Escuela Politécnica Superior
Departamento de Ingeniería Informática
Universidad Autónoma de Madrid

Propuesta de una Metodología de Aplicación de Técnicas de Descubrimiento del Conocimiento para la Ayuda al Estudiante en Entornos de Enseñanza Superior

TESIS DOCTORAL

César Vialardi Sacin

Madrid, setiembre 2010

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

TESIS DOCTORAL

Propuesta de una Metodología de Aplicación de Técnicas de Descubrimiento del Conocimiento para la Ayuda al Estudiante en Entornos de Enseñanza Superior

Autor:
César Vialardi Sacin

Tutor:
D. Álvaro Ortigosa

Madrid, setiembre 2010

Memoria presentada para optar por el grado de Doctor en Ingeniería informática

Título	**:**	Propuesta de una Metodología de Aplicación de Técnicas de Descubrimiento del Conocimiento para la Ayuda al Estudiante en Entornos de Enseñanza Superior
Autor	**:**	César Vialardi Sacin
Director	**:**	Álvaro Ortigosa
Dpto.	**:**	Ingeniería Informática

Miembros del Tribunal

Presidente :

Secretario :

Vocal 1 :

Vocal 2 :

Vocal 3 :

Fecha de Lectura :

Calificación :

Agradecimientos

Durante los cinco años de estudios doctorales, que incluyen mis estancias en Madrid y las jornadas en Lima, muchas personas han estado a mi lado ofreciéndome su amistad y apoyo. La culminación de esta investigación, luego de todo ese tiempo dedicado a un intenso trabajo, ha sido posible gracias a su ánimo constante y desinteresada colaboración. Aprovecho este espacio para dejar constancia de mi gratitud a todas ellas.

En primer lugar, quiero agradecer a la doctora Ilse Wisotzki, rectora de la Universidad de Lima, por haber confiado en mí desde que este proyecto se inició e incluso desde algunos años antes, cuando fue testigo de mis intentos por estudiar en España. Desde un principio me alentó y mantuvo intactas mis ganas de concretar esa gran oportunidad. Posteriormente, la doctora Wisotzki fue quien, a través de su confianza y respaldo, hizo viable el proyecto. Su auspicio y sus atinados consejos fueron y siguen siendo invalorables.

Quiero expresar mi especial agradecimiento a Pilar Rodríguez por ese espíritu tan generoso con el que me trató desde el primer día en que fui a la Escuela Politécnica Superior, y porque sin su ayuda, sus magníficas ideas y sus soluciones tan simples no habría sido posible siquiera terminar los estudios del primer año del doctorado. Asimismo, le estaré eternamente agradecido por permitirme trabajar en su despacho durante el invierno del 2009. No exagero al decir que el B326 se convirtió en mi hogar a lo largo de los dos primeros meses de aquel año.

Quiero dar las gracias a mi director Álvaro Ortigosa por el tiempo que me dedicó durante las tres estancias en la UAM, que, aunque breves, fueron intensas e intelectualmente fecundas. Su presencia resultó decisiva tanto para la maduración del proyecto como para su culminación, y con ello me refiero a aspectos netamente académicos y también a aquellos que la distancia muchas veces me ha impedido cumplir por mí mismo. Contar con sus buena voluntad, sus vastos conocimientos, su extraordinaria intuición, pero sobre todo con su solidaridad, ha sido para mí un verdadero privilegio. Por si ello fuera poco, su preocupación por mí en los momentos más difíciles reveló, incluso por sobre su rigor y profesionalismo, su inapreciable amistad.

También quiero agradecer a Eduardo Pérez por sus innumerables lecturas de la totalidad del texto y por sus numerosas sugerencias constructivas. He perdido la cuenta de las veces en que lo he llamado telefónicamente desde Lima para discutir pormenores sobre los intrincados problemas que planteó esta investigación. En todas ellas, siempre encontré, además de perspicaces comentarios, la mejor disposición del otro lado del teléfono.

No quiero dejar de mencionar a mis profesores del período docente, Rosa, Fernando y Pablo, cuyas enseñanzas supieron compartir tan generosamente conmigo.

Igual mención merece el personal administrativo, que con suma gentileza me ayudó siempre que lo requerí. Gracias especialmente a Juana por toda la amabilidad y sobre todo amistad. El personal de la Biblioteca, Marisol, Elisa y María, me sacó de apuros cuando tuve que lidiar con el acopio de la bibliografía, fundamental para profundizar en los temas de la investigación. Nada habría logrado sin su diligencia y buen ánimo, que las llevaron a hacer mucho más de lo que estaba a su alcance. Gracias a todas ellas.

Tampoco puedo dejar de agradecer a mis compañeros de primer año, Myriam y Mari Carmen, a quienes recuerdo con mucho cariño; a Pablo Martin, quien me recibió el 5 de noviembre del 2005, primer día de mi estancia en Madrid; y a Javi, con quien escribí la mayor parte de mis artículos (ambos sabemos que empezamos de descubrir la dirección de nuestros trabajos en esas largas reuniones en el B326).

Otras personas a los que me gustaría dedicar unas palabras de reconocimiento por su colaboración, que a la postre fue decisiva, son Jorge, Jhonny, Juan Pablo, Gustavo, Bruno y Daniel del instituto de investigación científica de la Universidad de Lima. Con ellos tuve la oportunidad de compartir los últimos resultados de esta investigación. Gracias por sus reflexiones, que nutrieron las mías.

Quiero hacer una mención especial a Fernando Iriarte, a quien agradezco por todas las veces que ha leído las distintas versiones de este documento, donde ha gastado no poca tinta roja. Consta mi agradecimiento por las ganas y, sobre todo, por el cariño con que ha hecho esa labor.

Por último, pero con igual consideración, debo mencionar la silenciosa labor de Jessica, mi hermana, quien supo estar a mi lado en cada momento para solucionar todos esos pequeños detalles que, si no son atendidos a tiempo, vuelven imposible la culminación de una tesis. Es por ella, por su entrega e infinita paciencia, que estas páginas alcanzaron su forma definitiva.

En este largo camino, hubo, además de amigos y compañeros, una serie de instituciones cuyos valiosos aportes posibilitaron este trabajo. Agradezco a la Universidad de Lima, institución donde laboro, por todas las facilidades y los beneficios recibidos. A la Fundación Carolina, por otorgarme la condición de becario para los estudios doctorales que inicié el año 2005 y culmino con la presentación de esta memoria. Y al proyecto HADA (TIN2007-64718), por hacer factible la difusión de gran parte de esta investigación.

Resumen

Uno de los principales problemas para el estudiante de una universidad es la toma de decisiones adecuadas con respecto a su itinerario académico, es decir, en qué asignaturas matricularse: de ellas dependen, en gran medida, su futuro universitario y en muchos casos profesional. Aun si el plan de estudios restringe la optatividad, el estudiante suele tener cierta libertad para elegir cuándo cursar ciertas asignaturas, o en cuántas asignaturas matricularse por período académico. Sin embargo, hoy por hoy, el alumno muchas veces carece del apoyo necesario para orientarse entre las diferentes variables en juego que se presentan ante de una elección de esta naturaleza. Algunas universidades ofrecen la posibilidad de que los estudiantes soliciten la asesoría de un docente (quien tiene la experiencia de matrículas anteriores). Sin embargo, en la mayoría de las universidades el uso de tecnología ha llevado a efectuar los proceso de matrícula a través de Internet. De este modo, en gran parte de los casos el estudiante realiza la matrícula individualmente, por lo que el proceso queda a expensas de su experiencia y de la información disponible.

Lamentablemente, la experiencia demuestra que ni la una ni la otra resultan suficientes. El alumno promedio no relaciona el tiempo, el esfuerzo y su nivel intelectual con los requisitos mínimos implícitos dentro de una asignatura o una combinación de ellas, a fin de culminarla con éxito. El criterio elegido por él para tomar la decisión, en cambio, suele estar relacionado con la velocidad con la que quiere terminar sus estudios. Además, por el lado de la información, si bien es cierto que la Universidad en general proporciona toda la información cuantitativa necesaria (es decir, cursos hábiles, secciones, horarios, aulas y profesores), existe una información cualitativa que está implícita en las experiencias previas de los estudiantes (es decir, en las matrículas que se han efectuado anteriormente y en los resultados derivados de ellas) que no se aprovecha.

Por otro lado, las instituciones deben tender a preservar el conocimiento necesario para su funcionamiento fuera de personas individuales. En este caso particular, las universidades deben evitar que el conocimiento derivado de la experiencia de años de matrículas se pierda cuando un docente deja la institución. Para ello es necesario crear modelos que hagan explícito este conocimiento y permitan utilizarlo de forma sistemática.

La presente propuesta busca resolver esta problemática. En particular, intenta cubrir el vacío que se genera cuando el estudiante realiza la matrícula sin un adecuado asesoramiento. La línea más importante de esta investigación es, por tanto, la adquisición de conocimiento a partir del rendimiento académico de un grupo de estudiantes, con la finalidad principal de que nuevos estudiantes que vayan a cursar las mismas asignaturas puedan usar las experiencias anteriores con miras a obtener

mejores resultados en su itinerario académico. En otras palabras, nuestro primer objetivo es facilitar una adecuada visión del comportamiento y rendimiento del grupo de estudiantes en una determinada carrera universitaria y, al mismo tiempo, proporcionar recomendaciones que aumenten la efectividad y la pertinencia de las decisiones sobre las asignaturas en que los alumnos busquen matricularse.

Para ello se propone una metodología a través de la cual se pueden desarrollar modelos que sistematicen el conocimiento implícito en los datos de anteriores matrículas. A partir de este modelo, se pueden implementar herramientas, idealmente integradas a sistemas de matrícula vía Web actualmente existentes, cuya función será predecir la conveniencia de cursar una asignatura específica para un estudiante determinado, sobre la base de los resultados obtenidos por estudiantes con rendimientos académicos similares que hayan cursado antes dicha asignatura. Este sistema automático de asesorías, similar a un sistema de recomendación colaborativo, usa un motor de recomendaciones basado en técnicas de minería de datos.

Los sistemas de recomendación colaborativos son agentes que sugieren opciones para que el usuario pueda elegir entre ellas. Están basados en la idea de que individuos con los mismos gustos suelen seleccionar o preferir lo mismo. Son altamente aceptados y ofrecen buenos resultados para una gran cantidad de aplicaciones. En el ámbito de la educación, particularmente, estos sistemas, de manera inteligente, tratan de sugerir acciones a los estudiantes a partir de otras anteriores de individuos con las mismas características, ya sea académicas, demográficas o personales [Vial-09b],[Zaia-02]. En el presente trabajo, se propone una metodología basada en preparación de datos y en minería de datos, como ya se mencionó, con la finalidad de ofrecer a los estudiantes elementos clave para que las decisiones relativas al itinerario académico se realicen sobre la base del rendimiento académico de alumnos con un perfil similar, lo que optimizaría su rendimiento académico

La investigación se ha concentrado en la experimentación usando datos reales, particularmente la base de matrículas de la Facultad de Ingeniería de Sistemas de la Universidad de Lima, acumuladas desde su creación (año 1991) hasta el presente. De esta manera, la propuesta metodológica, que entre otros aspectos considera la utilización de atributos específicos para este dominio de aplicación, está fundamentado en una rigurosa experimentación que proporciona resultados que comprueban la utilidad y eficacia de los sistemas de filtrado colaborativo clásicos basados en memoria, de la incorporación de los atributos sintéticos de potencial y dificultad, y de la manipulación del dominio de aplicación con la finalidad de crear conjuntos de clasificadores cuya eficiencia con respecto a los clásicos se comprueba estadísticamente. Por último, el objetivo final es proporcionar una herramienta basada en una técnica eficiente que permita brindar una mejor recomendación a cada usuario.

Abstract

One of the main problems faced by university students is making appropriated decisions related to their learning paths (subjects in which to enroll every semesters or year): their academic -and in many cases their professional- future depends on said decisions. Even if options in the study plan are restricted, students have some freedom to choose certain courses, or the number of courses for the academic term. Currently, students do not have the necessary support to get organized and decide among the different existing variables when facing this type of selection. Some universities offer the possibility of receiving advice from a professor (with experience from previous enrolment processes). But in most universities the enrolment process is done through Internet. And thus students mostly enroll individually, and the process depends on his experience and the available information.

Unfortunately, this experience is insufficient, as average students do not link time, effort and intellectual level with the implicit minimal requirements for the subject or a combination of them, in order to successfully culminate their studies. Instead, the criteria chosen by the student in order to make a decision is generally related with the speed to finish the studies. Besides, in relation to information, if it is true that the University provides the required quantitative information (this is subjects available, sections, schedules, classrooms and professors), there is qualitative information implicit in students previous experiences (from previous enrollments and their outcomes) that is not being used.

On the other hand, institutions should preserve the required knowledge in order to be used not only by some individuals. In this particular case, universities should prevent that knowledge resulting from enrolment of previous years will be lost when the professors leave the university. In order to achieve this, it is necessary to create models that will make explicit this knowledge and allow using it in a systematic way.

Our study tries to fill this gap - generated when the student does not require advice- proposing a methodology that will finally allow developing a tool integrated to the current enrolment system through the web. In this way, the objective of the system is to predict the convenience of taking a specific subject for a certain student, based on results from students with similar academic performances that had taken said subjects. The most important line of this study is the acquisition of knowledge from students academic performance, the main purpose is that new students taking the same subjects will be able to use previous experiences, in order to obtain better results in their learning paths. This means, it facilitates an appropriated vision of behavior and performance of the group of students at certain university career and, at the same time, gives recommendations for

the student in order to increase his/her effectiveness and pertinence when making decisions in relation to subjects to enroll in.

In order to achieve said purpose, this current research proposes a methodology through which we can develop models that systematize the knowledge implicit in the data of previous enrolments. From this model we can implement tools integrated to enrolment systems through current web sites; the function of these models will be predicting the convenience for a specific student of enrolling in a certain course, based on results obtained by students with similar academic performance that have previously taken said course. This automatic advice system, similar to a collaborative recommendation system, uses a recommendation engine based in data mining techniques.

Collaborative recommendation systems are agents suggesting options for the users to choose among them. They are based on the idea that individuals with the same tastes usually choose or prefer the same things. These systems are highly accepted and offer good results for a great amount of applications. Particularly, in the field of education, a recommendation system is an agent trying, in an intelligent manner, to suggest actions to students from previous actions of other students with the same characteristics, whether academic, demographic or personal [Vial-09b], [Zaia-02]. In this current work, we propose a methodology based on data preparation and data mining -as it has already been mentioned- in order to offer students key elements so that decision, based in the academic performance of students with similar profile, will allow optimizing their academic performance.

The research has focused in the experimentation, using real data, mainly from the enrolment data base at Faculty of Computer Science at Universidad de Lima, accrued since its creation (1991) to the date. In this way the proposed methodology -which among other aspects takes into consideration the use of specific attributes for this domain of application- is based on an exhaustive experimentation that offers results confirming the usefulness and efficiency of the classic collaborative filtering systems based on memory, the inclusion of synthetic attributes of potential and difficulty, the manipulation of the domain of application in order to create set of classifiers whose efficiency has statistically been verified. Finally, the main objective is to contribute with a tool based on an efficient technique that will allow offering a better recommendation to each user.

Índice general

CAPÍTULO 7: CONCLUSIONES Y TRABAJO FUTURO

APÉNDICE A

REFERENCIAS BIBLIOGRÁFICAS

LISTA DE ABREVIATURAS

Índice de Figuras

Índice de Tablas

Propuesta de una Metodología de Aplicación de Técnicas de Descubrimiento del Conocimiento para la Ayuda al Estudiante en Entornos de Enseñanza Superior

Capítulo 1:
Introducción

1.1. Problema

En los últimos tiempos, el desarrollo de la World Wide Web ha provocado un incremento en el número de sus aplicaciones para la difusión de la información y en el acceso a diversos servicios. A partir de este desarrollo tecnológico, se produce un cambio en la manera como interviene el elemento humano en una serie de procesos de diversa índole en los que anteriormente era requisito indispensable su presencia física. Como consecuencia de este hecho, disminuye la frecuencia en el empleo de una serie de recursos humanos y materiales tales como infraestructura y personal administrativo, antes indispensables. Estos procesos interactivos, llevados a cabo desde cualquier lugar y con una eficacia igual o mayor, ahorran significativamente tiempo y espacio.

Las instituciones de educación superior han orientado sus procesos administrativos y académicos en esta línea [Vial-09a]. Es así como, actualmente, se llevan a cabo gestiones de diversa naturaleza a través de Internet. En particular, la búsqueda de documentos, registros de datos y el uso de sistemas de e-Learning, sistemas cuyo objetivo principal es proporcionar al usuario una mejor experiencia, brindándole una herramienta que le posibilite cumplir con su cometido de una manera simple, ágil y segura; igualmente, le permitirá a la institución lograr dinamismo y fluidez en sus procedimientos.

Estos sistemas ofrecen mejor interacción y mayor flexibilidad, debido a que pueden ser utilizados desde una gran variedad de dispositivos y distintas localizaciones y, algunos de ellos, incluso cuando el usuario lo desee. Sin embargo, como contraparte, la ausencia del ser humano en la mediación del proceso puede generar cierta inestabilidad e incertidumbre ante el desconocimiento de las acciones necesarias para lograr un manejo adecuado. Es fundamental, por tanto, contar con una herramienta que sirva de soporte y análisis.

En el contexto de todos los procesos involucrados en la educación universitaria, la matrícula es un procedimiento fundamental que los estudiantes deben realizar periódicamente. Conviene detenerse un momento en él, pues la presente investigación se concentra en su mejora y, específicamente, en su automatización a través de uso de técnicas del campo de la Inteligencia Artificial, y más específicamente técnicas de adquisición automática del conocimiento. Para ello se harán algunas suposiciones que no implicarán pérdida de generalidad, pero que permitirán establecer el marco de trabajo de la presente propuesta.

Un estudiante universitario debe matricularse cada período académico en las asignaturas correspondientes al plan curricular de la titulación que ha elegido. En la mayoría de los casos, tendrá algunas opciones disponibles; dependiendo del plan en cuestión, podrá tener libertad para elegir entre asignaturas optativas, o al menos decidir qué asignatura quiere tomar en este período académico y cuáles dejar para períodos subsiguientes. En tal situación, debe sopesar la dificultad de las asignaturas, sus requisitos, los horarios, el número de créditos, el promedio ponderado y el tiempo que le dedicará al estudio, a fin de establecer sus posibilidades de aprobación. Sin embargo, su decisión depende única y exclusivamente de la información que pueda conseguir por su propia cuenta.

Con el objetivo de tomar estas decisiones con la mayor información posible, las universidades pueden poner a disposición de los estudiantes diversas fuentes de información. Por ejemplo, en las universidades que ofrecen más recursos a los estudiantes, estos pueden asistir a sesiones de asesoramiento a cargo de tutores, es decir, profesores encargados de realizar la tarea compleja de evaluar la situación académica de cada alumno. Junto con ellos, luego de evaluar una serie de factores (por ejemplo, la disponibilidad de tiempo del estudiante, la facilidad o disposición individual que presente ante determinada área de estudio y los antecedentes de su desempeño académico), cada alumno podrá determinar cuál es el conjunto de asignaturas más conveniente para matricularse con vistas a obtener un buen rendimiento académico.

Pese a los esfuerzos que el alumno pueda realizar individualmente, incluso en el caso de que la institución lo provea de un asesor para mejorar el proceso, la información acopiada resulta parcial e insuficiente. Efectivamente, un análisis más preciso de las implicancias de esta tarea permite constatar la complejidad que conlleva aunar información útil para una toma de decisión racional en un contexto como este. Asimismo, permite observar en qué medida un esfuerzo individual no consigue una productividad efectiva en la provisión de información[1] previa al establecimiento de un itinerario académico, como veremos a continuación.

En el caso de que el alumno pudiera contar con un asesor, este debería conocer su situación académica al detalle. En este sentido, lograr una visión objetiva sobre el desempeño general del estudiante es una tarea que se complicaría por la intervención de múltiples variables: el número y el tipo de asignaturas que el estudiante pretende cursar en el nuevo período académico y si estas presentan requisitos; el desempeño del estudiante en cada asignatura específica cursada hasta el momento de la asesoría; o bien los resultados obtenidos por el alumno en todos los requisitos de una asignatura en

[1] La naturaleza de esta información es un factor medular del problema, al punto que podría decirse que, en cierto sentido, nuestro trabajo enfrenta la tarea de transformarla en información completa, objetiva, y transferible, es decir, cualitativamente distinta.

particular, en los dos requisitos inmediatos anteriores o solamente en el inmediato anterior.

Asimismo, el asesor tiene la tarea de investigar si la opción de matrícula elegida en el período anterior tuvo resultados positivos para el estudiante y, más aún, para el grupo en general, así como conocer de qué manera podría mejorar. En otras palabras, debería establecer un seguimiento de las recomendaciones para estimar su efectividad.

Por último, para que las recomendaciones sean válidas para cualquier estudiante, el asesor debe establecer un universo de datos amplio, pues, de lo contrario, las generalizaciones pueden resultar tendenciosas. Todo ello, no es necesario hacer mucho esfuerzo para notarlo, escapa a las posibilidades de un individuo, es decir, sale del campo de lo subjetivo e ingresa en el del trabajo automatizado.

Esta complejidad provoca que, incluso para asesores experimentados, no sea fácil realizar recomendaciones que atiendan las particularidades de cada estudiante. Además, desde el punto de vista de la organización, no es conveniente que el conocimiento necesario para realizar este asesoramiento resida sólo en los individuos.

Finalmente, hay que hacer una consideración adicional, relacionada con las nuevas posibilidades tecnológicas que se mencionaron más arriba: el uso de herramientas que permiten realizar la matrícula a través de Internet favorece el que los estudiantes tomen las decisiones sin asistencia de personal de la universidad y, en particular, sin consultar con los correspondientes asesores.

En el caso de que el alumno no cuente con una asesoría formal, la situación es básicamente la misma, aunque, dado que las estimaciones las tiene que realizar una sola persona, acarrearían una dificultad aun mayor. Asimismo, esta generalización es válida para los casos en los que el alumno decide voluntariamente pedir una recomendación a una persona en cuyo criterio confía (un compañero que ha llevado el curso y ha obtenido buena nota, por ejemplo). En todas esas situaciones, las fuentes de información (el asesor, compañeros de universidad, el propio estudiante que realiza una búsqueda personal) enfrentan los mismos problemas de insuficiencia cuando pretenden que la recomendación generada sea más productiva.

En síntesis, son muchos los obstáculos que encara aquel que busca obtener una recomendación objetiva y verdaderamente efectiva (del asesor, en caso de contar con uno, o de quien fuere): el elevado número de variables en juego; la imposibilidad de hacer un seguimiento personal del desempeño de cada alumno que llevó un curso por el cual está interesado el consultante; y el volumen de información que implica estimar la tendencia del rendimiento académico en un curso o en una serie de ellos.

Todo eso puede observarse desde el punto de vista del producto de la recomendación, es decir, a través de la información. Si centramos nuestra atención no tanto en los sujetos involucrados en la matrícula, sino en aquello que el alumno busca conseguir antes de la matrícula (información que responda a la pregunta: ¿será

beneficioso para mí que lleve estos cursos en este semestre?), podremos notar con más claridad el problema que enfrenta nuestro sistema. Efectivamente, cuando un alumno emprende la búsqueda solitaria de algunas certezas que lo ayuden a tomar una decisión en su matrícula, va a obtener información subjetiva (por ejemplo, algunos individuos le darán sus pareceres sobre la dificultad de un curso) y por tanto parcial y escasa, información que, si resulta útil, no será almacenada para aplicaciones posteriores. La naturaleza de esa información es, precisamente el obstáculo principal. En ese sentido, "¿cómo podemos brindarle al alumno información que sea diametralmente opuesta a la mencionada, es decir, que sea objetiva, completa y acumulable?" es la pregunta que nos anima.

A esos problemas de índole teórica y metodológica hay que sumarles los problemas prácticos derivados de ellos. La consecuencia directa de carecer de un apoyo en la matrícula es la desorientación de los estudiantes[2]. Esto se refleja en el hecho de que no son pocos los estudiantes que consideran que su bajo rendimiento se debe a una mala elección de los cursos en la matrícula, o bien porque se han matriculado en muchas asignaturas, o bien porque no han estado preparados para estas. Si se lograra enmendar el problema, podría potenciarse el nivel académico de los alumnos, lo que es, sin duda, uno de los principales objetivos de la universidad.

Ante tal situación problemática, se plantea la necesidad de un sistema de gestión universitaria académico-administrativa, en particular, la propuesta de una metodología que ayude a la creación de una herramienta predictiva capaz de responder la siguiente pregunta: ¿es posible proponer un sistema de recomendación que genere una predicción sobre el rendimiento académico de los estudiantes en ciertas asignaturas, predicción con la cual puedan orientarse en su proceso de matrícula? En ese sentido, se espera ampliar la información involucrada a través de la introducción y análisis de los datos pertenecientes al perfil académico del estudiante y del procesamiento de los datos preexistentes de cada una de las asignaturas. A partir de ello, el sistema podrá brindar al alumno información nueva y precisa que le permitirá orientarse de manera acertada al momento de elegir las materias en las que deberá matricularse.

Cabe señalar que el sistema de predicciones debe ser interpretado como tal, es decir, como una herramienta que presenta de manera estructurada un comportamiento muestral específico bajo circunstancias similares a la situación académica actual del estudiante. No debe entenderse de ninguna manera como un sistema que determine rígidamente las acciones humanas. Este sistema no considera toda una gama de información de carácter personal, no explícita en el registro académico del estudiante y

[2] Por el número de estudiantes en las universidades, la presencia física o virtual de un asesor se vuelve imposible. Sumado a ello, hay que considerar que la matrícula por Internet, pese a la ventaja que supone su rapidez y ubicuidad, ha agudizado el problema, pues ha alejado aún más al estudiante de las condiciones necesarias para enfrentar una elección racional de cursos.

que, bajo condiciones particulares, podría contribuir a la orientación del asesor y a la elección del estudiante. Así, como el sistema de recomendación no admite la consideración de otros factores que ocasionalmente podrían originar un cambio en la recomendación, los estudiantes deben considerar los resultados arrojados por el sistema como un punto de referencia entre otros.

Ahora bien, es posible asumir sin pérdida de generalidad que la información de registro, contenida en los sistemas de matrícula, proviene de tres fuentes. La primera está constituida por la información de registro que la institución obtiene al momento del ingreso del estudiante. La segunda la compone la información que se recibe cuando el alumno interactúa con el sistema; por ejemplo, cuando decide matricularse en tal o cual asignatura. La tercera está conformada por los datos que el sistema obtiene de los departamentos académicos, que pueden ser de dos tipos: la información creada por la institución, (por ejemplo, los planes curriculares y la carga de trabajo créditos y horas); y la información que se origina a partir de la interacción del estudiante con el sistema, es decir, a través de la introducción de sus resultados obtenidos en las diversas evaluaciones y de sus promedios por asignatura y por período académico.

En la práctica, los sistemas de gestión académico-administrativa almacenan toda esta información en bases de datos relacionales. Aunque todos los datos necesarios están disponibles en ellas, su tamaño es, generalmente, demasiado grande para permitir un aprovechamiento directo de la información contenida. Es por este motivo que ha sido necesario utilizar las técnicas de descubrimiento del conocimiento, cuyo principal proceso es la minería de datos. Esta es un componente dentro de este proceso, que consiste en analizar determinada información y aplicar los algoritmos apropiados capaces de producir y extraer información implícita contenida en los datos, previamente desconocida y potencialmente útil [Witt-00]. El reto de la minería de datos es, precisamente, trabajar con grandes volúmenes de información procedentes de sistemas que, por otro lado, pueden contener sus propios problemas (ruido, datos faltantes, volatilidad, etcétera).

La minería de datos es un campo multidisciplinario que involucra técnicas como el aprendizaje automático, el reconocimiento de patrones, la estadística y las bases de datos. Se lo considera como una evolución natural iniciada con la creación de las primeras bases de datos y que continuó con el Lenguaje Estructurado de Consultas (SQL, por sus siglas en inglés) y, con un mayor impacto, con el Online Analytical Processing (OLAP). Lo que pretende la minería de datos es automatizar el proceso, localizando y extrayendo patrones ocultos. En su forma más pura, no busca un tipo específico de información, sino que, simplemente, busca patrones que se encuentran en los datos.

1.2. Motivación

Los programas de estudio para obtener grados universitarios suelen tener una naturaleza flexible. En muchos casos, están compuestos de asignaturas obligatorias y optativas; aunque la optatividad sea baja, el estudiante puede tener la opción de elegir en qué semestre académico cursar cada asignatura, respetando ciertas restricciones. Antes de que cada semestre inicie, el estudiante debe matricularse en las pertenecientes al período que le corresponda o, en caso de que presenten requisitos, en aquellas que el propio plan habilite, lo que depende de su estructura curricular[3].

Dicho proceso puede ser enfocado desde tres puntos de vista. En primer lugar, está el aspecto administrativo. Forman parte de este nivel todos los procedimientos burocráticos que el alumno debe realizar para considerarse matriculado: etapas de la matrícula (inscripciones, pagos, presentación de documentos, asesorías), agentes de la matrícula, entre otros. Asimismo este proceso puede presentarse de múltiples maneras. Por ejemplo, en algunas instituciones, el estudiante, según su desempeño académico, goza de cierta prioridad que determina el orden en que se realizará su matrícula. Esa es la forma en la que enfrentan las limitaciones físicas que impone el número de aulas y profesores. Así, a medida que los primeros alumnos se inscriben en los cursos, los siguientes cuentan con menos posibilidades de elegir las secciones y los horarios de su preferencia.

De este primer aspecto de la matrícula se deriva el segundo, el aspecto técnico, es decir, los soportes y herramientas mediante los que se realiza la matrícula. Por ejemplo, si esta se desarrollase por medio de un soporte virtual, la página web habilitada podría presentar de manera clara el procedimiento que el alumno debe seguir paso a paso para efectuar su matrícula sin dificultades. En caso de que sea requerido por el estudiante, también podría ofrecer alternativas de apoyo técnico personalizado.

El tercer aspecto, que podríamos llamar “individual”, se encuentra alejado, en la actualidad, del ámbito técnico-administrativo. Se vincula, más bien, con una situación de carácter meramente personal: la selección, por parte del estudiante, de la asignatura o asignaturas en las que matricularse en un semestre académico.

Este último aspecto, la elección de matricularse en determinada asignatura para la cual se encuentre facultado depende de la decisión que tome el alumno, por un lado, a partir de la información que haya logrado agenciarse (por ejemplo, el grado de dificultad de un curso), y por otro lado, de sus necesidades personales (por ejemplo, culminar la carrera en cinco años). Para optimizar el uso de la información disponible, en algunas

[3] La sucesión de matrículas que efectúe el estudiante a lo largo de su carrera se denomina, en la presente investigación, “itinerario académico del alumno”.

instituciones educativas, el estudiante puede solicitar la asesoría de un docente, quien cuenta con experiencia obtenida a través del asesoramiento de alumnos en matrículas anteriores y con la información que le proporcione el estudiante acerca de su historial académico. Esto permite al docente conocer la situación actual del educando y orientarlo para que pueda decidir de la manera más acertada posible en cuántas y cuáles asignaturas hábiles debería matricularse. Sin embargo, este servicio no está generalizado y, cuando se encuentra disponible, no se solicita con mucha frecuencia, por lo que la matrícula, en general, depende de la información que el estudiante tenga sobre su propio desempeño, es decir, de su propia experiencia.

Las condiciones que motivan la presente investigación, en ese contexto, pueden resumirse en tres presunciones iniciales. Primero, la mayoría de estudiantes carece de dicha experiencia o esta le resulta insuficiente para decidir adecuadamente en el momento de la matrícula, debido a que no relaciona una serie de factores (como son el tiempo, su afinidad por el campo de estudio de la asignatura, el esfuerzo que se haya propuesto realizar y sus habilidades o aptitudes) con los requisitos mínimos que debe cumplir para culminar con éxito una asignatura o una combinación de ellas. Segundo, muchas veces el criterio elegido para tomar la decisión, dada la poca madurez del alumno, está íntimamente relacionado con la premura con la que quiere terminar sus estudios. Por último, por el lado del conocimiento de los datos relativos a la matrícula, si bien es cierto que una universidad proporciona toda la información cuantitativa necesaria (cursos hábiles, secciones, horarios, aulas y profesores), existe información que podría ser útil, pero que en la práctica es muy difícil que pueda ser aprovechada por los estudiantes o por los encargados de aconsejarlos.

Esta información de tipo cualitativo constituye el punto de partida de la presente propuesta. Así, el conjunto de información previa disponible, junto con los datos del historial académico de cada estudiante, puede resultar provechoso si se utiliza para construir un modelo predictivo. A partir de este modelo, es posible obtener una recomendación basada en hechos reales y pertinentemente orientada. Al respecto, debe tomarse en cuenta que, incluso si un profesor con experiencia asesora a los estudiantes, le resultaría muy difícil incorporar nuevas tendencias en los estudiantes (por ejemplo, las relacionadas con el rendimiento académico) y en las asignaturas (por ejemplo, la dificultad y los requisitos). En cambio, un análisis automatizado de los registros sí puede detectar tales tendencias o cambios aun cuando el volumen de información sea muy grande.

Así, en el presente trabajo se propone utilizar este tipo de información para construir sistemas de recomendación colaborativos. Estos sistemas son agentes que sugieren opciones [Badr-01] para que el usuario pueda elegir entre ellas. Uno de sus fundamentos es la idea de que individuos con los mismos gustos o preferencias (que se manifiestan a través de compras, ventas, decisiones de todo tipo que configuran caminos

e itinerarios) suelen seleccionar lo mismo. En la actualidad, son altamente aceptados y ofrecen buenos resultados para una gran cantidad de aplicaciones [Adom-05].

En un contexto donde existe un evidente vacío relativo a la disposición de información sobre matrículas y rendimiento académico de años anteriores, la puesta en ejecución de la presente metodología es favorable, pues convierte un aspecto del proceso de matrícula basado, hasta hoy, en análisis mayormente subjetivos, en uno estadístico matemático. Esto lo ubicará dentro del aspecto técnico del proceso de matrícula y, por tanto, lo volverá más objetivo y preciso, lo cual se traducirá en la elección de asignaturas de una manera más certera, y, en consecuencia, en el aumento del porcentaje de éxito alcanzado en cada uno de los grupos de usuarios.

1.3. Objetivos y propuesta

En el ámbito de la educación, un sistema de recomendación es un agente que, de manera inteligente, trata de sugerir acciones a los estudiantes a partir de experiencias anteriores de usuarios con las mismas características, ya sean académicas, demográficas o personales [Vial-09b], [Zaia-02]. Como se deduce de la sección anterior, dentro del marco en el que se desarrolla la presente investigación, lo más destacado del proceso de matrícula no reside en la matrícula misma, sino en la decisión previa del alumno, decisión referida a la elección de las asignaturas que cursará en el período académico en el que se está matriculando. El problema que trata de resolver este trabajo de investigación está centrado en formular una correcta recomendación basada en referencias del rendimiento académico obtenido por otros estudiantes como él, que han cursado las asignaturas en períodos anteriores, para contar con un elemento adicional de información con el que pueda efectuar su matrícula de manera más conveniente.

El problema planteado se resuelve en dos fases. En primer lugar, se propone un método de preparación de datos para el ámbito de la educación superior [Vial-09a], que se revisará en el capítulo correspondiente a la metodología de preparación de datos. Este consiste en generarle atributos nuevos a la base de datos con el objetivo de mejorar la representación de la realidad estudiada. Esta realidad a la que se hace mención es propia de dominio de la educación y consiste en considerar que el análisis de datos que realiza el sistema de recomendación no sólo depende de los atributos originales, sino también de datos externos propios del sistema, como son: a) los promedios de las asignaturas relacionadas con la asignatura en cuestión, b) la dificultad de las asignaturas y otros factores que se pueden ir generando a partir de la información de la base de datos y que sustenten un acercamiento de los datos a las reglas del sistema de matrícula y a las particularidades del dominio (por ejemplo, las dependencias de cada

asignatura con otras, los cambios de planes de estudio, la eliminación y creación de asignaturas).

En segundo lugar, se propone un nuevo método para el aprendizaje automático, cuyo fundamento son las técnicas basadas en remuestreo de datos de entrenamiento

. Específicamente, se propone una variante del método de Baggin. Este nuevo método lo aplica el sistema de recomendación usando los datos manipulados en la fase previa, a fin de poder entregar recomendaciones más precisas al estudiante.

En general, los sistemas de recomendación más extendidos en el mercado son los de filtrado colaborativo. Estos trabajan en base al cálculo de una medida llamada "similitud". Con esta, generan una predicción numérica correspondiente a los alumnos que han cursado las asignaturas; así, el sistema estaría apto para recomendar al estudiante si le conviene o no cursarlas.

El problema de aplicar técnicas tradicionales de filtrado colaborativo usadas comercialmente reside en el hecho de que estas calculan la predicción en base a una medida de similitud, lo que se complica a medida que aparecen más atributos o características de los usuarios o artículos (ítems). En los sistemas de comercio vía Web, estas trabajan sin inconvenientes ni restricciones, debido a que el usuario asigna puntajes a los artículos que desea. Si bien es cierto que este tipo de sistema tiene el riesgo de que el consumidor se encuentre con pocos artículos puntuados, tiene la ventaja, debido a las características del propio dominio, de requerir un solo atributo. Por ello, se facilita el cálculo de la similitud y la consecuente predicción.

Sin embargo, en un dominio de aplicación más complejo, como es el que nos ocupa, el estudiante obtiene una calificación que es un número (que corresponde a cierto intervalo de valores) que indica que puede aprobar o desaprobar una asignatura. Es entonces que se presentan dos diferencias significativas con respecto al dominio de aplicación tradicional de los sistemas de filtrado colaborativo. En primer lugar, en este dominio de aplicación las puntuaciones son calificaciones y, lo que es más importante, se requiere más de una variable, lo que eventualmente permitiría el cálculo directo de la predicción, pues hacen falta más atributos que deben ser considerados; por ejemplo, el promedio, los créditos cursados, la dificultad de la asignatura, etc. Por otro lado, se observa que, para que el modelo se asemeje a la realidad, es necesario considerar, como se mencionó previamente, otros atributos, lo que vuelve al método tradicional de filtrado colaborativo, hasta cierto punto, inaplicable.

El objetivo es que el sistema de filtrado colaborativo pueda dar recomendaciones efectivas, que maneje las mismas variables que un tutor consideraría en un sistema presencial. Para ello se propone poner en práctica, en lugar del cálculo de las similitudes, el análisis de datos mediante el uso de técnicas de minería de datos.

En el presente trabajo de investigación, se propone el sistema SPRS (Student Performance Recommedation System), un sistema de recomendación basado en

técnicas de minería de datos cuyo objetivo es generar un modelo, dependiendo de la técnica, fundamentado en la información histórica. Su finalidad principal es ofrecer a los alumnos elementos básicos para que la consulta o solicitud de recomendación se realice a partir del rendimiento académico de estudiantes con un perfil similar, y así permita optimizar su desempeño académico.

De esta manera, integrando una herramienta a los sistemas vía Web actualmente existentes, se busca, entre otras cosas, cubrir el vacío que se genera cuando el estudiante no solicita una asesoría. Esta herramienta ofrecerá al estudiante una facilidad con la que pueda, mediante una consulta muy simple, visualizar una recomendación basada en la experiencia de estudiantes con similar rendimiento académico.

Para realizar este trabajo, se han utilizado los datos pertenecientes a las matrículas de la Facultad de Ingeniería de Sistemas de la Universidad de Lima desde su creación en el año 1992. Dicha base está compuesta por datos demográficos de cada estudiante, las matrículas en las asignaturas, las notas que obtuvieron en ellas, la cantidad de asignaturas que cursaron en cada semestre académico, sus promedios ponderados y acumulados por semestre académico. Después de filtrarlos y limpiarlos, se aplicaron diversas técnicas de aprendizaje al sistema.

1.4. Contribución

El uso de técnicas de Minería de Datos en el campo de la educación es relativamente nuevo. La mayoría de trabajos están mayormente enfocados a la educación a distancia, vía Web o virtual [Rome-05]. Si tomamos como referencia el ámbito de la educación en general, el trabajo hecho en el ámbito de la predicción del rendimiento académico usando técnicas de Minería de Datos es incluso menor. Muchas de las aplicaciones usadas actualmente en los sistemas de recomendación son simples consultas a bases de datos [Schafer-05]. Algunos sistemas usan técnicas que van desde vecinos cercanos hasta análisis bayesiano. Los primeros sistemas de recomendación usan vecinos cercanos, basados en el cálculo de la distancia entre consumidores sobre la base de su histórico de preferencias [Schafer-01]. Las predicciones de cuánto le gustará un producto a determinado consumidor son calculadas tomando el promedio ponderado de las opiniones de un conjunto de vecinos. Algunas de las técnicas de minería de datos (DM) más comunes utilizadas en sistemas de recomendación son agrupamiento o clustering y reglas de asociación [Rome-07], [Vial-07].

No obstante esos antecedentes, aparte de los estudios mencionados, son pocos los que aplican técnicas de minería de datos en el ámbito de los sistemas de recomendación y ninguno con el fin específico de recomendar a los estudiantes sobre sus futuras matrículas.

Nuestro sistema de recomendación, que usa técnicas de minería de datos en el ámbito de la educación superior y cuyo diseño, implementación y evaluación es el objetivo central de este trabajo, supera los sistemas tradicionales, debido a que propone un espacio conceptual entre los estudiantes, sus preferencias y características académicas [Vial-09c].

Precisamente, este espacio es tomado por el sistema de recomendación, y es a partir de él que las técnicas empleadas construyen los modelos que servirán para predecir el rendimiento de un estudiante en el curso en que piensa matricularse.

Por ello, hemos cuidado que el método presentado en esta tesis conste de todos los pasos necesarios, con la finalidad de que una institución de educación superior pueda seguirlos y, a partir de ellos, obtener dos tipos de información: a) por el lado de la institución, este método permite encontrar patrones o tendencias ocultas a partir de los atributos de los estudiantes, con los que se podrá analizar su rendimiento en determinadas asignaturas a lo largo de los años [Brav-08a]; b) por el lado de los estudiantes, les permite predecir, con cierto grado de credibilidad, la calidad de su rendimiento en determinadas asignaturas o, dicho de otro modo, cuán bien rindieron estudiantes con características similares a las suyas en las asignaturas que consultan.

Las principales contribuciones que esta tesis propone a la comunidad científica se resumen a continuación:

1.4.1. Se estableció una metodología para aplicar recomendación colaborativa usando técnicas de minería de datos.

Inicialmente, se pensó resolver el problema aplicando las técnicas del filtrado colaborativo clásico. Debido a que los datos en este dominio de aplicación contienen más de un atributo y a que el cálculo de la similitud en estas condiciones no sería de aplicación práctica para la obtención de una adecuada predicción, se estimó la posibilidad de aplicar técnicas de minería de datos. Para ello, se ordenaron lo datos de manera distinta a la que exige el filtrado colaborativo clásico, con el objetivo de eliminar el problema de escases de datos y otros problemas asociados al dominio de aplicación. A fin de comparar los resultados, se aplicó el filtrado colaborativo clásico y se tuvo que hacer uso de una matriz reducida de usuarios y notas obtenidas por cada uno de ellos en las diferentes asignaturas, lo que trajo consigo una escases de datos muy grande, dado que no todos los estudiantes habían cursado todas las asignaturas[4]. Para la aplicación de las técnicas, los datos se tuvieron que ordenar de tal forma que un registro signifique el desempeño académico de un estudiante desde el momento de la matrícula en una asignatura determinada hasta la obtención de la nota final.

[4] Esta es una de las características más importantes de los datos del presente dominio de aplicación.

Bajo esta disposición de datos, es evidente que cada estudiante llegará a tener tantos registros o instancias como asignaturas curso, o bien en determinado período académico, o bien en total, considerando el histórico de la universidad (siempre que la base de datos abarque el lapso desde que el estudiante hizo su ingreso). Por otro lado, es evidente que en este arreglo de datos, los problemas de "campos vacios" no existen, debido a que sólo los datos que corresponden a los estudiantes que en algún momento obtuvieron sus calificaciones, se consideran como conjunto de entrenamiento.

1.4.2. Se usó clasificación supervisada para un sistema recomendador.

Como se mencionó en párrafos anteriores, generalmente los sistemas de recomendación están desarrollados en base a memoria (*memory based*), dada su facilidad de uso, pues la mayoría de estos controlan un solo atributo. En nuestro caso, la experimentación con filtrado colaborativo clásico, basado en memoria, no dio los resultados esperados, ya que no se pudo considerar atributos que, para los expertos en el dominio, eran de principal consideración. Por ese motivo, se propone la aplicación de técnicas de aprendizaje supervisado con una clase de dos valores.

1.4.3. Se diseñó una metodología de preparación de datos en este dominio de aplicación.

Los datos provenientes de las matrículas de estudiantes en el ámbito universitario no constituyen un terreno donde los investigadores especializados en técnicas de minería de datos hayan incursionado mucho. Son pocos los trabajos hechos en este campo: apenas hay intentos de aplicar filtrado colaborativo clásico basado en similitudes. Por otro lado, los datos que generan los estudiantes, los profesores y la universidad (mediante sus consejos de gobierno) son muy complicados y con una tendencia al cambio muy fuerte.

Se sabe que los estudiantes cursan asignaturas por un período determinado, los planes curriculares cambian, la mayoría de asignaturas presentan requisitos, los estudiantes no tiene un solo ritmo de trabajo y, por ende, un rendimiento uniforme. Todo ello lleva a pensar que, si se quiere hacer predicciones o hacer recomendaciones a un estudiante en la actualidad y, a la vez, utilizar información de períodos donde esta no tenga vigencia, la eficacia no será la adecuada.

Por este motivo, el presente trabajo propone el uso de una metodología de preparación de datos que considera todas las características dinámicas de este dominio, tales como la actualización de planes curriculares, las equivalencias de asignaturas, los cambios que se den en ellas y los requisitos de los estudiantes antes de cursarlas.

En relación con ello y como parte de la metodología propuesta, se incluye la generación de dos atributos sintéticos que describen el espacio conceptual de interrelación entre preferencias, obligaciones y características académicas globales de los estudiantes. La necesidad de incorporar dichos atributos sintéticos está basada en la falta de información que existe en los datos iniciales, ya que estos, sin la información de requisitos y dificultad de asignatura, dejarían de reflejar lo que ocurriría si la asesoría y su consecuente recomendación fuesen hechas por un experto antes de la matrícula.

1.4.4. Propusieron mejoras a las herramientas de clasificación supervisada aplicables a problemas aquí expuestos, lo que permitió, a su vez, mejoras en la precisión de la clasificación.

Las técnicas basadas en conjuntos de clasificadores son aquellas que introducen perturbaciones en los datos de entrada para obtener clasificadores individuales. La variabilidad requerida dentro del conjunto es obtenida mediante modificaciones de la distribución de entrenamiento (que se supone debe parecerse a la distribución real) con las que se inducen las variaciones en los clasificadores individuales.

En el presente trabajo, se propuso un nuevo método basado en el algoritmo de construcción de conjuntos *bagging* [Brei-96]. Este algoritmo genera, utilizando otro algoritmo base como motor de aprendizaje, tantos modelos como conjuntos se hayan creado. Una vez producidos los modelos, se usa un conjunto de prueba para obtener, por intermedio de una votación ponderada, la predicción de cada una de sus instancias.

Las presentes propuestas se centran en el uso de conjuntos creados en nuestro propio dominio de aplicación. La primera está basada en la aplicación de un clasificador que utiliza conjuntos provenientes de un muestreo estratificado. La segunda proviene de la generación de conjuntos con repetición por medio de un muestreo aleatorio.

1.5. Estructura de la tesis

El resto de este trabajo está organizado como se precisa a continuación. En el capítulo 2, se resume el estado del arte detallando los sistemas que usan minería de datos en el ámbito de la educación. En ese mismo capítulo, describiremos dichos sistemas y su relación con las técnicas de minería de datos. En el capítulo 3, dedicado a la presentación de los sistemas de recomendación, estos son agrupados en dos grandes categorías: los basados en modelos y los basados en memoria.

A diferencia de los primeros, los basados en memoria presentan antecedentes y se dividen, a su vez, en tres tipos: los basados en contenidos, los basados en filtrado colaborativo y los híbridos. Se concluye que es el de filtrado colaborativo el más conveniente, siempre que se potencie adaptándolo a los sistemas basados en modelos; en otras palabras, en el dominio de la tesis, se ha recontextualizado el algoritmo de filtrado colaborativo para aprovechar las ventajas que ofrecen los basados en modelos.

Para terminar el capítulo, se desarrollan las diferentes métricas de evaluación que serán aplicadas en el presente trabajo.

En el capítulo cuarto, se describen el proceso del descubrimiento del conocimiento y las técnicas asociadas con él, necesarias para hacer una recomendación basada en modelos, pero con la filosofía del filtrado colaborativo: árboles de decisión, reglas basadas en árboles, técnicas de Naïve Bayes, clasificación de vecinos cercanos y conjuntos de clasificadores (principalmente Bagging y Boosting). Asimismo, se propone una nueva técnica sobre la base de la perturbación/alteración del conjunto de entrenamiento. Para finalizar, se trabaja con los métodos de poda, que serán utilizados en los capítulos relativos a la experimentación. Se evidencia aquí que la propuesta es exclusiva para este dominio de aplicación, pues la elección del método está determinada por la estructura de los datos y los sujetos involucrados (periodicidad, aumento de competencia, etc.).

El capítulo quinto, que desarrolla la propuesta de una metodología para preparar los datos, incide en la necesidad de plantear una propuesta a partir de las características particulares del ámbito académico: cambios de plan curricular (cambios del nombre de la asignatura, creditaje, desaparición de alguna asignatura, requisitos), relaciones de determinación entre los cursos, etc. En efecto, cuando la recomendación es presencial, lo primero que se pregunta es por el desempeño del alumno en las materias que constituyen el fundamento del curso que se desea elegir. Por eso, nuestra propuesta debe contener implícitamente el atributo requisito. Si se hiciera un filtrado directo, se perdería mucha información, dado que la base de datos original contiene muchos en mal estado. Por el contrario, la metodología planteada propone incluir conocimiento estático en cada una de sus instancias, así como la secuencia de aplicación del motor sobre registros dotados de cierto conocimiento.

Finalmente, en el sexto capítulo se revisan la experimentación y la evaluación mediante cinco tipos de experimentos, reflejando una íntima coherencia con los conceptos del capítulo anterior. Los experimentos son los siguientes: primero, la aplicación del filtrado colaborativo con nuestra base de datos. Segundo, la comparación entre los resultados de la aplicación del motor a las bases de datos que tienen conocimiento estático y a aquellas que no lo tienen. El aporte en este punto es que se demuestra que la base de datos con conocimiento estático arroja estadísticamente mejores resultados que la original. Tercero, la segmentación de los datos en períodos,

debido a que inicialmente se conjeturaba que para períodos académicos muy antiguos los datos no aportaban mucha información útil. Por lo tanto, se decide aplicar el algoritmo de árboles de decisión con validación cruzada a los conjuntos provenientes de los cortes propuestos, a partir de lo cual se detectan patrones y tendencias relativas al rendimiento académico y a los cambios curriculares propuestos por la universidad. Cuarto, un estudio comparativo de los diferentes métodos de poda para árboles de decisión. Y por último, un estudio comparativo de todos los algoritmos de aprendizaje automático considerados, teniendo en cuenta cada una de las métricas involucradas en la presente investigación.

Capítulo 2:
Estado del Arte

Las técnicas de minería de datos son herramientas que actualmente se están aplicando con éxito en diferentes áreas del conocimiento humano. Su utilidad resalta sobre todo en contextos en los que se requiere analizar una gran cantidad de datos, pues, a partir de ellos, es posible extraer patrones con la finalidad de utilizarlos en la construcción de modelos de predicción. Finanzas y manejo de información, bancos [Han-01], telecomunicaciones [Han-01a], [Luan-02b], medicina [Han-01a], [Han-01b], [Feld-03], industria al por menor [Han-01], [Edel-00], explotación de la información en la web [Moba-96] y en educación [Luan-02a], [Luan-01], [Waiy-03], [Luan-02b], [Rada-06], [Dela-05], [Cort-08][cast-09] son, por ejemplo, algunos de los escenarios y situaciones en los que hoy en día se utiliza.

En cada uno de ellos, la minería de datos cumple funciones específicas, pero, en líneas generales, se concibe como un proceso de extracción e identificación de información útil que luego se transforma en conocimiento. Con el objetivo de extraer conceptos, patrones y relaciones empleados para informar al usuario a través de una consulta, combina técnicas de inteligencia artificial, aprendizaje automático, reconocimiento de patrones, administración de bases de datos, estadística y matemática. La minería de datos tiene, además, funciones tales como análisis de la asociación, clasificación, agrupamiento y predicción [Moba-96], que implican otras tantas técnicas o algoritmos utilizados para llevarlas a cabo.

Actualmente, aunque ha habido un incremento en el interés por el uso de las técnicas de minería de datos, muy pocos trabajos han sido desarrollados para el ámbito educativo. Las investigaciones que desde hace unos años se han enfocado en el uso de técnicas de minería de datos en el contexto de la educación todavía constituyen un número reducido en el conjunto de todas las aplicaciones. A continuación, reseñaremos los estudios realizados por Jing Luan, Naeimeh Delavari, Al Radaideh, E. Castellano, P. Cortez, N. Anand, F. Siraj y M. Ramaswami, sin duda los más significativos.

2.1. Aproximación de F. Siraj y M.A. Addoulha

Siraj en [Sira-09] presentó el resultado de aplicar estas técnicas a bases de datos de educación superior, con el propósito de entender la información relativa a la matrícula de los estudiantes de la Universidad de Sebha, en Libia. Utiliza aproximaciones descriptivas y predictivas para descubrir información escondida. En particular, para agrupar los datos en conjuntos basados en similitudes, se empleó el análisis de *clusters*.

Estos se usaron como “insumo” para los experimentos de predicción. Para el análisis predictivo, se aplicaron redes neuronales, regresión logística y árboles de decisión. El estudio mostró que, de las tres técnicas, las redes neuronales fueron las de mayor precisión.

En este estudio, se utilizó la metodología CRISP considerando las particularidades del dominio. Las fases de entendimiento del negocio y de entendimiento de los datos se realizaron en paralelo, debido a que el objetivo final no podía identificarse hasta que la oficina de registro de los estudiantes no se familiarizase con los datos. El conjunto de datos contiene 8510 instancias desde 1998 hasta 2006. Se incluyen 38 atributos, ocho numéricos y el resto de tipo categórico. Como resultado de la fase de preproceso, se obtuvieron 6830 instancias.

El primer estudio consistió en hacer estadísticas descriptivas basadas en la naturaleza de los datos, tales como tablas de frecuencia y relaciones entre atributos. El análisis de *clusters* se utilizó para determinar la similitud entre atributos del conjunto de datos.

Primero se determinó la distribución de alumnos por carrera, debido a que la Universidad tiene varios locales. Este estudio sirvió para predecir cuáles, en el futuro, tendrían menos alumnos en alguna carrera. También se observó que la mayoría de los estudiantes eran de género femenino y que preferían carreras como ciencias, artes, medicina y odontología.

Luego se aplicó *clustering*. Como resultado, con cada *cluster* se identificaron tres conjuntos y las relaciones entre atributos tales como religión, género, nacionalidad, estado, grado de pertenencia, facultad, estatus de estudio, tipo de admisión, estatus de vivienda y tipo de registro.

Para el análisis predictivo, se emplearon tres técnicas. Para el análisis de regresión logística, los atributos de facultad y nacionalidad fueron los únicos significativos en cada *cluster*; por ello, el modelo arroja 99.44 por ciento de precisión. En el análisis del árbol de decisión, también se usaron los mismos atributos, por lo que se obtuvo el 99.77 por ciento de precisión. Por último, el mismo análisis se aplicó en las redes neuronales y se obtuvo 99.98 por ciento de precisión.

En conclusión, el estudio de Siraj hace estadística descriptiva para visualizar datos, luego aplica una técnica de agrupamiento para formar grupos y, por último, aplica técnicas de aprendizaje a cada grupo por separado considerando únicamente dos variables. A diferencia de nuestro trabajo, F. Siraj predice en forma general a qué carreras, facultades y universidades, según su situación geográfica, los alumnos se matricularían según los atributos elegidos por ellos. Por esta razón tiene un alto porcentaje de aciertos. Nosotros, en cambio, aplicamos técnicas de aprendizaje automático para predecir el rendimiento académico de los estudiantes. La información que brindamos ayuda a los estudiantes a decidir su matrícula en el semestre académico

para el cual hacen la consulta. En nuestro caso, finalmente, la aplicación de técnicas de *clustering* sería inapropiada, debido a que nosotros, desde el principio, hemos decidido trabajar un problema de predicción con clasificación supervisada.

2.2. Aproximación de Naeimeh Delavari

Delavari en [Dela-05] presenta y justifica las capacidades y habilidades de la tecnología de minería de datos en el contexto de sistemas de educación superior al proponer un modelo para mejorar la eficiencia y efectividad de sus procesos. Parte de la constatación de que, a partir de datos relativos a la gestión en la educación superior, la tecnología de minería de datos puede descubrir patrones ocultos, asociaciones y anomalías, y, mediante este conocimiento, mejorar el proceso de toma de decisiones acertadas. En particular, el autor propone un modelo que sirva como guía para que las instituciones de educación superior puedan determinar cuán bien las técnicas de minería de datos pueden asistir a dichas organizaciones en la optimización de sus decisiones y en la identificación de las partes de los procesos educativos que deben ser mejoradas.

Para crear el modelo, se basaron en indicadores funcionales e indicadores básicos, los cuales fueron categorizados en tres grupos: los de entrada, los de proceso y los de salida. Los indicadores de entrada se refieren a recursos humanos (estudiantes, científicos, personal administrativo), recursos financieros (créditos, costos de estudio, investigación y construcción) y recursos sistemáticos (campos de estudio, laboratorios, plazas).

Los indicadores de proceso se usan para evaluar la funcionalidad y rendimiento del sistema educativo en general. Estos se refieren a los métodos para la investigación y para la enseñanza, y a las mejoras educativas cuantitativas y cualitativas (tasa de registro, tasa de deserción, tasa de retención). Los indicadores de salida se refieren básicamente a los alumnos y graduados.

El modelo planteado consta de siete procesos principales, que son los más usuales en los sistemas de educación superior: evaluación, planeamiento, registro, consultoría, marketing, rendimiento y evaluación. Cada proceso es categorizado en subprocesos. Por ejemplo, el de evaluación tiene los subprocesos de evaluación de la asignatura y evaluación del entrenamiento en la empresa.

La principal idea del modelo es identificar cómo cada uno de estos procesos tradicionales puede mejorarse a través de técnicas de minería de datos.

Cada subproceso cuenta con la descripción de un cierto conocimiento implícito en los datos. Para cada descripción, se detalla la tarea que se efectuará para obtener dicho conocimiento y la técnica de minería que se utilizará.

En resumen, en este trabajo de investigación se plantea, en el marco general de la aplicación de las técnicas de minería de datos en el ámbito educativo, un modelo que describe los procesos, subprocesos, el conocimiento implicado en los subprocesos, la metodología para llegar a dicho conocimiento y, por último, la propia técnica de minería empleada.

Nuestro trabajo de investigación correspondería, si seguimos la propuesta de Delavari, al proceso principal de registro o matrícula y al subproceso de matrícula de los estudiantes en un curso. El conocimiento que debe adquirirse, según el modelo, son los patrones de estudiantes que previamente se han matriculado en dichas asignaturas. La finalidad de ese conocimiento es predecir qué tipo de estudiante cuenta con más posibilidades de pasar dicha asignatura y formular las técnicas de clasificación y asociaciones, técnicas que, según el modelo, deben aplicarse.

2.3. Aproximación de Ramaswami

Los objetivos de la investigación de Ramaswami en [Rama-10] fueron la generación de un conjunto de variables predictivas y la identificación de la alta influencia que este podría tener sobre el rendimiento académico de los estudiantes en educación superior. Por otro lado, la investigación necesitaba de una herramienta para la aplicación de técnicas de minería de datos. Para ello, se desarrolló una basada en la técnica CHAID para la construcción de árboles (CHI-squared Automatic Interaction Detection), y se validó su funcionamiento con datos de estudiantes de secundaria del sistema de educación en India. En total, se recolectaron datos para 35 atributos.

La metodología de este trabajo de investigación es experimental y generó una base de datos construida de una fuente primaria y secundaria. Los datos primarios fueron recolectados de estudiantes regulares y los secundarios fueron reunidos de bases de datos existentes en la oficina de registro de cinco escuelas distintas en tres diferentes distritos de Tamilnadu, en la India.

Se hizo un estudio piloto seleccionando dos colegios en el distrito de Madural de Tamilnadu. Toda la información demográfica, académica y socioeconómica se obtuvo de 224 estudiantes a través de cuestionarios. Las notas fueron extraídas de las bases de datos de la oficina de registro.

Al aplicar una regresión lineal simple, después de codificar las variables categóricas, la precisión predictiva del rendimiento de los estudiantes fue de 39.23 por ciento. El resultado de este estudio experimental reveló que había una correlación fuerte entre los factores referidos a la localización de la escuela, el tipo de escuela, la educación de los padres y sus notas obtenidas en el nivel secundario y el rendimiento académico de los estudiantes.

Adicionalmente, se desarrolló un estudio detallado basado en el estudio piloto. Los datos fueron recolectados de cinco colegios distintos de tres diferentes distritos, lo que resultó en un total de mil conjuntos de datos. Estos se reprocesaron por medio de transformaciones y condicionamientos con el objetivo de simplificar y robustecer el modelo. Como resultado, se obtuvieron 772 registros de estudiantes, los cuales se procesaron usando la herramienta CHAID para la construcción de árboles de predicción. Un total de 228 estudiantes de mil registros fueron eliminados debido a respuestas irrelevantes, ausencias en el examen final e información incompleta de la base de datos de la oficina de registro.

Se encontró que la precisión del modelo predictivo, después de aplicar el método CHAID, fue de 44.69 por ciento. Esto indica que el modelo clasifica correctamente a 345 estudiantes de 722, y que su precisión es mejor que la del propuesto por AL- Radaideh [Rada-06].

El presente trabajo se diferencia del descrito debido, principalmente, a los atributos usados y al objetivo que se tiene con cada uno de ellos. El objetivo de Ramaswami es caracterizar el rendimiento por factores demográficos; por ello, usa 34 atributos demográficos y solo uno de carácter académico, a diferencia de la presente investigación, que no usa ningún atributo demográfico y se concentra totalmente en lo académico.

Además, Ramaswami usó la técnica de clasificación CHAID, que es uno de los algoritmos de árboles de decisión que genera particiones recursivas de una población en conjuntos disjuntos. Estos conjuntos, llamados nodos, son particiones de tal modo que la variación de las variables son minimizadas dentro de los conjuntos y maximizada entre conjuntos. Con esta técnica, obtuvo una precisión del 44.69 por ciento, mientras que, en la presente investigación, la precisión predictiva no baja del 81 por ciento.

2.4. Aproximación de Jing Luan

Jing Luan ha dirigido las capacidades de la minería de datos a través de cuatro estudios. El primero de estos [Luan-01] responde a la siguiente pregunta: ¿qué hacen las instituciones educativas para conocer a sus estudiantes? Esta investigación se desarrolló con el propósito de crear resultados de tipologías significativas para quince mil estudiantes en varios niveles educativos. Los datos primarios utilizados para este análisis son los resultados educativos de los estudiantes en combinación con la duración de sus estudios. A partir de ello, se pudo observar que, al extraer los patrones de los resultados de aprendizaje de estudiantes anteriores en combinación con la duración del estudio, se pueden crear tipologías de resultados de aprendizajes significativos con algunas técnicas de *clustering.*

El uso de la minería de datos en este estudio de caso ayuda al sistema educativo a describir mejor los grupos de conjuntos de estudiantes homogéneos y a crear series de tipologías con un nombre predefinido. Estas pueden utilizarse más allá de la creación del perfil tradicional del estudiante. Los *clusters* ayudan a las universidades a identificar mejor los requerimientos propios de cada grupo y a tomar mejores decisiones en la formulación de ofertas de cursos, currículos, tiempo requerido para enseñar, entre otros. El resultado es la mayor satisfacción de los estudiantes con sus instituciones y sus estudios, los cursos que toman y los períodos de clases. A partir de ello, los docentes pueden formular planes diferenciados para cada grupo.

El otro estudio de caso realizado por Jing Luan [Luan-01], [Luan-02a] pretendió predecir las probabilidades de traslado de un estudiante y facilitar a la institución una intervención temprana dedicada a los que presentaban mayor riesgo. Se partía de la constatación de que, debido a las dificultades académicas, más de la mitad de estudiantes en una universidad terminaban trasladándose a lo largo de los cuatro años de estudio, y de que muchos tomaban un buen tiempo para trasladarse o simplemente lo solicitaban sin llegar a realizarlo. Fueron estas dos últimas variables las que se consideraron como iniciales para caracterizar el perfil de los estudiantes con riesgo de trasladarse.

Otras variables que sirvieron en la predicción de la transferencia de estudiantes fueron sus objetivos educacionales, número de cursos tomados en el traslado, número de unidades intentadas, el acceso a ayuda financiera y etnicidad. En este caso, se usaron las técnicas de clasificación supervisada; los analistas emplearon algoritmos de redes neuronales artificiales, que tuvieron una precisión del 72 por ciento, y la inducción de regla del C5.0) que tuvo una precisión del 80 por ciento e, incluso, llegó a comparar y a contrastar los resultados.

Este tipo de estudio permite a las universidades predecir la probabilidad de traslado de los estudiantes. También las ayuda a prestar más atención en aquellos que necesitan asistencia académica, al programarles clases adicionales y plantear horas de consejería con tutores y psicólogos de la universidad. Con ello se previene que los estudiantes desaprueben.

En el tercer caso de estudio llevado a cabo por Jing Luan [Luan-01], [Luan-02a], la matrícula en las instituciones estudiadas puede llegar a ser la décima parte del total del alumnado. La mayoría de las universidades envían correos a los alumnos sobre una base regular, llegando a asumir un costo muy fuerte por año debido a esta actividad. En este estudio, Luan demuestra que la minera de datos ayuda a las universidades a centrarse en los alumnos con mayores probabilidades de hacer donativos. En otras palabras, las ayuda a desarrollar un método rentable para con aquellos estudiantes cuyas probabilidades de hacer donativos sean más altas.

Los patrones de éxito de antiguos graduados que han contribuido a la universidad forman el conocimiento descubierto por un algoritmo de predicción. El resultado de estos patrones es facilitar a la universidad el envío de sus correos electrónicos, de tal manera que sea más efectivo, y así incrementar la cantidad de alumnos donantes. La ventaja principal de esta actividad para las universidades es el ahorro que implica dirigir correcta y efectivamente las comunicaciones con sus ex alumnos interesados en donar a su institución.

El último estudio hecho por Luan [Luan-02a] buscaba predecir la probabilidad de que los estudiantes persistan o no en sus estudios y agruparlos para que la institución pueda aplicar las estrategias adecuadas para mejorar la tasa de persistencia, lo que implica disminuir la tasa de deserción. Para ello se tomaron los datos de una universidad en Silicón Valley. Las dos técnicas de clasificación usadas fueron las redes neuronales artificiales y los árboles de decisión; junto con la técnica de agrupamiento de dos pasos, se aplicaron al conjunto de perfiles del estudiante.

Al extraer los patrones exitosos de estudiantes previos a través de técnicas de predicción, las universidades pueden predecir la probabilidad de persistencia de un alumno. La minería de datos, así, ayuda a las universidades a identificar a aquellos que son menos propensos a regresar a la universidad cada año. Como consecuencia, teniendo este conocimiento sobre el estudiante, resulta fácil intervenir a tiempo para ayudarlo. En el caso estudiado por Luan, la relación de estudiantes predichos como aquellos con menores probabilidades de regresar a la universidad es administrada por la facultad para tomar decisiones anticipadas que buscan no perderlos. En suma, la influencia de la minería de datos en las estrategias de marketing incrementa la tasa de persistencia del estudiante.

En los cuatro casos, Luan hace predicciones sobre tendencias de matrículas, de traslados, de donaciones de ex alumnos y de persistencia de estudiantes para concluir sus estudios. A diferencia de nuestro trabajo, que usa algunos atributos históricos de los estudiantes, Luan enfoca más sus resultados a crear un sistema de soporte de decisiones para las universidades, utilizando, con mayor énfasis, variables del tipo demográfico. Si bien es cierto que estas variables están disponibles en todo sistema universitario, estas, en la presente investigación, no tienen influencia, ya que, a diferencia del trabajo hecho por Luan, se intenta recomendar al estudiante sobre la matricula que debe hacer, utilizando para ello la predicción de su rendimiento académico futuro basado en el rendimiento de otros estudiantes de características similares. Estas variables, en nuestro caso, proporcionarían información sobre las calificaciones o sus predicciones que no necesariamente influiría en estudiantes futuros. Por ejemplo, las variables de género, carrera profesional, nivel socioeconómico etc., son variables que Luan usa, ya que sus objetivos, completamente distintos a los nuestros, se dirigen a

predecir acciones futuras que no tienen que ver directamente con el rendimiento académico.

2.5. Aproximación de Paulo Cortez

Paulo Cortez en [cort-08] desarrolló una investigación cuyo objetivo es modelar y predecir el rendimiento de los estudiantes de educación secundaria mediante el uso de minería de datos (DM) e inteligencia de negocios (BI). Para ello, se recolectaron datos reales recientes (notas, información demográfica, características relacionadas con lo social) a partir de la información proporcionada por los colegios y de la recogida a través de cuestionarios. Además, fueron modelados los dos cursos más importantes, matemática y portugués, usando clasificadores binarios de cinco niveles y tareas de regresión. También se aplicaron cuatro técnicas de DM (árboles de decisión, árboles aleatorios, redes neuronales y máquinas de soporte vectorial) y tres diferentes tipos de entrada de datos (con notas previas y sin ellas).

Los resultados indicaron que se obtiene una buena precisión predictiva siempre y cuando la información de las notas del primer y segundo período del colegio esté disponible. Aunque los logros de los estudiantes estuvieron altamente influenciados por las evaluaciones pasadas, un análisis exploratorio mostró que otras características también resultan relevantes (ausencias, educación y trabajo de padres y consumo de alcohol).

El objetivo de este documento fue predecir los logros de los estudiantes e identificar las variables clave que condicionan el éxito o el fracaso educativo. En lo esencial, buscaba responder a dos preguntas que los autores generalmente formulan en el ámbito de la educación: ¿es posible predecir el rendimiento de un estudiante?, ¿cuáles son los factores que afectan dicho rendimiento? Como se mencionó líneas antes, para responderlas, se analizaron datos reales de los colegios secundarios portugueses a partir del reporte de notas y cuestionario. Este permitió la recolección de muchos datos demográficos, sociales y atributos relacionados con la escuela (edad del estudiante, consumo de alcohol, educación de la madre), complementando la escasa información del reporte (solo notas y ausencias).

Para la experimentación, se utilizaron cuatro técnicas supervisadas, tres configuraciones y tres objetivos distintos de minería de datos. Las técnicas supervisadas fueron los árboles de decisión, los árboles aleatorios, las redes neuronales y las máquinas de soporte vectorial. Las tres configuraciones consideraban, en el conjunto de entrenamiento, las notas parciales, finales y los promedios. Para la primera configuración, se incluyeron en el conjunto todas las variables consideradas para el estudio excepto la nota final. De manera similar, la segunda configuración consideró los mismos atributos de la primera eliminando la nota del segundo período. Por último, la

tercera configuración consideró los mismos atributos tomados para la segunda configuración, pero sin involucrar la nota del primer período.

Por último, los tres objetivos involucrados en el diseño de los autores se refieren a las formas de manejar las clases cuyos valores son binarios, de cinco niveles y con salidas numéricas. Para la clasificación binaria, se consideraron los valores de las clases aprobado y desaprobado; para la clasificación con cinco niveles, se incluyeron las escalas desde “muy bueno” hasta “insuficiente”; y para las salidas numéricas, se usó regresión.

Este estudio consideró los datos correspondientes a los años 2005 y 2006 de dos escuelas públicas de la región de Alentejo en Portugal. Los datos fueron recolectados de dos fuentes: los reportes de notas de las dos escuelas que contienen las notas de los tres períodos con el respectivo número de ausencias de cada estudiante en cada signatura, y un cuestionario que complementa la información académica.

Los cuestionarios de 37 preguntas fueron aplicados a 788 estudiantes, de los cuales 111 fueron eliminados por falta de información. Finalmente, los datos fueron integrados en dos conjuntos de datos relacionados con las asignaturas de matemática y portugués.

Todos los experimentos fueron conducidos mediante la herramienta RMINER, librería de R que facilita la aplicación de técnicas de minería de datos. La herramienta presenta un conjunto de funciones para las tareas de clasificación y regresión. Para la experimentación, fue necesario estandarizar todos los atributos a media cero y desviación estándar uno. Se llevaron a cabo veinte corridas de diez veces validación cruzada para cada configuración.

Como una comparación base, el predictor de Naïve también fue probado. Para la primera configuración, resultó un modelo igual al obtenido considerando la nota del segundo período. Cuando las notas del segundo período no estuvieron disponibles (segunda configuración), se emplearon las del primero. En el caso de no considerar ninguna evaluación (tercera configuración), se tomó en cuenta la clase más común (para tareas de clasificación) o el valor de salida promedio (regresión).

Como se esperaba, la primera configuración logró los mejores resultados. La precisión predictiva decrece cuando la nota del segundo período no existe (segunda configuración) y el peor resultado se obtiene cuando no existen notas de los estudiantes.

Al igual que en algunos trabajos de investigación, Cortez en [cort-08] hace un estudio predictivo del rendimiento de estudiantes de escuelas secundarias tomando como base variables demográficas y una sola variable académica correspondiente a solo dos asignaturas clave: matemática y portugués.

Cortez considera que algunas variables demográficas combinadas con una variable académica (notas de asignaturas anteriores) son suficientes para predecir calificaciones. Sin embargo, no considera el nivel de conocimientos diferenciado que

cada estudiante trae de la asignatura o asignaturas anteriores, variable que sí es considerada en la presente investigación.

Asimismo, Cortez considera que el nivel de inasistencias es importante para la predicción, atributo que en la presente investigación no es considerado, debido a dos razones. En primer lugar, se cree intuitivamente que el nivel académico está íntimamente relacionado con el esfuerzo que pone el estudiante en preparar una determinada asignatura y que dicho esfuerzo implica, aparte del estudio, la asistencia rigurosa a clases. En segundo lugar, la universidad elimina automáticamente del examen final a aquellos alumnos que tienen más del 20 por ciento de inasistencias en la asignatura.

2.6. Aproximación de Anan Kumar

Anan Kumar en [Anan-09] desarrolló un estudio teórico introductorio y propuso el empleo de técnicas de clasificación para mejorar la calidad del sistema de educación superior, evaluando los datos de los estudiantes con la finalidad de saber cuáles son los principales atributos que pueden afectar su rendimiento académico. Usó la metodología CRISP y formuló una arquitectura basada en técnicas de agrupamiento para la limpieza de los datos y en árboles de decisión para la fase de descubrimiento de patrones.

La arquitectura consistió en un proceso general que se inició con la extracción de datos de las bases de datos de los estudiantes y de los profesores. Una vez reunidos los dos conjuntos, se procedió a limpiarlos, eliminando parte de ellos en forma manual, debido a que los investigadores los consideraban irrelevantes para el estudio.

Normalmente, en KDD, el proceso previo a la aplicación de las técnicas de minería es una etapa muy importante, en donde se agrupan y se limpian los datos. En esta aproximación se usaron algoritmos de agrupamiento, específicamente el de k-means, que para este caso resultó ser el mejor.

En esta investigación, luego del procesamiento de los datos, se procede a clasificarlos con la ayuda de los algoritmos de clasificación como el Naïve Bayes y el árbol de decisión. El conocimiento adquirido, según los autores, puede utilizarse, eventualmente, para profundizar el entendimiento de los patrones que siguen las matrículas en una determinada asignatura y para que la administración pueda tomar mejores decisiones a partir de los datos de los estudiantes registrados en años anteriores.

A diferencia de nuestro trabajo, Kumar primero trata de determinar atributos relevantes usando técnicas de clustering. En nuestro caso, estos atributos se consideran según el juicio del experto; más aun, en la presente investigación se consideran los cursos requisitos y se crea un atributo que caracteriza el rendimiento de este estudiante antes de llevar la asignatura que Kumar no considera. Por otro lado, Kumar usa

directamente árboles de decisión; en cambio, en la presente investigación usamos un metamodelo por votación que usa árboles de decisión como algoritmo base.

2.7. Aproximación de Vasile Bresfelean

El objetivo de esta investigación es proveer conocimiento valioso sobre el entendimiento, predicción y prevención de los fracasos académicos de los alumnos para que la institución, mediante decisiones acertadas, pueda revertir la situación. Se usan los algoritmos de *clustering* (FarthesFirst) y J48 de la herramienta Weka con la finalidad de encontrar perfiles de éxito y fracaso de los estudiantes.

Bresfelean en [Bres-08] construye un perfil de los estudiantes que fracasan en sus exámenes mediante la información extraída de dos fuentes: una directa, que proviene de los estudiantes de la facultad de economía y administración de negocios de Cluj-Napoca, quienes llenan una encuesta en línea; y otra indirecta, de las propias bases de datos de la universidad. La información recolectada consiste en datos generales: género, colegio de procedencia, situación escolar, becas obtenidas, interrupción de estudios, ausencia en sus exámenes, categoría de pago y opiniones de los estudiantes en general (cursos, materiales, infraestructura, conocimiento adquirido, educación continua, etcétera).

Se usó Excel para manejar los datos y los atributos fueron recodificados. Asimismo, se usaron palabras clave en el caso de las respuestas largas de las encuestas. En total, llegaron a emplearse 50 instancias. En el *clustering*, se aplicó el método Farthest First basado en K-Means. Se especificó el parámetro k=2, correspondiente a los dos grupos de estudiantes: el primer grupo fue identificado como el de los alumnos que pasaron todos los exámenes (42 instancias) y el segundo como aquel que falló uno o más exámenes (8 instancias). Los k puntos fueron elegidos aleatoriamente como centroides de cada *cluster*. Todas las instancias fueron asignadas al centro más cercano del *cluster* según la métrica de distancia euclidiana ordinaria. Luego los centroides de las instancias en cada *cluster* fueron calculados, y tomados como nuevos valores centrales de sus respectivos *clusters*. Finalmente, el proceso entero fue repetido con los nuevos centroides. La iteración continuó hasta que los mismos puntos fueran asignados a cada *cluster*.

Luego se separó a los estudiantes en segmentos con perfiles distintos. Los del mismo segmento eran los de perfil más cercano; los de diversos segmentos presentaban un perfil muy distinto.

Primero, aplicaron el j48 basado en *training set* y obtuvieron 96 por ciento de éxito; luego utilizaron la validación cruzada y obtuvieron el 76 por ciento. Cabe mencionar que esta versión de j48 usada por los autores emplea un estimador de

Laplace. El resultado es un árbol de decisión cuya raíz es el atributo que corresponde a la información sobre si el estudiante continúa su educación después de finalizar sus estudios de grado, en un segundo nivel, el árbol considera el tipo de admisión de los estudiantes. Es decir, si fueron admitidos por sus notas, por su bachillerato o porque ya poseían un grado.

A diferencia de la presente investigación, Vasile usa técnicas de minería de datos con información extraída de dos fuentes; sin embargo, es preciso mencionar que una de ellas, la encuesta, no siempre es fiable, pues muchas veces se completa sin la seriedad debida. En nuestro caso, solo se emplea información del rendimiento académico obtenido por los alumnos a partir del sistema y, si bien se usa un par de atributos sintéticos extraídos indirectamente, estos se construyen a partir de datos primitivos, lo que asegura su confiablidad.

2.8. Aproximación de Cristóbal Romero

El objetivo fundamental de esta investigación fue comparar diferentes técnicas de minería de datos para clasificar estudiantes mediante dos factores: los datos de uso al seguir cursos web y las notas finales al terminarlos. Romero et. al en [Rome-08] desarrollaron una herramienta específica de minería de datos dentro de Moodle, orientada para que fuese usada por los instructores y, de esa manera, para facilitar el uso de las técnicas de minería. En este sentido, los instructores pueden crear y mantener cursos vía Web y también llevar a cabo el proceso de minería de datos en la misma herramienta, con la finalidad de retroalimentarse y mejorar su curso por esa vía.

Adicionalmente, se llevaron a cabo algunos experimentos para evaluar el rendimiento y la utilidad de diferentes técnicas de clasificación, con el objetivo de predecir notas finales de estudiantes basadas en la información de datos de uso web, cuando los estudiantes llevaban sus cursos virtualmente.

Se usaron los datos de 438 estudiantes en siete cursos Moodle de la Universidad de Córdoba. Se emplearon diferentes atributos, como el número de identificación del curso, el número de asignaciones hechas, el número de pruebas tomadas, el número de pruebas aprobadas/suspendidas, el número de mensajes enviados/leídos al fórum, el total de tiempo usado en las asignaciones/pruebas y en el fórum, y, por último, la nota final.

Se discretizaron (manualmente) todos los valores de los atributos con el objetivo de que los instructores los interpreten mejor. Por ejemplo, las notas finales fueron discretizadas en: (FAIL: is value<5; PASS; is Value is >=5 and <7; GOOD: if value>=7 and<9; Exellent: if value is >9). Los otros atributos fueron discretizados en forma de LOW, MEDIUM and HIGH.

Dentro de todas las referencias del estado del arte, [rome-08] es la que más se aleja a los objetivos de la presente investigación, debido a que Romero aplica minería de datos para clasificar estudiantes con iguales notas finales en diferentes grupos según las actividades llevadas a cabo en un curso vía web. Para ello se estudian los datos de uso de cursos hechos en Moodle en la universidad de Córdova. En la presente investigación, en cambio, se predice el rendimiento de un estudiantes en determinada asignatura, según atributos y notas de otros estudiantes del pasado. Sin embargo, la presente investigación predice el rendimiento de un estudiante a partir de dos factores: los rastros que dejó el estudiante al seguir el curso vía web y las notas obtenidas por el estudiante al finalizar dicha asignatura.

Para llevar a cabo la experimentación, Romero usa técnicas de discretización de variables y técnicas de rebalanceo. Para la medida de la precisión, usa la media geométrica y obtiene rendimientos cercanos al 70 por ciento con varios métodos aplicados a los mismos datos. En este trabajo, en cambio, se mencionan varias métricas de precisión incluyendo las curvas ROC, que son especiales para la confiabilidad de aprendizaje de datos desbalanceados.

2.9. Aproximación de Nguyen Thay Nghe

El objetivo fundamental de esta investigación fue comparar la precisión de algoritmos de árboles de decisión contra los de Redes Bayesianas para dos diferentes instituciones: CTU (Can The University), Vietnam, y AIT (Asian Institute of Technology), en Asia. Aunque se trata de analizar dos poblaciones completamente distintas, se han obtenido resultados similares para la precisión del modelo predictivo que representa el rendimiento estudiantil.

Nguyen Thai et al. proponen en [Nguy-07] una metodología que se inicia desde la elección de una buena herramienta para hacer minería. Después de un estudio muy simple, deciden utilizar Weka, debido principalmente a la facilidad de empleo con grandes conjuntos de datos. El próximo paso fue la recolección y preparación de los datos. Para CTU se recolectaron 20,492 instancias para estudiantes admitidos desde 1995 a 2002, y para AIT 936 instancias para estudiantes admitidos dentro de los años 2003 y 2005. Se seleccionaron atributos relevantes disponibles y se crearon atributos nuevos a partir del conocimiento del dominio. Se calculó la ganancia de información para cada atributo, con la finalidad de identificar aquellos con una gran influencia en la clasificación.

Para CTU, se determinaron dos atributos (con una ganancia de información alta): CGPA2 (promedio ponderado acumulado del segundo año) y las habilidades en inglés.

Para AIT, el atributo con más alta ganancia de información fue el nivel del instituto de procedencia.

Para modelar el problema de la predicción del rendimiento académico, se entrenaron datos con árboles de decisión y con Redes Bayesianas. Con el uso de estos modelos, se afinaron los atributos de entrada del conjunto de datos subdividiendo el rango de valores en nuevas clases y evaluando los cambios en la precisión de la predicción. En algunos casos, esto conllevó una mejora significativa en la precisión. Por ejemplo, los autores experimentaron con cuatro distintas divisiones en el conjunto del promedio acumulado (GPA). Primero, consideraron valores continuos desde 2.0 hasta 4.0; luego, discretizaron dicho atributos con cuatro valores (Fail, fair, Good, Very Good) correspondientes a los intervalos (2.0-2.5, 2.5-3.0, 3.0-3.5, 3.5-4.0), de lo que resultó una precisión ligeramente menor. En tercer lugar, procedieron a discretizar el mismo atributo en tres valores discretos (Warning, Good, Very Good), con resultados mejores, pero aún menores que la distribución continua. Por último, se decidió dividir el atributo en tres conjuntos del mismo tamaño, con superiores logros en la predicción después de aplicar la clasificación.

Al aplicar los algoritmos de árboles de decisión y de redes bayesianas para determinar la predicción, se obtuvieron los siguientes resultados relativos a la precisión: con el conjunto de CTU y con cuatro valores en su clase, se obtuvo 75.95 por ciento en la precisión, y con el conjunto de AIT, 70.62 por ciento. Cabe destacar que, para dos clases, Thai obtuvo los resultados siguientes: 94.03 por ciento para CTU y 92.74 por ciento para AIT.

A diferencia de nuestra investigación, Thai obtiene mejores resultados en su precisión debido, quizá, al uso de variables demográficas que, para este caso, tienen una alta correlación con el desempeño de los estudiantes en ambos centros de educación.

2.10. Aproximación de Emilio Castellano

E. Castellano [Cast-08] propuso la aplicación del filtrado colaborativo en la recomendación de asignaturas y/o perfiles mediante la estimación de las calificaciones que obtendría un alumno en asignaturas que aún no ha cursado, basándose en las calificaciones sobre las mismas asignaturas que alumnos con el mismo perfil académico obtuvieron en el pasado. Lo que se pretendía estudiar es la validez del uso del filtrado colaborativo como herramienta para orientar al alumnado cuando tome decisiones que impliquen elección de materias, de perfiles o modalidades académicas, e incluso detección de asignaturas con potenciales problemas y necesidades de refuerzo para el individuo.

El objetivo principal de esta investigación fue a la pregunta acerca de si es posible utilizar el expediente académico de un estudiante para orientarlo al momento de elegir otras asignaturas o, en general, su futuro. Los autores definieron el término “expediente académico” como un conjunto de calificaciones obtenidas por un alumno para una serie de materias cursadas a lo largo de cierto período de tiempo. La respuesta, inicialmente, no fue del todo clara, puesto que se sabe que entran en juego factores subjetivos, psicológicos y aptitudinales.

Dado que las calificaciones de un individuo aportan información fiable sobre sus aptitudes, las áreas en las que mejor se comporta e, incluso, sus preferencias, la investigación pretendió evaluar si un sistema de recomendación Colaborativo, estimando la posible calificación que un alumno obtendría en una materia en caso de cursarla, puede proporcionar información relevante que, conjugada debidamente con otro tipo de informaciones, dé lugar a un sistema capaz de ayudar a los individuos cuando tomen decisiones sobre su futuro. Para ello, los autores realizaron una serie de experimentos a fin de obtener una respuesta fiable a la pregunta anterior.

El conjunto de datos que se utilizaron en los experimentos fueron las calificaciones de los alumnos en una serie de materias. Normalmente, los sistemas de recomendación trabajan o bien con datos explícitos (directamente aportados por el usuario sobre sus propias percepciones), o bien implícitos (obtenidos automáticamente por el sistema en función del comportamiento del usuario) [Herl-04]. En este caso los datos no corresponden a alguno de los dos tipos mencionados anteriormente, puesto que las calificaciones resultan de una interdependencia entre el estudiante y otro sistema que no es el de recomendación.

El conjunto de datos que se utilizó está formado por un total de 744 alumnos entre cuarto año de educación secundaria y 1º y 2º de bachillerato de nueve promociones procedentes de varios centros educativos andaluces. Considera hasta 100 asignaturas y un total de 15752 calificaciones, que contemplan valores enteros comprendidos entre el 0 y el 10. En las etapas educativas involucradas, los estudiantes empiezan a tener necesidades de tomar decisiones sobre su itinerario académico.

Para dar respuesta a la pregunta, base fundamental de esta investigación, se realizaron numerosos experimentos para medidas de similitud y predicciones basadas en memoria. Para evaluar el comportamiento de los distintos algoritmos de filtrado colaborativo, se utilizaron las métricas de evaluación del Error Medio Absoluto, que son aquellas que estiman la exactitud con la que el sistema realizará las predicciones, y la cobertura, que calcula el porcentaje de ítems para los que el sistema es capaz de proporcionar una predicción [Herl-04].

En los experimentos, se estudió el comportamiento de la mayoría de variantes conocidas de algoritmos de filtrado colaborativo [Bree-99], intentando optimizar

diferentes parámetros como porcentajes de conjunto de entrenamiento y prueba, número de vecinos, factor de relevancia y medidas de predicción.

Los resultados obtenidos corresponden al uso del coeficiente de correlación de Pearson como medida de similitud, a la que se aplica un factor de relevancia que parte de la idea de que serán más relevantes aquellas medidas de similitud en las que han participado un mayor número de valoraciones [Herl-1999]. Para ello, se multiplica el coeficiente de correlación de Pearson por el número de alumnos usados en el cálculo de similitud dividido entre una constante, que se ajustó a 30 tras haberse variado en los experimentos entre 5 y 60. El número de vecinos K escogidos para el cálculo de predicciones que mejor ha resultado ha sido 15 (después de realizar pruebas con valores entre 5 y 50). Las predicciones del sistema se han calculado mediante la suma media ponderada y se les aplicó una mejora denominada amplificación de casos [Bree-98], que enfatiza en las predicciones aquellas valoraciones aportadas por vecinos con mayor similitud y penaliza las más lejanas.

Se verificó que el MAE, en las predicciones, es aceptable y es del orden del 0.902, número muy aceptable para las necesidades que se pretendieron en esta investigación. Debido a ello, los autores desarrollaron el software Orieb, basado en la metodología, herramientas y técnicas usadas en la investigación.

Orieb es un sistema para ayudar a aquellos alumnos que quieran cursar Bachillerato, partiendo de los datos usados en la experimentación. El sistema lleva a cabo tres tipos de recomendaciones: la modalidad de Bachillerato más adecuada para el individuo (a elegir de entre cuatro posibles), las asignaturas de la modalidad más recomendadas y las asignaturas en las que el alumno puede requerir refuerzo educativo.

Dado que las recomendaciones se calculan en base a la similitud de los usuarios, y esta puede dar valores sesgados (alta similitud con números bajos de elementos en común), para aumentar la fiabilidad de las recomendaciones, el sistema proporcionará información adicional. Esta expresa no sólo lo adecuada que es una recomendación para el individuo, sino también el grado de confianza que merece teniendo en cuenta, según las predicciones, cómo esa recomendación es construida por el sistema.

A partir de ello, se construyó el grado de interés que una modalidad puede presentar para un alumno mediante tres factores: (i) la media de las calificaciones correspondientes a las materias propias de dicha modalidad, (ii) la varianza en las calificaciones de dichas materias y (iii) la cobertura de tales previsiones, considerando que, cuantas más materias de entre todas las propias de la modalidad se contemplen y cuanto menor sea la diferencia entre las calificaciones de dichas materias, mayor será el grado de confianza que puede aportar la media de las predicciones.

Para el caso de recomendaciones individuales, se presentan listas ordenadas mediante el interés calculado para cada ítem u F.C., obtenido en base a las predicciones

realizadas por el algoritmo; además, se muestra un porcentaje de confianza para la recomendación que tiene en cuenta dos factores: la varianza para la predicción concreta y el número de elementos que se utilizaron para elaborar la predicción, es decir, el número de materias similares usadas. Al recomendar las materias propias de modalidad, se muestran todas las modalidades, lo que favorece que el alumno pueda evaluar completamente las alternativas. Con respecto a las materias que requieren refuerzo, el cálculo es similar, aunque se muestran únicamente aquellas asignaturas cuya predicción es menor o igual que cuatro.

A diferencia de nuestro trabajo, este sistema sólo usa los datos de las calificaciones obtenidas por los estudiantes para predecir sus notas futuras en las asignaturas que aún no han cursado. Esto tiene la dificultad de que la vecindad se forma en base a las notas obtenidas y que, por lo tanto, no considera la dificultad de la asignatura ni el potencial del estudiante para la asignatura a cursar. Orieb es radicalmente diferente al sistemas de predicción propuesto en esta investigación, debido a que, en el caso del cuarto año de secundaria y de los dos años del bachillerato, un estudiante debe cursar un número mínimo de asignaturas; sin embargo, en el nivel universitario, esto no es cierto y es por ello que se usa una medida de carga académica llamada “créditos de asignatura” y “créditos matriculados”.

Se sabe que los algoritmos de filtrado colaborativo trabajan de dos maneras: los basados en memoria, que se establecen en una vecindad completa de usuarios y sus valoraciones para el cálculo de predicciones, y los basados en modelos, que usan esas valoraciones para aprender un modelo que será el empleado para predecir. En nuestro caso, a partir de los datos, aplicamos técnicas de minería para construir modelos del rendimiento de los estudiantes y, además, combinamos los modelos con los cálculos de las predicciones basadas en memoria.

Capítulo 3:

Sistemas de Recomendación

Los sistemas de recomendación, como su nombre lo indica, son aplicaciones de software que proveen recomendaciones personalizadas a usuarios acerca de productos o servicios que pueden interesarles, según las preferencias que ellos han expresado tanto implícita como explícitamente. Dicho de otro modo, los sistemas de recomendación son programas que crean un modelo de preferencias del usuario con el propósito de satisfacer sus intereses. Por ello, se dice que son sistemas de información diseñados para datos no estructurados o semiestructuradas. Están basados en descripciones de preferencias de personas o grupos comúnmente llamados “perfiles”.

En su formulación más común, el problema de la recomendación se reduce al problema de estimar *ratings* (en adelante, “evaluaciones”) para objetos que aún no han sido experimentados por un usuario. Intuitivamente, esta estimación está basada, por lo general, en las evaluaciones dadas por este usuario a otros objetos y en alguna otra información adicional que más adelante se describirá.

La recomendación no es un fenómeno nuevo propio de la era digital; es, sobre todo, un comportamiento social existente en la vida real [Tse-03]. Diariamente, confiamos en recomendaciones de la gente, ya sea a través de cartas de recomendación, resúmenes de películas o cuestionarios. Con la introducción de la Web, este proceso ha ido creciendo y actualmente los sistemas de recomendación asisten a las personas a fin de que puedan tomar decisiones más acertadas sobre sus preferencias.

Se estima que Internet contiene en la actualidad cientos de terabytes de datos de textos, gráficos, videos; debido a esto, el problema ya no es la falta de información, sino la sobrecarga. Cada vez es más complicado para un individuo elegir algo que satisfaga completamente sus necesidades por la distorsión que genera la ausencia de jerarquías entre la información recibida. Para solucionar ese problema, primero se utilizaron motores de búsqueda, pero estos, en la actualidad, devuelven una cantidad muy grande de información; de modo que, frente a este inconveniente, los expertos optaron por usar algunas técnicas de filtrado con las cuales acercarse a un conjunto cada vez más pequeño de información relevante. Además, dicha información se puede personalizar: el usuario podrá así obtenerla de un sistema que, de forma inteligente, maneje el perfil histórico, los intereses y gustos del usuario para, automáticamente, sugerirle objetos semejantes.

Los métodos de filtrado de la información fueron los más usados para los sistemas de recuperación. De la misma manera que los sistemas convencionales, tienen el objetivo de seleccionar ítems, documentos o, en general, aquello que satisfaga las necesidades de información del usuario. Sin embargo, los sistemas de recuperación

usualmente son diseñados para facilitar una búsqueda rápida de la información a fin de satisfacer necesidades a corto plazo de un individuo o población de usuarios muy diversa. En cambio, los sistemas de recomendación están hechos para soportar necesidades de información a largo plazo de un usuario o grupo de usuarios con necesidades similares. Otra diferencia fundamental es que los primeros operan con un conjunto de documentos estáticos, mientras que los segundos tienen el objetivo de identificar documentos relevantes de una base que cambia constantemente.

Se distinguen dos tipos de sistemas de recomendación de acuerdo con la manera en la que recomiendan un ítem para diferentes usuarios. Los sistemas de recomendación basados en contenidos, que le recomiendan al usuario ítems similares a los que él prefirió en el pasado, y los sistemas de filtrado colaborativo, que le recomiendan objetos al usuario, que generalmente recibe el nombre de "usuario activo", según los gustos y preferencias de otros usuarios con los que los comparta. Debido a las limitaciones de cada una de las estrategias antes mencionadas, han surgido otras de la combinación entre ellas, llamadas "sistemas de recomendación híbridos", que han demostrado empíricamente mayor efectividad.

En este capítulo, se desarrollará un resumen de la terminología y técnicas relacionadas con los sistemas de recomendación. En la sección 3.1, se formalizará el concepto de recomendación, se describirán los componentes básicos y generales de estos sistemas. Las diferentes aproximaciones, basadas en contenidos, el filtrado colaborativo y las hibridas, se revisan en las secciones 3.2, 3.3, 3.4, respectivamente. Para cada una de ellas, se presentan sus formas de uso, algunos ejemplos y sus limitaciones.

3.1. Sistemas de recomendación

El problema de recomendación puede ser formulado como sigue [Adom-05]. Sea $U = (u_1, u_2, \dots\dots\dots u_M)$ un conjunto de todos los usuarios registrados en un sistema de recomendación, y sea $I = (i_1, i_2, \dots\dots\dots i_N)$ el conjunto de todos los posibles ítems accesibles por los usuarios desde el sistema y que, además, pueden ser recomendados. Sea $g: UxI \rightarrow R$, donde R es un conjunto totalmente ordenado, una función de utilidad tal que$g(u_m, i_n)$ mide la ganancia o utilidad del ítem i_n para el usuario u_m. Entonces, para cada usuario $u_m \epsilon\, U$, queremos elegir un ítem $i^{max\,,\ u_n \epsilon}$ I, desconocido para el usuario, que maximice la función utilidad g. Formalmente tenemos:

$$\forall\, u_m \,\epsilon\, U,\ \ i^{max\,,\ u_m} = arg\max_{i_n\, \epsilon\, I} g(u_m, i_n) \quad \text{..... (3.1)}$$

La utilidad de un sistema de recomendación se representa por medio de un número llamado "evaluaciones", que mide cuánto está interesado un usuario específico

en un ítem. Dependiendo de la aplicación, las evaluaciones pueden ser especificadas directa o indirectamente por los usuarios, o calculadas por el sistema.

Cada elemento del conjunto U de usuarios puede ser descrito por medio de un perfil, el cual puede incluir algún tipo de información demográfica de usuario, como edad, género, nacionalidad, etc., y/o alguna información acerca de gustos, intereses y preferencias. Análogamente, cada elemento del conjunto I puede describirse con un conjunto de características. Por ejemplo, en un sistema recomendador de películas, donde I es el conjunto de todas ellas, cada película puede ser representada no solo por su ID, sino también por su título, género, director, año, etc.

La forma en la cual el perfil del usuario y la descripción de los ítems se definen es un punto clave de cualquier sistema recomendador. Sin embargo, este no es el único factor que influencia su eficiencia y efectividad. Por ejemplo, el mecanismo que captura las preferencias de los usuarios es crítico. Generalmente, los usuarios no están dispuestos a perder su tiempo llenando encuestas para informar al sistema sobre sus gustos o preferencias; por otro lado, si se capturan automáticamente, se tiende a generalizar estas características, lo que vuelve a los sistemas cada vez más ineficientes.

En la figura 3.1, se muestran los componentes básicos de un sistema recomendador. En primer lugar, el sistema debe capturar los gustos y preferencias de los usuarios de forma explícita o implícita. Una vez que el sistema "sabe" los gustos y preferencias del usuario, ejecuta algún algoritmo que usa (compara y/o combina) el perfil del usuario y la descripción de los ítems. Encontradas (extrapoladas) las evaluaciones para cada usuario o ítem, según sea el caso, se guardan en una base de datos. Es importante señalar que no todos los elementos guardados son aptos para ser usados, ello depende de la estrategia de recomendación. Finalmente, el sistema recomienda al usuario, según los datos y el algoritmo usado, lo que debería elegir.

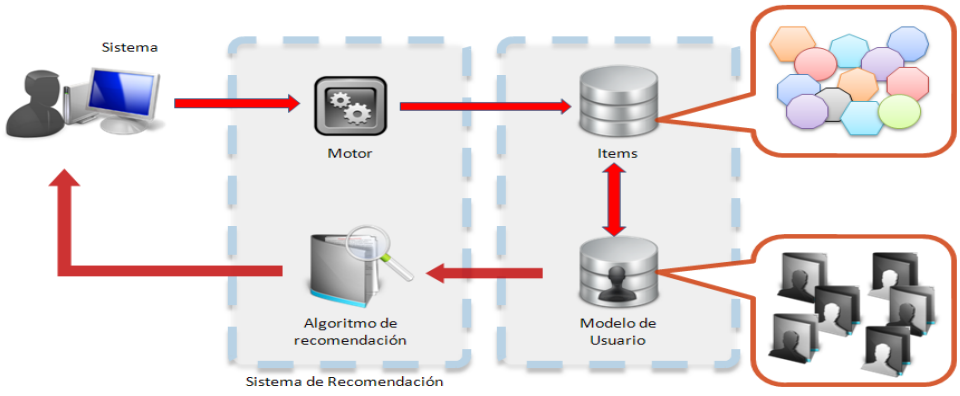

Figura 3.1. Componentes de un sistema de recomendación

La principal dificultad en todos los sistemas de recomendación recae en el hecho de que la función utilidad g no está usualmente definida en todo el espacio UxI, sino en un

subconjunto de este. Esto significa que g necesita ser extrapolado al espacio UxI. Así, por ejemplo, en los sistemas de recomendación, la función utilidad es representada por ranking y está inicialmente definida solo en los ítems previamente ranqueados por los usuarios.

La extrapolación mencionada de rankings conocidos a desconocidos se lleva a cabo siguiendo alguna de las dos diferentes aproximaciones [Bree-98]: especificar heurísticas que definen la función utilidad y validar empíricamente su rendimiento, o establecer modelos que estimen la función utilidad optimizando ciertos criterios de rendimiento -tales como Error Cuadrático Medio (MSE)- entre los rankings conocidos y predichos. En ambos casos, cuando los rankings desconocidos son estimados, la recomendación se realiza seleccionando los objetos con el más alto ranking, de acuerdo con la expresión (3.1). Opcionalmente, también se pueden recomendar los N mejores objetos a un usuario o a un conjunto de usuarios.

Los nuevos rankings de los objetos aún no ranqueados también se pueden estimar de muchas otras maneras; por ejemplo, usando métodos de aprendizaje automático, teoría de la aproximación y heurísticas en general. Los sistemas de recomendación algunas veces se clasifican según la aproximación que se emplea para estimar rankings. En el caso de esta memoria, se describirá la clasificación de los sistemas de recomendación de acuerdo con la manera en que se hace la recomendación [Bala-97]:

- Sistemas de recomendaciones basados en contenidos: A los usuarios se les recomienda ítems similares a los que ellos prefirieron usar en el pasado.
- Sistemas de recomendación colaborativos: A cada usuario se le recomiendan objetos que otros usuarios, con gustos y preferencias similares, han usado en el pasado.
- Sistemas de recomendación híbridos: Debido a los defectos de cada una de las estrategias anteriores, este método combina técnicas colaborativas y basadas en contenidos.

3.2 Sistemas de recomendación basados en contenidos

Este tipo de sistemas de recomendación está basado en el hecho de que un usuario o consumidor prefiere objetos con características similares a aquellos que él mismo usó en el pasado [Terv-01]. Así la función de utilidad $g(u_m, i_n)$ del ítem $i_n \in I$ para el usuario $u_m \in U$ se estima según la utilidad $g(u_m, i_j)$ asignada por u_m al ítem i_j , similar al ítem i_n . En este tipo de sistemas de filtrado, los objetos se seleccionan por correlación entre las preferencias del usuario y el contenido de los objetos. Estos deben estar de alguna forma tal que puedan ser analizados automáticamente, de modo que se puedan

detectar contenidos que el usuario clasificó previamente [Spec-00]. Este recibe el nombre de “filtrado por contenido”, porque es precisamente eso lo que se analiza de las fuentes de información que han sido evaluadas para crear un perfil de los intereses de los consumidores; en otras palabras, los objetos se definen por sus características asociadas. En el caso particular de un texto, el sistema en cuestión considera sus palabras como una característica propia. Un sistema de este tipo aprende un perfil de los intereses del usuario según las características presentes en los objetos evaluados anteriormente.

Por ejemplo, para sugerir películas a un usuario u_m , un sistema recomendador basado en contenidos busca las características comunes entre las películas que u_m ha evaluado positivamente en el pasado. Entonces, solo aquellas que han tenido un alto grado de similitud con las que el usuario prefirió serán recomendadas.

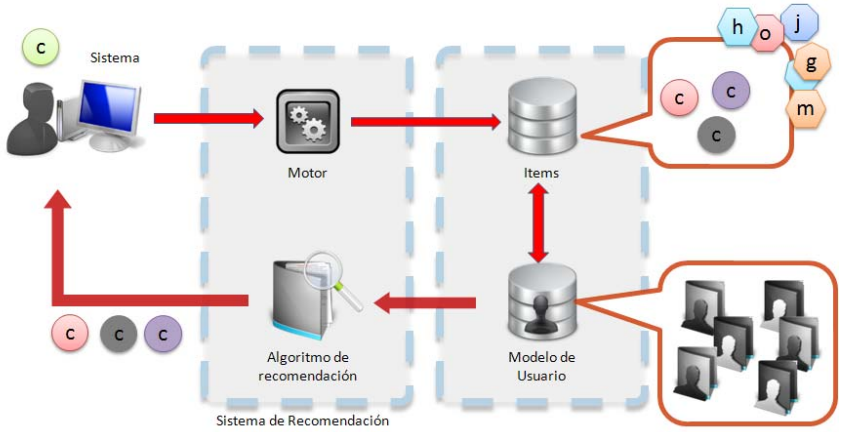

Figura 3.2. Recomendación basada en contenidos

En la figura 3.2, se muestra el proceso completo seguido por un sistema recomendador basado en contenidos. Primero, captura las preferencias del usuario (quien va ser recomendado) y construye su perfil personal. Después, cuando la recomendación deba producirse, las preferencias reunidas en el modelo del usuario se comparan con las características del ítem almacenado en el sistema; el objeto cuya característica es la más similar a las preferencias del usuario se extrae y se presenta como recomendación. Se debe notar que, en este escenario, solo los objetos que comparten rasgos basados en contenidos con el perfil del usuario podrán ser recomendados, lo cual, en la práctica, reduce de manera significativa el conjunto de ítems con dicha condición.

Los sistemas basados en contenidos se originan en los sistemas de recuperación de la información. La mejora de los primeros sobre los segundos proviene del uso de perfiles de usuario que contienen información acerca de gustos, preferencias y necesidades. Esta información se puede obtener implícitamente (es decir, aprendida a

partir de la navegación, compras y/o usos que el usuario tenga con el sistema) o explícitamente (a través de cuestionarios).

Formalmente, y siguiendo la notación usada en [Adom-05], sea $C(i_n)$ la descripción del contenido de ítem $i_n \epsilon I$, esto es, el perfil del ítem o el conjunto de los atributos que caracterizan a i_n. Generalmente, este se calcula extrayendo un conjunto de características del objeto i_n y se emplea para determinar cuán apropiado es el ítem para poder recomendarlo. Esta descripción es representada como un vector de números reales (pesos), en el cual cada componente mide la importancia (o información) de las características de los descriptores del ítem.

$$C(i_n) = \ i_n \ = \ (i_{n,1}, i_{n,2}, \dots\dots\dots\dots\dots i_{n,k}) \ \epsilon \ R^k \text{.....} \textit{(3.2)}$$

Análogamente, sea CBPU (el perfil del usuario basado en contenido); $CBPU(u_m)$ las preferencias de los usuarios $u_m \epsilon \ U$ basadas en contenidos, esto es, la ponderación (peso) de las características de los contenidos del ítem que describen los gustos, intereses y necesidades del usuario.

$$CBUP(u_m) = \ u_m \ = \ (u_{m,1}, u_{m,2}, \dots\dots\dots\dots\dots u_{m,k}) \ \epsilon \ R^k \text{ ..... } \textit{(3.3)}$$

La ganancia o utilidad del ítem i_n para el usuario u_m se calcula como una función que combina los diferentes descriptores del ítem y los componentes del perfil del usuario.

$$g(u_m, i_n) = \ score \ (CBUP(u_m), C(i_n)) \ \epsilon \ R \text{.....} (3.4)$$

En el caso de que $C(i_n)$ represente el contenido de un documento, el paradigma de recuperación de la información propone que ambos, $C(i_n)$ y $CBUP(u_m)$, pueden ser representados usando la técnica Frequency/Inverse Document Frequency (TF-IDF) [Baez-99] En los ambientes de recuperación de la información, la medida de TF-IDF se define como sigue: se asume que N es el número total de documentos que pueden ser recomendados a usuarios y que la palabra clave k_j aparece en n_i de ellos. Más aún, asumimos que $f_{i,j}$ es el número de veces que la palabra clave k_i aparece en el documento d_j. Entonces $TF_{i,j}$, el término frecuencia (o frecuencia normalizada) de la palabra clave k_i en el documento d_j se define de la siguiente manera:

$$TF_{i,j} \ = \ \frac{f_{i,j}}{max_z f_{z,j}} \text{.....} \textit{(3.5)}$$

Donde el máximo se calcula sobre la frecuencia $f_{z,j}$ de todas las palabras clave k_z que aparecen en el documento d_j. Sin embargo, no todas estas son útiles para distinguir entre documentos relevantes e irrelevantes. Por lo tanto, la medida de $IDF_{i,j}$ es frecuentemente usada en combinación con $TF_{i,j}$ y se define como sigue:

$$IDF_i = \log \frac{N}{n_i} \text{.....(3.6)}$$

Entonces, el peso TF-IDF para la palabra clave k_j en el documento d_j se define de esta manera:

$$w_{i,j} = TF_{i,j} \; x \; IDF_i \text{.....(3.7)}$$

Además, el contenido del documento d_j se define como sigue:

$$C(d_j) = \left(w_{1,j}, w_{2,j}, \ldots\ldots\ldots\ldots\ldots\ldots w_{k,j}\right) \text{.....(3.8)}$$

Lo que quiere decir que ambos, $C(i_n)$ y $CBUP(u_m)$, se pueden simbolizar por vectores pesos w_c y w_s, más aún la función $g(c,s)$ es usualmente representada en la literatura de recuperación de la información como una heurística generalmente asociada con la medida de similitud del coseno.

$$g_{c,s} = \cos(w_c, w_s) = \frac{w_c . w_s}{\|w_c\|_2 \; x \; \|w_s\|_2} = \frac{\sum_{i=1}^{K} w_{i,c} w_{i,s}}{\sqrt{\sum_{i=1}^{K} w_{i,c}{}^2} \sqrt{\sum_{i=1}^{K} w_{i,s}{}^2}} \text{..... (3.9)}$$

Donde K es el número total de palabras en el sistema.

La forma de la expresión 3.4 permite distinguir las diferentes técnicas de recomendación basadas en contenidos propuestas en la literatura actual. Como se sostuvo anteriormente, estas técnicas pueden ser clasificadas según la heurística y según ciertos modelos. La primera calcula la predicción de la función utilidad a partir de fórmulas heurísticas que están inspiradas, mayormente, en métodos de recuperación de la información tales como la medida de la similitud de cosenos. La segunda, por otro lado, obtiene sus predicciones a partir de un modelo aprendido mediante datos subyacentes, usando aprendizaje estadístico y modelos de aprendizaje automático, entre los que pueden contarse clasificadores bayesianos, algoritmos de *clustering*, árboles de decisión y redes neuronales artificiales.

3.2.1. Limitaciones de los sistemas de recomendación basados en contenidos

Los sistemas de recomendación basados en contenidos presentan muchas limitaciones, las cuales han sido identificadas en las referencias [Bala-97], [Burk-02], [Adom-05], y se describen a continuación.

- **Análisis de contenido restringido**

  Las recomendaciones basadas a partir de los contenidos están restringidas por las características que están explícitamente asociadas con los ítems que serán recomendados. Por ejemplo, en el caso de las películas, las recomendaciones pueden basarse solamente en características como el nombre de los actores, resúmenes, géneros, etc.

  La efectividad de esta técnica depende de los datos descriptivos disponibles. Existen muchos dominios de aplicación en los que la extracción automática de las características es muy difícil de realizar, y asignar características normalmente es muy poco práctico. Por ejemplo, es mucho más complicada la extracción automática de características a datos multimedia (como imágenes, video y audio) que a los de contenido textual.

  De igual modo, nunca podrá distinguirse si dos ítems están representados por las mismas características. Así, si dos textos comparten las mismas palabras clave, los sistemas basados en contenidos no podrán diferenciarlos.

- **Sobreespecialización**

  Los sistemas de recomendación basados en contenidos pueden recomendar solamente objetos altamente valorados por un usuario, dado su perfil; por ello, los usuarios se ven restringidos a buscar artículos similares a aquellos que ya han sido estimados. Esto quiere decir que ven solo los contenidos que ya se han observado antes. Además, el problema de la sobreespecialización no se produce solamente porque el sistema de recomendación no pueda recomendar ítems que son diferentes a los que el usuario ha visto antes. En algunos casos, este tipo de sistemas no recomienda objetos si ellos son muy similares a algo que el usuario ha visto con anterioridad, como, por ejemplo, algún artículo diferente que tenga usos similares.

- **Problema del nuevo usuario**

  El usuario tiene que haber estimado un número suficiente de objetos antes de que este tipo de sistema pueda realmente entender sus preferencias y presentarle recomendaciones confiables. Así, un usuario nuevo con muy pocas evaluaciones no podría conseguir recomendaciones precisas.

3.2.2. Ejemplos de los sistemas de recomendación basados en contenidos

	Nombre	Principales características
Sistemas de recomendación basados en contenidos	***LIBRA***	[Moon-98]*;* [Moon-00] Es un sistema recomendador de libros basado en contenidos que utiliza información semiestructurada acerca de textos provenientes de la Web. Las características que representan los libros están estructuradas como un conjunto de palabras equivalentes al título, al autor, al tema, etc. Con ellas, un clasificador bayesiano aprende y, así, el sistema tiene la capacidad de explicar su recomendación listando las características que más contribuyen.
	New Dudes	[Pazz-99]; [Bill-00] Es un agente que usa habla sintetizada para leer historias a los usuarios. Estas se recomiendan de acuerdo con un modelo de intereses de corto y largo plazo. Las preferencias son obtenidas tomando en cuenta no solo las evaluaciones de los usuarios, sino también el tiempo que ellos invierten escuchando la historia. Para determinar las recomendaciones a corto plazo, las historias se describen en términos del vector TF-IDF, el cual se compara con la similitud del coseno, y suministrados al módulo según el algoritmo de KNN. Por otro lado, para establecer la recomendación a largo plazo, las historias se representan como vectores de características booleanas, donde cada una de ellas indica la presencia o ausencia de la característica
	Syskill & Webert	[Pazz-97] Es un recomendador de páginas electrónicas que maneja perfiles del usuario que pueden determinar qué sitios de la Web (dado el tema) serían interesantes para él. Cada usuario tiene un conjunto de perfiles, uno para cada tema; el sistema identifica las 128 palabras más informativas de la página. Hace uso de un clasificador bayesiano que puede obtener perfiles a partir de una retroalimentación con el usuario y los sitios que visite.
	Info Finder	[Krul-97] Es un sistema recomendador de mensajes basado en contenidos que aprende la información de interés del usuario a partir de un conjunto de mensajes y otros documentos en línea clasificados por los usuarios. Específicamente, el sistema los utiliza para efectuar búsquedas y consultas de sus categorías personales que ejecuta para que pueda recomendar regularmente nuevos documentos. Para construir estas consultas, InfoFinder extrae semánticamente frases significantes de cada documento usando varias heurísticas basadas en características visuales también significantes, construye árboles de decisión [Duda-01] con las frases identificadas y los transforma en nuevas consultas.

Tabla 3.1. Sistemas de recomendación basados en contenidos

3.3. Sistemas de filtrado colaborativo

El Filtrado Colaborativo (FC) es una técnica empleada para la recomendación y predicción de decisiones. Su finalidad es sugerir nuevos objetos o predecir su utilidad a un usuario en particular empleando como referencia las preferencias de otros usuarios

que presentan características similares y que previamente han usado o valorado dichos objetos, es decir, las preferencias de un grupo de personas cuyas decisiones muestran tendencias similares a las del usuario evaluado. Esto supone que personas con apreciaciones similares sobre un producto o servicio también elegirán de manera similar en sus futuras decisiones. Dicho en términos más sencillos, su meta principal es la automatización del proceso de recomendación comúnmente llamado de "boca a boca", por el que la gente sugiere productos o servicios según sus propias experiencias. Este proceso parte de la constatación de que el usuario que necesita elegir entre varias opciones en las que no cuenta con experiencia confía en las opiniones de aquellos que sí la poseen.

El FC puede darnos resultados de dos tipos, que dependerán de su uso u objetivos:

- Predicciones.- Se utiliza la opinión de todos los usuarios similares al evaluado con el objetivo de presentar un solo valor final como resultado. Es un número que expresa la predicción de un ítem para un usuario determinado.
- Recomendaciones.- Se utiliza la opinión de todos los usuarios similares al evaluado con el objetivo de presentar una lista de N valores posibles como resultado, considerando que, dentro de esa lista N, no deben estar contenidos los ítems evaluados por el usuario con anterioridad.

En un típico sistema de FC, existe un conjunto de m usuarios $U = \{u_1, u_2, \ldots\ldots\ldots\ldots\ldots\ldots u_m\}$ y n objetos $I = \{i_1, i_2, \ldots\ldots\ldots\ldots\ldots\ldots i_n\}$. Cada usuario tiene una lista de objetos $I_{u_i} \subseteq I$, en la cual ha registrado sus preferencias, donde existe la posibilidad de que sea un conjunto vacío cuando no han sido evaluados. En el caso de una recomendación, la lista está conformada por los N-objetos $I_r \subset I$ adecuados para el usuario activo. Se debe notar que $I_r \cap I_{u_a} = \emptyset$, lo que significa que ningún elemento del conjunto I_r ha sido comprado por el usuario.

La tarea del FC es encontrar un ítem para un usuario $u_a \in U$, llamado "usuario activo", según los ítems previamente evaluados por otros usuarios. La predicción es un valor numérico $P_{a,j}$ que expresa en qué magnitud el ítem i_j es adecuado al usuario u_a $i_j \notin I_{u_a}$, y debe estar en la misma escala que usó u_a para dar sus evaluaciones. Esta predicción está representada por una función de utilidad $g(u_m, i_n)$ del ítem $i_n \in I$ para un usuario $u_m \in U$, estimada a partir de una función utilidad $g(u_j, i_n)$ asignada al ítem i_n por los usuarios u_j similares al usuario u_m.

En la figura 3.3, se muestra el esquema del proceso de FC, donde la matriz mxn representa la matriz de evaluaciones A. Cada elemento $a_{i,j}$ es un número que representa la opinión o el puntaje que consideró el i-ésimo usuario para el j-ésimo objeto. Cada evaluación individual está dentro de una escala numérica. Generalmente, en los

sistemas de recomendación colaborativos, se usa la escala del uno al cinco, donde el cero indica que el usuario no ha dado su evaluación.

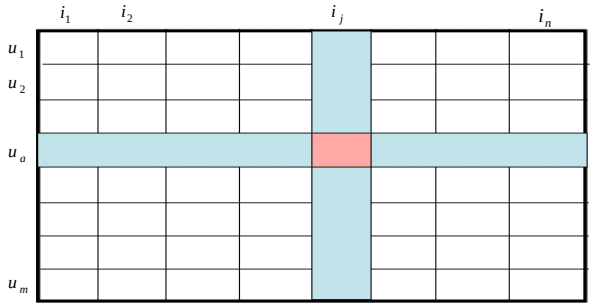

Figura 3.3. Matriz de evaluaciones usuario – ítem

A diferencia del método basado en contenidos, los sistemas de filtrado colaborativo predicen la utilidad del ítem para un usuario en particular de acuerdo con la evaluación previa hecha por otros usuarios. En este tipo de filtrado, los usuarios expresan sus preferencias evaluando ítems. Esas evaluaciones son tomadas como una representación aproximada de sus gustos, intereses y necesidades en el dominio de aplicación. Asimismo, son comparadas con las enviadas por otros usuarios; de este modo, se encuentra el conjunto de usuarios llamado “vecinos cercanos”. A partir de esto, los ítems que fueron estimados altamente por los vecinos cercanos al usuario, más no por este, finalmente son recomendados.

En la recomendación, estos sistemas forman una lista de N ítems, $\mathrm{I_r} \sqsubset \mathrm{I}$, que podrían gozar de la mayor preferencia de los usuarios activos, con la condición de que $\mathrm{I_r} \cap \mathrm{I_{u_a}} \neq \oslash$. Este proceso para los algoritmos de filtrado colaborativo se conoce con el nombre de recomendaciones top-N.

Todas estas aproximaciones comparten definiciones comunes para el perfil del usuario y la descripción del ítem, pero se diferencian de los sistemas basados en contenido. Formalmente, si se representa el perfil del usuario de la forma CUP (perfil colaborativo del usuario), sea $CUP(u_m) = r_m = (r_{m,1}, r_{m,2}, \dots\dots\dots\dots r_{m,n}) \epsilon R^n$ el perfil colaborativo del usuario u_m constituido por un conjunto de evaluaciones dadas por el usuario u_m a los n ítems del sistema, y sea $R(i_n) = r_n = (r_{1,n}, r_{2,n}, \dots\dots\dots\dots r_{m,n}) \epsilon R^m$ el conjunto de evaluaciones $r_{m,n} \epsilon\, R$ asignadas al ítem i_n por los m usuarios registrados en el sistema. En ambas definiciones, si el usuario u_m no ha evaluado el ítem i_n, entonces el valor de $r_{m,n} = 0$. La ganancia o utilidad del ítem i_n para el usuario u_m se calcula como una función que combina los diferentes descriptores del ítem y los componentes del perfil del usuario.

$$g(u_m, i_n) = \ score\ (CUP(u_m), R(i_n))\ \epsilon\ \ R \ \text{.....(3.10)}$$

La forma en la cual la expresión previa se formula permite distinguir las diferentes técnicas propuestas en el campo. La primera distinción es la que clasifica la técnica de filtrado colaborativo en basadas en usuarios e ítems

A continuación, examinaremos dos clases generales de algoritmos de filtrado colaborativo.

3.3.1. Basado en el usuario

El filtrado colaborativo basado en usuario y el basado en ítem están dentro del conjunto de técnicas conocidas como las técnicas de filtrado colaborativo basado en memoria (memory based). Estas usan bases de datos para generar predicciones. Para ello primero se emplean técnicas estadísticas que permiten encontrar un conjunto de ítems similares, conocidos como vecindades.

Los sistemas de filtrado colaborativo basados en usuarios, también llamado “vecinos cercanos” o “filtrado colaborativo basado en usuarios”, son los más populares en la práctica, estos comparan las evaluaciones del usuario activo (el que pide la recomendación) con las de otros usuarios a fin de identificar un grupo de gente similar a él, de tal manera que el ítem con evaluación máxima (y, además, compartida por la mayor cantidad de usuarios) sea el recomendado. En términos más simples, sugiere que el usuario que eligió el ítem A debería estar interesado en el ítem B si otros usuarios que eligieron A también estuvieron interesados en el ítem B.

Este sistema utiliza la base de datos de usuario-ítem para generar predicciones, y emplea técnicas estadísticas para encontrar un conjunto de usuarios, conocido como “vecindades”, que tienen un histórico de preferencias similares (tienden a evaluar de manera similar diferentes ítems o compran un conjunto similar de ítems). Cuando se forma el vecindario de usuarios, el sistema usa diferentes algoritmos para combinar las preferencias de los vecinos, a partir de las cuales puedan obtener una predicción o una lista de recomendaciones top-N para el usuario activo.

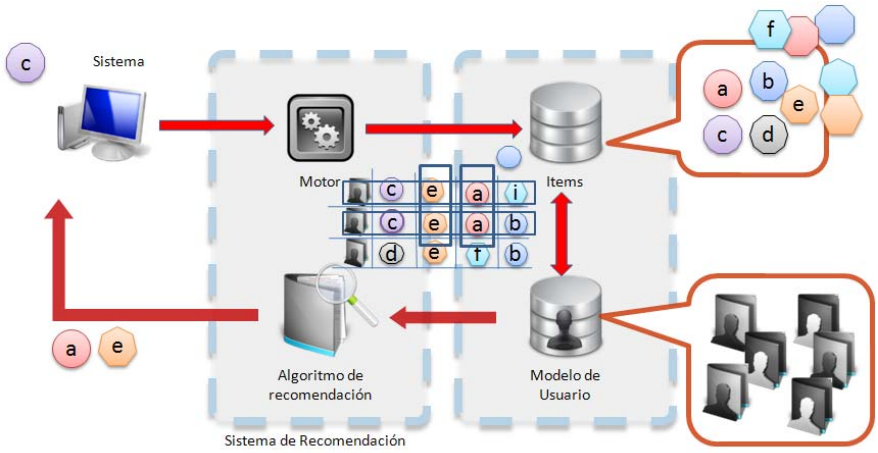

Figura 3.4. Proceso de recomendación que usa filtrado colaborativo basado en usuarios

En la figura 3.4, se muestra un proceso típico de recomendación de sistemas de filtrado colaborativo basado en el usuario. La base de datos consiste en una lista de ítems que han sido seleccionados, evaluados o comprados por el usuario (el resto de los ítems permanecen invisibles al sistema)[1]. En seguida, los gustos o preferencias se capturan observando la elección o las evaluaciones que este haya hecho. Cada elección o evaluación es guardada en su perfil, lo que crea un historial en forma de tabla.

Para generar la recomendación, el algoritmo correlaciona esas evaluaciones o elecciones con la lista de cada uno de los otros usuarios que están registrados en el sistema, y selecciona el grupo con más alta correlación. Luego, el sistema crea una lista de ítems elegidos o evaluados por los usuarios identificados por el sistema (aquellos con gustos muy parecidos), y la categoriza por su frecuencia o por su evaluación. Finalmente, se recomiendan los ítems mejor evaluados y con mayor número de frecuencia de aparición.

Como se mencionó, los algoritmos de filtrado colaborativo basado en los usuarios predicen las evaluaciones según el conjunto de todos los ítems evaluados por dichos usuarios previamente. Presentado de manera más formal, la función de utilidad $g(u_m, i_n)$ del ítem $i_n \epsilon I$ para un usuario $u_m \epsilon U$ representa dicha predicción y se calcula como un agregado de las evaluaciones $r_{j,n}$ de algún otro usuario u_j para el mismo ítem i_n:

$$g(u_m, i_n) = agre_{u_j \epsilon \hat{U}_m} r_{j,n} \ldots\ldots\ldots\ldots\ldots\ldots (3.11)$$

Donde $\hat{U}_m$ es el conjunto de los N usuarios más similares al usuario u_m (los que han evaluado al ítem i_n). El valor de N puede variar desde 1 hasta el número total de

[1] Se debe tener en cuenta que nunca recomendará ítems previamente elegidos por el usuario.

usuarios registrados en el sistema. Algunos ejemplos para la función de utilidad se muestran en [Adom-05] y se detallan a continuación:

a) $g(u_m, i_n) = \frac{1}{|\hat{U}_m|} \sum_{u_j \epsilon \hat{U}_m} r_{j,n}$

b) $g(u_m, i_n) = k \sum_{u_j \epsilon \hat{U}_m} sim(u_m, u_j) \text{ X } r_{j,n}$ .....*(3.12)*

c) $g(u_m, i_n) = \bar{r}_m + k \sum_{u_j \epsilon \hat{U}_m} sim(u_m, u_j) \text{ X } (r_{j,n} - \bar{r}_j)$

Donde k se considera como factor de normalización y es definido como:

$$k = \frac{1}{\sum_{u_j \epsilon \hat{U}_m} |sim(u_m, u_j)|} \text{.....} (3.13)$$

y donde el factor de evaluación promedio del usuario u_n , $\bar{r}_m$, en 3.12(C) es definido como:

$$\bar{r}_j = \frac{1}{|I_j|} \sum_{I_n \epsilon I_j} r_{j,n} \quad , \text{ donde } I_j = \{ i_n \epsilon I / r_{j,n} \neq 0 \} \text{.....(3.14)}$$

El caso más simple para el cálculo de la función de utilidad puede considerar la función de agregación como un simple promedio (3.12a). Sin embargo, la forma más común es la suma ponderada mostrada en (3.12b). La medida de similitud entre los usuarios u_m y u_j es, en esencia, la medida de una distancia, y se usa como una ponderación. Esto significa que los usuarios u_m y u_j más similares son aquellos que alcanzan una mayor ponderación en la evaluación $r_{j,n}$(que hace el usuario u_j para el ítem i_n) para predecir la evaluación que hará el usuario $u_m \epsilon U$ para el ítem $i_n \epsilon I$ que se mide por intermedio de $r_{i,m}$.

- **Cálculo de la similitud**

Se debe observar que la similitud $sim(x, y)$ es una heurística que se introduce a fin de diferenciar distintos niveles de similitud entre usuarios, es decir, con la finalidad de encontrar “pares más cercanos” o “vecinos más próximos” para cada usuario y, al mismo tiempo, simplificar el procedimiento de estimación de evaluaciones. El objetivo principal de la formación del vecindario es encontrar, para cada usuario u_m, una lista ordenada $\hat{U}_m$ de los N usuarios más similares tales que $u_m \notin \hat{U}_m$ y las similitudes resultantes mantengan algún orden (Top-N).

Se han usado muchas aproximaciones para calcular la similitud $sim(u_m, u_j)$ entre dos usuarios en sistemas de recomendación colaborativos. En numerosos casos, la similitud entre dos usuarios u_m y u_j se basa en las evaluaciones de los ítems hechas por ellos mismos. Las dos aproximaciones más populares son la de correlaciones y la basada en cosenos.

Sea $I_{m,j} = \{i_n \epsilon \ I / r_{m,,n} \neq , r_{j,n} \neq 0\}$ el conjunto de ítems evaluados por ambos usuarios que luego se tomarán en cuenta para el cálculo de la similitud. En muchas de las situaciones usadas en la empresa o en el ámbito comercial, los puntajes se consideran entre el cero y el cinco. En este trabajo de investigación, los puntajes serán considerados desde el cero hasta el veinte por tratarse de las evaluaciones obtenidas por los estudiantes:

- Cálculo de la similitud basada en coseno: [Bree-98]; [Sarw-01].

  Esta medida establece la similitud entre dos usuarios u_m y u_j calculando el coseno del ángulo formado entre sus dos vectores de evaluaciones:

$$r_m = (r_{m,1}, r_{m,2}, \dots \dots . r_{m,N}) \quad \text{y} \quad r_j = (r_{j,1}, r_{j,2}, \dots \dots . r_{j,N})$$

$$Sim(u_m, u_j) = cos(r_m, r_j) = \frac{r_m . r_j}{\|r_m\| x \| r_j\|} =$$

$$\frac{\sum_{i_n \in I_{m,j}} r_{m,n} . r_{j,n}}{\sqrt{\sum_{i_n \in I_{m,j}} r_{m,n}^2} \sqrt{\sum_{i_n \in I_{m,j}} r_{j,n}^2}} \dots\dots\dots\dots\dots (3.15)$$

- Cálculo de la similitud basada en correlación: [Resn-94]; [Shar-1995].

  Esta medida establece la similitud entre dos usuarios u_m y u_j calculando el coeficiente de correlación de Pearson:

$$r_m = (r_{m,1}, r_{m,2}, \dots \dots . r_{m,N}) \quad \text{y} \quad r_j = (r_{j,1}, r_{j,2}, \dots \dots . r_{j,N})$$

$$Sim(u_m, u_j) = \frac{\sum_{i_n \in I_{m,j}} (r_{m,n} - \overline{r_m})(r_{j,n} - \overline{r_j})}{\sqrt{\sum_{i_n \in I_{m,j}} (r_{m,n} - \overline{r_m})^2} \sqrt{\sum_{i_n \in I_{m,j}} (r_{j,n} - \overline{r_j})^2}} \dots\dots\dots (3.16)$$

  Donde las sumatorias de n son de los ítems que los dos usuarios m y j han evaluado, y donde $\overline{r_m}$ es el promedio de las evaluaciones hechas a los ítems por el m-ésimo usuario.

  Se puede utilizar otras correlaciones como *Constrained Pearson Correlation*, que es una variación de la correlación de Pearson, que usa un punto medio en vez de una tasa promedio. También se puede usar *Spearman Rank Correlation,* que es similar a la correlación de Pearson excepto que la calificación es ranking, y la correlación r de Kendall, la cual es similar a la de Spearman, pero, en vez de usar los rankings propios, usa sólo los relativos para calcular la correlación.

- Cálculo de la similitud basada en coseno ajustado.

  La similitud calculada en la matriz se realiza a partir de las filas de usuarios analizados. La fórmula del coseno ajustado es la siguiente:

$$\mathrm{sim}(w,s) = \frac{\sum i \in I\,(R_{i,w} - \bar{R}_i)(R_{i,s} - \bar{R}_i)}{\sqrt{\sum i \in I (R_{i,w} - \bar{R}_i)^2}\sqrt{\sum i \in I (R_{i,s} - \bar{R}_i)^2}} \text{.....(3.17)}$$

Donde $\bar{R}_i$ es el promedio de la i-ésima evaluación del ítem i.

Finalmente, en las recomendaciones top-N se utiliza la lista de los N ítems mejor evaluados que sean de interés para el usuario activo. En el caso de los modelos basados en el usuario, primero se identifican los k usuarios más similares (vecinos cercanos) al usuario activo. Después de esto, las correspondientes filas en la matriz usuario-ítem se agregan para identificar una lista C de ítems junto con la frecuencia de sus apariciones. Luego el sistema recomienda, con esta lista C, los N ítems mejor evaluados, sin considerar los evaluados por el usuario activo.

Se debe tener en cuenta que cada sistema de recomendación puede usar diferentes medidas de distancia para implementar el cálculo de la similitud y las correspondientes estimaciones de los ítems aún no evaluados. Una estrategia muy común consiste en calcular todas las similitudes de los usuarios por adelantado y recalcularlas de vez en cuando. Entonces, cuando quiera que el usuario pregunte por una recomendación, la evaluación podrá ser eficientemente calculada.

Se debe notar que los sistemas de recomendación basados en contenidos y los colaborativos usan la misma medida de cosenos proveniente de la literatura de recuperación de la información. Sin embargo, en los primeros, se utiliza una medida de similitud entre los vectores TF-IDF, mientras que, en los sistemas colaborativos, la similitud se mide entre vectores de las evaluaciones especificadas por los usuarios.

Las técnicas vistas hasta el momento se han utilizado para calcular la similitud entre usuarios. En [SARW-01] se propuso, usando las mismas técnicas, el cálculo de las similitudes entre ítems para obtener evaluaciones a partir de ellos. Esta idea ha sido estudiada también en [Desh-04] para las recomendaciones de los Top-N ítems.

3.3.2. Basado en ítems

Un sistema de filtrado colaborativo basado en el ítem siguiere que a un usuario a quien le agrada el ítem A se le debería recomendar el ítem B si este se establece como el más similar al primero según la opinión de otros usuarios. Como en la aproximación basada en usuarios, las estrategias basadas en el ítem también reconocen patrones. Sin embargo, en vez de identificarlos entre la elección de los usuarios, lo hacen entre ítems.

En [Desh-04] y en [Sarw-01], se presenta evidencia empírica de que los algoritmos basados en ítems pueden rendir mejor que los métodos de filtrado colaborativo tradicional basados en usuarios.

El modelo de agrupamiento trata el problema de filtrado colaborativo como uno de clasificación [Basu-98], [Bree-98] y [Unga-98], y trabaja agrupando usuarios similares en una misma clase y estimando la probabilidad de que un usuario en particular esté en alguna clase determinada; a partir de esto, calcula la probabilidad condicional de las evaluaciones. Las aproximaciones basadas en reglas aplican algoritmos de reglas de asociación para encontrar agrupaciones entre ítems.

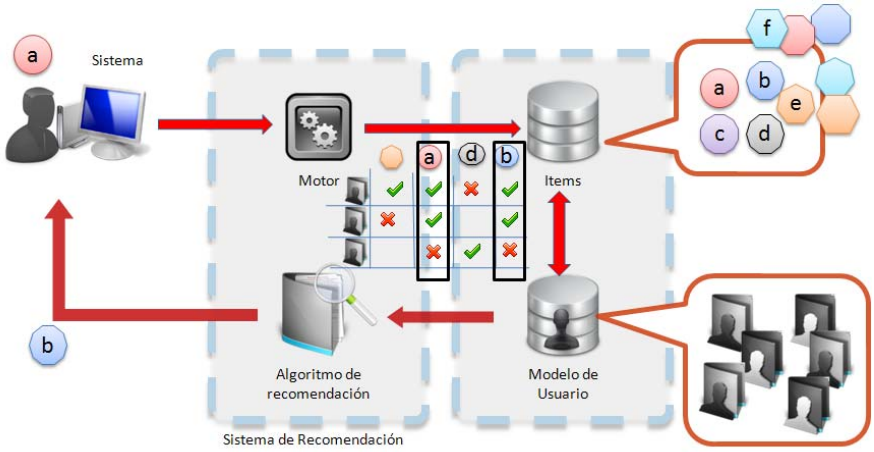

Figura 3.5. Proceso de recomendación que usa filtrado colaborativo basado en ítems

En la figura 3.5, se muestra el proceso de recomendación de un sistema de filtrado colaborativo basado en el ítem. Como en los basados en usuarios, la base de datos está compuesta por ítems seleccionados, evaluados o comprados por el usuario (el resto permanece invisible al sistema de recomendación). Las formas en que se capturan las preferencias también son comunes en ambas aproximaciones. Para generar recomendaciones, el sistema de filtrado colaborativo basado en el ítem encuentra ítems similares a los listados en el perfil del usuario activo, que pueden visualizarse como los que han sido evaluados altamente. Por último, se recomiendan al usuario activo los objetos con los mayores puntajes.

- **Cálculo de la similitud**

Un paso crítico en el algoritmo de filtrado colaborativo basado en ítems es el cálculo de la similitud entre ellos con el objetivo de seleccionar los artículos más afines. La idea básica en el cálculo de similitud $\mathrm{sim}(i_n, i_j)$ entre dos objetos i_n y i_j, paso crítico para la selección de artículos afines en el algoritmo de filtrado colaborativo basado en el ítem, es aislar a los usuarios que han evaluado ambos ítems en un conjunto U_m definido como $u_{n,j}=\{u_m \epsilon U / r_{m,n} \neq 0 , r_{m,j} \neq 0\}$ y luego aplicar una técnica de cálculo de similitud

para determinar la respectiva semejanza entre ellos o similitud $\mathrm{sim}(\mathrm{i_n}, \mathrm{i_j})$. Es posible hacer dicho cálculo de varias maneras. En este apartado, detallamos solo tres de ellas.

- Similitud basada en cosenos.
  Mide la similitud entre dos ítems calculando el coseno del ángulo formado por los vectores correspondientes.

$$Sim(i_n, i_j) = cos(r_n, r_j) = \frac{r_n . r_j}{\|r_n\| x \|r_j\|} = \frac{\sum_{u_m \in U_{n,j}} r_{m,n} . r_{m,j}}{\sqrt{\sum_{u_m \in U_{n,j}} r_{m,n}^2} \sqrt{\sum_{U_m \in U_{n,j}} r_{m,j}^2}} \text{.....(3.18)}$$

  Donde $\mathrm{r_{m,n}}$ es la evaluación que hace el usuario m al ítem n. Así mismo, $\mathrm{r_{m,j}}$ es la evaluacion que hace el usuario m al ítem j.

- Similitud basada en correlación.
  Mide la similitud entre los ítems calculando el coeficiente de correlación de Pearson en sus vectores de evaluaciones.

$$Sim(i_n, i_j) = \frac{\sum_{u_m \in u_{n,j}} (r_{m,n} - \overline{r_n})(r_{m,j} - \overline{r_j})}{\sqrt{\sum_{u_m \in u_{n,j}} (r_{m,n} - \overline{r_n})^2} \sqrt{\sum_{u_m \in u_{n,j}} (r_{m,j} - \overline{r_j})^2}} \text{.....(3.19)}$$

  Donde $\mathrm{r_{m,n}}$ es la evaluación que hace el usuario m al ítem n, $\overline{\mathrm{r_j}}$ es el promedio de la evaluación del j-ésimo ítem por esos usuarios.

- Cálculo de la similitud basada en coseno ajustado.
  La similitud calculada en la matriz se realiza a partir de las columnas de ítems analizadas. Sin embargo, el cálculo de la similitud del coseno tiene el inconveniente de no tomar en cuenta las diferencias de escala en la evaluación hecha por diferentes usuarios. Esto es resuelto por el cálculo del coseno ajustado extrayendo el promedio de evaluación de cada par coevaluado. La fórmula del coseno ajustado es la siguiente:

$$\mathrm{sim(i,j)} = \frac{\sum \mathrm{u} \in \mathrm{U} (\mathrm{R_{u,i}} - \overline{\mathrm{R}}_\mathrm{u})(\mathrm{R_{u,j}} - \overline{\mathrm{R}}_\mathrm{u})}{\sqrt{\sum \mathrm{u} \in \mathrm{U} (\mathrm{R_{u,i}} - \overline{\mathrm{R}}_\mathrm{u})^2} \sqrt{\sum \mathrm{u} \in \mathrm{U} (\mathrm{R_{u,j}} - \overline{\mathrm{R}}_\mathrm{u})^2}} \text{.....(3.20)}$$

  Donde $\overline{\mathrm{R}}_\mathrm{u}$ es el promedio de la u-ésima evaluación del usuario.

Finalmente, en las recomendaciones top-N, primero se calcula los k ítems más similares para cada ítem; luego se identifica la lista C como candidatos para la recomendación considerando la unión de los k ítems más similares y removiendo cada uno de los ítems en la lista U que el usuario activo ya ha evaluado. Posteriormente, se calcula las similitudes entre cada ítem de la lista C y la lista U. La lista C resultante,

ordenada de manera decreciente, será finalmente la lista de ítems top-N que se recomendará.

Predicción

El paso más importante en los mecanismos de recomendación de los sistemas de filtrado colaborativo basado en ítems es generar la predicción o, en términos más formales, encontrar el valor de la función utilidad $g(u_m, i_n)$ que mide la utilidad del ítem i_n para el usuario u_m.

- **Predicción por suma ponderada simple**

  Para generar la predicción, se usa el método de la suma ponderada [Adom-05]. Este método trata de establecer formalmente de qué manera el usuario activo evalúa objetos similares. Para obtener la predicción del objeto i_n para el usuario u_m, se calcula la suma de las evaluaciones $r_{m,j}$ dadas por u_m a los ítems i_j que pertenecen a una lista R (dentro de la lista se encuentran los ítems mas similares a i_n y que han sido calificados previamente). Cada evaluación es ponderada por la similitud correspondiente $\mathrm{sim}(\mathrm{i_n}, \mathrm{i_j})$ entre los objetos i_n y i_j.

  Así, la predicción para el usuario m en el ítem n es:

$$P_{m,n} = \frac{\sum_{j \in R} (\ r_{m,j} * Sim(n,j) \ )}{\sum_{j \in R} | Sim(n,j) |} \quad \text{.....(3.21)}$$

  $r_{m,j}$ es la evaluación del usuario m para el ítem j.

- **Suma ponderada de otras evaluaciones**

  Para realizar una predicción a un usuario m sobre un determinado ítem n, se puede tomar una suma ponderada de todas las evaluaciones de aquel ítem utilizando la siguiente fórmula:

$$P_{m,n} = \bar{r_n} + \frac{\sum_{j \in R} (\ (r_{m,j} - \bar{r_j}) \cdot Sim(n,j) \ )}{\sum_{j \in R} | Sim(n,j) |} \quad \text{.....(3.22)}$$

  Donde $\bar{r_n}$ and $\bar{r_j}$ son los promedios de las evaluaciones hechas en los ítems n y j, $r_{m,j}$ representa la calificación del ítem j realizada por el usuario m y $Sim(n,j)$ es el peso (relación) entre el ítem n y el ítem j. Esta sumatoria está realizada para los ítems que posean calificaciones del usuario m.

Como se puede observar, el resultado de este tipo de predicción se va ajustando por la calificación que otros usuarios han realizado para los ítems en particular.

- **Basada en regresión**

Según [Yu-04], en un escenario de regresión se busca la relación de una variable dependiente con distintas variables independientes. Un modelo de regresión típico puede estar expresado de la siguiente manera:

$$Y = a + B * X + \epsilon \text{.....(3.23)}$$

Donde Y representa la variable dependiente (en nuestro caso el curso a predecir) X, representa la variable independiente (la asignatura que ya tiene calificación), a representa el término constante, B representa el coeficiente de la variable independiente y $\boldsymbol{\epsilon}$ representa el error o valor aleatorio que puede estar presente en la regresión.

X e Y denotan los vectores que cumplen con el criterio de cocalificación (asignaturas que han sido cursadas por un mismo alumno). Estos vectores permiten la obtención de las constantes a y B de la siguiente manera:

$$a = \overline{Y} - B\overline{X} \text{.....(3.24)}$$

$$B = \frac{\sum(X-\overline{X})(Y-\overline{Y})}{\sum(X-\overline{X})^2} \text{.....(3.25)}$$

Donde $\bar{Y}$ representa el promedio del vector Y (promedio del curso a predecir) y $\bar{X}$ representa el promedio del vector X (promedio del curso que posee calificaciones). Con estos parámetros, podemos obtener el $\hat{Y}$ que nos servirá para la predicción. En nuestro caso de estudio, es necesario obtener un $\hat{R}$, que representa el $\hat{Y}$, por cada calificación que el usuario m haya realizado. De esta manera, el resultado final se define como:

$$\hat{R} = a + B * r_{m,j} \text{.....(3.26)}$$

El siguiente paso es obtener el valor que haga referencia al coeficiente de determinación R^2 que nos sirve para determinar el nivel de regresión que puede poseer el modelo y, a su vez, su error residual. Según [Yu-04], el valor de R^2 va ser denotado como r y representado por la siguiente expresión:

$$|r| = \frac{\sum_{u_m \in u_{n,j}} (r_{m,n} - \overline{r_n})(r_{m,j} - \overline{r_j})}{\sqrt{\sum_{u_m \in u_{n,j}} (r_{m,n} - \overline{r_n})^2} \sqrt{\sum_{u_m \in u_{n,j}} (r_{m,j} - \overline{r_j})^2}} = |Sim(n,j)| \text{.....(3.27)}$$

Se observa que para obtener el valor de r se aplica la correlación de Pearson entre dos ítems.

Finalmente, para lograr la predicción, se usa la siguiente expresión:

$$P_{m,n} = \frac{\sum_{j \in R} (\hat{R} * Sim(n,j))}{\sum_{j \in R} | Sim(n,j) |} \text{.....(3.28)}$$

Esto determina una predicción para un usuario m en un ítem n utilizando las j calificaciones que posea este.

3.3.3. Limitaciones de los sistemas de recomendación basados en filtrado colaborativo

Los sistemas de filtrado colaborativo han superado algunas debilidades de los basados en contenidos. Presentan, sin embargo, sus propias limitaciones, que se detallan a continuación [Bala-97], [Lee-01], [Burk-02], [Adom-05].

- ***Escasez de datos***

  Varios de los sistemas que emplean filtrado colaborativo para implementar sus recomendaciones se encuentran con una vasta lista de ítems por recomendar, una cifra alta en comparación con los usuarios que la evalúan. Por ello, en la matriz usuario-ítem, se obtienen pocas evaluaciones por cada ítem, lo que genera escasez de datos.

  Se pueden resaltar los siguientes problemas de escasez:

  o Cold Start

  El problema denominado “Cold Start” ocurre cuando un usuario o ítem recién ha ingresado al sistema y, por tanto, no se pueden encontrar similitudes con otros usuarios o ítems, debido a la escasez de información. Este limitante se puede describir de dos formas:

  <u>Problema del nuevo usuario</u>:

  Se sabe que los sistemas colaborativos aprenden las preferencias de los usuarios a partir de sus evaluaciones. Cuando un nuevo usuario entra, no se consideran sus evaluaciones, debido a que el sistema trabaja a partir de comparaciones entre las evaluaciones del usuario activo y las del resto. Por lo tanto, si la cantidad de evaluaciones es mínima o no existe, es extremadamente difícil formular la recomendación.

  Para tratar de superar este problema, [BURK-02] usó un sistema de recomendación híbrido, el cual combina las técnicas basadas en contenidos y las colaborativas. Por otro lado, ([Rash-02] y [Yu-04] plantearon que el sistema colaborativo podría proponer al nuevo usuario que evalúe el mejor ítem para el sistema colaborativo, es decir, el que presente mayor popularidad y mejor

entropía, a fin de que el sistema colaborativo pueda involucrar datos del nuevo usuario.

Problema del nuevo ítem

Es la contraparte simétrica del problema anterior. Se sabe que los sistemas colaborativos solo se fundamentan en las preferencias de los usuarios para hacer recomendaciones, y que no usan la información de los contenidos de los objetos existentes. Por lo tanto, hasta que un nuevo objeto sea evaluado por un usuario o un grupo de ellos, el sistema no podrá recomendarlo. Este problema aparece en dominios tales como noticias o, en general, en aquellos donde hay una rápida y constante rotación de nuevos objetos y cada usuario sólo evalúa algunos de ellos. Sin embargo, podrá superarse empleando sistemas de recomendación híbridos detallados en la próxima sección.

- Poca cobertura

  La cobertura puede definirse como el porcentaje de ítems a los que el algoritmo puede dar recomendación. Por ende, el problema de su escasez ocurre cuando el número de las evaluaciones dadas por el usuario es bastante pequeño comparado con el número de ítems en el sistema, así el sistema de recomendaciones podría ser incapaz de generar recomendaciones para ellos.

- Transitividad de usuarios

  Se refiere a un problema generado cuando dos usuarios con similitud en sus gustos no son tomados como tales por el sistema, debido a que ninguno de ellos ha coincidido en la evaluación de los mismos ítems. Esto podría reducir la confiabilidad de una predicción, dado que estas se basan en la comparación de usuarios en pares, comparación mediante la que se generan las predicciones.

- ***Dispersión***

  En la práctica, muchos sistemas comerciales de recomendación se utilizan para evaluar grandes cantidades de conjuntos de ítems. En muchos casos, el número de evaluaciones disponibles previamente obtenidas es muy pequeña comparada con la cantidad necesaria para realizar una predicción confiable. En estos sistemas, incluso los usuarios activos pueden llegar a comprar el uno por ciento de los objetos (cantidad que, por ejemplo, en el caso de los libros, alcanza el orden de los millones); por lo tanto, la recomendación puede ser no representativa con respecto al total.

  Para superar el problema de la dispersión, sería conveniente utilizar datos del perfil del usuario cuando se calcule su similitud. Así, dos usuarios pueden

considerarse similares no solo si evalúan los mismos ítems de manera parecida, sino si además pertenecen al mismo segmento demográfico.

Otra manera de superar esto se plantea en [HUAN-04], donde el problema de dispersión se enfrenta aplicando un esquema de recuperación asociativa.

- ***Escalabilidad***

  Este básicamente es un problema de cantidad de datos. Con millones de usuarios y objetos a la vez, la matriz de usuario-ítems puede sufrir problemas muy grandes de manipulación de datos. Este problema se trata según el dominio de aplicación y la técnica específica para encontrar similitud. Se produce cuando la base de datos es muy grande como para que el sistema pueda soportarlo. Es decir, la cantidad de usuarios e ítems registrados es tan grande, que las técnicas tradicionales de FC producirán recomendaciones más allá de lo aceptable o de lo práctico. Más aun cuando se necesita realizar predicciones en línea para la gran masa de usuarios de una manera inmediata, lo que genera una gran demanda de escalabilidad para el sistema de FC.

  Para solucionar este problema, se suelen utilizar técnicas para la reducción de dimensión, como la SVD, que rápidamente puede mejorar la calidad de las predicciones y ayudar al problema de escalabilidad.

- ***Sinonimia***

  Un problema de sinonimia aparece cuando, en un sistema de FC, se nombra un ítem de diferentes maneras. Este problema se hace común en sistemas donde un gran número de ítems carece de una adecuada terminología. De esta manera, pese a que es un mismo ítem, por presentar diferentes nombres, se trata como dos diferentes.

  Para la solución de este problema, podemos usar la técnica que en inglés se denomina Latent Semantic Indexing (LSI), que toma una matriz de términos y sinónimos asociando las palabras y construyendo un espacio semántico donde los términos y documentos que estén asociados se colocan cercanos entre sí. Así, el sistema de FC puede detectar qué términos están fuertemente asociados y descartar los demás.

3.3.4. Ejemplos de los sistemas de filtrado colaborativo

	Nombre	Principales características
Sistemas de filtrado colaborativo	***Grundy***	[Rich-79] Fue el primer sistema de recomendación que permitió añadir opiniones de los usuarios. Propuso el empleo de estereotipos como un mecanismo para construir modelos basados en una cantidad limitada de información sobre cada usuario. El sistema posibilitó almacenar opiniones o anotaciones sobre los contenidos de mensajes como un tipo de metainformación. Por su parte, el sistema brindaba a los usuarios la posibilidad de realizar búsquedas sobre el contenido de un documento, así como de la metainformación producida por los usuarios. Se empleó para recomendar libros.
	Tapestry	[Gold-92] Uno de los primeros sistemas automatizados para filtrado colaborativo fue diseñado para apoyar a una comunidad pequeña de usuarios. Estos podían filtrar la información, incluidos el correo electrónico y los artículos de usenet. Cuando los usuarios evaluaban un documento, ellos podían anotarlo. Con estas evaluaciones numéricas, otros usuarios podían enviar preguntas como “muéstreme los documentos que Mary anotó con ‘excelente’ y Jack anoto con ‘Sam debe leer’”.
	Ringo	[Shar-95] Es un sistema de filtrado colaborativo que hace recomendaciones personalizadas sobre temas de música y artistas. La base de datos crece diariamente a medida que los usuarios la alimentan mediante la descripción de sus gustos y preferencias con ayuda de una escala; dicha evaluación constituye los perfiles personales. Ringo utiliza estos para generar recomendaciones comparándolos con el propósito de identificar a los usuarios con perfiles parecidos.
	Phoaks	[Terv-97] Es un sistema basado en filtrado colaborativo que reconoce y reutiliza recomendaciones Phoaks. Es, asimismo, un sistema experimental que encuentra información relevante de alta calidad en la red. A través de un grupo de evaluaciones, busca mensajes sobre páginas electrónicas y las contabiliza como recomendación; para ello, se especializa en funciones como roles de usuario y rehúso de mensajes. En el caso de roles, este sistema espera que cada tipo de usuario realice la misma actividad sin esperar el mismo tipo de beneficios.
	Grouplens	[Sarw-98] Define y desarrolla un modelo híbrido que consta de estimaciones basadas en contenidos dentro de un sistema de filtrado colaborativo. Es posible por la inclusión de un agente inteligente llamado “filterbot”, que permite enfrentar con éxito los problemas de dispersión para los usuarios de sistemas de filtrado colaborativo escasamente poblados.

Tabla 3.2. Sistemas de filtrado colaborativo

3.4. Sistemas híbridos

Los sistemas híbridos de recomendación combinan técnicas del filtrado basado en contenidos y de filtrado colaborativo. Así, las recomendaciones híbridas se generan considerando características descriptivas y correlaciones según las evaluaciones dadas por los usuarios.

Existen numerosas formas de combinar métodos de recomendación colaborativos y basados en contenidos [Burk-02], [Adom-05]. Entre todas ellas, el paradigma más común es el de la recomendación colaborativa a través de contenidos [Pazz-99], donde los perfiles son construidos para detectar similitudes entre usuarios.

A continuación, se clasifican los sistemas híbridos de recomendación según [Burk-02].

- **Sistema híbrido con pesos**

Los sistemas de recomendación híbridos pueden construirse por medio de la implementación separada de sistemas colaborativos y basados en contenidos. Un sistema recomendador híbrido con pesos es uno en el cual la evaluación de un ítem se calcula a partir de los resultados de todas las técnicas de recomendación presentes en el sistema. Podemos combinar sus evaluaciones dentro de una recomendación final usando tanto una combinación lineal de evaluaciones como un sistema de votaciones.

Su beneficio es que todas las capacidades del sistema se utilizan en el proceso de recomendación de manera directa y sencilla. La suposición implícita de esta técnica es que el valor relativo de las diversas técnicas es más o menos uniforme a lo largo del universo de posibles objetos. Sin embargo, a partir de la discusión anterior, sabemos que esto no es siempre así: un recomendador colaborativo será más débil para aquellos ítems con un número reducido de evaluaciones o instancias.

- **Sistema híbrido alternante**

También pueden construirse sistemas de recomendación híbridos usando algunos criterios para alternar entre diferentes técnicas de recomendación. Estos criterios, en todos los casos, están basados en alguna métrica definida. En primer lugar, emplean las técnicas basadas en contenidos; si estas no dan buenos resultados, el sistema cambia a la de filtrado colaborativo. La alternancia aporta el alejamiento, de forma semántica, de los ítems previamente evaluados con puntuaciones altas, y el acercamiento a los ítems relevantes. Su beneficio es que las recomendaciones hechas son sensibles a las fortalezas y las debilidades de los métodos que la constituyen. Por otro lado, presenta una complejidad adicional, ya que el criterio de alternancia necesita un nivel más de parametrización.

- **Sistema híbrido mixto**

Este método se usa cuando se tiene que formular una gran cantidad de recomendaciones simultáneamente o cuando las recomendaciones de más de una técnica se presentan juntas. Tiene como ventaja evitar el problema del nuevo ítem, es decir, su componente basado en contenidos; puede dar nuevas recomendaciones, inclusive si los objetos no han sido evaluados. Estos sistemas explotan muy bien los beneficios de las técnicas basadas en contenidos y las colaborativas. Sin embargo, para su funcionamiento óptimo, requieren una lista ordenada de ítems, lo que implica el desarrollo de una técnica de priorización.

- **Sistema híbrido de combinación de rasgos**

Estos sistemas usan la información colaborativa como rasgos adicionales asociados a cada ítem y, sobre esta, aplican técnicas basadas en contenidos. Sin embargo, sus recomendaciones no dependen exclusivamente de los datos colaborativos, de aquellos que han evaluado los ítems, sino también de rasgos inherentes a ellos.

- **Sistema híbrido basado en cascada**

En este tipo de sistemas, se emplea primero una técnica con el objetivo de producir una gran lista de candidatos, y luego otra para refinarlos. Evita emplear técnicas secundarias para ítems que no serán recomendados dado su bajo nivel de estimación. Es más eficiente que aquellas que tratan con toda la lista de datos a la vez, debido a que, en una primera instancia, se filtran datos, y luego de filtrarlos, usando otra técnica, se enfocan únicamente al conjunto resultante.

- **Sistema híbrido basado en aumento de rasgos**

Al igual que en los sistemas de recomendación por cascada, la recomendación se obtiene en dos procesos. En primera instancia, se produce una clasificación de cada ítem mediante un modelo aprendido que genera características. Luego, estas se emplean como insumo en una segunda técnica, para explotar la información obtenida y enriquecerla. La diferencia entre esta técnica y la de cascada es que, en este caso, los rasgos utilizados en la segunda recomendación incluyen los resultados de la primera.

- **Sistema híbrido basado en metanivel**

Estos sistemas combinan dos técnicas de recomendación mediante el modelo generado por una de ellas como entrada para la otra. En este tipo de sistemas, el modelo aprendido, basado en recomendaciones por contenidos, es una representación de los intereses del usuario; el segundo paso del proceso de metanivel puede procesarse de una manera más simple trabajando con procedimientos de filtrado colaborativo.

3.4.1. Ejemplos de los sistemas híbridos

	Nombre	Principales características
Sistemas híbridos	***Con pesos***	**[Clay-99]** Sistema híbrido que combina evaluaciones obtenidas de sistemas de recomendación mediante combinación lineal. **[Pazz-99]** Sistema híbrido que combina evaluaciones obtenidas de sistemas de recomendación mediante un sistema de votaciones.
	Alternante	**[Pazz-00]** El sistema Daily learner emplea tanto la técnica basada en contenidos como la colaborativa. Primero usa la basada en contenidos; si esta no da resultados suficientemente fidedignos, se introduce la colaborativa. Actualmente, este sistema posee dos algoritmos basados en contenidos, el de corto y el de largo plazo; la técnica colaborativa se usa como intermedio entre las dos. **[Tran-00]** Este sistema alternante elige la técnica de mejor resultado según las evaluaciones pasadas de los usuarios.
	Mixto	**[Smyt-00]** PTV es un sistema hibrido que usa una aproximación mixta para ensamblar un sistema de recomendación de programas de televisión. Emplea la técnica basada en contenidos que, a su vez, se basa en contenidos y descripciones textuales de presentaciones televisivas e información colaborativa acerca de las preferencias de otros usuarios. Ambas recomendaciones se combinan para realizar la recomendación final.
	Combinación de rasgos	**[Basu-98]** Este sistema de combinación de rasgos se aplica a la recomendación de películas mediante una combinación entre rasgos basados en contenidos y estimaciones de usuarios, con lo que logra mejoras muy significativas en la precisión sobre los sistemas puramente colaborativos.
	Cascada	**[Burk-02]** EntreeC es un sistema en cascada basado en conocimiento y recomendación colaborativa. Aplica el conocimiento sobre restaurantes para hacer las recomendaciones según los intereses establecidos por los usuarios. Estas se colocan en grupos de iguales preferencias, y la técnica colaborativa se emplea para reordenar la recomendación final.
	Aumento de rasgos	**[Moon-99]** El sistema Libra hace recomendaciones de libros mediante la técnica basada en contenidos. Para lograrlo, emplea los datos de Amazon y aplica un clasificador de textos de Naïve Bayes. Dentro del texto utilizado como datos en este sistema, se encuentran incluidos los títulos y los autores de cada libro que Amazon genera a partir de su sistema colaborativo propio. Estas características son consideradas para contribuir significativamente con la calidad de la recomendación inicial.
	Metanivel	**[Bala-97]** El sistema FAB es un sistema de recomendación basado en técnicas colaborativas tradicionales manteniendo el perfil de cada basado en contenidos. Fue diseñado para ayudar a los usuarios a discernir entre una gran cantidad de información disponible en el red.

		Funciona desde fines de 1994 y combina los métodos basados en contenidos y colaborativos de recomendación aprovechando las ventajas de los dos y evitando sus limitaciones. La estructura híbrida de Fab permite un reconocimiento automático de problemas emergentes relacionados con varios grupos de usuarios y resuelve también problemas de escalabilidad.

Tabla 3.3. Sistemas híbridos

3.5. Evaluación de sistemas de recomendación

Una de las fases más importantes de cualquier proceso de recomendación, ya sea basado en memoria o en modelo, es la evaluación de los resultados obtenidos.
Generalmente, estos sistemas de recomendación, después del proceso, presentan alternativas para generar sus propias predicciones. Es imprescindible que estas alternativas sean evaluadas basadas en su calidad y efectividad antes que estos modelos sean desplegados en el dominio de aplicación para el cual fueron creados. El despliegue sin evaluación previa causaría una inversión y, como consecuencia, un gasto irrecuperable.

Algunas métricas utilizadas para evaluar sistemas de recomendación miden qué tan cerca están las predicciones hechas por el sistema de recomendación de las reales evaluaciones hechas por el usuario. Otras toman en cuenta la frecuencia con las que un sistema de recomendación hace predicciones correctas o no y otras miden la habilidad del algoritmo de recomendación para producir una lista ordenada de ítems.

Las métricas de precisión de predicción miden qué tan cerca están los valores predichos por el sistema de recomendación al valor real del usuario. Las métricas de precisión de predicción son particularmente importantes para las tareas de evaluación en las cuales el valor predicho será mostrado al usuario. Por ejemplo, el recomendador de películas MovieLens [Dahl-98] predice el número de estrellas que un usuario otorgará a cada película y muestran las predicciones al usuario. Así, estas métricas evalúan qué tan cerca estuvieron las predicciones de MovieLens a la cantidad real de estrellas que un usuario otorgó a cada película. Debido a que los valores predichos crean un ordenamiento entre los ítems, la precisión predictiva puede ser utilizada también para medir la habilidad de un sistema de recomendación para ordenar ítems considerando las preferencias del usuario. Esta métrica esta necesariamente limitada al cálculo de la diferencia los valores predichos y valores verdaderos.

Las métricas de precisión de clasificación (métricas de soporte de decisiones) miden la frecuencia con la cual ofrece una decisión correcta o errada sobre la posibilidad de que un ítem sea bueno. Así, este tipo de métrica es apropiada para tareas

encargadas de encontrar buenos ítems para el usuario que generalmente tiene preferencias.

Cuando se aplican en experimentos fuera de línea, las métricas de precisión de clasificación pueden ser amenazadas por la esparcidad de la información. El problema se da cuando el sistema de filtrado colaborativo, que está siendo evaluado, está generando una lista de los top ítems recomendados. Cuando la calidad de la lista es evaluada, las recomendaciones pueden tropezarse con el hecho de que no han sido evaluadas.

Una aproximación para la evaluación que utiliza bases de datos esparcidas, es ignorar las recomendaciones para ítems que no poseen evaluaciones. La lista de recomendación es primero procesada para remover todos estos ítems.

Otra aproximación es asumir valores por defecto, frecuentemente negativos, para los ítems recomendados que no han sido evaluados [Bree-98]. El problema de esta aproximación yace en que la evaluación por defecto podría ser muy diferente de la real (no observada) para un ítem.

Una tercera aproximación, aunque no sea aplicable en nuestro caso, es calcular cuántos de los ítems evaluados con puntuación muy alta son encontrados en la lista de recomendación generada por el sistema de recomendación. En esencia, estamos midiendo qué tan bien el sistema puede identificar los ítems de los que el usuario ya estuvo al tanto. Esta aproximación de evaluación puede resultar en algoritmos de filtrado colaborativo que son sesgados hacia recomendaciones que no son novedosas y que son obvias, o quizá algoritmos que pueden generar sobreajuste. Las métricas de precisión de clasificación no pretenden medir directamente la habilidad de un algoritmo para predecir evaluaciones en forma precisa.

Las métricas de precisión de ordenamiento miden la habilidad del algoritmo de recomendación para producir una lista ordenada de ítems recomendados que pueda compararse con la lista de cómo el usuario ordenaría los mismos ítems. A diferencia de las métricas de clasificación, estas son más apropiadas para evaluar algoritmos que serán usados para presentar listas de recomendación raqueadas a los usuarios en dominios donde las preferencias de los usuarios son no binarias.

Un estudio muy claro y profundo de la evaluación de los sistemas de recomendación basados en filtrado colaborativo se encuentra en [Herl-04]. En las siguientes secciones, y siguiendo los lineamientos de este documento, delinearemos las métricas más usadas para evaluar sistemas de recomendación.

3.5.1. Métricas de precisión de predicción

Las métricas de precisión de predicción miden qué tan cerca están los valores predichos por el sistema de recomendación al valor real del usuario. Estas juzgan la

precisión de una predicción simple $P_{m,n}$ para los ítems i_n comparándola con su valor actual $r_{m,n}$. Las métricas más representativas con estas propiedades son:

3.5.1.1 Error absoluto medio (MAE): Se usa comúnmente en sistemas de recuperación de la información y en sistemas de recomendación. Es un valor que mide la desviación de la recomendación a partir del valor especificado por el usuario. Para cada par de valores $(r_{m,n}, P_{m,n})$ en un sistema de recomendación clásico, donde $r_{m,n}$ es el valor actual y $P_{m,n}$ es el valor predicho, el MAE es calculado primero sumando los errores absolutos de todas las N predicciones para todos los M usuarios utilizando la siguiente relación:

$$MAE = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left|r_{m,n} - P_{m,n}\right|$$

A menor MAE, más precisa es la predicción que conduce a proporcionar una recomendación.

Desde la perspectiva de la clasificación, el MAE se puede definir mediante la siguiente relación:

$$MAE = \frac{1}{M}\sum_{i=1}^{M}|actual(C_i) - Pred(C_i)|$$

Donde *Pred(C)* es el valor de la predicción de cada individuo, *Actual(C)* es el valor actual del individuo (propio de cada individuo) antes de hacer la predicción, que generalmente está en el conjunto de prueba, y M es el número de individuos. En este caso, los predictores están en el rango [0,1] y son comparados con los valores actuales que son binarios, es decir, en el conjunto $\{0,1\}$.

3.5.1.2. Raíz del error cuadrático medio (RMS): Usado comúnmente en sistemas de recuperación de la información y en sistemas de recomendación, esta sigue la misma lógica que el MAE, pero extrayendo la raíz a la suma de las diferencias al cuadrado entre el valor predicho y el valor real. Para cada par de valores $(r_{m,n}, P_{m,n})$ en un sistema de recomendación clásico, donde $r_{m,n}$ es el valor actual y $P_{m,n}$ es el valor predicho, tenemos:

$$RMS = \sqrt{\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left(r_{m,n} - P_{m,n}\right)^2}$$

Esta métrica es muy usada en análisis de regresión, mide cuánto la predicción se desvía de su valor real.

Desde la perspectiva de la clasificación, el error RMSE se puede definir mediante la siguiente relación:

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\big(actual(C_i) - Pred(C_i)\big)^2}$$

Desde el punto de vista de la clasificación, el RMS es aplicable solo cuando los predictores están en el rango [0,1] y son comparados con los valores actuales (reales) que son binarios, es decir, en el conjunto $\{0,1\}$. En la presente memoria, el valor predicho se considera como la probabilidad y el actual se considera uno si es aprobado y cero si está suspendido (desaprobado).

A menor RMS, más precisa es la predicción que conduce a proporcionar una recomendación.

3.5.1.3 Error absoluto medio normalizado (NMAE): NMAE (Normalized Mean Absolute Error): Proviene del MAE y es utilizado para sistemas recomendadores con diferentes escalas numéricas cuando se quiere comparar con un MAE. Se tiene un rango de [0;1] siendo cero el valor ideal.

$$NMAE = \frac{MAE}{rmax - \mathrm{rmin}}$$

Donde:

rmax representa la máxima calificación posible;

rmin representa la mínima calificación posible.

3.5.1.4. La media cruzada de la entropía (MXE): Es usada en un sentido probabilístico cuando interesa predecir la probabilidad de que un ejemplo sea positivo. Se puede probar que minimizando la entropía cruzada se da la máxima hipótesis de verosimilitud. La media cruzada de la entropía es definida como:

$$MXE = -\frac{1}{N}\sum\big(Actual(C) * ln(Pred(C))\big) + (1 - Actual(C) * \ln(1 - pred(C)))$$

En este caso, se asume que $Pred(C) \in [0,1]$ y que $Actual(C) \in \{0,1\}$.

Para los efectos de las experimentaciones, solo usaremos error absoluto medio (MAE), y la raíz del error cuadrático medio (RMS).

3.5.2. Métricas de precisión de clasificación (Decisión Support)

Este tipo de métrica es apropiada para tareas encargadas de encontrar buenos ítems para el usuario que generalmente prefiere, sobre todo, elegir ítems de alta calidad para él. Estas métricas también reciben el nombre de "métricas de soporte de decisión" y determinan qué tan bien un sistema de recomendación puede hacer predicciones de ítems de alta relevancia; esto es, miden la frecuencia con la cual el sistema ofrece una decisión correcta o errada sobre la posibilidad de que un ítem sea bueno para el usuario o, mejor dicho, ítems que podrían ser altamente ranqueados por los usuarios.

Este tipo de métricas se podría dividir en dos segmentos: métricas de umbral y métricas de rango [Caru-04].

Para las métricas de umbral, no es importante cuán cerca una predicción está de su umbral, sino si está por debajo o encima del umbral. Estas pueden ser Accuracy, Lift y F-Score.

3.5.2.1. Accuracy (ACC): Es probablemente la métrica más usada en el campo del aprendizaje automático. Se define como la proporción de predicciones correctas que el clasificador hace con respecto al total de los datos. Si un clasificador tiene salidas continuas, como las redes neuronales, se fija un umbral y cada ejemplo sobre el umbral se considera como una predicción correcta.

$$Accuracy = \frac{No.\,de\ instancias\ Bien\ Clasificadas}{No.\,de\ Instancias\ Totales}$$

A mayor ACC, más precisa es la predicción que conduce a proporcionar una recomendación.

3.5.2.2. Lift (LFT): Se usa frecuentemente en estudios de mercado, Mide cuánto más un clasificador predice ejemplos positivos que un clasificador aleatorio:

$$Lift = \frac{\%\ de\ falsos\ positivos\ arriba\ del\ umbral}{\%\ de\ ejemplos\ totales\ arriba\ del\ umbral}$$

Donde: $\%\ de\ falsos\ positivos\ arriba\ del\ umbral = \frac{tp_s}{N_s}$ siendo tp_s y N_s la cantidad de verdaderos positivos de una muestra. Usualmente se dispone de un umbral para que un porcentaje fijo de los datos totales sean clasificados como positivos.

A mayor LFT, más precisa es la predicción que conduce a proporcionar una recomendación.

3.5.2.3. F- Score (PRF): Fue propuesta por Lewis en [Lewi -94] y se define como la media armónica entre la precision y el recall.

$$PRF = \frac{(\beta).Precision.Recall}{\beta^2(Precision + Recall)}$$

Donde el parámetro $\beta\epsilon[0,1]$ determina la influencia relativa de ambas métricas. En la mayoría de los casos, se usa el valor $\beta = 1$.

Las métricas de Rango miran las predicciones de forma distinta que las métricas de umbral. Si los casos están ordenados por sus valores predichos, la métrica de rango mide cuantos casos positivos hay sobre casos negativos. La métrica de rango puede ser vista como un resumen del rendimiento del modelo sobre todos los posibles umbrales y depende únicamente del orden de las predicciones y no de sus valores reales. Las métricas de rango estudiadas son el área bajo la curva (AUC) y precision/recall break even point (BEP).

3.5.2.4. Área bajo la curva ROC (AUC):

Una curva ROC es una técnica gráfica para elegir clasificadores en base a su desempeño. Esta curva muestra la concesión que se tiene que hacer del porcentaje de falsos positivos para obtener un mayor porcentaje de verdaderos positivos [Fawc-04] .Para ser utilizada, es necesario que un algoritmo dé un puntaje a la posibilidad de que un registro pertenezca a una clase.

La curva ROC se grafica en un sistema de coordenadas rectangulares. Para ello necesita un puntaje que está en relación con la probabilidad que obtiene el clasificador con respecto a la clase positiva, independiente de la clase real que posea. Para cada punto de la curva en el plano cartesiano, se consideran cada uno de los valores distintos del puntaje, la tasa de verdaderos positivos se calcula en el eje vertical y la tasa de falsos positivos en el eje horizontal.

Las abscisas y ordenadas que corresponden a cada punto que pertenece a la curva ROC son la cantidad de registros positivos y negativos, respectivamente, cuyos puntajes sean mayores o iguales que el valor del puntaje actual.

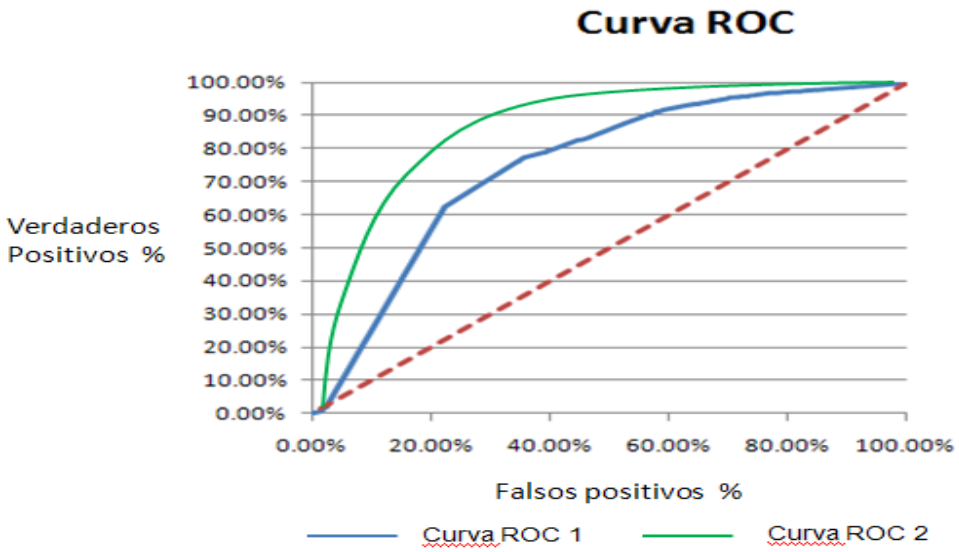

Figura 3.6. Dos curvas ROC correspondientes a dos clasificadores distintos

El área bajo la curva (AUC) indica el desempeño promedio de un clasificador a través de todos los posibles puntajes. No obstante, una curva puede ser mejor que otra para un punto particular del espacio ROC. Todos los clasificadores deberían tener más que 0.5, que es el valor que obtiene un clasificador aleatorio [Fawc-04].

Hablando específicamente del campo del diagnóstico, por ejemplo, ciertos errores resultan más costosos que otros. En ese sentido, en términos generales, conviene diferenciar muy claramente el costo de un error del error mismo. Así, si bien no podemos evitar que un clasificador cometa errores, deberíamos estar en condiciones de discriminar sus costos, pues es a partir de ellos que se determina la calidad del clasificador.

Efectivamente, a partir de lo dicho, se deduce que la minimización de errores no repercute en la calidad del modelo tanto como la minimización de costos. Por lo tanto, el trabajo deberá estar enfocado en el momento previo al aprendizaje, al presentar las matrices de costos. Ello, no obstante, se enfrenta con un obstáculo: muchas veces, los costos no se conocen de antemano o los modelos han sido seleccionados previamente.

Precisamente, es en esas circunstancias que el análisis ROC presenta una altísima utilidad como método de selección de clasificadores. Gracias a él se separan aquellos con un comportamiento óptimo y aquellos con una utilidad menor. Dicho análisis, fácilmente aplicable para clasificadores binarios, se lleva a cabo construyendo una envoltura convexa a partir de todos los clasificadores, de la que resulta una curva que, conjuntamente a los ejes, grafica un polígono convexo. El resultado de involucrar la discriminación de los costos de los errores en el proceso, a través del análisis ROC, es

evidente: se obtiene un análisis predictivo más realista. Asimismo, los modelos resultantes permitirán que las decisiones sean más adecuadas.

3.5.2.5. Precisión/Recall (BEP): Estas medidas son ampliamente usadas en recuperación de la información y representa aquel punto en donde la precisión y el *recall* son iguales.

3.5.2.6. Precisión: Es la métrica que representa la probabilidad de que un ítem recomendado como relevante lo sea verdaderamente. En otras palabras, es la fracción de ejemplos predichos correctamente como positivos sobre todos los ítems clasificados de esa manera.

$$Precision = \frac{VP}{VP + FP}$$

3.5.2.7. Recall: Es la métrica que representa la probabilidad de que un ítem relevante sea recomendado como tal. En otras palabras, es la fracción de ejemplos correctamente predichos como positivos entre todos los ejemplos que son realmente positivos.

$$Recall = \frac{VP}{VP + FN}$$

3.6. Experimentación en nuestro dominio de aplicación

Con el objetivo de probar la eficacia del método de filtrado colaborativo, en el capítulo 6 se llevará a cabo la experimentación correspondiente. Cabe mencionar que esta experimentación tiene fuertes limitaciones, debido a que los datos considerados no constan de todos los atributos que posee el estudiante y que necesita el sistema para simular la asesoría que se lleva a cabo en la realidad.

Debido a las limitaciones de los sistemas de filtrado colaborativo basado en memoria, que son los que usan en su mayoría una sola variable que permita determinar las similitudes y por consiguiente las predicciones, solo se consideran las notas para el cálculo de la predicción, dejándose de considerar elementos importantes, como el promedio ponderado acumulado, las veces que cursa la misma asignatura, los créditos cursados, etc.

Capítulo 4:

Mecanismo de Recomendación

La recomendación no es un fenómeno nuevo y, desde luego, no surge de la era digital, sino de un comportamiento social existente en la vida cotidiana e inherente a ella. Sin embargo, en un mundo donde la información es abundante y la cantidad de decisiones sobrepasa cualquier límite, la recomendación, como proceso natural, cobra particular importancia y puede ayudar a encontrar y a evaluar ítems de interés. Con la introducción de la red, este proceso social ha ido creciendo. Actualmente, existen los sistemas de recomendación cuya finalidad es asistir automáticamente a los usuarios para que decidan de manera más acertada sobre sus preferencias. Conectan a los usuarios con ítems [Scha-05] asociando el contenido del ítem recomendado o la opinión de otros individuos con las acciones u opiniones de los usuarios originales del sistema.

El desarrollo de los sistemas de recomendación es una actividad cuya complejidad depende de la precisión que se desee ofrecer. Por esta razón, no suele ser suficiente una limpieza y un filtrado adecuado de los datos, sino que, en la mayoría de los casos, es necesaria una evaluación posterior basada en los resultados obtenidos por los usuarios. Para este tipo de evaluación, se utilizan técnicas de descubrimiento del conocimiento, que asisten al que ofrece la herramienta de recomendación en la evaluación y validación del sistema recomendador. Estas técnicas permiten descubrir nuevos conocimientos a partir de los datos de utilización o datos históricos.

El descubrimiento del conocimiento en bases de datos (KDD) está orientado al desarrollo de métodos y técnicas que les den sentido a los datos. El dato se define, según Elmasri [Elma-00], como algo que podemos almacenar y que tiene un significado implícito. Para Davenport [Dave-98], el conocimiento es una mezcla de experiencia, valores e información que sirve como un marco de referencia para la incorporación de nuevas experiencias.

Una definición más precisa de descubrimiento del conocimiento es la de Fayad, que lo describe como: “el proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y entendibles en los datos” [Fayy-97]. Por válido se entiende que los patrones descubiertos deben seguir siendo precisos (con algún grado de certidumbre) para integrar datos nuevos que representen un aporte acerca de algo previamente desconocido. Los términos “útiles” y “entendibles” de la definición son más subjetivos y dependen del punto de vista del analista.

Históricamente, el concepto y la acción de buscar patrones útiles derivados de datos han sido tratados con diferentes nombres, incluyendo los de “minería de datos”, “extracción de conocimiento”, “descubrimiento de la información”, “arqueología de datos” y “procesamiento de patrones de datos”. El término “minería de datos” ha sido

mayormente usado por las comunidades de estadísticos, analistas de datos y administradores de datos.

De modo general, el objetivo de KDD y de DM es extraer información útil de grandes cantidades de datos. Enmarcada en dicho contexto general, esta investigación tiene como objetivo utilizar estas técnicas como base para un motor de recomendación, de modo que la orientación que asista al usuario pueda basarse en los patrones y tendencias ocultos en los datos; por lo tanto, en este capítulo, se presentarán los conceptos básicos de KDD y se enfocará, de modo detallado, las tareas y técnicas de minería de datos.

4.1. Descubrimiento del conocimiento (KDD)

Algunos autores interpretan el KDD como un sinónimo de DM, como, por ejemplo, Adrians [Adri-96], mientras que otros enfocan DM como un paso esencial del KDD [Mani-97] y [Fayy-96]. Adoptaremos, en este trabajo, la definición de Fayad para los términos de KDD y de DM: “KDD es un proceso que abarca el descubrimiento de conocimiento útil a partir de datos, en cuanto que DM se restringe a la aplicación de algoritmos de extracción de patrones”.

El descubrimiento del conocimiento se ocupa, entonces, del proceso completo: del almacenamiento de los datos, de la manera de acceder a ellos, de la aplicación de los algoritmos a grandes conjuntos de datos y, finalmente, de la interpretación de los resultados y de su presentación. La minería de datos es, en cambio, un proceso más dentro del procedimiento del descubrimiento del conocimiento.

Después de definir el dominio de aplicación y aclarar los objetivos del proceso de KDD, se detallarán los subprocesos que componen el descubrimiento del conocimiento:

- ***Selección de los datos***: Proceso que se encarga de seleccionar los datos sobre los cuales se efectuará el descubrimiento. Estos se almacenan en una base de datos.
- ***Preprocesamiento de los datos***: Es un primer análisis que busca limpiar los datos, recolectar información sobre el modelo y definir estrategias para manejar los datos faltantes.
- ***Transformación de datos***: Se trata de encontrar características útiles para representar los datos según los objetivos de la tarea. Esto se logra aplicando métodos de reducción o transformación de la dimensionalidad. Con ello se puede reducir el número de variables.
- ***Minería de datos***: Consiste en encontrar un método adecuado de minería de datos alineado a los objetivos que se quieren alcanzar.

- ***Interpretar los patrones***: En este proceso, es posible visualizar los patrones extraídos o, dado un modelo, visualizar los datos. También es posible que se decida volver al primer paso y comenzar una nueva iteración del proceso.
- ***Difusión***: En este proceso, se trata de obtener el conocimiento en sí. Posiblemente, se incorpora a otro sistema para más acciones o, simplemente, se documenta.

Usama Fayyad, en [Fayy-96], sugiere la naturaleza iterativa e interactiva del descubrimiento del conocimiento. Al final de cada paso, el analista puede decidir si regresa a un proceso anterior y lo reinicia.

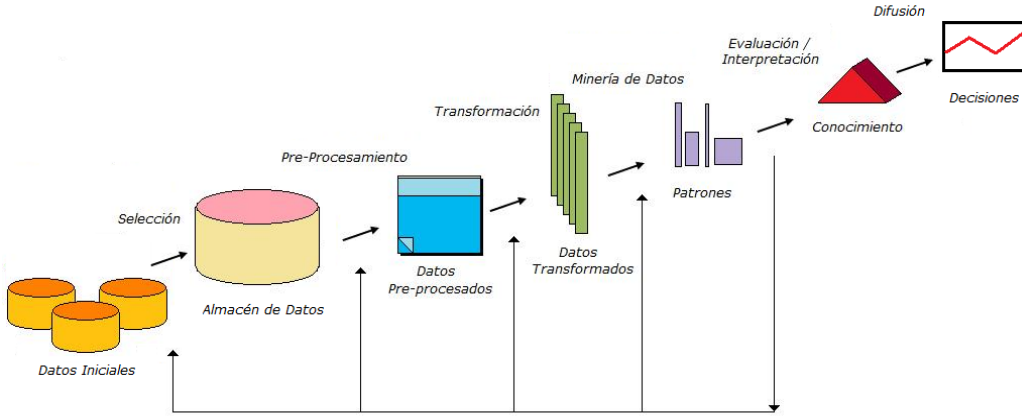

Figura 4.1. Proceso de descubrimiento de conocimiento

Cabe precisar que existen dos tipos de modelos de descubrimiento del conocimiento: aquellos que lo predicen y aquellos que lo describen. Los primeros pretenden estimar valores futuros o desconocidos a partir de alguna variable de interés, mientras que los segundos identifican patrones que explican los datos, es decir, sirven para explorar las propiedades de los datos examinados, no para predecir nuevos.

En los últimos años, las aplicaciones de KDD van incrementándose cada vez más en diferentes áreas como, por ejemplo, astronomía, medicina, biología, climatología, márquetin, publicidad y en todos los campos del ámbito empresarial, donde ha tenido sus mayores desarrollos. En este trabajo, se describe una propuesta para utilizar el KDD en el área de la recomendación aplicada al ámbito educativo.

4.2. El aprendizaje automático

El aprendizaje automático (Machine Learning) es el área de la inteligencia artificial que se ocupa de desarrollar algoritmos capaces de “aprender”; en otras palabras, trata de crear programas capaces de generar comportamientos a partir de una información no necesariamente estructurada con anterioridad, llamada “ejemplo de

entrenamiento". El resultado del aprendizaje se refleja en un modelo de comportamiento que, de ser exitoso, produce una mejora en el rendimiento de la tarea aprendida [Mitc-97]. Constituye, junto con la estadística, el núcleo del análisis inteligente de los datos.

4.3. Minería de datos

La minería de datos (DM), campo multidisciplinario que involucra técnicas como el aprendizaje automático, el reconocimiento de patrones, la estadística y las bases de datos, es el proceso más importante dentro del descubrimiento del conocimiento. Como ya se precisó, se trata de un componente más en una serie. Consiste en analizar determinada información y aplicar los algoritmos apropiados con la finalidad de que produzcan nuevos patrones a partir de los datos originales. Su reto es, precisamente, trabajar con grandes volúmenes de datos procedentes de sistemas de información que, por otro lado, pueden contener sus propios problemas (ruido, datos faltantes, volatilidad, etcétera).

Se lo considera como una evolución natural iniciada con la creación de las primeras bases de datos y que continuó con el Lenguaje Estructurado de Consultas (SQL, por sus siglas en inglés) y, con un mayor impacto, con el Online Analytical Processing (OLAP). Lo que pretende la minería de datos es automatizar el proceso localizando y extrayendo patrones ocultos. En su forma más pura, no busca un tipo específico de información, sino que, simplemente, busca patrones entre los datos.

Según Witten [Witt-00], "la minería de datos es la extracción de información implícita, previamente desconocida y potencialmente útil contenida en los datos". Desde un punto de vista operacional, la minería de datos se define como el proceso automático o semiautomático de descubrir patrones en grandes cantidades de datos, bajo la premisa de que los patrones descubiertos deben ser útiles. Así también, [Tan-06] define la minería de datos como la tecnología que combina las técnicas y métodos tradicionales de análisis con sofisticados algoritmos para procesar grandes cantidades de datos.

Es así como la minería de datos, en nuestro contexto, se puede definir como el conjunto de métodos y técnicas que mediante un análisis de grandes volúmenes de datos permiten descubrir información oculta que facilita su entendimiento. Las técnicas de minería de datos son numerosas; en el presente trabajo se estudiarán algunas de ellas así como las diferentes métricas de precisión, que son las que se encargan de medir su rendimiento.

Un sistema de minería de datos puede ser clasificado, según Chen [Chen-96], mediante los siguientes criterios:

- Tipo de banco de datos sobre el cual la minería se aplica.

- Tipo de conocimiento que será minado (reglas de asociación, reglas de clasificación, agrupamiento, etcétera).
- Tipo de técnica que debe ser utilizada.

4.3.1. Tareas de minería de datos

Las tareas de minería de datos están divididas en dos grandes categorías. Unas son las tareas predictivas, cuyo objetivo es predecir el valor de un atributo particular a partir de los valores de otros atributos. El que será predicho se conoce comúnmente como “variable dependiente”, mientras que los atributos usados para hacer predicciones se llaman “variables explicativas” o “independientes”. Las otras tareas son las descriptivas, cuyo objetivo es producir patrones (correlaciones, tendencias, clústers y trayectorias) que resumen la relación subyacente de los datos.

Dentro de esta clasificación, existen cuatro tipos más específicos de tareas de minería de datos:

Modelo predictivo: Se refiere a la construcción de un modelo para la variable dependiente como función de las variables explicativas. Existen dos tipos de modelos predictivos: la clasificación (variables discretas) y la regresión (variables continuas).

La asociación: Se usa para descubrir patrones que describen características fuertes de asociación entre los datos. Estos patrones son representados en forma de reglas. Dado que el espacio de búsqueda es muy amplio, el objetivo del análisis de asociación es extraer los patrones más interesantes de la forma más eficiente posible. Surgieron como una respuesta a la necesidad de analizar las cestas de compra; por este motivo, a las instancias se les suele llamar “transacciones”, y a los atributos, “productos”. Así, las reglas de asociación describen la posibilidad de que la presencia de un producto implique una fuerte tendencia a la presencia de otros dentro de la misma transacción.

Análisis de conglomerados: Son un conjunto de técnicas que permiten clasificar los objetos en grupos relativamente homogéneos llamados “conglomerados”. Los objetos en cada grupo tienden a ser similares entre sí y a diferenciarse de otros grupos. Se entiende que estos deben ser mutuamente exclusivos y colectivamente exhaustivos.

4.4. Técnicas de minería de datos

En minería de datos, las técnicas supervisadas y no supervisadas tienen diferentes propósitos. Si bien es cierto que ambas se usan generalmente para la extracción de información útil a partir de grandes volúmenes de datos, las dos difieren radicalmente en el tratamiento de los datos. Mientras que las técnicas supervisadas entrenan ejemplos etiquetados, de tal manera que al evaluar un elemento nuevo lo clasifican de acuerdo con el conjunto de entrenamiento, las técnicas no supervisadas no

poseen dichas etiquetas, de tal manera que la clasificación se produce sin conocer absolutamente nada de los ejemplos de entrenamiento más que los atributos de entrada.

Una técnica de clasificación o clasificador es una aproximación sistemática para construir modelos de clasificación desde un conjunto de datos de entrada. La tarea consiste en diferenciar individuos de acuerdo con sus atributos y agruparlos en clases. Esta tarea sería fácil si se conocieran las reglas que asignan valores de clases a los individuos de acuerdo con las características de cada uno de ellos. El problema fundamental reside en inducir estas reglas de clasificación, cuando son desconocidas, a partir de la información contenida en un conjunto inicial de datos que llamaremos datos de entrenamiento o conjunto de entrenamiento.

Para obtener dichas reglas se han diseñado un gran número de algoritmos o técnicas de aprendizaje. Tratándose de sistemas de aprendizaje supervisado, el fin principal es predecir la etiqueta de un nuevo elemento basándose en los atributos que lo caracterizan y utilizando las reglas inducidas a partir del conjunto de entrenamiento. Dentro del aprendizaje supervisado, se distinguen dos tipos de problemas, dependiendo de la naturaleza de la etiqueta de clase. Se llama “clasificación” cuando las posibles etiquetas toman valores discretos y se llama “regresión” cuando dichos valores son continuos.

Un algoritmo de clasificación supervisado se puede definir como aquel que construye un modelo tal que, dado un vector de atributos a_j(j puede ser o no igual a i), pueda obtener una clase C con $c_i = \{1,2,3,, \dots \dots \dots C\}$ usando el conocimiento contenido en el conjunto de datos iniciales B, siendo B un conjunto de n ejemplos de la forma $B = \{(a_i, c_i);\ i = 1,2,3, \dots . . n\ \}$

Como ejemplos, tenemos los árboles de decisión [Quin-97], los clasificadores basados en reglas, las redes neuronales [Hayk-99], Support Vector Machine [Vapn-95] y los clasificadores Naïve Bayes [Pear-88]. Cada técnica emplea un algoritmo de aprendizaje para identificar el modelo que mejor se ajuste a los datos de entrenamiento. El modelo generado por el algoritmo de aprendizaje muestra la entrada de datos y la predicción creada por el modelo de nuevos registros. Por lo tanto, un objetivo clave del algoritmo de aprendizaje es construir modelos con buena capacidad de generalización**.**

4.4.1. Árboles de decisión

Un árbol de decisión es un diagrama construido a partir de un conjunto de observaciones, es decir, de instancias propias de algún dominio en particular, y que poseen atributos que describen a cada elemento. El diagrama representa y clasifica una serie de condiciones sobre los valores de los atributos. Su objetivo es obtener alguna conclusión, llamada “función objetivo”, acerca de dichas observaciones.

En su forma básica, un árbol de decisión está compuesto por hojas, arcos y nodos. Estos están asociados a los atributos que se evalúan para determinar el camino que se debe seguir a través de las dos o más alternativas posibles (representadas por los arcos). El primer nodo que se evalúa se denomina nodo raíz. Un nodo que no se ramifica se conoce como hoja, y es el resultado que devolverá el árbol (la función objetivo). Finalmente, una ruta que se inicia en el nodo raíz y termina en una hoja se denomina rama, y representa todas aquellas instancias que cumplen con las condiciones de los nodos que se evalúan en el recorrido. En otras palabras, los árboles de decisión clasifican instancias y, por lo tanto, también pueden utilizarse como una generalización de aquellas que no se han incluido en las observaciones. En ese sentido, es un modelo de predicción.

Los árboles de decisión representan una "disyunción de conjunciones de restricciones" [Mitc-97] de los valores de los atributos de las instancias. Cada ruta corresponde a la conjunción de las valoraciones a las que se ha sometido al atributo, y el árbol en sí es la disyunción de estas conjunciones.

Un atributo puede tener valores binarios (cuando solamente existen dos posibilidades), nominales (cuando el atributo puede tener uno de varios valores categóricos posibles), ordinales (cuando los valores del atributo pueden ser varios, pero, a la vez, existe la posibilidad de agruparlos siempre y cuando no se vulnere el orden que los caracteriza) o continuos (cuando los valores de los atributos son números reales).

El algoritmo básico utilizado para construir un árbol de decisión se llama ID3 (Inductive Decision Tree) y utiliza una búsqueda de arriba-hacia-abajo a través de todo el espacio de árboles de decisión posibles. El algoritmo ID3 y su sucesor C4.5 fueron presentados por Ross Quinlan en 1986 y 1993, respectivamente.

El algoritmo ID3 va construyendo el árbol de decisión hasta que este clasifique perfectamente los ejemplos de entrenamiento o hasta que todos los atributos hayan sido utilizados. Los datos de entrada para el algoritmo ID3 se conocen como los conjuntos de instancias de "entrenamiento" o de "aprendizaje" que el algoritmo utilizará para generar el árbol de decisión. El algoritmo ID3 construye el árbol de decisión preguntándose: ¿qué atributo debe ser analizado en la raíz del árbol? Para responder esta pregunta, cada instancia se evalúa para determinar lo bien que clasifica, por sí sola, a los ejemplos de entrenamiento.

El algoritmo C4.5 se construye mediante el método Hunt. La base para este método consiste en construir un árbol de decisión. A partir de un conjunto T de datos de entrenamiento, se definen las clases $\{c_1, c_2, c_3, \dots\dots\dots\dots c_k\}$. Luego, existen tres posibilidades:

1. T contiene uno o más casos que pertenecen a una única clase c_j. En esta situación, el árbol de decisión es una hoja cuya etiqueta es la clase c_j.
2. T no contiene ningún caso. En esta situación, el árbol de decisión también es una hoja cuya etiqueta podrá ser determinada por información que no pertenece al conjunto T. En la mayoría de casos, esta información depende del dominio de aplicación.
3. T contiene casos que pertenecen a varias clases. En esta situación, lo que conviene es refinar el conjunto T en subconjuntos de casos que tiendan a pertenecer a una única clase. Para llevar a cabo esto, primero se elige una prueba basada en un único atributo que tiene uno o más resultados mutuamente excluyentes. $\{o_1, o_2, \dots \dots \dots o_n\}$; de aquí T se divide en los conjuntos $\{T_1, T_2, \dots \dots \dots T_n\}$, donde t_i contiene todos los casos de T que tienen el resultado o_i para la prueba elegida. Cada nodo identifica la prueba, y las ramas, a sus resultados posibles. El mecanismo de construcción del árbol se aplica recursivamente a cada subconjunto de datos de entrenamiento.

El mejor atributo se selecciona para el nodo raíz. Luego se desprende una flecha por cada valor posible del atributo. Los ejemplos de entrenamiento se clasifican de acuerdo con las alternativas existentes y, por cada una de ellas, se crean nuevos nodos repitiendo el mismo procedimiento, es decir, seleccionando siempre el mejor atributo. Durante la construcción de un árbol de decisión, el algoritmo (en su versión original) nunca retrocede para reconsiderar las elecciones previas (a esta característica se le denomina “voracidad”).

Para seleccionar el atributo que estará asociado a cada nodo del árbol, se define una propiedad llamada “ganancia de información”, que mide cuán bien un atributo determinado separa los ejemplos de entrenamiento de acuerdo con la clasificación que se persigue. C4.5 utiliza esta información para seleccionar atributos entre candidatos en cada paso, mientras el árbol se va construyendo.

La ganancia de información se basa en el concepto de entropía, el cual se utiliza en la teoría de la información para medir el grado de pureza de una determinada colección de datos de entrenamiento. En general, si los posibles valores del atributo a_i ocurren con probabilidades $P(a_i)$, entonces el contenido de información o entropía E del conjunto de observaciones está dado por:

$$E(T) = E(P(a_i), \dots \dots P(a_n)) = \sum_{i=1}^{n} -P(a_i) \log_2 P(a_i) \quad \dots\dots\dots (4.1)$$

Una vez realizada la prueba sobre los distintos atributos, se divide el conjunto de entrenamiento según el mejor atributo. Para encontrarlo, se utilizan los principios de la teoría de la información, que sostiene que esta se maximiza cuando la entropía se

minimiza. Una manera de cuantificar la bondad de un atributo en dicho contexto consiste en considerar la cantidad de información que proveerá, tal y como esto se define en la teoría de la información. Un bit de información es suficiente para determinar el valor de un atributo booleano (por ejemplo, sí/no, verdadero/falso, 1/0, etc.) sobre el cual no sabemos nada.

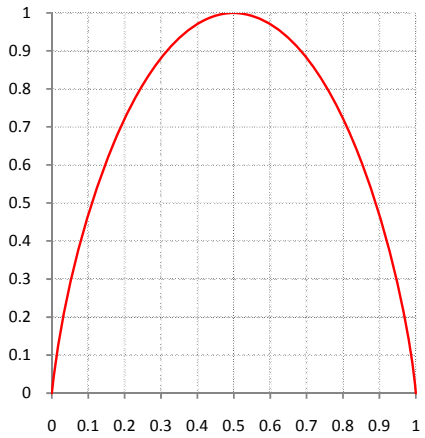

Figura 4.2. Función entropía relativa a una clasificación booleana

Consideremos el caso booleano aplicando esta ecuación a un lanzamiento de una moneda. Tenemos que la probabilidad de obtener cara o sello es de 50% para cada una:

$$\mathrm{E}\left(\frac{1}{2},\frac{1}{2}\right) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} - = 1$$

Ejecutar el lanzamiento nos provee 1 bit de información; de hecho, nos provee la clasificación del experimento: si fue cara o sello. Si el mismo experimento se ejecuta con una moneda cargada que da 99% de las veces sello, entonces E(1/100, 99/100) = 0,08 bits de información, menos que en el caso anterior, porque ahora tenemos más evidencia sobre el posible resultado del experimento. La gráfica de la función de entropía se muestra en la figura 4.2.

Si todos los ejemplos son positivos o negativos (por ejemplo, si pertenecen todos a la misma clase), la entropía será 0. Una posible interpretación de esto es considerarla como una medida de ruido o desorden en los ejemplos.

Definimos la ganancia de información (gain) como la reducción de la entropía causada por dividir un conjunto de entrenamiento con respecto a un atributo.

Supongamos que tenemos una prueba posible con n resultados que dividen al conjunto T de entrenamiento en los subconjuntos $\{T_1, T_2, \ldots\ldots\ldots T_n\}$. Si la prueba se

realiza sin explorar las divisiones subsiguientes de los subconjuntos T_i, la única información disponible para evaluar la partición es la distribución de clases en T y sus subconjuntos.

Consideremos una medida similar luego de que T ha sido dividido de acuerdo con los n resultados de la prueba X. La entropía puede determinarse como la suma ponderada de los subconjuntos de la siguiente manera:

$$E(T,X) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} . E(T_i) \quad \ldots\ldots\ldots (4.2)$$

La cantidad $\text{gain}(T,X) = E(T) - E(T,X)$ mide la información ganada al partir T de acuerdo con la prueba X. El criterio de ganancia, entonces, selecciona la prueba que maximice la obtención de información. Es decir, antes de dividir los datos en cada nodo, se calcula la ganancia que resultaría de dividir el conjunto de datos según cada uno de los atributos posibles. Finalmente, se realiza la partición que culmina en la mayor ganancia.

El criterio de ganancia tiene la desventaja de que presenta una tendencia muy fuerte a favorecer los atributos que contienen muchos valores. Analicemos, por ejemplo, una prueba sobre un atributo que sea la clave primaria de un conjunto de datos. Obtendremos un único subconjunto para cada caso, y para cada subconjunto tendremos $gain(T,X) = 0$; entonces, la ganancia de información será máxima. Desde el punto de vista de la predicción, este tipo de división no es útil.

Esta tendencia inherente al criterio de ganancia puede corregirse con una normalización, mediante la cual se ajusta la ganancia aparente, atribuible a pruebas con muchos resultados. Como ejemplo, consideremos el contenido de información de un mensaje correspondiente a los resultados de las pruebas. Por analogía con la definición de la $gain(T)$, tenemos:

$$split\ info(T,X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} . \log_2 \left(\frac{|T_i|}{|T|}\right) \quad \ldots\ldots\ldots (4.3)$$

Esto representa la información potencial generada al dividir T en n subconjuntos, mientras que la ganancia de información mide la información relevante a una clasificación que nace de la misma división.

Entonces:

$$gain\ ratio(T,X) = \frac{gain\ (T,X)}{split\ info(T,X)} \quad \ldots\ldots\ldots (4.4)$$

El C4.5 usa por defecto el $\text{gain ratio}(X)$ para el cálculo de ganancias.

En la figura 4.3, se muestra el algoritmo para construir árboles de decisión.

ENTRADA: Conjunto de entrenamiento T
Conjunto de atributos $\{a\} = < a_1, a_2 \ldots . a_n >$
Clase C = $\{C_1, C_2 \ldots .\}$
PROCESO: Función C4.5 (T, $\{a\}$, C)

1. Si T se encuentra vacio
   Devolver un árbol vacio sin nodos. Ir al paso 8
2. Si todas las instancias de T tienen el mismo valor de C
   Devolver un árbol con un solo nodo, de valor C. Ir al paso 8
3. Calcular $\forall\ a_i \in \{a\}$ el $gain\ ratio(T, a_i)$
4. Sea el atributo $a_q \leftarrow Argmax_{(a_i \in \{a\})}\ gain\ ratio(T, a_i)$
5. Sean $d_1, d_2 \ldots . d_m$ los valores del atributo a_q y $T_1, T_2 \ldots T_m$ los respectivos conjuntos correspondientes a los valores de d_j.
6. Crear un nodo t con etiqueta a_q y crear ramas etiquetadas con los valores de d_j
7. Regresar al paso 1 con, C4.5 (T_1, $\{a\} - a_q, C$)…. C4.5 (T_m, $\{a\} - a_q, C$).
8. Terminar proceso

SALIDA: Un árbol de decisión.

Figura 4.3. Pseudocódigo para la creación del árbol de decisión

Para entender mejor cómo se que trabaja esta técnica, veremos un ejemplo sencillo. Se trata de uno muy parecido al difundido en la literatura de árboles de decisión [Mitc-97]. Este describe la situación de un conjunto de estudiantes que cursan diferentes asignaturas, teniendo en consideración tres atributos adicionales (ver tabla 4.1): el número de asignaturas matriculadas, la vez de matrícula y su promedio ponderado acumulado que, por esta vez, se ha considerado como un valor no numérico.

Cursos matriculados	Cursos	Vez de matrícula	PPA	Promedio
uno	C1	segunda	malo	SUSP
uno	C1	segunda	bueno	SUSP
uno	C2	segunda	malo	APROB
dos	C3	segunda	malo	APROB
tres	C3	primera	malo	APROB
tres	C3	primera	bueno	SUSP
tres	C2	primera	bueno	APROB
dos	C1	segunda	malo	SUSP
tres	C1	primera	malo	APROB
dos	C3	primera	malo	APROB
dos	C1	primera	bueno	APROB
dos	C2	segunda	bueno	APROB
uno	C2	primera	malo	APROB
dos	C3	segunda	bueno	SUSP

Tabla 4.1. Conjunto de instancias en el ámbito de la educación [Mitc-97]

Tenemos que, si consideramos que T representa al conjunto de entrenamiento, $p(c_1)$ es la probabilidad de que la clase sea SUSP (asignatura desaprobada) y $P(c_2)$ la

probabilidad de que la clase sea APROB (asignatura aprobada), la entropía está dada por:

$$E(T) = -\sum_{i=1}^{n} P(c_i) \log_2 P(c_i) \quad \text{……. (4.5)}$$

A partir de todos los datos disponibles, el C4.5 analiza todas las divisiones posibles según los distintos atributos, y calcula la ganancia y/o la proporción de ganancia.

Del conjunto de datos, se observa en la clase que SUSP = 5, APROB = 9; entonces, $P(c_1) = \frac{5}{14}$ y $P(c_2) = \frac{9}{14}$; por lo tanto, la entropía del sistema es:

$$\boldsymbol{E(T) = -\frac{5}{14}\log_2\frac{5}{14} - \frac{9}{14}\log_2\frac{9}{14} = 0.9403\ bits}$$

Luego se analiza el atributo cursos matriculados (x=cursos matriculados)

$$E(T,X) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} . E(T_i)$$

Donde T_1 es un conjunto cuyos valores son uno; T_2, dos; T_3, tres, y X corresponde al atributo cursos matriculados.

$$E(T,X) = \frac{4}{14}(E(T_1)) + \frac{6}{14}(E(T_2)) + \frac{4}{14}(E(T_3)) = 0.9111$$

$$gain(T, Cursos\ Matri) = E(T) - E(T, Cursos\ Matri) = \mathbf{0.0292\ bit}$$

$$split\ gain(Curso\ Matri) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} . \log_2\left(\frac{|T_i|}{|T|}\right) = -\frac{4}{14}\log_2\frac{4}{14} - \frac{6}{14}\log_2\frac{6}{14} - \frac{4}{14}\log_2\frac{4}{14}$$

$\boldsymbol{split\ gain(Curso\ Matri)}$ **= 1.5567 bits**

Entonces, la proporción de ganancia es:

$$gain\ ratio(Curso\ Matri) = \frac{gain\ (Curso\ Matri)}{split\ gain(Curso\ Matri)} = \mathbf{0.0188}$$

De igual forma para los demás atributos, de los que se obtiene:

- $\boldsymbol{gain\ ratio(Curso\ Matri) = 0.0188}$
- $\boldsymbol{gain\ ratio(Curso) = 0.1565.}$
- $\boldsymbol{gain\ ratio}(Vez\ de\ matricula) = \mathbf{0.1518}$
- $\boldsymbol{gain\ ratio}(PPA) = \mathbf{0.0488}$

Se puede observar que el atributo curso tiene la mayor proporción de ganancia, es decir, maximiza la ganancia cuando se elige ese atributo; por lo tanto, se elige como nodo raíz. Haciendo el proceso recursivo, finalmente obtenemos el árbol mostrado en la figura 4.4 y cuyo detalle se presenta en el anexo A.1.2.

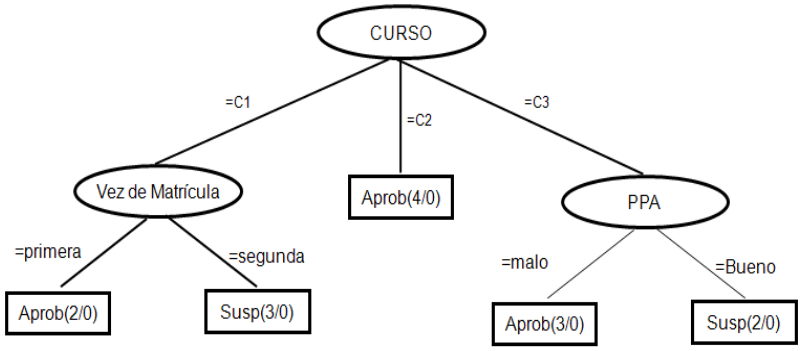

Figura 4.4. Árbol de decisión correspondiente a las instancias de la tabla 4.2

Los árboles de decisión pueden generarse a partir de atributos discretos o continuos. Cuando se trabaja con atributos discretos, la partición del conjunto, según el valor de algunos atributos, es simple. Por ejemplo, agrupamos a todos los alumnos que llevaron una asignatura por primera vez y los aislamos de los que lo hicieron en más de una oportunidad. En el caso de los atributos numéricos, esta división no es tan simple. Por ejemplo, si queremos dividir los conjuntos según el promedio ponderado acumulado, es casi imposible encontrar en un conjunto muchos estudiantes que lo tengan idéntico.

Para solucionar este problema, puede recurrirse a la binarización. Este método consiste en formar dos rangos de valores de acuerdo con un atributo, que puede tomarse como simbólico. Por ejemplo, el alumno calificado con más de 12.5 de promedio ponderado acumulado y el que tenga una nota menor.

Se construye el árbol y, si alguna de las ramas tiene una entropía igual a cero, el nodo se convierte en una hoja. Para el resto de nodos, el procedimiento se repite descartando los atributos que se hayan considerado en el árbol, de modo que estos solo aparezcan una vez, como máximo, en la rama (aunque existen variaciones de esta regla). El proceso continúa en cada rama hasta que todos los atributos se incluyan en el árbol o todos los ejemplos de entrenamiento asociados con el nodo tengan el mismo atributo objetivo (es decir, la entropía cero).

El algoritmo C4.5 busca el conjunto de todos los árboles de decisión consistentes con los datos de entrenamiento, lo que se denomina “espacio de hipótesis”. El algoritmo C4.5 ejecuta una búsqueda comenzando con el árbol vacío y considerando poco a poco hipótesis más elaboradas hasta encontrar un árbol de decisión que clasifique correctamente los datos de entrenamiento. El algoritmo prefiere, en primer lugar, árboles

pequeños (en términos de proximidad a la raíz del árbol), y, en segundo lugar, aquellos árboles que colocan los atributos informativos más cerca de la raíz.

El sobreajuste ocurre cuando el modelo resultante se restringe específicamente a un conjunto de entrenamiento que presenta ruido o tiene un número pequeño de ejemplos, y se descubre cuando el modelo no puede predecir correctamente ejemplos distintos a los de entrenamiento. Las supuestas regularidades de los datos con los que se construyó el modelo provocaron que fallaran las predicciones hechas sobre una nueva base de datos. En estos casos, la hipótesis tiene un error de entrenamiento bajo y uno de generalización alto. En la figura 4.5 [Mitc-97], podemos observar que los nuevos casos producen errores en la clasificación debido al sobreajuste del árbol generado.

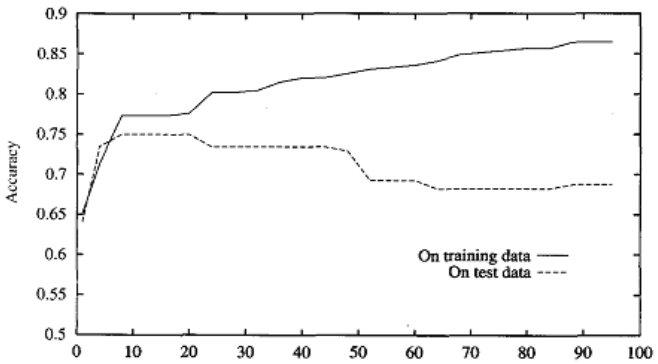

Figura 4.5. Efecto de la reducción del error basado en poda

La sobre generalización, la evaluación de atributos poco importantes o significativos y el gran tamaño del árbol obtenido causan el sobreajuste; para evitar eso, se aplican técnicas que se encargan de recortarlo y hacerlo más pequeño. Estas reciben el nombre de técnicas de poda.

En el primer caso, un árbol puede construirse a partir de ejemplos con ruido, por lo cual algunas ramas pueden ser engañosas. En cuanto a la evaluación de atributos no relevantes, estos deben podarse, ya que solo agregan niveles, mas no contribuyen a la ganancia de información. Por último, si el árbol obtenido es demasiado profundo o demasiado frondoso, se dificulta la interpretación por parte del usuario.

Existen, básicamente, dos maneras de modificar el método de división recursiva para producir árboles más simples: decidir no dividir más de un conjunto de casos de entrenamiento, o remover retrospectivamente alguna parte de la estructura construida por la división recursiva.

El primer enfoque, conocido como prepoda, tiene la ventaja de que ahorra tiempo al construir una estructura que luego será simplificada en el árbol final. Los sistemas que lo aplican generalmente buscan la mejor manera de partir el subconjunto y evalúan la partición desde el punto de vista estadístico mediante la teoría de la ganancia de información, reducción de errores, etc. Si esta evaluación es menor que un límite

predeterminado, la división se descarta y el árbol para el subconjunto es simplemente la hoja más apropiada. Sin embargo, este tipo de método tiene la desventaja de que no es fácil detener una división en el momento adecuado: un límite muy alto puede terminar con la partición antes de que los beneficios de particiones subsiguientes parezcan evidentes, mientras que un límite demasiado bajo resulta en una simplificación muy leve.

El C4.5 utiliza el segundo enfoque, el método de “divide y reinarás”: procesa los datos de entrenamiento libremente, y el árbol sobreajustado producido se poda después. Los procesos computacionales adicionales invertidos en la construcción de partes del árbol que luego se podarán pueden ser sustanciales, pero el costo no supera los beneficios de explorar una mayor cantidad de particiones posibles. El crecimiento y la poda de los árboles son más lentos, pero más confiables.

Cualquiera sea la técnica elegida, se deberá tener un criterio definido para determinar el tamaño final y correcto del árbol. Para tal efecto, se suelen aplicar algunas de las siguientes técnicas:

- Usar un conjunto separado de ejemplos distintos de los de entrenamiento con la finalidad de evaluar las técnicas de poda.

- Usar todos los datos disponibles para el entrenamiento, pero aplicar una prueba estadística con el propósito de saber si se debe expandir (o podar) un nodo determinado.

La más común es la primera y se conoce como “técnica de prueba y validación”. Todos los datos disponibles se dividen en dos conjuntos de ejemplos: uno de entrenamiento (que se usa para obtener la hipótesis) y uno de validación (que se emplea para evaluar la exactitud de la hipótesis y, además, para evaluar el impacto de podar el espacio de hipótesis). La justificación es que, aun cuando existen errores aleatorios y coincidencias regulares en los datos de entrenamiento, es poco probable que los de validación exhiban esas mismas fluctuaciones. Por lo tanto, el conjunto de validación puede proporcionar una verificación del posible sobreajuste existente. Ahora bien, el conjunto de datos debe ser lo suficientemente grande; se acostumbra usar dos tercios de los ejemplos disponibles para el entrenamiento, y el tercio restante, para la validación. Con ese tamaño, la poda puede realizarse efectivamente.

Algunos otros métodos de poda que serán revisados en este trabajo aplican el método que consiste en dividir el conjunto total de datos en tres subconjuntos: conjunto de crecimiento, conjunto de poda y conjunto de prueba, en donde la unión de los dos primeros corresponde al conjunto de entrenamiento. En estos procedimientos, el conjunto de poda y el conjunto de entrenamiento se emplean para aprender dos árboles de decisión, los cuales reciben el nombre de “árbol de crecimiento” y “árbol de entrenamiento”, respectivamente. El primero se usa para los métodos que requieren un conjunto independiente para podar un árbol de decisión. Recíprocamente, el árbol de

entrenamiento se utiliza para los métodos que explotan el conjunto de entrenamiento solo.

En el árbol, los nodos se podan iterativamente, eligiendo siempre el nodo que, una vez eliminado, incremente la exactitud del árbol de acuerdo con los datos de validación. Se continúa podando hasta afectar la exactitud.

La poda de los árboles de decisión llevará, sin duda, a clasificar erróneamente una mayor cantidad de los casos de entrenamiento. Por lo tanto, las hojas de un árbol podado no contendrán necesariamente una única clase, sino, como se explicó con anterioridad, una distribución de clases. Asociada con cada hoja, habrá una que especificará la probabilidad de que un caso de entrenamiento en la hoja pertenezca a determinada clase.

Generalmente, la simplificación de los árboles de decisión se realiza descartando uno o más subárboles y reemplazándolos por hojas. Al igual que en la construcción de árboles, las clases asociadas con cada una de las hojas se encuentran al examinar los casos de entrenamiento cubiertos por ellas y eligiendo el caso más frecuente. Además de este método, el C4.5 permite reemplazar un subárbol por alguna de sus ramas.

En el supuesto de que fuera posible predecir la tasa de error de un árbol y sus subárboles, esto inmediatamente llevaría al siguiente método de poda [Quin-93]:
"*Comenzar por las hojas y examinar cada subárbol. Si un reemplazo del subárbol por una hoja o por su rama más frecuentemente utilizada lleva a una menor tasa de errores predicha, entonces se debe podar el árbol de acuerdo con ello, recordando que las proporciones de errores predichas para todos los subárboles que lo contienen se verán afectadas*".

Como la tasa de errores predicha para un árbol disminuye si disminuyen las proporciones de errores predichas en cada una de sus ramas, este proceso generaría un árbol con una tasa de errores mínima.

Está claro que calcular la tasa de errores a partir de los datos de entrenamiento para los cuales el árbol se construyó no es un estimador útil, ya que, en lo que respecta al conjunto de entrenamiento, la poda siempre aumenta dicha tasa. Existen, sin embargo, dos clases de técnicas para predecirla:

- Poda según la complejidad del costo [Brei-84], en la cual la tasa de errores predicha para un árbol se modela como la suma ponderada de su complejidad y sus errores en los casos de entrenamiento mediante los casos extras utilizados para determinar los coeficientes de la ponderación.
- Poda de reducción de errores (Reduced-error pruning) [Quin-87], que evalúa la tasa de errores de un árbol y sus componentes directamente a partir del nuevo conjunto de casos.

El enfoque tomado por el C4.5 pertenece a la segunda familia de técnicas que utilizan únicamente el conjunto de entrenamiento a partir del cual se construyó el árbol.

El problema reside en decidir cómo y cuándo realizar la poda, cuyo objetivo principal es disminuir el sobreajuste. Quinlan propone utilizar la tasa de error como indicador de poda mediante el método de poda pesimista. Este método consiste en estimar la tasa de error para cada rama, que se define como la suma de las tasas de los errores de las hojas que se derivan de dicha rama, y la tasa de error para la hoja que la sustituiría. Si la tasa de error estimada para la nueva hoja es menor que la tasa de error calculada para la rama (la que se va a reemplazar por la hoja), se sustituye la rama por la nueva hoja. Bajo el supuesto de que la tasa de error es N/E, siendo E el número de eventos y N el número de pruebas, se puede decir que la probabilidad de que se produzca un suceso sigue una distribución Binomial $B(N,p)$. Por tanto, E es una variable aleatoria binomial y su probabilidad puede ser estimada mediante un intervalo de confianza.

Quinlan, en [Quinn-93], define la tasa de error estimada$\mathrm{U_{CF\%}(E,N)}$ como el límite superior del intervalo de confianza de nivel CF para una distribución binomial $B(N,p)$, donde CF es el factor de confianza, E el número de instancias mal clasificadas(instancias erróneas) y N el número de instancias totales de la hoja.

La probabilidad de error no puede determinarse de forma exacta, pero cuenta con límites de confianza. Para un límite de confianza CF, el límite superior de esta probabilidad puede encontrarse a partir de los límites de confianza para la distribución binomial; el límite superior se expresa como $\mathrm{U_{CF\%}(E,N)}$. Como en la distribución binomial los límites superior e inferior son simétricos, la probabilidad de que el promedio real de errores exceda $\mathrm{U_{CF\%}(E,N)}$ es CF/2.

El C4.5 simplemente iguala el estimador de error predicho de la hoja con su límite superior, bajo el argumento de que el árbol se construyó para minimizar la tasa de error observada.

Para simplificar el cálculo, las proporciones de error para las hojas y subárboles se calculan asumiendo que se utilizaron para clasificar un conjunto de nuevos casos del mismo tamaño que el conjunto de entrenamiento. Entonces, una hoja que cubre N casos de entrenamiento con un estimador de error predicho de $\mathrm{U_{CF\%}(E,N)}$ generaría N x $\mathrm{U_{CF\%}(E,N)}$ errores predichos. Análogamente, la cantidad de errores predichos asociados con un (sub)árbol es la suma de estos para cada una de sus ramas.

Definición 4.4.1. Se define la tasa de error estimada de una instancia (TEE) como el límite superior del intervalo de confianza del nivel de CF% para una distribución binomial. Esta TEE se denota con $\mathrm{U_{CF\%}(E,N)}$ y se expresa mediante la siguiente expresión:

$$\mathrm{U_{CF\%}(E,N)} = \frac{\left(\mathrm{f}+\frac{\mathrm{z}^2}{\mathrm{N}}+\mathrm{z}\sqrt{\frac{\mathrm{f}}{\mathrm{N}}-\frac{\mathrm{f}^2}{\mathrm{N}}+\frac{\mathrm{z}^2}{4\mathrm{N}^2}}\right)}{\left(1+\frac{\mathrm{z}^2}{\mathrm{N}}\right)}\;;\quad \mathrm{f}=\frac{\mathrm{E}+0.5}{\mathrm{N}} \qquad \mathrm{CF}\ \rightarrow\ \mathrm{z} \quad \text{..... (4.6)}$$

E son los errores cubiertos por la hoja (número de errores de la hoja), N son los casos cubiertos por la hoja (número total de casos o eventos), CF es el factor de confianza y f la probabilidad de error normalizado. En la tabla 4.2, se muestran los factores de confianza establecidos por el c4.5.

CF	Z
0.1%	3.09
0.5%	2.58
1%	2.33
5%	1.65
10%	1.28
20%	0.84
25%	0.69
40%	0.25

Tabla 4.2. Factores de confianza preestablecidos por el programa c4.5

Definición 4.4.2. Sea N el número de instancias totales de una hoja y E el número de instancias incorrectamente clasificadas, se define el **error estimado** (EE) en una hoja con la siguiente expresión:

$$\mathrm{EE} = \mathrm{N}\ \mathrm{x}\ \mathrm{U}_{\mathrm{CF\%}}(\mathrm{E}, \mathrm{N}) \quad \ldots\ldots\ldots (4.7)$$

Para entender mejor esto, podríamos considerar una hoja con seis instancias (casos) en total, tres de las cuales fueron mal clasificadas; entonces, su tasa de error para CF=25% (Z=0.69), es:

$$\mathrm{EE} = \mathrm{N}\ \mathrm{x}\ \mathrm{U}_{\mathrm{CF\%}}(3{,}6) = 4.27$$

4.4.2. Reglas basadas en los árboles de decisión

Para obtener las reglas, [Quin-93] propone un método basado en podas. Este implica que cada camino desde la raíz del árbol hasta llegar a un nodo hoja genere una condicional $A \rightarrow C$, llamada regla. Esta condicional tiene un antecedente (A), que representa la conjunción de la prueba (conjunción de atributos de una o un conjunto de instancias), y el consecuente (C), que representa a la clase o al nodo hoja. Todo ello se hace, asumiendo que el árbol debe ser recorrido de arriba hacia abajo. Luego se procede a ejecutar la poda al antecedente de cada regla. Para entenderlo mejor, se proponen cuatro pasos que se detallan a continuación:

Paso 1:

Partiendo del árbol de decisión sin podar, el primer paso natural es transformar cada una de las ramas en reglas, empezando por una hoja y subiendo por el árbol hasta alcanzar el nodo raíz. De esta forma, se generarán tantas reglas como hojas tenga el

árbol. Las reglas obtenidas no resuelven la dificultad de legibilidad de los árboles, debido a que existe una regla por cada hoja y pueden aparecer condiciones irrelevantes que no afectan la precisión. Por este motivo, es necesario un segundo paso en el que se eliminen las condiciones irrelevantes.

Paso 2:

Quinlan en [Quin-93] propone utilizar la "tasa error pesimista" TEP para determinar qué condiciones afectan en menor medida a la precisión de la regla.
En el ejemplo correspondiente a la figura 4.1, después de aplicar el algoritmo de reglas de decisión, obtenemos las siguientes reglas:

Rule 1:	**Curso = c1 ; Vez de matrícula = segundo**	**> class SUSP**
Rule 2:	**Vez de matrícula = primera**	> class APROB
Rule 3:	**Curso = c2**	> class APROB
Rule 4:	**Curso = c3 ; PPA = malo**	> class APROB
Rule 5:	**Curso = c3 ; PPA = bueno**	> class SUSP

Tabla 4.3. Reglas de decisión obtenidas a partir del árbol

Los antecedentes de las reglas provenientes del árbol sin podar, pueden contener condiciones irrelevantes, con lo cual la regla puede generalizarse eliminándolas. Para decidir cuándo una condición debe eliminarse, [Quin-93] propone el método basado en la tasa error pesimista, explicado a continuación:

Dada una regla R : $A \rightarrow C$, que proviene de recorrer cada nodo del árbol desde la raíz hasta la hoja, siendo la hoja el consecuente de la regla y cuyo valor está determinado por algunos de los valores que corresponden a la clase propuesta y el antecedente la conjunción de los elementos del conjunto de atributos A, que corresponden a los atributos instanciados de la forma: $\mathrm{a}_1 \wedge \mathrm{a}_2 \ldots\ldots\ldots \mathrm{a}_\mathrm{m}$. Se define la tasa de error pesimista, denotado por $\mathrm{TEP}(\mathrm{R},\mathrm{C})$ siendo R la regla y c su respectivo consecuente, de la siguiente manera:

Definición 4.4.3. La tasa de error pesimista, denotada por $\mathrm{TEP}(\mathrm{R},\mathrm{C})$, es la tasa de error estimada para la instancia que pertenece a una hoja después de la clasificación y se define por:

$$\mathrm{TEP}(\mathrm{R},\mathrm{C}) = \mathrm{U}_{\mathrm{CF\%}}(\mathrm{E},\mathrm{N})$$

Donde R es la regla y C es la clase, cuyo valor instanciado corresponde al valor del consecuente de dicha regla. Cabe recordar que E es el número de instancias mal clasificadas por la regla de un total de N instancias que la cubren.

Para proceder a eliminar los atributos con menor importancia de las reglas, [Quin-93] propone el siguiente algoritmo:

ENTRADA: Una regla R de la forma $a_1 \wedge a_2 \wedge a_3 \wedge a_4 \ldots\ldots\ldots \wedge a_n \longrightarrow c$
PROCESO:

1. Calcular $\mathrm{TEP}(\mathrm{R}, \mathrm{C})$
2. $\forall\, \mathrm{a_i}$ Calcular $\mathrm{TEP}(\mathrm{R} - \mathrm{a_i}, \mathrm{C})$
3. Determinar $\mathrm{TEP}(\mathrm{R} - \mathrm{a_m}, \mathrm{C}) = \min\{\mathrm{TEP}(\mathrm{R} - \mathrm{a_i}, \mathrm{C})\}$
4. Si $TEP < TEP(\mathrm{R}, \mathrm{C})$ pasar al paso 5
   4.1.- Si no terminar
5. Eliminar la condición a_m
6. Regresar al primer paso con $R = R - a_m$

SALIDA: Una Regla simplificada

Figura 4.6. Pseudocódigo para la eliminación de atributos irrelevantes de las reglas

Para entender mejor el algoritmo, lo aplicaremos a la regla 1 de la tabla 4.3 que corresponde al árbol de la figura 4.4.

Sea la regla 1:

Rule 1:	Curso = c1 ; Vez de matrícula = segunda	> class SUSP (3/0)

Donde

a_1 es la asignatura C1 y a_2 el atributo correspondiente a la vez de matrícula = segunda

1. Calcularemos TEP: $TEP(R_1, C) = U_{CF\%}(0.3) = 37\%$

2. Eliminar cada una de las condiciones por separado y determinar el nuevo TEP
   2.1. Al eliminar a_1 de R, obtenemos:

Rule 1:	Vez de matrícula = primera	> class APROB	(7/3)

   Por lo tanto:

   $TEP(R_1 - a_1, C) = U_{CF\%}(3{,}7) = 62.6\%$

   2.2. Al eliminar a_2 de R, obtenemos:

Rule 1:	Curso= C1	> class APROB	(5/2)

   Por lo tanto:

   $TEP(R_1 - a_2, C) = U_{CF\%}(2{,}5) = 64.7\%$

3. Determinar el mínimo TEP: $TEP(R - a_m, C) = min\{TEP(R_1 - a_1, C),\ TEP(R_1 - a_2, C)\}$

   $TEP(R - a_m, C) = min\{TEP(R_1 - a_1, C)\}$

   Por lo tanto, $a_m = a_1$

4. Como $TEP(R - a_1, C) > TEP(R_1, C)$

   Entonces no se elimina el atributo de la condicional $\mathrm{a_1}$.

5. Por lo tanto, la regla quedará igual:

Rule 1:	Curso=C1, Vez de matrícula = primera	> class SUSP	(63.0%)

Debido a que el error pesimista de la regla base es menor que el generado cuando se suprime una condición, las demás reglas no se podan.

Rule 2:	Vez de matrícula = primera	**> class APROB 66.2%**
Rule 3:	**Curso = c2**	> class APROB 70.7%
Rule 4:	**Curso = c3 ; PPA = malo**	> class APROB 63%
Rule 5:	**Curso = c3 ; PPA = bueno**	> class SUSP 50%

Tabla 4.4. Reglas de decisión que no se podan

Paso 3:

Una vez obtenidas las reglas, Quinlan, en [Quin-93], sugiere obtener un subconjunto de reglas SR que minimice los errores. Esto debido a que algunas de las reglas obtenidas en el paso anterior pueden tener un error muy grande. Para llevar a cabo esto, primero suponemos que, del conjunto de todas las reglas desarrolladas, seleccionamos un subconjunto de ellas, que lo denotaremos por SR que cubre una clase en particular C. La efectividad de este subconjunto puede ser obtenida mediante el número de casos de entrenamiento cubiertos por SR que no pertenecen a la clase C (falsos positivos), y el número de clases C de casos de entrenamiento que no están cubiertos por alguna regla en SR (falsos negativos).

Quinlan propone, para calcular la pureza del subconjunto SR y seleccionar el mejor, utilizar el principio de Minimum Description Length (MDL) [Quin-93]. Este principio consiste en que el mensaje óptimo para enviar entre un emisor y un receptor es aquel que minimice el número de bits requerido para codificar el mensaje. Supongamos que el emisor y el receptor tienen el mismo conjunto de entrenamiento, pero tan sólo el receptor conoce la clase de cada caso de entrenamiento. El emisor debe enviar esta información al receptor por medio de un mensaje. Este está conformado, en nuestro caso, por dos conjuntos. El primero, el subconjunto de reglas con las que el receptor pueda clasificar cada uno de sus casos, y el segundo, errores (falsos positivos y falsos negativos).

Ahora será necesario calcular el costo de codificar una regla, para ellos se define:

Definición 4.4.4. Sea R una regla R de la forma R: $A \rightarrow C$ con $A = \mathrm{a}_1 \wedge \mathrm{a}_2 \ldots\ldots\ldots \mathrm{a}_\mathrm{n}$, se define el costo de codificar una regla como [Quin-93].

$$C_R = Log_2(n!) \ldots\ldots\ldots (4.8)$$

En nuestro caso:

Rule 1: $C_{R1} = \log_2 2! = 1$

Rule 2: $C_{R2} = \log_2 1! = 0$

Rule 3: $C_{R3} = \log_2 1! = 0$

Rule 4 $C_{R4} = \log_2 2! = 1$

Rule 5 $C_{R5} = \log_2 2! = 1$

Definimos los subconjuntos asociados a cada clase: reglas que cubren APROB {R2, R3, R4}, subconjunto SR_1 y reglas que cubren SUSP {R1, R5}, subconjunto SR_2.

Definición 4.4.5. Sea SR el subconjunto de N reglas, se define el costo de codificar todas esta reglas como [Quin-93].

$$C_T = \sum_{i=1}^{N} C_{Ri} - \text{Log}(N!) \quad \ldots\ldots\ldots \quad (4.9)$$

Para nuestro caso, tomado el subconjunto de aprobados SR_1.

$$\mathbf{C_t = \sum_{i=1}^{N} C_{Ri} - \log_2(N!) = C_{R2} + C_{R3} + C_{R4} - \log_2(3!)}$$

$$\mathbf{C_t = 1 - 2.58 = -1.58}$$

Definición 4.4.6. Dado un conjunto de entrenamiento de **n** casos y sea **r** el número de casos cubiertos por el subconjunto de reglas SR, se define el costo de codificar los errores como:

$$C_t = Log_2\left(\left(\frac{r}{fp}\right)\right) + Log_2\left(\left(\frac{n-r}{fn}\right)\right) \quad \ldots\ldots\ldots (4.10)$$

Siendo fp y fn el número de ejemplos falsos positivos y falsos negativos presentes.

Para nuestro caso:

Para S_1

Casos cubiertos $r = 10$ y fp = 1 , fn = 0

$$C_e = \log_2\left(\binom{10}{1}\right) + \log_2\left(\binom{4}{0}\right) = \log_2 10 + \log_2 1 = 3.32$$

Definición 4.4.7. Se define el costo total de codificación como [Quin-93]:

$$C_T = C_e + Wx\, C_t \text{ Siendo } W\epsilon\,[0,1] \quad \ldots\ldots\ldots (4.11)$$

Entonces calculamos el costo total para el subconjunto S_1 (Tomamos $W = 0.5$)

$$C_T = 3.32 + 0.5x(-1.58) = 2.53$$

En conclusión, el subconjunto SR que minimice el número total de falsos negativos y de falsos positivos es el que tendrá el mínimo costo total de codificación.

Para ilustrar el paso tres, regresemos al conjunto de datos de entrenamiento de la tabla 4.1:

S	C_t	fp	fn	C_e	C_T
{-}	0	0	9	10.96	10.96
{R2}	0	1	3	7.93	7.93
{R3}	0	0	5	7.97	7.97
{R4}	1	0	6	8.85	9.35
{R2,R4}	0	1	2	6.90	6.90
{R2,R3}	-1	1	1	5.49	4.99
{R3,R4}	0	0	2	4.39	4.39
{R2,R3,R4}	-1.58	1	0	3.32	2.53

Tabla 4.5. Costo de codificación de subconjuntos

Por lo tanto, se observa que el menor costo asociado a la codificación de las reglas está presente en {R2, R3, R4}.

El siguiente paso es calcular el número de falsos positivos por cada subconjunto encontrado; por lo tanto, aquel que posea la menor cantidad de falsos positivos, se posicionara en la cabecera del conjunto de reglas; además, en cada subconjunto de reglas existe un ordenamiento interno en base al acierto estimado de probabilidad.

Paso 4:

Una vez que ya se ha encontrado un subconjunto de reglas para representar cada clase, queda determinar el ordenamiento para las clases y seleccionar un valor por defecto. Al decidir el ordenamiento de las clases, es importante considerar los falsos positivos, ya que ocasionarán clasificaciones incorrectas. Entonces, cuando se dispone el ordenamiento, se elige primero la clase que los representa en menor cantidad. Luego, los falsos positivos de los casos de entrenamiento que aún no han sido seleccionados se recalculan y se selecciona nuevamente la clase que presente menos casos, y así sucesivamente.

Como la clase por defecto se utilizará cuando un caso no sea cubierto por ninguna de las reglas, estas deberían involucrarse para determinarla. El C4.5 elige aquella que cubre la mayoría de los casos de entrenamiento no considerados por ninguna regla, y resuelve empates a favor de la clase con la mayor frecuencia absoluta. Cuando ya se ha determinado el ordenamiento y la clase por defecto, el conjunto de reglas se examina por última vez. Si existe alguna regla cuya eliminación reduzca el número de errores de clasificación, se elimina y se procede a calcular los errores. El conjunto completo vuelve a verificarse. Este paso se diseñó para evaluar el conjunto de reglas en la forma en que será utilizado.

Ahora procederemos a calcular el número de falsos positivos por cada subconjunto encontrado; por lo tanto, aquel que posea la menor cantidad de falsos positivos, se posicionara en la cabecera del conjunto de reglas y, además, en cada subconjunto de reglas existe un ordenamiento interno en base al acierto estimado de probabilidad.

Después de ellos, se obtiene la tabla de salida que proporciona el programa C4.5; este se puede visualizar en la figura 4.6:

Rule	Size	Error	Used	Wrong	Advantage	Class
1	2	37.0%	3	0(0.0%)	3(3|0)	SUSP
5	2	50.0%	2	0(0.0%)	2(2|0)	SUSP
3	1	29.3%	4	0(0.0%)	0(0|0)	APROB
2	1	33.8%	4	0(0.0%)	0(0|0)	APROB
4	2	37.0%	1	0(0.0%)	0(0|0)	APROB

Tabla 4.6. Estadísticas de las reglas de decisión

4.4.3. Clasificación por la técnica de Naïve Bayes

El clasificador bayesiano se aplica a las tareas de aprendizaje donde cada instancia x es descrita por un conjunto de atributos y donde la función objetivo $F(x)$ puede tomar algún valor de algún conjunto finito V. Al igual que en los árboles de decisión, el clasificador bayesiano depende de un conjunto de ejemplos de entrenamiento, y lo que se pide es clasificar a una nueva instancia, descrita por valores en forma vectorial $< a_1, a_2, a_3 \dots \dots a_n >$.

En muchos escenarios de aprendizaje, se considera un cierto conjunto de hipótesis candidatas H y generalmente estamos interesados en encontrar la hipótesis más probable $h \in H$ dado un dato observado D. Cualquier hipótesis probable máxima es llamada una hipótesis máxima a posteriori (MAP); esta se puede determinar usando el teorema de Bayes para calcular la probabilidad a posteriori de cada hipótesis candidata. Más precisamente, diremos que:

$$h_{MAP} = \operatorname{argmax}_{h \epsilon H} P(h|D) = \operatorname{argmax}_{h \epsilon H} \frac{P(D|h).P(h)}{P(D)} =$$

$$\operatorname{argmax}_{h \epsilon H} P(D|h)P(h) \dots\dots\dots\dots\dots\dots (4.12)$$

La aproximación bayesiana para clasificar la nueva instancia consiste en encontrar el valor objetivo que es una hipótesis máxima a posteriori V_{MAP} , dados los valores de los atributos que describen una cierta instancia.

$$V_{MAP} = \operatorname{argmax}_{V_j \epsilon V} P(V_j | a_1, a_2, \dots \dots \dots, a_n) \quad \dots\dots\dots\dots\dots\dots (4.13)$$

Usando el teorema de Bayes, rescribiremos la expresión:

$\mathrm{V_{MAP}} = \mathrm{argmax_{V_j \epsilon V}}\, \mathrm{P}(\mathrm{a_1}, \mathrm{a_2}, \ldots\ldots\ldots, \mathrm{a_n} | \mathrm{V_j})\, \mathrm{P}(\mathrm{V_j})$ ……..(4.14)

Ahora se podría intentar determinar los dos términos de la ecuación anterior basados en datos de entrenamiento. Es fácil estimar cada uno de los $P(V_j)$, simplemente contando la frecuencia con la cual cada valor objetivo V_j aparece en el dato de entrenamiento.

El clasificador de Bayes está basado en la asunción simplificada de que los valores de los atributos son condicionalmente independientes dado el valor objetivo. En otras palabras, la asunción es que, dado el valor objetivo de la instancia, la probabilidad de observar la conjunción $a_1, a_2, \ldots\ldots\ldots a_n$ es justo el producto de probabilidades para los atributos individuales; es decir:

$\mathrm{P}(\mathrm{a_1}, \mathrm{a_2}, \ldots\ldots\ldots, \mathrm{a_n} | \mathrm{V_j}) = \prod_{i=1}^{n} P(a_i|V_j)$……..(4.15)

Sustituyendo esta última ecuación en la anterior, tenemos la fórmula de Naïve Bayes:

$V_{NB} = argmax_{V_j \epsilon V}\, P(V_j). \prod_{i=1}^{n} P(a_i|V_j)$……..(4.16)

Donde V_{NB} denota el valor de la salida de la función objetivo por intermedio del clasificador.

El clasificador bayesiano simple es un método probabilístico de clasificación que puede ser usado para determinar la probabilidad de que un elemento j pertenezca a una clase C_i dados los valores $[V_{1j}, V_{2j}, V_{3j} \ldots\ldots\ldots V_{nj}]$ de los atributos $[A_1, A_2, A_3 \ldots\ldots\ldots A_n]$ del elemento j:

$P(C_i \,|\, A_1 = V_{1j}\ \&\ A_2 = V_{2j}\ \&\ \ldots\ldots\ldots\ldots\ldots\ldots\ \&A_N = V_{Nj})$ ……..(4.17)

Si los valores de los atributos son independientes, esta probabilidad es proporcional a:

$P(C_i). \prod_{k=1}^{n} P(A_k = V_{kj}|C_i)$……..(4.18)

La probabilidad de cualquier elemento de pertenecer a la clase C_i

La probabilidad del atributo $A_k = V_{kj}$ de pertenecer a elementos de la clase C_i

Para determinar la clase más probable de un elemento, se calcula la probabilidad de cada clase y el elemento se atribuye a la clase con mayor probabilidad.

ENTRADA: Conjunto de entrenamiento T
Conjunto de atributos $\{a\} = < a_1, a_2 \ldots . a_n >$
Clase C = $\{C_1, C_2 \ldots .\}$

PROCESO:

1. Para $\forall\, c_i \in C$ hacer
   1.1 Para $\forall\, a_j \in \{a\}$ hacer
      1.1.1 Sean $d_1, d_2 \ldots . d_m$ los valores del atributo a_j hacer
         -. Determinar la frecuencia de instancias de d_p como frec(d_p) y el número de instancias del atributo a_j como frec(a_j)
         -. Calcular la probabilidad P usando el estimador de Laplace
         $$P = \frac{frec\,(d_p)+1}{frec\,(a_j)+m}$$
   1.2 Calcular $P(C_i) = \frac{|C_i|}{N}$ siendo N el número total de instancias
   1.3 Sea una nueva instancia x_q y los valores de sus atributos $< a_{1x_q}, a_{2x_q} \ldots . a_{nx_q} >$
      Calcular $P_i = P(C_i) * \prod_{j=1}^{n} P(a_j = a_{jx_q} | C_i)$
2. Normalizamos las probabilidades $\forall i\ P_i = \frac{P_i}{P_1+P_2+\cdots.}$
3. $V_{NB} = Argmax_{\forall\, c_i \in C} P_i$
4. Terminar proceso

SALIDA: Función objetivo estimado V_{NB} para la instancia x_q .

Figura 4.7. Pseudocódigo del algoritmo de Naïve Bayes

Para entender mejor cómo es que trabaja esta técnica, predeciremos la clase correspondiente a un estudiante con los siguientes atributos:

$$a_i = curso\ matric. = tres; curso = c3, vez = primera; PPA = bueno)$$

De acuerdo con los datos de entrenamiento de la tabla 4.1., nuestra tarea es predecir el valor objetivo (aprobado/suspenso) del concepto objetivo promedio para la nueva instancia de prueba. El objetivo es calcular el valor V_{NB} en:

$$V_{NB} = argmax_{V_j \epsilon V}\, P(V_j). \prod_{i=1}^{n} P(a_i|V_j) \ldots\ldots\ldots\ldots\ldots (4.19)$$

Instanciando la expresión anterior, obtenemos:

$$V_{NB} = argmax_{aprob}\, P(aprob). P(C.Mat = tres|aprob). P(curso = c3|aprob).$$
$$P(vez = prim|aprob). P(PPA = bueno|aprob)$$

Procederemos a calcular cada una de las probabilidades condicionales según los datos del conjunto de entrenamiento para $V_j = aprob$:

$$P(C.Mat = tres|V_j = aprob) = \frac{P(tres \cap aprob)}{P(aprob)} = \frac{\frac{4}{14}\cdot\frac{3}{4}}{\frac{9}{14}} = \frac{1}{3}$$

$$P(curso = c3|V_j) = \frac{1}{3} \quad , \quad P(vez = prim|V_j) = \frac{2}{3} \quad , \quad P(PPA = bueno|V_j = aprob) = \frac{1}{3}$$

$$V_{NB}|aprob = \frac{9}{14}.\frac{1}{3}.\frac{1}{3}.\frac{2}{3}.\frac{1}{3} = 0.015873$$

$$V_{NB} = argmax_{desap}\, P(desap).P(C.Mat = tres|desap).P(curso = c3|desap).$$
$$P(vez = prim|desap).P(PPA = bueno|desap)$$

Procederemos a calcular cada una de las probabilidades condicionales según los datos del conjunto de entrenamiento para $V_j = desap$:

$$P\big(C.Mat = tres\big|V_j = desp\big) = \frac{P(tres \cap desap)}{P(desap)} = \frac{\frac{4}{14}.\frac{1}{4}}{\frac{5}{14}} = \frac{1}{5}$$

$$P\big(curso = c3\big|V_j\big) = \frac{2}{5} \quad , \quad P\big(vez = prim\big|V_j\big) = \frac{1}{5} \quad , \quad P\big(PPA = bueno\big|V_j\big) = \frac{3}{5}$$

$$V_{NB}|desap = \frac{5}{14}.\frac{1}{5}.\frac{2}{5}.\frac{1}{5}.\frac{3}{5} = 0.00342857$$

Esta clasificación predice la instancia en consulta con la clase aprobado, debido a que le otorga la mayor probabilidad. Para determinar dicha probabilidad normalizada, procedemos a calcular:

$$\frac{0.0158733}{0.0158733 + 0.00342857} = \frac{0.0158733}{0.0193018} = 0.8216546$$

Hasta ahora se ha calculado la probabilidad de manera clásica como el numero de eventos observados entre el número total de eventos (usando el factor $\frac{N_c}{N}$). Debido a que esta forma de calcular la probabilidad presenta problemas cuando no existe eventos, dado que es un solo factor de la probabilidad total y haría cero a toda la expresión, [Mitc-97] propone usar la estimación de Laplace.

$$estimador = \frac{N_c + mp}{N + m} \quad \text{……. (4.20)}$$

Donde N_c y N, son las mismas variables usadas anteriormente, p es la estimación de la probabilidad que deseamos obtener, se obtiene al dividir 1 entre el número posible de valores del atributo en cuestión (se asume que todos los valores de los atributos son igualmente probables), m es una constante llamada (equivalent simple size), que determina qué tanto peso darle a la variable p, y es igual a la cantidad de valores de pueda tomar el atributo analizado:

$$P(C.Mat = tres\ |V_j) = \frac{N_c + 1}{N + m} = \frac{4}{9+3} = \frac{1}{3}$$

$$P(curso = c3|aprob) = \frac{1}{3}\ ,$$

$$P(vez = prim|aprob) = \frac{7}{11}\quad ,\quad P(PPA = bueno|aprob) = \frac{4}{11}$$

$$V_{NB}|\ aprob\ =\ \frac{9}{14}.\frac{1}{3}.\frac{1}{3}.\frac{7}{11}.\frac{4}{11} = 0.016528926$$

de igual manera el suspenso(desaprobado)

$$V_{NB}|\ desap\ =\ \frac{5}{14}.\frac{1}{4}.\frac{3}{8}.\frac{2}{7}.\frac{4}{7} = 0.005466472$$

Normalizando obtenemos el nuevo valor de la probabilidad de que la predicción sea aprobada para nuestra instancia analizada.

$$\frac{0.016528926}{0.016528926 + 0.005466472} =\ 75.14\%$$

4.4.4. Clasificación por la técnica de KNN

El Algoritmo KNN, llamado también K-Nearest Neighbor [Ahad-91], basa su funcionamiento en la búsqueda de las K instancias más cercanas. Generalmente, se usa como medida de vecindad la distancia euclídea en el caso de KNN, uno de los problemas de usar esta distancia radica en que, si existen datos con un rango amplio, los de menor valor aportan menos información; por ello, es necesaria la normalización.

Si usamos únicamente la distancia a un vecino, el resultado suele ser muy sensible al ruido. Es por ello que se suele usar K vecinos más cercanos. Si la variable de salida es numérica, entonces el resultado de la predicción nos dará el valor medio, pero si es del tipo categórica (SUSP, APROB), el resultado es el valor más frecuente.

A diferencia de otros algoritmos de aprendizaje que consumen más tiempo en la etapa de construcción del modelo, KNN prácticamente dedica mayor tiempo en la etapa de clasificación de nuevas instancias; por ello, se le suele llamar “algoritmo perezoso”.

ENTRADA: Conjunto de entrenamiento T de la forma $\{ (x_1, C_1), \ldots . (x_N, C_N)\}$,
Instancia a clasificar x_q
Numero de vecinos K

PROCESO:

1. Calcular $d(x_i, x_q)$ $\forall\, x_i \in X$, siendo d la distancia euclidiana.
2. Seleccionar los $\{x_1, x_2 \ldots . x_k\}$ vecinos más cercanos de x_q denotado por $D_{x_q}^K$
3. Asignar a x_q la clase más frecuente C_j de $D_{x_q}^K$
4. Terminar proceso

SALIDA: Clase más frecuente en $D_{x_a}^K$ asignada a x_q.

Figura 4.8. Algoritmo de vecinos próximos (KNN-IBK)

x_i describe a una instancia cualquiera de atributos $< a_1(x), a_2(x), \ldots . a_n(x) >$
La distancia euclidiana está definida como:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \ldots\ldots\ldots (4.21)$$

Existe una variación en la cual se considera el peso asociado a las distancias; entonces solo se modifica la siguiente expresión:

$$f'(x_q) \quad \leftarrow \quad argmax_{v \in V} \sum_{i=1}^{k} w_i . \delta(v, f(x_i)); \text{ donde } \quad w_i = \frac{1}{d(x_q, x_j)^2} \ldots\ldots\ldots (4.22)$$

Usando simulación, Enas y choi, en [Enas-86], hicieron un estudio en el cual se determina el k óptimo cuando solo existen dos clases presentes y se determinó que si los tamaños muestrales de las clases son comparables, entonces $k = n^{\frac{3}{8}}$. Mientras que si existe un desbalance entre la distribución de clases, se puede usar $k = n^{\frac{1}{4}}$.

Se suele usar K impar para evitar posibles empates en la clasificación de nuevas instancias. Para entender mejor cómo se trabaja esta técnica predeciremos la clase correspondiente a un estudiante con los siguientes atributos:

$$\mathrm{a_i} = \text{curso matric.} = \text{uno; curso} = \text{c3, vez} = \text{primera; PPA} = \text{bueno})$$

Cabe mencionar que el ejemplo a visualizar usa los datos de la tabla 4.1. A continuación, se muestra una tabla de distancias generadas cuando el algoritmo compara la instancia a clasificar con cada una de las instancias del conjunto de entrenamiento.

Curso Matriculado=uno	Curso=C3	Vez=primera	PPA=bueno	distancia
0	1	1	1	1.73205081
0	1	1	0	1.41421356
0	1	1	1	1.73205081
1	0	1	1	1.73205081
1	0	0	1	1.41421356
1	0	0	0	1
1	1	0	0	1.41421356
1	1	1	1	2
1	1	0	1	1.73205081
1	0	0	1	1.41421356
1	1	0	0	1.41421356
1	1	1	0	1.73205081
0	1	0	1	1.41421356
1	0	1	0	1.41421356

Tabla 4.7. Tabla de comparación entre cada instancia del conjunto de entrenamiento

En la tabla 4.7, se puede observar que los valores que corresponden a las celdas son binarios. Esto obedece a que los datos de entrenamiento son atributos nominales y al comparar dichos valores con los de la instancia de prueba ellos asignan el valor de cero cuando los valores son coincidentes y el valor de uno cuando difieren.

Una vez hecha la comparación, se procede calcular el valor que corresponde a la distancia para dichos atributos nominales usando la expresión de la distancia euclideana. Una vez obtenidos estos valores, se elije el menor de ellos asegurando que la distancia correspondiente a la menor distancia será el vecino más próximo. Debido a que la instancia más cercana a la que está en consulta tiene la clase original suspenso, entonces la clase en consulta toma el mismo valor.

4.4.5. Decision Stumps

Un Stump es un árbol de decisión de un solo nivel, con un nodo raíz y sus hijos. Este árbol tiene como nodo raíz aquel atributo que maximiza la ganancia de la información entre todo el conjunto de instancias de entrenamiento [Holt-93] [Ye-07].

Pseudocódigo

ENTRADA: Conjunto de entrenamiento T
Conjunto de atributos $\{a\} = < a_1, a_2 \ldots . a_n >$
Clase C = $\{C_1, C_2 \ldots.\}$

PROCESO:

1. Calcular $\forall\ a_i \in \{a\}$ el $gain\ ratio(T, a_i)$
2. Sea el atributo $a_q \leftarrow Argmax_{(a_i \in \{a\})}\ gain\ ratio(T, a_i)$
   2.1. Sean $d_1, d_2 \ldots . d_m$ los valores del atributo a_q
   2.2. Crear un nodo raíz t_0 con etiqueta a_q y una rama d_j , siendo este el valor más frecuente de dicho atributo (Para que produzca menor error en la clasificación.)
   2.3. Crear una rama que contenga los valores del atributo $\{d_1 .. d_m\} - d_j$
   2.4. Las instancias que no puedan ser clasificadas en los pasos previos, se les asignara una clase por defecto (aquella con más frecuencia en el conjunto de entrenamiento T).
3. Terminar proceso

SALIDA: Un árbol de decisión de un solo nivel.

Figura 4.9. Algoritmo del método de decisión Stump [Witt-05]

Generamos un pequeño árbol Stump con las instancias que han sido usadas en anteriores ejemplos.

Número de cursos matriculados:

	SUSP	APROB	
Uno	2	2	4
Dos	2	4	6
tres	1	3	4

Tabla 4.8. Distribución del atributo número de cursos por cada clase

Cursos matriculados:

	SUSP	APROB	
C1	3	2	5
C2	0	4	4
C3	2	3	5

Tabla 4.9. Distribución del atributo cursos matriculados por cada clase

Vez de matricula:

	SUSP	APROB	
Primera	1	6	7
Segunda	4	3	7

Tabla 4.10. Distribución del atributo vez de matrícula por cada una de las clases

PPA

	SUSP	APROB	
malo	2	6	8
bueno	3	3	6

Tabla 4.11. Distribución del atributo PPA por cada una de las clases

Calculamos el $\mathrm{GainRatio}$ y comparamos los GainRatios de cada uno de los atributos:

$\text{gain ratio}(\text{Curso Matri}) = 0.0188$

$\text{gain ratio}(\text{Curso}) = 0.1565.$

$\text{gain ratio}(\text{Vez de matricula}) = 0.1518$

$\text{gain ratio}(\text{PPA}) = 0.0488$

Entonces se elige el atributo curso como raíz del árbol. A continuación, elegimos el valor del atributo que produzca menor cantidad de errores.
De la tabla, se observa que elige C2=APROB por contener solo aciertos.

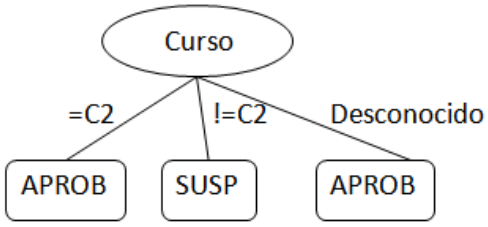

Figura 4.10. Estructura de grafo final generado por el algoritmo de decisión Stump

Finalmente, calculamos la distribución de probabilidades para predecir nuevos casos

Curso = C2	SUSP	APROB
	0.0	1.0

Curso != C2	SUSP	APROB
	0.5	0.5

Curso = C2	SUSP	APROB
	0.357	0.64

Decisión Stumps suele ser usado como clasificador base de Boosting.

4.4.6. Conjunto de clasificadores

Los conjuntos de clasificadores son sistemas que clasifican nuevos ejemplos combinando las decisiones individuales de los clasificadores que los componen. Estos conjuntos se construyen en dos pasos: en el primer paso, se genera un conjunto de clasificadores con algún algoritmo en particular, y en el segundo paso se combinan las hipótesis generadas.

En esta investigación, se usan las técnicas basadas en remuestreo de los datos de entrenamiento, técnicas que introducen perturbaciones en los datos de entrada (por ejemplo, repetición de ejemplos, eliminación de ejemplos, asignación artificial de pesos etc.) con el objetivo de obtener cada uno de los clasificadores individuales.

Para que este tipo de técnicas sean útiles es necesario que los clasificadores generados sean diversos. Por tanto, para construir un conjunto de clasificadores, hay que elegir el algoritmo base y diseñar una metodología que sea capaz de construir clasificadores que cometan errores distintos en los datos de entrenamiento. Los árboles de decisión poseen características que los convierten en buenos algoritmos para estas técnicas, debido a que pequeñas variantes en los datos de entrenamiento pueden lograr que los modelos generados sean muy diferentes.

Los métodos de generación de conjuntos, estudiados en este trabajo de investigación, usan técnicas basadas en remuestreo de datos de entrenamiento; dos de los algoritmos más usados son Bagging [Brei-96] y Boosting [Freu-95], que se explican brevemente a continuación.

4.4.6.1. Clasificación por la técnica de Bagging

Bagging (bootstrap aggregating) fue propuesto por [Brei-96] como un método multiclasificador que combina las predicciones de un conjunto de hipótesis. En este caso, cada clasificador del conjunto se obtiene utilizando una muestra aleatoria con repetición del mismo número de ejemplos que el conjunto de datos de entrenamiento (muestra bootstrap). Cabe mencionar que, en un muestreo con reemplazo, muchas instancias del conjunto original pueden repetirse, mientras que otras podrían quedarse al margen [Opit-99].

La combinación de clasificadores se realiza mediante la votación ponderada de la predicción de cada hipótesis, luego se calcula el voto promedio de los clasificadores en conjunto y se elige el más probable.

Cada muestra utilizada para construir cada uno de los clasificadores contiene el 63.2% de los datos originales; el resto de ejemplos, con muy alta probabilidad, son repetidos; por lo tanto, en este tipo de técnicas, cada clasificador se genera con un conjunto reducido de datos de entrenamiento. Esto significa que los clasificadores individuales representan peores hipótesis que los clasificadores construidos con todos

los datos. Esta peor capacidad de generalización se compensa mediante la combinación de los clasificadores y se traduce en una mejor capacidad de generalización por parte de la técnica de Bagging en comparación a la de un solo clasificador construido con todos los datos de entrenamiento.

ENTRADA: Conjunto de entrenamiento T
Numero de conjuntos I (número de muestras bootstrap)
Algoritmo de aprendizaje A Clase C = {C_1, C_2 ….}

PROCESO: Para i=1 hasta I hacer

1. Generar la muestra bootstrap T_i utilizando el conjunto original T
   $T_i = boostrap(T)$ //
2. Con la muestra T_i generada, creamos una hipótesis H_i, utilizando el algoritmo de aprendizaje A.
   $H_i = ConstruyeClasificador(A, T_i)$
3. Incrementar la iteración $i = i + 1$, volver al paso 1.
4. Sea x_q una nueva instancia a clasificar,
   $C^*(x) = Argmax_{C_j \in C} \sum_{i=1}^{I} \delta(H_i(x_q), c_j)$ // La clase predicha con más frecuencia
   Donde $\delta(a,b) = 1 \; Si \; a = b$ y $\delta(a,b) = 0$ en otros casos

SALIDA: Clase más frecuente $C^*(x_q)$ para la instancia x_q.

Figura 4.11. Pseudocódigo para el algoritmo de clasificación de Bagging

Bagging recibe como parámetros de entrada el número de iteraciones o modelos a generar, esto tiene un efecto en cuanto al sesgo y la varianza. La reducción del error con respecto al algoritmo utilizado se debe a la reducción en varianza [Baue-99]. Bagging reduce más el error cuando los clasificadores individuales tienen errores de sesgo pequeño y a la vez errores de varianza grandes [Brei-98].

Estas medidas se relacionan con la capacidad de ajuste y generalización de un modelo. Cuando se logra un gran ajuste, la diferencia entre los datos reales y la estimación del modelo es pequeña; en este caso, el sesgo también es pequeño, pero estos buenos resultados de ajuste van de la mano con el aumento en la complejidad del modelo. Cuando se aumenta la complejidad del modelo, este se vuelve sensible a pequeñas variaciones en los datos de entrada, fluctuando en función de estos. Es así como la varianza aumenta [Witt-05].

Al combinar múltiples clasificadores, se reduce el error esperado al reducir la varianza.

El método Bagging presenta dos fases para su funcionamiento: generación de modelos y clasificación de una nueva instancia.

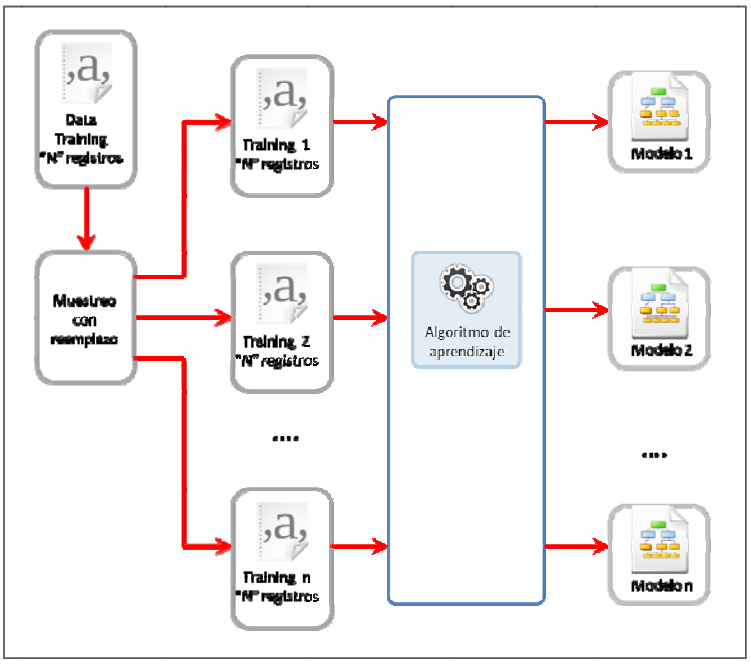

Figura 4.12. Modelo Bagging para la generación de hipótesis

Se generan T conjuntos (muestreo Bootstrap). Estos sirven como entrada para el algoritmo de aprendizaje, el cual genera tantos modelos como subconjuntos se crearon.

A continuación, una nueva instancia es clasificada en todos los modelos generados y mediante la votación se elige el resultado más probable, el cual es la salida final de la etapa de clasificación.

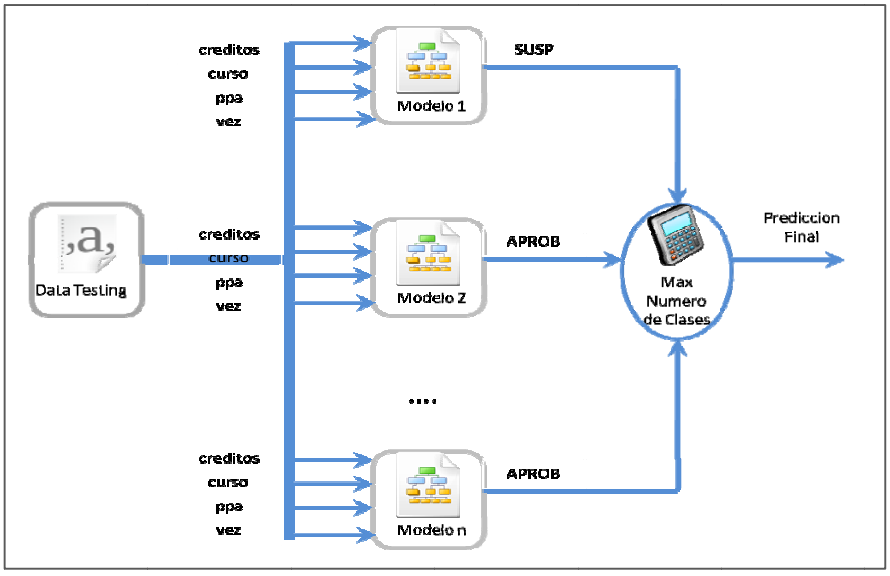

Figura 4.13. Modelo Bagging para clasificación de instancias

Algunas consideraciones adicionales del método Bagging son:

- Muestra buenos resultados cuando los datos contienen ruido [Hand-06].
- Se usa con clasificadores inestables, es decir, aquellos que al cambiar una pequeña porción de datos cambian totalmente el modelo. Un ejemplo de clasificador inestable son los árboles de decisión [Alpa-10].
- Un número de iteraciones mayor reduce el error en la predicción. Existen trabajos relacionados en esta área, los cuales calculan un número óptimo de iteraciones [Opit-99].

4.4.6.2. Clasificación por la técnica de Boosting

Boosting fue introducido por Schapire en [Scha-04] y posteriormente fue mejorado por Freund [Freu-95]. Él la concibió como una técnica que construye clasificadores mediante la asignación de pesos a los ejemplos de forma adaptativa. En cada iteración, se construye un nuevo clasificador que intenta compensar los errores cometidos previamente, aprovechando el comportamiento de los previamente construidos. Para lograr que en cada iteración cada nuevo clasificador mejore los resultados en regiones donde fallan los anteriores, se utiliza un conjunto de datos ponderado cuyos pesos son actualizados tras cada iteración.

Breiman, en [Brei-96], diseñó este tipo de algoritmos adaptativos con el nombre de “arcing” (adaptively resample and combine). En Boosting, este proceso adaptativo se consigue asignando pesos a los ejemplos de entrenamiento y modificando dichos pesos de acuerdo con los resultados del último clasificador generado. La modificación de pesos se hace de forma que los ejemplos mal clasificados aumenten en importancia para construir el siguiente clasificador. Boosting es el primer algoritmo de arcing desarrollado y el más difundido.

El primer algoritmo de este tipo fue desarrollado por Freund en [Freu-95] y se denominó “AdaBoost” (AdaptativeBoosting). AdaBoost genera un conjunto de modelos o hipótesis y sus respectivos votos. Los modelos son creados de una manera secuencial; en cada iteración, el algoritmo modifica los pesos asociados a cada instancia con la finalidad de penalizar aquellas instancias que no fueron clasificados correctamente.
El objetivo es minimizar el error esperado en diferentes distribuciones. Veamos a continuación el funcionamiento de AdaBoost. El pseudocódigo de este algoritmo se encuentra en la figura 4.14.

ENTRADA: Conjunto de entrenamiento T de la forma $\{ (x_1, C_1), \ldots . (x_N, C_N)\}$
I iteraciones
Algoritmo de aprendizaje A

PROCESO:

1. Asignamos pesos a cada instancia del conjunto T, $D_1(i) = \frac{1}{N} \forall i$

Para t=1 hasta I hacer

2. Generamos el modelo H_t , $H_t = ConstruyeClasificador(T, D_t, A)$
3. Calculamos el error de H_t : $e_t = \sum_{i=1}^{N} (1 - \delta(H_t(x_i), C_i)) * D_t(i)$ (ejemplos mal clasificados)
   (Donde $\delta(a, b) = 1 \; Si \; a = b$ y $\delta(a, b) = 0$ en otros casos)
   Si $e_t \geq 0.5$ // descartar clasificador
   Si $e_t = 0$ // salir del bucle e ir al paso 8
4. $\beta_t = \frac{e_t}{1 - e_t}$

Para j=1 hasta N hacer

5. Si $H_t(x_j) = C_j$ entonces $D_{t+1}(j) = D_t(j).\beta_t$
   Si no $D_{t+1}(j) = D_t(j)$
6. Normalizamos los pesos $D_{t+1}(j) = D_{t+1}(j) * \frac{1}{Z_t}$(Siendo Z_t la suma de todos los pesos en la iteración t).

7. $t = t + 1$, volver al paso 2.
8. Sea x_q una nueva instancia a clasificar,

$$C^*(x_q) = \text{Arg}\, max_{c_j \in C} \sum_{i=1}^{I} \delta(H_i(x_q), c_j) * \log \frac{1}{\beta_i}$$

SALIDA: Clase con votación mayoritaria $C^*(x_q)$ para la instancia x_q.

Figura 4.14. Pseudocódigo para el algoritmo de clasificación de Boosting

Dado un conjunto de datos de entrenamiento $T = \{ (x_1, C_1), \ldots . (x_N, C_N)\}$ y un número I de Iteraciones (cada una construye un clasificador), se asigna un peso D a cada instancia del conjunto de entrenamiento, inicializando los pesos con $D_1(i) = \frac{1}{N}$ (línea 1). Es decir, al principio, todos los ejemplos tienen igual importancia para construir el primer clasificador individual.

A continuación, se realiza un bucle de I iteraciones donde se construye, usando el clasificador A, un modelo individual H_t , aplicando todas las instancias de entrenamiento ponderadas con pesos D_t (línea 2). A continuación, se calcula el error ϵ_t para el modelo H_t con respecto a las instancias de entrenamiento T, como la suma de los pesos de los ejemplos mal clasificados (línea 3). Si $\epsilon_t \geq 0.5$ entonces se descarta este clasificador. En el caso que el error $\epsilon_t = 0$ entonces el proceso se da por concluido.

Si ninguno de estos casos se cumple, se calculan los pesosD_{t+1} para construir el modelo H_{t+1} de forma que H_{t+1}) con los pesos D_{t+1} tenga un error menor que 0.5. Esto se consigue reduciendo los pesos de los ejemplos bien clasificados por un factor $\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$ y normalizando (línea 6).

Finalmente, en la línea 8, la clasificación del conjunto se obtiene a través del voto ponderado de todos los clasificadores H_t mediante $\log\frac{\epsilon_t}{1-\epsilon_t} = \log\frac{1}{\beta_t}$. Esto significa que clasificadores con menor error tienen más peso en el proceso de votación.

A continuación, en la figura 4.15, se puede ver la generación del modelo para la técnica de Boosting.

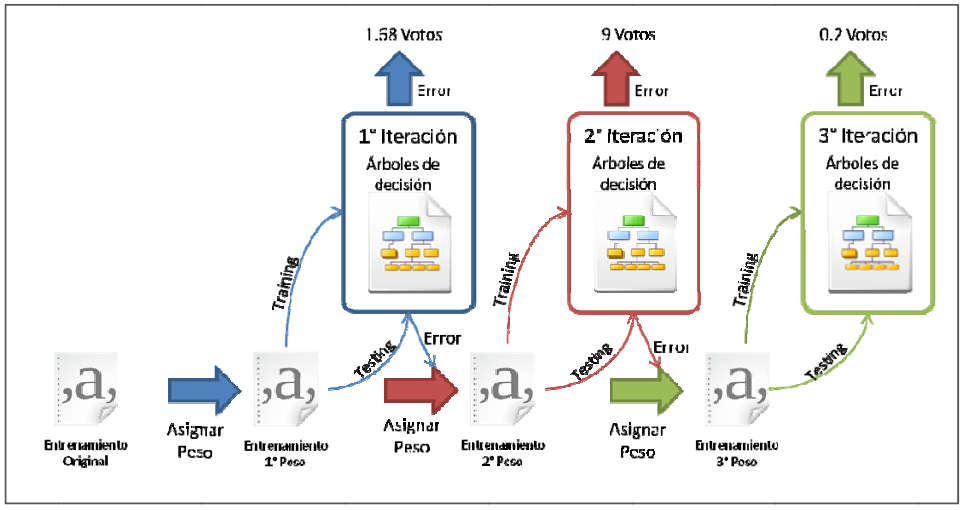

Figura 4.15. Modelo Boosting para la obtención del modelo a partir de un conjunto de entrenamiento

Para la clasificación de una instancia, se sigue el siguiente procedimiento:

1. Cada modelo clasifica una nueva instancia otorgándole a la clase predicha sus votos.
2. Se suman los votos por cada clase.
3. Se escoge la clase con más votos.

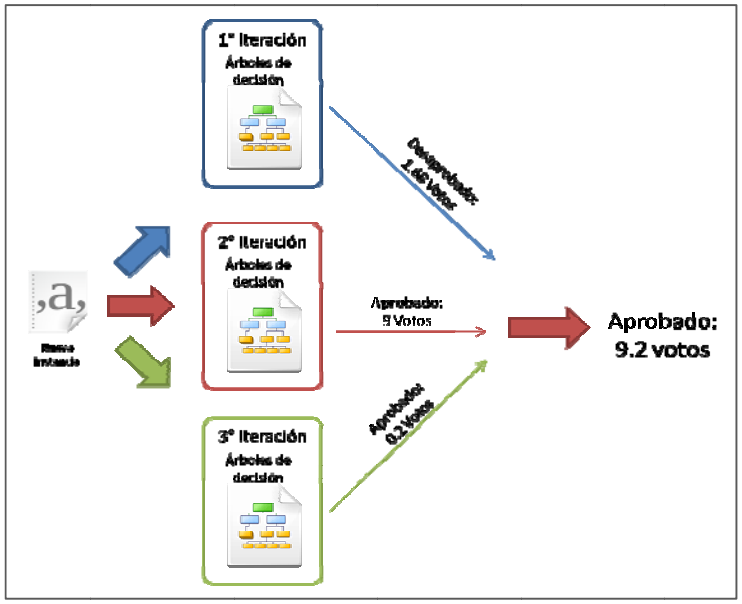

Figura 4.16. Modelo Bagging para clasificación de una nueva instancia

4.4.7. Modelos propuestos

Como se sabe, un conjunto de clasificadores clasifica nuevos ejemplos por decisión conjunta de sus componentes. Las decisiones de los clasificadores individuales se combinan, mediante voto, para obtener una clasificación final. Normalmente, de esta combinación resulta un conjunto de clasificadores que tiene más precisión que cada uno de los clasificadores de los que está compuesto. En estos casos, se evita combinar clasificadores similares entre sí, debido a que la precisión del conjunto será aproximadamente igual a la de sus componentes.

Por tanto, para mejorar el resultado de la clasificación, lo importante es generar clasificadores diversos cuyos errores no estén correlacionados, de forma que, al combinar, los errores de éstos tiendan a compensarse.

En este trabajo de investigación, se propone un nuevo método de generación de conjuntos de clasificadores que llamamos BAG. La propuesta, que tiene sus orígenes en el algoritmo de Bagging, plantea dos alternativas que están basadas en diferentes métodos de formación de conjuntos. La primera está basada en un muestreo estratificado BAG-E y la segunda en un muestreo probabilístico BAG-P.

El método BAG-E está basado en los métodos de conjuntos de clasificadores que tienen la tarea de clasificar nuevos ejemplos, combinando las decisiones individuales de los clasificadores provenientes de conjuntos que resultan de estratificar la base de datos original bajo alguna condición. Los subconjuntos estratificados provienen del dominio de aplicación y son determinados subdividiendo la base de datos total tomado en cuenta alguna condición establecida que sea coherente con el dominio de aplicación. En este caso, dicha condición tiene que ver con la periodicidad con que se generan los datos, como se explicará más adelante. Por esto, se ha considerado que esta condición debe corresponder a un período equivalente a un año académico. Cada uno de estos estratos (subconjuntos) será entrenado con el algoritmo base. Luego de ello, se le aplicará la votación, de acuerdo al pseudocódigo de la figura 4.17.

ENTRADA: Conjunto de entrenamiento T
Número de conjuntos I (muestras estratificadas)
Algoritmo de aprendizaje A
Clase C = {C_1, C_2 ….}

PROCESO: Para i=1 hasta I hacer

1. Generar la muestra T_i. (En nuesto caso $T_i = 1$ año acaémico)
   T_i =*CortePorPeriodo*(T) //
2. Con la muestra T_i generada, creamos una hipótesis H_i, utilizando el algoritmo de aprendizaje A.
   $H_i = ConstruyeClasificador(A, T_i)$
3. Incrementar la iteración $i = i + 1$, volver al paso 1.
4. Sea x_q una nueva instancia a clasificar,
   $C^*(x) = Argmax_{C_j \in C} \sum_{i=1}^{I} \delta(H_i(x_q), c_j)$ // La clase predicha con más frecuencia

SALIDA: Clase más frecuente $C^*(x_q)$ para la instancia x_q.
Donde $\delta(a, b) = 1\ Si\ a = b$ y $\delta(a, b) = 0$ en otros casos

Figura 4.17. Pseudocódigo para el algoritmo de clasificación de Bag-E

Como se puede observar, la diferencia entre el pseudocódigo del algoritmo de Bagging correspondiente a la figura 4.11 y el pseudocódigo de la figura 4.17 está en la generación de los conjuntos de entrenamiento para la construcción de los clasificadores. Mientras que en el Bagging clásico, los conjuntos son formados por muestreo aleatorio con reemplazo, en el Bag-E dicho conjunto corresponden a la función *CortePorPeriodo (figura 4.17)*, función que está determinada por las características y ventajas propias de nuestro dominio de aplicación.

En el caso de muestreo estratificado, el conjunto base, que es el que se ha utilizado como conjunto de entrenamiento en todos los casos, se ha divido en tantos subconjuntos independientes como años se quieran considerar en el estudio (cada subconjunto corresponde a un año de estudio). Se debe tener en cuenta que este método puede ser aplicado a cualquier dominio. Sin embargo, la justificación para que sea aplicado al presente es la siguiente:

- En la Universidad, cada año académico está compuesto de dos cuatrimestres regulares. Los estudiantes, al matricularse en un determinado período (cuatrimestre), obtienen calificaciones según el nivel de estudios al que pertenecen y las asignaturas en que se hayan matriculado. Esto significa que dentro del conjunto de datos de un año académico se considera, por lo menos en forma teórica, las matriculas, el rendimiento y, en general, todas las características de un grupo mayoritario de estudiantes que son aquellos que habiendo hecho su ingreso a la universidad, estudiarán su carrera en cinco años consecutivos como mínimo. Esto último implica que los datos (registros) presentes en un año (estrato) provienen de un mismo

proceso que involucra el hecho de que un estudiante se matricule, que curse sus asignaturas, que rinda sus evaluaciones y que obtenga resultados.

- Así como todos los estudiantes son caracterizados en cada estrato considerado en esta propuesta, también se puede decir que, en la base de datos correspondiente a cada año, se toman en cuenta todas las asignaturas con sus respectivas evaluaciones. Esto se debe a que, en cada semestre, la institución oferta todas sus asignaturas y que, de semestre a semestre, tanto la oferta como la demanda de estas es relativamente constante.
- Estos dos fundamentos tienen sus orígenes en que el sistema universitario es casi un sistema cíclico y que cada año son cursadas las mismas asignaturas, pero por un grupo distinto de estudiantes y las modificaciones curriculares son tomadas en cuenta por la metodología propuesta en este trabajo (capítulo 5). Además, es importante destacar que se trata de un sistema colaborativo y que lo que se pretende es descubrir los patrones ocultos que existen en la base de datos y que cualquier modificación que se pueda hacer en la aplicación del algoritmo deberá tener en cuenta para el análisis todos los datos o, por lo menos, una gran parte de ellos.

Con la fundamentación anterior, se puede tener una idea intuitiva de que los datos utilizados en esta propuesta tienen una distribución de probabilidad cuya variación es suave debido a la aleatoriedad que presenta la selectividad dentro de cada año y a la poca variabilidad de los resultados obtenidos por el grupo de alumnos entre año y año. Además, es muy claro reconocer que, debido a que los modelos generados son estáticos, consideramos que la distribución de probabilidad del problema en su conjunto es estacionaria con respecto al tiempo.

En la figura 4.18, se puede observar que el conjunto de entrenamiento es particionado en subconjuntos (estratos), en donde cada instancia de los conjuntos pertenece a un año distinto. Luego de entrenar cada conjunto, se aplica el método de votación que permite determinar el rendimiento del conjunto de prueba basado en la votación que se haya obtenido de cada unos de los n modelos generados a partir de los n estratos iniciales y en cada una de las instancias. En la sección correspondiente al diseño de los experimentos, se considerará una validación basada en la aleatoriedad de la elección del conjunto de prueba la cantidad de veces que se quiera.

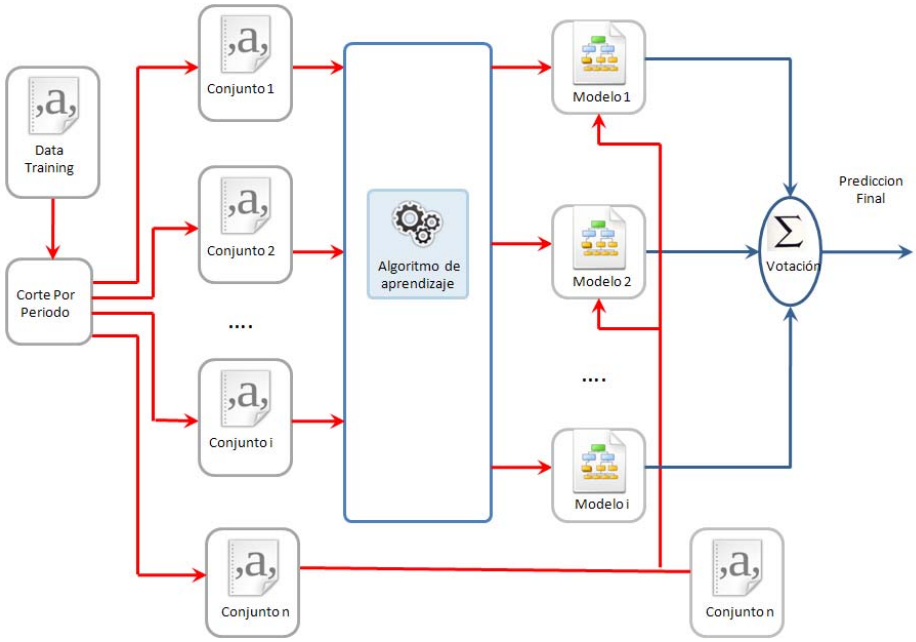

Figura 4.18. Propuesta de método de votación con conjuntos estratificados (Bag-E)

El segundo método de clasificación mediante conjunto, que se aplica en el presente trabajo de investigación, es el que llamamos **BAG-P.**

BAG-P está basado en el método de Bagging y consiste en introducir una perturbación al conjunto de entrenamiento basada en los cortes aplicados en el experimento referido a la independencia de los datos con respecto al tiempo, explicados en el punto 6.2.2.

Debido a las características del dominio de aplicación, los subconjuntos se van formando por eliminación. Esto quiere decir que el primer subconjunto será el conjunto inicial de datos, el segundo subconjunto estará conformado por los mismos registros que el anterior menos los que corresponden al conjunto de datos más antiguos de la base de datos con alguna condición establecida, y así sucesivamente. Cabe mencionar que, con esta propuesta, los conjuntos sucesivos tendrán cada vez menos instancias, por lo que este proceso deberá ser detenido en algún momento, como se puede observar en el pseudocódigo de la figura 4.19.

La condición que se establece para este caso particular es que los datos eliminados en los conjuntos sucesivos pertenecen a semestres académicos sucesivos.

ENTRADA: Conjunto de entrenamiento T, conjunto de prueba Te
Número de conjuntos I (número de muestras probabilísticas)
Algoritmo de aprendizaje A
Clase C = {C_1, C_2 ….}
Muestra a extraer $P = \{P_1, P_2 \ldots\ldots P_n\}$ (conjunto de periodos académico, que puede ser año o semestre)

PROCESO:

1. $T_e = ExtraerTesting(T)$.(Conjunto de prueba es extraído del conjunto original T)
2. T_1 = T -{T_e}

Para $i = 1$ hasta I hacer

3. Generar una muestra T_{i+1} del conjunto T_1 .
   T_{i+1} =T_i-{P_i} /
4. Con la muestra T_i generada, creamos una hipótesis H_i, utilizando el algoritmo de aprendizaje A.
   $H_i = ConstruyeClasificador(A, T_i)$
5. Incrementar la iteración $i = i + 1$, volver al paso 3.

6. Sea x_q una nueva instancia del conjunto de prueba a clasificar,
   $C^*(x) = Argmax_{C_j \in C} \sum_{i=1}^{I} \delta(H_i(x_q), c_j)$ // La clase predicha con más frecuencia
   Donde $\delta(a, b) = 1\ Si\ a = b$ y $\delta(a, b) = 0$ en otros casos

SALIDA: Clase más frecuente $C^*(x_q)$ para la instancia x_q.
Nota: El proceso de clasificación se repite con todas las instancias del conjunto de prueba
Donde $\delta(a, b) = 1\ Si\ a = b$ y $\delta(a, b) = 0$ en otros casos

Figura 4.19. Pseudocódigo para el algoritmo de clasificación de Bag-P

Como se puede observar, la diferencia entre el pseudocódigo de la figura 4.11 y el pseudocódigo de la figura 4.19 reside en la generación de los conjuntos de entrenamiento para la construcción de los clasificadores. Mientras que en el Bagging clásico los conjuntos son formados por muestreo aleatorio con reemplazo, en el Bag-P dicho conjunto corresponde a la generación de subconjuntos, llamados “corte” en esta investigación, tal como se muestra en las líneas 1), 2) y 3) del pseudocódigo.

En la figura 4.20, se pueden observar los conjuntos después de aplicar el muestreo probabilístico (dirigido). Una vez formados dichos conjuntos, se procede a entrenar cada uno de ellos. Obtenido un modelo por cada conjunto, en seguida se elige una cantidad determinada de conjuntos de prueba, que depende del máximo número de particiones del último subconjunto que cumplen las condiciones establecidas. Después de este proceso, se aplica la votación.

Cabe mencionar que, antes de proceder a aplicar el algoritmo de clasificación a los subconjuntos de entrenamiento, se extrae de cada uno de ellos el subconjunto de prueba seleccionado.

Los resultados se pueden ver en la sección correspondiente del capítulo de experimentación.

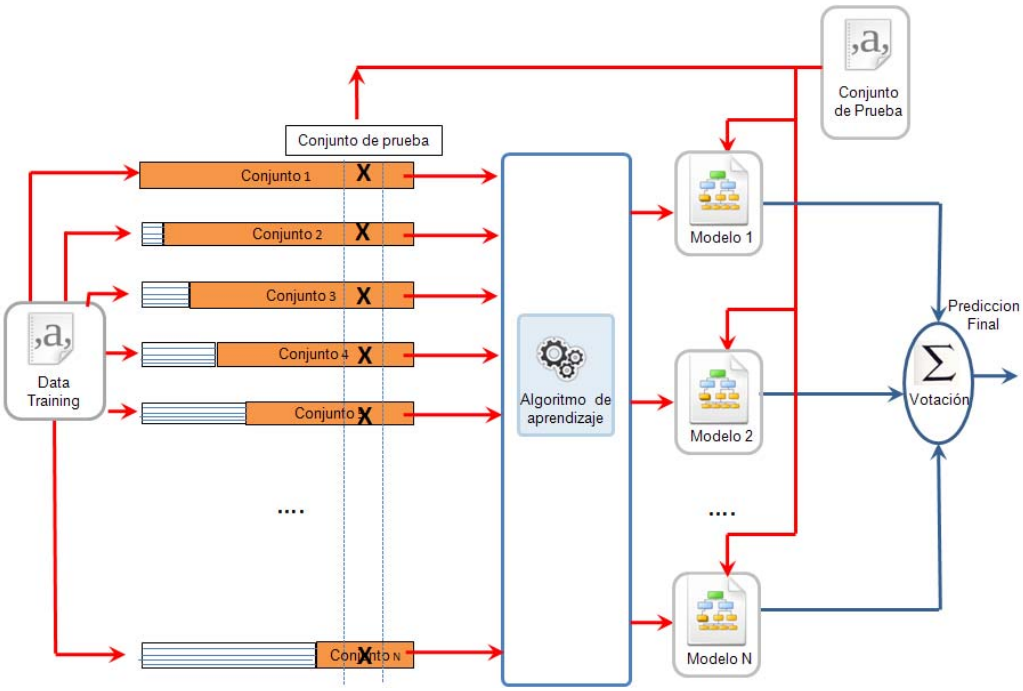

Figura 4.20. Método de clasificación basado en la división de datos por cortes (Bag-P)

Es bueno aclarar que los fundamentos para aplicar este tipo de generación de conjuntos están basados principalmente en las características de nuestro dominio de aplicación. Los experimentos realizados en el capítulo 6 ilustran que los métodos propuestos, después de la experimentación, producen resultados equivalentes o mejores que las técnicas tradicionales (Bagging y Boosting) en los conjuntos de datos explorados. Además de esto, presentan un importante ahorro computacional respecto a conjuntos creados con algoritmos clásicos.

4.5. Métodos de poda para árboles de decisión

Los árboles de decisión descritos en el punto 4.4.1 son generalmente complejos. Aunque en algunas circunstancias se puedan definir condiciones que hagan al árbol menos complejo, tales como el número mínimo de ejemplos en las hojas de cada árbol (m), la máxima profundidad del árbol, etc. En general, estos métodos no resultan suficientes para que el árbol sea legible, entendible y con buena capacidad de generalización.

Es así como los métodos de poda se abren un camino en la necesidad de ser aplicados a los árboles entrenados con la finalidad de simplificarlos. La característica general de ellos es reemplazar un nodo hoja por un subárbol determinado, basado en la

observación de que la tasa del error esperada dentro de un subárbol pueda ser reducida al reemplazar dicho subárbol por un nodo hoja. Por ejemplo, en el árbol de la figura 4.4, tenemos cinco hojas. Entre las hojas que pertenecen a la rama Curso=C1, una de ellas clasifica dos aprobados y la otra tres suspenso. Cuando se prueba con un conjunto de test, supongamos que este tenga tres elementos, se obtiene que la rama vez=primera obtiene un aprobado y un suspenso y la rama vez=segunda no obtiene aprobados y un suspenso. Es evidente que en este caso conviene convertir estos tres nodos en un nodo hoja debido a que el número de errores son iguales en ambos casos y siempre se prefiere tener un árbol más pequeño (con menos nodos) a un árbol con muchos nodos.

Para llevar a cabo la poda en los árboles de decisión, se requiere un conjunto de datos independientes. Este conjunto recibe el nombre de "conjunto de poda", se utiliza para instancias del árbol construido y sirve para hacer los cálculos que definirán si un determinado subárbol se convierte en hoja o no. Para obtener el conjunto de pruning se utiliza el conjunto de entrenamiento y se divide con una proporción previamente establecida en dos conjuntos: uno de ellos servirá para que el árbol se genere (*tree growing set*) y el segundo conjunto para que el árbol proceda a instanciarse y se pode (*pruning set*).

A continuación, se describen los métodos revisados en este trabajo de investigación; para llevar esto a cabo, se emplea la siguiente nomenclatura:

Árbol sin podar: T_{max}
Nodo interno (no hoja): t
Nodo cualquiera (puede ser una hoja): t^*
Subárbol con raíz en t : T_t
Número de casos cubiertos por un nodo t : $n(t)$
Hojas que están debajo de un nodo t : L_t
Errores en un nodo t : $e(t)$
Nota $e(t)$ son los errores en un nodo cuando se simplifican las hojas que están debajo y se clasifica la clase mayoritaria.

4.5.1. Reduced Error Pruning (REP)

Método propuesto por J.R. Quinlan [Quin-87], conceptualmente es el más simple. Utiliza un conjunto de instancias independientes que sirve para podar a T_{max}.Su estrategia de poda es del tipo Bottom-up, realiza un recorrido desde las hojas hacia el nodo raíz de $\mathrm{T_{max}}$ siempre en busca del error mínimo de clasificación. La condición de poda de este método es $e(t) \leq e(T_t)$, siendo $e(t)$ los errores de un nodo t y $e(T_t)$ los errores de un subárbol con raíz t. Además, se cumple:

$$e(T_t) = \sum_{s \in L_t} e(s) \text{ …….(4.24)}$$

Una característica que hace a este método muy exitoso es su complejidad computacional lineal, ya que cada nodo es visitado sólo una vez para evaluar la posibilidad de poda y el trabajo simple de considerar el error numérico y no como una tasa de error. Las desventajas de este método son, en primer lugar, que requiere un conjunto de instancias independientes y, en segundo lugar, que tiene una tendencia a la sobrepoda (*overpruning)*. Esto puede ocurrir cuando el conjunto de poda es mucho menor al conjunto de entrenamiento [Espo- 97].

ENTRADA:

- Un árbol T_{max} con todas las instancias del conjunto de poda clasificadas en los nodos.

PROCESO: Hacer el recorrido del árbol y de cada nodo comenzando por las hojas.

1. Determinar el $e(t)$ para un nodo t y $e(T_t)$ para el subárbol T_t.
2. Podar si se cumple la condición $e(t) \leq e(T_t)$.
3. Pasar al siguiente nodo t.
4. Si se visitaron todo los nodos, terminar.

SALIDA: Un árbol podado.

Figura 4.21. Algoritmo para podar árboles de decisión por el método de poda REP

Para entender mejor este método, se usará el árbol construido a partir de las instancias de entrenamiento de la tabla 4.1. En la tabla 4.12 se puede observar el conjunto de poda que se usará en lo sucesivo para los métodos que lo requieran.

Cursos matriculados	Cursos	Vez de matrícula	PPA	Promedio
Uno	C2	Primera	Malo	APROB
Tres	C1	Primera	Malo	APROB
Uno	C1	Segunda	Malo	SUSP
Tres	C3	Primera	Bueno	SUSP
Dos	C3	Segunda	Bueno	APROB
Tres	C1	Primera	Malo	SUSP

Tabla 4.12. Conjunto de *pruning* utilizado para instanciar el árbol construido en la figura 4.4

Por lo tanto, el árbol instanciado será:

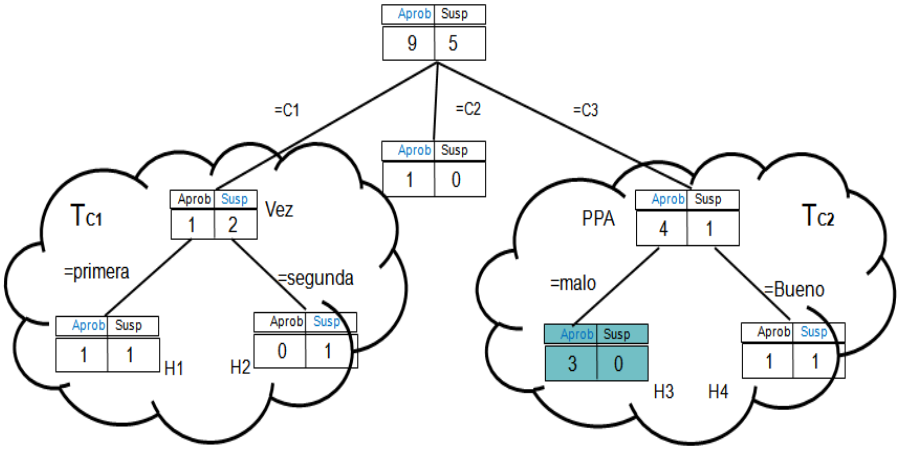

Figura. 4.22. Árbol instanciado con el conjunto de *pruning*

En la Figura 4.22, se puede observar que la hoja H3, correspondiente a la rama Curso=C3 y PPA=malo, se mantiene igual debido a que no existe instancia alguna, en el conjunto de poda, que contenga la combinación de atributos. Así también, en este árbol, los nodos agrupados y etiquetados con las letras Tc1 y Tc2 pueden convertirse en nodos hojas debido a que en ambos casos los errores son iguales (en número) y es preferible tener un árbol más pequeño. Es así que en la rama que corresponde al subárbol Tc1, el error de la hoja H1 es uno y el de la hoja H2 cero. El error del subárbol Tc1 es $e(T_t) = e(H1) + e(H2)$=1 y, además, el error que se genera al convertir Tc1 en hoja (en el supuesto caso que se pode) es $e(t) = 1$. Por lo tanto, la condición de poda $e(t) \leq e(T_t)$ se verifica, por lo que se decide podar.

4.5.2. Pessimistic Error Pruning (PEP)

Este método, propuesto por J.R. Quinlan [Quin-87], se caracteriza por usar el mismo conjunto de instancias para entrenar y podar el árbol. Define la tasa de error (TE) como medida de exactitud de los nodos y subárboles. La tasa de error aparente es optimista y no se puede usar para podar el árbol. Por ello, se introduce la corrección de la continuidad para la distribución binomial [Wild-95]; con ello, obtiene una tasa de error más realista.

La condición de poda de este método es:

$$r(t) \leq r(T_t) + SE \text{ .....(4.25)}$$

Siendo $r(t)$ la tasa de error corregida del nodo t y $r(T_t))$ la tasa de error corregida del subárbol con raíz t. Las expresiones para $r(t)$ y $r(T_t))$ son:

$$r(t) = (e(t) + 1/2)/n(t) \quad \text{.....(4.26)}$$

$$r(T_t) = \frac{1}{n(t)} \sum_{s \in L_t} (e(s) + 1/2) \quad \text{.....(4.27)}$$

Además, se sabe que $\mathrm{SE} = \sqrt{\frac{\mathrm{r(T_t)} * (1 - \mathrm{r(T_t)})}{\mathrm{n(t)}}}$ .....(4.28)

Este método, a diferencia de los demás, es del tipo Top-Down, es decir, el árbol es evaluado desde la raíz hacia las hojas y, en la práctica, su coste computacional es menor que los demás, ya que, en ciertos casos, no se necesitará visitar todos los nodos del árbol. Una de sus desventajas es el Efecto Horizonte [Bres-97] que se refiere al hecho de que un árbol puede ser podado incluso cuando contiene un subárbol que "no habría sido podado por el mismo criterio".

La introducción de la corrección de la continuidad puede producir *overpruning* o *underpruning* en el árbol podado [Espo- 97].

Pseudocódigo

ENTRADA:

- Un árbol T_{max} con todas las instancias de entrenamiento clasificadas en los nodos.

PROCESO: Hacer el recorrido del árbol y de cada nodo comenzando por la raíz.

1. Determinar la TE corregida para un nodo t y el subárbol T_t.
2. Añadir el error estándar SE a este.
3. Podar si se cumple la condición $r(t) < r(T_t) + SE$.
4. Finalizar si se han visitado todos los nodos t.

SALIDA: Un árbol podado.

Figura 4.23. Algoritmo para podar árboles de decisión por el método de poda PEP

Para entender mejor este método, se usará el árbol construido a partir de las instancias de entrenamiento de la tabla 4.1. En este caso, no se usa conjunto de poda; por lo tanto, se analizará el mismo árbol de la figura 4.4. Cabe mencionar que la poda se lleva a cabo con el conjunto de entrenamiento.

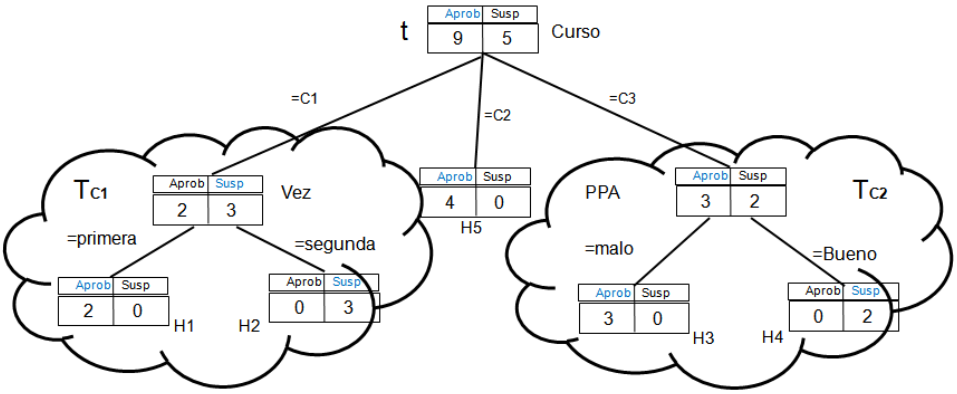

Figura 4.24. Árbol original que será sometido al método de poda PEP

Al analizar el grupo de nodos Tc1, observamos que el número de casos clasificados por el nodo t, denotado por n(t), es cinco. La tasa de error del mismo nodo, denotada por R(t), es representada por la siguiente expresión:

$r_{(t)} = \frac{e(t)}{n(t)}$; $r_{(t)} = \frac{e(t)+1/2}{n(t)}$ considerando su factor de corrección por la distribución binomial.

Como el procedimiento es desde la raíz hacia las hojas, se procede a evaluar, en primer lugar, la tasa de error correspondiente a la raíz $R(t)$.

$$r_{(t)} = \frac{e(t)}{n(t)} = \frac{5 + 1/2}{14} = 0.39$$

Luego se procede a calcular la tasa de error de todas las hojas que se encuentran ubicadas por debajo del nodo t, es decir, las hojas H1, H2, H3, H4 y H5

$$r(T_t) = \sum_{s \in Lt} \frac{(e(s) + 0.5)}{n(t)} = \frac{(0 + 0.5) + (0 + 0.5) + (0 + 0.5) + (0 + 0.5) + (0 + 0.5)}{14}$$

$$= 0.1785$$

El error estándar SE se calcula y aplica para hacer más pesimista la condición de poda. Este es:

$$\mathrm{SE} = \sqrt{\frac{\mathrm{r(T_t)} * (1 - \mathrm{r(T_t)})}{\mathrm{n(t)}}} = \sqrt{\frac{0.1785 * (0.8214)}{14}} = 0.010473$$

Por último, todo lo anterior se aplica a la condición de poda:

$$r(t) < r(T_t) + SE$$

Por lo tanto, no se poda.

Cabe mencionar que este proceso es recursivo. La siguiente iteración comprenderá el nodo raíz del subárbol Tc1 y sus respectivas hojas y así sucesivamente cubriendo todos los nodos internos que no sean hojas.

4.5.3. Minimun Error Pruning (MEP)

Niblett y Bratko [Cest-91] proponen este método Bottom-up en la búsqueda del árbol que produzca la tasa mínima de error con respecto a un conjunto independiente de instancias; es decir, se clasifican las instancias de entrenamiento en T_{max} y luego se poda en base a la m-probabilidad estimada. Por cada valor de m, se genera un árbol podado con su respectiva tasa de error. El mejor árbol será aquel m que produzca menor error de estimación.

Probabilidad de estimación en un nodo t^*

$$p_i(t^*) = \frac{n_i(t^*)+m*p_{ai}}{n(t^*)+m} ; i = \{1,2 \dots Nclases\} \text{ …….(4.29)}$$

p_{ai} = probabilidad a priori de la clase i

Error Estático en un nodo t^* :

$$STE(t^*) = \min\{1 - p_i(t^*)\} \quad i = \{1,2 \dots Nclases\} \quad \text{…….(4.30)}$$

Error Dinámico nodo t :

$$DYE(t) = \sum_{s \in L_t}(STE(s) * \frac{n(s)}{n(t)}) \text{ …….(4.31)}$$

Condición de poda:

$$STE(t) \leq DYE(t) \text{ …….(4.32)}$$

ENTRADA:

- Un árbol T_{max} con todas las instancias de entrenamiento clasificadas en los nodos.
- Conjunto M de valores del parámetro m.

PROCESO: $\forall\ j \in M$

Hacer el recorrido del árbol y de cada nodo comenzando por las hojas.

1. Determinar $STE(t)$ para $i = \{1,2 \dots Nclases\}$.
2. Determinar $DYE(t)$.
3. Podar si se cumple la condición $STE(t) \leq DYE(t)$.
4. Calcular tasa de error TE clasificando instancias del conjunto de poda.

Salida Parcial: Árbol podado, con su respectiva tasa de Error TE.

5. Determinar el menor TE de todos los ÁRBOLES generados.

SALIDA: Un árbol podado con error mínimo de poda.

Figura 4.25. Algoritmo para podar árboles de decisión por el método de poda MEP

Para entender mejor este método se usará el árbol construido a partir de las instancias de entrenamiento de la tabla 4.1. En este caso, no se usa conjunto de poda; por lo tanto, se analizará el mismo árbol de la figura 4.4., que está mejor representado en la figura 4.26.

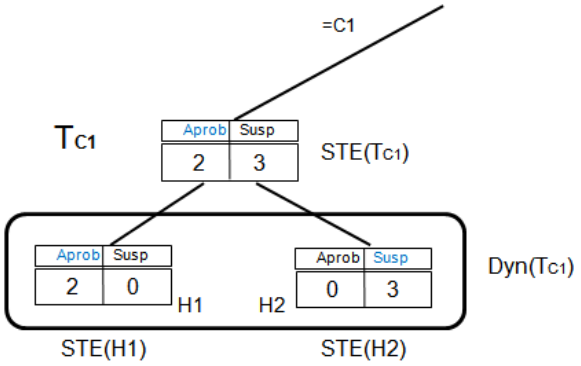

Figura 4.26. Nodos analizados para el cálculo del error estático y dinámico

Al analizar los nodos hojas H1 y H2, se puede observar que el error estático es $STE(t^*) = \min\{1 - p_i(t^*)\}$ y $p_i(t^*)$, es la probabilidad esperada para que una instancia alcance el nodo t con el valor de la i-ésima clase. Utilizando el valor para m=2 en la expresión 4.29.

Lo que desea calcular es el error estático de cada una de las hojas del árbol; para ello se usa la siguiente expresión:

$$STE(\text{nodo}) = \text{Min}\{1 - p_i(\text{nodo})\}$$

$$= Min\{1 - p_i(nodo)\}, donde\ i\ son\ los\ valores\ de\ la\ clase$$

En los tres casos, es claro que el valor de la clase aprobado es el adecuado para encontrar el menor valor que corresponde al error estático, como se verá a continuación

Para el nodo H1:

$$p_{aprobado}(\text{H1}) = \frac{n_i(t^*) + m * p_{ai}}{n(t^*) + m} = \frac{2 + 2 * 0.6428}{2 + 2} = 0.1786$$

Para el nodo H2:

$$p_{aprobado}(\text{H2}) = \frac{n_i(t^*) + m * p_{ai}}{n(t^*) + m} = \frac{0 + 2 * 0.6428}{3 + 2} = 0.2571$$

Para el nodo, en el supuesto caso que sea conveniente podar:

$$p_{aprobado}(\text{Tc1}) = \frac{n_i(t^*) + m * p_{ai}}{n(t^*) + m} = \frac{2 + 2 * 0.6428}{5 + 2} = 0.4693$$

El error dinámico, como su expresión lo indica, es la suma ponderada de los errores estáticos para cada una de las hojas del sub árbol que se quiere podar. En nuestro caso, tendremos:

$$\text{Dyn(Tc1)} = \frac{2}{5}\big(\text{STE(H1)}\big) + \frac{3}{5}\big(\text{STE(H2)}\big) = 0.2257$$

Debido a que la condición de poda no se cumple, se decide no podar el subárbol Tc1 y se procede a utilizar el conjunto de poda.

4.5.4. Critical Value Pruning (CVP)

J. Mingers [Ming-87] propone este método muy similar a una técnica de prepoda, debido a que la condición impuesta para podar se basa en la información que sirvió para generar el árbol.

El algoritmo propuesto basa su funcionamiento en la definición del umbral: si determinado nodo contiene un valor crítico menor al umbral, entonces se poda. Este proceso genera una serie de árboles $\{T_{max} \ldots T_1, T_0\}$, los cuales determinan una tasa de error e cuando han sido clasificados con el conjunto de instancias independiente de poda. Para este experimento, se usa una el $Gainratio$ [Espo- 97] como el valor crítico para la poda.

Condición de poda $Gainratio(t) \leq Threshold$

ENTRADA:

- Un árbol T_{max} con su respectiva ganancia $Gainratio$ en cada nodo t.
- $Threshold$ inicial = 0.1

PROCESO:

Hacer el recorrido del árbol y de cada nodo comenzando por las hojas.
For ($Threshold = 0.1$; $Threshold < 1$; $Threshold = Threshold + 0.1$)

1. Podar todos los nodos t si se cumple la $Gainratio(t) \leq Threshold$.
2. No podar si existe algún nodo hijo de t que contenga un mayor valor de ganancia que el umbral.
3. Calcular tasa de error e clasificando instancias del conjunto de poda en el árbol generado.

Salida Parcial: Árboles podados $\{T_{max} \ldots \ldots T_1, T_0\}$, y su respectiva tasa de error e

4. Determinar el menor e de todos los ÁRBOLES generados.

SALIDA: Un árbol podado con tasa mínima de error con respecto a un conjunto de poda.

Figura 4.27. Algoritmo para podar árboles de decisión por el método de poda CVP

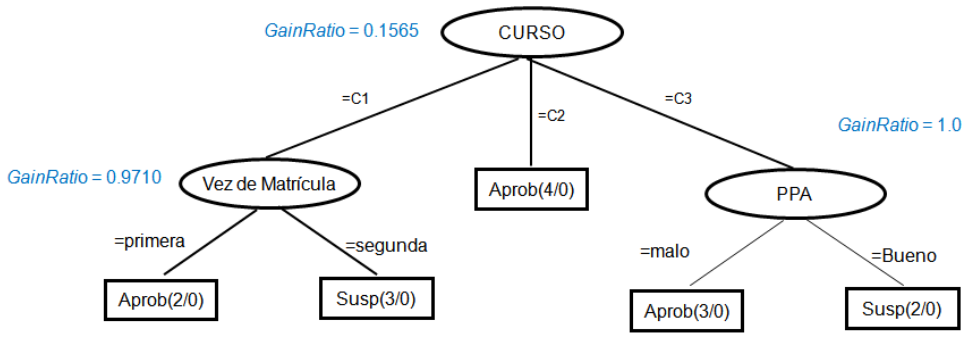

Figura 4.28. Árbol de decisión con respectivas ganancias en cada nodo

Observamos que si comenzamos la iteración con $Threshold = 0.1$ no se realiza la poda, porque ningún nodo cumple la condición $Gainratio(t) \leq 0.1$, mientras que si se usa la siguiente iteración para el nodo Curso, vemos que $Gainratio(t) = 0.1565 \leq 0.2$. Si podamos, estaríamos dejando de lado los nodos hijo Vez y PPA, los cuales no cumplen con la condición de poda. Para este caso, a pesar de que el nodo Curso debería podarse, no se poda para evitar la pérdida de nodos con una mayor $Gainratio$.

4.5.5. Cost-Complexity Pruning (CCP)

Este método fue propuesto por Breiman [Brie-84] y forma parte del software CART. Trata de encontrar el mejor compromiso entre los errores estimados de un conjunto de poda o validación cruzada y el tamaño del árbol de decisión. En otras palabras, el objetivo del CCP es reducir al mínimo el error y la complejidad (igual al número de hojas de un árbol) de un árbol de decisión.

Costo Total nodo subárbol: T_t $R(T_t) + \alpha * L_t$ ……. (4.33)

Costo Total nodo hoja t^* : $R(t^*) + \alpha$ ……. (4.34)

Medida de aumento de la tasa de error: $\alpha = \frac{R(T_t) - R(t)}{L_t - 1}$ ……. (4.35)

Desviación estándar: $SE = \sqrt{\frac{e*(1-e)}{N}}$ ……. (4.36)

N: instancias del conjunto independiente de prueba y e: Tasa de error en los sucesivos árboles $\{T_{max} .. T_1, T_0\}$.

El árbol se debe podar cuando el T_t que contenga el $min\{\alpha_0, \alpha_1 \ldots . \alpha_n\}$.

Se definen dos métodos de costo-complejidad:

0SE: Método que busca el árbol con una tasa de error mínima e_{min} en el conjunto de todos los árboles generados $\{T_{max} \ldots\ldots\ T_1, T_0\}$. Este error e_{min} se calcula con el conjunto de prueba independiente.

1SE: Método que agrega un factor de corrección en la búsqueda del tasa se error mínima "$e_{min} + SE$".

Cuando α = 0 se tiene T_{max} y si $\alpha = \propto$, entonces se tiene solo en nodo raíz.

Pseudocódigo 0SE ,1SE

ENTRADA:

- Un árbol T_{max} con todas las instancias de entrenamiento clasificadas en los nodos.
- Conjunto independiente de ejemplos
- Una medida X que representa el error estándar para la poda.
- Hacer $\alpha = 0$ para la primera iteración

PROCESO: Mientras $L_{Tmax} > 1$

Hacer el recorrido del árbol y de cada nodo comenzando por las hojas.

1. Determinar el parámetro α para cada nodo t y subarbol T_t.
2. Clasificar las instancias independientes y calcular la tasa de error e.
3. Encontrar el $min\{\alpha_0, \alpha_1 \ldots . \alpha_n\}$ y podar los T_t que cumplen con α.

Salida Parcial: Árboles podados, con su respectiva tasa de error e.

4. if (Método = 0SE)

4a. Determinar e_{min} para los ÁRBOLES $\{T_{max} \ldots \ldots T_1, T_0\}$,

else

if(Método = 1SE)

4b. Si T contiene el e_{min}, entonces buscar el mayor e que cumpla la condición $e < e_{min} + SE$ y devolver su respectivo T'.

SALIDA: Un árbol podado

Figura 4.29. Algoritmo para podar árboles de decisión por el método de poda CCP

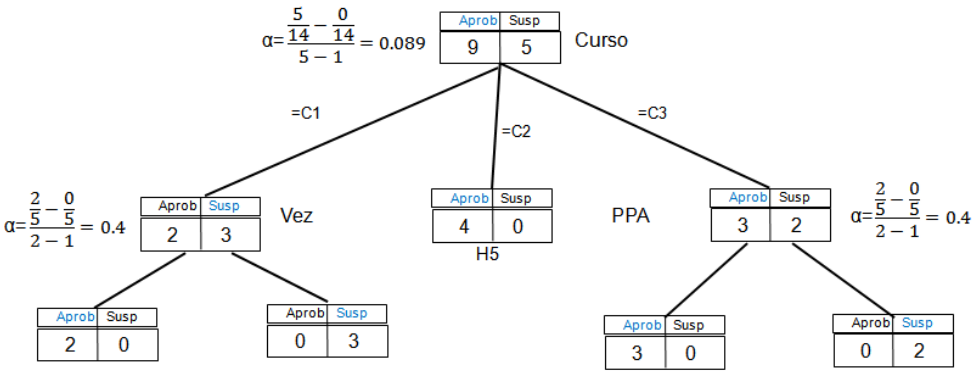

Figura 4.30. Cálculo del parámetro α para cada nodo diferente de una hoja

Primero calculamos el parámetro de aumento de error α en el nodo Vez:

$$\alpha_{VEZ} = \frac{R(t) - R(T_t)}{L_t - 1} = \frac{\frac{5}{14} - \frac{0}{14}}{5 - 1} = 0.089$$

Así sucesivamente en los nodos PPA y Curso:

$$\alpha_{CURSO} = \frac{\frac{2}{5} - \frac{0}{5}}{2 - 1} = 0.4 \quad , \alpha_{PPA} = \frac{\frac{2}{5} - \frac{0}{5}}{2 - 1} = 0.4$$

A continuación, podamos aquel nodo o nodos que posean el valor min α , que en nuestro caso es el nodo Curso, y se clasifican nuevamente todas las instancias, y se obtiene un árbol reducido con un solo nodo.

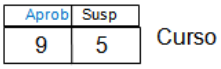

Figura 4.31. Árbol de clasificación después de la poda

El proceso es recursivo y se van generando series de árboles $\{T_{max} \ldots\ldots T_1, T_0\}$, los cuales son probados con nuevas instancias para medir su efectividad de clasificación.

4.5.6. Error Based-Pruning (EBP)

Fue propuesto e implementado por J.R Quinlan [Quin-93] en el software C4.5. Usa las instancias de entrenamiento para determinar un estimador de tasa de errores, asumiendo que los errores en las hojas se modelan como una distribución binomial.

Cuando una hoja cubre N instancias y E de ellas son incorrectas, se puede interpretar como E "eventos" en N "ensayos". Es entonces que se puede determinar la probabilidad de error en la población cubierta por la hoja dado un límite de confianza CF. Se usa el límite de confianza superior $U_{CF}(E,N)$ para minimizar la tasa de error observada. Cuando una hoja cubre N casos con un estimador de error $U_{CF}(E,N)$, entonces la tasa de errores predicha es $N * U_{CF}(E,N)$.

La diferencia de este método con los demás estudiados es la posibilidad de hacer *grafting*. Esto consiste en no reemplazar el subárbol T_t por el nodo t cuando se cumple la condición de poda, sino en redibujar otro subárbol para reemplazar a T_t.

Límite de confianza superior: $U_{CF}(E,N) = \left(f + \frac{z^2}{2N} + z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}}\right) / \left(1 + \frac{z^2}{N}\right)$ ……. (4.37)

Corrección de la distribución binomial: $f = \frac{E+0.5}{N}$ ……. (4.38)

CF y la distribución normal: $CF = f(z)$ ……. (4.39)

Predicción de error en nodo t: $P_e(t) = N * U_{CF}(E,N)$ ……. (4.40)

Predicción de error subárbol T_t: $P_e(T_t) = \sum_{s \in L_t} P_e(s)$……. (4.41)

Condición de Poda $P_e(t) \leq P_e(T_t)$ ……. (4.42)

Pseudocódigo

ENTRADA:

- Un árbol T_{max} con todas las instancias de entrenamiento clasificadas en los nodos.

PROCESO: Hacer el recorrido del árbol y de cada nodo comenzando por las hojas.

1. Para un nodo t determinar $P_e(t)$ y $P_e(T_t)$.
2. Podar si se cumple la condición $P_e(t) \leq P_e(T_t)$.
   2a.- Reemplazar subárbol T_t por t.
   2b.- Caso contrario aplicar *grafting* y redibujar el subárbol T_t.

SALIDA: Un árbol podado.

Figura 4.32. Pseudocódigo del algoritmo de poda EBP

Como ejemplo de este método, analicemos la figura 4.31 para el nodo Vez y sus hijos.

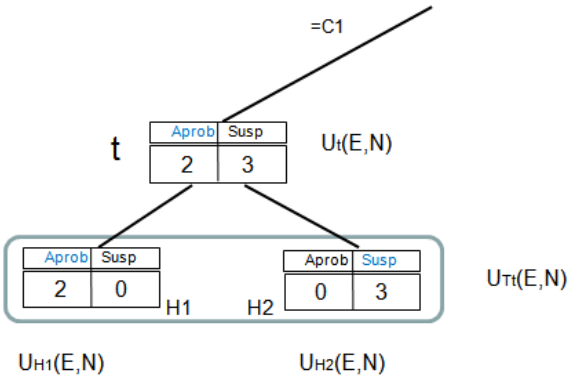

Figura 4.33. Proceso de poda EBP para los árboles de decisión

Primero se calcula la probabilidad de error predicho $\mathrm{U_{H1}(E,N)}$, $\mathrm{U_{H2}(E,N)}$ en las hojas H1 y H2 respectivamente:

$$U_{CF\%H1}(0,2) = (1 - \exp(\log(0.25) / 2)) \quad ;$$

$$U_{CF\%H1}(0,2) = 0.5$$

De igual forma

$$U_{CF\%H2}(0,3) = 0.37$$

A continuación, hacemos lo mismo en el nodo t asumiendo que se podan las hojas y volvemos a clasificar las instancias en el nodo padre; entonces:

$$U_{CF\%t}(2,5) = \left(\frac{2+0.5}{5} + \frac{0.69^2}{2*5} + 0.69 * \sqrt{\frac{0.5}{5} - \frac{0.5^2}{5} + \frac{0.69^2}{4*5^2}}\right) / \left(1 + \frac{0.69^2}{5}\right) = 0.64$$

Finalmente, calculamos los errores predichos y comparamos:

$$P_e(Tt) = 2 * U_{CF\%H1}(0,2) + 3 * U_{CF\%H1}(0,3) = 2.11$$

$$P_e(t) = 5 * U_{CF\%t}(2,5) = 3.2$$

Como observamos, no se cumple $P_e(t) \leq P_e(T_t)$; por lo tanto, no se poda.

4.6. Técnicas de evaluación

En lo que va de este capítulo, hemos presentado varias de las técnicas más utilizadas en minería de datos. Estas técnicas permiten crear un modelo basado en un conjunto de datos que recibe el nombre de "conjunto de entrenamiento". Una vez creado el modelo y aplicado a un conjunto de prueba, cabe preguntarse cómo podemos saber si los resultados de aplicar una técnica determinada de clasificación a un conjunto de entrenamiento son lo suficientemente validas. Para responder a esta pregunta, se tienen que aplicar algunos métodos que sirven para evaluar la calidad de un modelo a partir de un conjunto de datos.

Por otro lado, estimar el comportamiento de un clasificador inducido por alguna técnica de minería de datos utilizando alguna métrica determinada es importante, no solo para predecir el valor de la futura predicción en términos de la métrica elegida, sino también para escoger un clasificador a partir de un conjunto dado. Esta última tarea recibe el nombre de "Selección del modelo" (Model Selection) [Wolp-92].

Una primera aproximación nos llevaría a utilizar el propio conjunto de entrenamiento como referencia para evaluar la calidad de un modelo. Sin embargo, esto es equivocado, ya que la evaluación estaría sesgada al conjunto de datos de entrenamiento, premiando al modelo que se ajusta a estos datos dejando de generalizar la evaluación para otros tipos de datos.

Una segunda aproximación intuitiva sería evaluar el modelo construido por un clasificador sobre un conjunto de datos diferente al conjunto de datos de entrenamiento. El hecho de utilizar dos conjuntos de datos independientes, uno para el aprendizaje del modelo o de las hipótesis y el otro para evaluarlas, permite evitar el problema de premiar el sobreajuste. Sin embargo, aún existe el problema de que el resultado del modelo aprendido sea demasiado dependiente del conjunto de entrenamiento. Esta última afirmación contraviene el hecho de que la construcción del modelo debería ser muy general como para que cualquier conjunto de datos de prueba tenga porcentajes de error parecidos a los del conjunto de prueba inicial. En este caso, diremos que el modelo aprendido es el óptimo para el modelo de aprendizaje utilizado.

Existen otro tipos de aproximaciones, como la de evaluar el modelo por error o por costos. En esta última, se evalúa el costo de los errores cometidos por un modelo, siendo el mejor el que menor costo presenta.

A continuación, presentaremos algunas de las técnicas más usada para evaluar de manera confiable a las técnicas de clasificación revisadas en este capítulo.

4.6.1. Evaluación del modelo basado en precisión

Frecuentemente, las medidas para evaluar el modelo están basadas en la precisión de la hipótesis o, de manera equivalente, en el porcentaje de error entre la hipótesis y la función. En el presente trabajo, en el apartado 3.5, se ha hecho un estudio detallado de las métricas de evaluación que luego será aplicado en el capítulo referido a la experimentación.

4.6.1.1. Oversampling

Oversampling es una técnica cuyo objetivo principal es la reducción del porcentaje de desaciertos donde la clase correctamente predicha es la clase minoritaria y la clase incorrectamente predicha es la clase mayoritaria. Este tipo de error se produce principalmente porque el clasificador, al tratar de minimizar su error, se ajusta más a la clase mayoritaria.

La solución que plantea este método consiste en igualar el número de registros que corresponden a cada una de las clases. Para lograr esto, debe repetirse las instancias de las clases minoritarias hasta que se alcance el número de registros de la clase mayoritaria. El resultado de este proceso es un aumento del error global del clasificador sobre datos de *testing*, pero una disminución de estos desaciertos en la clase minoritaria que, en algunos dominios, pueden considerarse nocivos.

4.6.1.2. Aproximación de Holdout

El método Holdout, llamado algunas veces “Test Sample Estimation”, particiona los datos en dos conjuntos mutuamente excluyentes llamados “conjunto de entrenamiento” y “conjunto de prueba”, o “prueba Holdout”. Es común designar 2/3 de los datos como conjunto de entrenamiento y el 1/3 restante como conjunto de prueba. El conjunto de entrenamiento se somete a un algoritmo de clasificación (inductor), y la función inducida (clasificador) es probada con el conjunto de instancias restantes (conjunto de prueba).

El rendimiento de un clasificador, utilizando el método de Holdout, es una medida que depende de la división que se haga en el conjunto de entrenamiento y conjunto de prueba. En el remuestreo aleatorio, muchas veces llamado Holdout Resampling (usado en el punto 6.2.3.1), el método de Holdout es repetido k veces, y las métricas de rendimiento, según la que se tenga que usar, son los promedios de aplicar el inductor con el método de Holdout a cada uno de los conjuntos.

4.6.1.3 Aproximación de validación cruzada

La validación cruzada es un mecanismo que reduce en gran medida la dependencia del resultado del experimento al conjunto que se utiliza para entrenarlo. Consiste en dividir el conjunto de entrenamiento o el conjunto total de evidencias en n subconjuntos disjuntos de similares tamaños y que conserven la clase. Luego, de ellos se toman n1 subconjuntos como conjuntos de entrenamiento, y el restante se emplea como uno de prueba. Este procedimiento se repite n veces, utilizando siempre un conjunto de prueba diferente. El error final de la validación cruzada se calcula como el promedio de los n errores obtenidos en los n experimentos.

Una ventaja clave de la validación cruzada es que la varianza de los n errores de cada una de las pruebas, permite estimar la variabilidad del método de aprendizaje con respecto a los datos de entrenamiento. Asimismo, debido al diseño de la validación cruzada, es evidente que los n conjuntos de prueba son disjuntos. No obstante, esto no es cierto para cada uno de los conjuntos de entrenamiento.

Esto último podría representar una desventaja, debido a que si el conjunto inicial se parte en n subconjuntos independientes, cada uno de los conjuntos de entrenamiento tendrá siempre $\frac{n-2}{n}100\%$ de instancias en común. Este solapamiento podría desvirtuar la calidad de la estimación, por lo que se puede considerar una variante de validación cruzada $\frac{k}{2}$x2 como alternativa. Esta técnica consiste en aplicar $\frac{k}{2}$ repeticiones de una validación cruzada con n=2. En cada una de las $\frac{k}{2}$ iteraciones, el conjunto de datos se divide en dos subconjuntos disjuntos (de entrenamiento y de prueba) con el mismo número de instancias.

En la figura 4.32, se esquematiza el proceso de validación cruzada con 4 pruebas (4-fold Cross Validation).

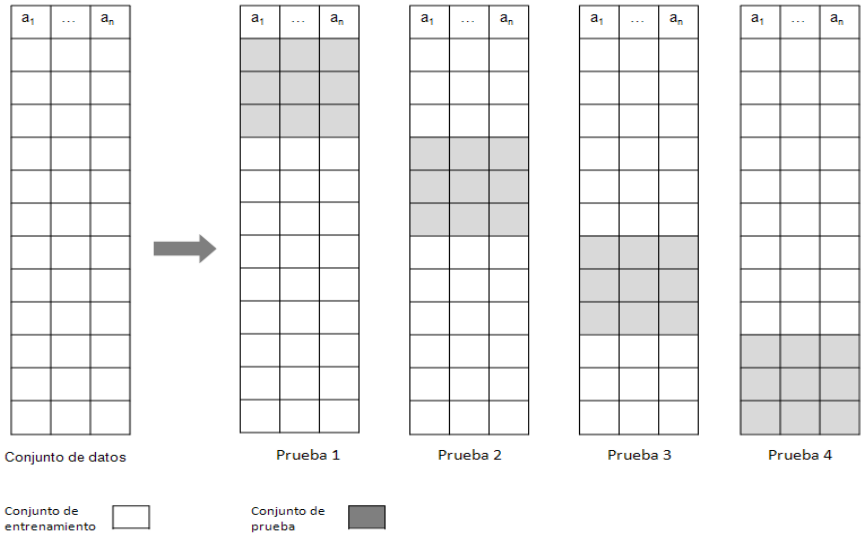

Figura 4.34. Diseño esquemático de la validación cruzada

4.6.2. Evaluación del modelo en base a costos

Se sabe que la finalidad de los clasificadores es predecir, mediante el uso de experiencias previas, la decisión que debe tomarse ante un problema que presenta más de una acción como respuesta. Sin embargo, los clasificadores, pese a ser instrumentos sumamente productivos para tal fin, tienen un límite, no por el hecho de que algunas de sus predicciones puedan ser erróneas, sino porque dichos errores conllevan efectos diversos, lo que equivale a decir que las consecuencias de los errores no siempre son equivalentes, incluso las referidas a un mismo problema.

De hecho, es bastante usual el caso en que los errores de clasificar ejemplos de clases minoritarias en la clase mayoritaria (decir que aprobará cuando, en realidad, suspende) tiene asociado un mayor costo que la situación inversa. Es evidente que los costos dependen del dominio de aplicación, pero, en cualquier caso, es excepcional que los costos sean uniformes para un determinado problema. Por lo tanto, muchas veces, la precisión no es la mejor medida para evaluar la calidad de un modelo o de un algoritmo de aprendizaje. En nuestro caso particular, es muy relativo diferenciar los errores debido a que si se presentase el caso de un falso positivo, este podría tener explicación en diversos factores, más subjetivos que en el campo cuantitativo.

El aprendizaje sensible al costo puede considerarse como una aproximación más realista del aprendizaje; en este contexto, la calidad de un determinado modelo, algunas veces, se mide en términos de minimización de costos en vez de minimización de errores.

Una manera habitual de expresar los costos en problemas de clasificación es mediante una matriz asociada de costos. Esta es de dimensión nxn en el caso de que la clase tenga n valores. En ella se expresan los costos de todas las combinaciones posibles entre la clase predicha y la clase real. Por ejemplo, en nuestro caso, el costo de predecir a un estudiante que aprobará una asignatura, cuando, en realidad, al finalizar el curso la suspende (desaprueba) puede ser (es solo un estimado) diez veces más grave que decirle a un estudiante que suspenderá cuando realmente aprueba.

Para estimar el costo total de un clasificador, se utiliza la matriz de costos combinada con la matriz de confusión, detallada en el punto 3.5.2.4. Cada celda de la matriz de confusión $M_{i,j}$enumera, para cada clase, los casos en los cuales el clasificador ha predicho una clase. De manera similar, cada celda de la matriz de costos $C_{i,j}$enumera, para cada clase, el costo asociado a cada uno de los errores (falsos positivos y negativos). Es obvio que la diagonal principal de la matriz de costos tiene valores nulos debido a que, en esos lugares, la matriz de confusión representa los aciertos.

Otra forma de enfrentar errores con diferentes costos es aplicar la técnica del *oversampling*. Esta es una técnica cuyo objetivo principal es la reducción del porcentaje de desaciertos, donde la clase real es una clase minoritaria y la clase incorrectamente predicha es la clase mayoritaria. Este tipo de error se produce principalmente porque el clasificador, al tratar de minimizar su error, se ajusta más a la clase mayoritaria.

La solución que plantea este método consiste en igualar el número de registros que corresponden a cada una de las clases. Para lograr esto, deben repetirse las instancias de las clases minoritarias hasta que se alcance el número de registros de la clase mayoritaria. El resultado de este proceso es un aumento del error global del clasificador sobre data de prueba, pero una disminución de estos desaciertos en la clase minoritaria que en algunos dominios puede considerarse nocivos.

En el caso particular del presente trabajo, es decir, el de la predicción de si un alumno aprueba o no, el error con mayor costo es decirle a un estudiante que aprobará la asignatura cuando realmente no es así. Esto ocurre porque se trabaja con un conjunto desbalanceado y la clase mayoritaria es aprobado; por lo tanto, cualquier algoritmo tendrá una tendencia a predecir más de la clase mayoritaria que sobre la minoritaria.

Capítulo 5:

Metodología para la preparación de datos

El objetivo de la metodología consiste en construir un modelo a partir de los datos históricos de los alumnos que sea capaz de representar la realidad de manera precisa [Vial-10a]. Este modelo permite predecir si la decisión de un alumno de inscribirse en un curso lo llevará a aprobar o desaprobar mediante un proceso de clasificación. Consecuentemente, esta predicción ayuda al alumno a tomar una mejor decisión acerca del conjunto de cursos que llevará en un semestre. El caso de estudio del presente documento es una problemática generalizada en los datos extraídos. Por esto, es necesario explicar de manera muy sucinta dicho contexto y las normas académicas pertinentes.

En primer lugar, la matrícula se realiza mediante un sistema web que permite que los alumnos escojan los cursos que desean llevar del conjunto de cursos hábiles[1]. Un alumno podrá intentar aprobar un curso un máximo de tres veces, para lo cual deberá obtener un puntaje mayor o igual a 11 en la escala vigesimal. En caso de no lograrlo, será impedido de continuar sus estudios en la facultad. Si un alumno desaprueba un curso y el semestre académico coincide con un cambio de plan de estudios, el alumno deberá llevar el curso correspondiente al nuevo plan de estudios y no al de aquel que cambia.

Debido a que en el presente estudio se usan datos históricos, es necesario tomar en cuenta los cambios del plan de estudios tanto en el nivel de contenido de los cursos como en el nivel de su estructura.

Los posibles cambios son los siguientes:

- Inclusión de nuevos cursos.
- Modificación de las relaciones de dependencia entre cursos.
- Eliminación de cursos.
- Reemplazo de un curso por otro.

En la Figura 5.1, se puede observar la relación de dependencia de Cálculo II tanto en la malla 1996-1 como en la 2006-1 para, de esta manera, contrastar los cambios realizados en ella. El gráfico nos muestra que el curso de Matemática Básica II fue eliminado y que, además, se incluyó el curso de Álgebra Lineal a modo de reemplazo del curso retirado.

[1] Un curso hábil es aquel que pertenece al plan de estudios de la carrera y cuyos requisitos, en el caso de tenerlos, han sido aprobados.

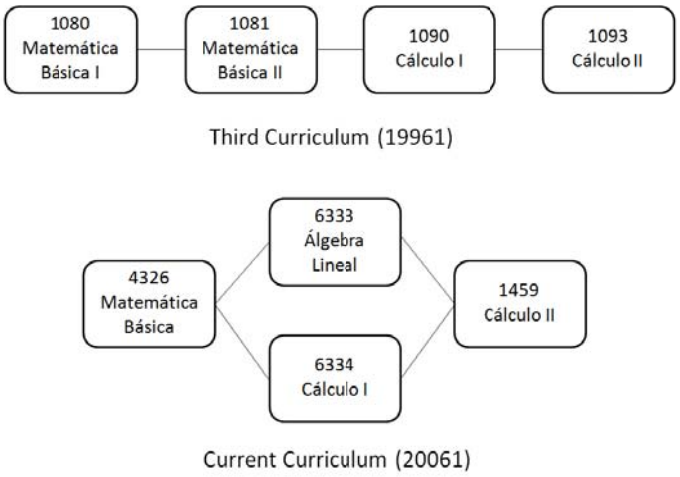

Figura 5.1. Submalla curricular del área académica de matemática y de operaciones

Un concepto que se usará en secciones posteriores es la distancia entre dos cursos. De manera muy intuitiva, la distancia entre Cálculo II y Matemática Básica es de dos, mientras que la distancia entre Matemática Básica y Cálculo I es de uno.

Para un mejor entendimiento de este concepto, usamos como ejemplo la figura 5.2, en la cual se representa una submalla curricular, que en nuestro caso es un extracto del plan curricular de la carrera, correspondiente al curso de Gráficos por Computadora. Esta muestra sus cursos prerrequisitos, los cuales, como se observa, pertenecen a dos áreas académicas.

Por un lado, el área de ciencias básicas, las asignaturas de Cálculo III, Cálculo II Cálculo I, Algebra Lineal y Matemática Básica. Por el lado del área de software, las asignaturas de Programación, Estructuras de Datos y Algoritmos, Estructuras Discretas en Computación, Fundamentos de Programación, Álgebra Lineal y Matemática Básica. Además, se muestra un concepto de nivel que agrupa los cursos de acuerdo con su ubicación en el grafo:

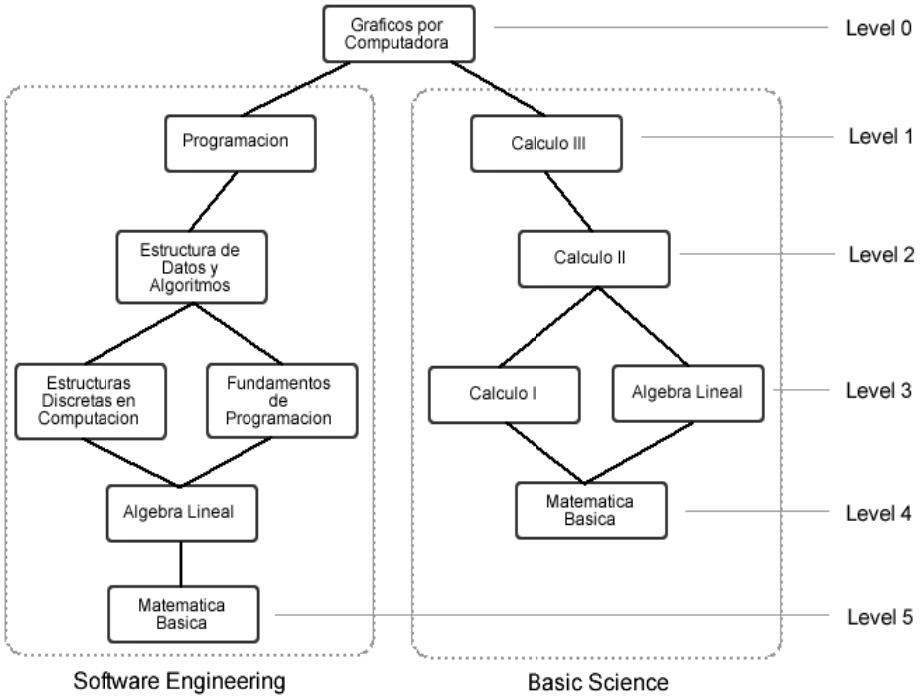

Figura 5.2. Submalla curricular de la asignatura de gráficos por computadora

Por lo tanto, podemos definir la distancia como la diferencia de niveles entre un curso y uno de sus requisitos en el grafo de dependencia elaborado para un curso determinado.

En la figura 5.2., se observa que la distancia entre el curso Gráficos por Computadora y sus requisitos inmediatos, Programación y Cálculo III, es de 1, mientras que la distancia respecto a Matemática Básica es de 4. En el caso de que un curso requisito pertenezca a dos o más áreas académicas distintas, el valor correspondiente a la distancia entre ese curso y el curso de nivel 0 será la menor de ellas.

Además del grafo de dependencia, podemos derivar una función que llamaremos "conjuntos de cursos requisitos", que nos devuelve el conjunto de requisitos de un curso determinado de acuerdo con un valor de distancia. La relación puede ser descrita como:

$$SPC_{(curso,distancia)} = Conjunto\ de\ asignaturas\ requisitos$$

A continuación, presentamos algunos ejemplos:

$$\text{SPC}(\text{Gráficos por Computadora}\ , 1) = \{\text{Programación, Cálculo III}\}$$

$$\text{SPC}(\text{Cálculo II}\ , 2) = \{\text{Cálculo I, Algebra Lineal, Matemática Básica}\}$$

Para esta experimentación, se utilizaron los datos de la Facultad de Ingeniería de Sistemas que, a lo largo de toda su historia, ha tenido un total de ocho planes de estudios distintos. Cada uno de estos ha tenido un período de vigencia acotado por un Ciclo de Inicio y un Ciclo de Fin. Los datos fueron organizados en forma de tabla, donde cada fila (registro) representa los datos correspondientes a un estudiante y a una asignatura. Así, si un estudiante determinado ha cursado C asignaturas hasta el momento, en la tabla habrá C registros que contengan sus datos.

El resultado de la disposición de los datos es una tabla m x n (m registros y n atributos) estudiantes-atributo, donde las columnas contienen los datos correspondientes a cuando el estudiante cursó la asignatura: el número de asignaturas cursadas simultáneamente, nombre y código de la asignatura, nota obtenida, promedio ponderado acumulado (PPA), etc.

La clase que se considera en la aplicación del algoritmo supervisado de aprendizaje es la nota que obtuvo el estudiante en la asignatura que se representa en el registro. Esta ha sido discretizada siguiendo las normas actuales de la institución: suspenso (desde cero hasta 10.99) y aprobado (desde 11.00 hasta 20).

Para efectos de esta investigación, al estudiante que solicita una recomendación se lo llamará “estudiante activo”. Dado que nuestro objetivo fundamental es recomendar a un estudiante sobre su matrícula en determinada asignatura, este no podrá estar representado en la tabla en ningún registro que corresponda a la asignatura en consulta. Así mismo, debido al plan de estudios de cada carrera universitaria, en donde existe un número limitado de asignaturas por carrera, y al dominio de esta investigación, los problemas de escalabilidad y dispersión, inherentes a este tipo de representación en sistemas de filtrado colaborativo tradicionales [Sarw-01], no se discutirán.

5.1. El proceso de KDD en el ámbito educativo

Esta sección esboza el proceso de procesamiento de datos desde su extracción hasta la generación del modelo. Es decir, considera los subprocesos de Extracción de Datos, Selección, Transformación y Limpieza, y Minería de datos. Los procesos se muestran en la (Fig. 5.3) y se detallan a continuación:

5.1.1. Extracción de Datos

Este proceso consiste en copiar los datos de la base de datos OLTP de la Universidad hacia una base de datos distinta para poder procesarlos fuera de línea. Básicamente, consta de dos fases. La primera consiste en extraer los datos normalizados, es decir, los planes de estudios, los cursos y las notas de los alumnos de

la base de datos de la Universidad de Lima en forma de archivos con un formato adecuado (en nuestro caso, por su eficacia, se empleó el formato CSV). La segunda fase consiste en cargar los datos de estos archivos a la base de datos de nuestra aplicación.

5.1.2. Selección, Transformación y Limpieza

Consta de cinco fases que a continuación se detallan.

- **Normalización de datos**: Consiste en cargar los datos de la base de datos de carga a la base de datos procesada y normalizada.
- **Agregación de datos**: Consiste en construir tres tablas de ayuda (Linear Dependency, Backwards Equivalence y Forward Equivalence), que permitirán realizar con mayor facilidad la tercera fase.
- **Generación de datos**: Consiste en calcular los valores de las variables utilizadas en este estudio, la dificultad del curso y el potencial de un alumno en determinada rama antes de que asuma una asignatura.
- **Filtrado de datos**: Consiste en eliminar los registros que no son necesarios para realizar el descubrimiento de patrones. En este caso, se han considerado los datos correspondientes a los ciclos regulares y que pertenecen a la facultad de Ingeniería de Sistemas, los registros de los cursos de primer semestre llevados por primera vez y los de los cursos convalidados de los alumnos de traslado externo.
- **Selección de atributos**: Consiste en permitir al administrador del sistema la selección de los campos (atributos) que serán parte del conjunto de entrenamiento y que se utilizarán en el siguiente subproceso.

Para efectos de este estudio, hemos seleccionado los siguientes campos:

1. Nombre del curso : Cadena
2. Vez : Número discreto
3. Promedio Inicio : Número continuo (Promedio Ponderado Acumulado)
4. Potencial : Número continuo
5. Creditaje : Número continuo
6. Créditos Matriculados : Número continuo
7. Dificultad : Número continuo
8. Clase : Aprobado o Desaprobado

5.1.3. El subproceso de minería de datos

Se lleva a cabo utilizando algoritmos de clasificación, como son c4.5, Naïve Bayes, algoritmo de vecinos cercanos, etc. Consiste en el descubrimiento de los patrones de rendimiento académico representados como árboles de decisión.

En la figura 5.3 [Vial-10a], se detalla el proceso considerando tres subprocesos. Los dos primeros son los que corresponden específicamente a la preparación de los datos de entrenamiento. El tercero es la aplicación del algoritmo de aprendizaje.

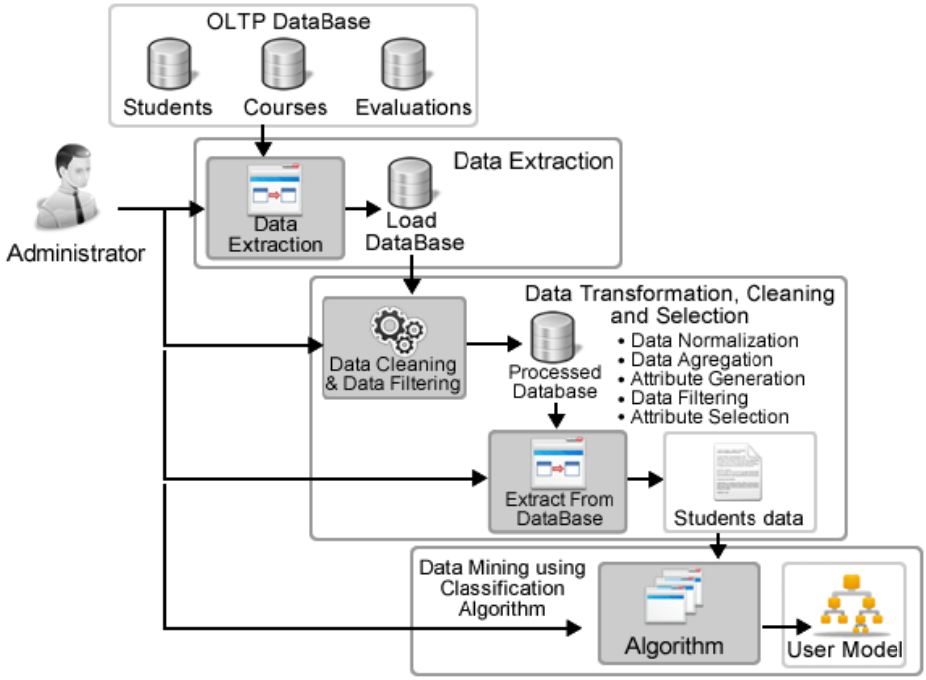

Figura 5.3. Proceso de descubrimiento del conocimiento

5.2. Descripción de los datos

Los estudiantes de la Universidad poseen datos estáticos y dinámicos. Los primeros son asignados al estudiante por la Universidad y generalmente corresponden a datos demográficos como, por ejemplo, su código, nombre, dirección, etc. La mayor parte de datos en este estudio son dinámicos. Reciben este nombre, porque van cambiando con el tiempo. Estos datos los asigna el sistema automáticamente a medida que la transición de los estados del sistema se va modificando, ya sea por el propio sistema, cuando detecta que tiene que hacerlo (por ejemplo: nivel del alumno en la actualidad) o cuando el estudiante selecciona determinadas condiciones para que el estado del sistema cambie (por ejemplo, cantidad de cursos matriculados o nombre o código del curso). A continuación, se detallan los datos dinámicos que corresponden a cada estudiante cuando cursa una determinada asignatura.

Atributos	Descripción tabla nota
Ciclo de ingreso a UL	Período académico en el cual postuló e ingresó a la Universidad.
Nivel del alumno en la actualidad	El nivel del alumno se determina por la asignatura pendiente de menor nivel. En este caso, el nivel del alumno corresponde a aquel en que está ubicado el estudiante al momento de extraer los datos y hacer las consultas.
PPA del alumno en la actualidad	Promedio ponderado acumulado de las notas del alumno obtenidas en cada asignatura en el momento de hacer la consulta. El valor de este atributo corresponde a números reales en el rango de 0 a 20. No fue segmentado, porque no corresponde al valor de la clase que considera el algoritmo de aprendizaje supervisado; sin embargo, el significado para un registro que corresponde a un estudiante es: [0, 11> Suspenso (SUSP): El estudiante tiene PPA desaprobado. [11, 20] Aprobado (APROB): El estudiante tiene PPA aprobado.
Situación actual del alumno en Ingeniería de Sistemas	Corresponde al atributo que indica si el alumno es aún estudiante en la carrera. Los posible valores que toma son los siguientes: estudiante (en caso de que tenga asignaturas pendientes), completó estudios (si aún no cumplió los requerimientos, no solicitó ser graduado o está en proceso al momento de la consulta), graduado (aquel que cumplió con todos los requisitos para serlo) o titulado (quien defendió una tesis de título profesional).
Período de matrícula	La carrera dura diez semestres académicos. Cada semestre es llamado también "período académico" y es representado por cinco cifras, las cuatro primeras corresponden al año calendario y la última puede ser 0, 1 ó 2 según el período: 0 para el período de verano enero-febrero, 1 para el período de abril-julio y 2 para el período de agosto-diciembre.
Cursos matriculados	Valor del número de asignaturas en las que el estudiante se matriculó en el mismo período en el cual se efectuó la matrícula en la asignatura representada en el correspondiente registro.
Créditos matriculados	Cada asignatura tiene un número de créditos asignados por la Universidad. Este campo corresponde a la suma de todos los créditos de todas las asignaturas en las que el alumno se ha matriculado, inclusive la que corresponde al registro actual.
Código del curso	Cadena numérica de cuatro dígitos que identifica de manera única la asignatura.
Curso	Nombre de la asignatura.
Créditos de curso	Es el número asignado a una asignatura, que representa la medida del número de horas teóricas y prácticas semanales que posee dicha asignatura.
Vez de Matrícula	Número que corresponde a la cantidad de veces que se ha realizado la matrícula en la asignatura representada en el registro.
Promedio	Nota final obtenida por el alumno después de haber cursado la asignatura. Rango de 0 a 20. [0, 11) Suspenso (SUSP): El estudiante deberá matricularse en el ciclo inmediato posterior en la misma asignatura. [11, 20] Aprobado (APROB): El estudiante podrá matricularse en la siguiente asignatura según su plan de estudios.
PPP con que inicia el ciclo	Es el promedio ponderado del período anterior a aquel en que se realizó la matrícula. En el caso de que el estudiante haya dejado de estudiar más de un período académico regular, el valor para dicho campo será 0.
PPP con que termina el ciclo	Es el promedio ponderado del período que corresponde a las asignaturas matriculadas (notas de las asignaturas del período anterior). Tres períodos son posibles en un año académico: 1 (abril – julio), 2 (agosto – diciembre) y 0 (enero – febrero). Rango de 0 a 20.
PPA con que inicia el ciclo	Es el promedio ponderado acumulado de todas las asignaturas cursadas anteriores a la matrícula.
PPA con que termina el ciclo	Promedio ponderado acumulado de todas las asignaturas cursadas, inclusive las del período actual.
Ciclo con que se calcula el inicio del ciclo	Es el último período académico en que realizó su matrícula.

Tabla 5.1. Descripción de atributos del conjunto inicial de datos

5.2.1. Descripción de las tablas

Los datos originalmente fueron extraídos de la base de datos OLTP. La información consta de cuatro tablas. La descripción de cada una de las tablas se puede observar en la tabla 5.2. Estas son tabla Nota, tabla Plan de Estudios, tabla Equivalencia y tabla Requisito. La existencia de cada una de estas tablas se justifica al momento de procesar los datos.

Tabla	Descripción	Registros
Nota	Contiene las notas de todos los alumnos desde el período 1991-2 hasta el período 2009-1.	250 843
Plan de estudios	Contiene el listado de asignaturas con los planes de estudios a los que corresponden.	667
Equivalencia	Contiene el listado de asignaturas por plan de estudios y su respectiva asignatura equivalente inmediata anterior.	311
Requisito	Contiene el listado de asignaturas y sus requisitos inmediatos por cada plan de estudios.	579

Tabla 5.2. Descripción de las tablas de datos extraídas de la base original

En la figura 5.4., se pueden visualizar las tablas con sus respectivas variables.

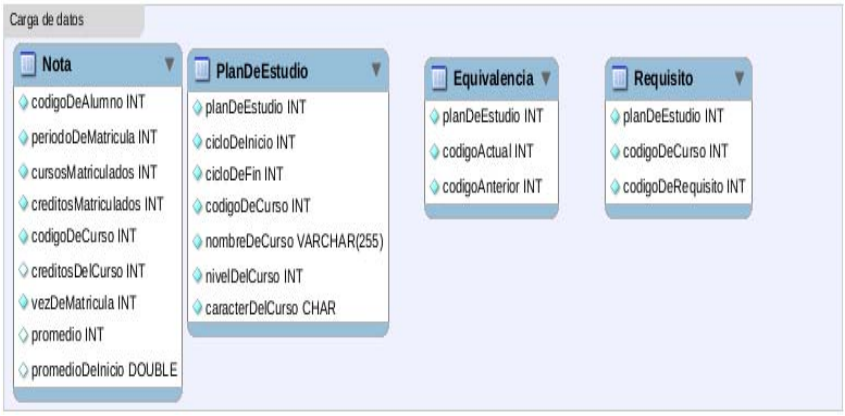

Figura 5.4. Tablas extraídas de la base original

En la tabla 5.3., se detalla cada uno de los atributos que pertenecen a los datos de la tabla Nota. Cabe destacar que estos pertenecen a los estudiantes de la Facultad de Ingeniería de Sistemas desde su creación y que efectuaron su matrícula a partir del año 1991. El total de registros que contiene esta tabla es de 250843. Uno de estos registros equivale a la nota que obtuvo un estudiante de la Facultad en un curso determinado cuando realizó la matrícula en algún semestre.

Atributos	Descripción tabla Nota
Código del alumno	Cadena numérica de ocho dígitos que identifica de manera única al estudiante.
Período de matrícula	La carrera dura diez semestres académicos. Cada semestre se llama también "período académico" y se representa con cinco cifras. Las cuatro primeras corresponden al año calendario y la última puede ser 0, 1 ó 2 según el período: 0 para el período de verano enero-febrero, 1 para el período de abril-julio y 2 para el período de agosto-diciembre.
Cursos matriculados	Valor que corresponde al número de asignaturas en las que el estudiante se matriculó durante el período en que se efectuó la matrícula en la asignatura representada en el correspondiente registro.
Créditos matriculados	Cada asignatura tiene un número de créditos asignados por la Universidad. Este campo corresponde a la suma de todos los créditos de todas las asignaturas en las que el alumno se ha matriculado, inclusive la que corresponde al registro actual.
Código del curso	Cadena numérica de cuatro dígitos que identifica de manera única la asignatura.
Créditos del curso	Es el número asignado a una asignatura, que representa la medida del número de horas teóricas y prácticas semanales que esta posee.
Vez de matrícula	Número que corresponde a la cantidad de veces que se ha realizado la matrícula en la asignatura representada en el registro.
Promedio	Nota final del alumno después de haber cursado la asignatura. Rango de 0 a 20. [0, 11) Suspenso (SUSP): El estudiante deberá matricularse en el ciclo inmediato posterior en la misma asignatura. [11, 20] Aprobado (APROB): El estudiante podrá matricularse en la siguiente asignatura según su plan de estudios.
PPA con que inicia el ciclo	Es el promedio ponderado acumulado de todas las asignaturas cursadas anteriores a la matrícula.

Tabla 5.3. Atributos de la tabla Nota

En la tabla 5.4, se detallan los atributos de la tabla Plan de Estudio (667 registros).

Atributos	Descripción tabla Plan de estudio
Plan de estudios	Cada malla curricular ha sido codificada. En nuestro caso, el valor de este atributo representa el número de cambios, aumentado en una unidad, efectuado al plan de estudios original. Desde la creación de la Facultad, se han realizado ocho cambios de plan curricular; por ello, se han considerado ocho planes de estudios.
Ciclo del inicio	Representa el ciclo o período en el cual se implementó la modificación curricular.
Ciclo del fin	Representa el ciclo o período en el cual finalizó un plan curricular para iniciar el siguiente.
Código del curso	Cadena numérica de cuatro dígitos que identifica de manera única la asignatura.
Nombre del curso	Nombre de la asignatura.
Nivel del curso	Es el número que representa la ubicación de la asignatura en la malla curricular. Debido a que la carrera tiene diez semestres académicos, el nivel del curso es un número entero que identifica en qué semestre se encuentra ubicada dicha asignatura.
Carácter del curso	En el plan curricular, existen asignaturas obligatorias (O) y electivas (E).

Tabla 5.4. Atributos de la tabla Plan de Estudio

La tabla 5.5. corresponde a la tabla Equivalencia. Esta posee tres atributos y un total de 311 registros. Los atributos son los siguientes:

Atributos	Descripción tabla Equivalencia
Plan de estudios	Cada malla curricular ha sido codificada. En nuestro caso, el valor de este atributo representa el número de cambios, aumentado en una unidad, efectuado al plan de estudios original. Desde la creación de la Facultad, se han efectuado ocho cambios de plan curricular; por ello, se han considerado ocho planes de estudios.
Código actual	Representa el código de la asignatura que se encuentra vigente en la malla curricular que corresponde al plan de estudios.
Código anterior	Representa el código de la asignatura que ha sido reemplazada en el plan anterior al actual.

Tabla 5.5. Atributos de la tabla Equivalencias

La tabla 5.6 corresponde a la tabla de requisitos. Esta posee tres atributos y un total de 579 registros. Los atributos son los siguientes:

Atributos	Descripción tabla Requisitos
Plan de estudios	Cada malla curricular ha sido codificada. En nuestro caso, el valor de este atributo representa el número de cambios, aumentado en una unidad, efectuado al plan de estudios original. Desde la creación de la Facultad, se han efectuado ocho cambios de plan curricular; por ello, se han considerado ocho planes de estudios.
Código de curso	Cadena numérica de cuatro dígitos que identifica de manera única la asignatura.
Código de requisito	Representa el código de la asignatura, que es el requisito para poder efectuar la matrícula en el curso al que representa el registro actual.

Tabla 5.6. Atributos de la tabla Requisitos

5.2.2. Preparación de datos

A partir de los datos del apartado anterior, se inicia el proceso de generación de los datos finales, los cuales serán usados para entrenar el modelo y obtener predicciones adecuadas. El proceso de generación de datos se divide en cinco subprocesos:

5.2.2.1. Normalización de datos

En este subproceso, los datos se normalizan y se cargan dentro de un modelo relacional de base de datos apropiado para que estos puedan manipularse más fácilmente. Por ejemplo, la tabla Plan de Estudios que aún no está normalizada y que posee 667 registros se convierte en las tres tablas siguientes: la tabla Curso, con 244 registros; la tabla CursoMalla, con 667 registros; y, por último, la tabla Malla, con ocho registros, debido a que desde la creación de la Facultad se han registrado solo ocho

modificaciones curriculares. Las tablas mencionadas anteriormente se detallan en la tabla 5.7:

Tabla	Atributos	Proceso
Curso	Código del curso, nombre del curso, dificultad	Los dos primeros atributos se extraen directamente de la tabla original. Para calcular el tercer campo, se emplea la función que calcula la dificultad para cada una de las asignaturas.
Curso Malla	Malla del curso	Para obtener esta tabla, se extrae la asignatura junto con la correspondiente malla a la cual pertenece; para ello, se identifica en qué malla(s) ha estado cada uno de los cursos.
Malla	Código de malla, fecha de inicio de malla (implementación), fecha de caducidad.	Cada malla tiene su fecha de implementación y su fecha de caducidad; esta última depende de la fecha en que se proponga una nueva malla. El término "malla curricular" es equivalente al de plan de estudios

Tabla 5.7. Modelo normalizado del plan de estudio

5.2.2.2. Transformación de datos

En este subproceso, se construyen tres tablas adicionales a la vez que se producen datos desnormalizados para acelerar su manipulación y búsqueda. Particularmente, se trabaja con las asignaturas que son requisitos, lineales y no lineales[2], con distancia mayor o igual que la unidad. Como ya se indicó, cuando se menciona la distancia entre dos requisitos, se refiere al número de períodos que existen entre un curso y su requisito no lineal. Por ejemplo, una asignatura que dependa de otra que está ubicada en el semestre inmediato anterior tiene distancia igual a uno. En este trabajo de investigación, se hará referencia a N1 debido a que es condición indispensable para determinar el potencial por el método que se define más adelante. De este proceso, se obtienen tres tablas: dependencia lineal, con 2652 registros; equivalencia hacia delante, con 244 registros; y equivalencia hacia atrás, con 404 registros. Estas se describen en la tabla 5.8.

Tabla	Atributos	Proceso
Dependencia lineal	Código del curso, código de la malla, código del curso padre	Se identifican todas y cada una de las asignaturas pertenecientes a cada una de las mallas con todos sus respectivos requisitos (cursos padres) hasta llegar al primero.
Equivalencia atrás	Curso, curso antiguo	Para cada curso, se identifica su equivalente en mallas anteriores.
Equivalencia adelante	Curso, curso actual	Para cada curso, se identifica cuál es su equivalente en la malla más reciente donde lo tenga.

Tabla 5.8. Tablas de aceleración de proceso

[2] Se dice que un curso es requisito lineal cuando depende directamente de otra asignatura, y no lineal cuando hay una dependencia indirecta; generalmente, esto último ocurre en asignaturas que pertenecen a períodos académicos no consecutivos.

La tabla 5.9 corresponde a la tabla Dependencia lineal. Esta posee tres atributos y un total de 2652 registros. Los atributos son los siguientes:

Atributos	Descripción tabla Dependencia lineal
Curso	Código de cada una de las asignaturas que ha tenido la Facultad de Ingeniería de Sistemas de la Universidad de Lima. Una asignatura, conceptualmente, puede estar representada por más de dos registros. En este caso, se tratan de asignaturas comunes modificadas a lo largo del tiempo y que corresponden a asignaturas equivalentes.
Plan de estudios	Es un código de un dígito que corresponde al número de modificación curricular hecha en el tiempo. Hasta el momento del estudio, se han identificado ocho modificaciones curriculares.
Requisito	Cadena numérica de cuatro dígitos que identifica de manera única la asignatura padre de la representada en el primer atributo. En nuestro caso, la asignatura padre representa el requisito inmediato anterior de la asignatura hijo.

Tabla 5.9. Atributos de dependencia lineal

Cada una de las tablas descritas en la tabla 5.8 se representa conceptualmente por una relación matemática. La relación que corresponde a la tabla 5.9 se describe a continuación:

$$Dep - Lineal(curso, Plan\ de\ estudios) = SPC(conjunto\ de\ cursos\ requisitos)$$

Esta relación tiene como variables independientes (2652 combinaciones) a los cursos que ha tenido la Facultad de Ingeniería de Sistemas, y su valor corresponde a todos los requisitos de alguna asignatura considerando la malla curricular (plan de estudios) a la que perteneció.

En la tabla 5.10, se pueden observar la relaciones de dependencia lineal para la asignatura de Cálculo II (1459) que, en la malla ocho, tuvo tres requisitos: 6334, 6333, 6326, y en las mallas cinco, seis y siete tuvo también tres: 1458,1081 y 1080.

Curso	Malla	Requisito
1459	5	1080
1459	5	1081
1459	5	1458
1459	6	1080
1459	6	1081
1459	6	1458
1459	7	1080
1459	7	1081
1459	7	1458
1459	8 (del 2006-1 al 2009-1)	6326 – Matemática Básica
1459	8	6333 – Algebra Lineal
1459	8	6334 – Cálculo I

Tabla 5.10. Dependencia lineal de cálculo II

La tabla 5.11 corresponde a la tabla Equivalencia hacia atrás. Esta posee dos atributos y un total de 404 registros. Los atributos son los siguientes:

Atributos	Descripción tabla Equivalencia hacia atrás
Curso	Código de cada una de las asignaturas que ha tenido la Facultad de Ingeniería de Sistemas de la Universidad de Lima. Una asignatura, conceptualmente, puede estar representada por más de dos registros; en este caso, se trata de asignaturas comunes que han sido modificadas a lo largo del tiempo y que corresponden a asignaturas equivalentes.
Curso_Predecesor	Código de un curso equivalente en un plan de estudios anterior

Tabla 5.11. Atributos de la tabla Equivalencia hacia atrás

En su mayoría, cada asignatura de la malla curricular actual tiene su equivalente en asignaturas de mallas anteriores. Este paso, que hemos denominado "linealización hacia atrás", consiste en identificar, en una tabla, las asignaturas de la última malla, y relacionarlas con sus equivalentes en sus predecesoras. Se considera, además, que cada curso equivale a sí mismo, por lo que todas las asignaturas tendrán como mínimo un registro. Esta tabla resultó con un total de 404 registros y solamente con dos atributos, "código del curso actual" y "código del curso antiguo".
La relación que representa conceptualmente a esta tabla es:

$$Equiv - Atras(curso) = Conjunto\ de\ cursos\ antiguos$$

Esta presenta como variables independientes a los cursos que ha tenido la Facultad de Ingeniería de Sistemas, y su valor corresponde a sí mismo y a las asignaturas que han sido equivalentes a cada una de ellas en el pasado.

En la tabla 5.12, se puede observar un extracto de aquella tabla, que corresponde a las asignaturas de Cálculo I (6334) y Cálculo II (1459), valores actuales de dichas asignaturas.

Curso Cálculo I	Curso Antiguo	Curso Cálculo II	Curso Antiguo
1092	1092	1093	1093
1458	1092	1459	1093
1458	1458	1459	1459
6334	1092		
6334	1458		
6334	6334		

Tabla 5.12. Equivalencia hacia atrás para Cálculo I y Cálculo II

El proceso inverso al anterior recibe el nombre de "equivalencia hacia delante". En este caso, cada asignatura de código antiguo tendrá un equivalente en código actual.

Si la asignatura ha perdido vigencia, no podrá representarse y el registro se eliminará. La tabla final resultó con 244 registros; sus atributos son los siguientes:

Atributos	Descripción tabla Equivalencia hacia delante
Curso	Código de cada una de las asignaturas que ha tenido la Facultad de Ingeniería de Sistemas de la Universidad de Lima. Una asignatura conceptualmente puede representarse por más de dos registros; en este caso, se trata de asignaturas comunes que se han modificado a lo largo del tiempo y que corresponden a asignaturas equivalentes.
Curso actual	Es el código de la misma asignatura en mención que tiene su equivalente en la malla actual.

Tabla 5.13. Atributos de la tabla de Equivalencia hacia delante

La relación que representa conceptualmente a esta tabla es la siguiente:

$$Equiv - Adelante(curso) = curso\_Actual$$

Esta relación tiene como variable independiente a las asignaturas de la Facultad, y su valor corresponde a todo aquel curso que se creó con posterioridad (dada una modificación curricular) y que es equivalente al antiguo. El valor futuro a una asignatura vigente es el mismo para este caso.

En la tabla 5.14, se puede observar un extracto de aquella tabla, que corresponde a las asignaturas de Cálculo I (6334) y Cálculo II (1459).

Curso Cálculo I	Curso Actual	Curso Cálculo II	Curso Actual
1092	6334	1093	1459
1458	6334	1459	1459
6334	6334		

Tabla 5.14. Equivalencia hacia delante para Cálculo I y Cálculo II

5.2.2.3. Generación de datos

Así como el modelo del usuario (user Modeling) en cualquier dominio de aplicación presenta atributos que caracterizan al individuo, en nuestro caso, cada estudiante tiene un conjunto de atributos que identifican sus rasgos académicos. Uno de ellos es el atributo potencial, un número que representa la calidad académica de un estudiante en una asignatura en particular, basado en las notas que obtuvieron en las asignaturas que fueron cursadas previamente como requisito. Para calcularlo, se necesita transformar el período de matrícula en términos de malla curricular, así se identifica la malla curricular vigente cuando el estudiante hizo su matrícula. Por ejemplo, una asignatura que se lleva en el ciclo académico 1994-2 corresponde a la malla académica que va de 1993-2 a 1996-0; en nuestro caso, esa es la segunda malla curricular.

Existen excepciones a esta regla de transformación, puesto que alumnos que desaprueban un curso han podido históricamente llevar el mismo curso de mallas curriculares anteriores luego de que estas dejaron de estar vigentes.[3] En estos casos, en los cuales una transformación directa vía rango de fechas no funciona, es necesario buscar el curso en las mallas anteriores hasta encontrarlo. Para el ejemplo anterior, podría ocurrir que, si el curso no se encuentra en la segunda malla curricular, se busque en la primera, en donde se debería encontrar el equivalente. A pesar de que esta regla cubre la mayoría de los casos, existen alumnos que pertenecían anteriormente a otras Facultades y, por tanto, en su registro de notas se consignan los cursos que llevaron en estas, aunque no necesariamente son propias del plan de estudios de la Facultad de Ingeniería de Sistemas que se está analizando. Por esta razón, estos registros son eliminados luego.

Durante una fase preliminar de pruebas, se trabajó con los registros disponibles antes de que fueran normalizados, y se obtuvieron resultados que se consideraron perfectibles, por lo cual se vio conveniente generar datos que pudieran englobar las cualidades de un determinado curso y de un alumno para dicho curso. Era necesario, por ello, que el modelo relacional pudiera manejar de manera eficiente las relaciones existentes entre los cursos. Para este fin, se optó por dos métricas que se obtienen del procesamiento ejecutado sobre este modelo:

***Dificultad* [Vial-10a]:**

En el nivel del curso, se decidió que la dificultad está dada por el promedio de todos los alumnos que han cursado la asignatura. Este cálculo debe tomar en cuenta que, en algunos cambios curriculares, los cursos cambian de código a pesar de ser equivalentes, y que estas equivalencias deben involucrarse en el cálculo; es decir, si hay cuatro cursos equivalentes entre sí, se considera el promedio de sus notas como ponderado por el creditaje que tuvieron en su respectiva malla. La definición de la dificultad se detalla a continuación; su respectivo pseudocódigo se puede observar en la figura 5.5.

Así como el código de una asignatura y su nombre la caracterizan de manera definitiva en un plan de estudios, la dificultad caracteriza a la asignatura de manera temporal (por semestre académicos), ya que, al ser dependiente de las calificaciones que obtengan los alumnos semestre a semestre en cada asignatura, se va recalculando en cada período antes de que el archivo final sea usado como conjunto de entrenamiento.

[3] Esto debido a las normas transitorias que estipula cada unidad académica.

El grado de dificultad de un curso se mide como el promedio ponderado de las notas de todos los alumnos que han llevado el curso o alguno de sus equivalentes. Esto es representado por:

$$Dificultad_{\ c} = \frac{\sum_{t \in BE_c} \ \sum_{j=1}^{m_t} \ G_{j,t} * W_t}{\sum_{t \in BE_c} W_t * \ m_t}$$

Donde:

c: Curso Actual
t: Curso equivalente al actual
BE_c : Conjunto de cursos equivalentes en el pasado al curso c
$\mathrm{m_t}$: Total de alumnos en el curso t
$G_{j,t}$: Nota del alumno j en el curso t
W_t: Cantidad del créditos del curso t

A continuación, mostraremos un ejemplo para el cálculo de dificultad del curso de Algebra Lineal. Este tuvo en el pasado un equivalente llamado Matemática Básica II, el cual poseía un creditaje distinto al actual. Así, la dificultad de este curso para el ciclo 2006-2 es calculada de la siguiente manera:

ciclo	Nombre del curso	Estudiante	Nota	Créditos (depende de la malla)
2005-2	Matemática Básica II	Estudiante 1	12	4
		Estudiante 2	10	4
		Estudiante 3	14	4
2006-1	Álgebra Lineal	Estudiante 2	13	2
		Estudiante 4	14	2
		Estudiante 5	10	2

Tabla 5.15. Notas usadas para el ejemplo de cálculo de la dificultad

$$Difficulty_{Alg\ Lineal} = \frac{12\ x\ 4 + 10\ x\ 4 + 14\ x\ 4 + 13\ x\ 2 + 14\ x\ 2 + 10\ x\ 2}{4\ x\ 3 + 2\ x\ 3} = 12.11$$

Como se puede observar, para el cálculo de la dificultad, se utilizan las notas de los alumnos que cursaron una o varias veces esta asignatura o sus equivalentes en el pasado (por ejemplo, el estudiante 2).

Cabe destacar que este atributo debe ser recalculado en cada proceso de matrícula para considerar las notas de los nuevos estudiantes que cursan la asignatura. Además, como los valores de este atributo están relacionados con la nota promedio que los alumnos obtienen en dicho curso, mientras más bajo sea el valor obtenido, significará mayor complejidad de la asignatura.

Pseudocódigo de Dificultad

```
1 For Each Course
2    If Course == Course.CurrentVersion Then
3       Difficulty = NULL
4       numerator = 0
5       denominator = 0
6       For Each Grade
7          If Course.EquivalentCourses Contains Grade.Course Then
8             numerator = numerator + (Grade.FinalGrade x Grade.Course.Credits)
9             denominator = denominator + (Grade.Course.Credits)
10         End If
11      End For
12      Difficulty = numerator / denominator
13      For Each Course in Course.EquivalentCourses
14        Course.Difficulty = Difficulty
15      End For
16   End If
17 End For
```

Figura 5.5. Pseudocódigo de dificultad

***Potencial* [Vial-10a]:**

El potencial es la caracterización que identifica al alumno en cada asignatura que le corresponda cursar. Un mismo estudiante puede tener un potencial diferente en dos asignaturas distintas, ya que, en caso de que ellas dependan de requisitos distintos, se calcularán de diferente manera. Por otro lado, un estudiante podrá tener potenciales distintos en una misma asignatura según la vez que la curse.

El término potencial surge de la necesidad de tener un atributo que incluya el rendimiento académico de un determinado estudiante por áreas. No es lo mismo que se calcule la predicción de aprobar o reprobar una asignatura de Gestión de la Cadena de Suministro que una de Cálculo 3. La ausencia de este atributo dejaría en manos del promedio ponderado acumulado y de la dificultad de la asignatura la decisión para el clasificador. Sin embargo, considerando esta variable adicional, se está tomando en cuenta el rendimiento del estudiante en asignaturas que han sido requisitos lineales y no lineales de la asignatura en consulta.

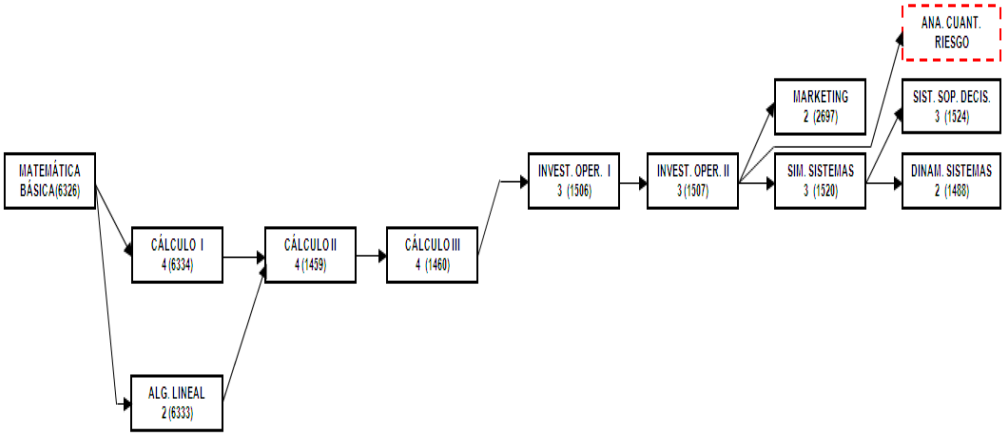

Figura 5.6. Subred curricular correspondiente al área de operaciones

En la figura 5.6, se ilustra un grafo de precedencia. Este concepto es importante para el término potencial debido a que es mediante este grafo que se calcula el potencial para un registro de la base de datos. Por ejemplo, para calcular el potencial de Cálculo III, se deben considerar las notas y las dificultades obtenidas por el estudiante y por el sistema, respectivamente, en las cuatro asignaturas que tiene como prerrequisito, según sea la modalidad del cálculo de potencial. Esta modalidad está relacionada con la distancia que se debe considerar para dicho cálculo.

El potencial representa la habilidad académica de un estudiante en un curso en particular, basado en las notas que obtuvo en los requisitos. Este es calculado como un promedio ponderado de las notas que ha obtenido un alumno en los cursos requisitos. Este promedio se ajusta con la dificultad. Además de los cursos que son requisitos inmediatos, también se considera para el cálculo aquellos que se encuentren a una distancia N del curso en cuestión. Durante la fase de experimentación, se consideraron cuatro valores distintos para el cálculo del potencial y su respectivas variantes(tratamientos) se pueden ver en la tabla 5.16.

Tratamiento	Descripción
Potencial NT	Esta metodología consiste en realizar el cálculo del potencial utilizando las asignaturas que sean pre requisitos de la asignatura en consulta.
Potencial PPA	Esta metodología es igual a la anterior, pero, para calcular el potencial, se toman todos los cursos que el alumno ha llevado hasta el momento.
Potencial N1	Para esta prueba, el potencial se calcula bajo la premisa de que las dependencias de un curso constan de sus requisitos inmediatos.
Potencial N2	Para esta prueba, el potencial se calcula bajo la premisa de que las dependencias de un curso tienen dos niveles de requisitos.

Tabla 5.16. Tratamientos para el cálculo del potencial

El potencial es representado por:

$$potential_{s,c,d} = \frac{\sum_{t \in SPC_{c,d}} (\frac{\sum_{v=1}^{H_t} G_{s,t,v} * W_t}{D_t})}{\sum_{t \in SPC_{c,d}} W_t * H_t}$$

Donde:

- s: Alumno
- c: Curso Actual (el que se va predecir)
- d: Distancia para el cálculo de potencial
- t: Curso requisito
- $SPC_{c,d}$: Conjunto de requisitos del curso c a una distancia d
- H_t: Número máximo de notas al cursar la asignatura t
- $G_{s,t,v}$: Nota del alumno s en el curso t y en el intento v
- W_t : Cantidad del créditos del curso t
- D_t : Dificultad del curso t

A continuación, tomando en cuenta la tabla 5.17, se procederá a calcular el potencial para el curso de Cálculo III para el estudiante 2 utilizando como referencia el grafo de dependencias de la Figura 5.2 para determinar los cursos a utilizar.

CURSO	DISTANCIA	CRÉDITOS	DIFICULTAD	NOTA	VEZ
CÁLCULO II	1	4	10.88	12	1
CÁLCULO I	2	4	10.83	15	1
ÁLGEBRA LINEAL	2	2	10.08	13	2
MATEMÁTICA BÁSICA II	2	4	10.08	10	1
MATEMÁTICA BÁSICA	3	4	10.87	16	1
LENGUAJE I	-	4	11.53	11	1

Tabla 5.17. Notas usadas para el ejemplo de cálculo de potencial

$$POTENCIAL\ N1 = \frac{\frac{12\ x\ 4}{10.88}}{4} = 1.1$$

$$Pot.N2 = \frac{\frac{12\ x\ 4}{10.88} + \frac{15\ x\ 4}{10.83} + \frac{13\ x\ 2}{10.08} + \frac{9\ x\ 2}{10.08} + \frac{10\ x\ 4}{10.08}}{4\ x\ 1\ +\ 4\ x\ 1\ +\ 2\ x\ 2 + 4\ x\ 1} = 1.14$$

$$Pot.NT = \frac{\frac{12\ x\ 4}{10.88} + \frac{15\ x\ 4}{10.83} + \frac{13\ x\ 2}{10.08} + \frac{9\ x\ 2}{10.08} + \frac{10\ x\ 4}{10.08} + \frac{16x\ 4}{10.87}}{4\ x\ 1\ +\ 4\ x\ 1\ +\ 2\ x\ 2 + 4\ x\ 1\ + 4\ x\ 1} = 1.21$$

$$Pot.PPA = \frac{\frac{12\ x\ 4}{10.88} + \frac{15\ x\ 4}{10.83} + \frac{13\ x\ 2}{10.08} + \frac{9\ x\ 2}{10.08} + \frac{10\ x\ 4}{10.08} + \frac{16x\ 4}{10.87} + \frac{11\ x\ 4}{11.53}}{4\ x\ 1 +\ 4\ x\ 1\ +\ 2\ x\ 2 + 4\ x\ 1\ + 4\ x\ 2 + 4\ x\ 1} = 1.17$$

En la tabla 5.17, se puede observar las notas del estudiante 2 en los cursos requisito de Cálculo III, tanto en cursos actuales como en cursos que fueron

reemplazados (Matemática Básica II sustituido por Álgebra Lineal). También se observa que la dificultad de Álgebra Lineal es la misma que Matemática Básica II por ser cursos equivalentes.

En este caso, se utiliza la fórmula de SPC para determinar los distintos conjuntos de cursos requisitos para el cálculo del potencial dependiendo de la distancia. Dicha función retorna el conjunto de requisitos de una asignatura que haya sido cursada por el estudiante, considerando las equivalencias entre cursos y los diferentes cambios curriculares.

Luego, se procede a realizar el promedio ponderado usando las notas del estudiante en los cursos requisitos tantas veces haya intentado cursar la asignatura requisito actual o la equivalente en el pasado con su respectivo creditaje para luego ser dividido por la dificultad del curso respectivo. De acuerdo con la fórmula, los valores de potencial más altos representan que un alumno se va a desenvolver óptimamente en el curso a predecir.

En caso de no poder realizar el cálculo del potencial de un alumno en una asignatura determinada, se procederá a tomar el potencial PPA, el cual no es más que la división ponderada de la división de la nota obtenida en un curso entre su dificultad; este utiliza todas las asignaturas que el alumno haya cursado hasta el momento de la consulta. En el ejemplo Potencial PPA, se observa que el curso de Lenguaje I ha sido agregado junto con las demás asignaturas del área de ciencias básicas para obtener un resultado final.

El respectivo pseudocódigo que corresponde al cálculo del potencial se muestra en la figura 5.6.

Potential Pseudo code

```
1 For Each Grade1
2    Potential = NULL
3    numerator = 0
4    denominator  = 0
5    For Each Grade2
6       If (Grade2.Student == Grade1.Student)
7          And (Grade2.Semester < Grade1.Semester)
8          And (Grade1.Course.EquivalentCourses Contains Grade2.Course
9             Or Grade1.Course.Prerequisites Contains Grade2.Course) Then
10       numerator = numerator + Grade2.FinalGrade / Grade2.Course.Difficulty *
Grade2.Course.Credits
11       denominator = denominator + Grade2.Course.Credits
12    End For
13    If denominator == 0 Then
14       For Each Grade2
15          If (Grade2.Student == Grade1.Student)
16             And (Grade2.Semester < Grade1.Semester) Then
17          numerator = numerator + Grade2.FinalGrade / Grade2.Course.Difficulty *
Grade2.Course.Credits
18          denominator = denominator + Grade2.Course.Credits
19       End For
20    End If
21    Potential = numerator / denominator
22    Grade1.Potential = Potential
23 End For
```

Figura 5.7. Pseudocódigo de potencial

En la tabla 5.18, se muestra un extracto de ocho registros correspondientes al desempeño de un estudiante que llevó el curso 6326, luego el curso 6333 por tres veces y 6334 por tres veces, y aprobó estos dos últimos en la tercera vez. Se puede notar que cada registro presenta un potencial distinto debido a que este se calcula cada vez que el estudiante cursa la asignatura que está representada en dicho registro. Este potencial depende únicamente de los cursos que pertenecen al gráfico de dependencias.

A	B	C	D	E	F	G	H	I	J	K	L
alumno	malla	curso	vez	ciclo	promedio	promedio De Inicio	potencial	creditaje	tipo	cursos Matriculados	creditos Matriculados
20071083	8	6326	1	20071	4			4	3	7	20
20071083	8	6326	2	20072	11	9.75	0.36790928	4	1	6	19
20071083	8	6333	1	20081	2	9.5121	0.6898299	2	1	7	20
20071083	8	6333	2	20082	5	7.4262	0.591553121	2	1	5	15
20071083	8	6333	3	20091	11	7.5263	0.575646769	2	1	4	11
20071083	8	6334	1	20081	3	9.5121	0.6898299	4	1	7	20
20071083	8	6334	2	20082	1	7.4262	0.552070898	4	1	5	15
20071083	8	6334	3	20091	11	7.5263	0.437099248	4	1	4	11

Tabla 5.18. Representación final de datos

Es evidente que el registro nueve equivaldría a que el estudiante cursase la asignatura de Cálculo 2. Dicho registro también tendría un potencial asignado calculado por la ecuación del potencial.

Se denomina grafo de dependencias al que está formado por todos los caminos de la subred, desde el nodo inicial hasta el nodo que representa el curso en consulta. En la figura 5.8, se muestra el gráfico de dependencia para la asignatura 1459 para un estudiante. Cada nodo del gráfico puede representar uno o más registros en el conjunto de entrenamiento final, lo que depende de la cantidad de veces que dicho estudiante ha cursado la materia. Este gráfico de dependencia es un subgrafo que pertenece al de las asignaturas del área correspondiente a soporte de decisiones.

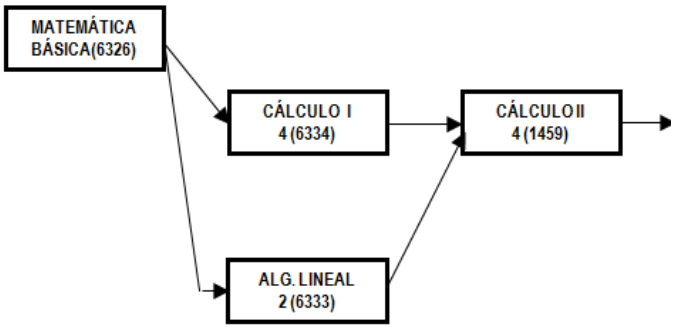

Figura 5.8. Subgrafo para la subred del área de operaciones

Existe la posibilidad de que el potencial dependa sólo de la asignatura requisito inmediata anterior; en este caso, se llama potencial con distancia euclidiana n=1 y lo indicaremos, a partir de ahora, potencial N1, tal como se mencionó anteriormente. Como una generalización, se definirá el gráfico de dependencia de un curso $c \in C$ como el par (C, E), donde C es el conjunto de todos los nodos, que en nuestros caso representan cursos, y E son los arcos que unen dos nodos distintos y cuyo significado es la relación de dependencia padre – hijo, es decir, la relación de una asignatura con su respectivo requisito.

El grafo de la figura 5.6 muestra, a su vez, un sub grafo del gráfico total de asignaturas de la Facultad o del plan curricular (llamada malla curricular) completa de asignaturas electivas y no electivas de primero a décimo nivel. Es importante destacar que este subgrafo puede contener, a su vez, otros según la asignatura por la que se consulta. Por ejemplo, si el curso por el que se consulta fuese el 1459, como en el caso de la figura 5.8, el subgrafo resultante constaría de cuatro nodos y un camino no lineal. Se dice que un camino es lineal si existe dependencia unívoca de un requisito con su respectivo curso; en nuestro ejemplo, el camino es no lineal, ya que, para alcanzar el nodo que corresponde al curso de Cálculo II, se pudo pasar por dos caminos. Estos dos

nodos intermedios, en el contexto del cálculo del potencial, se consideran como si fueran únicos.

Como se mencionó anteriormente, para efectos de la experimentación que se desarrollará en el próximo capítulo, se consideran cuatro metodologías para el cálculo del potencial. Esto significa que se experimentará con cuatro conjuntos disjuntos que contienen los mismos datos, siendo la única diferencia el atributo que corresponde al potencial. Los cuatro conjuntos presentan los mismos atributos, como se puede observar en la tabla 5.19:

Atributos	Tipo de atributo
Nombre de asignatura	Cadena
Vez	Número discreto
Promedio inicio	Número Continuo(PPA)
Potencial	Número Continuo
Creditaje	Número Continuo
Créditos matriculados	Número Continuo
Dificultad	Número Continuo
Nota (clase)	Aprobado/Suspenso(desaprobado)

Tabla 5.19. Atributos considerados para el algoritmo de aprendizaje

5.2.2.4. Filtrado y limpieza de datos

En todo proceso de minería de datos, la limpieza de los datos para eliminar ítems irrelevantes es de mucha importancia. El descubrimiento de patrones será útil sólo si los datos en el conjunto de entrenamiento proporcionan una representación real del rendimiento académico y de las acciones y/o decisiones que ha tomado el estudiante en el pasado.

Inicialmente, la Universidad proporcionó una base de 250843 registros correspondientes a 5938 alumnos. En la tabla 5.20, se describe detalladamente cada uno de los pasos realizados para filtrar los datos.

Paso	Descripción	Justificación	No. de registros eliminados	No. de registros restantes
1	Eliminar los registros correspondientes a asignaturas que alumnos de la Facultad en estudio llevaron en otras Facultades.	La Facultad en la cual se enfoca el presente estudio es la de Ingeniería de Sistemas, por lo cual estos registros no pueden tomarse en cuenta.	5327	250363
2	Eliminar los registros para los cuales no se puede calcular el potencial.	Para este grupo, no existen registros previos al ciclo que le corresponde y, por tanto, no puede realizarse ninguna predicción al respecto.	37575	212788
3	Eliminar todos los registros que no se pueden traducir al plan de estudios actual.	Estos registros no pueden incluirse, porque los registros no tienen equivalentes en el currículo actual.	32647	180141
4	Eliminar los registros de los semestres de verano.	La Universidad ofrece ciclos no regulares durante el período de enero a marzo; no obstante, en estos ciclos los alumnos generalmente llevan menos cursos, 1 ó 2, pero tienen una presión adicional, dado que el tiempo es más corto, por lo cual estos registros no pueden compararse contra la masa de registros de ciclos regulares.	18894	161247

Tabla 5.20. Pasos considerados para el filtrado y limpieza de datos

5.2.2.5. Selección de atributos

El principal objetivo de nuestra investigación es el descubrimiento de patrones que servirán para dar recomendaciones positivas o negativas respecto a la matrícula de un estudiante en determinada asignatura, según las notas que obtuvieron otros estudiantes con rendimientos académicos similares. En este sentido, y como se conoce la función que cumple cada atributo y las relaciones implícitas que existen entre ellos, se consideró que el aprendizaje automático debía realizarse con los atributos detallados en la tabla 5.21.

Atributos	Justificación de la elección del atributo
Nombre de asignatura	Es deseable que el sistema de aprendizaje construya reglas o haga una clasificación que ayude al estudiante a tomar una buena decisión sobre su matrícula en cada asignatura. En nuestro caso, el nombre de las asignaturas será el identificador que las relacione con la recomendación positiva o negativa.
Vez que cursa la asignatura	En la Universidad de Lima, las asignaturas sólo se pueden cursar tres veces; esto quiere decir que el estudiante que suspende una asignatura tiene solo un par de oportunidades adicionales para aprobarlas. En caso de que esto no ocurra, dicho estudiante es separado de la carrera a la que pertenece dicha asignatura.
PPA con que inicia el ciclo	Tratándose de un sistema de recomendación colaborativo, una variable importante para la clasificación es el promedio ponderado acumulado con que inicia el período, debido a que este atributo tiene la función de distinguir a los estudiantes según su rendimiento académico en general.
Potencial	Aptitud o habilidad que posiblemente tiene el alumno para este curso.
Créditos de la asignatura	Corresponden al creditaje de la asignatura matriculada.
Créditos matriculados	Muchas veces el rendimiento del estudiante se ve afectado por la cantidad de asignaturas en que se matricula. Tal decisión se debe a la falta de experiencia del estudiante o a la falta de información cualitativa referida al rendimiento de otros alumnos que se matricularon en el mismo número de asignaturas. Una medida más exacta de esta dificultad es el creditaje, indicativo del número de horas del curso.
Dificultad	Indica la dificultad de la asignatura en base al promedio de alumnos que la han llevado.
Nota	Representa la variable de clasificación. Por su condición, fue discretizada de la siguiente forma: [0, 11) Suspenso (SUSP): Significa que el estudiante deberá matricularse en el ciclo inmediato posterior en la misma asignatura. [11, 20] Aprobado (APROB): Significa que el estudiante podrá matricularse en la siguiente asignatura según su plan de estudios.

Tabla 5.21. Atributos finales

5.2.2.6. Datos resultantes

Finalmente, se obtuvo una tabla de 161247 registros que representa las notas obtenidas por cada alumno que se matriculó en determinada asignatura por primera, segunda y tercera vez. Las columnas de esta matriz representan los atributos mencionados en la tabla 5.19.

Cabe mencionar que los sistemas de recomendación, en general, trabajan con una matriz cuyos registros representan a los usuarios (estudiantes) y cuyas columnas representan ítems que, en nuestro caso, son las características de cada materia. Al aplicar alguna de las técnicas de aprendizaje a esta distribución de datos, tendremos como resultado relaciones entre asignaturas, debido a que es una matriz cuyos atributos son los nombres o códigos de las asignaturas, lo que podría ser, eventualmente, una información adicional para la recomendación.

En la figura 5.9, se observa el modelo de datos que engloba las fases de pre-procesamiento y limpieza. La capa normalizada muestra las tablas generadas luego de normalizar los datos originales. La capa agregada agrupa las tablas creadas por conveniencia para agilizar las consultas. La capa de cursos traducidos muestra una fase

preliminar de filtrado en la cual se han eliminado los registros que no pueden convertirse a la malla actual, mientras que los registros de la vista final contienen solo los datos que van a usarse como data de entrenamiento. Para este fin, se debe realizar una selección de los atributos disponibles.

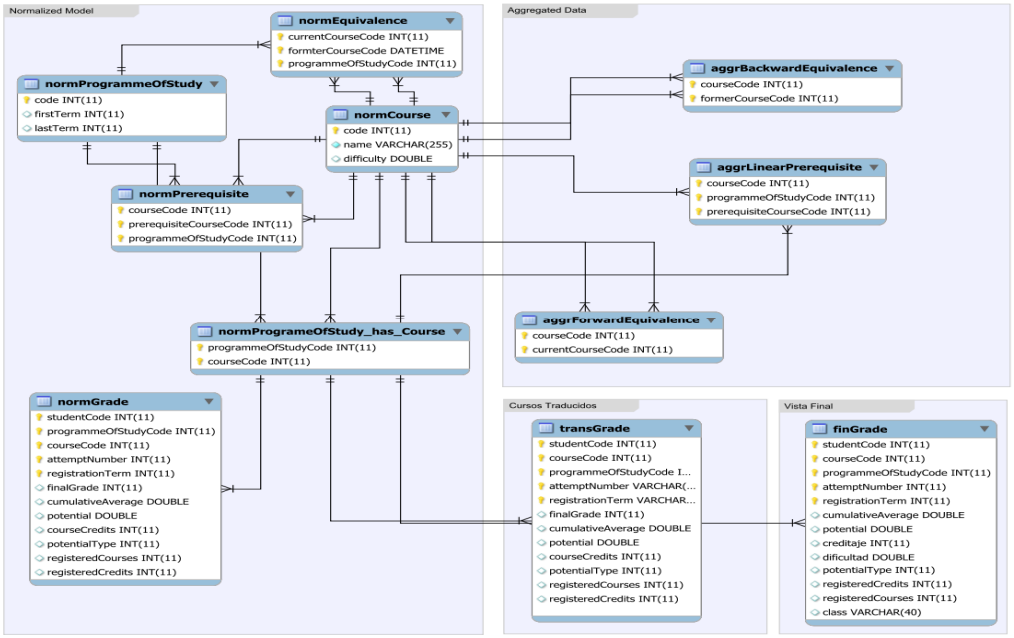

Figura 5.9. Diagrama del modelo normalizado

5.3. Arquitectura del sistema de consulta

Debido a que el proceso de KDD involucra la parte de la minería de datos y la difusión y uso, en la figura 5.10. podemos apreciar la arquitectura del proceso completo [Vial-10b].

Este proceso trabaja con un Servicio Web de Consulta que se integra a la aplicación de matrícula de la Universidad de Lima. Para responder efectivamente a las peticiones de la aplicación de matrícula, el servicio web interactúa contra la base de datos procesada y contra el ejecutable consult.exe del respectivo algoritmo de aprendizaje utilizado sobre la base de datos que se prepara en los procesos precedentes.

Para poder explicar mejor el diagrama, se han numerado los pasos que interconectan los componentes de este subproceso.

Primer paso: El alumno se autentifica contra la aplicación de matrícula (1) ingresando su usuario y su contraseña. La aplicación de matrícula muestra al alumno su promedio ponderado acumulado, el listado de cursos a los que puede matricularse (cursos hábiles) y la vez de matrícula correspondiente a cada uno de ellos.

Segundo paso: Cuando el alumno selecciona un curso, la aplicación de matrícula automáticamente invoca al Servicio Web de Consulta (2) enviándole como datos el código del alumno, los cursos que el estudiante seleccionó, la cantidad total de créditos que corresponden a los cursos elegidos (la sumatoria de créditos de los cursos seleccionados), la vez de matrícula en cada uno de los cursos y el promedio ponderado acumulado.

Tercer paso: El Servicio Web de Consulta, en ese momento, deberá obtener la dificultad del curso y el potencial del alumno para dicho curso, por lo que consultará la Base de Datos Procesada (3).

Cuarto paso: El servicio Web de consultas recibe la información solicitada de la base de datos que ha sido previamente procesada(4).

Quinto paso: Con esta información, el Servicio Web de Consulta crea un nuevo grupo de datos (5), el cual servirá para realizar una llamada al consult.exe. Los datos enviados son el promedio ponderado acumulado, el curso, la vez de matrícula en dicho curso, el número total de créditos, la dificultad y el potencial.

Sexto paso: El ejecutable consult.exe conocerá la respuesta a tal petición (6) realizando una búsqueda sobre los archivos .names y .tree.

La respuesta obtenida es la clase (aprobado o desaprobado), además del factor y de un intervalo de confianza. Esta respuesta es enviada desde el consult.exe hasta el usuario a través de los pasos (7), (8), (9) y (10).

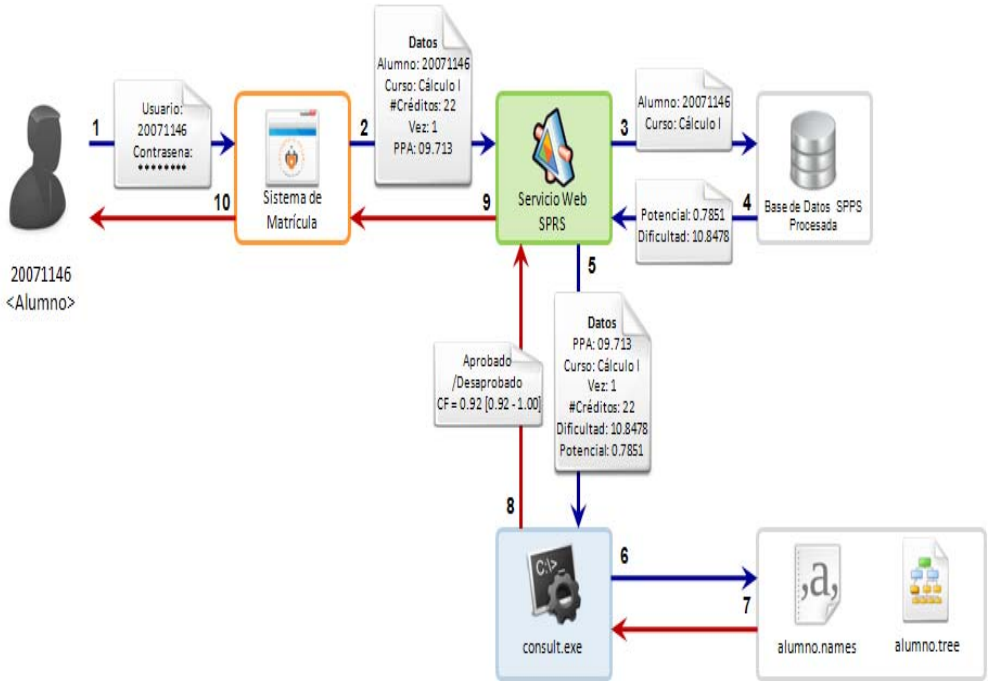

Figura 5.10. Arquitectura del sistema de recomendación propuesto [Vial-10b]

El sistema está conformado por tres subsistemas:

- Consulta Dinámica
- Student Performance Recommendation System(SPRS)
- Aplicación de Matrícula de la Universidad de Lima

5.3.1. Consulta dinámica

La consulta dinámica está conformada por tres elementos:

- **Componente de Acceso al Ejecutable de Consola.**
  Este componente, al cual hemos denominado ConsultaDinamicaDL, se encarga de realizar la llamada asíncrona al ejecutable de consola consult.exe, para controlar las lecturas (solicitudes) de los valores de los atributos del modelo analizado, y luego obtener el resultado de la predicción.
- **Ejecutable de Consola consult.exe.** Este ejecutable utiliza el archivo binario que almacena el algoritmo utilizado y ofrece una interfaz de línea de comando (CLI) para realizar consultas.
- **Archivos de datos del algoritmo.** Los archivos de datos utilizados por el consult.exe son el archivo alumno.names y el archivo alumno.model.

5.3.2. Student Performance Recommendation System (SPRS)

El SPRS comprende cinco capas [Vial-10a]:

- **ULima.SPRS.DL.SA.** Esta capa tiene el método llamado ObtenerResultado que se encarga de encapsular la lógica de la operación a ejecutar contenida en el componente ConsultaDinamicaDL, para poder utilizarlo de una forma más simple.

- **ULima.SPRS.DL.DALC.** Esta capa de acceso a datos, a través de sus métodos ObtenerPotencial y ObtenerDificultad, se encarga de obtener el potencial y la dificultad de cada curso que el alumno ha seleccionado al momento de matricularse. Esta información será utilizada para hacer la consulta en la capa ULima.SPRS.DL.SI. El potencial y la dificultad de cada instancia han sido previamente calculados MySql.

- **ULima.SPRS.BL.BE.** Esta capa contiene la descripción de las entidades de negocio utilizadas para transferir la información a través de toda la aplicación, desde el acceso a datos hasta su exposición en el servicio web.

- **ULima.SPRS.BL.BC** Esta capa negocio se encarga de invocar en un solo método (ObtenerResultado) tres funciones. Dos de ellas (ObtenerPotencial y

- ObtenerDificultad) implementadas en la capa Ulima.SPRS.DL.DALC (Data Access Layer Component) y la última (Ejecutar) implementada en la capa ULima.SPRS.DL.SA, que es el Service Agent.

- **Web Service wsSPRS.** Este Servicio Web simplemente expone la funcionalidad del método implementado en la capa Bussiness Component Layer (ULima.SPRS.BL.BC).

En la figura 5.11 se muestra la interfaz de usuario del sistema SPRS[Vial-10a]

Figura 5.11. Interfaz de usuario del sistema SPRS [Vial-10b]

5.3.3. Aplicación de Matrícula de la Universidad de Lima

La aplicación de matrícula de la Universidad de Lima está conformada por cuatro componentes:

Capa de negocio:

- **Capa de acceso de datos (edu.ulima.gama.dao)**: Esta capa de acceso a datos se encarga de obtener toda la información necesaria para que los alumnos puedan matricularse, es decir, los cursos hábiles, sus secciones, los profesores de cada curso y los horarios de cada sección. La base de datos donde se encuentra dicha información es un IBM DB2 v.

- **Business Entities:** Esta capa contiene la descripción de las entidades de negocio utilizadas para transferir la información a través de toda la aplicación, desde el acceso a datos hasta su visualización en la interfaz de usuario.
- **Business Component:** Esta capa contiene las operaciones que implementan a las reglas de negocio de la matrícula a través de diversas llamadas a la capa edu.ulima.gama.dao, como, por ejemplo, mostrar los cursos en que el alumno puede matricularse en base a las normas descritas por la malla curricular vigente.

5.3.4. Capa de Presentación

J2EE web Aplication: Esta aplicación es la interfaz de usuario de la aplicación. Para efectos de realizar el proceso de matrícula, llama a las operaciones implementadas en la capa Business Layer, y, para realizar recomendaciones sobre el rendimiento académico del alumno, consume el servicio Web del SPRS.

En la figura 5.12, se puede observar la arquitectura del sistema SPRS integrado al sistema de matrícula de la Universidad.

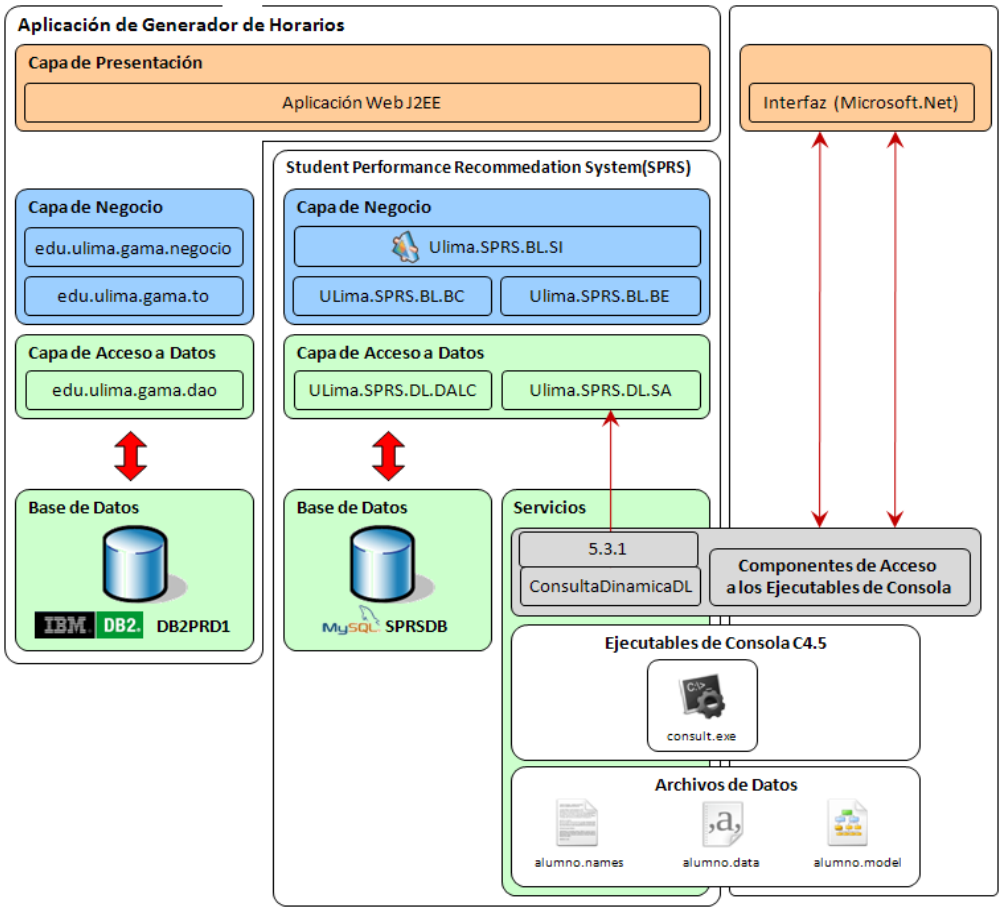

Figura 5.12. Sistema SPRS integrado al sistema de matrícula

Capítulo 6:

Experimentación y Evaluación

Teniendo en consideración que el objetivo principal de este trabajo de investigación es entregarle al estudiante una recomendación adecuada referida al éxito o fracaso que podría tener si efectuara su matrícula en una o varias asignaturas, este capítulo está dedicado a la fase de experimentación. Para efectos de aplicar las diferentes técnicas de descubrimiento del conocimiento desarrolladas en la presente memoria, se ha hecho uso de la base de datos obtenida con la metodología propuesta en el capítulo correspondiente a la preparación de datos.

Se sabe que la predicción del éxito o el fracaso del estudiante en una asignatura se determinan mediante algún clasificador. El resultado de la clasificación, aplicando diferentes métodos, genera la recomendación a partir del análisis de un conjunto de datos históricos relativos al rendimiento académico que otros alumnos han obtenido en la misma asignatura durante períodos académicos anteriores.

Para ello, se realizaron diversas pruebas con el conjunto de datos a fin de obtener modelos clasificadores que permitirán optimizar las predicciones sobre el resultado de los estudiantes. En estas pruebas, se tuvieron en cuenta dos diferentes factores:

1. La técnica utilizada para aprender el modelo debía considerar que, en situación de uso real, las condiciones podrían cambiar año a año y el clasificador podría reflejar patrones desactualizados. Por ejemplo, las recomendaciones que se realicen en el primer semestre del año académico 2009 se harán utilizando, en el mejor caso, datos históricos hasta el 2008.

2. Como cualquier clasificador aprendido, se espera que tenga un porcentaje de error en las predicciones. Sin embargo, para el contexto de uso, no todos los errores son iguales. En particular, partiendo del conocimiento de los especialistas (es decir, el criterio de la universidad), sería más grave equivocarse recomendando a un estudiante que se matricule en un curso que a la postre no aprobará, que recomendarle no matricularse en un curso que podía haber aprobado. Esto podría determinar que el porcentaje de error o la llamada precisión pueda ser relativamente útil en nuestro caso, debido a que los errores no tienen el mismo costo.

Estos criterios orientaron la configuración de las pruebas realizadas para determinar los patrones capaces de conducir el proceso a una correcta predicción del rendimiento académico.

Para los experimentos, se han considerado dos grandes fases. La primera corresponde a la que usa las técnicas tradicionales de los sistemas de recomendación basados en filtrado colaborativo. Un extenso desarrollo ha sido considerado en el capítulo 3 de la presente memoria.

6.1. Experimentación que usa filtrado colaborativo basado en memoria

En principio, para entender bien el proceso de la experimentación que usa filtrado colaborativo basado en memoria, nos ayudaremos de la figura 6.1 para tener en cuenta que en esta primera fase de la experimentación trabajaremos única y exclusivamente con las notas obtenidas por los estudiantes en cada una de las asignaturas. En la Tabla 6.1, se presenta un extracto de la tabla de datos alumno-asignatura que es similar a la matriz de usuarios-ítems, presente en cualquier sistema de filtrado colaborativo clásico. En este caso, los usuarios se representaran por los alumnos; y los ítems, por los cursos.

Alum./Asig.	Asig_1	Asig_2	Asig_3	Asig_4	Asig_5
1001	18		12	12	0
1002		17	13	17	16
1003		17	14	13	17
1004	16	18	13	12	11
1005	16	16	14	18	9
1006	14	17	15	18	10
1007			14	12	9
1008	18	18	13	12	

Tabla 6.1. Extracto de la matriz de notas de Alumno-asignatura

En la matriz de la figura 6.1, se observa que existen espacios en blanco. Los espacios en blanco en la matriz alumno-asignatura representan a estudiantes que aún no han cursado dichas asignaturas, y los espacios que presentan un número cero, a los estudiantes que, habiéndose matriculado en las asignaturas, procedieron a retirarse de ellas antes de culminar el período académico para el cual se matricularon.

A continuación, se procede a calcular la similitud entre asignaturas. Debido a que existen varias formas de hacerlo y que estamos en un dominio de aplicación específico,

en la experimentación, se usarán las diferentes formas de obtener la similitud. Luego de calcular la similitud, se procederá a calcular la predicción.

Si, por ejemplo, queremos encontrar la similitud entre las dos primeras asignaturas presentes en la tabla 6.1, en primer lugar, se extrae sólo las notas que son covaloradas y se obtiene la tabla 6.2; luego se usa la expresión correspondiente a la ecuación 3.18, 3.19 o 3.20, según si se quiere calcular similitud por coseno, o usando la expresión correspondiente a la correlación o al coseno ajustado para obtener los valores que corresponden a la similitud entre la asignatura 1 y la asignatura 2.

Alumno	Asignatura_1	Asignatura-2
1004	16	18
1005	16	16
1006	14	17
1008	18	18

Tabla 6.2. Extracto de la figura 6.1 que corresponde a las asignaturas covaloradas y sólo de las asignaturas cuya similitud quiere encontrarse

Así, al calcular la similitud (basada en cosenos) entre todas las asignaturas de la figura 6.2, se obtiene la tabla de la figura 6.3.

Ci/Cj	Asig_1	Asig_2	Asig_3	Asig_4	Asig_5
Asig_1	1	0.99677022	0.98684494	0.96185866	0.8239759
Asig_2	0.99677022	1	0.89483848	0.97757604	0.96922309
Asig_3	0.98684494	0.89483848	1	0.9871153	0.90399241
Asig_4	0.96185866	0.97757604	0.9871153	1	0.89982138
Asig_5	0.8239759	0.96922309	0.90399241	0.89982138	1

Tabla 6.3. Tabla de similitudes entre todas las asignaturas presentes en la figura 6.1

La experimentación tuvo como objetivo final reconocer la mejor forma de calcular la similitud y, por ende, la mejor forma de calcular la predicción. Así, se identifica como la mejor forma a aquella que genera menor error de predicción.

En el anexo A.2., se puede observar que la manera en que la predicción genere menor error de prueba y obtenga altos índices de especificidad y sensibilidad es aquella que usa la correlación de Pearson para el cálculo de la similitud, y la suma ponderada de otras notas (todas las notas del estudiante) para la predicción.

Los resultados obtenidos después de la ejecución del filtrado colaborativo se pueden visualizar en las siguientes figuras:

En las figuras 6.1, 6.2, 6.3 se puede observar que el método del cálculo de la similitud influye fuertemente en la predicción de las notas. Así mismo, en la figura 6.1., se observa que el mejor MAE (el menor de todos), para la predicción que usa la suma ponderada de todas las evaluaciones del estudiante, corresponde a la que utilizó la correlación de Pearson (Ecuación 3.22) para el cálculo de la similitud.

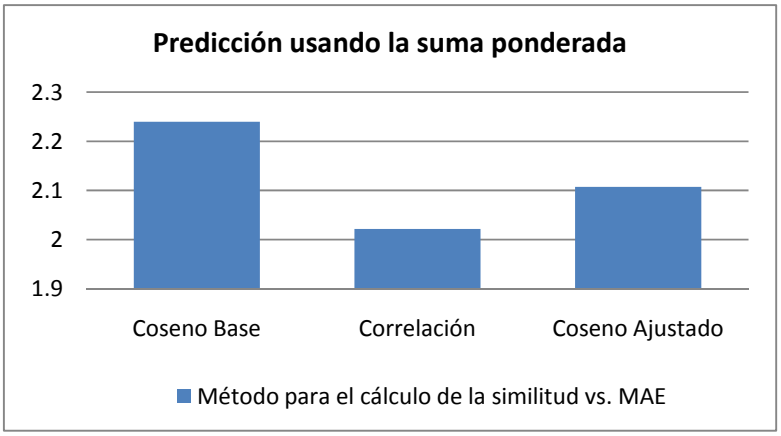

Figura 6.1. Predicción que utiliza la suma ponderada de Otros

Así mismo, en la figura 6.2., se observa que el mejor MAE para la predicción que usa la suma ponderada simple de las evaluaciones de los estudiantes (Ecuación 3.21) es el que corresponde al cálculo de la similitud por correlación de Pearson.

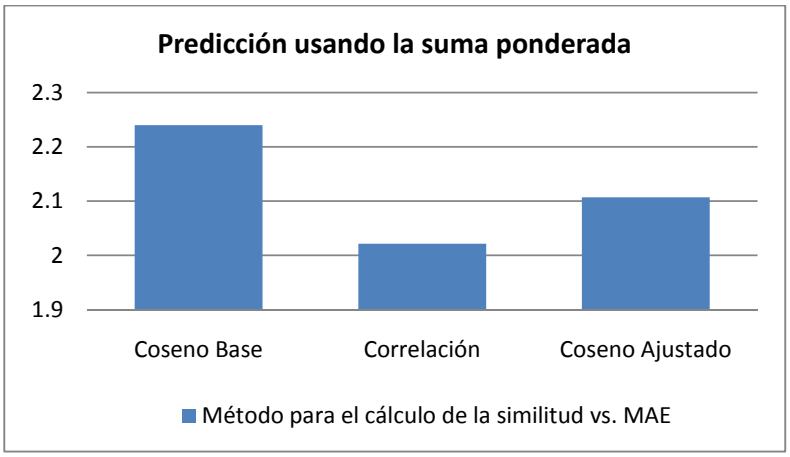

Figura 6.2. Predicción que utiliza la suma ponderada simple

En la figura 6.3., se observa que el mejor MAE para la predicción que usa el método de regresión (Ecuación 3.28) es el que corresponde al cálculo de la similitud por intermedio del coseno ajustado.

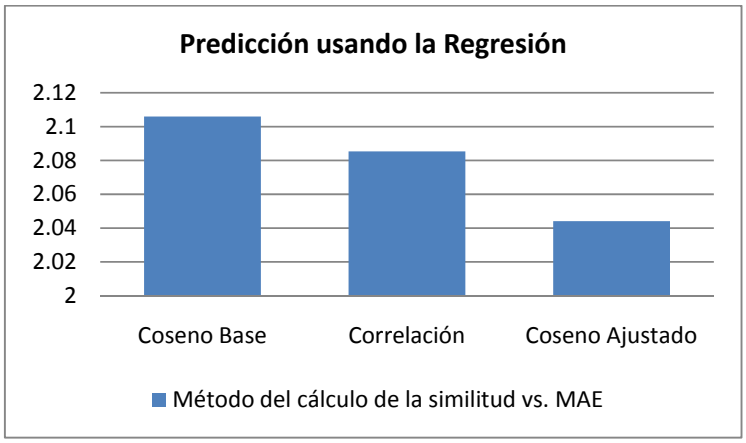

Figura 6.3. Predicción que utiliza la regresión

Finalmente, haciendo una comparación entre los diferentes métodos que se pueden utilizar para el cálculo de la predicción, utilizando, para el cálculo de la similitud, únicamente el método basado en correlaciones, podemos observar que la mejor predicción se obtiene mediante la suma ponderada de todas las evaluaciones del estudiante para el cual se está llevando a cabo la predicción.

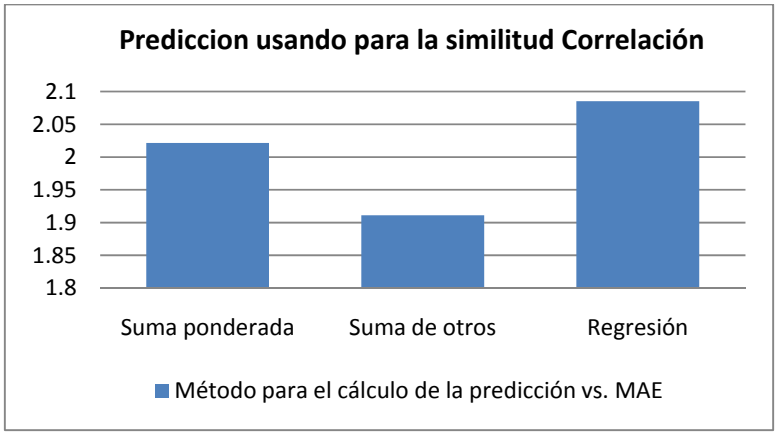

Figura 6.4. Predicción que utiliza correlaciones para el cálculo de la similitud

Observando el gráfico de la figura 6.4., se puede concluir que la mejor combinación para determinar la predicción usando filtrado colaborativo es:

- Cálculo de la similitud basada en correlación de Pearson.
- Cálculo de la predicción que utiliza la suma ponderada de otros.

Cabe mencionar que no se ha utilizado ningún concepto relacionado con el potencial, malla académica ni requisito de asignaturas explicadas en el capítulo 5, debido a que el método adoptado para el cálculo de la predicción incluye todas las notas que el estudiante ha obtenido en el pasado.

La totalidad de experimento y pruebas, llevadas a cabo en este contexto con el uso de otras métricas de rendimiento, están detalladas en el anexo A.2.

6.2. Experimentación para la aplicación de filtrado colaborativo basado en modelos

Es conocido que ningún algoritmo de aprendizaje automático trabaja de manera eficiente para todos los conjuntos de datos y mucho menos para todos los dominios de aplicación. Debido a que el objetivo del presente trabajo es proponer un sistema de recomendación que utilice técnicas de predicción con estudiantes que deseen asesorarse al efectuar su matrícula en un semestre académico, en esta sección, nos centraremos en presentar las experimentaciones más importantes hechas en este campo, así como la propuesta final de motor de aprendizaje más apropiado para los datos en estudio.

Para lograr este objetivo, en primer lugar, se evaluará el rendimiento (basado en precisión) que presentan las técnicas de árboles de decisión, vecinos cercanos y Naïve Bayes [Brav-07], cuando se analizan y comparan los datos originales extraídos a partir de la base de datos contra el conjunto de datos resultantes de la aplicación de la metodología propuesta en el capítulo 5 del presente trabajo de investigación. Con el diseño de experimentos planteado en el presente capítulo, se pretende aceptar o rechazar las siguientes hipótesis:

- *Los atributos sintéticos (potencial y dificultad) agregados a la base de datos original logran que los algoritmos de aprendizaje automático obtengan mejores predicciones en nuestro dominio de aplicación.*
- *La técnica de árboles de decisión, como técnica base, resulta ser la más eficiente para que el sistema de recomendación pueda entregar mejores recomendaciones.*

Debido a las características cíclicas de nuestro dominio de aplicación, es decir, que cada asignatura que pertenece al plan curricular siempre es cursada por diversos estudiantes, que los períodos académicos son consecutivos, que existe modificaciones curriculares, que los estudiantes van concluyendo sus estudios y que nuevos estudiantes ingresan en cada período académico, se tiene una ligera sospecha que los periodos académicos más antiguos influyen en forma negativa al momento de aplicar alguna técnica de aprendizaje.

Es por ello que se desarrolla la segunda fase de experimentación que recibe el nombre de experimentación de cortes. Para desarrollar este experimento, se usaron los resultados obtenidos en la fase anterior. Es decir, con el mejor conjunto de datos y con la mejor técnica, se procede a desarrollar un experimento que consiste en el aprendizaje de modelos a partir de una variedad de conjuntos de datos. El primer subconjunto de datos corresponde a todas las instancias que registran las matrículas y el rendimiento de cada

estudiante, desde que se creó la facultad (19912) hasta el período 20091. El segundo grupo de datos es el mismo que el primero sin los datos que corresponden al período más antiguo, es decir, al primer período (19912). El siguiente grupo de datos, es igual que el anterior, sin el grupo de instancias que corresponden a los dos primeros períodos. Así se forman 31 subconjuntos, siendo el último corte el que empieza en el período 20062 y termina en el 20091.

Con el diseño de experimentos planteado en el presente capítulo, se pretende aceptar o rechazar la siguiente hipótesis:

- *El conjunto de las instancias que registran la matrícula y las notas obtenidas por los estudiantes es afectado por el tiempo.*

En una siguiente etapa, se hace un estudio empírico de los métodos de poda para los árboles de decisión. Debido a los resultados de los experimentos anteriores, se estudian los métodos de poda y luego se hace un diseño de experimentos considerando los conjuntos que resultan de algunos subconjuntos y de la aplicación de los cortes (subconjuntos) más importantes y con menor error encontrados en experimento anterior.

Con el diseño de experimentos planteado en el presente capítulo, se pretende aceptar o rechazar la siguiente hipótesis:

- *Los métodos de poda en el aprendizaje de árboles de decisión afectan los modelos aprendidos.*

Finalmente, se prueba la eficacia de los algoritmos que propone la presente investigación. La experimentación incluye la ejecución de los algoritmos propuestos mediante el uso de la base de datos obtenida después de la preparación de los datos y de la metodología propuesta en el presente trabajo. Los resultados de estos algoritmos son comparados utilizando diferentes métricas de rendimiento.

Para llevarlos a cabo, se han utilizado las diferentes técnicas revisadas en el presente trabajo con sus diferentes versiones. Una versión, para algún algoritmo, corresponde a su instancia, que resulta de introducir parámetros para su ejecución. Estos parámetros externos, diferentes en cada algoritmo de aprendizaje, se utilizan para condicionar al modelo y obtener diferentes resultados, que en algunos casos son mejores que los algoritmos aplicados con sus opciones por defecto. Por ejemplo, para aplicar el algoritmo de árboles de decisión se usó un factor de confianza de 0.4 y número mínimo de objetos en las hojas de 40 instancias.

6.2.1. Determinación de las mejores condiciones para el aprendizaje automático

Objetivo:

El experimento es secuencial; esto quiere decir que los resultados parciales de alguna experimentación se usan para las que siguen. Tiene como objetivos la comprobación empírica de tres factores:

6.2.1.1. El algoritmo de clasificación más eficaz.

6.2.1.2. El mejor conjunto de datos.

6.2.1.3. El mejor tratamiento de cálculo de potencial.

Procedimiento:

Se han considerado dos conjuntos del mismo tamaño (número de instancias). Ambos poseen 161247 instancias, que son las que corresponden a cuando un estudiante cursó una determinada asignatura en la Facultad de Ingeniería de Sistemas desde el período académico 19912, período en que dio inicio a sus actividades hasta el periodo 20091.

En la tabla 6.4., se observa el primer conjunto de datos; este posee cinco atributos y una clase. Este conjunto proviene de la extracción de los datos del conjunto original, previamente preparado (limpieza). El segundo conjunto (tabla 6.5) posee dos atributos sintéticos adicionales, que son los resultantes de la aplicación de la metodología a la base de datos original. Cabe mencionar que ambos conjuntos poseen las mismas instancias y que su única diferencia está en los atributos.

Curso	vez	PPA	Creditaje de Asignatura	Número de créditos matriculados	Clase
Diseño Lógico	1	15.2173	3	13	Aprob
Programación	1	11.8890	4	19	Aprob
Prod. de Software para la Gestión	1	10.6384	3	16	Aprob
Estructuras Discretas	1	11.9710	5	20	Aprob
Ing. Software I	1	12.5750	4	22	Susp
Cálculo III	2	12.0322	4	22	Aprob
Teoría Sistemas	1	12.7714	2	17	Aprob
Lenguaje I	2	6.6500	4	17	Aprob
Historia Universal	1	8.6750	3	14	Aprob

Tabla 6.4. Conjunto de datos primitivos

Curso	Vez	PPA	Potencial.	Creditaje de Asignatura	Número de créditos Matriculados	Dific.	Class
Diseño Lógico	1	15.2173	1.317	3	13	11.61	Aprob
Programación	1	11.8890	0.974	4	19	13.50	Aprob
Prod. de Software para la Gestión	1	10.6384	0.894	3	16	13.11	Aprob
Estructuras Discretas	1	11.9710	1.199	5	20	12.70	Aprob
Ing. Software I	1	12.5750	0.993	4	22	12.94	Susp
Cálculo III	2	12.0322	0.920	4	22	10.82	Aprob
Teoría Sistemas	1	12.7714	1.161	2	17	12.27	Aprob
Lenguaje I	2	6.6500	0.519	4	17	11.54	Aprob
Historia Universal	1	8.6750	0.813	3	14	11.78	Aprob

Tabla 6.5. Conjunto de datos con dificultad y potencial

Uno de estos atributos es el potencial. Este posee cuatro variantes que corresponden, como se vio en la sección correspondiente a la metodología propuesta, a sus cuatro formas de cálculo (tratamientos).

Nuestro diseño experimental está basado en *Holdout Resampling* que consiste en ordenar los datos aleatoriamente y particionarlos en dos conjuntos: 70 por ciento para el conjunto de entrenamiento y 30 por ciento para el conjunto de prueba. Así, este proceso se repite diez veces. Finalmente, los errores de predicción se promedian sobre todas las pruebas para calcular la predicción media del error y su correspondiente varianza o error estándar.

Para este experimento, se usan los algoritmos de aprendizaje de árboles de decisión, vecinos cercanos y Naïve Bayes. Cada algoritmo es ejecutado con cada uno de los diez conjuntos de entrenamiento, construidos aleatoriamente, usando diferentes configuraciones. Es así que para los árboles de decisión usamos las opciones de factor de confianza (F.C.=0.4) y también consideramos que el número mínimo de instancias en cada hoja no debe ser menor de 40 (M=40). Estos valores son encontrados como óptimos luego de varias experimentaciones con conjuntos de datos del mismo dominio. El algoritmo de vecinos cercanos, en nuestro caso, acepta como parámetro de entrada el valor K, que corresponde al número de vecinos que se consideran en el aprendizaje. Usaremos K=91, debido a que es conocido que la relación $k = n^{\frac{3}{8}}$ obtiene el mejor aprendizaje [Enas-96], siendo n el numero de instancias de entrenamiento y k el número de vecinos más cercanos. Por otro lado, para aplicar los algoritmos de clasificación por conjuntos, se toma como algoritmo base los árboles de decisión, con sus mismos parámetros de entrada, y se construye la votación utilizando 25 iteraciones. [Opit-99].

Una vez construidos los modelos, estos son validados con su respectivo conjunto de prueba, y se obtiene el porcentaje de error.

En la tabla 6.6., se pueden observar los errores resultantes después de ejecutar cada uno de los tres algoritmos de aprendizaje sobre la base de datos con cinco atributos (conjunto de datos primitivos). Así mismo, en la tabla 6.7(a) (b) (c) (d), se puede observar el porcentaje de error obtenido al entrenar cada una de las bases de datos correspondientes a los diferentes formas de obtener el atributo potencial que en este trabajo las hemos llamado potencial N1, potencial N2, potencial NT y potencial PPA.

Para las pruebas estadísticas, usaremos la prueba t pareada. Esta prueba se utiliza, en general, cuando se registran datos de un mismo individuo antes y después de la aplicación de un efecto que se desea analizar. En esta fase de experimento, la prueba t pareada se empleó de la siguiente manera: los efectos a estudiarse fueron el algoritmo de clasificación, los conjuntos de datos y la metodología de cálculo de potencial respectivamente, y los individuos, los diferentes conjuntos de datos obtenidos después de hacer el Holdout Resampling. Las observaciones que se registraron fueron las tasas de error que se generaron al aplicar las diferentes técnicas a cada uno de los conjuntos producidos con la técnica propuesta en los experimentos correspondientes.

En los resultados, se puede observar un signo que corresponde al valor de la estadística de prueba y el valor que corresponde al p-value. Aquí se puede observar la significancia de diferencias entre los algoritmos de clasificación, los conjuntos de datos y la metodología de cálculo de potencial, según corresponda. Un signo +(-) delante del valor de p-value indica que alguna de las condiciones lleva a cabo mejor (peor) el aprendizaje. En el caso de que al valor del p-value no le anteceda un signo, sino un número cero, esto indicará que no hay diferencias significativas entre un tratamiento y otro. Los valores que aparecen entre paréntesis, representan los p-values de las pruebas t pareada. Un p-value es la probabilidad de obtener un valor más extremo (más pequeño o más grande) que el observado. En nuestro caso, si para el p-value se obtiene un nivel de significancia por debajo del 5%, se rechazará la hipótesis nula. Esto quiere decir que se debe considerar que los errores de predicción tienen promedios diferentes.

Split	C4.5	KNN IBK	Naïve Bayes
1	18.8238	19.0326	22.1085
2	18.9664	19.0760	22.3959
3	18.9354	19.1049	22.2842
4	18.7225	18.8486	22.0899
5	18.9106	18.9871	22.3318
6	18.8258	19.0264	22.1251
7	19.1070	19.0760	22.1933
8	18.9664	19.0739	22.2057
9	18.7762	19.1276	22.1003
10	19.1339	19.1711	22.5488
	18.92±0.13	19.05±0.09	22.24±0.15

Tabla 6.6. Conjunto de datos primitivos

Split	C4.5	KNN IBK	Naïve Bayes
1	18.7638	18.9127	23.7519
2	18.7225	18.9726	23.9214
3	18.6398	18.586	23.4687
4	18.5468	18.7018	23.8574
5	18.8548	18.9209	23.9276
6	18.6481	18.8362	23.9401
7	18.6997	18.6956	23.6775
8	18.8424	18.9643	23.8615
9	18.7494	18.83	23.7912
10	18.677	18.6357	23.6382
	18.71±0.09	18.81±0.14	23.78±0.15

Tabla 6.7(a). Resultados con Potencial N1

Split	C4.5	KNN IBK	Naïve Bayes
1	18.369	18.6708	23.7416
2	18.8651	18.7618	23.9483
3	18.5199	18.677	23.7912
4	18.7163	18.8238	23.7747
5	18.8258	18.6667	23.8698
6	18.8155	18.7307	23.6196
7	18.7886	18.7018	24.093
8	18.5881	18.6977	23.5142
9	18.799	18.801	23.6713
10	18.7473	18.7597	23.6858
	18.7±0.16	18.73±0.06	23.77±0.17

Tabla 6.7(b). Resultados con Potencial N2

Split	C4.5	KNN IBK	Naïve Bayes
1	18.9189	18.9333	24.0579
2	18.9395	18.8672	24.1385
3	18.6315	18.708	23.63
4	18.7659	18.9809	23.8388
5	18.5819	18.7866	23.7933
6	18.7225	18.8134	23.754
7	18.6873	18.8279	23.7457
8	18.6543	18.5798	23.7499
9	18.6481	18.7928	23.6837
10	18.8134	18.9023	23.8718
	18.74±0.12	18.82±0.12	23.83±0.16

Tabla 6.7(c). Resultados con Potencial NT

Split	C4.5	KNN IBK	N Bayes
1	18.8713	18.8734	23.5473
2	18.708	18.7473	23.3778
3	18.9788	18.9437	23.439 8
4	18.8775	18.9147	23.3116
5	18.8713	19.0098	23.6941
6	18.7783	19.1276	23.3902
7	18.7349	19.0636	23.4481
8	18.7969	19.0904	23.5618
9	18.6956	18.7514	23.4398
10	18.6253	18.9003	23.0842
	18.79±0.11	18.94±0.13	3.43±0.16

Tabla 6.7(d). Resultados con Potencial PPA

6.2.1.1. Experimento referido a la eficacia del algoritmo de clasificación

Con este experimento, se pretende responder a la siguiente pregunta: ¿cuál de los algoritmos usados en la presente investigación produce menor error de predicción?

Para responderla, se utilizan los porcentajes de error obtenidos. Para ello, se hace un análisis estadístico dentro de cada una de las tablas, que consiste en aplicar la prueba t-pareada a los resultados de la tabla 6.6 y los de las tablas 6.7(a), 6.7 (b), 6.7 (c) y 6.7 (d). En la tabla 6.8., se puede observar una descripción de las diez pruebas realizadas.

Comprobación del efecto del algoritmo de clasificación	
KNN vs. C4.5	Dentro de las tablas 4, 5(a), 5(b), 5(c), 5(d)
Naive Bayes vs. C4.5	Dentro de las tablas 4, 5(a), 5(b), 5(c), 5(d)

Tabla 6.8. Pruebas estadísticas para encontrar el mejor algoritmo

Los resultados obtenidos se pueden visualizar en la tabla 6.9:

Prueba	*Datos Primitivos*	*Potencial N1*	*Potencial N2*	*Potencial NT*	*Potencial PPA*
KNN vs. C4.5	- (0.0028)	- (0.0188)	0 (0.5842)	- (0.0299)	- (0.0115)
NB vs C4.5	- (0)	- (0)	- (0)	- (0)	- (0)

Tabla 6.9. Resultados de la prueba t-pareada para la comprobación de la eficacia de la técnica

Interpretación:

Los resultados de la prueba t-pareada indican que:

- La tasa promedio de error del algoritmo C4.5 es menor que la tasa promedio de error del algoritmo Naive Bayes tanto en el conjunto de datos primitivos como en los cuatro tratamientos (formas de calcular el valor del potencial) del potencial.
- La tasa promedio de error del algoritmo C4.5 es menor que la tasa promedio de error del algoritmo KNN cuando se utiliza el conjunto de datos primitivos y los tratamientos de cálculo potencial N1,NT y PPA, siendo igual en el tratamiento N2.

Conclusión:

Se concluye que C4.5 es el algoritmo más eficaz prediciendo nuevas instancias en nuestro dominio de aplicación. Además, es el más adecuado debido a su capacidad de representación, facilidad de interpretación y menor costo computacional [Rock-08].

6.2.1.2. Experimento referido a la eficacia de los conjuntos de datos

Con el siguiente experimento se pretende responder a la siguiente pregunta: ¿cuál de los conjuntos de datos produce menor error de predicción? Cabe mencionar que para este experimento usamos el resultado del anterior. Quiere decir que solo utilizaremos, para nuestra comparación, los árboles de decisión.

Para responder a esta pregunta, se utilizan los porcentajes de error obtenidos. Para ello se hace un análisis estadístico entre las diferentes tablas, que consiste en aplicar la prueba t-pareada a los resultados de la tabla 6.6 y los de las tablas 6.7(a), 6.7 (b), 6.7 (c) y 6.7 (d). En la tabla 6.10., se puede observa una descripción de las cuatro pruebas realizadas.

Comprobación del efecto de los nuevos atributos (potencial y dificultad)		
C4.5 usando los datos de la tabla 4	vs.	C4.5 usando los datos de la tabla 5(a)
C4.5 usando los datos de la tabla 4	vs.	C4.5 usando los datos de la tabla 5(b)
C4.5 usando los datos de la tabla 4	vs.	C4.5 usando los datos de la tabla 5(c)
C4.5 usando los datos de la tabla 4	vs.	C4.5 usando los datos de la tabla 5(d)

Tabla 6.10. Pruebas estadísticas para encontrar el mejor conjunto de atributos

Los resultados obtenidos se pueden visualizar en la tabla 6.11:

	Potencial N1 (C4.5)	**Potencial N2 (C4.5)**	**Potencial NT (C4.5)**	**Potencial PPA (C4.5)**
Datos primitivos (C4.5)	+ (0.0018)	+ (0.0069)	+ (0.0104)	0 (0.0877)

Tabla 6.11. Resultados de la prueba t-pareada para la comprobación de la eficacia del conjunto

Interpretación:

- Los tratamientos N1, N2 y NT tienen menores promedios de tasas de error cuando se usa el algoritmo C4.5 comparados con el conjunto de datos primitivo.
- Para el caso del potencial PPA, se observa que no existe una diferencia significativa en las tasas de error entre este tratamiento y el conjunto de datos primitivo.

Conclusión:

Se concluye que el conjunto de datos con atributos sintéticos obtiene mejores promedios de tasas de error y logra representar mejor la realidad estudiada. Cabe destacar que el tratamiento de cálculo de potencial PPA no presenta diferencia significativa con respecto al conjunto de datos primitivo, por lo que este será descartado en las siguientes secciones.

6.2.1.3. Experimento referido a la eficacia de los tratamientos de cálculo de potencial

Con el siguiente experimento se pretende responder a la siguiente pregunta: ¿cuál de las metodologías usadas para el cálculo del potencial es la que produce menor error de predicción? Cabe mencionar que para este experimento usamos el resultado del anterior. Quiere decir que solo utilizaremos, para nuestra comparación, los árboles de decisión y las bases de datos con siete atributos.

Para responder a esta pregunta, se utilizan los porcentajes de error obtenidos. Para ello se hace un análisis estadístico entre las diferentes tablas, que consiste en aplicar la prueba t-pareada a los resultados de las tablas 6.7(a), 6.7 (b) y 6.7(c). En la tabla 6.12., se puede observar una descripción de todas las tres pruebas realizadas.
Para esto, se usaron tres pruebas t pareadas:

Comprobación de la mejor metodología de cálculo de potencial		
C4.5 usando los datos de la tabla 5(a) N1	vs.	C4.5 usando los datos de la tabla 5(b) N2
C4.5 usando los datos de la tabla 5(a) N1	vs.	C4.5 usando los datos de la tabla 5(c) NT
C4.5 usando los datos de la tabla 5(b) N2	vs.	C4.5 usando los datos de la tabla 5(c) NT

Tabla 6.12. Pruebas estadísticas para encontrar el mejor potencial

Los resultados obtenidos se pueden visualizar en la tabla 6.13:

Prueba	Signo (P-value)
Potencial N1 vs. Potencial N2	0 (0.8596)
Potencial N1 vs. Potencial NT	0 (0.6925)
Potencial N2 vs. Potencial NT	0 (0.6426)

Tabla 6.13. Resultados de la prueba t-pareada para la comprobación de la eficacia de la metodología

Interpretación:
Se observa que no hay diferencias significativas entre los promedios de las tasas de error en las tres metodologías de cálculo de potencial.

Conclusión:
Se concluye que los tratamientos de cálculo de potencial N1, N2 y NT obtienen estadísticamente iguales promedios de tasas de error. Cabe mencionar que en el proceso real de asesorías en la universidad, el asesor de matrícula considera, en su mayoría, la nota del curso requisito inmediato anterior. Por lo tanto, nosotros decidimos utilizar el conjunto de datos con potencial N1.

6.2.2. Experimento referido a la independencia de los datos con respecto al tiempo

Objetivo:

El presente experimento tiene el objetivo de comprobar el efecto que tienen los datos de semestres regulares cercanos a la creación de la facultad (los más antiguos) en la predicción.

Consideraciones:

El ámbito universitario y en especial nuestro dominio de aplicación presentan algunas características que hacen que los datos puedan ser analizados bajo ciertos

escenarios. En primer lugar, desde el punto de vista de los estudiantes, estos no son los mismos períodos a período. Si bien es cierto que de un período a otro no hay cambios significativos en su rendimiento, sí es posible que, después de algunos períodos, el cambio generacional lleve consigo un cambio de tendencias como para que un conjunto de entrenamiento represente la realidad de manera coherente.

Desde el punto de vista de las asignaturas, existe un gran grupo de ellas que va cambiando cada cierto tiempo. Al momento de la modificación curricular, la institución crea ciertas reglas de juego con el objetivo de que sus estudiantes no sean afectados por esta modificación. Una de las reglas más importantes está referida a las equivalencias entre las asignaturas. Sin estas, es evidente que sería imposible transportar en el tiempo una asignatura que ha sufrido más de una modificación, lo que llevaría consigo una eliminación de tales registros, en la fase de limpieza, y por ende una pérdida de información.

En este sentido, existe la hipótesis de que, mientras más lejos esté el período de análisis, este influirá negativamente en las predicciones que genera el algoritmo de aprendizaje utilizado con la finalidad de recomendar a estudiantes futuros sobre las asignaturas a cursar.

En el caso de probarse nuestra hipótesis, tendríamos que proceder a determinar, por alguna técnica estadística, el conjunto de datos que mejor represente la realidad. En caso contrario, lo más adecuado sería considerar todo el conjunto de datos. Es decir, el que corresponde a los períodos 19912 hasta el período 20091, debido que presenta la mayor cantidad de datos, lo que influiría en una mejor estimación en la clasificación al momento de generar la recomendación, debido al concepto de consistencia [Lehm-98].

Procedimiento:

Se crearon 31 subconjuntos de entrenamiento distintos de acuerdo con la figura 1. Dichos conjuntos, que para efectos de esta investigación llamaremos “cortes”, son estudiados utilizando el algoritmo C4.5.

El hecho de utilizar Holdout Resampling, es decir, dos conjuntos de datos independientes, uno para el aprendizaje del modelo o de las hipótesis y el otro para evaluarlas, permite evitar el problema de premiar el sobreajuste. Sin embargo, aún existe el problema de que el resultado del modelo aprendido sea demasiado dependiente del conjunto de entrenamiento. Esta última afirmación contraviene el hecho de que la construcción del modelo debería ser muy general como para que cualquier conjunto de datos de prueba tenga porcentajes de error parecidos a los del conjunto de prueba inicial. En este caso, diremos que el modelo aprendido es el óptimo para el modelo de aprendizaje utilizado.

La aplicación de Holdout Resampling implica la reducción, en gran medida, de la dependencia del resultado del experimento al conjunto que se utiliza para entrenarlo. La

ventaja de este método es que la varianza de los n errores de cada una de las pruebas permite estimar la variabilidad del método de aprendizaje con respecto a los datos de entrenamiento.

En la figura 6.5., se pueden observar los 31 cortes que corresponden a 31 subconjuntos distintos de datos de entrenamiento. Estos conjuntos, inicialmente, se obtuvieron al separar los datos según el período académico al que correspondían. Como se puede ver, desde el período 19912 hasta el período 20062, existen 31 cortes. Estos han sido construidos separando gradualmente los datos que corresponden a los períodos más antiguos, con el objetivo de tener una idea del comportamiento de los datos en cada uno de los conjuntos, además de verificar la estabilidad de cada conjunto sin considerar el período académico más antiguo.

Una vez que los datos fueron separados, se procedió a aplicar el algoritmo de aprendizaje con la metodología de evaluación Holdout Resampling. Luego de ello, se obtiene un modelo y su respectivo porcentaje de error de predicción, que se pueden observa en la tabla 6.14.

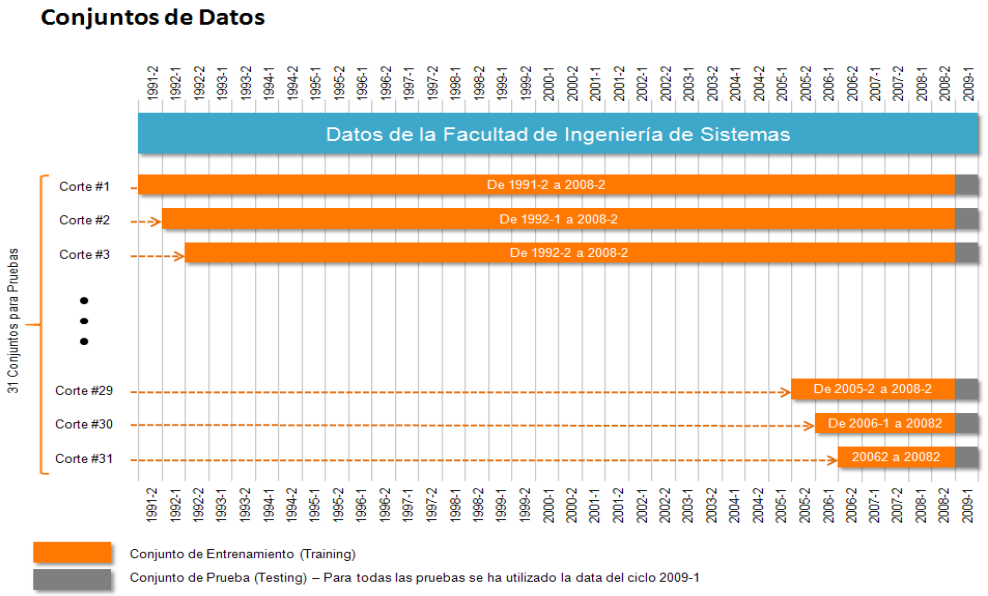

Figura 6.5. Formación de los subconjuntos (cortes) tomando en consideración períodos académicos

Experimento:

Con el siguiente experimento, se pretende responder a la pregunta: ¿los datos con mayor antigüedad generan alteraciones significativas en el error de las predicciones obtenidas? Cabe mencionar que para este experimento se usaron los resultados de la fase anterior. Quiere decir que solo utilizaremos, para nuestra comparación, el algoritmo C4.5 y los tres tratamientos de cálculo de potencial (N1, N2 y NT).

Año	N1	N2	NT
1991-2	18.71443	18.70346	18.73633
1992-1	18.66667	18.72351	18.84734
1992-2	18.75663	18.74977	18.73937
1993-1	18.69761	18.67613	18.75992
1993-2	18.5975	18.61778	18.67632
1994-1	18.63431	18.61908	18.62431
1994-2	18.66304	18.62731	18.64019
1995-1	18.56569	18.58781	18.51238
1995-2	18.45229	18.58835	18.70223
1996-1	18.54976	18.61871	18.6715
1996-2	18.68794	18.74084	18.70938
1997-1	18.77143	18.7193	18.83452
1997-2	18.97613	19.02178	19.10364
1998-1	19.15082	19.20933	19.10132
1998-2	19.30548	19.28156	19.32113
1999-1	19.34892	19.5113	19.48591
1999-2	19.51384	19.56671	19.57482
2000-1	19.61255	19.68224	19.51549
2000-2	19.44221	19.50245	19.60065
2001-1	19.38029	19.34249	19.52245
2001-2	19.45067	19.48999	19.56831
2002-1	19.4081	19.57295	19.47302
2002-2	19.19991	19.44703	19.32823
2003-1	19.01441	19.07525	19.16089
2003-2	19.00024	19.08816	19.05019
2004-1	18.78572	18.91395	19.06188
2004-2	18.75145	18.77759	18.91261
2005-1	18.79554	18.75035	18.95615
2005-2	18.90656	18.99523	18.90884
2006-1	18.79187	18.89706	19.01105
2006-2	18.89723	19.0661	19.06095

Tabla 6.14. Porcentajes de error por cada corte y tratamiento

Para responder a esta pregunta, se utilizan los promedios de las tasas de error obtenidos en cada una de las pruebas (Tabla 6.14); luego se aplica una prueba de hipótesis de igualdad de proporciones. En esta prueba, la distribución chi-cuadrado es la estadística de prueba. El objetivo es determinar si existe diferencia significativa entre las tasas promedios de error registradas en los 31 cortes.

	N1	N2	NT
Chi-Cuadrado	7.5529	8.5929	7.7875
P-Value	0.9999	0.9999	0.9999

Tabla 6.15. Resultados de la prueba chi-cuadrado

Interpretación:
En la tabla 6.15., se observa que los valores p-value son mayores que el nivel de significación del 0.05; por lo tanto, no hay evidencia estadística suficiente para sostener que las tasas promedios de error son diferentes en los 31 cortes para cada uno de los tratamientos N1, N2 y NT.

Conclusión:
Con la prueba de proporciones, se concluye que no existen diferencias significativas entre los porcentajes de error obtenidos con los datos de mayor antigüedad y los datos de períodos más recientes para cada uno de los tratamientos. Con estos resultados, es conveniente utilizar el conjunto de datos que corresponden al primer corte. Es decir, los registros acumulados desde el 19912 hasta el 20091.

6.2.3. Eficiencia de los diferentes métodos de poda para árboles de decisión en nuestro dominio de aplicación (comparación empírica)

6.2.3.1. El diseño del experimento

En esta sección, se presentan los resultados de una comparación empírica de los diferentes métodos de poda detallados en esta investigación. Cabe destacar que el diseño del experimento tiene sus bases en el diseño propuesto por Floriana Espósito en [Espo-97].

Objetivo:

Con este experimento, se pretende determinar cuál de los métodos de poda produce menor tasa de error en nuestro dominio de aplicación.

Procedimiento:

Las principales características de los conjuntos de datos considerados en nuestro experimento se pueden visualizar en la Tabla 6.16.

Se han elegido doce conjuntos distintos de datos. Estos conjuntos fueron seleccionados a partir del grupo de treintaiún conjuntos que corresponden a la fase de experimentación 6.2.2. Cada corte, a su vez, corresponde a la agrupación de instancias relativas a los períodos académicos desde el 19921 hasta el 20062. El último corte contiene instancias que corresponden a seis períodos académicos, es decir, desde el

período 20062 hasta el período 20091, último considerado para efectos de esta investigación.

Se eligieron doce cortes a partir de tres subconjuntos distintos, que corresponden a los períodos 19952, 20021, 20041. Los dos últimos corresponden a aquellos en donde se aplicó cambio de plan curricular, y el primero corresponde a aquel en donde el aprendizaje obtuvo el menor porcentaje de error del tratamiento con Holdout Resampling.

Cada uno de los tres períodos académicos mencionados anteriormente corresponden a grupos con un número distinto de instancias: el conjunto que corresponde al corte 19952 contiene 142 573 instancias, el que corresponde al corte 20021 contiene 79 064 instancias y el que corresponde al corte 20041 contiene 59 264 instancias. A cada uno de estos grupos se les aplicó tres métodos distintos para el cálculo del potencial del estudiante, lo que corresponde a los primeros nueve conjuntos. Los tres últimos conjuntos corresponden a la aplicación de la técnica de *oversampling* (se detalla en el punto 4.6.1.1) para el conjunto del corte 19952 (199 251 registros) con los tres tratamientos para de cálculo del potencial, por ser este el que obtiene el mejor resultado en el tratamiento N1.

Conjunto		**# de Clases**	**# de Atributos**	**# Real**	**# Mult.**	**% EB**	**Distribución Uniforme**
Tratamiento	Corte						
Potencial NT	19952	2	7	5	2	21.60%	No
Potencial N1	19952	2	7	5	2	21.60%	No
Potencial N2	19952	2	7	5	2	21.60%	No
Potencial NT	20021	2	7	5	2	22.61%	No
Potencial N1	20021	2	7	5	2	22.61%	No
Potencial N2	20021	2	7	5	2	22.61%	No
Potencial NT	20041	2	7	5	2	21.69%	No
Potencial N1	20041	2	7	5	2	21.69%	No
Potencial N2	20041	2	7	5	2	21.69%	No
Potencial NT Over	19952	2	7	5	2	50.00%	Sí
Potencial N1 Over	19952	2	7	5	2	50.00%	Sí
Potencial N2 Over	19952	2	7	5	2	50.00%	Sí

Tabla 6.16. Descripción de datos

En la Tabla 6.16, las columnas “#Real” y “#Mult” presentan la cantidad de atributos tratados como valores reales y como atributos discretos, respectivamente.

La columna de % EB se refiere al porcentaje de error que se obtendría si la clase más frecuente fuese predicha. Lo que esperamos es que, después del aprendizaje, obtengamos tasas de error más bajas que las presentadas en % EB. La última columna establece si la distribución de ejemplos por clase es uniforme.

Para el experimento, cada conjunto de datos ha sido dividido en tres subconjuntos: conjunto crecimiento (Crecimiento, 49 por ciento), conjunto poda (Poda, 21 por ciento) y conjunto de prueba (Test, 30 por ciento). La unión del conjunto de crecimiento y el de poda se llama "conjunto de entrenamiento". El conjunto de crecimiento y el conjunto entrenamiento se usan para aprender dos árboles de decisión, llamados "árboles de crecimiento" y "de entrenamiento" respectivamente.

El árbol de crecimiento se utiliza para aquellos métodos que requieren un conjunto independiente para podar un árbol de decisión (REP, MEP, CVP), detallados en los puntos 4.5.1, 4.5.3, 4.5.4, así como la poda de costo complejidad, basada en un conjunto de poda independiente que adopta dos distintas reglas de selección 0SE Y 1SE. En cambio, el árbol de entrenamiento es usado por aquellos métodos que explotan solo los conjuntos de entrenamiento tales como el PEP**,** EBP.

La evaluación de la tasa de error siempre se hace en un conjunto de prueba usando el conteo empírico de errores [Kitt-82], el cual es un estimador objetivo e imparcial.

La distribución de casos en los conjuntos de crecimiento, poda, entrenamiento y prueba para cada base de datos se muestra en la Tabla 6.17.

Conjunto		**# de Datos**	**# de Training**	**# Grow**	**# Prune**	**# Test**
Tratamiento	Corte					
Potencial NT	19952	142573	99800	69860	29940	42773
Potencial N1	19952	142573	99800	69860	29940	42773
Potencial N2	19952	142573	99800	69860	29940	42773
Potencial NT	20021	79064	55344	38741	16603	23720
Potencial N1	20021	79064	55344	38741	16603	23720
Potencial N2	20021	79064	55344	38741	16603	23720
Potencial NT	20041	59264	41484	29039	12445	17780
Potencial N1	20041	59264	41484	29039	12445	17780
Potencial N2	20041	59264	41484	29039	12445	17780
Potencial NT	19952 - O	199251	156478	109534	46944	42773
Potencial N1	19952 - O	199251	156478	109534	46944	42773
Potencial N2	19952 - O	199251	156478	109534	46944	42773

Tabla 6.17. Descripción de datos

Se aplica el método de Holdout Resampling dividiendo aleatoriamente en conjuntos de crecimiento, poda y prueba diez veces. Para cada ejecución del algoritmo C4.5, se toman en cuenta dos estadísticas: el número de nodos del árbol y la tasa de error del árbol en el conjunto de Test. Esto se aplica a los árboles de poda, de crecimiento y de entrenamiento.

Nuestro diseño experimental, basado en Holdout Resampling*,* ha sido usado en algunos otros estudios empíricos tales como los realizados por Mingers [Ming-89],

Buntine [Bunt-92] y Holte [Holt-93]. Los errores de predicción se promedian sobre todas las pruebas para calcular la predicción media del error y su correspondiente varianza o error estándar.

La media del error de la predicción podría ser una estimación imparcial en el caso de que el error observado de la predicción en los sucesivos conjuntos de prueba fuera independiente. Sin embargo, esto no es cierto debido a que el conjunto de prueba puede superponerse como consecuencia de la aleatoriedad del *resampling*. Consecuentemente, cuando un conjunto de prueba se usa para evaluar el significado de la diferencia en errores de predicción, el resultado debe ser cuidadosamente interpretado. En particular, la significancia estadística debería leerse como muy probable para mantener alguna expectativa sobre los datos dados, y no como muy probable para mantenerla para datos futuros.

Para estudiar el efecto de la poda en la precisión predictiva de los árboles de decisión, comparamos las tasas de error del árbol podado con aquellas correspondientes a árboles entrenados. Para verificar si las técnicas de simplificación de árboles son beneficiosas, comparamos dos estrategias de inducción: una estrategia sofisticada que, de una manera u otra, poda el árbol T_{max}, y una estrategia simple (ingenua) que únicamente retorna T_{max} .

Otra característica interesante de los métodos de poda es su tendencia a sobrepodar árboles de decisión. Para estudiar el problema de la sobrepoda y subpoda de los métodos de poda, tomaremos en cuenta dos árboles de decisión para cada ensayo, el primero llamado OPGT (Árbol de Crecimiento Podado Óptimamente), por sus siglas en inglés y el segundo llamado OPTT (Árbol de Entrenamiento Podado Óptimamente). El primero es un árbol de crecimiento que ha sido simplificado aplicando el método REP al conjunto de prueba (como test). Con esto se puede afirmar que este es el mejor árbol podado que podríamos producir a partir del árbol de crecimiento [Espo-97]. De forma similar, el OPTT es el mejor árbol que podríamos obtener podando algunas ramas del árbol de entrenamiento. Los árboles OPGT y OPTT definen, en la mejora de precisión, una cota superior a la que una técnica de poda puede producir, así como una cota inferior en la complejidad del árbol podado.

Los árboles OPGT y OPTT se usan para investigar algunas propiedades de conjuntos de datos. Por ejemplo, comparando la precisión de los árboles Crecimiento/Entrenamiento con la precisión de los correspondientes $OPGT_S$ y $OPTT_S$, es posible evaluar la máxima mejora producida por un algoritmo ideal de poda. La magnitud de diferencia en la precisión de *OPGT* y *OPTT* puede ayudar a entender si el objetivo ideal de estos métodos de simplificación que requieren un conjunto de poda es similar al objetivo ideal para los otros métodos.

Por el contrario, una comparación de la precisión de los correspondientes árboles de crecimiento y de poda nos provee una indicación de la ventaja inicial que

algunos métodos pueden tener sobre otros. Además, el tamaño del árbol podado óptimamente puede ser explotado para señalar una tendencia de métodos de simplificación hacia una sobrepoda o subpoda.

En este caso, debemos comparar el tamaño de un *OPGT* con aquellos árboles producidos por los métodos que usan un conjunto independiente de poda, mientras el tamaño de un OPTT debe estar relacionado con el resultado de otro método.

Algunos resultados experimentales, los cuales son independientes de un método de poda en particular, se muestran en la Tabla 6.18(a)(b). Ellos están dados en forma de 18.74 ± 0.14, donde el primer número es el valor promedio para los diez ensayos, y el segundo número se refiere al correspondiente error estándar.

Según lo esperado, el tamaño promedio del árbol crecimiento (entrenamiento) es siempre mayor que el de OPGT (OPTT). La relación (Grown Tree Size/OPTG Size) va desde 3.11 para el conjunto del período 20021 con el método de potencial hasta 6.55 para el conjunto correspondiente al período 19952 con la metodología de potencial y mediante la aplicación de *oversampling*. El ratio (Trained Tree Size/OPTT Size) es ligeramente más grande para el conjunto correspondiente al período 19952 con *oversampling*.

		Grown		OPGT	
		EBP		REP	
Conjunto		Size	E.R.	Size	E.R.
Tratamiento	Corte		**% Testing**		**% Testing**
Potencial NT	19952	9163.9 ± 435.33	19.63 ± 0.15	2661.9 ± 281.14	18.04 ± 0.13
Potencial N1	19952	10519 ± 687.49	19.76 ± 0.16	2952.5 ± 302.61	17.88 ± 0.13
Potencial N2	19952	9696.3 ± 419.8	19.67 ± 0.16	2808.8 ± 299.18	17.97 ± 0.13
Potencial NT	20021	5279.7 ± 274.66	20.6 ± 0.23	1695.3 ± 178.78	18.98 ± 0.24
Potencial N1	20021	6060.2 ± 266.66	20.58 ± 0.27	1890.6 ± 242.08	18.74 ± 0.24
Potencial N2	20021	5488.8 ± 216.77	20.57 ± 0.27	1753.7 ± 230	18.95 ± 0.25
Potencial NT	20041	4331.4 ± 223.6	20.15 ± 0.34	1368.4 ± 194.38	18.49 ± 0.21
Potencial N1	20041	5029.5 ± 423.08	20.22 ± 0.3	1605.4 ± 275.93	18.24 ± 0.29
Potencial N2	20041	4647.3 ± 199.06	20.15 ± 0.31	1453.1 ± 121.03	18.44 ± 0.26
Potencial NT	19952 - O	24064.8 ± 841.71	29.01 ± 0.23	3672.6 ± 454.48	20.33 ± 0.16
Potencial N1	19952 - O	26235.4 ± 1094.03	28.68 ± 0.2	4077.6 ± 359.52	19.87 ± 0.24
Potencial N2	19952 - O	25796.4 ± 1549.85	28.9 ± 0.22	4093 ± 440.44	20.22 ± 0.25

Tabla 6.18 (a). Porcentaje de tamaño y tasa de error de OPTT y OPGT para cada conjunto

		Trained		OPTT	
		EBP		REP	
Conjunto		Size	E.R.	Size	E.R.
Tratamiento	Corte		% Testing		% Testing
Potencial NT	19952	11591 ± 261.31	19.32 ± 0.16	3237.4 ± 249.11	17.96 ± 0.13
Potencial N1	19952	13006.3 ± 730.62	19.37 ± 0.12	3642.9 ± 483.15	17.83 ± 0.12
Potencial N2	19952	12107 ± 703.14	19.35 ± 0.14	3441.2 ± 363.05	17.86 ± 0.13
Potencial NT	20021	6620.3 ± 222.35	20.31 ± 0.17	2196.9 ± 195.97	18.85 ± 0.18
Potencial N1	20021	7467.1 ± 588.94	20.37 ± 0.23	2530.3 ± 245.52	18.64 ± 0.24
Potencial N2	20021	7079.1 ± 472.15	20.34 ± 0.18	2091.3 ± 185.14	18.77 ± 0.19
Potencial NT	20041	5039.3 ± 472.53	19.83 ± 0.27	1559.4 ± 245.08	18.33 ± 0.27
Potencial N1	20041	6154.9 ± 488.79	19.86 ± 0.25	2000.9 ± 225.95	18.12 ± 0.29
Potencial N2	20041	5396 ± 486.11	19.85 ± 0.19	1739.8 ± 262.18	18.29 ± 0.26
Potencial NT	19952 - O	33961.5 ± 2087.64	28.13 ± 0.28	4951.4 ± 584.63	19.93 ± 0.21
Potencial N1	19952 - O	36221.9 ± 2286.18	27.76 ± 0.26	5866.4 ± 743.14	19.44 ± 0.19
Potencial N2	19952 - O	35369.4 ± 1118.79	28 ± 0.38	5762.9 ± 533.32	19.6 ± 0.23

Tabla 6.18 (b). Porcentaje de tamaño y tasa de error de OPTT y OPGT para cada conjunto

No hay ningún conjunto de datos en el cual el árbol entrenado presente una tasa de error más grande que el error base. En caso de que esto ocurriera, sería un ejemplo típico de sobreajuste, para lo cual las técnicas de poda deberían ser beneficiosas. Vale la pena observar que el promedio tamaño/error de los datos mostrados en la columna Trained de la Tabla 6.18(a) (b) es siempre mayor que la unidad, lo que es concordante con lo observado por Schaffer [Scha-92], quien afirmó que en el caso de no haber sobreajuste siempre el ratio sería un número mayor que la unidad.

Comparando la tasa de error de los árboles de crecimiento y de entrenamiento, podemos concluir que los árboles de entrenamiento son generalmente más precisos que sus correspondientes árboles de crecimiento (se puede ver con una prueba T y con un nivel de significancia de 0.1). Esto significa que los métodos que requieren conjuntos poda trabajan bajo desventaja. Es más, la tasa de mala clasificación de OPTT es siempre más baja que la tasa de error de OPGT.

La prueba t pareada se utiliza cuando se registran datos de un mismo individuo antes y después de la aplicación de un efecto que se desea analizar. Supóngase que se tienen x1, x2,…, xn observaciones de n individuos antes de aplicar un efecto; a continuación, se tiene y1, y2, …, yn observaciones luego de aplicar el mencionado efecto. Entonces se define: di = xi - yi, y se calculan x_d y s_d, es decir, el promedio y la desviación estándar de las diferencias, respectivamente. La estadística de prueba es $(\bar{x}_d - \mu_d)\sqrt{n}/s_d$ con distribución t con (n-1) grados de libertad.

En este trabajo de investigación, la prueba t pareada fue utilizada de la siguiente manera: el efecto a estudiarse fue el método de poda, y los individuos, los diferentes conjuntos de datos obtenidos en base a los tratamientos N1, N2, NT. Las observaciones que se registraron fueron las tasas de error que se generaron con los respectivos árboles de decisión.

Experimento

En esta sección, se discutirán los resultados de 840 diferentes experimentos de diferentes conjuntos y con diferentes métodos de poda. El primer factor que analizaremos en esta sección es la tasa de error de los árboles podados (simplificados). Como se estableció en líneas anteriores, nuestro objetivo es descubrir cuándo y cómo una estrategia sofisticada, la cual poda árboles de decisión inducidos a partir de datos, es mejor que una estrategia simple (ingenua), la cual no poda en absoluto, devolviendo el árbol tal cual T_{max}. Ambas estrategias pueden acceder a los mismos datos, pero la estrategia sofisticada puede tanto usar algunos datos para hacer crecer el árbol y el resto para podarlo, como explorar todos los datos de una sola vez para construir y podar el árbol de decisión. Por esta razón, probamos la significancia de diferencia en tasas de error entre los árboles de decisión podados y los árboles entrenados.

La sensibilidad para la elección del método de poda parece afectar dominios artificiales más que datos reales[Espo-97]. Ciertamente, en nuestro estudio, todos los conjuntos son propensos a la poda, mientras que los conjuntos que resultaron de un *oversampling* son resistentes a la poda.

Tal heurística incrementa la varianza de la tasa de error estimada, como se muestra en la Tabla 6.19(a)(b), en la cual se puede observar la tasa promedio de error junto con el error estándar para cada base de datos y cada método aplicado. El error más bajo es obtenido por el método de poda EBP y el más alto, en su mayoría, por CVP.

Vale la pena señalar que casi todas las bases de datos no propensas a la poda tienen el más alto porcentaje de error base (tabla 6.16) como son las bases construidas por *oversampling,* que tienen el 50 por ciento de error base, mientras que estas bases de datos con un relativo error base bajo se benefician de los métodos de poda. Parece que la técnica de poda solo produce mejoras orientadas al entendimiento de los árboles, pero no puede incrementar la precisión predictiva si ninguna clase domina sobre otra.

Conjunto		REP	MEP	PEP	CVP
Metodología	**Corte**				
Potencial NT	19952	18.7 ± 0.15	19.02 ± 0.19	18.71 ± 0.14	19.59 ± 0.14
Potencial N1	19952	18.6 ± 0.12	19.06 ± 0.15	18.71 ± 0.19	19.72 ± 0.19
Potencial N2	19952	18.63 ± 0.16	18.99 ± 0.18	18.74 ± 0.19	19.58 ± 0.16
Potencial NT	20021	19.68 ± 0.29	19.95 ± 0.17	19.62 ± 0.26	20.57 ± 0.24
Potencial N1	20021	19.55 ± 0.26	19.87 ± 0.19	19.53 ± 0.23	20.51 ± 0.33
Potencial N2	20021	19.64 ± 0.23	19.94 ± 0.21	19.67 ± 0.19	20.51 ± 0.26
Potencial NT	20041	19.19 ± 0.24	19.53 ± 0.28	19.94 ± 1.11	20.1 ± 0.34
Potencial N1	20041	19.13 ± 0.19	19.49 ± 0.32	20.53 ± 1.2	20.15 ± 0.31
Potencial N2	20041	19.21 ± 0.2	19.54 ± 0.32	20.15 ± 1.14	20.09 ± 0.24
Potencial NT	19952 - O	28.35 ± 0.24	28.91 ± 0.42	28.28 ± 0.53	28.91 ± 0.43
Potencial N1	19952 - O	28.43 ± 0.29	28.86 ± 0.23	27.9 ± 0.41	28.88 ± 0.25
Potencial N2	19952 - O	28.49 ± 0.3	29 ± 0.34	28 ± 0.37	28.95 ± 0.31

Tabla 6.19 (a). Promedio de tasa de errores para diferentes métodos de poda

Conjunto		0SE	1SE	EBP
Metodología	Corte			
Potencial NT	19952	18.73 ± 0.14	18.84 ± 0.13	18.57 ± 0.18
Potencial N1	19952	18.73 ± 0.13	18.83 ± 0.11	18.57 ± 0.18
Potencial N2	19952	18.71 ± 0.17	18.81 ± 0.16	18.55 ± 0.19
Potencial NT	20021	19.65 ± 0.25	19.78 ± 0.23	19.57 ± 0.19
Potencial N1	20021	19.5 ± 0.23	19.65 ± 0.24	19.49 ± 0.19
Potencial N2	20021	19.58 ± 0.23	19.77 ± 0.21	19.55 ± 0.21
Potencial NT	20041	19.15 ± 0.2	19.39 ± 0.3	19.05 ± 0.27
Potencial N1	20041	19.04 ± 0.3	19.18 ± 0.37	19.01 ± 0.32
Potencial N2	20041	19.19 ± 0.25	19.3 ± 0.3	19 ± 0.25
Potencial NT	19952 -O	29.07 ± 0.43	28.95 ± 0.51	28.14 ± 0.24
Potencial N1	19952- O	29.09 ± 0.23	28.98 ± 0.29	27.65 ± 0.25
Potencial N2	19952- O	29.18 ± 0.34	29.03 ± 0.38	27.95 ± 0.42

Tabla 6.19 (b). Promedio de tasa de errores para diferentes métodos de poda

La Tabla 6.20 reporta los resultados de las pruebas para un nivel de confianza igual a 0.10.

Conjunto		REP	MEP	PEP	CVP	0SE	1SE	EBP	Total	Total +	Total -
Metodología	**Corte**										
Potencial NT	19952	+	+	+	-	+	+	+	6 / 1	6	1
Potencial N1	19952	+	+	+	-	+	+	+	6 / 1	6	1
Potencial N2	19952	+	+	+	-	+	+	+	6 / 1	6	1
Potencial NT	20021	+	+	+	-	+	+	+	6 / 1	6	1
Potencial N1	20021	+	+	+	0	+	+	+	6 / 0	6	0
Potencial N2	20021	+	+	+	-	+	+	+	6 / 1	6	1
Potencial NT	20041	+	+	0	-	+	+	+	5 / 1	5	1
Potencial N1	20041	+	+	0	-	+	+	+	5 / 1	5	1
Potencial N2	20041	+	+	0	-	+	+	+	5 / 1	5	1
Potencial NT	19952 -O	-	-	0	-	-	-	0	0 / 5	0	5
Potencial N1	19952-O	-	-	0	-	-	-	+	1 / 5	1	5
Potencial N2	19952-O	-	-	0	-	-	-	0	0 / 5	0	5

Tabla 6.20. Tabla de significancia de la mejora de los métodos de poda

Un “+” en la tabla 6.20 significa que, en promedio, la aplicación del método de poda realmente mejora la precisión predictiva del árbol de decisión, mientras que un “-” indica un decrecimiento significante en la precisión predictiva. Cuando el efecto de la poda no es ni bueno ni malo, ello se indica con un 0.

A primera vista, podemos afirmar que, generalmente, la poda no hace decrecer la precisión predictiva. De modo más preciso, es posible dividir los conjuntos de datos en tres categorías principales: los propensos a la poda, los insensibles a la poda y los resistentes a la poda. Los más representativos de esta última categoría son, ciertamente, los conjuntos a los que se les aplicó *oversampling*. En este caso, casi todos los métodos que usan un conjunto de poda independiente producen significantemente árboles menos precisos, seguidos del conjunto que corresponde al corte 20041con sus tres formas de cálculo de potencial.

Los métodos que operan en el árbol de entrenamiento (PEP y EBP) parecen más apropiados para dominios complicados. Esto es evidente, ya que se puede observar que dichos métodos de poda trabajan mejor con los conjuntos correspondientes al corte 1995-2 con *oversampling*, verificándose que, en ningún caso, los métodos (PEP, EBP) empeoran la precisión predictiva (en el peor de los casos, la mantienen igual).

En general, una explicación sobre el comportamiento de los métodos de poda puede deducirse de la Tabla 6.20 comparando columnas más que filas. El número de bases de datos en las cuales cada método reportó un “+” o un “-” puede indicar la tendencia apropiada de cada método de poda para varios conjuntos considerados. Si

pensamos en los casos en los cuales observamos un cierto decrecimiento en la precisión, deberíamos concluir que CVP es el método con el peor rendimiento, seguido de PEP (considerando sólo el conjunto sin *oversampling*).

En este último caso, sin embargo, el número de "+" es también alto. Como esperábamos, se observó un comportamiento estático para PEP, el cual mejora su precisión en seis de los conjuntos y siguió igual en los seis restantes. Al menos cuatro métodos, REP, MEP, 0SE y 1SE, trabajan igual de bien, de modo que es posible postular su equivalencia a pesar de las diferencias en la formulación. Por esto, es posible afirmar que producen significantemente iguales árboles para el mismo conjunto de datos.

Resumiendo estos resultados, podemos concluir que no hay indicios de que los métodos que explotan un conjunto independiente de poda trabajen mejor que los otros.

Para completar el análisis de las tasas de error, observaremos si la significancia de diferencias entre los métodos lleva a una mejora. Debido a que no es posible reportar todas las posibles comparaciones, decidimos comparar uno de los métodos que parece el más estable, llamado EBP, con los otros.

El signo de valor P (P-Value) y el correspondiente nivel de significancia se muestran en la Tabla 6.21:

Conjunto		**REP**	**MEP**	**CVP**	**0SE**	**1SE**	**PEP**	**Total +**	**Total 0**	**Total -**
Tratamiento	Corte									
P. NT	19952	- (0.0019)	- (0)	- (0)	- (0.0133)	- (0.0002)	- (0.0128)	0	0	6
P. N1	19952	0 (0.3434)	- (0)	- (0)	- (0.002)	- (0.0002)	- (0.0013)	0	1	5
P. N2	19952	- (0.0107)	- (0)	- (0)	- (0.0084)	- (0.0002)	- (0.0002)	0	0	6
P. NT	20021	0 (0.1022)	- (0.0001)	- (0)	0 (0.1679)	- (0.0002)	0 (0.2443)	0	3	3
P. N1	20021	0 (0.3434)	- (0.0001)	- (0)	0 (0.8402)	- (0.0191)	0 (0.3732)	0	3	3
P. N2	20021	0 (0.2042)	- (0.0001)	- (0)	0 (0.5763)	- (0.0048)	- (0.0735)	0	2	4
P. NT	20041	- (0.0607)	- (0.0001)	- (0)	0 (0.1582)	- (0.0046)	- (0.0263)	0	1	5
P. N1	20041	0 (0.1188)	- (0.0001)	- (0)	0 (0.5414)	- (0.0095)	- (0.0031)	0	2	4
P. N2	20041	- (0.0024)	- (0)	- (0)	- (0.0025)	- (0.0009)	- (0.0114)	0	0	6
P. NT	19952-O	- (0.0124)	- (0)	- (0)	- (0)	- (0.0001)	0 (0.368)	0	1	5
P. N1	19952-O	- (0.0001)	- (0)	- (0)	- (0)	- (0)	- (0.0729)	0	0	6
P. N2	19952-O	- (0.0044)	- (0)	- (0)	- (0)	- (0)	0 (0.6262)	0	1	5
Total +		0	0	0	0	0	0			
Total 0		5	0	0	5	0	4			
Total -		7	12	12	7	12	8			

Tabla 6.21. Nivel de significancia de la diferencia entre EBP y los otros métodos de poda

Un signo +(-) en la intersección de la i-ésima fila y la j-ésima columna indica que el EBP obtiene mayores (menores) porcentajes de error cuando poda que el método en la j-ésima columna para la base de datos en la i-ésima fila.

Los valores que aparecen en paréntesis representan los p-values de las pruebas t pareada. Un p-value es la probabilidad de obtener un valor más extremo (más pequeño o más grande) que el observado. Es decir: P-value = P (|t |>) $(\bar{x}_d - \mu_d)\sqrt{n}/s_d$ en una distribución t con (n-1) grados de libertad. De una rápida mirada en la tabla, podemos concluir que EBP siempre es mejor que los otros métodos (Tabla 6.18).

En la prueba t pareada, la hipótesis nula es μd = 0, es decir, no hay diferencia significativa entre EBP y REP (MEP, CVP,…). Si se calculan las diferencias di = xi - yi, donde los xi representan las tasas de error de EBP y yi las tasas de error de un método de poda, y si además se rechaza la hipótesis nula con $\mu_d < 0$, se concluirá que EBP es mejor por contar con una menor tasa promedio de error. Así como en la tasa de error, para el tamaño de los árboles nuevamente testeamos el nivel de significancia de las diferencias por medio de la prueba pareada de dos colas con T-Test.

La Tabla 6.22 resume los resultados cuando el nivel de confianza es 0.10. Debe considerarse que la comparación involucra OPGT para aquellos que operan en el conjunto de *pruning*, y OPTT para los otros.

Conjunto		REP	MEP	CVP	0SE	1SE	PEP	EBP	Total o	Total u	Total -
Metodología	**Corte**										
Potencial NT	19952	-	u	u	o	o	o	o	4	2	1
Potencial N1	19952	-	u	u	o	o	o	o	4	2	1
Potencial N2	19952	-	u	u	o	o	o	o	4	2	1
Potencial NT	20021	-	u	u	o	o	o	o	4	2	1
Potencial N1	20021	-	u	u	o	o	o	o	4	2	1
Potencial N2	20021	-	u	u	o	o	o	o	4	2	1
Potencial NT	20041	-	u	u	o	o	o	o	4	2	1
Potencial N1	20041	-	u	u	o	o	o	o	4	2	1
Potencial N2	20041	-	u	u	o	o	o	o	4	2	1
Potencial NT	19952-O	u	u	u	u	u	u	u	0	7	0
Potencial N1	19952-O	u	u	u	u	u	u	u	0	7	0
Potencial N2	19952-O	u	u	u	u	u	u	u	0	7	0
Total o		0	0	0	9	9	9	9			
Total u		3	12	12	3	3	3	3			
Total -		9	0	0	0	0	0	0			

Tabla 6.22. Prueba para el tamaño de árbol

Aquí “u” se establece para una subpoda significante, “o” para una sobrepoda significante y “-” para indicar que no hay diferencias relevantes estadísticamente. La prueba confirma que MEP, CVP tiende a subpodar, y que las pruebas 0SE, 1SE, PEP y EBP tienden a sobrepodar, mientras que el método REP no evidencia ni una sobrepoda ni una subpoda.

La Tabla 6.18(a-b) muestra en forma detallada el tamaño promedio de los árboles podados óptimamente (OPGT, OPTT). Puede ser virtualmente partida en dos subtablas. La tabla 6.18(a) se usará en una comparación de métodos de poda que operan en los árboles de crecimiento, y la tabla 6.23 (b), basada en los métodos que podan con los árboles de entrenamiento.

Conjunto		**REP**	**MEP**	**CVP**
Metodología	Corte			
Potencial NT	19952	2650.1 ± 229.66	3577 ± 500.49	8605.8 ± 376.69
Potencial N1	19952	3009.3 ± 273.49	4122 ± 387.42	9884.6 ± 738.79
Potencial N2	19952	2735.3 ± 323.81	3829.8 ± 275.18	9068.8 ± 427.56
Potencial NT	20021	1542.8 ± 351.93	2080.6 ± 290.39	4941.6 ± 295.42
Potencial N1	20021	1926.6 ± 357.62	2460.2 ± 227.63	5600 ± 267.12
Potencial N2	20021	1602.3 ± 304.13	2093.7 ± 168.44	5085.9 ± 228.24
Potencial NT	20041	1322.1 ± 193.47	1875.5 ± 237.89	4017.3 ± 206.11
Potencial N1	20041	1620.2 ± 233.75	2125.7 ± 292.39	4687.8 ± 405.13
Potencial N2	20041	1374.5 ± 231.16	1974.4 ± 184.41	4347.3 ± 206.3
Potencial NT	19952-O	15406.5 ± 762.47	19584.7 ± 914.78	23702.4 ± 974.63
Potencial N1	19952-O	16747.6 ± 715.96	21007.9 ± 784.17	26061 ± 1237.61
Potencial N2	19952-O	16463.6 ± 506.96	20764.6 ± 736.84	25149.3 ± 922.16

Tabla 6.23 (a). Prueba para el tamaño de árbol (REP,MEP,CVP)

Así en la Tabla 6.23(a)(b) encontramos una confirmación de que REP tiende a sobrepodar debido a que, en seis de las nueve bases de datos consideradas (sin *oversampling*), esto es, para los tratamientos N2 y NT y para los tres cortes considerados, se producen árboles con un menor tamaño promedio que aquellas obtenidas por el OPGT. La explicación de esta tendencia puede atribuirse al hecho de que la decisión para podar una rama se basa solo en la evidencia del conjunto de poda.

Conjunto		0SE	1SE	OPGT
Metodología	Corte			
Potencial NT	19952	494.6 ± 327.7	44.6 ± 51.54	2661.9 ± 281.14
Potencial N1	19952	795.4 ± 573.13	25 ± 4.71	2952.5 ± 302.61
Potencial N2	19952	590.1 ± 401.37	49.7 ± 45.5	2810.8 ± 298.89
Potencial NT	20021	281.5 ± 328.02	45.2 ± 80.77	1695.1 ± 178.7
Potencial N1	20021	274.5 ± 235.81	50.4 ± 64.87	1890.6 ± 242.08
Potencial N2	20021	313.9 ± 236.66	58.7 ± 88.77	1753.7 ± 230
Potencial NT	20041	432.9 ± 224.11	63.7 ± 58.38	1368.4 ± 194.38
Potencial N1	20041	379.4 ± 226.14	94.5 ± 64.56	1613.8 ± 271.51
Potencial N2	20041	432.3 ± 245.3	93.9 ± 67.18	1453.1 ± 121.03
Potencial NT	19952-O	20824.5 ± 1031.75	16890 ± 624.21	3672.6 ± 454.48
Potencial N1	19952-O	22490.2 ± 966.66	18697.3 ± 995	4077.6 ± 359.52
Potencial N2	19952-O	21953.2 ± 619.79	17898.4 ± 1028.89	4093 ± 440.44

Tabla 6.23 (b). Prueba para el tamaño de árbol (0SE,1SE,OPGT)

En la Tabla 6.23 (c), confirmamos que PEP y EBP tienden a sobrepodar debido a que, en todas las nueve bases de datos consideradas (sin *oversampling*), se producen árboles con un menor tamaño promedio que aquellas obtenidas por el OPTT.

Conjunto		PEP	EBP	OPTT
Metodología	**Corte**			
Potencial NT	19952	539.4 ± 130.51	1590.3 ± 240.77	3237.4 ± 249.11
Potencial N1	19952	766.2 ± 182.11	1850.8 ± 274.43	3642.9 ± 483.15
Potencial N2	19952	633.6 ± 197.86	1648.3 ± 197.88	3441.2 ± 363.05
Potencial NT	20021	402.4 ± 107.36	1128.1 ± 288.18	2188.7 ± 203.31
Potencial N1	20021	433.2 ± 115.98	1284 ± 234.34	2530.3 ± 245.52
Potencial N2	20021	400.6 ± 87.76	1244.1 ± 157.38	2091.3 ± 185.14
Potencial NT	20041	250.2 ± 180.69	773.3 ± 155.71	1559.4 ± 245.08
Potencial N1	20041	145.6 ± 192.49	864.8 ± 152.29	2000.9 ± 225.95
Potencial N2	20041	244.1 ± 231.18	740.7 ± 123	1739.8 ± 262.18
Potencial NT	19952-O	16429.5 ± 1364.27	24775.3 ± 1156.98	4951.4 ± 584.63
Potencial N1	19952-O	17195 ± 1116.37	26032.4 ± 1339.65	5866.4 ± 743.14
Potencial N2	19952-O	16924.5 ± 1637.5	25924.2 ± 967.56	5762.9 ± 533.32

Tabla 6.23 (c). Prueba para el tamaño de árbol

Conclusión:

Después de este experimento, podemos concluir:

- Las técnicas de poda solo producen mejoras orientadas al entendimiento de los árboles, pero no pueden incrementar la precisión predictiva si ninguna clase domina sobre otra.

- Podríamos concluir que CVP es el método con el peor rendimiento, seguido de PEP.
- Al menos cuatro métodos, REP, MEP, 0SE y 1SE, trabajan igual de bien, de modo que es posible postular su equivalencia a pesar de las diferencia en la formulación.
- No hay indicios de que los métodos que explotan un conjunto independiente de poda trabajen mejor que los otros.
- En general, el método de poda EBP es el que menor tasas de error proporciona.

6.2.4. Análisis de las técnicas de clasificación por conjuntos: Bagging y Boosting

Objetivo

El presente experimento busca determinar que las técnicas de clasificación por conjuntos (como Bagging y Boosting) tienen un mejor desempeño que las técnicas base en cada tratamiento de cálculo de potencial. Para ello se usarán los resultados de las tres fases anteriores, es decir, se usará como algoritmo base C4.5 y los tres tratamientos de cálculo de potencial N1, N2 y NT (Fase 1). Además, utilizará el conjunto de datos desde el período 19912 y el método de poda EBP tal como se confirmó en las fases dos y tres respectivamente.

Procedimiento

Aplicamos la técnica de Holdout Resampling a los datos del corte 19912 para generar un conjunto que consta de 70 por ciento de datos de entrenamiento y 30 por ciento de datos de prueba. Esto nos permitirá generar los modelos tanto para el algoritmo base C4.5 como para los conjuntos de clasificadores.

Para estos últimos, se definieron veinticinco iteraciones (modelos) para el algoritmo de Bagging y diez iteraciones para el de Boosting. Todo el proceso descrito anteriormente se realiza diez veces.

Experimento

Con este experimento, se pretende responder a la pregunta: ¿es mejor el modelo obtenido a partir de un conjunto de clasificadores que uno obtenido a partir de un clasificador individual? Para responder a esta pregunta, se utilizan las tasas de error obtenidas al aplicar Bagging, Boosting y el algoritmo base C4.5 a todos los datos.

Split	N1	N2	NT
1	18.7638	18.369	18.9189
2	18.7225	18.8651	18.9395
3	18.6398	18.5199	18.6315
4	18.5468	18.7163	18.7659
5	18.8548	18.8258	18.5819
6	18.6481	18.8155	18.7225
7	18.6997	18.7886	18.6873
8	18.8424	18.5881	18.6543
9	18.7494	18.799	18.6481
10	18.677	18.7473	18.8134
	18.71±0.09	18.7±0.16	18.74±0.12

Tabla 6.24. Tasas de error de C4.5

Split	**N1**	N2	NT
1	18.3504	18.2305	18.7204
2	18.6109	18.5013	18.6233
3	18.2057	18.1705	18.3483
4	18.4269	18.5034	18.6481
5	18.5984	18.5178	18.3731
6	18.4889	18.4351	18.5592
7	18.2925	18.4661	18.4124
8	18.7618	18.3669	18.3711
9	18.4786	18.5984	18.3876
10	18.5116	18.524	18.584
	18.47±0.16	18.43±0.14	18.5±0.14

Tabla 6. 25 (a). Tasa de error de Bagging

Split	N1	N2	NT
1	19.6176	19.3488	19.6693
2	19.5452	19.7292	19.9711
3	19.5059	19.4398	19.5928
4	19.4894	19.6093	19.7623
5	19.6837	19.816	19.3282
6	19.3116	19.7251	19.6072
7	19.601	19.5245	19.6693
8	19.6858	19.2951	19.3344
9	19.5969	19.7168	19.6734
10	19.5225	19.5659	19.7891
	19.56±0.11	19.58±0.18	19.64±0.2

Tabla 6.25 (b). Tasa de error de Boosting

Se procede a determinar estadísticamente qué algoritmo tiene menor tasa de error promedio. Para ello se procede a realizar una prueba t pareada a los resultados de la tabla 6.24 y a los de las tablas 6.25(a) y 6.25(b). En la tabla 6.26., se puede observar una descripción de las nueve pruebas realizadas.

		N1	**N2**	**NT**
C4.5	vs. Bagging	**+ (0.0003)**	**+ (0)**	**+ (0)**
C4.5	vs. Boosting	- **(0)**	**- (0)**	**- (0)**
Bagging	vs. Boosting	- **(0)**	**- (0)**	**- (0)**

Tabla 6.26. Resultados de la prueba t-pareada entre algoritmos

Interpretación:

Los resultados indican que la tasa promedio de error disminuye al usar el clasificador Bagging con respecto a las tasas de error obtenidas tanto para el algoritmo base C4.5 como para el de Boosting. Además, se puede observar que la tasa promedio de error del algoritmo base C4.5 es menor que la obtenida con el clasificador Boosting.

Conclusión:

De acuerdo con los resultados obtenidos, se concluye que, al aplicar la técnica de clasificación por conjunto Bagging, se logra disminuir significativamente las tasas de error en cada uno de los tratamientos en comparación con el uso de otros algoritmos.

6.2.5. Análisis de las técnicas de clasificación propuestas: Bag-E y Bag-P

Objetivo

El presente experimento busca determinar la eficacia de los métodos propuestos (Bag-E y Bag-P) para nuestro dominio de aplicación en comparación con los mejores métodos encontrados en las fases anteriores. Dicho en otras palabras, se comparan los resultados de cada uno de los métodos propuestos con los resultados que se obtienen de la aplicación del Bagging con 25 iteraciones sobre el algoritmo base de C4.5 con un número mínimo de instancias en las hojas de 40 y un factor de confianza 0.40. Para este experimento, solo se considera el tratamiento de potencial N1 y se utiliza como conjunto base el conjunto original de datos desde el período 19912. Aparte de esto, se busca encontrar una métrica que aproxime el rendimiento de dichos clasificadores en el caso de que se considere que los costos de los errores obtenidos en la clasificación sean distintos. Para ello se hace uso de los resultados obtenidos con la métrica AUC (área bajo la curva ROC).

Procedimiento

En este experimento, se han considerado las siguientes reglas para la formación de los conjuntos según la técnica que fue aplicada:

Para la aplicación del experimento con la técnica de Bagging, se usó la técnica de Holdout Resampling a los datos del corte 19912 para generar un conjunto que consta de 70 por ciento de datos de entrenamiento y 30 por ciento de datos de prueba.

- Para el Bag-E, usamos como conjuntos de entrada los datos correspondientes a cada año, según el pseudocódigo de la figura 4.17.
- Para el Bag-P, usamos como conjuntos de entrada los definidos según el pseudocódigo de la figura 4.19., es decir, los que corresponden a cada corte.

Adicionalmente se introduce un estudio comparativo entre los resultados que arrojan las técnicas propuestas y los resultados de la técnica de Bagging encontrados en la fase de experimentación anterior.

Experimento

6.2.5.1. Experimento relativo al método BAG-E.

Como se explicó en la propuesta del algoritmo en la sección 4.4.7, el método BAG-E está basado en los métodos de conjuntos de clasificadores que tienen la tarea de predecir las clases de nuevos ejemplos, combinando las decisiones individuales de los clasificadores provenientes de conjuntos que resultan de estratificar la base de datos original. Los conjuntos estratificados, son determinados, como se explicó anteriormente, dividiendo la base de datos total por años y considerando cada año como un estrato.

Par llevar a cabo esta prueba, se elijen aleatoriamente tantos subconjuntos como se desea y en seguida se aplica el algoritmo de C4.5(con las condiciones antes establecidas) a cada uno de ellos con el objetivo de crear un modelo(hipótesis) por subconjunto. Una vez obtenido el modelo, se aplica votación sobre un conjunto de prueba como se puede observar en la figura 6.6.

En la figura 6.6, se puede observar que el conjunto inicial de datos se ha dividido en varios subconjuntos (dieciocho en nuestro caso). Luego se selecciona aleatoriamente uno de ellos para que sea considerado como conjunto de prueba y se toma cada subconjunto restante como conjunto de entrenamiento.

Después de esta selección, se aplica el algoritmo de clasificación a cada subconjunto con el objetivo de obtener una hipótesis. Luego de obtenida dicha hipótesis, se clasifica el conjunto de prueba en cada submodelo, y se obtiene una clasificación por cada instancia para cada modelo, las que son utilizadas para la votación. Luego de la votación y determinada la clase final para cada instancia, se obtiene la precisión del modelo.

Este proceso representa una sola ejecución de la técnica de Bag-E. En nuestro caso, para que sea posible comparar esta nueva propuesta con las técnicas clásicas, se validará repitiendo este proceso diez veces con conjuntos de prueba distintos.

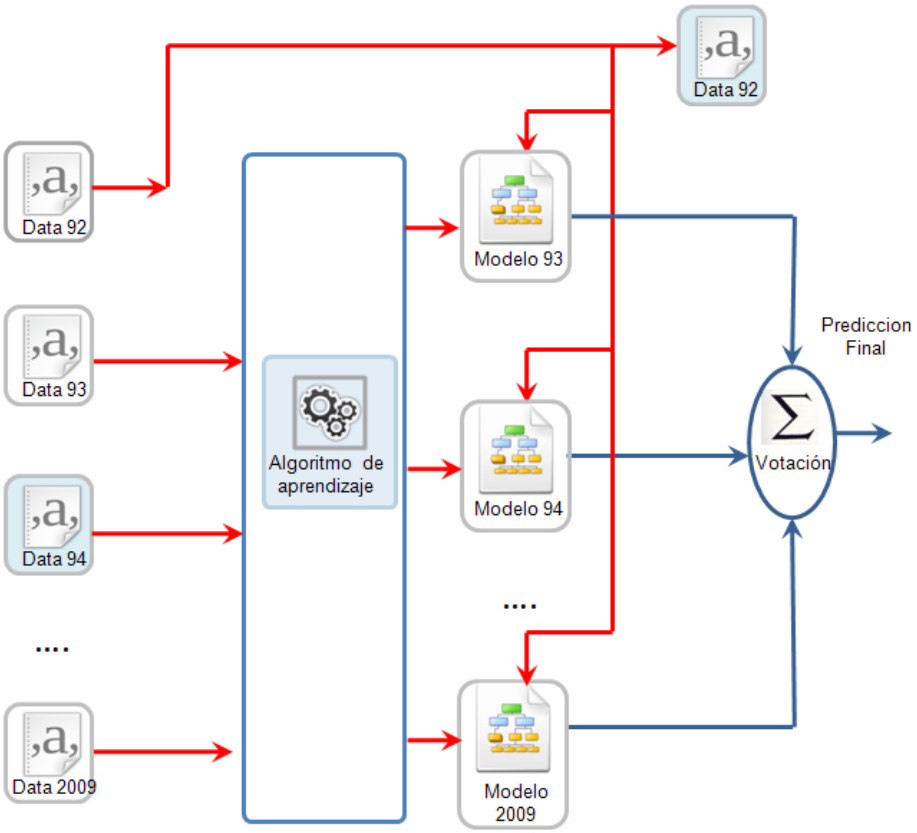

Figura 6.6. Clasificador mediante conjunto (BAG-E)

Los resultados obtenidos al aplicar dicho algoritmo a nuestro conjunto de datos son:

BAG-E.	ACC	ROC
Testing95	0.81995	0.78784
Testing96	0.83867	0.81208
Testing97	0.84575	0.80527
Testing98	0.83482	0.80519
Testing99	0.81339	0.80382
Testing01	0.80673	0.80765
Testing04	0.80971	0.79211
Testing05	0.81387	0.76618
Testing06	0.80945	0.77817
Testing09	0.82073	0.80047
Promedio	0.8213±0.0137	0.7959±0.0147

Tabla 6.27. Resultados finales de la aplicación de algoritmo BAG-E

Como se puede observar en la tabla 6.27, se presentan los resultados usando las métricas de precisión y el área bajo la curva ROC.

Es importante recordar que uno de los objetivos de esta investigación es encontrar métodos que sirvan para una mejor precisión en la recomendación. Por ello, se ha hecho un estudio comparativo, usando análisis de varianza (ANOVA), entre los métodos propuestos y el resultante de las fases de experimentación anterior, para determinar la técnica que proporcione mejores resultados en la recomendación.

Cabe recordar que el algoritmo base para la aplicación de BAG-E es el C4.5 y que su configuración fue de 40 instancias como máximo en las hojas y un factor de confianza de 0.4.

6.2.5.2. Experimento relativo al método BAG-P.

Como se explicó en la sección 4.4.7, el método BAG-P está basado en los métodos de clasificación por conjuntos, que tiene la tarea de clasificar nuevos ejemplos combinando las decisiones individuales de los clasificadores provenientes de conjuntos que resultan de introducir una perturbación al conjunto de entrenamiento. Dicha perturbación está basada en la extracción de ciertos subconjuntos con alguna condición establecida, que en nuestro caso representan los datos correspondientes a un semestre académico.

Para llevar a cabo esta prueba, se elijen aleatoriamente tantos subconjuntos como se desea (figura 6.7) y en seguida se aplica el algoritmo de C4.5 (con las condiciones antes establecidas) a cada uno de ellos con el objetivo de crear un modelo (hipótesis). Cabe mencionar que, antes de aplicar el algoritmo de aprendizaje, se tuvo que seleccionar el conjunto de prueba para que dichos datos sean extraídos de los respectivos subconjuntos de entrenamiento con el objetivo de evitar el sobreajuste.

En la figura 6.8, se puede observar que el conjunto inicial de datos se ha dividido en varios subconjuntos (veintiuno en nuestro caso). Cada uno de ellos es entrenado con el objetivo de obtener una hipótesis. Luego de obtenida dicha hipótesis, se clasifica el conjunto de prueba en cada uno con la finalidad de aplicar la respectiva votación. Con este proceso, se obtiene una precisión determinada. En nuestro caso, para que sea posible comparar esta nueva propuesta con las técnicas clásicas, se validará repitiendo este proceso diez veces con todos los conjuntos de prueba seleccionados.

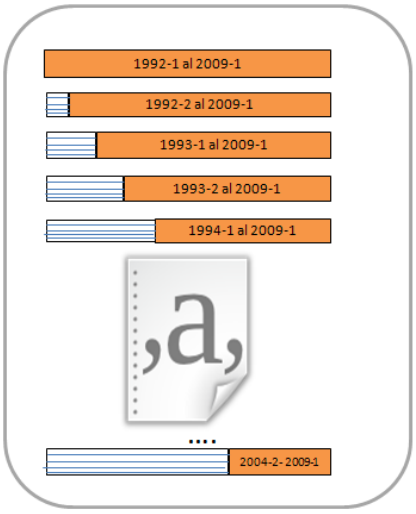

Figura 6.7. División del conjunto total de datos por cortes

En la figura 6.8, se puede observar que, por ejemplo, se ha elegido el conjunto 20062 como conjunto prueba (en alguna corrida). Cada vez que se elige un conjunto de prueba (algún semestre académico), este es extraído de cada uno de los cortes para luego ser sometidos al algoritmo de aprendizaje.

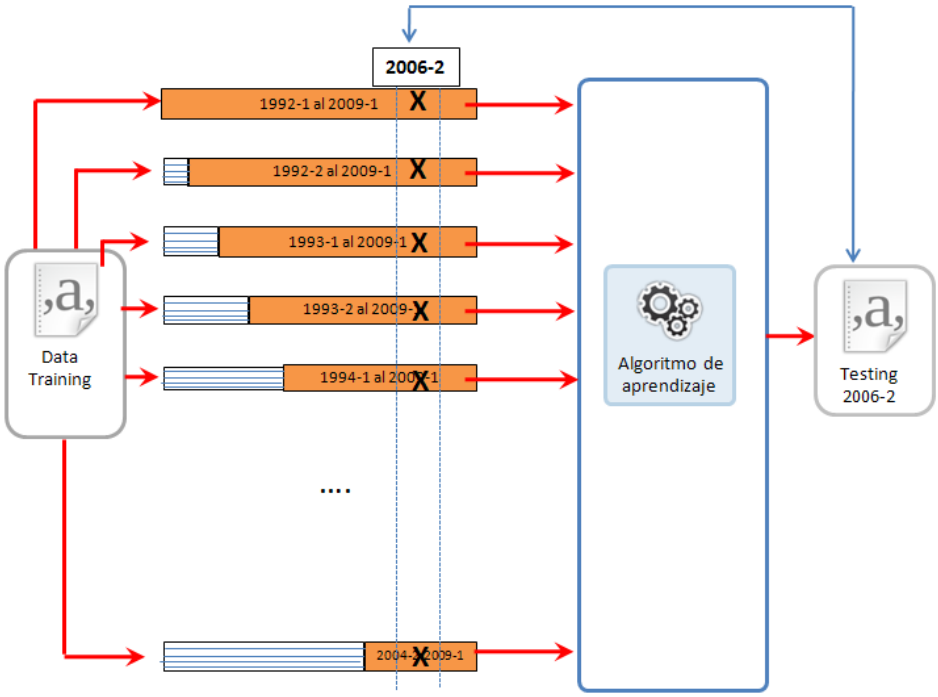

Figura 6.8. Experimento que usa el método BAG-P

Los resultados obtenidos al aplicar dicho algoritmo a nuestro conjunto de datos son:

BAG-P.	ACC	ROC
Testing 20042	0.82529	0.80026
Testing 20051	0.82286	0.78756
Testing 20052	0.81075	0.77324
Testing 20061	0.81485	0.79103
Testing 20062	0.80905	0.79201
Testing 20071	0.81349	0.7985
Testing 20072	0.80662	0.80668
Testing 20081	0.80998	0.79739
Testing 20082	0.81832	0.81281
Testing 20091	0.82314	0.8089
Promedio	0.8154±0.0066	0.7968±0.0116

Tabla 6.28. Resultados finales de la aplicación de algoritmo BAG-P

En el anexo A.3.5., se hace una prueba estadística de análisis de varianza, y nuestra propuesta BAG-E resulta como la más precisa, y la segunda propuesta BAG-P, como igualmente buena que los resultados de los mejores algoritmos clásicos.

Experimento comparativo

Con este experimento se pretende responder a la pregunta: ¿qué tan eficientes son los modelos propuestos en comparación con la mejor técnica encontrada en la fase de experimentación anterior? Para responder a esta pregunta, se utilizan las tasas de error y el área bajo la curva (AUC), obtenidas al aplicar Bagging, BAG-E y BAG-P.

Estas se pueden observar en las tablas 6.29 y 6.30 respectivamente:

Accuracy	BAG-E.	BAG-P.	Bagging_opcion_25
1	18.005	17.471	18.3504
2	16.133	17.714	18.6109
3	15.425	18.925	18.2057
4	16.518	18.515	18.4269
5	18.661	19.095	18.5984
6	19.327	18.651	18.4889
7	19.029	19.338	18.2925
8	18.613	19.002	18.7618
9	19.055	18.168	18.4786
10	17.927	17.686	18.5116
	17.8693±1.37	18.4565±0.6619	18.4726±0.1637

Tabla 6.29. Resultados comparativos de tasa de error

Roc	BAG-E.	BAG-P.	Bagging_opcion_25
1	0.78784	0.80026	0.79928
2	0.81208	0.78756	0.80612
3	0.80527	0.77324	0.80909
4	0.80519	0.79103	0.79686
5	0.80382	0.79201	0.80333
6	0.80765	0.7985	0.80486
7	0.79211	0.80668	0.80448
8	0.76618	0.79739	0.80369
9	0.77817	0.81281	0.80203
10	0.80047	0.8089	0.80039
	0.7959±0.0147	0.7968±0.0116	0.803±0.0035

Tabla 6.30. Resultados comparativos con la métrica AUC

Se procede a determinar estadísticamente qué algoritmo tiene menor tasa de error promedio. Para ello se aplica una prueba de análisis de varianza (ANOVA) a los resultados de las tablas 6.29 y 6.30 respectivamente. En la tabla 6.31., se pueden observar los resultados de la prueba:

Métrica	P-value	Hay diferencia significativa entre los algoritmo
Precisión	0.238	No
AUC	0.305	No

Tabla 6.31. Resultado de la prueba estadística de ANOVA

Capítulo 7:

Conclusiones y trabajo futuro

En este capítulo, se presentan los principales resultados de la investigación, las consideraciones realizadas al respecto, las limitaciones de la propuesta, las futuras líneas de trabajo y un listado de las publicaciones a las que ha dado lugar el desarrollo de la presente memoria.

7.1. Generalidades

Este trabajo de investigación surgió con el propósito de ayudar a los estudiantes en el proceso regular de matrícula proporcionándoles un criterio adicional en la toma de sus decisiones. Para ello se propuso una metodología, aprovechable por cualquier institución de educación superior, cuyo objetivo es preparar los datos académicos de los alumnos, a fin de darles la forma adecuada para que puedan ser tratados mediante técnicas del descubrimiento del conocimiento. De esa manera, se logró hacer predicciones sobre el rendimiento académico a partir de los datos históricos de estudiantes con características similares.

La línea más importante de esta investigación, por lo tanto, fue la adquisición del conocimiento sobre el rendimiento académico de los estudiantes en ciertas asignaturas, con el objetivo de ponerlo a disposición de aquellos que vayan a cursarlas en el futuro. De esa manera, estos contarán con un elemento de juicio adicional, basado en una predicción, durante su matrícula.

En ese sentido, los sistemas de recomendación tuvieron un rol fundamental en esta investigación, pues estos se vinculan estrechamente con dicho propósito, en tanto y en cuanto la técnica utilizada en el motor de recomendación es la que marca su precisión y efectividad. Dentro de ese marco conceptual, la investigación, usando datos reales, se concentró en la experimentación de la aplicación de técnicas y herramientas de minería de datos.

Particularmente, se analizaron los datos reales provenientes de las matrículas de los estudiantes de la Facultad de Ingeniería de Sistemas de la Universidad de Lima desde su creación. El análisis se llevó a cabo utilizando técnicas de filtrado colaborativo basadas en memoria y en modelos, y en estas últimas se aplicaron técnicas tales como árboles de decisión (DT), clasificadores de Naïve Bayes (NB), vecinos cercanos (KNN), Decision Stumps y técnicas de clasificación mediante conjuntos (Bagging y Boosting),

con el propósito de descubrir tendencias y patrones de soporte para una correcta toma de decisión cuando, en el momento de la matrícula y en un determinado semestre académico, se busque elegir una asignatura como parte de un itinerario académico.

Asimismo, la presente investigación incluyó una descripción detallada de los diferentes pasos, basados en KDD, necesarios para hacer una correcta predicción, orientada al tipo de datos que se podrían obtener de cualquier institución académica. Así se pudo encontrar que, mediante estas técnicas, es posible construir un modelo que represente el comportamiento de estudiantes en su paso por diversas asignaturas. La construcción de este modelo facilitó una adecuada visión del comportamiento y rendimiento del grupo de estudiantes en una determinada carrera universitaria. Al mismo tiempo, permitió cumplir con el objetivo, que era el ayudar al alumno en su matrícula.

7.2. Conclusiones

Para enfrentar el propósito inicial y realizar los procesos de análisis y experimentación mencionados en el apartado anterior, se formularon dos propuestas. La primera estuvo centrada en la metodología de preparación de los datos. Estos, que se presentaban llenos de ruido y, en muchos casos, sin utilidad para las predicciones, resultaron expeditos luego de la aplicación de la metodología. La segunda estuvo enfocada en los métodos de aprendizaje automático que, basados en minería de datos, permitieron clasificar instancias a fin de predecir eficientemente situaciones futuras mediante datos históricos. A continuación, el examen de ambas propuestas permitirá precisar las conclusiones acuñadas a lo largo de la tesis.

7.2.1. Sobre la metodología de creación y preparación de los datos

Para la consecución del primer objetivo, es decir, la creación y propuesta de una metodología de preparación de datos basada en el proceso de KDD, debió considerarse su importancia primordial como subproceso. Efectivamente, se pudo observar que la precisión predictiva dependería, en gran parte, de la calidad de los datos que resultaran de él. Por ello, el principal aporte en esta etapa de la investigación fue incluir en el proceso de preparación de datos un subproceso de creación de dos variables sintéticas, que en este trabajo se llaman “dificultad” y “potencial”. Como se explica en el capítulo 5, la generación de estas dos variables, que fueron las que dotaron a cada instancia y al conjunto de ellas del conocimiento estático sobre las particularidades de nuestro dominio de aplicación, se sumó a la generación de un proceso estándar de limpieza de datos. Ambas fueron calculadas y se agregaron al conjunto original como si fueran atributos o características naturales de los estudiantes.

En primer lugar, la dificultad de un curso es un número que mide cuán fácil o difícil ha sido la asignatura; por ello, depende únicamente de las evaluaciones ponderadas de cada estudiante por asignatura según el conjunto de datos que se tomen. Por ejemplo, si una asignatura ha sido cursada por varios estudiantes durante algún período determinado, entonces su dificultad se asignará al registro que corresponde a cada uno de los que vayan a cursar esta asignatura. Con este atributo adicional, se logró un solo valor para cada una de las asignaturas ofertadas[1].

El segundo atributo sintético, el potencial, se definió como la caracterización que identifica al alumno en cada asignatura que le corresponda cursar y, más específicamente, como el valor numérico que mide las capacidades y habilidades de un estudiante para enfrentar determinado curso, el cual se establece mediante su rendimiento desde que ingresó hasta el momento en que solicitó la asesoría. En esta investigación, se propusieron varios tipos de potencial que dependen de las distancias entre las asignaturas que son sus requisitos: los que consideran un solo requisito (N1), dos requisitos (N2), todos los cursos requisitos (NT) y todas las asignaturas correspondientes a su plan curricular (potencial PPA). Debido a que cada asignatura se relaciona con el conjunto de sus requisitos, esta información se utilizó a fin de saber con qué conocimiento se presentaba el estudiante al curso y, sobre todo, cuál había sido su desempeño anterior en aquellas asignaturas básicas, lo que marcaba una tendencia relevante para las asignaturas futuras.

Finalmente, luego de proponer esta metodología como parte del proceso de preparación de datos en el ámbito de la educación superior, se concluyó que, en este dominio de aplicación, no basta con la limpieza del conjunto de datos primitivos, que son los que cualquier institución almacena en sus registros y los que cualquier proceso de minería se encargaría de filtrar, sino que, adicionalmente, deben considerarse otros que provengan de una combinación de datos históricos y de las propias reglas de la institución. Entre ellos se encontró, por ejemplo, requisitos de asignaturas, áreas académicas, malla curricular, modificaciones curriculares, creditaje, equivalencias, etcétera.

7.2.2. Sobre el método propuesto para el aprendizaje automático

Para explicar las conclusiones derivadas del diseño y la aplicación de los métodos de aprendizaje automático propuestos en el presente trabajo de investigación, antes se debe hacer referencia a su dominio de aplicación, es decir, a los datos que se generan cuando un estudiante interacciona dentro del sistema de educación superior. En

[1] También se experimentó con diferentes valores, uno por semestre académico.

él se observa que los datos son dinámicos y muy cambiantes, debido a que la información es gobernada por un conjunto de reglas que se pueden clasificar, según su flujo y sus valores, en dos tipos: las que dependen de la universidad y las que dependen del factor humano (profesores o estudiantes).

Las reglas que dependen de la universidad son aquellas que influyen directamente en los datos y podrían, en caso de que no se tomen las medidas necesarias, hacer cambiar los patrones correspondientes al conjunto de datos históricos. Dentro de este tipo de reglas están las modificaciones curriculares, los períodos académicos, los nuevos estudiantes, los que egresan, los que abandonan, etcétera. Como se demuestra claramente en esta memoria de investigación, la regularidad en la precisión predictiva relativa al tiempo[2]acaso indica que esta tendencia se debe mantener para el futuro. Esto permite afirmar que no existe un subconjunto de datos que sea el que, una vez analizado, proporcione mejor precisión predictiva. Por lo expuesto, el presente trabajo propone el uso de la totalidad de los datos para el respectivo análisis.

Lo más probable, en un contexto como aquel, era encontrar diferencias significativas entre los períodos. Sin embargo, debido a la metodología de preparación de datos propuesta, dicha posibles diferencias se estabilizaron en el tiempo. Esto debido a que existió el cuidado de incluir tanto la equivalencia entre las asignaturas establecidas en las normas de modificación curricular, como la respectiva eliminación de registros que representaban asignaturas que dejarían de tomarse en cuenta en el nuevo plan. Si bien es cierto que, en algunos períodos académicos, hubo ligeros cambios de tendencias en los porcentajes de error de predicción, debe considerarse que este factor presenta un índice de perturbación tan leve que no influye sobre el aprendizaje de patrones.

Las reglas que dependen del factor humano son aquellas que no se pueden medir ni cuantificar, es decir, los aspectos inherentes al estudiante, quien es el que obtiene la nota, y los inherentes al docente, quien califica al el desempeño del estudiante.

Esta última dimensión resultó ser una de las limitaciones más grandes del trabajo de investigación, debido a que poco a casi nada que provino de este tipo de reglas se pudo cuantificar, siendo estas las que causan la mayor cantidad de ruido en los datos Efectivamente, después de toda la experimentación que se ha llevado a cabo, se obtuvo la evidencia de tres situaciones que eventualmente serían las causantes de generarlo. En primer lugar, estudiantes cuyo rendimiento académico no proporciona una regla fija, debido a que en un semestre alcanzan evaluaciones sobresalientes y, en el siguiente, evaluaciones que no guardan relación con sus datos históricos. En segundo lugar, existe una cierta variabilidad en las matrículas, dado que no todos los estudiantes llevan una

[2] Cuando se pretenda incluir un período académico al estudio, los datos de este nuevo período son agregados a la base de datos.

misma cantidad de asignaturas por cuatrimestre, pues ello depende única y exclusivamente de su decisión.

En tercer lugar, se han detectado patrones de desniveles en notas diferenciadas por docentes. Si bien es cierto que el número que corresponde al atributo nota de asignatura proviene de un promedio entre las notas del examen parcial, el examen final y la tarea académica, estas no presentan ninguna uniformidad. Esto se debe a la relativa subjetividad en el establecimiento del nivel de dificultad de las evaluaciones y en la calificación correspondiente. Todo ello, indudablemente, agrega ruido a los datos y no permite fijar patrones fuertes que los clasifiquen categóricamente.

Una vez comprendido y establecido el dominio de aplicación, que son los datos a los cuales se les aplicarán las nuevas técnicas de aprendizaje automático que esta investigación propone, será más simple entender el porqué de su propuesta. Estos métodos, como se sabe, están basados en la técnica de Bagging y se diferencian en que en vez de usar muestreo aleatorio con reemplazo, aplican el concepto de períodos o años académicos. Los resultados son tan buenos como los obtenidos al aplicar la técnica de Bagging[3].

Como se sabe, en la propuesta de Baggin tradicional, antes de clasificar una nueva instancia por votación, se deben generar tantos modelos como cantidad de conjuntos *bootstrap* existan, de tal forma que, a mayor cantidad de períodos académicos, más datos se tendrán que analizar, con el consecuente costo computacional y tiempo invertido en la aplicación de la técnica base para generar los modelos. Por el contrario, la técnica Bag- E propuesta en el presente trabajo no presenta el problema del costo computacional ni el de tiempo, debido a que los modelos, una vez construidos, permanecen fijos en el tiempo, y el único modelo generado es el correspondiente al último período o año académico considerado.

Por otro lado, la propuesta Bag-P tiene mayor costo computacional que Bag-E, pero menor que el Bagging clásico, debido a que la cantidad de datos que deben ser analizados en cada nuevo conjunto es menor que en el Bagging tradicional.

Esta última propuesta, sin embargo, no estuvo exenta de problemas. Como en todo estudio empírico sobre la aplicación de una técnica, la media del error de la predicción puede resultar una estimación imparcial en el caso de que el error observado en los conjuntos de prueba sea independiente. Como se sabe, en la propuesta de Bag-P dicho error no es independiente debido a que el conjunto se superpone como consecuencia de la forma de seleccionar nuestros datos de entrenamiento[4]. Consecuentemente, cuando el conjunto de prueba es usado para evaluar el significado

[3] Si bien es cierto que la mejora fue significativa a simple vista, en una prueba estadística de análisis de varianza presentó resultados iguales que los de las configuraciones clásicas más eficientes.

[4] Lo que no ocurre en el Bag-E, ya que los datos son completamente independientes.

en errores de predicción, el resultado debe ser cuidadosamente interpretado. En particular, la significancia estadística debería leerse como muy probable para mantener alguna expectativa sobre los datos dados, no para mantenerla para datos futuros.

7.2.3. Sobre los resultados de los experimentos

En el capítulo correspondiente a la evaluación de la propuesta referida a la metodología de preparación de datos y de las nuevas técnicas, se llevaron a cabo dos tipos de experimentos. En el primero, el de filtrado colaborativo basado en memoria, se usó una base datos de estudiantes con sus notas. En esta primera fase, se concluyó que la predicción basada en la suma ponderada de todas las notas de los estudiantes, utilizando el cálculo de la similitud mediante la correlación de Pearson, presenta el menor error de predicción.

El experimento que usó el filtrado colaborativo basado en modelos se desarrolló en cinco fases. En la primera, se determinaron las mejores condiciones para el aprendizaje automático, lo que incluyó establecer el mejor algoritmo de clasificación, el mejor conjunto de datos y, por último, el mejor tratamiento en el cálculo del potencial. Finalmente, se pudo concluir que el algoritmo de árboles de decisión es el más eficaz para este dominio en particular, que el conjunto que mejor representa la realidad estudiada es el conjunto que incluye los atributos sintéticos y que el atributo potencial es igual de eficaz en cualquiera de sus tratamientos.

En la segunda fase de experimentación, se determinó si existía una diferencia significativa entre un semestre y el que resultaba de adicionarle a aquel los datos del semestre nuevo. Finalmente, se pudo concluir, por intermedio de la prueba de estadística de proporciones, que no existe diferencia significativa entre un semestre y otro. Con estos resultados, se decidió utilizar, para el resto del trabajo, la información de la base de datos desde la creación de la Facultad.

En la tercera fase, se determinó el mejor método de poda para los árboles de decisión en nuestro dominio de aplicación. En este experimento, se convino usar conjuntos de datos, correspondientes a nuestro dominio de aplicación, con comportamientos que, al parecer, eran extremos e, inclusive, a uno de ellos se le aplicó remuestreo con el objeto de nivelar los valores de su clase. Los resultados mostraron que las técnicas de poda producen una mejora en la precisión predictiva, así como también mejoras orientadas al entendimiento de los árboles. En la tabla 6.21, se puede observar que el método EBP es el que mejor precisión predictiva produce en nuestro dominio de aplicación y, a su vez, el que optimiza el entendimiento, como se puede observar en la figura 6.22.

En la cuarta fase de la experimentación, se probaron las técnicas de clasificación por conjuntos Bagging y Boosting, y se concluyó que el Bagging obtuvo mejor precisión predictiva que la mejor configuración del algoritmo de árboles de decisión.

Por último, en la quinta fase, se experimentó con los datos del presente dominio mediante los métodos propuestos en esta investigación. Aquí se utilizaron dos métricas de rendimiento: la de precisión y la que corresponde al área bajo la curva ROC. Si bien es cierto que el algoritmo Bag-E (tabla 6.29) obtiene, en promedio, menor tasa de error que todos aquellos con los que se compara, al hacer una prueba estadística de ANOVA se pudo observar que no existen diferencias significativas entre los algoritmos estudiados. Eso significa que aquellos que fueron propuestos funcionan tan bien como la mejor configuración encontrada para el algoritmo de Bagging. Esto último es válido en relación con la métrica de precisión y con la métrica del área bajo la curva ROC

Es necesario resaltar, a modo de conclusión, que los experimentos presentados en este trabajo, aparte de utilizar datos reales, mostraron que las herramientas utilizadas son técnicas muy poderosas, pero que, a su vez, presentan puntos débiles cuando buscan patrones en un dominio de conocimiento en donde el factor humano participa y tiene consecuencias directas en los datos de una manera muy importante.

7.3. Consideraciones finales

El sistema implementado, construido en base a la metodología propuesta, provee dos ventajas. Por un lado, el estudiante puede inferir que una recomendación del sistema, proveniente de un patrón determinado, está relacionada, de alguna manera, con su rendimiento académico global o con el de determinadas asignaturas, en base a datos históricos de estudiantes que con un perfil parecido. Por lo tanto, el estudiante podrá decidir libremente, tomando en cuenta factores más subjetivos, si efectúa o no la matrícula. Por el lado de la institución, el sistema, a la vez que recomienda, conserva información relevante respecto al rendimiento de los estudiantes, información que no hubiera sido posible conservar considerando algún método tradicional de almacenamiento y menos si la institución decide la participación humana por intermedio de un asesor, dejando de lado el aspecto tecnológico.

Así, tomando en cuenta su objetivo principal (apoyar al estudiante mediante recomendaciones en el proceso muchas veces complicado de decidir en cuántas y cuáles asignaturas matricularse sobre la base de su rendimiento académico y el de otros con características similares a él), se puede concluir que la propuesta cumple con proporcionar el mayor beneficio de este trabajo: brindar información útil para que el estudiante cuente con un elemento de juicio adicional al momento de decidir en qué asignaturas matricularse en cada semestre académico.

7.3.1. Sobre la metodología de creación y preparación de los datos

Nuestra investigación partió de la constatación del vacío que se genera cuando un estudiante, sin la experiencia debida, se matricula en un determinado período académico. Debido a esta problemática, la presente investigación presenta un sistema de asesoría virtual automática cuya inteligencia proviene de los datos históricos de otros estudiantes que durante años anteriores han cursado asignaturas similares a las recomendadas.

En caso de que la presente metodología o una variante de ella se tenga que aplicar en alguna institución de educación superior, podría surgir un problema al tener que decidir sobre los atributos que habría que involucrar en el estudio. La decisión de considerar tales o cuales atributos obedece, entre otras cosas, a la calidad con la que los que sean seleccionados representen a los individuos o sus acciones. Por lo tanto, el número de atributos provenientes de los datos históricos pueden variar dependiendo de los objetivos y de la realidad inserta en la institución. Por ejemplo, en el presente estudio se decidió por eliminar atributos demográficos tales como nacionalidad, sexo, colegio de procedencia, lugar de residencia, debido a que no eran representativos para nuestro estudio y para nuestra realidad[5].

Sin embargo, pese a que con el grupo de atributos se pudieron obtener modelos que se ajustaban a nuestra realidad, faltaba controlar algunos otros que, como se mencionó anteriormente, tienen que ver con las reglas de la institución[6]. En nuestro caso, se relacionaban con la dificultad de la asignatura y con el potencial de cada estudiante. Esto no implica que no exista otro atributo que subyaga a los datos primitivos, que pueda ayudar a representar al individuo o sus acciones. Por lo tanto, se requiere entonces enfocar la solución al problema en la búsqueda de atributos adicionales y sus respectivas caracterizaciones.

La necesidad de plantear los atributos sintéticos es coherente con la observación del modus operandi del docente asesor en el momento de la asesoría. En dicha situación, como cada carrera tiene un plan curricular, el asesor solicita información al sistema sobre las notas que obtuvo el alumno en la asignatura inmediata anterior, es decir, la que fue requisito de la consultada. En nuestro sistema automático, era necesario, por tanto, un atributo que, de alguna manera, tuviera implícita la información que el docente asesor procesaba con dos fuentes de información: la primera, proveniente de la malla curricular, y la segunda, proveniente del sistema que proporciona la(s) nota(s) de la(s) asignaturas que son requisitos.

[5] Esto no implica que tengan que ser excluidos en otras realidades.
[6] Los datos existentes en las bases de datos de instituciones de educación, cuando interactúan con reglas propias de las mismas instituciones, pueden generar datos adicionales.

Ahora bien, una vez que se establecieron estos atributos en la base de datos y se calcularon, surgieron nuevos problemas que se consideran como una limitación en el presente trabajo de investigación. En primer lugar, la dificultad y el potencial son conceptos heurísticos derivados de la experiencia de las asesorías presenciales. La idea en esta investigación es cuantificar este tipo de conocimiento, para que la información se entregue mejor al momento de aplicar la técnica de aprendizaje en la clasificación.

En segundo lugar, existe información sobre cada variable cuya valía se sospecha, pero que no se tiene registrada y no se puede medir. Por ejemplo, para la variable vez, cuáles son las condiciones que llevaron a un estudiante a suspender una asignatura. Estas no siempre son normales, por lo que el número de veces que lleve el curso posiblemente no guardará relación con el histórico de dicho estudiante, lo que incluye ruido a los datos analizados en este estudio. Tampoco se registran los problemas o causas externas que determinaron el fracaso del estudiante. Para la variable dificultad, es evidente que esta no solo debería depender de la nota que otros estudiantes han obtenido en dicha asignatura, sino del tiempo y el esfuerzo que el estudiante le dedica. Como es evidente, estos factores no son medibles, ni siquiera observables en el ámbito universitario.

Para la clase nota, se puede considerar el tiempo que el estudiante dedica a la asignatura, la exigencia del docente (que muchas veces no es constante), la diferencia de niveles de exámenes de semestre a semestre, el rendimiento académico del grupo y los factores externos que influyen sobre sus individuos. Toda esta información, eventualmente, podría ser muy valiosa; sin embargo, no se puede medir y no está registrada en ninguna base de datos de la Universidad.

En tercer lugar, al hacer la prueba estadística para comparar el rendimiento entre las bases de datos sin la aplicación de la metodología propuesta y aquellas en las que sí se aplicó, se observó que la mejora, en términos de precisión, no es tan significativa como se esperaba.

Por último, se debe considerar que, al tratar con seres humanos, sus reacciones y los resultados (que implican un trabajo dedicado, concentración adecuada, estudio constante y una cierta estabilidad emocional), se trabaja en un campo donde existe un amplio margen para los resultados desconocidos, que definitivamente afectan la precisión de las estimaciones.

Sobre esto último, lo que no debe perderse de vista es que la predicción proveniente de una clasificación generada por el sistema es, para el estudiante, en términos de su uso, una simple recomendación basada en estadísticas, que podrá tomar en cuenta si así lo desea y a sabiendas de que proviene de estudiantes que, como él, han seguido cursos, se han matriculado, han hecho sus mejores esfuerzos y han obtenido notas con las que el sistema de aprendizaje automático hizo la clasificación.

7.3.2. Sobre el método propuesto para el aprendizaje automático

Una vez que los datos que serían analizados estuvieron bien definidos, se procedió a aplicar la técnica encargada del análisis, es decir, la técnica que iba a transformar dichos datos en un modelo. Como se sabe, no existe una técnica de clasificación supervisada que sirva para todos los dominios; por lo tanto, en este trabajo de investigación, el problema que se presentó radicaba en determinar la mejor técnica en términos de precisión predictiva, capaz de proporcionar recomendaciones por intermedio de predicciones más acertadas. Por ello, fue necesario hacer un estudio exhaustivo de aplicación de varias técnicas de aprendizaje a fin de encontrarla.

Este enfoque de experimentación, así asumido, estuvo plenamente justificado, debido a que únicamente los patrones con regularidad absoluta pueden resolverse con la sola aplicación de cualquier algoritmo. Por ello, parte del esfuerzo del presente trabajo residió en encontrar una técnica de aprendizaje capaz de mejorar el problema de entregar recomendaciones más confiables al estudiante en un ámbito sin regularidades absolutas.

Al encontrar resultados de clasificación que se podrían considerar explicables hasta cierto punto, se decidió estudiar una forma de aplicar clasificación mediante conjuntos, aún con la sospecha de que había algunos semestres o períodos académicos con mucho más ruido que otros. Es así como el estudio se centra en el algoritmo de Bagging, una de las técnicas basadas en remuestreo más comunes y cuya filosofía consiste en aplicar repetidas veces el algoritmo de aprendizaje que se elija a los conjuntos de entrenamiento obtenidos previamente por muestras aleatorias con reemplazo.

Los algoritmos que propone la presente investigación combinan conceptos de técnicas clásicas de clasificación por conjuntos con un concepto propio del conjunto de datos en estudio. Como se sabe, dicho conjunto se dividió en subconjuntos con el objetivo de obtener varios modelos que representen a sus elementos. Lo novedoso de la propuesta es el concepto utilizado en la división, si bien es cierto que este concepto representa una de las reglas más simples que gobierna a los datos provenientes de cualquier institución de educación[7].

7. 4. Trabajos futuros

En los últimos años, la necesidad de los centros de educación superior por entregar información a sus alumnos, profesores y personal administrativo a partir de los

[7] En toda institución de educación superior, existen períodos de estudio; dichos períodos son repetitivos para la institución y progresivos para el estudiante, que tiene que cumplir con una cierta cantidad de ellos para dar por concluida su carrera.

datos que generan se ha ido incrementando. Ello se justifica debido a que las instituciones deben tomar decisiones acertadas respecto a su conducción, en particular, sobre ciertas acciones concretas: crear o cerrar una carrera profesional, advertir la pérdida de estudiantes, predecir el índice de abandono, tener un sistema de alertas, etcétera. De esa manera, se pueden crear estrategias de decisión adecuadas, con las cuales evitar problemas futuros, que hasta el momento se han ido resolviendo con el diseño de sistemas de gestión que, a partir de datos históricos, utilizan técnicas de descubrimiento del conocimiento.

En ese sentido, la predicción del rendimiento académico abre muchas posibilidades, en tanto y en cuanto son numerosas las aplicaciones que se podrían conseguir a partir de la predicción de notas en un ámbito académico en general y en el de la educación superior en particular.

Como muestra de ello, a continuación se exponen los trabajos que, sobre la base de los diferentes tipos de resultados obtenidos a partir de la experimentación realizada, pueden continuar los lineamientos y los temas de la presente investigación.

7.4.1. Trabajos futuros referidos al dominio de aplicación

Un alto porcentaje de trabajos de investigación en el área de aprendizaje basado en minería de datos se lleva a cabo con datos sintéticos. Estos, en su mayoría, están alejados de la realidad o de un entorno donde participan las decisiones o acciones de los seres humanos. En la presente investigación, en cambio, se estudiaron datos reales provenientes de interacciones y, sobre todo, sujetos a reglas de la institución y a situaciones de la vida diaria: que un alumno se matricule, que decida en cuántas y cuáles asignaturas hacerlo, que obtenga una calificación (con todo lo que eso implica).

Todo ello amplía la variedad de nuestros datos. Por esa razón, urge involucrar más variables en el estudio, que podrían ser dependientes del entorno (una información más detallada de la dificultad de las asignaturas, de las evaluaciones obtenidas, de la asistencia e interés del estudiante, de las notas obtenidas en la secundaria, entre otras) o dependientes del estudiante (el tiempo que dedica a estudiar, su capacidad y habilidad para ciertas materias, su disposición para enfrentarlas, etcétera).

También es imprescindible un análisis más profundo para detectar ruido. Sería de mucho interés aplicar técnicas estadísticas basadas en regresión, modelos lineales y diseño de experimentos que ayuden en dicha tarea, pues si bien es cierto que en la presente investigación se plantea una metodología específica de preparación de los datos, por los diferentes experimentos y resultados obtenidos se observa que el error, casi constante, se debe a que el ruido no ha sido eliminado por completo.

7.4.2. Trabajos futuros referidos a las técnicas de aprendizaje supervisado

Resulta de mucho interés un futuro análisis de la diversidad de clasificadores generados mediante el método de Bagging y Boosting. Es verdad que en esta investigación se formula la propuesta de un nuevo método de clasificación por votación, pero un examen de tal diversidad posibilitaría la generalización del tratamiento estratificado que se da en nuestros datos y, con dicha tendencia, la capacidad de probar si ocurre lo mismo con los datos de cada una de las otras facultades y, quizá, con los de otras instituciones.

7.4.3. Trabajos futuros referidos a las métricas de rendimiento

Es indispensable hacer un estudio futuro de los costos de clasificar erróneamente a un estudiante como aprobado o suspendido, trabajo que deberá estar enfocado en métricas confiables que presenten una alta utilidad para seleccionar clasificadores. La necesidad de este estudio reside en el hecho de que no existe solamente una métrica buena para cada clasificador y mucho menos para cada dominio de aplicación, pues tan solo se cuenta con algunos indicios sobre qué métrica podría ser adecuada cuando los datos tienen cierto sesgo o tendencia.

Por ese motivo, hay que determinar la pertinencia de unas sobre las otras, para lo cual los costos son un criterio decisivo, dado que, como se sabe, los clasificadores, pese a ser instrumentos sumamente productivos, tienen un límite, no tanto por el hecho de que algunas de sus predicciones puedan ser erróneas, como porque dichos errores conllevan efectos diversos. Esto equivale a decir que las consecuencias de los errores no siempre son equivalentes, incluso las relacionadas con un mismo problema. Si se hace referencia específicamente al campo del diagnóstico, se notará que ciertos errores resultan más costosos que otros (por ejemplo, el costo de errar el tiempo de vida de un paciente es psicológicamente insignificante si se calculan años adicionales al plazo real, pero puede ser fatal si se trata de algunos años menos). En ese sentido, en términos generales, conviene diferenciar muy claramente el costo de un error del error mismo, a lo cual tendrán que avocarse los trabajos futuros referidos a las métricas de rendimiento. Con ello, si bien no se puede evitar que un clasificador cometa errores, están dadas las condiciones para discriminar sus costos y, por lo tanto, la calidad del clasificador.

7.5. Publicaciones

- *A Case Study: Data Mining Applied to Student Enrollment* César Vialardi, Jorge Chue Alfredo Barrientos, Daniel Victoria, Jhonny Estrella, Álvaro Ortigosa. Proceedings of Second Educational Data Mining conference*., p. 333-335, Pittsburg Pensilvania, Usa , Julio 2010 ISBN:978-0-615-37529-8.*

- *A Methodology for Student Performance Prediction Based on Data Mining César Vialardi, Alfredo Barrientos, Daniel Victoria, Jhonny Estrella and Álvaro Ortigosa. In* Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2010 *(pp. 388-393). Chesapeake, VA: AACE. ISBN: 1-880094-81-9*

- *La predicción del rendimiento académico en la Universidad: Una aplicación de sistemas de* recomendación basada en minería de datos. César Vialardi, Jorge Chue, Alfredo Barrientos, Daniel Victoria, Johnny Estrella, Álvaro Ortigosa II EIM@ - 2009 II Encontro Interactivo de Matemática Aplicada. Universidad Autónoma Metropolitana. México - Brazil – Perú Octubre 2009

- *Recommendation in Higher Education Using Data Mining Techniques url, bibtex presentación* César Vialardi*, Javier Bravo, Leila Shafti y Alvaro Ortigosa* Proceedings of Second Educational Data Mining conference*., p. 190-199, Universidad de Córdoba, Córdoba, Spain, Julio 2009 (ISBN: 978-84-613-2308-1).*

- *Improving AEH Courses through Log Analysis* César Vialardi, Javier Bravo y Alvaro ortigosa Journal of Universal Computer Science - J.UCS, Vol. 14, Issue 17, p. 2777-2798 ,Graz, Austria, Febrero 2009 (ISSN: 0948-6968).

- *Using Decision Trees for Discovering Problems on Adaptive Courses* Javier Bravo, César Vialardi y Alvaro Ortigosa. Proceedings of E-Learn 2008: World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education, p. 268-277 ,Las Vegas, EE.UU., Noviembre 2008 (ISBN: 1-880094-66-5).

- *ASquare: A Powerful Evaluation Tool for Adaptive Hypermedia Course System Javier Bravo,* César Vialardi *y Alvaro Ortigosa. Proceedings of Hypertext 2008 Conference, p. 219-220, Pittsburgh, EE.UU., Junio 2008 (ISBN: 978-1-59593-998-2).*

- *A Problem-Oriented Method for Supporting AEH Authors through Data Mining* Javier Bravo, César Vialardi y Alvaro Ortigosa. Proceedings of International Workshop on Applying Data Mining in e-Learning (ADML07) held at the Second European Conference on Technology Enhanced Learning (EC-TEL 2007), p. 53-62, Creta, Grecia, Septiembre 2007 (ISSN: 1613-0073).

- *Empowering AEH Authors Using Data Mining Techniques.* César Vialardi, Javier Bravo y Alvaro Ortigosa. Proceedings of Fifth International Workshop on Authoring of Adaptive and Adaptable Hypermedia (A3H 2007) held at the 11th International Conference on User Modeling (UM2007), p. 33-43, Corfu, Grecia, Junio 2007.

Apéndice A

A.1. Archivo de datos

Esta sección contiene el archivo de datos utilizado como ejemplo en la explicación de cada una de las técnicas desarrolladas en la presente investigación.

A.1.1. Archivo de datos alumnos.csv

```
Cursos matriculados, Cursos, Vez de matrícula, PPA, Promedio
uno,C1,segunda,malo,SUSP
uno,C1,segunda,bueno,SUSP
uno,C2,segunda,malo,APROB
dos,C3,segunda,malo,APROB
tres,C3,primera,malo,APROB
tres,C3,primera,bueno,SUSP
tres,C2,primera,bueno,APROB
dos,C1,segunda,malo,SUSP
tres,C1,primera,malo,APROB
dos,C3,primera,malo,APROB
dos,C1,primera,bueno,APROB
dos,C2,segunda,bueno,APROB
uno,C2,primera,malo,APROB
dos,C3,segunda,bueno,SUSP
```

A.1.2. Árbol de decisión correspondiente al conjunto de datos alumnos.csv

```
== Run information ===
Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     alumnos.csv
Instances:    14
Attributes:           5
                      Cursos matriculados
                      Cursos
                      Vez de matricula
                      PPA
                      Promedio
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree

 Cursos = C1
|    Vez de matr¡cula = segunda: SUSP (3.0)
|    Vez de matr¡cula = primera: APROB (2.0)
 Cursos = C2: APROB (4.0)
 Cursos = C3
|   PPA  = malo: APROB (3.0)
|   PPA  = bueno: SUSP (2.0)
Number of Leaves  :         5
Size of the tree :  8
```

A.1.3. Archivo de datos de dominio de aplicación Alumno _Ing_ Sistemas.csv

Lo que se muestra es un extracto del archivo original debido a que este tiene 161247 registros.

Curso, Vez, Promedio Inicio, Potencial, Creditaje, CreditosMatriculados, Dificultad, Clase

INGENIERIA ECONOMICA, 1, 13.24, 1.11, 2, 19, 11.78, Aprobado

INGENIERIA DE DATOS, 1, 12.18, 0.84, 5, 20, 12.42, Desaprobado

FISICA II, 1, 13.26, 0.95, 5, 22, 11.76, Desaprobado

FISICA I, 1, 12.20, 1.10, 4, 11, 11.52, Desaprobado

SIMULACION DE SISTEMAS, 1, 14.42, 1.17, 3, 16, 11.10, Desaprobado

CALCULO III, 1, 11.28, 1.00, 5, 20, 10.82, Desaprobado

ESTADISTICA Y PROBABILIDAD I, 1, 11.42, 0.91, 3, 20, 10.82, Desaprobado

INTRODUCCION A LAS CIENCIAS SOCIALES, 1, 13.57, 1.12, 2, 19, 12.51, Desaprobado

ESTADISTICA Y PROBABILIDAD I, 1, 11.28, 0.73, 3, 19, 10.82, Desaprobado

DISENO LOGICO, 1, 10.35, 0.86, 2, 20, 11.61, Desaprobado

ESTRUCTURAS DISCRETAS EN COMPUTACION, 1, 9.59, 0.83, 5, 22, 12.70, Desaprobado

PROGRAMACION, 1, 11.39, 1.02, 3, 12, 12.31, Desaprobado

HISTORIA UNIVERSAL CONTEMPORANEA, 2, 11.13, 0.71, 3, 9, 11.11, Desaprobado

DISENO LOGICO, 1, 9.01, 0.77, 3, 17, 11.61, Desaprobado

INGENIERIA ECONOMICA, 1, 11.43, 0.92, 2, 22, 11.78, Desaprobado

ALGEBRA LINEAL, 1, 9.78, 0.91, 4, 10, 10.07, Desaprobado

ECONOMIA GENERAL, 1, 12.36, 1.04, 3, 16, 11.08, Desaprobado

CALCULO I, 2, 7.3, 0.87, 4, 15, 10.84, Desaprobado

TEORIA DE SISTEMAS, 1, 10.92, 0.85, 2, 19, 12.27, Desaprobado

FISICA II, 1, 13.36, 1.12, 4, 19, 11.76, Desaprobado

CALCULO II, 1, 13, 1.00, 4, 22, 10.89, Desaprobado

ALGEBRA LINEAL, 1, 10.70, 0.96, 4, 17, 10.07, Desaprobado

ECONOMIA GENERAL, 2, 7.36, 0.72, 3, 10, 11.08, Desaprobado

ESTRUCTURAS DE DATOS Y ALGORITMOS, 2, 10.38, 0.78, 4, 15, 12.30, Desaprobado

CALCULO III, 1, 8.91, 0.74, 4, 17, 10.82, Desaprobado

INGENIERIA DE SOFTWARE I, 1, 11.73, 0.68, 4, 21, 12.94, Desaprobado

ESTADISTICA Y PROBABILIDAD II, 1, 12.37, 1.38, 3, 13, 11.38, Desaprobado

ESTADISTICA Y PROBABILIDAD I, 1, 13.17, 1.19, 3, 19, 10.82, Desaprobado

SIMULACION DE SISTEMAS, 2, 10.18, 0.58, 3, 6, 11.10, Desaprobado

ALGEBRA LINEAL, 3, 10, 0.86, 4, 14, 10.07, Desaprobado

A.1.4. Análisis descriptivo de los datos del conjunto Alumno_Ing_ Sistemas.csv

A continuación, se hace un análisis descriptivo de los datos. Para interpretar correctamente los gráficos, se debe tener en cuenta que un registro en la tabla del anexo A.1.3. significa la vez que un estudiante se matriculó en una asignatura, cursó dicha asignatura y, una vez terminado el semestre, obtuvo su calificación y esta se actualizó en dicho registro, a la vez que se actualizó el promedio ponderado acumulado. Por lo tanto, es evidente que hablar de un registro y de un estudiante son dos cosas completamente distintas.

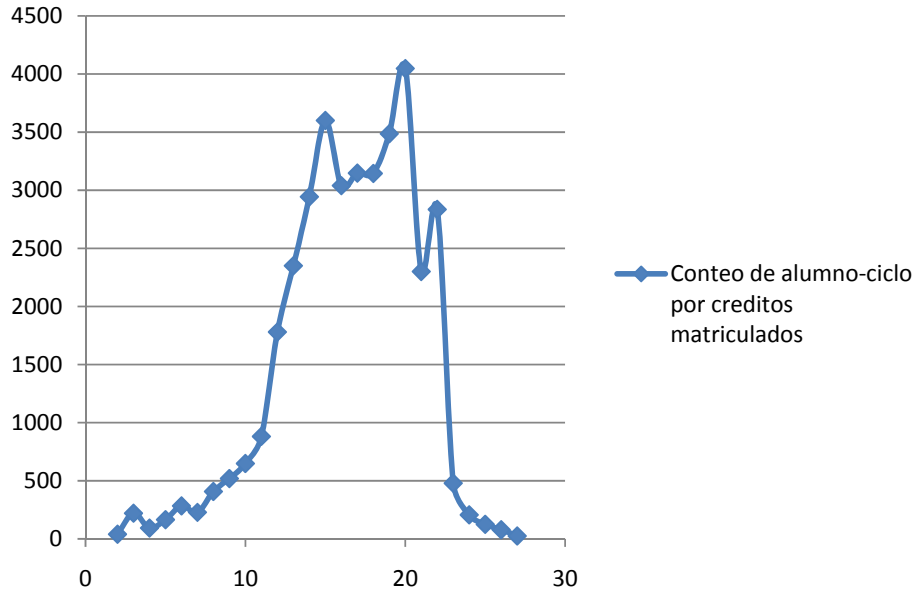

Figura A.1.1. Créditos matriculados versus cantidad de matrículas

La figura A.1.1. corresponde a un gráfico de frecuencias. Por ejemplo, el par ordenado (10; 500) significa que en toda la base de datos existen 500 veces en que los estudiantes se matricularon en 10 créditos.

En total, se tiene 37084 matrículas distintas; ellas corresponden a los 161247 registros de nuestra base de datos original. De aquí se puede observar que en un proceso de matrícula, en promedio, cada estudiante se matricula en 4.35 asignaturas.

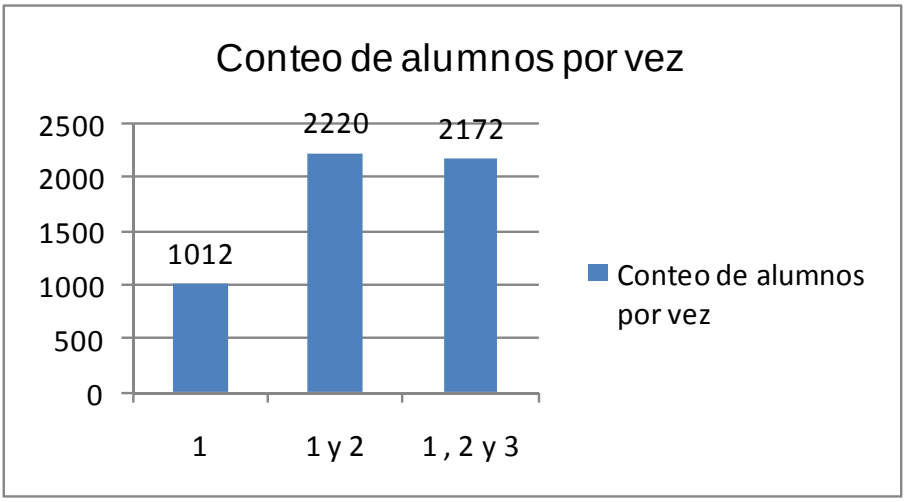

Figura A.1.2. Cantidad de alumnos por vez de matrícula

En la figura A.1.2., se observa que existen 1012 alumnos que han cursado todas sus asignaturas solo una vez, 2220 alumnos que han cursado, al menos, una asignatura dos veces y 2172 alumnos que han cursado, al menos, una asignatura tres veces.

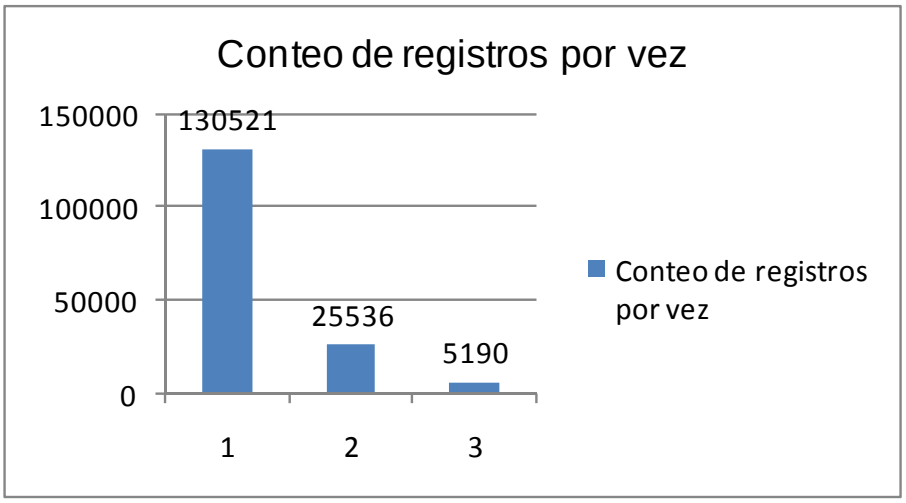

Figura A.1.3. Cantidad de registros por vez de matrícula

En la figura A.1.3., se observa que 130521 registros corresponden a asignaturas cursadas por primera vez, 25536 registros que corresponden a asignaturas cursadas por segunda vez, y 5190 registros corresponden a asignaturas cursados por tercera vez.

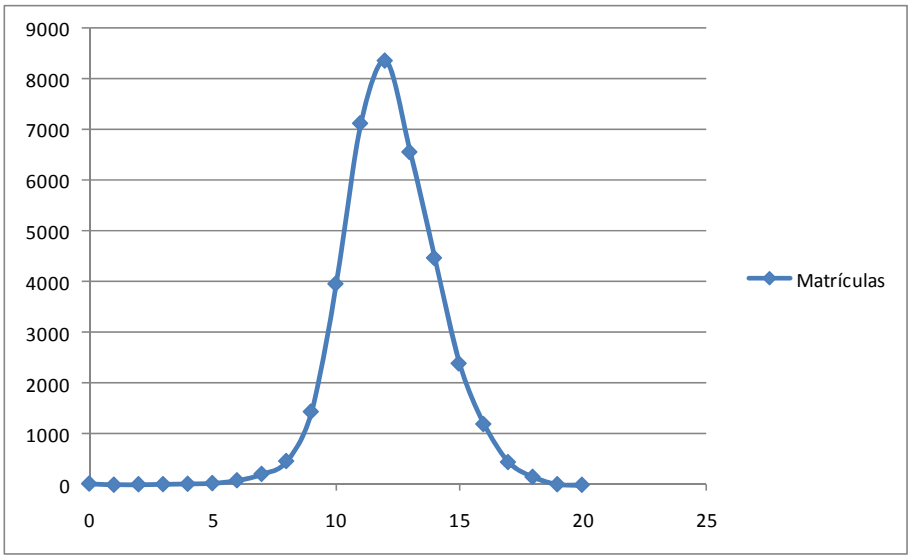

Figura A.1.4. Créditos matriculados versus matrículas

En la Figura A.1.4., el par ordenado (10, 4000) significa que 4000 matrículas fueron hechas por los estudiantes teniendo un promedio ponderado inicial de 10.

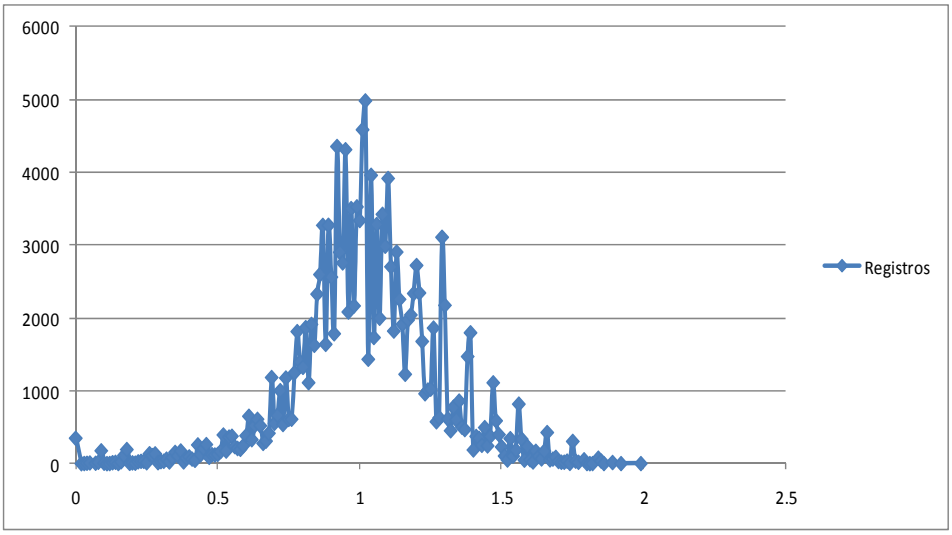

Figura A.1.5. Potencial versus cantidad de registros

En la Figura A.1.5., el par ordenado (1, 2000) significa que 2000 registros presentan un valor de 1 para el atributo potencial.

A.1.5. Análisis descriptivo de la distribución de errores

En esta sección, se muestra la gráfica de la distribución de los errores cometidos al entrenar la base de datos original de 161247 registros, separar el último semestre 2009-1(4563 registros) y luego aplicar el algoritmo c4.5 para entrenar el conjunto de registros restantes que contienen 156684. Al comparar los datos reales con los predichos, se obtiene que el algoritmo se equivoca en predecir 810 veces, de las cuales 320(40%) errores corresponden a valores cercanos a la nota 10. En la figura A.1.6., se observa el grafico de distribución entre las diferencias de los valores reales y los predichos:

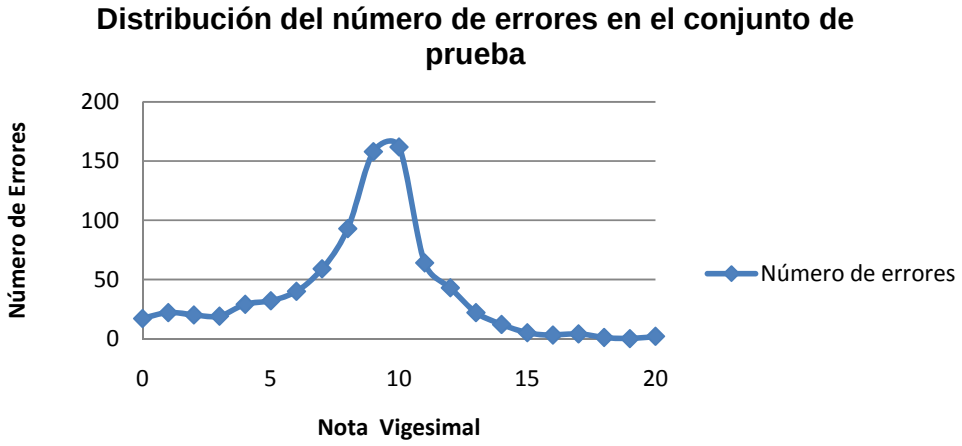

Figura A.1.6. Diferencias de los valores reales y los predichos

A.2. Experimento correspondiente al filtrado colaborativo basado en memoria

En este anexo se presenta un resumen de las configuraciones y experimentaciones de los experimentos llevados a cabo cuando se aplicó el filtrado colaborativo. Para llevar a cabo esto, se consideran las tres formas de obtener la similitud, expresadas en las ecuaciones 3.18, 3.19 y 3.20, y las tres formas de obtener la predicción, expresadas en las ecuaciones 3.21, 3.22 y 3.28.

Para el cálculo de la similitud, se toma en cuenta el valor de la calificación obtenida por el estudiante al cursar la asignatura por primera vez.

Para identificar el tipo de cálculo usado para determinar la similitud y la predicción, se considerará las abreviaturas de las tablas A.2.1 y A.2.2 respectivamente.

Acrónimo	Tipo de similitud
COSB	Coseno
COR	Correlación
COSA	Coseno Ajustado

Tabla A.2.1. Diferentes maneras de obtener la similitud

Acrónimo	Tipo de Predicción
SUMA	Suma Ponderada
OTRO	Suma Ponderada de otros
REG	Modelo de Regresión

Tabla A.2.2. Diferentes maneras de obtener la predicción

A continuación, se mostrarán las nueve formas de ejecutar las evaluaciones según los factores mencionados:

Número de Prueba	Tipo de Similitud	Tipo de predicción
1	COSB	SUMA
2	COSB	OTRO
3	COSB	REG
4	COR	SUMA
5	COR	OTRO
6	COR	REG
7	COSA	SUMA
8	COSA	OTRO
9	COSA	REG

Tabla A.2.3. Diferentes combinaciones para el cálculo de la predicción

En la tabla A.2.3., se pueden observar todas las combinaciones consideradas para el cálculo de la predicción.

El primer paso para la preparación de los datos es crear una tabla que contenga los datos de los alumnos con los promedios, vez, notas y el código del curso actualizado. Se necesita generar una tabla de alumnos que contenga el código del alumno y el promedio de sus notas relacionadas con la primera vez. También se diseñó una tabla con la relación de los cursos.

La clase real y clase predicha fueron discretizadas con el concepto de aprobado y desaprobado. Para los valores menores de once, como desaprobados, y para valores mayores o iguales que once, como aprobado.

De esta manera, se buscó predecir la clase a la que pertenece una nueva instancia.

Antes de ejecutar la experimentación, fue necesario definir las siguientes reglas:

- En los casos de modificación curricular, para los estudiantes que cursaron asignaturas equivalentes a una del plan de estudios actual, se ha considerado tomar las evaluaciones de todas las asignaturas. Esto se debe a que la asignatura del plan de estudios antiguo puede no ser la misma, en contenidos, que la del plan curricular vigente.
- Para evitar el problema del Cold Start, la predicción se obtendrá considerando las notas de todas las asignaturas cursadas por el alumno.
- Debido a que los promedios de las evaluaciones de cada asignatura son redondeados al valor inmediato superior, decidimos redondear las predicciones.
- Se tomarán en cuenta las evaluaciones de todos los ciclos anteriores. Luego de realizar las ejecuciones y comparar las predicciones con las clases reales, obtenemos los siguientes resultados considerando el período 2009-2 como conjunto de prueba.

La métrica de precisión utilizada en estas pruebas fue la precisión (Accuracy). Cabe mencionar que la precisión predictiva obtenida por cada una de las pruebas se puede visualizar en la tabla A.2.4:[1]

Ranking	Tipo de (similitud-predicción)	Porcentaje de Error
1	`COSB-REG	15.91%
2	`COR-REG	15.93%
3	`COSA-REG	16.41%
4	`COSA-OTRO	20.60%
5	`COR-OTRO	20.65%
6	`COSA-SUMA	20.76%
7	`COR-SUMA	21.13%
8	`COSB-SUMA	21.47%
9	`COSB-OTRO	21.53%

Tabla A.2.4. Porcentaje de error de cada una de las pruebas hechas a la base de datos

De la tabla A.2.4, se observa que los experimentos que obtienen mejor precisión son:

Puesto	Tipo de Predicción	Porcentaje de Error
1	Similitud usando Coseno Base Predicción tomando regresión	15.91%
2	Similitud usando Correlación Predicción tomando regresión	15.93%

Tabla A.2.5 Relación de los mejores experimentos y su respectivo porcentaje de error

[1] El error base de los datos analizados es de 16.04 por ciento.

Debido a que se está haciendo una predicción por clasificación, y que se sabe que, para nuestro dominio, una nota menor que 11 significa desaprobado, sin tener en cuenta qué tan próximo se encuentre del valor umbral (nota >= 11). Además, se sabe que un sistema de filtrado colaborativo, al generar predicciones o recomendaciones, ejecuta los cálculos con las notas de los alumnos para obtener un valor numérico como predicción. Por todo esto, se calculan las métricas de desempeño que hagan una diferencia entre lo que predice y el valor real que obtuvo un estudiante.

Las métricas de evaluación usadas son MAE y NMAE. Los resultados del cálculo se pueden observar en la tabla A.2.6:

Tipo de Combinación	MAE	NMAE
`COR-OTRO`	2.10312831	0.10515642
`COSA-REG`	2.11611877	0.10580594
`COR-SUMA`	2.12963945	0.10648197
`COR-REG`	2.15509014	0.10775451
`COSB-REG`	2.16516437	0.10825822
`COSA-SUMA`	2.24469777	0.11223489
`COSA-OTRO`	2.26458112	0.11322906
`COSB-SUMA`	2.28844115	0.11442206
`COSB-OTRO`	2.29029692	0.11451485

Tabla A.2.6. Resultados del MAE y NMAE ordenados de menor a mayor

De esta manera, se puede observar que la mejor evaluación, en términos de precisión numérica, se obtiene utilizando la correlación para el cálculo de la similitud y la suma ponderada de otros para el cálculo de la predicción.

Debido a que un clasificador hace predicciones y muchas de ellas pueden ser erróneas, es importante conocer cuál es su efecto cuando estas son incorrectas. En muchas situaciones, los errores tienen distintas consecuencias. Algunos errores tienen costos más elevados que otros, especialmente en el campo del diagnóstico. Obviamente, los costos de cada clasificación dependen del dominio de aplicación y del problema, pero casi nunca se da el caso de que ellos sean uniformes para un solo problema. Consecuentemente, algunas veces, la precisión no es la mejor manera de evaluar la calidad de un clasificador.

Por ejemplo, decirle a un alumno que va a aprobar una asignatura cuando realmente la desaprueba genera un tipo de error. Sin embargo, este error es distinto a decirle a otro alumno que desaprobará una asignatura cuando realmente la aprueba. Por esta razón, una de las maneras de observar el rendimiento de las predicciones de un clasificador con respecto a sus valores reales es utilizando los valores de una matriz llamada “de confusión” y las métricas que derivan de ella.

Ahora usaremos las métricas de precisión de la clasificación, específicamente la sensibilidad, la especificidad y el área bajo la curva (AUC).

En el caso de estudio, la sensibilidad mide la proporción de estudiantes que aprueban una asignatura y que, después de hacer el experimento, son correctamente clasificados. Por otro lado, la especificidad mide la proporción de estudiantes que desaprobaron y que, después de hacer el experimento, son correctamente clasificados

. Una buena predicción es aquella que maximiza la especificidad y la sensibilidad. Para el caso en estudio, un error en los falsos positivos es peor que uno en los verdaderos negativos; por lo tanto, se busca tener una alta especificidad.

En la tabla A.2.7, se muestra las combinaciones ordenadas de mayor a menor respecto a la columna de especificidad.

Nombre de la evaluación	Especificidad	Sensibilidad	Área Bajo la Curva
`COR-OTRO`	51.40%	84.69%	0.8165
`COSB-OTRO`	43.64%	85.13%	0.8122
`COSA-OTRO`	43.14%	86.33%	0.8168
`COSA-SUMA`	38.35%	87.05%	0.8160
`COR-SUMA`	35.21%	87.21%	0.8142
`COSB-SUMA`	27.27%	88.32%	0.8124
`COSA-REG`	22.98%	95.17%	0.8378
`COR-REG`	20.00%	96.31%	0.8401
`COSB-REG`	19.17%	96.50%	0.8403

Tabla A.2.7. Especificidad, sensibilidad y AUC para cuando el umbral es once

Si se observa la tabla A.2.7., encontramos una cierta correspondencia con los resultados obtenidos en la tabla A.2.6.

Se podría decir, en este caso, que la similitud basada en correlación y la predicción que utiliza la suma ponderada de otros reportan los mejores porcentajes en la métrica MAE y en la especificidad, mas no en la precisión predictiva.

Debido a que las clases se encuentran desbalanceadas, el valor de la especificidad no es tan alto, y se obtienen mejores predicciones para los aprobados. Para eliminar este problema, decidiremos aumentar el umbral de la predicción a un valor de trece para ser considerado aprobado.

Los resultados de la ejecución se pueden visualizar en la tabla A.2.8:

Nombre de la combinación	Especificidad	Sensibilidad	Área Bajo la Curva
`COR-OTRO-NOMAL-NOAR`	86.28%	55.95%	0.7239
`COR-SUMA-NOMAL-NOAR`	83.14%	52.64%	0.7074
`COSB-OTRO-NOMAL-NOAR`	78.02%	59.27%	0.7312
`COSA-OTRO-NOMAL-NOAR`	78.02%	60.40%	0.7359
`COSA-REG-NOMAL-NOAR`	76.69%	66.40%	0.7601
`COR-REG-NOMAL-NOAR`	76.53%	65.68%	0.7569
`COSA-SUMA-NOMAL-NOAR`	76.36%	59.96%	0.7328
`COSB-REG-NOMAL-NOAR`	76.03%	65.52%	0.7558
`COSB-SUMA-NOMAL-NOAR`	72.40%	60.91%	0.7336

Tabla A.2.8. Especificidad, sensibilidad y AUC para cuando el umbral es trece

Se puede observar que la especificidad aumenta y la sensibilidad baja demasiado. Por lo tanto, el umbral se disminuye a un valor de doce para ser considerado aprobado.

Los resultados de la ejecución se pueden visualizar en la tabla A.2.9:

Nombre de la combinación	Especificidad	Sensibilidad	Área Bajo la Curva
`COR-OTRO-NOMAL-NOAR`	71.24%	72.59%	0.7817
`COSB-OTRO-NOMAL-NOAR`	62.64%	74.04%	0.7809
`COSA-OTRO-NOMAL-NOAR`	62.15%	75.78%	0.7878
`COR-SUMA-NOMAL-NOAR`	61.65%	72.12%	0.7720
`COSA-SUMA-NOMAL-NOAR`	58.02%	75.56%	0.7835
`COSA-REG-NOMAL-NOAR`	51.24%	85.16%	0.8184
`COR-REG-NOMAL-NOAR`	50.25%	85.85%	0.8205
`COSB-REG-NOMAL-NOAR`	48.76%	86.11%	0.8204
`COSB-SUMA-NOMAL-NOAR`	48.26%	76.92%	0.7814

Tabla A.2.9. Especificidad, sensibilidad y AUC para cuando el umbral es doce

A.3. Determinación de las mejores condiciones para el aprendizaje automático

	C4.5 vs. KNN	C4.5 vs. NB
Prom. Dif	-0.13472	-3.32065
Desv. Est. Dif	0.1046393	0.1042676
LI	-0.2095744	-3.3952385
LS	-0.0598655	-3.2460614
t - pareada	-4.07133702	-100.71023
P-value	0.00279404	4.760E-15
	C4.5 es mejor	**C4.5 es mejor**

Tabla A.3.1. Comprobación del efecto del algoritmo de clasificación en el conjunto primitivo

	C4.5 vs. KNN	C4.5 vs. NB
Prom. Dif	-0.09115	-5.06912
Desv. Est. Dif	0.100767458	0.153604404
LI	-0.163234697	-5.179001971
LS	-0.019065303	-4.959238029
t - pareada	-2.860463238	-104.3587588
P-value	0.018766212	3.45655E-15
	C4.5 es mejor	**C4.5 es mejor**

Tabla A.3.2. Comprobación del efecto del algoritmo de clasificación en el conjunto N1

	C4.5 vs. KNN	C4.5 vs. NB
Prom. Dif	-0.02564	-5.06749
Desv. Est. Dif	0.142838177	0.193143
LI	-0.127820276	-5.205656
LS	0.076540276	-4.929323
t - pareada	-0.567640954	-82.96823
P-value	0.584158974	2.71E-14
	C4.5 es mejor	**C4.5 es mejor**

Tabla A.3.3. Comprobación del efecto del algoritmo de clasificación en el conjunto N2

	C4.5 vs. KNN	C4.5 vs. NB
Prom. Dif	-0.08289	-5.09003
Desv. Est. Dif	0.101723585	0.071565588
LI	-0.155658669	-5.141224938
LS	-0.010121331	-5.038835062
t - pareada	-2.576798647	-224.9137961
P-value	0.029852984	3.4547E-18
	C4.5 es mejor	**C4.5 es mejor**

Tabla A.3.4. Comprobación del efecto del algoritmo de clasificación en el conjunto NT

	C4.5 vs. KNN	C4.5 vs. Naive Bayes
Prom. Dif	-0.14843	-4.63568
Desv. Est. Dif	0.148311767	0.139543612
LI	-0.254525846	-4.735503486
LS	-0.042334154	-4.535856514
t - pareada	-3.164798612	-105.051798
P-value	0.011461427	3.2568E-15
	C4.5 es mejor	**C4.5 es mejor**

Tabla A.3.5. Comprobación del efecto del algoritmo de clasificación en el conjunto PPA

C4.5 (Conj Prim.) Vs	C4.5 N1	C4.5 N2	C4.5 NT	C4.5 PPA
Prom. Dif	0.20327	0.21424	0.18137	0.12391
Desv. Est. Dif	0.14710057	0.19460188	0.177896275	0.204609169
LI	0.09804059	0.0750302	0.054110672	-0.022458582
LS	0.30849941	0.3534498	0.308629328	0.270278582
t - pareada	4.36977353	3.48139683	3.224026474	1.915055061
P-value	0.00179768	0.00692331	0.010421918	0.087748657
	N1 es mejor	**N2 es mejor**	**NT es mejor**	**(Conj. Prim.) = PPA**

Tabla A.3.6. Comprobación del efecto de los nuevos atributos (potencial y dificultad)

	C4.5 N1 vs. C4.5 N2	C4.5 N1 vs. C4.5 NT	C4.5 N2 vs. C4.5 NT
Prom. Dif	0.01097	-0.0219	-0.03287
Desv. Est. Dif	0.190596246	0.169585901	0.216527155
LI	-0.125374341	-0.143214445	-0.187764196
LS	0.147314341	0.099414445	0.122024196
t - pareada	0.182008758	-0.40837051	-0.48005095
P-value	0.859609321	0.692546791	0.64264321
	N1=N2	**N1=NT**	**N2=NT**

Tabla A.3.7. Comprobación de la mejor metodología de cálculo de potencial

	N1	N2	N3
$\chi_0^2 = \sum \frac{(o_i - e_i)^2}{e_i}$	7.55292	8.59290	7.78745
$P(\chi^2 > \chi_0^2)$	0.99998	0.99995	0.99998

Tabla A.3.8 Resultados de la prueba de proporciones (Chi cuadrado)

Potencial N1			
	C4.5 vs. Bagging	**C4.5 vs. Boosting**	**Bagging vs Boosting**
Prom. Dif	0.24186	-0.84153	-1.08339
Desv. Est. Dif	0.135521193	0.072115263	0.167985789
LI	0.144913979	-0.893118152	-1.203559794
LS	0.338806021	-0.789941848	-0.963220206
t - pareada	5.643607892	-36.90136307	-20.39446316
P-value	0.000316165	3.90647E-11	7.64225E-09
	Bagging es mejor	**C 4.5 es mejor**	**Bagging es mejor**

Tabla A.3.9. Comprobación del mejor algoritmo con la consideración de los tratamientos N1

Potencial N2			
	C4.5 vs. Bagging	**C4.5 vs. Boosting**	**Bagging vs Boosting**
Prom. Dif	0.27206	-0.87359	-1.14565
Desv. Est. Dif	0.082583375	0.09456085	0.122426661
LI	0.212983413	-0.941234757	-1.233228757
LS	0.331136587	-0.805945243	-1.058071243
t - pareada	10.4177053	-29.21435398	-29.59211159
P-value	2.54273E-06	3.14678E-10	2.8062E-10
	Bagging es mejor	**C 4.5 es mejor**	**Bagging es mejor**

Tabla A.3.10. Comprobación del mejor algoritmo con la consideración de los tratamientos N2

Potencial NT			
	C4.5 vs. Bagging	**C4.5 vs. Boosting**	**Bagging vs Boosting**
Prom. Dif	0.23358	-0.90338	-1.13696
Desv. Est. Dif	0.061814054	0.130385538	0.150639262
LI	0.189360889	-0.996652195	-1.244720836
LS	0.277799111	-0.810107805	-1.029199164
t - pareada	11.94946395	-21.90993298	-23.86750421
P-value	7.98166E-07	4.05598E-09	1.90001E-09
	Bagging es mejor	**C 4.5 es mejor**	**Bagging es mejor**

Tabla A.3.11. Comprobación del mejor algoritmo con la consideración de los tratamientos NT

A.4. Análisis de las técnicas de clasificación y sus configuraciones

Objetivo

El presente experimento busca determinar cuál de las configuraciones de los algoritmos estudiados en la presente investigación es más eficiente para el dominio de aplicación.

Procedimiento

Aplicamos la técnica de Holdout Resampling a los datos correspondientes a los periodos 19912 hasta el 20091 para generar un conjunto que consta de 70 por ciento de datos de entrenamiento y 30 por ciento de datos de prueba. Esto nos permitirá generar los modelos para los clasificadores descritos en la tabla A.4.1 . En esta tabla, se definen todas las configuraciones que serán usadas.

ALGORITMO	Opciones	Pruebas
K-Nearest Neighbors (KNN)	K=74,75,76,77,78,79,80,81,82,83,84 (donde se uso k=n^(3/8))	11
Naive Bayes (NB)	Default/KernelEstimator	2
Árboles de decisión (DT)	CF=0.25,M=2 , CF=0.4,M=40	2
Bagging con árboles (BAG-DT)	it=10,15,20, 25 (CF=0.25,M=2) it=10,15,20, 25 (CF=0.4 , M=40)	8
Boosting con árboles (BST-DT)	it=10,15,20, 25 (CF=0.25,M=2) it=10,15,20, 25 (CF=0.4 , M=40)	8
Boosted stumps (BST-STMP)	it=10,15,20, 25	4
Total		35
	10 Holdout Resampling	350

Tabla A.4.1. Tabla descriptiva de las variantes de los algoritmos aplicados en el experimento comparativo entre algoritmos de clasificación

Cabe indicar que las opciones por defecto del algoritmo de árboles de decisión son las que consideran dos instancias como mínimo en las hojas de los árboles de decisión y un factor de confianza de 0.25. Además, la técnica base para los algoritmos de Bagging y Boosting es C4.5.

Experimento

Con este experimento, se pretende responder a la pregunta: ¿cuál de las configuraciones planteadas tiene mayor tasa de precisión para cada algoritmo estudiado (C4.5, Naïve Bayes, KNN, Bagging, Boosting) en nuestro dominio de aplicación. Para responder a esta pregunta, se aplicó el análisis de varianza a los resultados obtenidos y mostrados en la tabla A.4.2(a)(b), que corresponde a la precisión a las tablas A.4.3(a)(b) relativas a la métrica que representa el área bajo la curva ROC.

	corrida 01	corrida 02	corrida 03	corrida 04	corrida 05
Bagging_default_10	0.81056	0.81019	0.81127	0.81067	0.80637
Bagging_default_15	0.81153	0.81015	0.81278	0.81184	0.80794
Bagging_default_20	0.81201	0.81106	0.8143	0.81176	0.80862
Bagging_default_25	0.81247	0.81089	0.81422	0.81247	0.8092
Bagging_opcion_10	0.81641	0.81433	0.81621	0.81495	0.81321
Bagging_opcion_15	0.81594	0.81433	0.81728	0.81542	0.81368
Bagging_opcion_20	0.81579	0.81387	0.81784	0.81565	0.81366
Bagging_opcion_25	0.81641	0.81387	0.8179	0.81575	0.81402
Boosting_default_10	0.79541	0.79752	0.79806	0.79512	0.79256
Boosting_default_15	0.79568	0.79799	0.79905	0.79301	0.79293
Boosting_default_20	0.79463	0.79735	0.7988	0.79407	0.7932
Boosting_default_25	0.79597	0.79822	0.79812	0.79361	0.79355
Boosting_opcion_10	0.80382	0.80455	0.80494	0.80511	0.80316
Boosting_opcion_15	0.80502	0.80405	0.80211	0.80488	0.80279
Boosting_opcion_20	0.80473	0.80506	0.80331	0.80556	0.80256
Boosting_opcion_25	0.80473	0.80387	0.80263	0.80556	0.80347
ibk_74	0.81077	0.81005	0.81348	0.81309	0.81073
ibk_75	0.81063	0.81038	0.81381	0.81352	0.81098
ibk_76	0.81067	0.80978	0.81416	0.81284	0.81089
ibk_77	0.81087	0.81044	0.81426	0.81344	0.81087
ibk_78	0.81071	0.80959	0.81412	0.81275	0.81075
ibk_79	0.81087	0.81027	0.81414	0.81298	0.81079
ibk_80	0.81063	0.80978	0.81395	0.81247	0.81083
ibk_81	0.81067	0.80998	0.8141	0.81292	0.811
ibk_82	0.81083	0.80961	0.81364	0.81257	0.81036
ibk_83	0.8106	0.80965	0.81383	0.8128	0.81058
ibk_84	0.81063	0.80963	0.81342	0.81249	0.81038
DecisionS_10	0.79227	0.79243	0.79636	0.79161	0.79039
DecisionS_15	0.78696	0.79766	0.79762	0.79824	0.79473
DecisionS_20	0.79853	0.79766	0.80304	0.79824	0.79465
DecisionS_25	0.80432	0.80294	0.80428	0.80484	0.80599
C4.5_default	0.81143	0.81261	0.81373	0.81129	0.81007
C4.5_opcion	0.81236	0.81278	0.8136	0.81453	0.81145
NB_Opcion	0.76242	0.76091	0.76531	0.7617	0.76066
NB_default	0.75992	0.75808	0.76223	0.75814	0.75771

Tabla A.4.2 (a). Tasa de precisión de los conjuntos de entrenamiento 1,2,3,4,5

	corrida 06	corrida 07	corrida 08	corrida 09	corrida 10
Bagging_default_10	0.81027	0.8124	0.8081	0.81127	0.81015
Bagging_default_15	0.81189	0.81234	0.80827	0.81255	0.81098
Bagging_default_20	0.81187	0.81255	0.80814	0.81261	0.81153
Bagging_default_25	0.81178	0.81315	0.80841	0.81242	0.81218
Bagging_opcion_10	0.81476	0.81645	0.81211	0.81503	0.81457
Bagging_opcion_15	0.81451	0.81575	0.81216	0.81569	0.81482
Bagging_opcion_20	0.81515	0.81631	0.81182	0.8154	0.81449
Bagging_opcion_25	0.81509	0.81699	0.81242	0.81521	0.8149
Boosting_default_10	0.79547	0.7958	0.79237	0.79481	0.79642
Boosting_default_15	0.79281	0.79326	0.79086	0.79531	0.79564
Boosting_default_20	0.79463	0.79425	0.7909	0.79599	0.7964
Boosting_default_25	0.79396	0.79363	0.79093	0.79622	0.79543
Boosting_opcion_10	0.80688	0.80399	0.80314	0.80403	0.80484
Boosting_opcion_15	0.80571	0.80546	0.80252	0.80496	0.80604
Boosting_opcion_20	0.80651	0.80471	0.80207	0.80542	0.80661
Boosting_opcion_25	0.8061	0.80498	0.80244	0.80488	0.80659
ibk_74	0.81131	0.81253	0.80953	0.81151	0.81395
ibk_75	0.81162	0.81251	0.81017	0.81199	0.8143
ibk_76	0.81131	0.81278	0.80978	0.81201	0.81379
ibk_77	0.81112	0.81282	0.80998	0.81174	0.81435
ibk_78	0.81122	0.81259	0.80994	0.81199	0.81366
ibk_79	0.81164	0.81304	0.81036	0.8117	0.81364
ibk_80	0.81122	0.81296	0.80959	0.81131	0.81321
ibk_81	0.81176	0.81344	0.81029	0.81168	0.81354
ibk_82	0.81156	0.813	0.80986	0.8111	0.81327
ibk_83	0.81176	0.81319	0.80982	0.81151	0.81389
ibk_84	0.81168	0.81288	0.80984	0.81125	0.81323
DecisionS_10	0.78166	0.78549	0.79039	0.78394	0.79303
DecisionS_15	0.79026	0.79824	0.79281	0.79543	0.78857
DecisionS_20	0.79564	0.79506	0.79704	0.79138	0.79539
DecisionS_25	0.80244	0.80515	0.7999	0.80312	0.79494
C4.5_default	0.81184	0.81199	0.80994	0.81211	0.81207
C4.5_opcion	0.81352	0.813	0.81158	0.81251	0.81323
NB_Opcion	0.76068	0.76333	0.76159	0.76211	0.76366
NB_default	0.75828	0.76149	0.75905	0.76099	0.76035

Tabla A.4.2 (b). Tasa de precisión de los conjuntos de entrenamiento 6,7,8,9,10

	corrida 01	corrida 02	corrida 03	corrida 04	corrida 05
Bagging_default_10	0.7939	0.80225	0.80143	0.79103	0.79719
Bagging_default_15	0.79545	0.80425	0.80363	0.79301	0.79911
Bagging_default_20	0.79665	0.80465	0.80439	0.79423	0.80002
Bagging_default_25	0.79729	0.8047	0.80545	0.79468	0.80029
Bagging_opcion_10	0.79859	0.80578	0.80221	0.79622	0.80253
Bagging_opcion_15	0.79879	0.80588	0.80331	0.79637	0.8029
Bagging_opcion_20	0.79902	0.80591	0.80676	0.79676	0.80319
Bagging_opcion_25	0.79928	0.80612	0.80909	0.79686	0.80333
Boosting_default_10	0.77721	0.78146	0.77909	0.77644	0.78024
Boosting_default_15	0.7778	0.78182	0.77982	0.77677	0.78074
Boosting_default_20	0.77794	0.78225	0.78003	0.77667	0.78121
Boosting_default_25	0.77804	0.78219	0.78023	0.77675	0.78122
Boosting_opcion_10	0.7932	0.79777	0.7931	0.79324	0.79425
Boosting_opcion_15	0.79399	0.79817	0.79362	0.79453	0.79581
Boosting_opcion_20	0.79396	0.79848	0.79371	0.79466	0.79583
Boosting_opcion_25	0.79414	0.79839	0.79394	0.79463	0.79609
ibk_74	0.8075	0.81026	0.81044	0.80892	0.80952
ibk_75	0.80746	0.81029	0.81042	0.80894	0.80935
ibk_76	0.8075	0.81031	0.81041	0.80901	0.80922
ibk_77	0.80758	0.81033	0.81047	0.80891	0.80934
ibk_78	0.8076	0.81036	0.81042	0.80883	0.80936
ibk_79	0.80757	0.81029	0.81027	0.80886	0.80933
ibk_80	0.80751	0.81013	0.81008	0.80875	0.80934
ibk_81	0.80744	0.81013	0.81007	0.80872	0.80934
ibk_82	0.80746	0.80995	0.81003	0.80866	0.80933
ibk_83	0.80742	0.80983	0.8101	0.80876	0.80935
ibk_84	0.80734	0.80986	0.8101	0.80878	0.8094
DecisionS_10	0.76161	0.76215	0.76256	0.75611	0.76114
DecisionS_15	0.77614	0.77322	0.77321	0.7718	0.77805
DecisionS_20	0.78139	0.78037	0.78231	0.77474	0.78139
DecisionS_25	0.78552	0.78492	0.78608	0.77894	0.78608
C4.5_default	0.78086	0.77635	0.77747	0.73271	0.77784
C4.5_opcion	0.78962	0.78806	0.78991	0.78367	0.79275
NB_Opcion	0.77367	0.77569	0.77817	0.77192	0.7756
NB_default	0.77075	0.77254	0.77463	0.76774	0.77231

Tabla A.4.3 (a). Área bajo la curva ROC de los conjuntos de entrenamiento 1,2,3,4,5

	corrida 06	corrida 07	corrida 08	corrida 09	corrida 10
Bagging_default_10	0.79732	0.79747	0.79895	0.79694	0.79573
Bagging_default_15	0.80273	0.80216	0.79955	0.7987	0.79649
Bagging_default_20	0.80387	0.80308	0.80099	0.79969	0.79748
Bagging_default_25	0.80438	0.804	0.80215	0.80214	0.79839
Bagging_opcion_10	0.79971	0.80113	0.80239	0.80038	0.79976
Bagging_opcion_15	0.80394	0.80405	0.80321	0.80069	0.80021
Bagging_opcion_20	0.80482	0.80418	0.80318	0.80089	0.80031
Bagging_opcion_25	0.80486	0.80448	0.80369	0.80203	0.80039
Boosting_default_10	0.77789	0.77974	0.77514	0.77988	0.77662
Boosting_default_15	0.77847	0.77968	0.77548	0.78106	0.77705
Boosting_default_20	0.77872	0.78022	0.77557	0.7815	0.77706
Boosting_default_25	0.77897	0.7802	0.77514	0.78175	0.77714
Boosting_opcion_10	0.79559	0.79327	0.79449	0.79767	0.79792
Boosting_opcion_15	0.79666	0.79389	0.79533	0.79844	0.79867
Boosting_opcion_20	0.79671	0.79411	0.79533	0.79879	0.79858
Boosting_opcion_25	0.79681	0.79423	0.79557	0.79873	0.79877
ibk_74	0.81026	0.80936	0.80895	0.80896	0.81181
ibk_75	0.81023	0.80932	0.80899	0.80899	0.81167
ibk_76	0.81022	0.80928	0.80903	0.80902	0.8116
ibk_77	0.81016	0.80937	0.80909	0.80897	0.81159
ibk_78	0.81018	0.80936	0.80932	0.80901	0.81153
ibk_79	0.81029	0.80932	0.8093	0.80886	0.81158
ibk_80	0.81021	0.80928	0.80929	0.80882	0.81151
ibk_81	0.81021	0.8094	0.80929	0.80874	0.81145
ibk_82	0.81015	0.80928	0.80914	0.80873	0.81139
ibk_83	0.81021	0.80942	0.80908	0.80876	0.81131
ibk_84	0.81029	0.80945	0.80912	0.80882	0.81121
DecisionS_10	0.7521	0.75087	0.7596	0.75037	0.76357
DecisionS_15	0.77482	0.77434	0.76988	0.77208	0.77443
DecisionS_20	0.77776	0.7832	0.77877	0.77951	0.78153
DecisionS_25	0.78319	0.787	0.78266	0.78456	0.78253
C4.5_default	0.74368	0.78395	0.73917	0.7812	0.77979
C4.5_opcion	0.79274	0.79076	0.78878	0.79331	0.78989
NB_Opcion	0.77465	0.77448	0.77502	0.77534	0.77706
NB_default	0.77152	0.77141	0.77195	0.77201	0.77327

Tabla A.4.3 (b). Área bajo la curva ROC de los conjuntos de entrenamiento 6,7,8,9,10

En la tabla A.4.4(a), se muestra el resultado del análisis de varianzas (ANOVA), aplicado a los datos de las tablas A.4.2(a) (b)(precisión), con el uso del método de Tukey con 95 por ciento de nivel de confianza. Cabe mencionar que en el análisis de varianza las medias que no comparten una letra son significativamente diferentes.

	N	Media	Agrupación
Bagging_opcion_25	10	0.815256	A
Bagging_opcion_20	10	0.814998	A B
Bagging_opcion_15	10	0.814958	A B C
Bagging_opcion_10	10	0.814803	A B C D
C4.5_opcion	10	0.812856	A B C D E
ibk_75	10	0.811991	A B C D E
ibk_77	10	0.811989	A B C D E
ibk_79	10	0.811943	A B C D E
ibk_81	10	0.811938	A B C D E
ibk_76	10	0.811801	B C D E
ibk_83	10	0.811763	B C D E
ibk_78	10	0.811732	B C D E
Bagging_default_25	10	0.811719	B C D E
C4.5_default	10	0.811708	B C D E
ibk_74	10	0.811695	B C D E
ibk_80	10	0.811595	B C D E
ibk_82	10	0.811580	B C D E
ibk_84	10	0.811543	C D E
Bagging_default_20	10	0.811445	D E
Bagging_default_15	10	0.811027	E
Bagging_default_10	10	0.810125	E
Boosting_opcion_20	10	0.804654	F
Boosting_opcion_25	10	0.804525	F
Boosting_opcion_10	10	0.804446	F
Boosting_opcion_15	10	0.804354	F
DecisionS_25	10	0.802792	F
DecisionS_20	10	0.796663	G
Boosting_default_10	10	0.795354	G
Boosting_default_20	10	0.795022	G
Boosting_default_25	10	0.794964	G
Boosting_default_15	10	0.794654	G
DecisionS_15	10	0.794052	G
DecisionS_10	10	0.789757	H
NB_Opcion	10	0.762237	I
NB_default	10	0.759624	I

Tabla A.4.4 (a). Resultados del ANOVA Tukey para la métrica de precisión

En la tabla A.4.4(b), se muestra el resultado del análisis de varianzas (ANOVA), aplicado a los datos de las tablas A.4.3(a) (b)(AUC), con el uso del método de Tukey con 95 por ciento de nivel de confianza.

	N	Media	Agrupación
ibk_74	10	0.809598	A
ibk_78	10	0.809597	A
ibk_77	10	0.809581	A B
ibk_79	10	0.809567	A B
ibk_75	10	0.809566	A B
ibk_76	10	0.809560	A B
ibk_80	10	0.809492	A B
ibk_81	10	0.809479	A B
ibk_84	10	0.809437	A B
ibk_83	10	0.809424	A B
ibk_82	10	0.809412	A B
Bagging_opcion_25	10	0.803013	A B C
Bagging_opcion_20	10	0.802502	B C D
Bagging_opcion_15	10	0.801935	C D E
Bagging_default_25	10	0.801347	C D E
Bagging_opcion_10	10	0.800870	C D E
Bagging_default_20	10	0.800505	C D E
Bagging_default_15	10	0.799508	C D E
Bagging_default_10	10	0.797221	C D E
Boosting_opcion_25	10	0.796130	C D E F
Boosting_opcion_20	10	0.796016	C D E F
Boosting_opcion_15	10	0.795911	D E F
Boosting_opcion_10	10	0.795050	E F
C4.5_opcion	10	0.789949	F G
DecisionS_25	10	0.784148	G H
DecisionS_20	10	0.780097	H I
Boosting_default_25	10	0.779163	H I
Boosting_default_20	10	0.779117	H I
Boosting_default_15	10	0.778869	H I J
Boosting_default_10	10	0.778371	H I J
NB_Opcion	10	0.775160	I J
DecisionS_15	10	0.773797	I J K
NB_default	10	0.771813	J K
C4.5_default	10	0.767302	K
DecisionS_10	10	0.758008	L

Tabla A.4.4 (b). Resultados del ANOVA Tukey para la métrica de AUC

En ambos casos, como se puede observar en la tabla A.4.5, el análisis de varianza reporta un p-value nulo(P=0), lo que indica que se rechaza la hipótesis nula, y a la vez, determina que si hay diferencia significativa entre los métodos.

Para el caso de la precisión, el mejor método es Bagging 25, y para el caso de la métrica AUC, el mejor método es el de vecinos cercanos con 74 y 78 vecinos respectivamente.

Métrica	P-value	Hay diferencia significativa entre los algoritmos
Precisión	0	Si
AUC	0	Si

Tabla A.4.5. Resultado de la prueba del ANOVA

A.5. Prueba del sistema de recomendación en el ámbito de la matrícula

En la presente sección, se muestra la experimentación y puesta en marcha del sistema recomendador. Este sistema después de ser implementado usando los datos hasta el semestre académico 20092 y con los resultados correspondientes al capítulo de los experimentos, fue probado para la matrícula del periodo correspondiente al semestre académico 20101.

En dicho periodo, los estudiantes hábiles para la matricula fueron de 804 estudiantes, de ellos se eligió un total de 50 estudiantes para llevar a cabo la prueba. Después de haberles explicado las características del sistema SPRS, se les comunicó que antes de la matrícula, si ellos consideraban conveniente, podían hacer uso del sistema. Del total de estudiantes seleccionados, solo 39 lo usaron. Este número de estudiantes generaron 198 instancias para la prueba de nuestro sistema. Debido a que la matricula es vía web y a que el experimento tenía que simular las condiciones reales, los estudiantes hicieron su matrícula sin supervisión y usando solo el sistema de recomendación.

Al hacer el contraste entre las predicciones del sistema y los resultados reales se obtuvo una precisión de 85.36%. La interpretación de esta tasa de acierto debe ser considerada tomando en cuenta lo siguiente:

- El porcentaje de los estudiantes que al usar el sistema obtuvieron una recomendación de aprobado en sus asignaturas y que al finalizar el semestre realmente aprobaron fueron el 82.32 % del total.
- El porcentaje de los estudiantes que al usar el sistema obtuvieron una recomendación de desaprobado en sus asignaturas y que al finalizar el semestre realmente desaprobaron fueron el 3.03% del total.

Para este análisis se consideró el factor de confianza de la predicción. Este factor representa el porcentaje de aciertos del sistema, cuando un alumno lleva una asignatura que previamente ha sido cursada por otros alumnos con características similares y en situaciones análogas.

En la prueba del sistema de recomendación el estudiante tuvo la oportunidad de visualizar el factor de confianza que el sistema le proporcionó y basar en ella su decisión de matricularse o no en determinada asignatura.

Los resultados de las tablas A.5.1 se pueden interpretar de la siguiente manera:

- Observamos que cuando el sistema predice aprobado con un factor de confianza entre [0.75-1] y los alumnos realmente aprobaron, el número de aciertos del sistema fue de 154, mientras que el número de desaciertos fue de 20. Esto indica que cuando se tiene un mayor factor de confianza el sistema es más eficaz. Cabe

destacar que la mayoría de los desaciertos se dieron debido a que los alumnos tuvieron un comportamiento fuera de lo habitual o simplemente abandonaron la asignatura.

- Cuando la predicción es desaprobado, observamos que en la mayoría de casos el sistema les da un factor de confianza bajo, entre [0.5-0.7], valores muy cercanos al umbral de decisión, lo cual indica que en estos casos el sistema puede realizar predicciones ambiguas.

FC	**Predicción**	**Aprobado (Real)**	**Desaprobado (Real)**
		Cantidad de registros	Cantidad de Registros
[0.50, 0.70 >	Desaprobado	3	5
[0.70 , 0.80 >	Desaprobado	1	1
[0.50 , 0.75 >	Aprobado	9	5
[0.75 , 1.00 >	Aprobado	154	20

Tabla A.5.1. Resultados de las predicciones del sistema

Referencias Bibliográficas

[Adom-05] Adomavicius, G., & Tuzhilin, A. (2005): *Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions.* IEEE Transactions on Knowledge and Data Engineering, 17 (6), 734-749.

[Ahad-91] D. Aha and D. Kibler,(1991): *Instance-based learning algorithms*, Machine Learning, vol.6, pp. 37-66.

[Alpa-10] Alpaydin, E. (2010). *Introduccion to Machine learning* Second Edition. Mit Express

[Anan-09] N. Anand , G. Kumar ; (2009): *Improving Academic Performance Students by applying data Mining Techniques*. European Journal Of Scientific research, 34, no. 4, 526-534

[Baez-99] R. Baeza-Yates, B. Ribeiro Neto; (1999): Modern Information Retrieval. Harlow, UK: Addison-Wesley.

[Bala-97] M. Balabanovic Y. Shoham; (1997): *Fab: Content-Based Collaborative Recommendation*. Communications of the ACM archive, 40 (3), 66-72.

[Bala-98] M. Balabanovic; (1998): *Exploring versus Exploiting when Learning User Models for Text Representation,* User Modeling and User-Adapted Interaction 8(1-2), 71-102.

[Basu-98] C. Basu, H. Hirsh, and W. Cohen, (1998): *Recommendation as Classification: Using Social and Content-Based Information in Recommendation.* Recommender Systems. Papers from 1998 Workshop, Technical Report WS-98-08, AAAI Press .

[Bau-99] Eric Bauer y Ron Kohavi ; (1999): *An empirical comparison of voting classi_cation algorithms: Bagging, boosting, and variants*. Machine Learning, 36(1- 2):105.139.

[Bill-00] D. Billsus, M. Pazzani ; (2000): *User Modeling for Adaptive News Access*. User Modeling and User-Adapted Interaction, 10 (2-3), 147-180.

[Brav-07] J. Bravo, C. Vialardi and A. Ortigosa; (2007): *A Problem-Oriented Method for Supporting AEH Authors through Data Mining*. Proceedings of International Workshop on Applying Data Mining in e-Learning (ADML07) held at the Second European Conference on Technology Enhanced Learning (EC-TEL 2007), p. 53-62, Creta, Grecia, (ISSN: 1613-0073).

[Brav-08a] J. Bravo, C. Vialardi and A. Ortigosa*; (2008): ASquare: A Powerful Evaluation Tool for Adaptive Hypermedia Course System. Proceedings of Hypertext 2008 Conference, p. 219-220, Pittsburgh, EE.UU., (ISBN: 978-1-59593-998-2).*

[Brav-08b] J. Bravo, C. Vialardi and A. Ortigosa; (2008): *Using Decision Trees for Discovering Problems on Adaptive Courses*. Proceedings of E-Learn 2008: World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education, p. 268-277 ,Las Vegas, EE.UU. (ISBN: 1-880094-66-5).

[Bree-98] J.S. Breese, D. Heckerman and C. Kadie ; (1998): *Empirical Analysis of Predictive Algorithms for Collaborative Filtering.* Proc. 14th Conf. Uncertainty in Artificial Intelligence, .

[Brei-84] L. Breiman, J. Friedman, R. Olshen, and C. Stone ;(1984): *Classification and Regression Trees*. Belmont, Calif.: Wadsworth Int'l,

[Brei-96] L. Breiman ; (1996): *Bagging predictors*. Machine Learning, 24(2):123–140.

[Brei-98] L. Breiman ; (1998): *Arcing classifiers*. The Annals of Statistics, 26(3):801.849.

[Bres-08] V.P. Bresfelean, M. Bresfelean, N. Ghisoiu, and C. Comes ; (2008): *Determining students' academic failure profile founded on data mining methods*. Proceedings of the *ITI 2008 30th Int. Conf. on Information Technology Interfaces,* June 23-26, Cavtat, Croatia

[Bres-97] L.A. Breslow, and D.W. Aha ;(1997): *Comparing tree-simplification procedures*. In Proc of the 6th Int Workshop on AI &Statistics .Ft. Lauderdale, Florida: unpublished.

[Bunt-92] W.L. Buntine and T. Niblett ; (1992): *A Further Comparison of Splitting Rules for Decision-Tree Induction.* Machine Learning, vol. 8, no. 1, pp. 75-85,

[Burk-02] R Burke; (2002): *Hybrid Recommender Systems: Survey and Experiments*. User Modeling and User-Adapted Interaction, 12 (4), 331-370.

[Caru-04] R. Caruana, A. Niculescu-Mizil; (2004): *Data mining in metric space: An empirical analysis of suppervised learning performance criteria*. Knowledge Discovery and Data Mining (KDD'04).

[Cast-08] Emilio J. Castellano and L. Martínez; (2008): *ORIEB, A CRS For Academic Orientation Using Qualitative Assessments*. Proceedings of the IADIS International Conference E-Learning, pp 38-42.

[Cest-91] B. Cestnik and I. Bratko ;(1991): *On Estimating Probabilities in Tree Pruning* Machine Learning. EWSL-91, Y. Kodratoff, ed., Lecture Notes in Artificial Intelligence. Berlin: Springer-Verlag, no. 482, pp. 138-150.

[Clay-99] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin; (1999): *Combining Content-Based and Collaborative Filters in an Online Newspaper.* Proc. ACM SIGIR '99 Workshop Recommender Systems: Algorithms and Evaluation.

[Cort-08] P. Cortez, A. Silva; (2008): *Using Data Mining to Predict Secondary School Student Performance*. In A. Brito and J. Teixeira Eds., Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008) pp. 5-12, Oporto, Portugal, April, 2008. EUROSIS, ISBN:978-9077381-39-7.

[Dahl-98] B. Dahlen, J. Konstan, J. Herlocker, N. Good, A. Borchers and J. Riedl; (1998): *Jumpstarting movielens: User benefits of starting a collaborative filtering system with "dead data.* TR 98-017. University of Minnesota.

[Dave-98] T. Davenport, L. Prusak ; (1998): *Working Knowledge: How Organizations Manage What They Know*. Harvard Business School Press. <http://www.gestiondelconocimiento.com/conceptos_diferenciaentredato.htm>

[Dela-05] N. Delavari, M. Beikzadeh, S. Amnuaisuk; (2005): *Application of Enhanced Analysis Model for Data Mining Processes in Higher Educational System*. 6th Annual International Conference: ITEHT July 7-9. Juan Dolio, Dominican Republic.

[Desh-04] M. Deshpande, G. Karypis ; (2004): *Item-Based Top-N Recommendation Algorithms*. ACM Trans. Information Systems, vol. 22, no. 1, pp. 143-177.

[Duda-01] R. Duda, P. Hart, and D. Stork; (2001): *Pattern Classification*. New York, NY,USA: John Wiley.

[Edel-00] H. Edelstein; (2000): *Two Cross Corporation*, SPSS white paper-executive briefing

[Efro-83] B. Efron and G. Gong ; (1983): *A Leisurely Look at the Bootstrap, the Jackknife, and Cross-Validation*. The American Statistician, vol. 37, pp. 36-48.

[Elma-00] Elmasri, R. y Navathe,(2000): S. Fundamentals of Databases System. Addinson-Wesley, 3.a edición.

[Enas-86] G. Enas, S. Choi ; (1986): *Choice of the Smoothing Parameter and Efficiency of k-nearest Neighbor Classification.* Computers and Mathematics with Applications, Vol. 12, 235-244,

[Espo- 97] F. Esposito, D. Malerba and G. Semeraro; (1997): *A comparative analysis of methods for pruning decisión trees*. IEEE Transactions on Pattern Analysis and Machine Intelligence ,476–491.

[Fayy-97] U. Fayyad, G.Piatetsky-Shapiri and P. Smyth; (1997): *From Data mining to knowledge Discovery in Databases*. AAAI, pp. 37-54.

[Feld-03] K. Feldman; (2003): *Mining the Biomedical Literature using Semantic Analysis and Neural Language Processing Techniques, a link analysis approaches*. ClearForest Corporation. New York.

[Freu-95] Y. Freund and R. Schapire; (1995): *A decision-theoretic generalization of on-line learning and an application to boosting*. In Proc. 2nd European Conference on Computational Learning Theory, 23.37.

[Gold-92] D. Goldberg, et al; (1992): *Using collaborative filtering to weave an information tapestry*. Communication of ACM, 35(12), 61-70.

[Han-01a] J. Han and M. Kamber; (2001): *Data Mining: Concepts and Techniques*. Simon Fraser University, Morgan Kaufmann publishers. ISBN 1-55860-489-8.

[Han-01b] J. Han; (2002): *How can Data mining Help Bia-Data Analysis*. BIOKllDO2:Woskhop on data mining in Bioinformatics.

[Hand-06] J. Hand and K. Micheline; (2006): *Data Mining: Concepts and Techniques* Morgan Kaufmann Publishers - Academic Press,

[Hayk-99] Simon Haykin; (1999): *Neural Networks: A Comprehensive Foundation*. Prentice Hall.

[Herl-04] J. Herlocker, J. A. Konstan, L. Terveen, and J. Riedl; (2004): *Evaluating Collaborative Filtering Recommender Systems*. ACM Transactions on Information Systems, 22 (1), 5-53.

[Holt-93] R.C. Holte; (1993): *Very Simple Classification Rules Perform Well on Most Commonly Used Datasets.* Machine Learning, vol. 11, no. 1, pp. 63-90, 1993.

[Huan-04] Z. Huang, H. Chen, D. Zeng; (2004): *Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering* ACM Trans. Information Systems, vol. 22, no. 1, pp. 116-142.

[Kitt-82] J. Kittler and P.A. Devijver ; (1982): *Statistical Properties of Error Estimators in Performance Assessment of Recognition Systems* IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 4, no. 2, pp. 215-220.

[Koha-95] R. Kohavi and G.H. John; (1995): *Automatic Parameter Selection by Minimizing Estimated Error.* Proc. 12th Int'l Conf. on Machine Learning, Lake Tahoe, Calif., pp. 304-312

[Kots-04] S. Kotsiantis, C. Pierrakeas and P. Pintelas; (2004): *Predicting Students' Performance in Distance Learning Using Machine Learning Techniques*. Applied Artificial Intelligence (AAI), 18, no. 5, 411–426.

[Krul-97] B. Krulwich, C. Burkey; (1997): *The Info finder Agent: Learning User Interests Through Heuristic Phrase Extraction*. IEEE Intelligent Systems and Their Applications, 12 (5), 22-27.

[Lee -01] W. Lee; (2001): *Collaborative Learning for Recommender Systems.* Proceedings of the 18th International Conference on Machine Learning, (pp. 314-321). Williamstown, MA, USA.

[Lehm-98] E. Lehmann, G. Casella; (1998): *Theory of Point Estimation*. Springer, page 54.

[Lewi -94] D.A. Lewis, W.A. Gale ;(1994): *A Sequential Algorithm for Training Text Classifiers*. Proceedings of the 17th ACM Conference on Research and Development in Information Retrieval (SIGIR 1994), (pp. 3-12). Dublin, Ireland.

[Luan-01] J. Luan; (2001): *Data mining and Knowledge Management, A System Analysis for Establishing a Tiered Knowledge Management Model (TKMM)*, Proceedings of Air Forum, Toronto. Canada.

[Luan-02a] J. Luan; (2002): *Data Mining and Knowledge Management in Higher Education-Potential Applications.* Proceedings of AIR Forum, Toronto, Canada.

[Luan-02b] J.Luan ; (2002): *Data Mining Application in Higher Education*. SPSS Executive Report.

[Mani-97] H. Manila; (1997): *Methods and problems in data mining*. International Conference on Database theory. Delphi, Grecia: Proceedings. Springer Verlag.

[Ming-87] J. Mingers; (1987): *Expert Systems—Rule Induction With Statistical Data* Journal of Operational Research Society, vol. 38, pp. 39-47.

[Ming-89] J. Mingers; (1989): *An Empirical Comparison of Pruning Methods for Decision Tree Induction.* Machine Learning, vol. 4, no. 2, pp. 227-243.

[Mitc-97] T. Mitchell; (1997): *Machine Learning.* Portland: WCB/McGraw-Hill.

[Moba-96] B. Mobasher. N. Jain, E. Han and J. Srivastava; (1996): *Web Mining: Pattern Discovery from World Wide Web Transactions.* Technical Report TR96-OS0, Department of Computer Science. University of Minnesota.

[Moon-98] R. J. Mooney, P.N. Bennett, and L. Roy, L.; (1998): *Book Recommending Using Text Categorization with Extracted Information*. Proceedings of the AAAI 1998 Workshop on Recommender Systems, (pp. 70-74). Madison, WI, USA.

[Moon-99] R.J. Mooney and L. Roy; (1999): *Content-Based Book Recommending Using Learning for Text Categorization* Proc. ACM SIGIR '99 Workshop Recommender Systems: Algorithms and Evaluation..

[Moon-00] R. Mooney and L. Roy; (2000): *Content-based Book Recommending Using Learning for Text Categorization*. Proceedings of the 5th ACM Conference on Digital Libraries, (pp. 195-240). San Antonio, TX, USA.

[Nguy-07] T. Nguyen, P. Janecek, and P. Haddawy; (2007): *A Comparative Analysis of Techniques for Predicting Academic Performance*. Proceedings of the 37th ASEE/IEEE Frontiers in Education Conference October 10 – 13, Milwaukee, WI

[Opit-99] D. Opitz, R. Maclin; (1999): *Popular Ensemble Methods: An Empirical Study* Journal of Artificial Intelligence Research 11 169-198

[Pazz-97] M. Pazzani, D. Billsus; (1997): *Learning and Revising User Profiles: The Identification of Interesting Websites*. Machine Learning, 27 (3), 313-331.

[Pazz-99] M. Pazzani; (1999): *A Framework for Collaborative, Content-based, and Demographic Filtering*. Artificial Intelligence Review, 13 (5-6), 393-408.

[Pazz-00] D. Billsus and M. Pazzani; (2000): *User Modeling for Adaptive News Access.* User Modeling and User-Adapted Interaction, vol. 10, nos. 2-3, pp. 147-180.

[Pear-88] Judea Pearl; (1998): *Probabilistic reasoning in intelligent systems networks of plausible inference*. Morgan Kaufmann.

[Quin-87] J.R. Quinlan; (1987): *Simplifying Decision Trees* Int'l J. Man-Machine Studies, vol. 27, pp. 221-234.

[Quin-93] J.R. Quinlan; (1993): *C4.5: Programs for Machine Learning*. San Mateo, Calif.: Morgan Kaufmann.

[Rada-06] Q. A. Al-Radaideh, E. M. Al-Shawakfa, and M. I. Al-Najjar; (2006): *Mining Student Data using Decision Trees*. International Arab Conference on Information Technology(ACIT'2006), Yarmouk University, Jordan.

[Rama-10] M. Ramaswami and R. Bhaskaran; (2010): *A CHAID Based Performance Prediction Model in Educational Data Mining* IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 1, No. 1, January

[Rash-02] A.M. Rashid, I. Albert, D. Cosley, S.K. Lam, S.M. McNee, J.A. Konstan, and J. Riedl; (2002): *Getting to Know You: Learning New User Preferences in Recommender Systems* Proc. Int'l Conf. Intelligent User Interfaces.

[Resn-94] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl; (1994): *GroupLens: An Open Architecture for Collaborative Filtering of Netnews*. Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work (CSCW 1994), (pp. 175-186). Chapel Hill, NC, USA.

[Rich-79] E. Rich; (1979): *User Modeling via Stereotypes*. Cognitive Science, vol. 3, no. 4, pp. 329-354.

[Rock-08] L. Rokach,O. Maimon; (2008): *Data Mining con Decision Trees:Theory and Applications*. World Scientific Publishing Company

[Rome-07] C. Romero, S. Ventura, J.A. Delgado, P.De Bra; (2007): *Personalized Links Recommendation Based on Data Mining in Adaptive Educational Hypermedia Systems*. EC-TEL 2007: 292-306

[Rome-08] C. Romero, S. Ventura, P. Espejo, C. Hervas; (2008): *Data mining algorithms to classify students*. Educational Data Mining Conference EDM 2008. Montreal. Junio 20-21. Pag. 182-185.

[Sarw-98] B. Sarwar, et al; (1998): *Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System*. Proceedings of the ACM Conferences on computer Supported Cooperative Work (CSCW) p. 345-354, Seattle, Washington, November 14th-18th.

[Sarw-01] B. Sarwar, G. Karypis, J.A. Konstan and J. Riedl; (2001): *Item-Based Collaborative Filtering Recommendation Algorithms*. Proceedings of the 10th International World Wide Web Conference (WWW 2001), (pp. 285-295). Hong Kong, China.

[Scha-90] R. E. Schapire; (1990): The strength of weak learnability. Machine Learning, 5(2):197.227.

[Scha-93] C. Schaffer; (1993): *Overfitting Avoidance As Bias* Machine Learning, vol. 10, no. 2, pp.153-178.

[Scha-01] J.A. Konstan and J. Riedl; (2001): *E-Commerce Recommender Applications.* Data Mining and Knowledge Discovery, vol 5 nos.1/2, pp 115-152. (Also appeared as a chapter in Kohavi and Provost (editors) (2001), Applications of Data Mining to Electronic Commerce, Kluwer Academic Publishers.)

[Scha-05] J.B. Schafer; (2005): *The application of data-mining to recommender systems*. J. Wang (Eds.), Encyclopedia of data warehousing and mining, Hershey, PA. Idea Group p. 44-48.

[Shar-95] U. Shardanand and P. Maes; (1995): *Social Information Filtering: Algorithms for Automating 'Word of Mouth'*. Proceedings of the Conference on Human Factors in Computing Systems (CHI 1995), (pp. 210-217). San Francisco, CA, USA.

[Shih-04] Y. Shih; (2004): *Extending Traditional Collaborative Filtering with Attributes Extraction to Recommender New Products*. Master´s Thesis. Deparment of Business Administration. National Sun Yat-sen University, Taiwan.Availabe via ethesys(digital Library) under URN 91421019

[Sira-09] F. Siraj, and M. A. Abdoulha; (2009): *Uncovering Hidden Information Within University's Student Enrollment Data Using Data Mining.* MASAUM Journal of Computing, Volume 1 Issue 2, September

[Spec-00] G. Specht; (2000): *Information Filtering and Personalisation in Databases using Gaussian Curves* Proc. of the IEEE 4th Int. Database Engineering and Application Symposium (IDEAS 2000), 18.-20, Yokohama, Japan, IEEE Computer Society, 2000, pp. 16-24)

[Smyt-00] B. Smyth and P. Cotter; (2000): *A Personalized TV Listings Service for the Digital TV Age*. Knowledge-Based Systems 13: 53-59.

[Tan-06] P. Tan, M. Steinbach and V. Kumar; (2006): *Introduction to Data Mining*. Pearson-Addison Wesley publishers, Boston, USA, 2006.

[Terv-97] L. Terveen, et al; (1997): *Phoaks: A system for sharing recommendations*. Communication of ACM, 40(3), p.59-62

[Terv-01] L. Terveen, W. Hill; (2001): *Beyond Recommender Systems: Helping People Help Each Other*. In J. M. Carroll, Human-Computer Interaction in the New Millennium (pp. 487-509). New York: Addison-Wesley.

[Tran-00] T. Tran and R. Cohen; (2000): *Hybrid Recommender Systems for Electronic Commerce.* Proc. Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, Technical Report WS-00-04, AAAI Press.

[Unga-98] L.H. Ungar and D.P. Foster; (1998): *Clustering Methods for Collaborative Filtering*," Proc. Recommender Systems, Papers from 1998 Workshop, Technical Report WS-98-08.

[Unga-98] L.H. Ungar and D.P. Foster; (1998): *Clustering Methods for Collaborative Filtering*. Proceedings of the AAAI 1998 Workshop on Recommendation Systems. Madison, WI, USA.

[Vapn-95] V. Vapnik; (1995): *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.

[Vial-07] C. Vialardi, J.Bravo and A. Ortigosa; (2007): *Empowering AEH Authors Using Data Mining Techniques.* Proceedings of Fifth International Workshop on Authoring of Adaptive and Adaptable Hypermedia (A3H 2007) held at the 11th International Conference on User Modeling (UM2007), p. 33-43, Corfu, Grecia

[Vial-09a] C. Vialardi, J. Bravo and A. Ortigosa; (2009): *Improving AEH Courses through Log Analysis*. Journal of Universal Computer Science J.UCS, Vol. 14, Issue 17, p. 2777-2798, Graz, Austria, (ISSN: 0948-6968).

[Vial-09b] C. Vialardi, J. Bravo, L. Shafti and A. Ortigosa; (2009): *Recommendation in Higher Education Using Data Mining Techniques* url, bibtex presentación Proceedings of Second Educational Data Mining conference., p. 190-199, Universidad de Córdoba, Córdoba, Spain, (ISBN: 978-84-613-2308-1).

[Vial-09c] C. Vialardi, J. Chue, A. Barrientos, D. Victoria, J. Estrella, A. Ortigosa; (2009): *La predicción del rendimiento académico en la Universidad: Una aplicación de sistemas de recomendación basada en minería de datos*. II EIM@ - 2009 II Encontro Interactivo de Matemática Aplicada. Universidad Autónoma Metropolitana. México - Brazil – Perú

[Vial-10a] C. Vialardi, A. Barrientos, D. Victoria, J. Estrella and A. Ortigosa (2010): *A Methodology for Student Performance Prediction Based on Data Mining*. In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (pp. 388-393). Chesapeake, VA: AACE. ISBN: 1-880094-81-9

[Vial-10b] C. Vialardi, J. Chue A. Barrientos, D. Victoria, J. Estrella, A. Ortigosa; (2010): *A Case Study: Data Mining Applied to Student Enrollment* Proceedings of Second Educational Data Mining conference., p. 333-335, Pittsburg Pensilvania, Usa. ISBN:978-0-615-37529-8

[Waiy-03] K. Waiyamai; (2003): *Improving Quality of Graduate Student by Data Mining*. Dept. of Computer Engineering, Faculty of Engineering, Kasetsat University, Bangkok, Thailand.

[Wild-95] C. Wild and G. Weber; (1995): Introduction to Probability and Statistics University of Auckland.

[Witt-00] Ian H. Witten and Eibe Frank; (2000): *Data Mining: practical machine learning tools and techniques with java implementations*. Morgan Kaufmann Publishers - Academic Press.

[Witt-05] Ian H. Witten and Eibe Frank; (2005): *Data Mining: practical machine learning tools and techniques*. Morgan Kaufmann Publishers - Elsevier, second edition.

[Wolp-92] D.H. Wolpert; (1992): *Stacked generalization.* Neural Networks 5, 241-259

[Ye-07] Ye Wang, Bin Bi; (2007): *A Solution to PAKDD'07 Data Mining Competition*. PAKDD 2007 Data Mining Competition. Under the supervision of: Dehong Qiu

[Yu-04] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, H.P. Kriegel; (2004): *Probabilistic Memory-Based Collaborative Filtering.* IEEE Trans. Knowledge and Data Eng., vol. 16, no. 1, pp. 56-69.

Lista de abreviaturas

0SE	Regla 0 pruning CCP
1SE	Regla 1 pruning CCP
ACC	Accuracy
AIT	Asian Institute of Technology
ANOVA	Análisis de Varianza
APROB	Clase que corresponde a cuando un estudiante aprobó una asignatura
AUC	Area Under Curve
BAG-E	Bagging Estratificado
BAGGING	Boostrop Aggregaring
BAG-P	Bagging Probabilistico
BEP	Break even point
BI	Inteligencia de Negocios
C4.5	Algoritmo de arbol de decisión, sucesor de ID3
C5.0	Algoritmo sucesor de C4.5
CART	Clasification and Regresion Trees
CBPU	Content Based Profile User
CCP	Cost-Complexity Pruning
CF	Confidence Factor
CGPA2	Promedio Ponderado Acumulado del segundo año
CRISP	**Cross-Industry Standard Process for Data Mining**
CSV	Formato de Texto separado por comas
CTU	Can The University
CUP	Collaborative User Profile
CVP	Critical Value Pruning
DL	Dependencia Lineal
DM	Data Mining
DM-EDU	Data Mining Education(Jing Luan)
DT	Decisión Trees
EA	Equivalencia Atras
EBP	Error Based-Pruning
EE	Error Estimado
FC	Filtrado Colaborativo
GPA	Grade Point Average
IBK	Instance Based Classifier
ID3	Inductive Decisión Tree

J48	Implementación en Weka del algoritmo C4.5(versión java)
KDD	Knowledge Discovery in Databases
K-MEANS	K-medias , Algorithm of clustering
KNN	K nearest Neighbors
LFT	Lift
LSI	Latent Semantic Indexing
MAE	Mean absolute Error
MAP	Hipótesis Máxima a Posteriori
MDL	Minimum Description Lenght
MEP	Minimum Error Pruning
ML	Machine Learning
MSE	Mean Squared Error
MXE	Mean Cross Entropy
N1	Potencial Considerando una sola asignatura requisito
N2	Potencial Considerando dos asignaturas requisitos
NB	Naïve Bayes
NMAE	Normalized Mean Absolute Error
NN	Neural Netwok
OLAP	Online Analytical Processing
OPGT	Árbol de Crecimiento Podado Óptimamente
OPTT	Árbol de Entrenamiento Podado
ORIEB	Sistemas de Orientación para el bachillerato
PEP	Pessimistic Error Pruning
PPA	Promedio Ponderado Acumulado
PRF	F-Score
R	Lenguaje y entorno de programación para análisis estadístico y gráfico.
REP	Reduced Error Pruning
RF	Random Forest
RMINER	Software de Minería de Datos de código libre
RMS	Raíz del Error Cuadrático Medio
ROC	Receiver Operating Characteristic
SE	Standar Error
SIM	Similitud
SPRS	Student Prediction Recomended System
SQL	Structured Query Language
SR	Subconjunto de Reglas
SUSP	Clase que corresponde a cuando un estudiante desaprobó una asignatura
SVM	Support Vector Machine

TE	Tasa de Error
TEE	Tasa de Error Estimada
TEP	Tasa Error Pesimista
TF-IDF	Frequency/Inverse Document Frequency
TOP-N	Lista de N-Items
WEKA	Software de minería de datos, de código libre