# «COMPARACIÓN DE MÉTODOS DE DETECCIÓN DEL FRAUDE EN OPERACIONES DE BANCA MÓVIL»

by

Israel Beltrán Reyes

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

University of Huelva & International University of Andalusia

September/November 2017

# UNA COMPARACIÓN DE DIFERENTES MÉTODOS DE DETECCIÓN DEL FRAUDE EN OPERACIONES DE BANCA MÓVIL

Israel Beltrán Reyes

Máster en Economía, Finanzas y Computación

Mónica Carmona Arango
Universidad de Huelva y Universidad Internacional de Andalucía

2017

## Abstract

This paper presents a process for fraud detection in electronic banking transactions through mobile banking platforms. We use, for that purpose, a set of data mining technics that allow us the classify into genuine and dishonest transactions.

These techniques are used in a synthetic database provided by INDRA because a mutual collaboration agreement between INDRA and University of Huelva.

In order to obtain conclusions, we compare classification´s techniques designed, trained and validated, using common metrics. A set of methods are applied, ranging from basic methods as logistic regression to more powerful methods as Xgboost.

**Keywords:** Key words: Data Mining, Decision Tree, Random forest, Naive Bayes, SVM, Logistic Regresion, KNN, Xgboost.

# Resumen

Presentamos en este trabajo un procedimiento para la detección del fraude en operaciones financieras realizadas a través de plataformas de “Banca Móvil”. Se aplican para ello, un conjunto de técnicas de minería de datos que permitan la clasificación de los registros entre operaciones legítimas y fraudulentas.

Estas técnicas son aplicadas a una base de datos sintéticos proporcionada por la empresa INDRA, en virtud de un convenio de colaboración entre la empresa y la Universidad de Huelva.

Se extraen conclusiones derivadas de la comparación de los métodos clasificación diseñados, entrenados y validados, sobre el establecimiento de una métrica común. Se emplean un grupo de técnicas que van desde las más básicas como la regresión logística, a métodos más potentes como Xgboost, que serán comparados para determinar el método de clasificación más efectivo para este tipo de análisis.

**Palabras clave:** Minería de Datos, Árboles de Decisión, Naive Bayes, SVM, Regresión Logística, KNN, Xgboost.

# Agradecimientos

Quiero expresar mi agradecimiento mi tutora la profesora Mónica Carmona que se aventuró conmigo sin pensarlo, al profesor Manuel E. Gegundez, sin cuyos consejos y guía no hubiera sido posible, y a mis compañeros Pepe, Diego y David

# Tabla de Contenidos


| | |
|---|---|
| 1.- Introducción | p. 1 |
| 2.- Revisión de la bibliografía | p. 3 |
| 3.- Metodología | p. 5 |
| 4.- Resultados | p. 17 |
| 5.- Conclusiones | p. 20 |
| Bibliografía | p. 21 |

# Lista de Tablas

Tabla 1 Matriz confusión algoritmos I

Tabla 2 Matriz confusión algoritmos II

Tabla 3 Métricas algoritmos

Tabla 4 Curvas mejores algoritmos

Tabla 5 Comparación algoritmos

Tabla 6 Resultados base de datos original

Figura7 Matriz de confusión

Figura 8 Ejemplo curva AUC-ROC

# Lista de Figuras

Tabla 1 Matriz confusión algoritmos I

Tabla 2 Matriz confusión algoritmos II

Tabla 3 Métricas algoritmos

Tabla 4 Curvas mejores algoritmos

Tabla 5 Comparación algoritmos

Tabla 6 Resultados base de datos original

# 1 Introducción

Cada año se pierden, a consecuencia del fraude, miles de millones de dólares en operaciones con tarjetas de crédito. Sólo en el 2015 el fraude alcanzó 21,84 miles de millones de dólares, un 20.6% más que en el 2014 [1]. Este fraude se comete en operaciones tales como la retirada de dinero en efectivo, el pago en efectivo o a crédito a través de tarjetas, transferencias electrónicas, etc. El fraude afecta tanto a particulares como a empresas, y se comete tanto en operaciones con tarjeta como en operaciones en las que las tarjetas no están presentes. En éstas últimas no es posible comprobar, por una de las partes, si la identidad de la otra coincide con el tenedor de la tarjeta. En este sentido, cobra especial relevancia las operaciones financieras y comerciales que se realizan a través de internet o del teléfono móvil.

Las entidades bancarias y emisores de tarjetas de crédito dedican gran parte de sus recursos a reducir el impacto económico del fraude en este tipo de transacciones, incrementando las medidas de seguridad en el comercio electrónico o través del teléfono móvil. Estos intentos han ido encaminados tanto a la prevención como a la detección de prácticas fraudulentas. La detección actúa una vez que las medidas de prevención no han funcionado, tratando de minimizar el tiempo transcurrido desde que se comete el fraude hasta que se conoce.

La detección del fraude se ha convertido, por tanto, en un tópico de interés en investigaciones desarrolladas en el campo de la minería de datos. Pero al mismo tiempo que mejoran los sistemas de detección del fraude, los criminales evolucionan en la forma en la que cometen sus actividades delictivas para evitar la detección. Esto implica que será necesario seguir avanzando en la detección del fraude para adelantarse a los mecanismos de los defraudadores.

Como hemos comentado el control de la actividad fraudulenta se pueden abordar desde dos enfoques: la detección y la prevención. El objetivo de la detección es conocer la existencia de un fraude en el menor tiempo desde que se comete.

Por otro lado, existen diferentes técnicas o metodologías para la prevención de fraude en las transacciones financieras que se realizan por cualquier medio, como por ejemplo la verificación de claves de acceso a la banca por internet. Es cuando fallan las técnicas de prevención del fraude cuando la detección se convierte en el único mecanismo para minimizar las pérdidas

económicas, por ello, detectar patrones de comportamiento de fraude se convierte en un factor clave para evitar que éste se produzca.

La minería de datos permite a partir de técnicas cada vez más sofisticadas la detección del fraude en operaciones bancarias, a través de la identificación de patrones de comportamiento relacionados con esta práctica. El término minería de datos engloba todas las técnicas estadísticas y matemáticas utilizadas para detectar patrones en un conjunto de datos. La aplicación de determinadas herramientas dentro de la minería de datos a una base de registros de transacciones financieras, permite determinar la probabilidad de que una determinada operación pueda ser considerada como fraudulenta.

Técnicas estadísticas tradicionales como la aplicación de métodos lineales o la utilización de la regresión logística se han revelado como efectivos métodos en la detección del fraude en operaciones bancarias [2], pero la posibilidad de aplicar métodos más sofisticados de análisis como las aplicaciones de las redes neuronales han sido extensamente utilizados, convirtiéndose en un *hot topic* en los últimos años.

En esta dirección, el objetivo de este trabajo consiste en el diseño de diferentes algoritmos de clasificación que nos permitan identificar si un nuevo registro, es decir, una nueva operación de traspaso de dinero a través del teléfono móvil, se puede catalogar como operación legítima o fraudulenta. Así como, la comparación entre los métodos de clasificación diseñados, entrenados y validados sobre una base de datos sintéticos.

La idea de este trabajo parte del convenio firmado entre la empresa Indra y el programa de Máster en Economía, Finanzas y Computación de la Universidad de Huelva, para desarrollar trabajos de Fin de Máster que resuelvan una problemática empresarial concreta. En este caso, la detección del fraude en las operaciones monetarias realizadas a través de plataformas de banca móvil. Para ello Indra proporciona la base de datos necesaria para el diseño y validación de los métodos propuestos en este trabajo.

# 2 Revisión de la bibliografía

La revisión de la literatura reciente, aporta numerosos trabajos que tratan de determinar reglas para clasificar las operaciones electrónicas en legítimas o fraudulentas, demostrando que este tema se ha convertido en los últimos años en un foco de atracción de los investigadores. Una prueba de ello la encontramos en los trabajos de recopilación publicados [3, 4, 5, 6, 7] que analizan las diferentes técnicas y enfoques utilizados para la detección del fraude en un sentido amplio.

Uno de los problemas básicos y comunes a muchos de los estudios analizados, es el trabajo con bases de datos con clases muy desiguales debido a la baja tasa de existencia de fraude entre la base de datos de registros. Por ejemplo, nos encontramos [8, 9, 21, 22]con trabajos que manejan tasas de fraude muy por debajo del 10%. Para resolver este inconveniente de trabajar con datos no balanceados se han adoptado diferentes soluciones. Por un lado, las técnicas de reducción de datos que balancean los registros reduciendo las clases minoritarias, en contraste con las técnicas de sobremuestreo (oversampling) que aumentan el tamaño de las clases minoritarias para balancear la base de datos [10].

También es relativamente frecuente en los trabajos examinados [11, 12, 13] la utilización de datos sintéticos para el entrenamiento y validación de los sistemas de detección, bien por la imposibilidad de obtener datos anonimizados, tanto por cuestiones legales, por razones competitivas, o por cuestiones prácticas.

En cuanto a las áreas de trabajo en las que se han aplicado técnicas de detección del fraude encontramos trabajos [14] relacionados con la detección del fraude en los seguros agrarios, seguros del hogar, seguros del automóvil, seguros de salud, tarjetas de crédito, blanqueo de dinero, etc. Y más concretamente para el caso de España hemos revisado trabajos relacionados con las operaciones con tarjetas de crédito [15] y con los seguros del automóvil [16].

Se han empleado un amplio rango de técnicas dentro de la minería de datos para detectar el fraude, que van desde los métodos más simples de regresión lineal o regresión logística [17] a técnicas más sofisticadas como las redes neuronales que han sido ampliamente utilizadas en este tipo de investigaciones. Encontramos trabajos que, por ejemplo, utilizan las técnicas de razonamiento basado en casos (*case based reasoning*, CBR) [18], clasificadores bayesianos [19],

árboles de decisión [17, 19, 20] , *random forest* [21,23], máquinas de soporte vectorial (*support vector machine*, SVM) [24], método de los k vecinos más próximos (*k nearest neighbors*, KNN) [24,25] y por último, algoritmos de extreme gradient boosting (XGBoosting) [26].

# 3 Metodología

El proceso seguido (figura 1) para alcanzar los objetivos propuestos queda representado en la siguiente figura y guiará la estructura de este apartado metodológico.

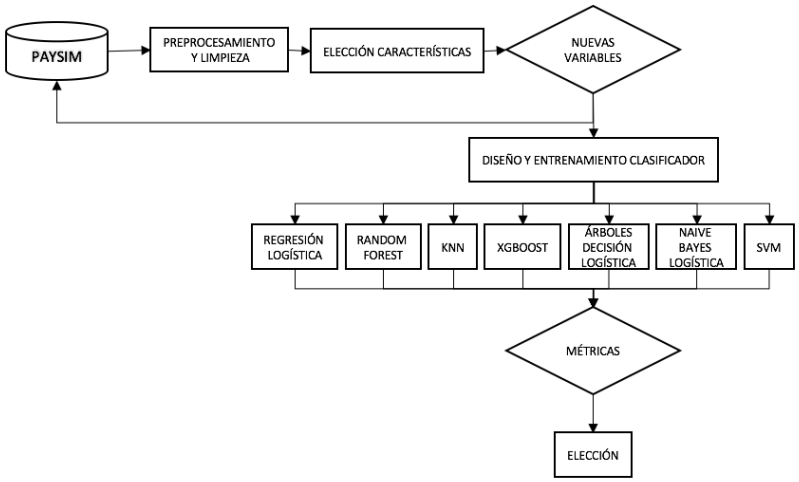



**Figura 1:** Flujo del proceso

## 3.1 Obtención de datos

Uno de los principales inconvenientes en el desarrollo de algoritmos de clasificación que permitan detectar operaciones económicas o financieras que podrían ser catalogadas como fraudulentas es la dificultad de contar con un conjunto de registros reales, suficientes, confiables y que identifiquen los comportamientos fraudulentos, tanto para entrenar como para validar los algoritmos de clasificación.

Por ello hacemos uso de la base de datos PaySim (disponible en www.kaggle.com), compuesta por un conjunto de datos generados artificialmente a partir de una muestra real de operaciones financieras realizadas a través del teléfono móvil facilitada por la empresa telefónica Ericsson, que provee de este sistema de transacciones monetarias a 14 países de todo el mundo. Por lo tanto, PaySim genera un conjunto de datos sintéticos reproduciendo los comportamientos observados en una muestra real.

El estudio de este tipo de datos adquiere relevancia en los últimos años ya que las entidades financieras facilitan y promueven el movimiento del dinero online, en gran medida a través del

teléfono móvil, lo que ha generado un incremento en las transacciones móviles y por tanto una reducción del dinero físico en las transacciones diarias.

Para cada una de las transacciones recogidas en la base de datos se retienen once variables, que definimos a continuación [27]:

1. Step: Variable discreta que representa la unidad de tiempo. La base de datos recoge transacciones realizadas durante las 24 horas en un día.
2. Type: Variable categórica que recoge cada uno de los cinco tipos de transacciones consideradas:
   a. Cash-in: Supone la entrada de efectivos en la cuenta, a través de un establecimiento comercial. Esto es, entregas de dinero en la caja de un establecimiento comercial para ingresar dinero en la cuenta.
   b. Cash-out: Esta es una operación inversa a la anterior. Implica una retirada de fondos a través de un establecimiento comercial.
   c. Debit: Es un ingreso en una cuenta bancaria enviando dinero desde la cuenta asociada al teléfono móvil.
   d. Payment: Es un proceso de pago de bienes y servicios a un establecimiento comercial.
   e. Transfer: Implica una transferencia de dinero a otro usuario del servicio de banca móvil.
3. Amount: Es la cantidad de dinero implicada en cada una de las operaciones.
4. NameOrig: Es una variable categórica que identifica al usuario de la cuenta de origen de las operaciones descritas anteriormente.
5. NewbalanceOrig: Estado de la cuenta de origen después de la transacción.
6. OldbalanceDest: Balance de la cuenta de origen antes de la operación financiera.
7. NameDest: Identifica al usuario de destino.
8. NewbalanceDest: Estado de la cuenta de destino después de la transacción
9. OldbalanceDest: Balance de la cuenta de destino antes de la operación financiera
10. IsFraud: Variable dicotómica que identifica las transacciones catalogadas como fraudulentas. En esta base de datos sintética se han considerado básicamente dos tipos de fraude. En primer lugar, la pérdida sobre el control de las cuentas por parte de su legítimo propietario y por otro lado las estafas a dichos usuarios. En el primer caso una persona ajena al efectivo usuario de la cuenta consigue acceder a la misma vaciándola a través de cuentas “mulas” o directamente extrayendo a través de un establecimiento comercial la máxima cantidad permitida.

La segunda forma de estafa se realiza a través de los comerciantes, intermediarios entre los clientes y el servicio de pago por móvil. Estos establecimientos prestan servicios como la retirada de efectivo a través de los cajeros, el pago en caja para realizar imposiciones en una cuenta, emisión de cupones, gestión de reclamaciones, etc.

11. IsFrlaggedFraud: Es una alerta que paraliza operaciones por encima de las 200.000 unidades monetarias.

## 3.2 Elección de características

En esta fase debemos decidir cuáles son las variables que van a intervenir en el proceso y limpiarlas. Este proceso puede marcar la diferencia entre obtener buenos resultados o no.

Algunas de las variables descritas en la fase anterior son transformadas para poder trabajar con datos numéricos, por ello la mayoría de las variables categóricas se transforman o son eliminadas de la base de datos original.

En primer lugar, tratamos la variable type que contiene diferentes tipos de transacciones (Figura 2), convirtiéndola en cinco variables dicotómicas que identifican cada uno de los tipos de operaciones, pasando por tanto de una variable a 5 variables dummies.

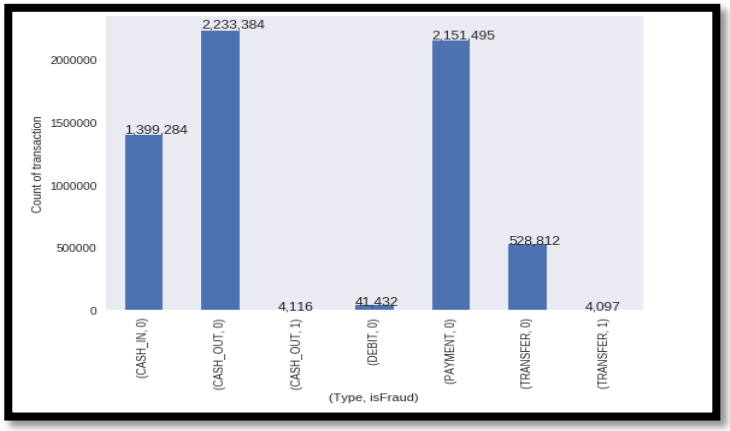



**Figura 2.** Variables type normales y fraudulentas

Por otro lado, al descubrir que sólo dos de los tipos de transacciones definidos están implicados en operaciones fraudulentas: cash-out y transfer en una proporción similar dentro del grupo de las operaciones fraudulentas, eliminaremos de nuestros registros el resto de transacciones que impliquen un tipo diferente a cash-out o transfer (Figura 3). Pasando por tanto de una base de

más de seis millones de registros que incluyen cinco tipos de transacciones, a una base de 2,8 millones de registros incluyendo sólo dos clases de transacciones. Esto va a simplificar el trabajo y permitirá reducir el tiempo de cálculo de cada uno de los algoritmos de clasificación propuestos.

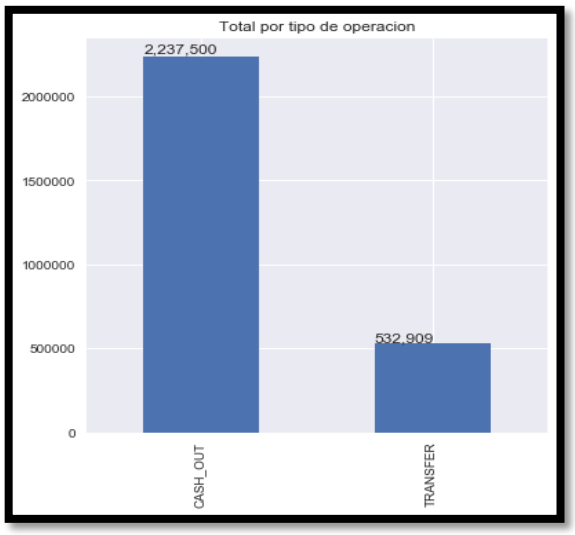



**Figura 3.** Cantidad de operaciones en type

En este primer análisis descriptivo (Figura 4) de los datos comprobamos que la base de datos no está balanceada, esto es, el porcentaje de operaciones identificadas como fraudulentas representa una muy pequeña proporción respecto a todos los registros (0.2965%).

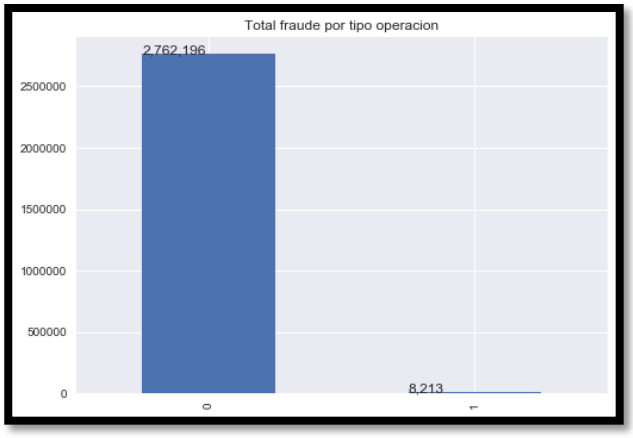



**Figura 4** Cantidad de operaciones fraudulentas

Para salvar este inconveniente utilizaremos un método de submuestreo. Estos métodos eliminan observaciones de la clase mayoritaria con el fin de igualar los tamaños de las clases.

Utilizamos la técnica under-sampling o muestreo aleatorio dentro de la base original para equilibrar la muestra, esto es, quedarnos con la misma proporción de datos fraudulentos que de transacciones legítimas. Para ello, después de dividir el conjunto de datos entre train y test, dentro de train aplicaremos la técnica under-sampling. Obtendremos dos subconjuntos, X_train e Y_train, balanceados con el mismo número de casos normales y fraudulentos como vemos en la Figura 5.

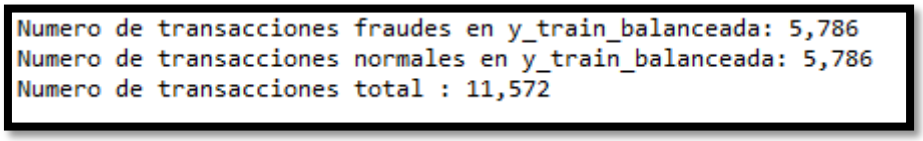



**Figura 5** Número de datos para y train balanceada

Ante bases de datos de tanto volumen, la técnica under-sampling es muy recomendable, ya que la principal ventaja de esta decisión es el ahorro de tiempo de ejecución.

Dado que no descubrimos ningún patrón de comportamiento esperado referido a las variables nameOrig y nameDest se eliminarán de la misma forma del conjunto de registros original. Estas variables me permiten identificar además si el usuario de la cuenta es una persona física o un comerciante. Así no hallamos ninguna operación fraudulenta en la que esté implicado un comerciante y además no encontramos relación alguna entre los identificadores de los usuarios de las cuentas de origen o destino y el fraude. Por tanto, al considerar que no aportan ninguna información que pueda ser relevante para la clasificación de operaciones como legítimas o fraudulentas, son eliminados, aunque previamente se trata de extraer la máxima información de ellas. Para ello se crean nuevas variables que retienen el número de veces que estos usuarios aparecen en el momento anteriores. Creando cuatro variables que informan de si un individuo concreto aparece en n periodos anteriores.

Dentro de este primer análisis realizado en los datos, comprobamos que la última de las variables descritas, isFlaggedFraud, nos arroja un total de sólo 16 operaciones que se paralizan por ser operaciones que superan una determinada cantidad. La descartamos del análisis realizado porque

entendemos que no es una variable importante en la determinación de si una operación es fraudulenta o no y porque es muy simple establecer este mecanismo de alerta en cualquier registro de transacciones.

A partir de las variables referentes a los balances de las cuentas, se crean e incorporan al análisis dos nuevas variables, errorbalanceDest y errorbalanceOrig, que representan los errores calculados tanto en los balances de origen como en los de destino, de la siguiente forma:

ErrorbalanceOrig = newbalanceOrig + amount – oldbalanceOrig

ErrorbalanceDest = oldbalanceDest + amount - newbalanceDest

Por lo tanto, una vez realizada la tarea de depuración y limpieza de la base de datos, trabajamos con 2.769.745 operaciones y las siguientes 14 variables:

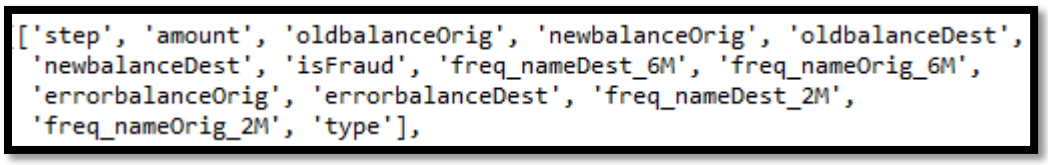



**Figura 6** Variables finales del dataset

Con la creación de freq_nameOrig_2M, freq_nameDest_2M, freq_nameOrig_6M y freq_nameDest_6M, intentamos aportar información relevante a la hora de solucionar el problema, las cuatro variables podemos verlas en la Figura 6.

Reflejan la frecuencia en n periodos anteriores. La primera en el conjunto sesgado de 2 millones donde habíamos filtrado solo registros de cash-out y transfer, y la segunda se hace para el data set completo con 6.362.620 millones de operaciones.

Utilizamos Python, en su versión 3.6.3 como lenguaje de programación, porque además de ser un lenguaje multiparadigma y multiplataforma, es de código abierto, gratuito, veloz y posee una gran cantidad de librerías que facilitan el trabajo y el tratamiento de datos.

Una vez realizadas las tareas de depuración o limpieza de los registros iniciales disponemos de una base de 2.769.745 registros con 14 variables que utilizaremos tanto para el entrenamiento y validación de los métodos de clasificación utilizados.

## 3.3 Elección del clasificador

Aplicamos diferentes algoritmos procedentes de diferentes paradigmas, para explorar las distintas posibilidades y elegir el método más adecuado.

**Árboles de decisión (Decision Trees)**

Es un tipo de algoritmo de aprendizaje supervisado que se utiliza principalmente para problemas de clasificación. Funciona para variables dependientes categóricas y continuas. Los árboles de decisión se basan en reglas dentro del conjunto de datos entrenamiento, donde se saca la información y construye esas reglas. Dichas reglas se aplicarán al conjunto de datos test.

El algoritmo intenta resolver el problema, mediante el uso de la representación de árbol . Cada nodo interno del árbol corresponde a un atributo, y cada nodo de hoja corresponde a una etiqueta de clase.

**Bosques aleatorios (Random Forest)**

Random forest es uno de los algoritmos de clasificación más popular, y usado tanto para problemas de regresión o de clasificación como el nuestro. Es un método que combina una cantidad grande de árboles de decisión independientes probados sobre conjuntos de datos aleatorios con igual distribución.

Es un algoritmo de clasificación supervisado, y consiste en crear un “bosque” con varios árboles de decisión. Cuantos más arboles creemos, más robusto será el sistema y mayor precisión.

Mejora la precisión en la clasificación mediante la incorporación de aleatoriedad en la construcción de cada clasificador individual.

En la fase de entrenamiento creamos diferentes árboles de forma independiente, cada uno con valores de entrada distintos. Se selecciona aleatoriamente un porcentaje de los datos, esto se hace también con los atributos, y en cada nodo se selecciona la partición óptima. Una vez que están todos los árboles, se evalúa de forma independiente y la predicción será la media de todos los árboles.

Una de sus ventajas es la eficiencia trabajando con grandes volúmenes y determinación de las variables más importantes en la clasificación.

**Máquina de soporte vectorial (Support Vector Machines, SVM)**

Utiliza un método de aprendizaje supervisado, en el que se aprende de un conjunto de registros previamente etiquetados con su clase correspondiente. El algoritmo representa los puntos de la muestra en el espacio y busca un gap de separación que me permita diferenciar las clases previamente definidas. El clasificador SVM es principalmente utilizado en el tratamiento de problemas de clasificación múltiple, esto es, con más de dos clases para predecir.

**Naive Bayes**

Es una técnica de clasificación basada en el teorema de Bayes con un supuesto de independencia entre los predictores. En términos simples, un clasificador de Naive Bayes supone que la presencia de una característica particular en una clase no está relacionada con la presencia de ninguna otra característica. Incluso si estas características dependen unas de otras o de la existencia de otras características, un clasificador ingenuo de Bayes consideraría que todas estas propiedades contribuyen independientemente a la probabilidad de que esta fruta sea una manzana.

El modelo Naive Bayes es fácil de construir y particularmente útil para conjuntos de datos muy grandes.

- P ( c | x ) es la probabilidad posterior de la clase ( objetivo ) dado predictor ( atributo ).
- P ( c ) es la probabilidad previa de clase .
- P ( x | c ) es la probabilidad que es la probabilidad de clase dada por el predictor .
- P ( x ) es la probabilidad previa de predictor .

**k Vecinos más próximos (k Nearest Neighbors KNN)**

Los métodos KNN se utilizan para resolver problemas de clasificación y regresión. El método KNN es un algoritmo simple que clasifica un caso en función de su parecido a otros casos

previamente clasificados. El caso, por tanto, se asigna a la clase que es la más común entre sus k vecinos más cercanos medidos por una función de distancia.

Como comentamos, el funcionamiento del algoritmo clasificador kNN consiste en encontrar k número predefinido de muestras de entrenamiento más cercanas en la distancia al nuevo punto y predecir la etiqueta a partir de ellas.

Estas funciones de distancia pueden ser distancia Euclidiana, Manhattan, Minkowski y Hamming. La distancia euclidiana es la medida de distancia más comúnmente utilizada y medida entre dos puntos representa la longitud del camino que los conecta.

**Regresion logística**

Es un método de clasificación y no un algoritmo de regresión como podría parecer por el nombre. Se utiliza para estimar valores discretos (valores binarios como 0/1, sí/no, verdadero/falso) en función de un conjunto determinado de variables independientes. En palabras simples, predice la probabilidad de ocurrencia de un evento ajustando los datos a una función logit. Por lo tanto, también se conoce como regresión logit. Dado que, predice la probabilidad, sus valores de salida se encuentran entre 0 y 1 .

**XGboost**

El algoritmo XGBoost tiene una potencia de predicción inmensamente alta que lo convierte, a priori, en la mejor opción para la precisión en los eventos, ya que posee el modelo lineal y el algoritmo de aprendizaje en árbol, lo que hace que el algoritmo sea más rápido que las técnicas existentes de aumento de gradiente. El soporte incluye varias funciones objetivas, que incluyen regresión y clasificación.

Internamente, XGBoost representa todos los problemas como un caso de modelado predictivo de regresión que sólo toma valores numéricos como entrada.

## 3.4 Entrenamiento del modelo

Los algoritmos que estamos usando están basados en un conjunto de datos previamente conocido y tratado sobre el que vamos a extraer un conocimiento que queremos aplicar y contrastar sobre otro conjunto no conocido o nuevo. De nada servirá medir una predicción sobre datos que ya han

sido estudiados. Debemos evaluar el rendimiento sobre un conjunto diferente al que hemos usado para extraer información, por ello dividiremos el conjunto global en dos. Una parte será de entrenamiento y la otra servirá para evaluar (entrenamiento y test). Una de las particiones más usuales y la que usaremos en nuestro algoritmo será un 70% entrenamiento y 30% test. Nos quedarán 1.938.821 registros en entrenamiento y 830.924 registros para evaluar.

Debemos ajustar bien el modelo para que el conjunto de entrenamiento sea capaz de sacar toda la información y no caer en problemas de sesgo-varianza. El sesgo mide el error medio del modelo usando distintos conjuntos de entrenamiento, mientras que la varianza mide la sensibilidad a cambios en los datos.

A nuestro conjunto de entrenamiento le aplicaremos la técnica de balanceo anteriormente descrita (under-sampling). Con estos datos seremos capaces de extraer la información necesaria para predecir sobre nuestro conjunto de test.

## 3.5 Métricas

Algunos de los métodos más usados para medir el rendimiento de un algoritmo de clasificación son la matriz de confusión, la exactitud de la clasificación (accuracy) y el área bajo la curva Roc (AUC-ROC).

| | | **VALORES MODELADOS: XM** | |
|---|---|---|---|
| | | Falso | Verdadero |
| **VALORES ACTUALES: X** | Falso | Verdadero Negativo (True Negative//TN) | Falso Positivo (False Positive//FP) |
| | Verdarero | Falso Negativo (False Negative//FN) | Verdadero Positivo (True Positive//TP) |

**Figura 7** Matriz de confusión

La matriz de confusión permite visualizar y medir la capacidad del algoritmo para clasificar un nuevo registro entre las dos clases establecidas. Los verdaderos positivos (TP) y verdaderos negativos (TN) representan la proporción en la que los nuevos registros quedan perfectamente

clasificados, mientras que los falsos positivos (FP) y los falsos negativos (FN) representan los casos en los que el clasificador no clasifica correctamente.

La exactitud de la clasificación (accuracy), es una métrica muy buena cuando todas las variables tienen la misma importancia.

- Accuracy = (TP + TN) / Total
- Recall = TP / (TP + FN)
- Precision = TP / (TP + FP)

Tendremos en cuenta como medidas más apropiadas a la hora de tomar una decisión otras como por ejemplo la tasa de verdaderos positivos (True positive ratio TPR) y la tasa de falsos positivos (false positive ratio FPR).

El TPR o Recall, es la cantidad de operaciones fraude clasificadas de forma correcta respecto al total de fraude. TPR = TP / P

FPR, es la cantidad de operaciones normales que han sido marcadas como fraude respecto del total de transacciones normales. FPR = FP / N.

Estas dos métricas anteriores nos sirven para la curva ROC (característica operativa del receptor), que será la métrica a la que más peso vamos a dar a la hora de decidir un buen clasificador. La curva ROC representa el TPR en función del FPR. Veamos un ejemplo de curva AUC-ROC con la siguiente figura (Figura 8).

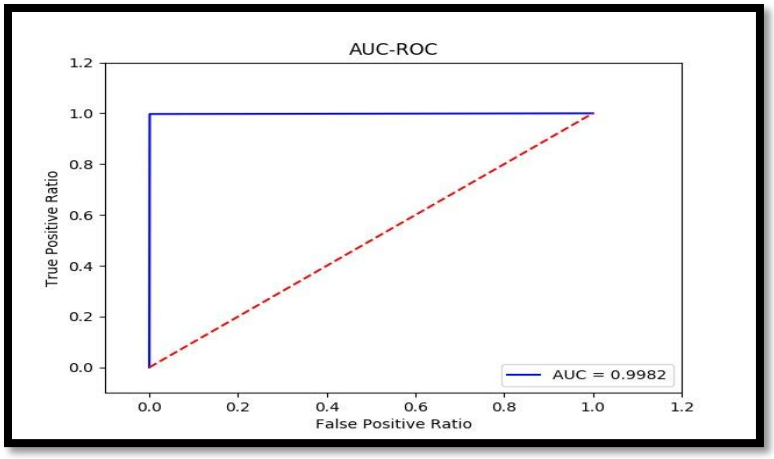



**Figura 8** Ejemplo curva AUC-ROC

A partir de dicha curva, una forma muy extendida de medir el rendimiento global del modelo es calcular el área bajo la curva (AUC), el cual es un valor que oscila entre 0 y 1 (todos los ejemplos mal clasificados o todos los ejemplos bien clasificados, respectivamente).

- 90-100% = clasificador excelente
- 80-90% = clasificador bueno
- 70-80% = clasificador aceptable
- 60-70% = clasificador pobre
- 50-60% = clasificador erróneo

# 4 Resultados

Una vez hemos ejecutado los algoritmos comparamos los resultados en función de las métricas elegidas para determinar el método más efectivo en nuestro problema. En primer lugar, obtendremos las métricas de cada método por separado, para posteriormente pasar a una comparación entre los siete métodos empleados. Finalizamos este apartado observando los resultados que obtendríamos sin hacer ningún tipo de procesado a la base de datos, utilizando los mismos algoritmos.

## 4.1 Resultados individuales

Mostraremos para cada algoritmo empleado su matriz de confusión, accuracy, recall, precisión, promedio de la precisión y graficaremos los dos mejores AUC-ROC.

Vemos en la tabla 1 y 2 las matrices de confusión para todos los algoritmos empleados y podemos observar de que forma cada clasificador distribuye las observaciones entre falsos positivos y falsos negativos.

| **Decision tree** | | **Random forest** | | **SVM** | | **NaiveBayes** | |
|---|---|---|---|---|---|---|---|
| TN=825770 | FP=2743 | TN=828165 | FP=348 | TN=828513 | FP=0 | TN=810523 | FP=17990 |
| FN=6 | TP=2405 | FN=7 | TP=2404 | FN=2399 | TP=12 | FN=1251 | TP=1160 |

**Tabla 1** Matriz confusión algoritmos I

| **KNN** | | **Logistic Regression** | | **XGBoost** | |
|---|---|---|---|---|---|
| TN=783875 | FP=44638 | TN=724712 | FP=103801 | TN=828251 | FP=262 |
| FN=93 | TP=2318 | FN=66 | TP=2345 | FN=6 | TP=2405 |

**Tabla 2** Matriz confusión algoritmos II

Se presentan en la tabla 3 los resultados obtenidos para las métricas importantes en la elección del método de clasificación.

| | Decision Tree | Random Forest | SVM | Naive Bayes | KNN | Logistic Regression | XGBoost |
|---|---|---|---|---|---|---|---|
| **Accuracy = (TP+TN)/TOTAL** | 0.996691 | 0.999572 | 0.997112 | 0.976843 | 0.946167 | 0.874998 | 0.999677 |
| **Recall = TP/(TP+FP)** | 0.997511 | 0.997096 | 0.004977 | 0.481128 | 0.961426 | 0.972625 | 0.997511 |
| **Precision = TP/(TP+FP)** | 0.467171 | 0.873546 | 1.0 | 0.060574 | 0.049365 | 0.022092 | 0.901762 |
| **Nivel de promedio de precisión** | 0.732345 | 0.935325 | 0.503932 | 0.271604 | 0.505452 | 0.497398 | 0.899525 |
| **AUC** | 0.9971 | 0.9983 | 0.5025 | 0.7297 | 0.9538 | 0.9237 | 0.9986 |

**Tabla 3** Métricas algoritmos

En la tabla 4 vemos representada la curva AUC-ROC para los dos modelos con mayor valor de la tabla 3, la interpretación de esta curva será otra de los criterios de elección del mejor método.

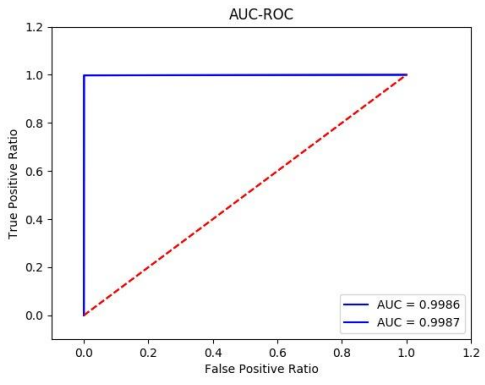
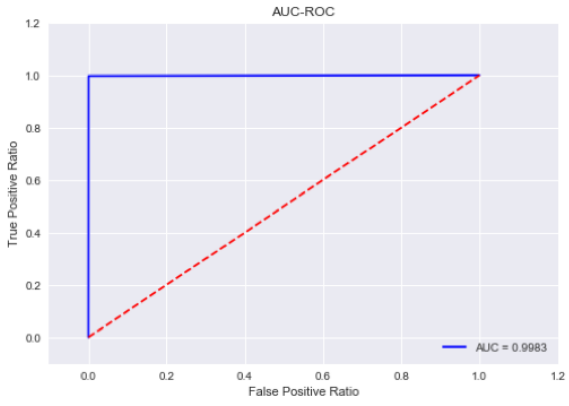
| XGBoost | Random Forest |
|---|---|
|  |  |



**Tabla 4** Curvas mejores algoritmos

## 4.2 Comparación resultados

Elegiremos dos de los mejores algoritmos en base a los resultados obtenidos, fijándonos en tres métricas de las anteriores que consideramos importantes. Las métricas en las que basaremos nuestra decisión son en primer lugar la curva AUC-ROC, recall y por ultimo el nivel de precisión.

| | Decision Tree | Random Forest | SVM | Naive Bayes | KNN | Logistic Regression | XGBoost |
|---|---|---|---|---|---|---|---|
| **Recall** | 0.997511 | 0.997096 | 0.004977 | 0.481128 | 0.961426 | 0.972625 | 0.997511 |
| **Precision** | 0.467171 | 0.873546 | 1.0 | 0.060574 | 0.049365 | 0.022092 | 0.901762 |
| **AUC-ROC** | 0.9971 | 0.9983 | 0.5025 | 0.7297 | 0.9538 | 0.9237 | 0.9986 |

**Tabla 5** Comparación algoritmos

## 4.3 Resultados sin procesar

Con objeto de conocer si el preprocesamiento o preparación inicial de los datos realizados en la segunda fase del proceso mejora los resultados, realizamos el mismo análisis, es decir, aplicamos los mismos algoritmos en el conjunto de datos originales. Los resultados obtenidos se muestran en la tabla 6.

| | Decision Tree | Random Forest | SVM | Naive Bayes | KNN | Logistic Regression | XGBoost |
|---|---|---|---|---|---|---|---|
| **Recall** | 0.878898 | 0.769947 | 0.704272 | 0.398136 | 0.739975 | 0.423653 | 0.996681 |
| **Precision** | 0.891170 | 0.982428 | 1.0 | 0.080003 | 0.895149 | 0.475238 | 0.999168 |
| **AUC-ROC** | 0.9393 | 0.8850 | 0.8521 | 0.6922 | 0.8699 | 0.7111 | 0.9983 |

**Tabla 6** Resultados base de datos original

# 5 Conclusiones

Las operaciones online o a través del teléfono móvil se han convertido en una práctica diaria para las empresas y para los ciudadanos en general. Como consecuencia, tanto empresas como instituciones financieras dedican una parte de sus recursos económicos en el diseño e implantación de medidas para garantizar la seguridad en este tipo de transacciones. La identificación de estas prácticas fraudulentas confiere confiabilidad al sistema.

Hemos presentado en este trabajo un conjunto de técnicas que permitan la identificación de operaciones deshonestas, con diferentes niveles de complejidad tanto en su diseño como en su implementación, que han sido aplicadas sobre una base de datos sintética con el objetivo de determinar qué método permite una mejor detección de este tipo de prácticas. A la vista de los resultados podemos concluir que de entre los algoritmos analizados, los que mejor precisión y fiabilidad alcanzan son el XGBoost y Random Forest, resultando clasificadores casi perfectos en los datos utilizados. De la misma forma podemos afirmar que las tareas de preprocesamiento de los datos iniciales, tales como limpieza de los datos y selección de características, creación de nuevas variables y aplicación de técnicas de balanceo, han permitido la mejora del rendimiento, aunque las técnicas que vuelven a ser determinantes a la hora de afrontar este problema son XGboost, Random forest.

Pretendemos continuar con la mejora de la aplicación de estas técnicas de análisis de datos a través de la mejora de los parámetros utilizados en los algoritmos y transformación e introducción de nuevas variables.

# Bibliografía

**[1]** The Nilson Report" 2016 Issue 1096. https://www.nilsonreport.com/publication_newsletter_archive_issue.php?issue=1096

**[2]** Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical science*, 17(3) 235-249.

**[3]** Kou, Y., Lu, C. T., Sirwongwattana, S., & Huang, Y. P. (2004). Survey of fraud detection techniques. In *Networking, sensing and control, 2004 IEEE international conference on* (Vol. 2, pp. 749-754). IEEE.

**[4]** Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*.

**[5]**Wang, S. (2010, May). A comprehensive survey of data mining-based accounting-fraud detection research. In *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on* (Vol. 1, pp. 50-53). IEEE.

**[6]** Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, *50*(3), 559-569.

**[7]** Flegel, U., Vayssière, J., & Bitz, G. (2010). A state of the art survey of fraud detection technology. *Insider threats in cyber security*, 73-84.

**[8]** Foster, D. P., & Stine, R. A. (2004). Variable selection in data mining: Building a predictive model for bankruptcy. *Journal of the American Statistical Association*, *99*(466), 303-313.

**[9]** Bentley, P. J., Kim, J., Jung, G. H., & Choi, J. U. (2000, October). Fuzzy darwinian detection of credit card fraud. In *the 14th Annual Fall Symposium of the Korean Information Processing Society* (Vol. 14).

**[10]** Padmaja, T. M., Dhulipalla, N., Bapi, R. S., & Krishna, P. R. (2007, December). Unbalanced data classification using extreme outlier elimination and sampling techniques for

fraud detection. *In Advanced Computing and Communications, 2007. ADCOM 2007. International Conference on* (pp. 511-516). IEEE.

**[11]** Barse, E. L., Kvarnstrom, H., & Jonsson, E. (2003, December). Synthesizing test data for fraud detection systems. In Computer Security Applications Conference, 2003. Proceedings. 19th Annual (pp. 384-394). IEEE.

**[12]** Lundin, E., Kvarnström, H., & Jonsson, E. (2002). A synthetic fraud data generation methodology. *Information and Communications Security*, 265-277.

**[13]** Barse, E. L., Kvarnstrom, H., & Jonsson, E. (2003, December). Synthesizing test data for fraud detection systems. In *Computer Security Applications Conference, 2003. Proceedings. 19th Annual* (pp. 384-394). IEEE.

**[14]** Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, *50*(3), 559-569.

**[15]** Dorronsoro, J. R., Ginel, F., Sanchez, C., & Cruz, C. S. (1997). Neural fraud detection in credit card operations. *IEEE transactions on neural networks*, *8*(4), 827-834.

**[16]** *Artís, M., Ayuso, M., & Guillen, M. (1999).* Modelling different types of automobile insurance fraud behaviour in the Spanish market. *Insurance: Mathematics and Economics*, *24*(1), 67-81.

**[17]** Shen, A., Tong, R., & Deng, Y. (2007, June). Application of classification models on credit card fraud detection. In *Service Systems and Service Management, 2007 International Conference on* (pp. 1-4). IEEE.

**[18]** Wheeler, R., & Aitken, S. (2000). Multiple algorithms for fraud detection. *Knowledge-Based Systems*, *13*(2), 93-99.

**[19]** Kirkos, E., Spathis, C., & Manolopoulos, Y. (2007). Data mining techniques for the detection of fraudulent financial statements. *Expert systems with applications*, *32*(4), 995-1003.

**[20]** Şahin, Y. G., & Duman, E. (2011). Detecting credit card fraud by decision trees and support vector machines.

**[21]** Dal Pozzolo, A., Caelen, O., Le Borgne, Y. A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert systems with applications*, *41*(10), 4915-4928.

**[22]** Wei, W., Li, J., Cao, L., Ou, Y., & Chen, J. (2013). Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web*, *16*(4), 449-475.

**[23]** Zhang, J., Zulkernine, M., & Haque, A. (2008). Random-forests-based network intrusion detection systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(5), 649-659.

**[24]** Whitrow, C., Hand, D. J., Juszczak, P., Weston, D., & Adams, N. M. (2009). Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery*, *18*(1), 30-55.

**[25]** He, H., Graco, W., & Yao, X. (1998, November). Application of genetic algorithm and k-nearest neighbour method in medical fraud detection. In *Asia-Pacific Conference on Simulated Evolution and Learning* (pp. 74-81). Springer, Berlin, Heidelberg.

**[26]** Tavares, G. M., Mastelini, S. M., & Barbon Jr, S. (2017). User Classification on Online Social Networks by Post Frequency. *CEP*, *86057*, 970.

**[27]** Lopez-Rojas, E., Elmir, A., & Axelsson, S. (2016). PaySim: A financial mobile money simulator for fraud detection. In *28th European Modeling and Simulation Symposium, EMSS, Larnaca* (pp. 249-255). Dime University of Genoa.