Universidad Internacional de La Rioja (UNIR)

Escuela de Ingeniería

Máster en Análisis y Visualización de Datos Masivos

Análisis y optimización de algoritmos de clasificación supervisada sobre operaciones impagadas en tarjetas de crédito

Trabajo Fin de Máster

Presentado por: de Juan de Llano, Iozu

Director/a: Mora García, Antonio Miguel

Ciudad: Madrid
Fecha: 21-9-2017

Resumen

El presente proyecto tiene como objetivo identificar el algoritmo con mayor exactitud, para detectar patrones de impago en tarjetas de crédito partiendo a partir de un dataset con operaciones anónimas reales. Para poder identificar patrones de impago se han utilizado veinticuatro algoritmos de inteligencia artificial basados en clasificación supervisada, así como técnicas de clustering, eliminación de valores fuera de rango, normalización de variables, tratamiento de valores ausentes, selección de variables o tratamiento de datasets no balanceados, entre otras técnicas, con el fin de intentar mejorar el nivel de exactitud inicial. A partir de los resultados iniciales, en los que se evalúan los algoritmos principalmente en base a su nivel de exactitud, se han identificado los algoritmos con mejores resultados para analizar adicionalmente su precisión, sensibilidad, área ROC, tiempo de ejecución, número de variables utilizadas, su rendimiento y complejidad en cuanto al modelo generado y la interpretación de los resultados obtenidos, identificando el algoritmo que mejor clasifica las instancias, comparando, además, los resultados obtenidos con estudios previos realizados sobre el mismo dataset, así como sobre estudios similares realizados sobre otros datasets con información de tarjetas de crédito. Concretamente, en el estudio previo existente el algoritmo seleccionado es la red neuronal. En el presente trabajo se concluye que la combinación de selección de atributos junto con el algoritmo de clasificación J48 proporcionan niveles similares de exactitud, precisión y sensibilidad, si bien reducen sensiblemente el tiempo de ejecución (es 60 veces más rápido que la red neuronal) así como mejoran la facilidad de interpretación de los resultados (utiliza sólo 6 de 23 atributos distribuidos en un árbol de 25 nodos con 24 conexiones, frente a la red neuronal que utiliza 23 de 23 atributos y 37 nodos con 338 conexiones).

Palabras Clave: Impago en tarjetas de crédito, Clasificación supervisada, preprocesamiento de datos, selección de algoritmos de clasificación

Abstract

The present project has as aim to identify the algorithm with better accuracy values in the detection of default in credit card real operations from an anonymous dataset. In order to identify patterns of default, there have been tested twenty four artificial intelligence algorithms based on supervised classification, as well as technics like clustering, outliers elimination,

variable normalization, absent values treatment, feature selection, or data balancing dataset techniques. These have been applied trying to improve the level of the initial accuracy. Starting from initial results on the basis of the accuracy level, the methods with higher accuracy have been chosen to conduct an analysis on additional parameters such as precision, sensitivity, area ROC, running time, number of used variables, complexity regarding the generated model, and about the interpretation of the obtained results. The objective is to identify the algorithm that better classifies the instances, and compare it with the results obtained with the previous studies performed on the same date set, as well as compare it with similar studies carried out on other datasets with credit card information. Specifically, in the previous existing study the selected algorithm was the neuronal network. However, the combination of selection of attributes together with the J48 classification algorithm provide similar accuracy, precision and sensibility levels, as well as much better execution times (it is 60 times faster than neuronal networks) and a better improvement in results interpretation (J48 uses only 6 out of 23 attributes distributed in a tree of 25 nodes with 24 connections opposite to the neuronal network that uses 23 out of 23 attributes and 37 nodes with 338 connections).

Keywords: Credit Card default, data pre processing, classification algorithms selection

Contenido

Resumen....2
Abstract....2
1 Introducción y objetivos....8
2 Estructura del trabajo....10
3 Contexto y estado del arte....11
3.1 Datasets identificados para el análisis de algoritmos de aprendizaje supervisado....11
3.2 Estudios previos realizados sobre operaciones en tarjetas de crédito....15
4 Arquitectura software y hardware utilizada para las pruebas....18
5 Proceso de análisis....23
5.1 Estructura del dataset....23
5.2 Identificación inicial de algoritmos de predicción....24
5.3 Análisis de los valores de los atributos del dataset....30
5.4 División del dataset en entrenamiento y prueba....33
5.5 Identificación de atributos relevantes del dataset – selección de características....33
5.6 Tratamiento de los atributos: missing, outliers, normalización, balanceo de datos....39
5.6.1 Identificación de campos nulos....39
5.6.2 Identificación de valores anómalos – outliers....39
5.6.3 Normalización de variables....41
5.6.4 Generación de casos de impago sintéticos con SMOTE....42
5.7 Segmentación de clientes.....44
5.8 Identificación de algoritmos de mayor precisión....55
6 Conclusiones....61
7 Referencias y enlaces....64

Índice de tablas

Tabla 1 – Resultados obtenidos en el estudio de I-Cheng Yeh y Che-hui Lien identificando los niveles de exactitud iniciales 15
Tabla 2 – Resultados obtenidos en el estudio de I-Cheng Yehn y Che-hui Lien identificando el nivel de dispersión respecto a la recta de regresión entre los valores predichos y la predicción 15
Tabla 3 – Resultados del estudio “Comparative Evaluation of Predictive Modeling Techniques on Credit Card Data” 17
Tabla 4 – Distribución inicial la clase sobre la población del dataset inicial 24
Tabla 5 - Detalle del nivel de exactitud inicial de los algoritmos de clasificación 26
Tabla 6 - Detalle de los tiempos de ejecución inicial de los algoritmos de clasificación 27
Tabla 7 - Detalle de las variables seleccionadas por los algoritmos de clasificación 29
Tabla 8 – Distribución del nivel de balanceo de la clase sobre la población de los dataset de training y de test 33
Tabla 9 – Matriz de correlación de Pearson 34
Tabla 10 – Variables identificadas por los algoritmos de selección de variables 35
Tabla 11 – Variables identificadas por los algoritmos de selección de variables ajustada 36
Tabla 12 – Comparativa de la exactitud de los algoritmos de clasificación con y sin selección de atributos 37
Tabla 13 – Comparativa de la exactitud de los algoritmos de clasificación con y sin eliminación de outliers 40
Tabla 14 – Comparativa de la exactitud de los algoritmos de clasificación con y sin atributos normalizados 41
Tabla 15 – Comparativa de la exactitud de los algoritmos de clasificación con y sin instancias sintéticas generadas con SMOTE 43
Tabla 16 – Comparativa de la exactitud de los algoritmos de clasificación, reevaluando los modelos mejorados con SMOTE sobre el dataset de prueba 44
Tabla 17 – Muestra de la distribución de las instancias de cada tipo de cluster 46
Tabla 18 – Distribución de las instancias por algoritmo KMeans en los cluster 0, 1, 2, 3, y 4 46
Tabla 19 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 0 47
Tabla 20 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 1 47

Tabla 21 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 2 48
Tabla 22 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 3 48
Tabla 23 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 4 49
Tabla 24 – Resumen de los resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 0, 1, 2, 3, y 4 50
Tabla 25 – Resumen de los resultados por algoritmo de clasificación con segmentación de clientes 50
Tabla 26 – Resumen de los resultados de exactitud de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 0, 1, 2, 3, y 4 51
Tabla 27 – Distribución de las instancias por algoritmo KMeans en los cluster 0 y 4, divididos en dos subclusters 52
Tabla 28 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 0 – Subcluster 0 52
Tabla 29 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 0 – Subcluster 1 53
Tabla 30 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 4 – Subcluster 0 53
Tabla 31 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 4 – Subcluster 1 54
Tabla 32 – Resumen de los resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – cluster 0 y 4, divididos en dos subclusters, identificando la diferencia entre la clasificación inicial y la clasificación final aplicando subclusters 54
Tabla 33 – Tres algoritmos con mayor nivel de exactitud 55
Tabla 34 – Tres algoritmos con mayor nivel de exactitud, clasificados por nivel de precisión, sensibilidad y media armónica de ambos F no ponderada 56
Tabla 35 – Relación de algoritmos evaluados por el meta algoritmo AutoWEKAClassifier 57
Tabla 36 – Tres algoritmos con mayor nivel de exactitud, clasificados por nivel de precisión, sensibilidad y media armónica de ambos F no ponderada, tiempo de ejecución, nº de variables seleccionadas y complejidad del algoritmo 60
Tabla 37 – Comparación del algoritmo óptimo identificado en el estudio previo realizado por I-Cheng Yeh y Che-hui Lien y el obtenido en el presente trabajo 63

Índice de ilustraciones

Ilustración 1 – Flujo desarrollado para la ejecución de algoritmos 19
Ilustración 2 - Nivel de exactitud inicial de los algoritmos de clasificación 25
Ilustración 3 – Tiempo de ejecución de algoritmos de clasificación 28
Ilustración 4 – Representación del número de veces que ha sido seleccionada cada variable 30
Ilustración 5 – Distribución de los campos del dataset 30
Ilustración 6 – Identificación de los atributos más relevantes 36
Ilustración 7 – Identificación de los atributos más relevantes mejorada 36
Ilustración 8 – Árbol de clasificación generado por el algoritmo Attribute Selected Classifier . 59
Ilustración 9 – Árbol de clasificación generado por el algoritmo LMT 60

1 Introducción y objetivos

Las empresas se enfrentan al reto de predecir el comportamiento de las personas en relación a los modelos económicos, de marketing o sanitarios, el funcionamiento de los procesos industriales o la maquinaria, o del medio ambiente, entre otros, en muy diferentes áreas: prevención del impago en tarjetas de crédito, prevención del delito en transacciones por internet, previsión del tiempo, prevención del fallo en maquinaria industrial o la predicción de enfermedades o de gasto sanitario. Es por ello que en las empresas se han desarrollado muy diversos sistemas de monitorización cuya finalidad es conocer el funcionamiento actual, así como predecir el comportamiento futuro del objeto analizado para realizar acciones de carácter previo al hecho que se intenta predecir. Para ello, la inteligencia artificial ha desarrollado diferentes algoritmos que permiten, en base a conocimiento previo, predecir el comportamiento futuro. Los algoritmos de clasificación supervisada permiten establecer métodos de búsqueda para instancias que no han sido clasificadas previamente con la finalidad de predecir su clase.

No obstante, la información de que se dispone no siempre contiene suficientes muestras (conocimiento previo) del hecho buscado, a fin de poder establecer algoritmos de predicción suficientemente robustos como para predecir correctamente dicho hecho. Asimismo dicha información puede presentar problemas, como por ejemplo: existir valores en los atributos fuera de los rangos habituales, existir patrones de los que no se disponga de todos los atributos completos o contener variables continuas/discretas/nominales, que pueden alterar o limitar significativamente el comportamiento de los algoritmos de predicción.

El presente trabajo tiene como finalidad analizar diferentes algoritmos de clasificación supervisada para determinar cuál de ellos es más efectivo a la hora de generar un modelo para identificar operaciones de fraude a partir del dataset inicial. La identificación de modelos de predicción efectivos puede servir como punto de partida para entidades centradas en esta área de conocimiento, de cara a iniciar el proceso de búsqueda de modelos de predicción directamente con los algoritmos identificados en el presente trabajo. La efectividad se medirá en función de diferentes variables: exactitud, precisión, sensibilidad, área ROC, tiempo de ejecución, número de variables utilizadas, complejidad del modelo generado y complejidad en el proceso de interpretación del modelo. Para ello, se ha seleccionado un dataset de libre distribución (Default of credit card clients Data Set, 2016) que contiene datos sobre operaciones de tarjetas de crédito de Taiwán disponible en el Center for Machine Learning and Intelligent Systems, con 23 atributos y 30.000 instancias, las cuales han sido previamente

etiquetadas como pagadas o impagadas. Por tanto, el resultado de este trabajo es determinar qué algoritmo de clasificación predice mejor el impago en tarjetas de crédito para nuevas instancias no clasificadas.

El proceso de selección de algoritmos se basa en diferentes factores. Inicialmente se ejecutan todos los algoritmos de clasificación incluidos en el alcance sobre el dataset original sobre el que no se han realizado modificaciones, obteniendo sus resultados: matriz de confusión con falsos positivos y falsos negativos, de la que se obtiene la exactitud (ver Ecuación 1 – Exactitud), que mide la relación entre los casos verdaderos identificados por el algoritmo y la población total. No obstante, por comodidad en el presente trabajo se ha utilizado la tasa de error en la exactitud (ver Ecuación 2 – Error en la exactitud) con la finalidad de identificar aquellos algoritmos cuya tasa de error en la exactitud sea la más baja.

Por otra parte, la ejecución de los algoritmos facilita la identificación preliminar de las variables más relevantes seleccionadas por cada uno de ellos, de cara a poder analizar con posteridad los resultados obtenidos en procesos de clasificación basados únicamente en dichas variables.

Posteriormente se realizan operaciones sobre el dataset original modificándolo para tratar las instancias de cara a eliminar outliers, normalizar variables, tratar los valores ausentes en los atributos, ejecutar algoritmos específicos para la selección de características, aplicar métodos para el tratamiento de datasets no balanceados o de segmentación.

Finalmente se ejecutan de nuevo los algoritmos de clasificación analizando los resultados para determinar el impacto de los procesos de tratamiento del dataset sobre cada uno de ellos. Los algoritmos se ordenan según la exactitud alcanzada por el modelo de predicción de clases generado. A continuación, sobre los tres algoritmos más relevantes en función del nivel de exactitud alcanzado, es decir, por su baja tasa de error, se analizan variables adicionales como son la precisión, sensibilidad, área ROC, tiempo de ejecución, número de variables utilizadas, su complejidad en cuanto al modelo generado y la interpretación de los resultados obtenidos. De esta manera se identifica el algoritmo que mejor clasifica las instancias a todos los niveles, comparando después los resultados obtenidos por éste con los estudios previos realizados sobre el mismo dataset, así como sobre otros datasets diferentes que contienen información sobre tarjetas de crédito con impagos o concesión de crédito.

Como resultado se obtiene el algoritmo que mejor predice la clase asociada a los impagos de tarjetas de crédito, minimizando lo máximo posible los errores de falsos positivos y falsos

negativos, es decir, la tasa de error en la exactitud, lo que implica que tendrán mejores predicciones que el resto de algoritmos.

2 Estructura del trabajo

En este apartado se describe brevemente la estructura de los capítulos del presente trabajo:

- *Contexto y estado del arte*. En este apartado se describe el trabajo realizado para seleccionar el dataset objeto de análisis, así como la existencia de estudios previos sobre el citado dataset y las conclusiones obtenidas por dichos trabajos.
- *Arquitectura software y hardware utilizada para la pruebas*. En este apartado se describe la arquitectura hardware y software utilizada para la realización del presente trabajo.
- *Estructura del dataset*. En este apartado se describe la estructura del dataset utilizado para el análisis del impago en tarjetas de crédito.
- *Identificación inicial de algoritmos de predicción*. En este apartado se identifican de forma preliminar, sin realizar tratamientos sobre del dataset inicial, el nivel de exactitud sobre el dataset identificado, de los algoritmos explicados previamente en la arquitectura. Para ello se mide su nivel de exactitud, el tiempo de ejecución, así como el proceso de selección de variables que realizan.
- *Análisis de los valores de los atributos del dataset*. En este apartado se analizan los valores de los atributos del dataset, identificando valores ausentes, anómalos o fuera de rango, máximos, mínimos, medias y desviaciones estándar.
- *División del dataset en entrenamiento y prueba*. En este apartado se explica cómo se divide del dataset inicial en dos datasets independientes de entrenamiento y prueba.
- *Identificación de atributos relevantes del dataset – selección de características*. En este apartado se analiza la correlación entre los diferentes atributos y se ejecutan diferentes algoritmos que permiten la identificación de los atributos más relevantes, con el fin de valorar la posible ejecución de los algoritmos de clasificación sobre un dataset reducido con similares resultados de exactitud.
- *Tratamiento de los atributos: missing, outliers, normalización y balanceo de datos*. En este apartado se aplican técnicas para intentar mejorar la calidad de la información del dataset con la finalidad de que los algoritmos de clasificación evaluados inicialmente puedan obtener un nivel de exactitud superior.

- *Segmentación de clientes*. En este apartado se realiza una segmentación de clientes, con la finalidad de establecer grupos homogéneos que permitan generar modelos específicos para cada segmento de forma que éstos sean más precisos, al centrarse en poblaciones más específicas.
- *Identificación de algoritmos de mayor precisión*. En el presente apartado se analizan los algoritmos que mejor clasifican la información del dataset. Para ello, y teniendo en cuenta todos los tratamientos realizados en los apartados previos, se analizan los tres algoritmos en función de su nivel de exactitud, precisión, sensibilidad, área ROC, tiempo de ejecución, variables utilizadas, complejidad del modelo generado (Ej.: número de nodos y hojas en el caso de árboles), así como complejidad del modelo a la hora de interpretar los resultados y ejecutar el modelo obtenido con nuevas instancias.

3 Contexto y estado del arte

En el presente apartado se describen los datasets identificados inicialmente para la realización del presente trabajo relativos, en su mayoría, al fraude en tarjetas de crédito, así como las características principales de cada dataset que facilitaron la selección de uno de ellos y el descarte de los demás. Por otra parte, se describe el estudio previo existente sobre el dataset seleccionado y las conclusiones obtenidas en él, así como sobre otros datasets diferentes que contienen información sobre tarjetas de crédito con impagos o concesión de crédito.

3.1 Datasets identificados para el análisis de algoritmos de aprendizaje supervisado

Para la selección del dataset inicial, se han consultado múltiples fuentes de información. A continuación se muestran algunas de ellas:

- *Dataset: KDD Cup 1999*. (The UCI KDD Archive, 1999)
    - Descripción: Dataset en el que se registra información sobre conexiones de internet.
    - Dominio de conocimiento: Información de los protocolos utilizados.
    - Tipo de conocimiento: Supervisado. Posee la clase.

  - Número de instancias: Contiene nueve datasets y un muy elevado número de atributos, sobre los que es necesario componer un modelo de datos que los relacione, para poder determinar el número de instancias y poder trabajar con ellos.
  - Este fichero no fue seleccionado, debido a su complejidad inicial no fue seleccionado.
- *Dataset: Statlog (German Credit Data) Data Set.* (Hofmann, 1994)
  - Descripción: Dataset en el que se registra información sobre operaciones con tarjetas de crédito.
  - Dominio de conocimiento: Información de los datos bancarios utilizados.
  - Tipo de conocimiento: Supervisado. Posee la clase.
  - Número de instancias: Contiene únicamente 1000 instancias, lo cual no proporciona un margen suficiente para que los algoritmos puedan dividirse en datasets de entrenamiento y de prueba con suficientes clases de entrenamiento y validación.
  - Este fichero no fue seleccionado, debido al reducido número de instancias no fue seleccionado.
- *Dataset: Credit Approval Data Set.* (Credit Approval Data Set, s. f.)
  - Descripción: Dataset en el que se registra información sobre operaciones con tarjetas de crédito.
  - Dominio de conocimiento: Información de los datos bancarios utilizados.
  - Tipo de conocimiento: Supervisado. Posee la clase.
  - Número de instancias: Contiene únicamente 690 instancias, lo cual no proporciona un margen suficiente para que los algoritmos puedan dividirse en datasets de entrenamiento y de prueba con suficientes clases de entrenamiento y validación.
  - Este fichero no fue seleccionado, debido al reducido número de instancias no fue seleccionado.
- *Dataset: Japanese Credit Screening Data Set.* (Sano, 1992)
  - Descripción: Dataset en el que se registra información sobre operaciones con tarjetas de crédito.
  - Dominio de conocimiento: Información de los datos bancarios utilizados.
  - Tipo de conocimiento: Supervisado. Posee la clase.
  - Número de instancias: Contiene únicamente 125 instancias y no dispone de descripción de los atributos, lo cual no proporciona un margen suficiente para

que los algoritmos puedan dividirse en datasets de entrenamiento y de prueba con suficientes clases de entrenamiento y validación.

  - Este fichero no fue seleccionado, debido al reducido número de instancias no fue seleccionado.

- *Dataset: Bank Marketing Data Set.* (Moro, Cortez and Rita, 2014)
  - Descripción: Dataset en el que se registra información sobre campañas de marketing.
  - Dominio de conocimiento: Información de las acciones comerciales utilizadas así como datos bancarios.
  - Tipo de conocimiento: Supervisado. Posee la clase.
  - Número de instancias: Contiene 45.211 instancias divididas en cuatro datasets.
  - Este fichero no fue seleccionado. Este fichero se valoró para su selección ya que posee un número elevado de instancias, así como un dominio de conocimiento adecuado, si bien, fue descartado por seleccionar posteriormente otro dataset.
- *Dataset: Spambase Data Set.* (Hopkins, Reeber, Forman and Suermindt, 1999)
  - Descripción: Dataset en el que se registra información sobre correos electrónicos.
  - Dominio de conocimiento: Información sobre las características de los campos habituales de correo electrónico.
  - Tipo de conocimiento: Supervisado. Posee la clase.
  - Número de instancias: Contiene únicamente 4.601 instancias, lo cual no proporciona un margen suficiente para que los algoritmos puedan dividirse en datasets de entrenamiento y de prueba con suficientes clases de entrenamiento y validación.
  - Este fichero no fue seleccionado. Este fichero se valoró para su selección ya que posee un número elevado de instancias, así como un dominio de conocimiento adecuado, si bien, fue descartado por seleccionar posteriormente otro dataset.
- *Dataset: FICO UCSD Data Mining Contest.* (FICO UCSD Data Mining Contest, 2009)
  - Descripción: Dataset con información sobre fraude en tarjetas de crédito.
  - Dominio de conocimiento: Información de los datos bancarios utilizados.
  - Tipo de conocimiento: Supervisado. Posee la clase.
  - Número de instancias: Contiene únicamente 100.000 instancias, si bien están divididas en seis data sets, lo cual hace más complejo determinar su unión

para tratarlos de forma única. Asimismo está claramente desbalanceado al tener sólo un 3% de operaciones con fraude.

  - Este fichero no fue seleccionado. Este fichero se valoró para su selección ya que posee un número elevado de instancias, así como un dominio de conocimiento adecuado, si bien, fue descartado por estar muy desbalanceado así como por la necesidad de unir todos los datasets.

- *Dataset: Statlog (Australian Credit Approval) Data Set.* (Statlog (Australian Credit Approval) Data Set, s. f.)
  - Descripción: Dataset con información sobre concesión de crédito en tarjetas de crédito.
  - Dominio de conocimiento: Información de los datos bancarios utilizados.
  - Tipo de conocimiento: Supervisado. Posee la clase.
  - Número de instancias: Contiene únicamente 690 instancias, lo cual no proporciona un margen suficiente para que los algoritmos puedan dividirse en datasets de entrenamiento y de prueba con suficientes clases de entrenamiento y validación.
  - Este fichero no fue seleccionado, debido al reducido número de instancias no fue seleccionado.
- *Dataset: Default of credit card clients Data Set*. (Yeh, 2016)
  - Descripción: Dataset en el que se registra información sobre operaciones con tarjetas de crédito.
  - Dominio de conocimiento: Información de los datos bancarios utilizados.
  - Tipo de conocimiento: Supervisado. Posee la clase.
  - Número de instancias: Contiene 30.000 instancias y 23 atributos, lo cual permite que los algoritmos puedan ejecutarse con datasets de entrenamiento y de prueba.
  - Fichero finalmente sí fue seleccionado. Este es el fichero que se ha seleccionado para el presente proyecto, ya que posee un número elevado de instancias, así como un dominio de conocimiento adecuado. Por otra parte, el presente dataset ha sido publicado en el año 2016, siendo el más completo y actualizado de los datasets descritos.

3.2 Estudios previos realizados sobre operaciones en tarjetas de crédito

Sobre el dataset analizado del presente trabajo publicado por el Center for Machine Learning and Intelligent Systems (Default of credit card clients Data Set, 2016), se han identificado los siguientes estudios:

- "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients" (Yeh and Lien, 2009).

  En dicho estudio los autores evaluaron los siguientes algoritmos de clasificación sobre el dataset obteniendo los siguientes resultados (ver Tabla 1):

Tabla 1 – Resultados obtenidos en el estudio de I-Cheng Yeh y Che-hui Lien identificando los niveles de exactitud iniciales

Algoritmo	Exactitud entrenamiento	Exactitud prueba
K-nearest neighbor	0,18	0,16
Logistic regression	0,20	0,18
Discriminant analysis	0,29	0,26
Naıve Bayesian	0,21	0,21
Neural networks	0,19	0,17
Classification trees	0,18	0,17

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Posteriormente realizó un análisis adicional sobre los datos utilizando una técnica nueva, llamada Sorting Smoothing Method (SSM), que determinó que el nivel de exactitud es superior en las redes neuronales respecto al resto de métodos, ya que casi no existía dispersión (desviación típica sobre la media) entre los resultados obtenidos en la predicción y los predichos, identificando asimismo altos niveles de dispersión en los resultados de los árboles de clasificación pese a su alto nivel de exactitud inicial (ver Tabla 2).

Tabla 2 – Resultados obtenidos en el estudio de I-Cheng Yehn y Che-hui Lien identificando el nivel de dispersión respecto a la recta de regresión entre los valores predichos y la predicción

Algoritmo	Coeficiente de regresión	Regresión R^2
K-nearest neighbor	0,770	0,876
Logistic regression	1,233	0,794
Discriminant analysis	0,837	0,659
Naıve Bayesian	0,502	0,899

Algoritmo	Coeficiente de regresión	Regresión R^2
Neural networks	0,998	0,965
Classification trees	1,111	0,278

Este último algoritmo no será considerado para el presente proyecto, ya que la finalidad es aplicar tratamientos de datos sobre los atributos con la finalidad de intentar mejorar el nivel de exactitud obtenido en el citado estudio, mientras que el algoritmo SSM establece dos niveles de predicción diferentes. Como resultado del trabajo realizado se concluyó que el algoritmo con mejor nivel de clasificación son las redes neuronales, basándose en el bajo nivel de dispersión entre los resultados obtenidos en la predicción y los predichos. Asimismo establece que el algoritmo de K vecinos posee un elevado nivel de exactitud y bajo nivel de dispersión como segundo mejor algoritmo de clasificación.

- "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients" (Gómez and Leur, s .f.). En este estudio se crea un data mart en MySQL, y se aplica el algoritmo de clasificación Naive Bayes, aplicando previamente técnicas de preprocesamiento agrupando datos (como el BILL_AMT, y APY_AMT que se suman en dos variables), se discretizan valores numéricos en rangos, se eliminan registros para mejorar el balanceo de la clase positiva, consiguiendo una tasa de error en la exactitud (ver Ecuación 2 – Error en la exactitud) del 29%, el cual es un resultado que claramente es mejorable, razón por la que no se tendrá en cuenta en el presente trabajo.

Adicionalmente se han identificado los siguientes estudios sobre fraude en tarjetas de crédito, si bien se han realizado sobre datasets diferentes al utilizado en el presente trabajo:

- "Comparative Evaluation of Predictive Modeling Techniques on Credit Card Data" (Singh and Aggarwal, 2011). En este estudio se comparan técnicas estadísticas (paramétricas, pues se debe conocer la distribución de la población) y técnicas de inteligencia artificial (técnicas no paramétricas pues no se conoce la distribución de la

población). Los resultados obtenidos se muestran en la tabla 3:

Tabla 3 – Resultados del estudio "Comparative Evaluation of Predictive Modeling Techniques on Credit Card Data"

	Exactitud	Valores positivos predichos	Valores negativos predichos	Tasa de verdaderos positivos	Tasa de verdaderos negativos
Linear Discriminant Analysis	15.96	84.78	95.11	94.44	86.42
Logistic Regression	13.7	82.82	89.72	*	*
Multilayer perceptron	12.91	85.11	88.68	85.95	87.99
Decision tree	14.36	78.67	79.90	92.81	79.90
Genetic Programming	16.54	77.76	89.91	88.89	79.11
Support Vector Machines	13.49	82.97	89.23	87.58	85.64
Kernel density estimation	19.72	80.08	87.89	82.80	87.80
Kneighbourhood	18.23	79.08	85.02	84.50	82.20

* Estos parámetros no pueden ser calculados.

Exactitud = ver Ecuación 2 – Error en la exactitud, utilizada en el presente trabajo.

Valores positivos predichos = Precisión, utilizada en el presente trabajo.

Valores negativos predichos = equivalente a la precisión con valores negativos, no utilizada en el presente trabajo.

Tasa de verdaderos positivos = Sensibilidad, utilizada en el presente trabajo.

Tasa de verdaderos negativos = equivalente a la sensibilidad con valores negativos, no utilizada en el presente trabajo.

En el estudio se analiza, a partir del dataset de concesión de crédito en tarjetas de crédito "Statlog (Australian Credit Approval) Data Set" (Statlog (Australian Credit Approval) Data Set, s. f.) el nivel de exactitud de cada algoritmo, así como la influencia del número de validaciones cruzadas realizadas, entre otros aspectos, concluyendo acerca de la influencia en los diferentes algoritmos analizados a través de los ratios indicados en la tabla 3 del número de validaciones cruzadas realizadas. No obstante, y si bien no concluyen a cerca de qué algoritmo clasifica mejor, se puede observar en la tabla que el nivel de exactitud es superior tanto en la red neuronal, como en la regresión logística.

- "Data Mining Techniques for Credit Card Fraud Detection: Empirical Study" (Fahmi, Hamdy and Nagati, 2016). En este estudio se comparan empíricamente los algoritmos: Kneighbourhood, J48, Naive Bayes y Support Vector Machines, en base a la tasa de verdaderos positivos, falsos negativos, la media ambos y el MCC - Matthews Correlation Coefficient, sobre un dataset de operaciones de fraude en tarjetas de crédito "UCSD-FICO Data Mining Contest 2009" (UCSD-FICO Data Mining Contest

2009, 2009). El objetivo es determinar cuál de las cuatro técnicas es mejor en base a los indicadores. Como resultado concluyen que todas las técnicas tienen resultados similares y que por tanto no existe un algoritmo que claramente sea superior al resto en esta área de concomimiento.

El presente trabajo, tal y como se ha explicado previamente con mayor detalle, realiza una revisión de veinticuatro algoritmos de inteligencia artificial, para lo cual no se ha identificado un estudio previo de similares características. El presente estudio puede servir de punto de partida para entidades centradas en este área de conocimiento, de cara a iniciar el proceso de búsqueda de modelos de predicción directamente con los algoritmos identificados en el presente trabajo como los algoritmos más óptimos en los procesos de predicción dentro de un proceso de investigación global.

Por último, indicar que se han analizado otros muchos estudios ya que existe una amplia bibliografía sobre esta área de conocimiento, que proporcionan diversa información para conocer las técnicas utilizadas actualmente, si bien no cuantifican sus resultados para poder compararlos con los obtenidos en el presente trabajo: “Toward Scalable Learning with Non-uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection” (Chan and Stolfo, 1998), - “Understanding Credit Card Frauds” (Bhatla, Prabhu and Dua, 2003), “Credit card fraud and detection techniques: a review” (Delamaire , Abdou and Pointon, 2009), “A Proposed Classification of Data Mining Techniques in Credit Scoring” (Keramati and Yousefi, 2011), “Comparative analysis of data mining methods for predicting credit default probabilities in a retail bank portfolio” (Tudor, Bâra and Vasilica Oprea, s. f.).

4 Arquitectura software y hardware utilizada para las pruebas

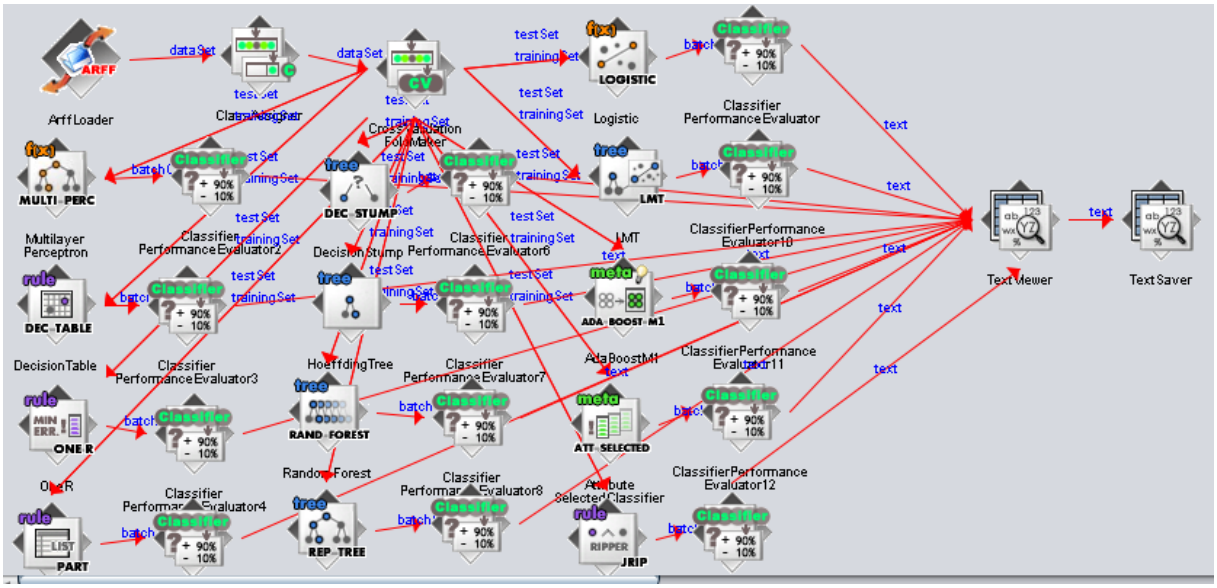
Para la realización de las pruebas se ha dispuesto de un equipo portátil con las siguientes características de hardware y software:

- Arquitectura hardware:
  - Memoria RAM: 16 Gb
  - Microprocesadores Intel: 7

Indicar que se ha probado a utilizar arquitecturas inferiores basadas en 4 Gb de memoria RAM y 2 Microprocesadores Intel, para los que se triplicaba el tiempo de ejecución de los algoritmos.

- Arquitectura software:
  - Sistema operativo: Windows 10
  - Software de Data Mining y Machine Learning: Weka versión 3.8.1.
  - Componentes de Weka utilizados:
    - Package manager: para la instalación de paquetes adicionales como SMOTE (Chawla, Bowyer, Hall and Kegelmeyer, 2002), clasificadores logísticos, etc.
    - Explorer: para el preprocesado del fichero de datos (normalización, eliminación de outliers, etc.)
    - KnowledgeFlow: para el desarrollo de un proceso batch para el tratamiento de los datos mediante algoritmos de inteligencia artificial. Inicialmente todas las pruebas se han realizado mediante la ejecución de algoritmos de forma manual en Weka, si bien, tras el estudio del funcionamiento del sistema y el mejor conocimiento de los diferentes componentes de Weka, se ha desarrollado el siguiente flujo en la herramienta KnowledgeFlow (ver Ilustración 1).

Ilustración 1 – Flujo desarrollado para la ejecución de algoritmos



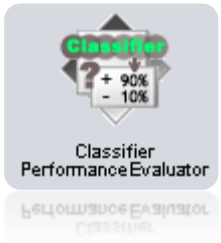
No obstante, indicar que se han desarrollado otras muchas versiones del flujo para intentar automatizar la generación de los resultados en secuencia, con el fin de facilitar su posterior análisis en tablas, si bien no ha sido posible mediante la utilización de bloques, por lo que su traspaso a tablas se ha realizado de forma manual. En el flujograma se distinguen principalmente siete tipos de componentes:

Este componente permite cargar el fichero de datos en formato `arff` utilizado por Weka dentro del flujo.

Este componente determina qué atributo del fichero de datos cargado es el que contiene la clase, aspecto fundamental para la ejecución de todos los algoritmos de clasificación supervisada.

Este componente genera dos datasets de forma automática, uno para training y otro para test (entrenamiento y pruebas).

Este componente, es similar a cualquiera de los utilizados para clasificar. Determina qué algoritmo de clasificación estamos utilizando.



Este componente genera, a partir de los resultados obtenidos por el algoritmo de clasificación, todos los datos de evaluación del mismo, como son: confusion matrix, precision, recall, TP Rate, FP Rate, etc.

Este componente transforma los resultados obtenidos del componente anterior en formato texto.

Este componente guarda los datos del componente anterior en un fichero de texto plano.

Los algoritmos utilizados en el presente trabajo, incluidos en el flujo de KnowledgeFlow, han sido los siguientes:

- Algoritmos de clasificación de funciones:
  - Logistic
  - multilayerperceptron
- Algoritmos de clasificación de reglas:
  - Decisiontable
  - ONER
  - Part
  - JRIP
- Algoritmos de clasificación de árboles:
  - Decisionstump
  - Hoeffdingtree
  - Randomforest
  - Reptree
  - LMT
- Meta-algoritmos de clasificación:
  - Adaboostm1
  - Attributeselectedclassifier

No obstante, indicar que adicionalmente se han ejecutado los siguientes algoritmos en Weka de forma manual:

- Algoritmos de clasificación bayesianos:
  - BayesNet
  - NaiveBayesMultinomialText
  - NaiveBayesUpdateable
- Algoritmos de clasificación de funciones:
  - SGD
  - SMO
  - VotedPerceptron
- Algoritmos de clasificación lazy:
  - LBK
- Algoritmos de clasificación de reglas:
  - ZeroR
- Algoritmos de clasificación de árboles:
  - J48
  - RandomTree

A continuación se describen brevemente la tipología de algoritmos aplicados:

- Algoritmos de clasificación bayesianos. Este tipo de algoritmos se basan en aplicar probabilidad bayesiana sobre sucesos incompatibles condicionados sobre otro suceso.
- Algoritmos de clasificación de funciones. Este tipo de algoritmos se basan en aplicar funciones matemáticas sobre los datos, como puedan ser técnicas de regresión logística o funciones matemáticas aplicadas a los nodos de las redes neuronales.
- Algoritmos de clasificación lazy. Este tipo de algoritmos se basan en no desarrollar un modelo generalizando una solución hasta que se presentan casos de prueba, siendo necesario almacenar los datos de entrenamiento para poder clasificar los casos de prueba, a diferencia de los algoritmos tradicionales que generan el modelo a partir de los datos de entrenamiento y no necesitan almacenar los datos de entrenamiento para clasificar a un caso de prueba.
- Algoritmos de clasificación de reglas. Este tipo de algoritmos analizan los datos para encontrar dependencias entre ellos con la finalidad de establecer reglas basadas en los valores de los atributos, de forma que si se cumple el antecedente de la regla se obtiene el consecuente.
- Algoritmos de clasificación de árboles. Este tipo de algoritmos se basan en la identificación de clases mediante la generación de árboles de decisión basados en atributos. Este tipo e algoritmos también pueden representarse mediante reglas.
- Meta-algoritmos de clasificación. Este tipo de algoritmos se basan en aplicar múltiples algoritmos para la identificación de clases. Por ejemplo, de selección de atributos y de clasificación posterior, o de ejecución de múltiples algoritmos de clasificación para evaluar el mejor de todos ellos, mediante el ajuste paramétrico de ellos.

Dichos algoritmos han sido seleccionados a partir del listado de algoritmos de la aplicación Weka, para los cuales los atributos son compatibles, ya que no todos los algoritmos pueden manejar atributos numéricos. El objetivo era identificar qué algoritmos se podían seleccionar sin modificar los datos de entrada, ya que, por ejemplo, discretizar variables continuas puede implicar un sesgo o pérdida de información inherente, al tener que establecer rangos de valores cuyo criterio de asignación puede variar, y con él la exactitud de los algoritmos de clasificación asociados.

5 Proceso de análisis

En el presente apartado se describe la estructura del dataset, así como los resultados de los algoritmos al aplicar las diversas técnicas de tratamiento comentadas previamente (selección de variables, eliminación de valores fuera de rango, etc.), para finalmente seleccionar los algoritmos más efectivos.

5.1 Estructura del dataset

El data set analizado ha sido publicado en el año 2016, y contiene 23 atributos, lo que permite caracterizar a cada cliente por su perfil de pago de forma muy completa, identificando, entre otros aspectos, el importe a cobrar, el importe cobrado, y los días de impago, historificados por meses, además de otros datos relativos al perfil del pagador, como puedan ser el género, el nivel educativo o la edad. Asimismo contiene un elevado número de instancias - 30.000 - para que los algoritmos tengan ejemplos suficientes para poder generar modelos precisos y confiables. Asimismo posee valores numéricos en todos sus atributos, si bien se pueden discretizar sin sesgo aquellos que se basan en un listado de valores numéricos. Por último, el dataset se encuentra suficientemente balanceado para poder inferir nuevas clases sin necesidad de realizar técnicas de balanceo previas.

El dataset original se encuentra en formato Excel, en la página web del Center for Machine Learning and Intelligent Systems (Default of credit card clients Data Set, 2016), disponiendo de la siguiente estructura interna de campos del data set, que serán los atributos a analizar:

- Clase (variable explicada): 0 implica que no hay impago; 1 implica que sí hay impago.
- 23 Atributos (variables explicativas):
    - X1: Cantidad del crédito concedido al cliente en dólares taiwaneses sin decimales.
    - X2: Género: 1 = masculino; 2 = femenino.
    - X3: Nivel educativo: 1 = colegio; 2 = universidad; 3 = instituto; 4 = otros.
    - X4: Estado civil: 1 = casado; 2 = soltero; 3 = divorciado; 0 = otros.
    - X5: Edad en años.
    - X6 a X11: Historia de pagos mensuales pasados realizados por el cliente desde de abril a septiembre de 2005 hacia atrás. Ej.: X6 = el estado de reembolso en septiembre de 2005; X7 = el estado de reembolso en agosto de 2005; ...; X11 = el estado de reembolso en abril de 2005. Para cada uno de los campos

anteriores, se pueden tomar uno de los siguiente valores: -2 = ningún consumo; -1 = pagan debidamente; 0 = uso de tarjetas de crédito renovables; 1 = el pago se retrasa un mes; 2 = el pago se retrasa dos meses; ...; 8 = el pago se retrasa ocho meses; 9 = el pago se retrasa nueve meses o más.

- X12-X17: Cantidad facturada al cliente en cada mes en dólares taiwaneses sin decimales. X12 = cantidad de facturada en septiembre de 2005; X13 = cantidad facturada en agosto de 2005; ...; X17 = cantidad de facturada en abril de 2005.
- X18-X23: Cantidad de pago anterior (NT dólar). X18 = cantidad pagada en septiembre de 2005; X19 = cantidad pagada en agosto de 2005; ...; X23 = cantidad pagada en abril de 2005.

Se considera que el dataset se encuentra balanceado, presentando la siguiente distribución inicial (ver Tabla 4):

Tabla 4 – Distribución inicial la clase sobre la población del dataset inicial

CLASE	Dataset completo	
0	77,88%	23.364
1	22,12%	6.636
TOTAL		**30.000**

5.2 Identificación inicial de algoritmos de predicción

Este apartado tiene como objetivo identificar, de forma preliminar, sin realizar modificaciones sobre el dataset, el nivel de exactitud de los diferentes algoritmos de clasificación seleccionados en Weka. Tal y como se ha explicado previamente con mayor detalle en el apartado 1 Introducción y objetivos, se considerado la tasa de error de la exactitud**¡Error! No se encuentra el origen de la referencia.**.

Para ello se han ejecutado los algoritmos abajo indicados (ver Ilustración 2) con la finalidad de identificar el nivel de exactitud definido, junto con el tiempo de ejecución de cada uno de ellos. A continuación se muestra cómo se mide la exactitud:

Ecuación 1 – Exactitud

$$Exactitud = \frac{Nº\ de\ verdaderos\ positivos + Nº\ de\ verdaderos\ negativos}{Nº\ de\ verdaderos\ positivos + Nº\ de\ verdaderos\ negativos + Nº\ de\ falsos\ positivos + Nº\ de\ falsos\ negativos}$$

No obstante, por comodidad en el presente trabajo se ha utilizado la tasa de error en la exactitud, es decir, se ha considerado como exactitud:

Ecuación 2 – Error en la exactitud

$$Error\ en\ la\ exactitud = 1 - \frac{Nº\ de\ verdaderos\ positivos + Nº\ de\ verdaderos\ negativos}{Nº\ de\ verdaderos\ positivos + Nº\ de\ verdaderos\ negativos + Nº\ de\ falsos\ positivos + Nº\ de\ falsos\ negativos}$$

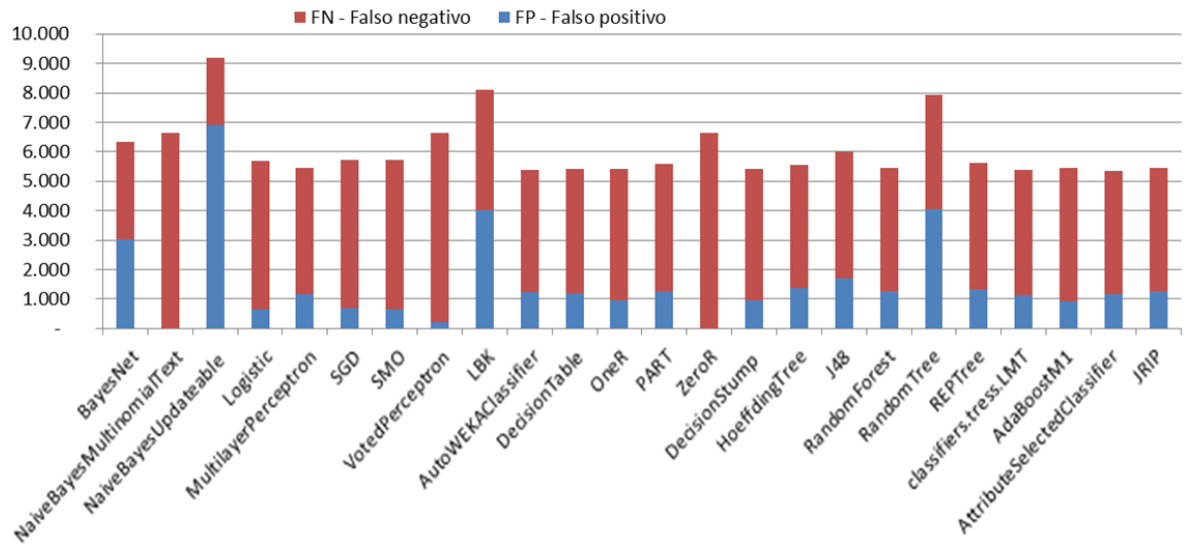
Que equivale a la siguiente fórmula:

Ecuación 3 – Error en la exactitud equivalente

$$Error\ en\ la\ exactitud = \frac{Nº\ de\ falsos\ positivos + Nº\ de\ falsos\ negativos}{Nº\ de\ verdaderos\ positivos + Nº\ de\ verdaderos\ negativos + Nº\ de\ falsos\ positivos + Nº\ de\ falsos\ negativos}$$

El resultado se pude ver en la Ilustración 2, que representa en valor absoluto la suma de errores obtenidos por cada algoritmo sumando falsos positivos y falsos negativos, ya que se ha utilizado la Ecuación 3 – Error en la exactitud equivalente, es decir, la tasa de error en la exactitud para ver cuál es más baja.

Ilustración 2 - Nivel de exactitud inicial de los algoritmos de clasificación



Como se puede observar, en general los algoritmos tienden a sacar más falsos negativos que falsos positivos, lo cual supone un riesgo en el modelo de predicción, es decir, es mejor

detectar impago cuando realmente no lo es (falso positivo), que no detectarlo cuando realmente sí lo es (falso negativo).

En la Tabla 5 se puede ver el detalle de los resultados obtenidos por cada algoritmo:

Tabla 5 - Detalle del nivel de exactitud inicial de los algoritmos de clasificación

	FP	FN	Total	FP %	FN %	Total %
BayesNet	3.048	3.295	6.343	10,16%	10,98%	21,14%
NaiveBayesMultinomialText	-	6.636	6.636	0,00%	22,12%	22,12%
NaiveBayesUpdateable	6.912	2.273	9.185	23,04%	7,58%	30,62%
Logistic	638	5.054	5.692	2,13%	16,85%	18,97%
MultilayerPerceptron	1.168	4.283	5.451	3,89%	14,28%	18,17%
SGD	699	5.011	5.710	2,33%	16,70%	19,03%
SMO	653	5.069	5.722	2,18%	16,90%	19,07%
VotedPerceptron	214	6.440	6.654	0,71%	21,47%	22,18%
LBK	4.005	4.109	8.114	13,35%	13,70%	27,05%
AutoWEKAClassifier	1.215	4.167	5.382	4,05%	13,89%	17,94%
DecisionTable	1.181	4.234	5.415	3,94%	14,11%	18,05%
OneR	944	4.479	5.423	3,15%	14,93%	18,08%
PART	1.279	4.309	5.588	4,26%	14,36%	18,63%
ZeroR	-	6.636	6.636	0,00%	22,12%	22,12%
DecisionStump	953	4.459	5.412	3,18%	14,86%	18,04%
HoeffdingTree	1.383	4.168	5.551	4,61%	13,89%	18,50%
J48	1.712	4.276	5.988	5,71%	14,25%	19,96%
RandomForest	1.270	4.178	5.448	4,23%	13,93%	18,16%
RandomTree	4.059	3.889	7.948	13,53%	12,96%	26,49%
REPTree	1.342	4.278	5.620	4,47%	14,26%	18,73%
LMT	1.138	4.236	5.374	3,79%	14,12%	17,91%
AdaBoostM1	941	4.516	5.457	3,14%	15,05%	18,19%
AttributeSelectedClassifier	1.166	4.180	5.346	3,89%	13,93%	17,82%
JRIP	1.249	4.201	5.450	4,16%	14,00%	18,17%

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Indicar que todos los algoritmos se han ejecutado con la parametrización que viene en Weka por defecto, mediante validación cruzada de 10 folders con el dataset completo con todas las instancias.

En la Tabla 5, se puede observar que el mayor índice de exactitud obtenido corresponde a los siguientes algoritmos: AttributeSelectedClassifier, LMT y AutoWEKAClassifier, ya que todos ellos se encuentran con un nivel de error inferior al 18%.

Por otra parte, se puede observar que el mayor índice de exactitud teniendo en cuenta sólo el menor número de falsos negativos corresponde a los siguientes algoritmos: BayesNet y NaiveBayesUpdateable, ya que ambos se encuentran con un nivel de error igual o inferior al 11%, si bien su el nivel de error en la exactitud es igual o superior al 21%.

Para la evaluación de algoritmos posterior, se utilizarán los algoritmos con un nivel de exactitud inferior al 19%, con la finalidad de seleccionar los algoritmos más precisos de partida descartando el resto: Logistic, MultilayerPerceptron, DecisionTable, OneR, PART, DecisionStump, HoeffdingTree, J48, RandomForest, REPTree, LMT, AdaBoostM1, AttributeSelectedClassifier, JRIP.

En la tabla 6, se puede ver el tiempo de ejecución de cada uno de los algoritmos:

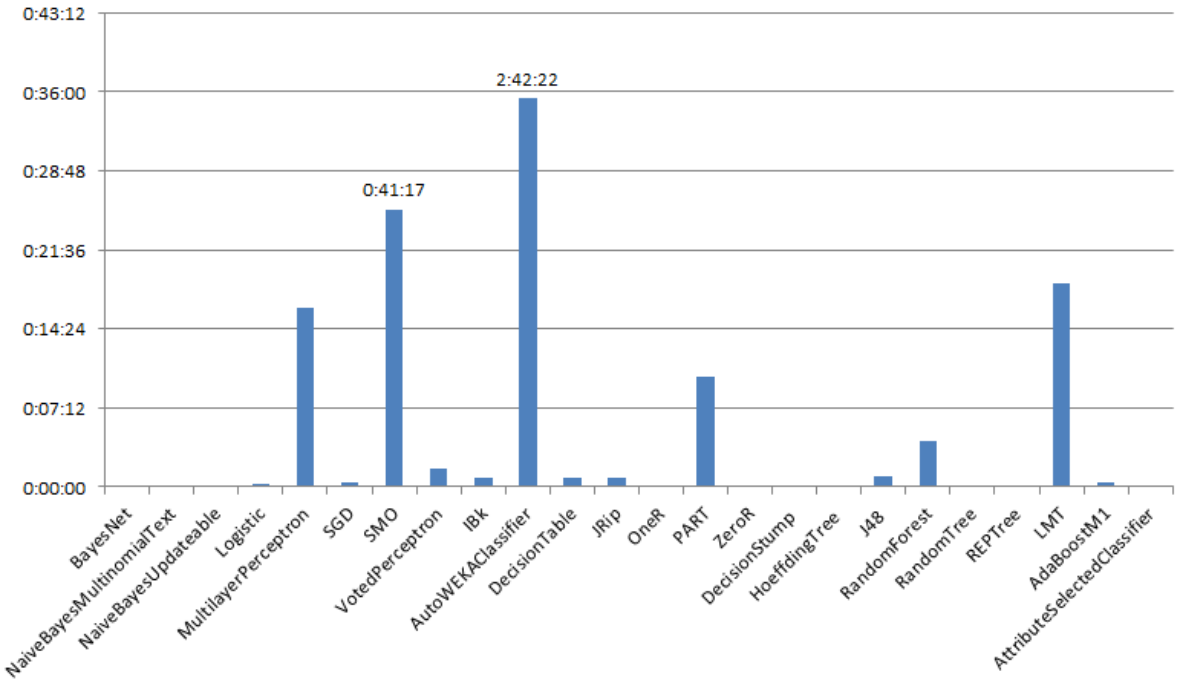
Tabla 6 - Detalle de los tiempos de ejecución inicial de los algoritmos de clasificación

	Tiempo hh:mm:ss		Tiempo hh:mm:ss
BayesNet	0:00:04	PART	0:09:58
NaiveBayesMultinomialText	0:00:00	ZeroR	0:00:01
NaiveBayesUpdateable	0:00:03	DecisionStump	0:00:03
Logistic	0:00:16	HoeffdingTree	0:00:06
MultilayerPerceptron	0:16:16	J48	0:00:56
SGD	0:00:27	RandomForest	0:04:06
SMO	0:25:17	RandomTree	0:00:05
VotedPerceptron	0:01:39	REPTree	0:00:11
LBk	0:00:47	LMT	0:18:29
AutoWEKAClassifier	0:35:22	AdaBoostM1	0:00:24
DecisionTable	0:00:51	AttributeSelectedClassifier	0:00:09
OneR	0:00:02	JRip	0:00:51

Nota: Los tiempos cercanos a cero se expresarán como 0:00:00.

Gráficamente se ve claramente forma más sencilla la diferencia de tiempos, existiendo dos algoritmos que claramente son superiores en tiempo a la media de los algoritmos (ver Ilustración 3):

Ilustración 3 – Tiempo de ejecución de algoritmos de clasificación



Como se puede observar, los algoritmos que menos tiempo tardan son los siguientes: BayesNet, NaiveBayesMultinomialText, NaiveBayesUpdateable, Logistic, SGD, VotedPerceptron, LBk, DecisionTable, JRip, OneR, ZeroR, DecisionStump, HoeffdingTree, J48, RandomTree, REPTree, AdaBoostM1, AttributeSelectedClassifier, ya que todos ellos tienen un tiempo inferior a un minuto.

Cabe destacar, que sólo uno de ellos, AttributeSelectedClassifier, tiene un tiempo de ejecución inferior a un minuto con un nivel de error inferior al 18%, por tanto es el algoritmo a priori más eficiente de los ejecutados.

Adicionalmente a los criterios de exactitud y tiempos de ejecución, determinados algoritmos de clasificación seleccionan las variables que consideran más relevantes para el modelo. En la tabla 7 se muestran los atributos seleccionados por los algoritmos de clasificación, identificando con una “S” aquellos atributos seleccionados, y en blanco aquellos algoritmos

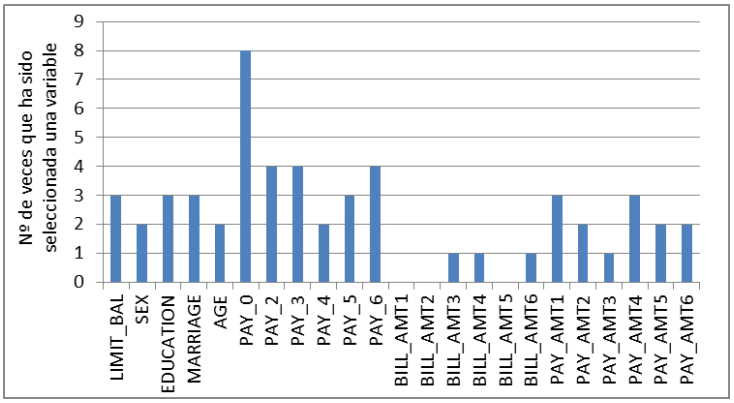
que no realizan selección de variables:

Tabla 7 - Detalle de las variables seleccionadas por los algoritmos de clasificación

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	Default payment next month
BayesNet																								
NaiveBayesMultinomialText																								
NaiveBayesUpdateable																								
Logistic		S	S	S	S	S	S	S	S	S	S													
MultilayerPerceptron																								
SGD																								
SMO																								
VotedPerceptron																								
IBk																								
AutoWEKAClassifier																								
DecisionTable																								
JRip	S				S	S	S			S	S			S				S			S	S	S	
OneR						S																		
PART																								
ZeroR																								
DecisionStump						S																		
HoeffdingTree				S		S									S		S	S	S	S	S	S	S	
J48	S	S	S	S																				
RandomForest																								
RandomTree																								
REPTree																								
LMT			S			S	S	S			S													
AdaBoostM1	S					S		S			S							S	S		S			
AttributeSelectedClassifier						S	S	S	S	S														
Atributos seleccionados	**S**		**S**	**S**		**S**	**S**	**S**		**S**	**S**							**S**			**S**		**S**	
TOTAL columna	**3**	**2**	**3**	**3**	**2**	**8**	**4**	**4**	**2**	**3**	**4**	**0**	**0**	**1**	**1**	**0**	**1**	**3**	**2**	**1**	**3**	**2**	**2**	

En la siguiente gráfica se va más claramente (ver Ilustración 4):

Ilustración 4 – Representación del número de veces que ha sido seleccionada cada variable

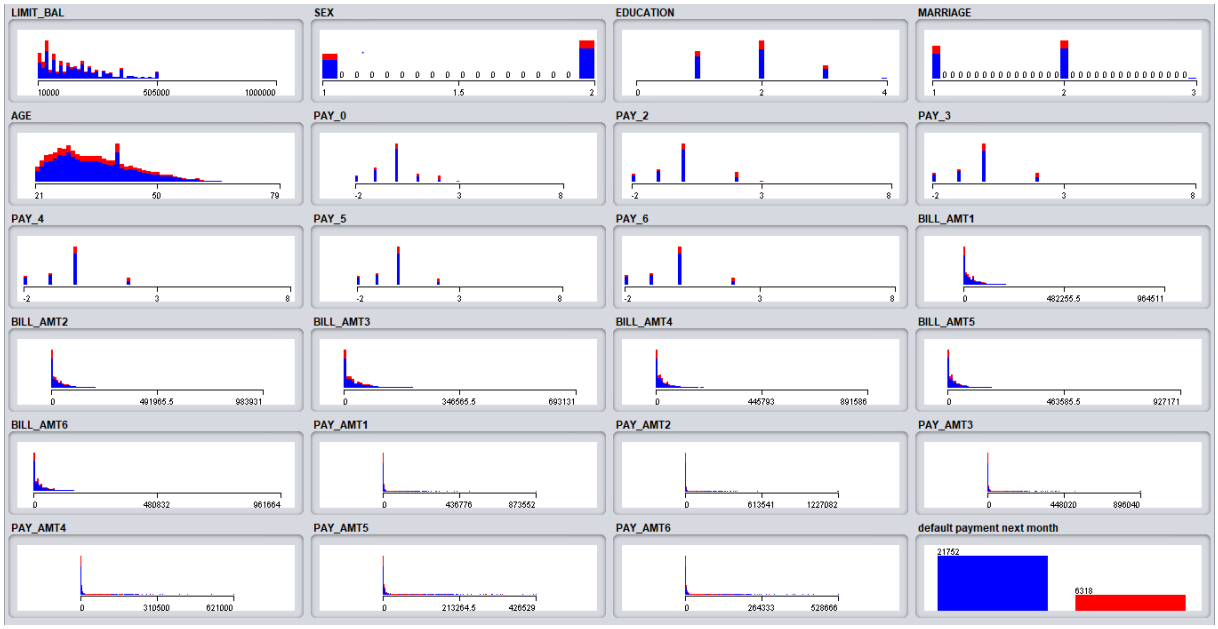


Como se puede observar en la Ilustración 4, los atributos más relevantes, de acuerdo al número de veces que han sido seleccionados por los diferentes algoritmos son los siguientes: LIMIT_BAL, EDUCATION, MARRIAGE, PAY_0, PAY_2, PAY_3, PAY_5, PAY_6, PAY_AMT1, PAY_AMT4, PAY_AMT6, ya que todos ellos han sido seleccionadas por tres o más algoritmos de clasificación. Asimismo, cabe destacar que el importe facturado es poco relevante para la mayoría de algoritmos de clasificación BILL_AMT1 a 6.

5.3 Análisis de los valores de los atributos del dataset

La distribución de los datos del dataset es la siguiente (ver Fichero 5):

Ilustración 5 – Distribución de los campos del dataset



A continuación, se detalla el análisis realizado de los campos del dataset:

- Clase (variable explicada) DEFAULT PAYMENT NEXT MONTH:
  - *Valores que toman las instancias*: 0, 1.
  - *Valores nulos*: no tiene.
  - *Valores anómalos*: no tiene.
- 23 Atributos (variables explicativas):
  - LIMIT_BAL: Cantidad del crédito.
    - *Máximo*: 1.000.000
    - *Mínimo*: 10.000
    - *Media*: 167.484,323
    - *Desviación estándar*: 129.747,662
    - *Valores nulos*: no tiene.
    - *Valores anómalos*: no tiene.
  - SEX: Género.
    - *Valores que toman las instancias:* 0, 1*.*
    - *Valores nulos*: no tiene.
    - *Valores anómalos*: no tiene.
  - EDUCATION: Nivel educativo.
    - *Valores que toman las instancias:* 1, 2, 3, 4, 5 y 6*.*
    - *Valores nulos*: no tiene.
    - *Valores anómalos*: 0, 5 y 6 no descritos por el autor. Puesto que el autor considera el valor 4 como “otros valores”, he procedido a unificar los valores 0, 5 y 6 en el valor 4.
  - MARRIAGE: Estado civil.
    - *Valores que toman las instancias: 0,* 1, 2 y 3*.*
    - *Valores nulos*: no tiene.
    - *Valores anómalos*: no tiene, una vez resueltas dudas por el autor del dataset.
  - AGE: Edad en años.
    - *Máximo*: 79
    - *Mínimo*: 21
    - *Media*: 35,486
    - *Desviación estándar*: 9.218.
    - *Valores nulos*: no tiene.
    - *Valores anómalos*: no tiene.

- PAY_0 a PAY_6: Historia de pagos mensuales pasados.
  - *Valores que toman las instancias:* -2, -1, 0, 2, 3, 4, 5, 6, 7 y 8.
  - *Valores nulos*: no tiene.
  - *Valores anómalos*: no tiene, una vez resueltas dudas por el autor del dataset.
- BILL_AMT0 a BILL_AMT6: Cantidad facturada al cliente en cada mes.
  - *Máximo*: 1.664.089 - Valor calculado sobre todos los campos.
  - *Mínimo*: -339.603 - Valor calculado sobre todos los campos, ya que todos ellos toman valores negativos en el valor mínimo.
  - *Valores nulos*: no tiene.
  - *Valores anómalos*: puesto que la cantidad a facturar no puede ser negativa, he optado por eliminar del dataset los valores negativos de facturación.
- PAY_AMT0 a PAY_AMT6: Cantidad de pago anterior (NT dólar). X18 = cantidad pagada en septiembre de 2005; X19 = cantidad pagada en agosto de 2005; ...; X23 = cantidad pagada en abril de 2005.
  - *Máximo*: 1.684.259
  - *Mínimo*: 0
  - *Valores nulos*: no tiene.
  - *Valores anómalos*: no tiene.

Para poder obtener esta información sobre el data set, ha sido necesario solicitarla al autor I-Cheng Yeh, ya que la descripción inicial de los campos es incompleta, en relación a los valores que toman los campos, existiendo valores en los campos X3 a X17 (sin X5) que no son coherentes o no vienen en la descripción de la tipología de valores posibles indicada inicialmente por el autor existiendo valores anómalos.

El resultado una vez aplicados todos los filtros es el siguiente:

- Instancias iniciales: 30.000
- Instancias con impagos iniciales: 6.636 – 22,12%
- Instancias con valores anómalos:
  - EDUCATION. Nivel educativo. No supone reducción de instancias. Cambio de valores 4, 5 y 6 por valor 4.
  - BILL_MT0 a BILL_AMT6. Cantidad facturada al cliente en cada mes. Eliminación de instancias con valores negativos de facturación.
- Instancias incorrectas eliminadas: 1.930

- Instancias finales una vez desechadas las incorrectas: 28.070
- Instancias con impagos finales: 6.318 – 22,51%

Por tanto, para tratamientos posteriores, se tomará como fichero de referencia el obtenido en el presente apartado, con 28.070 instancias.

En los apartados siguientes, se muestran los diferentes procesos realizados sobre el dataset obtenido en el presente apartado para intentar mejorar el nivel de exactitud de los algoritmos así como disminuir el tiempo de ejecución.

5.4 División del dataset en entrenamiento y prueba

Para la realización posterior de pruebas, se ha dividido el dataset obtenido en el presente apartado como dataset de partida, con 28.070 instancias, en dataset de entrenamiento y de prueba, mediante la función de Weka `Resample`, obteniendo la siguiente población (ver Tabla 8):

Tabla 8 – Distribución del nivel de balanceo de la clase sobre la población de los dataset de training y de test

Clase	TEST		TRAINING	
0 – Pago	77,82%	8.738	77,27%	13.014
1 - Impago	22,18%	2.490	22,73%	3.828
TOTAL		**11.228**		**16.842**

Ello permitirá utilizar datos distintos para el entrenamiento de los modelos, para finalmente verificar sus valores de exactitud, precisión, etc. con datos de prueba.

Como se puede observar los porcentajes de balanceo del data set inicial se mantienen, ya que la función `Resample` mantiene la distribución que tiene la clase sobre el dataset inicial, sobre los nuevos datasets generados de entrenamiento y pruebas.

5.5 Identificación de atributos relevantes del dataset – selección de características

Para la identificación de los atributos relevantes, se han ejecutado los siguientes algoritmos que ofrece Weka con la finalidad de tener un conjunto de algoritmos que permita determinar qué atributos son seleccionados por la mayoría de ellos, es decir, si existe una tendencia

clara en el proceso de selección: OneRAttributeEval, InfoGainAttributeEval, GainRatioAttributeEval, CorrelationAttributeEval, SymmetricalUncertAttributeEval, ReliefFAttributeEval y PrincipalComponents.

Por otra parte, se ha calculado la matriz de correlación de Pearson calculada mediante la herramienta Microsoft Excel entre las variables para determinar el nivel de afinidad de las mismas, que es a su vez utilizada por el algoritmo CorrelationAttributeEval para la selección de atributos(ver Tabla 9):

Tabla 9 – Matriz de correlación de Pearson

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default payment next month
LIMIT_BAL	1																							
SEX	0,024	1																						
EDUCATION	-0,23	0,013	1																					
MARRIAGE	-0,1	-0,03	-0,14	1																				
AGE	0,138	-0,08	0,156	**-0,37**	1																			
PAY_0	-0,27	-0,06	0,111	0,021	-0,04	1																		
PAY_2	-0,29	-0,08	0,125	0,023	-0,05	**0,69**	1																	
PAY_3	-0,28	-0,07	0,119	0,036	-0,06	**0,586**	**0,766**	1																
PAY_4	-0,26	-0,07	0,113	0,036	-0,05	**0,548**	**0,664**	**0,779**	1															
PAY_5	-0,25	-0,06	0,1	0,035	-0,05	**0,516**	**0,623**	**0,691**	**0,823**	1														
PAY_6	-0,23	-0,05	0,086	0,035	-0,05	**0,482**	**0,576**	**0,638**	**0,722**	**0,82**	1													
BILL_AMT1	**0,349**	-0,03	-0,02	-0,04	0,062	0,106	0,117	0,101	0,099	0,105	0,107	1												
BILL_AMT2	**0,344**	-0,03	-0,02	-0,03	0,06	0,103	0,114	0,114	0,108	0,111	0,114	**0,853**	1											
BILL_AMT3	**0,333**	-0,02	-0,02	-0,04	0,057	0,118	0,134	0,137	0,146	0,149	0,151	**0,793**	**0,824**	1										
BILL_AMT4	**0,344**	-0,02	-0,03	-0,03	0,058	0,101	0,113	0,115	0,121	0,136	0,139	**0,757**	**0,799**	**0,817**	1									
BILL_AMT5	**0,338**	-0,02	-0,03	-0,03	0,054	0,097	0,109	0,11	0,116	0,13	0,142	**0,731**	**0,765**	**0,776**	**0,836**	1								
BILL_AMT6	**0,335**	-0,02	-0,04	-0,03	0,052	0,092	0,106	0,108	0,115	0,127	0,139	**0,707**	**0,74**	**0,749**	**0,809**	**0,86**	1							
PAY_AMT1	0,08	0,001	-0,01	0,002	0,004	-0,03	-0,04	-0	-0,01	-0	-0	0,017	0,15	0,123	0,11	0,089	0,088	1						
PAY_AMT2	0,08	-0,01	-0,01	0,004	0,007	-0,03	-0,03	-0,03	-0	-0	-0	0,031	0,018	0,128	0,116	0,093	0,086	0,025	1					
PAY_AMT3	0,088	-0,01	-0,02	0,009	0,009	-0,03	-0,03	-0,03	-0,03	-0	6E-04	0,038	0,042	0,017	0,16	0,11	0,114	0,086	0,216	1				
PAY_AMT4	0,095	-0,01	-0,01	-0,01	0,006	-0,03	-0,03	-0,03	-0,03	-0,04	-0	0,071	0,043	0,044	0,026	0,148	0,118	0,04	0,022	0,044	1			
PAY_AMT5	0,127	-0,02	-0,02	-0	0,017	-0,03	-0,03	-0,03	-0,03	-0,03	-0,04	0,078	0,067	0,067	0,055	0,044	0,151	0,078	0,026	0,074	0,07	1		
PAY_AMT6	0,124	-0,01	-0,02	9E-05	0,015	-0,03	-0,02	-0,02	-0,02	-0,02	-0,02	0,097	0,089	0,098	0,07	0,074	0,043	0,104	0,069	0,093	0,078	0,08	1	
default payment next month	-0,15	-0,04	0,037	-0,03	0,006	0,337	0,271	0,24	0,222	0,21	0,193	-0,02	-0,02	-0,02	-0,01	-0,01	-0,01	-0,03	-0,02	-0,02	-0,02	-0,03	-0,03	1

En la Tabla 9, se observan en color resaltado aquellas celdas que tienen un nivel de correlación igual o superior a +0,3, e igual o inferior a -0,3, es decir, que existe cierta dependencia entre los valores de dos variables porque guardan algún tipo de relación, como AGE y MARRIAGE. Lo que se observa en la tabla 9 es que, en general, no existe una elevada dependencia de las variables entre ellas, aunque a priori pudiera suponerse la existencia de una relación entre el retraso en el pago PAY_0 a PAY_6 y la cantidad pagada PAY_AMT0 a PAY_AMT6, o entre dichas variables y la probabilidad de impago. La ausencia de dependencia entre las variables dificulta el proceso de predicción para determinar tendencias así como de selección de variables.

Tras la ejecución de los algoritmos de selección de atributos, se obtienen los resultados mostrados en la Tabla 10:

Tabla 10 – Variables identificadas por los algoritmos de selección de variables

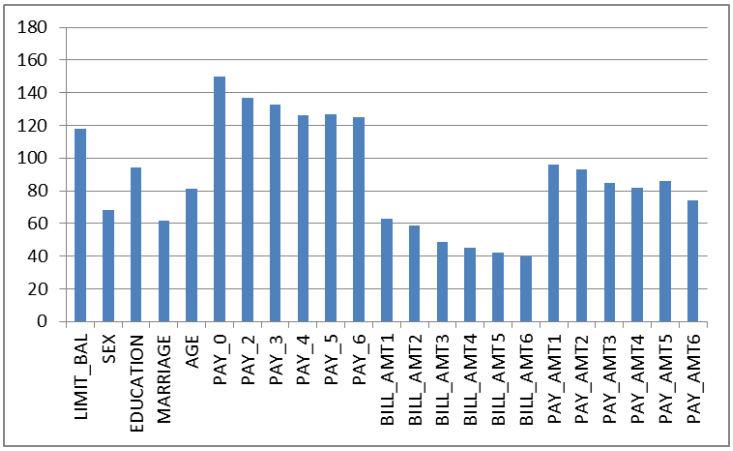
	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6
TOTAL columna	95	45	71	39	58	127	114	110	103	104	102	40	36	26	22	19	17	73	70	62	59	63	51
OneRAttributeEval	8	9	7	10	11	1	2	5	4	3	6	22	23	20	21	18	19	16	12	15	14	13	17
InfoGainAttributeEval	8	16	14	19	15	1	2	3	4	5	6	17	18	22	20	23	21	7	9	10	11	12	13
GainRatioAttributeEval	10	15	14	19	16	3	2	1	4	5	6	18	17	23	23	23	23	7	8	9	12	11	13
CorrelationAttributeEval	7	14	15	16	21	1	2	3	4	5	6	17	18	19	20	22	23	8	9	11	12	10	13
SymmetricalUncertAttributeEval	9	16	14	19	15	1	2	3	4	5	6	17	18	23	23	23	23	7	8	10	11	12	13
ReliefFAttributeEval	1	23	3	16	2	4	14	13	15	11	6	7	8	5	9	10	12	20	22	21	19	17	18

En la tabla 10, se indica para cada algoritmo y para cada variable el orden en el que algoritmo ha seleccionado la variable, es decir, la más relevante será la que tenga número 1, y la menos relevante la que tenga el número 23. Indicar que las variables BILL_AMT3 a 6 no han sido seleccionadas por los algoritmos GainRatioAttributeEval y SymmetricalUncertAttributeEval, por lo que se les ha asignado el valor más desfavorable, es decir el 23.

En la fila en la que se indica TOTAL columna, refleja el número total de atributos (23) multiplicado por el número de algoritmos analizados (5), menos la suma de los valores de cada fila. La finalidad es que si por ejemplo, el atributo LIMIT_BAL fuese relevante para los 5 algoritmos, tendría en todos ellos la posición 1, y por tanto la suma de la columna sería 5, que sería el valor más bajo de todos los atributos. Para invertir este proceso, y que tenga el valor más alto posible y poder representarlos gráficamente, en el ejemplo anterior el cálculo realizado sería el siguiente: 23*5 – 5 = 110, que sería el valor más alto posible de todos los atributos.

Como resultado se obtiene la gráfica de la Ilustración 6:

Ilustración 6 – Identificación de los atributos más relevantes



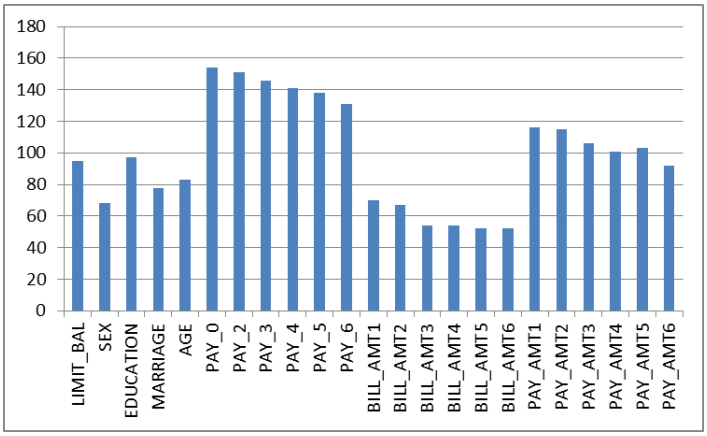
No obstante, en la Tabla 10 se observa que el algoritmo ReliefFAttributeEval difiere sensiblemente de todos los anteriores, ya que por ejemplo, todo los campos relativos a la forma de pago del clientes PAY_0 a 6 son los más relevantes para todos los algoritmos con posiciones de 1 a 6, mientras que para el citado algoritmo toman posiciones superiores a 10 en 4 de los 6 atributos. Es por ello, que se ha eliminado del total el citado algoritmo obteniendo una clasificación total ajustada a los atributos seleccionados mayoritariamente (ver Tabla 11):

Tabla 11 – Variables identificadas por los algoritmos de selección de variables ajustada

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6
TOTAL columna	95	68	97	78	83	154	151	146	141	138	131	70	67	54	54	52	52	116	115	106	101	103	92

La tabla 11 se muestra gráficamente a continuación (ver Ilustración 7):

Ilustración 7 – Identificación de los atributos más relevantes mejorada



Como se puede observar en la gráfica anterior, no hay ningún valor a cero puesto que no hay ningún atributo que para todos los algoritmos esté en el orden 23, que sería el caso más desfavorable, así que si establecemos un punto de corte en 75 puntos ya que visualmente selecciona las variables más relevantes, los atributos seleccionados son los siguientes:

- Variables relativas al retraso en el pago de los clientes, PAY_0 a 6. Es decir, si el cliente ha pagado debidamente o si tiene 'n' meses de retraso.
- Variables relativas al pago real de las facturas por parte de los clientes, PAY_AMT1 a 6. Es decir, si el cliente ha pagado las facturas en su totalidad o en parte.
- El límite máximo concedido LIMIT_BAL.
- El nivel de educación EDUCATION.
- La edad AGE.
- Y el estado civil MARRIAGE.

Asimismo, tal y como se indicó previamente en el proceso de selección de algoritmos, cabe destacar que:

- Las variables seleccionadas son prácticamente las mismas, con la única deferencia de la edad AGE, PAY_4, PAY_AMT2, PAY_AMT3 y PAY_AMT5 que en los algoritmos de clasificación no fueron seleccionadas.
- El importe facturado es poco relevante para la mayoría de algoritmos BILL_AMT1 a 6, así como el estado civil de la persona.

Una vez identificadas las variables más relevantes, vamos a proceder a reejecutar los algoritmos de clasificación con la finalidad de determinar: (1) si se mantienen los mismos niveles de error pese a tener menos datos; (2) si se mejoran los tiempos de respuesta (ver Tabla 12).

Tabla 12 – Comparativa de la exactitud de los algoritmos de clasificación con y sin selección de atributos

	FP	FN	Total con selección atributos	FP %	FN %	Total con selección atributos %	Total sin selección atributos %	Diferencia %
BayesNet	2.997	3.051	6.343	10,68%	10,87%	22,60%	21,14%	-1,45%
NaiveBayesMultinomialText	-	6.318	6.636	0,00%	22,51%	23,64%	22,12%	-1,52%
NaiveBayesUpdateable	5.215	2.332	9.185	18,58%	8,31%	32,72%	30,62%	-2,11%
Logistic	648	4.673	5.692	2,31%	16,65%	20,28%	18,97%	-1,30%
MultilayerPerceptron	1.183	3.945	5.451	4,21%	14,05%	19,42%	18,17%	-1,25%
SGD	846	4.463	5.710	3,01%	15,90%	20,34%	19,03%	-1,31%

	FP	FN	Total con selección atributos	FP %	FN %	Total con selección atributos %	Total sin selección atributos %	Diferencia %
SMO	653	5.069	5.722	2,33%	18,06%	20,38%	19,07%	-1,31%
VotedPerceptron	-	6.318	6.654	0,00%	22,51%	23,71%	22,18%	-1,53%
LBK	3.843	3.800	8.114	13,69%	13,54%	28,91%	27,05%	-1,86%
AutoWEKAClassifier	975	4.093	5.382	3,47%	14,58%	19,17%	17,94%	-1,23%
DecisionTable	1.105	3.987	5.415	3,94%	14,20%	19,29%	18,05%	-1,24%
OneR	926	4.181	5.423	3,30%	14,89%	19,32%	18,08%	-1,24%
PART	1.092	4.051	5.588	3,89%	14,43%	19,91%	18,63%	-1,28%
ZeroR	-	6.318	6.636	0,00%	22,51%	23,64%	22,12%	-1,52%
DecisionStump	2.969	3.020	5.412	10,58%	10,76%	19,28%	18,04%	-1,24%
HoeffdingTree	1.074	4.079	5.551	3,83%	14,53%	19,78%	18,50%	-1,27%
J48	1.389	3.941	5.988	4,95%	14,04%	21,33%	19,96%	-1,37%
RandomForest	1.355	3.860	5.448	4,83%	13,75%	19,41%	18,16%	-1,25%
RandomTree	3.793	3.747	7.948	13,51%	13,35%	28,31%	26,49%	-1,82%
REPTree	1.285	3.947	5.620	4,58%	14,06%	20,02%	18,73%	-1,29%
LMT	1.135	3.977	5.374	4,04%	14,17%	19,14%	17,91%	-1,23%
AdaBoostM1	1.499	3.948	5.457	5,34%	14,06%	19,44%	18,19%	-1,25%
AttributeSelectedClassifier	1.003	4.063	5.346	3,57%	14,47%	19,05%	17,82%	-1,23%
JRIP	1.255	3.849	5.450	4,47%	13,71%	19,42%	18,17%	-1,25%

Nota: La columna “Total sin selección de atributos”, corresponde a la columna calculada previamente “Total” de la Tabla 5 - Detalle del nivel de exactitud inicial de los algoritmos de clasificación, en la que no se realizó una selección de atributos previa. El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Tal y como se observa en la tabla 12, la selección de atributos no mejora el nivel de exactitud, ya que empeora de media un 1,39% el volumen de errores (falsos positivos y falsos negativos). No obstante, indicar que el volumen de errores no es significativamente superior, por lo que en caso de identificar modelos de predicción superiores mediante la aplicación de otras técnicas como la eliminación de outliers, normalización de variables, o segmentación previa y división de la población por modelos-segmentación, es un nivel de error que puede ser valorado como asumible, en función de si se desea disponer de un modelo preciso o de un modelo eficiente, para lo que es necesario ponderar otros aspectos como el riesgo en la identificación de falsos negativos, o el volumen de información gestionada y los tiempos de respuesta del modelo.

5.6 Tratamiento de los atributos: missing, outliers, normalización, balanceo de datos

El presente apartado tiene como objetivo aplicar técnicas para intentar mejorar la calidad de la información del dataset con la finalidad de que los algoritmos de clasificación evaluados inicialmente puedan obtener un nivel de exactitud superior al identificado en la Tabla 5 - Detalle del nivel de exactitud inicial de los algoritmos de clasificación. Para ello, se han aplicado las siguientes técnicas:

- Identificación de campos nulos
- Identificación de valores anómalos – outliers
- Normalización de variables.
- Generación de clases sintéticas con el algoritmo SMOTE (Chawla, Bowyer, Hall and Kegelmeyer, 2002).

5.6.1 Identificación de campos nulos

Tras la revisión de los valores de los campos, detallada en el apartado 5.3 se pude concluir que los campos no tienen valores nulos.

5.6.2 Identificación de valores anómalos – outliers

Para eliminar los outliers o valores fuera de rango, he utilizado la función `IntercuartileRange`, que determina los valores máximos y mínimos a eliminar según la parametrización dada al algoritmo. La eliminación de outliers supone eliminar aquellos valores que se encuentran muy lejanos a la media del grupo, por ejemplo, tres desviaciones típicas, suponiendo que los datos se distribuyan normalmente. En general, en el apartado 5.1, podemos ver que en general los datos tienden a estar concentrados para los atributos no discretos. El objetivo es determinar la influencia de este tipo de valores en los algoritmos de predicción.

Al aplicar la citada función, se crean dos nuevos atributos:

- Outlier. Toma los valores: yes, no. Indica si el valor se considera fuera de rango parametrizado. El resultado es que existen 4.136 instancias con valores fuera de rango, que son, a modo resumen, los que se encuentran en el primer y cuarto cuartil de acuerdo a una serie de factores.

- ExtremeValue. Toma los valores: yes, no. Indica si la instancia contiene un valor extremo. El resultado es que existen 3.596 instancias con valores fuera de rango.

Posteriormente, para eliminar las instancias que contengan valores fuera de rango, utilizo la función de instancia `RemoveWithValues`, señalando el campo Outlier con valor 'yes'.

Una vez eliminadas las instancias, el estado del dataset es el siguiente:

- Instancias eliminadas: 6.374
- Instancias finales que permanecen, una vez desechadas las anteriores: 21.696
- Instancias con impagos finales: 5.256 – 24,23%

Una vez eliminados, se ejecutan los algoritmos más rápidos y con mejor nivel de exactitud, de acuerdo a lo indicado previamente en el apartado "5.2 Identificación inicial de algoritmos de predicción" obteniendo el siguiente nivel de exactitud (ver Tabla 13):

Tabla 13 – Comparativa de la exactitud de los algoritmos de clasificación con y sin eliminación de outliers

	FP	FN	Total con eliminación outliers	FP %	FN %	Total con eliminación outliers %	Total sin eliminación outliers %	Diferencia %
Logistic	721	3.679	4.400	3,32%	16,96%	20,28%	18,97%	-1,31%
MultilayerPerceptron	866	3.505	4.371	3,99%	16,16%	20,15%	18,17%	-1,98%
DecisionTable	909	3.296	4.205	4,19%	15,19%	19,38%	18,05%	-1,33%
OneR	717	3.500	4.217	3,30%	16,13%	19,44%	18,08%	-1,36%
PART	1.226	3.227	4.453	5,65%	14,87%	20,52%	18,63%	-1,89%
DecisionStump	2.451	2.468	4.919	11,30%	11,38%	22,67%	18,04%	-4,63%
HoeffdingTree	775	3.443	4.218	3,57%	15,87%	19,44%	18,50%	-0,94%
RandomForest	1.135	3.179	4.314	5,23%	14,65%	19,88%	18,16%	-1,72%
REPTree	1.085	3.259	4.344	5,00%	15,02%	20,02%	18,73%	-1,29%
LMT	845	3.363	4.208	3,89%	15,50%	19,40%	17,91%	-1,49%
AdaBoostM1	1.763	2.971	4.734	8,13%	13,69%	21,82%	18,19%	-3,63%
AttributeSelectedClassifier	741	3.452	4.193	3,42%	15,91%	19,33%	17,82%	-1,51%
JRIP	981	3.237	4.218	4,52%	14,92%	19,44%	18,17%	-1,27%

Nota: La columna "Total sin eliminación outliers", corresponde a la columna calculada previamente "Total" de la Tabla 5 - Detalle del nivel de exactitud inicial de los algoritmos de clasificación, en la que no se realizó una selección de atributos previa. El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Tal y como se observa en la tabla 13, la eliminación de los outliers o valores fuera de rango no mejora el nivel de exactitud, ya que empeora de media un 1,87% el volumen de errores

(falsos positivos y falsos negativos). Por tanto, la eliminación de valores cuya desviación típica sea excesiva no afecta al nivel de exactitud de los algoritmos, sino que por contra, disminuye el número de instancias de entrenamiento por lo que empeora los resultados. Asimismo indicar que el número de falsos negativos es muy elevado en el porcentaje de exactitud, lo que implica que el modelo no es bueno.

5.6.3 Normalización de variables

El proceso de normalización de variables es relevante, ya que hay algoritmos, como las redes neuronales, que son sensibles a las variables no normalizadas. El proceso de normalización permite establecer escalas iguales para todas las variables, de forma que la diferencias de valores para el mismo atributo entre dos instancias, puede ser calculada dentro de un rango definido, que es idéntico para todos los atributos numéricos. Para poder normalizar las variables, se utiliza la función `Normalize` estableciendo, en este caso, una escala de 1 a 10.

Una vez normalizado, se ejecutan los algoritmos más rápidos y con mejor nivel de exactitud, de acuerdo a lo indicado previamente en el apartado "5.2 Identificación inicial de algoritmos de predicción" obteniendo el siguiente nivel de exactitud (ver Tabla 14):

Tabla 14 – Comparativa de la exactitud de los algoritmos de clasificación con y sin atributos normalizados

	FP	FN	Total con atributos normalizados	FP %	FN %	Total con atributos normalizados %	Total sin atributos normalizados %	Diferencia %
Logistic	675	4.647	5.322	2,40%	16,56%	18,96%	18,97%	0,01%
MultilayerPerceptron	-	-	-	-	-	-	18,17%	-
DecisionTable	1.153	3.953	5.106	4,11%	14,08%	18,19%	18,05%	-0,14%
OneR	926	4.181	5.107	3,30%	14,89%	18,19%	18,08%	-0,11%
PART	1.233	4.033	5.266	4,39%	14,37%	18,76%	18,63%	-0,13%
DecisionStump	2.969	3.020	5.989	10,58%	10,76%	21,34%	18,04%	-3,30%
HoeffdingTree	1.137	4.088	5.225	4,05%	14,56%	18,61%	18,50%	-0,11%
RandomForest	1.312	3.874	5.186	4,67%	13,80%	18,48%	18,16%	-0,32%
REPTree	1.266	3.943	5.209	4,51%	14,05%	18,56%	18,73%	0,17%
LMT	1.096	3.961	5.057	3,90%	14,11%	18,02%	17,91%	-0,11%
AdaBoostM1	1.499	3.948	5.447	5,34%	14,06%	19,41%	18,19%	-1,22%
AttributeSelectedClassifier	1.003	4.063	5.066	3,57%	14,47%	18,05%	17,82%	-0,23%
JRIP	1.216	3.857	5.073	4,33%	13,74%	18,07%	18,17%	0,10%

Nota: La columna “Total sin atributos normalizados”, corresponde a la columna calculada previamente “Total” de la Tabla 5 - Detalle del nivel de exactitud inicial de los algoritmos de clasificación, en la que no se realizó una selección de atributos previa. El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Tal y como se observa en la tabla 14, el uso de atributos normalizados no mejora el nivel de exactitud, salvo excepciones puntuales en las que sí mejora hasta un 0,17%, ya que, en términos generales, empeora de media un 0,45% el volumen de errores (falsos positivos y falsos negativos). Los algoritmos que no tienen resultados, se debe a que en la ejecución inicial ya se realiza desde Weka la normalización previa de los datos. Asimismo indicar que el número de falsos negativos es muy elevado en el porcentaje de exactitud, lo que implica que el modelo no es bueno.

5.6.4 Generación de casos de impago sintéticos con SMOTE

Un problema habitual en los procesos de estimación de modelos, es la falta o el exceso de casos etiquetados para una de las clases que se pretende predecir. Es por ello que se han desarrollado técnicas para que los algoritmos puedan realizar estimaciones en ambos casos. Algunas de ellas son las siguientes:

- *Undersampling*. Eliminación de instancias en las que la clase es negativa, en nuestro caso clientes que sí han pagado, para equilibrar el conjunto total de forma que aumente el porcentaje de clases positivas (clientes que no han pagado) respecto a la población total que se ha reducido.
- *Oversampling*. Eliminación de instancias en las que la clase es positiva, en nuestro caso clientes que no han pagado, para equilibrar el conjunto total de forma que disminuya el porcentaje de clases positivas (clientes que no han pagado) respecto a la población total que se ha reducido.
- *SMOTE*. Generación de clases sintéticas positivas, aplicada generalmente en el caso de que exista undersampling, de forma que no sea necesario eliminar instancias para equilibrar el conjunto total para que aumente el porcentaje de clases positivas (clientes que no han pagado). El algoritmo que genera clases sintéticas se denomina SMOTE - Synthetic Minority Over-sampling Technique. (Chawla, Bowyer, Hall and Kegelmeyer, 2002).

En el presente trabajo, utilizaremos SMOTE por ser un método contrastado en la literatura para generar instancias con clases de impago (default payment next month = 1) sintéticamente. Se utiliza principalmente cuando la clase del dataset no está balanceado, y

presenta proporciones del tipo 98% de pago y 2% de impago, con la finalidad de ayudar a los algoritmos en el proceso de entrenamiento generando de forma sintética nuevas instancias de la clase minoritaria de impago. No obstante, el dataset analizado no presenta falta de balanceo, ya que posee aproximadamente (depende de si estimamos el dataset completo, o los datasets de entrenamiento y prueba) un 22% de instancias con impago en la clase. Aun así, se ha ejecutado el algoritmo SMOTE para verificar si la generación de casos artificiales ayuda en el proceso de entrenamiento.

Una vez generadas las instancias sintéticas, se ejecutan los algoritmos más rápidos y con mejor nivel de exactitud, de acuerdo a lo indicado previamente en el apartado 5.2 obteniendo el siguiente nivel de exactitud (ver Tabla 15):

Tabla 15 – Comparativa de la exactitud de los algoritmos de clasificación con y sin instancias sintéticas generadas con SMOTE

	FP	FN	Total con instancias sintéticas impago	FP %	FN %	Total con instancias sintéticas impago %	Total sin instancias sintéticas impago %	Diferencia %
Logistic	1.188	4.077	5.265	5,75%	19,72%	25,47%	18,97%	-6,50%
MultilayerPerceptron	1.232	2.640	3.872	5,96%	12,77%	18,73%	18,17%	-0,56%
DecisionTable	3.078	2.640	5.718	14,89%	12,77%	27,66%	18,05%	-9,61%
OneR	17	4.081	4.098	0,08%	19,74%	19,83%	18,08%	-1,75%
PART	904	2.788	3.692	4,37%	13,49%	17,86%	18,63%	0,77%
DecisionStump	1.807	3.078	4.885	8,74%	14,89%	23,63%	18,04%	-5,59%
HoeffdingTree	1.590	3.263	4.853	7,69%	15,79%	23,48%	18,50%	-4,98%
RandomForest	960	2.358	3.318	4,64%	11,41%	16,05%	18,16%	2,11%
REPTree	1.058	2.842	3.900	5,12%	13,75%	18,87%	18,73%	-0,14%
LMT	1.112	2.732	3.844	5,38%	13,22%	18,60%	17,91%	-0,69%
AdaBoostM1	1.807	2.615	4.422	8,74%	12,65%	21,39%	18,19%	-3,20%
AttributeSelectedClassifier	753	2.701	3.454	3,64%	13,07%	16,71%	17,82%	1,11%
JRIP	733	3.048	3.781	3,55%	14,75%	18,29%	18,17%	-0,12%

Nota: La columna “Total sin instancias sintéticas impago”, corresponde a la columna calculada previamente “Total” de la Tabla 5 - Detalle del nivel de exactitud inicial de los algoritmos de clasificación, en la que no se realizó una selección de atributos previa. El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Tal y como se observa en la tabla 15, el uso de instancias sintéticas de la clase minoritaria generadas a través del algoritmo SMOTE no mejora el nivel de exactitud, salvo excepciones puntuales en las que sí mejora hasta un 2,11%, ya que, en términos generales, empeora de

media un 2,24% el volumen de errores (falsos positivos y falsos negativos). A partir de los resultados obtenidos, se puede concluir que en este caso la generación de instancias y sintéticas no mejora el nivel de exactitud de los algoritmos.

No obstante, para los dos algoritmos en los que sí mejora el nivel de exactitud, se ha reevaluado el modelo con el dataset de prueba, obteniéndose los siguientes resultados (ver Tabla 16):

Tabla 16 – Comparativa de la exactitud de los algoritmos de clasificación, reevaluando los modelos mejorados con SMOTE sobre el dataset de prueba

	Nuevo modelo generado con las instancias sintéticas de entrenamiento con SMOTE, utilizando el dataset de prueba							
	FP	FN	Total con instancias sintéticas impago	FP %	FN %	Total sin instancias sintéticas impago %	Total inicial %	Diferencia %
RandomForest	616	1.447	2.063	5,49%	12,89%	18,37%	18,16%	-0,21%
AttributeSelectedClassifier	520	1.487	2.007	4,63%	13,24%	17,87%	17,82%	-0,05%

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Tal y como se observa en la tabla 16, la aplicación de los modelos obtenidos, mediante el dataset de entrenamiento al que se le ha aplicado el algoritmo SMOTE para generar clases de impago sintéticas con el fin de mejorar el modelo de estimación, sobre dataset de pruebas sin clases sintéticas, no mejora el nivel de predicción con datos reales, ya que en ambos casos empeora el nivel de exactitud.

5.7 Segmentación de clientes.

El presente apartado tiene como objetivo, realizar una segmentación de clientes, con la finalidad de establecer grupos homogéneos que permitan generar modelos específicos para cada segmento de forma que éstos sean más precisos, al centrarse en poblaciones más concretas.

Para determinar los segmentos, se han ejecutado varios algoritmos de clustering:

- Canopy. Determina el número de clusters de forma autónoma.
- FarthestFirst. No determina el número de clusters de forma autónoma. Se ha programado para crear 5 clusters.

- FilteredCluster. No determina el número de clusters de forma autónoma. Se ha programado para crear 5 clusters ya que se basa en KMeans.
- MakeDensityBasedCluster. No determina el número de clusters de forma autónoma. Se ha programado para crear 5 clusters ya que se basa en KMeans.
- SimpleKMeans. No determina el número de clusters de forma autónoma. Se ha programado para crear 5 clusters.
- EM. Determina el número de clusters de forma autónoma.

Los algoritmos se han ejecutado utilizando 5 clusters inicialmente, ya que se ha estimado que con aproximadamente 30.000 instancias, disponer de 5 clusters de 6.000 instancias cada uno permite por una parte, disponer de instancias suficientes en cada cluster, y por otra parte, en caso de ser necesarios subdividirlos, se dispondría de dos cluster de 3.000 instancias para realizar nueves estimaciones con suficientes instancias.

El resultado de los algoritmos que establecen el número de clusters de forma autónoma es el siguiente:

- Canopy. 12 clusters.
- EM. 6 clusters.

Todos los algoritmos han sido ejecutados con la parametrización por defecto, eliminando la variable dependiente del dataset.

Se ha analizado la distribución de las instancias de cada cluster con la finalidad de determinar la distribución más homogénea de acuerdo a los diferentes algoritmos. En la siguiente tabla 17 se muestran las primeras instancias del dataset, a modo de ejemplo,

incluyendo a la derecha los resultados de agrupación de los algoritmos de clustering:

Tabla 17 – Muestra de la distribución de las instancias de cada tipo de cluster

Nº instancia	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default payment next	SIMPLE KMEAN	CANOPY	EM	DENSITY BASED	FURTHEST FIRST	FILTERED
0	20000	2	2	1	24	2	2	-1	-1	-2	-2	3913	3102	689	0	0	0	0	689	0	0	0	0	1	2	2	7	2	0	2
1	120000	2	2	2	26	-1	2	0	0	0	2	2682	1725	2682	3272	3455	3261	0	1000	1000	1000	0	2000	1	2	2	3	2	0	2
2	90000	2	2	2	34	0	0	0	0	0	0	29239	14027	13559	14331	14948	15549	1518	1500	1000	1000	1000	5000	0	0	1	6	0	0	0
3	50000	2	2	1	37	0	0	0	0	0	0	46990	48233	49291	28314	28959	29547	2000	2019	1200	1100	1069	1000	0	0	1	6	0	0	0
4	50000	1	2	1	57	-1	0	-1	0	0	0	8617	5670	35835	20940	19146	19131	2000	36681	10000	9000	689	679	0	4	0	9	4	0	4
5	50000	1	1	2	37	0	0	0	0	0	0	64400	57069	57608	19394	19619	20024	2500	1815	657	1000	1000	800	0	1	0	6	1	0	1
6	500000	1	1	2	29	0	0	0	0	0	0	367965	412023	445007	542653	483003	473944	55000	40000	38000	20239	13750	13770	0	1	7	4	4	4	1
7	100000	2	2	2	23	0	-1	-1	0	0	-1	11876	380	601	221	-159	567	380	601	0	581	1687	1542	0	0	1	7	0	0	0
8	140000	2	3	1	28	0	0	2	0	0	0	11285	14096	12108	12211	11793	3719	3329	0	432	1000	1000	1000	0	0	1	3	0	0	0
9	20000	1	3	2	35	-2	-2	-2	-2	-1	-1	0	0	0	0	13007	13912	0	0	0	13007	1122	0	0	1	0	7	1	0	1
10	200000	2	3	2	34	0	0	2	0	0	-1	11073	9787	5535	2513	1828	3731	2306	12	50	300	3738	66	0	0	1	7	0	0	0
11	260000	2	1	2	51	-1	-1	-1	-1	-1	2	12261	21670	9966	8517	22287	13668	21818	9966	8583	22301	0	3640	0	0	1	9	0	0	0
12	630000	2	2	2	41	-1	0	-1	-1	-1	-1	12137	6500	6500	6500	6500	2870	1000	6500	6500	6500	2870	0	0	0	1	0	0	0	0
13	70000	1	2	2	30	1	2	2	0	0	2	65802	67369	65701	66782	36137	36894	3200	0	3000	3000	1500	0	1	3	3	3	3	4	3
14	250000	1	1	2	29	0	0	0	0	0	0	70887	67060	63561	59696	56875	55512	3000	3000	3000	3000	3000	3000	0	1	0	1	1	0	1
15	50000	2	3	3	23	1	2	0	0	0	0	50614	29173	28116	28771	29531	30211	0	1500	1100	1200	1300	1100	0	0	1	3	0	0	0
16	20000	1	1	2	24	0	0	2	2	2	2	15376	18010	17428	18338	17905	19104	3200	0	1500	0	1650	0	1	3	3	3	3	2	3
17	320000	1	1	1	49	0	0	0	-1	-1	-1	253286	246536	194663	70074	5856	195599	10358	10000	75940	20000	195599	50000	0	4	0	4	4	0	4
18	360000	2	1	1	49	1	-2	-2	-2	-2	-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
19	180000	2	1	2	29	1	-2	-2	-2	-2	-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	7	0	0	0
20	130000	2	3	2	39	0	0	0	0	0	-1	38358	27688	24489	20616	11802	930	3000	1537	1000	2000	930	33764	0	0	1	2	0	0	0
21	120000	2	2	1	39	-1	-1	-1	-1	-1	-1	316	316	316	0	632	316	316	316	0	632	316	0	1	2	2	0	2	0	2

Como resultado, se ha determinado que la distribución más homogénea, debido en parte a que 2 algoritmos se basan en KMeans, además del propio algoritmo SimpleKMeans, es la distribución de clusters del algoritmo KMeans.

Posteriormente, se ha dividido el dataset inicial en los cinco clusters determinados por el algoritmo KMeans, y se les ha añadido la clase, ya que para la generación del cluster no se había incluido la clase (en caso de añadirse la clase, el algoritmo encuentra similitud entre todos los elementos de la misma clase), y se han realizado los análisis con la finalidad de determinar si, pese a tener menor número de ejemplos de entrenamiento, al ser más homogéneos se mejora el nivel de exactitud de los algoritmos.

La distribución de las instancias en los cluster ha sido la siguiente (ver Tabla 18):

Tabla 18 – Distribución de las instancias por algoritmo KMeans en los cluster 0, 1, 2, 3, y 4

	Nº registros	Clase 0	Clase 1	Clase 0 %	Clase 1 %
Cluster0	2.876	2.117	759	73,61%	35,85%
Cluster1	4.963	3.957	1.006	79,73%	25,42%
Cluster2	4.369	3.403	966	77,89%	28,39%
Cluster3	868	664	204	76,50%	30,72%
Cluster4	3.766	2.873	893	76,29%	31,08%
Total	**16.842**	**13.014**	**3.828**	**77,27%**	**29,41%**

En la tabla 18 se muestra que la distribución de la clase 1 – Impago, permanece homogénea entre todos los cluster con un porcentaje suficiente como para no considerarse desbalanceados. Por otra parte, el cluster 3 presenta un número de instancias reducido, lo que implica que los resultados obtenidos de su entrenamiento pueden no ser representativos de la realidad.

A continuación, se muestran los resultados obtenidos para los cinco clusters, ejecutando los algoritmos más rápidos y con mejor nivel de exactitud, de acuerdo a lo indicado previamente en el apartado 5.2 obteniendo el siguiente nivel de exactitud (ver Tablas 19, 20, 21, 22 y 23):

Tabla 19 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 0

Cluster 0	FP	FN	Total con segmentación	FP %	FN %	Total con segmentación %	Total sin segmentación %	Diferencia %
Logistic	112	510	622	3,89%	17,73%	21,63%	18,97%	-2,66%
MultilayerPerceptron	184	471	655	6,40%	16,38%	22,77%	18,17%	-4,60%
DecisionTable	464	128	592	16,13%	4,45%	20,58%	18,05%	-2,53%
OneR	95	504	599	3,30%	17,52%	20,83%	18,08%	-2,75%
PART	198	435	633	6,88%	15,13%	22,01%	18,63%	-3,38%
DecisionStump	299	355	654	10,40%	12,34%	22,74%	18,04%	-4,70%
HoeffdingTree	1	759	760	0,03%	26,39%	26,43%	18,50%	-7,93%
RandomForest	159	447	606	5,53%	15,54%	21,07%	18,16%	-2,91%
REPTree	167	474	641	5,81%	16,48%	22,29%	18,73%	-3,56%
LMT	100	513	613	3,48%	17,84%	21,31%	17,91%	-3,40%
AdaBoostM1	239	414	653	8,31%	14,39%	22,71%	18,19%	-4,52%
AttributeSelectedClassifier	139	458	597	4,83%	15,92%	20,76%	17,82%	-2,94%
JRIP	139	468	607	4,83%	16,27%	21,11%	18,17%	-2,94%

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Tabla 20 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 1

Cluster 1	FP	FN	Total con segmentación	FP %	FN %	Total con segmentación %	Total sin segmentación %	Diferencia %
Logistic	101	793	894	2,04%	15,98%	18,01%	18,97%	0,96%
MultilayerPerceptron	192	681	873	3,87%	13,72%	17,59%	18,17%	0,58%
DecisionTable	169	665	834	3,41%	13,40%	16,80%	18,05%	1,25%
OneR	136	689	825	2,74%	13,88%	16,62%	18,08%	1,46%
PART	152	707	859	3,06%	14,25%	17,31%	18,63%	1,32%
DecisionStump	136	689	825	2,74%	13,88%	16,62%	18,04%	1,42%

Cluster 1	FP	FN	Total con segmentación	FP %	FN %	Total con segmentación %	Total sin segmentación %	Diferencia %
HoeffdingTree	126	720	846	2,54%	14,51%	17,05%	18,50%	1,45%
RandomForest	193	656	849	3,89%	13,22%	17,11%	18,16%	1,05%
REPTree	177	671	848	3,57%	13,52%	17,09%	18,73%	1,64%
LMT	141	696	837	2,84%	14,02%	16,86%	17,91%	1,05%
AdaBoostM1	136	693	829	2,74%	13,96%	16,70%	18,19%	1,49%
AttributeSelectedClassifier	143	693	836	2,88%	13,96%	16,84%	17,82%	0,98%
JRIP	178	660	838	3,59%	13,30%	16,88%	18,17%	1,29%

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Tabla 21 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 2

Cluster 2	FP	FN	Total con segmentación	FP %	FN %	Total con segmentación %	Total sin segmentación %	Diferencia %
Logistic	116	716	832	2,66%	16,39%	19,04%	18,97%	-0,07%
MultilayerPerceptron	202	637	839	4,62%	14,58%	19,20%	18,17%	-1,03%
DecisionTable	139	661	800	3,18%	15,13%	18,31%	18,05%	-0,26%
OneR	120	665	785	2,75%	15,22%	17,97%	18,08%	0,11%
PART	128	686	814	2,93%	15,70%	18,63%	18,63%	0,00%
DecisionStump	120	663	783	2,75%	15,18%	17,92%	18,04%	0,12%
HoeffdingTree	122	663	785	2,79%	15,18%	17,97%	18,50%	0,53%
RandomForest	210	628	838	4,81%	14,37%	19,18%	18,16%	-1,02%
REPTree	159	663	822	3,64%	15,18%	18,81%	18,73%	-0,08%
LMT	141	649	790	3,23%	14,85%	18,08%	17,91%	-0,17%
AdaBoostM1	120	663	783	2,75%	15,18%	17,92%	18,19%	0,27%
AttributeSelectedClassifier	140	641	781	3,20%	14,67%	17,88%	17,82%	-0,06%
JRIP	167	641	808	3,82%	14,67%	18,49%	18,17%	-0,32%

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Tabla 22 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 3

Cluster 3	FP	FN	Total con segmentación	FP %	FN %	Total con segmentación %	Total sin segmentación %	Diferencia %
Logistic	46	105	151	5,30%	12,10%	17,40%	18,97%	1,57%
MultilayerPerceptron	69	104	173	7,95%	11,98%	19,93%	18,17%	-1,76%
DecisionTable	57	93	150	6,57%	10,71%	17,28%	18,05%	0,77%

Cluster 3	FP	FN	Total con segmentación	FP %	FN %	Total con segmentación %	Total sin segmentación %	Diferencia %
OneR	48	96	144	5,53%	11,06%	16,59%	18,08%	1,49%
PART	71	105	176	8,18%	12,10%	20,28%	18,63%	-1,65%
DecisionStump	75	81	156	8,64%	9,33%	17,97%	18,04%	0,07%
HoeffdingTree	45	157	202	5,18%	18,09%	23,27%	18,50%	-4,77%
RandomForest	53	102	155	6,11%	11,75%	17,86%	18,16%	0,30%
REPTree	59	94	153	6,80%	10,83%	17,63%	18,73%	1,10%
LMT	47	104	151	5,41%	11,98%	17,40%	17,91%	0,51%
AdaBoostM1	67	99	166	7,72%	11,41%	19,12%	18,19%	-0,93%
AttributeSelectedClassifier	49	96	145	5,65%	11,06%	16,71%	17,82%	1,11%
JRIP	70	84	154	8,06%	9,68%	17,74%	18,17%	0,43%

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Tabla 23 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 4

Cluster 4	FP	FN	Total con segmentación	FP %	FN %	Total con segmentación %	Total sin segmentación %	Diferencia %
Logistic	116	669	785	3,08%	17,76%	20,84%	18,97%	-1,87%
MultilayerPerceptron	220	540	760	5,84%	14,34%	20,18%	18,17%	-2,01%
DecisionTable	202	548	750	5,36%	14,55%	19,92%	18,05%	-1,87%
OneR	142	602	744	3,77%	15,99%	19,76%	18,08%	-1,68%
PART	201	557	758	5,34%	14,79%	20,13%	18,63%	-1,50%
DecisionStump	419	413	832	11,13%	10,97%	22,09%	18,04%	-4,05%
HoeffdingTree	-	892	892	0,00%	23,69%	23,69%	18,50%	-5,19%
RandomForest	216	560	776	5,74%	14,87%	20,61%	18,16%	-2,45%
REPTree	215	554	769	5,71%	14,71%	20,42%	18,73%	-1,69%
LMT	200	531	731	5,31%	14,10%	19,41%	17,91%	-1,50%
AdaBoostM1	328	499	827	8,71%	13,25%	21,96%	18,19%	-3,77%
AttributeSelectedClassifier	220	533	753	5,84%	14,15%	19,99%	17,82%	-2,17%
JRIP	236	520	756	6,27%	13,81%	20,07%	18,17%	-1,90%

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Como resumen del resultado de las cuatro tablas 19, 20, 21, 22 y 23, se obtiene la siguiente tabla (ver Tabla 24):

Tabla 24 – Resumen de los resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 0, 1, 2, 3, y 4

	Diferencia entre algoritmo con y sin segmentación				
	Cluster0 %	Cluster1 %	Cluster2 %	Cluster3 %	Cluster4 %
Logistic	-2,66%	0,96%	-0,07%	1,57%	-1,87%
MultilayerPerceptron	-4,60%	0,58%	-1,03%	-1,76%	-2,01%
DecisionTable	-2,53%	1,25%	-0,26%	0,77%	-1,87%
OneR	-2,75%	1,46%	0,11%	1,49%	-1,68%
PART	-3,38%	1,32%	0,00%	-1,65%	-1,50%
DecisionStump	-4,70%	1,42%	0,12%	0,07%	-4,05%
HoeffdingTree	-7,93%	1,45%	0,53%	-4,77%	-5,19%
RandomForest	-2,91%	1,05%	-1,02%	0,30%	-2,45%
REPTree	-3,56%	1,64%	-0,08%	1,10%	-1,69%
LMT	-3,40%	1,05%	-0,17%	0,51%	-1,50%
AdaBoostM1	-4,52%	1,49%	0,27%	-0,93%	-3,77%
AttributeSelectedClassifier	-2,94%	0,98%	-0,06%	1,11%	-2,17%
JRIP	-2,94%	1,29%	-0,32%	0,43%	-1,90%

Como se observa en la tabla 24, los cluster 1, 2 y 3 sí mejoran el nivel de exactitud de los algoritmos, en torno al 1%, si bien los cluster 0 y 4 no mejoran en ninguno de los algoritmos la exactitud, empeorando en todos ellos. Existe la posibilidad, de que los cluster 0 y 4 necesiten segmentarse de nuevo en grupos más homogéneos, si bien, dicha división implicaría que el número de instancias de cada segmento sería demasiado pequeño para poder entrenarlo adecuadamente. No obstante, dichos cluster poseen 2.876 y 3.766 instancias respectivamente, pudiendo dividirse en dos cluster de menor tamaño, ya que el cluster 3 aun teniendo 868 instancias, sí ha mejorado el nivel de clasificación.

Por otra parte, en la siguiente tabla 25 se muestran los resultados de la exactitud por algoritmo independientemente del cluster, identificando los niveles máximos, mínimos y medios en los niveles de exactitud:

Tabla 25 – Resumen de los resultados por algoritmo de clasificación con segmentación de clientes

	Diferencia entre algoritmo con y sin segmentación		
	Máximo %	Mínimo %	Media %
Logistic	1,57%	-2,66%	-0,41%
MultilayerPerceptron	0,58%	-4,60%	-1,76%

	Diferencia entre algoritmo con y sin segmentación		
	Máximo %	**Mínimo %**	**Media %**
DecisionTable	1,25%	-2,53%	-0,53%
OneR	1,49%	-2,75%	-0,27%
PART	1,32%	-3,38%	-1,04%
DecisionStump	1,42%	-4,70%	-1,43%
HoeffdingTree	1,45%	-7,93%	-3,18%
RandomForest	1,05%	-2,91%	-1,01%
REPTree	1,64%	-3,56%	-0,52%
LMT	1,05%	-3,40%	-0,70%
AdaBoostM1	1,49%	-4,52%	-1,49%
AttributeSelectedClassifier	1,11%	-2,94%	-0,62%
JRIP	1,29%	-2,94%	-0,69%

En la tabla 25 se observa, que ningún algoritmo tiene un promedio de exactitud positivo, lo que implica que en conjunto, no mejora la exactitud inicial obtenida tras el proceso de segmentación de clientes. No obstante, también se observa que existen mejoras en el proceso de segmentación por cluster. En la siguiente tabla 26, se muestra el nivel de exactitud de respecto a la ejecución inicial distribuida por cluster:

Tabla 26 – Resumen de los resultados de exactitud de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 0, 1, 2, 3, y 4

Diferencia en la exactitud	Cluster0 %	Cluster1 %	Cluster2 %	Cluster3 %	Cluster4 %
Mejor aumento	-2,53%	1,64%	0,53%	1,57%	-1,50%
Peor aumento	-7,93%	0,58%	-1,03%	-4,77%	-5,19%
Media aumento	-3,75%	1,22%	-0,15%	-0,13%	-2,43%

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Se puede observar que el cluster 1 aumenta el nivel de exactitud con independencia de los algoritmos utilizados.

Con la finalidad de determinar si los cluster 0 y 4 que presentan peores niveles de clasificación pueden mejorar sus niveles de clasificación realizando una nueva clasificación de dichos clusters, se ha realizado una nueva subdivisión en dos sub-clusters, obteniendo a partir de los dos clusters 0 y 4 iniciales, 4 nuevos subclusters.

Los resultados obtenidos se muestran en la siguiente tabla (ver Tabla 27):

Tabla 27 – Distribución de las instancias por algoritmo KMeans en los cluster 0 y 4, divididos en dos subclusters

	Nº registros	Clase 0	Clase 1	Clase 0 %	Clase 1 %
Cluster0 – Subcluster 0	2.358	1.738	620	73,71%	26,29%
Cluster0 – Subcluster 1	518	139	379	26,83%	73,17%
Cluster4 – Subcluster 0	3.198	2.410	788	75,36%	24,64%
Cluster4 – Subcluster 1	568	463	105	81,51%	18,49%
Total	**6.642**	**4.750**	**1.892**	**71,51%**	**28,49%**

En la tabla 27 se muestra que la distribución de la clase 1 – Impago permanece, en términos generales, homogénea entre todos los cluster con un porcentaje suficiente como para no estar desbalanceados. Por otra parte, los dos sub-cluster 1 presentan un número de instancias muy reducido, lo que implica que los resultados obtenidos de su entrenamiento pueden no ser representativos de la realidad.

A continuación, se muestran los resultados obtenidos para los cuatro sub-clusters, ejecutando los algoritmos más rápidos y con mejor nivel de exactitud, de acuerdo a lo indicado previamente en el apartado 5.2 obteniendo el siguiente nivel de exactitud (ver Tablas 28, 29, 30 y 31):

Tabla 28 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 0 – Subcluster 0

Cluster 0 – Subcluster 0	FP	FN	Total con segmentación	FP %	FN %	Total con segmentación %	Total sin segmentación %	Diferencia %
Logistic	86,0	446,0	532	3,65%	18,91%	22,56%	18,97%	-3,59%
MultilayerPerceptron	149,0	423,0	572	6,32%	17,94%	24,26%	18,17%	-6,09%
DecisionTable	312,0	423,0	735	13,23%	17,94%	31,17%	18,05%	-13,12%
OneR	65,0	447,0	512	2,76%	18,96%	21,71%	18,08%	-3,63%
PART	95,0	457,0	552	4,03%	19,38%	23,41%	18,63%	-4,78%
DecisionStump	242,0	312,0	554	10,26%	13,23%	23,49%	18,04%	-5,45%
HoeffdingTree	-	620,0	620	0,00%	26,29%	26,29%	18,50%	-7,79%
RandomForest	141,0	379,0	520	5,98%	16,07%	22,05%	18,16%	-3,89%
REPTree	135,0	407,0	542	5,73%	17,26%	22,99%	18,73%	-4,26%
LMT	64,0	448,0	512	2,71%	19,00%	21,71%	17,91%	-3,80%
AdaBoostM1	201,0	333,0	534	8,52%	14,12%	22,65%	18,19%	-4,46%
AttributeSelectedClassifier	89,0	418,0	507	3,77%	17,73%	21,50%	17,82%	-3,68%
JRIP	129,0	408,0	537	5,47%	17,30%	22,77%	18,17%	-4,60%

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Tabla 29 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 0 – Subcluster 1

Cluster 0 – Subcluster 1	FP	FN	Total con segmentación	FP %	FN %	Total con segmentación %	Total sin segmentación %	Diferencia %
Logistic	32	62	94	6,18%	11,97%	18,15%	18,97%	0,82%
MultilayerPerceptron	49	66	115	9,46%	12,74%	22,20%	18,17%	-4,03%
DecisionTable	57	66	123	11,00%	12,74%	23,75%	18,05%	-5,70%
OneR	30	57	87	5,79%	11,00%	16,80%	18,08%	1,28%
PART	35	71	106	6,76%	13,71%	20,46%	18,63%	-1,83%
DecisionStump	35	57	92	6,76%	11,00%	17,76%	18,04%	0,28%
HoeffdingTree	29	97	126	5,60%	18,73%	24,32%	18,50%	-5,82%
RandomForest	29	69	98	5,60%	13,32%	18,92%	18,16%	-0,76%
REPTree	44	53	97	8,49%	10,23%	18,73%	18,73%	0,00%
LMT	33	57	90	6,37%	11,00%	17,37%	17,91%	0,54%
AdaBoostM1	31	62	93	5,98%	11,97%	17,95%	18,19%	0,24%
AttributeSelectedClassifier	36	55	91	6,95%	10,62%	17,57%	17,82%	0,25%
JRIP	39	53	92	7,53%	10,23%	17,76%	18,17%	0,41%

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Tabla 30 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 4 – Subcluster 0

Cluster 4 – Subcluster 0	FP	FN	Total con segmentación	FP %	FN %	Total con segmentación %	Total sin segmentación %	Diferencia %
Logistic	112,0	577,0	689	3,50%	18,04%	21,54%	18,97%	-2,57%
MultilayerPerceptron	185,0	480,0	665	5,78%	15,01%	20,79%	18,17%	-2,62%
DecisionTable	368,0	480,0	848	11,51%	15,01%	26,52%	18,05%	-8,47%
OneR	123,0	541,0	664	3,85%	16,92%	20,76%	18,08%	-2,68%
PART	177,0	526,0	703	5,53%	16,45%	21,98%	18,63%	-3,35%
DecisionStump	382,0	368,0	750	11,94%	11,51%	23,45%	18,04%	-5,41%
HoeffdingTree	-	788,0	788	0,00%	24,64%	24,64%	18,50%	-6,14%
RandomForest	177,0	501,0	678	5,53%	15,67%	21,20%	18,16%	-3,04%
REPTree	191,0	506,0	697	5,97%	15,82%	21,79%	18,73%	-3,06%
LMT	169,0	504,0	673	5,28%	15,76%	21,04%	17,91%	-3,13%
AdaBoostM1	267,0	460,0	727	8,35%	14,38%	22,73%	18,19%	-4,54%
AttributeSelectedClassifier	183,0	487,0	670	5,72%	15,23%	20,95%	17,82%	-3,13%
JRIP	193,0	478,0	671	6,04%	14,95%	20,98%	18,17%	-2,81%

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Tabla 31 – Resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – Cluster 4 – Subcluster 1

Cluster 4 – Subcluster 1	FP	FN	Total con segmentación	FP %	FN %	Total con segmentación %	Total sin segmentación %	Diferencia %
Logistic	24	66	90	4,23%	11,62%	15,85%	18,97%	3,12%
MultilayerPerceptron	31	64	95	5,46%	11,27%	16,73%	18,17%	1,44%
DecisionTable	45	64	109	7,92%	11,27%	19,19%	18,05%	-1,14%
OneR	26	60	86	4,58%	10,56%	15,14%	18,08%	2,94%
PART	29	60	89	5,11%	10,56%	15,67%	18,63%	2,96%
DecisionStump	37	45	82	6,51%	7,92%	14,44%	18,04%	3,60%
HoeffdingTree	-	5	5	0,00%	0,88%	0,88%	18,50%	17,62%
RandomForest	28	62	90	4,93%	10,92%	15,85%	18,16%	2,31%
REPTree	36	56	92	6,34%	9,86%	16,20%	18,73%	2,53%
LMT	28	67	95	4,93%	11,80%	16,73%	17,91%	1,18%
AdaBoostM1	37	45	82	6,51%	7,92%	14,44%	18,19%	3,75%
AttributeSelectedClassifier	35	53	88	6,16%	9,33%	15,49%	17,82%	2,33%
JRIP	35	50	85	6,16%	8,80%	14,96%	18,17%	3,21%

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Como resumen del resultado de las cuatro tablas 28, 29, 30 y 31 se obtiene la siguiente tabla (ver Tabla 32):

Tabla 32 – Resumen de los resultados de la ejecución de los algoritmos de clasificación con segmentación de clientes – cluster 0 y 4, divididos en dos subclusters, identificando la diferencia entre la clasificación inicial y la clasificación final aplicando subclusters

	Cluster0 – Subcluster 0 %	Cluster0 – Subcluster 1 %	Cluster4 – Subcluster 0 %	Cluster4 – Subcluster 1 %
Logistic	-3,59%	0,82%	-2,57%	3,12%
MultilayerPerceptron	-6,09%	-4,03%	-2,62%	1,44%
DecisionTable	-13,12%	-5,70%	-8,47%	-1,14%
OneR	-3,63%	1,28%	-2,68%	2,94%
PART	-4,78%	-1,83%	-3,35%	2,96%
DecisionStump	-5,45%	0,28%	-5,41%	3,60%
HoeffdingTree	-7,79%	-5,82%	-6,14%	17,62%
RandomForest	-3,89%	-0,76%	-3,04%	2,31%
REPTree	-4,26%	0,00%	-3,06%	2,53%
LMT	-3,80%	0,54%	-3,13%	1,18%
AdaBoostM1	-4,46%	0,24%	-4,54%	3,75%
AttributeSelectedClassifier	-3,68%	0,25%	-3,13%	2,33%
JRIP	-4,60%	0,41%	-2,81%	3,21%

Como se observa en la tabla 32, los dos subclusters 0 empeoran el nivel de exactitud de los algoritmos, en torno al 4,5%. Teniendo en cuenta que entre los dos subclusters incluyen 5.556 instancias, que representan el 83,65% de los registros de los dos clusters principales, podemos concluir que aunque los dos subclusters 1 sí mejoran el nivel de exactitud en torno al 1,5%, el nuevo proceso de segmentación no mejora en su conjunto el nivel de clasificación obtenido previamente en la división de 5 clusters.

5.8 Identificación de algoritmos de mayor precisión

En el presente apartado, se van a identificar los tres algoritmos con mejor nivel de exactitud de acuerdo a la clasificación inicial.

Tabla 33 – Tres algoritmos con mayor nivel de exactitud

	FP	FN	Total	FP %	FN %	Total %	Área ROC %	Tiempo de ejecución *hh:mm:ss*
AutoWEKAClassifier	1.215	4.167	5.382	4,05%	13,89%	17,94%	74,20%	00:15:00[1]
LMT	1.138	4.236	5.374	3,79%	14,12%	17,91%	74,00%	00:23:35
AttributeSelectedClassifier	1.166	4.180	5.346	3,89%	13,93%	17,82%	69,00%	00:00:07

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

1 - El tiempo de este algoritmo no es comparable al resto, ya que supone la ejecución paralela de un conjunto de algoritmos. Para calcular el tiempo, se ha ejecutado únicamente el algoritmo MultilayerPerceptron con la parametrización identificada por el algoritmo AutoWEKAClassifier.

En la tabla 33 se muestra que el nivel de exactitud de los tres algoritmos es similar, así como el área ROC, no existiendo diferencias significativas, si bien podemos considerar que el algoritmo con mayor área ROC es el AutoWEKAClassifier.

Por otra parte, para poder analizar con solvencia los tres algoritmos, además de la exactitud, es necesario tener en cuenta el nivel de:

- Precisión, que mide la relación de positivos reales identificados por los algoritmos, en relación con la población total de positivos reales y erróneos identificados por los algoritmos:

Ecuación 4 – Precisión

$$Precisión = \frac{Nº\ de\ verdaderos\ positivos}{Nº\ de\ verdaderos\ positivos + Nº\ de\ falsos\ positivos}$$

- Sensibilidad, que mide la relación de positivos reales identificados por los algoritmos, en relación con la población total real de positivos existentes en el dataset:

Ecuación 5 – Sensibilidad

$$Sensibilidad = \frac{Nº\ de\ verdaderos\ positivos}{Nº\ de\ verdaderos\ positivos + Nº\ de\ falsos\ negativos}$$

- Así como la media armónica no ponderada de ambos denominada “F”:

Ecuación 6 – Media armónica no ponderada “F”

$$Media\ armónica\ no\ pondenrada\ "F" = \frac{2 * Sensibildiad * Precisión}{Precisión + Sensibilidad}$$

En la siguiente tabla 34 se muestran los valores obtenidos por los tres algoritmos para cada una de sus clases, así como la media de éstas:

Tabla 34 – Tres algoritmos con mayor nivel de exactitud, clasificados por nivel de precisión, sensibilidad y media armónica de ambos F no ponderada

	Precisión			Sensibilidad			F		
	Clase 0 %	Clase 1 %	Media %	Clase 0 %	Clase 1 %	Media %	Clase 0 %	Clase 1 %	Media %
AutoWEKAClassifier	84,00%	67,60%	**80,40%**	95,00%	36,40%	**82,10%**	89,20%	47,30%	**79,90%**
LMT	84,00%	68,00%	**80,50%**	95,10%	36,40%	**82,10%**	89,20%	47,40%	**80,00%**
AttributeSelectedClassifier	84,20%	67,80%	**80,50%**	95,00%	37,00%	**82,20%**	89,30%	47,90%	**80,10%**
Media	**84,07%**	**67,80%**	**80,47%**	**95,03%**	**36,60%**	**82,13%**	**89,23%**	**47,53%**	**80,00%**

Para poder analizar los datos de la tabla 34, es necesario previamente establecer las siguientes premisas:

- Como la finalidad del dataset es identificar a clientes morosos, la clase más relevante es la clase 1.
- Para poder establecer un criterio de análisis entre precisión y sensibilidad, consideramos como prioritaria la sensibilidad, ya que en el caso de análisis es más relevante identificar correctamente todas las personas morosas, que determinar si los objetos seleccionados son correctos, ya que tendríamos en cuenta únicamente los falsos positivos, pero no tendríamos en cuenta los falsos negativos, aspecto que sí contempla la sensibilidad.

Una vez establecidas estas premisas, podemos concluir que la sensibilidad media es del 36,6% en los tres algoritmos seleccionados, la cual es baja, es decir, existe una gran cantidad para la clase de impago de falsos negativos, y por tanto, clientes morosos que no son detectados. Este aspecto fue puesto de manifiesto ya al inicio del presente trabajo en el apartado 5.2. Por otra parte, la sensibilidad es prácticamente la misma en todos ellos, no existiendo variaciones significativas.

No obstante, conviene analizar los resultados con mayor detalle, ya que:

- El algoritmo AutoWEKAClassifier es un meta algoritmo que incluye la revisión de los siguientes algoritmos (ver Tabla 35):

Tabla 35 – Relación de algoritmos evaluados por el meta algoritmo AutoWEKAClassifier

Learners					
BayesNet	2	Logistic	1	REPTree*	6
DecisionStump*	0	M5P	4	SGD*	5
DecisionTable*	4	M5Rules	4	SimpleLinearRegression*	0
GaussianProcesses*	10	MultilayerPerceptron*	8	SimpleLogistic	5
IBk*	5	NaiveBayes	2	SMO	11
J48	9	NaiveBayesMultinomial	0	SMOreg*	13
JRip	4	OneR	1	VotedPerceptron	3
KStar*	3	PART	4	ZeroR*	0
LinearRegression*	3	RandomForest	7		
LMT	9	RandomTree*	11		
Ensemble Methods					
Stacking	2	Vote	2		
Meta-Methods					
LWL	5	AttributeSelectedClassifier	2	RandomSubSpace	3
AdaBoostM1	6	Bagging	4		
AdditiveRegression	4	RandomCommittee	2		
Attribute Selection Methods					
BestFirst	2	GreedyStepwise	4		

Figure 1: Learners and methods supported by Auto-WEKA 2.0, along with number of hyperparameters $|\Lambda|$. Every learner supports classification; starred learners also support regression.

"Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA" (Kotthoff, Thornton, Hoos, Hutter and Leyton-Brown, 2016)

Cabe destacar que entre los diferentes tipos de algoritmos, se encuentra el algoritmo AttributeSelectedClassifier, si bien, como resultado se ha identificado que el mejor algoritmo de clasificación es el MultilayerPerceptron. Dicho método no presenta en la tabla inicial Tabla 5 - Detalle del nivel de exactitud inicial de los algoritmos de clasificación el mismo nivel de exactitud, ya que el algoritmo AutoWEKAClassifier evalúa los metaparámetros, y los ajusta identificando el mejor nivel de exactitud.

El algoritmo MultilayerPerceptron ejecutado por Weka por defecto, presenta 12 neuronas de entrada, y dos de salida, considerando en las doce neuronas de entrada

las 23 variables del modelo, existiendo pesos que ponderan las variables en cada una de las entradas.

En este caso, no es sencillo interpretar el modelo generado ya que no se puede saber fácilmente qué variables está utilizando y cuáles no, ya que a priori utiliza todas ponderadas en los 12 nodos de entrada, y tampoco es sencillo generar manualmente un pronóstico navegando a través del árbol, ni interpretar los resultados obtenidos, si bien, sí es replicable el modelo para nuevas variables.

- El algoritmo AttributeSelectedClassifier ha sido probado como parte del algoritmo anterior en el que se han intentado ajustar dos hiperparámetros, en los que varían:
    - El evaluador: BestFirst o GreedyStepWise.
    - El método de búsqueda: CfsSubsetEval.

  De acuerdo a las especificaciones descritas en el “User Guide for Auto-WEKA version 2.6” (Kotthoff, Thornton and Hutter, 2017). Por otra parte, indicar que dicho algoritmo utiliza el clasificador J48.

Dicho algoritmo genera un árbol de 13 hojas que se muestra a continuación (ver Ilustración 8):

Ilustración 8 – Árbol de clasificación generado por el algoritmo Attribute Selected Classifier

```
PAY_0 <= 1
|   PAY_2 <= 1: 0 (24599.0/3514.0)
|   PAY_2 > 1
|   |   PAY_5 <= 0
|   |   |   PAY_2 <= 2: 0 (1484.0/533.0)
|   |   |   PAY_2 > 2
|   |   |   |   PAY_3 <= 2: 1 (89.0/38.0)
|   |   |   |   PAY_3 > 2: 0 (13.0/4.0)
|   |   PAY_5 > 0
|   |   |   PAY_5 <= 2
|   |   |   |   PAY_4 <= 2
|   |   |   |   |   PAY_0 <= 0: 0 (55.0/24.0)
|   |   |   |   |   PAY_0 > 0
|   |   |   |   |   |   PAY_3 <= 2: 1 (500.0/241.0)
|   |   |   |   |   |   PAY_3 > 2: 0 (32.0/14.0)
|   |   |   |   PAY_4 > 2: 1 (39.0/14.0)
|   |   |   PAY_5 > 2
|   |   |   |   PAY_4 <= 6: 1 (56.0/21.0)
|   |   |   |   PAY_4 > 6: 0 (3.0)
PAY_0 > 1
|   PAY_3 <= -1
|   |   PAY_2 <= -1: 0 (91.0/31.0)
|   |   PAY_2 > -1: 1 (99.0/43.0)
|   PAY_3 > -1: 1 (2940.0/850.0)
```

Dicho árbol consta de veinticinco nodos. En el proceso de selección de variables identifica como variables relevantes únicamente la capacidad de pago del cliente, es decir, seis variables, desechando el resto.

En este caso, es posible interpretar el modelo generado, así como los resultados obtenidos, ya que se puede saber qué variables está utilizando y cuáles no, así como realizar manualmente un pronóstico navegando a través del árbol, siendo replicable el modelo para nuevas variables.

- El algoritmo LMT – Logistic Model Trees no ha sido probado como parte del algoritmo anterior AutoWEKAClassifier. Dicho algoritmo genera el siguiente árbol de clasificación (ver Ilustración 9):

Ilustración 9 – Árbol de clasificación generado por el algoritmo LMT

```
PAY_0 <= 1
|   PAY_2 <= 1: LM_1:48/144 (24599)
|   PAY_2 > 1
|   |   PAY_6 <= 0: LM_2:48/192 (1625)
|   |   PAY_6 > 0
|   |   |   PAY_6 <= 2: LM_3:48/240 (588)
|   |   |   PAY_6 > 2: LM_4:48/240 (58)
PAY_0 > 1: LM_5:48/96 (3130)
```

Dicho árbol consta de nueve nodos y cinco hojas. En el proceso de selección de variables identifica como variables relevantes únicamente la capacidad de pago del cliente, en este caso tres variables, desechando el resto de variables.

En este caso, es incluso más sencillo que en el algoritmo anterior AttributeSelectedClassifier interpretar el modelo generado, así como los resultados obtenidos ya que el número de variables es menor, así como realizar manualmente un pronóstico navegando a través del árbol, siendo replicable el modelo para nuevas variables.

Finalmente, el resumen de todo lo identificado anteriormente se muestra en la siguiente tabla 36:

Tabla 36 – Tres algoritmos con mayor nivel de exactitud, clasificados por nivel de precisión, sensibilidad y media armónica de ambos F no ponderada, tiempo de ejecución, nº de variables seleccionadas y complejidad del algoritmo

Métrica	MultilayerPerceptron	LMT	AttributeSelectedClassifier – BestFirst, CfsSubsetEval, J48
Exactitud %	17,94%	17,91%	17,82%
Área ROC %	74,20%	74,00%	69,00%
Tiempo de ejecución *hh:mm:sss*	0:15:00	0:23:35	0:00:07
Precisión - Clase1 %	67,60%	68,00%	67,80%
Sensibilidad - Clase1 %	36,40%	36,40%	37,00%
F - Clase1 %	47,30%	47,40%	47,90%
Número de nodos	37	9	25
Número de hojas	2	5	13
Número de atributos utilizados	23	3	6
Número de conexiones entre nodos	338	8	24

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Como resumen del análisis realizado, a partir de la tabla 36 se puede concluir que el algoritmo más sencillo en base a su número de nodos y hojas, su elevado nivel de exactitud y de sensibilidad, el reducido número de variables utilizadas, la facilidad de interpretación de los resultados obtenidos, es el algoritmo LMT, si bien, su tiempo de ejecución es el más alto. Es necesario tener en cuenta que la realización de procesos de calibrado con un número mayor de instancias, podría dilatar el tiempo de ejecución en proporción, lo que en modelos reales con medio millón de instancias de muestra, suponiendo que el tiempo de ejecución aumentase de manera lineal, sería de 06:33:03 (hh:mm:ss). Para el algoritmo AttributeSelectedClassifier – BestFirst, CfsSubsetEval, J48, el incremento de tiempo para medio millo de instancias sería de 00:01:56. Es por ello, que teniendo en cuenta que la complejidad del algoritmo AttributeSelectedClassifier – BestFirst, CfsSubsetEval, J48 al no ser muy superior a la del algoritmo LMT, pero que en cambio el tiempo de ejecución es 202 veces más rápido, podemos concluir que el algoritmo más eficiente es el AttributeSelectedClassifier – BestFirst, CfsSubsetEval, J48.

6 Conclusiones

En relación con los estudios realizados por otros autores se puede concluir lo siguiente:

- Los análisis previos realizados por I-Cheng Yeh y Che-hui Lien en el esutdio “The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients” (Yeh and Lien, 2009) sobre el mismo dataset utilizado en el presente trabajo “Default of credit card clients Data Set” (Yeh, 2016), y en base a los algoritmos analizados: Kneighbourhood, Logistic regression, Discriminant analysis, Naive Bayesian, Neural networks y Classification tres, establecen que el mejor algoritmo de clasificación supervisada son las redes neuronales, utilizando entre otros como criterios de selección de los algoritmos el nivel de exactitud.

- Por otra parte, en el estudio “Data Mining Techniques for Credit Card Fraud Detection: Empirical Study” (Fahmi, Hamdy and Nagati, 2016), concluyen que entre los algoritmos analizados: Kneighbourhood, J48, Naive Bayes y Support Vector Machines, sobre un dataset de operaciones de fraude en tarjetas de crédito “UCSD-FICO Data Mining Contest 2009” (UCSD-FICO Data Mining Contest 2009, 2009), todas las técnicas tienen resultados similares y que por tanto no existe un algoritmo que claramente sea superior al resto en esta área de concomimiento.

- Por último, en el estudio “Comparative Evaluation of Predictive Modeling Techniques on Credit Card Data” (Singh and Aggarwal, 2011), se observa en las mediciones realizadas que entre los algoritmos analizados: Linear Discriminant Analysis, Logistic Regression, Multilayer perceptron, Decision tree, Genetic Programming, Support Vector Machines, Kernel density estimation, Kneighbourhood, sobre un dataset de operaciones de fraude en tarjetas de crédito “Statlog (Australian Credit Approval) Data Set” (Statlog (Australian Credit Approval) Data Set, s. f.), el nivel de exactitud es superior tanto en la red neuronal, como en la regresión logística.

En el presente proyecto se analizan un total de veinticuatro algoritmos de clasificación supervisada, utilizando como criterios de selección la exactitud, precisión, sensibilidad, área ROC, tiempo de ejecución, y su complejidad en cuanto al modelo generado y a la interpretación de los resultados obtenidos.

También se ha analizado la respuesta a técnicas de segmentación mediante clustering, selección de atributos, eliminación de outliers, creación de clases sintéticas con SMOTE, normalización de variables y revisión de valores ausentes, para intentar mejorar los resultados de la predicción.

Como resultado del análisis realizado se puede concluir que los tres algoritmos que mejores valores obtienen en conjunto son: MultilayerPerceptron, LMT, AttributeSelectedClassifier – BestFirst, CfsSubsetEval, J48. Finalmente, tras analizar los resultados obtenidos por los citados algoritmos, se puede concluir que el que presenta mejores características es el algoritmo de selección y clasificación de atributos AttributeSelectedClassifier - BestFirst, CfsSubsetEval, J48.

En relación a los estudios previos realizados, indicar que en el primer estudio citado, cuyos resultados se obtienen sobre el mismo data set utilizado en el presente trabajo, se identifican las redes neuronales como el mejor algoritmo de clasificación. No obstante, en el presente trabajo se documenta cómo el algoritmo el algoritmo de selección y clasificación de atributos AttributeSelectedClassifier - BestFirst, CfsSubsetEval, J48 presenta un nivel de exactitud, precisión, sensibilidad similares, pero mejora sensiblemente el tiempo de ejecución (7 segundos) así como disminuye la complejidad (genera un árbol de 25 nodos con 24 conexiones), entre otros aspectos, facilitando resultados en un tiempo reducido y siendo fáciles de interpretar, utilizando únicamente 6 atributos de los 23 iniciales, a diferencia de las redes neuronales que poseen resultados similares en relación a la

exactitud, sensibilidad y precisión, si bien utilizan todos los atributos como entrada, su tiempo de ejecución es muy elevado (15 minutos), y su complejidad (posee 37 nodos con 338 conexiones) hace difícil la interpretación de los resultados obtenidos. Los resultados de las pruebas realizadas se resumen en la siguiente tabla 37:

Tabla 37 – Comparación del algoritmo óptimo identificado en el estudio previo realizado por I-Cheng Yeh y Che-hui Lien y el obtenido en el presente trabajo

Métrica	Resultados estudio realizado por I-Cheng Yeh y Che-hui Lien	Resultados del presente trabajo
	MultilayerPerceptron	AttributeSelectedClassifier - BestFirst, CfsSubsetEval, J48
Exactitud %	17,94%	17,82%
Área ROC %	74,20%	69,00%
Tiempo de ejecución *hh:mm:ss*	0:15:00	0:00:07
Precisión - Clase1 %	67,60%	67,80%
Sensibilidad - Clase1 %	36,40%	37,00%
F - Clase1 %	47,30%	47,90%
Número de nodos	37	25
Número de hojas	2	13
Número de atributos utilizados	23	6
Número de conexiones entre nodos	338	24

El valor de la exactitud está medido de acuerdo a la Ecuación 2 – Error en la exactitud.

Es importante indicar, que en la tabla 37 se presenta el algoritmo óptimo identificado en el estudio realizado por I-Cheng Yeh y Che-hui Lien (Yeh and Lien, 2009), si bien, los resultados de la tabla 37 no se muestran en el citado estudio, habiendo sido obtenidos en el presente trabajo utilizando el mismo algoritmo y dataset (que es el utilizado para el presente trabajo) que se utilizó en su estudio.

El escaso tiempo de ejecución del modelo generado por el algoritmo de selección de atributos (AttributeSelectedClassifier) permite su calibración periódica, a partir de los datos reales en función de los cambios en las acciones de pago realizadas por los clientes. Asimismo permite la realización de procesos de calibrado con un número mayor de instancias, manteniendo ventanas de tiempo inferiores a 2 minutos para medio millón de instancias.

Por otra parte, en relación al estudio estudio “Comparative Evaluation of Predictive Modeling Techniques on Credit Card Data” (Singh and Aggarwal, 2011), indicar que entre los tres algoritmos más eficientes identificados en el presente trabajo, se encuentran los algoritmos MultilayerPerceptron y LMT, que se basan en redes neuronales así como en árboles de

regresiones logísticas. Si bien el algoritmo LMT no es exactamente el aplicado en el citado estudio, los análisis realizados en el presente trabajo permiten coincidir con los resultados obtenidos en el citado estudio, considerando la exactitud, precisión y sensibilidad.

Por último, en relación al estudio estudio “Data Mining Techniques for Credit Card Fraud Detection: Empirical Study” (Fahmi, Hamdy and Nagati, 2016) en el presente trabajo sí se identifican un algoritmo que destaca por encima del resto, tal y como ya se ha explicado con mayor detalle en los párrafos previos de las conclusiones.

7 Referencias y enlaces

1. Abbas Keramati & Niloofar Yousefi. (2011). A Proposed Classification of Data Mining Techniques in Credit Scoring. Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management, No especificado, 9.
2. Adela Ioana Tudor, Adela Bâra & Simona Vasilica Oprea. (No especificada). Comparative analysis of data mining methods for predicting credit default probabilities in a retail bank portfolio. Latest Trends in Information Technology, ISBN: 978-1-61804-134-0, 6.
3. Autor confidencial. Año publicación no especificado). Credit Approval Data Set. No especificada fecha publicación, de University of California, Irvine Sitio web: http://archive.ics.uci.edu/ml/datasets/Credit+Approval
4. Autor confidencial. (No especificado). Statlog (Australian Credit Approval) Data Set . No especificado, de University of California, Irvine Sitio web: Statlog (Australian Credit Approval) Data Set. Sitio web: http://archive.ics.uci.edu/ml/datasets/statlog+(australian+credit+approval)
5. Autor no especificado. (2009). UCSD-FICO Data Mining Contest 2009. 2009, de University of California San Diego Sitio web: https://www.cs.purdue.edu/commugrate/data/credit_card/
6. Chiharu Sano. (1992). Japanese Credit Screening Data Set. 1992-03-19, de University of California, Irvine Sitio web: http://archive.ics.uci.edu/ml/datasets/Japanese+Credit+Screening
7. Dr. Hans Hofmann. (1994). Statlog (German Credit Data) Data Set. 1994-11-17, de University of California, Irvine Sitio web: https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

8. Eibe Frank, Mark A. Hall & Ian H. Witten. (2016). The WEKA Workbench. http://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf: Morgan Kaufmann.Dataset: KDD Cup 1999.
9. I-Cheng Yeh, Che-hui Lien. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. ScienceDirect, 2473-2480, 8.
10. I-Cheng Yeh. (2016). Default of credit card clients Data Set. 2016-01-26, de University of California, Irvine Sitio web: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients#
11. Jesus Gomez &Jammy Loeur. (Año no publicado). The Default of Credit Card Clients. Fecha no publicada, de California State University Sitio web: http://athena.ecs.csus.edu/~gomezja/FinalProjectReport.pdf
12. Lars Kotthoff, Chris Thornton & Frank Hutter. (2017). User Guide for Auto-WEKA version 2.6. 2017-7-11, de cs.ubc.ca Sitio web: http://www.cs.ubc.ca/labs/beta/Projects/autoweka/manual.pdf.
13. Lars Kotthoff, Chris Thornton, Holger H. Hoos, Frank Hutter & Kevin Leyton-Brown. (2016). Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. Journal of Machine Learning Research, 17, 5.
14. Linda Delamaire, Hussein Abdou & John Pointon. (2009). Credit card fraud and detection techniques: a review. Banks and Bank Systems, Volume 4, Issue 2, 12
15. Mark Hopkins, Erik Reeber, George Forman & Jaap Suermondt. (1999). Spambase Data Set. 1999-07-01, de University of California, Irvine Sitio web: https://archive.ics.uci.edu/ml/datasets/spambase
16. Marwan Fahmi, Abeer Hamdy & Khaled Nagati. (2016). Data Mining Techniques for Credit Card Fraud Detection: Empirical Study. 2016-11-08, de British University in Egypt, Faculty of Informatics and Computer Science Sitio web: http://www.bue.edu.eg/pdfs/Research/ACE/5%20Online%20Proceeding/9%20Artificial%20Intelligence%20Techniques%20&%20Applications%20(ETI02)/Data%20Mining%20Techniques%20for%20Credit%20Card%20Fraud%20Detection%20Empirical%20Study.pdf
17. Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall & W. Philip Kegelmeyer. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357.
18. Philip K. Chan & Salvatore J. Stolfo. (1998). Toward Scalable Learning with Non-uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection. 1998, de

American Association for Artificial Intelligence Sitio web: http://www.aaai.org/Papers/KDD/1998/KDD98-026.pdf

19. Ravinder Singh & Rinkle Rani Aggarwal. (Octubre 20111). Comparative Evaluation of Predictive Modeling Techniques on Credit Card Data. International Journal of Computer Theory and Engineering, Vol. 3, No. 5, 6.
20. S. Moro, P. Cortez & P. Rita. (2014). Bank Marketing Data Set. 2012-02-14, de University of California, Irvine Sitio web: https://archive.ics.uci.edu/ml/datasets/bank+marketing
21. Tej Paul Bhatla, Vikram Prabhu & Amit Dua. (2003). Understanding Credit Card Frauds. Cards Business Review, 01, 17.
22. The UCI KDD Archive. (1999). KDD Cup 1999 Data. October 28, 1999, de Information and Computer Science University of California, Irvine. Sitio web: https://archive.ics.uci.edu/ml/machine-learning-databases/kddcup99-mld/kddcup99.html