Universidad Autónoma de Madrid
Escuela Politécnica Superior

Doble Grado en Ingeniería Informática y Matemáticas

Trabajo Fin de Grado

SISTEMA INFORMÁTICO PARA EL ANÁLISIS DE DATOS EN ENTORNOS EDUCATIVOS

Autor: Lara Olmos Camarena
Tutor: Ruth Cobos Pérez

Febrero 2018

SISTEMA INFORMÁTICO PARA EL ANÁLISIS DE DATOS EN ENTORNOS EDUCATIVOS

Autor: Lara Olmos Camarena
Tutor: Ruth Cobos Pérez

Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Febrero 2018

Resumen

Resumen

El área de las analíticas de aprendizaje es un tema de estudio emergente relacionado con el desarrollo de métodos para la exploración, análisis y visualización de datos procedentes de entornos educativos en línea con la intención de mejorar los procesos de enseñanza-aprendizaje y los entornos educativos.

Los cursos MOOC (*Massive Open Online Course*) permiten recopilar gran cantidad de datos y motivan estudios que predicen la adquisición de certificado o abandono de los estudiantes planteando una clasificación de los mismos con técnicas de Aprendizaje Automático. Como variables de entrada se consideran indicadores de interacción con las plataformas e indicadores que reflejan la actividad, constancia y eficacia de los estudiantes a lo largo de su aprendizaje, todos ellos obtenidos del procesamiento de eventos de navegación (*clickstreams*), entre otros.

El objetivo de este Trabajo de Fin de Grado consiste en la creación de un sistema informático a partir de la herramienta *edX-MAS* (*edX-MAS: Model Analyzer System for edX MOOC*) que permita aplicar procesos de Minería de Datos de Educación y Data Science a los datos de los cursos MOOC de UAMx con nuevos algoritmos de aprendizaje automático e indicadores para predecir qué estudiantes aprobarán (y por tanto obtendrán un certificado) o abandonarán el curso (predicción a *posteriori*).

El sistema añade extracción, preprocesamiento y exportado de datos para la definición de abandono basada en reglas de clasificación. Incluye evaluación y visualización de las estadísticas de los modelos predictivos y exportado para análisis posteriores.

Palabras Clave

Analíticas de Aprendizaje, Cursos Online Masivos y Abiertos, Minería de Datos en Educación, Modelos Predictivos, Aprendizaje Automático.

Abstract

The area of learning analytics is an emerging study topic related to the development of methods for exploring, analyzing and visualizing data from online educational environments with the intention of improving teaching-learning processes and learning environments.

MOOC courses (*Massive Open Online Course*) allow the collection of a large amount of data and motivate studies that predict the acquisition of certificates or dropout of students by proposing a classification of them with Machine Learning techniques. Input variables are considered as indicators of interaction with platforms and indicators that reflect the activity, constancy and effectiveness of students throughout their learning, all of them obtained from the processing of navigation events (*clickstreams*), among others.

The objective of this end-of-grade paper is to create a computing system from the *edX-MAS* tool (*edX-MAS: Model Analyzer System for edX MOOC*) to implement Educational Data Mining and Data Science processes to UAMx data courses with new Machine Learning algorithms and indicators to predict which students are going to pass (and also gain a certificate) or left the course (*a posteriori* prediction).

The system adds extraction, pre-processing and export of data for the definition of dropout based on classification rules. It includes evaluation and visualization of predictive model statistics and allows to export them for further analysis.

Key Words

Learning Analytics, Massive Online Free Curses, Educational Data Mining, Predictive Models, Machine Learning.

Agradecimientos

Agradecer a Ruth Cobos por la oportunidad de desarrollar este trabajo. A Víctor Macías por todo su esfuerzo y trabajo anterior.

Gracias a mi familia por su infinito amor, paciencia, ayuda y guía.

A todos los amigos de la carrera, ¡llegamos a la meta! Ha sido un placer conoceros y aprender junto a vosotros InfoMates: Alfonso, Ana, Antonio, Carlos, Dani R., Dani V., Gonzalo, Harry, Jesús, Manu, Sara D., Sara S., Víctor M., Victor P., Mateo, Pinar, Ferpi y Sofía. También a las matemáticas Silvia y Elena.

A Carlota, Ricardo, Ana y Ulises por sus sonrisas, cariño y apoyo.

A Melucha, Ana y Rafa por introducirme en la aventura matemática.

Índice general

Índice de Figuras

Índice de Tablas

Glosario y abreviaturas

MOOC (*Massive Open Online Courses*). Se caracterizan por no tener limitación en las matriculaciones, poder ser seguido online y tener carácter abierto y gratuito (con materiales accesibles de forma gratuita). El primero se creó en el año 2008.

edX. Plataforma en línea fundada por la Universidad de Harvard y el MIT en 2012. Es proveedor de MOOC de universidades e instituciones del mundo a estudiantes de todo el mundo, [2].

LAK (Learning Analytics and Knowledge). *International Conference on Learning Analytics and Knowledge* es un foro de investigación que proporciona un terreno común para que académicos, administradores, desarrolladores de software y empresas formen y debatan el estado del arte sobre analíticas de aprendizaje.

L@S (ACM Conference on Learning @ Scale). Learning at Scale investiga entornos de aprendizaje a gran escala, mediados por la tecnología. La conferencia fue creada por la Association for Computing Machinery (ACM), inspirada en la aparición de los MOOCs y en el cambio de pensamiento sobre la educación.

EDUCON (Global Engineering Education Conference).

eMOOCs (European MOOC Stakeholder Summit).

SIIE (Simposio Internacional de Informática Educativa). Foro internacional para la presentación y debate de los últimos avances en investigación sobre las tecnologías para el aprendizaje y su aplicación práctica en los procesos educativos.

CEDI (Congreso Español de Informática).

TEEM (Technological Ecosystems for Enhancing Multiculturality).

UAMx (Oficina para cursos MOOC de la UAM). Portal de formación online de la UAM: [3].

Curva ROC (*Reciever Operating Characteristics curve*). El espacio *ROC* es un cuadrado de dimensiones $1x1$ cuyo vértice inferior izquierdo se sitúa en el punto (0,0), el eje del plano X es la tasa de fracasos (tasa de falsos positivos, FPR) y el eje Y es la tasa de éxitos (tasa de verdaderos positivos, TPR). Un modelo tiene mayor precisión si sus puntos están cerca de la recta $y = 1$. Para más información, ver [4].

Area Under the ROC. Mide la **precisión** de clasificación de un algoritmo de aprendizaje automático. Un área de 1 representa una clasificación perfecta; un área de 0,5 representa una clasificación sin valor. Para más detalle, [5].

1 Introducción

En este primer capítulo se detalla la motivación, propuesta y objetivos del presente Trabajo de Fin de Grado. Posteriormente se detalla la estructura del documento.

1.1. Motivación del proyecto

Los **MOOC** (*Massive Open Online Courses*) son cursos de educación a distancia gratuitos ofrecidos por universidades y plataformas online que plantean un nuevo paradigma de aprendizaje. Introducen diversas formas de transmisión de conocimiento (vídeos, problemas interactivos o *gamification*), evaluación de la calidad del mismo y nuevas formas de certificación: permiten al alumno decidir su ritmo de aprendizaje en función de su disponibilidad e interés. En su **desarrollo tecnológico** plantean los siguientes retos: defensa de la privacidad de los estudiantes en la recopilación de datos, escalabilidad y coste en los sistemas para soportar gran cantidad de alumnos inscritos y la consecuente necesidad de lidiar con gran volumen y variedad de datos (*big data*, [6]), para su posterior análisis.

Las **analíticas de aprendizaje** (o *Learning Analytics*) es un tema de estudio emergente relacionado con el proceso y desarrollo de métodos para la exploración de gran cantidad de datos procedentes de **entornos educativos en línea** con la intención de mejorar las técnicas de aprendizaje, [7], contenidos de los cursos, monitorización, tutorización y seguimiento de los estudiantes [8], descubrir patrones de aprendizaje e interacciones entre alumnos y profesores en foros o redes sociales, [9]. Este análisis requiere la realización de fases de minería de datos (*Educational Data Mining*) y *Data Science*: entendimiento del contexto y recolección de los datos *online*, extracción, filtrado y limpiado de los datos, el almacenamiento, reconocimiento de patrones, visualización, interpretación y análisis de los resultados.

Uno de los problemas a los que se enfrentan los MOOC es la **elevada tasa de abandono**, [10, 11, 12]. Un estudiante puede no terminar el curso por tener intención solamente

de curiosear, por tener dificultad con las actividades o contenidos, o perder el interés o falta de tiempo para la realización del mismo, o por haber tomado la decisión de no examinarse o no pagar por la adquisición de un certificado una vez que ha completado las actividades que ha considerado [13]. Por tanto, la predicción temprana de un estudiante en riesgo de suspender o abandonar es importante [14]: permite tomar medidas de tutorización (a través de sistemas de alertas) y recomendación de tareas relevantes para la mejora de su rendimiento (con análisis de contenido y *text mining*).

En este contexto la Universidad Autónoma de Madrid (UAM, [15]) ofrece desde el año 2015 distintos cursos en la plataforma edX, [2], de la que es miembro desde 2014. Esta plataforma facilita a sus entidades académicas los datos que recopila de los estudiantes y la oficina de UAMx realiza su anonimización. Con ellos, se han desarrollado los siguientes proyectos: estudio de la estructura de ficheros de eventos, generación de indicadores, desarrollo de cuadros de mando para ver las estadísticas de los cursos y el uso de aprendizaje automático para predecir usuarios aprobados con certificado, los más recientes [1] y [16].

Además, se han presentado distintos artículos en conferencias de analíticas de aprendizaje: congreso LAK (Learning Analytics and Knowledge) [17], la conferencia EDUCON (Global Engineering Education Conference) [19, 20], la cumbre eMOOCs (European MOOC Stakeholder Summit) [10, 18, 21, 22], la conferencia SIIE dentro del CEDI 2016 [23], la conferencia L@S (ACM Conference on Learning @ Scale) [24]. El más reciente es el congreso TEEM' 17 en Cádiz en el que se presenta la herramienta *edX-MAS*, [25].

Por último, en el ámbito de predicción en MOOCs no hay modelos predictivos establecidos, [11]. Con la motivación en la investigación en este ámbito, partimos de la herramienta ***edX-MAS: Model Analyzer System for edX MOOC*** [1, 25]. Es una herramienta modular y escalable que permite al usuario visualizar, analizar y comparar datos importados de un curso MOOC, observar la importancia de variables de entrada (o indicadores) y comparar métricas de los modelos predictivos con los que cuenta.

1.2. Propuesta

La propuesta principal de este trabajo es el desarrollo de un sistema denominado *edX-MAS+* a partir de la herramienta *edX-MAS*, [1, 25], que amplíe dicha herramienta con las siguientes funcionalidades, todas ellas aportaciones del actual trabajo de fin de grado:

1. **Modelado predictivo para el abandono** de los estudiantes en los MOOCs de UAMx. Para ello se preprocesarán los datos de ediciones pasadas de los cursos de UAMx y se realizará una clasificación de alumnos que han abandonado el curso, ya que no se tienen datos de abandono etiquetados previamente. Para ello se han explorado técnicas de exploración de datos y descubrimiento de patrones versus a la clasificación manual de los mismos.

2. **Generación de modelos predictivos** para obtención de certificado (aprobado) y abandono con **nuevos algoritmos** de **aprendizaje automático**. Previamente se realizará un estudio del estado del arte de aprendizaje automático y *Data Science* con enfoque al análisis predictivo.

3. Creación de **nuevos indicadores**. Para ello, se requiere un análisis de los conjuntos de datos de eventos que los usuarios realizan en la plataforma edX y exploración de los indicadores empleados en otros estudios de analíticas de aprendizaje.

4. Mejora de aspectos de almacenamiento, opciones de generación de modelos y visualización. Entre ellas, opción de **frecuencia** de la generación de modelos **diaria o semanal**, adición de menú para la visualización de estadísticas y nuevas gráficas para la comparación de métricas de modelos para ambas variables de salida y algoritmos de los modelos predictivos.

1.3. Objetivos

El desglose de objetivos para la realización de las propuestas y redacción de este documento es el siguiente.

1. Entender la contextualización de los cursos MOOC y analíticas de aprendizaje a partir de la lectura de artículos.

2. Instalación de la herramienta *edX-MAS*. Analizar el contexto de partida a partir de la instalación y realización de pruebas en la herramienta *edX-MAS*.

3. Modelado predictivo para el abandono de los estudiantes:

   3.1.) Leer referencias sobre el abandono en MOOCs.

   3.2.) Analizar los datos de los cursos de UAMx. Seleccionar, extraer y limpiar los datos de eventos relevantes para la caracterización de abandono. Exportar los datos obtenidos para el abandono. Diseñar, codificar y probar el submódulo de extracción y preprocesado de datos.

   3.3.) Modificar el modelo de datos de la herramienta para su almacenamiento.

   3.4.) Explorar técnicas de reconocimiento de patrones de agrupación y reglas.

   3.5.) Explorar y análizar la muestra de datos para la detección de patrones y obtención de reglas para la clasificación de abandono.

   3.6.) Integrar en todas las fases de la herramienta de partida: importación de un curso, generación de modelos, almacenamiento de estadísticas y visualización.

   3.7.) Automatizar la generación de modelos predictivos para pruebas.

4. Generación de modelos predictivos para aprobado y abandono con nuevos algoritmos de aprendizaje automático.

   4.1.) Realizar un estudio del estado del arte de análisis de datos, aprendizaje automático, modelos predictivos.

   4.2.) Integrar nuevos algoritmos escogidos con el estado del arte y bibliografía.

   4.3.) Mejorar el modelo de datos relativo al almacenamiento de ejecuciones de modelos y sus estadísticas.

5. Creación de nuevos indicadores.

5.1.) Crear indicadores a partir de la matriz de actividad.

5.2.) Procesar eventos de navegación en problemas y vídeos. Diseñar, codificar y probar.

5.3.) Crear indicadores de número de problemas y vídeos distintos accedidos. Diseñar, codificar y probar.

5.4.) Codificar la distinción en la recuperación de indicadores agregados versus no agregados, gestionar indicadores de eventos y de actividad.

6. Mejora en las funcionalidades de la herramienta de partida.

6.1.) Añadir opción de modelos semanales. Cambiar el módulo de generación de modelos, almacenar la frecuencia en el modelo, crear las consultas a la base de datos en la aplicación y visualizar (botón para seleccionar diario o semanal).

6.2.) Realizar cambios en la visualización de la herramienta: menú de visualización de estadísticas de los modelos creados distinto del menú de creación de modelos. Mejorar el modelo de datos para almacenamiento de los modelos creados.

6.3.) Añadir gráficas para comparación de estadísticas de los modelos para las variables de salida (aprobado y abandono).

6.4.) Ampliar el modelo de datos para adición de datos en tabla de visualización de indicadores.

7. Desarrollar contenidos a partir de la bibliografía y estados del arte: MOOCs, analíticas de aprendizaje, aprendizaje automático, Data Science, modelos predictivos y predicción en cursos en línea. Redactar y revisar la memoria del trabajo de fin de grado.

1.4. Estructura del documento

En el *primer capítulo* se ha introducido, detallado la propuesta y objetivos del trabajo.

En el *segundo capítulo* se presenta el estudio del estado del arte general sobre análisis de datos, modelos predictivos, algoritmos de aprendizaje automático y *Data Science*; para después detallar su uso en analíticas de aprendizaje. Como análisis del contexto se describe la herramienta *edX-MAS*: su arquitectura lógica y módulos, lenguajes de desarrollo y librerías, indicadores, modelos predictivos, visualización y evaluación.

En el *tercer capítulo*, la extracción de datos y la clasificación de estudiantes que han abandonado el curso. En el *cuarto capítulo*, las ampliaciones de modelos predictivos e indicadores, módulos y diseño por cada uno. En el *quinto capítulo*, el desarrollo, la estructura de ficheros, implementación y pruebas del sistema. En el *sexto capítulo* se resumen los nuevos resultados obtenidos. En el *séptimo capítulo* se incluyen las conclusiones y puntos de continuación para trabajos futuros.

Como *anexos* se incluye la planificación (diagrama de Gantt), más detalle sobre indicadores de analíticas de aprendizaje, el árbol de decisión obtenido a partir de las reglas de clasificación de abandono y visualizaciones de la interfaz gráfica del sistema.

2

Estado del arte

En este capítulo se estudia el estado del arte de análisis de datos y modelos predictivos y su uso en el ámbito de analíticas de aprendizaje y en cursos online masivos y abiertos. En el análisis del contexto se describe la herramienta *edX-MAS*: su arquitectura, módulos, lenguajes de programación, indicadores y modelos predictivos.

2.1. Análisis de datos y modelos predictivos

El **análisis predictivo** agrupa técnicas del **análisis** y **minería de datos**, **aprendizaje automático** y **reconocimiento de patrones** para analizar datos históricos y obtener de forma precisa y fiable predicciones del valor de una variable de salida o clasificación de resultados futuros, [26]. Por tanto, el **ciclo de diseño**, [27], de un sistema para análisis predictivo consiste en:

1. **Entender** el escenario, definir un objetivo y **recolectar datos** en crudo. Limpieza y preprocesado.

2. **Exploración** de la muestra de datos, reconocimiento de patrones y **selección de las variables**. En esta fase es útil **conocimiento del dominio** (o experto) y técnicas de reducción de la dimensionalidad. **Transformación** de los datos.

3. **Almacenamiento** de los datos. Diseño del modelo de datos (diagrama entidad-relación o clave-valor) y elección entre las tecnologías de bases de datos: relacionales o NoSQL (orientadas a documento u objeto).

4. **Generación del modelo**. En primer lugar, elegir una representación. Por ejemplo, **modelos vectoriales** (dimensión el número de características), **probabilísticos** o **lógicos** (transforman relaciones entre los datos en reglas y se forman árboles de decisión). En el caso de aprendizaje automático, se definen las variables de entrada y la de salida y se realiza el entrenamiento y test de los algoritmos elegidos.

5. **Visualización** de los resultados y **evaluación** del modelo. En aprendizaje automático se compara los modelos a partir de la **matriz de confusión** y el **tiempo de entrenamiento y test** del algoritmo. La matriz de confusión es una matriz 2x2 en la que se indica los verdaderos y falsos positivos (TP, FP respectivamente) en la primera fila y, en la segunda, los falsos y verdaderos negativos (FN, TN, respectivamente). Las métricas de evaluación más comunes son exactitud (*accuracy*), exhaustividad (*recall*), precisión, tasa de falsos positivos, F1-score [28]. Para clasificadores binarios la curva ROC, [4] y AUROC (Area Under the Receiver Operating Characteristics curve), [5].

Métrica	Fórmula
Exactitud	$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
Exhaustividad	$Recall = \frac{TP}{TP+FN}$
Precision	$Precision = \frac{TP}{TP+FP}$
Tasa de Falsos Positivos	$FPR = \frac{FP}{FP+TN}$

Cuadro 2.1: Tabla de métricas de evaluación de modelos predictivos obtenidas a partir de la matriz de confusión.

Los **modelos predictivos** se construyen con una serie de variables de entrada (o **indicadores**) y pueden emplear los siguientes algoritmos, [27, 29] en función del tipo de variable de salida a predecir: categórica o cualitativa (con dos, binaria, o más clases) o paramétrica (en espacio discreto o continuo).

- Estadísticos basados en **regresión lineal, múltiple** para predicción de variables en espacios continuos (ver [30] y [31]). Para variables categóricas binarias se emplea **regresión logística**, explicado en detalle en [32].
- **Clasificación** o **aprendizaje automático supervisado**. El entrenamiento del algoritmo se realiza a partir de unos datos ya etiquetados y se conocen un número de clases fijo.
  - **Árboles de decisión**. Es un modelo jerárquico que crea recursivamente particiones en el espacio de datos tomando una variable como test y dividiendo la muestra en función de sus valores. Para saber qué variable utilizar en cada nivel del árbol se emplea una medida de pureza: impureza de Gini (frecuencia con la que un elemento sería etiquetado incorrectamente si el etiquetado fuera aleatorio) o la ganancia de información (entropía de información). Así se obtienen los nodos de decisión hasta un criterio de parada y la clasificación en los nodos hoja, nodos de respuesta. El recorrido completo por el árbol forma unas reglas de decisión. Los algoritmos más conocidos son CART (*Classification And Regression Tree*), ID3 y C4.5.
  - **Máquinas de vectores de soporte** (SVM). Trabajan en un espacio n-dimensional dado por n variables de entrada. Busca el hiperplano soporte separador a partir de las funciones de kernel. Se centran en la minimización del riesgo estructural: vectores de soporte (subconjunto de vectores de entrenamiento con propiedades específicas) y encontrar el hiperplano separador que maximice la distancia a la que se encuentra de los vectores de soporte.

  - **Vecinos próximos** (kNN). Crea un modelo vectorial y emplea predicción por similitud (con la función coseno o distancia euclídea, por ejemplo) de un nuevo vector con otros k vectores más cercanos. Se clasifica el nuevo vector en la clase que contenga a más de los vectores vecinos.
  - **Clasificador Naïve Bayes**. Emplea el teorema de Bayes de la probabilidad condicionada, [33]. Asume que las variables de entrada son independientes.

- **Aprendizaje en conjunto**, [34]. Es el proceso en el que se generan y combinan con distintas estrategias múltiples modelos (clasificadores débiles) para resolver el problema de predicción o clasificación con mayor tasa de acierto o mejorando las métricas de evaluación, dividiendo la muestra de entrenamiento aleatoriamente con reemplazamiento (*bootstrap*).

  - ***Bagging*** (*bootstrap aggregation*). Se replica el conjunto de datos de entrenamiento y se emplea cada uno como entrada de los distintos clasificadores. La clase elegida por la mayoría de los clasificadores es la decisión del conjunto. Un ejemplo es ***Random Forest***, emplea distintos árboles de clasificación y elige la clasificación con más votos.
  - ***Boosting***. En primer lugar se toma un clasificador con precisión mayor que la media y después se añaden de forma escalonada nuevos clasificadores con la estrategia de proporcionar los datos de entrenamiento más informativos para cada clasificador. El **meta-algoritmo** más común es ***AdaBoost*** (*adaptative boosting*): añade clasificadores hasta que se ha minimizado el error, usa de forma ponderada la salida de otros algoritmos de aprendizaje, creando un **árbol de decisión**. Otro es ***Gradient Boosting***, en particular *Stochastic Gradient Boosting* (o GBM) [35], *Extreme Gradient Boosting* (o *XGBoost*), [36]. También *Boosted Logistic Regression* (o *LogitBoost*) [37].

- **Redes neuronales artificiales**. Son métodos no lineales y se pueden usar para modelizar relaciones complejas entre variables de entrada y salidas, buscar patrones, clasificar y predecir, [38]. Por ejemplo, redes de una capa oculta versus al perceptrón multicapa, las redes neuronales recurrentes (RNN), redes convolucionales (CNN) y redes neuronales profundas, como *Long Short Term Memory*, LSTM, [16].

- **Modelos bayesianos**. Por ejemplo, *Bayesian Regularized Neuronal Networks* (red bayesiana), *Bayesian Generaliazed Linear Model* y *Bayesian Additive Regression Trees*, [39].

Otras técnicas que se emplean en minería de datos, [40, 41], para el reconocimiento de patrones y descubrimiento de conocimiento para toma de decisiones y clasificación son:

- Aprendizaje **automático no supervisado o *clustering***. Típicamente, K-means. Para formar los clusters toma k puntos, los centroides. Cada punto pertenecerá al cluster con centroide más cercano a él. Se busca de nuevo los centroides de cada clúster con los nuevos puntos. Repitiendo los pasos anteriores se agregan más puntos hasta que los centroides no varíen significativamente.

- **Reglas de asociación** o aprendizaje basado en reglas, [42]: a priori, *eclat* (búsqueda en profundidad), búsqueda en anchura, *FP-Growth* (algoritmo que crea un

árbol de compresión y decisión por valores frecuentes en la base de datos en orden descendente, FP-Tree), [40, 43]. Estas reglas implican causalidad entre los valores de las variables de entrada para un porcentaje de los datos, soporte, y con cierta confianza (probabilidad de la causalidad).

Para concluir este apartado, se adjunta resultados del informe *Kaggle 2017 Data Science Survey*, [44], donde se muestra un estudio del estado del arte de 2017 de *Data Science* y aprendizaje automático. Contiene estadísticas sobre el uso de los diversos algoritmos, el apartado *What data science methods are used at work?* concluye que el algoritmo de **Regresión Logística** es el más empleado.

Selections	Count	Percent
Regression/Logistic Regression	4636	63.5 %
Decision Trees	3640	49.9 %
Random Forests	3378	46.2 %
Neural Networks	2743	37.6 %
Bayesian Techniques	2236	30.6 %
Ensemble Methods	2078	28.5 %
SVMs	1948	26.7 %
Gradient Boosted Machines	1742	23.9 %
CNNs	1383	18.9 %
RNNs	895	12.3 %

Cuadro 2.2: Algoritmos más empleados según el estudio *Kaggle 2017 Data Science Survey*

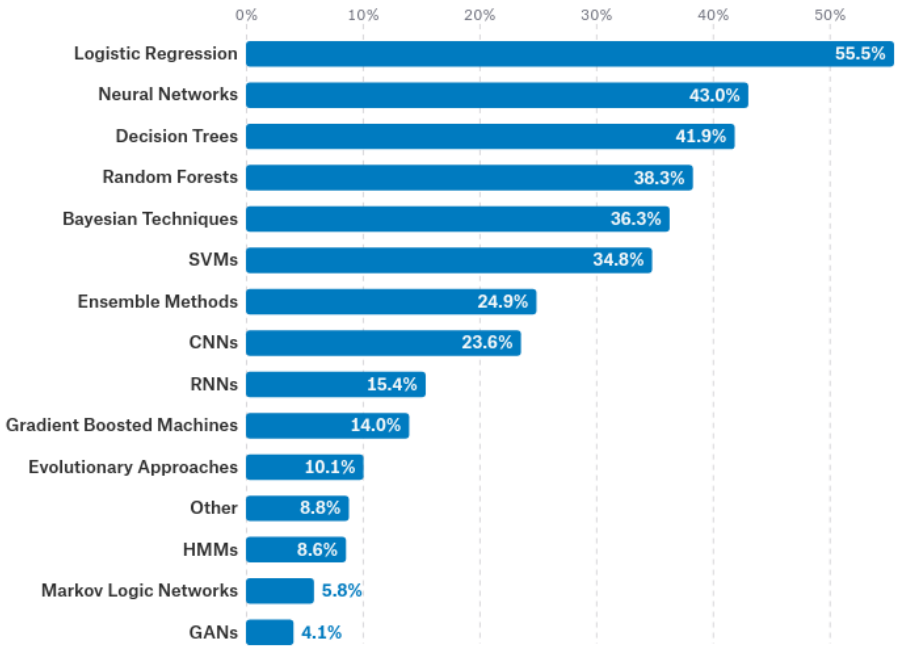


Figura 2.1: Gráfica de métodos más empleados de Data Science en el ámbito académico en 2017

Estos resultados se han empleado para la justificación de los modelos predictivos integrados en el sistema *edX-MAS+*.

2.2. Predicción en cursos educativos en línea

A continuación se presenta un estudio del estado del arte de la predicción en cursos educativos en línea, para más detalle consultar [9, 10, 11, 12, 45].

Los estudios de analíticas de aprendizaje considerados generan modelos predictivos tras la extracción y limpiado de distintos tipos de datos procedentes de los entornos educativos en línea para la **detección de estudiantes en riesgo de no obtener su certificado** [12] o **abandonar** empleando diversas técnicas: minería de datos, *text mining*, análisis de contenido [46], técnicas de procesamiento de lenguaje natural y análisis de sentimiento [11], análisis social (foros con estructuras de seguimiento [9, 46]), gamificación (con insignias o recompensas en los cursos, [47]) y aprendizaje automático [10].

Los **conjuntos de datos** más frecuentes son *clickstreams* y *video clickstream* de los usuarios con las plataformas y sus recursos, [45, 47]. También, se consideran mensajes en foros, datos demográficos y encuestas [20]. A partir de ellos, algunos estudios crean clasificaciones de estudiantes por sus interacciones [10, 47], otros miden constancia y efectividad, interés, progreso, capacidad y participación de los estudiantes [8, 47].

Los **indicadores** empleados se pueden agrupar en las siguientes categorías: indicadores de **interacción en las plataformas en línea** (por ejemplo número de eventos, tiempos totales de interacción, tiempos entre eventos), **indicadores de actividad** (por ejemplo número de días conectado, número de vídeos y problemas distintos accedidos, intentados, abandonados y completados), indicadores de material **generado por los alumnos** (obtenidos del procesado de mensajes de foros, trabajos, tareas, número de *likes*) e indicadores particulares a los estudiantes (demográficos, datos de registro). Para más detalle sobre los indicadores consultar el anexo B.

Respecto a los algoritmos de aprendizaje automático empleados en los modelos predictivos para la **predicción de aprobado**, se ha observado que los estudios [10], [11] y [12] coinciden en el uso de *Random Forest* y Regresión Logística.

El trabajo [11] añade los algoritmos de Naïve Bayes, árboles de decisión y máquinas de vectores de soporte (ordenados por mejor AUC).

El estudio [10] emplea también modelado con *boosted regression* (GBM) y vecinos próximos en la generación de modelos semanales. Se obtiene que el algoritmo GBM da mayor estabilidad en la predicción en las primeras semanas, aunque Random Forest y Regresión Logística también aportan buenos resultados.

El estudio [12] utiliza máquinas de vectores de soporte, Naïve Bayes y árboles *boosting XGBoost*.

El trabajo [16] emplea los algoritmos para la predicción del aprobado (ordenados de mayor a menor tasa de acierto): árbol de decisión CART, AdaBoost con árboles de decisión, Naïve Bayes, redes neuronales recurrentes (Long Short-Term Memory) y máquinas de vectores de soporte.

En el caso de **predicción de abandono**, los estudios no emplean una definición homogénea. Las más comunes son que el estudiante no ha completado la última semana del curso o no ha presentado actividad en la semana anterior más reciente. En los conjuntos de datos de *clickstreams*, los métodos utilizados son regresión logística, máquinas de vectores de soporte, *Random Forest*, redes neuronales, entre otros (consultar [9, 45]).

2.3. Análisis del contexto: herramienta *edX-MAS*

La herramienta ***edX-MAS: Model Analyzer System para edX MOOC*** [25] es una aplicación web que permite importar una edición de un curso MOOC, genera modelos **diarios** para la predicción de obtención de **certificado del curso** (o de aprobado) de un alumno, incluye recomendación del mejor algoritmo, visualización de gráficas para evaluar la calidad de los mismos y exportar resultados y gráficas.

2.3.1. Arquitectura lógica y módulos

Su **arquitectura lógica** es de **tres capas**: presentación, negocio y datos. La **capa de presentación** muestra la información al usuario y captura sus interacciones. La **capa de negocio** se encarga de recibir las peticiones que realiza el usuario (comunicación con la capa de presentación) y de generar una respuesta tras el proceso (comunicación con la capa de datos para almacenar o extraer datos). Esta capa incluye los módulos de **generación de indicadores** y el módulo de **generación de modelos**. Por último, la **capa de datos** se encarga de almacenar los datos de los cursos y contiene los ficheros en crudo de los cursos. El sistema *edX-MAS+* mantiene esta arquitectura lógica. La capa de datos se ha ampliado con el submódulo de preprocesado de datos (ver 4.2.1).

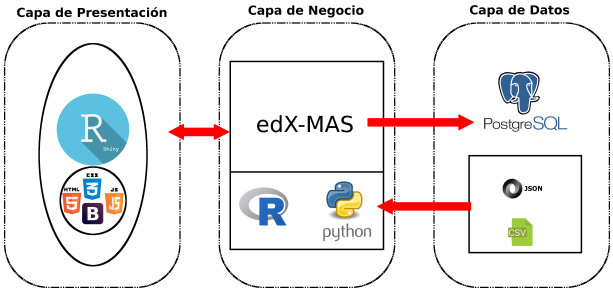


Figura 2.2: Capas de la herramienta *edX-MAS* (tomada de [1])

Sus **módulos** son: **importación** (almacenamiento y generación de indicadores), **generación de modelos**, **visualización** y **exportado de datos**.

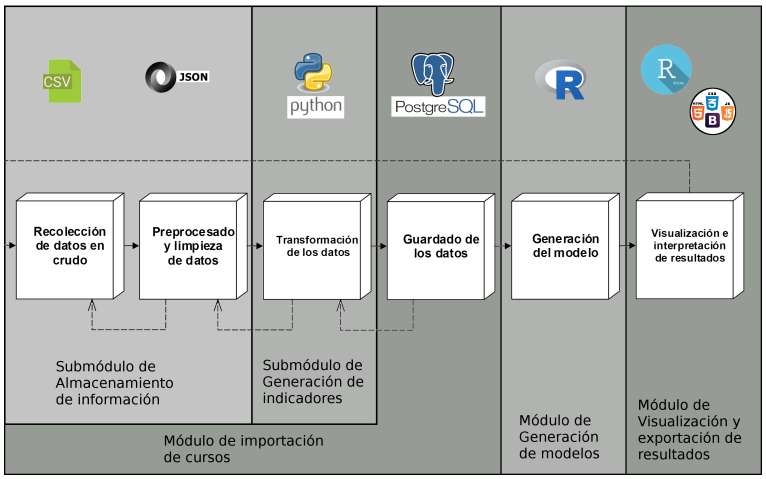


Figura 2.3: Módulos de la herramienta *edX-MAS* (tomada de [1])

2.3.2. Lenguajes de desarrollo y librerías

En la programación de la herramienta se ha empleado **R** y ***Python***. Ambos son lenguajes *Open Source*, multiplataforma e interpretados. R se emplea para análisis estadístico y gráfico, cuenta con librerías extensas: paquetes CRAN. *Python* es un lenguaje de propósito general, orientado a objetos y de alto nivel. Su sintaxis es clara, sencilla y escueta, es uno de los lenguajes con mayor información por volumen de código.

- Se ha utilizado ***R*** para el **módulo de generación y evaluación** de modelos predictivos. En concreto la librería ***caret*** (acrónimo de *Classification and Regression Training*, [48]). También, para la **capa de presentación** (módulo de visualización e interfaz de usuario) se ha realizado con la librería ***Shiny*** de R y ***Plotly*** (para gráficas interactivas). ***Shiny*** permite crear aplicaciones web basadas en programación reactiva: vincula valores de entrada con los de salida (las modificaciones de las primeras provocan variaciones en las segundas).
- Se ha empleado **Python** para la **extracción, limpiado y procesado** de los datos para la **generación de indicadores**, por su facilidad en el tratamiento de ficheros JSON.
- **Base de datos relacional** *PostgreSQL* y el gestor *pgAdmin*.

Estos lenguajes y tecnologías también se han empleado en *edX-MAS+*.

2.3.3. Indicadores

En el **módulo de generación de indicadores** se trabaja con distintos indicadores que resumen la actividad **por día** del alumno en el curso:

- Número total de **eventos** generados por el alumno (*num_events*).
- Número total de **sesiones** del alumno durante el día (*num_sessions*). Se determinan sesiones como conexiones distanciadas por una diferencia temporal mayor a un umbral prefijado en la aplicación.
- **Tiempo** total aplicado en un curso (*total_time*).
- Número total (*nav_events*) y tiempo (*nav_time*) entre eventos relacionados con la **navegación** entre diferentes contenidos del curso.
- Número y tiempo entre eventos relacionados con las **interacciones de vídeo** (*video_events*, *video_time*).
- Número y tiempo entre eventos del **foro** del curso (*forum_events*, *forum_time*).
- Número total de interacciones con los **problemas** del curso y tiempo entre dichas interacciones (*problem_events*, *problem_time*).

Con la herramienta *edX-MAS*, [25], se ha obtenido que los indicadores más relevantes son los relacionados con los eventos y tiempos de problemas y vídeos. Las variables menos relevantes son las relacionadas con los foros de los cursos.

2.3.4. Modelos predictivos

Para la predicción de aprobado del alumno se ha empleado como variables de entrada los indicadores detallados anteriormente. Se ha utilizado algoritmos de aprendizaje en conjunto de **boosting** con **AdaBoost**: ***Boosted Logistic Regression***, ***Stochastic Gradient Boosting***, ***Extreme Gradient Boosting***. Como clasificadores, ***Support Vector Machine (SVM)*** y ***k-Nearest Neighbors***. Todos están presentes en el paquete *caret* de R: *LogitBoost*, *gbm*, *xgLinear*, *svmLinear* y *kknn*, respectivamente.

En el proceso de entrenamiento de los algoritmos se dividen los datos del curso en dos subconjuntos: entrenamiento y test, proporción de 0.75 y 0.25, respectivamente. Se emplea validación cruzada repetidas veces tras centrar y normalizar los datos.

En la predicción de adquisición de certificado se ha concluido que el **algoritmo kNN tiene peor precisión** en todos los conjuntos de datos y que *Extreme Gradient Boosting* necesita más tiempo de entrenamiento y test, [25].

2.3.5. Visualización y evaluación de los modelos

En su interfaz gráfica, *edX-MAS* tiene dos menús deslizables en la columna lateral izquierda. Permiten importar un curso (tras introducir el nombre y edición, seleccionar la localización de su carpeta de datos, indicadores a calcular) y crear un modelo (seleccionando el curso, edición, método, indicadores y variable de salida - solo predicción de obtención de certificado). Para más detalle consultar el *Manual de usuario* de [1].

Tras pinchar el enlace de *Stats Models* del menú de creación de modelos, se muestran distintas pestañas en el panel principal: *AUC and Times*, *ROC*, *Importance Variables* e *Indicators*. En *AUC and Times* se incluye una gráfica de los valores AUC, gráfica del tiempo de entrenamiento y de predicción (de línea). En *ROC* e *Importance Variable* las gráficas de línea de los valores asociados al nombre de la pestaña, respectivamente. Por último, en *Indicators* se incluye una tabla que carga los indicadores y sus valores en una tabla que permite filtrar por valor de la columna o por búsqueda general, limitar el número de filas visualizado y navegar por distintas páginas de resultados.

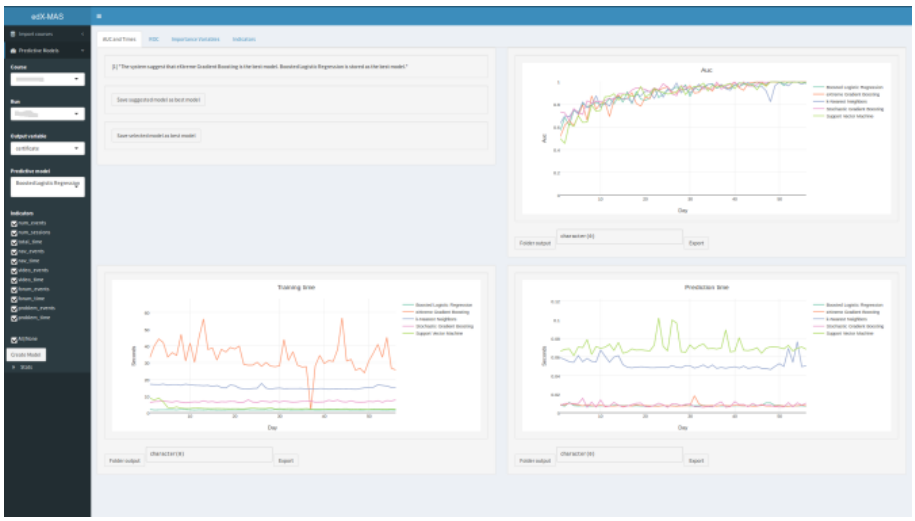


Figura 2.4: Interfaz gráfica de la herramienta *edX-MAS*

3

Abandono del curso

Este capítulo detalla la extracción, exploración de los datos y creación de una definición de abandono basada en reglas para la clasificación de los estudiantes por abandono, dado que no se dispone de datos etiquetados.

3.1. Extracción de datos relevantes

Los paquetes de los cursos de UAMx contienen distintos directorios con datos, entre ellos *certificates* y *events* (más detalle en el anexo de [1]). Después de un análisis de los mismos, la información relevante para la clasificación del abandono del estudiante es: el identificador de usuario, su nota y tipo de certificado, fecha (o día del curso) de última conexión y número total de días activos. El **tipo de certificado** puede tomar los valores de *passing* (aprobado), *notpassing* (no aprobado), *downloadable* (aprobado y descargado), *audit_passing* (aprobado), *audit_notpassing* (no aprobado) y *unverified* (aprobados que no pagaron certificado). Se han obtenido a partir de:

- Datos de la **obtención de certificados**. Contiene los **identificadores**, **nota**, identificador del curso, **tipo** y la URL del certificado si se ha descargado.

- Ficheros de **eventos de navegación por día** de los usuarios en el curso. Los datos se representan en formato JSON y recogen el identificador del estudiante, la fecha en formato UTC del evento, tipo de evento y detalles asociados. Se ha obtenido la **fecha máxima disponible** en todos los ficheros como **última fecha de conexión del alumno**, los **eventos por día** de los alumnos, la **fecha de inicio** y de **finalización del curso**. La actividad por alumno se ha modelizado como un **vector** con una componente binaria por día de curso, en función de si ha registrado o no actividad del alumno para ese día. A partir del vector, con la suma de las componentes tenemos el **número total de días activos** del estudiante en el curso.

3.2. Exploración de los datos extraídos

La extracción de datos se ha realizado sobre una muestra de los cursos de UAMx: *Equidad801x* (ediciones de 2016 1T y 3T), *Quijote501x* (ediciones 2015 1T y 3T, 2016 3T) y *Renal701x* (ediciones 2016 1T y 3T). Se han obtenido 10.919 estudiantes con eventos de los que se ha podido extraer información para su clasificación de abandono.

Una primera exploración con **R** ha permitido obtener los siguientes resultados.

- Frecuencia de los valores nominales del tipo de certificado. Los alumnos que han aprobado el curso no son útiles para la definición de abandono, por lo que nos centraremos en los 9.544 alumnos que no han aprobado, el 87,4 %.

Tipo de certificado	Equivale a	Número de alumnos
audit_passing	Aprobado	276
downloadable	Aprobado	1.091
unverified	Aprobado	8
notpassing	No aprobado	6.788
audit_notpassing	No aprobado	2.756

Cuadro 3.1: Tipo de certificados y el número de alumnos en los cursos de muestra

- Histograma del número de días activos para los alumnos que no han aprobado. Se observa una distribución de **ley de potencia con cola larga**, es decir, gran número de estudiantes no aprobados se conectaron menos de **diez días** (el 94,28 %) y pocos estudiantes se conectaron más de diez días durante el curso.

- Histograma del día de última conexión del curso para alumnos no aprobados. Se observa que el 51,55 % de los alumnos que no han aprobado tienen como día de última conexión alguno entre los veinte primeros días del mismo. Dicho porcentaje se obtiene de las cifras del histograma (primeras cuatro columnas) dividido por el número total de no aprobados.

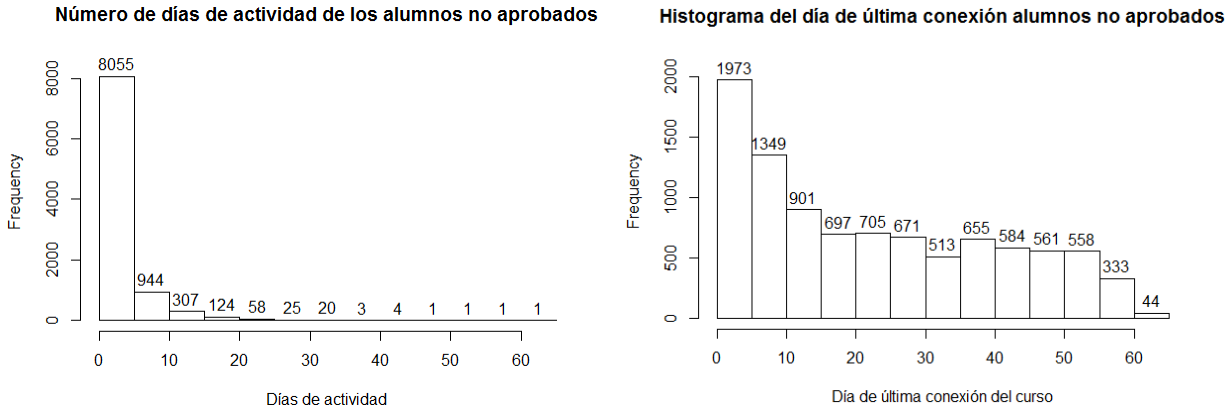


Figura 3.1: Histogramas del número de días de actividad y día de última conexión para alumnos no aprobados

Por otro lado, con el **modo de exploración** de **Weka** [40] se ha realizado una extracción de patrones comunes en los estudiantes con métodos de **aprendizaje automático no supervisado** sobre toda la muestra de estudiantes: agrupación con **K-Means** (en [47] se emplea para perfilado de usuarios y búsqueda de comportamientos comunes). Los **centroides** obtenidos permiten observar los **valores medios** de días de actividad, última conexión y nota; y el tipo de certificado que tienen en común la mayoría de estudiantes de cada clúster.

Variable	Clúster 0	Clúster 1	Clúster 2	Clúster 3	Clúster 4
grade	0,7712	0,0348	0,0619	0,0089	0,0354
status	*downloadable*	*notpassing*	*notpassing*	*notpassing*	*audit_notpassing*
last connection	51,1818	23,692	46,7945	5,7816	23,2025
num_activity_days	18,0662	3,091	5,7424	1,6257	3,2997
Instancias	1.375 (13 %)	1.945 (18 %)	1.995 (18 %)	2.848 (26 %)	2.756 (25 %)

Cuadro 3.2: Tabla de centroides de K-Means

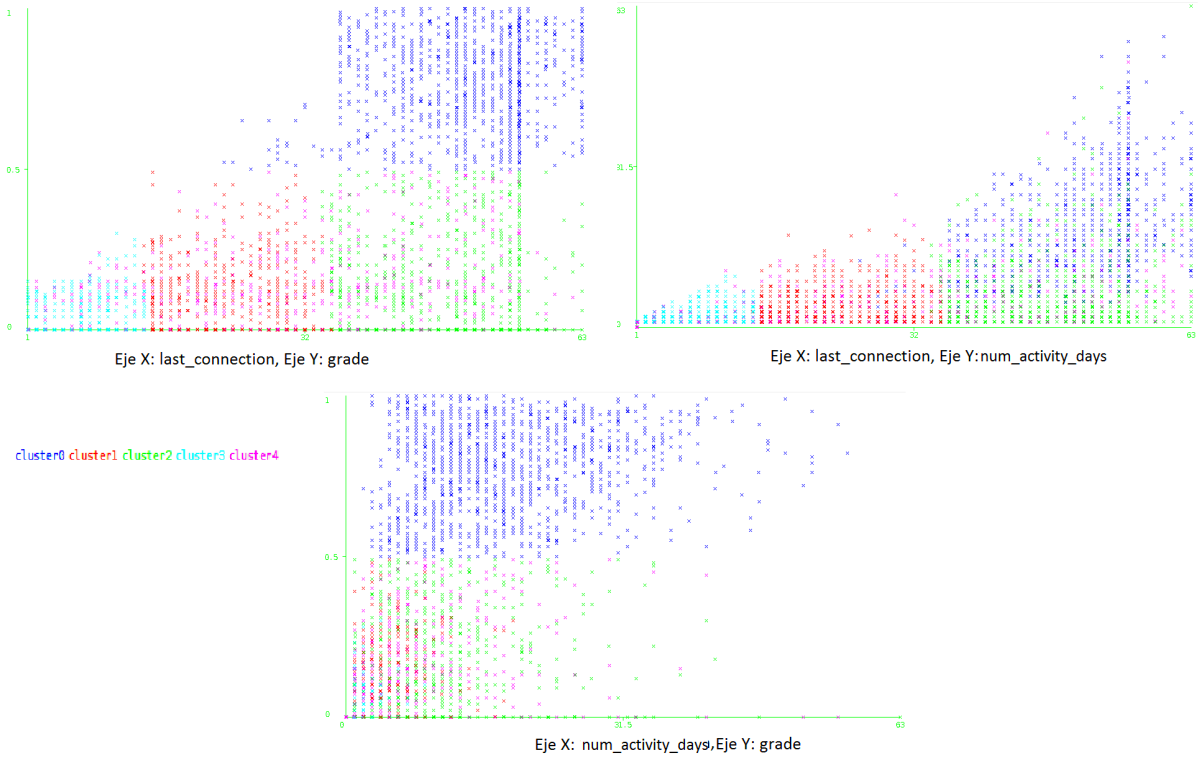


Figura 3.2: Visualización de los clústeres de K-Means

Gracias a los centroides y a la visualización de Weka, se observa que el **clúster 0** agrupa a los estudiantes **aprobados**. Se caracterizan por conectarse hasta la finalización del curso (51 último día medio, 59 es la duración media de los cursos) y estar activos casi un tercio de los días del curso. El **clúster 3** representa a alumnos **no aprobados** que dejan el curso en sus primeros días y no se conectan apenas, son usuarios que solo se matriculan en el curso por **curiosidad**. Los clústeres 1, 2 y 4 representan usuarios **no aprobados** activos hasta fechas más avanzadas del curso pero que se conectan **menos del 10 %** de los días, así que permiten establecer un **umbral mínimo de días de actividad**. Además el **clúster 2** muestra que hay alumnos que **se conectan al final** para realizar las tareas pero aun así no aprueban el curso.

3.3. Clasificación de los estudiantes

Para obtener la variable binaria de abandono se ha aplicado una **clasificación basada en reglas** considerando la exploración de los datos del apartado anterior. Estas reglas se pueden traducir en un árbol de decisión, se adjunta en el anexo C. Las reglas son las siguientes.

1. Filtro de los estudiantes cuyo **certificado** es algún tipo de **aprobado**, de tal forma que **se descarta que hayan abandonado** el curso.

2. Tras calcular el **número de días de actividad mínimo** asociado a un **umbral** de un 10 %, se considera como abandono aquellos que no lo cumplen.

3. Comparación de la **fecha de última conexión** del estudiante con la **fecha de finalización del curso**. Si el estudiante se ha conectado alguna vez durante las dos últimas semanas después de la fecha de finalización del curso para entregar ejercicios o completar tareas; o durante la **última semana del curso**, se considera que **no ha abandonado** el curso. En caso contrario, sí. Este criterio se ha tomado de los estudios valorados en [9], en los que se emplea como definición de abandono que el estudiante no haya tenido actividad en la semana más reciente o última semana del curso.

Tras aplicar la clasificación a la muestra se obtiene 2.318 estudiantes considerados como no abandono y 8.601 que sí. La gráfica 3.3 muestra en rojo los puntos de datos de estudiantes que se han clasificado como abandono, azul los que no.

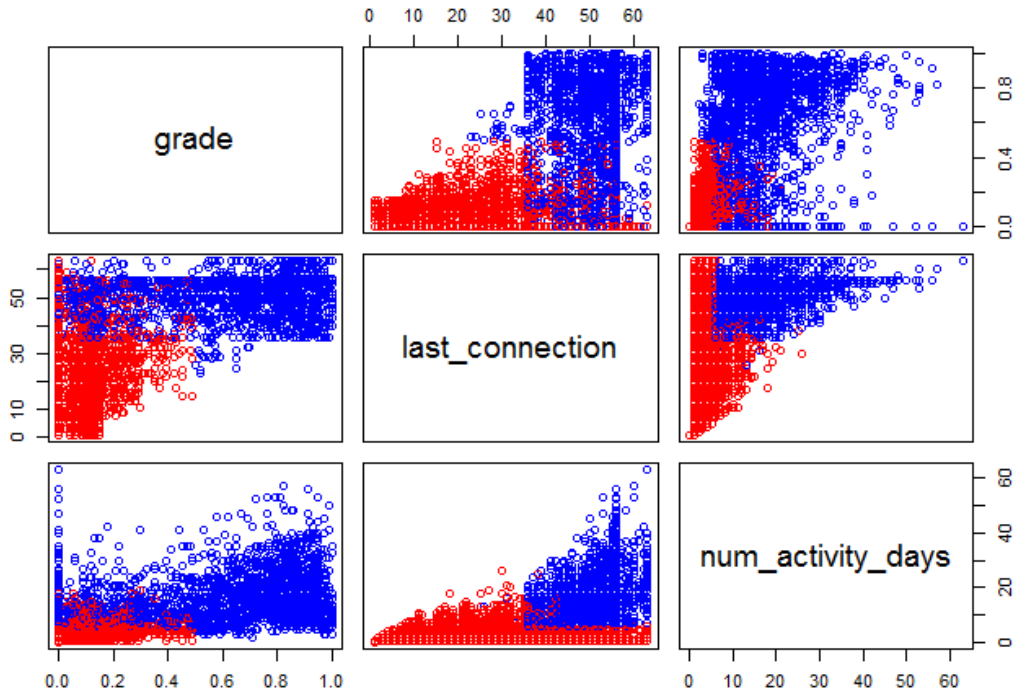


Figura 3.3: Gráfica de puntos de la muestra tras clasificación de abandono por reglas

4

Sistema *edX-MAS+*

A continuación se detalla los modelos predictivos e indicadores integrados en el sistema *edX-MAS+*. Después se describe los módulos y diseño del sistema en comparación con la herramienta de partida.

4.1. Ampliaciones en modelado predictivo

4.1.1. Modelos predictivos

Con el estudio del estado del arte presentado, se han elegido los siguientes algoritmos a integrar en el sistema *edX-MAS+*.

- **Técnicas bayesianas**. Se han valorado por estar entre las cinco técnicas más empleadas en 2017, [44], y por no haberse tratado en *edX-MAS*. Se ha escogido dos algoritmos de esta rama: **Naïve Bayes** (por haber obtenido buenos resultados en los estudios valorados) y **regresión logística bayesiana**, [39], *Bayesian Generalized Linear Model* (por utilizar además regresión logística, de uso tan extendido). Ambos se encuentran en el paquete *caret* de R con el identificador *nb* y *bayesglm*, respectivamente.
- **Random Forest**, por emplear árboles de decisión y ser un algoritmo de aprendizaje en conjunto de *bagging*, tipo no incluído en la herramienta de partida. En *caret* se identifica con *rf* y es necesario el paquete *randomForest*.
- **Árbol de decisión**. Se ha escogido CART (*Classification And Regression Tree*), por combinar regresión logística también. Se incluye en el paquete *rpart*, nombre también de su identificador.
- **Redes neuronales**. Es la segunda técnica más empleada en el ámbito académico, [44]. Se ha escogido un algoritmo *nnet* del paquete *nnet* que consiste en el ajuste de una red neuronal con una capa oculta para clasificación.

4.1.2. Indicadores

Se han añadido los siguientes indicadores de actividad de los estudiantes.

- **Número de días conectado**, *connected_days*. Se obtiene a partir del tratado de la matriz de actividad obtenida en el procesamiento de datos para el abandono. Se calcula tomando el subvector de actividad del usuario (desde el día uno del curso hasta día en el que se calcula) y se cuenta el número de componentes que son uno, es decir, que se ha conectado.
- **Número de días de inactividad consecutivos**, *consecutive_inactivity_days*. También se ha calculado a partir de la matriz de actividad y el subvector del usuario y día para el que se calcula el indicador. Se encuentra el último día que el usuario se ha conectado y se toma la diferencia de días con respecto al día de cálculo.
- **Número de vídeos distintos accedidos**, *num_diff_videos*. Se ha procesado los ficheros de eventos para obtener un listado de los usuarios y sus eventos de vídeos y en qué día del curso se ha realizado. Para calcular el indicador, se consideran los eventos hasta la fecha de cálculo y se construye un conjunto de identificadores de vídeos (distintos y sin repetición). El indicador es el cardinal del conjunto.
- **Número de problemas distintos accedidos**, *num_diff_problems*. Se calcula de forma análoga al anterior indicador, con la diferencia de que los eventos que se ha filtrado están relacionados con las interacciones del usuario con problemas del curso.

4.2. Módulos

El sistema está dividido en módulos heredados de la herramienta *edX-MAS*. A continuación se detalla su funcionalidad y ampliaciones realizadas.

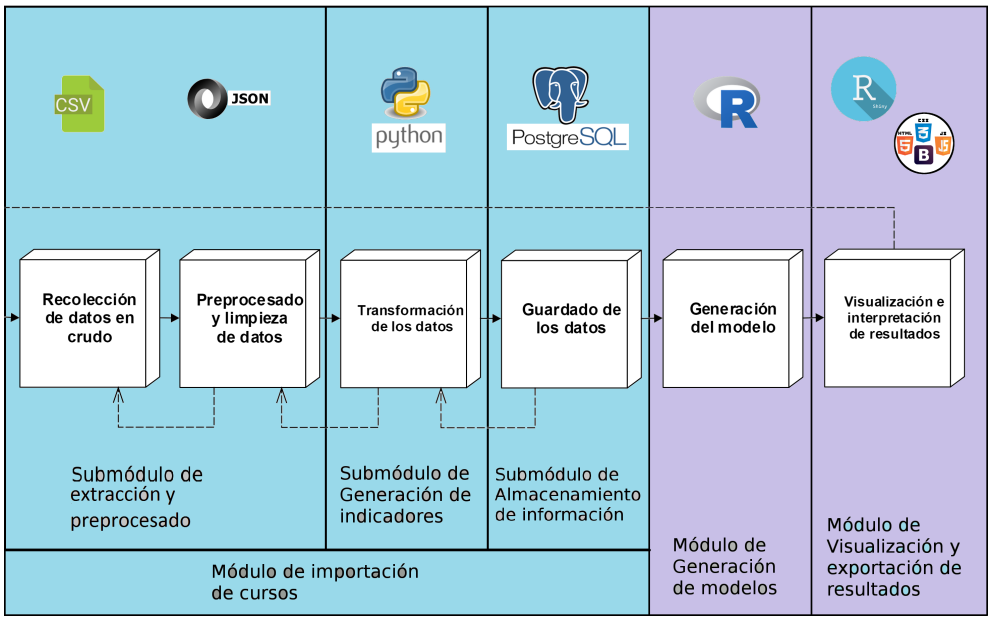


Figura 4.1: Módulos de la herramienta *edX-MAS+*

4.2.1. Módulo de importación curso

Este módulo recoge todo el proceso de importación de datos de una edición de un curso MOOC en la base de datos y la generación de ficheros tras distintos procesados de datos. Incluye las fases de recolección y extracción, filtrado y limpieza, preprocesado, transformación y guardado en la base de datos. Este módulo se divide en tres submódulos: extracción y preprocesado, almacenamiento de la información y generación de indicadores.

Submódulo de extracción y preprocesado

Este submódulo se encarga de extraer los datos de certificados y de todos los ficheros de eventos del curso (datos en crudo). Tras el limpiado y filtrado se exporta a ficheros CSV dentro de directorios específicos del directorio del curso, para su carga y uso por los submódulos de almacenamiento de la información y generación de indicadores.

Los datos exportados consisten en información obtenida para y con la clasificación de los estudiantes por abandono (fichero de abandono y de última conexión) y de la actividad total (matriz de actividad) o lista de eventos filtrada por tipo (problemas, vídeos).

Submódulo de almacenamiento de información

Su labor principal es el almacenamiento de los datos de un curso MOOC en la base de datos. Dichos datos pueden tener un filtrado sin preprocesado (provienen de ficheros CSV de certificados o eventos de un día directamente) o provenir del submódulo de extracción y preprocesado.

Submódulo de generación de indicadores

Este módulo se encarga de generar indicadores a partir de ficheros de eventos de estudiantes, de la matriz de actividad o de ficheros CSV de eventos filtrados por tipo (ficheros resultado del preprocesado de todos los eventos, resultado del submódulo de preprocesado). Este módulo corresponde con la labor de transformación.

Los indicadores se calculan para cada estudiante y día y se emplean como variables de entrada de los modelos predictivos. Se distinguen dos tipos de indicadores:

- **Indicadores no acumulativos temporalmente**. Coincide con los indicadores de interacción heredados de la herramienta *edX-MAS* (ver el anexo B). Se generan a partir de datos de navegación diarios y se obtienen con la suma de resultados parciales diarios desde el inicio del curso hasta el día de generación del modelo.
- **Indicadores acumulativos temporalmente**. Se construyen a partir de algún fichero resultado del submódulo de preprocesado, es decir, son variables creadas a partir de los datos de eventos de navegación del curso **desde su comienzo hasta el día de cálculo**. Se diferencian de los anteriores en que no se pueden obtener con la suma de resultados parciales de cada día. Por ejemplo, el número de problemas distintos accedidos no es la suma de los problemas distintos accedidos cada día porque se puede acceder más de una vez al mismo problema en distintas jornadas si no se ha terminado.

El sistema *edX-MAS+* cuenta con los siguientes indicadores.

num_events	num_sessions
nav_events	nav_time
video_events	video_time
forum_events	forum_time
problem_events	problem_time
	total_time
connected_days	consecutive_inactivity_days
num_diff_problems	num_diff_videos

Cuadro 4.1: Tabla de indicadores de *edX-MAS+*

4.2.2. Módulo de generación de modelos

Se encarga de la generación de modelos predictivos para las variables de salida de obtención de certificado (aprobado) y de abandono a partir de los datos transformados en el módulo de generación de indicadores y en la recuperación de variables de entrada de la base de datos (agregando o sin agregar). Los algoritmos disponibles para la generación de los modelos predictivos son los siguientes.

Boosted Logistic Regression	Stochastic Gradient Boosting
Extreme Gradient Boosting	Support Vector Machine
k-Nearest Neighbors	Random Forest
Naïve Bayes	Bayesian Generalized Linear Model
Classification and Regression Tree	Neuronal Network

Cuadro 4.2: Tabla de algoritmos para la generación de modelos predictivos de *edX-MAS+*

4.2.3. Módulo de visualización y exportado de resultados

Este módulo se encarga de la parte visual de la aplicación y de la comunicación con la capa lógica del sistema. En detalle:

- Aporta la interfaz gráfica para importar cursos a la base de datos. Para ello se comunica con los submódulos de extracción y preprocesado, almacenamiento de la información y de generación de indicadores (módulo de importado de un curso).
- Representa una interfaz gráfica para la creación de un modelo predictivo y almacenamiento de las características seleccionadas y estadísticas obtenidas. Intercambia datos con el submódulo de almacenamiento de información y el submódulo de generación de modelos.
- Añade una interfaz gráfica para la visualización de estadísticas de los modelos predictivos generados y recomendación del mejor modelo.
- Permite exportar gráficas, estadísticas e indicadores recuperados de la base de datos para un modelo predictivo.

4.3. Diseño por módulos

4.3.1. Módulo de importación de cursos

Submódulo de preprocesado de datos

Para extraer, filtrar y procesar los datos de cursos de UAMx se ha diseñado las siguientes clases.

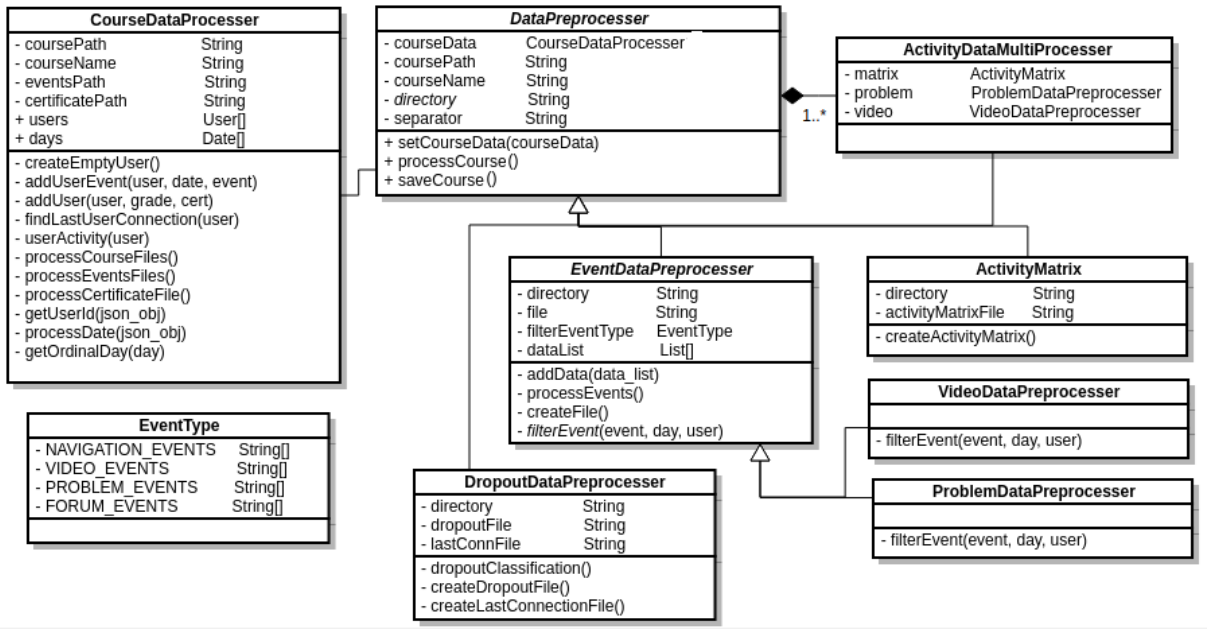


Figura 4.2: Diagrama de clases del submódulo de preprocesado de datos del sistema *edX-MAS+*

CourseDataProcesser extrae la información de los usuarios del curso (eventos, nota, tipo de certificado, fecha de última conexión, vector de actividad) y una lista de fechas de los días del curso. Sus métodos permiten tomar y limpiar el identificador del usuario, fecha de última conexión de los objetos JSON de los ficheros de eventos.

La clase abstracta **DataPreprocesser** contiene una clase *CourseDataProcesser* (instancia única y compartida, patrón *Singleton*), la ruta del curso, ruta de exportado y nombre. Contiene métodos para exportar los datos obtenidos a subdirectorios del curso. Heredando de ella y sobrescribiendo sus métodos se pueden implementar nuevos procesados sobre otros ficheros de datos de los cursos de UAMx. Sus clases hijas son *DropoutDataPreprocesser*, *EventDataPreprocesser* y *ActivityMatrix*. **DropoutDataPreprocesser** implementa las reglas para la clasificación por abandono y crea los ficheros CSV de última conexión (identificador del usuario, nota, tipo de certificado, fecha de última conexión) y de abandono (identificador de usuario, nota, tipo de certificado, último día de conexión, número de días conectado y la variable binaria de abandono). **EventDataPreprocesser** engloba filtros sobre **todos** los eventos del curso. Para añadir más filtrados se puede emplear los tipos enumerados de *EventType* y crear más clases hijas, siguiendo el diseño de **ProblemDataPreprocesser** y **VideoDataPreprocesser**. **ActivityMatrix** exporta a un fichero CSV la matriz que tiene como filas vectores de actividad de los estudiantes, es decir, una componente binaria (uno si se ha conectado, cero sino) por día del curso.

Submódulo de almacenamiento de información

En este subapartado se describe el modelo de la base de datos del sistema. Se ha obtenido a partir del modelo de datos de la herramienta *edX-MAS*, con algunas modificaciones, como se puede observar en las siguientes tablas. Se tiene más información sobre las tablas coincidentes en el apartado 4.2.1 de [1].

Nombre	**Datos que contiene**
course_runs	Id. edición y curso (course_run_id, **único**), nombre y edición del curso
course_users	Id. estudiante (único), course_run_id y nombre usuario
indicators	Id. estudiante, nombre indicador, día y valor
course_indicator_names	Id. indicador, course_run_id y nombre
course_duration	course_run_id y duración
course_prediction_meths_names	course_run_id y método predictivo
course_ouputs	course_run_id y salida
course_models	course_run_id, método predictivo, indicadores, salida, día y caracteres de R variable del modelo
course_bestmodel	course_run_id y mejor método predictivo, salida e indicadores
course_statsmodels	course_run_id, método predictivo, indicadores, salida y variable estadísticas de R en texto plano
certificates	Id. certificado, id. estudiante, nota y variable binaria aprobado

Cuadro 4.3: Tablas del modelo de datos de *edX-MAS*

Nombre	**Datos que contiene**
course_models_execution	Id. ejecución (único), course_run_id, método predictivo, indicadores, salida y **frecuencia**
course_models	**Id. ejecución**, día y caracteres de R variable modelo
course_statsmodels	**Id. ejecución** y variable estadísticas de R en texto plano
dropout	Id. dropout, id. estudiante, tipo, fecha última conexión, días de actividad y variable binaria abandono

Cuadro 4.4: Tablas añadidas y modificadas en *edX-MAS+*

La tabla *course_models_execution* permite guardar los datos asociados a la creación de un modelo: algoritmo, lista de indicadores, variable de salida (certificado o abandono), frecuencia (diaria o semanal) y relaciona el curso y edición con el identificador *course_run_id*.

La tabla *dropout* almacena los datos extraídos y la clasificación realizada.

La carga de datos de todas las tablas menos *course_models_execution*, *course_models* y *course_statsmodels* se realiza en las fases de extracción, preprocesado, limpieza, transformación y guardado de datos. Las tres tablas mencionadas se poblarán al crear un modelo desde la interfaz gráfica o con la automatización de creación de modelos.

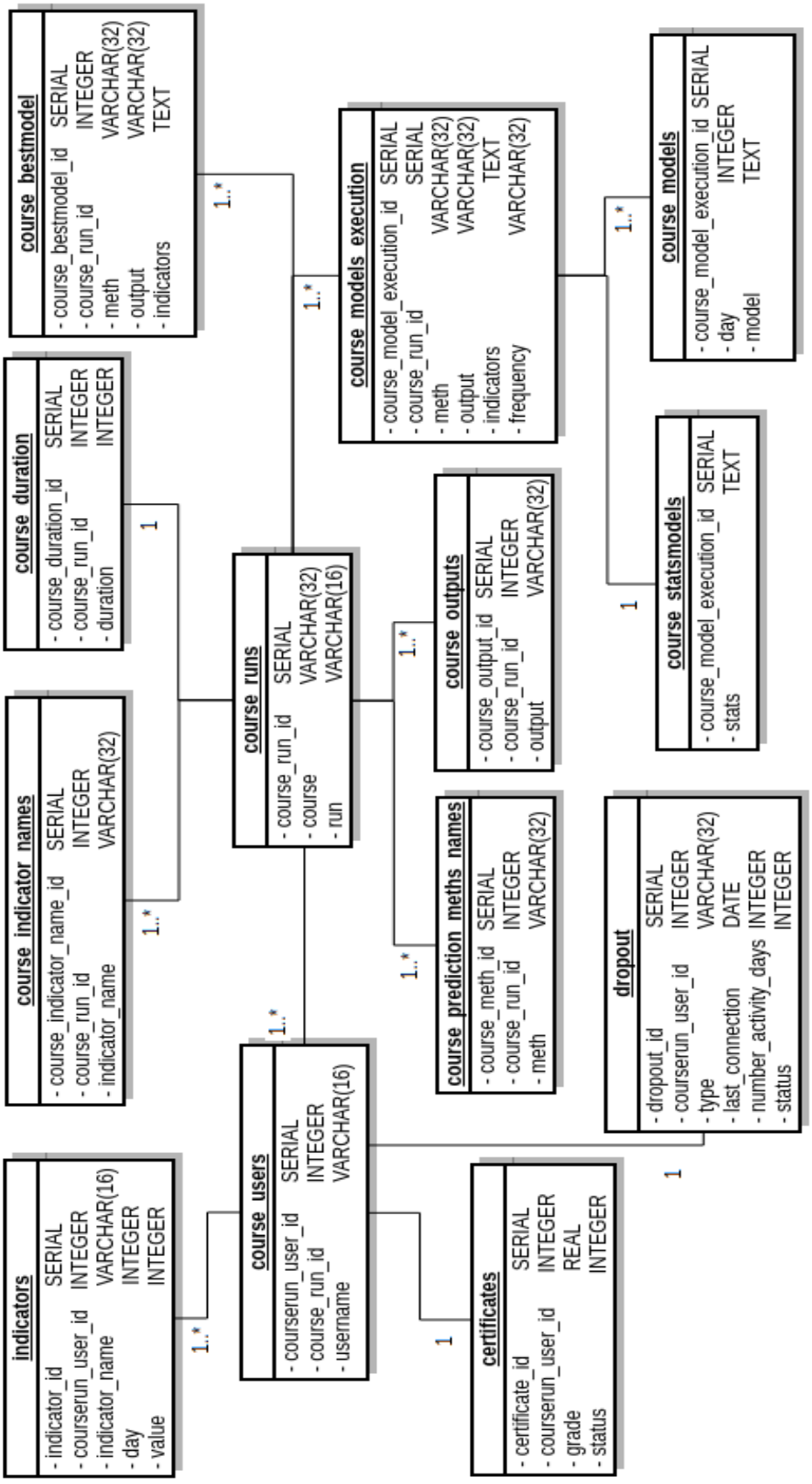


Figura 4.3: Modelo de datos del sistema *edX-MAS+*

Submódulo de generación de indicadores

Se muestra el diseño que se ha realizado para generar indicadores de eventos y de actividad. Se ha creado a partir del diseño de indicadores de eventos de *edX-MAS* y de forma que se mantenga la escalabilidad. Se incluye un diagrama de clases en la figura 4.4.

El indicador base es **EdxIndicator**. Incluye los atributos y métodos básicos para calcular todos los indicadores de cada estudiante en un día especificado. Para ello, permite el procesado (método *processFile*) de un fichero asociado (atributo *file*) a un día o que recoja información desde el principio del curso hasta el día de cómputo (o finalización del curso). Creando clases hijas de ella y sobrescribiendo el método de procesado se permite crear indicadores más complejos que añadan fórmulas, lectura (de ficheros obtenidos tras preprocesados) y/o procesados rápidos de otras fuentes de datos (no solo eventos).

EdxIndicator tiene asociada la clase **EdxPostgresqlStore** que conecta con la base de datos y permite guardar toda la información del indicador. También está contenida en la clase **EdxMultipleIndicators** que permite calcular los indicadores a partir de listas diferenciadas de nombres de indicadores de eventos o de actividad gracias al método *processAllindicators*. *EdxIndicator* tiene dos clases hijas: *EdxEventIndicator* y *EdxActivityIndicator*.

EdxEventIndicator, denominada *EdxIndicator* en [1], incluye métodos que permiten tomar una lista de eventos para un día prefijado (*getDayEvents*) y usuario (*getDayEventsPartition*) con el procesado rápido del fichero de eventos asociado a ese día. Sus clases hijas son las siguientes (para detalle sobre las clases nietas consultar [1]).

- **EdxIndicatorFilterEvents**. Permite filtrar eventos asociados a un usuario y tipo de evento y generar como indicador la longitud de la lista de filtrado. El filtrado de eventos se gestiona con los parámetros globales creados para el módulo de preprocesado, gracias al anexo de [1] (disponibles en el fichero *event_ settings.py*).
- **EdxIndicatorTimes** cuenta el tiempo transcurrido entre eventos de un mismo tipo definiendo sublistas de eventos transcurridos durante un tiempo menor a cinco minutos de conexión del usuario. Como también requieren filtros de eventos diarios, heredan de las correspondientes clases hijas de *EdxIndicatorFilterEvents*.

La clase **EdxActivityIndicator** cuenta con un atributo de datos genérico que las hijas (*EdxMatrixActivityIndicator* y *EdxNumDiffActivityIndicator*) sobrescriben con la estructura creada tras la carga de información de los ficheros de actividad total del curso (obtenidos en el módulo de preprocesado). También tiene el método abstracto para obtener la actividad del usuario desde el primer día hasta el día de cálculo.

EdxMatrixActivityIndicator, hija de *EdxActivityIndicator*, procesa la matriz de actividad y devuelve el subvector de actividad. Los indicadores cuentan el número de días con actividad de ese subvector (*EI_ connected_ days*) y el número de días sin conexión, ceros del subvector desde el último uno, (*EI_ consecutive_ inactivity_ days*).

La clase **EdxNumDiffActivityIndicator**, hija de *EdxActivityIndicator*, procesa el listado de eventos filtrados de **todos los ficheros de eventos** (eventos de problemas para *EI_ num_ diff_ problems* y de vídeo para *EI_ num_ diff_ videos*). En *userActivityUntilDay* devuelve el **conjunto** de identificadores de los recursos a los que el usuario **ha accedido** entre el primer y el día de cómputo. El indicador es el cardinal de ese conjunto.

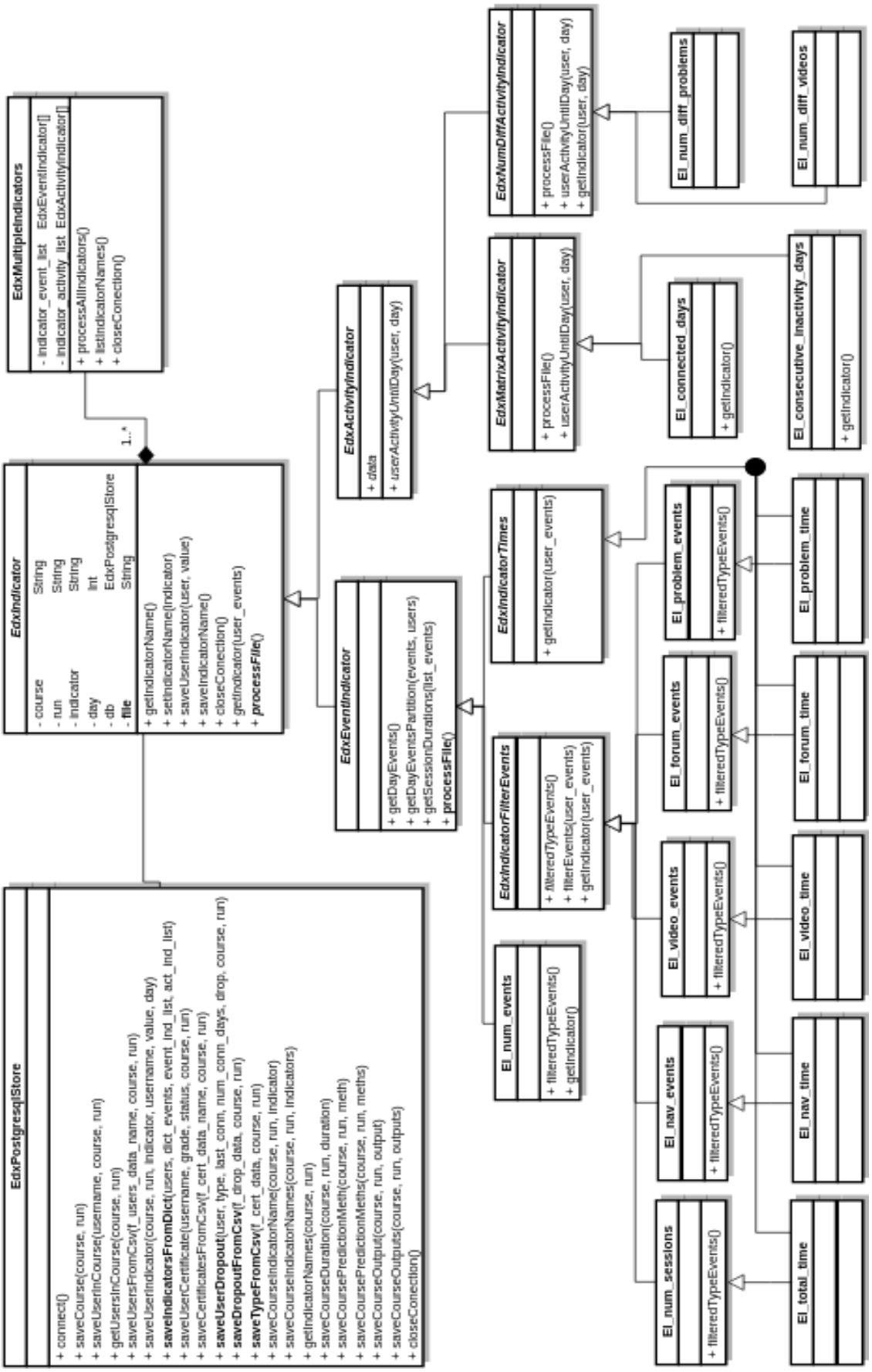


Figura 4.4: Diagrama de clases de los indicadores del sistema *edX-MAS+*

4.3.2. Módulo de generación de modelos

En este apartado se muestra el diseño de las clases utilizadas en la generación de modelos y la recuperación de datos del sistema previa. Se parte de las clases diseñadas en la herramienta *edX-MAS*: *Predictor*, *PredictorFunctions*, *EdxRPostgresql* y una clase para cada algoritmo predictivo (*PredictorLogitBoost*, *PredictorGbm*, *PredictorXGBoost*, *PredictorKnn* y *PredictorSvm*). Se han creado clases para los nuevos métodos predictivos: *PredictorRandomForest*, *PredictorNaiveBayes*, *PredictorBayesianGLM*, *PredictorCART* y *PredictorNNet*.

La clase base es **PredictorFunctions**. Tiene adjuntas las clases **Predictor** (para los datos que identifican el modelo, se ha añadido el atributo para la frecuencia diaria o semanal) y **EdxRPostgresql** (para el acceso a la base de datos). Las funciones de *PredictorFunctions* permiten:

1. Recuperar de la base de datos los indicadores asociados a un día de curso con el método *indicators_df_merged_acc*. Devuelve una matriz que tiene como columnas el identificador del estudiante y valor de cada indicador. Se ha alterado para distinguir los indicadores acumulativos y no acumulativos, requieren consultas diferentes a la base de datos, reflejadas en la clase *EdxRPostgresql*: en el método *get_merged_indicators* y adición del método *get_acumulative_indicators*.

2. Tomar de la base de datos la variable de salida, con el método *output_df_merged*. Se ha modificado para tomar una de las dos posibles salidas: certificado o abandono, y recuperar los datos de las variables binarias asociadas de la base de datos. Se invoca a un nuevo método de *EdxRPostgresql*: *get_course_dropout*. Se devuelve el identificador de usuario y su valor de la variable de salida.

3. Se combinan ambas matrices en el método *all_indicators_output_df_merged* para obtener la matriz ampliada, es decir, la matriz de variables de entrada (indicadores) y la variable de salida (certificado o abandono).

4. Se crea una partición de las filas de la matriz ampliada para entrenamiento y predicción (test) con el método *divide_all_data*.

5. Se genera el modelo predictivo, se mide el tiempo de entrenamiento y predicción en el método *train_model*.

6. Con el modelo se generan las estadísticas asociadas gracias al método *predict* que devuelve el tiempo empleado en predicción y más información en la estructura de retorno. Esta estructura es la entrada para *roc_performance*, *auc_performance* e *importance_variable* que devuelven matrices de puntos para formar la curva ROC, la medida AUC y la importancia de las variables, respectivamente.

El método *create_predictor_model* integra las fases descritas (invocando al método interno *predictor_model_iterator*) y tiene la lógica de decisión para generación de modelo diario o semanal. En el caso de semanal, se realiza una copia de los resultados obtenidos el primer día de la semana a los demás con el método *predictor_model_iterator_copy*.

La clase **EdxRPostgresql** permite almacenar los resultados obtenidos con los métodos *store_model*, *store_model_stats*, *store_model_execution*.

Las modificaciones detalladas se marcan en negrita en el diagrama de clases 4.5.

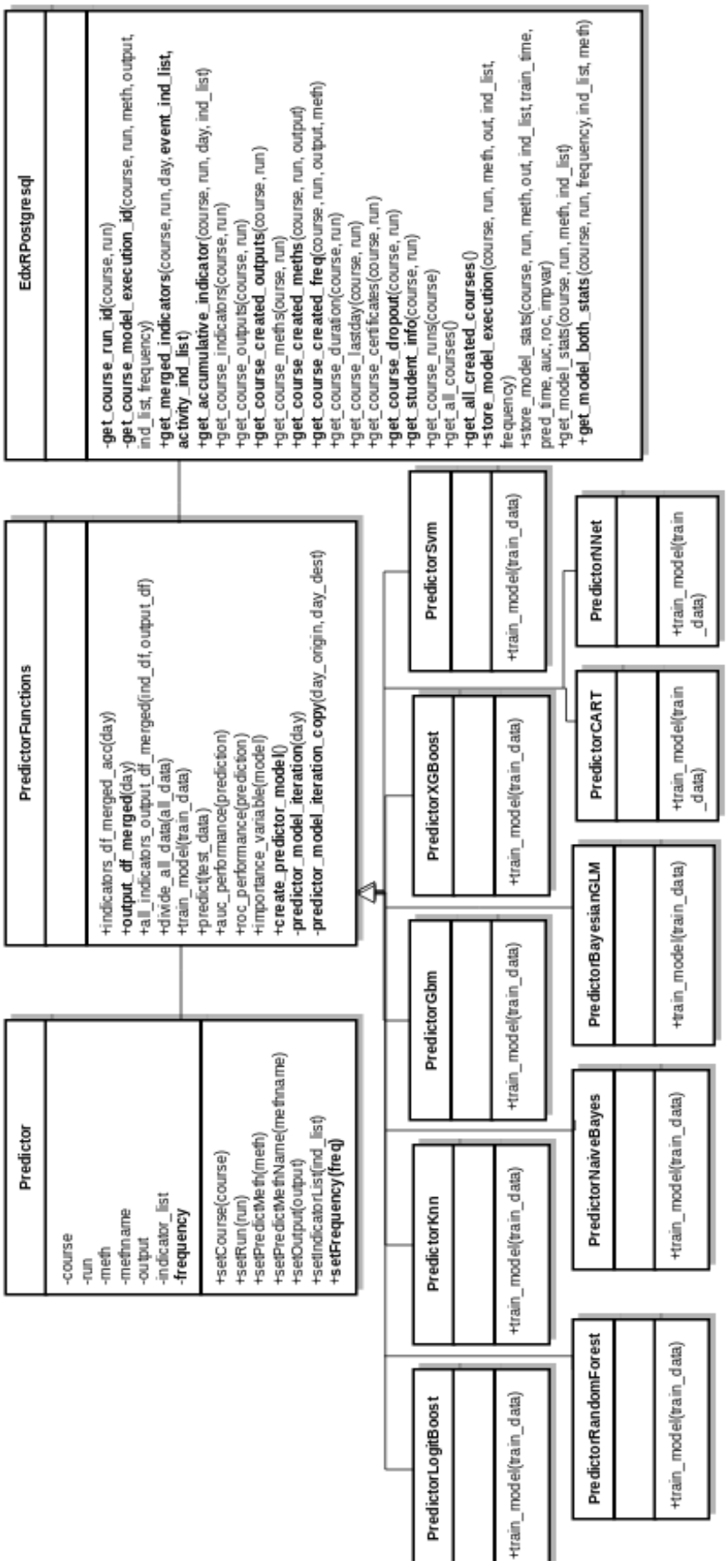


Figura 4.5: Diagrama de clases para la generación de modelos en el sistema *edX-MAS+*

4.3.3. Módulo de visualización y exportado de resultados

Este módulo se encarga de gestionar la interacción con el usuario y de definir el flujo de información entre la interfaz gráfica y el resto de componentes. El sistema *edX-MAS+* emplea el diseño realizado con el patrón *modelo vista controlador* (MVC) de la herramienta de partida.

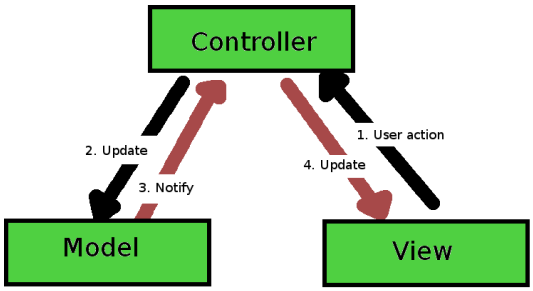


Figura 4.6: Patrón modelo vista controlador (tomada de [1])

En la **vista**, la **interfaz gráfica del sistema** incluye tres menús deslizables en la columna lateral izquierda para importar un curso, crear modelos y visualizar estadísticas de los modelos generados. Las estadísticas de los modelos se muestran en un panel central con las siguientes pestañas. Se adjuntan imágenes en el anexo D.

- En *AUC and Times* se incluye la recomendación del mejor modelo, las gráficas AUC, tiempo de entrenamiento y predicción para los modelos del curso y edición con la frecuencia, indicadores y variable de salida seleccionados.
- En *ROC* se muestran las gráficas de las curvas ROC para la predicción de obtención de certificado y de abandono del curso, edición del modelo con algoritmo, frecuencia e indicadores seleccionados.
- En *Importance Variables*, las gráficas de la importancia de las variables en la predicción de obtención de certificado y de abandono del curso, edición del modelo con algoritmo, frecuencia e indicadores seleccionados.
- En *Indicators* se muestra la tabla de indicadores e información del estudiante.
- En *AUC Comparatives*, las gráficas AUC para certificado, abandono y comparativa entre ambas para los modelos del curso y edición con indicadores y frecuencia seleccionadas.

El **servidor de R *Shiny*** cumple la funcionalidad de **controlador**. También la clase **EdxRPostgresql** porque permite **recuperar nuevos datos** de la base de datos de los modelos generados para las opciones de los desplegables del menú de estadísticas (*get_all_created_courses*, *get_course_created_outputs*, *get_course_created_freq*), estadísticas para gráficas comparativas de modelos con distintas variables de salida (*get_model_both_stats*), datos de los estudiantes para la tabla de indicadores (*get_student_info*). Además incluye los métodos para recuperar datos para los desplegables de los menús de importación de modelos y de creación de modelos, para la recomendación del mejor modelo predictivo para un curso seleccionado y estadísticas (las cadenas de texto de R) para su representación en las gráficas detalladas (*get_model_stats*).

5

Desarrollo, implementación y pruebas

El desarrollo del sistema *edX-MAS+* se ha realizado teniendo en cuenta el análisis del contexto de la herramienta de partida y siguiendo sus principios de modularidad y escalabilidad. Para la realización de cada modificación e integración de los objetivos se ha estudiado la reutilización, adaptación y mejora del modelo de datos, estructuras y clases del código disponible.

En este capítulo se describe factores escalables de la herramienta y su implementación para trabajos futuros. Después las pruebas realizadas.

5.1. Estructura de ficheros

La estructura de ficheros del sistema se compone de dos directorios diferenciados por funcionalidad y lenguaje de programación empleado: *postgresqlstore* (Python) y *R*. En el primero se encuentran los ficheros del módulo de importación de cursos. En el segundo la lógica de los módulos de generación de modelos, visualización y exportado de datos.

/postgresqlstore	*/R*
createdatabase.sql	/src/server/courseimporter.R
course_data.py	/src/server/interactiveUI.R
data_preprocesser.py	/src/server/modelcreator.R
edxindicator.py	/src/server/statsgraphics.R
edxindicatoractivity.py	/src/ui/importcourses.R
edxindicatornumevents.py	/src/ui/predictivemodel.R
edxindicatorstore.py	edxrpostgresqldb.R
edxindicatortime.py	global.R
events_settings.py	model_script.R
script_all_course.py	predictor.R
	server.R
	ui.R

Cuadro 5.1: Estructura de ficheros de *edX-MAS+*

A continuación se describe la funcionalidad de cada fichero de *postgresqlstore*.

- *createdatabase.sql* crea la estructura de la base de datos del modelo descrito en el capítulo del sistema *edX-MAS+*, diseño del submódulo de almacenamiento de la información.
- *course_ data.py* contiene el módulo que procesa todos los eventos y certificados del curso y carga la información en las estructuras detalladas en el diseño del submódulo de extracción y preprocesado.
- *data_ preprocesser.py* contiene el código para el preprocesado de datos para la clasificación de abandono y obtención de problemas y vídeos realizados por los estudiantes (filtrado de eventos). Corresponde al código del diseño del submódulo de exportado y preprocesado.
- *edxindicator.py* contiene las clases abstractas detalladas en el diseño del submódulo de generación de indicadores.
- *edxindicatoractivity.py* contiene el código de las clases hijas de los indicadores de actividad.
- *edxindicatornumevents.py* contiene el código de las clases hijas de los indicadores de interacción de conteo de eventos.
- *edxindicatorstore.py* contiene el código para la generación de múltiples indicadores simultáneamente, distinguiendo entre las dos clases hijas de la clase abstracta base: indicadores de eventos e indicadores de actividad.
- *edxindicatortime.py* contiene el código de las clases hijas de los indicadores de interacción temporales.
- *edxpostgresqlstore.py* contiene una clase que se encarga de la conexión y de la gestión de la base de datos al importar cursos (consultas INSERT).
- *event_ settings.py* contiene la clasificación de eventos por su recurso asociado (navegación general, vídeos, foros o problemas).
- *script_ all_ course.py* equivale a un ***main*** que a partir de una ruta donde se encuentran los datos del curso, nombre, edición y lista de indicadores preprocesa los datos, los carga a la base de datos del sistema y genera los indicadores solicitados.

Respecto a los ficheros del directorio *R*, se distingue entre los ficheros del módulo de generación de modelos y los de la funcionalidad de visualización y exportación de resultados.

- Generación de modelos. Contiene dos ficheros: *predictor.R* y *edxpostgresqldb.R*. El primero cuenta con las clases detalladas en el diseño del submódulo de generación de modelos. El segundo contiene la clase que se encarga de la conexión y la gestión de la base de datos al generar modelos.
- Módulo de visualización y exportado de datos.

- *edxpostgresqldb.R.* Este fichero pertenece al controlador del patrón (MVC) entre la capa de visualización y de datos, es necesaria para recuperar datos al recomendar el mejor modelo, para las gráficas y tabla de indicadores e información del estudiante. Es decir, contiene consultas del tipo SELECT.
- Ficheros base para la aplicación *Shiny* de R: *ui.R* y *server.R*. El primero define la interfaz gráfica (pestañas del panel principal y la columna lateral con menús). El segundo sirve como controlador y maneja la respuesta de la aplicación tras las interacciones del usuario. Para que estos ficheros no sean extensos, en la herramienta *edX-MAS* se subdividió su codificación en las carpetas *R/src/server* y *R/src/ui*. Se ha ampliado el fichero *R/src/ui/predictivemodel.R* para la adición de nuevas gráficas comparativas y el menú de visualización de estadísticas.
- *global.R* contiene las funciones que conectan la capa de servidor de *Shiny* con el resto de ficheros desarrollados.

5.2. Implementación y escalabilidad

Como se ha comentado en el apartado de diseño, es posible ampliar las funcionalidades del sistema en los siguientes puntos.

- **Creación de nuevos extractores y preprocesadores**. Para un procesado de ficheros de datos de los cursos de UAMx (distintos a eventos y certificados) se debe integrar la carga, extracción y limpiado con nuevos atributos y métodos en la clase *CourseDataProcesser* (fichero *course_ data.py*). Después, se debe codificar una clase hija de la clase **DataPreprocesser** y sobreescribir sus métodos de *processCourse* y *saveCourse* (fichero *data_ preprocesser.py*) si se quiere exportar los datos extraídos. Si se quiere preprocesar datos de eventos, se debe heredar de la clase *EventDataPreprocesser* y modificar los tipos de eventos a filtrar en un nuevo tipo enumerado en el fichero *events_ settings.py*.
- **Obtención y representación de nuevas métricas para evaluación de modelos predictivos**. Se podrían añadir nuevas métricas: exactitud, exhaustividad, precisión, tasa de falsos positivos, *F1-score*, coeficiente de *kappa de Cohen*. Para ello habría que emplear la matriz de confusión disponible en R *caret* en el método *predict* de *predictor.R* o las funciones *twoClassSummary*, *multiClassSummary* (de *caret*) en los métodos *train* de *predictor.R*, dependiendo de la métrica. Se guardarían los valores para cada día en la variable *self$predictor* en los métodos *predictor_ model_ iteration* y *predictor_ model_ iteration_ copy* para representarlos en gráficas de líneas de forma similar al tiempo de entrenamiento, por ejemplo. Se almacenarían en la base de datos en el método *store_ model_ stats* de *edxpostgresqldb.R*, en la cadena de texto de R de la tabla *course_ statsmodels*.
- **Generación de nuevos indicadores**. Se pueden crear indicadores relacionados con los no acumulativos (heredando de *EdxIndicatorFilterEvents*, *EdxIndicatorTimes*) o acumulativos (heredando de *EdxActivityIndicator*, *EdxNumDiffActivityIndicator*) disponibles. Los segundos requieren ficheros de preprocesados y filtrados

de eventos de todo el curso. Si se crea un indicador con definición distinta a los del sistema (con cálculos como medias, dispersiones) se debe heredar de *EdxIndicator*. En todos los casos, se debe modificar el diccionario de indicadores de *script_all_course.py* y la función de *get_default_indicators* del fichero *global.R*, añadiendo su nombre.

- **Generación de nuevos modelos predictivos**. Se alteraría la función *get_course_meths*, la variable *methdict* de *global.R* con el nombre del modelo y su nomenclatura en la librería de *caret*. Si se quiere crear modelos personalizados se debe añadir una clase en *predictor.R* que herede de *predictorFunctions* y sobrescribir sus métodos.

5.3. Pruebas

Para la realización de las pruebas se han empleado datos reales de los MOOC de UAMx: *Equidad801x* (ediciones de 2016 1T y 3T), *Quijote501x* (ediciones 2015 1T y 3T, 2016 3T) y *Renal701x* (ediciones 2016 1T y 3T).

5.3.1. Pruebas unitarias

Permiten controlar el funcionamiento correcto de cada módulo del sistema por separado. Se ha realizado con ejecuciones de rutinas principales sencillas en cada módulo.

- **Submódulo de extracción y preprocesado**. Se subdividen estas pruebas unitarias en dos, por tener código para la extracción (*course_data.py*) y código para el preprocesado y exportado tras la extracción (*data_preprocesser.py*).
  - Se ha creado una rutina principal para visualización de número de usuarios y eventos extraídos. Estas pruebas se pueden ejecutar desde el *main* de *course_data.py*.
  - Se ha creado un *main* para la extracción y exportado de los datos de abandono y de actividad a los ficheros de abandono, matriz de actividad, problemas y vídeo. Estas pruebas se pueden ejecutar desde el *main* de *data_preprocesser.py*.
- **Submódulo de almacenamiento de información**. Se ha probado la conexión con la base de datos y el almacenamiento de nuevos datos de abandono con la función *main* del fichero *edxpostgresqlstore.py*.
- **Submódulo de generación de indicadores**. Se ha tomado de la rutina de *scrip_all_course.py* la parte relativa a la generación de indicadores para su ejecución sin necesidad de importar un curso al completo. Se ha probado la creación simultánea de múltiples indicadores con la clase *EdxMultipleIndicators*. Estas pruebas se pueden ejecutar desde el *main* de *edxindicatorstore.py*.
- **Módulo de generación de modelos**. Se ha codificado en el fichero *model_script.R* una rutina similar a la realizada por el servidor de la aplicación R *Shiny* para la creación de modelos en la consola de *RStudio* y almacenamiento en la base de datos

del sistema (su ejecución y estadísticas), sin necesidad de lanzar la aplicación web y seguir su interfaz gráfica. Sobre ella, distintas funciones permiten la ejecución de más de un modelo para algunos parámetros fijos. Por ejemplo, dado un nombre de curso, edición y método se puede ejecutar todos los modelos para las dos variables de salida y las dos frecuencias. También, ejecutar todos los algoritmos para un curso y edición con una variable de salida y frecuencia fijadas. Estas pruebas unitarias son las denominadas como automatización en la generación de modelos predictivos en los objetivos.

5.3.2. Pruebas de integración y funcionamiento

Para comprobar el flujo de información entre cada capa se ha realizado las siguientes pruebas.

- Para el módulo de importación de cursos se ha procedido a la ejecución del fichero *script_all_courses.py*. En él se comprueba la comunicación entre sus submódulos y funcionamiento completo de dicha funcionalidad.
- Para la generación de modelos, se ha ejecutado el script de *model_script.R* y a continuación se ha realizado consultas a la base de datos para comprobación de la correcta creación del modelo e indicadores asociados. Posteriormente se ha visualizado sus estadísticas en la vista de la aplicación web de *Shiny*.
- En la creación de nuevas gráficas, menú de visualización de estadísticas y soporte de la elección de distintas variables de salida y de frecuencia en el menú de creación de los modelos predictivos se ha comprobado que la comunicación entre la interfaz gráfica y sus respectivos controladores era correcta.

5.3.3. Pruebas de sistema

Se ha probado el sistema en dos entornos diferenciados en sistemas operativos.

El entorno de desarrollo y el primero en el que se ha probado es un ordenador portátil comercial con Windows 8, 16GB de RAM y Core i7 4500U. El segundo también es un portátil con Windows 10 y 8GB de RAM, Core i5 7th.

Las pruebas han consistido en la instalación de cero y correcta ejecución y funcionamiento del sistema, coincidiendo con las pruebas de validación de las propuestas del proyecto.

6
Resultados

Este capítulo incluye los resultados obtenidos tras el uso del sistema *edX-MAS+* con datos de UAMx: *Equidad201x* (edición de 2016 3T), *Quijote501x* (ediciones 2015 1T y 3T, 2016 3T) y *Renal701x* (ediciones 2016 1T y 3T). Tras importar los cursos, se han creado modelos diarios de cada edición (6 en total) con todos los algoritmos (10) para las variables de salida (2). Es decir, se ha generado 7.080 modelos.

6.1. Resultados generales

En todas las ediciones se ha obtenido, tanto en la predicción de adquisición de certificado como de abandono, los siguientes resultados.

- El algoritmo que más tiempo de entrenamiento requiere es *Extreme Gradient Boosting*, seguido de *Neuronal Network* y *Random Forest* (ver la figura 6.1).
- El algoritmo con mayor tiempo de predicción es *Naive Bayes* (ver la figura 6.1).
- El algoritmo con menor precisión es *Classification and Regression Tree* (CART), es decir, la línea de valores AUC queda por debajo de las demás.
- Los indicadores menos importantes son *forum_events* y *forum_time*, tanto para la predicción de certificado como para predicción de abandono del curso.
- Los valores AUC de la predicción de certificado son mayores que los de abandono para todos los algoritmos. Es decir, obtenemos más precisión en la predicción de certificado que la de abandono (ver la figura 6.3).

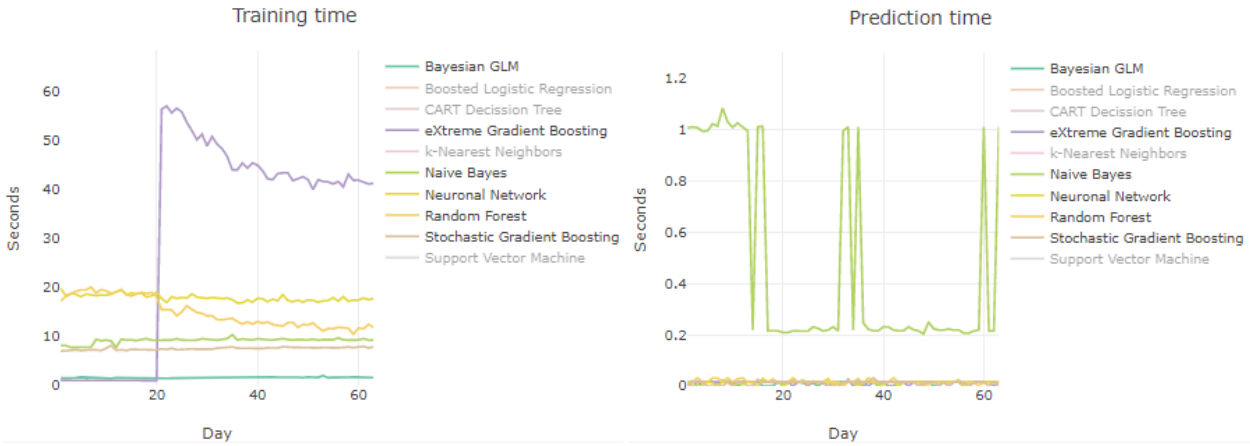


Figura 6.1: Ejemplos de tiempos de entrenamiento y predicción para *Quijote501x 3T2015*

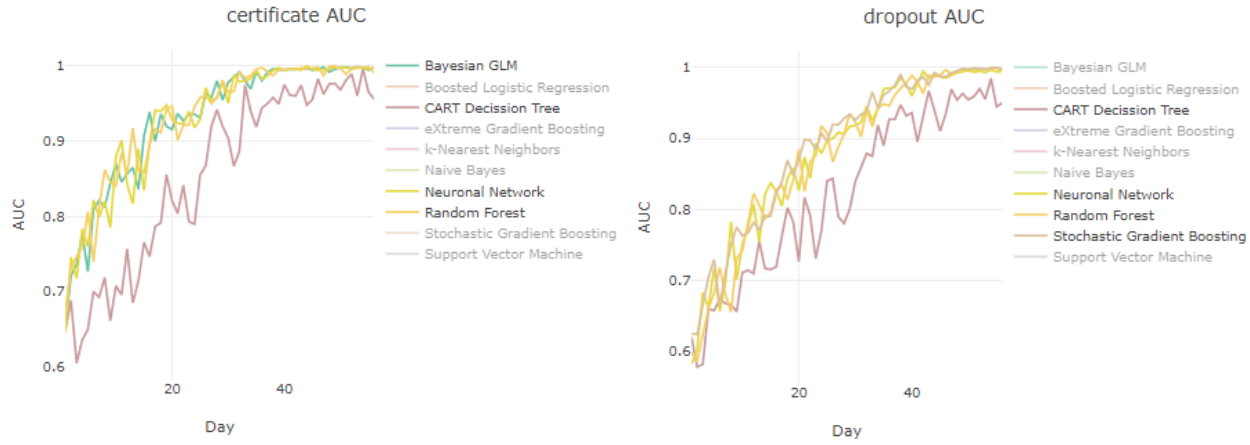


Figura 6.2: Ejemplo de AUC para ambas variables de salida del curso *Renal701x 1T 2016*

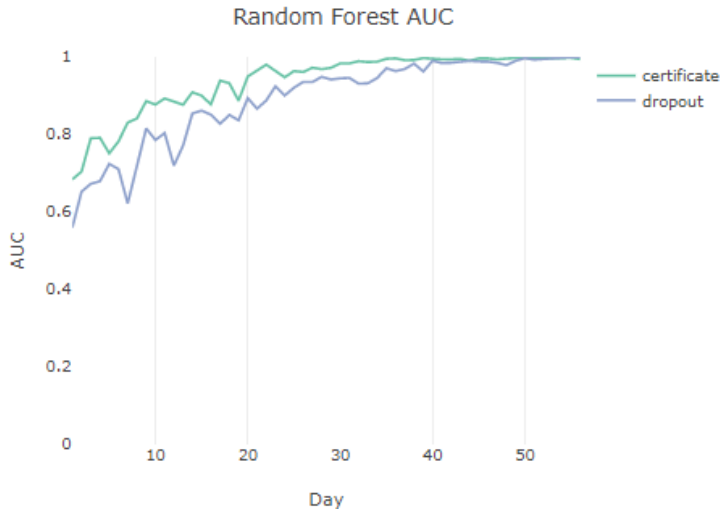


Figura 6.3: Ejemplo de comparativa AUC para el curso *Equidad701x 3T 2016*

6.2. Mejores algoritmos e indicadores

6.2.1. Predicción de adquisición de certificado

Los algoritmos con mayor precisión (mejores valores AUC) son: *Neuronal Network*, *RandomForest* y *Bayesian Generalized Linear Model* (ver la figura 6.2). Considerando también el tiempo de entrenamiento y la estabilidad de predicción el mejor algoritmo de los tres es *Bayesian Generalized Linear Model*.

Los mejores indicadores para los algoritmos mencionados coinciden: *num_ diff_ problems*, *problem_ events* y *num_ sessions*.

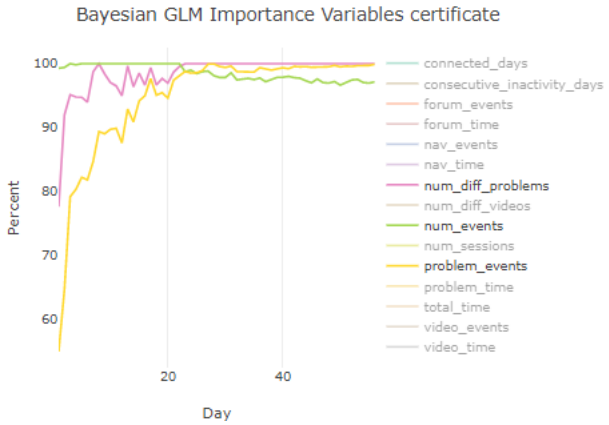


Figura 6.4: Importancia de las variables para certificado de *Quijote501x 3T2015*

6.2.2. Predicción de abandono del curso

El mejor algoritmo para la predicción de abandono es *Stochastic Gradient Boosting*, seguido de *Neuronal Network* y *Random Forest* (ver la figura 6.2). En la gráfica 6.5 se observa la importancia de las variables para el mejor. En el primer tercio del curso son muy importantes *num_ events* y *num_ sessions*, a medida que pasa el tiempo disminuye su importancia. A partir de la mitad del curso es muy importante *connected_ days*.

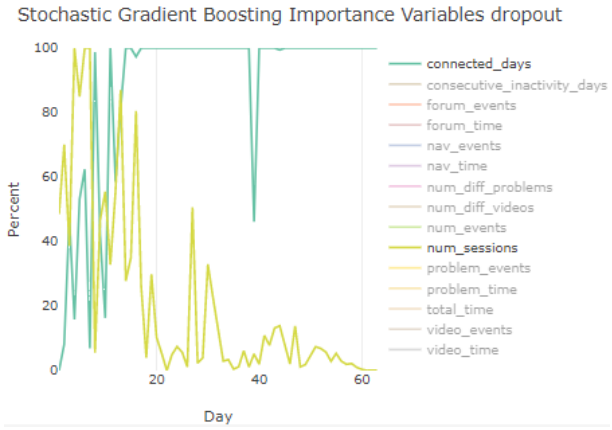


Figura 6.5: Importancia de las variables para abandono de *Quijote501x 3T2016*

7

Conclusiones y trabajo futuro

En este capítulo se incluye las conclusiones y los puntos de continuación para futuros trabajos.

7.1. Conclusiones

La Universidad Autónoma de Madrid ofrece cursos MOOC en la plataforma edX desde el año 2015. Gracias a la oficina de UAMx es posible al acceso de datos de dichos cursos para su análisis, permitiendo así realizar proyectos sobre analíticas de aprendizaje, entre ellos la herramienta *edX-MAS*, punto de partida de este trabajo.

El sistema presentado propone el modelado de abandono de los usuarios, la generación de modelos con nuevos algoritmos de aprendizaje automático para la predicción de adquisición de certificado (aprobado) y abandono del curso, la investigación y creación de nuevos indicadores de entrada y ampliación de generación semanal de modelos predictivos, mejoras en el modelo de datos, visualización y comparación de las estadísticas de los modelos.

Para el cumplimiento de las propuestas se ha completado los objetivos expuestos en el capítulo de introducción, sintetizados a continuación.

Se ha realizado un estudio del estado del arte general de los cursos MOOC y su principal problema, el abandono de los usuarios; de análisis de datos, modelos predictivos y algoritmos de aprendizaje automático, para después entrar en detalle de su aplicación en estudios recientes de analíticas de aprendizaje.

Para el modelado del abandono se ha procedido a la extracción y exploración de datos reales de cursos de UAMx disponibles para la creación de unas reglas de clasificación como definición de abandono versus al etiquetado manual de los usuarios.

El sistema amplía el diseño por módulos de *edX-MAS* manteniendo su escalabilidad.

Se ha probado con datos de tres cursos de UAMx y seis ediciones, generando 7.080 modelos durante la última semana del trabajo. Se ha obtenido que el algoritmo con mayor precisión para la predicción de aprobado es *Bayesian Generalized Linear Model* con indicadores más importantes *num_ diff _problems*, *problem_ events* y *num_ sessions*. Para el abandono, *Stochastic Gradient Boosting* con indicadores relevantes *connected_ days*, *num_ events* y *num_ sessions*. Los algoritmos que requieren mayor tiempo de entrenamiento son *Extreme Gradient Boosting*, *Random Forest* y *Neuronal Network*; más tiempo de predicción *Naive Bayes*. El algoritmo menos preciso es *Classification and Regression Tree* (CART) y los indicadores menos relevantes son los relativos a los foros.

7.2. Trabajo futuro

Hay varias líneas de desarrollo, posibles gracias a los criterios de escalabilidad y ampliación del sistema aplicados.

Una de ellas es la investigación de indicadores que midan eficiencia, progreso, tiempos y hábitos de aprendizaje, tomando como punto de partida el anexo B. Para su integración en el sistema conviene analizar su implementación como indicador acumulativo o no acumulativo, con procesado de eventos diarios o todos los eventos del curso mediante preprocesado. Algunos de ellos también implicarían el desarrollo de nuevos extractores de información de otros ficheros de datos de los cursos de UAMx. Por ejemplo, para medir el progreso en problemas, es necesario obtener el número de problemas de un curso a partir de los ficheros de estructura del mismo.

Otra línea de avance es el almacenamiento de más métricas para la evaluación y comparación de los modelos generados con el sistema, siguiendo los detalles facilitados en la implementación y escalabilidad.

Este sistema se utilizará para importar más cursos de UAMx y visualizar las estadísticas de los modelos predictivos para la obtención de resultados que generalicen los ya obtenidos.

Además de este trabajo, hay otros desarrollos en paralelo que también aprovechan los datos de UAMx, con distintos objetivos. Uno de ellos es un sistema de alarmas que alerte a los estudiantes que están en riesgo de suspender. Otro es la recomendación de ejercicios relevantes para mejora del rendimiento del alumno.

En un futuro, el sistema expuesto en este trabajo y los mencionados anteriormente se podrían integrar en un único sistema web.

Bibliografía

[1] Víctor Macías Palla. Herramienta para el modelado predictivo en entornos educativos en línea. *Trabajo de Fin de Grado. Escuela Politécnica Superior, Universidad Autónoma de Madrid*, Junio 2017.

[2] edX. `https://www.edx.org/`, [En línea. Noviembre 2017].

[3] Cursos gratuitos en línea de la Universidad Autónoma de Madrid. `https://www.edx.org/es/school/uamx`, [En línea. Noviembre 2017].

[4] Plotting and intrepretating an ROC curve. `http://gim.unmc.edu/dxtests/roc2.htm`, 2017. [En línea. Accedido Diciembre 2017].

[5] The area under an ROC curve. `http://gim.unmc.edu/dxtests/roc3.htm`, 2017. [En línea. Accedido Diciembre 2017].

[6] Yanyan Zheng. Big Data Analytics in MOOCs. *2015 IEE International Conference on Computer and Information Technology*, 2015.

[7] University of Washington. What is learning analytics? `http://www.uwb.edu/learningtech/elearning/learning-analytics?utm_campaign=elearningindustry.com&utm_source=%2F5-reasons-why-learning-analytics-are-important-for-elearning&utm_medium=link`, 2013. [En línea. Accedido Noviembre 2017].

[8] Daniel Jaramillo-Morillo; Mario Solarte Sarasty; Gustavo Ramírez González; Mar Pérez-Sanagustín. Follow-Up of Learning Activities in Open edX: A Case Study at the University of Cauca. *Digital Education. Out to the world and back to the Campus*, 2017.

[9] Ayse Saliha Sunar; Aniza Abdullah; Hugh C. Davis. How learners' interactions sustain engagement: a MOOC case study. *Journal of LATEX Class Files*, 2016.

[10] L.A. Ruipérez Valiente; R. Cobos; P.J. Muñoz-Merino; A. Andújar; C. Delgado Kloos. Early Prediction and Variable Importance of Certificate Accomplishment. *European MOOC Stakeholder Summit 2017 (eMOOCs 2017). Leganés, Madrid, España.*, 22-26 Mayo, 2017.

[11] Pedro Manuel Moreno Marcos; Carlos Alario Hoyos y Pedro José Muñoz Merino. Analítica del aprendizaje social en MOOCs: herramienta de visualización y predicción para los foros de edx. *Trabajo Fin de Máster. Universidad Carlos III*, 2017.

[12] Martin Hlosta; Zdenek Zdrahal; Jaroslav Zendulka. Early identification of at-risk students without models based on legacy data. *LAK '17 Vancouver, BC, Canada*, 2017.

[13] R. Boyatt D.F.O. Onah, J. Sinclair. Droupout Rates of Massive Open Online Courses: Behavioural Patterns. *The University of Warwick (United Kingdom).*

[14] eLearning Infographics. How can educational data mining and learning analytics enhance education infographic. `https://elearninginfographics.com/how-can-educational-data-mining-and-learning-analytics-enhance-%education-infographic`, 2014. [En línea. Accedido Noviembre 2017].

[15] Universidad Autónoma de Madrid. `http://www.uam.es/`, [En línea. Noviembre 2017].

[16] Cristina Isidro Estradas. Propuesta de un método basado en Deep Learning para learning analytics en MOOCs. *Trabajo de Fin de Máster. Escuela Politécnica Superior, Universidad Autónoma de Madrid*, 2017.

[17] R. Cobos; A. Wilde; Zaluska. Predicting attrition from Massive Open Online Courses in FutureLearn and ed. Comparing attrition prediction in FutureLearn and edX MOOCs. *Proceedings of the LAK FutureLearn Worshop in the Learning Analytics and Knowledge 2017 Conference (LAK17), Canada.*, 13-17 Mar 2017.

[18] I.D. Claros; R. Cobos; G. Sandoval; M. Villanueva. Creating MOOCs by UAMx: experiences and expectations. *European MOOC Stakeholder Summit 2015. Mons, Bélgica.*, 18-20 Mayo, 2015.

[19] I.D. Claros; L. Echeverría; A. Garmendia; R. Cobos. Towards a Collaborative Pedagogical Model in MOOCs. *Global Engineering Education Conference (EDUCON), 2014 IEEE (EDUCON 2014). Estambul, Turquía.*, 3-5 Abril, 2014.

[20] Ruth Cobos; Francisco Jurado. An Exploratory Analysis on MOOCs Retention and Certification in Two Courses of Different Knowledge Areas. *Global Engineering Education Conference (EDUCON), 2017 IEEE (EDUCON 2017).*, 2017.

[21] M. Leon; R. Cobos; K. Dickens. Internal Perspectives of MOOCs in Universities. *European MOOC Stakeholder Summit 2017 (eMOOCs 2017). Leganés, Madrid, España.*, 22-26 Mayo, 2017.

[22] M.L. Leon; R. Cobos; K. Dickens; Su White; H. Davis. Visualising the MOOC experience: a dynamic MOOC dashboard built through institutional collaboration. *eMOOCs 2016. 4th European MOOCs Stakeholders Summit.*, pages 461–470, Feb 22-24, 2016.

[23] C. Delgado Kloos; C. Alario-Hoyos; C. Fernández-Panadero; I. Estévez-Ayres; P. Muñoz-Merino; R. Cobos; J. Moreno; E. Tovar; R. Cabedo; N. Piedra; J. Chicaiza; J. López. Proyecto eMadrid: MOOCs y Analítica del Aprendizaje. SIIE16, CEDI2016. Salamanca, Castilla y León, España. 2016.

[24] R. Cobos; S. Gil; A. Lareo; F.A. Vargas. Open-DLAs: An Open Dashboard for Learning Analytics. *LS 2016 Third (2016) ACM Conference on Learning Scale Edinburgh, Scotland Uk*, pages 265–268, April 25 - 26, 2016.

[25] Ruth Cobos; Víctor Macías Palla. edx-MAS: Model Analyzer System. *TEEM 2017, Cádiz*, 2017.

[26] Predictive analytics. `https://en.wikipedia.org/wiki/Predictive_analytics`. [En línea. Accedido Noviembre 2017].

[27] Richard O. Duda; Peter E. Hart; David G. Stork. *Pattern Classification*. Wiley-Interscience Publication, Reading, Massachusetts, 2001.

[28] F1-score. `https://en.wikipedia.org/wiki/F1_score`, 2017. [En línea. Accedido Diciembre 2017].

[29] Essentials of machine learning algorithms. `https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/`. [En línea. Accedido Noviembre 2017].

[30] José R. Berrendero. Modelo de regresión simple. `http://verso.mat.uam.es/~joser.berrendero/cursos/adatos/ad2-tema3-12.pdf`.

[31] José R. Berrendero. Modelo de regresión múltiple. `http://verso.mat.uam.es/~joser.berrendero/cursos/adatos/ad2-tema4-12.pdf`.

[32] José R. Berrendero. Introducción al modelo de regresión logística. `https://caminosaleatorios.files.wordpress.com/2013/05/logistica.pdf`, 2013.

[33] Bayes' theorem. `https://en.wikipedia.org/wiki/Bayes%27_theorem`, 2017. [En línea. Accedido Noviembre 2017].

[34] Ensemble learning. `http://www.scholarpedia.org/article/Ensemble_learning`. [En línea. Accedido Noviembre 2017].

[35] J. H. Friedman. Stochastic Gradient Boosting. 1999.

[36] Extreme Gradient Boosting. `http://xgboost.readthedocs.io/en/latest/model.html`. [En línea. Accedido Noviembre 2017].

[37] Jerome Friedman; Trevor Hastie; Robert Tibshirani. Additive Logistic Regression: A Statistical view of Boosting. 2000.

[38] Priyanka Gaur. Neuronal Network in Data Mining. *International Journal of Electronics and Computer Science Engineering (JECSE)*.

[39] Mike Grosskopf. A short talk on bayesian generalized linear models. `http://rstudio-pubs-static.s3.amazonaws.com/15619_15a956c6bda84c3f97ef32793895689a.html#/`.

[40] University of Waikato. Trasparencias del libro Data Mining practical machine learning tools and techniques. `https://www.cs.waikato.ac.nz/ml/weka/book.html`. [FP-Growth en el capítulo 6] [Aprendizaje por reglas capítulos 3 y 4] [En línea. Accedido Noviembre 2017].

[41] Haydeé Cerón Reyes, María de los Ángeles y Gómez Díaz. Minería de datos. `https://www.slideshare.net/04071977/mineria-de-datos`, 2010. [En línea. Accedido Noviembre 2017].

[42] Association rule learning. `https://en.wikipedia.org/wiki/Association_rule_learning`, 2017. [En línea. Accedido Noviembre 2017].

[43] Vasiljevic Vladica. FP-Growth Algorithm. `https://es.slideshare.net/pradip8051/fp-growth-algorithm`.

[44] 2017 The State of Data Science Machine Learning. `https://www.kaggle.com/surveys/2017;https://www.kaggle.com/amberthomas/what-do-data-scientists-do`.

[45] Jacob Whitehill; Kiran Mohan; Daniel Seaton; Yigal Rosen; Dustin Tingley. MOOC Dropout Prediction: How to Measure Accuracy? *Cambridge, MA, USA*, 2017.

[46] Tim O'Riordan; David E. Millard; John Schulz. How should we measure online learning activity? *Annual Conference 2015. School of Electronics and Computer Science, University of Southampton (UK)*, 2015.

[47] José Antonio Ruipérez Valiente; Pedro José Muñoz Merino. Analyzing the behavior of students regarding learning activities, badges and academic dishonesty in MOOC environments. *PH.D. Thesis, IMDEA*, 1999.

[48] Max Kuhn. The caret package. `https://topepo.github.io/caret/`, 2017. [En línea. Accedido Noviembre 2017].

A

Planificación y dinámica de trabajo

En el desarrollo de este trabajo se ha empleado las siguientes tecnologías.

- Para realización de consultas *ad hoc* a la base de datos *pgAdmin*.
- *Sublime Text 3* como editor de texto en la codificación.
- *RStudio* para exploración de datos y generación de gráficas.
- Para control de versiones un repositorio privado en *Bitbucket*.
- Para la redacción de este documento LaTeX.
- Como gestor bibliográfico una *tablet*.
- Para la creación del diagrama de Gantt se ha empleado *SmartDraw* y los diagramas de clases y del modelo de datos *Gliffy*.

A continuación se presenta un diagrama de Gantt para el detalle de la realización de los objetivos del trabajo. También, diagramas para las tareas y lecturas de cada objetivo.

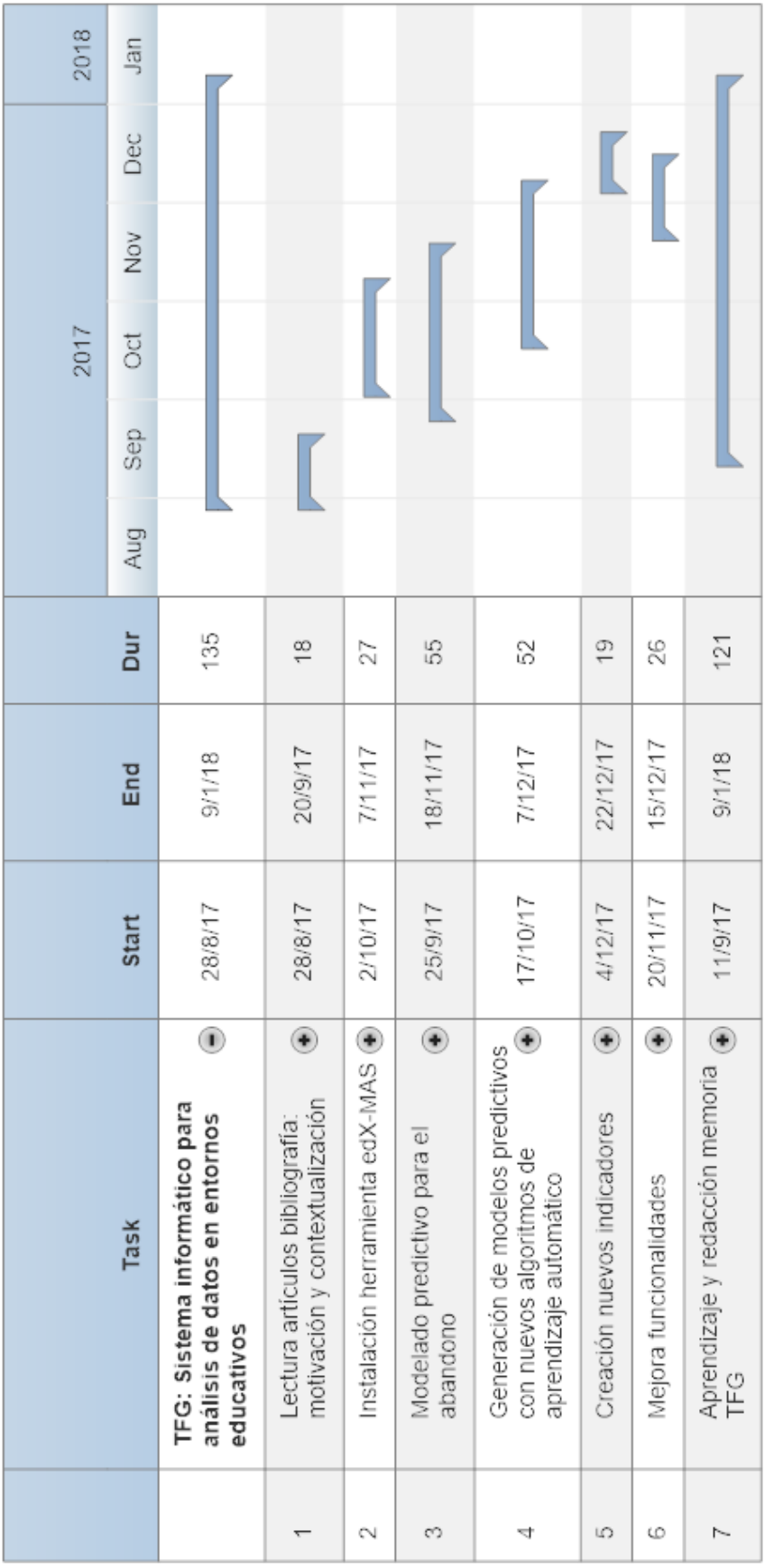
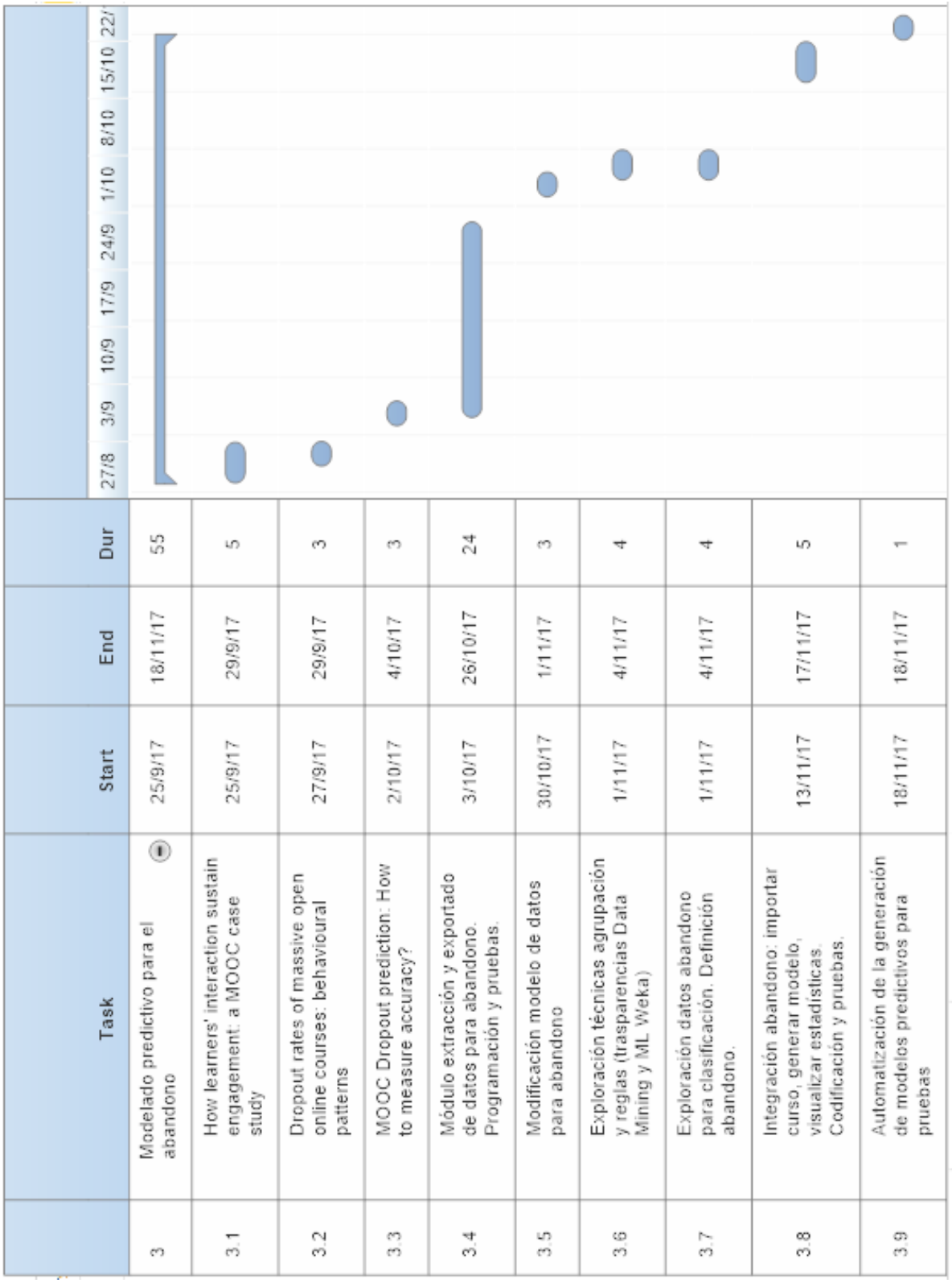


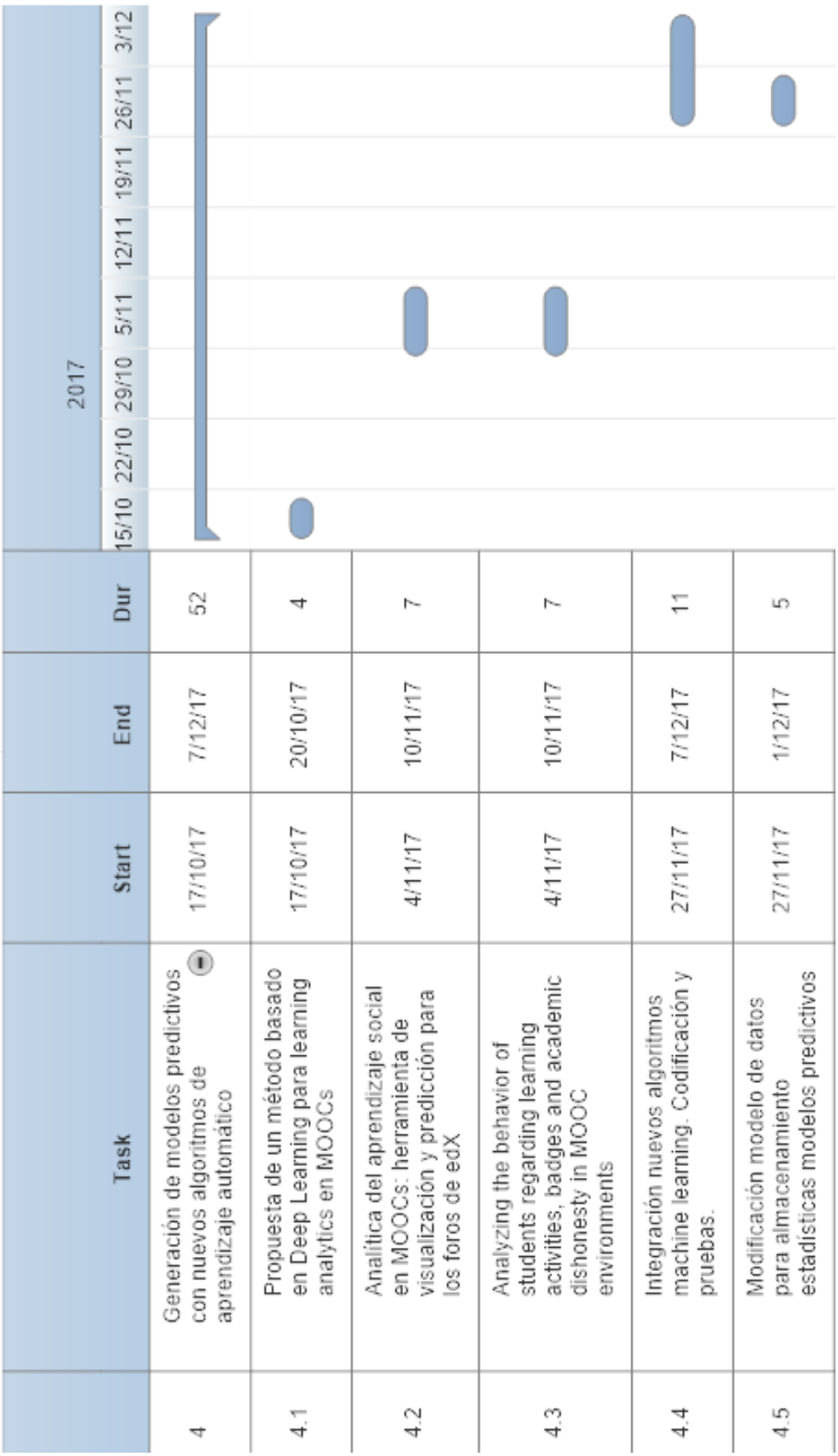
Figura A.1: Diagrama de Gantt reducido de los principales objetivos del trabajo

Figura A.2: Detalle del diagrama de Gantt del primer y segundo objetivo



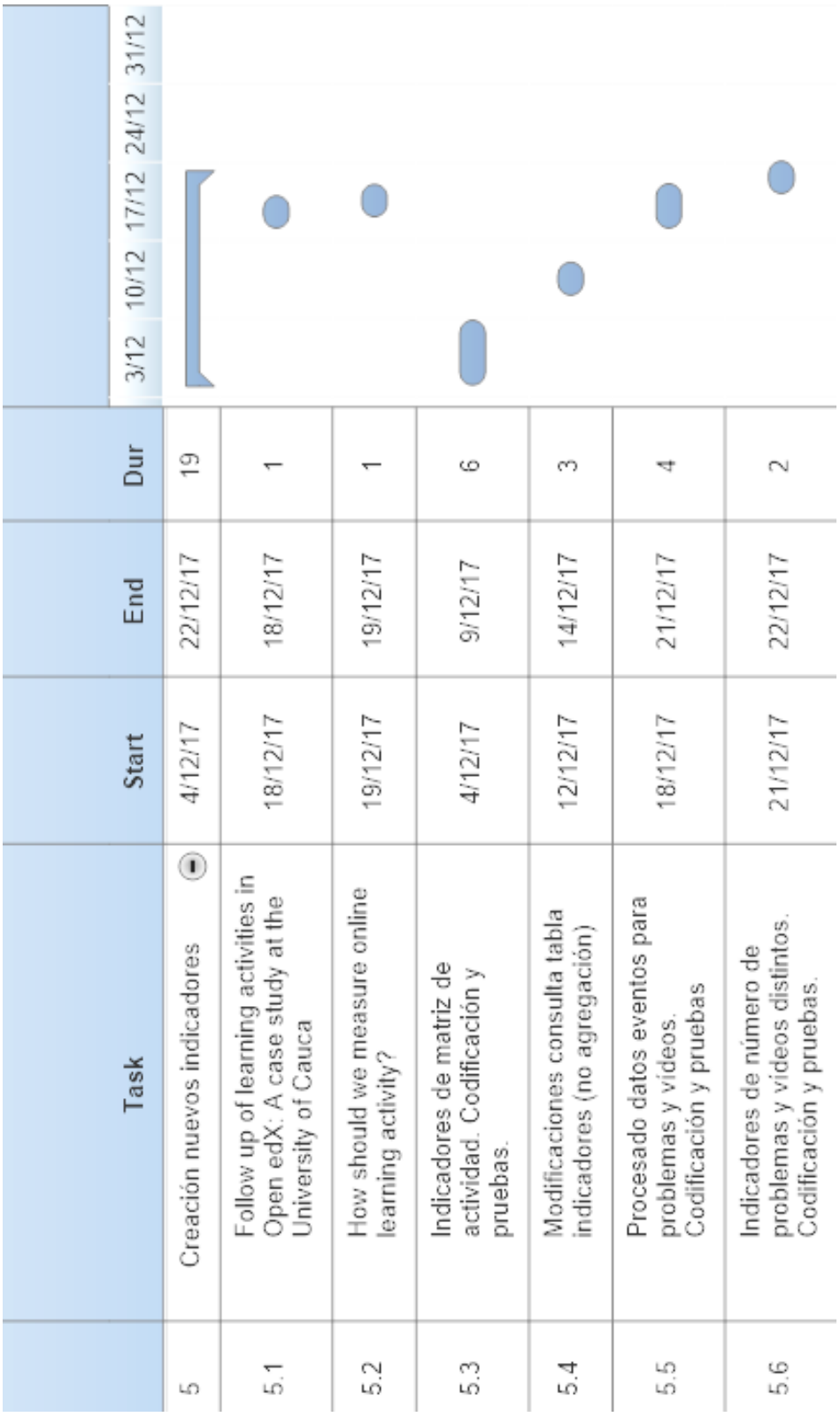
	Task	Start	End	Dur
3	Modelado predictivo para el abandono	25/9/17	18/11/17	55
3.1	How learners' interaction sustain engagement: a MOOC case study	25/9/17	29/9/17	5
3.2	Dropout rates of massive open online courses: behavioural patterns	27/9/17	29/9/17	3
3.3	MOOC Dropout prediction: How to measure accuracy?	2/10/17	4/10/17	3
3.4	Módulo extracción y exportado de datos para abandono. Programación y pruebas.	3/10/17	26/10/17	24
3.5	Modificación modelo de datos para abandono	30/10/17	1/11/17	3
3.6	Exploración técnicas agrupación y reglas (trasparencias Data Mining y ML Weka)	1/11/17	4/11/17	4
3.7	Exploración datos abandono para clasificación. Definición abandono.	1/11/17	4/11/17	4
3.8	Integración abandono: importar curso, generar modelo, visualizar estadísticas. Codificación y pruebas.	13/11/17	17/11/17	5
3.9	Automatización de la generación de modelos predictivos para pruebas	18/11/17	18/11/17	1

Figura A.3: Detalle del diagrama de Gantt del tercer objetivo



	Task	Start	End	Dur
4	Generación de modelos predictivos con nuevos algoritmos de aprendizaje automático	17/10/17	7/12/17	52
4.1	Propuesta de un método basado en Deep Learning para learning analytics en MOOCs	17/10/17	20/10/17	4
4.2	Analítica del aprendizaje social en MOOCs: herramienta de visualización y predicción para los foros de edX	4/11/17	10/11/17	7
4.3	Analyzing the behavior of students regarding learning activities, badges and academic dishonesty in MOOC environments	4/11/17	10/11/17	7
4.4	Integración nuevos algoritmos machine learning. Codificación y pruebas.	27/11/17	7/12/17	11
4.5	Modificación modelo de datos para almacenamiento estadísticas modelos predictivos	27/11/17	1/12/17	5

Figura A.4: Detalle del diagrama de Gantt del cuarto objetivo



	Task	Start	End	Dur
5	Creación nuevos indicadores	4/12/17	22/12/17	19
5.1	Follow up of learning activities in Open edX: A case study at the University of Cauca	18/12/17	18/12/17	1
5.2	How should we measure online learning activity?	19/12/17	19/12/17	1
5.3	Indicadores de matriz de actividad. Codificación y pruebas.	4/12/17	9/12/17	6
5.4	Modificaciones consulta tabla indicadores (no agregación)	12/12/17	14/12/17	3
5.5	Procesado datos eventos para problemas y vídeos. Codificación y pruebas	18/12/17	21/12/17	4
5.6	Indicadores de número de problemas y videos distintos. Codificación y pruebas.	21/12/17	22/12/17	2

Figura A.5: Detalle del diagrama de Gantt del quinto objetivo

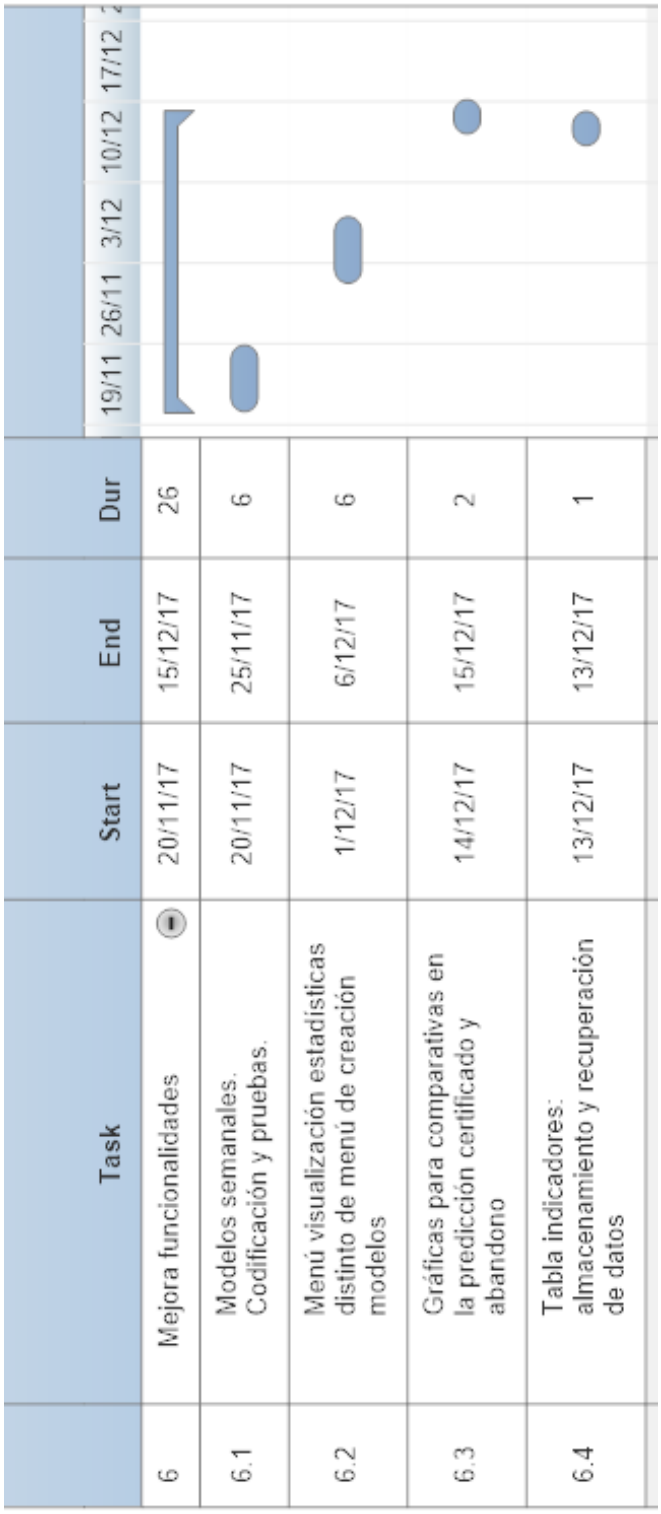
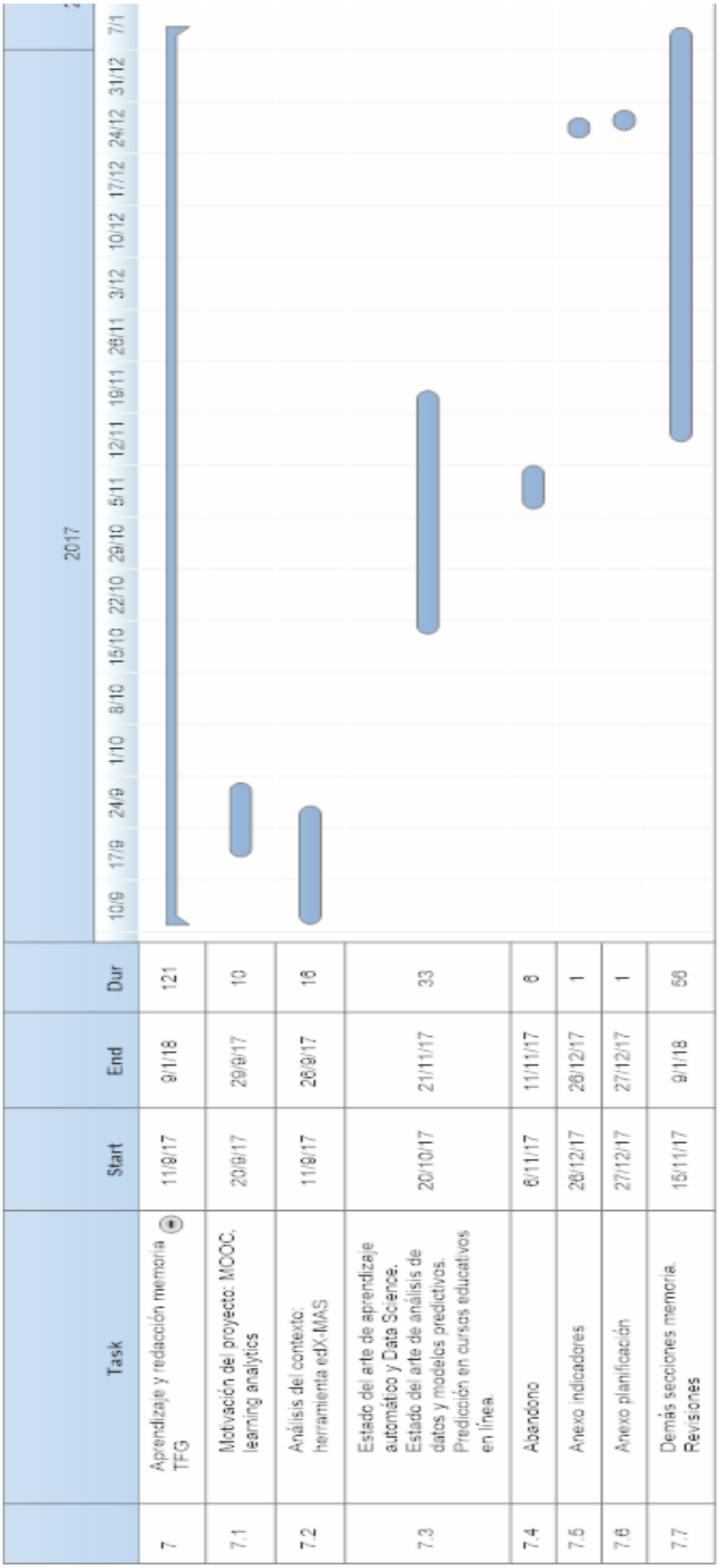


Figura A.6: Detalle del diagrama de Gantt del sexto objetivo



	Task	Start	End	Dur
7	Aprendizaje y redacción memoria TFG	11/8/17	9/1/18	121
7.1	Motivación del proyecto: MOOC, learning analytics	20/9/17	29/9/17	10
7.2	Análisis del contexto: herramienta edX-MAS	11/8/17	28/9/17	18
7.3	Estado del arte de aprendizaje automático y Data Science. Estado del arte de análisis de datos y modelos predictivos. Predicción en cursos educativos en línea.	20/10/17	21/11/17	33
7.4	Abandono	6/11/17	11/11/17	6
7.5	Anexo indicadores	26/12/17	26/12/17	1
7.6	Anexo planificación	27/12/17	27/12/17	1
7.7	Demás secciones memoria. Revisiones	15/11/17	9/1/18	58

Figura A.7: Detalle del diagrama de Gantt del séptimo objetivo

B

Indicadores de analíticas de aprendizaje

En este anexo se ha agrupado los **indicadores** encontrados en los estudios de la bibliografía en las siguientes categorías: **interacción en las plataformas en línea**, **actividad**, **generados por los alumnos** e indicadores **particulares** de los estudiantes (demográficos, datos de registro y del contexto de los estudiantes). Otra clasificación encontrada en [47] es: indicadores del uso de la plataforma, correcto avance del usuario en la plataforma, tiempo en la plataforma, comportamientos en la resolución de ejercicios. Se recomienda ver la sección *Selected indicators* para más detalle.

Los indicadores de interacción y de actividad permiten monitorizar el aprendizaje y comportamiento de los estudiantes, además de predecir aprobado y suspenso. Permiten medir su progreso, interés, persistencia, constancia, efectividad, capacidad y participación. Para una asociación de estudios bibliográficos por cada detalle mencionado, consultar la tabla de [8], sección de *Indicators*.

B.1. Indicadores particulares y generados por los estudiantes

Por ejemplo, en el estudio [12], se referencian indicadores de edad, región y sexo del estudiante. También la fecha de alta del estudiante en el curso. Estos indicadores se pueden considerar en cursos que requieran un registro previo y permitan matriculacularse aunque ya haya comenzado.

Para indicadores de materiales generados por los estudiantes se han encontrado procesados con técnicas de procesamiento de lenguaje natural y análisis de sentimiento y de contenido [11, 46], análisis social (foros con estructuras de seguimiento [9]) y gamificación (con insignias o recompensas en los cursos, [47]).

B.2. Indicadores de interacción

Indicador	Definición	Referencias
number_events	Número total de eventos	[10], [16], [25], [47]
number_sessions	Número total de sesiones	[10], [16], [25], [47]
total_time	Tiempo total aplicado en un curso	[10], [16], [25], [47]
nav_events	Número de eventos de nav. entre contenidos	[25]
nav_time	Tiempo entre eventos de nav. entre contenidos	[25]
video_events	Número de eventos de interacciones de vídeo	[25]
video_time	Tiempo entre eventos de vídeo	[25]
total_video_time	Tiempo total de reproducción de vídeos	[10], [47]
forum_events	Número de eventos del foro	[25]
forum_time	Tiempo entre eventos del foro	[25]
problem_events	Número de interacciones con los problemas	[25]
problem_time	Tiempo entre interacciones con los problemas	[10], [25]
total_problem_time	Tiempo total aplicado en problemas	[10], [16] , [47]
page_time	Tiempo total aplicado en páginas del curso	[47]
average_time_per_day	Tiempo medio empleado por día del curso	[47]

Cuadro B.1: Tabla de indicadores de interacción de analíticas de aprendizaje

B.3. Indicadores de actividad

Indicador	Definición	Referencias
number_days	Días que el usuario ha entrado en el curso (activos)	[10], [47]
num_videos	Vídeos distintos accedidos	[16], [47]
num_problems	Problemas distintos accedidos	[16], [47]
problem_progress	Porcentaje de problemas completados	[10], [47]
video_progress	Porcentaje de vídeos vistos	[10], [47]
completed_videos	Vídeos visualizados por completo	[47]
num_comments	Comentarios aportados al foro	[16], [47]
optional_activities	Actividades opcionales realizadas	[47]
attempts_required	Intentos hasta contestar correctamente un problema	[16], [47]
successful_problems	Problemas contestados correctamente	[16]
avg_time_correct	Tiempo medio hasta contestar correctamente	[47]
avg_num_hits	Número medio de pistas visualizadas	[47]
problem_left	Porcentaje de vídeos accedidos sin completar	[16], [47]
video_left	Porcentaje de problemas accedidos sin completar	[16], [47]
problem_homogeneity	Distribución de actividad con dispersión del tiempo	[10]
video_homogeneity	Dispersión del tiempo empleado en un vídeo	[10]

Cuadro B.2: Tabla de indicadores de actividad de analíticas de aprendizaje

C

Árbol de decisión de la clasificación de abandono

Se incluye el árbol de decisión obtenido con las reglas definidas para la clasificación de abandono.

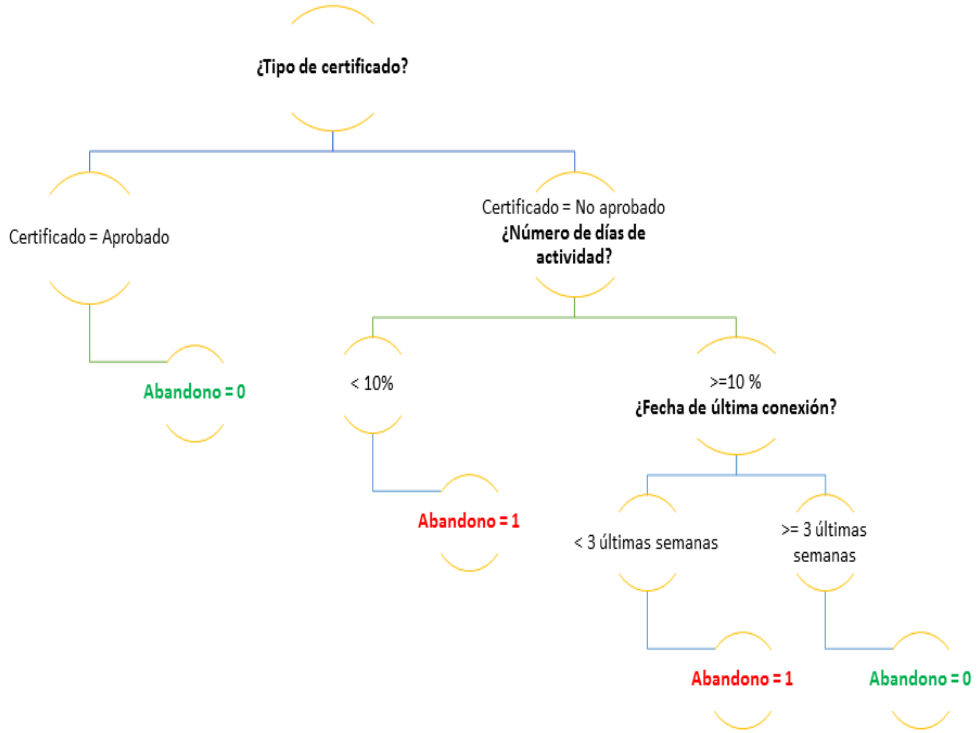


Figura C.1: Árbol de decisión obtenido de las reglas de clasificación de abandono

D

Visualización del sistema

A continuación se incluye imágenes de la interfaz gráfica del sistema *edX-MAS+*.

En primer lugar, se observa la pantalla inicial y los menús de la columna lateral izquierda.

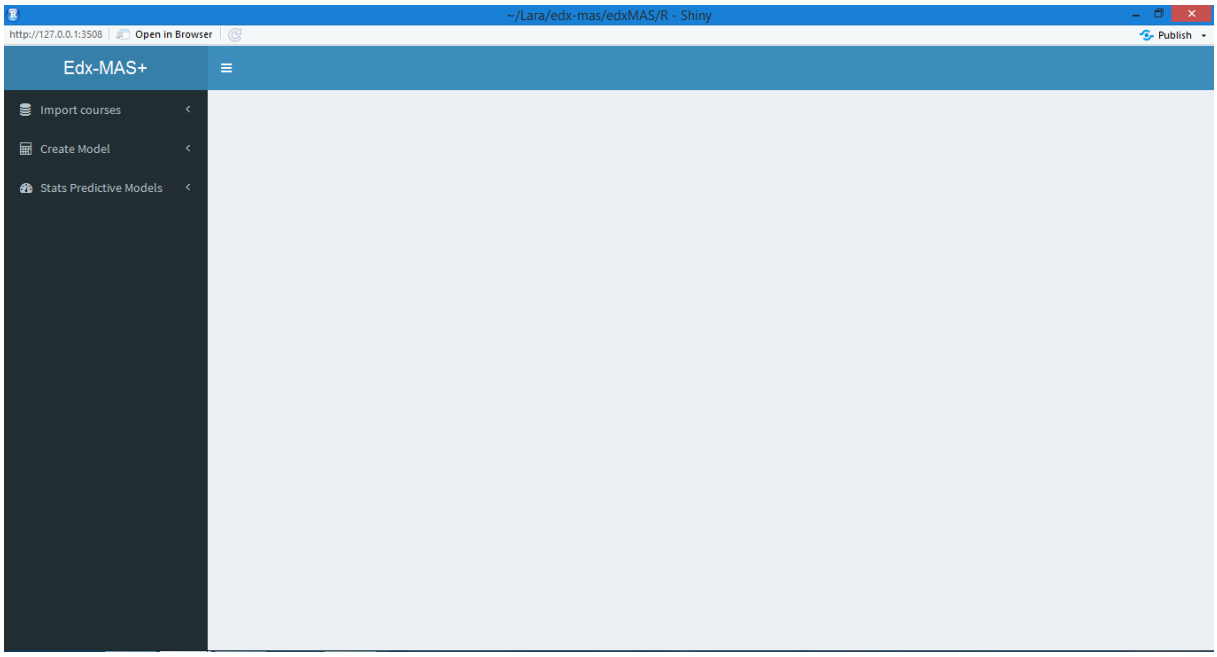


Figura D.1: Visualización inicial

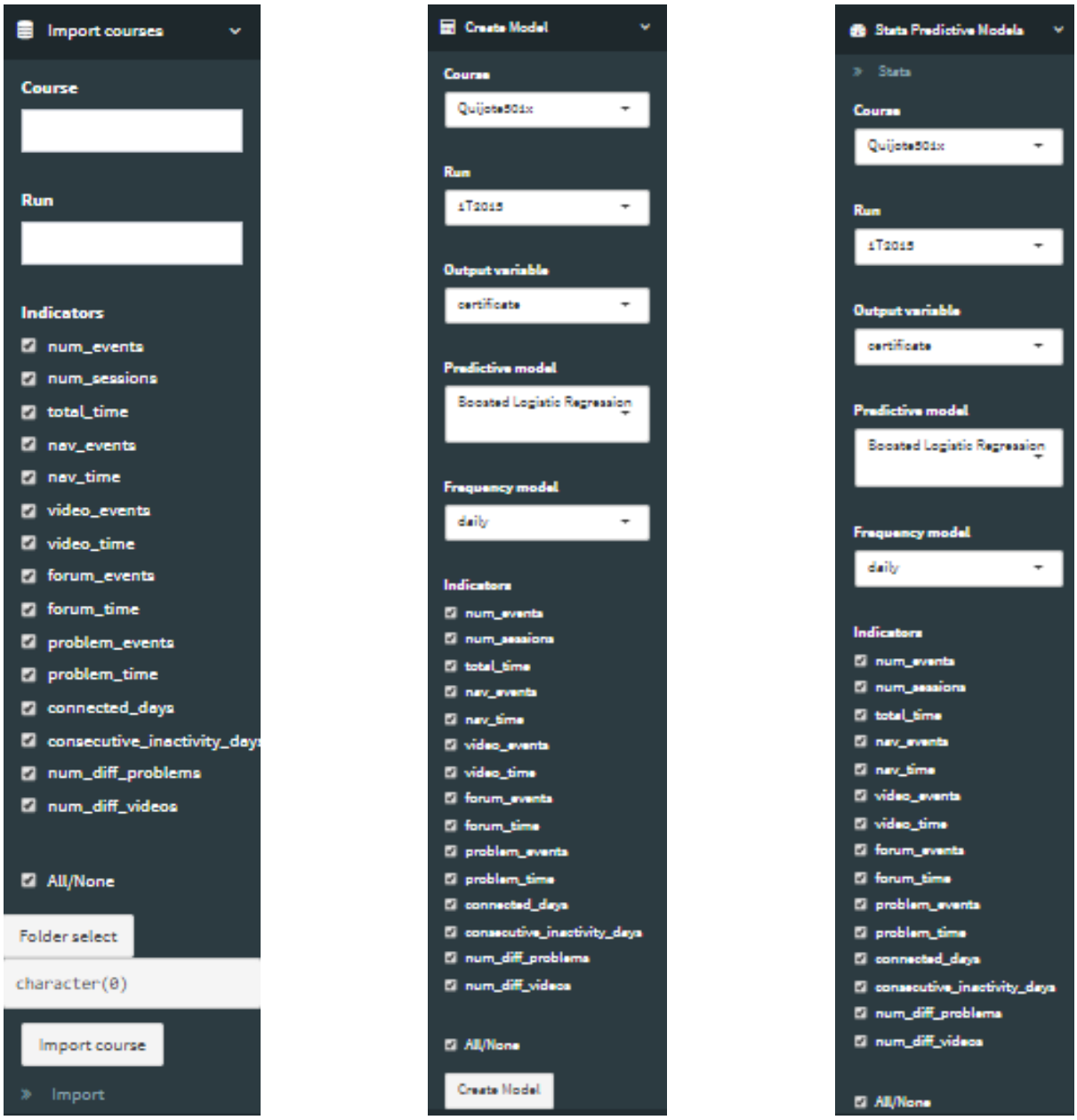


Figura D.2: Visualización menús: importar curso, crear modelo y ver estadísticas

Tras pulsar el enlace de *Stats* del menú de visualización de estadísticas se muestra el panel central con las pestañas de D.3.

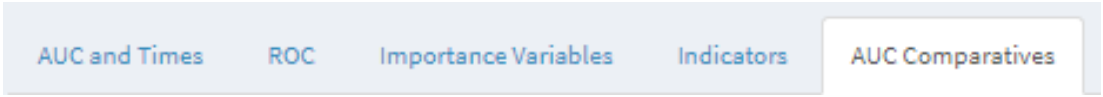


Figura D.3: Pestañas de visualización de estadísticas

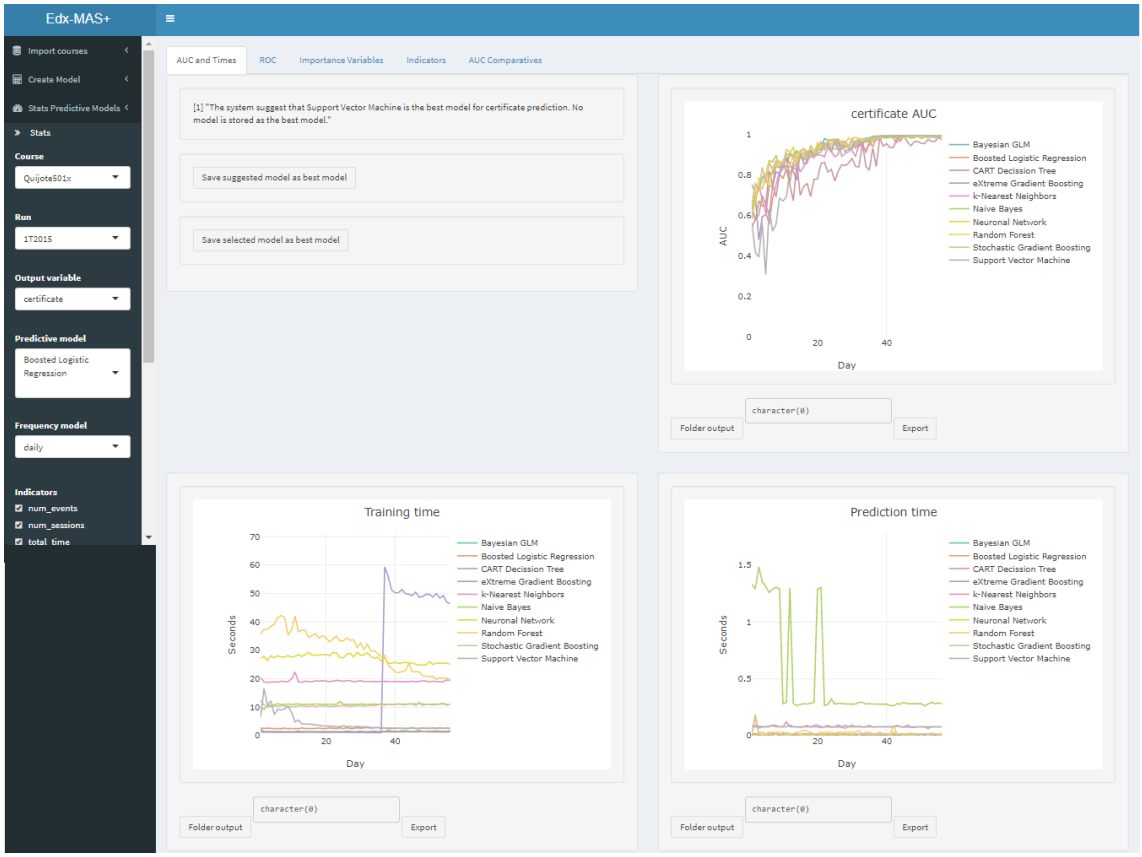


Figura D.4: Visualización pestaña *AUC and Times*

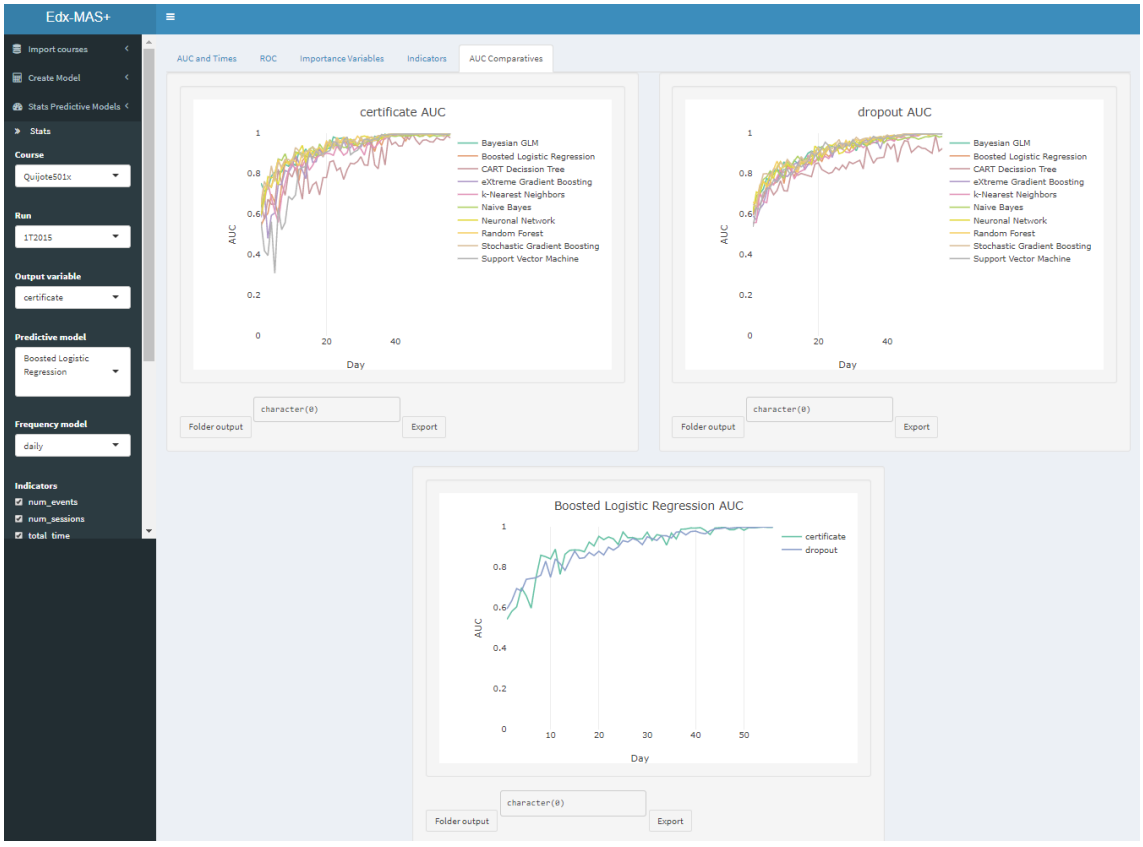


Figura D.5: Visualización pestaña *AUC Comparatives*

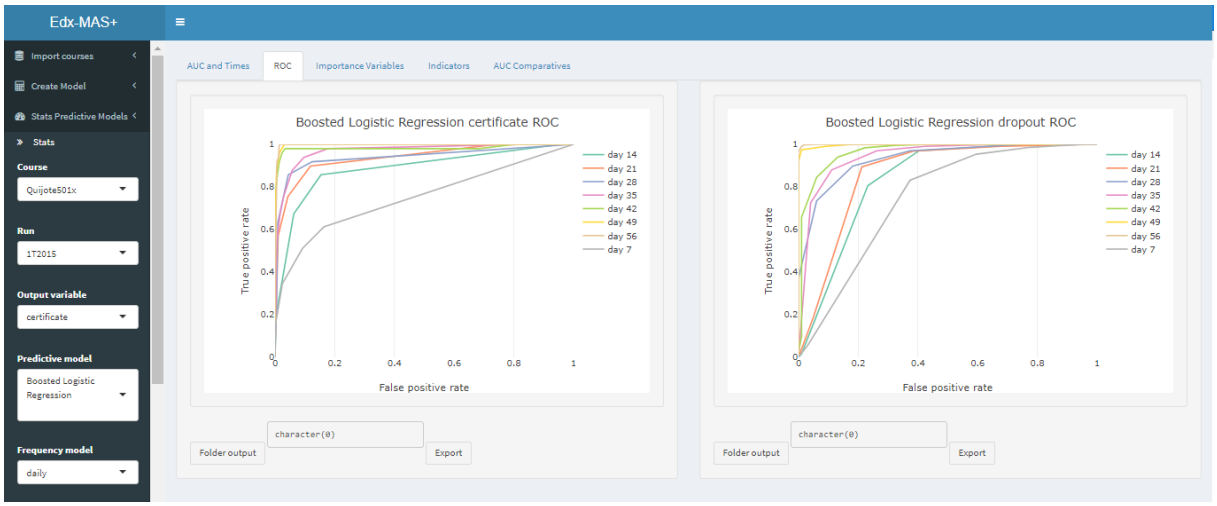


Figura D.6: Visualización pestaña *ROC*

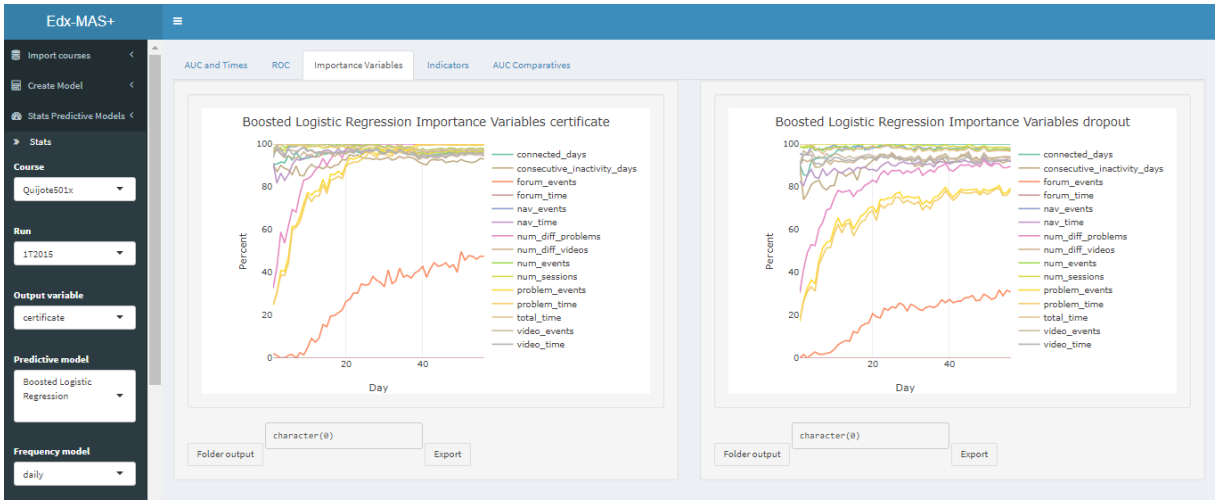


Figura D.7: Visualización pestaña *Importance Variables*

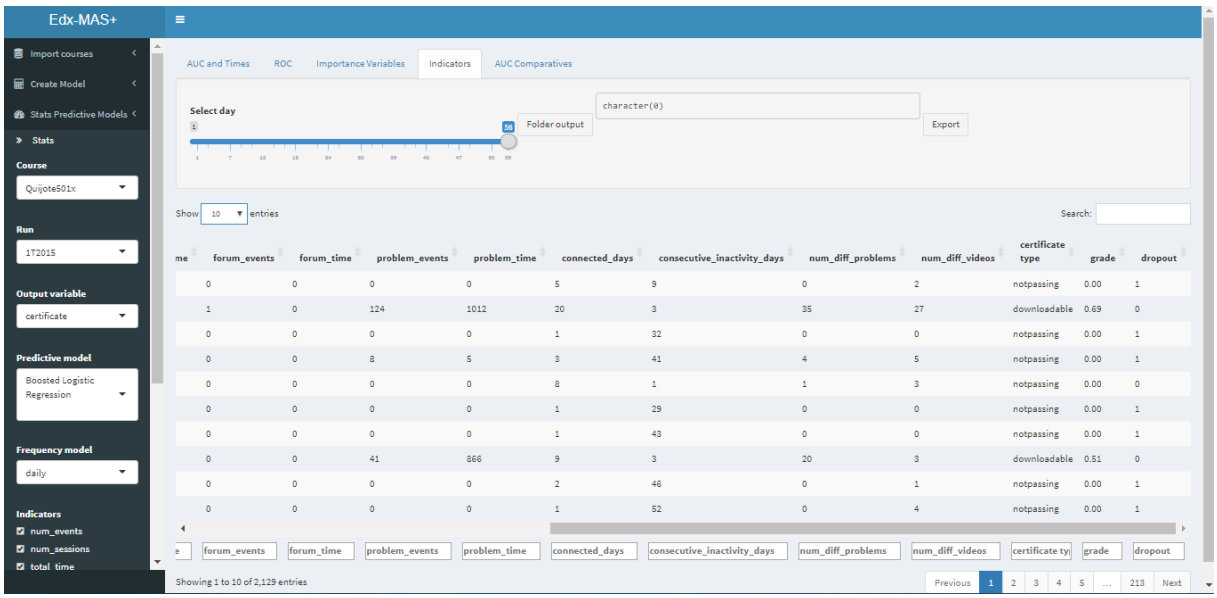


Figura D.8: Visualización pestaña *Indicators*