Universidad Autónoma de Madrid
Escuela politénica superior


Grado en Ingeniería Informática

Trabajo Fin de Grado

# TÉCNICAS DE APRENDIZAJE ACTIVO EN INTELIGENCIA ARTIFICIAL

Autor: Víctor Salgado Rodríguez
Tutor: Manuel Sánchez-Montañés Isla

JUNIO 2016

# TÉCNICAS DE APRENDIZAJE ACTIVO EN INTELIGENCIA ARTIFICIAL

Autor: Víctor Salgado Rodríguez
Tutor: Manuel Sánchez-Montañés Isla

Escuela Politécnica Superior
Universidad Auténoma de Madrid

# Resumen


## Resumen

En un mundo tan globalizado y multiculturizado como en el que vivimos ahora se ha introducido como elemento indispensable la tecnología, siendo fundamental e imprescindible en el día a día de nuestra sociedad. En un ámbito tan extenso cómo este, uno de los motores de desarrollo y avance es la inteligencia artificial.

Una de las principales ideas que surgen en torno a la inteligencia artificial es explotar la gran cantidad de información resultante del uso de esta tecnología. Para conseguir que esta idea se convierta en realidad se han empezado a desarrollar técnicas que permiten a los computadores extraer patrones y realizar predicciones partiendo de esa gran cantidad de información. Este problema lo intentan solventar diferentes ramas como el aprendizaje automático, el big data o el data mining.

Dentro del campo del aprendizaje automático, una de las técnicas es el aprendizaje activo, cuyo objetivo principal es obtener resultados igual o mejores que técnicas de aprendizaje automático estándar, pero utilizando el número de datos etiquetados posible. Para conseguir esto, los algoritmos de aprendizaje activo deben ser capaces de detectar queé datos son los más informativos. En este trabajo se evaluarán, entre otras cosas, diferentes técnicas de seleccionar dichos datos.

Durante todo este Trabajo Fin de Grado se ha realizado un estudio sobre el comportamiento y funcionamiento de la técnica de aprendizaje activo. En primer lugar se han llevado a cabo estudios sobre diferentes comportamientos del propio algoritmo, utilizando diferentes conjuntos de datos, diferentes clasificadores, diferentes técnicas de aprendizaje, y diferentes criterios de parada. En paralelo se ha comparado el rendimiento de los algoritmos activos con algoritmos estándar que no implementan esta técnica, con el fin de evaluar si el aprendizaje activo tiene un rendimiento mejor que el pasivo.

Finalmente se ha realizado un análisis comparativo de todos los resultados obtenidos y se ha presentado una serie de conclusiones sobre el tema abordado.



## Palabras Clave

Inteligencia Artificial, Aprendizaje Automático, Aprendizaje Activo, Aprendizaje Semisupervisado

## Abstract

In a globalized and multicultural world like the one we currently live at we introduced an indispensable element, technology, which is now fundamental for our society in the day to day routine. In such an extense topic, one of the motors of development and progress is artificial inteligence.

One of the principal ideas that arrise regarding artificial inteligence is to be able to explore the great quantity of information, result of the use of this technology. To be able to make this idea come to life, they have started to develop techniques that allow computers extract patterns and produce predictions from the large amount of information gatherd. This problem tries to be fixed in different ways such as machine learning, the big data or data maining.

In this case, we will focus on the study of artificial intenligence.

Inside the field of artificial inteligence, one of the techniques is active learning, which it's main objective is to obtain results like or better than machine learning, but using a smaller number of data. To achieve this, the algorithms of active learning have to be capable of detecting wich information is the most informative. In this essay we will evalue, amongst others, different techniques to select this data.

In this Bachelor's Thesis a study has been made about the behaviour and functionality of the technique of active larning. On first place, there have been several studies about the different behaviours of the alorithm, using different data compounds, different classifiers, and diferent stop criteria. At the same time I have been comparing the progress of the active eith the standard algorithms that do not implement this technique, with the aim of evaluating if active larning has greater efficiency than the passive algorithm.

Finally an evaluation has been made comparing all of the results obtained, and have been presented in a series of conclussions about the subject issued.



## Key words

Artificial Intelligence, Machine Learning, Active Learning, Semi-supervised Learning.

# Agradecimientos

A mis padres, por hacer de mi la persona que soy ahora y haberme inculcado sus valores y principios. Enseñarme que la constancia y el trabajo son el camino para conseguir todo aquello que me proponga.

A mis hermanos, novia y amigos porque gracias a su apoyo durante estos cuatro años he podido sobreponerme a las adversidades y afrontar nuevos retos con más fuerza y ganas.

A la Escuela Politécnica Superior de la Universidad Autónoma de Madrid por haberme dado la oportunidad de conocer a un grupo de amigos informáticos que quedarán para siempre, y donde juntos hemos pasado tan buenos momentos a lo largo de este ciclo.

A todos los profesores que han contribuido a que durante este tiempo me haya ido enriqueciendo con nuevos conocimientos, en especial a Manuel por su apoyo en este trabajo.

# Índice general

# Índice de Figuras

# Índice de Tablas

# 1 Introducción

## 1.1. Motivación del proyecto

Dentro del ámbito de la Inteligencia Artificial (IA), uno de los campos más importantes es el aprendizaje automático (AA ó ML), cuyo objetivo principal consiste en el desarrollo de algoritmos capaces de aprender a realizar una tarea determinada a partir de ejemplos, es decir, identificar patrones complejos entre una cantidad de datos [1]. Es en ese punto donde aparece el problema que nosotros abarcaremos. En muchos casos el número de datos necesarios para obtener un resultado satisfactorio es muy elevado, lo que se puede traducir en un alto coste de esfuerzo y económico. Por ejemplo, si una empresa desea realizar un modelo predictivo para detectar a qué clientes les interesará más un producto, deberá recopilar primero una base de datos con clientes interesados y otros no interesados. Pero para recopilar dicha base de datos necesitará contactar con ellos y consultarles si les interesa el producto, lo que conlleva un coste que puede llegar a ser elevado (y aumenta linealmente con el número de clientes consultados).

Con el objetivo de solventar este tipo de problemas, en el que se desea (o hay un límite) en el número de patrones etiquetados (con la clase objetivo a predecir), surge el campo del Aprendizaje Activo (AL). En este campo se desarrollan algoritmos que se encargan de ir solicitando las etiquetas de sólo aquellos ejemplos que contribuirán más al aprendizaje del modelo, minimizando de esta forma el número de patrones etiquetados que necesita el algoritmo.

En este trabajo se estudiará el comportamiento de diferentes técnicas de aprendizaje activo comparándolas entre ellas y con técnicas no activas. Para ello se ha desarrollado un algoritmo genérico que permite convertir cualquier técnica de aprendizaje pasivo en activo. Con él, se comparará el funcionamiento de las mejores técnicas de aprendizaje activo hasta el momento según la literatura. Para ello se estudiaron bases de datos de naturaleza muy diferente para obtener conclusiones lo más generales posibles.

## 1.2. Objetivos y enfoque

El objetivo principal de este Trabajo Fin de Grado es realizar un estudio experimental del comportamiento de los algoritmos más relevantes actualmente en Aprendizaje Activo. Se

identificarán sus pros y contras, y bajo qué condiciones su capacidad de generalización es óptima. Para ello se abordarán los siguientes objetivos secundarios:

- Realizar pruebas experimentales para evaluar el comportamiento del algoritmo activo.
- Realizar pruebas experimentales con el algoritmo equivalente no activo.
- Comparar estadísticamente los resultados obtenidos del aprendizaje activo frente al aprendizaje pasivo.
- Realizar un estudio sobre queé clasificador base es el mejor para el algoritmo de aprendizaje activo.
- Realizar un estudio sobre queé criterio de selecci'on de patrones es mejor.
- Realizar un estudio sobre los diferentes criterios de parada y realizar pruebas experimentales de los resultados obtenidos en el estudio.

## 1.3. Organización de la memoria

En el capítulo 2 se presentará el estudio del estado del arte realizado con la finalidad de conocer el estado actual en esta materia y explicar en detalle sus bases.

En el capítulo 3 se explica en detalle la metodología llevada a cabo durante todo el proyecto, detallando los aspectos más relevantes y explicando en detalle los algoritmos utilizados.

En el cuarto capítulo se muestran las pruebas y resultados obtenidos. En él se mostrarán los resultados en función de la base de datos base de datos estudiada y evaluando el algoritmo activo sobre varios escenarios en función de los diferentes parámetros estudiados.

Finalmente, en el capítulo cinco encontraremos nuestras conclusiones sobre el trabajo realizado, y las posibles líneas de investigación en el trabajo futuro.

Acto seguido y para finalizar el cuerpo del documento se presenta la bibliografía utilizada durante todo el trabajo.

# 2

# Estado del arte

## 2.1. Introducción

Típicamente, en un problema complejo cualquier algoritmo de aprendizaje automático necesita muchas instancias etiquetadas (es decir, cuya clase es conocida y se le da al sistema) para obtener un buen rendimiento. En numerosas ocasiones la obtención de estos datos puede resultar difícil, costosa, o simplemente no es deseable proporcionar al algoritmo una gran cantidad de datos etiquetados.

Un ejemplo es una empresa que fabrique motores y tenga como objetivo construir un modelo que prediga el tiempo de vida de los mismos, o de un nuevo modelo. Supondría un coste muy alto tener que forzar los motores hasta el final de su uso, el coste económico en caso de fabricar tantos motores como para comprobar sus patrones de funcionamiento sería extremadamente alto. Otro ejemplo es una empresa de videovigilancia que quiere desarrollar un sistema que, a partir de los vídeos obtenidos por una cámara, detecte automáticamente si hay alguna situación de alarma (por ejemplo, un sistema que detecte hurtos en los andenes del Metro). Aunque la empresa disponga de filmaciones a lo largo de meses, necesitará de un experto que detecte y etiquete las situaciones concretas de hurto, lo cual puede ser muy costoso en tiempo y esfuerzo.

Dentro del aprendizaje automático, las denominadas técnicas de aprendizaje activo construyen el modelo y el conjunto de entrenamiento de manera incremental con el objetivo de que este sea lo menor posible. Para ello, en cada iteración el algoritmo elige dentro de un conjunto de datos no etiquetados aquellos que considera más informativos, y solicita al usuario que sean etiquetados. Antes de continuar puntualizaremos que el término de aprendizaje activo se usa en otros campos como el de la educación, pero no tiene ninguna relación con la técnica que estudiaremos en este TFG.

Un entorno relevante donde se puede reflejar la utilidad de esta técnica son las redes sociales, un ámbito en auge en la actualidad y que recoge una cantidad masiva de información. Un ejemplo donde el aprendizaje activo puede ser muy relevante es una empresa que quiere desarrollar un sistema que detecte cuándo se está hablando de ella en las redes sociales, y si es de manera positiva o no (análisis de sentimiento). Para desarrollar el modelo se deberá construir una base de datos de textos, etiquetando cada uno de ellos como positivo o negativo. La obtención de datos en este caso no es muy costosa, ya que toda la cantidad de información de la red es accesible. Pero sí obtendríamos un coste muy alto al etiquetar dicha información, ya que en este

caso quien evaluaría la información obtenida sería un experto, con lo que conllevaría tiempo, esfuerzo y coste proporcional al tamaño de la base de datos creada.

En definitiva el aprendizaje activo intenta aliviar el cuello de botella que supone el coste de obtención de información etiquetada, realizando para ello consultas a un proveedor de información que a partir de ahora denominaremos oráculo. De este modo, el objetivo del sistema será lograr una buena tasa de error usando el menor número de datos etiquetados posible, minimizando de esta forma el coste de obtención de la información.

En la siguiente imagen se muestra el ciclo de un algoritmo de aprendizaje activo.

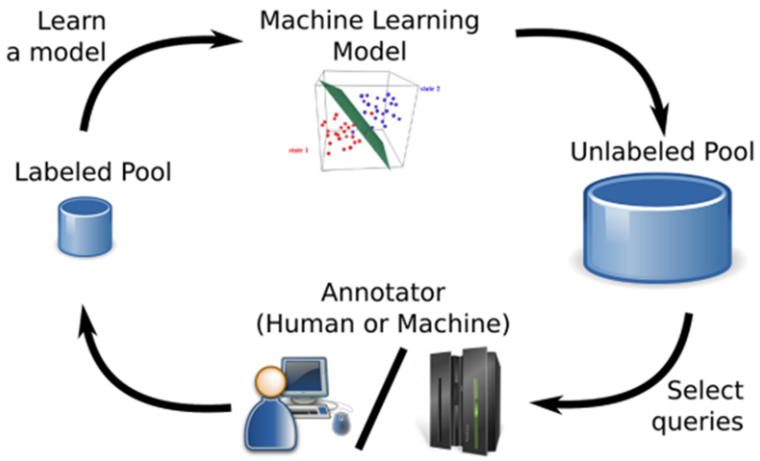


Figura 2.1: Representación del ciclo de vida aprendizaje activo [https://www.cs.cityu.edu.hk/]

Tal y como se ilustra en la 2.1, el ciclo de vida de un algoritmo activo es el siguiente:

- El algoritmo comienza con un conjunto pequeño de instancias etiquetadas en su conjunto de entrenamiento, y un conjunto de datos no etiquetados. Se fija además el valor de K (número de instancias que el algoritmo solicita que se etiqueten en cada iteración).
- El algoritmo construye un clasificador usando el conjunto de entrenamiento
- Dicho clasificador se usa para seleccionar los K patrones más informativos dentro del conjunto no etiquetado, y solicita sus etiquetas. En este punto el algoritmo está realizando la consulta.
- Se incorporan dichos datos etiquetados al conjunto de entrenamiento, y se eliminan del conjunto de datos no etiquetados.
- El proceso de aprendizaje se repite hasta satisfacer la condición de parada elegida.

Hay diferentes escenarios donde los algoritmos de aprendizaje activo pueden plantear qué preguntar. Así mismo hay varias estrategias de consulta que han sido utilizadas para decidir que instancias son más informativas.

## 2.2. Escenarios de consulta

En esta sección se describirán diferentes escenarios de consulta, es decir, situaciónes en las que el algoritmo decide qué información solicita al oráculo, y nos enfocaremos en los criterios más utilizados para realizar esa decisión.

### 2.2.1. Síntesis de consulta de pertenencia

Históricamente este fue el primer escenario que fue estudiado en aprendizaje activo [2]. En esta configuración el algoritmo puede solicitar etiquetas para cualquier instancia del espacio de entrada incluyendo patrones que no se encuentran en el conjunto de datos no etiquetado, en lugar de instancias la distribución natural subyacente. La eficiencia de la síntesis de consultas es tratable y eficiente para problemas de dominio finito. La idea de sintetizar consultas ha sido también extendida a tareas de regresión logística, como aprender a predecir las coordenadas absolutas de una mano robot dados los ángulos de las articulaciones de su brazo mecánico como entrada [3].

La limitación principal de este escenario de consulta es que puede solicitar etiquetas para instancias que no tienen ningún sentido en el problema en concreto que se pretende resolver. Por ejemplo, en problemas de clasificación de caracteres escritos a mano este algoritmo genera muchas consultas que incluyen símbolos con caracteres sin ningún significado [4].

### 2.2.2. Muestreo selectivo

En este escenario al algoritmo se le presentan las nuevas instancias de una en una, y este decide para cada una de ellas si es informativa (por tanto, si debe ser etiquetada) o no. Este proceso se repite hasta completar el número de etiquetas permitido en cada iteración [5].

En este algoritmo se asume que la obtención de una instancia no etiquetada no es costosa. De este modo, en primer lugar podremos obtener muestras de la distribución real, y posteriormente el algoritmo podrá decidir si solicita o no la etiqueta. Si la entrada de datos es uniforme, el comportamiento del muestreo selectivo será igual que el aprendizaje de consulta de pertenencia (apartado anterior), pero si la distribución no es uniforme, y lo más importante, desconocida, tenemos la garantía de que las consultas seguirán teniendo sentido, ya que proceden de una distribución subyacente.

La decisión de si el algoritmo quiere consultar o no una instancia se puede abordar de diferentes maneras. Una de ellas es evaluar cada instancia utilizando una medida de la cantidad de información que proporciona, por lo que cada muestra tendrá un valor. La selección sobre este conjunto de muestras es una decisión aleatoria donde cada una de ellas tendrá una probabilidad proporcional al valor informativo de cada una de ellas. Por tanto será más probable seleccionar las consultas más informativas [6].

Otra idea consiste en calcular una región de incertidumbre explícita. Es decir, imponer una condición sobre la cantidad de información proporcionada por cada muestra, y que todas las que cumplan dicha condición sean consultadas [7]. Por ejemplo, si medimos en bits la cantidad de información de cada instancia (por ejemplo, la incertidumbre en su clase que tiene el clasificador), una región de incertidumbre explícita puede ser la formada por todas las instancias cuya información sea mayor de 0.8 bits.

Otro posible enfoque basado en principios consiste en definir una región todavía desconocida para la clase general de modelo [8]. Si dos muestras tienen el mismo modelo, pero diferentes parámetros, y comparten todos los datos etiquetados menos en alguna instancia no etiquetada, entonces esta instancia se encontrará en la región de incertidumbre. El coste de este cálculo es elevado, por lo que debe mantenerse después de cada nueva consulta.

Algunos ejemplos donde este tipo de muestreo ha sido utilizado son el etiquetado gramatical [6], la programación de sensores [9], el aprendizaje de las funciones de recuperación de información de visitantes [10], y la desambiguación lingüística [11].

### 2.2.3. Muestreo basado en conjunto

El muestreo se realiza de forma que de un conjunto muy grande de K muestras potenciales de entrenamiento (el conjunto de datos), el algoritmo selecciona N (secuencialmente) [12].

En este caso, suponemos tener un pequeño conjunto de datos y un gran conjunto de datos de datos sin etiquetar, que normalmente es cerrado y estático (esto no es estrictamente necesario). Las consultas se seleccionan de acuerdo a una medida que cuantifica el nivel informativo de cada instancia, seleccionando aquellas que son más informativas, o en caso de ser demasiadas, un subconjunto de ellas.

La principal diferencia entre el muestreo basado en conjunto y el muestreo selectivo, mencionado en el apartado anterior, es que en el caso de muestreo selectivo se realizan pruebas a través de secuencias y se toman decisiones de forma individual para cada una de las instancias. Por el contrario, en el muestreo basado en conjunto se evalúa y califica el subconjunto de N instancias antes de decidir la mejor consulta. Aunque el más común es el muestreo en conjunto, hay escenarios donde es mejor utilizar muestreo selectivo, por ejemplo, cuando la memoria o procesamiento sea limitada.

## 2.3. Marcos estratégicos de consulta

En este apartado explicaremos los distintos tipos de algoritmos existentes para determinar qué datos se deben consultar.

### 2.3.1. Muestreo basado en incertidumbre

En esta estrategia, la más simple y utilizada, el algoritmo solicita en la consulta las etiquetas de aquellos puntos donde el modelo actual tiene menos seguridad sobre la etiqueta correcta, por ejemplo, los puntos más cercanos a la frontera de decisión. [12].

Podemos considerar tres estrategias posibles: puntos de menor confianza posible, puntos con menor margen de decisión, o puntos con mayor incertidumbre, utilizando medidas como la entropía para calcular dicha incertidumbre para el clasificador.

Este enfoque es a menudo sencillo para los modelos probabilísticos de aprendizaje. El funcionamiento de esta estrategia en problemas de clasificación binaria consultará instancias donde la probabilidad posterior sea muy cercana a 0.5. Este tipo de estrategia es muy frecuente en aprendizaje activo. Un ejemplo de su uso son los modelos de secuencia estadística sobre tareas de extracción de información [13].

### 2.3.2. Consulta por comité

En este marco de consulta se entrenan varios modelos con las etiquetas disponibles y se vota sobre la salida de los datos no etiquetados. La consulta más informativa se considera que es la instancia en la cual los votantes están en mayor desacuerdo [14].

Para la implementación de un algoritmo que cumpla esta estrategia de consulta se requiere, en primer lugar, ser capaz de construir un comité de los modelos que representen las diferentes regiones del conjunto de datos. En segundo lugar, debe haber un cierto grado de desacuerdo entre los miembros que forman los comités creados ya que, en otro caso, la estrategia no sería capaz de satisfacer con éxito el objetivo propuesto.

### 2.3.3. Cambio esperado del modelo

Esta estrategia etiqueta los puntos que más podrían alterar el modelo actual [15]. Es decir, la idea es seleccionar los puntos no etiquetados más influyentes para el modelo, independientemente del resultado obtenido al etiquetar las instancias. Cabe destacar que este enfoque se ha probado que funciona correctamente, salvo en algunos casos donde puede ser computacionalmente costoso si el conjunto de datos es muy grande [16].

### 2.3.4. Reducción del error esperado

Roy y McCallumy fueron los primeros en proponer este marco de consulta para clasificación de textos mediante Naïve Bayes [17]. En la estrategia de reducción del error esperado se etiquetan los puntos que más podrían reducir el error de generalización. El algoritmo planteado por Roy-McCallum sigue los siguientes pasos:

- Entrenar a un clasificador utilizando los ejemplos etiquetados actuales.
- Considerar cada ejemplo no etiquetado X como un posible candidato a consultar.
- Considerar una etiqueta posible para cada candidato X y añadirlo al conjunto de entrenamiento.
    - Volver a entrenar con el conjunto de entrenamiento ampliado.
    - Estimar la pérdida esperada resultante del fallo para cada X.
    - Asignar a cada uno de los posibles candidatos X una medida de error.
- Seleccionar el candidato que genera menor error esperado en todos los demás ejemplos.

En detalle, el algoritmo realiza su clasificación con los datos de entrenamiento etiquetados, y en el cálculo de test, para cada uno de los ejemplos del conjunto de entrenamiento no etiquetados, lo añade al conjunto de etiquetas junto con la etiqueta que predice para dicho dato. Teniendo el conjunto nuevo, se reentrena sobre este nuevo conjunto, y se evalúa este resultado frente al obtenido sin el dato añadido. Esta comparación nos da como resultado una estimación del error para esta instancia del conjunto de entrenamiento de test.

Calcularemos este valor para cada dato de test, y finalmente seleccionaremos los datos que produzcan menor error esperado.

Este método es costoso computacionalmente, ya que para cada instancia de test se requiere un reentrenamiento para calcular el error de dicho dato.

### 2.3.5. Reducción de la varianza

Como se ha visto en el apartado anterior, obtener el error esperado de forma directa es muy costoso. Para evitar ese coste, podemos estimar de forma indirecta este valor. Esta forma es concretamente minimizando la varianza, por lo que se solicitarán los puntos que más podrían minimizar la varianza de salida [18].

La ventaja sobre la reducción de error esperado del apartado anterior (2.3.4), es que el modelo no debe reentrenarse, evitando el coste tan alto que obteníamos en la estrategia anterior.

Sin embargo, estos métodos son empíricamente mucho más lentos que las estrategias de consulta más simples como el muestreo basado en incertidumbre.

### 2.3.6. Métodos de densidad ponderada

La idea principal en la que se basa esta estrategia de consulta es que las instancias más informativas no son solo aquellas en las que el clasificador tiene más incertidumbre, sino también las instancias que son más representativas de la distribución subyacente [19].

Tal y como se ilustra en la imagen, podemos entender mejor como funciona el método de consulta mediante densidad ponderada. En este caso, el punto A es claramente el punto de mayor incertidumbre. Sin embargo, en este caso ese no sería el punto de consulta, ya que no es lo suficientemente representativo de la distribución presentada. En cambio, el punto B, sin ser el más cercano a la frontera de decisión, sí que es más informativo que A.

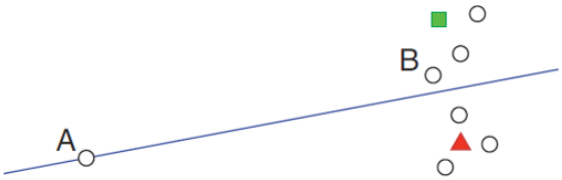


Figura 2.2: Ilustración del método densidad ponderada. [ http://image.slidesharecdn.com/]

Un ejemplo más claro es el que llevó a cabo Fujii [11] utilizando una estrategia de consulta de densidad de ponderación para vecinos próximos, la cual selecciona las instancias que son menos similares al conjunto de datos etiquetado y más similares al conjunto de datos no etiquetado.

## 2.4. Criterio de parada

Otro aspecto muy importante en el aprendizaje activo, y del que todavía no hemos hablado, es cuándo poner fin al aprendizaje: hay un punto donde el algoritmo alcanza su çonocimiento"óptimo del problema en cuestión y en caso de seguir solicitando datos, estos datos ya no van a proporcionar información útil al clasificador, sino que incluso van a hacer que pierda valor la información correcta obtenida anteriormente.

Tal y como menciona Olsson [20] hay diferentes estrategias para determinar cuándo ha llegado a su nivel de aprendizaje más alto el algoritmo de aprendizaje activo.

### 2.4.1. Rendimiento absoluto mínimo

El criterio de parada establece un rendimiento mínimo al clasificador, por lo que el clasificador irá evaluando el rendimiento en cada una de las iteraciones, en caso de obtener un rendimiento por debajo del rendimiento absoluto mínimo fijado el algoritmo finalizará. Este rendimiento mínimo se pasa por parámetro al algoritmo antes de empezar el proceso de aprendizaje [21].

### 2.4.2. Máximo rendimiento posible

Por otro lado, y con un enfoque opuesto al criterio de parada anterior, ahora el clasificador tendrá como condición de parada alcanzar un rendimiento determinado. En el momento que se alcance este rendimiento se finalizará el aprendizaje. De la misma forma que en el criterio de rendimiento absoluto mínimo valor de este rendimiento máximo deseado es un parámentro que recibe el clasificador y es conocedor del mismo antes de inicial el proceso de aprendizaje [21].

### 2.4.3. Conjunto de validación

El criterio de parada se puede basar en el comportamiento del modelo en un subconjunto de training llamado conjunto de validación. Dicho subconjunto no se utilizará para entrenar el modelo sino para monitorizar su comportamiento. Así pues, a medida que el proceso de aprendizaje va avanzando, se intenta utilizar la tasa de acierto del modelo en este subconjunto para detectar cuándo ha llegado el clasificador a su punto óptimo de aprendizaje.

### 2.4.4. Basado en la confianza

Sugiere utilizar la confianza del clasificador para determinar cuándo finalizar el algoritmo de aprendizaje. La idea principal propuesta [22] es finalizar el proceso cuando la confianza del clasificador en un conjunto de pruebas externas se mantiene en el mismo nivel o disminuye durante un número de iteraciones consecutivas.

En su informe Vlachos [22] realiza un estudio utilizando un clasificador SVM con un kernel lineal. En dicho estudio opta por medir la confianza de dicho clasificador utilizando el margen de decisión medio de los datos de entrenamiento desconocidos por el clasificador. El margen de decisión muestra la distancia de cada uno de los puntos al hiperplano generado.

### 2.4.5. Otros criterios de parada

Otros criterios de parada que también han sido utilizados proponen la combinación de varias medidas sobre el resultado del clasificador [23]:

- **Máxima confianza**: se basa en una medida de incertidumbre sobre los datos no etiquetados. Para el cálculo de dicha incertidumbre se hará uso de la entropía obtenida sobre los propios datos no etiquetados.
- **Mínimo error esperado**: se basa en la precisión de clasificación de las etiquetas predichas por el clasificador en comparación con las etiquetas proporcionadas por el oráculo (el valor real de la clase del dato).

Zhu y Hovy ampliaron su trabajo y presentaron otras dos estrategias [24]:

- **Incertidumbre general**: esta estrategia es similar a la estrategia basada en máxima confianza, pero en lugar de tomar los casos más informativos en consideración, se calcula utilizando todos los datos restantes en el conjunto de datos no etiquetados.
- **Clasificación basada en el cambio**: se basa en la suposición de que la instancia más informativa es la que provoca que el clasificador cambie la etiqueta predicha. En este caso el proceso de aprendizaje finalizará cuando una etiqueta de un dato del conjunto de datos no etiquetados cambia su etiqueta predicha durante dos iteraciones consecutivas.

# 3
# Metodología

En este apartado se presentan los aspectos más relevantes en cuanto a la metodología realizada para el desarrollo de los algoritmos de comparación y evaluación del aprendizaje activo. Bien es cierto que el trabajo que se ha llevado a cabo durante este Trabajo Fin de Grado no ha sido en gran medida un trabajo de desarrollo y sí un trabajo de investigación y estudio de algoritmos existentes. A pesar de ello debemos tener en cuenta una serie de decisiones tomadas cuidadosamente para el desarrollo de los programas utilizados para obtener y evaluar dicha información.

## 3.1. Visión general

El sistema que se detallará a continuación es un sistema de clasificación para aprendizaje tanto pasivo como activo. La idea principal y sobre la que se ha diseñado el algoritmo es la de poder clasificar tanto activamente como pasivamente una serie de datos y evaluar los resultados de una manera tal que sean directamente comparables. De esta forma, se podrán evaluar los diferentes comportamientos del algoritmo activo, y de la misma forma se podrá comparar el comportamiento activo frente al pasivo.

Un aspecto importante es la elección del clasificador. En nuestro caso el diseño se ha hecho de forma flexible dando opción a seleccionar qué tipo de clasificador se quiere utilizar en cada prueba.

Por lo tanto, la premisa principal sobre la que está implementado el algoritmo es intentar que ambos algoritmos, tanto el de aprendizaje activo como el de aprendizaje pasivo, se encontrasen en igualdad de condiciones. Es decir, proporcionar los mismos datos, tanto para el clasificador activo como para el pasivo. En este aspecto cabe destacar que solo trabajarán con los mismos datos inicialmente, ya que como hemos comentado en la sección 2, el algoritmo activo irá seleccionando los datos que considere más informativos para su clasificación, mientras que la solución por la que se ha optado en el caso del algoritmo pasivo es seleccionar los datos aleatoriamente sobre el conjunto de datos no etiquetados.

## 3.2. Algoritmo general del aprendizaje activo

En este apartado procederemos a explicar un poco más en detalle el código referente al algoritmo general de aprendizaje activo que se ha implementado. El desarrollo del código de este Trabajo Fin de Grado se ha realizado en dos bloques: el primero de ellos, el código correspondiente a la obtención de los resultados, y el otro el código correspondiente a la evaluación y comparación de los resultados obtenidos. En este apartado entraremos a describir el algoritmo correspondiente al primer bloque mencionado. La estructura del código se divide en dos ficheros: uno de ellos contiene el propio algoritmo activo, y el segundo contiene un programa que controla la carga de la base de datos, los parámetros y el funcionamiento del algoritmo. Por último, también se encarga de guardar los resultados obtenidos en ficheros que serán utilizados para evaluar el rendimiento de dicho algoritmo.

Algoritmo (main):

Inicio

1. Cargar datos de la base de datos.
2. Normalizar los datos en caso de necesitarlo.
3. Inicializar variables y parámetros (incluidos el clasificador y el número de pruebas).
4. Dividir el conjunto de datos en datos de entrenamiento y datos de test.
5. Para cada prueba, hasta completar el número de pruebas total:
   *a*) Reordenar los datos aleatoriamente utilizando una semilla dada.
   *b*) Llamada algoritmo activo.
   *c*) Llamada algoritmo pasivo.
   *d*) Guardar los datos obtenidos en estructuras internas.
6. Guardar en un fichero los valores guardados previamente en las estructuras internas.

Fin

Algoritmo general de aprendizaje activo:

Inicio

1. Dividimos los datos de entrenamiento en etiquetados y no etiquetados (conocidos y desconocidos).
2. Para cada una de las iteraciones desde el número de etiquetas inicial al número de etiquetas final, de N en N (siendo N el número de etiquetas que el algoritmo consulta en cada iteración):
   *a*) Entrenar el modelo con los datos de entrenamiento etiquetados.
   *b*) Calcular las entropías de cada instancia de entrenamiento no etiquetada.
   *c*) Seleccionar las N instancias más informativas para el clasificador, que serán las que se consulten en esa iteración. Actualizar el subconjunto etiquetado y el no etiquetado del conjunto de entrenamiento.
   *d*) Calcular y guardar los datos de evaluación del algoritmo.

Fin

## 3.3. Selección de datos a etiquetar

### 3.3.1. Variable por la que hacer el ranking

En esta sección abordaremos el problema que surge en el algoritmo activo sobre cómo decidir qué datos consultar, más concretamente, qué estrategia de consulta utilizar. De todas las comentadas en el estado del arte, las que hemos considerado para realizar el trabajo son la consulta basada en incertidumbre, la consulta basada en confianza, y por último, se ha valorado hacer una ligera modificación a la estrategia de consulta basada en incertidumbre.

- **Consulta basada en incertidumbre**: tal y como se comenta en la sección 2, la consulta basada en incertidumbre consulta las instancias no etiquetadas que el clasificador considera más inciertas. Para medir dicha incertidumbre, nos basaremos en la entropía estimada por el clasificador sobre los datos no etiquetados. Se solicitarán las instancias que mayor entropía tengan asignada.
- **Consulta basada en confianza**: antes de evaluar esta estrategia de consulta, debemos destacar que esta estrategia la hemos llevado a cabo únicamente con el clasificador SVM. En este caso, la consulta se realizará sobre los datos sobre los que el clasificador tenga menos confianza. Para medir esta confianza, calcularemos el margen de decisión de cada punto respecto al hiperplano separador. Se solicitarán las instancias que menor margen tengan, independientemente del signo asignado a cada instancia (estén por encima o por debajo del hiperplano).
- **Modificación de la consulta basada en incertidumbre**: esta idea pretende evitar el sobreajuste que puede surgir obteniendo las instancias más informativas de un bloque. Para evitar este sobreajuste, se modifica la consulta de instancias de tal forma que el bloque de instancias solicitadas en cada iteración se rellena con la mitad de los datos más informativos (los de mayor entropía), y la otra mitad restante se completa con instancias seleccionadas aleatoriamente.

### 3.3.2. Tamaño inicial

El tamaño inicial del conjunto de datos etiquetados de entrenamiento supone un elemento de estudio aparte. El número de estos datos etiquetados inicialmente lo consideraremos bajo o alto en función del número de datos del conjunto de entrenamiento.

Esto supone que, en caso de dar un número de datos etiquetados iniciales muy bajo, el rendimiento inicial del algoritmo será ínfimo ya que el conocimiento del clasificador sobre el problema no es lo suficiente como para poder obtener buenos resultados. Por el contrario, si el número inicial de datos etiquetados es muy alto, estaríamos olvidando una de las premisas principales del algoritmo, ya que partimos de la base de reducir el coste de los datos etiquetados. Además, en este caso, el proceso de aprendizaje se verá limitado pues el rango de acción será menor, ya que al tener un número de datos alto no tardaremos en obtener el número total de datos de entrenamiento.

### 3.3.3. Tamanño del bloque

El tamaño del bloque es otro factor a tener en cuenta a la hora de la evaluación del algoritmo activo. Este valor representa el número de instancias que el algoritmo solicitará en cada una de las iteraciones. Durante este trabajo se ha realizado un estudio para poder concluir en qué afecta este valor al rendimiento del algoritmo.

## 3.4. Clasificadores básicos utilizados

### 3.4.1. k-NN

El clasificador k-NN (k nearest neighbours), también conocido como de vecinos más próximos, es un clasificador supervisado que predice para cada instancia de test la clase mayoritaria entre sus k vecinos más cercanos dentro de los datos de entrenamiento. Es considerado un algoritmo perezoso ("lazy") ya que el clasificador no construye ningún modelo, sino que simplemente guarda las instancias de entrenamiento. Así mismo es un algoritmo no paramétrico, es decir, no hace suposiciones sobre la distribución concreta que siguen los datos.

Uno de los problemas de este algoritmo lo encontramos en la sensibilidad frente a atributos irrelevantes. Además el algoritmo es sensible a la escala de los atributos, ya que tiende a dar más importancia a los atributos con mayor escala. Para evitar estos problemas, las estrategias típicas son la estandarización de los datos (evitando así problemas de escalabilidad) y la selección de variables, eliminando los atributos que sólo aportan ruido. Los parámetros principales de este algoritmo son k y la medida de distancia d. K representa el número de vecinos sobre el que se calcula la clase mayoritaria. Con un valor muy bajo las instancias ruidosas tendrán una gran influencia; por el contrario, un valor muy alto de k hará perder la localidad del clasificador.

La medida de distancia d será la encargada de determinar la proximidad entre dos instancias y, por lo tanto, de seleccionar qué instancias de entrenamiento entran en el conjunto de k influencias de la clase que se pretende determinar. Normalmente la medida utilizada es la distancia Euclídea.

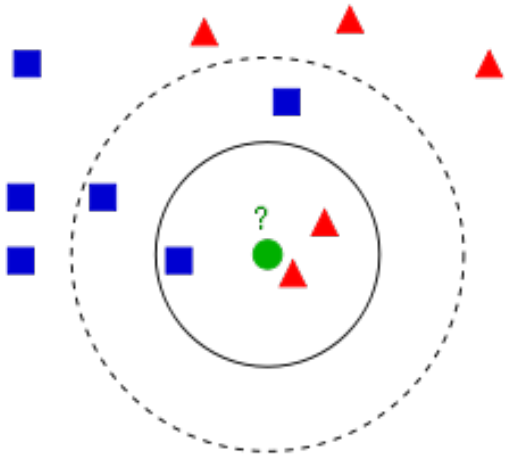


Figura 3.1: Representación k-NN. [https://es.wikipedia.org/wiki/]

En la figura 3.1 se muestra un ejemplo del funcionamiento de la clasificación k-NN. Como se puede observar, en el ejemplo se utiliza k=3 (circunferencia de línea continua, en este caso se asigna a la instancia de test la clase "triángulo rojo") o k=5 (circunferencia de línea discontinua, en este caso se asigna a la instancia de test la clase cuadrado azul).

### 3.4.2. SVM

SVM (Support Vector Machine) es un modelo de clasificación que se basa en transformar el espacio de entrada a otro de dimensión mayor donde sea posible separar linealmente los datos de diferente clase. Por la matemática en la que está basado este método podemos evitar realizar esta transformación explícitamente si se proporciona la función de kernel, que evalúa el

producto escalar de dos instancias en el espacio transformado. Hay diferentes kernels estándar que representan la utilización de diferentes espacios transformados: por ejemplo el kernel lineal, el kernel polinómico, o el kernel de funciones de base radial (RBF). En este trabajo se hicieron pruebas iniciales con distintos kernels, observándose que el lineal era superior a los otros en las bases de datos utilizadas. Por tanto, es el kernel con el que se trabajaeá aquí cuando hablemos de SVMs.

Como hemos dicho anteriormente, las SVMs construyen un separador lineal en el espacio transformado. Este separador lleva asociado un margen, que para problemas linealmente separables no es más que la menor de las distancias de las instancias al hiperplano. Las SVMs tratan de construir este separador lineal maximizando su margen lo máximo posible, ya que cuanto mayor sea, más seguridad tendremos sobre las predicciones realizadas por el clasificador [25].

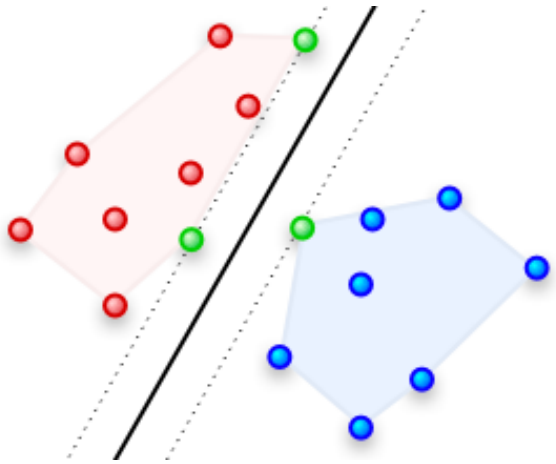

Figura 3.2: Representación SVM. [ http://www.m8j.net/ ]

Debemos destacar que muchas veces el clasificador no será capaz de dividir todos los datos por clase completamente. Esto quiere decir que habrá errores. En este punto el clasificador deberá considerar la importancia del error obtenido en detrimento del margen que produzca. El parámetro que calibra este compromiso es el llamado C.

### 3.4.3. Árboles de decisión

Un árbol de decisión es un clasificador que va construyendo una estructura en forma de árbol. El árbol está formado por nodos interiores, nodos terminales denominados hojas, y ramas. Cada nodo interior del árbol corresponde a una pregunta sobre un atributo, por ejemplo una pregunta lógica sobre un posible valor del atributo de la instancia. Las hojas son nodos etiquetados con una clase o una probabilidad estimada a cada una de las clases. Esto se produce si la profundidad del árbol no es lo suficientemente alta como para determinar la clase con una certeza absoluta. Cuando el clasificador recibe una instancia del conjunto de entrenamiento utilizará el árbol como medio de clasificación, hasta llegar a un nodo hoja que determine la clase que se asignará a esa instancia.

### 3.4.4. Random forest

Este algoritmo nace como una mejora sustancial de los árboles de clasificación simples. Su funcionamiento consiste en combinar una serie de árboles de decisión independientes construidos sobre diferentes conjuntos de datos aleatorios con igual distribución. [26] El algoritmo inicial presentado por Tin Kam Ho fue extendido posteriormente por Breiman [27].

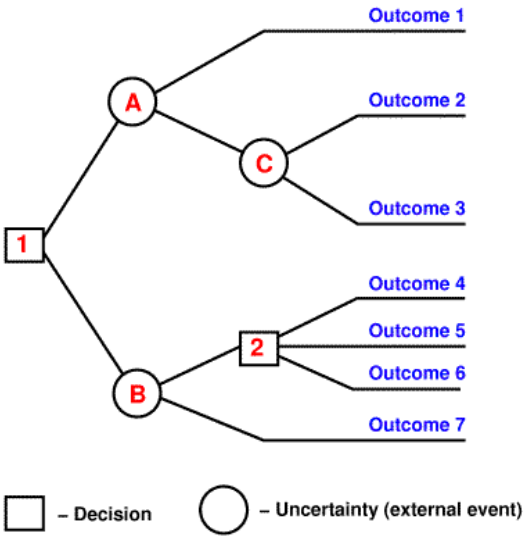


Figura 3.3: Representación de un árbol de decisión [ http://www.time-management-guide.com ]

Durante la fase de entrenamiento se construyen todos los árboles de decisión independientes que formarán el bosque, construyéndolos a partir de datos de entrada ligeramente diferentes. Esto se consigue con el factor de aleatoriedad. Para ello se selecciona aleatoriamente con reemplazamiento un porcentaje de los datos de entrenamiento.

Finalmente, a la hora de obtener el resultado de la clasificación se realiza una evaluación de la instancia a clasificar en cada uno de los árboles, obteniendo así las diferentes probabilidades de cada clase. La probabilidad de cada clase se obtiene calculando la proporción de árboles dentro del bosque que obtienen ese resultado.

Algunas de sus ventajas son la eficiencia en el tratamiento de una alta cantidad de datos, y que es uno de los algoritmos de aprendizaje más certeros para un conjunto de datos lo suficientemente grande.

Por el contrario, se han encontrado algunas desventajas que hacen referencia a un cierto sobreajuste en problemas de clasificación con datos ruidosos [28].

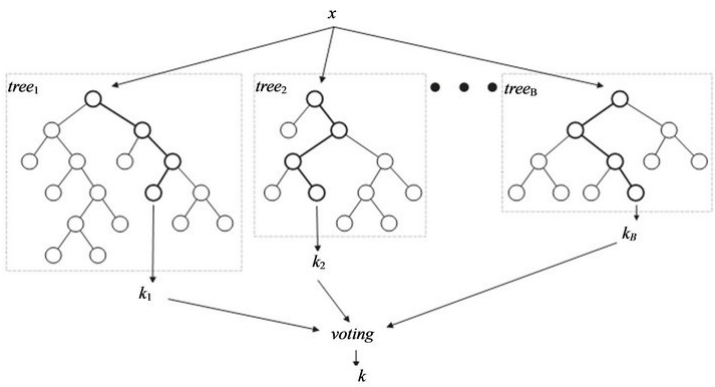


Figura 3.4: Representación Random Forest. [ http://file.scirp.org ]

## 3.5. Método de parada

En esta sección hablaremos de los métodos de parada utilizados y estudiados durante el trabajo. Uno de estos métodos de parada estudiado ha sido el que utiliza un subconjunto de validación. En primer lugar debemos tener clara la división que hemos realizado sobre los datos. Sobre el conjunto total de datos hemos realizado una primera división obteniendo dos subconjuntos, uno con una cantidad de datos superior que será el conjunto de entrenamiento y otro subconjunto que contendrá los datos utilizados para test. Esta subdivisión se ha realizado con porcentajes de datos en torno al setenta-treinta. Debemos destacar que el clasificador solo utilizará el conjunto de datos de test resultante para estimar el error de generalización del clasificador una vez construido el modelo con los datos de entrenamiento. Del conjunto de entrenamiento sacaremos el conjunto de datos que ahora sí serán utilizados para entrenar el clasificador, y otro conjunto de datos que llamaremos conjunto de validación que lo utilizaremos con el fin de parar el aprendizaje en un punto óptimo.

Una vez hemos aclarado todos los subconjuntos de datos, a la hora de ejecutar todo el algoritmo de aprendizaje iremos guardando la información correspondiente al rendimiento obtenido en cada iteración por el clasificador sobre el conjunto de validación y el conjunto de test.

Finalmente cuando el proceso de aprendizaje termina y tenemos todos los datos guardados, realizaremos el estudio y evaluación del criterio de parada. Concretamente este criterio de parada se centra en la evaluación del rendimiento en cada iteración. Se obtendrá el punto de parada cuando tras un número de iteraciones el rendimiento en el conjunto de validación es igual o peor a dicho punto.

Otro método de parada también estudiado emplea la misma idea que el método de validación mentado, pero utilizará en lugar del rendimiento sobre un conjunto de datos de validación, un conjunto que recogerá el margen medio de los puntos no etiquetados en cada iteración. Este margen medio nos dará con resultado la confianza del clasificador sobre los datos desconocidos del problema. También se ha estudiado el momento de parada en el punto más alto de rendimiento en validación.

## 3.6. Evaluación del clasificador

Los algoritmos de aprendizaje activo en general son evaluados mediante la construcción de curvas de aprendizaje, las cuales representan la medida de evaluación de interés (como la precisión) como una función del número de nuevas instancias etiquetadas y añadidas al conjunto. Podemos concluir que el algoritmo de aprendizaje activo es superior a otra aproximación (en este caso, línea base como pasivo tradicional supervisado aprendizaje), si se domina a la otra en la mayoría de los puntos a lo largo de la curva de aprendizaje.

Se va a evaluar el rendimiento sobre el conjunto de test en cada iteración y se van a realizar promedios para calcular medias y desviaciones estándar con el fin de representar las curvas de aprendizaje promedio.

También se han comparado mediante la medida de Wilcoxon las distribuciones de los dos tipos de algoritmos para comprobar que ambos siguen distribuciones diferentes y comprobar que no son fruto de la aleatoriedad.

Para comprobar el comportamiento del aprendizaje activo se ha evaluado el porcentaje de datos que se han necesitado hasta cumplirse la condición de parada y se ha comparado el rendimiento obtenido en ese punto frente al rendimiento resultante con el total de datos de entrenamiento.

## 3.7. Software utilizado

El desarrollo de los algoritmos de evaluación de aprendizaje activo utilizados en este trabajo se ha realizado en el lenguaje de programación Python (versión 2.7). Se ha utilizado la distribución Anaconda 2-4.0 del mismo. Con el fin de utilizar los algoritmos básicos citados en el apartado 3.1.3, se ha usado la librería de aprendizaje automático scikit-learn para Python [29].

## 3.8. Bases de datos utilizadas

Las bases de datos que se han utilizado para evaluar los resultados son las siguientes:

### 3.8.1. PIMA

Esta base de datos se corresponde a una colección de diagnósticos médicos donada por Vincent Sigillito. Contiene informes de 768 mujeres mayores de veintiún años [30] y cada una de las instancias contiene un total de ocho atributos, más la clase asignada a dicha instancia. Estos atributos son:

- Número de veces que ha estado embarazada.
- Concentración de glucosa.
- Presión sanguínea.
- Espesor de pliegue del tríceps.
- Insulina en suero.
- Función de diabetes.
- Edad.

El problema de esta base de datos es binario, ya que la clase toma únicamente los valores 1 y 0 (representando si la mujer que representa sufre diabetes o no).

### 3.8.2. Dígitos

Esta base de datos corresponde a una colección de 1797 imágenes. Cada una de estas imágenes corresponde con la representación de un dígito escrito a mano. Con el fin de poder representar cada dígito como una imagen, cada una de ellas contiene 64 atributos. Cada uno de ellos se corresponden con uno de los píxeles que forman una imágen de 8x8. Cada píxel obtiene un valor entre 0 y 100 en una escala de grises donde el valor 0 corresponde a un píxel totalmente blanco, y 100 a un píxel totalmente negro.

El problema de esta base de datos es un problema multiclase donde cada imágen puede representar cualquier dígito de 0 a 9, teniendo en total 10 clases posibles. [30].

### 3.8.3. Incidencias

La base de datos de incidencias, proporcionada por la empresa Cognodata Consulting, consta de 57735 registros, cada uno descrito por un texto con la descripción proporcionada por el cliente u operador que la ha recogido. El problema consiste en predecir diferentes aspectos de la incidencia tales como el motivo de la incidencia (por ejemplo si es debido a un cargo indebido, o a un servicio no satisfactorio dado por la empresa, etc.), canal a través del cual se ha recogido la incidencia (atención telefónica, email, etc.) o producto sobre el que trata la incidencia.

La empresa nos proporciona la información de los textos ya codificada. Para ello ha eliminado las stop words y ha hecho una selección de las palabras que desde el punto de vista del negocio son más relevantes, quedándose con las 728 más representativas. La información de los textos que nos dan viene en formato BOW, donde hay tantas columnas como palabras en el vocabulario, y la aparición o no de las palabras está codificada como 0/1 (1 si aparece la palabra en algún lugar del texto, 0 si no aparece).

# 4

# Experimentos Realizados y Resultados

En esta sección se describirán en detalle los resultados obtenidos utilizando la metodología explicada en la sección anterior. Primero se mostrará el funcionamiento general del algoritmo y se continuará estudiando el efecto de los diferentes parámetros del algoritmo activo. Se realizará un estudio comparativo entre dos técnicas de consulta tales como la técnica de consulta basada en confianza y de consulta basada en incertidumbre, y finalmente se estudiará el efecto en la eficiencia del algoritmo de los diferentes criterios de parada. En todas las pruebas se contrastarán los resultados obtenidos con diferentes bases de datos y con los diferentes clasificadores utilizados.

## 4.1. Funcionamiento general del algoritmo

A continuación, con el fin de obtener una primera idea del comportamiento general del algoritmo que se va a estudiar, valoraremos un caso donde se han escogido los siguientes valores en los parámetros del algoritmo: 25 % de datos inicialmente etiquetados, con un incremento del 1 % por iteración (ambos porcentajes son sobre el total de datos de entrenamiento disponibles). En este ejemplo se utiliza el clasificador de vecinos próximos (kNN) con k=7, y la base de datos PIMA [30].

En la figura 4.1 se representa el score promedio en test utilizando los parámetros anteriormente citados. Se representa en rojo el comportamiento del algoritmo activo, frente al color azul que representa el algoritmo pasivo. Así mismo se representa la desviación obtenida tanto para el rendimiento activo como el rendimiento pasivo. Como se puede observar el rendimiento del algoritmo activo y el pasivo es inicialmente el mismo, lo cual tiene sentido, ya que el número inicial de datos etiquetados es el mismo. Partiendo de este punto, observamos cómo la curva de score del algoritmo activo está por encima del pasivo hasta los trescientos datos de entrenamiento. En este punto el algoritmo activo parece reflejar un sobreajuste y reduce su tasa de acierto en test mientras que el algoritmo pasivo continua incrementado su score linealmente hasta finalizar igualando la tasa de acierto del pasivo. Ambos algoritmos finalizan con el mismo score ya que en ese punto ambos utilizan las etiquetas de todos los datos de entrenamiento.

Si observamos el score máximo obtenido por cada uno de los algoritmos encontramos un rendimiento muy similar, siendo ligeramente inferior el pasivo. Sin embargo, es destacable que el número de datos etiquetados utilizados por el algoritmo activo cuando alcanza su score máximo

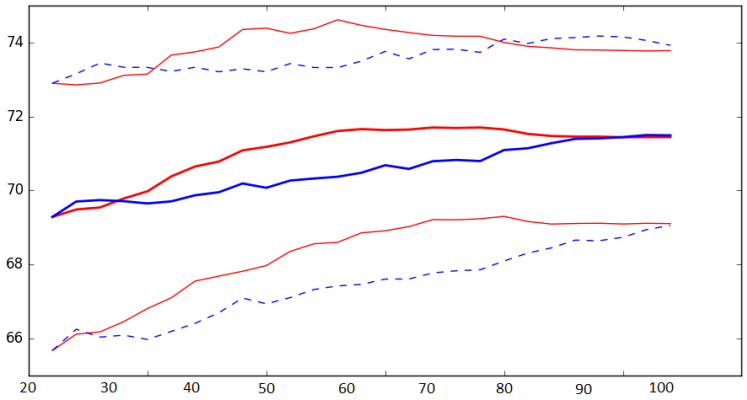


Figura 4.1: Rojo: promedio y desviación estándar del score en test del algoritmo activo a lo largo de 100 repeticiones frente al porcentaje de datos utilizados de entrenamiento. Azul: mismos datos para el algoritmo pasivo.

es considerablemente inferior al número de datos etiquetados utilizados por el algoritmo pasivo en el punto con el mismo score.

También hemos realizado pruebas estadísticas sobre estos datos con el fin de asegurarnos de que los comportamientos del algoritmo activo y del pasivo son cualitativamente diferentes. Para ello hemos realizado el test de Wilcoxon [31]. Este test es una prueba no paramétrica que compara dos muestras seleccionadas y determina si hay diferencias entre ellas. No es necesario una distribución específica y requiere un nivel ordinal en la variable. Pretende determinar que dos mediciones son diferentes por motivos estadísticos y no se deben al azar. En nuestro caso, chequearemos con el test de Wilcoxon si el hecho de que el método activo tenga mejor score o no que el pasivo es fruto del azar o no.

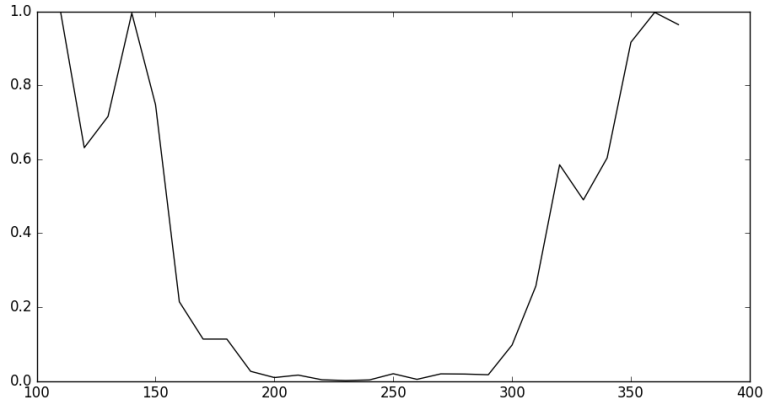


Figura 4.2: Probabilidad de la hipótesis nula del test de Wilcoxon aplicado en cada uno de los volúmenes de training estudiados.

En la figura 4.2 se refleja el resultado obtenido del test de Wilcoxon mediante la representación de la probabilidad de la hipótesis nula. La hipótesis nula está asociada a que el aprendizaje activo tenga un comportamiento exactamente igual que el pasivo salvo fluctuaciones, por lo que, tal y como se refleja en la imagen, inicialmente es alta pero luego tiende a ser cero, corroborando que los scores del aprendizaje activo son estadísticamente diferentes a los del pasivo. En el último tramo de la evaluación se vuelven a asemejar ya que en ese momento tienden a utilizar los mismos datos etiquetados.

## 4.2. Número inicial de datos

A continuación se mostrarán los resultados obtenidos sobre el estudio de la cantidad de datos iniciales del algoritmo. Se plantean como valores de estudio los porcentajes de datos de entrenamiento de 5, 12.5 y 25. Los resultados se estructuran en diferentes tablas donde se exponen los valores utilizados en cada una de las pruebas y se muestra el porcentaje de datos utilizados en el punto de parada (*Punto de parada*), el rendimiento en ese punto (*Score parada*) y el rendimiento máximo durante todo el aprendizaje (*Score max*). Los resultados se realizaron sobre cien pruebas con el objetivo de resultados lo más fiables posibles. Al mostrar los resultados promedio añadimos a cada valor porcentual la desviación estándar.

Hemos seleccionado el criterio de parada por score en validación, y un número fijo de tamaño de bloque para todos los casos igual al 1 %. Así mismo el método de consulta utilizado es el basado en incertidumbre.

| Experimiento | | Resultados (%) | | |
|---|---|---|---|---|
| Clasificador | BBDD | Punto de parada | Score parada | Score max |
| kNN | Digits | 41.3 ± 8.6 | 97.1 ± 1.0 | 97.6 ± 0.6 |
| SVM | Digits | 31.8 ± 6.9 | 97.0 ± 0.9 | 97.5 ± 0.6 |
| DTree | Digits | 35.3 ± 13.6 | 66.4 ± 3.9 | 69.0 ± 3.4 |
| RF | Digits | 37.5 ± 6.6 | 96.9 ± 0.8 | 96.8 ± 0.8 |
| kNN | PIMA | 15.0 ± 9.4 | 69.3 ± 3.4 | 71.5 ± 2.4 |
| SVM | PIMA | 19.4 ± 9.6 | 75.4 ± 3.2 | 77.1 ± 2.2 |
| DTree | PIMA | 17.5 ± 8.1 | 72.6 ± 3.6 | 73.3 ± 3.1 |
| RF | PIMA | 18.5 ± 9.9 | 73.5 ± 3.0 | 74.5 ± 2.4 |

Cuadro 4.1: Resultados del estudio con un volumen de datos etiquetados iniciales del 5 % sobre training.

| Experimiento | | Resultados (%) | | |
|---|---|---|---|---|
| Clasificador | BBDD | Punto de parada | Score parada | Score max |
| kNN | Digits | 40.5 ± 8.5 | 96.7 ± 0.8 | 98.0 ± 0.2 |
| SVM | Digits | 30.1 ± 3.6 | 96.9 ± 3.6 | 96.4 ± 0.0 |
| DTree | Digits | 34.8 ± 12.8 | 67.0 ± 4.2 | 69.3 ± 0.6 |
| RF | Digits | 40.3 ± 7.9 | 97.1 ± 0.6 | 97.1 ± 0.3 |
| kNN | PIMA | 23.6 ± 10.8 | 70.1 ± 3.4 | 71.4 ± 2.3 |
| SVM | PIMA | 25.0 ± 8.8 | 76.0 ± 2.4 | 77.1 ± 2.2 |
| DTree | PIMA | 21.4 ± 7.9 | 72.3 ± 3.3 | 73.5 ± 3.2 |
| RF | PIMA | 22.9 ± 8.4 | 73.7 ± 2.8 | 74.5 ± 2.4 |

Cuadro 4.2: Resultados del estudio con un volumen de datos etiquetados iniciales del 12.5 % sobre training.

En este caso, debemos saber que idealmente el conjunto de datos inicial no debería ser muy grande ya que esto es precisamente lo que pretendemos evitar. Por lo que a la hora de evaluar los resultados representados en las tablas anteriores deberemos tener en cuenta que en nuestro caso ideal el número de datos con el que deberíamos iniciar nuestro aprendizaje debe de tener un valor bajo.

En los resultados mostrados destaca negativamente la tabla 4.3, dicha tabla representa un tamaño inicial de datos del 25 %, por lo que el porcentaje de datos utilizados en este caso es

| Experimiento | | Resultados (%) | | |
|---|---|---|---|---|
| Clasificador | BBDD | Punto de parada | Score parada | Score max |
| kNN | Digits | 47.9 ± 8.0 | 97.2 ± 0.9 | 97.6 ± 0.6 |
| SVM | Digits | 42.9 ± 6.8 | 97.1 ± 0.8 | 97.5 ± 0.6 |
| DTree | Digits | 40.9 ± 11.6 | 67.7 ± 3.3 | 68.7 ± 3.3 |
| RF | Digits | 49.8 ± 7.2 | 97.0 ± 0.7 | 96.7 ± 0.7 |
| kNN | PIMA | 33.8 ± 7.0 | 70.9 ± 2.3 | 72.9 ± 0.5 |
| SVM | PIMA | 34.8 ± 6.5 | 74.3 ± 1.6 | 76.2 ± 0.7 |
| DTree | PIMA | 37.4 ± 9.5 | 72.4 ± 2.4 | 74.8 ± 0.8 |
| RF | PIMA | 36.1 ± 7.4 | 72.6 ± 1.8 | 72.6 ± 0.9 |

Cuadro 4.3: Resultados del estudio con un volumen de datos etiquetados iniciales del 25 % sobre training.

mayor que en las otras dos tablas anteriores. A pesar de que el score de test en el momento de parada es bueno, el porcentaje de datos inicial ya supone un alto coste que se suma al coste de datos hasta cumplir el criterio de parada. Por tanto este caso no es satisfactorio para nuestro objetivo.

Evaluando el rendimiento entre las dos tables restantes, 4.2 y 4.1 observamos que el porcentaje de datos utilizados hasta satisfacer el criterio de parada es muy similar, siendo ligeramente inferior el porcentaje utilizado en el caso con un porcentaje inicial del 12.5 %. Respecto al rendimiento en ambos casos se acerca bastante al obtenido con el conjunto de datos de entrenamiento completo.

## 4.3. Número de etiquetas por bloque

En este apartado se muestran los resultados obtenidos sobre el estudio de la cantidad de datos solicitados por el algoritmo en cada iteración. Se plantean como valores de estudio los porcentajes de datos de 0.1 %, 1 % y 5 %. Los resultados se estructuran en diferentes tablas donde se muestran los valores estudiados en cada caso.

En las tablas se muestran los siguientes datos: el punto de parada (*Punto de parada*), el rendimiento en ese punto (*Score parada*) y el rendimiento máximo durante todo el aprendizaje (*Score max*). Al tratarse de nuevo de valores promedio volvemos a añadir los valores de la desviación estándar en cada valor. Hemos seleccionado el criterio de parada de validación y un número fijo de tamaño de datos inicial para todos los casos igual, que se corresponde con un 12.5 %. El método de consulta utilizado es consulta basada en incertidumbre.

Los resultados en estos tres posibles escenarios de tamaño por bloque nos dejan varios aspectos relevantes. En la figura 4.4 podemos observar que el punto de parada seleccionado mediante validación es muy temprano, con valores entre el 15 % de los datos y el 22 %. Esto es un buen síntoma ya que nos estamos ahorrando por lo menos el 75 % de los datos. Sin embargo ese buen resultado en el punto de parada se ve contrarrestado por el rendimiento obtenido en ese punto, quedando lejos del rendimiento obtenido con el total de datos de entrenamiento. Por otro lado, en la figura 4.6 sucede lo contrario, en los resultados obtenidos observamos un score de test en el punto de parada muy próximo al score en el conjunto total de datos de entrenamiento. En este caso el score de test es muy cercano al obtenido utilizando todos los datos de entrenamiento pero la cantidad de datos necesarios para obtener dicho score llga en algunos casos a superar el 50 % en algunos experimentos. Finalmente, en los resultados obtenidos en la figura 4.5 obtenemos el

| Experimiento | | Resultados ( %) | | |
|---|---|---|---|---|
| Clasificador | BBDD | Punto de parada | Score parada | Score max |
| kNN | Digits | 21.8 ± 4.7 | 93.1 ± 2.5 | 98.0 ± 0.2 |
| SVM | Digits | 17.8 ± 3.2 | 94.8 ± 1.8 | 97.4 ± 0.0 |
| DTree | Digits | 15.7 ± 2.8 | 59.1 ± 5.4 | 69.4 ± 5.3 |
| RF | Digits | 19.8 ± 4.1 | 93.7 ± 2.2 | 97.1 ± 0.3 |
| kNN | PIMA | 14.8 ± 2.2 | 68.8 ± 3.5 | 71.4 ± 2.4 |
| SVM | PIMA | 14.8 ± 2.0 | 74.1 ± 3.5 | 77.1 ± 2.2 |
| DTree | PIMA | 14.2 ± 1.6 | 71.3 ± 4.7 | 73.5 ± 3.2 |
| RF | PIMA | 15.1 ± 1.9 | 72.7 ± 3.5 | 74.5 ± 2.4 |

Cuadro 4.4: Resultados del estudio con un volumen tamaño del bloque del 0.1 %.

| Experimiento | | Resultados ( %) | | |
|---|---|---|---|---|
| Clasificador | BBDD | Punto de parada | Score parada | Score max |
| kNN | Digits | 40.5 ± 8.5 % | 96.7 ± 0.7 | 98.0 ± 0.2 |
| SVM | Digits | 30.1 ± 3.6 | 96.9 ± 0.3 | 97.4 ± 0.0 |
| DTree | Digits | 34.8 ± 12.8 | 67.0 ± 4.2 | 69.3 ± 0.6 |
| RF | Digits | 40.3 ± 7.9 | 97.1 ± 0.6 | 97.1 ± 0.3 |
| kNN | PIMA | 23.6 ± 10.8 | 70.1 ± 3.3 | 71.4 ± 2.3 |
| SVM | PIMA | 25.0 ± 8.8 | 76.0 ± 2.4 | 77.1 ± 2.2 |
| DTree | PIMA | 21.4 ± 7.9 | 72.3 ± 3.3 | 73.5 ± 3.2 |
| RF | PIMA | 22.9 ± 8.35 | 73.7 ± 2.8 | 74.5 ± 2.4 |

Cuadro 4.5: Resultados del estudio con un volumen tamaño del bloque del 1 %.

punto medio entre las dos situaciones anteriores. En este caso encontramos una estabilidad entre el punto de parada seleccionado y el rendimiento obtenido en dicho punto. Para la base de datos PIMA el porcentaje de datos utilizados ronda el 20-25 % mientras que en la base de datos Digits este valor es entorno al 35 %. El rendimiento en el punto de parada obtenido es generalmente muy próximo al rendimiento obtenido utilizando el conjunto de entrenamiento completo.

## 4.4. Estudio de los diferentes métodos de consulta

En esta sección estudiaremos el comportamiento de la consulta basada en incertidumbre y la consulta basada en confianza. Para ello se utilizará el clasificador SVM. También evaluaremos el comportamiento de consulta dividiendo los datos de consulta a partes iguales, entre los de mayor incertidumbre y otra serie de datos aleatorios hasta completar los datos requeridos por iteración, este método en las tablas que se ilustrarán a continuación se denomina *Mixto*.

Se muestran los resultados de la comparativa entre la consulta basada en incertidumbre y la modificación citada. Hemos realizado las pruebas con el método de parada de validación, tamaño inicial de 12.5 % y tamaño del bloque 1 %.

En esta tabla queda reflejada la comparativa entre el método de consulta basado en incertidumbre y la variación citada de elegir únicamente la mitad de los datos inciertos y completar los restantes con datos aleatorios.

Los datos reflejan que en porcentaje de datos utilizados previos a la parada del aprendizaje es menor en caso del método de consulta basado en incertidumbre, así mismo el porcentaje en ese

| Experimiento | | Resultados ( %) | | |
|---|---|---|---|---|
| Clasificador | BBDD | Punto de parada | Score parada | Score max |
| kNN | Digits | 61.8 ± 14.4 % | 97.6 ± 0.4 | 98.0 ± 0.2 |
| SVM | Digits | 38.5 ± 7.7 | 97.2 ± 0.3 | 97.4 ± 0.0 |
| DTree | Digits | 77.4 ± 14.3 | 69.9 ± 2.2 | 69.4 ± 1.4 |
| RF | Digits | 66.9 ± 14.9 | 97.3 ± 0.4 | 97.2 ± 0.3 |
| kNN | PIMA | 47.2 ± 21.1 | 71.1 ± 3.1 | 71.5 ± 2.4 |
| SVM | PIMA | 42.4 ± 18.2 | 76.6 ± 2.3 | 77.1 ± 2.2 |
| DTree | PIMA | 49.3 ± 23.4 | 73.1 ± 3.0 | 73.1 ± 3.2 |
| RF | PIMA | 45.4 ± 22.9 | 74.2 ± 2.5 | 74.5 ± 2.6 |

Cuadro 4.6: Resultados del estudio con un volumen tamaño del bloque del 5 %.

| Experimiento | | | Resultados ( %) | | |
|---|---|---|---|---|---|
| Clasificador | BBDD | Consulta | Punto de parada | Score parada | Score max |
| kNN | Digits | Incertidumbre | 40.5 ± 8.5 | 96.7 ± 0.8 | 98.0 ± 0.2 |
| kNN | Digits | Mixto | 52.2 ± 10.9 | 96.6 ± 1.3 | 97.6 ± 0.6 |
| DTree | Digits | Incertidumbre | 34.8 ± 12.8 | 67.0 ± 4.2 | 69.3 ± 0.6 |
| DTree | Digits | Mixto | 27.6 ± 9.5 | 62.0 ± 4.7 | 65.30 ± 3.8 |
| RF | Digits | Incertidumbre | 40.3 ± 7.9 | 97.1 ± 0.6 | 97.1 ± 0.3 |
| RF | Digits | Mixto | 52.2 ± 10.9 | 96.6 ± 1.3 | 97.6 ± 0.6 |
| kNN | PIMA | Incertidumbre | 23.6 ± 10.8 | 70.1 ± 3.4 | 71.5 ± 2.3 |
| kNN | PIMA | Mixto | 25.1 ± 13.2 | 69.8 ± 3.6 | 71.5 ± 2.4 |
| DTree | PIMA | Incertidumbre | 21.4 ± 7.9 | 72.3 ± 3.3 | 73.5 ± 3.2 |
| DTree | PIMA | Mixto | 21.6 ± 9.8 | 72.2 ± 3.5 | 73.5 ± 2.9 |
| RF | PIMA | Incertidumbre | 22.9 ± 8.35 | 73.7 ± 2.8 | 74.5 ± 2.4 |
| RF | PIMA | Mixto | 25.2 ± 10.3 | 73.3 ± 3.3 | 74.5 ± 2.5 |

Cuadro 4.7: Comparación método de consulta basado en incertidumbre.

punto es más próximo al porcentaje obtenidos con todos los datos de entrenamiento utilizando este método frente al mixto.

En la próxima tabla se presentan los diferentes métodos de consulta sobre el clasificador SVM (Support Vector Classifier), incluyendo el método de consulta basado en confianza.

En este caso, el método de consulta que menor número de patrones necesita hasta satisfacer el criterio de parada es el basado en incertidumbre seguido por el basado en confianza. En cuanto al score en el punto de parada observamos que se obtiene un resultado bueno acercándose al score cuando se usan todos los datos etiquetados del conjunto de entrenamiento.

Finalmente, podemos concluir que el comportamiento entre los diferentes criterios de consulta es muy similar siendo sutilmente mejor el método basado en incertidumbre.

## 4.5. Criterio de parada

En este apartado se mostrarán los resultados obtenidos de diferentes criterios de parada estudiados. Entre ellos se encuentra el ya utilizado en las pruebas anteriores, el criterio de parada de validación. En este criterio de parada evaluamos el rendimiento del clasificador sobre

| Experimiento | | | Resultados (%) | | |
|---|---|---|---|---|---|
| Clasificador | BBDD | Consulta | Punto de parada | Score parada | Score max |
| SVM | Digits | Incertidumbre | 30.1 ± 3.6 | 96.9 ± 0.4 | 97.5 ± 0.6 |
| SVM | Digits | Mixto | 38.7 ± 12.7 | 96.2 ± 1.6 | 97.5 ± 0.6 |
| SVM | Digits | Confianza | 35.0 ± 18.1 | 94.6 ± 1.9 | 97.5 ± 0.6 |
| SVM | PIMA | Incertidumbre | 25.0 ± 8.8 | 76.0 ± 2.4 | 77.1 ± 2.2 |
| SVM | PIMA | Mixto | 26.2 ± 10.4 | 75.7 ± 2.8 | 77.1 ± 2.1 |
| SVM | PIMA | Confianza | 26.2 ± 9.3 | 76.2 ± 2.8 | 77.1 ± 2.2 |

Cuadro 4.8: Comparación método de consulta del clasificador SVM.

un conjunto de validación en cada iteración: si tras un número de iteraciones determinado el score en validación se ha mantenido igual o menor se satisface la condición de parada.

Otro criterio utilizado es el criterio de parada basado en la confianza del clasificador SVM. En este caso, en vez de utilizar como medida informativa el score en el conjunto de validación se evalúa la confianza media del clasificador SVM que se obtiene sobre los puntos no etiquetados. El sistema de de parada es idéntico al anterior, pero evaluando en cada iteración la confianza obtenida. Otro método implementado utiliza como época de parada la época en la que se ha obtenido el máximo score en el conjunto de validación. Este método se denomina *Validación max* en la tabla.

Al igual que en tablas anteriores, los resultados son los promedios obtenidos tras diferentes pruebas, se añade la desviación estándar a dichos valores. En todas las pruebas realizadas se utilizan bloques de 1 %, y un tamaño inicial del 12.5 % de total de datos etiquetados en training. El criterio de consulta seleccionado para las pruebas es el criterio basado en incertidumbre.

| Experimiento | | | Resultados (%) | | |
|---|---|---|---|---|---|
| Clasificador | BBDD | Criterio | Punto de parada | Score parada | Score max |
| kNN | Digits | Validación | 40.5 ± 8.5 | 96.7 ± 0.8 | 98.00 ± 0.2 |
| kNN | Digits | Validación Max | 61.0 ± 18.1 | 97.4 ± 0.5 | 98.0 ± 0.2 |
| SVM | Digits | Validación | 30.1 ± 3.6 | 96.9 ± 0.4 | 97.5 ± 0.6 |
| SVM | Digits | Confianza | 15.6 ± 3.5 | 92.8 ± 2.5 | 97.5 ± 0.2 |
| SVM | Digits | Validación Max | 31.2 ± 4.4 | 96.9 ± 0.4 | 97.5 ± 0.2 |
| kNN | PIMA | Validación | 23.6 ± 10.8 | 70.1 ± 3.4 | 71.5 ± 2.3 |
| kNN | PIMA | Validación Max | 45.4 ± 20.2 | 71.3 ± 3.0 | 71.5 ± 2.3 |
| SVM | PIMA | Validación | 25.0 ± 8.8 | 76.0 ± 2.4 | 77.1 ± 2.2 |
| SVM | PIMA | Confianza | 14.7 ± 2.9 | 73.4 ± 3.6 | 77.1 ± 2.2 |
| SVM | PIMA | Validación Max | 38.1 ± 20.2 | 76.3 ± 2.2 | 77.1 ± 2.2 |

Cuadro 4.9: Comparación criterio de parada.

Los resultados expuestos en la tabla 4.10 reflejan que el criterio que mayor cantidad de datos etiquetados utiliza hasta cumplir el criterio de parada es el método de validación máxima. Por el contrario el que menor número de datos utiliza hasta cumplir el criterio de parada es el de basado en confianza siendo bastante inferior a los otros dos casos. El método de parada que utiliza la confianza obtiene un score en test en ese punto algo lejano al score en test obtenido al entrenar con todos los datos de entrenamiento etiquetados. Sin embargo, el criterio de parada por validación obtiene un buen score en test en el momento de parada quedando ligeramente inferior a dicho score si se utiliza el total de los datos de training etiquetados.

Nos encontramos en un punto donde en caso de querer ahorrar más datos estaremos perdiendo algo de score en test, mientras que si elegimos el método de validación asumiremos el coste de obtener un porcentaje mayor de datos etiquetados a costa de obtener mayor score.

## 4.6. Resultados obtenidos con la base de datos de incidencias

En esta sección se exponen los resultados obtenidos sobre la base de datos de incidencias proporcionada por la empresa Cognodata Consulting, con el fin de poder corroborar los resultados expuestos en apartados anteriores. Esta base de datos, más completa y con una cantidad de datos mayor respecto a las utilizadas anteriormente representa un problema real, donde en nuestro caso el objetivo será predecir sobre un conjunto de incidencias qué motivo ha causado cada una de ellas.

Se han utilizado los siguientes valores de los parámetros del algoritmo: 1 % de tamaño de bloque, 5 % tamaño inicial. En este caso el criterio de parada viene dado por un número de datos de entrenamiento finito, esta limitación asemeja un posible caso de la vida real donde una empresa sólo tiene presupuesto para etiquetar una serie de patrones. En nuestro caso, esta limitación la estamos poniendo en 1000 patrones, de una base de datos de 57735. Los resultados expuestos a continuación comparan el score obtenido entest por el algoritmo activo en el momento que se alcanzan los 1000 patrones etiquetados, frente al algoritmo pasivo que selecciona esos 1000 patrones aleatoriamente.

De nuevo al obtener valores promedio se añade el valor de la desviación estándar.

| Experimiento | | | Resultados ( %) | |
|---|---|---|---|---|
| Clasificador | BBDD | Criterio | Score activo | Score pasivo |
| kNN | Incidencias | 1000 patrones | 79.5 ± 0.7 | 77.1 ± 0.7 |
| SVM | Incidencias | 1000 patrones | 79.3 ± 0.5 | 77.1 ± 0.7 |
| DTree | Incidencias | 1000 patrones | 72.0 ± 3.2 | 72.6 ± 2.3 |
| RF | Incidencias | 1000 patrones | 80.7 ± 0.7 | 78.9 ± 0.8 |

Cuadro 4.10: Comparación rendimiento activo / pasivo base de datos de incidencias.

Los resultados muestran que para los clasificadores kNN, SVM y Random Forest el algoritmo activo supera en un 2 % de score en test al pasivo. Sin embargo, para el clasificador árbol de decisión no obtenemos tan buen resultado, siendo el score muy parejo en ambos casos.

Los resultados en los tres primeros clasificadores mostrados obtienen un score en test cercano al 80 %, mejorando el valor del rendimiento prior sobre esta base de datos que se encuentra en un 58 %.

Por lo tanto, realizando la prueba sobre un escenario donde la empresa que estudia el problema limita el gasto en el etiquetado de instancias, los resultados evidencian un mejor rendimiento del algoritmo activo.

# 5

# Conclusiones y trabajo futuro

En este Trabajo Fin de Grado se ha realizado un estudio sobre un tipo de algoritmos del ámbito del aprendizaje automático y la inteligencia artificial, el aprendizaje activo.

Se han realizado estudios sobre el comportamiento general del algoritmo activo frente al pasivo, el número de datos etiquetados que conoce inicialmente el algoritmo, el número de nuevos datos etiquetados que solicita el algoritmo en cada iteración, diferentes criterios de parada y diferentes métodos de consulta.

En todos los resultados obtenidos queda reflejada la superioridad del algoritmo activo frente al pasivo, obteniendo un porcentaje de acierto que en general es mayor que obtenido por el equivalente pasivo. Así mismo en las pruebas donde se estudia el criterio de parada, el score en test en la época de parada elegida por el algoritmo se aproxima al score utilizando todo el conjunto de entrenamiento etiquetado.

Los resultados obtenidos respecto al volumen inicial de datos etiquetados reflejan gran similitud entre el comportamiento de las pruebas con un valor inicial de 5 % y de 12.5 %, tanto en porcentaje de datos utilizados como en la calidad del score de test obtenida. El valor de 12.5 % es ligeramente mejor para la base de datos Digits pero por el contrario para la base de datos PIMA el volumen de datos etiquetados utilizados es menor.

En cuanto al tamaño del bloque, los resultados reflejan que la peor opción es la que selecciona un 5 % de los datos en cada iteración. Entre las dos restantes, 0.1 % y 1 % el resultado más favorable es algo discutible. En el caso de elegir bloques de 0.1 % el porcentaje de datos etiquetados utilizados hasta parar es menor, pero el score de test en este punto es algo lejano al obtenido cuando se usa todo el conjunto de entrenamiento etiquetado. Por el contrario, en caso de bloques de 1 % el algoritmo para con un número de datos etiquetados mayor pero la calidad del score de test en este punto es muy cercana a la obtenida cuando se usa todo el conjunto de training etiquetado. Para la elección de uno u otro deberemos estudiar el problema en cuestión con detenimiento, evaluando si podemos asumir un coste adicional en el etiquetado de datos sabiendo que el rendimiento será muy bueno, frente al caso donde no podamos asumir ese coste y nos conformemos con un porcentaje de acierto menor.

Al evaluar diferentes criterios de parada, comparamos la consulta basada en incertidumbre con otra estrategia planteada por nosotros que mezcla en la consulta datos elegidos al azar con datos elegidos por el criterio de incertidumbre. El objetivo de este método era no sobrecargar el

conjunto etiquetado con instancias demasiado complejas. En los resultados obtenidos la consulta basada únicamente en incertidumbre es claramente mejor. Así mismo y utilizando el clasificador SVM, hemos comprobado estos dos criterios y un tercer método de consulta basado en la confianza del clasificador sobre los datos no etiquetados REF. El criterio que menor número de datos utiliza hasta satisfacer el criterio de parada es el basado en incertidumbre con un rendimiento de parada muy bueno. Entre los otros dos métodos se refleja un comportamiento mejor en el basado en confianza.

En las últimas pruebas realizadas se comparan diferentes criterios de parada, entre los que se encuentran el basado en el score en validación, que selecciona la época con el valor máximo en validación, y un criterio que utiliza la confianza media sobre los puntos no etiquetados. El mejor de todos ellos es el criterio de validación, que a pesar de obtener un porcentaje mayor de datos utilizados obtiene un acierto muy alto en el punto de parada. En cuanto al criterio del punto con mayor acierto de validación resalta la cantidad de datos que son necesarios hasta cumplir el criterio, siendo un método descartable para nuestro estudio.

Si evaluamos el rendimiento de los clasificadores utilizados, observamos que sobre la base de datos Digits los resultados son realmente buenos salvo para el clasificador árbol de decisión. Respecto a los otros tres clasificadores obtienes resultados muy parejos, destacando mínimamente Random Forest. En cuanto a la base de datos de PIMA el clasificador que mejor solventa el problema es SVM estando por encima de los otros tres, que obtienen valores similares.

Los resultados obtenidos nos muestran que el comportamiento del algoritmo activo es superior frente al pasivo. Planteado el problema en el que la obtención de los datos etiquetados supone un coste elevado, el rendimiento obtenido utilizando una cantidad menor se aproxima al que se obtiene con el conjunto de datos de entrenamiento completo.

Viendo estos resultados, concluimos que el aprendizaje activo puede ser de gran utilidad en muchos problemas de clasificación del mundo real. El ahorro que esto supone puede abrir puertas a abordar problemas de clasificación de patrones que suponen un coste inasumible usando algoritmos estándar, como la clasificación de textos en situaciones con una cantidad masiva de información. Así mismo puede ser interesante en problemas de clasificación en los que, aunque los costes sean asumibles, esta técnica los reduce considerablemente.

Durante este proyecto se han llevado a cabo pruebas y se han obtenido resultados sobre tres bases de datos elegidas por su naturaleza diferente. En el futuro podría ser interesante realizar estudios sobre otros tipos de problemas para confirmar los resultados obtenidos en este trabajo, tanto para problemas multiclase como problemas binarios. También se propone extender el estudio sobre otros clasificadores tales como redes bayesianas, Naïve Bayes o redes neuronales para comprobar si las conclusiones son similares a las obtenidas en este TFG.

# Glosario de acrónimos

- **IA**: Inteligencia Artificial
- **AA**: Aprendizaje Automático
- **AL**: Active Learning
- **ML**: Machine Learning
- **RF**: Random Forest
- **DTree**: Decision Tree
- **SVC**: Support Vector Classifier
- **SVM**: Support Vector Machine

# Bibliografía


[1] Friedhelm Schwenker and Edmondo Trentin. Pattern classification and clustering: a review of partially supervised learning approaches. *Pattern Recognition Letters*, 37:4–14, 2014.

[2] Dana Angluin. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988.

[3] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 1996.

[4] Eric B Baum and Kenneth Lang. Query learning can work poorly when a human oracle is used. In *International Joint Conference on Neural Networks*, volume 8, page 8, 1992.

[5] Atlas R. Ladner M. El-Sharkawi R. Marks IIm M. Aggoune Cohn, David L. and D. Park. *Training connectionist networks with queries and selective sampling.* University of Washington, Dept. of Computer Science, 1990.

[6] Ido Dagan and Sean P Engelson. Committee-based sampling for training probabilistic classifiers. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 150–157. The Morgan Kaufmann series in machine learning,(San Francisco, CA, USA), 1995.

[7] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.

[8] Tom M Mitchell. Generalization as search. *Artificial intelligence*, 18(2):203–226, 1982.

[9] Vikram Krishnamurthy. Algorithms for optimal scheduling and management of hidden markov model sensors. *Signal Processing, IEEE Transactions on*, 50(6):1382–1397, 2002.

[10] Hwanjo Yu. Svm selective sampling for ranking with application to data retrieval. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 354–363. ACM, 2005.

[11] Atsushi Fujii, Takenobu Tokunaga, Kentaro Inui, and Hozumi Tanaka. Selective sampling for example-based word sense disambiguation. *Computational Linguistics*, 24(4):573–597, 1998.

[12] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

[13] Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *AAAI*, volume 5, pages 746–751, 2005.

[14] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM, 1992.

[15] Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *Advances in neural information processing systems*, pages 1289–1296, 2008.

[16] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[17] Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, pages 441–448, 2001.

[18] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

[19] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1070–1079. Association for Computational Linguistics, 2008.

[20] Fredrik Olsson. A literature survey of active machine learning in the context of natural language processing. *SICS Technical Report*, 2009.

[21] Florian Laws and Hinrich Schätze. Stopping criteria for active learning of named entity recognition. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 465–472. Association for Computational Linguistics, 2008.

[22] Andreas Vlachos. A stopping criterion for active learning. *Computer Speech & Language*, 22(3):295–312, 2008.

[23] Jingbo Zhu and Eduard H Hovy. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *EMNLP-CoNLL*, volume 7, pages 783–790, 2007.

[24] Jingbo Zhu, Huizhen Wang, and Eduard Hovy. Multi-criteria-based strategy to stop active learning for data annotation. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 1129–1136. Association for Computational Linguistics, 2008.

[25] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[26] Tin Kam Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

[27] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[28] Mark R Segal. Machine learning benchmarks and random forest regression. *Center for Bioinformatics & Molecular Biostatistics*, 2004.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[30] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[31] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.