Universitat de les Illes Balears

TRABAJO DE FIN DE MÁSTER

ESTUDIOS EXPLORATORIOS SOBRE DETECCIÓN DE FRAUDE CON TARJETAS DE CRÉDITO

Jorge Lazo Rosado

Máster Universitario en Análisis de Datos Masivos en Economía y Empresa

(Especialidad / Itinerario Herramientas en Gestión y Análisis Inteligente de Datos)

Centro de Estudios de Postgrado

Año Académico 2022-23

ESTUDIOS EXPLORATORIOS SOBRE DETECCIÓN DE FRAUDE CON TARJETAS DE CRÉDITO

Jorge Lazo Rosado

Trabajo de Fin de Máster

Centro de Estudios de Postgrado

Universidad de las Illes Balears

Año Académico 2022-23

Palabras clave del trabajo:

fraud, card, credit

Nombre Tutor/Tutora del Trabajo Prof. Isaac Lera Castro

Nombre Tutor/Tutora (si procede) Prof. Antoni Jaume Capó

Nombre Tutor/Tutora (si procede)

Universitat de les Illes Balears

Indice

1. Introducción 4

1.1. Motivación 4

1.2. Objetivos 5

2. Antecedentes 5

2.1. Bases de datos 5

2.2. Validación cruzada 5

2.3. Evaluación de algoritmos de clasificación 6

3. Implementación 6

3.1. Metodología 6

3.2. Muestra (Dataset) 7

3.3. Depuración de datos 7

3.4. Redenominación de variables 7

3.5. Conversión de variables categóricas a ficticias (dummies) 7

3.6. División de la Muestra 7

3.7. Análisis exploratorio 7

3.8. Selección de variables 8

3.9. Validación cruzada 8

4. Algoritmos de clasificación 9

4.1. Regresión Logística 9
Clasificación con vector de probabilidades 9
Validación del modelo de regresión logística 10
Matriz de confusión con regresión logística 10

4.2. K-Nearest Neighbors (KNN) 10
Validación del Modelo de predicción 10

4.3. Árboles de decisión 10
Validación del modelo de árboles de decisión 11

4.4. Bosques aleatorios (Random Forest) 12
Matriz de Decisiones 12
Validación del modelo 12

4.5. Curvas ROC 12

4.6. Aplicación Práctica 13

5. Análisis Resultados 14

5.1. Eficiencia de Predicción (scores) 14

5.2. Matrices de confusión 14

6. Conclusiones 15

7. Referencias 15

Apéndice A. 15

Estudios exploratorios sobre detección de fraude con tarjetas de crédito

Jorge Lazo Rosado
Tutores: *Prof. Isaac Lera Castro, Prof. Antoni Jaume Carbó*

Treball de fi de Màster Universitari en Analisi de Dades Massives en Economia i Empresa (MADM)

Universitat de les Illes Balears
07122 Palma de Mallorca
j.lazo-rosado1@estudiant.uib.cat

Resumen

En este trabajo voy a analizar el problema del fraude con tarjetas de crédito. Asimismo, voy a plantear una metodología para crear modelos de predicción para clasificar las transacciones de pago (observaciones) mediante algoritmos de aprendizaje estadístico supervisado a partir de una muestra (base de datos).

1. Introducción

Es evidente que varios factores, como cambios en el comportamiento de compra de los consumidores a través de Internet, han incrementado el uso extendido de tarjetas de créditos como medio de pago.

Los pagos online benefician tanto a los negocios como a los consumidores en términos de conveniencia, velocidad y flexibilidad. La dependencia en este medio de pago es enorme tanto para los comercios como para los consumidores.

Sin embargo, este incremento ha traído como consecuencia el aumento del fraude en el uso de medios de pago electrónico.

Cada año se pierden miles de millones de euros en todo el mundo debido al fraude con tarjetas de crédito. [1], obligando así a las entidades financieras a mejorar continuamente sus sistemas de detección de fraude. En los últimos años, varios estudios han propuesto el uso de técnicas de aprendizaje automático y minería de datos para abordar este problema.

Las pérdidas generadas impactan mucho al comercio, a la banca y a las empresas de seguro.

La detección del fraude es el proceso de identificar transacciones financieras sospechosas.

El fraude tiene lugar mediante el pago en el punto de venta (points of sales) POS, transacción de pago electrónica no presencial (card-not-present) CNP con tarjetas extraviadas, robadas, copiadas. Siendo el porcentaje de fraude mayor en operaciones donde el uso de la tarjeta de crédito no es presencial. [2]

1.1. Motivación

La motivación de este trabajo es aportar más conocimiento sobre la aplicación de los métodos de

aprendizaje automático (*Machine Learning*) en la detección del fraude con medios de pago (ej. tarjetas de crédito).

En este contexto, considero que el aprendizaje automático (ML) es una metodología idónea para este fin y dispone de un conjunto de herramientas que pueden servir a los sistemas informáticos a reconocer patrones de datos (a través del análisis y tratamiento de datos masivos), aprender de ellos y desarrollar habilidades para llevar a cabo ciertas tareas de forma automática.

Asimismo, quiero demostrar desde un punto de vista estadístico cuán eficientes pueden ser las técnicas de clasificación comparando los resultados y también cómo se desarrolla la programación de estas técnicas en Python.

Estos desarrollos se podrían implementar dentro de sistemas de seguridad de bancos, instituciones financieras, medios de pago, plataformas de comercio online, etc.

El aprendizaje automático está considerado como una de las técnicas más exitosas para la identificación del fraude. Este utiliza los métodos de clasificación y regresión para el reconocimiento del fraude con tarjetas de crédito. [3].

Por ejemplo, en el trabajo *A Review on Credit Card Fraud Detection using Machine Learning (International Journal of scientific & technology research)* los autores aplicaron los algoritmos de regresión logística, k-vecinos más próximos (KNN), bosques aleatorios, árboles de decisión máquinas de soporte vectorial (SVM) y clasificador bayesiano ingenuo (naive bayes). Los resultados del estudio presentaron las ratios de eficiencia de predicción, sensibilidad y especificidad. Donde los algoritmos de bosques aleatorios, regresión logística y árboles de decisión presentaron los mejores resultados. [4]

1.2. Objetivos

Los objetivos de este trabajo son:

I. Hacer una breve reseña de la problemática del fraude con tarjetas de crédito, mencionando algunos estudios y trabajos de investigación ya publicados sobre la aplicación del Aprendizaje Automático (*ML)* en este campo.
II. Presentar una visión general sobre los métodos de clasificación
III. Crear modelos de predicción para clasificar aquellas transacciones de pagos en regulares o irregulares (fraude / no fraude) mediante algoritmos de clasificación.
IV. Ajustar dichos modelos a los datos de prueba para clasificarlos mediante validación cruzada
V. Evaluar su eficiencia mediante ratios y gráficos
VI. Presentar resultados y conclusiones

2. Antecedentes

2.1. Bases de datos

La mayoría de los datos disponibles sufren el problema de desequilibrio (imbalanced / unbalanced data) donde la proporción de una categoría es muy dominante sobre la otra [5]. Por ejemplo, categorías: 1 = fraude, 0 = no fraude. Esto puede generar un sesgo en el análisis.

Por otra parte, no es fácil disponer de bases de datos que faciliten las variables de medición originales debido a las restricciones por protección de datos. Con el análisis de componentes principales (PCA), se convierten las variables en nuevas dimensiones explicativas. Por tanto, al disponer sólo de las dimensiones, ya no se puede identificar las variables explicativas que determinan la clasificación de las observaciones.

Por ejemplo, los autores del artículo *Fraud Detection of Credit Card Using Logistic Regression* comentan que, dado que no pueden proporcionar las variables originales de la base datos, han transformado las variables originales en componentes principales con la técnica PCA "… *Features V1, V2, … V28 are the principal components obtained with PCA"*.[6]

2.2. Validación cruzada

Según los autores Payam Refaeilzadeh, Lei Tang, Huan Lui: "La validación cruzada es una manera de predecir el ajuste de un modelo a un hipotético conjunto de datos de prueba cuando no disponemos del conjunto explícito de datos de prueba." [7]

Para poder validar un modelo voy a dividir la muestra en datos de entrenamiento y datos de prueba.

Asimismo, voy a aplicar la validación cruzada con el método k-folds cross-validation.

Primero definimos una partición de la muestra en k partes (k-fold), tal que una parte será utilizada como

conjunto de prueba en cada iteración. De esta manera todas las particiones serán utilizadas. Mientras cada partición es utilizada como datos de prueba, las particiones restantes (k-1) serán utilizadas como datos de entrenamiento.

Repetimos el experimento varias veces partiendo los datos tantas veces como hayamos fijado el número de particiones (k-partes). Finalmente, calculamos la media aritmética obtenida de los resultados de cada iteración llevada a cabo. En cada una de las k iteraciones de este tipo de validación se realiza un cálculo de error. El resultado final lo obtenemos a partir de la media aritmética de los K valores de errores obtenidos.

2.3. Evaluación de algoritmos de clasificación

La matriz de confusión es un criterio de evaluación para medir la eficiencia de un modelo predictivo mediante la clasificación en 4 categorías

Real (actual)	Predicción	
	Negativo	Positivo
Ha dado negativo	TN (verdadero negativo)	FP (falso positivo)
Ha dado positivo	FN (falso negativo)	TP (verdadero positivo)

Tabla 1: Matriz de confusión

Voy a utilizar los siguientes indicadores de evaluación:

- TP (verdaderos positivos) = N.º de casos positivos clasificados correctamente
- TN (verdaderos negativos) = N.º de casos negativos clasificados correctamente
- FN (falsos negativos) = N.º de casos negativos clasificados incorrectamente
- FP (falsos positivos) = N.º de casos positivos clasificados incorrectamente
- FPR: Matriz de confusión (False-Postive Rate)
- Curva ROC (receiving operating characteristic)
- Ratio de precisión (Accuracy) es la proporción de observaciones bien clasificadas entre todas las observaciones de prueba.
- Sensibilidad (Recall) es la tasa de verdaderos positivos, calculado como el número de verdadreos positivos dividido entre la suma de verdaderos positivos y falsos negativos.
- Tasa de error equivalente (Equal Error Rate) para determinar los umbrales de aceptación y de rechazo
- Ratio de precisión (Accuracy) = T P + TN / (TP +TN +FP +FN)
- Sensitivilidad (Recall) = (TPR) = TP / (TP + FN)
- Especificity = TN / (TN + FP)
- FPR: (False-Postive Rate) = 1 - Specificity

El área debajo de la curva (AuC) es una medida de la capacidad de un clasificador binario para distinguir entre categorías y se utiliza como resumen de la curva ROC. Cuanto mayor sea el AUC, mejor será el rendimiento del modelo para distinguir entre las clases positivas y negativas [8].

3. Implementación

3.1. Metodología

Voy a aplicar una metodología de aprendizaje supervisado estructurándola en los siguientes pasos:

I. Realizar una búsqueda y obtención de una base de datos (fuente secundaria) de fraude con tarjetas de créditos
II. Descargar la base de datos (Dataset)
III. Hacer un análisis preliminar de los datos para detectar errores en la muestra.
IV. Depurar la muestra, eliminando aquellos registros que contengan errores o estén incompletos.
V. Convertir aquellas variables categóricas en variables booleanas (0, 1), según la presencia de cada categoría por registro.
VI. Hacer una elección preliminar de las variables relevantes
VII. Hacer una partición de la muestra en conjuntos de datos de entrenamiento y de prueba.
VIII. Hacer un análisis exploratorio de los datos
IX. Aplicar los siguientes algoritmos de clasificación: regresión logística, k-vecinos cercanos, árboles de decisión y bosques aleatorios
X. Crear modelos de predicción y clasificación de las observaciones
XI. Realizar la validación cruzada
XII. Medición de las ratios de eficiencia de predicción
XIII. Construir matrices de confusión
XIV. Graficar curvas ROC
XV. Comparar la eficiencia entre los modelos
XVI. Conclusiones
XVII. Programación del código en Python

3.2. Muestra (Dataset)

La muestra (dataset) seleccionada la he descargado desde el repositorio Kaggle. Concretamente, se trata de una muestra con datos de fraude de tarjetas de crédito recogidos en Nigeria con el título de *Credit Card Fraud Detection Dataset (Emmanuel Onwuegbusi)* [9].

En África existe un gran problema de fraude con el uso de tarjetas de crédito, donde las incidencias más frecuentes se dan a través de las transacciones de pago por Internet y por aplicaciones móviles. También en los retiros en cajeros automáticos. [10]

A primera vista podemos ver que la proporción entre fraude y no fraude es de aproximadamente poco menos de dos tercios contra poco más de un cuarto respectivamente.

Categoría	Observaciones	Proporción
1	27370	73,78 %
0	9727	26,22%

Tabla 2: Porcentaje de categorías

El dataset contiene las siguientes variables:

- Account Number: tipo numérico
- CVV (Card Verification Value): tipo numérico
- Customers age: tipo numérico (num)
- Gender: tipo carácter (char)
- Marital Status: tipo carácter (char)
- Card color: tipo carácter (char)
- Card type: tipo carácter (char)
- Domain: Domestic, International: tipo carácter (char)
- Amount: tipo numérico (num)
- Average income expenditure: tipo numérico (num)
- City: tipo carácter (char)
- Outcome: tipo booleano

3.3. Depuración de datos

Se ha efectuado una depuración de los registros con datos ausentes (NaN). Number of NaN values: 8851

3.4. Redenominación de variables

Con la finalidad de abreviar las nomenclaturas, he rebautizado algunas variables:

- CustomerAge = Age
- Gender = Gen
- Marital Status = Status
- CardColour = Color
- CardType = Type
- Domain = Area
- AverageIncomeExpendicture = AIE
- Customer_City_Address = City

3.5. Conversión de variables categóricas a ficticias (dummies)

He convertido las variables categóricas en variables ficticias, tal que puedan tomar valores (1,0)

- Gen_Female, Gen_Male,
- Status_Divorced, Status_Married, Status_Single, Status_Unknown,
- Color_Gold, Color_White,
- Type_MasterCard,Type_Verve, Type_Visa,
- Area_International, Area_Local
- City_Abuja, City_Enugu, City_Ibadan, City_Kano, City_Lagos, City_Ota, City_Other,City_Port Harcourt

3.6. División de la Muestra

Si partimos la muestra en conjuntos de datos de entrenamiento y de prueba, es posible que el modelo de clasificación que primero hayamos ajustado con los datos de entrenamiento, no se ajuste bien a los datos de prueba. Esta discrepancia podría darse debido un problema de sobreajuste (overfitting).

Para evitar un posible problema de sobreajuste y optimizar el ajuste con los datos de prueba, se recomienda hacer una validación cruzada. [11]

3.7. Análisis exploratorio

Se ha efectuado un análisis exploratorio de los datos, principalmente cruzando las variables explicativas con la variable a explicar (outcome) con un análisis de frecuencias e histogramas.

Observamos diferencias significativas en cuanto al género. La frecuencia mayor de fraude se da en el género masculino.

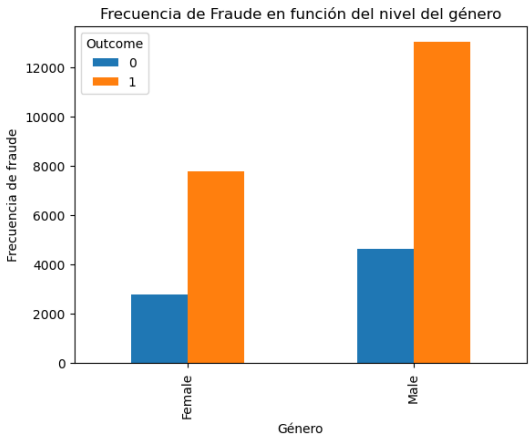

Figura 1: Frecuencia de fraude por género

Observamos diferencias significativas entre los casados, solteros y divorciados. La frecuencia mayor se da en los casados y solteros

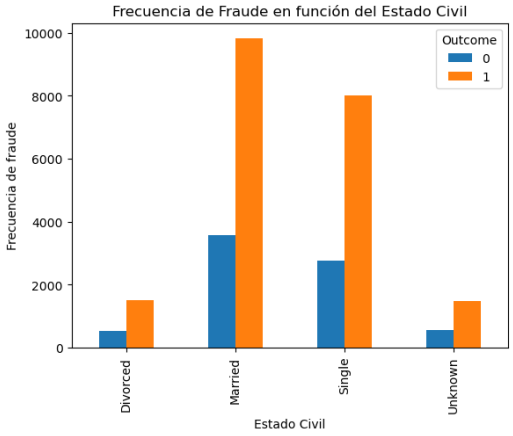

Figura 2: Frecuencia de fraude por estado civil

Observamos diferencias significativas entre Master Card, Verte, Visa. La frecuencia mayor de fraude se da con la tarjeta Verte.

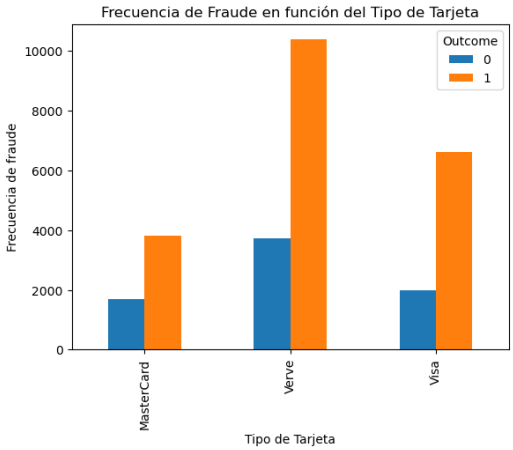

Figura 3: Frecuencia de fraude por género

Observamos una incidencia mayor de fraude agrupada en el segmento de usuarios jóvenes (entre los 18 años y 32).

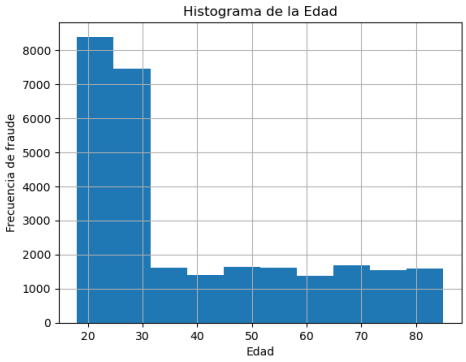

Figura 4: Frecuencia de fraude por edad

3.8. Selección de variables

He aplicado la función Recursive Feature Elimination (RFE) de la biblioteca scikit-learn de Python para efectuar la selección de variables relevantes.

Dado un estimador externo que asigna pesos a las características, el objetivo es seleccionar variables considerando recursivamente conjuntos de características cada vez más pequeños.
Primero, el estimador se entrena en el conjunto inicial de variables y la importancia de cada variable se obtiene a partir de cualquier atributo específico. Posteriormente las características menos importantes se eliminan [12]

3.9. Validación cruzada

Esta técnica sirve para evaluar los resultados del análisis estadístico y garantizar que sean independientes de la partición que se ha hecho entre el conjunto de entrenamiento y el conjunto de prueba.

Primero definimos una partición de la muestra en k partes (k-fold), tal que cada una sea utilizada como conjunto de prueba en cada iteración. De esta manera todas las particiones serán utilizadas. Mientras cada partición es utilizada como datos de prueba, las particiones restantes (k-1) serán utilizados como datos de entrenamiento [13]

- k-subconjuntos
- k-fold = 10 iteraciones.

Por tanto, cada partición sea utilizada como datos de prueba al menos una vez. O sea, cada observación es parte del conjunto de entrenamiento y de prueba.

4. Algoritmos de clasificación

Voy a generar 4 modelos de clasificación con: regresión logística, K-vecinos más próximos, Árbol de decisión y bosques aleatorios.

4.1. Regresión Logística

Crear un modelo de regresión logística con la función de Newton-Raphson

Model:	Logit	Method:	MLE
Dependent Variable:	Outcome	Pseudo R-squared:	0.468
Date:	2023-08-26 00:07	AIC:	17335.8268
No. Observations:	28246	BIC:	17418.3138
Df Model:	9	Log-Likelihood:	-8657.9
Df Residuals:	28236	LL-Null:	-16278.
Converged:	1.0000	LLR p-value:	0.0000
No. Iterations:	8.0000	Scale:	1.0000

Tabla 2: Parámetros de la regresión logística

- Pseudo R-Squared: 0.468
- Converged = 1.0000 ha convergido
- Iteraciones = 8
- Grados de Libertad = 10 – 1 = 9
- Log-Likelihood: logaritmo natural
- No consideramos AIC (Akaike Information Criterion), BIC (Bayes Information Criterion)

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
Age	-0.0009	0.0009	-0.9249	0.3550	-0.0027	0.0010
Amount	0.0000	0.0000	77.6742	0.0000	0.0000	0.0000
AIE	0.0000	0.0000	30.1271	0.0000	0.0000	0.0000
Gen_Female	0.0052	0.0395	0.1311	0.8957	-0.0722	0.0825
Status_Divorced	0.0094	0.0730	0.1287	0.8976	-0.1338	0.1525
Status_Unknown	-0.0887	0.0734	-1.2090	0.2267	-0.2325	0.0551
Color_Gold	-5.3234	0.1025	-51.9591	0.0000	-5.5242	-5.1226
Color_White	-3.6087	nan	nan	nan	nan	nan
Type_MasterCard	-2.2180	nan	nan	nan	nan	nan
Type_Visa	-1.3907	nan	nan	nan	nan	nan
Area_International	0.4126	0.0421	9.7936	0.0000	0.3300	0.4951

Tabla 3: Parámetros de la regresión logística

Coef: coeficientes de las variables independientes

P>|z| estadístico de Wald. A menor valor de P, mayor significación. Así podemos determinar qué variables explicativas del modelo contribuyen de manera significativa en la clasificación.

Variables con p bajo tienen una relevancia alta:
Importe (Amount)
Ingreso promedio (AIE)
Tarjeta Dorada (Color_Gold)

Variables con p muy alto tienen poca relevancia:
Género femenino (Gen_Female)
Estado civil divorciado (Status Divorced)

Aplicamos el algoritmo linear_model.Logistic Regression() [14]

	0	1
0	Age	[-3.151068852977285e-09]
1	Amount	[5.599974875599324e-06]
2	AIE	[-5.575895760719262e-06]
3	Gen_Female	[-2.9392895636177286e-11]
4	Status_Divorced	[-5.545901847498879e-12]
5	Status_Unknown	[-7.1020492060009655e-12]
6	Color_Gold	[-4.0187771256429226e-11]
7	Color_White	[-3.876842628060737e-11]
8	Type_MasterCard	[-3.036483886184219e-11]
9	Type_Visa	[-8.403587418765382e-12]
10	Area_International	[-4.009565032320506e-11]

Tabla 4: Estimación de coeficientes

La variable de los coeficientes indica los cambios en escala logarítmica por cada unidad de cambio en la variable. Entonces podría interpretar que:

- Si el importe (amount) aumenta en una unidad, el logaritmo del cociente de probabilidades del fraude se incrementará en 5.60
- Si el Ingreso promedio (AIE) aumenta en una unidad, el logaritmo del cociente de probabilidades del fraude disminuirá en 5.57.
- Si la proporción de divorciados aumenta en una unidad, el logaritmo del cociente de probabilidades del fraude disminuirá en 5.54.
- Si el uso de tarjeta visa aumenta en una unidad, el logaritmo del cociente de probabilidades del fraude disminuirá en 8,40.

Clasificación con vector de probabilidades

La función de estimación de máxima verisimilitud (MLE) calcula la probabilidad condicionada de observar un resultado (1, 0)

Primera Columna = 0, Segunda Columna = 1

```
array([[0.20870786, 0.79129214],
       [0.22702206, 0.77297794],
       [0.48740507, 0.51259493],
       ...,
       [0.58653583, 0.41346417],
       [0.02781027, 0.97218973],
       [0.35051264, 0.64948736]])
```

Probabilidad del valor de salida: fraude, no fraude.

Probabilidad estándar = 0.5

Fraude = 0	Fraude = 1	Clasificación
20,87%	79,13%	1
22,70%	77,3%	1
48,74%	51,26%	1
…	…	…
58,65%	41,35%	0
2,78%	97,22%	1
35,05	64,95%	1

Tabla 5: Porcentaje de categorías

Si > 0.5 → la observación se clasifica como resultado positivo (fraude)
Si < 0.5 → la observación se clasifica como resultado negativo (no fraude)

Validación del modelo de regresión logística

Estimación de la ratio de eficiencia para el conjunto de datos de prueba, obtenemos una ratio de eficiencia de 0.807. O sea, 80,76%.

Matriz de confusión con regresión logística

La clasificación obtenida con las muestras de entrenamiento y de validación

actual	0	1
prediction		
0	1032	464
1	1166	5812

Tabla 6: Matriz de confusión de logit

- TP (verdaderos positivos) = 5812
- TN (verdaderos negativos) = 1032
- FN (falsos positivos) = 1166
- FP (falsos negativos) = 464
- Ratio de precisión = (TP + TN) / (TP +TN +FP + FN) = (5812 + 1032) / 8474 = 80,76%

4.2. K-Nearest Neighbors (KNN)

Para el método de K-vecinos más próximos, he aplicado el algortimo neighbors.KNeighborsClassifier [15].
Tenemos 2 grupos (clusters) a priori: fraude y no fraude

El algoritmo sigue los siguientes pasos:

- Encontrar los k puntos más cercanos
- Comparar cada uno de los grupos que existan en el conjunto de datos contra lo que quiera clasificar
- Crear una lista con las distancias seguidas de la clase (cluster) para cada uno de los puntos de la muestra.
- Luego ordenar la lista por distancia más cercana y tomar los primeros k-valores de la lista para encontrar el más popular y obtener una clasificación.

Validación del Modelo de predicción

Dividimos la muestra en un conjunto de entrenamiento y otro de prueba.

Con el conjunto de datos de prueba, obtenemos una ratio de eficiencia de 0.843. O sea, un 84,29%.

Matriz de confusión con KNN

La clasificación obtenida con las muestras de entrenamiento y de validación

Predictions	0	1
Actual		
0	1592	606
1	725	5551

Tabla 7: Matriz de confusión KNN

- TP (verdaderos positivos) = 5551
- TN (verdaderos negativos) = 1592
- FN (falsos positivos) = 606
- FP (falsos negativos) = 725

4.3. Árboles de decisión

Crear un modelo para clasificar una observación en fraudulenta o no fraudulenta en función de las variables explicativas.

Mediante este algoritmo se van generando los nodos de decisión, que también suelen tener subnodos que sirven para "afinar" la predicción del nodo anterior hasta alcanzar un nodo que no se divida. Este último nodo se le conoce como nodo hoja o *leaf node*.

El nodo hoja debe ser lo más homogéneo posible, pero evitando problemas de overfitting.

Aplicamos el algoritmo DecisionTreeClassifier para estimar el modelo [16]

Validación Cruzada para la poda (Prunning)

- Fijamos número de particiones (k-folds) = 10
- Profundidad máxima (max_depth) = 5. Es el número máximo de profundidad de árboles
- Corte mínimo (min_samples_split) = 20. Es el número mínimo de muestras para dividir un nodo hoja en un sub-nodo
- El número mínimo de muestras (min_samples_leaf) = 5 para convertirse en una hoja nodo Visualización del árbol de decision

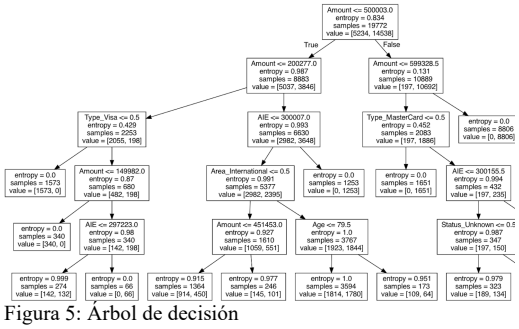

Figura 5: Árbol de decisión

- Nivel 1: se parte en 2 nodos en función del criterio de partición Importe (Amount) < = 500.000 a la izquierda y > 500.000 a la derecha
- Nivel 2: obtenemos 2 nodos en función del Importe (Amount) <= 220.277 para el nodo izquierdo y de <= 599.328,5 para el nodo derecho con un valor muy bajo de entropía
- Nivel 3: obtenemos 4 subnodos por cada uno de los 2 anteriores. Para el primer subnodo, en el lado izquierdo el criterio de partición es la tarjeta Visa con una probabilidad del 50%, pero un valor de entropía mucho más bajo que su par. Para el Segundo subnodo el criterio de partición es el Ingreso promedio (AIE) <= 300.007
- Nivel 3: subnodo al lado derecho, el criterio. De partición es la tarjeta Master Card con una probabilidad <= 50% y con un valor bajo de entropía. El cuarto subnodo se convierte en un nodo hoja (leaf node).
- En el cuarto nivel, aparece un nuevo criterio de partición por área internacional o procedencia extranjera de la tarjeta con una probabilidad menor igual al 50%
- En el quinto nivel, aparece un nuevo criterio de partición por la edad <= 79.5

Validación del modelo de árboles de decisión

Estimación de los puntajes de eficiencia de la predicción (scores) en función de la profundidad del árbol. Probando hasta 10 niveles de profundidad (1,2,3,4,5,6,7,8,9,10):

- Score para i = 1 es de 0.795
- Score para i = 2 es de 0.829
- Score para i = 3 es de 0.858
- Score para i = 4 es de 0.863 es el score máximo
- Score para i = 5 es de 0.863 es el score máximo

…

Criterios más importantes de partición:

- Amount = Importe
- AIE = Ingreso promedio
- Type Visa
- Type Master Card

Si decidimos que el árbol solo crezca hasta 4 o 5 niveles de profundidad (i = 4 o 5) nos va a dar la clasificación óptima.

Matriz de confusión con árboles de decisión

La clasificación obtenida con las muestras de entrenamiento y de validación

Predictions	0	1
Actual		
0	1610	588
1	641	5635

Table 8: Matriz de confusión de árboles

- TP (verdaderos positivos) = 5635
- TN (verdaderos negativos) = 1810

- FN (falsos positivos) = 588
- FP (falsos negativos) = 641

4.4. Bosques aleatorios (Random Forest)

Aplicamos el algoritmo RandomForestClassifier para estimar el modelo de bosque aleatorios [17]

Creamos un modelo para clasificar una observación en fraudulenta o no fraudulenta en función de las variables explicativas.

Matriz de Decisiones

```
array([[1.        , 0.        ],
       [0.        , 1.        ],
       [0.65714286, 0.34285714],
       ...,
       [0.71794872, 0.28205128],
       [0.74358974, 0.25641026],
       [1.        , 0.        ]])
```

Cada uno de ellos ha decidido con que probabilidad caemos dentro de cada categoría:

Primer árbol: [1, 0] → 1 fraude
Segundo árbol: [0, 1] → no fraude
Tercer árbol: 65,7% probabilidad de fraude contra 34,3% de no fraude
……
un empate 0.5, 0.5 no es bueno.

Validación del modelo

Aplicamos OBB (out-of-bagging) para medir el error de predicción de bosques aleatorios para estimar las ratios de eficiencia de la predicción en función del número de árboles

- número de árboles = 10, score = 0.855
- número de árboles = 50, score = 0.856
- número de árboles = 100, score = 0.855
- número de árboles = 500, score = 0.857
- número de árboles = 1000, score = 0.854

Grado de eficiencia con 5 niveles = 85,71%.

Podemos ver que con 500 árboles alcanzamos el máximo puntaje. Sin embargo, la diferencia entre 10, 50, 100, 500, 1000 árboles es pequeña.

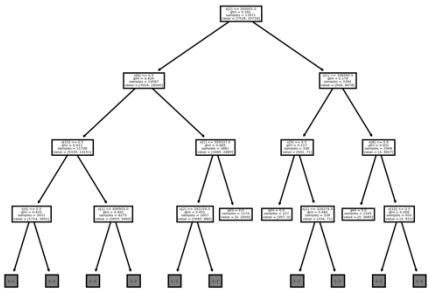

Figura 6: Bosques aleatorios

Matriz de confusión con bosques aleatorios

La clasificación obtenida con las muestras de entrenamiento y de validación

Predictions	0	1
Actual		
0	2198	0
1	0	6276

Tabla 9: Matriz de confusión bosques

4.5. Curvas ROC

La curva ROC (receiving operating characteristic) es un gráfico que ilustra la capacidad de diagnóstico de un sistema clasificador binario a medida que varía su umbral de discriminación.

La curva ROC es la gráfica de la tasa de verdaderos positivos (TPR) frente a la tasa de falsos positivos (FPR), en diferentes niveles de un umbral.

La curva ROC mide la sensibilidad (Recall)

- N.º de positivos reales = TP + FN
- N.º de negativos reales = TN + FP
- N.º de predicciones correctas = TP + TN
- N.º de predicciones incorrectas = FP + FN

Estimación de la sensibilidad y la especificidad

- Sensibilidad = (TPR) = TP / (TP + FN)
- Especificidad = (TNR) = TN / (TN + FP)]
- FPR: (False-Postive Rate) = 1 – Especificidad

Probar diferentes umbrales para generar diferentes valores de sensibilidad y especificidad:

Vector de umbrales = [0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70]

Curva ROC representa FPR = 1 – Especificidad

```
1 sensitivities
```

```
[1.0,
 0.9961759082217974,
 0.9837476099426387,
 0.9633524537922243,
 0.9260675589547482,
 0.8738049713193117,
 0.8153282345442957,
 0]
```

```
1 especifities_1
```

```
[0.9963603275705186,
 0.9367606915377616,
 0.8548680618744313,
 0.7106460418562329,
 0.5304822565969063,
 0.4017288444040037,
 0.2725204731574158,
 0]
```

Table 10: sensibilidades y especificidades

Con estos valores podemos generar las curvas ROC

Gráfico de la curva ROC

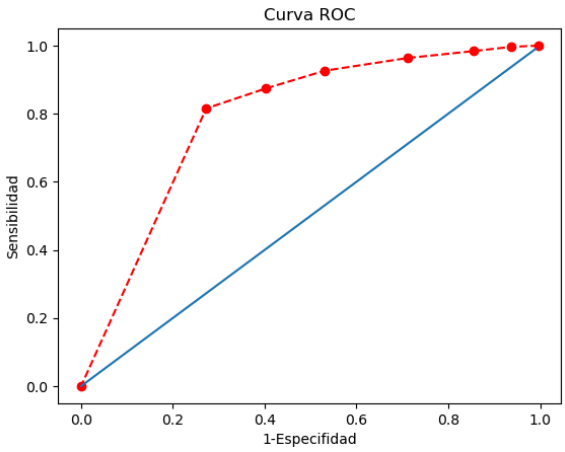

Figura 7: Curva ROC

Dado que la curva se acerca al extremo superior izquierdo, nos indica que el modelo es un buen predictor.

- La diagonal (Benchmarking) representa el 50%
- La diagonal es una decisión totalmente aleatoria
- La diagonal es el peor modelo que existe
- Cualquier cosa que esté encima de la diagonal es un predictor mejor que una elección totalmente aleatoria (cara o cruz)

Área debajo de la curva

Compute Area Under the Curve (AUC) = 0.877

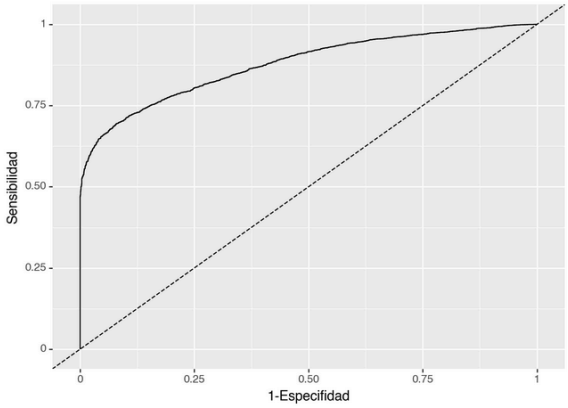

Figura 8: Curva ROC

- Incrementar la sensibilidad → reduce la especificidad
- Cuanto mayor se ajuste la curva (tangente) al borde superior izquierdo → mayor es la predicción del modelo
- Cuanto más cerca de la diagonal → peor es el modelo
- Cuanto mayor sea el área bajo la curva → mejor será la predicción

4.6. Aplicación Práctica

Por ejemplo, una institución financiera nos encarga un proyecto para implementar un sistema de clasificación de pagos con de tarjeta de crédito.
Si la institución financiera conoce un % promedio de fraude a priori, podemos utilizar dicha probabilidad como un umbral de probabilidades.

Establecer un umbral de probabilidades tal que estas probabilidades sean clasificadas en 2 categorías:

Si la probabilidad < umbral → se considera como 0
Si la probabilidad > umbral → se considera 1.

Por ejemplo, la institución financiera supone de antemano que el 10% de los pagos con tarjeta de crédito son fraudulentos (p = 1/10)

Si una observación tiene más del 10% de probabilidad de ser clasificada como fraude, entonces se le puede calificar como posible fraude.

Si > 10%, entonces clasificar como fraude = 1
Si < 10%, entonces clasificar como no fraude = 0

	0	prediction
0	0.791292	1
1	0.772978	1
2	0.512595	1
3	0.682343	1
4	0.985585	1

Tabla 9: Matriz de probabilidades

Si hiciéramos una clasificación con este umbral, tendríamos una matriz de confusión muy exagerada

col_0	count
prediction	
0	8
1	8466

Tabla 10: Matriz de confusión

Predicción = 99.90%

Vemos que con un umbral del 10%, el modelo de regresión logística clasifica a más del 99% de observaciones de la muestra como fraudulentas.

Si aumentamos dicho umbral a un 75%, la predicción decrece a 62.95%. O sea, que con un umbral del 75%, el modelo clasifica a más del 62% de observaciones de la muestra como fraudulentas.

col_0	count
prediction	
0	3139
1	5335

Tabla 11: Matriz de confusión

5. Análisis Resultados

5.1. Eficiencia de Predicción (scores)

- El modelo de regresión logística obtenemos una ratio de eficiencia de un 80,76%.
- El modelo de KNN obtenemos una ratio de eficiencia de un 84,29%.
- Con el modelo de árboles de clasificación obtenemos una ratio de eficiencia para i = 4 (hasta 4 niveles de profundidad) de 86,30%
- Con el modelo de clasificación de bosques aleatorios con 500 árboles podemos obtener un porcentaje de eficiencia es de un 85,71%.

5.2. Matrices de confusión

Concatenamos las matrices de confusión con las clasificaciones de predicción de los modelos generados por cada algoritmo

	0	1
0	1599	1159
1	599	5117
0	1592	606
1	725	5551
0	1610	588
1	641	5635
0	2198	0
1	0	6276

Table 12: sensibilidades y especificidades

Si comparamos los valores reales con las predicciones hechas por los algoritmos de clasificación utilizando la muestra de prueba, vemos que los porcentajes de verdaderos negativos y verdaderos positivos son muy similares:

- el número de casos positivos clasificados correctamente (verdaderos positivos) oscilan entre 60,38%, 65,50%, 66,49% respectivamente.
- el número de casos negativos clasificados correctamente (verdaderos negativos) oscilan entre 18,86%, 18,78%, 18,99% respectivamente.

Sin embargo, vemos que las tasas de clasificación de predicción por el algoritmo de bosques aleatorios con respecto a los valores reales, es mucho mejor:

- Número de casos positivos clasificados correctamente (verdaderos positivos) = 25,93%

- Número de casos negativos clasificados correctamente (verdaderos negativos) = 74,07%

6. Conclusiones

Como el objetivo general de este trabajo es analizar datos de transacciones de pago con tarjetas de crédito, aplicando una metodología con modelos de clasificación; he comprobado cómo funcionan los algoritmos de clasificación y medido su eficiencia en cada modelo de predicción obtenido (regresión logística, KNN, árboles de decisión, bosques aleatorios)

El hecho de que los porcentajes de eficiencia en la predicción son muy aproximados entre los algoritmos, -es en mi opinión- un indicio de que los modelos de predicción son consistentes.

Asimismo, el hecho de que los porcentajes de clasificación de verdaderos positivos y verdaderos negativos sean muy aproximados, también reflejan de que los modelos de predicción son consistentes y eficientes. Siendo el modelo predicción de bosques aleatorios el más preciso.

Los coeficientes de regresión indican los cambios en escala logarítmica por cada unidad de cambio de las variables. Por ejemplo, a mayor importe de pago, mayor probabilidad de fraude. Por el contrario, a mayor ingreso promedio, menor probabilidad de fraude. A mayor proporción de tarjeta visa, menor probabilidad de fraude. A mayor frecuencia de divorciados menor probabilidad de fraude.

Los criterios de clasificación más importantes son el importe, el gasto promedio, la tarjeta visa y master.

En función de los resultados obtenidos, podemos intuir que los métodos de regresión logística, K-vecinos más próximos, árboles de decisión y bosques aleatorios son algoritmos óptimos para clasificar las observaciones (transacciones de pago con tarjeta de crédito) en fraudulentas y no fraudulentas.

7. Referencias

[1] Informe 2022 Payment Threats and Fraud Trends Report (23.11.2022) publicado por el European Payments Council. (htt1)
[2] [5] []Fraud Detection in Credit Cards using Logistic Regression (2020), autores Hala Z Alenzi, Nojood O. Aljehane (htt2)
[3] [4] A Review on Credit Card Fraud Detection using Machine Learning (October 2019). International Journal of scientific & technology research (htt3)
[6] Fraud Detection of Credit Card Using Logistic Regression. June 2022 (Mohammed Nasser Hussain, Maram Sai Charan Reddy). (htt4)
[7] Payam Refaeilzadeh, Lei Tang, Huan Lui: Cross-Validation, 2009 (Arizon State University) (htt11)
[8] [11] Credit card fraud detection using machine learning: A survey (octubre, 2020) Yvan Lucas (INSA Lyon) y Johannes Jurgovsky (Universidad de Pasau, Alemania) (htt5)
[9] Credit Card Fraud Detection Dataset, May 2023 (Emmanuel Onwuegbusi) (htt6)
[10] Business Day: Cashless policy traps Nigerians between fraud, failed transactions. December 2022 (htt7)
[12] Función sklearn.feature_selection.RFE
[13] Fundamentals of Machine Learning for Predictive Data Analysis. Algorithms, Worked Examples, and Case Studies. John D. Kelleher, Brian Mac Namee, Aoife D'Arcy. 2nd Edition. The MIT Press Cambridge, Massachusetts (2015)
[14] Función linear_model.LogisticRegression() de Scikit-Learn (Python) (htt8)
[15] Función neighbors.KNeighborsClassifier Scikit-Learn (Python) (htt12)
[16] Función sklearn.tree / DecisionTreeClassifier (Python) (htt10)
[17] Función Algoritmo RandomForestClassifier slearn.tree (Python) (htt9)

Apéndice A.

Se adjuntan como apéndice la siguiente documentación:

Código de programación en Python:

https://github.com/jlazoros-uib/UIB-MADM-TFM-Open/blob/main/TFM%20Deteccion%20Fraude%20v0.4.ipynb

Muestra (dataset):

https://github.com/jlazoros-uib/UIB-MADM-TFM-Open/blob/main/credit%20card%20fraud%20v.01.csv

Árbol de Decisión:

https://github.com/jlazoros-uib/UIB-MADM-TFM-Open/blob/main/TFM%20A%CC%81rboles%20de%20Clasificacio%CC%81n.pdf