UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

MASTER UNIVERSITARIO DE INVESTIGACIÓN EN INGENIERÍA DE SOFTWARE Y SISTEMAS INFORMATICOS

ITINERARIO INGENIERÍA DE SOFTWARE

TRABAJO FIN DE MASTER
(Código 31105151)

TOMA DE DECISIONES BASADA EN EL CLASIFICADOR DE NAIVE BAYES PONDERADO

ALUMNO: RAFAEL JAVIER BERNAL VAZQUEZ

DIRECTOR: PEDRO JAVIER HERRERA CARO

JUNIO, 2018

MASTER UNIVERSITARIO DE INVESTIGACIÓN EN INGENIERÍA DE SOFTWARE Y SISTEMAS INFORMATICOS

ITINERARIO INGENIERÍA DE SOFTWARE

TRABAJO FIN DE MASTER
(Código 31105151)

TOMA DE DECISIONES BASADA EN EL CLASIFICADOR DE NAIVE BAYES PONDERADO

TIPO B: TRABAJO ESPECÍFICO PROPUESTO POR EL ALUMNO

ALUMNO: RAFAEL JAVIER BERNAL VAZQUEZ

DIRECTOR: PEDRO JAVIER HERRERA CARO

JUNIO, 2018

CALIFICACIONES

DECLARACIÓN JURADA DE AUTORÍA DEL TRABAJO CIENTÍFICO, PARA LA DEFENSA DEL TRABAJO FIN DE MASTER

Fecha: 20/05/2018

Quién suscribe:

Autor(x): Rafael - Javier Bernal Vázquez.
D.N.I/N.I.E/Pasaporte.: 48813872-T

Hace constar que es la autor(x) del trabajo:

Título completo del trabajo.
Toma de decisiones basada en el clasificador de Naive Bayes

En tal sentido, manifiesto la originalidad de la conceptualización del trabajo, interpretación de datos y la elaboración de las conclusiones, dejando establecido que aquellos aportes intelectuales de otros autores, se han referenciado debidamente en el texto de dicho trabajo.

DECLARACIÓN:

- ✓ Garantizo que el trabajo que remito es un documento original y no ha sido publicado, total ni parcialmente por otros autores, en soporte papel ni en formato digital.
- ✓ Certifico que he contribuido directamente al contenido intelectual de este manuscrito, a la génesis y análisis de sus datos, por lo cual estoy en condiciones de hacerme públicamente responsable de él.
- ✓ No he incurrido en fraude científico, plagio o vicios de autoría; en caso contrario, aceptaré las medidas disciplinarias sancionadoras que correspondan.

R. J. Bernal

Impreso TFdM05_Autor. Autorización de publicación y difusión del TFdM para fines académicos

Autorización

Autorizo/amos a la Universidad Nacional de Educación a Distancia a difundir y utilizar, con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firma del/los Autor/es

Resumen

Este Trabajo Fin de Máster se enmarca dentro de los sistemas de apoyo para la toma de decisiones en el desarrollo de software, una de las líneas de investigación que cubre el Máster Universitario de Investigación en Ingeniería de Software y Sistemas Informáticos de la UNED. El Aprendizaje Automático es una disciplina que trata de crear algoritmos que, a partir de una serie de datos, sean capaces de encontrar patrones complejos que ayuden a predecir comportamientos futuros. Existe una gran variedad de este tipo de algoritmos y a grandes rasgos se pueden clasificar como supervisados y no supervisados. La mejora en el rendimiento de estos algoritmos, aunque sólo sea en un determinado rango, puede tener un importante impacto en multitud de áreas que van desde el ámbito empresarial hasta el científico.

En concreto, este trabajo propone una mejora del algoritmo clásico de Naive Bayes, basada en un método ponderado. Dicho algoritmo se basa en el teorema de Bayes junto con algunas hipótesis simplificadoras y se caracteriza por su sencillez y buenos resultados. El rendimiento de este algoritmo incluso puede llegar a ser comparable con árboles de decisión y redes neuronales en determinados casos. Sin embargo, debido a la hipótesis de independencia entre atributos, hace que la calidad de los resultados obtenidos en aquellos dominios donde haya una fuerte dependencia entre variables decaiga.

El método propuesto es probado utilizando 6 colecciones de datos cada una de ellas con características diferentes tales como: número de atributos, número de registros, posibles valores de clasificación, etc. Asimismo, se utilizan ejemplos de algoritmos supervisados y no supervisados para poner en valor dicho método propuesto. Los algoritmos escogidos son: agrupamiento difuso, KNN y SVM.

Además, las limitaciones del algoritmo de Naive Bayes son conocidas desde hace más de 40 años por lo que han surgido toda una serie de métodos ponderados que han contribuido a la mejora del método original. Por lo que las 6 colecciones citadas se comparan con uno de los métodos ponderados más conocidos: la selección de atributos de Naive Bayes utilizando el algoritmo de árboles de decisión C4.5

Palabras Clave

Sistema de soporte a la Toma de Decisiones, Almacén de Datos, Aprendizaje Automático, Proceso de Extracción del Conocimiento, Minería de Datos, Naive Bayes.

Abstract

This Final Master Project is part of the support systems for decision making in Software Development, one of the lines of research that covers the Master's Degree in Software Engineering and Computer Systems of the UNED. Machine Learning is a discipline that tries to create algorithms that, based on a series of data, are able to find complex patterns that help predict future behaviors. There is a wide variety of algorithms and can be classified as supervised and unsupervised. The improvement in the performance of these algorithms, even if only in a certain range, can have an important impact in many areas ranging from business to scientific.

In particular, this work proposes an improvement of the classic Naive Bayes algorithm, based on a weighted method. This algorithm is based on Bayes' theorem together with some simplifying hypotheses and is characterized by its simplicity and good results. The performance of this algorithm can even be comparable with decision trees and neural networks in certain cases. However, due to the hypothesis of independence between attributes, the quality of the results obtained in those domains where there is a strong dependency between variables falls.

The proposed method is tested using 6 collections of data with differents characteristics such as: number of attributes, number of records, possible classification values, etc. Also, examples of supervised and unsupervised algorithms are used to value said method proposed. The chosen algorithms are FC, KNN and SVM.

In addition, the limitations of the Naive Bayes algorithm have been known for more than 40 years, and a whole series of weighted methods have emerged and have contributed to the improvement of the original method. Therefore, the six mentioned collections will be compared with one of the best-known weighted methods: the selection of Naive Bayes attributes using the decision tree algorithm C4.5

Keywords

Decision Support System, Data Warehouse, Machine Learning, Knowledge Discovery in Databases, Data Mining, Naive Bayes.

Índice de contenido

Índice de tablas

Índice de ilustraciones

1. Introducción

En los últimos tiempos se ha reconocido un nuevo potencial de gran valor que es la gran cantidad de datos almacenados informáticamente en los sistemas de información de empresas, gobiernos, instituciones, etc. Inicialmente estos datos se consideraban el producto final de una determinada solución de software, sin embargo, hoy en día estos datos son una nueva materia prima a partir de la cual se puede obtener un nuevo valor cada vez más en alza: el conocimiento.

La forma de obtener este conocimiento ha variado a lo largo del tiempo. Inicialmente el análisis de una base de datos se realizaba mediante consultas realizadas directamente sobre una BBDD operacional u OLTP. Poco a poco, esta forma de explotación se ha demostrado poco flexible y muchas veces poco escalable a grandes volúmenes de datos.

La tecnología de base de datos respondió a estos problemas con la aparición del *Data Warehouse* o Almacén de Datos. Su arquitectura consta de un repositorio de fuentes heterogéneas de datos integrados y organizados bajo un esquema unificado para facilitar su análisis. Esta tecnología incluye operaciones de tipo OLAP[1] (procesamiento analítico en línea), así como la posibilidad de ver la información desde distintas perspectivas.

Sin embargo, a pesar de que estas herramientas OLAP aportan cierto análisis descriptivo, no genera conocimiento que pueda ser aplicado a otros datos. Por ejemplo, se puede saber que el 10% de los ancianos sufren Alzheimer, lo cual puede ser de utilidad, pero es mucho más valioso tener un conjunto de reglas a partir de la cuales se pueda determinar si una persona tendrá Alzheimer o no en el futuro.

Por ello ha surgido una nueva generación de herramientas y técnicas para ayudar a extraer conocimiento útil a partir de la información disponible. Ejemplos de estas técnicas son los árboles de decisión, redes neuronales, Bayes Naive, etc. La mejora de estas herramientas y técnicas tiene una repercursión en bastantes áreas que van desde aplicaciones en medicina, vehículos autónomos, robots, análisis de imágenes, previsión del clima, etc.

1.1 Objetivos del TFM

El objetivo principal de este Trabajo de Fin de Máster es implementar una mejora a uno de los principales clasificadores probabilísticos aplicados en la toma de decisiones actuales: Naive Bayes (en adelante NB).

El método NB es un clasificador basado en el teorema de Bayes. Se trata de un algoritmo supervisado ya que necesita de datos de entrenamiento para que pueda ser aplicado. La principal característica de este método se basa en la suposición de que los diferentes sucesos que puedan ocurrir son independientes de cara al resultado final, es por ello que se le conoce como ingenuo. Este método destaca por su sencillez y buenos resultados obtenidos.

Para comprobar estas mejoras aplicadas se comparan los resultados obtenidos con respecto al

1 OLAP es el acrónimo en inglés On Line Analytical Processing. Su objetivo es agilizar consultas de base de datos para lo cual utiliza estructuras multidimensionales que contienen datos resumidos de grandes bases de datos.

método de NB original, además de compararlo con otros métodos existentes como son: *Fuzzy Clustering* (FC), *K-Nearest Neighbors* (KNN), *Support Vector Machine* (SVM) y otro método que utiliza NB combinado con el algoritmo C4.5 de árboles de decisión. De esta forma se puede poner en valor la implementación propuesta y considerar si es una solución válida.

Dentro del objetivo se encuentra no sólo el mejorar el algoritmo respecto al original, sino que se la optimización realizada esté, al menos, a un nivel similar a otros algoritmos más eficientes.

1.2 Justificación del TFM

La justificación de este Trabajo Fin de Máster se encuentra relacionada con la principal debilidad que presenta el clasificador NB que es la hipótesis de independencia entre atributos. Este es el motivo por el cual en aquellos dominios donde hay una relación evidente entre dos o más atributos, los resultados obtenidos no tengan la precisión que podría obtenerse con otros algoritmos (como por ejemplo, C4.5).

Desde hace más de 40 años se han venido realizando diferentes propuestas para mitigar esta debilidad. Dichas propuestas pueden clasificarse fundamentalmente en dos tipos. Por un lado, la introducción de factores de ponderación sobre los atributos, que no es otra cosa mas que introducir un peso a cada uno de los atributos, incluso pudiendo llegar a tener el valor 0 si un atributo no aporta valor a la clasificación.Y por otro lado, se encuentran aquellas propuestas que se basan en relajar la hipótesis de independencia.

La propuesta que se realiza, aunque no establezca pesos a cada uno de los atributos, se encuentra estrechamente relacionada con la ponderación de atributos. Por tanto, podría catalogarse dentro de esta categoría.

1.3 Estructura del TFM

La estructura del Trabajo de Fin de Máster seguirá los siguientes apartados:

- Introducción
- Estado del arte
- Solución propuesta
- Resultados obtenidos
- Conclusiones y Trabajo Futuro
- Bibliografía

En el segundo capítulo, dedicado al estado del arte se repasa brevemente lo que es la minería de datos y en concreto, su aplicación en la toma de decisiones. Se detalla cómo es el proceso de extracción de información y de cómo se obtienen los patrones a través de los cuales se realiza dicha extracción. Dentro de la mineria de datos se distingue entre tareas y métodos. Las tareas pueden ser tanto descriptivas como predictivas y dentro de los métodos se detallan los más utilizados como son: máquinas de vectores de soporte, redes neuronales artificiales, métodos basados en vecindad, Naive Bayes, etc. Por último, se detallan los principales métodos ponderados de NB que han realizado correcciones sobre el método original.

En el tercer capítulo se presenta la solución propuesta, donde se explica en qué consiste el método

ponderado de NB, cuáles son los criterios que se han establecido, cómo se ha llegado a tales conclusiones, trabajos utilizados de referencia, colecciones de datos utilizadas, etc.

En el cuarto capítulo se presentan los resultados obtenidos, se aplica el método ponderado y se compara no sólo con lo que se hubiera obtenido aplicando el método de NB original, sino que también se utilizan otros algoritmos que son ampliamente utilizados actualmente. Para esta comparativa se utilizan algoritmos supervisados como son KNN y SVM, uno no supervisado como es FC, y además, con otro algoritmo de NB ponderado. En el caso del algoritmo no supervisado se espera que la mejora con respecto a éste sea la más evidente, ya que este tipo de algoritmos no dispone de una batería de ejemplos previamente clasificados.

Por último, en el quinto capítulo se reflejan las conclusiones y las propuestas de trabajo futuro. Se realiza una lectura crítica de los resultados obtenidos en el capítulo anterior y se evalúa el grado de consecución de los objetivos propuestos en este TFM. En base a estas conclusiones se proponen las líneas de trabajo futuras.

1.4 Software utilizado y colecciones de datos del TFM

Existe una amplia variedad de software disponible, tanto comercial como de libre distribución, para la realización de labores de minería de datos. Desde el punto de vista del software comercial destacan: Saldford Systems, Stasoft and IBM Spss Predictive Analytics. En lo que al software libre se refiere destacan: RapidMiner, Weka y R.

Para el desarrollo de este Trabajo Fin de Master se ha escogido R (www.r-project.org), el cual se encuentra disponible en la url www.r-project.org. Las razones principales para esta elección son:

- Es un software de libre distribución.
- Se encuentra disponible en multitud de plataformas (UNIX, Windows y MacOS).
- Es un lenguaje orientado a objetos fácil de aprender para aquellas personas que ya han desarrollado previamente en otros lenguajes tipo java o C sharp.
- Dispone de una amplia gama de librerías que permiten implementar funcionalidades estadísticas, de representación gráficas, algoritmos supervisados y no supervisados …
- Las librerías de las que dispone se encuentran perfectamente documentadas.
- La comunidad de usuarios que lo utiliza puede reportar bugs que encuentre utilizando esta herramienta y cada cierto tiempo una nueva versión de este software está disponible. Esto hace que se trate de una herramienta altamente contrastada.
- Muy utilizada en el mundo académico y científico.

Por otro lado, las colecciones de datos utilizadas en dicho trabajo se han extraído de la web de UCI (Center for Machine Learning and Intelligent Systems) la cual dispone de más de 400 colecciones de datos a libre disposición para la investigación realizada dentro del ámbito del Machine Learning.

2. Estado del Arte

2.1 Visión General

Debido a las nuevas tecnologías, en concreto a los sistemas de información, que han ido desarrollándose progresivamente durante la segunda mitad del siglo XX se ha llegado a mejorar significativamente la gestión de la información en diferentes ámbitos que van desde la investigación, la empresa privada o la administración pública.

De forma paralela a estas mejoras, toda esta información gestionada ha venido generando una gran cantidad de datos. Esta ingente cantidad de datos - que a priori podría ser considerada de poca utilidad una vez que ha pasado un cierto tiempo - se ha convertido en una auténtica nueva materia prima.

Es en este punto donde surge como nueva disciplina la minería de datos, que de una manera formal se puede definir como *"el descubrimiento eficiente de información valiosa, no-obvia de una gran colección de datos"* (Bigus, 1996). Dicho de otra forma, todas estas colecciones de datos almacenadas durante todo este tiempo permiten generar conocimiento. Este conocimiento, a su vez, se puede aplicar en la toma de decisiones.

Para encontrar este conocimiento, no obvio inicialmente, es necesario la búsqueda de patrones. Al comienzo esta búsqueda de patrones se realizaba mediante consultas ejecutadas directamente sobre bases de datos usando lenguajes como SQL[2]. Posteriormente, llegó la arquitectura de *Data Warehouse* o almacén de datos, donde a partir de un esquema denominado de *staging* donde se procedía a volcar información procedente de sistemas heterogéneos, para posteriormente aplicar procesos de transformación y homogeneización y llegar a obtener un *datamart*. Este *datamart* contenía información específica sobre un área determinada que finalmente ayuda a la toma de decisiones.

Asociado a la tecnología del *Data Warehouse* están las operaciones OLAP, las cuales permiten un mayor análisis comparado con lo que se obtiene aplicando consultas SQL directamente sobre una base de datos. Esto es posible al permitir obtener información desde una amplia gama de perspectivas y de una forma mucho más rápida y eficiente.

Sin embargo, estas tecnologías presentan serías limitaciones a la hora de generar conocimiento que pueda ser utilizado a otros datos, o lo que es lo mismo limita la toma de decisiones. Es en este contexto es donde surge una nueva disciplina que es el *Machine Learning* o Aprendizaje Automático/Máquina. En esta disciplina se toman decisiones a partir de algoritmos que son capaces de revisar grandes volúmenes de datos y predecir comportamientos futuros. Dicho en otras palabras,

2 SQL es el acrónimo en inglés Structured Query Language. Entre sus características se encuentran el uso de álgebra y cálculo relacional, los cuales permiten realizar consultas con el fin de recuperar información de base de datos.

esta disciplina trata de responder a la pregunta: ¿Cómo podemos construir sistemas informáticos que automáticamente mejoran con la experiencia, y cuáles son las leyes fundamentales que gobiernan todos los procesos de aprendizaje? (Dietterich, 1990)

En la Tabla 1 se muestra el detalle de los principales hitos en la búsqueda de conocimiento desde la segunda mitad del siglo pasado:

Evolución	Tecnologías	Características
Colecciones de datos (1960 -)	Computadoras, cintas, disco	Manipulación estadística
Acceso a datos (1980-)	Base de datos relacionales, lenguaje de búsqueda estructurados (SQL)	Resultados dinámicos de búsqueda a nivel de registros.
Almacenes de datos (1990-)	Base de datos multidimensionales	Resultados dinámicos de búsqueda en múltiples niveles
Minería de datos (2000-)	Algoritmos avanzados, computadoras multiproceasador	Información prospectiva y proactiva

Tabla 1: Evolución de la minería de datos desde 1960 a la actualidad. Fuente: (Aldana, 2000)

2.2 Sistemas de Soporte a la Toma de Decisiones (DSS)

Como se ha expuesto anteriormente, poco a poco se ha ido pasando a un nuevo paradigma: el conocimiento. Incluso yendo más lejos se debería hablar del conocimiento útil o inteligencia de negocio. En este entorno tecnológico ha surgido el concepto de sistemas de soporte a la toma de decisiones.

Un sistema de soporte a la toma de decisiones (DSS por sus siglas en inglés Decision Support System) puede definirse de muchas maneras debido al amplio rango de aplicaciones que tiene. A continuación se muestran las definiciones principales:

- Un DSS, en términos generales, es "un sistema basado en computador que ayuda en el proceso de toma de decisiones" (Finlay, 1994).

- De forma más específica, se puede definir un DSS como "un sistema de información basado en un computador interactivo, flexible y adaptable, especialmente desarrollado para apoyar la solución de un problema de gestión no estructurado para mejorar la toma de decisiones. Utiliza datos, proporciona una interfaz amigable y permite la toma de decisiones en el propio análisis de la situación" (Turban, 1995)

Los sistemas DSS se pueden distinguir del resto sistemas que pueda tener una compañía por los siguientes componentes característicos (Turban y col., 2005):

- **Administrador de datos**. Posee una base de datos propia que puede ser interconectada con el *Data Warehouse* de la corporación.

- **Administrador del modelo**. Este componente facilita modelos financieros, científicos, cuantitativos, etc., que provee de capacidad analítica al sistema.

- **Interfaz del usuario.** Se caracteriza por ser bastante intuitiva y consistente para el usuario.

- **Administrador del conocimiento.** Este componente facilita la toma de decisiones.

Entre los principales beneficios de los DSS se pueden destacar (Yañez, 2008) :

- Respuestas inmediatas a situaciones imprevistas.
- Manejo de varias estrategias bajo distintas condiciones de manera rápida y objetiva.
- Mejora del control y desempeño administrativo.
- Mejora el desempeño para análisis.

Desde el punto de vista de lo que caracteriza a un sistema DSS destacan (Yañez, 2008):

- Administración del conocimiento
- Modelado
- Fácil de construir y de usar
- Dirigido a grupos o a individuos, directivos a distintos niveles.
- Decisiones secuenciales
- Efectividad sobre eficiencia.
- Adaptabilidad y flexibilidad.
- Variedad de estilos de decisión y procesos.
- Control humano sobre la máquina
- Evolución del sistema
- Inteligencia, diseño y elección.

Las aplicaciones de los sistemas DSS abarcan la mayor parte de industrias y negocios. Entre los beneficios de aplicar este tipo de soluciones se encuentran (Stair y Reynolds, 2000):

- Los administradores de universidades pueden utilizar un DSS para programar los horarios de manera efectiva.

- Un DSS ayuda a la planificación de la producción en base a los datos de ventas, programas de trabajo y flujo de trabajo.

- En el área de inversiones, los analistas financieros utilizan un DSS para diversificar los fondos de un cliente entre un grupo apropiado de opciones de inversión para minimizar el riesgo y proporcionar una tasa de rendimiento adecuada sobre la inversión.

2.3 Proceso de Extracción del Conocimiento

El proceso de extracción de conocimiento o Knowledge Discovery from Database (KDD) se define como "*el proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y, en última instancia, comprensibles a partir de los datos*" (Fayyad y col., 1996). Este proceso consta de una secuencia iterativa de etapas o fases, ya que es posible que sean necesarias varias iteraciones para poder llegar a obtener una extracción de conocimiento de alta calidad. A grosso modo, puede decirse que un KDD transforma la información en conocimiento.

En la Ilustración 1 se pueden apreciar las etapas en las que se organiza un proceso KDD:

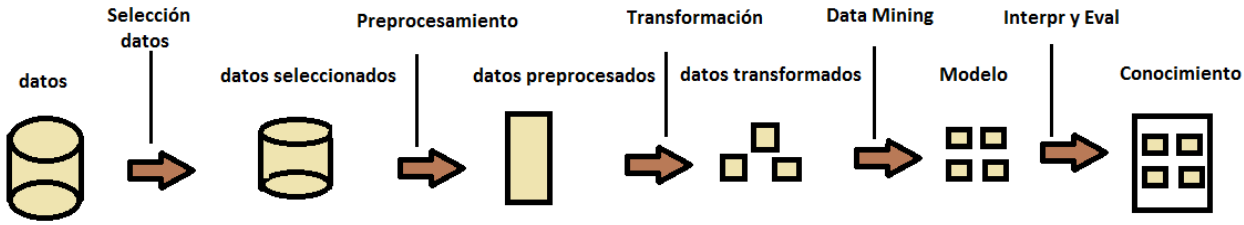

Ilustración 1: Etapas del proceso KDD

- **Fase de Selección de datos**.

  Normalmente los datos que se necesitan explotar no se encuentran en una sola base de datos, sino que por el contrario suelen proceder de fuentes muy diversas, tanto internas como externas a la empresa o institución. Este hecho supone una dificultad añadida ya que el formato de los datos o criterios utilizados puede ser diferente dependiendo de la fuente.

  El D*ata Warehouse* o almacén de datos surge como arquitectura donde almacenar todos estos datos de origen diverso. De esta forma, las nuevas tecnologías responden al incremento de colecciones de datos de bases de datos transaccionales por parte de las empresas e instituciones.

- **Fase de Preprocesamiento**

  Tras haber recopilado toda la información en la fase previa y volcado a un repositorio común, el siguiente paso es seleccionar y preparar el conjunto de datos que se pretende estudiar.

  En esta fase se toman decisiones importantes acerca de la calidad de los datos y, por lo tanto, del conocimiento final adquirido. Por ejemplo, es de crucial importancia decidir qué hacer con datos anómalos, ya que pueden ser errores en la medición del dato o por el contrario son datos correctos de un evento singular que se deben de tener en cuenta.

  En general, es necesario reflexionar previamente acerca del significado de aquellos datos que se desvían de lo esperado antes de eliminarlos o decidir incluirlos en las siguientes fases. Es necesario distinguir el mal funcionamiento de un dispositivo, un valor por defecto …

La calidad del conocimiento descubierto no sólo depende del algoritmo de minería utilizado, sino también de la calidad de los datos. Es por ello, que después de recopilar los datos, es necesario prepararlos para que puedan ser explotados mediante técnicas de minería de datos (Hernández y col., 2004).

- **Fase de Transformación**

  En esta fase se procede a seleccionar a aquellos atributos que son más relevantes para las tareas de minerías de datos. Aunque los propios algoritmos de minería de datos se encarguen de seleccionar aquellas variables más importantes o ignorar aquellas que menos impacto tienen, es importante utilizar el conocimiento que se tenga sobre el dominio del problema a la hora de mejorar los atributos a utilizar.

  Se utilizan métodos de reducción de dimensiones o de transformación para disminuir el número efectivo de variables bajo consideración o para encontrar representaciones invariantes de los datos (Fayyad y col., 1996). Las técnicas de reducción que se utilizan son agregaciones, compresión de datos, histogramas, segmentación, discretización basada en entropía, etc. (Han y Kamber, 2001).

- **Minería de datos**

  El objetivo de esta fase es producir nuevo conocimiento que pueda utilizar el usuario. Se trata de buscar patrones insospechados y de interés, aplicando tareas de descubrimiento como clasificación (Quinlan, 1986), agrupamiento (Ng y Han, 1994), patrones secuenciales (Agrawal y Srikant, 1995) y asociaciones (Agrawal y Skrikant, 1994).

  Para ello hay que construir un modelo basado en los datos recopilados para este propósito. El modelo es una descripción de los patrones y relaciones entre los datos que puede usarse para poder hacer predicciones, entender mejor los datos o entender sucesos pasados.

  Por todo esto es importante tomar una serie de decisiones antes de comenzar esta fase que pueden sintetizarse en las siguientes cuestiones:

  - Qué tipo de tarea de minería es el más apropiado (por ejemplo clasificación)
  - Qué tipo de modelo utilizar (por ejemplo un árbol de decisión)
  - Qué algoritmo de minería de datos resuelve mejor la tarea y obtiene el modelo deseado.

- **Interpretación y evaluación**

  Por último, en esta fase es donde se valora la calidad del resultado obtenido. Idealmente, los patrones descubiertos deben tener tres cualidades a destacar: precisos, comprensibles e interesantes (novedoso y útil). Existen diversas técnicas de evaluación pero entre ellas destacan la validación simple y la cruzada con n pliegues:

  - Validación Simple

    En la validación simple se reserva un porcentaje de los datos que se tienen como conjunto de prueba. De esta forma, al no utilizar estos datos para construir el modelo

es seguro que la prueba es totalmente independiente. Normalmente el porcentaje de datos utilizados suele estar entre el 5% - 50% y suelen ser lo más hetergéneos posibles para que la prueba sea representativa.

- Validación cruzada con *n* pliegues

  En la validación cruzada con *n* pliegues los datos se dividen en *n* grupos. Una vez obtenidos los *n* grupos, uno de ellos se reserva como conjunto de pruebas y los *n*-1 restantes se utilizan para construir el modelo. Este proceso se repite *n* veces y en cada iteración se deja un grupo diferente para la prueba. Finalmente, se construye un modelo con todos los datos y se obtienen sus ratios de error promediando los *n* ratios de error obtenidos en las diferentes iteraciones. La validación cruzada es una manera de predecir el ajuste de un modelo a un hipotético conjunto de datos de prueba cuando no se dispone del conjunto explícito de datos de prueba (Payam y col., 2008).

Un analista experto (o varios) en el área que se investiga recomendará a los responsables de la empresa o institución, las acciones a tomar en base a los resultados obtenidos tras construirse y validarse el modelo. Además de estas recomendaciones también propondrá otras líneas futuras de investigación.

Por último, es importante que todo este conocimiento no sea restringido, ya que es bueno que se comunique y distribuyan los resultados obtenidos a los usuarios o empleados implicados dentro de la organización mediante seminarios o charlas. Y es que este nuevo conocimiento adquirido debe integrar el *know-how* de la organización.

2.4 Minería de datos

Tal y como se ha visto en el apartado anterior la minería de datos es una etapa dentro del proceso de extracción del conocimiento (KDD). En concreto, es "*el paso consistente en el uso algoritmos concretos que generan una enumeración de patrones a partir de los datos preprocesados"* (Fayyad y col., 1996).

La minería de datos también puede ser definida como el proceso de descubrir conocimiento interesante de grandes cantidades almacenadas en base de datos, *Data Warehouses* u otro repositorio de información (Han y Kamber, 2001). Algunas de sus principales aplicaciones son: toma de decisiones, procesos industriales, investigaciones científicas, etc.

Las raíces de la minería de datos se remontan a mediados del siglo anterior cuando eran los propios departamentos de informática los que preparaban resúmenes de información para los responsables de cada área. Incluso en esta época, ya los estadísticos utilizaban términos como *Data Fishing, Data Mining* o *Data Archeology*. La forma de proceder era poco flexible a la hora de procesar la información, sobre todo cuando ésta era voluminosa y compleja.

En las siguientes décadas aparecieron los sistemas gestores de base de datos, los cuales eran bastante rígidos ya que apenas tenían flexibilidad a la hora de realizar consultas. Además las bases de datos eran cada vez de origen más heterogéneo, por lo que cada vez era más difícil examinar la información de una forma integrada.

En la década de los 80 aparece el *Data Warehouse* donde no sólo se comienzan a solucionar los problemas anteriores de gestión de cantidades voluminosas de datos y heterogeneidad de los mismos, sino que además es cuando se comienza a potenciar realmente a la minería de datos. Es en esta época cuando Rakesh Agrawal, Gio Wiederhold, Robert Blum y Gregory Piatetsky-Shapiro, entre otros, comienzan a utilizar los términos de *Data Mining* y KDD (Ramirez, 2003).

Es frecuente al utilizar el término de minería de datos que se utilice de forma indistinta con otros términos como estadística u OLAP. Sin embargo, estas son disciplinas distintas cuyas diferencias principales son (Beltran, 2016):

.

- Minería de datos versus Estadística

  Ambos tienen como objetivo construir modelos que describan una determinada situación a partir de unos datos. Sin embargo, la minería de datos construye el modelo de forma automática, mientras que la estadística necesita de un estadístico profesional.

  Además la minería de datos utiliza técnicas de Inteligencia Artificial, a mayor dimensionalidad del problema ofrece mejor soluciones y usa técnicas menos restrictivas (pueden ser utilizadas con los mínimos supuestos posibles) que las estadísticas.

  Por otro lado, las técnicas estadísticas se centran en técnicas confirmatorias, mientras que las técnicas de minería de datos suele utilizar técnicas exploratorias. Es decir, en la minería de datos se delega parte del conocimiento analítico en técnicas de aprendizaje.

- Minería de datos versus OLAP

  Las herramientas OLAP permiten navegar rápidamente por lo datos, pero no se genera conocimiento en el proceso.

Dentro de la minería de datos hay que distinguir entre tareas y métodos. Las tareas de minería de datos pueden considerarse como un tipo de problema para ser resuelto por un algoritmo de minería de datos (Hernández y col., 2004). Por ejemplo, si se necesita clasificar determinados tipos de piezas es una tarea de clasificación. La forma de resolverlo es el método, por ejemplo, en el caso descrito podrían utilizarse los árboles de decisión. Dicho de otra forma, los métodos son técnicas que permiten resolver la tarea en cuestión.

2.4.1 Tareas de Minería de Datos

Las tareas de minería de datos pueden clasificarse como:

- **Predictivas**. Son problemas en los que hay que predecir uno o más valores para uno o más ejemplos. Se pueden diferenciar los siguientes tipos:
  - Clasificación o discriminación:

    Los ejemplos se presentan como un conjunto de pares de elementos de dos conjuntos, $\delta = \{<e,s>: e \in E, s \in S\}$, donde S es el conjunto de valores de salida. El objetivo es aprender una función $\lambda : E \rightarrow S$, denominada clasificador, que

represente la correspondencia existente entre los ejemplos, es decir, para cada valor de E se tiene un unico valor para S.

- Clasificación suave:

  La presentación del problema es la misma que la de la clasificación, pares de elementos de dos conjuntos, $\delta = \{ <e,s> : e \in E, s \in S \}$. Además de la función $\lambda : E \rightarrow S$ se aprende otra función $\Theta_i : E \rightarrow R$ que significa el grado de certeza de la predicción hecha por la función λ. Es decir, nos proporciona una medida de la fiabilidad de dichas clasificaciones.

- Estimación de probabilidad de clasificación:

  La presentación del problema es la misma que la clasificación normal y suave, pares de elementos de dos conjuntos, $\delta = \{ <e,s> : e \in E, s \in S \}$. Eso sí, la función a aprender es diferente. Se trata de aprender exclusivamente m funciones $\Theta_i : E \rightarrow R$, donde m es el número de clases. Es decir, cada función a aprender retorna para cada ejemplo m un valor real p_i. Cada uno estos valores p_i se denomina probabilidad de la clase i y significa el grado de certeza de que un ejemplo sea de la clase i.

- Categorización:

  Aquí cada ejemplo de $\delta = \{ <e,s> : e \in E, s \in S \}$, así como la correspondencia a aprender $\lambda : E \rightarrow S$, puede asignar varias categorías a un mismo e, a diferencia de la clasificación, que solo asigna una.

- Preferencias o Priorización:

  El aprendizaje de preferencias consiste en determinar a partir de dos o más ejemplos, un orden de preferencia.

- Regresión:

  El conjunto de evidencias son correspondencias entre dos conjuntos $\delta : E \rightarrow S$, donde S es el conjunto de valores de salida. El objetivo es aprender una función $\lambda : E \rightarrow S$ que represente la correspondencia existente entre los ejemplos. La diferencia respecto a la clasificación es que S es numérico, es decir, que puede ser un valor entero o real.

- **Descriptivas**. En este caso no se pretende predecir nuevos datos, sino describir los existentes. Se pueden diferenciar diferenciar los siguientes tipos:

  - Agrupamiento o Clustering:

    El objetivo de esta tarea es obtener grupos entre los elementos de δ, de tal manera que los elementos asignados al mismo grupo sean similares.

  - Correlaciones y factorizaciones:

El objetivo es ver la relevancia de los atributos, detectar atributos redundantes o dependencias entre atributos.

- Reglas de asociación:

  El objetivo, en cierto modo, es similar a los estudios de correlaciones y factoriales, pero para los atributos nominales, muy frecuentes en las bases de datos. Este tipo de estudio también recibe el nombre de análisis de vínculos.

- Dependencias funcionales:

  Este tipo de tareas consideran todos los posibles valores, por ejemplo, una dependencia funcional podría ser la edad (discretizada en varios intervalos), el nivel de ingresos (discretizado en varios niveles), el código postal, si está casado, si tiene vehículo...

- Detección de valores e instancias anómalas.

  Este tipo de tareas puede ser muy útil para detectar comportamientos anómalos como pueden ser fraudes, fallos, intrusos o comportamientos diferenciados.

Cada una de las tareas descritas requiere de técnicas, métodos o algoritmos para resolverlas.

2.4.2 Técnicas de Minería de datos

Las técnicas de minería de datos crean modelos que pueden ser predictivos o descriptivos. Los modelos predictivos realizan predicciones a futuro, mientras que los modelos descriptivos proporcionan información acerca de las relaciones existentes entre los datos.

Para la creación de estos modelos es necesario extraer patrones. Los seres humanos tienen la capacidad de poder extraer patrones de forma innata de lo que ven en su entorno. Ejemplos de esto son las figuras en las nubes o las cuadrigas en las estrellas. Las diferentes técnicas de minería de datos han querido emular esta capacidad intentando incluso ir mucho más allá en determinados aspectos. Con estos patrones se pretende descubrir conocimiento que debe ser válido, novedoso, interesante y sobre todo comprensible.

De acuerdo a la función de salida estas técnicas se pueden distinguir varios tipos de aprendizaje: aprendizaje supervisado, aprendizaje no supervisado y semi-supervisado (Rebollo, 2009).

En el aprendizaje supervisado se trata de deducir una función a partir de los datos de entrenamiento. Esta función debe ser capaz de predecir el valor correspondiente a cualquier objeto de entrada válida después de haber visto una serie de ejemplos (datos de entrenamiento). Para ello, tiene que generalizar a partir de los datos presentados las situaciones no previstas inicialmente.

En el caso del aprendizaje no supervisado no se dispone de una batería de ejemplos previamente clasificados, sino que únicamente a partir de las propiedades de los ejemplos se procede dar una agrupación de cada caso según su similaridad.

Por último, aprendizaje semi-supervisado se encuentra a medio camino entre el aprendizaje

supervisado y no supervisado. Hay que considerar que la adquisición de un número grande datos de entrenamiento muchas veces puede ser inviable, mientras el obtener datos sin clasificar tiene un coste muy inferior. Cuando se combinan datos no marcados en conjunción con una pequeña cantidad de datos marcados, se puede producir una mejora considerable en la precisión alcanzada en el aprendizaje.

Las principales técnicas utilizadas en minería de datos son:

- Máquinas de Vectores de Soporte

  Las Máquinas de Vectores Soporte o *Support Vector Machines* (SVM) son un conjunto de algoritmos de aprendizaje supervisado desarrollados por Vladimir Vapnik y su equipo AT&T.

  Estos métodos están relacionados con problemas de clasificación (Burgues, 1998) y regresión (Gunn, 1998). Dado un conjunto de ejemplos de entrenamiento se pueden etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra.

  Dado un conjunto de puntos, subconjunto de un conjunto mayor, en el que cada uno de ellos pertenece a una de dos posibles categorías, un algoritmo basado en SVM construye un modelo capaz de predecir si un punto nuevo pertenece a una categoría o la otra. La SVM busca un hiperplano que separe de forma óptima a los puntos de una clase de la otra, que eventualmente han podido ser previamente proyectados a un espacio de dimensionalidad superior.

  Desgraciadamente, en la mayoría de los universos a estudiar la separación no puede realizarse mediante una línea o plano recto, ya que normalmente se debe tratar con más de dos variables predictoras o conjuntos de datos no totalmente separados. Debido a las limitaciones computacionales de las máquinas de aprendizaje lineal, éstas no pueden ser utilizadas en la mayoría de las aplicaciones del mundo real. Las representación por medio de funciones Kernel ofrece una solución a este problema, proyectando la información a un espacio de características de mayor dimensión el cual aumenta la capacidad computacional de la máquina de aprendizaje lineal.

  Las SVM han demostrado tener un gran desempeño en muchas aplicaciones del mundo real, llegando a ser un clasificador muy preciso en muchos casos, incluso superando a las redes neuronales (Betancourt, 2005). Entre las principales aplicaciones destacan el reconocimiento facial, OCR, Bioinformática, Minería de texto...etc.

  Sin embargo, a pesar de sus buenos fundamentos teóricos y buen desempeño al generalizar, las SVM no son adecuadas para clasificación con grandes conjuntos de datos, ya que la matriz del kernel crece de forma cuadrática con el tamaño del conjunto de datos, provocando que el entrenamiento de las SVM sobre conjuntos de datos grandes sea un proceso muy lento.

- Reglas de Inducción

Se extraen reglas de la forma si-entonces de un conjunto de datos, combinadas e incluso utilizando variables negadas (Gryzmala-Busee, 2010). Esta información puede utilizarse, por ejemplo, para temas relacionados con el marketing de un supermercado como es la ubicación de los productos.

- Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (RNA) o sistemas conexionistas son sistemas de procesamiento de la información cuya estructura y funcionamiento están inspirados en las redes neuronales biológicas (Montaño, 2002). Pueden usarse para el reconocimiento de patrones, la compresión de información y la reducción de la dimensionalidad, la visualización, etc. Es decir, se pueden utilizar como una herramienta para llevar a cabo minería de datos.

A grandes rasgos se pueden considerar que funcionan como un conjunto de elementos simples de procesamiento llamados nodos o neuronas conectadas entre sí por conexiones que tienen un valor numérico modificable llamado peso. La actividad que una unidad de procesamiento o neurona artificial realiza en un sistema de este tipo es simple. Normalmente, consiste en sumar los valores de las entradas (inputs) que recibe de otras unidades conectadas a ella, comparar esta cantidad con el valor umbral, y si lo iguala o supera, envía una salida (output) a las unidades a las que esté conectada. Tanto las entradas que la unidad recibe como las salidas que envía dependen a su vez del peso o fuerza de las conexiones por las cuales se realizan dichas operaciones.

El objetivo es encontrar la combinación que mejor se ajusta entrenando a la red neuronal. Este entrenamiento, aprendizaje, es la parte crucial de la RNA, ya que nos marcará la precisión del algoritmo. Hay dos tipos de aprendizajes en RNA que son: supervisado y no supervisado.

- Métodos basados en casos y vecindad

Estos métodos se basan en que la predicción se basa fundamentalmente en el conjunto de ejemplos vecinos al dato que se pretende procesar. O dicho de otra forma, de la distancia entre cada ejemplo y el dato en cuestión.

Lo que se hace es aprender de ejemplos de casos conocidos, y en base a estos casos, se toma una decisión sobre nuevos casos. Para ello hay que definir qué se entiende por similitud o distancia. De hecho, se pueden definir las distancia entre dos vectores, puntos o instancias x e y de dimensión n de muy distintas formas:

  - Distancia Euclídea
  - Distancia de Manhattan
  - Distancia de Chebychev
  - Distancia del coseno
  - Distancia de Mahalanobis

Pero el concepto de distancia también se puede utilizar cuando los ejemplos están formados por atributos nominales, algo que es habitual en minería de datos. Por ejemplo, para los atributos nominales se suele utilizar la función delta, es decir, $\delta(a,b) = 1$ si y sólo si $a=b$ y $\delta(a,b)= 0$ en caso contrario.

Dentro de los métodos basados en casos y vecindad destacan:

- Agrupamiento difuso o *Fuzzy Clustering*

  Los algoritmos de clustering difuso permiten que un elemento pertenezca a más de un grupo o clúster mediante un grado de pertenencia (Beca, 2007). Se trata de una de las principales técnicas de algoritmos no supervisados y se encuentra englobado dentro de los métodos basados en casos y vecindad. El algoritmo de Fuzzy C-Means es el más utilizado para realizar clustering difuso. Dicho algoritmo permite encontrar un conjunto de prototipos representativos de cada clúster y los grados de pertenencia a cada dato. Este tipo de algoritmo es muy utilizado para tareas de segmentación y clasificación.

  Fuzzy C-Mean fue propuesto por Dunn en 1973 y fue modificado por Bezdek en 1981. Posee dos puntos fundamentales que son (Thomas y Nashipudimath, 2012) :

  - El cálculo del centro de los clústers y el uso de la distancia euclídea para la asignación de los centros.

  - El algoritmo asigna a cada punto un valor que va de 0 a 1 que indica el grado de pertenencia a un determinado cluster. Cuanto más cercano a 1 mayor grado de pertenencia tiene el punto al cluster, mientras que cuanto más cercano esté a 0 menos grado de pertenencia tiene al cluster. Además provee un parametro de fuzzificación que puede tomar que va desde 1 a n, e indica el grado de borrosidad en los clústeres.

  El último punto viene a decir que el algoritmo permite a un determinado punto pertenecer a varios clústeres a la vez, eso sí con un grado de pertenencia distinto para cada uno de ellos.

  Por otro lado, este método presenta las siguientes desventajas (Thomas y Nashipudimath, 2012):

  - El método tiende a dar un alto grado de pertenencia a puntos atípicos, por lo que este algoritmo no es bueno en el manejo de puntos atípicos.

  - Para un clúster dado el grado de pertenencia de un punto depende directamente del grado de pertenencia a otros centros de clústers, lo que puede llevar a resultados poco óptimos.

  - Este método presenta también problemas a la hora de manejar grandes dimensiones de conjunto de datos.

  Además de los puntos anteriormente descritos, cuando un punto es equidistante al

centro de dos clústeres se le otorga igual grado de pertenencia a ambos clústers, cuando en realidad se le debería asignar un valor muy bajo por no decir nulo. Para solucionar este problema Krishnapuram y Keller propusieron un nuevo algoritmo conocido como C-Means (PCM).

El algoritmo PCM ayuda a identificar a aquellos puntos atípicos (*noise points*) otorgándoles un valor muy bajo de pertenencia a cualquier clúster. Sin embargo, es bastante sensible a una buena inicialización, de no ser así puede llegar a dar a lugar clústeres coincidentes.

- KNN o Vecinos más próximos

  El método de los vecinos más próximos o KNN es un método utilizado como clasificación de objetos, aunque también se puede utilizar para regresión. La idea principal de este algoritmo es clasificar a un determinado objeto con la clase mas frecuente de sus K vecinos más próximos (Moujahid y col., 2000).

  Para clasificar al nuevo objeto se tendrá en cuenta el número de vecinos más próximos dependiendo del valor que se escoja para k. Por ejemplo, si k = 1, se asignará el valor del vecino más próximo. Si se elige un valor de K muy pequeño entonces el método será muy sensible al ruido, por otro lado si se elige un valor de K muy elevado los puntos con los que que se compare podrán pertenecer a otras clases. Por tanto, el valor ideal de K debe ser aquel que sea lo suficientemente grande para evitar el ruido, pero debe ser pequeño en comparación con el número de puntos.

  Este tipo de algoritmo es de los conocidos como lazy o perezoso, ya que durante la fase del entrenamiento no construye ningún modelo, sino que simplemente guarda los ejemplos. Una vez que llegan los elementos a clasificar comparando su similitud con los ejemplos es cuando realiza la clasificación. Asimismo, se dice que es local, ya que asume que la clase de un dato depende sólo de los k vecinos más cercanos.

  Entre las principales limitaciones de este algoritmo se encuentran:

  - Tiene una alta sensibilidad a los atributos que son irrelevantes y al ruido.
  - Es lento si hay muchos datos de entrenamiento.
  - Depende de que la función de distancia sea la adecuada.
  - Le afecta la maldición de la dimensionalidad

  La maldición de la dimensionalidad (Bellman, 1961) se refiere a que cuando el volumen del espacio aumenta debido a un incremento de ésta, los datos disponibles se vuelven dispersos provocando complicaciones a cualquier método que use términos estadísticos. Es decir, el método KNN funciona correctamente incluso con una gran cantidad de datos pero siempre y cuando la dimensionalidad no sea alta.

- Algoritmos evolutivos

Los algoritmos evolutivos son métodos que se basan en los postulados de la evolución biológica para conseguir soluciones óptimas (Hidalgo y Cervigón, 2004). Estos algoritmos

son de gran utilidad con espacios de búsquedas extensos y no lineales, en donde otros métodos no son capaces de encontrar soluciones en un tiempo razonable.

Estos algoritmos trabajan con una población de individuos que representan las soluciones candidatas a un problema. Tras someter a esta población a determinadas transformaciones y un proceso de selección, se queda con aquellas soluciones mejores. Cada ciclo de transformación y selección constituye una generación, por lo que después de n generaciones queda una sola solución que es la más óptima.

- Arboles de decisión

Un árbol de decisión es un conjunto de condiciones organizadas en una estructura jerárquica, de tal manera que la decisión final se puede determinar siguiendo las condiciones que se cumplen desde la raíz del árbol hasta alguna de sus hojas. El conocimiento obtenido durante el proceso de aprendizaje inductivo se representa mediante un árbol (Barrientos y col., 2009).

Un árbol gráficamente se representa por un conjunto de nodos, hojas y ramas. Se basa en la aplicación de un conjunto de reglas SI-ENTONCES, utilizando funciones lógicas que llevan a disyunciones de posibles resultados (Calancha, 2011).

Entre las principales características es que las diferentes opciones posibles a partir de una determinada condición son excluyentes, por lo que siguiendo el árbol se llega a una sola solución.

Los valores que pueden tomar las entradas/salidas son discretos o continuos. Cuando son discretos se denomina clasificación y cuando son continuos se denomina regresión.

Un aspecto esencial en los algoritmos basados en árboles de decisión es la elección del mejor criterio para la división de los datos, por lo que para ello se han desarrollado diferentes algoritmos que tratan de buscar el criterio ideal. Entre los algoritmos que destacan están ID3 y C4.5. Ambos algoritmos generan árboles de decisión a partir de datos de entrenamiento.

Quinlan propuso en 1986 el algoritmo denominado ID3, el cual se basa en la entropía para elegir el mejor atributo dentro de los existentes para crear el árbol. Una vez elegido el atributo se realiza una partición de los datos basándose en dicho atributo que representa el mejor candidato para ramificar el árbol, después se sigue desarrollando el árbol repitiendo este proceso (Martinez y col., 2013). Este criterio seleccionado favorece a los atributos con muchos valores lo que no necesariamente es más adecuado para construir el árbol de decisión.

Posteriormente, Quinlan propuso en 1993 una mejora de este algoritmo al que denominó C4.5 (Quinlan, 1993). Este algoritmo se basa en la utilización del criterio de ratio de ganancia o *gain ratio* a la hora de escoger cada nodo en el árbol (Sharma y col., 2013). De esta forma, se consigue evitar que las variables con mayor número valores salgan beneficiadas en la selección. Además, el algoritmo incorpora una poda basada en un test de hipótesis que trata de responder a la pregunta de si merece la pena expandir una determinada rama o no.

La ganancia de información se define según la siguiente expresión:

$$Gan\,Inf\,(S\,,A) = Entropia\,(S) - \sum_{(v \in V(A))} \frac{|S_v|}{|S|} \quad (1)$$

donde S es una colección de objetos, A son los atributos de los objetos y V(A) es el conjunto de valores que A puede tomar. La entropía por su parte se define como:

$$Entropia\,(S) = \sum_i p_i \log_2 p_i \quad (2)$$

Donde S es una colección de objetos, p_i es la probabilidad de los posibles valores e i son las posibles respuestas de los objetos.

Entre las ventajas de los árboles de decisión destacan (Santa y Veloza, 2013):

- Facilita la interpretación de la decisión adoptada.
- Proporciona un alto grado de comprensión del conocimiento utilizado en la toma de decisiones.
- Reduce el número de variables independientes.
- Explica el comportamiento respecto a una determinada tarea de decisión.
- Permite la clasificación de nuevos casos siempre y cuando no existan modificaciones sustanciales en las condiciones bajos las cuales se generaron los ejemplos que sirvieron para su construcción.

- Métodos Bayesianos

Los Métodos Bayesianos hacen un uso explícito de la teoría de la probabilidad para cuantificar la incertidumbre. Estos modelos admiten tanto un uso descriptivo como predictivo. En lo que al modelo descriptivo se refiere, estos métodos facilitan el descubrimiento de relaciones de dependencia y/o relevancia de sus variables. Por otro lado, desde el punto de vista predictivo suelen utilizarse como clasificadores.

Hay dos razones principales para considerar estos métodos de gran interés dentro de la minería de datos (Mitchell, 1997):

- Son un método práctico para realizar inferencias a partir de los datos, induciendo modelos probabilísticos que después serán usados para razonar sobre nuevos valores observados. Además, permiten calcular de forma explícita la probabilidad asociada a cada una de las hipótesis posibles, lo que constituye una gran ventaja sobre otras técnicas.

- Facilitan un marco de trabajo útil para la comprensión y análisis de numerosas técnicas de aprendizaje y minería de datos que no trabajan explícitamente con probabilidades.

Entre las desventajas de estos métodos se encuentra el elevado coste computacional de ponerlo en práctica. Es por ello, que se han realizado suposiciones para poder reducir la

complejidad de estos modelos. Sin embargo, y pese a las simplificaciones algunos de estos modelos, como por ejemplo el clasificador de NB, son realmente competitivos y ofrecen mejores resultados que otras técnicas más complejas. Además, la aparición de las Redes Bayesianas han conseguido simplificar el coste computacional del modelo probabilístico sin pérdida de información por parte del mismo.

Entre estos métodos destacan:

- Redes Bayesianas

  Las Redes Bayesianas (Rbs) son un modelo probabilístico que relaciona un conjunto de variables aleatorias mediante un grafo dirigido. Son redes gráficas sin ciclos en el que se representan variables aleatorias (Rivera, 2011). En las relaciones de independencia/dependencia de las variables que componen el modelo es donde se articula el conocimiento. Además, de forma numérica se expresa el peso de estas relaciones cualitativas, para lo que se emplean relaciones de probabilidad.

  El hecho de usar una representación gráfica para la especificación del modelo hace de las Rbs una herramienta muy atractiva en su uso como representación del conocimiento, aspecto muy importante en la minería de datos. Estas redes son utilizadas en diversas áreas de aplicación como por ejemplo medicina (Beinlinch y col., 1989), ciencia (Breese y Blake, 1995) y economía (Hernández y col., 2004).

- Clasificador NB

  El clasificador de NB es el modelo con redes de base bayesianas más simple. Se encuentra fundamentado en el teorema de Bayes con algunas hipótesis que lo simplifican. De forma resumida se podría decir que este clasificador considera que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de cualquier otra característica.

Entre las principales características de los métodos bayesianos destacan (Malagón, 2003):

- Cada ejemplo observado va a modificar la probabilidad de que la hipótesis formulada sea correcta, reduciéndola o incrementándola.

- Estos métodos son robustos al posible ruido presente en los ejemplos de entrenamiento, e incluso a la posibilidad de tener datos erróneos en los mismos.

- Los métodos bayesianos permiten tener en cuenta en la predicción de la hipótesis el conocimiento a priori o conocimiento del dominio en forma de probabilidades.

2.4.3 Aplicaciones de la Minería de datos

La minería de datos abarca una gran variedad de aplicaciones donde están involucradas tanto empresas como organismos de investigación, gubernamentales, universitarios, etc. A continuación

se listan algunos ejemplos de los principales ámbitos de aplicación (Vallejos, 2006):

- Ámbito de la empresa

    - Detección de fraudes en las tarjetas de créditos. En Estados Unidos se desarrolló el *Falcon Fraud Manager,* el cual, es un sistema inteligente que examina transacciones, propietarios de tarjetas y datos financieros para detectar y mitigar fraudes. Este sistema está permitiendo ahorrar más de seiscientos millones de dólares americanos cada año.

    - Conocer por qué se dan de baja los clientes en una compañía de telefonía móvil. Este estudio lo realizó una operadora española que tenía como principales objetivos: saber qué perfil tenían los clientes que se daban de baja y la predicción del comportamiento de sus nuevos clientes. En base a este conocimiento adquirido la compañía aplicó cambios de cara a mantener e incrementar la cartera de clientes.

    - Predecir la audiciencia televisiva. La British Broadcasting Corporation (BBC) de Reino Unido emplea un sistema basado en redes neuronales y árboles de decisión aplicados a datos históricos que posee la cadena para ayudar a determinar la audiencia de un determinado programa y su mejor momento de emisión.

- Ámbito de la Universidad

    - Conocer la relación entre el trabajo de los recién titulados en la universidad y los estudios cursados. En Méjico se llevó a cabo un estudio acerca de los recién titulados de la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Chihuahua II. Lo que se pretendía conocer era si el plan de estudios facilitaba la inserción laboral a los que acababan de terminar la carrera. Dicho estudio utilizaba las variables de sexo, edad, escuela de procedencia, nivel económico del estudiante ….

- Ámbito Científico

    - Proyecto SKYCAT. El Second Palomar Observatory Sky Survey tras recopilar más de tres terabytes en imágenes, digitalizó 3000 fotografías con una resolución de 16 bits por píxel con el objetivo de formar un catálogo con todos esos objetos.

      El sistema Sky Image Cataloging and Analysis Tool (SKYCAT) se basa en técnicas de agrupamiento y árboles de decisión para poder catalogar objetos tales como: estrellas, planetas, sistemas, galaxias … Estos datos permiten a los astrofísicos mejorar la comprensión del universo.

Por otro lado, entre los tipos de datos a los que se les pueden aplicar técnicas de minería de datos destacan (Zaïne, 1999):

- Archivos de texto. Como es el texto simple o binario

- Bases de datos relacionales. Poseen tablas que contienen entidades o atributos y se hallan relacionadas con otras tablas.

- Base de datos transaccionales. Son un conjunto de datos que representan transacciones.

- Multimedia. Como pueden ser vídeos, imágenes o audios. Se pueden almacenar en base de datos relacionales u orientados a objetos.

- Series de tiempo. Son flujos continuos de información como pueden ser actividades en sitios web.

- Internet. Es el repositorio de datos más heterogéneo y dinámico. La información reside en textos, audios, etc.

2.4.4 Algoritmo clasificador bayesiano ingenuo o NB

Desde el punto de vista de las redes bayesianas se trata de uno de los casos más simples. Su estructura de red es fija y sólo es necesario conocer las probabilidades. La hipótesis de independencia asumida por este clasificador da un lugar a un modelo gráfico probabilístico en el que existe un único nodo raíz, y en el que todos los atributos son nodos hojas que tienen como único padre tiene al nodo raíz. En la Ilustración 2 se puede observar esta característica:

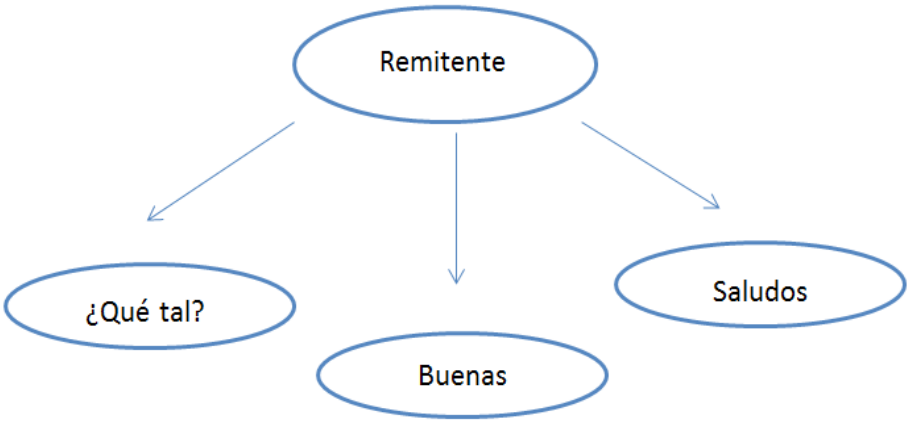

Ilustración 2: Grafo de Naive Bayes

Donde Remitente es el nodo raíz, y "¿Qué tal?", "Buenas" y "Saludos" son los nodos hojas. Esta red se modela sobre el hecho de que una carta puede ser escrita por un remitente que comienza diciendo: "¿Qué tal?", "Buenas" o "Saludos" (por su puesto, una carta puede empezar de muchas más formas) teniendo una determinada probabilidad de ocurrencia cada palabra dado el remitente.

En las Ilustraciones 3 y 4 se muestran dos ejemplos con dos remitentes distintos: Juan y Manuel donde cada uno de ellos tiene una probabilidad de ocurrencia diferente. En dicho ejemplo, se muestra que Juan es más probable que sea un remitente que Manuel, y por otro lado indica que la palabra más probable con la que comienza un correo Juan es "Buenas", mientras que las palabras que con más probabilidad puede empezar un correo Manuel son "Qué tal" y "Buenas".

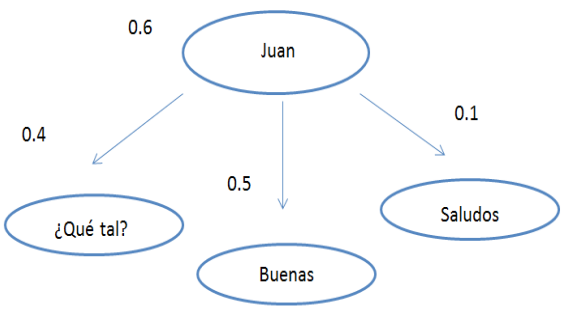

Ilustración 3: Ejemplo grafo Juan

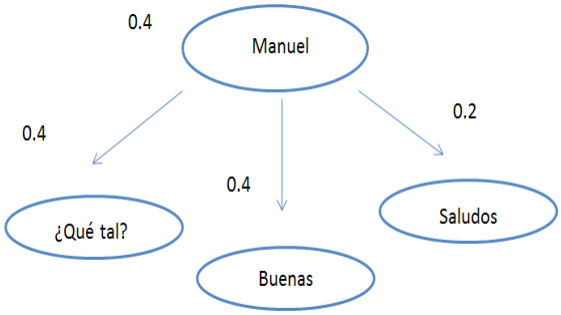

Ilustración 4: Ejemplo grafo Manuel

Este modelo descrito se dice que es de NB ya que la forma del grafo indica que existe independencia entre cada posible palabra una vez que se conoce al remitente.

A pesar de la larga tradición que siempre ha tenido este algoritmo dentro del mundo del reconocimiento de patrones (Duda y Hart, 1973), no fue hasta finales de los años ochenta cuando aparece por primera vez dentro de la literatura relacionada con el aprendizaje automático (Cestnik y col., 1987). Ha sido utilizado con bastante éxito dentro del aprendizaje automático a la hora de clasificar textos. De hecho, una de las principales aplicaciones es la de reconocer emails que son spam y separarlos de aquellos emails que sí son deseados por el receptor.

Desde un punto de matemático, este algoritmo se basa en el teorema de Bayes, el cual da nombre a los métodos bayesianos, y fue publicado en 1763 y dice:

Sea $\{A_1,....,A_i,.....,A_n\}$ un conjunto de sucesos mutuamente excluyentes y exhaustivos, y tales que la probabilidad de cada uno de ellos es distinta de cero. Sea B un suceso cualquiera del que se conocen las probabilidades condicionales $P(B|A_i)$. Entonces, la probabilidad $P(A_i|B)$ viene dada por la expresión:

$$P(A_i|B)=\frac{P(B|A_i)P(A_i)}{P(B)} \qquad (3)$$

donde $P(A_i)$ son las probabilidades a priori, $P(B|A_i)$ es la probabilidad de B en la hipótesis A_i y $P(A_i|B)$ son las probabilidades a posteriori.

El anterior teorema vincula la probabilidad de A dado B con la probabilidad de B dado A. Dicho de otra forma, sabiendo la probabilidad de tener por ejemplo un dolor de cabeza dado que se tiene gripe, se podría saber (si se tiene algún dato más) la probabilidad de tener gripe si se tiene dolor de cabeza.

El clasificador de NB introduce la simplificación de considerar que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de cualquier otra característica (Mitchell, 2005). Es decir, introduce la independencia condicional de las variables predictoras dada la variable clase.

Dentro de las ventajas de este clasificador destacan:

- Ventajas

    - Es un algoritmo sencillo de implementar y entender (Cheng y Greiner 1999)
    - Es bastante eficiente, robusto y tiene una alta precisión (Cheng y Greiner 1999).
    - Posee un fundamento matemático sólido.
    - Puede discriminar atributos importantes de lo que no lo son.
    - En dominios naturales ha demostrado comportarse tan bien como otros métodos más sofisticados. Llega incluso a ser comparable con los árboles de decisión y redes de neuronas.

- Desventajas

    - El asumir la independencia de sus atributos hace que la calidad de los resultados obtenidos decaiga en el caso de que los atributos se encuentren relacionados entre sí. (Martínez-Arroyo, 2006)
    - En el caso de que hay variables continuas es necesario discretizarlas previamente.
    - Coste computacional alto.
    - Exige conocimiento a priori.
    - Es necesario tener un conjunto de entrenamiento con un volumen medio o alto.

2.5 Métodos ponderados de NB

El clasificador de NB es extremadamente simple y aún así llega a obtener muy buenas aproximaciones como clasificador (Lewis, 1998). Estos buenos resultados se producen a pesar de la hipótesis simplificadora de independencia de atributos. De hecho, llega a obtener resultados comparables a los obtenidos con otros clasificadores más complejos como son los árboles de decisión (Zaidi y col., 2013).

Sin embargo, en la práctica la independencia entre atributos suele violarse, por lo que los resultados obtenidos están por debajo de todo lo óptimo que deberían ser. Debido a esto, se han generado toda una gran cantidad de trabajos que han tenido como objetivo tratar de paliar este problema.

Las diferentes propuestas de mitigación que se han realizado a lo largo del tiempo pueden clasificarse de acuerdo a dos categorías (Zaidi y col., 2013):

- Métodos semi-Naive Bayes. Esto métodos se caracterizan por relajar la hipótesis de independencia.

- Métodos de ponderación de atributos. En estos métodos se trata de establecer pesos a cada uno de los atributos "*ya que no han recibido toda la atención merecida*" (Zaidi y col., 2013).

En lo que a los métodos semi-Naive Bayes se refiere muchos investigadores han extendido el algoritmo original junto con un pequeño número adicional de dependencias entre atributos con la idea de mejorar los resultados (Zheng y Webb, 2000). Ejemplos de este tipo podemos encontrar es el *Tree-Augmented Naive Bayes* (Friedman y col., 1997) y *Average n-Dependence Estimators*

(Webb y col., 2011). Normalmente estos tipos de algoritmos limitan la estructura de red de dependencia a estructuras de tipo árbol, o más en general a estructuras de tipo gráfico que pueden ser aprendidas.

Es de destacar que el uso de selección de atributos y ponderación de atributos estableciendo pesos se encuentran estrechamente relacionados. Ya que, por ejemplo, la selección de atributos previos a la aplicación de NB sería similar a un método que establezca pesos a los atributos y asigne el valor 0 a aquellos que considere menos relevantes para la predicción.

En 1994 Langley y Sage propusieron el clasificador selectivo de Bayes, utilizando selección de características o atributos con la idea de eliminar aquellos atributos redundantes. La técnica se basaba en la búsqueda a través del espacio completo de todos los subconjuntos de atributos. El algoritmo se basa en un proceso iterativo de ir añadiendo un atributo a un conjunto de atributos (que se inicializa vacío) e ir midiendo su capacidad de predicción como clasificador. El algoritmo termina cuando la adición de cualquier otro atributo reduce la precisión del resultado. El orden en el que han sido añadidos los atributos al conjunto determina lo significativos que son para la clasificación. Basado en este método surgió el algoritmo de *Correlation-Based Feature Selection* que utilizaba una medida de correlación como métrica para determinar la relevancia de los atributos (Hall, 2000).

Es de destacar la gran tendencia que ha habido en la utilización de árboles de decisión para mejorar algoritmos de aprendizaje automático, y entre ellos, el algoritmo de NB. En esta línea destaca la mejora propuesta por los investigadores Chotirat Ratanamahatana y Dimitrios Gunopulos del departamento de ciencias de la computación de la Universidad de California en el 2003, proponiendo una combinación entre los métodos de NB y el algoritmo C4.5 de los árboles decisión (Ratanamahatana y Gunopulos, 2003). En dicho trabajo se demuestra que el método de NB funciona mejor si se utilizan sólo aquellas características que el algoritmo C4.5 selecciona para la construcción de árboles de decisión. Sólo aquellos atributos que aparecen en los 3 niveles más altos de un árbol de decisión son utilizados para la construcción del modelo de NB. Dicha mejora es evidente en aquellos dominios donde C4.5 funciona mejor que NB, llegando incluso a mejorar los resultados del propio algoritmo C4.5.

Desde el punto de vista de la ponderación de los pesos de los atributos, uno de los primeros trabajos relacionados fue realizado por Hilden y Bjerregaard en 1976 (Hilden y Bjerregaard, 1976). Esta propuesta usa solo un peso, por lo que no realiza estrictamente una ponderación de atributos, sino trata más bien de reducir los efectos de la hipótesis de independencia cuando no se cumple. Por ejemplo, establecer el peso de los atributos a 1 es una buena idea pero sólo cuando la hipótesis de independencia se cumple en gran medida. Variando el peso a otros valores entre 0-1 para problemas en los que sí existían dependencias entre variables se obtienen mejores resultados.

Posteriormente, Harry Zhang y Shengli Sheng de la Universidad de New Brunswick y la Universidad de Western Ontario presentaron el estudio científico *Learning Weighted Naive Bayes with Accurate Ranking* (Zhang y Sheng, 2004). En dicho trabajo se estudia cual es la forma de obtener los pesos de los diferentes atributos de forma más precisa para obtener resultados más ajustados a la realidad.

Para dicho fin exploran varios métodos: el método de la ganancia de ratio, el método de *Hill -Climbing* y el método de *Markov Chain Monte Carlo*, así como combinaciones entre dichos métodos. Para medir la calidad de los resultados obtenidos se basaron en el AUC[3].

3 AUC significa "área bajo la curva ROC". La curva ROC es un gráfico que demuestra el rendimiento de un modelo

Dicho estudio llega a la conclusión que el uso combinado del método de la ganancia de ratio para el cálculo de los pesos de los atributos, para posteriormente modificarlo usando el método de *Hill-Climbing* obtiene los mejores resultados. Los resultados obtenidos mejoran los resultados que se obtienen para el método clásico.

Posteriormente han surgido otros métodos como el propuesto por Hall (2007) donde el peso asignado a cada atributo es inversamente proporcional a la profundidad a la que se encuentran en un árbol de decisión, el cual, en cierta forma es bastante parecido al propuesto por Chotirat Ratanamahatana y Dimitrios Gunopulos.

Por último, es de destacar el método propuesto en Ferreira y colaboradores en el 2001, al utilizar discretización basada en la entropía para atributos numéricos y asignar un peso a cada partición del atributo de forma proporcional a su capacidad predictiva de clasificación.

2.6 Resumen del Análisis Bibliográfico

La búsqueda de conocimiento útil y no obvio dentro de grandes volúmenes de datos es lo que se conoce como minería de datos. En el actual paradigma donde el conocimiento se sitúa como un valor muy cotizado no sólo por organizaciones empresariales, sino además dentro del mundo académico y científico, es de vital importancia la obtención y/o mejora de los actuales algoritmos de extracción de patrones de conocimiento a partir de grandes volúmenes de datos.

Aunque estas mejoras sólo sean significativas en un determinado rango de datos, o sólo mejoren un porcentaje relativamente pequeño con respecto a otros algoritmos previamente existentes, pueden dar lugar a aplicaciones en la vida real que sean importantes.

Desde el punto de vista de una organización empresarial es de gran importancia su aplicación dentro de *Bussiness Intelligence*. Muchas organizaciones sólo se guían por los resultados económicos, lo cual equivale a conducir guiado solamente por el espejo retrovisor. Dicho en otras palabras, cuando se presenta un mal balance económico ya es tarde porque no se ha podido evitar un problema que ya estaba dando signos de existir previamente. Un ejemplo sería la de una empresa de seguros de automoción que no es capaz de perfilar correctamente aquellos clientes con más riesgo de siniestro, o el de una empresa de telefonía que no es capaz de detectar a tiempo los cambios en el mercado relativo a las preferencias del usuario.

Otras aplicaciones muy importantes son aquellas relativas a la investigación científica. Por ejemplo, detectar qué impacto tiene en el medio ambiente el uso de un determinado insecticida, o conocer los efectos o causas de una determinada enfermedad. Por supuesto, existen muchas más disciplinas donde estos algoritmos pueden aplicarse como son la robótica o el reconocimiento facial.

El proceso de extracción del conocimiento se realiza en diversas etapas siendo la minería de datos una de ellas: fase de selección de datos, fase de preprocesamiento, fase de transformación, fase de minería de datos y fase de interpretación y evaluación.

Tal y como se ha visto hay una amplia variedad de tareas y métodos dentro de la minería de datos. Entre las tareas existentes destacan la clasificación, categorización y regresión. Por otro lado, entre

de clasificación en todos los umbrales de clasificación. La curva representa dos parámetros: Tasa de verdaderos positivos y tasa de falsos positivos.

las principales técnicas destacan: redes neuronales, SVM, método de los vecinos más próximos, árboles de decisión, agrupamiento difuso y los métodos bayesianos. Estas técnicas puedes ser de aprendizaje no supervisado, supervisado y semi-supervisado. Las diferencias entre una u otra técnica radica en disponer de un conocimientoa priori a la hora de tratar los datos. Como es de esperar aquellos algoritmos supervisados ofrecen mejores resultados dado que se dispone de un conocimiento previo.

Los métodos bayesianos destacan porque admiten tanto un uso predictivo como descriptivo de una tarea. Es decir, se pueden utilizar para realizar predicciones así como para descubrir relaciones de interés dentro de las variables de un problema. A pesar de lo complejo que puede ser trabajar con probabilidades, dentro de los métodos bayesianos hay simplificaciones como es el caso del algoritmo clasificador de NB que facilitan bastante los cálculos.

El clasificador de NB si bien es sencillo de entender e implementar tiene algunas desventajas importantes derivadas de la hipótesis de independencia condicional. El hecho de que existan relaciones de dependencia entre sus variables hace que este método pierda eficacia en sus resultados.

Para corregir dichas deficiencias se han realizado diversas propuestas, las cuales pueden dividirse en dos tipos principales: métodos de semi-Naive Bayes, basados en relajar la hipótesis de independencia, y ponderación de atributos, basados en introducir para cada uno de los atributo un peso.

El objetivo de este Trabajo de Fin de Máster es corregir esta deficiencia que presenta el algoritmo supervisado de NB cuando existen relaciones de dependencia entre las variables que componen el problema. Para ello, se propondrá un método ponderado de este algoritmo. De esta forma se mejora uno de los algoritmos que presentan mayores ventajas tales como que puede tener un uso descriptivo y predictivo, y que es sencillo de entender.

Para poner en valor la mejora obtenida se comparan los resultados obtenidos con otros algoritmos de relevancia tanto supervisados como no supervisados. Por parte de los algoritmos no supervisados se propone el algoritmo de agrupamiento difuso o *Fuzzy Clustering*, el cual permite encontrar un conjunto de prototipos representativos de cada clúster y los grados de pertenencia a cada dato. Por otro lado, los algoritmos supervisados con los que se comparan son: KNN o vecinos más próximos, el cual es un algoritmo clásico donde cada nuevo objeto se considerará en base al número K de vecinos más próximos al mismo; y SVM, el cual es una algoritmo relativamente reciente con el que se han obtenido resultados muy óptimos. Dicho algoritmo busca un hiperplano que separe de una forma óptima a los puntos de una clase y de la otra. Por último, se comparan los resultados obtenidos con otro algoritmo de NB ponderado que utiliza el algoritmo C4.5 para seleccionar los atributos a aplicar.

3. Solución Propuesta

Tal y como se ha establecido en capítulos previos, el objetivo de este Trabajo de Fin de Máster es encontrar un clasificador de NB Ponderado (en adelante NBP) que mejore las limitaciones que presenta el original. El clasificador de NB se basa en el teorema de Bayes pero con la hipótesis simplificadora de que las variables que intervienen en el resultado final son independientes entre sí.

En el presente capítulo se presenta formalmente el clasificador de NB original y después se presenta el método ponderado. Dicho método ponderado trata de mejorar dicho clasificador corrigiendo al menos de forma parcial el error cometido al considerar que todas las variables que intervienen son independientes entre sí. Asimismo, se expone el pseudocódigo resaltando de esta manera las características principales del método propuesto.

Por último, se expone un método que combina los algoritmos C4.5 y de NB utilizando el primero para seleccionar aquellos atributos a tener en cuenta por el clasificador de NB.

3.1 Formalismo matemático del clasificador de NB

Dado un conjunto de datos tanto de entrenamiento como de test, y dada la función f(x) que clasifique a dichos ejemplos, la idea de usar el teorema de Bayes radica en estimar las probabilidades a posteriori consistente con el conjunto de datos de entrenamiento para escoger la hipótesis más probable.

Dado un ejemplo x representado por una serie de valores. Si la descripción del ejemplo viene dado por los valores $<a_1, a_2, a_3, \ldots a_n>$, el clasificador de NB estimará como hipótesis más probable aquella que cumpla la máxima probabilidad a posteriori V_{map} :

$$v_{MAP} = arg \max_{vj \in V} P(v_j | a_1, a_2, a_3 \ldots, a_n) \tag{4}$$

donde v_j es el valor de la función de clasificación f(x) en el conjunto finito V. Aplicando el teorema de Bayes la anterior expresión queda:

$$= arg \max_{vj \in V} \frac{(P(a_1, a_2, a_3 \ldots a_n | v_j) p(v_j))}{P(a_1, a_{2,}, a_3 \ldots a_n)} \tag{5}$$

En realidad, solo importa el numerador ya que el denominador no depende de v_j y además los valores $a_1, a_2 \ldots$ son constantes, por lo que:

$$= arg \max_{vj \in V} P(a_1, a_2, a_3 \ldots a_n | v_j) p(v_j) \tag{6}$$

Donde $p(v_j)$ puede ser estimado contando con la frecuencia con la que ocurre cada valor v_j en el conjunto de entrenamiento y dividiéndolo por el número total de ejemplos que forman este conjunto. Para estimar el término $P(a_{1,} a_{2,} a_3 \ldots a_n | v_j)$ que son las veces que para cada categoría aparecen los valores del ejemplo x, es necesario recorrer todo el conjunto de entrenamiento. Este cálculo es muy complicado cuando hay un número muy elevado de ejemplos, por lo que se recurre a

la hipótesis de independencia condicional.

La anterior suposición de que los atributos son independientes entre sí nos permiten factorizar la anterior expresión que queda de la forma:

$$P(a_1, a_2, a_3 \ldots a_n | v_j) = \prod_i P(a_i | v_j) \tag{7}$$

Por último, sustituyendo este nuevo término en la expresión la máxima probabilidad a posteriori, la aproximación del clasificador de Bayes Naive queda:

$$v_{nb} = arg\max_{vj \in V} P(v_j) \prod_i P(a_i | v_j) \tag{8}$$

donde as probabilidades $P(a_i | v_j)$ resultan mucho más sencillas de estimar que las $P(a_{1,} a_{2,} a_3 \ldots a_n)$.

3.2 Clasificador NBP

El método ponderado que se propone en este Trabajo de Fin de Máster se trata básicamente de realizar un ajuste de las probabilidades $P(a_i | v_j)$ obtenidas en el apartado anterior. Dicho ajuste no se realiza al azar sino que sigue una serie de criterios.

3.2.1 Hipótesis/criterios seguidos para el método propuesto

La idea de realizar un ajuste de dichas probabilidades originales ha sido extraída del trabajo *Learning Weighted Naive Bayes with Accurate Ranking* (Zhang y Sheng, 2004). En dicho trabajo, se defiende la ponderación de los diferentes atributos que participan en el problema a clasificar. La necesidad de ponderar los atributos surge de la hipótesis de independencia condicional, la cual, apenas se da en la realidad. Para ello se proponen diferentes métodos como son: Gain Ratio, Hill Climbing y Montecarlo.

En el método propuesto no se utilizarán ninguno de estos métodos pero nos sirve de idea a la hora de proponer otra serie de criterios que se enumeran a continuación:

1. Se ordenan los atributos $< a_1, a_2, a_3, \ldots a_n >$ en orden mayor a menor de ganancia de información $\text{GainInf}(a_i)$ a la hora de comenzar a ajustar las probabilidad de cada uno de ellos. Esta idea se basa en el algoritmo ID3 de los árboles de decisión.

2. Para cada uno de los atributos, la iteración consiste en variaciones del valor de los coeficientes de probabilidad original $P(a_i | v_j)$ del orden 1%, 2%, 3% y así sucesivamente.

3. Para cada uno de los atributos, se incrementa en un determinado % el coeficiente de probabilidad de una determinada probabilidad a priori $P(a_k | v_l) \cdot \text{Inc}(\%)$ y se reduce en el

mismo % para el resto de probabilidades a priori $\sum_{i \neq k\, j \neq l} P(a_i|v_j) \cdot \text{Desinc}(\%)$. De tal forma que si para un coeficiente de probabilidad se incrementa un 3%, la reducción en conjunto para el resto de coeficientes debe ser un 3%.

4. Se da más peso a aquellas probabilidades a priori que sean mayores $P'(a_i|v_j)$, y se reduce para el resto. Es decir, si un atributo puede tomar los valores "alto" o "bajo", y el valor "alto" está relacionado con el resultado de clasificación más probable y el valor "bajo" con el menos probable, se da más peso al coeficiente de probabilidad del valor "alto" y menor peso al coeficiente de probabilidad del valor "bajo".

5. Una vez que se han modificado los coeficientes de probabilidad de un atributo en concreto se vuelve aplicar el modelo obtenido a los datos. Si el resultado obtenido es mejor o igual que el que se tenía previamente, se continúan modificando los siguientes atributos. En caso contrario, se para el proceso y se recuperan con los coeficientes previos que daban mejor resultado. Para saber si el resultado obtenido es mejor o peor se comparan las máximas probabilidades a posteriori v'_{nb} obtenidas con el nuevo modelo con el resultado que se conoce previamente con los datos de prueba p_{nb} :

$$\sum_i F(p_{nb_i} - v'_{nb_i}) \qquad (9)$$

   Donde i toma los valores de cada caso a clasificar. El resultado de la función $F(p_{nb_i} - v'_{nb_i})$ es 0 si v'_{nb} y p_{nb} son coincidentes y toma el valor 1 si no coinciden. Por tanto, si el número que se obtiene del sumatorio es mayor tras la aplicación de un determinado modelo significa que la adecuación es peor.

6. Una vez que el punto anterior devuelve un modelo, se vuelve a aplicar la misma lógica pero variando el punto 4. La variación consiste en dar más peso al coeficiente de probabilidad con menos probabilidad a priori $P''(a_i|v_j)$. Es decir, volviendo al ejemplo anterior del atributo que puede tomar los valores "alto" o "bajo", se da menos peso al coeficiente de probabilidad asociado al valor "alto" y se le da más peso al coeficiente de probabilidad asociado al valor "bajo".

7. Una vez finalizado el modelo obtenido en el punto anterior, se vuelve a aplicar la misma lógica de los pasos del 1 al 5, pero modificando el punto 1. Dicha modificación consiste en ordenar los atributos $<a_1, a_2, a_3, \ldots a_n>$ en orden de menor a mayor de ganancia de información $\text{GainInf}(a_i)$

8. Por último, se evalúan los modelos obtenidos en los puntos 5, 6, y 7, y se escoge el que mejor resultado obtenga después de aplicarlos sobre el conjunto de datos.

3.2.2 Pseudocódigo del método propuesto

Los 7 pasos expuestos en el subapartado anterior se pueden expresar en el siguiente pseudocódigo:

```
Entero i, l y m;
Array atributos;
```

Array probabilidades a priori;

Inicio

leer (**atributos**); en orden de mayor a menor de ganancia de información.
leer (**probabilidades a priori**); en orden creciente de probabilidades a priori.

∀ atributo ← i donde i = 1, 2, 3, 4...
∀ probabilidad a priori ← l donde l = 1, 2, 3, 4

MIENTRAS m = 1, 2, 3... hasta modelo óptimo según $\sum_i F(p_{nb_i} - v'_{nb_i})$

datos ← NB; aplicar el modelo Naive Bayes

MIENTRAS i = 1, 2, 3 ... **máximo de atributos**

∀ l = 1, 2, 3 ... **máximo de probabilidades a priori**

ordenar $(P(a_i|v_l))$ de mayor a menor respecto probabilidad a priori

Si $P(a_i|v_l) >= \forall P(a_i'|v_l')$ Entonces

$P(a_i|v_l) \cdot$ m% ; Se incrementa

sino

$P(a_i|v_l) \cdot$ Desinc(m%) ; Se decrementa

fin si

datos ← NB'; aplicar el modelo Naive Bayes

Evaluar(**datos**) según $\sum_i F(p_{nb_i} - v'_{nb_i})$

Si es igual o mejor
entonces
SEGUIR_MIENTRAS i
SEGUIR_MIENTRAS m
sino
FIN_MIENTRAS i
FIN_MIENTRAS m
fin si

MIENTRAS m = 1, 2, 3... hasta modelo óptimo según $\sum_i F(p_{nb_i} - v'_{nb_i})$

datos ← NB; aplicar el modelo Naive Bayes

MIENTRAS i = 1, 2, 3 ... **máximo de atributos**

∀ l = 1, 2, 3 ... **máximo de probabilidades a priori**

ordenar $(P(a_i|v_l))$ de menor a mayor respecto probabilidad a priori

Si $P(a_i|v_l)$ <= $\forall P(a_i'|v_l')$ Entonces

$P(a_i|v_l)\cdot$ m% ; Se incrementa

sino

$P(a_i|v_l)\cdot$ Desinc(m%) ; Se decrementa

fin si

datos ← NB"; aplicar el modelo Naive Bayes

Evaluar(**datos**) según $\sum_i F(p_{nb}-v'_{nb})$

Si es igual o mejor
entonces
SEGUIR_MIENTRAS i
SEGUIR_MIENTRAS m
sino
FIN_MIENTRAS i
FIN_MIENTRAS m
fin si

leer (**atributos**); en orden menor a mayor de ganancia de información.
MIENTRAS m = 1, 2, 3... hasta modelo óptimo según $\sum_i F(p_{nb}-v'_{nb})$

datos ← NB; aplicar el modelo Naive Bayes

∀ atributo ← i donde i = 1, 2, 3, 4...

MIENTRAS i = 1, 2, 3 ... **máximo de atributos**

∀ l = 1, 2, 3 ... **máximo de probabilidades a priori**

ordenar $(P(a_i|v_l))$ de mayor a menor respecto probabilidad a priori

Si $P(a_i|v_l)$ >= $\forall P(a_i'|v_l')$ Entonces

$P(a_i|v_l)\cdot$ m% ; Se incrementa

sino

$P(a_i|v_l) \cdot$ Desinc(m%) ; Se decrementa

fin si

datos ← NB'''; aplicar el modelo Naive Bayes

Evaluar(**datos**) según $\sum_i F(p_{i\,nb} - v'_{i\,nb})$

Si es igual o mejor
entonces
SEGUIR_MIENTRAS i
SEGUIR_MIENTRAS m
sino
FIN_MIENTRAS i
FIN_MIENTRAS m
fin si

Si NB' < NB'' < NB''' o NB'' < NB' < NB'''
entonces
devolver (NB''')
si NB' < NB''' < NB'' o NB''' < NB' < NB''
entonces
devolver(NB'')
si NB'' < NB''' < NB' o NB''' < NB'' < NB'
entonces
devolver(NB')
fin si

Fin

3.3 Selección de características de NB utilizando árboles de decisión

La selección de características de NaiveBayes utilizando árboles de decisión (Ratanamahatana, Gunopulos, 2003) ha demostrado tener un mejor rendimiento comparado con el método de NB original y el algoritmo C4.5 utilizado en los árboles de decisión. Chotirat Ratanamahatana y Dimitros Gunopulos del departamento de ciencias de computación de la Universidad de California, desarrollaron este método donde combinaron ambos algoritmos de naturaleza diferente.

Este trabajo se basa en el hecho de que en dominios donde el algoritmo de NB presenta peores comportamientos debido a que los atributos están relacionados entre sí, el algoritmo C4.5 de árboles de decisión presenta mejores resultados. La idea principal es utilizar el algoritmo C4.5 para seleccionar a aquellos atributos que son más importantes (llegando en algunos casos a la mitad de los que originalmente existían) y posteriormente aplicar el algoritmo de NB utilizando solamente los atributos previamente seleccionados.

La combinación de ambos algoritmos ha llegado a tener incluso mejor comportamiento en aquellos dominios donde el algoritmo C4.5 mejora al de NB. La combinación propuesta de ambos

algoritmos tiene los siguientes pasos:

1. De los datos de entrenamiento se toma una muestra del 10%
2. Se ejecuta el algoritmo C4.5 sobre los datos obtenidos en el paso 1.
3. Se seleccionan los atributos que aparecen solamente en los 3 primeros niveles del árbol de decisión simplificado.
4. Se repite 5 veces los pasos (1 al 3)
5. Se forma una unión de todos los atributos de las 5 iteraciones.
6. Se ejecuta el modelo de NB sobre los datos de entrenamiento y de prueba utilizando sólo las características obtenidas en el paso 5.

4. Resultados obtenidos

4.1 Colecciones de datos utilizadas

En el presente Trabajo de Fin de Master se han utilizado 6 conjuntos de base de datos que se encuentran en la página web http://cml.ics.uci.edu/. Esta web pertenece a la Universidad de California en Irvine (en adelante UCI) y contiene bases de datos para el desarrollo de algoritmos relacionados con los sistemas inteligentes y de aprendizaje automático. De entre todas las colecciones se escogen las siguientes:

- **Colección 1**. Base de datos del servicio de transfusión de sangre del hospital de Hsin-Chu en Taiwan (I-Cheng Yeh, 2008). Estos datos han sido donados por el profesor I-Cheng Yeh del departamento de Gestión de la información de la Universidad de Chung-Hua. Se compone de 748 conjuntos de datos. Los atributos de los que se compone son:

    - *Recency*: Son los meses desde la última donación
    - *Frecuency*: Total de número de donaciones
    - *Monetary*: Total de sangre donada en centímetros cúbicos
    - *Time*: Meses desde la última donación
    - *Donated*: Una variable que puede tener dos valores 0 o 1 que indica si la persona donó sangre en marzo del 2007. El valor 0 significa que no lo hizo, y el valor 1 indica que sí.

- **Colección 2**. Base de datos de autentificación de billetes de banco.

    El propietario de estos datos es el profesor Volker Lohweg de la Universidad de Ciencias Aplicadas de Ostwestfalen-Lippe (Lohweg, 2012). Estos datos contienen información acerca de imágenes que fueron tomadas de billetes de bancos genuinos y falsos. Se compone de 1372 conjuntos de datos. Estos datos contienen cinco atributos:

    - *Variance of Wavelet Transformed image*. Atributo relacionado con la varianza de la transformada de wavelet de la imagen.
    - *Skewness of Wavelet Transformed image*. Atributo relacionado con la asimetría de la transformada de wavelet de la imagen.
    - *Curtosis of Wavelet Transformed image*. Atributo relacionado con la nitidez del pico de la transformada de wavelet de la imagen.
    - *Entropy of Image*. Atributo relacionado con la entropía de la imagen.
    - *Class*. Clasifica los billetes dependiendo de si son genuinos o falsos.

- **Colección 3**. Base de datos de la planta de iris

    Esta es quizás una de las colecciones más utilizadas para el reconocimiento de patrones y fue creada por R.A Fisher (Fisher, 1936). Se compone de 150 conjuntos de datos. Los atributos de los que se compone son:

- *Sepal length*: Longitud del sépalo
- *Sepal width*: Anchura del sépalo
- *Petal width*: Anchura del pétalo.
- *Petal length*: Longitud del pétalo.
- *Class*: Setosa, Versicolour y Virginica.

- **Colección 4**. Base de datos de Evaluación de coches

  Este conjunto de datos pertenece a los trabajos realizados por M. Bohanec y V. Rajkovic cuya finalidad era construir un modelo que evaluara los autos conforme a una serie de características (Bohanec y Rajkovic, 1997). Se compone de 1728 conjunto de datos. Los atributos de los que se compone son:

  - *Buying*: Precio de compra.
  - *Maint*: Precio del mantenimiento.
  - *Doors*: Número de puertas.
  - *Persons*: Nº de personas que puede llevar el coche.
  - *Lug_boot*: Tamaño del maletero
  - *Safety*: Seguridad del vehiculo.

- **Colección 5**. Conjunto de datos de Hayes-Roth

  Se trata de un conjunto de datos utilizado para estudios sociales creados por Barbara y Frederick Hayes-Roth (Hayes-Roth, 1989). Se compone de 132 conjuntos de datos. Los atributos son los siguientes:

  - *Name*. Nombre (atributo representado numéricamente para instancia)
  - *Hobby*. Afición (atributo representado numéricamente)
  - *Edad*. Edad (divido en rangos y se representa numéricamente)
  - *Educación*. Nivel de estudios (se representa numéricamente)
  - *Marital Status*. Estado civil (se representa numéricamente)
  - *Class*. Clase social (Se representa según categorías dependiendo de la clase social)

- **Colección 6**. Colecciones de datos de puentes de Pittsburg.

  Se trata de un conjunto de datos acerca de los puentes de Pittsburgh. Los creadores de esta base de datos son Yoram Reich & Steven J. Fenves del departamento de ingeniería civil y del centro de investigación en diseño de ingeniería de la Universidad de Caregie Mellon en Pittsburg (Reich y Feves, 1990). Inicialmente el número de variables de esta colección eran 13 pero para este trabajo se han reducido a 7. Se compone de 106 conjuntos de datos (inicialmente eran más). Los atributos son los siguientes:

  - *Purpose*. Propósito del puente.
  - *Lanes*. Tipo de Carril
  - *Clear-G*. Atributo no especificado.

- *T-or-D*. Indica si está cubierto
- *Material*. Tipo de material
- *Span*. Longitud del puente.
- *Class*. Indica el tipo de puente.

En la tabla 2 se describen las principales características de las colecciones anteriormente descritas:

Colección	Nº Filas	Nº atributos	Nº Result. Clasific.	Media	Mediana	Desv. Típica
1	748	4	2	0.02399705	0.02268229	0.02072269
2	1372	4	2	0.1544309	0.133703	0.1506597
3	150	4	3	0.6521269	0.6962069	0.3503707
4	1728	6	4	0.08762667	0.06017212	0.07276803
5	132	5	3	0.09100734	0.1516789	0.08307796
6	106	7	7	0.1217417	0	0.1681043

Tabla 2: Colecciones de datos

Donde los siguientes conceptos significan:

- Colección. Es el identificador de la colección
- Nº Filas. Es el número de registros que tiene la colección de datos.
- Nº Atributos. Es el número de atributos que tiene la colección en función de los cuales se clasifica. Por ejemplo, en el caso de la colección 1, estos atributos son 4: recency, frecuency, monetary y time.
- Nº Resul. Clasific. Es el número de posibles resultados que puede tener la clasificación. En el caso de la colección 1 son dos: si y no.
- Media. Es la media del valor de las Ganancias de Información de cada uno de los atributos.
- Mediana. Es la mediana del valor de las ganancias de información de cada uno de los atributos.
- Desv. Típica. Es la desviación típica de las Ganancias de Información de cada uno de los atributos.

Dichas colecciones se han escogido de acuerdo a una serie de criterios con la finalidad de comprobar cómo se comporta la solución propuesta con respecto a la original. Los criterios seguidos son:

- 3 colecciones con el mismo número de atributos.
- Colecciones con diferente número de atributos

- Colecciones con diferente número de registros o filas.
- Colecciones con diferentes valores de media, mediana y desviación típica.
- Colecciones con diferentes números de resultados que puede tener una clasificación.

4.2 Aplicación del Modelo de NB

El primer paso es aplicar el modelo de NB original sobre las colecciones expuestas en el apartado anterior. Dichas colecciones se utilizarán como datos de entrenamiento y sobre las mismas se aplicará el modelo de NBP.

La idea es ver cómo se comporta el método propuesto con respecto al método original considerando las diferentes características que tienen las colecciones y que se han expuesto anteriormente. Al considerar todos los registros como datos de entrenamiento se eliminan errores cometidos de una selección no lo suficientemente heterogénea de los datos. Esto último podría suponer conclusiones erróneas a la hora de valorar dicho método ponderado dependiendo de las características de los datos sobre los que se evalúa.

Para comprobar la calidad del modelo construido se compara la predicción prevista por el modelo de NB si se corresponde con la clasificación esperada de acuerdo a los datos de entrenamiento. Para ello se utiliza el concepto de % Adecuación, el cual tomará en valor 100% en el caso de que todos los valores se correspondan con lo esperado y 0% si no se corresponde ningún valor.

En la tabla 3 se muestran los resultados de aplicar el modelo de NB original a las colecciones expuestas en el apartado anterior.

	Colección 1	Colección 2	Colección 3	Colección 4	Colección 5	Colección 6
Media	0.02399705	0.1544309	0.6521269	0.08762667	0.09100734	0.1217417
Desv. Típica	0.02072269	0.1506597	0.3503707	0.07276803	0.08307796	0.1681043
Nº Registros	748	1372	150	1728	132	106
NB	186	606	6	440	25	66
% Adec. NB	75,13%	55,83%	96,00%	74,54%	81,06%	33,96%

Tabla 3: Resultados de Aplicar el Modelo NB

Donde los siguientes conceptos significan:

- NB. Es el número de registros que no se han correspondido con lo previsto por el modelo de NB.
- % Adec. NB. Es el % sobre el total de registros que sí se han correspondido con lo previsto por el modelo de NB.

4.3 Aplicación del Modelo Ponderado de NB

Después de aplicar sobre las 6 colecciones el modelo de NB original se procede a aplicar el método

ponderado propuesto en este Trabajo Fin de Máster. De esta forma y gracias a las diferentes características que poseen las colecciones se puede comprobar el grado de validez de las hipótesis propuestas en el capítulo anterior.

En la tabla 4 se recogen los resultados obtenidos tras la aplicación del método ponderado y se comparan con lo obtenido con el método original junto con las características de las colecciones de datos utilizadas.

	Colección 1	Colección 2	Colección 3	Colección 4	Colección 5	Colección 6
Desv. Típica	0.02072269	0.1506597	0.3503707	0.07276803	0.08307796	0.1681043
% Adec. NB	75,13%	55,83%	96,00%	74,54%	81,06%	33,96%
Nª Atributos	4	4	4	6	5	7
Nº Result. Clasific	2	2	3	4	3	7
Desc. NBP	176	580	2	310	20	35
NBP Cambiando Ord. Gan.Inf	180	605	5	249	20	42
NBP Camb. Ord. Apriori Inv.	177	605	2	294	16	36
% Adec. NBP	76,47%	57,72%	98,67%	85,59%	87,88%	66,98%
% Mejora NBP	1,75%	3,27%	2,71%	12,91%	7,76%	49,30%

Tabla 4: Resultado de Aplicar el Modelo de NBP

Donde los siguientes conceptos significan:

- Desc. NBP. Es el número de registros que no se han correspondido con lo previsto por el modelo de NBP. Es el resultado de aplicar las hipótesis expuestas en el subapartado 3.2.1 de ordenar de mayor a menor los atributos según la ganancia de información, y dar mayor peso a los coeficientes asociados al resultado con mayor probabilidad a priori. Es decir, se siguen los puntos 1 y 4.

- NBP cambiando Ord. Gan. Inf. Es el número de registros que no se han correspondido por el modelo de NBP, pero cambiando el orden de los atributos según la ganancia de información. Es decir, se sigue el punto 7 del subapartado 3.2.1.

- NBP Camb. Ord. Apriori Inv. Es el número de registros que no se han correspondido con el modelo de NBP, pero otorgándole menor peso a los coeficientes de probabilidad asociados al resultado de mayor probabilidad a priori. Es decir, se sigue el punto 6 del subapartado 3.2.1.

- % Adec. NBP. Es el % sobre el total de registros que sí se han correspondido con lo previsto

por el modelo de NBP considerando al mejor de los resultados obtenidos. Es decir, se escoge el modelo resultante de seguir el punto 8 del subapartado 3.2.1.

- % Mejora NBP. Es la mejora obtenida en la adecuación con respecto al NB original considerando al mejor de los resultados obtenidos del NBP. Es decir, según el criterio del punto 8 del subapartado 3.2.1.

De los resultados obtenidos de la tabla 4 se pueden obtener diversas conclusiones, pero en este apartado se centrará en aquellas relacionadas con las hipótesis establecidas en el subapartado 3.2.1:

- La hipótesis de ordenar los parámetros en orden ascendente de mayor a menor según el punto 1 del subapartado 3.2.1 no mejora necesariamente los resultados. De hecho, si se comparan los valores obtenidos para los conceptos de *NB Pond. Cambiando Ord. Gan.Inf* y *Desc. NB Pond* se observa que el cambiar el orden de los atributos de menor a mayor según la ganancia de información en la colección 4 se obtienen resultados sensiblemente mejores.

- La hipótesis de dar mayor peso a aquellos coeficientes de probabilidad con mayor probabilidad a priori tal y como se detalla en el punto 4 del subapartado 3.2.1. muestra buen comportamiento cuando la variable clasificadora puede tomar pocos valores, como por ejemplo Si/No. Sin embargo, cuando el problema se hace más complejo pudiendo la clasificación tomar más valores esta hipótesis no es correcta. Por ejemplo, cuando la variable clasificadora puede tomar 4 valores como es el caso de la colección 4 al comenzar dando mayor peso a los coeficientes asociados al resultado con menos probabilidad a priori, se obtienen mejores resultados que haciéndolo al revés.

4.4 Comparativas entre el modelo de NB y el modelo ponderado

Una vez que se han aplicado los métodos originales y ponderados sobre las 6 colecciones se puede realizar una comparativa más exhaustiva entre ambos. En este apartado se representan gráficamente los diferentes conceptos expuestos anteriormente con la idea de encontrar relaciones que sean significativas.

Para poder comprobar estos aspectos se representa en siguiente gráfico el % de adecuación de aplicar NB y el % de mejora tras aplicar la mejora del método NBP para cada una de las 6 colecciones.

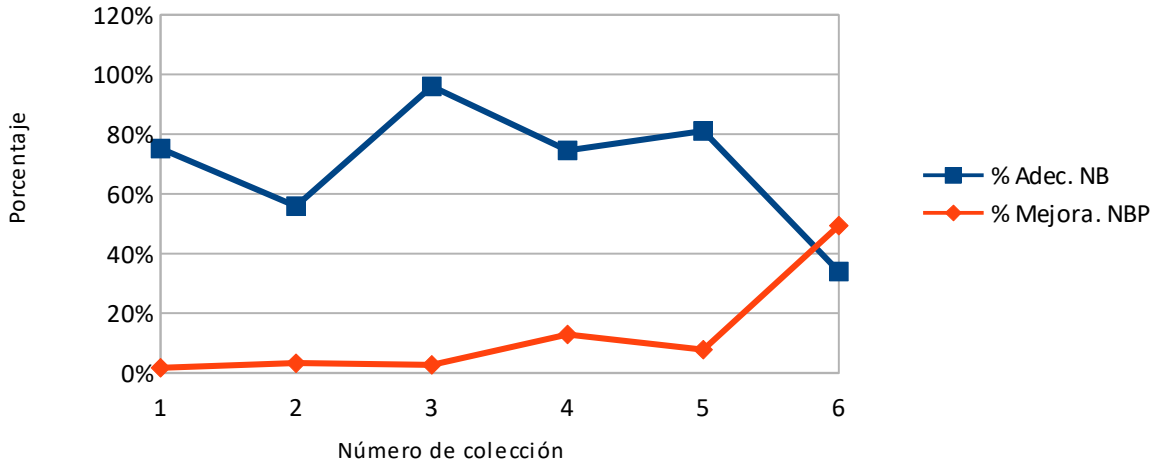

Ilustración 5: Comparativa % Adec NB - % Mejora NBP

Tal y como se puede comprobar no existe una relación clara que indique que conforme sea mayor la adecuación de NB, mayor mejora produce el NBP. De hecho, los datos obtenidos en la colección 6 ocurre justo lo contrario, es decir, tras aplicar el método original y obtener una mala adecuación, el método ponderado obtuvo una gran mejora.

En el siguiente gráfico se representa el % de adecuación de ambos métodos sobre las 6 colecciones de datos.

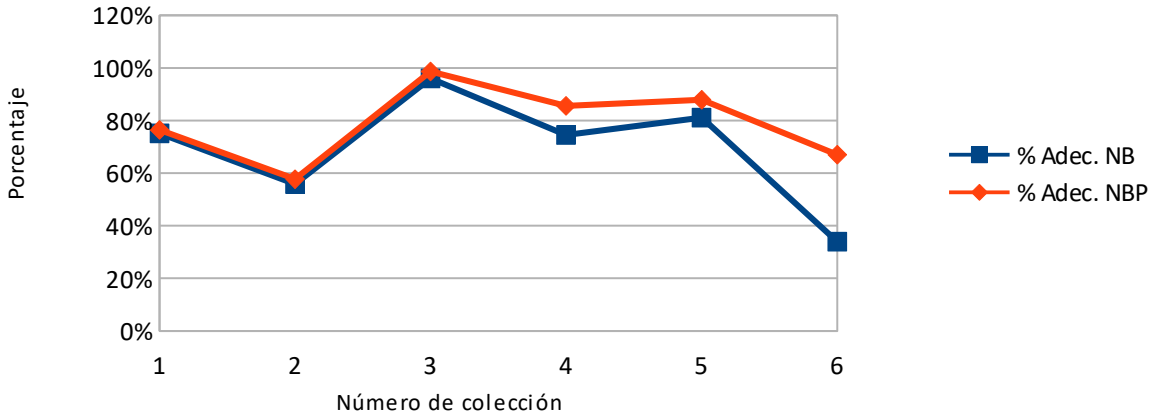

Ilustración 6: Comparativa % Adec NB - % Adec NBP

En este gráfico se puede comprobar que existe una relación lógica entre ambos % de adecuación. Sin embargo, a partir de la colección 4 se observa un margen de mejora mayor. Las colecciones 4, 5 y 6 coinciden en que tienen una mayor complejidad con respecto a las tres primeras. Es decir, tienen más atributos y mayor número de posibles resultados de clasificación.

En el siguiente gráfico se representa el % de mejora del método NBP respecto la desviación típica de las ganancias de información de cada una de las colecciones.

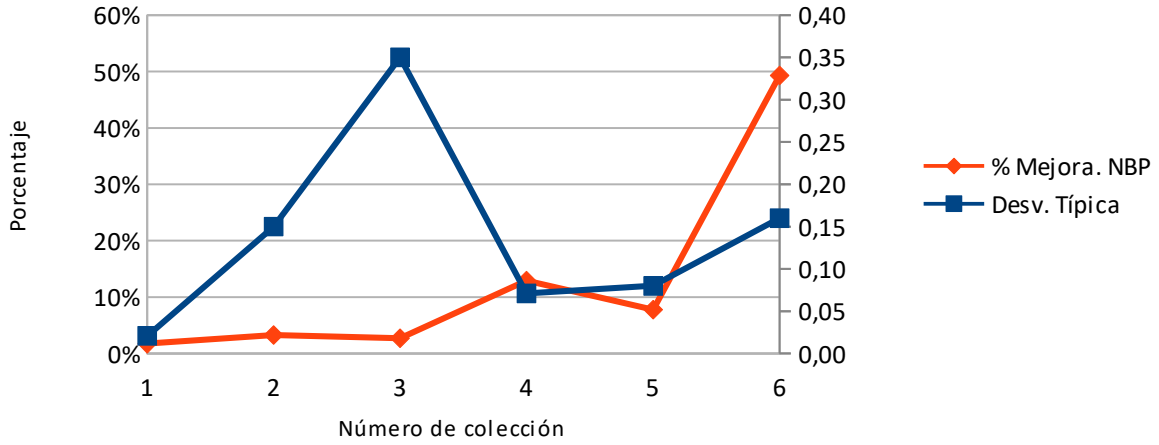

Ilustración 7: Comparativa Desv. Típica - % Mejora NBP

Como se puede comprobar en el gráfico no hay una relación entre la desviación típica de las ganancias de información de cada una de las colecciones y el porcentaje de mejora de NBP.

En el siguiente gráfico se representa el % de mejora del método NBP respecto la media de las ganancias de información de cada una de las colecciones.

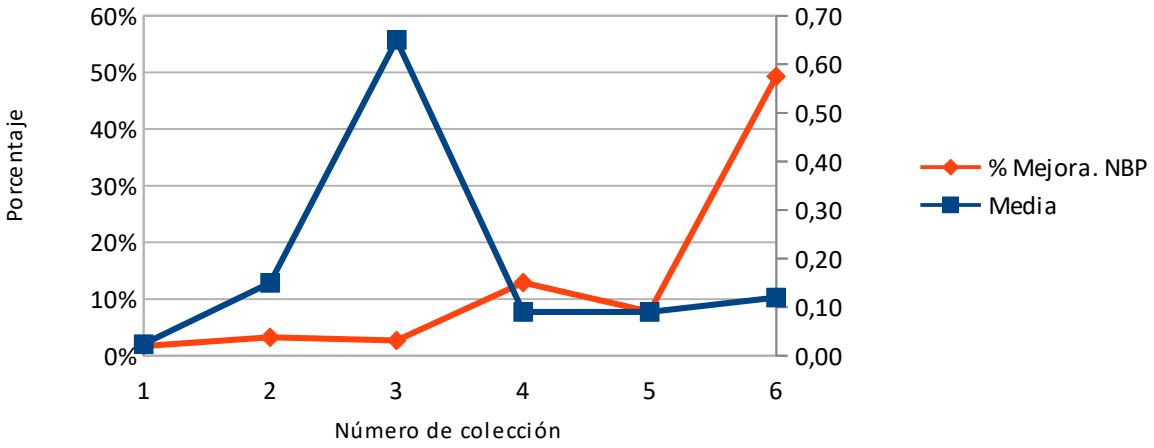

Ilustración 8: Comparativa Media - % Mejora Pond

Tal y como ocurría en el anterior gráfico no hay tampoco una relación clara entre la media de la ganancia de información y el porcentaje de mejora de NBP.

En el siguiente gráfico se representa el % de mejora del método NBP respecto al número de registros de cada una de las colecciones.

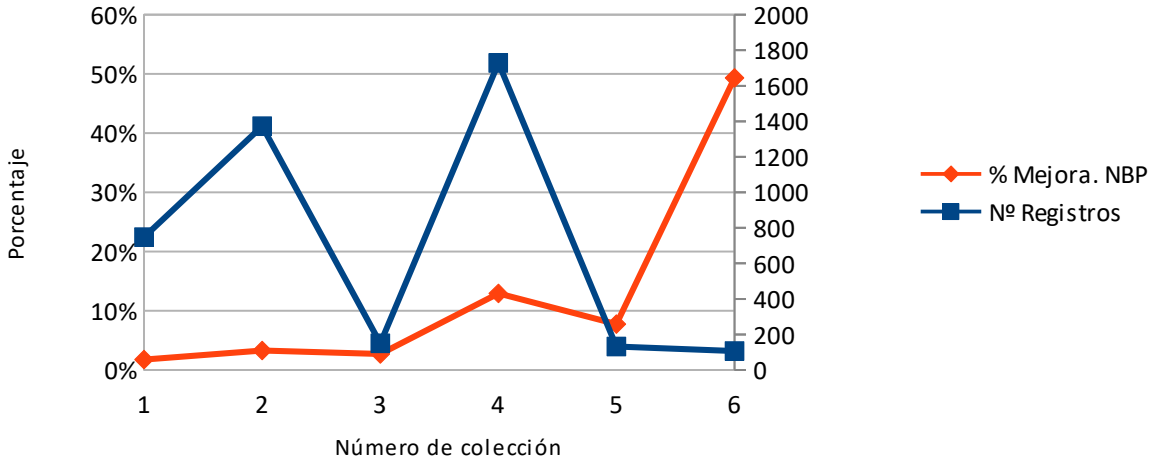

Ilustración 9: Comparativa Nº Reg - % Mejora NBP

Al igual que en los anteriores gráficos es sencillo de comprobar que no hay una relación existente entre el número de registros y el porcentaje de mejora de NBP.

En el siguiente gráfico se representa el % de mejora del método NBP respecto al número de posibles resultados que puede tener la clasificación de cada una de las colecciones.

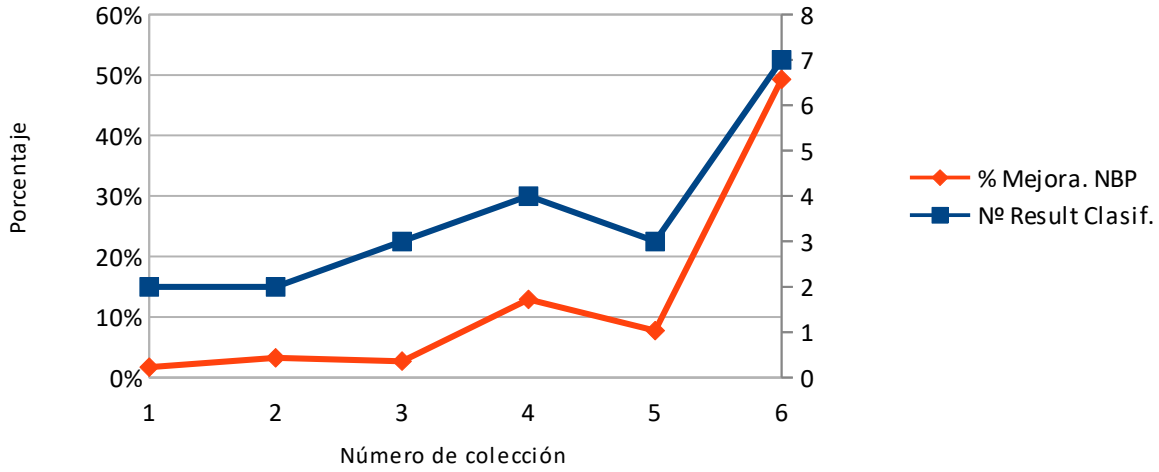

Ilustración 10: Comparativa Nº Result Clasif - % Mejora NBP

Como se puede comprobar en este gráfico sí existe una relación entre el número de resultados que puede tener la clasificación y el porcentaje de mejora de NBP. Es decir, a medida que más compleja es la clasificación mayor margen de mejora se obtiene con el método ponderado.

Por último, en el siguiente gráfico se representa el % de mejora del método NBP respecto al número de atributos de cada una de las colecciones.

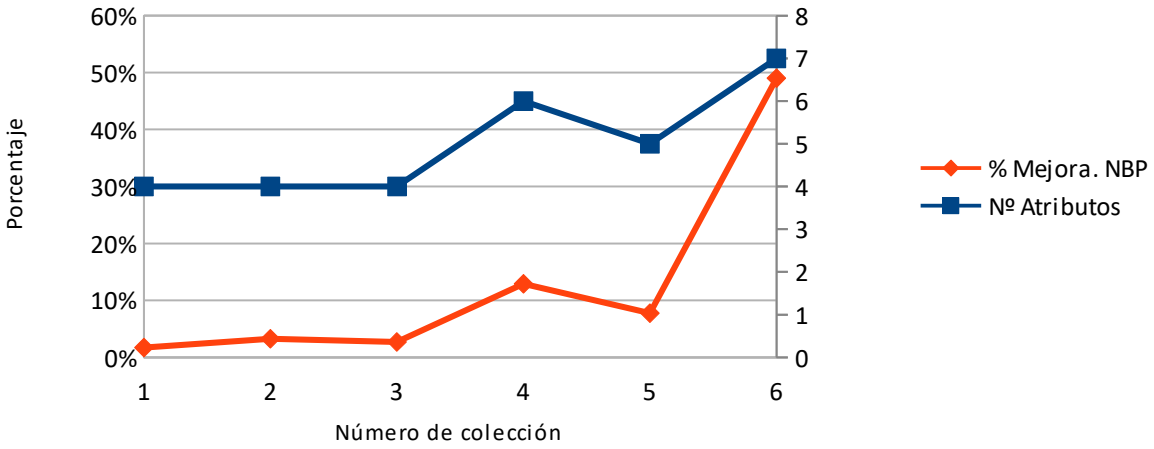

Ilustración 11: Comparativa Nº Atributos - % Mejora NBP

Como se puede comprobar en el gráfico también hay una relación clara existente entre el número de atributos que puede tener la clasificación y el porcentaje de mejora de NBP. Conforme mayor número atributos tiene la colección mejores resultados se obtienen al aplicar el método ponderado.

4.5 Comparativa con otros algoritmos

Una vez que se ha realizado la comparativa del modelo de NBP con respecto al modelo original, se procede a ponerlo en valor con respecto a otros algoritmos existentes. De esta forma se puede poner en valor cuál es la mejora producida por el algoritmo propuesto.

Para dicha comparativa se van a dividir cada una de las 6 colecciones de datos en dos conjuntos: conjunto de entrenamiento y de prueba. Esta división se realiza de la forma más heterogénea posible.

Los algoritmos con los que se va a comparar son los siguientes:

- *Fuzzy Clustering* o Agrupamiento difuso (FC)

  Este algoritmo no supervisado trata de asignar una probabilidad de pertenencia de cada elemento con respecto a un clúster determinado. Es decir, no asigna cada elemento unívocamente a un clúster determinado, sino que da un grado de pertenencia a cada uno de ellos.

  Al ser un algoritmo no supervisado, no tiene un conjunto de datos que sirva de entrenamiento previo, sino que son los propios datos que intenta clasificar los que sirven de entrenamiento del algoritmo.

  El hecho de que no sea supervisado le confiere una importante desventaja respecto a otros métodos que sí tienen una muestra de entrenamiento previa, por lo que es lógico que este algoritmo sea el que peor rendimiento presente con respecto al resto con los que se compara

en el presente trabajo.

- KNN (*K Nearest Neighbor*) o Vecinos más próximos (KNN)

  Se trata de un algoritmo supervisado por lo que es de la misma clase que el método propuesto. La idea principal de este algoritmo es que cada elemento se clasificará en base a sus K vecinos más próximos. El valor de K debe estar en un punto medio. Es decir, si K es demasiado pequeño se obtendrán resultados muy sensibles a puntos ruidosos, mientras que si K es demasiado grande se podrán obtener puntos de otras clases lejanas.

- *Support Vector Machine* o Máquinas de vectores de soporte (SVM)

  Se trata también de un algoritmo supervisado por lo que le confiere valor a la hora de comparar el método propuesto. Este algoritmo busca de la forma más óptima posible un hiperplano que separe de forma óptima los puntos de una clase de la otra. Al vector formado por los puntos más cercanos al hiperplano se le llama vector de soporte.

- Algoritmo de selección de características de NB usando árboles de decisión C4.5 (NB-C4.5)

  En el apartado 3.3 se expuso el algoritmo desarrollado por Ratanamahatana y Gunopulos acerca de la utilización del algoritmo C4.5 para seleccionar a aquellos atributos más importantes a la de aplicar el modelo de NB. Este algoritmo destaca por su sencillez y buenos resultados obtenidos por lo que se ha escogido para comprobar cómo se comporta el método propuesto con respecto a otros métodos existentes.

  Por ejemplo, la colección 1 posee los siguientes atributos o características: recency, frecuency, monetary y time. Tras aplicar los pasos 1 al 4 expuesto en el punto 3.3 se obtienen los siguientes atributos relevantes: recency, frecuency y time, eliminando de esta forma al atributo monetary. Tras seleccionar los atributos más importantes se vuelve a aplicar el método de NB.

- Algoritmo de selección de características de NBP usando árboles de decisión C4.5 (NBP-C4.5)

  Por último, se aplica el método de NBP al algoritmo que combina NB y C4.5. Es decir, se utiliza el algoritmo C4.5 para seleccionar aquellos atributos más representativos y posteriormente en vez de aplicar el método de NB original se utiliza el ponderado propuesto.

- Clasificador de NB/NBP

  Se aplican los métodos de NB y NBP utilizando los conjuntos de entrenamiento y prueba obtenidos para cada una de las colecciones.

En la tabla 5 se presentan los resultados obtenidos de la comparativa de la colección 1 con los algoritmos propuestos:

Comparativa de la Colección 1	
Nº Registros de prueba	330
NB descuadre	60
% Adec. NB	81,81%
NBP descuadre	52
% Adec. NBP	84,24%
% Mejora NBP	2,88%
FC descuadre	101
% Adecuación FC	69,39%
KNN descuadre	75
% Adecuación KNN	77,27%
SVM descuadre	49
% Adecuación SVM	85,15%
Selección de atributos usando C4.5 (NB – C4.5)	54
% Adecuación Selección atributos usando C4.5 (NB – C4.5)	83,63%
Selección de atributos usando c4.5 (NBP – C4.5)	53
% Adecuación Selección atributos usando C4.5 (NBP – C4.5)	83,94%

Tabla 5: Comparativa colección 1

Donde los siguientes conceptos significan:

- FC Descuadre. Número de registros del conjunto de prueba que no se han correspondido con el resultado correcto tras aplicar el algoritmo de *Fuzzy Clustering*.
- % Adecuación FC. Es el % sobre el total de registros que sí se han correspondido con lo previsto tras aplicar el algoritmo de *Fuzzy Clustering*.
- KNN descuadre. . Número de registros del conjunto de prueba que no se han correspondido con el resultado correcto tras aplicar el algoritmo de KNN.
- % Adecuación KNN. Es el % sobre el total de registros que sí se han correspondido con lo previsto tras aplicar el algoritmo de KNN.
- SVM descuadre. Número de registros del conjunto de prueba que no se han correspondido con el resultado correcto tras aplicar el algoritmo de SVM.
- % Adecuación SVM. Es el % sobre el total de registros que sí se han correspondido con lo previsto tras aplicar el algoritmo de SVM.

- Selección de atributos usando NB – C4.5. Número de registros del conjunto de prueba que no se han correspondido con el resultado correcto tras aplicar el algoritmo de NB – C4.5.

- % Adecuación Selección atributos usando NB – C4.5. Es el % sobre el total de registros que sí se han correspondido con lo previsto tras aplicar el algoritmo de NB – C4.5.

- Selección de atributos usando NBP – C4.5. Número de registros del conjunto de prueba que no se han correspondido con el resultado correcto tras aplicar el algoritmo de NBP-C4.5).

- % Adecuación Selección atributos usando NBP – C4.5 . Es el % sobre el total de registros que sí se han correspondido con lo previsto tras aplicar el algoritmo de NBP-C4.5.

En el siguiente gráfico se representan los valores de % adecuación para todos los algoritmos que se han comparado utilizando la coleccion 1.

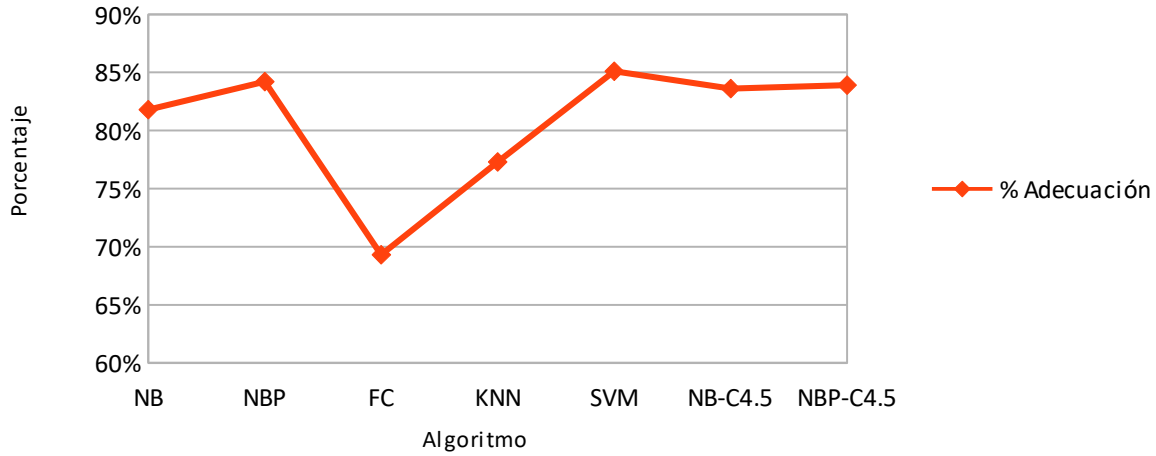

Ilustración 12: Comparativa del algoritmo ponderado con otros algoritmos col. 1

Como se puede comprobar en la ilustración 12 el método NBP tiene resultados similares a los que se obtienen utilizando el algoritmo SVM, el cual, es el que muestra mejores resultados. Asimismo, los resultados obtenidos utilizando los algoritmos NB-C4.5 y NBP-C4.5 son bastantes similares. Por último, el algoritmo no supervisado FC como era de esperar es el que ofrece peor rendimiento con diferencia.

En la tabla 6 se presentan los resultados obtenidos de la comparativa de la colección 2 con los algoritmos propuestos:

Comparativa de la Colección 2	
Nº Registros de prueba	626
NB descuadre	191
% Adec. NB	69,48%
NBP descuadre	180

% Mejora NBP	71,25%
% Adec. NBP	2,47%
FC descuadre	234
% Adecuación FC	62,61%
KNN descuadre	199
% Adecuación KNN	68,21%
SVM descuadre	48
% Adecuación SVM	92,33%
Selección de atributos usando C4.5 (NB – C4.5)	191
% Adecuación Selección atributos usando C4.5 (NB – C4.5)	69,69%
Selección de atributos usando c4.5 (NBP – C4.5)	180
% Adecuación Selección atributos usando C4.5 (NBP – C4.5)	71,25%

Tabla 6: Comparativa colección 2

En el siguiente gráfico se representan los valores de % adecuación para todos los algoritmos que se han comparado utilizando la colección 2.

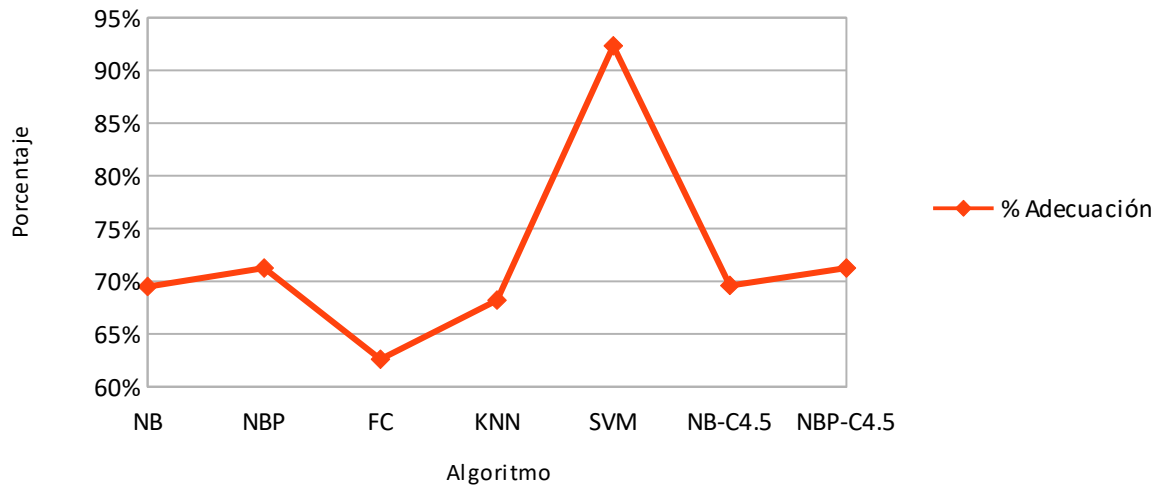

Ilustración 13: Comparativa del algoritmo ponderado con otros algoritmos col. 2

Como se puede comprobar en la ilustración 13 el método NBP obtiene resultados por encima o similares en comparación con el resto de algoritmos, a excepción del algoritmo SVM que alcanza unos resultados muy por encima del resto de algoritmos. Como era de esperar el algoritmo no supervisado de FC es el que obtiene peores resultados.

En la tabla 7 se presentan los resultados obtenidos de la comparativa de la colección 3 con los

algoritmos propuestos:

Comparativa de la Colección 3	
Nº Registros de prueba	86
NB descuadre	2
% Adec. NB	97,67%
NBP descuadre	1
% Adec. NBP	98,83%
% Mejora NBP	3,52%
FC descuadre	10
% Adecuación FC	88,37%
KNN	6
% Adecuación KNN	93,02%
SVM descuadre	3
% Adecuación SVM	96,51%
Selección de atributos usando C4.5 (NB – C4.5)	5
% Adecuación Selección atributos usando C4.5 (NB – C4.5)	94,18%
Selección de atributos usando c4.5 (NBP – C4.5)	4
% Adecuación Selección atributos usando C4.5 (NBP – C4.5)	95,35%

Tabla 7: Comparativa colección 3

En el siguiente gráfico se representan los valores de % adecuación para todos los algoritmos que se han comparado utilizando la colección 3.

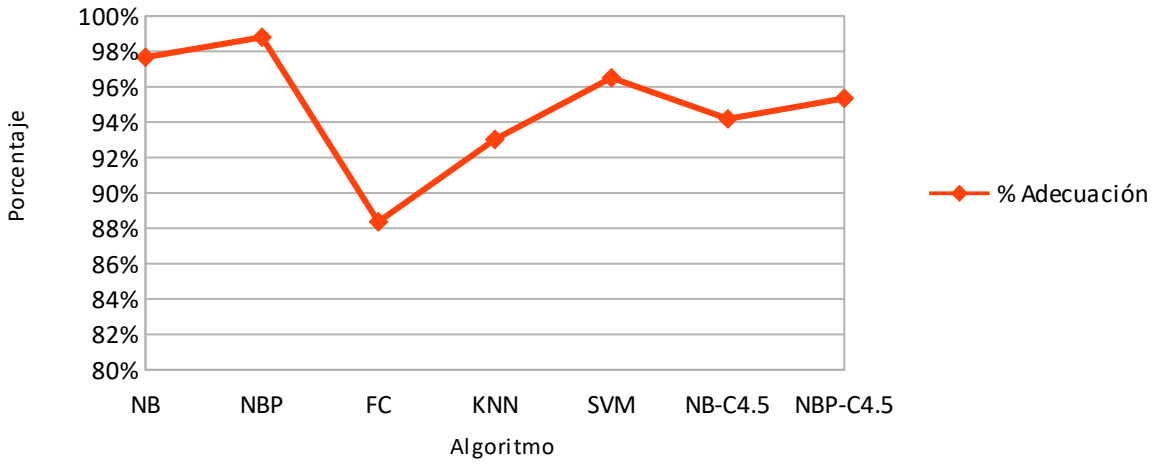

Ilustración 14: Comparativa del algoritmo ponderado con otros algoritmos col. 3

Como se puede comprobar en la Ilustración 14, el algoritmo NBP obtiene resultados claramente por encima del resto de algoritmos. Es de destacar, que se sitúa por encima del algoritmo SVM que suele mostrar mejores resultados. Una vez más el algoritmo FC es el que muestra peores resultados.

En la tabla 8 se presentan los resultados obtenidos de la comparativa de la colección 4 con los algoritmos propuestos:

Comparativa de la Colección 4	
Nº Registros de prueba	525
NB descuadre	189
% Adec. NB	64,00%
NBP descuadre	104
% Adec. NBP	80,19%
% Mejora NBP	20,20%
FC descuadre	
% Adecuación FC	
KNN descuadre	146
% Adecuación KNN	72,19%
SVM descuadre	43
% Adecuación SVM	91,81%
Selección de atributos usando C4.5 (NB – C4.5)	189
% Adecuación Selección atributos usando C4.5 (NB – C4.5)	64,00%
Selección de atributos usando c4.5 (NBP – C4.5)	100
% Adecuación Selección atributos usando C4.5 (NBP – C4.5)	80,95%

Tabla 8: Comparativa colección 4

En el siguiente gráfico se representan los valores de % adecuación para todos los algoritmos que se han comparado utilizando la colección 4.

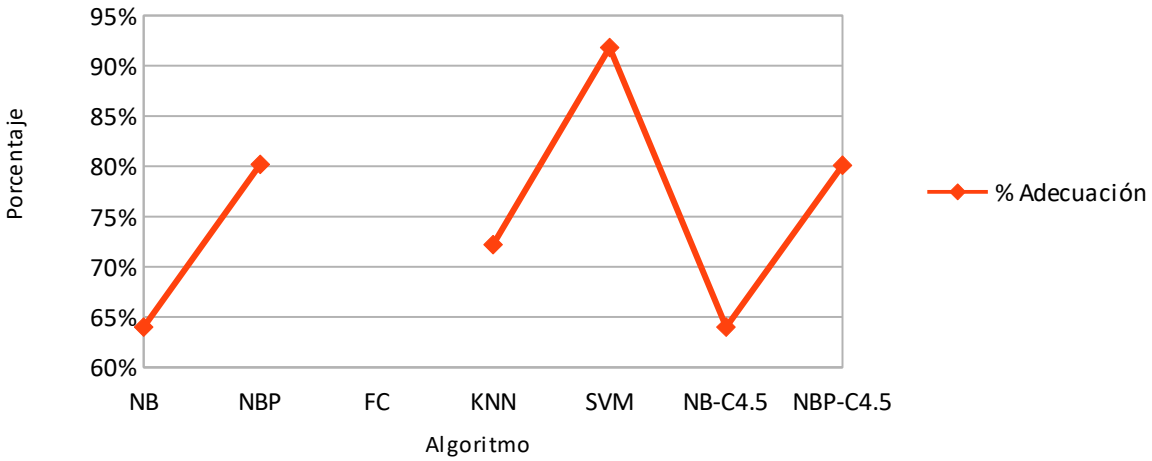

Ilustración 15: Comparativa del algoritmo ponderado con otros algoritmos col. 4

Como se puede comprobar en la ilustración 15 el algoritmo NBP obtiene resultados mejores o similares que la mayoría de los obtenidos por el resto de algoritmos, pero como ocurría con la colección 2, el algoritmo SVM vuelve a obtener el mejor resultado con diferencia para esta colección de datos. Para el algoritmo FC no se han podido obtener resultados debido a que no era posible aplicar la colección de datos.[4]

En la tabla 9 se presentan los resultados obtenidos de la comparativa de la colección 5 con los algoritmos propuestos:

Comparativa de la Colección 5	
Nº Registros de prueba	63
NB descuadre	20
% Adec. NB	68,26%
NBP descuadre	7
% Adec. NBP	88,89%
% Mejora NBP	23,21%
FC descuadre	48
% Adecuación FC	23,80%
KNN descuadre	22
% Adecuación KNN	65,08%
SVM descuadre	10
% Adecuación SVM	84,13%
Selección de atributos usando C4.5 (NB – C4.5)	12

4 La función fanny del paquete cluster de R tiene un parámetro que indica el número de clusters a construir, es decir, el número de posible resultados de la clasificación. Para el número de clusters necesarios para construir de colección 4 la función fanny da error al considerarlo muy alto.

% Adecuación Selección atributos usando C4.5 (NB – C4.5)	80,95%
Selección de atributos usando c4.5 (NBP – C4.5)	7
% Adecuación Selección atributos usando C4.5 (NBP – C4.5)	88,89%

Tabla 9: Comparativa colección 5

En el siguiente gráfico se representan los valores de % adecuación para todos los algoritmos que se han comparado utilizando la colección 5.

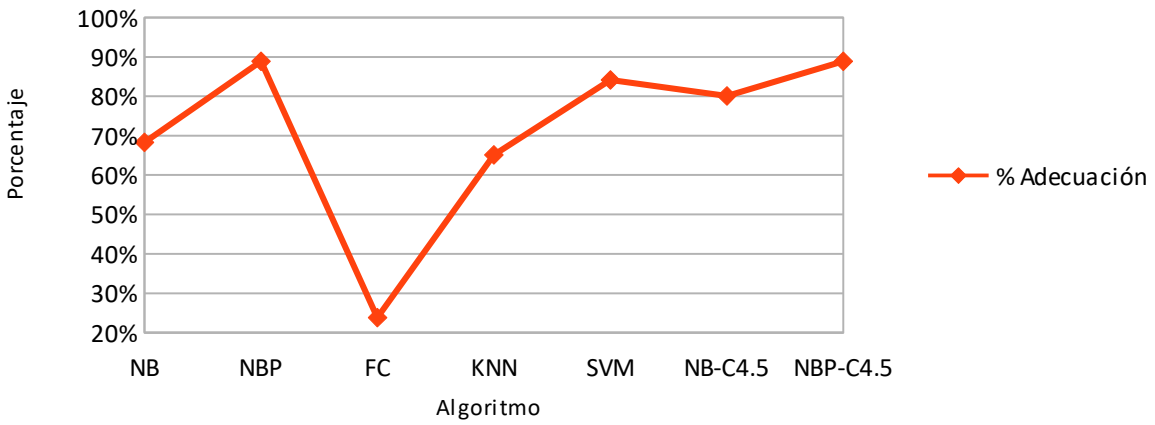

Ilustración 16: Comparativa del algoritmo ponderado con otros algoritmos col. 5

Como se puede comprobar en la ilustración 16, el algoritmo NBP junto con NBP-C4.5 obtienen los mejores resultados de la comparativa. Es de destacar que se consiguen mejores resultados que el algoritmo SVM.

Por último, en la tabla 10 se presentan los resultados obtenidos de la comparativa de la colección 6 con los algoritmos propuestos:

Comparativa de la Colección 6	
Nº Registros de prueba	54
NB descuadre	22
% Adec. NB	59,26%
NBP descuadre	18
% Adec. NBP	66,67%
% Mejora NBP	11,11%
FC descuadre	48
% Adecuación FC	11,11%
KNN descuadre	17

% Adecuación KNN	68,52%
SVM descuadre	22
% Adecuación SVM	59,26%
Selección de atributos usando C4.5 (NB – C4.5)	22
% Adecuación Selección atributos usando C4.5 (NB – C4.5)	59,26%
Selección de atributos usando c4.5 (NBP – C4.5)	18
% Adecuación Selección atributos usando C4.5 (NBP – C4.5)	66,67%

Tabla 10: Comparativa colección 6

En el siguiente gráfico se representan los valores de % adecuación para todos los algoritmos que se han comparado utilizando la colección 6.

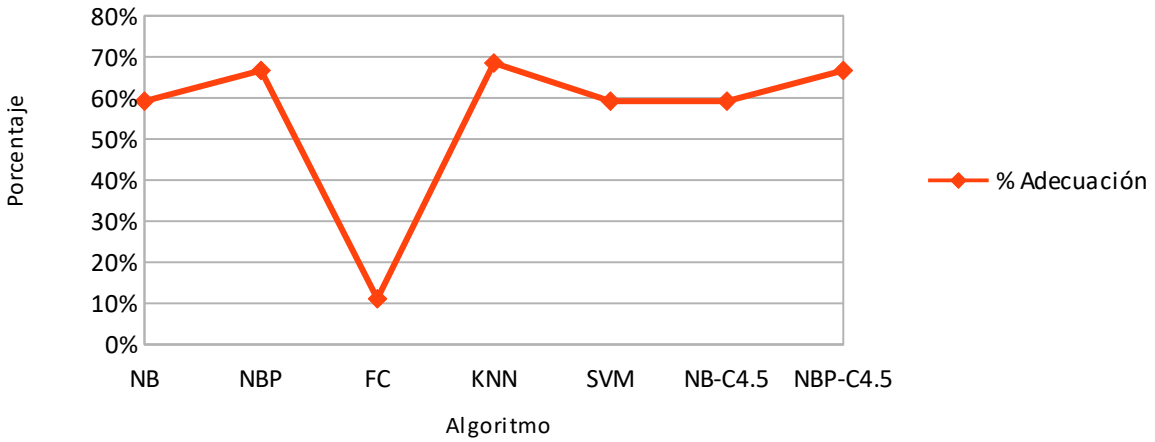

Ilustración 17: Comparativa del algoritmo ponderado con otros algoritmos col. 6

Tal y como puede observarse en la Ilustración 17, el algoritmo NBP tiene un comportamiento mejor que la mayoría de los resultados, aunque en este caso el algoritmo KNN obtiene resultados muy similares siendo el único caso en el que ocurre. Es de destacar que los algoritmos NB-C4.5 y NBP-C4.5 obtienen valores idénticos a NB y NBP esto es debido a que tras aplicar el filtro de atributos utilizando el algoritmo C4.5 no se llegó a eliminar ninguno, por lo que los resultados son iguales. Posiblemente con una colección con mayor número de registros sí se hubiera podido eliminar algún atributo.

5. Conclusiones

En este Trabajo Fin de Máster se han presentado los principales algoritmos relacionados con el Aprendizaje Automático o *Machine Learning.* Estos algoritmos pueden ser supervisados, no supervisados y semi-supervisados. Entre ellos destacan las máquinas de vectores de soporte o SVM, métodos basados en vecindad (como por ejemplo el KNN), árboles de decisión, redes neuronales y los métodos bayesianos.

Dentro de los métodos bayesianos destaca el clasificador de Naive Bayes (NB), el cual posee la ventaja de poder utilizarse tanto para tareas de uso predictivo como de uso descriptivo, y además es sencillo de entender. Dicho algoritmo se basa en el Teorema de Bayes pero estableciendo una hipótesis simplificadora que consiste en establecer la independencia entre las variables que componen el problema. Esto da como resultado un algoritmo mucho más sencillo de implementar, y además, en determinados dominios llega a obtener resultados tan buenos como los árboles de decisión o las redes neuronales.

Sin embargo, cuando la relación entre las variables que componen el problema es fuerte, el clasificador de NB encuentra uno de sus principales puntos débiles y es por ello que para paliar, al menos en parte, esta debilidad se han desarrollado distintos métodos ponderados.

El método ponderado que se ha propuesto en el presente Trabajo Fin de Máster establece una serie de hipótesis establecidas en la sección 3.2 del capítulo 3. Dicho método ha sido aplicado a 6 colecciones de datos extraídas del repositorio UCI. Como principales conclusiones se tienen que:

- Conforme mayor número de atributos y de posibilidades de clasificación tiene la colección mejores resultados se obtienen al aplicar el método ponderado. Es decir, conforme más compleja es la colección mejores resultados se obtienen.

- No hay una relación que indique que cuanta mejor adecuación tenga el método NB original, mejor comportamiento tendrá el método NBP. De hecho, puede ocurrir justo lo contrario.

- El método ponderado propuesto es independiente del número de registros de la colección.

- No hay una relación entre la desviación típica o media de las ganancias de información de cada una de las colecciones y el porcentaje de mejora de NBP.

- El comenzar a iterar por aquellos coeficientes de probabilidad, siguiendo el orden de mayor a menor ganancia de información de las variables no necesariamente da mejores resultados. De hecho, cambiando el orden puede ocurrir justamente lo contrario.

- Asimismo, el dar mayor peso a aquellos coeficientes de probabilidad relacionados con mayor probabilidad a priori sólo parece dar mejores resultados cuando la variable clasificadora puede tomar pocos valores. De hecho, cuando el problema se hace más complejo pudiendo la clasificación tomar más valores esta hipótesis no es correcta.

Estas conclusiones se han establecido comparando el método propuesto en este trabajo (NBP) respecto al original. Sin embargo, para poner en valor dicha mejora se han realizado comparativas con otros algoritmos existentes llegando a las siguientes conclusiones:

- Con respecto al algoritmo no supervisado FC es donde presenta mayor mejora. Esto era de esperar ya que se trata de un algoritmo no supervisado, y ya de por sí el propio algoritmo de NB original presenta mejores resultados.

- Con respecto al algoritmo supervisado KNN también se obtienen buenos resultados, tanto el método original y ponderado, lo cual también era de esperar ya que el algoritmo de NB original suele construir un modelo más óptimo con respecto a este método clásico. Sin embargo, esto no ocurre para la colección 6 donde el algoritmo KNN obtiene el mejor rendimiento de todos los algoritmos. Esto puede ser debido a que la colección 6 tiene pocos registros tanto de prueba como de entrenamiento.

- Con respecto al algoritmo de SVM, el algoritmo NBP tiene un comportamiento bastante dispar dependiendo del caso. Como mucho el método ponderado llega a igualarlo o superarlo por poco margen. Sin embargo, cuando SVM tiene mejor comportamiento que el algoritmo NBP puede llegar a superarlo con creces.

- Con respecto al algoritmo NB-C4.5 este método, suele llegar a mejorar al NB original pero llega a ser mejorado por el NBP propuesto en este trabajo. Hay algunos casos como la colección 3 donde el comportamiento del NB-C4.5 es peor que el método NB original, sin embargo esto coincide con una colección con pocos datos tanto de entrenamiento como de prueba, lo cual hace este caso poco significativo, ya que este algoritmo ponderado exige utilizar 5 conjuntos de entrenamiento diferentes.

- Por último, se prueba a combinar el algoritmo C4.5 con el método ponderado propuesto, en general, los resultados son bastantes similares a los obtenidos tras aplicar el método de NBP directamente.

5.1 Trabajos Futuros

En este Trabajo Fin de Máster se ha profundizado en la mejora del algoritmo de NB proponiendo un método ponderado siguiendo una serie de hipótesis. Dicha mejora ha conseguido un rendimiento superior que el método de NB original, llegando en algunos casos a tener un comportamiento similar a otros algoritmos considerados más eficaces como el SVM o superando al algoritmo C4.5-NB. Sin embargo, como líneas futuras de mejora de dicho trabajo se proponen las siguientes líneas:

- Simplificación del método propuesto. Para facilitar su implementación y entendimiento sería necesario simplificar las hipótesis establecidas en el capítulo 3 (Sección 3.2). Dicha simplificación mejoraría el rendimiento de este algoritmo en tiempo de ejecución.

- Establecer nuevas hipótesis sobre el método propuesto con la finalidad de obtener mejores resultados.

- Combinación del método ponderado con otros métodos ya existentes (SVM, ID3, FC, etc.)

Bibliografía

Barrientos R.; Cruz N.; Acosta H.; Rabatte I.; Gogeascoechea M.C; Pavón P. y Blázquez S. (2009) *Árboles de decisión como herramienta en el diagnóstico médico*. Revista médica la Universidad Veracruzana, número 2.

Beca S. (2007). *Clustering difuso con selección de atributos*. Universidad de Chile. Facultad de Ciencias Físicas y Matemáticas.

Beinlich, I.A., Suermondt, H.J., Chavez, R.M., Cooper, G.F. (1989). *The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks*. In proceedings of the 2nd European Conference on Artificial Intelligence in Medicine.

Belmman R.E (1961). *Adaptative Control Processes*. Princeton University Press, Princetion, NJ.

Beltran B. (2016). *Minería de datos*. Benemérita Universidad Autónoma de Puebla. Facultad de Ciencias de la Computación.

Biggus, J. P. (1996). *Data Mining With Neural Networks.* McGraw-Hill 1996.

Bohanec, M. (1997) UCI Machine Learning Repository http://archive.ics.uci.edu/ml. http://archive.ics.uci.edu/ml/datasets/Car+Evaluation

Breese, John S., Blake, Russ (1995). *Automating Computer Bottleneck Detection with Belief Nets*. Proceedings of the Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, CA.

Burgues C. (1998). *A tutorial on support vector machine for pattern recognition*. Data Mining and Knowledge Discovery, 2:121-167.

Calancha A. (2011). *Breve aproximación a la técnica de árbol de decisiones*. Extraído el 2 de marzo de: https://niefcz.files.wordpress.com/2011/07/breve-aproximacion-a-la-tecnica-de-arbol-de-decisiones.pdf

Cheng, J., and Greiner, R. (1999). *Comparing Bayesian Network Classifiers*. In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence

Coto M. (2013). *Minería de datos: Conceptos y Aplicaciones*. Universidad Autónoma Metropolitana y Universidad de Costa Rica.

Dietterich T. (1990). *Readings in machine learning*. Oregon State University

Domingos P. (2012) *A few useful things to know about machine learning*. University of Washington. Departament of computer science and engineering. Revista de Communications of the ACM, volumen 55, número 10. 78-87.

Duda, R. O. y Hart, P. E. (1973). *Pattern classificators and scene analysis*. Wiley, New York

Fayyad U. (1996). *Advances in knowledges Discovery and Data Mining*. AAAI Press.

Fernández E. *Análisis de clasificadores bayesianos*. Universidad de Buenos Aires. Extraído el 1 de febrero del 2018 de: http://materias.fi.uba.ar/7550/clasificadores-bayesianos.pdf

Ferreira J.T.A.S, Denison D.G.T y Hand D.J. (2001). *Weighted naïve Bayes modelling for data mining.*

Finlay, P. (1994). *Introducting decision support systems*. NCC Blackwell

Fisher, R. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. http://archive.ics.uci.edu/ml/datasets/Iris

Formia S. (2012) *Evaluación de técnicas de Extracción de conocimiento en Base de datos y su aplicación a la deserción de alumnos universitarios*. Universidad Nacional de la Plata. Facultad de Informática.

Grover N. (2014). *A study of various Fuzzy Clustering Algorithms.*International Journal of Engineering Research, volumen 3, número 3, 177-181.

Gunn S. (1998). *Support Vector Machines for Clasification and Regression*. ISIS Technical Report.

Hall M. A. (2000). *Correlation-based feature selection for discrete and numeric class machine learning*. In Proceedings of the Seventeenth International Conference on Machine Learning. pages 359–366. Morgan Kaufmann, 2000

Hall M.A. (2007). *A decision tree-based attribute weighting filter for naïve Bayes*. volumen 20, 120-126.

Hayes-Roth Barbara and Frederick (1989). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. http://archive.ics.uci.edu/ml/datasets/Hayes-Roth

Hernández J.; Ramírez M. J.; Ferri C. (2004). *Introducción a la minería de datos.* Pearson.

Hidalgo J.I. Y. Cervigón C. (2004). *Una revisión de los algoritmos evolutivos y sus aplicaciones. Revista del CES Felipe II, número 2.*

Hilden J. Bjerregaard B. (1976). *Computer-aided diagnosis and the atypical case. In In Decision Making and Medical Care: Can Information Science Help*. North-Holland Publishing Company

I-Cheng Yeh (2008). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Chung-Hua, Taiwan ;University of Chung-Hua, Departament of Information Management. http://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center

Lewis D. (1998). *Naive Bayes at forty: The independence assumption in information retrieval*. In ECML-98: Proceedings of the Tenth European Conference on Machine Learning, pages 4–15

Lohweg, V. (2012). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Ostwestfalen-Lippe, Gernany: University of Applied

Sciences.http://archive.ics.uci.edu/ml/datasets/banknote+authentication

Malagón C. (2003). *Clasificadores bayesianos. El algoritmo de Naive Bayes*. Extraído el 1 de febrero del 2018 de: https://www.nebrija.es/~cmalagon/inco/Apuntes/bayesian_learning.pdf

Martínez-Arroyo M. y Sucar L. (2006). *Learning an Optimal Naive Bayes Classifier.* Pattern Recognition, 2006. ICPR 2006. 18th International Conference

Martínez, S.; Ramírez C.; Rodríguez G. y Salazar W. (2013). *Implementación del algoritmo C4.5 de aprendizaje automático para la generación de árboles de decisión en forma inductiva en el proyecto AIPI*. Universidad José Simeón Cañas. Extraído el 2 de marzo del 2018 de: https://es.scribd.com/document/171594291/IMPLEMENTACION-DEL-ALGORITMO-C4-5-DE-APRENDIZAJE-AUTOMATICO-PARA-LA-GENERACION-DE-ARBOLES-DE-DECISION-EN-FORMA-INDUCTIVA-EN-EL-PROYECTO-AIPI

Mitchell T. (2015). *Machine Learning*. McGraw Hill.

Montaño J.J. (2002). *Redes neuronales artificiales aplicadas al análisis de datos*. Universitat de Les Illes Balears. Facultad de Psicología.

Moujahid A.; Inza I. Y Larrañaga P. (2008). *Clasificadores K-NN*. Universidad del País Vasco. Departamento de Ciencias de la Computación e Inteligencia Artificial.

Nashipudimath M. y Thomas B. (2012). Comparative analysis of fuzzy clustering algorithms in data mining. International Journal of Advance Research in Computer Science and Electronics Engineering, volumen 1, 221-225.

Quinlan (1993). C4.5: *Programs for Machine Learning Morgan Kaufmann*.

Ratanamahatana C. A. y Gunopulos D. (2003). Feature selection for the naive Bayesian classifier using decision trees. Applied Artificial Intelligence 17, 475–487

Rebollo M. (2009). Minería de datos con R. *Una reflexión sobre seguridad y ataques cibernéticos*. Universidad Nacional de Educación a Distancia. Departamento de Ingeniería de Software y Sistema Informáticos.

Reich Y. y Feves S. (1990). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Pittsburg Pensilvania. Department of Civil Engineering and Engineering Design Research Center Carnegie Mellon University. http://archive.ics.uci.edu/ml/datasets/Pittsburgh+Bridges.

Rivera M. (2011). *El papel de las redes bayesianas en la toma de decisiones*. Universidad del Rosario. Extraído el 1 de febrero del 2018 de: http://www.urosario.edu.co/Administracion/documentos/investigacion/laboratorio/miller_2_3.pdf

Ruiz A.;Hernández L. y Giraldo W. *"Aplicación de los sistemas de soporte a la decisión (DSS) en el comercio electrónico",* Revista ingeniería e investigación vol. 29 No 2 (94-99).

Santa J.J. y Veloza J.J. (2013). *Aplicación del aprendizaje automático con árboles de decisión al estudio de las variables del modelo de indicadores de gestión de las universidades públicas*. Universidad Tecnológica de Pereira. Facultad de Ciencias Básicas. Extraído el 2 de marzo de: http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/4916/51953S231.pdf;sequence=1.

Sehn T. (2014). *C4.5 algorithm and multivariate decision tree*. Nacional Institute for Space Research. Image Processing Division. Brasil. Extraído el 1 de febrero del 2018 de: https://www.researchgate.net/publication/267945462_C45_algorithm_and_Multivariate_Decision_Trees

Sharma, S.; Agrawal J. Y Sanjeev S. (2013). *Revista International Journal of Computer Applications*, volumen 82, número 16.

Sommerville, I. (2005). *Ingeniería de Software 7a Edición.* Pearson Addison Wesley

Timarán-Pereira, S. R., Hernández-Arteaga, I., Caicedo-Zambrano, S. J., Hidalgo-Troya, A. y Alvarado-Pérez, J. C. (2016). *Descubrimiento de patrones de desempeño académico*. Ediciones Cooperativa de Colombia.

Turban E. (1995). *Decision Support and Expert systems*. Macmillan

Turban E. y Aronson J. (2005). Decision Support Systems and Intelligence Systems. Pearson/Prentice Hall

Vallejos S. (2006). *Minería de datos*. Universidad Nacional del Nordeste. Facultad de Ciencias Exactas, Naturales y Agrimesura.

Webb G. I., Boughton J., Zheng F., Ting K. M. y Salem H. (2011). Learning by extrapolation from marginal to full-multivariate probability distributions. Machine Learning, 1-40

Yañez J. (2008). *La importancia de los DSS en la competitividad de las empresas*. Revista digital universitaria, volumen 9, número 12.

Zaidi N.; Cerquides J.; Carman M.J. Y Webb G. (2013). *Alleviating Naive Bayes Attribute Independence Assumption by Attribute Weighting*. Journal of Machine Learning Research, número 14, 1947-48

Zhang H. y Sheng S. (2004) *Learning Weighted Naive Bayes with Accurate Ranking*. Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on, 567-570.

Definición de siglas, abreviaturas y acrónimos

Término	Definición
AUC	Area Under Curve. Está relacionado con el rendimiento de un modelo de clasificación.
BBDD	Base de Datos
C4.5	Algoritmo utilizado para generar un árbol de decisión.
DSS	Decision support System. Son sistemas que proporcionan información y soporte a la toma de decisiones basados en OLAP y minería de datos.
FC	Fuzzy Clustering.
ID3	Algoritmo utilizado para generar un árbol de decisión.
KDD	Knowledge Discover from Database. Se refiere al proceso de extracción del conocimiento relacionado con la minería de datos.
KNN	K-Nearest Neighbors
NB	Naive Bayes.
NBP	Naive Bayes Ponderado.
OLAP	On Line Analytical Processing. Se utiliza dentro del área de la inteligencia de negocios y su finalidad es agilizar la consulta de grandes cantidades de datos.
OLTP	On Line Transaction Processing. Es un tipo de procesamiento que facilita y administra aplicaciones transaccionales.
Rbs	Redes Bayesianas
SQL	Structured Query Language. Es un lenguaje específico de bases de datos relacionales que permite especificar diversos tipos de operaciones en ellos
SVM	Support Vector Machine

Funciones utilizadas en R

Este Trabajo Fin de Máster se ha implementado utilizando librerías y funciones del lenguaje R, el cual, es uno de los lenguajes más utilizados en investigación dentro de áreas como: Estadística, minería de datos, investigación biomédica, etc.

Para el método propuesto NBP se han utilizado las siguientes funciones de R:

- **Función de naiveBayes**. Dicha función pertenece al paquete e1071 y construye un modelo de naive bayes para un conjunto de datos dado. Está sobrecargada ya que puede tener diferentes tipos de argumentos. En la implementación propuesta a esta función se la invoca de esta manera:

  Model <- naiveBayes(Class~.,data=datos)

  Donde model es el objeto model creado tras aplicar la función naiveBayes, class es la variable que se pretende clasificar, y datos el conjunto de datos sobre el que se construye el modelo.

- **Función information.gain**. Con esta función se obtienen las ganancias de información de cada uno de los atributos, y se invoca de la siguiente manera:

  IG.FSelector <- information.gain(class~,datos)

  Donde IG.Fselector es un objeto creado tras aplicar la función information.gain, class es la variable que se pretende clasificar, y datos es el conjunto de datos sobre el que se construye el modelo.

- **Función predict**. Esta función predice valores basados en un objeto modelo lineal. En el caso propuesto se invoca de la forma:

  Predicción <- predict(model,datos, type="class")

  Predicción es un vector generado después de aplicar la función predict sobre el objeto model previamente creado, datos son el conjunto de datos sobre los que se calcula la predicción y type indica el tipo de predicción que se requiere.

Para el método C4.5 – NB se han utilizado las siguientes funciones de R:

- **Función J48**. Dicha función pertenece al paquete Rweka. Esta función genera árboles de decisión utilizando el algoritmo C4.5. En la implementación de este algoritmo a esta función se la invoca de esta manera:

  fit <- J48(Class~.,data=datos)

  Donde fit es un objeto que se crea tras aplicar la función J48, class es la variable que se pretende clasificar y datos es el conjunto de datos sobre el que se construye el modelo.

- **Función sample**. Esta función toma una muestra aleatoria de un conjunto de datos de un tamaño determinado. En la implementación de este algoritmo a esta función se la invoca de esta manera:

  dataTrain = datos[sample(1:nrow(datos),40,replace=FALSE),]

  Donde dataTrain es la muestra de datos creada tras aplicar la función sample, datos es el conjunto de datos original, 40 es el tamaño de la muestra y replace indica si se hace con reemplazo (si cada elemento puede repetirse).

Para el método FC se utiliza las función de R:

- **Función fanny**. Dicha función pertenece al paquete Cluster. Esta funcíon calcula una agrupación difusa de los datos en k clusters. En la implementación de este algoritmo se invoca de la siguiente manera:

  FANNY <- fanny(Pred,k=3,maxit = 2000)

  donde FANNY es el objeto que crea después de aplicar la función fanny, Pred es el conjunto de datos de prueba, k es el número de clusters que se desea crear y maxit es el número máximo de iteraciones.

Para el método KNN se utilizan las siguientes funciones de R:

- **Normalize**. Se normalizan los datos antes de aplicarles la función knn de acuerdo a

  normalize <- function(x) { return ((x – min(x))/(max(x)-min(x)))}

  Posteriormente, se aplica dicha normalización a los datos de entrenamiento y prueba. En el siguiente ejemplo se consideran conjuntos con solo 4 atributos.

  ```
  data_n <- data.frame(lapply(datos[,c(1,2,3,4)],normalize))
  pred_n <- data.frame(lapply(Pred[,c(1,2,3,4)],normalize))
  ```

- **Función knn**. Dicha función pertenece al paquete class y aplica sobre un conjunto de prueba la predicción del algoritmo knn en base a un conjunto de entrenamiento. Dicho algoritmo se implementa de la siguiente manera:

  Prediction <- knn(train = data_n, test = pred_n,cl= datos$class ,k = 3)

  Donde Prediction es el resultado de aplicar la función knn, train y test son los datos de entrenamiento y prueba respectivamente, cl son las posibles clasificaciones y k es el número de vecinos cercanos considerados para el cual se considera el más óptimo.

Para el método SVM se utiliza la siguiente función de R:

- **Función ksvm**. Esta función pertenece al paquete kernlab y permite aplicar el algoritmo

SVM. Dicha función admite diversos modelos y se aplica de la siguiente forma:

```
model <- ksvm(class~.,data=datos,type ="C-svc",kernel=rbf,C=10,prob.model=TRUE)
```

donde model es el objeto creado tras aplicar la función ksvm, class indica la variable clasificadora, type = “C-svc” indica que es un problema de clasificación, kernel es la función kernel utilizada siendo en este caso kernel=rbf donde rbf es rbfdot(sigma=0.1) ya que en este caso se construye un modelo basado en una gaussiana, C es el costo de la violación de restricciones, prob.model = True indica que construye un modelo para calcular las probabilidades de clase.

Si en vez de crear un modelo basado en una función gaussiana, se pretendiera crear un modelo basado en un polinomio se utiliza:

```
model <- ksvm(class ~ ., data = datos, kernel = "polydot", kpar = list(degree = 3), C=1 ,cross = 3)
```

donde kernel = “polydot” indica que es un polinomio, kpar = list(degree = 3) indica que es de grado 3 y cross = 3 está relacionado con la validación cruzada. Por último, para crear un modelo lineal

```
model <- ksvm(class~.,data=datos,type="C-svc",kernel="vanilladot",C=5,scaled=FALSE)
```

donde kernel = “vanilladot” indica que se pretende construir un modelo lineal. Para cada una de las colecciones se trata de buscar el modelo que mejor se ajuste a la distribución de los datos.