Aplicación de técnicas múltiples de minería de datos para toma de decisiones en un caso real de gestión de reclamaciones de una compañía de seguros

by

Paola Santana Morales

A thesis submitted in conformity with the requirements
for the MSc in Economics, Finance and Computer Science

University of Huelva & International University of Andalusia

September 2022

APLICACIÓN DE TÉCNICAS MÚLTIPLES DE MINERÍA DE DATOS PARA TOMA DE DECISIONES EN UN CASO REAL DE GESTIÓN DE RECLAMACIONES DE UNA COMPAÑÍA DE SEGUROS

Realizado por:

Paola Santana Morales

Máster en Economía, Finanzas y Computación

Supervisado por:

Antonio J. Tallón Ballesteros

Universidad de Huelva & Universidad Internacional de Andalucía

2022

Agradecimientos

A mi tutor, Antonio Tallón, por su dedicación, tiempo y conocimientos; por su apoyo y consejos que han hecho que este trabajo saliese adelante.

A mis padres y hermana, por sus consejos, confianza y apoyo día a día en la distancia.

A mis amigos de toda la vida, que son parte de mí.

A los amigos que he tenido la oportunidad de hacer en Huelva durante el Máster que, a partir de ahora, serán insustituibles.

Paola Santana Morales

Huelva, 2022

Abstract

This Master's Thesis deals with the application of Machine Learning Techniques, with special emphasis on Data Preprocessing Techniques from the point of view of balancing instances in classes and the imputation of missing values in a real problem of the insurance company BNP Paribass Cardiff. The database classes refer to whether the approval of a claim can be accelerated or whether it requires additional information for processing. The development of the project was divided into two phases. First, two data rebalancing techniques were compared: undersampling and a variant of the oversampling technique called SMOTE. The next phase focused on comparing two missing value imputation techniques using firstly the Expectation Maximization and secondly the replacement of missing values using mean and mode, depending on the nature of the attribute in question. The empirical study focused on obtaining classifiers based on decision trees, bayesian and lazy, which were evaluated using metrics such as the area under the ROC curve called AUC, Cohen's kappa coefficient and accuracy, among others. It is concluded that, on average, better performance is obtained with the classifier based on the *Hellinger* decision tree, highlighting that better test results are acquired with the dataset that has not been subjected to the data preparation process.

JEL Classification: G14, G22, O31, O33.

Key Words: Machine Learning, Data Preprocessing Techniques, Undersampling, SMOTE, Missing Value Imputation, Expectation Maximization.

Resumen

Este Trabajo Fin de Máster (TFM) aborda la aplicación de Técnicas de Aprendizaje Automático, con especial énfasis en Técnicas de Preprocesamiento de Datos desde el punto de vista del balanceo de instancias en clases y la imputación de valores perdidos, en un problema real de la compañía de seguros BNP Paribass Cardiff. Las clases de la base de datos hacen referencia a si la aprobación de una reclamación puede acelerarse o si en cambio requiere información adicional para su tramitación. El desarrollo del TFM se ha distribuido en dos fases. En primer lugar, se compararon dos técnicas de rebalaceo de datos: submuestreo (*undersampling*) y una variante de la técnica de sobremuestreo (*oversampling*) denominada SMOTE, *Synthetic Minority Oversampling TEchnique*. La siguiente fase se centró en comparar dos técnicas de imputación de valores perdidos utilizando en primer lugar el Algoritmo de Maximización de la Esperanza Matemática y en segundo lugar la sustitución de los valores perdidos mediante la media y la moda del atributo, según la naturaleza del mismo. El estudio empírico se dedicó a obtener clasificadores basados en árboles de decisión, bayesianos y perezosos, los cuales se evaluaron mediante métricas como el área bajo la curva ROC (*Receiver Operating Characteristic curve*) denominada AUC, el coeficiente kappa de Cohen y la exactitud (*Accuracy*), entre otras. Se concluye que, en media, se obtiene mejor rendimiento con el clasificador basado en el árbol de decisión de *Hellinger* destacando que se adquieren mejores resultados de test con el conjunto de datos que no han sido sometidos al proceso de preparación de los datos.

Tabla de contenidos

Índice de figuras

Índice de tablas

1

Introducción

1.1. Preliminares

El Aprendizaje Automático, *Machine Learning* en inglés, creado por Arthur Samuel [67] en 1959, es un término general que se refiere a una amplia gama de algoritmos que realizan predicciones a un conjunto de datos de gran tamaño. En los recientes años se ha convertido en una herramienta poderosa no sólo por parte de los investigadores del campo de la Inteligencia Artificial, *Artificial Intelligence* [22], sino también por parte de expertos en otras áreas que emplean estos métodos para realizar sus propios objetivos [37], además que son promovidas por el crecimiento exponencial del volumen de datos [85]. Una de las transformaciones más significativas de la vida en el último medio siglo se debe a la informática y la tecnología digital [6]. Un resumen de las aplicaciones del Aprendizaje Automático se muestra en la Figura 1.1.

Según los diferentes requisitos del conjunto de datos, así como el objetivo que presente la tarea a llevar a cabo, las Técnicas de Aprendizaje Automático (TAA) [96] pueden clasificarse en Aprendizaje Supervisado, No Supervisado, Semisupervisado y Aprendizaje por Refuerzo, como muestra la taxonomía de la Figura 1.2. El Aprendizaje Supervisado, *Supervised Learning*, tiene como objetivo construir o entrenar un modelo que utilice datos ya etiquetados. En general, el Aprendizaje Supervisado se utiliza tanto en algoritmos de regresión, cuando el atributo de la clase es numérico, como en algoritmos de clasificación, cuando este atributo es categórico. En cambio, el Aprendizaje No Supervisado, *Unsuper-*

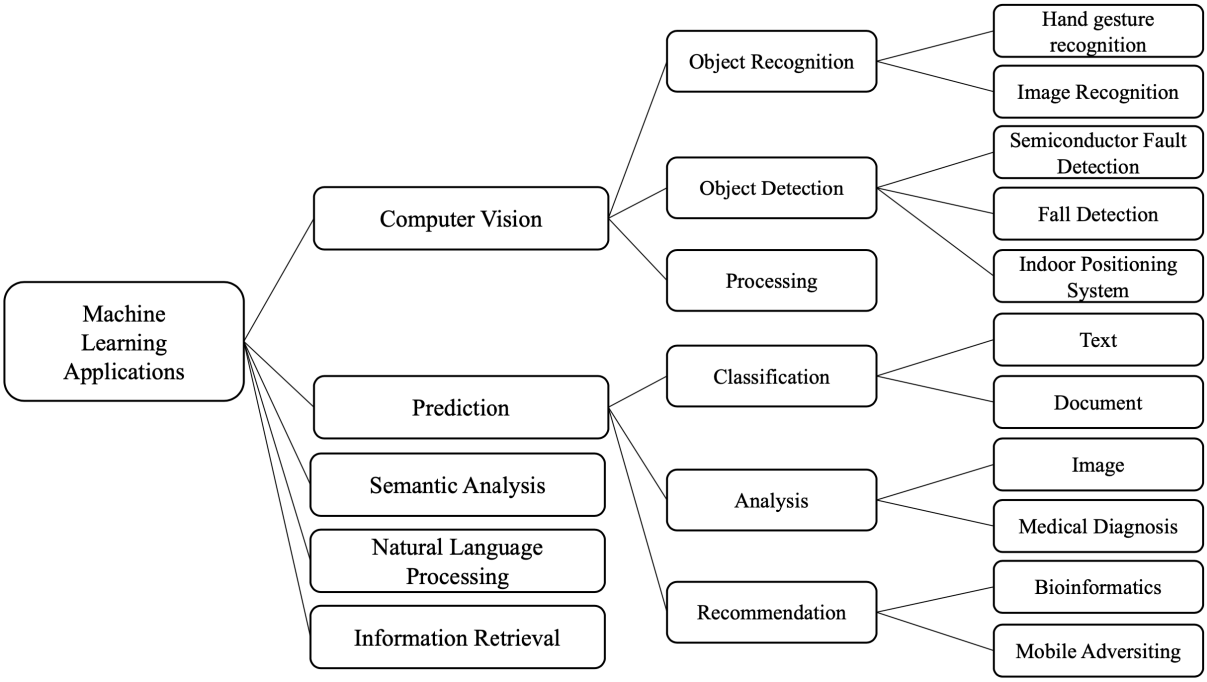

Figura 1.1. Aplicaciones del Aprendizaje Automático. **Fuente:** [103]

vised Learning, se basa en crear un modelo a partir de datos no etiquetados, incluye la agrupación y la reducción de la dimensión, para encontrar una serie de características o comportamientos en el conjunto de datos que permita posteriormente clasificarlos. El Aprendizaje Semisupervisado, *Semi-Supervised Learning*, es una estrategia que combina las ventajas del aprendizaje supervisado junto con las del no supervisado. Finalmente, el Aprendizaje por Refuerzo, *Reinforcement Learning*, es un método orientado a la recompensa con el que los agentes de refuerzo realizan acciones óptimas para conseguir la máxima recompensa.

Un problema de clasificación surge cuando es necesario asignar un objeto o instancia a una clase o grupo determinado que está previamente definido. Los procedimientos tradicionales de clasificación estadística [59] (p. ej., análisis discriminante) se basan en la teoría bayesiana. Estos procedimientos diseñan una función de probabilidad que se utilizará para la toma de decisiones de la clasificación. Sin embargo, en este tipo de técnicas se exige que los datos cumplan una serie de condiciones para que se adapten bien al modelo

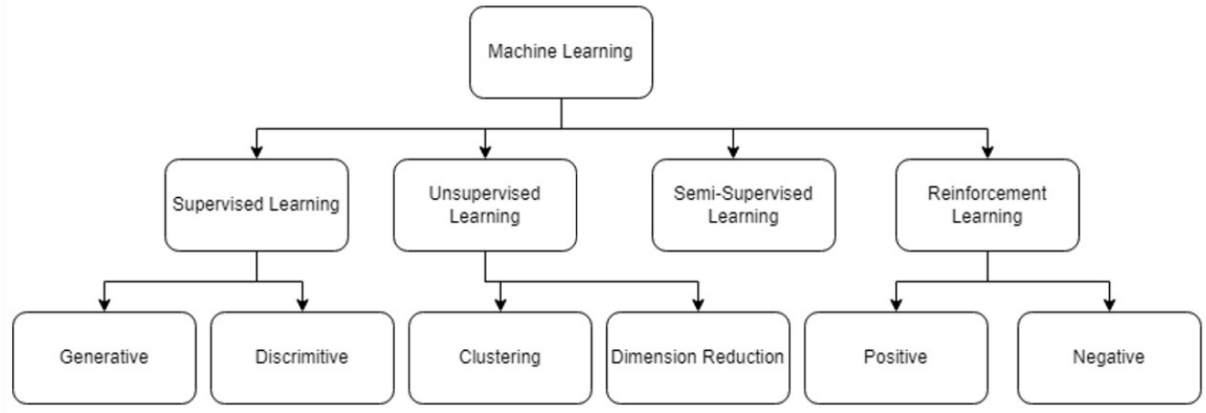

Figura 1.2. Taxonomía de las Técnicas de Aprendizaje Automático. **Fuente:** [83]

probabilístico al cual son sometidos. Asimismo, se debe conocer la naturaleza, comportamiento y propiedades de los datos para que la clasificación sea lo suficientemente precisa [123].

Las organizaciones, tanto públicas como privadas, han comprendido que el Análisis de Datos (*Data Analysis*, *en inglés*) es una herramienta esencial para conocer cómo mejorar su modelo de negocio y la toma de decisiones [15]. Las entidades financieras han sido pioneras en utilizar el Análisis de Datos y el Aprendizaje Automático. La mayoría de estas entidades disponen de diferentes servicios para los clientes, como el seguimiento de los datos para la apertura de una cuenta de ahorro, servicios de préstamos, de inversión o servicios de seguros [48] que utilizan la Minería de Datos para su desarrollo. Otras aplicaciones financieras del Aprendizaje Automático y la Minería de Datos son la predicción de acontecimientos financieros futuros como los mercados de valores, los tipos de cambio, la quiebra, el análisis financiero y de inversión predictivo, el comercio de futuros y la gestión de riesgo en los bancos [33]. Actualmente, las instituciones financieras y la mayoría de los bancos están invirtiendo en la tecnología de la información para aportar técnicas de Minería de Datos y Aprendizaje Automático para gestionar los conjuntos de datos con el fin de operar con el mayor éxito posible en presencia de un negocio competitivo [74].

1.2. Estado del arte

El sector de seguros [28] es una fuente de multitud de problemas empresariales que competen a un investigador operativo: evaluación de riesgos, clasificación de los asegurados u optimización de la cartera, entre otros. Dichos asuntos pueden resolverse mediante técnicas tradicionales de investigación operativa como regresión o programación lineal [104]. La tecnología de Minería de Datos puede aplicarse en diversos campos, como el aprendizaje en el aula [63], los recursos comunitarios [98], la gravedad de los accidentes de autobús [95] y otros campos relacionados. Sin embargo, basándose en una revisión de la literatura publicada, existen pocos estudios dedicados al estudio de los seguros que utilicen Minería de Datos [27]. No obstante, existen proyectos como [7, 114] donde se han aplicado estas técnicas para la detección de fraudes, el descubrimiento de riesgos en los seguros o la mejora de los servicios a los clientes [61].

Uno de los primeros estudios que implicaba la Minería de Datos en el sector de los seguros que realizó en 1999 por Edwin Pednault et al. [7]. En esta investigación, los autores recabaron información de las pólizas y siniestros de seguros de propiedad y accidentes de *International Business Machines Corporation* (IBM) para construir modelos predictivos de los riesgos de los seguros, de manera que utilizaron como núcleo la Minería de Datos para descubrir reglas de caracterización del riesgo mediante el análisis de grandes conjuntos de datos cuya característica más destacada era la presencia de mucho ruido.

En el año 2000, Kate Smith, Robert J. Willis y Malcolm Brooks [104] estudiaron dos problemas. El primero de ellos era la creación de patrones de retención de clientes mediante la clasificación de los asegurados según la probabilidad de renovar o anular sus pólizas. El segundo se centraba en comprender los patrones de siniestralidad e identificar los tipos de asegurados que sufrían más riesgos. Para resolver dichas situaciones, se utilizó un enfoque de Minería de Datos que considera el proceso de descubrimiento de conocimiento dentro de un marco integral utilizando test de hipótesis, árboles de decisión y redes neuronales de múltiples etapas.

Posteriormente, en el año 2004, Arnold F. Shapiro [101], se centró en aplicar técnicas de lógica difusa en áreas de seguros como la clasificación, la suscripción, la fijación de precios, la asignación de activos y las inversiones. La principal herramienta que utilizó fue el *clustering* siguiendo *c-means*.

En el año 2005, Roosevelt Mosley en [82] describe cómo se puede mejorar la situación financiera de una compañía de seguros utilizando técnicas como los árboles de decisión, la modelización lineal generalizada o la regresión logística. Asimismo, en el sector de seguros se han aplicado técnicas de Inteligencia Computacional [100] como redes neuronales [124], lógica difusa [122] y algoritmos genéticos [13].

Un estudio experimental sobre la detección del fraude en el seguro de invalidez individual en 2007 [86] demostró que los modelos predictivos de Bayes Naïve (o *Naïve Bayes* en inglés) superaban a los modelos de árbol de decisión y de Programación Lineal de Criterios Múltiples en términos de precisión de la clasificación. Al año siguiente, 2008, en [61], Anna Jurek y Dannuta Zakszewska plantean el uso de técnicas de Minería de Datos para clasificar las solicitudes de seguros de vida como buenas o malas desde el punto de vista del riesgo asumido por la compañía.

En 2010, Ming-Chang Lee y Chang To [72] aplicaron nuevas técnicas de Minería de Datos para evaluar las dificultades de las empresas y predecir la solvencia de estas; utilizando los datos de una empresa de seguridad de Taiwán. Se mejoró el rendimiento de los algoritmos utilizando Máquinas de Vectores Soporte, *Support Vector Machine (SVM)*, con validación cruzada en tres ocasiones y la Red Neuronal de Retropropagación, *Back Propagation Neural Network*, para cuatro atributos medidos. Se demostró que la SVM obtenía mejores resultados que la BNP.

En el año 2012 existieron varios estudios que aplicaron técnicas de Minería de Datos en el sector financiero y de seguros. En [29] se estudiaron técnicas de árboles de decisión para el análisis del riesgo crediticio, cuyo objetivo era reducir los errores manuales en el banco. En la investigación [78] los autores sugirieron un modelo basado en árboles

de decisión para la evaluación de créditos. El objetivo de este estudio era identificar los factores necesarios para que un banco rural de Bali evaluase las solicitudes de crédito. En [26], Chen et al. utilizaron una técnica de Minería de Datos híbrida para construir un modelo de puntuación de crédito para evaluar el riesgo de este. La investigación se produjo en dos etapas de las cuales la primera fue una etapa de agrupación utilizando la técnica *K-means*; mientras que la segunda, utilizó la SVM para la clasificación. En este mismo año, en [84] se propuso una nueva medida de distancia con datos de panel reales sobre la solicitud de tarjetas de crédito en China; los autores utilizaron el método de agrupación *K-means*.

En 2015, Ajay Byanjankaret et al. [19] describieron la aplicación de Redes Neuronales Artificiales, *Artificial Neural Networks*, en la construcción de modelos de puntuación de crédito en los préstamos entre particulares, cuyo fin era ganar una cuota de mercado en la industria financiera. Este estudio permitió obtener resultados prometedores en la clasificación de las solicitudes de crédito y así poder predecir el riesgo de crédito.

Recientemente, en el año 2022, en el estudio realizado por [27], se compararon 10 modelos de predicción de clasificación mixta utilizando una base de datos de clientes de seguros. La metodología de preprocesamiento utilizada fue la selección de atributos y la discretización de datos. Los principales objetivos de dicha investigación eran: encontrar el mejor clasificador, encontrar el modelo de clasificación mixto óptimo o comparar el rendimiento de los diferentes métodos de validación cruzada de datos, entre otros.

1.3. Objetivos

Los objetivos del presente Trabajo Fin de Máster (TFM) son realizar una revisión del estado del arte de la aplicación de Técnicas de Preprocesamiento de Datos y de Aprendizaje Automático en el sector financiero y de seguros; comparar dos técnicas de rebalanceo de datos: submuestreo (*undersampling*) y sobremuestreo (*oversampling*); así como la comparación de dos técnicas de imputación de valores perdidos: la sustitución mediante el

algoritmo de la maximización de la esperanza matemática y la imputación de valores perdidos, para cada atributo, por su media y moda, según corresponda. Para la realización de este estudio se aplicarán clasificadores basados en árboles decisión, bayesianos y perezosos.

2

Descripción de la base de datos

La base de datos estudiada proviene de la compañía internacional de seguros BNP Paribass Cardiff. Está formada por 114,321 instancias, 132 atributos y dos clases. La clase 0 hace referencia a las reclamaciones cuya aprobación podría acelerarse, de forma que se agilizarían los pagos; y la clase 1, a las reclamaciones para las que se requiere información adicional antes de su aprobación. Se aplicarán, por tanto, técnicas de clasificación binaria.

Cabe destacar que la base de datos contiene una gran cantidad de valores perdidos.

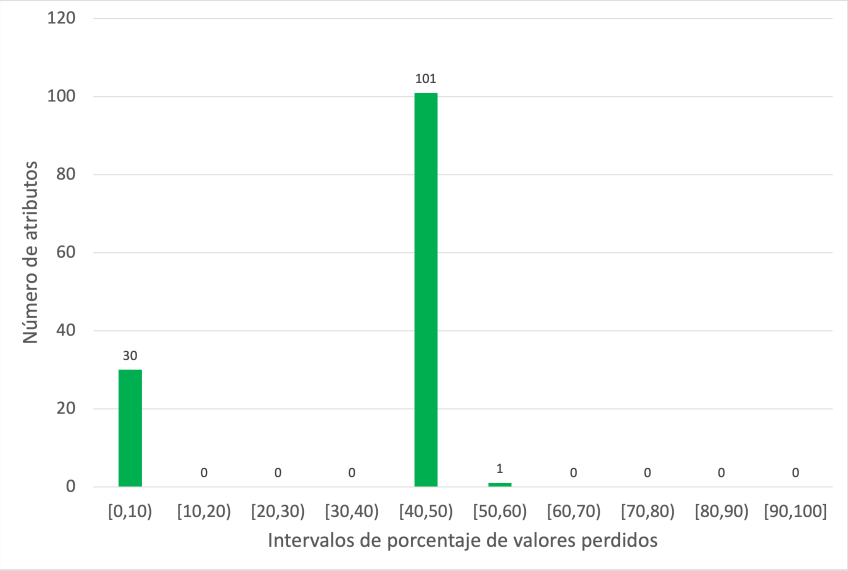

Figura 2.1. Distribución porcentual de valores perdidos en la base de datos. **Fuente:** Elaboración propia

En la Figura 2.1 se muestra la distribución porcentual y la frecuencia de atributos. Adicionalmente, el detalle de cada uno de los atributos puede observarse en la Figura A.1 en el Anexo. Nótese que 101 atributos de los 132 que existen en la base de datos tienen más del 40% de valores perdidos. Por esta razón, tratar este problema será esencial en el TFM. Esta situación es posible abordarla mediante la imputación de valores perdidos, dejando al margen técnicas más simples y elaboradas como la eliminación de instancias y atributos con valores perdidos. Para ello, se compararán dos técnicas: imputación de valores perdidos utilizando el algoritmo de maximización de la esperanza matemática, *Expectation-Maximization (EM)* en inglés; e imputación por la media y moda (MM, en adelante), descritas en el capítulo siguiente.

Otra de las propiedades significativas de esta base de datos es que las clases están muy desbalanceadas, según refleja la Figura 2.2. La barra azul (con 27300 instancias) representa la clase 0, mientras que la barra roja (con 87021 instancias) hace referencia a la clase 1. El problema del desbalanceo de clases puede provocar resultados poco eficientes e imprecisos [110]. Por esta razón, resolver esta cuestión será uno de los objetivos clave en el preprocesamiento de los datos.

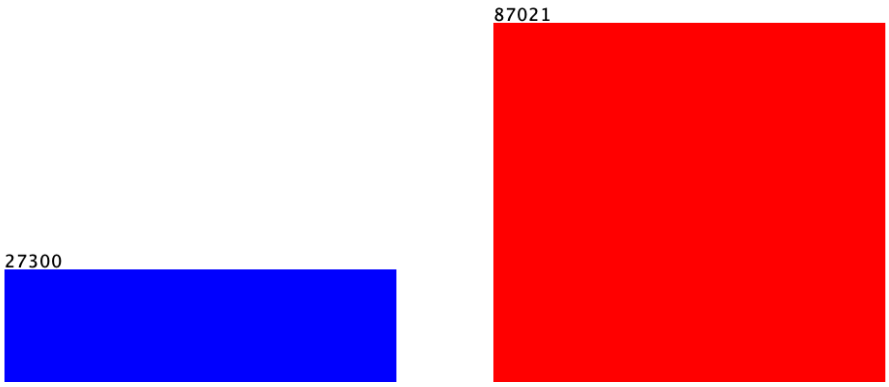

Figura 2.2. Distribución de las instancias en las diferentes clases de la base de datos.
Fuente: Elaboración propia

El desbalanceo de instancias en las clases se puede abordar de diferentes formas. Una de las opciones es el sobremuestreo de la clase minoritaria, lo que es conocido como *over-*

sampling; y la otra es el submuestreo de la clase mayoritaria, que se denomina *undersampling*. En particular, aplicaremos la técnica de SMOTE [24], *Synthetic Minority Oversampling Technique*, que se trata de una alternativa del *oversampling*, y la técnica de *undersampling*. Cabe recordar que, el objetivo principal será concluir qué conjunto de técnicas de preprocesamiento obtienen mejor rendimiento para la base de datos a tratar.

Otra propiedad destacable de la base de datos es que la mayoría de los atributos son numéricos, en concreto 113 de los 132. Sin embargo, los atributos nominales toman los mismos valores. En particular, estos valores están formados por una o varias letras del alfabeto, lo que podría corresponder a diferentes localizaciones donde se producen las reclamaciones.

Tabla 2.1. Muestra de posibles valores que puede tomar el atributo nominal v125

Valor	Número de apariciones
AU	807
AF	831
AE	645
CJ	1068
Z	1191
X	635
BJ	3333
BY	2477
S	609
E	1906
AR	1650
AM	322
AQ	351
AZ	1838
U	817
CD	1515
Total	19995

Los atributos v22 y v125 muestran un número de posibles valores muy superior al resto de los atributos nominales, 18210 y 90, respectivamente. En la tabla 2.1 se muestran

16 de los 90 posibles valores que puede tomar el atributo v125 junto con el número de apariciones que presentan en la base de datos y su peso.

El *grid search* [40] se utiliza para encontrar los hiperparámetros óptimos del modelo que obtenga las predicciones más precisas. El uso del *grid search* que aquí hacemos, en el diseño experimental previo, es más abstracto y el objetivo es encontrar qué técnica de rebalanceo, según el clasificador utilizado, es más apropiada en el conjunto de datos. A modo de ejemplo, la Tabla 2.2 muestra los resultados obtenidos en el *grid search* para el clasificador C4.5 con la métrica de precisión *Accuracy*.

Tabla 2.2. Ejemplo de *grid search* para el clasificador C4.5.

Técnica de Rebalanceo	Porcentaje	Número de vecinos	Media	
			EM	MM
Undersampling	25	5	75,089	75,837
Undersampling	50	5	66,297	68,385
SMOTE	100	5	72,089	68,535
SMOTE	150	5	69,624	70,175
			EM vs. MM (W/T/L): 1/0/3	

Partiendo de los promedios de la Tabla 2.2, se construye un gráfico para cada técnica de imputación de valores perdidos aplicada. En particular, en la Figura 2.3 la técnica aplicada ha sido la imputación por el algoritmo de la maximización de la esperanza matemática. De esta forma, se observa qué porcentaje de qué técnica obtiene mayor media y dicha técnica será la que se aplique al conjunto de datos inicial. En este caso, la técnica que se aplicaría sería *undersampling* con el 25% y 5 vecinos (este dato se observa en la Tabla 2.2).

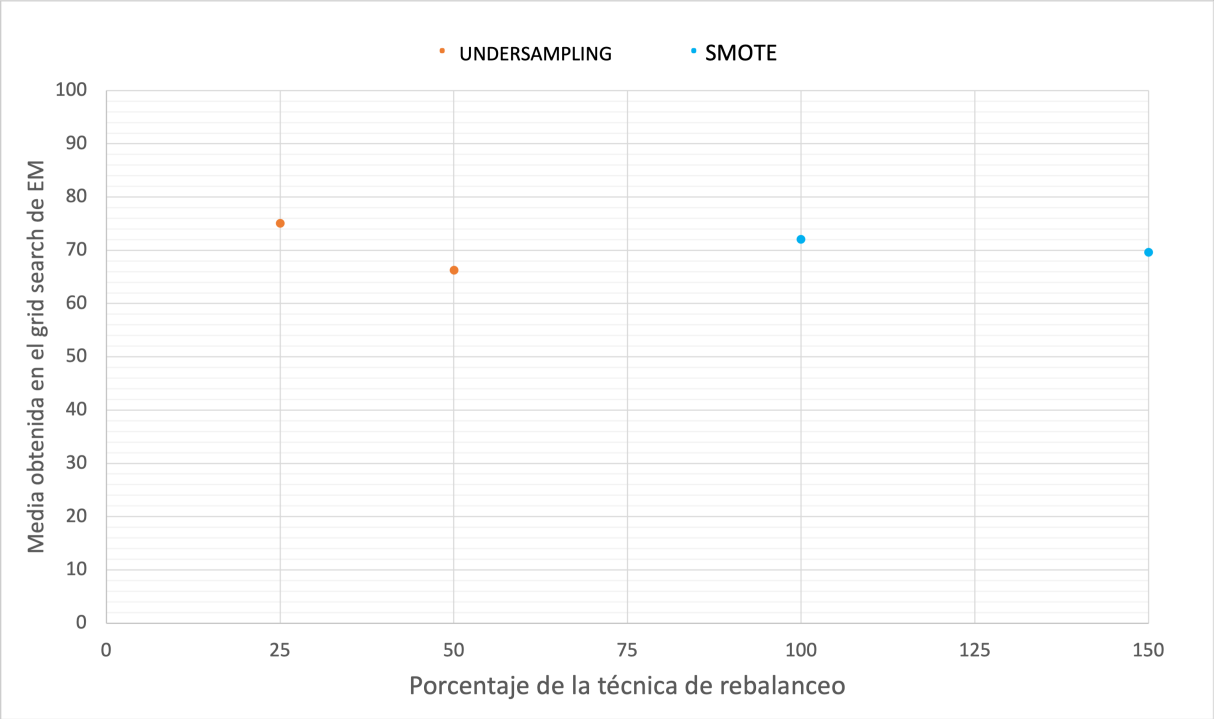

Figura 2.3. *Grid search* para EM con el clasificador C4.5.

Fuente: Elaboración propia

3

Marco teórico

En este capítulo se pretenden dar a conocer las nociones básicas teóricas sobre las técnicas de preprocesamiento a los que son sometidos los datos, así como los algoritmos de aprendizaje automático que se aplicarán y las métricas que se tendrán en cuenta para evaluar estos modelos.

Es conveniente explicar las diferencias entre los tres conceptos nombrados anteriormente. Las técnicas de preparación de datos tienen como objetivo corregir las inconsistencias de los datos que serán la base de análisis en procesos de Minería de Datos. Con el preprocesamiento se pretende que los datos que se van a utilizar tengan coherencia [52], sean veraces, describan fielmente el problema y tengan "calidad" como objetivo final [10] y, por tanto, generarían precisiones de calidad. Esta etapa se encarga de la limpieza de los datos, su integración, transformación y reducción de la dimensión para la siguiente fase de Minería de Datos [41]. En cambio, los algoritmos de Aprendizaje Automático son los algoritmos que utilizan los datos preprocesados anteriormente para aprender de forma autónoma a realizar una tarea o hacer predicciones con el fin de mejorar su rendimiento con el tiempo. Una vez entrenado este algoritmo de Aprendizaje Automático, será capaz de predecir con objetos desconocidos. Finalmente, las métricas de evaluación del modelo se centran en medir la precisión de la generalización de un modelo sobre datos no evaluados con anterioridad. Entre ellas destacan la métrica *Accuracy*, la matriz de confusión (*confusion matrix*) o la kappa de Cohen (*Cohen's kappa*) [30].

3.1. Técnicas de preprocesamiento de datos

La etapa de preprocesamiento de datos en el Aprendizaje Automático es una parte esencial que debe realizarse antes de la creación de cualquier modelo de predicción [8]. Existen multitud de técnicas de preparación de datos que abordan diferentes problemas como el gran volumen de atributos, instancias, valores atípicos y valores perdidos [69].

La Metodología de Minería de Datos para Aplicaciones de Ingeniería [57], *Data Mining Methodology for Engineering Applications* (DMME), es el enfoque más actualizado que abarca el proceso de preparación de los datos para la creación de un modelo de predicción [99]. Se trata de una extensión holística del modelo CRISP-DM, *Cross Industry Standard Process for Data Mining*. Los pasos adicionales pueden clasificarse convenientemente como las fases "Comprensión técnica y conceptualización" y "Realización técnica y pruebas", así como una tarea adicional "Implementación técnica" en la fase de despliegue en el flujo de trabajo de CRISP-DM. En la Figura 3.1 se observan los pasos del DMME.

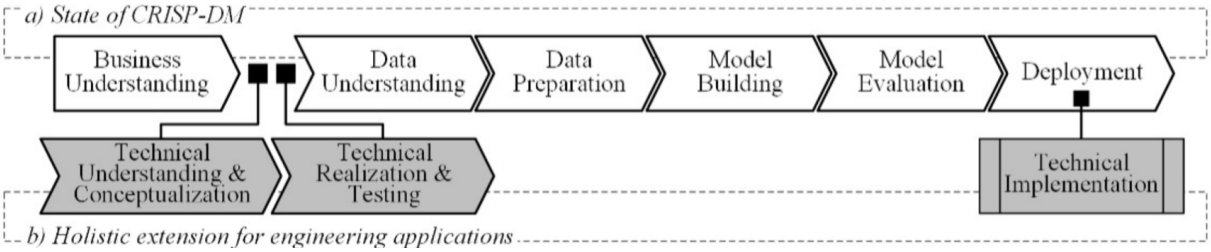

Figura 3.1. Fases del modelo de referencia *Data Mining Methodology for Engineering Applications* (DMME). a) Estado del *Cross Industry Standard Process for Data Mining* (CRISP-DM).
b) Extensión holística para la aplicación en ingeniería.

Fuente: [116]

Técnicas de imputación de valores perdidos

Una de las principales tareas en el preprocesamiento de datos [76] es el estudio de valores ausentes, valores perdidos o *missing values*. Los analistas deben identificar la presencia de estos y llevar a cabo las acciones necesarias para minimizar los efectos que puedan causar [87].

En la base de datos a tratar, como se comentó en el capítulo anterior, existe un gran número de valores perdidos. Para resolver este problema es necesario seleccionar un método adecuado, que dependerá de las particularidades de dicha base de datos. Esto conlleva a la necesidad de un estudio detallado sobre los métodos de reemplazo [5]. Algunas técnicas, como las descritas en [93] imputan los valores perdidos considerando los atributos de forma independiente a los demás. Otras técnicas más complejas utilizan las redes bayesianas, la regresión, los árboles de decisión y otros métodos descritos en [36, 55, 43, 71].

En particular, en el presente TFM se aplicarán dos técnicas: i) sustituir por la media si el atributo es numérico, o por la moda si el atributo es categórico [80]; y ii) sustituir el valor ausente utilizando el algoritmo de maximización de la esperanza matemática (EM).

El primero de estos algoritmos es una técnica de imputación en la que las observaciones ausentes se sustituyen por la media del valor observado, para variables numéricas y, por la moda, para variables nominales [97]. Sea x un atributo de la base de datos en cuestión, y denótese por x_{mis} a la observación de dicho atributo ausente, y por x_{obs} a la observación sin dato ausente. Sea n el número de observaciones de un atributo y r el número de observaciones de valores perdidos, el método de sustitución de la media seguirá fórmula 3.1.

$$\bar{x}_{obs} = \frac{\sum_{i=1}^{n-r} x_{obs,i}}{n-r} \tag{3.1}$$

Por otro lado, se ha utilizado el algoritmo de maximización de la esperanza matemática (*Expectation-Maximization Imputation*). Sea x un conjunto de datos que contiene valores ausentes, x_{mis}, y componentes observados, x_{obs}. En cualquier problema que tenga valores ausentes, la función de densidad y la función de probabilidad logarítmica pueden escribirse como muestran las Fórmulas 3.2 y 3.3, respectivamente.

$$f(x|\theta) = f(x_{obs}|\theta) f(x_{mis}|x_{obs}, \theta) \tag{3.2}$$

$$\ell(\theta|x) = \ell(\theta|x_{obs}) + \log f(x_{mis}|x_{obs}, \theta) \tag{3.3}$$

El segundo término de la ecuación 3.3 no puede calcularse ya que el componente x_{mis} es inobservable. El algoritmo EM resuelve este problema calculando la esperanza condicional $Q(\theta|\theta^{(m)})$ de la función logarítmica de probabilidad dados x_{obs} y el ajuste de θ.

$$Q(\theta|\theta^{(m)}) = E_{\theta(m)}(\ell_c(\theta : x)|x_{obs}) \tag{3.4}$$

A continuación, en el siguiente paso se maximiza $Q(\theta|\theta^{(m)})$ para obtener nuevas estimaciones de los parámetros $\theta^{(m+1)}$. Los pasos se repiten hasta obtener la convergencia [1].

Técnicas de rebalanceo de clases

Otro de los aspectos fundamentales a tratar es el desbalanceo de los datos, *Imbalanced Data*. En la mayoría de las aplicaciones de clasificación binaria del mundo real, como la detección de fraudes o la predicción de impagos, el desbalanceo de clases se ha convertido en un gran reto para los analistas de datos [108]. Cuando ocurren este tipo de problemas, los algoritmos de aprendizaje automático predicen erróneamente o pueden estar sesgadas y dar como predicción que todas las muestras, o la mayoría, de la clase minoritaria son de la clase mayoritaria. Sin embargo, la clase minoritaria suele tener gran importancia en los datos. El estudio del desbalanceo de clases ha sido un tema de bastante interés en el Aprendizaje Automático en los últimos años [34]. Por lo tanto, la investigación de la clasificación de datos desequilibrados ha atraído cada vez más la atención de los investigadores [35]. Los métodos para mejorar el rendimiento global del clasificador de datos desequilibrados incluyen principalmente la tecnología de remuestreo a nivel de reconstrucción de datos [120]. Este problema de desequilibrio entre clases influye en el rendimiento de los algoritmos de aprendizaje [51].

En el presente TFM, este problema se aborda aplicando dos técnicas de preprocesamiento: *undersampling* y *oversampling*. La primera de ellas consiste en disminuir las instancias de la clase mayoritaria. La técnica de submuestreo se está utilizando de forma

prominente para manejar el desequilibrio de clases por dos razones: la primera, este tipo de métodos de submuestreo son más rápidos [73, 115] y la segunda, no sufren sobreajuste como las técnicas de sobremuestreo [105, 25, 62]. En particular, la técnica de sobremuestreo que se aplicará será *SMOTE*, en este caso, la clase minoritaria se sobremuestrea tomando cada muestra de dicha clase e introduciendo ejemplos sintéticos a lo largo de los segmentos de línea que unen a los k vecinos más cercanos de la clase minoritaria [24].

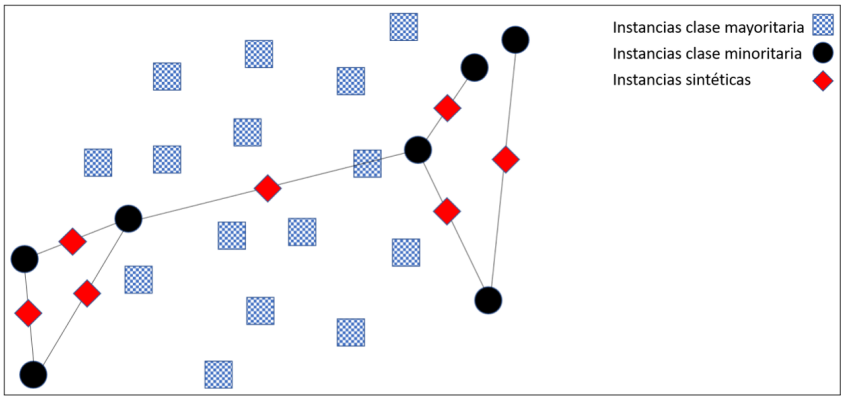

Figura 3.2. Ejemplo de utilización de SMOTE sobre un conjunto de datos no balanceados.
Fuente: [42]

Reducción de datos

La reducción de datos se considera uno de los pasos más importantes en el Descubrimiento de Conocimiento en la Base de Datos (*Knowledge Discovery in Databases*, *KDD*). El KDD se define en [39] como un proceso no trivial de identificación de patrones válidos, novedosos, potencialmente útiles y comprensible en los datos. En este proceso, destacan la selección de atributos y la selección de instancias.

El primero de estos, *Feature Selection (FS)*, desempeña un papel fundamental en los sistemas de clasificación. En general, se utilizan conjuntos de atributos de los cuales algunos son relevantes y otros irrelevantes o redundantes. Desde el punto de vista computacional, utilizar más atributos de los necesarios provoca que el algoritmo tenga que esforzarse más. Por ello, la selección de los atributos relevantes es totalmente recomendable [70]. Los

métodos de selección de atributos se pueden agrupar en tres grandes grupos: métodos filtro (*Filter Methods*), de envoltura [66] (*Wrapper Methods*) y métodos embebidos (*Embedded Methods*). En el primer grupo se encuentran el método de la Chi-Cuadrado [60], ReliefF [64], CFS (*Correlation-base Feature Selection*) [47]. En el segundo, Bayes Naïve [31] y Máquinas de Vector Soporte (*Support Vector Machine, SVM*) [14]. Finalmente, algunos ejemplos de métodos incorporados son CART [16] y C4.5 [89].

Por otro lado, la selección de instancias, *Instance Selection*, puede dar lugar a: reducir la complejidad espacial del problema de clasificación [20], disminuir el tiempo computacional requerido [32], disminuir el tamaño de las fórmulas obtenidas por un algoritmo de inducción en los conjuntos de datos requeridos [75] y acelerar el proceso de extracción de conocimiento [117]. En general, los algoritmos de selección de instancias se basan en el cálculo de la distancia entre observaciones del conjunto de entrenamiento. Algunos algoritmos, como el algoritmo del vecino más cercano condensado [49], *Condensed Nearest Neighbour Algorithm (CNN)*, o el algoritmo basado en dos instancias (IB2) [3], se han propuesto para superar las limitaciones que suponía utilizar funciones de distancias inapropiadas o inadecuadas para atributos lineales y nominales [23].

Detección de valores atípicos

Otro de los aspectos a tratar en el proceso de KDD es la detección de valores atípicos o *outliers*. David Hawkings [50] definió como outlier a una observación que se desvía tanto de otras observaciones que despierta la sospecha de haber sido generada por un mecanismo diferente. Algunas aplicaciones, como la detección de fraudes con tarjetas de crédito y el monitoreo de actividades criminales en el comercio electrónico se benefician con la detección de outliers. La detección de valores atípicos [4] se divide en dos tipos: detección global y local. En el primer caso se tiene en cuenta todo el conjunto de datos de manera que el dato se considera atípico si dicho punto está alejado de los demás [65]. Mientras que, en la detección local de valores atípicos, se realiza dicha búsqueda en un subconjunto de los datos. Un valor atípico local se basa en la probabilidad de que dicho

punto sea atípico en comparación con sus vecinos locales, es decir, utiliza el algoritmo *k*NN [106].

Detección de ruido

Uno de los principales retos de la clasificación es la presencia de ruido en los conjuntos de datos. La presencia de ruido puede aumentar la complejidad del modelo y el tiempo de aprendizaje provocando, además, que el rendimiento de los algoritmos disminuya [94]. Por tanto, es necesario identificar y manejar el ruido presente en el conjunto de datos [45].

3.2. Algoritmos de Aprendizaje Automático

El Aprendizaje Automático se basa en diferentes algoritmos que se utilizan para entrenar un modelo que resuelve un problema. El tipo de algoritmo empleado depende de la naturaleza del problema a resolver o el número de variables, entre otros [77]. Los principales algoritmos de clasificación [119] se dividen en Árboles de Decisión, Técnicas Perezosas, Bayes Naïve, Técnicas basadas en Reglas y Técnicas de Vector Soporte (TVS).

Algoritmos basados en Árboles de Decisión

Los árboles de decisión son algoritmos de partición recursiva que dan lugar a una estructura en forma de árbol que representa patrones en un conjunto de datos [9]. Los algoritmos de árboles de decisión consisten en averiguar la forma en la que se comporta el vector de atributos para un número de instancias. Este tipo de algoritmos genera las reglas para la predicción de la variable clase. La creación del árbol de decisión se lleva a cabo aplicando el algoritmo de Hunt [102] que se basa en la división en subconjuntos que buscan la separación óptima. Dado un conjunto de datos de entrenamiento de un nodo, si todos ellos pertenecen a una misma clase, dicho nodo se denomina terminal. En cambio, si los datos pertenecen a distintas clases, se vuelve a dividir en subconjuntos más pequeños en función de una variable y se repite el proceso. Dicha variable se puede elegir considerando el Índice de Gini o la Entropía.

El Índice de Gini mide el grado de pureza de un nodo, es decir, mide la probabilidad de no obtener dos registros de la misma clase en un nodo. Se seleccionará la variable con menor índice de Gini ponderado ya que cuanto mayor sea este, menor es la pureza del nodo. Para calcular dicho índice se sigue la Fórmula 3.5, siendo P_i la probabilidad de que un dato pertenezca a la clase i. En cambio, la Entropía es una medida que se aplica para cuantificar el desorden de un sistema. Si un nodo es puro su entropía será 0 de manera que solo tendrá observaciones de una clase, pero si la entropía es 1, la frecuencia de observaciones de cada una de las clases es igual. La Entropía se calcula siguiendo la Fórmula 3.6.

$$Gini(t) = 1 - \sum_{i=1}^{n} (P_i)^2 \tag{3.5}$$

$$H = \sum_{i=1}^{n} P_i \cdot \log_2(P_i) \tag{3.6}$$

En particular, se ha utilizado el algoritmo HTree (Hellinger Tree), propuesto por Geoff Hulten, Laurie Spencer y Pedro Domingos [58]. Este algoritmo se encarga de crear el árbol de Hoeffding y tiene la ventaja de que, como se genera de manera dinámica y continuamente, se logra un mayor nivel de adaptabilidad y oportunidad en la obtención de los modelos correspondientes a la descripción de una realidad temporal.

En un primer momento, se pensó en utilizar también el algoritmo J48 que, en Weka, es una implementación del algoritmo C4.5. Sin embargo, su ejecución requería una gran cantidad de memoria por lo que no se podía ejecutar en el equipo donde se han realizado los experimentos. Las características adicionales de J48 son la contabilización de los valores perdidos, la poda de los árboles de decisión, los rangos de valores de atributos continuos, etc. Este algoritmo genera las reglas a partir de las cuales se genera la identidad particular de los datos. El objetivo es la generalización progresiva de un árbol de decisión hasta conseguir un equilibrio de flexibilidad y precisión.

Algoritmos basados en Técnicas Perezosas

Los algoritmos de aprendizaje perezoso [2] presentan tres características que los distinguen de otros algoritmos de aprendizaje. En primer lugar, aplazan el procesamiento de sus entradas hasta que reciben las solicitudes de información; simplemente almacenan sus entradas para su uso futuro. A continuación, responden a las solicitudes de información combinando sus datos almacenados. Por último, descartan la respuesta construida y cualquier resultado intermedio. El algoritmo más reconocido en este sector es el método de los vecinos más cercanos [119] (*k Nearest Neighbour*, en inglés), *k*NN. Se trata de un método de clasificación no paramétrico, sencillo pero eficaz en muchos casos. Sin embargo, el éxito de la clasificación depende en gran medida del valor del parámetro k. Por ello, se dice que el método *k*NN [11] está sesgado por el valor de k [44]. Este parámetro puede determinarse automáticamente mediante el enfoque de validación cruzada "leave-one-out" hasta un límite superior especificado [113]. El algoritmo calcula la distancia entre una observación de prueba y de todos los objetos de entrenamiento para determinar la lista de vecinos más cercanos de manera que clasificará el objeto según la clase mayoritaria de dichos vecinos. Sea v la etiqueta de la clase, y_i la etiqueta de la clase del i-ésimo vecino más cercano e I la función indicadora que devolverá 1 si el argumento es verdadero y 0 en caso contrario. El algoritmo utiliza la Ecuación 3.7 para clasificar las instancias de prueba.

$$\text{Clase mayoritaria: } y' = \arg\max_v \sum_{(x_i, y_i) \in D_z} I(v = y_i) \tag{3.7}$$

Algoritmos basados en Bayes

Los algoritmos de Aprendizaje Automático basados en Redes Bayesianas ingenuas (NB) [68] son redes que tienen un solo nodo no observado y varios que sí. Se asume además que hay independencia entre los nodos hijos (observados) que en este caso son los atributos. El enfoque bayesiano para clasificar una nueva instancia consiste en asignar el valor más probable v_{MAP}, dados los valores de los atributos $\langle a_1, a_2, \ldots, a_n \rangle$. Véase la Fórmula 3.8. Utilizando el Teorema Bayesiano, la fórmula anterior se puede reescribir como la 3.9.

Este algoritmo es propenso a verse afectado indebidamente por probabilidades nulas. La suposición de la independencia entre los nodos hijos hace que los clasificadores Bayes sean menos preciosos que otros algoritmos de Aprendizaje Automático más sofisticados. La principal ventaja del clasificador ingenuo de Bayes es su corto tiempo de cálculo para el entrenamiento. Además, como el modelo tiene la forma de un producto, puede convertirse en una suma mediante el uso de logaritmos, con las consiguientes ventajas computacionales. Si un atributo es numérico, se suele discretizar durante el preprocesamiento de datos [121].

$$v_{MAP} = \arg\max \; P\Big(C_j \Big| a_1, a_2, \ldots, a_n\Big) \tag{3.8}$$

$$v_{MAP} = \arg\max \frac{P(a_1, a_2, \ldots, a_n) \cdot P(C_j)}{P(a_1, a_2, \ldots, a_n)} = \arg\max P(a_1, a_2, \ldots, a_n) \cdot P(C_j) \tag{3.9}$$

Algoritmos basados en Reglas

Los algoritmos de Aprendizaje Automático basados en reglas [53] se juzgan según su precisión de clasificación en un conjunto de prueba independiente y su complejidad [53]. Rendell & Seshu [91] señalaron que muchos conjuntos de datos tienen pocas observaciones extrañas o ”picos” y por lo tanto son fáciles de aprender, lo que supone que reglas simples obtengan un nivel de precisión elevado. Por otro lado, los estudios de métodos de poda [17] proporcionan pruebas de que la precisión no disminuye a medida que la poda es menor, aun cuando las reglas se podan al extremo [81]. El artículo [79] se centra en medir el rendimiento de reglas muy simples, denominadas “rules1”, árboles de decisión de un nivel, en conjuntos de datos de uso cotidiano en la investigación del Aprendizaje Automático. Entre los algoritmos basados en reglas destacan *Repeated Incremental Pruning to Produce Error Reduction* (RIPPER), *PARTial Decision Tree* (PART) [12] y *Nearest Neighbor with generalization* (NNge) [92].

Algoritmos basados en Técnicas de Vector Soporte

Finalmente, las Máquinas de Vector Soporte (SVM) son la técnica de Aprendizaje Automático más reciente [18]. SVM crea un hiperplano que separa las clases de datos. El

objetivo principal de dicho algoritmo es maximizar el margen y, por tanto, crear la mayor distancia entre el hiperplano de separación y las instancias del conjunto de datos. La complejidad del modelo de una SVM no se ve afectada por el número de atributos que presente el conjunto de datos de entrenamiento ya que cuando los datos son linealmente separables, la solución se representa como una combinación lineal de los dos puntos que se encuentran en el margen de separación óptimo hallado, ignorando así los demás puntos. Sin embargo, encontrar el hiperplano de separación suele ser complicado ya que los datos contienen instancias mal clasificadas. Este problema suele abordarse utilizando un margen suave que acepte algunas clasificaciones erróneas de las instancias de entrenamiento [111].

Sin embargo, en general, los datos no son separables, por lo que calcular el hiperplano de separación resulta una tarea difícil. Una solución a este problema es mapear los datos en un espacio de mayor dimensión, denominado espacio de características, y definir allí el hiperplano de separación. Con un espacio de características adecuadamente elegido y de suficiente dimensionalidad, cualquier conjunto de entrenamiento consistente puede hacerse separable. Los datos se mapean a otro espacio de Hilbert H tal que $\Phi : R^d \longrightarrow H$, entonces dicho algoritmo solo dependerá de los datos a través de productos en H, es decir, de funciones $\Phi(x_i) \cdot \Phi(x_j)$. Una de las limitaciones más destacable de las SVM es el tiempo de entrenamiento, debido a su lentitud. Nótese que los métodos SVM son binarios, por lo que si se trabaja con un problema multiclase habría que reducir dicho problema a un conjunto de múltiples problemas de clasificación binaria.

3.3. Métricas de evaluación de modelos

Las métricas de evaluación juegan un papel fundamental para conseguir el mejor clasificador durante el entrenamiento del modelo [54]. Según la literatura, estas métricas pueden clasificar en tres tipos: métricas con un determinado umbral, de probabilidad y de clasificación [21].

Accuracy

El rendimiento de los algoritmos de Aprendizaje Automático suele evaluarse mediante la métrica de *Accuracy* (Acc), sin embargo, esta medida tiene ciertas limitaciones. En [90, 118] se demostró que dada la simplicidad de esta métrica podría conducir a soluciones subóptimas, especialmente cuando las clases del problema se encuentran desbalanceadas. Acc mide la proporción de predicciones correctas sobre el número total de instancias evaluadas.

Denótese por t_p y t_n al número de instancias positivas y negativas bien clasificadas, respectivamente. Sea f_p las instancias que son clasificadas erróneamente como negativas. Análogamente, f_n serán las instancias clasificadas como positivas cuando en realidad son negativas. La fórmula 3.10 determina el cálculo de la métrica Accuracy. A medida que aumenta este coeficiente, mayor será la precisión del clasificador utilizado.

$$Acc = \frac{t_p + t_n}{t_p + f_p + t_n + f_n} \tag{3.10}$$

Dado un clasificador aleatorio, f_0 y f_1 denotan a las funciones de densidad de las clases 0 y 1, respectivamente. Por consiguiente, F_0 y F_1 son las funciones de distribución acumulada que se calculan como sigue la Fórmula 3.11, para $i = 0, 1$. Las funciones de distribución acumulada miden la sensibilidad o tasa de positivos de la clase en cuestión.

$$F_i(t) = \int_{-\infty}^{t} f_i(s)\, ds = P(s \leq t|i) \tag{3.11}$$

Análogamente, utilizando las funciones definidas anteriormente, la métrica de precisión *Accuracy* puede calcularse como muestra la Fórmula 3.12.

$$Acc(t) = \pi_0 F_0(t) + \pi_1 F_1(t) \tag{3.12}$$

AUC

Por otro lado, la curva ROC (*Receiver Operating Characteristic*) es una técnica estándar para resumir el rendimiento de un clasificador en un rango de compensaciones entre las tasas de error de verdaderos positivos y falsos positivos [109]. El área bajo la curva (*Area Under Curve*, *AUC*) es una métrica de rendimiento tradicional aceptada para una curva ROC [107]. Esta métrica puede interpretarse como la tasa de verdaderos positivos esperada, promediada sobre todas las tasas de falsos positivos, es decir, mide el rendimiento general de la clasificación.

La curva de ROC se define como un gráfico de $F_1(T)$ en el eje x frente a $F_0(t)$ en el eje y, con ambas cantidades no decrecientes monótonamente al aumentar t. El AUC se calcula mediante una operación de integración [38] como define la Fórmula 3.13.

$$AUC = \int_0^1 F_0(s)\, dF_1(s) = \int_{-\infty}^{+\infty} F_0(s)\, df_1(s) \tag{3.13}$$

Asimismo, cuando se trata de distribuciones empíricas la integral de la Fórmula 3.13 se puede sustituir por una suma. Sea S_p el número total de instancias clasificadas correctamente; y n_p y n_n el número de instancias positivas y negativas clasificadas, respectivamente. El valor AUC puede calcularse como muestra la Fórmula 3.14. Además, el objetivo es que el AUC esté lo más próximo a 1.

$$AUC = \frac{S_p - n_p(n_n - 1)/2}{n_p n_n} \tag{3.14}$$

El objetivo es que el coeficiente de AUC esté lo más próximo a 1 pues de esta manera se maximiza el valor de la integral. Se ha demostrado que el AUC es teórica y empíricamente mejor que el Accuracy para evaluar el rendimiento del clasificador y discriminar una solución óptima durante el entrenamiento [56].

Kappa de Cohen

El coeficiente kappa de Cohen, *Cohen's kappa*, es una medida estadística que mide la concordancia entre evaluadores. Se considera una medida más robusta que el simple cálculo del porcentaje de acuerdo, ya que tiene en cuenta el acuerdo que se produce al azar [112]. Cabe destacar que dicho coeficiente ha sido utilizado durante épocas en el ámbito de las ciencias sociales, la biología y la medicina, sin embargo, en el Aprendizaje Automático y en la Minería de Datos no ha recibido mucha atención como métrica de precisión.

Jacob Cohen [30] introdujo por primera vez esta métrica como una medida de acuerdo entre observadores de comportamientos psicológicos, se pretendía medir el nivel de acuerdo o desacuerdo entre dos personas que observaban el mismo fenómeno. Por tanto, el coeficiente kappa de Cohen mide la concordancia entre dos examinadores en sus correspondientes clasificaciones de N elementos en C categorías mutuamente excluyentes. Sea $Pr(a)$ la probabilidad de acuerdo total y, $Pr(e)$ la probabilidad hipotética de acuerdo con el azar, utilizando los datos observados para calcular las probabilidades de que cada observador diga al azar cada categoría. La ecuación 3.15 muestra cómo se calcula esta medida.

$$K = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{3.15}$$

Los clasificadores con mejor rendimiento deberían tener el coeficiente de la kappa de Cohen mayor ya que a medida que aumenta el acuerdo, dicho valor incrementa.

Recall

La métrica de exhaustividad, *Recall* en inglés, mide el número de observaciones positivas correctamente clasificadas del total de observaciones clasificadas como positivas. La métrica *Recall* se calcula como muestra la Fórmula 3.16.

$$Recall = \frac{t_p}{t_p + f_n} \tag{3.16}$$

Precisión

La medida de precisión, *Precision*, como su nombre indica, mide la calidad del modelo. Es decir, esta métrica mide el número de observaciones positivas clasificadas correctamente del total de positivas clasificadas. Se calcula como muestra la fórmula 3.17.

$$Precision = \frac{t_p}{t_p + f_p} \tag{3.17}$$

Medida F

La medida F se considera una media armónica que combina los valores de la precisión y la exhaustividad. Esta medida se calcula como se muestra en la Fórmula 3.18.

$$F = 2 \cdot \frac{Precision \cdot Exhaustividad}{Precision + Exhaustividad} \tag{3.18}$$

4

Experimentación

La experimentación se ha realizado en un equipo con la siguiente configuración: MacBook Air con procesador de 1,1 GHz Intel Core i3 de doble núcleo, memoria de 8GB 3733 MHz LPDDR4X. Los diferentes experimentos se han realizado en el framework Weka [46] (*Waikato Environment for Knowledge Analysis*) en la versión 3.8.6.

En el diseño experimental, la técnica de validación cruzada empleada ha sido *hold out* estratificado [88], de manera que el conjunto de entrenamiento y de prueba siguen la misma distribución en las clases que el conjunto de datos original. El conjunto de entrenamiento contendrá el 75% de las instancias totales, exactamente 85740. Mientras que, el conjunto de prueba contendrá el 25% restante, 28581 instancias.

La primera técnica aplicada al conjunto de datos de entrenamiento es la imputación de valores perdidos mediante EM y MM. De esta manera se pretende comparar qué técnica de imputación de valores ausentes obtiene mejor rendimiento con la base de datos a tratar. A continuación, se aplicó un algoritmo de *grid search* para encontrar el porcentaje de qué técnica de rebalanceo es óptimo en la base de datos. Todos los clasificadores aplicados utilizaron estos conjuntos de datos de entrenamiento con sus respectivos conjuntos de test. Finalmente, cuando se dedujo qué técnica de rebalanceo es mejor para cada técnica de imputación de valores perdidos según cada clasificador, se repitió el proceso de clasificación utilizando los conjuntos de entrenamiento y prueba iniciales.

El esquema de la experimentación sobre el proceso de preparación de los datos se puede observar en la Figura 4.1. Los conjuntos de entrenamiento ”Fi-entrenamiento” y ”Fi-test”, i=1,2,3,4; se forman mediante la técnica del *grid search* de manera que “F1-entrenamiento/test” corresponde al primer fold, “F2-entrenamiento/test” al segundo, etc. Estos subconjuntos de entrenamiento de entrenamiento se distribuyen, en número de instancias, como se muestra en la Tabla 4.1.

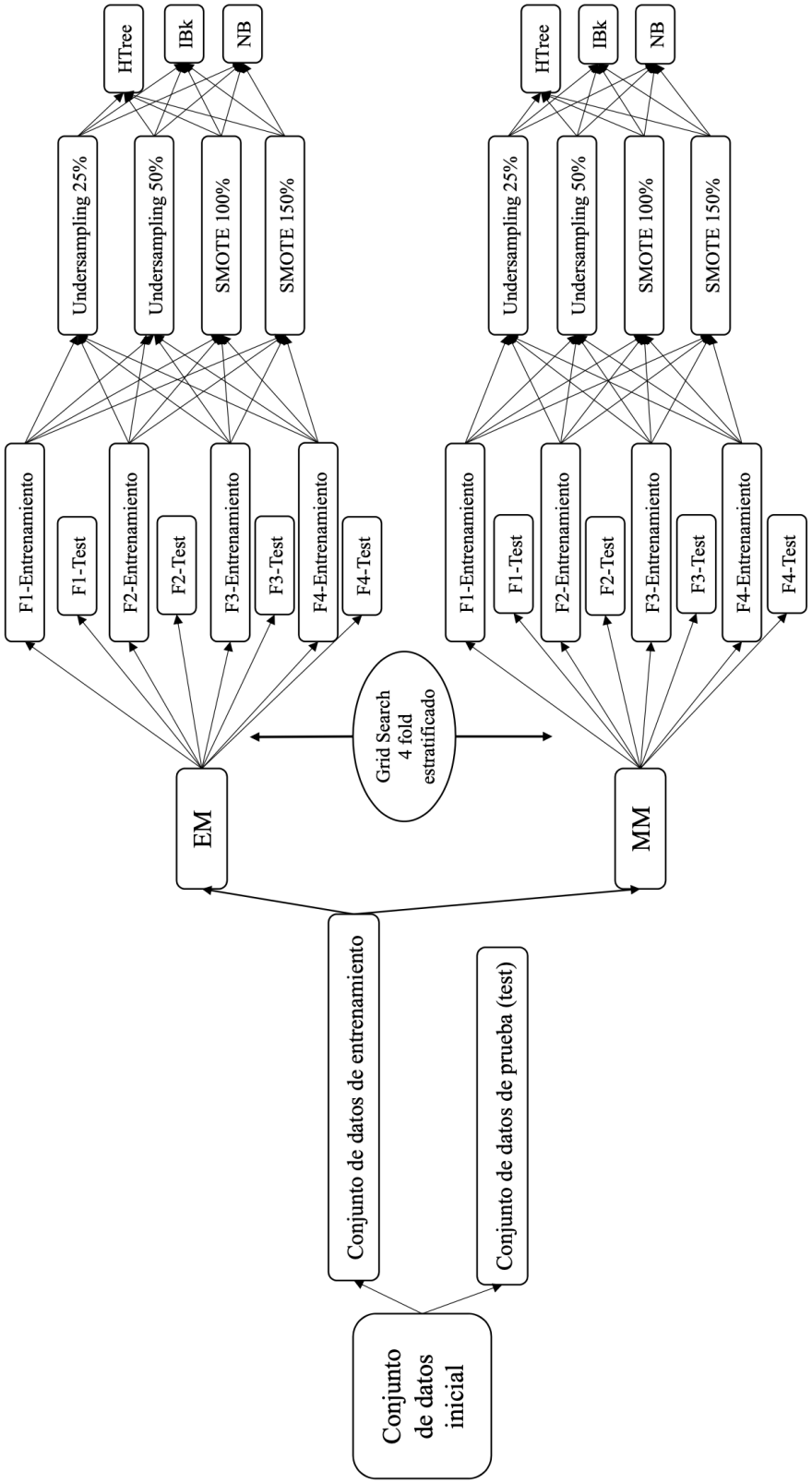

Figura 4.1. Esquema de la experimentación llevada a cabo en el preprocesamiento de datos.
Fuente: Elaboración propia

Por otro lado, los clasificadores que se encuentran en la última etapa de este esquema se entrenan con los conjuntos "Fi-entrenamiento" y se validan con los correspondientes "Fi-test", i=1,2,3,4. La mejor combinación de técnicas de prepocesamiento obtenida empíricamente es la que se utiliza con el conjunto de entrenamiento inicial y se utilizan los datos de test para evaluar dicho clasificador.

Tabla 4.1. Distribución de instancias en las clases tras el *grid search* (i=1,2,3,4)

	Número de instancias		
Conjunto de datos	Clase 0	Clase 1	Total
Fi-Entrenamiento	15357	48948	64305
Fi-Test	5118	16317	21435

Asimismo, la distribución de las instancias en las clases del conjunto de entrenamiento inicial, tras aplicar las técnicas de rebalanceo se muestra en la Tabla 4.2.

Tabla 4.2. Distribución de las instancias en las clases tras aplicar el rebalanceo de datos

	Número de instancias			
Técnica de preprocesamiento	Clase 0	Clase 1	Total	Diferencia[1]
Undersampling (25%)	15357	36711	52068	-12237
Undersampling (50%)	15357	24474	39831	-24474
SMOTE (100%)	30714	48948	79662	+15357
SMOTE (150%)	38392	48948	87340	+23035

[1] Esta columna muestra la diferencia entre el número de instancias del conjunto de datos de partida (conjunto "Fi-Entrenamiento" tras aplicar grid-search) y el conjunto de datos tras haber aplicado la técnica de preprocesamiento que corresponda.

Como se ha comentado con anterioridad, el principal objetivo de la fase de experimentación es encontrar la mejor combinación de técnicas de preprocesamiento aplicadas a los conjuntos de datos resultantes del *grid search* para, posteriormente, aplicar estas técnicas al conjunto de partida inicial. Los clasificadores se han probado con los parámetros por defecto ya que son los que los autores que han implementado estos algoritmos recomiendan. La Tabla 4.3 muestra los parámetros utilizados. Además, dichos parámetros se han

escrito en inglés para no perder la semántica y se han escrito en el orden de introducción del clasificador.

Tabla 4.3. Parámetros de los clasificadores

Clasificador	Parámetro	Valor
HTree	Binary Splits	True
	Collapse Tree	False
	Confidence Factor	0,25
	debug	False
	Max Equal	0,00
	minNumObj	2
	numFolds	3
	Reduce Error Pruning	False
	Save Instance Data	False
	seed	1
	Subtree Raising	False
	unpruned	True
	Use Laplace	True
Naïve Bayes	debug	False
	Display Model In Old Format	False
	Use Kernel Estimator	False
	Use Supervised Discretization	False
IB*k*	Número de vecinos	1
	crossValidate	False
	debug	False
	Distancia (distanceWeighting)	No distancia
	meanSquared	False
	Algoritmo de los vecinos más cercanos	LinearNNSearch
	windowSize	0

Los resultados de los experimentos realizados se muestran en la Tabla 4.4. Cada resultado consiste en la media de la precisión (*Accuracy*) de los cuatro conjuntos de datos "Fi-Entrenamiento" con sus respectivos "Fi-Test", para i=1,2,3,4.

Las mejores combinaciones de técnicas obtenidas se han escrito en negrita. Por tanto, estas combinaciones son las que se comprarán con el conjunto de datos original, veáse el Capítulo 5.

Tabla 4.4. Resultados obtenidos en el diseño experimental previo

Clasificador	Técnica de imputación de valores perdidos	Técnica de rebalanceo (media)			
		Undersampling (25%)	Undersampling (50%)	SMOTE (100%)	SMOTE (150%)
HTree	EM	65,49	61,59	68,57	**76,02**
	MM	66,91	63,86	**67,54**	67,37
Bayes Naïve	EM	**58,79**	56,17	54,17	54,68
	MM	**59,39**	58,97	53,07	55,04
IB*k*	EM	61,12	56,47	**70,72**	70,51
	MM	66,37	62,41	66,02	**66,59**

5

Resultados

Este capítulo presenta los resultados de test, es decir, los resultados obtenidos con el conjunto de datos que muestra la primera etapa de la Figura 4.1. Las métricas que se han recogido han sido la exactitud (*Accuracy*), el AUC, el coeficiente kappa de Cohen, la precisión (*Precision*), la exhaustividad (*Recall*) y la Medida F.

La Tabla 5.1 muestra los resultados de test de las medidas de rendimiento comparando tres situaciones. En primer lugar se aplican los clasificadores al conjunto de datos original, es decir, habiendo aplicado un proceso mínimo de preparación de datos. En las dos siguientes situaciones los datos han estado sometidas a un preprocesamiento de los datos. Como se explicó en capítulos anteriores, con este estudio se pretende comparar los resultados obtenidos sometiendo a los datos, en un primer lugar, a la imputación de valores perdidos mediante el Algoritmo de la Maximización de la Esperanza Matemática (EM) y la sustitución por la media y la moda (MM). A continuación, se aplicó la técnica de rebalanceo de datos (*undersampling* o una instancia de *oversampling* denominada *SMOTE*) más apropiada para cada clasificador según se pudo estudiar en la fase de experimentación. Este último aspecto se estudió aplicando la técnica de *grid search*, considerando las técnicas de preprocesamiento y los clasificadores elegidos para la experimentación (véase Tabla 2.2). Se han aplicado las técnicas en función de la conclusión extraída en el proceso de *grid search* (Tabla 4.4). Los clasificadores han sido probados con los parámetros que vienen por defecto tal como se expuso en el Capítulo 4.

El objetivo principal del TFM ha sido comparar las técnicas de preprocesamiento comentadas anteriormente con el conjunto de datos inicial. En este capítulo de resultados se añaden algunas tablas y figuras que permiten visualizar mejor los resultados obtenidos y decidir, según el clasificador, cual es la técnica de preprocesamiento más óptima para el conjunto de datos a tratar.

El clasificador basado en árboles de decisión, HTree, logra resultados similares para el valor de AUC aplicando técnicas de preprocesamiento, en concreto para el tercer escenario, MM+SMOTE(100 %), comparando este con el conjunto de datos que no han estado sometidos a un preprocesamiento, esto es, el conjunto de datos que presenta gran cantidad de valores perdidos y una distribución de las clases en instancias completamente desbalanceadas. En otros términos, la recuperación de información y el balanceo de datos son capaces de obtener rendimiento similares. Asimismo, en el coeficiente kappa de Cohen las diferencias entre el primer y tercer escenario son aún menores. En cambio, en los casos del valor de la exhaustividad (*Recall*) y la Medida F, los escenarios que obtienen valores similares son el primero y segundo. Además, cabe destacar que a pesar de que el valor de la exactitud (*Accuracy*) es superior en el primer escenario, este valor no es significativo ya que cuando se trata con un conjunto de datos desbalanceados esta métrica no es representativa. Observando este valor en los dos últimos escenarios, se concluye que, tras aplicar las técnicas de preprocesamiento descritas anteriormente, el clasificador HTree consigue clasificar correctamente un 70 % de los datos.

Tabla 5.1. Resultados de test

Clasificador	Técnica de Preprocesamiento	Exactitud (%) (*Accuracy*)	AUC	Kappa de Cohen	Precisión (*Precision*)	Exhaustividad (*Recall*)	Medida F
HTree	Original [1]	73,42	0,6850	0,1937	0,7100	0,7340	0,7190
	EM + SMOTE (150%)	71,67	0,6430	0,1681	0,6990	0,7170	0,7060
	MM + SMOTE (100%)	69,32	0,6670	0,1884	0,7050	0,6930	0,6990
Naïve Bayes	Original	59,18	0,7370	0,1822	0,7270	0,5920	0,6220
	EM + Undersampling (25%)	56,33	0,6390	0,1486	0,3540	0,7150	0,5630
	MM + Undersampling (25%)	57,55	0,6620	0,1736	0,3300	0,7280	0,5750
IB*k*	Original	67,09	0,5620	0,1201	0,6800	0,6710	0,6750
	EM + SMOTE (100%)	62,23	0,5270	0,0490	0,6550	0,6220	0,6360
	MM + SMOTE (150%)	66,05	0,5780	0,1426	0,6900	0,6610	0,6730

[1] El algoritmo HTree presenta problemas cuando trabaja con conjuntos de datos que presentan gran número de valores perdidos. Para la realización de este experimento se ha aplicado la técnica de eliminación de instancias que contengan valores perdidos al atributo v21 que contenía 1% de valores ausentes (véase Figura A.1). De esta manera, se elimaron 500 instancias y se ejecutó el algoritmo obteniendo los resultados que se muestran en la tabla.

Naïve Bayes obtiene, en algunos casos, resultados ligeramente peores con el conjunto de datos que han estados sometidos al preprocesamiento. El coeficiente kappa de Cohen se asemeja bastante en el primer y tercer escenario, coincidiendo este comportamiento con el clasificador HTree. Además, observando este coeficiente, se deduce que los datos se ajustan mejor a la imputación de los valores perdidos mediante la media y la moda del atributo. Por otro lado, el valor de *Recall* obtiene una considerable mejor puntuación cuando los datos han sido sometidos al proceso de preparación que en la primera situación donde estos no han sufrido ningún tratamiento. La métrica AUC es similar en las tres situaciones, se obtiene más del 0,6 del área bajo la curva que, en términos de rendimiento, es un valor bastante bueno. Lo mismo ocurre con el valor de la Medida F donde las tres situaciones obtienen valores en torno a 0,56 y 0,62. Finalmente, como se explicó anteriormente, la métrica de *Accuracy* para el primer escenario no es representativa. Por tanto, tras haber aplicado técnicas de preprocesamiento al conjunto de datos, el clasificador Naïve Bayes obtiene un porcentaje de acierto del 57 %.

El clasificador IB*k* obtiene, en general, resultados mejores en el tercer escenario, MM+SMOTE(150 %). En particular, en el caso de la precisión el primer y tercer escenario tan solo tienen una diferencia de 0,01. El coeficiente kappa de Cohen también es superior en el tercer escenario. Asimismo, destaca que para la segunda situación, EM+SMOTE(100 %), solo se obtiene un 0,0490 en este valor. La Medida F y el valor de *Recall* se comportan como en los otros dos clasificadores, es decir, se obtiene una leve mejor puntuación cuando el conjunto de datos no han sido sometidos a un preprocesamiento. Finalmente, el clasificador IB*k* obtiene un porcentaje de *Accuracy* superior al 60 %.

Para deducir qué combinación de técnicas de preprocesamiento de datos obtiene mejor rendimiento se ha calculado una nueva métrica que consiste en la media aritmética de la puntuación del AUC, el coeficiente kappa de Cohen y la Medida F, la cual se describe en la Fórmula 5.1.

$$MC1 = \frac{AUC + Kappa\ de\ Cohen + Medida\ F}{3} \tag{5.1}$$

La Tabla 5.2 muestra los resultados de test de esta nueva medida. Cabe destacar que esta medida no representa el nivel de acierto del clasificador. Se observa que, en el caso del clasificador HTree obtiene un ligero mejor resultado con el conjunto de datos inicial y, este valor es muy similar en los otros dos escenarios. Por otro lado, el clasificador Naïve Bayes se comporta análogamente al anterior clasificador. Sin embargo, en este caso, la diferencia entre el segundo y tercer escenario es superior. El clasificador IB*k* obtiene la mejor puntuación en el tercer escenario y, la diferencia entre el primero y segundo es mínima. Tras observar los resultados, se concluye que el conjunto de datos se adapta mejor a la imputación de valores perdidos mediante la media y la moda del atributo ya que en los tres clasificadores, la puntuación obtenida en este escenario es superior a la obtenida en el segundo, es decir, cuando los datos se someten a la imputación de valores perdidos utilizando el Algoritmo de la Esperanza Matemática. La Figura 5.1 muestra de forma gráfica los resultados obtenidos de esta nueva medida MC1.

Tabla 5.2. Resultados obtenidos con la medida MC1

Clasificador	Técnica de Preprocesamiento	MC1
HTree	Original	0,5326
	EM + SMOTE (150 %)	0,5057
	MM + SMOTE (100 %)	0,5181
Naïve Bayes	Original	0,5137
	EM + Undersampling (25 %)	0,3938
	MM + Undersampling (25 %)	0,4702
IB*k*	Original	0,3976
	EM + SMOTE (100 %)	0,4040
	MM + SMOTE (150 %)	0,4645

La Figura 5.1 muestra de forma gráfica los resultados obtenidos de esta nueva medida MC1.

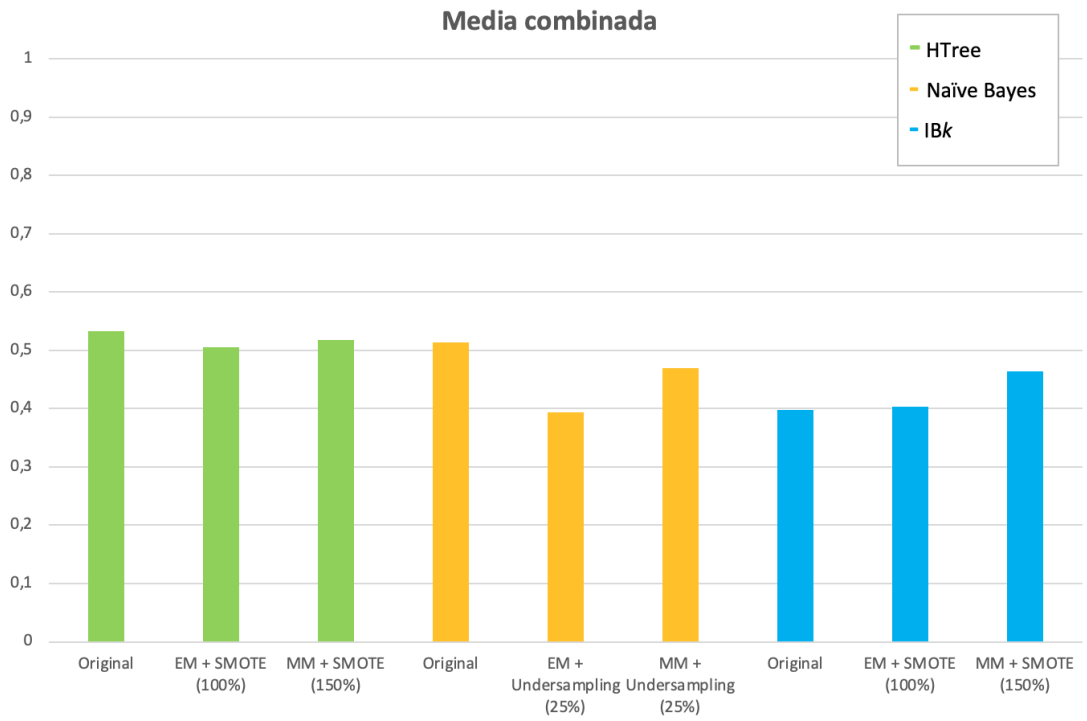

Figura 5.1. Histograma de la media combinada MC1

Fuente: Elaboración Propia

Una nueva métrica se ha añadido para el estudio de los resultados obtenidos. Esta medida es la media entre el porcentaje de exactitud (*Accuracy*) y el coeficiente de AUC. Esta nueva medida se denominará $MC2$ y se describe en la Fórmula 5.2. Esta métrica pretende comparar el rendimiento de las técnicas de preprocesamiento en el conjunto de datos a tratar, sin tener en cuenta los resultados que se obtuvieron con el conjunto de datos originales.

$$MC2 = \frac{\frac{Accuracy}{100} + AUC}{2} \tag{5.2}$$

Los resultados de esta medida obtenidos se encuentran en la Tabla 5.3. En promedio, los resultados obtenidos según cada clasificador son bastante similares. En el caso del clasificador basado en árboles de decisión, la media está en torno a un 0,67, siendo este el clasificador que mejores resultados obtiene. Por otro lado, los clasificadores de Naïve

Bayes e IB*k* obtienen resultados homogéneos, es decir, sus resultados se encuentran entre 0,57 y 0,62.

Tabla 5.3. Resultados obtenidos con la medida MC2

Clasificador	Técnica de Preprocesamiento	MC2
HTree	EM + SMOTE (150 %)	0,6799
	MM + SMOTE (100 %)	0,6801
Naïve Bayes	EM + Undersampling (25 %)	0,6012
	MM + Undersampling (25 %)	0,6188
IB*k*	EM + SMOTE (100 %)	0,5747
	MM + SMOTE (150 %)	0,6193

La Figura 5.2 muestra un histograma de los resultados obtenidos con la nueva medida MC2.

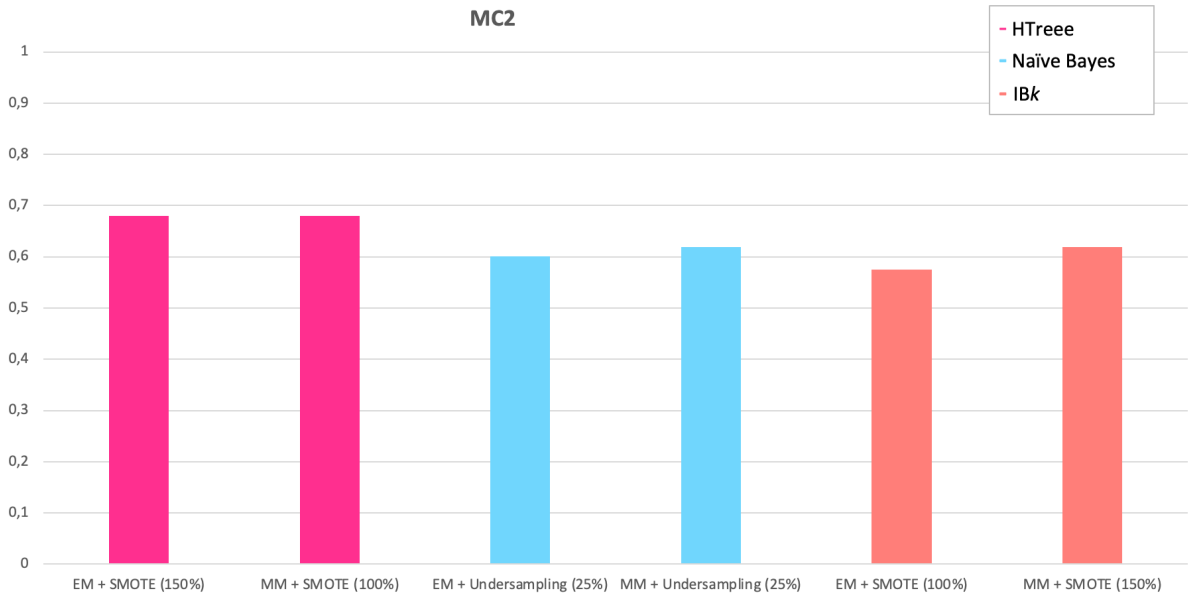

Figura 5.2. Histograma de la medida MC2 según el clasificador.

Fuente: Elaboración propia

6

Conclusiones

El primer objetivo marcado del TFM, correspondiente al estudio y revisión bibliográfica del estado del arte del Aprendizaje Automático en el sector de los seguros se ha sido abordado.

Respecto al segundo objetivo, se han propuesto diferentes técnicas de preprocesamiento de datos para resolver los dos principales problemas que presentaba la base de datos estudiada. El primero de estos es la presencia de gran cantidad de valores perdidos, lo que provocaba que no todos los algoritmos fuesen aplicables. El segundo problema es el considerable desbalanceo de las clases. Esta cuestión provocaba que los resultados obtenidos al aplicar clasificadores no fuesen precisos ya que estos se comportaban de manera que tendían a clasificar instancias que pertenecían a la clase minorita como si perteneciesen a la clase mayoritaria. Por tanto, tras realizar el estudio se ha conseguido plantear qué conjunto de técnicas de preprocesamiento se adaptan mejor a la base de datos tratada.

Una vez aplicada la técnica de imputación de valores perdidos, la técnica de sobremuestreo SMOTE obtuvo mejores resultados en comparación con la de submuestreo. La mejor adecuación del método de imputación de valores perdidos así como de la técnica de desbalanceo de datos se ha evaluado empíricamente utilizando el conjunto de entrenamiento exclusivamente haciendo uso de la técnica de *grid search*.

Las compañías de seguros manejan volúmenes de información que cada día aumentan considerablemente. Las Técnicas de Aprendizaje Automático permiten a estas empresas

descubrir patrones ocultos en sus conjuntos de datos y, además, crear sistemas que incorporan automáticamente estos nuevos patrones descubiertos, para así prevenir los riesgos. El sector asegurador es un ejemplo perfecto para la denominada Transformación Digital.

En cuanto a futuros trabajos de investigación abiertos a raíz de este TFM, se figuran, entre otros, la aplicación de técnicas de selección de datos, como la selección de instancias o atributos. Este aspecto se podría abordar estudiando la correlación entre las variables o bien alguna métrica que indique la relación entre estas, así como la consistencia. De esta manera, como determina la selección de atributos, se estudiaría qué subconjunto de estos obtiene mejores resultados de rendimiento o la calidad de los mismos individualmente.

Bibliografía

[1] Abd Razak, N., Zubairi, Y. Z., and Yunus, R. M. Imputing missing values in modelling the pm10 concentrations. *Sains Malaysiana 43*, 10 (2014), 1599–1607.

[2] Aha, D. W. *Lazy learning*. Springer Science & Business Media, 2013.

[3] Aha, D. W., Kibler, D., and Albert, M. K. Instance-based learning algorithms. *Machine learning 6*, 1 (1991), 37–66.

[4] Alghushairy, O., Alsini, R., Soule, T., and Ma, X. A review of local outlier factor algorithms for outlier detection in big data streams. *Big Data and Cognitive Computing 5*, 1 (2020), 1.

[5] Alonso Delgado, A. *Asistente para la sustitución de valores ausentes en bases de datos*. PhD thesis, Universidad Central"Marta Abreu"de la Villas, 2012.

[6] Alpaydin, E. *Machine learning*. MIT Press, 2021.

[7] Apte, C., Grossman, E., Pednault, E. P., Rosen, B. K., Tipu, F. A., and White, B. Probabilistic estimation-based data mining for discovering insurance risks. *IEEE Intelligent Systems and their Applications 14*, 6 (1999), 49–58.

[8] Ayele, W. Y. A toolbox for idea generation and evaluation: Machine learning, data-driven, and contest-driven approaches to support idea generation. *arXiv preprint arXiv:2205.09840* (2022).

[9] Baesens, B. *Analytics in a big data world: The essential guide to data science and its applications*. John Wiley & Sons, 2014.

[10] Batini, C., and Scannapieco, M. Introduction to information quality. In *Data and Information Quality*. Springer, 2016, pp. 1–19.

[11] Biau, G., and Devroye, L. *Lectures on the nearest neighbor method*. Springer, 2015.

[12] Blanco, J. A., and Tallón-Ballesteros, A. J. Supervised machine learning techniques in the bitcoin transactions. a case of ransomware classification. In *International Workshop on Soft Computing Models in Industrial and Environmental Applications* (2021), Springer, pp. 803–810.

[13] Booker, L. B., Goldberg, D. E., and Holland, J. H. Classifier systems and genetic algorithms. *Artificial intelligence 40*, 1-3 (1989), 235–282.

[14] Bradley, P. S., and Mangasarian, O. L. Feature selection via concave minimization and support vector machines. In *ICML* (1998), vol. 98, pp. 82–90.

[15] Braschler, M., Stadelmann, T., and Stockinger, K. *Applied Data Science*. Springer, 2019.

[16] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. *Classification and regression trees*. Routledge, 2017.

[17] Buntine, W., and Niblett, T. A further comparison of splitting rules for decision-tree induction. *Machine Learning 8*, 1 (1992), 75–85.

[18] Burges, C. J. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery 2*, 2 (1998), 121–167.

[19] Byanjankar, A., Heikkilä, M., and Mezei, J. Predicting credit risk in peer-to-peer lending: A neural network approach. In *2015 IEEE symposium series on computational intelligence* (2015), IEEE, pp. 719–725.

[20] Cano, J. R., Herrera, F., and Lozano, M. On the combination of evolutionary algorithms and stratified strategies for training set selection in data mining. *Applied Soft Computing 6*, 3 (2006), 323–332.

[21] Caruana, R., and Niculescu-Mizil, A. Data mining in metric space: an empirical analysis of supervised learning performance criteria. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (2004), pp. 69–78.

[22] Chai, C., Wang, J., Luo, Y., Niu, Z., and Li, G. Data management for machine learning: A survey. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[23] Chang, C.-L. Finding prototypes for nearest neighbor classifiers. *IEEE Transactions on Computers 100*, 11 (1974), 1179–1184.

[24] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research 16* (2002), 321–357.

[25] Chen, J., Nair, V., Krishna, R., and Menzies, T. "sampling" as a baseline optimizer for search-based software engineering. *IEEE Transactions on Software Engineering 45*, 6 (2018), 597–614.

[26] Chen, W., Xiang, G., Liu, Y., and Wang, K. Credit risk evaluation by hybrid data mining technique. *Systems Engineering Procedia 3* (2012), 194–200.

[27] Chen, Y.-S., Lin, C.-K., Lin, Y.-S., Chen, S.-F., and Tsao, H.-H. Identification of potential valid clients for a sustainable insurance policy using an advanced mixed classification model. *Sustainability 14*, 7 (2022).

[28] Cho, V., and Ngai, E. W. Data mining for selection of insurance sales agents. *Expert systems 20*, 3 (2003), 123–132.

[29] Chopde, K., Gosar, P., Kapadia, P., Maheshwari, N., and Chawan, P. M. A study of classification based credit risk analysis algorithm. *International Journal of Engineering and Advanced Technology 1*, 4 (2012), 142–144.

[30] Cohen, J. A coefficient of agreement for nominal scales. *Educational and psychological measurement 20*, 1 (1960), 37–46.

[31] Cortizo, J. C., and Giraldez, I. Multi criteria wrapper improvements to naive bayes learning. In *International Conference on Intelligent Data Engineering and Automated Learning* (2006), Springer, pp. 419–427.

[32] Czarnowski, I., and Jędrzejowicz, P. An approach to instance reduction in supervised learning. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence* (2003), Springer, pp. 267–280.

[33] Damrongsakmethee, T., and Neagoe, V.-E. Data mining and machine learning for financial analysis. *Indian Journal of Science and Technology 10*, 39 (2017), 1–7.

[34] Diamantini, C., and Potena, D. Bayes vector quantizer for class-imbalance problem. *IEEE Trans. Knowl. Data Eng. 21* (05 2009), 638–651.

[35] Fang, K., Wu, J., Zhu, J., and Xie, B. Forecasting of credit card credit risk under asymmetric information based on nonparametric random forests. *Economic Research Journal 39* (2010), 97–107.

[36] Farhangfar, A., Kurgan, L. A., and Pedrycz, W. Experimental analysis of methods for imputation of missing values in databases. In *Intelligent Computing: Theory and Applications II* (2004), vol. 5421, SPIE, pp. 172–182.

[37] Feng, N., Wang, H. J., and Li, M. A security risk analysis model for information systems: Causal relationships of risk factors and vulnerability propagation analysis. *Information sciences 256* (2014), 57–73.

[38] Flach, P. A., Hernández-Orallo, J., and Ramirez, C. F. A coherent interpretation of auc as a measure of aggregated classification performance. In *ICML* (2011).

[39] Frawley, W. J., Piatetsky-Shapiro, G., and Matheus, C. J. *Knowledge discovery in databases*, vol. 37. AAAI Press Menlo Park, CA, 1991.

[40] Gan, P.-T. The optimal economic uncertainty index: A grid search application. *Computational Economics 43*, 2 (2014), 159–182.

[41] García, S., Luengo, J., and Herrera, F. Dealing with missing values. In *Data preprocessing in data mining*. Springer, 2015, pp. 59–105.

[42] González López, M. *Pre-procesamiento de datos para aprendizaje de Distribución de Etiquetas*. PhD thesis, Universidad de Granada, 2021.

[43] Graham, J. W., et al. Missing data analysis: Making it work in the real world. *Annual review of psychology 60*, 1 (2009), 549–576.

[44] Guo, G., Wang, H., Bell, D., Bi, Y., and Greer, K. Knn model-based approach in classification. In *OTM Confederated International Conferences On the Move to Meaningful Internet Systems"* (2003), Springer, pp. 986–996.

[45] Gupta, S., and Gupta, A. Dealing with noise problem in machine learning datasets: A systematic review. *Procedia Computer Science 161* (2019), 466–474.

[46] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. The weka data mining software: an update. *ACM SIGKDD explorations newsletter 11*, 1 (2009), 10–18.

[47] Hall, M. A. *Correlation-based feature selection for machine learning*. PhD thesis, University of Waikato, Department of Computer Science, 1999.

[48] Han, J., and Kamber, M. Data mining: concepts and techniques, 2nd. *University of Illinois at Urbana Champaign: Morgan Kaufmann* (2006).

[49] Hart, P. The condensed nearest neighbor rule (corresp.). *IEEE transactions on information theory 14*, 3 (1968), 515–516.

[50] Hawkins, D. M. *Identification of outliers*. Springer, 1980.

[51] He, H., and Garcia, E. A. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering 21*, 9 (2009), 1263–1284.

[52] Hernández, C., Rodríguez, J. E. R., et al. Preprocesamiento de datos estructurados. *Revista vínculos 4*, 2 (2008), 27–48.

[53] Holte, R. C. Very simple classification rules perform well on most commonly used datasets. *Machine learning 11*, 1 (1993), 63–90.

[54] Hossin, M., and Sulaiman, M. N. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process 5*, 2 (2015), 1–11.

[55] Hu, M., Salvucci, S., and Cohen, M. Evaluation of some popular imputation algorithms. In *The survey research methods section of the ASA* (1998), pp. 308–313.

[56] Huang, J., and Ling, C. X. Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on knowledge and Data Engineering 17*, 3 (2005), 299–310.

[57] Huber, S., Wiemer, H., Schneider, D., and Ihlenfeldt, S. Dmme: Data mining methodology for engineering applications–a holistic extension to the crisp-dm model. *Procedia Cirp 79* (2019), 403–408.

[58] Hulten, G., Spencer, L., and Domingos, P. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (2001), Association for Computing Machinery, pp. 97–106.

[59] James, G., Witten, D., Hastie, T., and Tibshirani, R. *An introduction to statistical learning*. Springer, 2013.

[60] Jin, X., Xu, A., Bie, R., and Guo, P. Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles. In *International workshop on data mining for biomedical applications* (2006), Springer, pp. 106–115.

[61] Jurek, A., and Zakrzewska, D. Improving naïve bayes models of insurance risk by unsupervised classification. In *2008 International Multiconference on Computer Science and Information Technology* (2008), IEEE, pp. 137–144.

[62] Kaur, H., Pannu, H. S., and Malhi, A. K. A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys (CSUR) 52*, 4 (2019), 1–36.

[63] Khan, A., and Ghosh, S. K. Student performance analysis and prediction in classroom learning: A review of educational data mining studies. *Education and information technologies 26*, 1 (2021), 205–240.

[64] Kira, K., and Rendell, L. A. A practical approach to feature selection. In *Machine learning proceedings 1992*. Elsevier, 1992, pp. 249–256.

[65] Knox, E. M., and Ng, R. T. Algorithms for mining distancebased outliers in large datasets. In *Proceedings of the international conference on very large data bases* (1998), Springer, pp. 392–403.

[66] Kohavi, R., and John, G. H. Wrappers for feature subset selection. *Artificial intelligence 97*, 1-2 (1997), 273–324.

[67] Kohavi, R., and Provost, F. Glossary of terms journal of machine learning. *Machine Learning 30*, 2 (1998), 127–132.

[68] Kotsiantis, S. B., Zaharakis, I. D., and Pintelas, P. E. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review 26*, 3 (2006), 159–190.

[69] Kuhn, M., Johnson, K., et al. *Applied predictive modeling*. Springer, 2013.

[70] Kwak, N., and Choi, C.-H. Input feature selection for classification problems. *IEEE Transactions on Neural Networks 13*, 1 (2002), 143–159.

[71] Lakshminarayan, K., Harp, S. A., and Samad, T. Imputation of missing data in industrial databases. *Applied intelligence 11*, 3 (1999), 259–275.

[72] Lee, M.-C., and To, C. Comparison of support vector machine and back propagation neural network in evaluating the enterprise financial distress. *International Journal of Artificial Intelligence and Applications (IJAIA) 1*, 3 (2010), 31–43.

[73] Lin, W.-C., Tsai, C.-F., Hu, Y.-H., and Jhang, J.-S. Clustering-based undersampling in class-imbalanced data. *Information Sciences 409* (2017), 17–26.

[74] Lin, W.-Y., Hu, Y.-H., and Tsai, C.-F. Machine learning in financial crisis prediction: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42*, 4 (2011), 421–436.

[75] Liu, H., and Motoda, H. Data reduction via instance selection. In *Instance selection and construction for data mining*. Springer, 2001, pp. 3–20.

[76] Luengo, J., García-Gil, D., Ramírez-Gallego, S., García, S., and Herrera, F. Big data preprocessing. *Cham: Springer* (2020).

[77] Mahesh, B. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR) 9* (2020), 381–386.

[78] Mandala, I. G. N. N., Nawangpalupi, C. B., and Praktikto, F. R. Assessing credit risk: An application of data mining in a rural bank. *Procedia Economics and Finance 4* (2012), 406–412.

[79] Martin, B. *Instance-based learning: nearest neighbour with generalisation*. PhD thesis, University of Waikato, Department of Computer Science, 1995.

[80] McDermeit, M., Funk, R., and Dennis, M. Li analysis training series data cleaning and replacement of missing values, 1999.

[81] Mingers, J. An empirical comparison of pruning methods for decision tree induction. *Machine learning 4*, 2 (1989), 227–243.

[82] Mosley, R. The use of predictive modeling in the insurance industry. *Pinnacle Actuarial Resources Inc.* (2005).

[83] Nicolas, P. R. *Scala for machine learning*. Packt Publishing Ltd, 2015.

[84] Nie, G., Zhang, L., Liu, Y., Zheng, X., and Shi, Y. Decision analysis of data mining project based on bayesian risk. *Expert Systems with Applications 36*, 3 (2009), 4589–4594.

[85] Parekh, M., and Shukla, M. Survey of streaming clustering algorithms in machine learning on big data architecture. In *Information and Communication Technology for Competitive Strategies (ICTCS 2021)*. Springer, 2023, pp. 503–514.

[86] Peng, Y., Kou, G., Sabatka, A., Matza, J., Chen, Z., Khazanchi, D., and Shi, Y. Application of classification methods to individual disability income insurance fraud detection. In *International Conference on Computational Science* (2007), Springer, pp. 852–858.

[87] Pérez López, C. Técnicas de análisis multivariante de datos. Pearson Educación.

[88] Prechelt, L., et al. Proben1: A set of neural network benchmark problems and benchmarking rules. *Technical Report 21/94* (1994).

[89] Quinlan, J. R. *C4. 5: Programming for machine learning*. 1993.

[90] Ranawana, R., and Palade, V. Optimized precision-a new measure for classifier performance evaluation. In *2006 IEEE international conference on evolutionary computation* (2006), IEEE, pp. 2254–2261.

[91] Rendell, L., and Seshu, R. Learning hard concepts through constructive induction: Framework and rationale. *Computational Intelligence 6*, 4 (1990), 247–270.

[92] Roy, S. *Nearest neighbor with generalization*. PhD thesis, University of Canterbury, Christchurch, New Zealand, 2002.

[93] Rubin, D. B. Multiple imputation after 18+ years. *Journal of the American statistical Association 91*, 434 (1996), 473–489.

[94] Sáez, J. A., and Corchado, E. Ances: A novel method to repair attribute noise in classification problems. *Pattern Recognition 121* (2022), 108198.

[95] Samerei, S. A., Aghabayk, K., Mohammadi, A., and Shiwakoti, N. Data mining approach to model bus crash severity in australia. *Journal of safety research 76* (2021), 73–82.

[96] Santana-Morales, P., Merchán, A. F., Márquez-Rodríguez, A., and Tallón-Ballesteros, A. J. Feature ranking for feature sorting and feature selection: Fr4 (fs). In *International Work-Conference on the Interplay Between Natural and Artificial Computation* (2022), Springer, pp. 545–550.

[97] Schafer, J. L., and Graham, J. W. Missing data: our view of the state of the art. *Psychological methods 7*, 2 (2002), 147.

[98] Schorn, M. A., Verhoeven, S., Ridder, L., Huber, F., Acharya, D. D., Aksenov, A. A., Aleti, G., Moghaddam, J. A., Aron, A. T., Aziz, S., et al. A community resource for paired genomic and metabolomic data mining. *Nature Chemical Biology 17*, 4 (2021), 363–368.

[99] Shahidan, N. H., Latiff, A. S. A., and Wahab, S. A. Moving towards society 5.0: A bibliometric and visualization analysis. In *International Conference on Society 5.0* (2021), Springer, pp. 93–104.

[100] Shapiro, A. F. The merging of neural networks, fuzzy logic, and genetic algorithms. *Insurance: Mathematics and Economics 31*, 1 (2002), 115–131.

[101] Shapiro, A. F. Fuzzy logic in insurance. *Insurance: Mathematics and Economics 35*, 2 (2004), 399–424.

[102] Sharma, H., and Kumar, S. A survey on decision tree algorithms of classification in data mining. *International Journal of Science and Research (IJSR) 5*, 4 (2016), 2094–2097.

[103] Shinde, P. P., and Shah, S. A review of machine learning and deep learning applications. In *2018 Fourth international conference on computing communication control and automation (ICCUBEA)* (2018), IEEE, pp. 1–6.

[104] Smith, K. A., Willis, R. J., and Brooks, M. An analysis of customer retention and insurance claim patterns using data mining: A case study. *The Journal of the Operational Research Society 51*, 5 (2000), 532–541.

[105] Song, Q., Guo, Y., and Shepperd, M. A comprehensive investigation of the role of imbalanced learning for software defect prediction. *IEEE Transactions on Software Engineering 45*, 12 (2018), 1253–1269.

[106] Souiden, I., Brahmi, Z., and Toumi, H. A survey on outlier detection in the context of stream mining: review of existing approaches and recommadations. In *International Conference on Intelligent Systems Design and Applications* (2016), Springer, pp. 372–383.

[107] Stork, D. G., Duda, R. O., Hart, P. E., and Stork, D. Pattern classification. *A Wiley-Interscience Publication* (2001).

[108] Sundarkumar, G., and Vadlamani, R. A novel hybrid undersampling method for mining unbalanced datasets in banking and insurance. *Engineering Applications of Artificial Intelligence 37* (2015), 368–377.

[109] Swets, J. A. Measuring the accuracy of diagnostic systems. *Science 240*, 4857 (1988), 1285–1293.

[110] Valdovinos Rosas, R. M., Abad Sánchez, R., Alejo Eleuterio, R., Herrera Arteaga, E., and Trueba Espinosa, A. Tratamiento del desbalance en problemas con múltiples clases con ecoc. *Computación y Sistemas 17*, 4 (2013), 583–592.

[111] Veropoulos, K., Campbell, C., Cristianini, N., et al. Controlling the sensitivity of support vector machines. In *Proceedings of the international joint conference on AI* (1999), vol. 55, Morgan Kaufmann, pp. 60–65.

[112] Vieira, S. M., Kaymak, U., and Sousa, J. M. Cohen's kappa coefficient as a performance measure for feature selection. In *International conference on fuzzy systems* (2010), IEEE, pp. 1–8.

[113] Vijayarani, S., and Muthulakshmi, M. Comparative analysis of bayes and lazy classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering 2*, 8 (2013), 3118–3124.

[114] Viveros, M. S., Nearhos, J. P., and Rothman, M. J. Applying data mining techniques to a health insurance information system. In *VLDB* (1996), vol. 96, pp. 286–294.

[115] Wang, S., and Yao, X. Using class imbalance learning for software defect prediction. *IEEE Transactions on Reliability 62*, 2 (2013), 434–443.

[116] Wiemer, H., Drowatzky, L., and Ihlenfeldt, S. Data mining methodology for engineering applications (dmme)—a holistic extension to the crisp-dm model. *Applied Sciences 9*, 12 (2019), 2407.

[117] Wilson, D. R., and Martinez, T. R. Reduction techniques for instance-based learning algorithms. *Machine learning 38*, 3 (2000), 257–286.

[118] Wilson, S. W. Mining oblique data with xcs. In *International Workshop on Learning Classifier Systems* (2000), Springer, pp. 158–174.

[119] Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., et al. Top 10 algorithms in data mining. *Knowledge and information systems 14*, 1 (2008), 1–37.

[120] Wu, X., and Liu, H. Application of big data unbalanced classification algorithm in credit risk analysis of insurance companies. *Journal of Mathematics* (2022), 1–10.

[121] Yang, Y., and Webb, G. I. On why discretization works for naive-bayes classifiers. In *Australasian Joint Conference on Artificial Intelligence* (2003), Springer, pp. 440–452.

[122] Zadeh, L. Fuzzy sets. *Information and Control 8*, 3 (1965), 338–353.

[123] Zhang, G. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 30*, 4 (2000), 451–462.

[124] Zurada, J. *Introduction to artificial neural systems*. West Publishing Co., 1992.

A

Anexo

A.1. Tabla descriptiva la base de datos

Tabla A.1. Tabla descriptiva en detalle de los atributos de la base de datos

Var.	Tipo	Media	Desviación Típica	Mín	Máx	Valores Diferentes	Valores Perdidos (%)
v1	Numérico	1,63	1,08	-1,00e-06	20	24803	44
v2	Numérico	7,46	2,96	-1,00e-06	20	37465	44
v3	Nominal	-	-	-	-	3	3
v4	Numérico	4,15	1,15	-1,00e-06	20	37449	44
v5	Numérico	8,74	2,04	-1,00e-06	20	8994	43
v6	Numérico	2,43	0,59	-1,00e-06	20	32791	44
v7	Numérico	2,48	0,58	-1,00e-06	20	34655	44
v8	Numérico	1,49	2,78	-1,00e-06	20	10320	43
v9	Numérico	9,03	1,93	-1,00e-06	20	21547	44
v10	Numérico	1,88	1,39	-1,00e-06	18,53	973	0
v11	Numérico	15,44	0,79	0	20	30915	44
v12	Numérico	6,88	0,92	1,00e-06	18,71	70791	0
v13	Numérico	3,79	1,17	-1,00e-06	20	35113	44
v14	Numérico	12,09	1,44	-1,00e-06	20	26526	0
v15	Numérico	2,08	0,73	-1,00e-06	20	28818	44
v16	Numérico	4,92	1,79	-1,00e-06	20	11473	44

Var.	Tipo	Media	Desviación Típica	Mín	Máx	Valores Diferentes	Valores Perdidos (%)
v17	Numérico	3,83	1,91	-1,00e-06	20	37471	44
v18	Numérico	0,84	0,61	0	20	7579	44
v19	Numérico	0,22	0,17	-1,00e-06	20	28257	44
v20	Numérico	17,77	1,15	1,52	20	29909	44
v21	Numérico	7,03	1,07	0,11	19,29	76230	1
v22	Nominal	-	-	-	-	18210	0
v23	Numérico	1,09	4,00	-1,00e-06	20	5307	44
v24	Nominal	-	-	-	-	5	0
v25	Numérico	1,69	2,95	0.04	20	20837	43
v26	Numérico	1,88	0,55	-1,00e-06	20	32694	44
v27	Numérico	2,74	0,83	-1,00e-06	20	35332	44
v28	Numérico	5,09	2,67	-1,00e-06	19,84	11902	44
v29	Numérico	8,2	2,67	0	20	35837	44
v30	Nominal	-	-	-	-	7	53
v31	Nominal	-	-	-	-	3	3
v32	Numérico	1,62	0,56	-1,00e-06	17,56	33265	44
v33	Numérico	2,16	0,98	-1,00e-06	20	33392	44
34	Numérico	6,41	2,03	-1,00e-06	20	102477	0
vV35	Numérico	8,12	1,34	-1,00e-06	20	34849	44
v36	Numérico	13,37	2,36	0	20	6764	43
v37	Numérico	0,741	0,54	-1,00e-06	20	12522	44
v38	Numérico	0,09	0,58	0	20	12	0
v39	Numérico	1,24	2,36	-1,00e-06	19,92	28407	44
v40	Numérico	10,47	3,17	0	20	41842	0
v41	Numérico	7,18	1,00	-1,00e-06	20	35603	44
v42	Numérico	12,92	0,99	-1,00e-06	20	35471	44
v43	Numérico	2,22	0,64	-1,00e-06	20	30126	44
v44	Numérico	10,79	2,11	-1,00e-06	19,83	37535	44
v45	Numérico	9,14	2,06	-1,00e-06	20	35355	44
v46	Numérico	1,63	2,89	0,07	20	19739	43
v47	Nominal	-	-	-	-	10	0
v48	Numérico	12,54	2,19	-1,00e-06	20	37507	44
v49	Numérico	8,02	0,90	-1,00e-06	20	35219	44
v50	Numérico	1,50	1,16	-1,00e-06	20	10459	0
v51	Numérico	7,19	2,51	0	20	36858	44
v52	Nominal	-	-	-	-	12	0
v53	Numérico	15,71	0,79	-1,00e-06	20	29938	44
v54	Numérico	1,25	2,31	0,01	20	20982	43
v55	Numérico	1,56	0,83	-1,00e-06	20	31587	44
v56	Nominal	-	-	-	-	122	6
v57	Numérico	4,08	0,67	-1,00e-06	20	35047	44
v58	Numérico	7,70	6,80	-1,00e-06	20	33392	44
v60	Numérico	1,71	0,53	-1,00e-06	20	32805	44

Var.	Tipo	Media	Desviación Típica	Mín	Máx	Valores Diferentes	Valores Perdidos (%)
v61	Numérico	14,58	2,12	-1,00e-06	18,85	37561	44
v62	Numérico	1,03	0,69	0	7	8	0
v63	Numérico	1,69	2,96	0,05	20	20746	43
v64	Numérico	6,34	2,52	-1,00e-06	20	37461	44
v65	Numérico	15,85	1,87	0,66	20	31571	44
v66	Nominal	-	-	-	-	3	0
v67	Numérico	9,29	1,12	0	20	35580	44
v68	Numérico	17,56	2,28	1,50	20	33151	44
v69	Numérico	9,45	1,90	-1,00e-06	20	11539	44
v70	Numérico	12,27	2,31	0,43	19,82	19897	43
v71	Nominal	-	-	-	-	9	0
v72	Numérico	1,43	0,92	0	12	13	0
v73	Numérico	2,43	0,79	-1,00e-06	20	30266	44
v74	Nominal	-	-	-	-	3	0
v75	Nominal	-	-	-	-	4	0
v76	Numérico	2,41	1,38	-1,00e-06	20	37436	44
v77	Numérico	7,31	1,25	-1,00e-06	15,97	35980	44
v78	Numérico	13,33	1,84	-1,00e-06	20	11399	44
v79	Nominal	-	-	-	-	18	0
v80	Numérico	2,21	1,07	-1,00e-06	20	18126	44
v81	Numérico	7,29	2,22	0	20	12065	43
v82	Numérico	6,21	3,67	-1,00e-06	20	8148	43
v83	Numérico	2,17	1,06	-1,00e-06	20	32583	44
v84	Numérico	1,61	0,94	-1,00e-06	20	29623	44
v85	Numérico	2,82	1,42	-1,00e-06	20	26219	44
v86	Numérico	1,22	0,46	-1,00e-06	17,56	32342	44
v87	Numérico	10,18	3,00	0,87	19,84	20083	43
v88	Numérico	1,92	1,04	-1,00e-06	20	34510	44
v89	Numérico	1,52	2,81	0,02	20	20909	43
v90	Numérico	0,967	0,17	0	6,31	35099	44
v91	Nominal	-	-	-	-	7	0
v92	Numérico	0,58	0,24	0	8,92	32809	44
v93	Numérico	5,48	1,65	0	20	27198	44
v94	Numérico	3,85	0,85	-1,00e-06	19,02	34828	44
v95	Numérico	0,67	0,26	-1,00e-06	9,07	33113	44
v96	Numérico	6,46	1,12	-1,00e-06	20	35712	44
v97	Numérico	7,62	1,92	-1,00e-06	20	21932	44
v98	Numérico	7,67	2,32	-1,00e-06	19,06	19676	43
v99	Numérico	1,20	0,46	-1,00e-06	20	32126	44
v100	Numérico	12,09	6,88	-1,00e-06	20	33221	44
v101	Numérico	6,87	2,35	0	20	37593	44
v102	Numérico	2,89	1,82	-1,00e-06	20	6087	45
v103	Numérico	5,29	1,22	-1,00e-06	18,78	33819	44

Var.	Tipo	Media	Desviación Típica	Mín	Máx	Valores Diferentes	Valores Perdidos (%)
v104	Numérico	2,64	0,88	-1,00e-06	20	34478	44
v105	Numérico	1,08	2,24	0	20	20492	43
v106	Numérico	11,79	2,95	-1,00e-06	20	37490	44
v107	Nominal	-	-	-	-	7	0
v108	Numérico	2,15	0,91	0	20	18811	43
v109	Numérico	4,18	3,71	-1,00e-06	20	6676	43
v110	Nominal	-	-	-	-	3	0
v111	Numérico	3,37	1,48	-1,00e-06	20	31674	44
v112	Nominal	-	-	-	-	22	0
v113	Nominal	-	-	-	-	36	48
v114	Numérico	13,57	2,61	0	20	41690	0
v115	Numérico	10,55	1,90	-1,00e-06	20	11572	44
v116	Numérico	2,29	0,67	-1,00e-06	20	28945	44
v117	Numérico	8,30	3,61	-1,00e-06	20	8571	43
v118	Numérico	8,36	2,00	0	20	22160	44
v119	Numérico	3,17	4,24	-1,00e-06	20	28273	44
v120	Numérico	1,29	0,73	-1,00e-06	10,39	30291	44
v121	Numérico	2,74	1,35	-1,00e-06	20	29428	44
v122	Numérico	6,82	1,79	-1,00e-06	20	21250	44
v123	Numérico	3,55	2,60	0,02	19.69	37539	44
v124	Numérico	0,92	2,09	-1,00e-06	20	17762	43
v125	Nominal	-	-	-	-	90	0
v126	Numérico	1,67	0,50	-1,00e-06	15,63	33139	44
v127	Numérico	3,24	1,62	-1,00e-06	20	35188	44
v128	Numérico	2,03	1,07	1,00e-06	20	17877	43
v129	Numérico	0,31	0,69	0	11	10	0
v130	Numérico	1,93	1,26	-1,00e-06	20	26313	44
v131	Numérico	1,74	1,13	-1,00e-06	20	8499	44
class	Nominal	0,76	0,42	0	1	2	0

A.2. Distribución de los valores perdidos en la base de datos

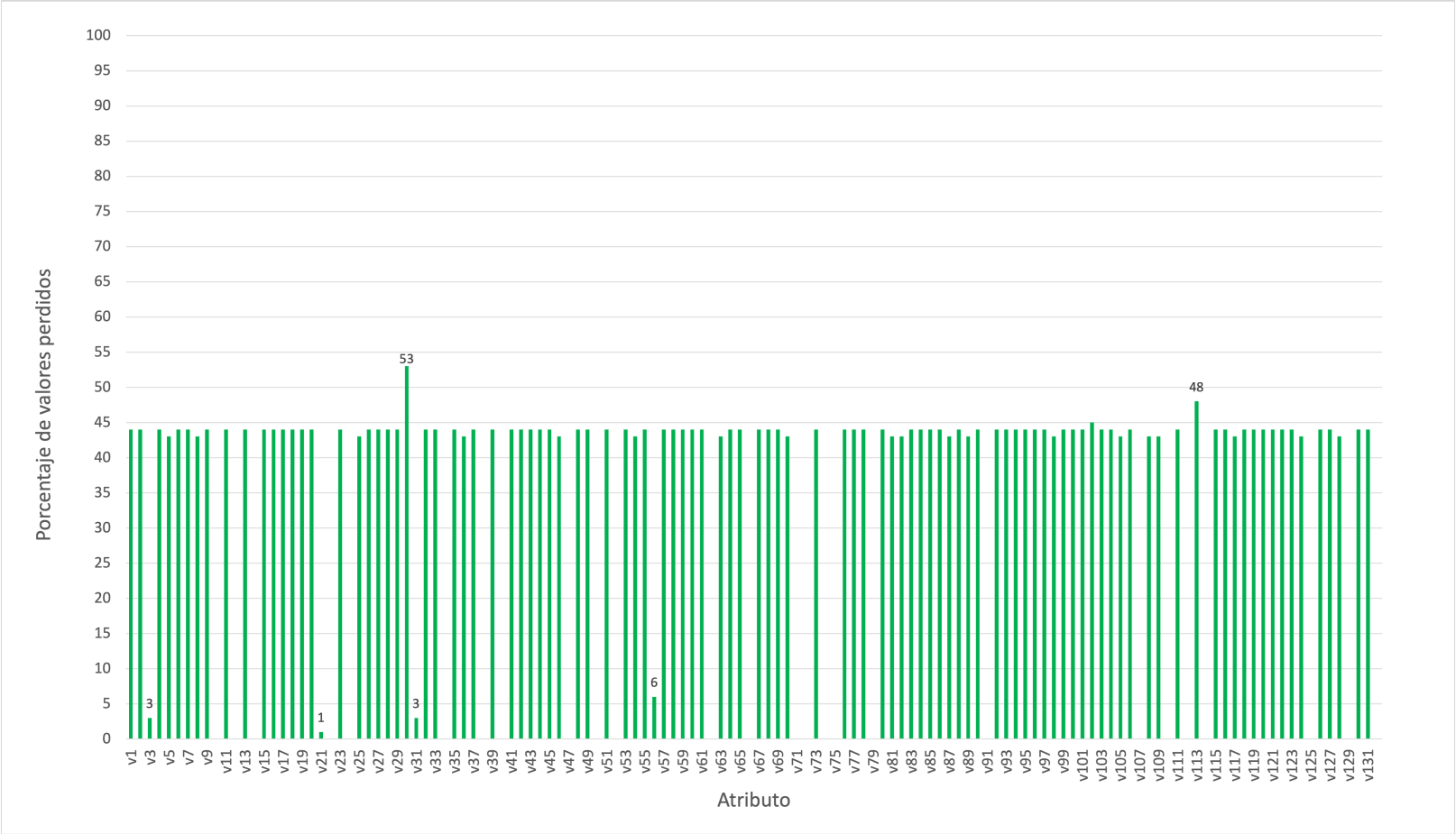

Figura A.1. Distribución de los valores perdidos en la base de datos en detalle. **Fuente:** Elaboración propia