PwC Challenge

Master en Business Analytics

Grupo 4:

Jerika Alejandra Castillero Latorraca
Noemi Luengo Crisóstomo
Jorge Enrique Morales Mercado
Salomea Maria Stepien Slomba

Máster: Business Analytics

ÍNDICE DEL INFORME PROPUESTA SOLUCION RETO PwC

Predicción del Fraude en Tarjetas de Crédito – Reto PwC..... 3
Introducción..... 3
Objetivo..... 3
Pasos para Predecir Fraude en Tarjetas de Crédito..... 3
A. Adquisición de Datos..... 3
B. Análisis Exploratorio de Datos (EDA)..... 5
Análisis de la variable isFraud..... 6
Atributos no relevantes..... 8
Identificación de patrones y conclusiones..... 8
C. Ingeniería de Características..... 9
D. Preparación de Datos para Modelado..... 11
E. Selección y Entrenamiento del Modelo..... 11
F. Evaluación del Modelo..... 14
G. Validación y Optimización del Modelo..... 16
H. Despliegue y Monitoreo..... 19
Repositorio del código..... 19
Conclusiones..... 19
Consideraciones Finales..... 19

Predicción del Fraude en Tarjetas de Crédito – Reto PwC

Introducción

La detección de fraude en transacciones con tarjetas de crédito es un desafío constante para las instituciones financieras. El uso de técnicas de machine learning puede ser fundamental para identificar patrones y anomalías en los datos que podrían indicar actividades fraudulentas.

Objetivo

Este informe describe los pasos esenciales, que realizaremos sobre el dataset (ver punto A) para predecir el fraude en transacciones con tarjetas de crédito utilizando técnicas de machine learning con diferentes herramientas, las mismas que se justifican por la premura en la entrega de presente informe, ;entre otras:

- PowerBI
- RapidMiner
- BigQuery
- GCP
- Firebase

Pasos para Predecir Fraude en Tarjetas de Crédito

A. Adquisición de Datos

Obtención de datos:

Nos entregan un dataset con transacciones de tarjetas de crédito, no hay fechas en las transacciones, incluyen características como cantidad, tipo de transacción, entre otros.

Descripción del dataset original del reto PwC

a) Nombre: Reto 2 - BA - DataSet.csv
b) Tipo: CSV
c) Observaciones : 6.362.620 observaciones
d) Atributos: 11 atributos.

En la siguiente figura se pueden observar los atributos junto con el tipo de variable.

Name	Type	Missing
amount	Real	0
oldbalanceOrg	Real	0
newbalanceOrig	Real	0
oldbalanceDest	Real	0
newbalanceDest	Real	0
step	Integer	0
isFlaggedFraud	Integer	0
Label isFraud	Nominal	0
type	Nominal	0
nameOrig	Nominal	0
nameDest	Nominal	0

Figura 1. Listado de variables y su tipo.

Antes de llevar a cabo el análisis de los datos, realizamos una limpieza preliminar de los mismos. Para este preprocesamiento planeamos eliminar datos duplicados, datos faltantes (null), transformar variables si fuese necesario (normalización, codificación de variables categóricas) y eliminar posibles atributos correlacionados.

En el dataset proporcionado, no se encuentra ningún duplicado ni ningún valor vacío, por lo que no es necesario realizar ningún paso para eliminarlos.

Sin embargo sí se pueden encontrar correlación entre atributos.

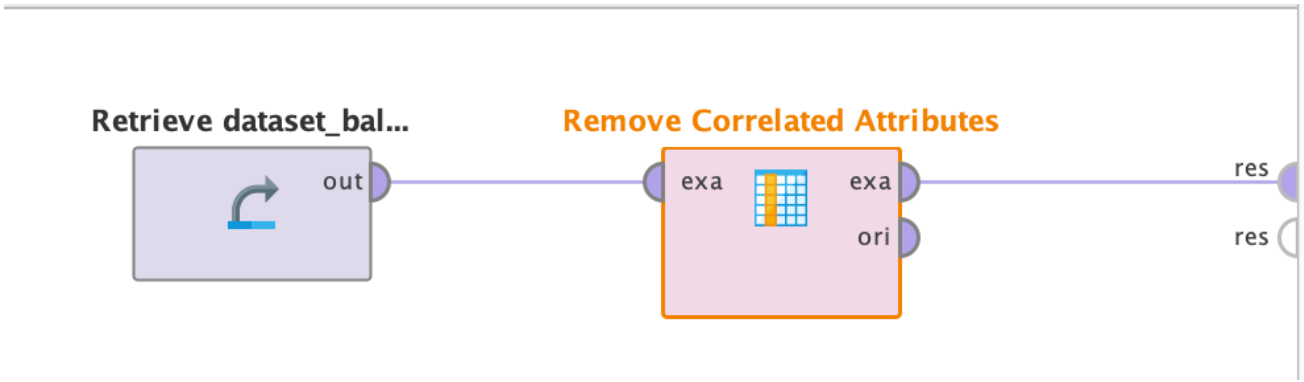

Figura 2. Eliminación de atributos correlacionados en Rapidminer

En la figura anterior se puede apreciar cómo sobre el dataset original aplicamos un operador en Rapidminer para eliminar los atributos correlacionados.

B. Análisis Exploratorio de Datos (EDA)

Después del preprocesamiento de los datos, procedemos a su análisis.

Para empezar vemos la distribución de las transacciones del dataset dado según el tipo (5 categorías).

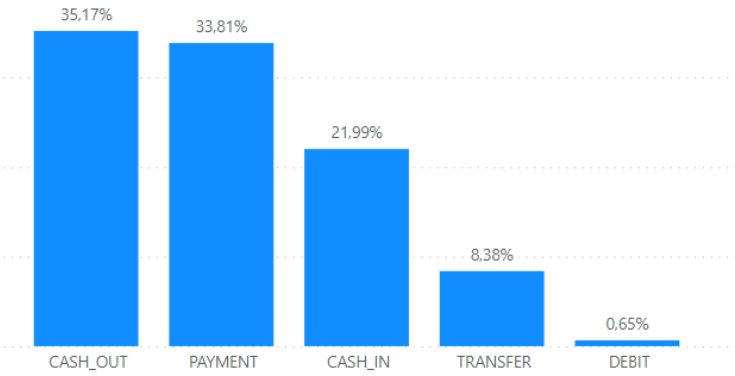

Figura 3. Distribución de las transacciones según la variable *type*.

En la gráfica de barras observamos que el 90,97% de los movimientos corresponden a ***cash_out, payment y cash_in.*** Esto supone una variación respecto a las estimaciones previas al EDA, ya que cash_out tiene más peso del previsto. La distribución de tipos de operaciones financieras es más cercana al de una cuenta bancaria que una tarjeta.

Como ya se mencionó en el preprocesado, no hay datos duplicados y nulos.

A continuación observamos algunos valores estadísticos sobre el dataset. Como podemos observar en la última columna el coeficiente de variación es muy elevado en todos los tipos. Podemos observar entonces que en el dataset obtenido algunas observaciones tienen valores atípicos, outliers, muy dispersos, que deben eliminarse o afectarán a valores estadísticos como la media, y al posterior modelo.

type	Promedio	Mín.	Máx.	Desv std	Coef_var
TRANSFER	$910.647,01	$2,60	$92.445.516,64	$1.879.571,77	206,40 %
CASH_OUT	$176.273,96	$0,00	$10.000.000,00	$175.329,71	99,46 %
CASH_IN	$168.920,24	$0,04	$1.915.267,90	$126.508,21	74,89 %
PAYMENT	$13.057,60	$0,02	$238.637,98	$12.556,45	96,16 %
DEBIT	$5.483,67	$0,55	$569.077,51	$13.318,37	242,87 %
Total	**$179.861,90**	**$0,00**	**$92.445.516,64**	**$603.858,18**	**335,73 %**

Figura 4. Datos estadísticos del dataset

Para este análisis se puede utilizar el operador Statistics en Rapidminer, para obtener un primer vistazo rápido sobre los valores del dataset.

Análisis de la variable isFraud

La variable principal en este dataset es isFraud.

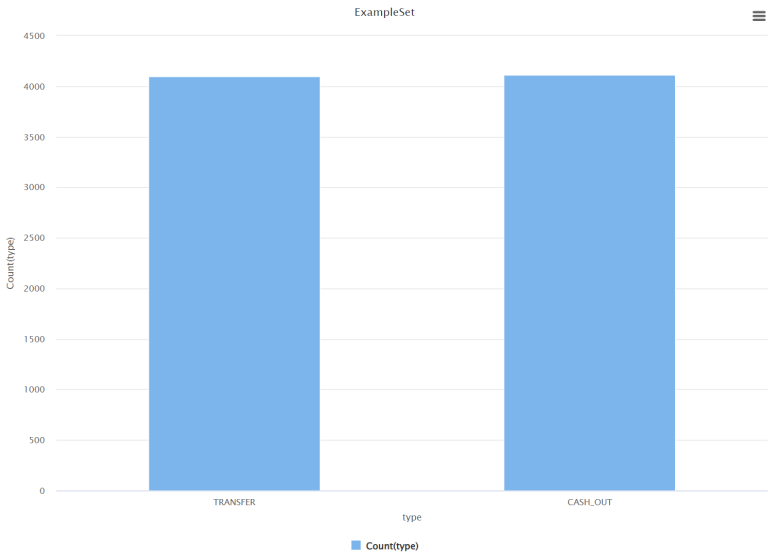

Figura 5. Distribución de isFraud según el atributo type

Como se puede visualizar en la figura anterior, solo hay observaciones marcadas como fraude en los tipos de transfer y cash_out. Por lo tanto, todas las observaciones de cash_in, debit y payment están marcadas como no fraude. Estos datos también suponen un cambio frente a las estimaciones previas al EDA, ya que la mayor parte de los fraudes suelen ocurrir en comercio electrónico, que se incluiría en el type payments, pero en este caso, no cuenta con ninguna observación marcada como fraude.

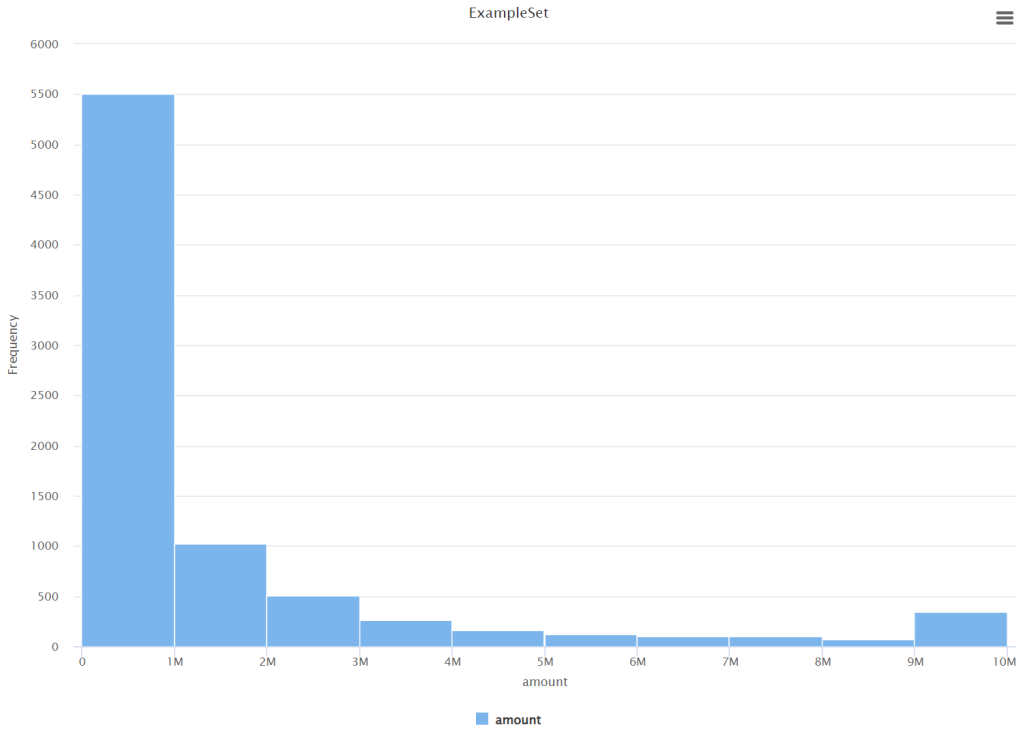

Figura 6. Distribución de amount en los datos con isFraud positivo

En la gráfica de barras anterior se puede observar como un gran número de las observaciones marcadas como fraude tienen un amount de 0. Esto indica que es necesario limpiar aún más los datos para eliminar las transacciones con amounts en 0, ya que no consideramos que no deben estar incluidas como fraude al no haber dinero en la acción, y puede desequilibrar los análisis estadísticos, y posteriormente al utilizar el modelo, generar un sesgo.

Atributos no relevantes

Al observar otros atributos más allá de isFraud, analizamos la relevancia de isFlaggedFraud. La cantidad de observaciones con isFlaggedFraud positiva es ínfima, 16 observaciones de un total de 6 millones, por lo que podemos eliminar este atributo al no aportar información relevante.

Identificación de patrones y conclusiones

Tras visualizar algunos valores estadísticos, y la información sobre las observaciones marcadas como fraude, procedemos a ver algunos datos en concreto. A simple vista se pueden observan algunos patrones, como por ejemplo:

- Todas las transacciones en cero están marcadas como isFraud
- Todas las transacciones que tienen el mismo valor pero la variable type es diferente están marcadas como fraude

type	nameDest	nameOrig	amount	isFraud
CASH_OUT	C958479953	C1861878353	$63,80	1
TRANSFER	C1368130863	C1293504491	$63,80	1
CASH_OUT	C2102058838	C1584512618	$119,00	1
TRANSFER	C1480876722	C1995557473	$119,00	1
CASH_OUT	C1518370196	C773613907	$119,65	1
TRANSFER	C543477940	C1497766467	$119,65	1
CASH_OUT	C517676411	C790340353	$151,00	1
TRANSFER	C315826176	C1172437299	$151,00	1
CASH_OUT	C1769947269	C1173659886	$164,00	1
TRANSFER	C2119910556	C1455969984	$164,00	1
CASH_OUT	C200064275	C1065370362	$170,00	1
TRANSFER	C26381896	C553034695	$170,00	1
CASH_OUT	C1009545186	C722054736	$174,92	1
TRANSFER	C380623487	C1554022122	$174,92	1

Figura 7. Identificación de patrones

Durante el EDA detectamos los siguientes puntos:

- La información del dataset se encuentra sesgada, dado que de los 6.236.600 registros, solo 8.213 (0,13%) están marcados como fraudes, lo cual para poder llevar a cabo un muestreo y modelado se debe balancear.
- El dataset no tiene campo de fecha, lo cual no permite tener una correlación con la variable isfraud que permita un mejor entrenamiento y precisión del modelo.
- Se tienen transacciones con valores en cero identificadas como fraude, lo cual nos indica que hay que realizar limpieza de los datos.
- Se observa un alta desviación de los datos respecto al promedio, lo cual se identifica al calcular el coeficiente de variación de la variable type. También es necesario limpiar los datos outliers.
- Todos las observaciones marcadas como fraude se encuentran en sólo dos tipos (cash out y payments), por lo que decidimos que hacer oversampling (con técnicas como smote) no aportará nueva información ni optimizará los modelos a utilizar. Sólo escalaría con respecto a los dos tipos de transacciones, y no mejoraría la solución.

C. Ingeniería de Características

Dadas las características del dataset el cual se pretende identificar cuáles transacciones son posibles fraudes se determinó que es un modelo de clasificación binaria y no de predicción o clasificación multiclase.

Se identificó que el atributo más importante para poder solucionar el problema de determinar si una transacción es fraude o no es el atributo “isFraud”.

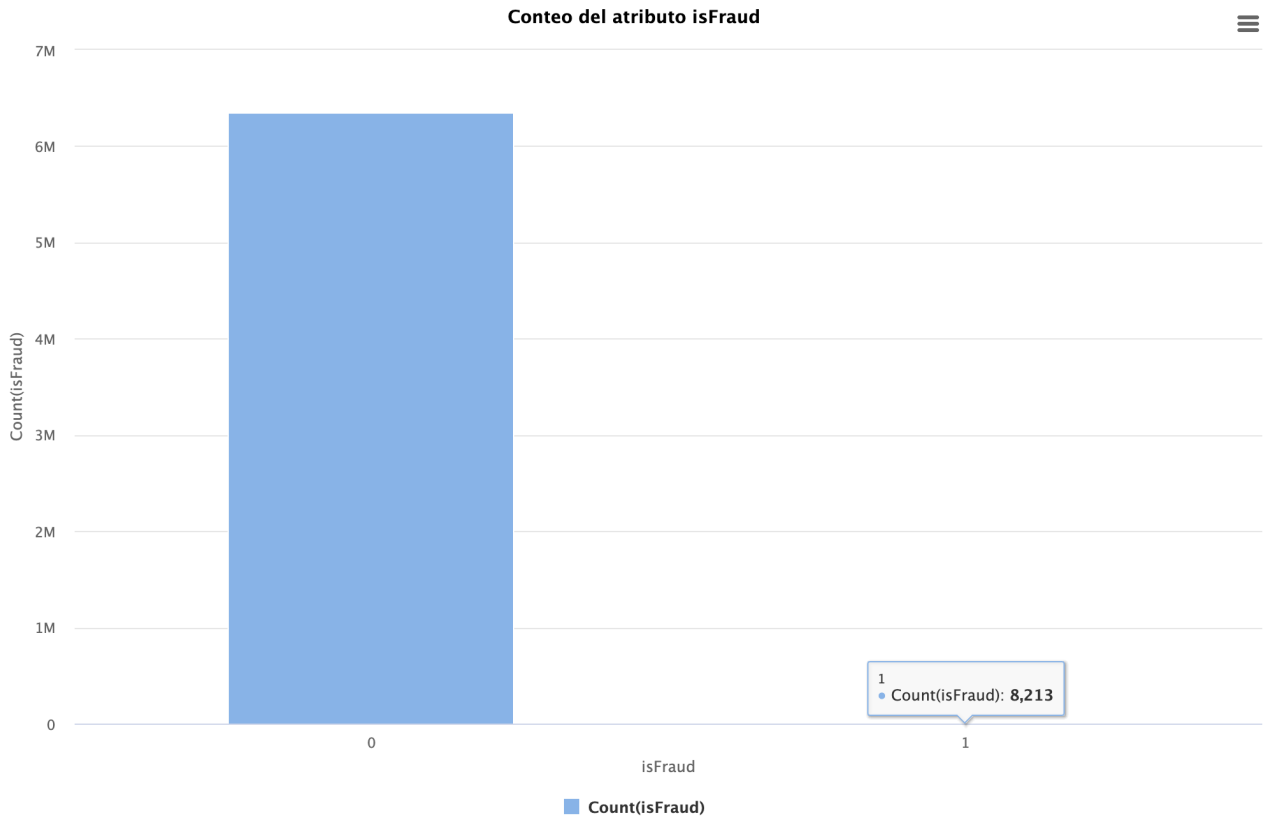

Figura 8. Demostración del desbalanceo del dataset (distribución de la variable isFraud)

Nominal value	Absolute count	Fraction
0	6354407	0.999
1	8213	0.001

Figura 9. Valores absolutos de la variable isFraud

Como se puede observar, de los más de 6 millones de transacciones que contiene el dataset únicamente 8,213 son categorizados como **fraude** (por medio del atributo isFraud). Con estos valores se puede concluir que el dataset está completamente desequilibrado para crear un modelo de clasificación preciso.

Para mejorar la capacidad del modelo a usar se realizó un undersampling que permitiera balancear el dataset el cual tuviera todos los casos marcados como fraude con el cual se pueda realizar un entrenamiento del modelo que tenga una mejor precisión. Adicionalmente mediante el método de codificación ***one hot coding*** convertimos los datos categóricos en binarios, lo cual permite un mayor número de iteraciones y un mejor aprendizaje del modelo.

D. Preparación de Datos para Modelado

Mediante el operador de RapidMiner "Remove Correlated Attributes" se removieron cualquier tipo de correlaciones entre los atributos del dataset para que así se pudiera tener un dataset sin información innecesaria, y no es necesario eliminar datos duplicados o datos vacíos, ya que el dataset no contiene ninguno; y mediante la técnica de ***undersampling*** se realizó un balanceo del dataset, obteniendo un total de **16.246** observaciones.

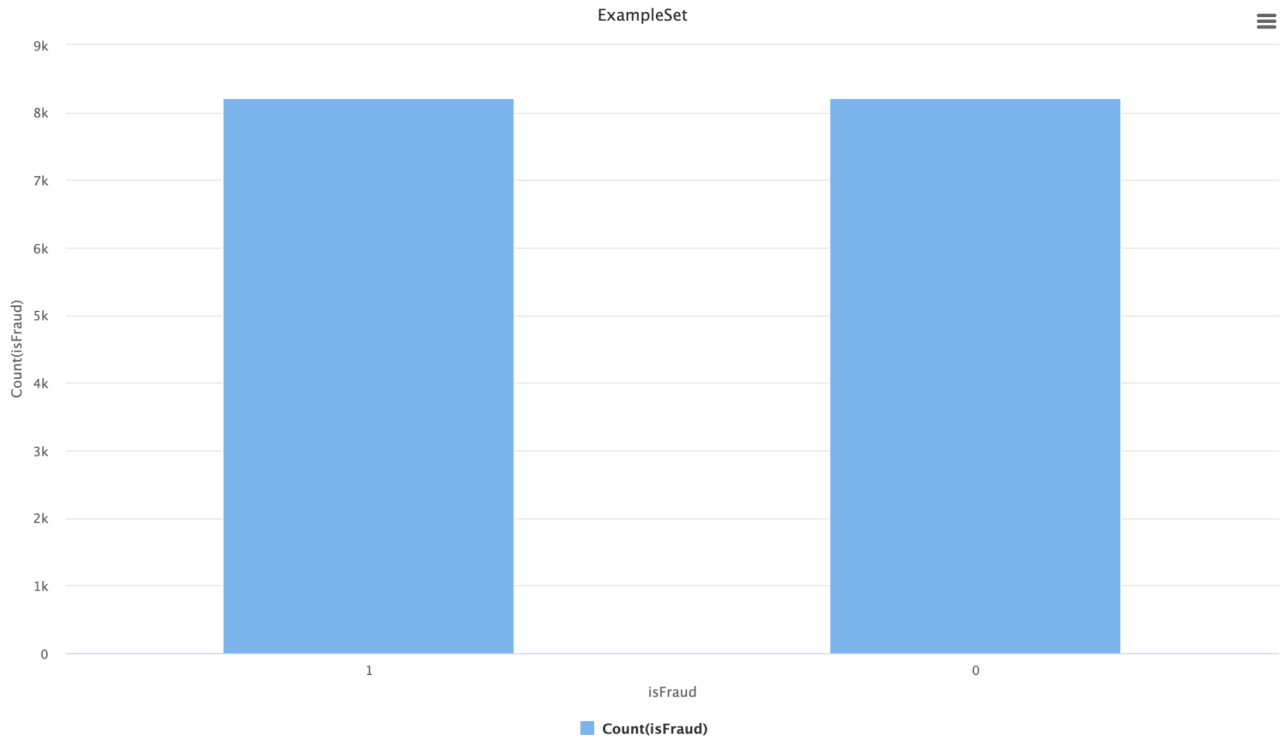

Figura 10. Distribución de la variable isFraud tras el undersampling.

Una vez realizado el sampleo de los datos (undersampling) se puede observar que ya hay un equilibrio entre las transacciones categorizadas como fraude y las que no. El nuevo dataset cuenta con un total de 16,426 registros de los cuales se mantienen los 8,213 registros categorizados como **fraude** junto con un conjunto de 8,213 registros de diferentes tipos de transacciones categorizados como **no fraudulentos**.

E. Selección y Entrenamiento del Modelo

Para la selección del modelo tomamos el dataset balanceado el cual contiene 16.426 observaciones y aplicamos los siguientes 5 modelos en rapidminer:

- Naive bayes
- KNN (Vecinos cercanos)
- Árbol de decisión

- Random Forest
- Deep Learning

Se toman estos cinco modelos los cuales son los más comunes y los de mejor precisión.

Para el entrenamiento y prueba del modelo se realizó una división 70/30 de los datos.

Resultados:

Modelo 1 - Naive Bayes

Rendimiento del entrenamiento: podemos observar que ambas clases tienen balanceadas las métricas de rendimiento, como lo vemos en el recall y la precisión(accuracy).

ceVector (Performance) | PerformanceVector (Performance Testing)

Table View | Plot View

accuracy: 93.68% +/- 0.94% (micro average: 93.68%)

	true true	true false	class precision
pred. true	5541	519	91.44%
pred. false	208	5230	96.18%
class recall	96.38%	90.97%	

Figura 11. Resultados del modelo Naive Bayes

Modelo 2 - Key-Nearest Neighbor (KNN)

Rendimiento del entrenamiento:

ceVector (Performance Testing) | **PerformanceVector (Performance)**

Table View | Plot View

accuracy: 90.88% +/- 0.80% (micro average: 90.88%)

	true true	true false	class precision
pred. true	5369	669	88.92%
pred. false	380	5080	93.04%
class recall	93.39%	88.36%	

Figura 12. Resultados del modelo KNN

Modelo 3 - Árbol de Decisión

Rendimiento del entrenamiento

Table View Plot View

accuracy: 77.98% +/- 0.97% (micro average: 77.98%)

	true true	true false	class precision
pred. true	5749	2532	69.42%
pred. false	pred. true	3217	100.00%
class recall	100.00%	55.96%	

Figura 13. Resultados del modelo Árbol de Decisión

Modelo 4 - Random Forest

Rendimiento del 98.66%

Table View Plot View

accuracy: 98.66% +/- 0.13% (micro average: 98.66%)

	true 1	true 0	class precision
pred. 1	8177	184	97.80%
pred. 0	36	8029	99.55%
class recall	99.56%	97.76%	

Figura 14. Resultados del modelo Random Forest

Modelo 5 - Deep Learning

Rendimiento del 93.20%

Table View Plot View

accuracy: 93.20% +/- 0.99% (micro average: 93.20%)

	true true	true false	class precision
pred. true	7680	584	92.93%
pred. false	533	7629	93.47%
class recall	93.51%	92.89%	

Figura 15. Resultados del modelo Deep Learning

F. Evaluación del Modelo

Validación del modelo: Evaluar el rendimiento del modelo utilizando métricas como precisión, exhaustividad, F1-score, matriz de confusión, ROC-AUC, entre otras. Para esta evaluación se tomaron los dos modelos con mayor accuracy dentro de las técnicas tradicionales.

Modelo Naive Bayes:

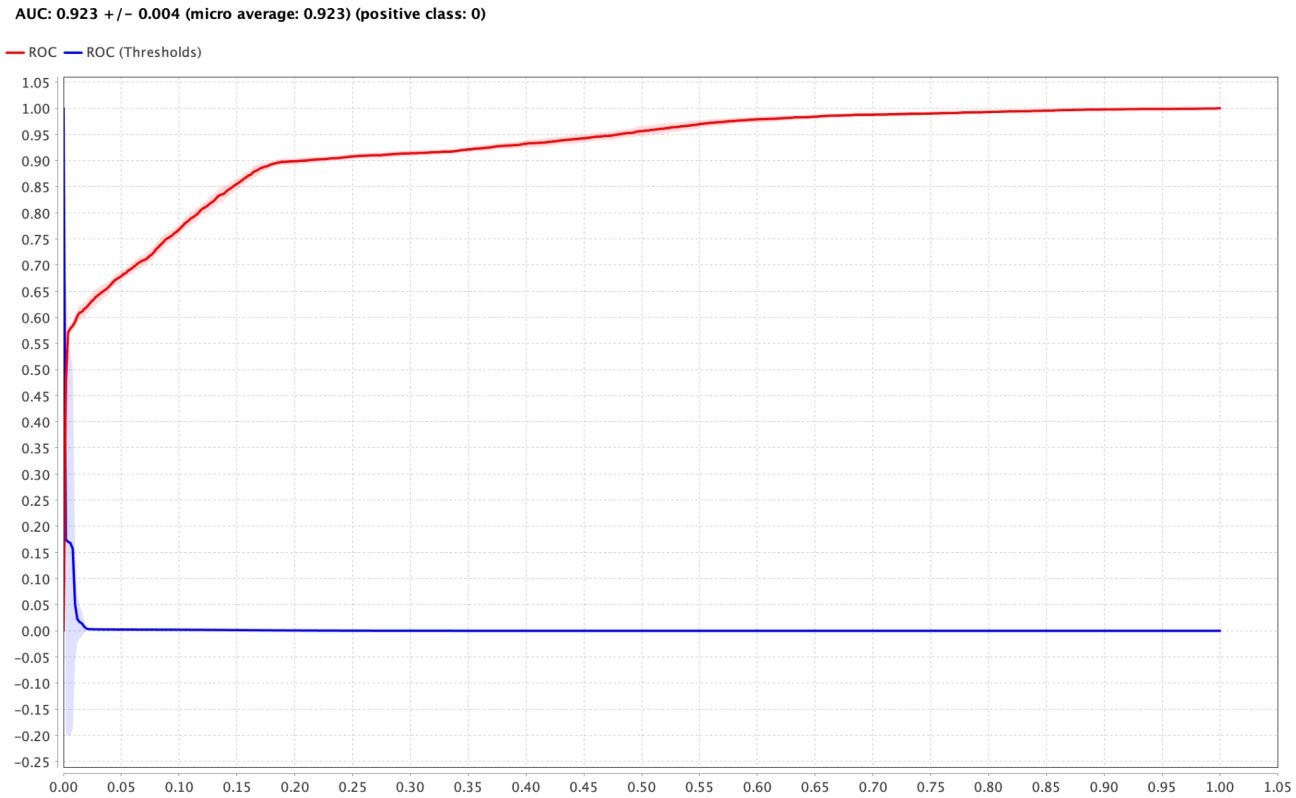

Figura 16. AUC del modelo Naive Bayes

Para el modelo Naive Bayes Model los resultados de los parámetros de medición fueron: accuracy de 93.68% +/-0.94%, AUC de 0.923, recall de 55.95% +/-0.98%, precision de 99.75% +/- 0.56% y f-measure de 71.68% +/- 0.71%.

Con estos resultados se puede interpretar que:

- El modelo tiene una alta accuracy, lo que indica un sólido rendimiento predictivo general.
- La puntuación AUC sugiere una buena capacidad de discriminación.
- La precisión es muy alta, lo que indica que cuando el modelo predice la clase positiva, suele ser correcto.
- La recuperación (recall) es relativamente menor, lo que indica que el modelo puede pasar por alto algunos casos positivos.
- La medida F (f-measure) proporciona un equilibrio entre precisión y recuperación.

Con mayores pruebas habría que determinar si el equilibrio entre precisión y recuperación es aceptable para esta aplicación.

Random Forest:

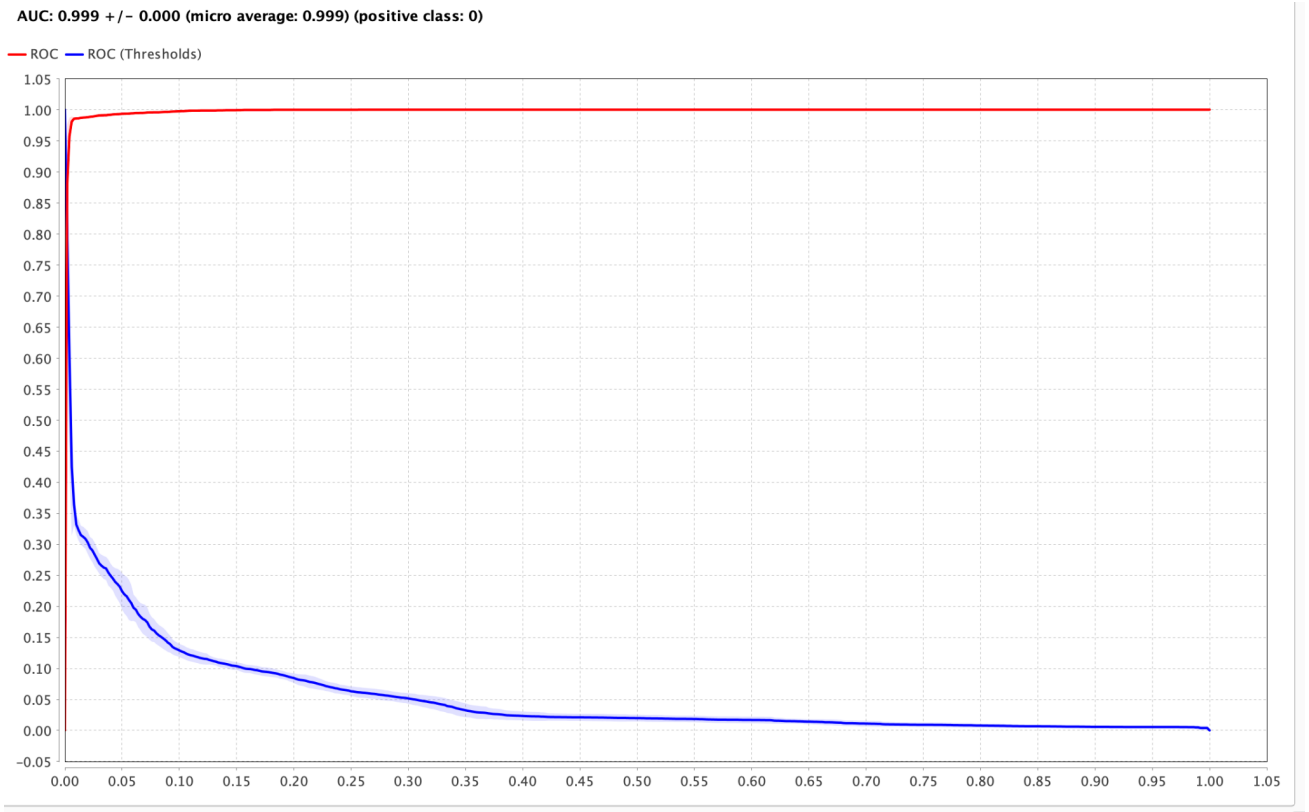

Figura 17. AUC del modelo Random Forest

Para el modelo Random Forest Model los resultados de los parámetros de medición fueron: accuracy de 98.66% +/-0.13%, AUC de 0.999, recall de 97.76% +/- 0.29%, precision de 99.55% +/- 0.19% y f-measure de 98.65% +/- 0.13%

El modelo Random Forest demuestra un rendimiento excepcional en todas las métricas.

La alta precisión y el AUC sugieren un fuerte rendimiento predictivo general y capacidad de discriminación.

La precisión y la recuperación son muy altas, lo que indica que el modelo es eficaz para identificar casos positivos y al mismo tiempo minimizar los falsos positivos.

Este modelo parece ser muy eficaz, pero como siempre, es crucial considerar las variaciones que pueden tener los datasets al ser balanceados y tener un componente de aleatoriedad de los registros seleccionados.

G. Validación y Optimización del Modelo

Validación cruzada: Verificar la generalización del modelo utilizando técnicas de validación cruzada.

Optimización adicional:

Para usar técnicas de Machine Learning más especializadas como lo son Deep Learning y Redes Neuronales se le realizó al dataset balanceado la técnica del One-Hot Encoding la cual optimiza el dataset y vuelve binarias las variables categóricas como lo es en este caso la variable Type (tipo de transacción). Esto mejora la capacidad de procesamiento y de entendimiento de los modelos de Machine Learning, tanto los modelos tradicionales como los más actuales.

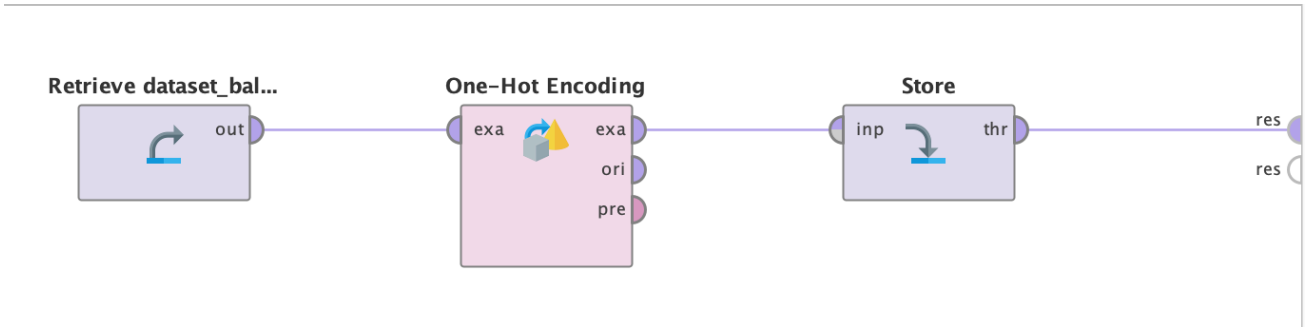

Figura 18. Operadores para utilizar One-Hot Encoding

Row No.	isFraud	type = TRANSFER	type = CASH_OUT	type = PAYMENT	type = CASH_IN	step	amount	oldbalance...	newbalanc...	oldbalance...
1	1	1	0	0	0	1	181	181	0	0
2	1	0	1	0	0	1	181	181	0	21182
3	1	1	0	0	0	1	2806	2806	0	0
4	1	0	1	0	0	1	2806	2806	0	26202
5	1	1	0	0	0	1	20128	20128	0	0
6	1	0	1	0	0	1	20128	20128	0	6268
7	1	0	1	0	0	1	416001.330	0	0	102
8	1	1	0	0	0	1	1277212.770	1277212.770	0	0
9	1	0	1	0	0	1	1277212.770	1277212.770	0	0

Figura 19. Dataset resultante tras el One-Hot Encoding

Una vez optimizado el código se aplicó el modelo de Redes Neuronales, así también como volver a probar algunos de los modelos previamente mostrados para observar la optimización de sus rendimientos. A continuación, se presentan los resultados.

Modelo de Redes Neuronales:

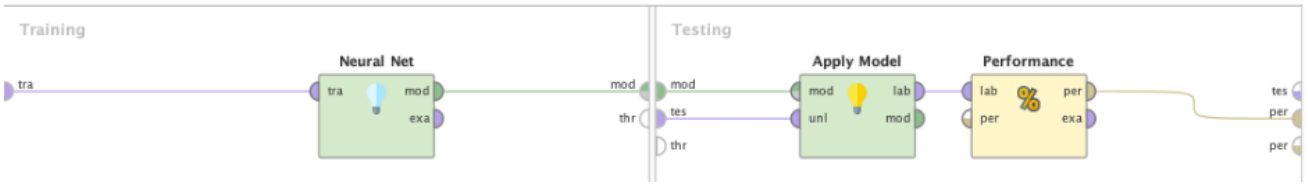

Figura 20. Cross Validation con el modelo Redes Neuronales

Table View Plot View

accuracy: 90.48% +/- 0.47% (micro average: 90.48%)

	true 1	true 0	class precision
pred. 1	7075	425	94.33%
pred. 0	1138	7788	87.25%
class recall	86.14%	94.83%	

Figura 21. Resultados del modelo Redes Neuronales

Se puede observar que presenta un rendimiento relativamente adecuado con una precisión de 90.48%, además de una desviación estándar del +/-0.47% que representa una buena estabilidad del modelo.

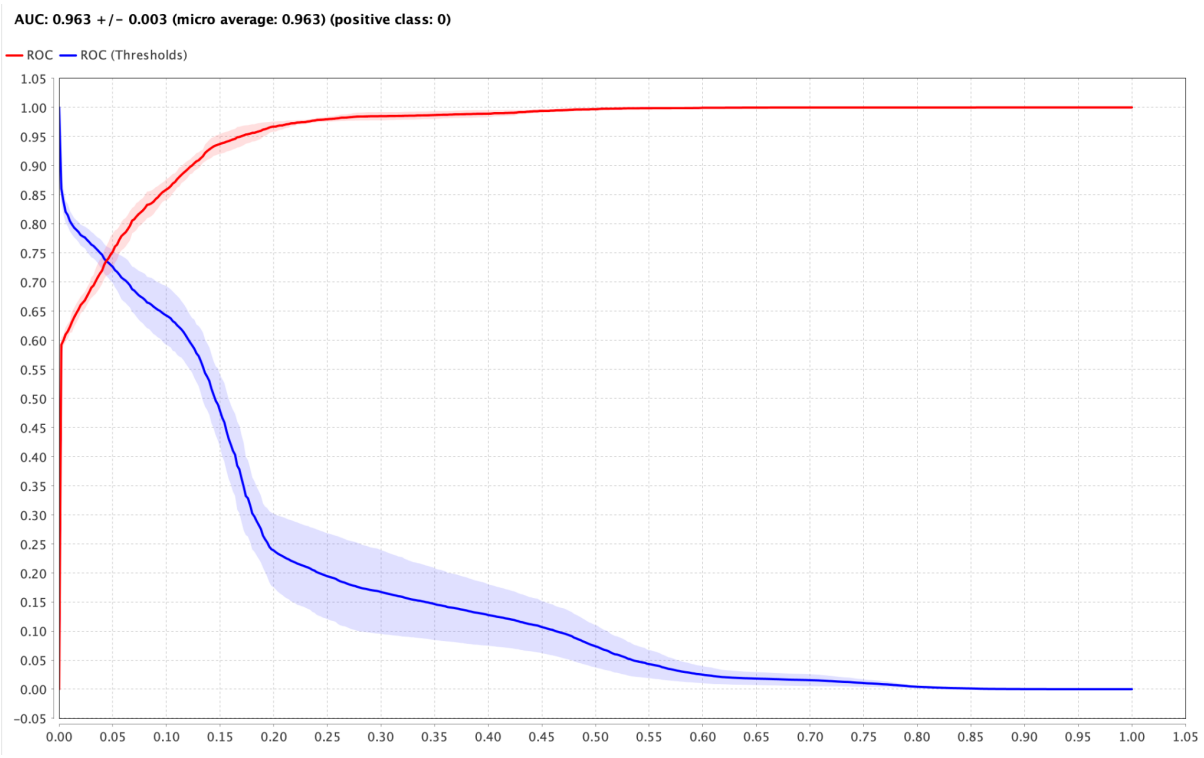

Figura 22. AUC del modelo Redes Neuronales

Podemos ver que el parámetro de AUC se encuentra por encima del 95% con una desviación estándar muy baja.

Deep Learning

Table View Plot View

accuracy: 95.26% +/- 0.39% (micro average: 95.26%)

	true 1	true 0	class precision
pred. 1	7866	432	94.79%
pred. 0	347	7781	95.73%
class recall	95.77%	94.74%	

Figura 23. Resultados del modelo Deep Learning

En el caso del modelo de Deep Learning se puede observar que tuvo una optimización del rendimiento de más de 3% utilizando el dataset con One-Hot Encoding.

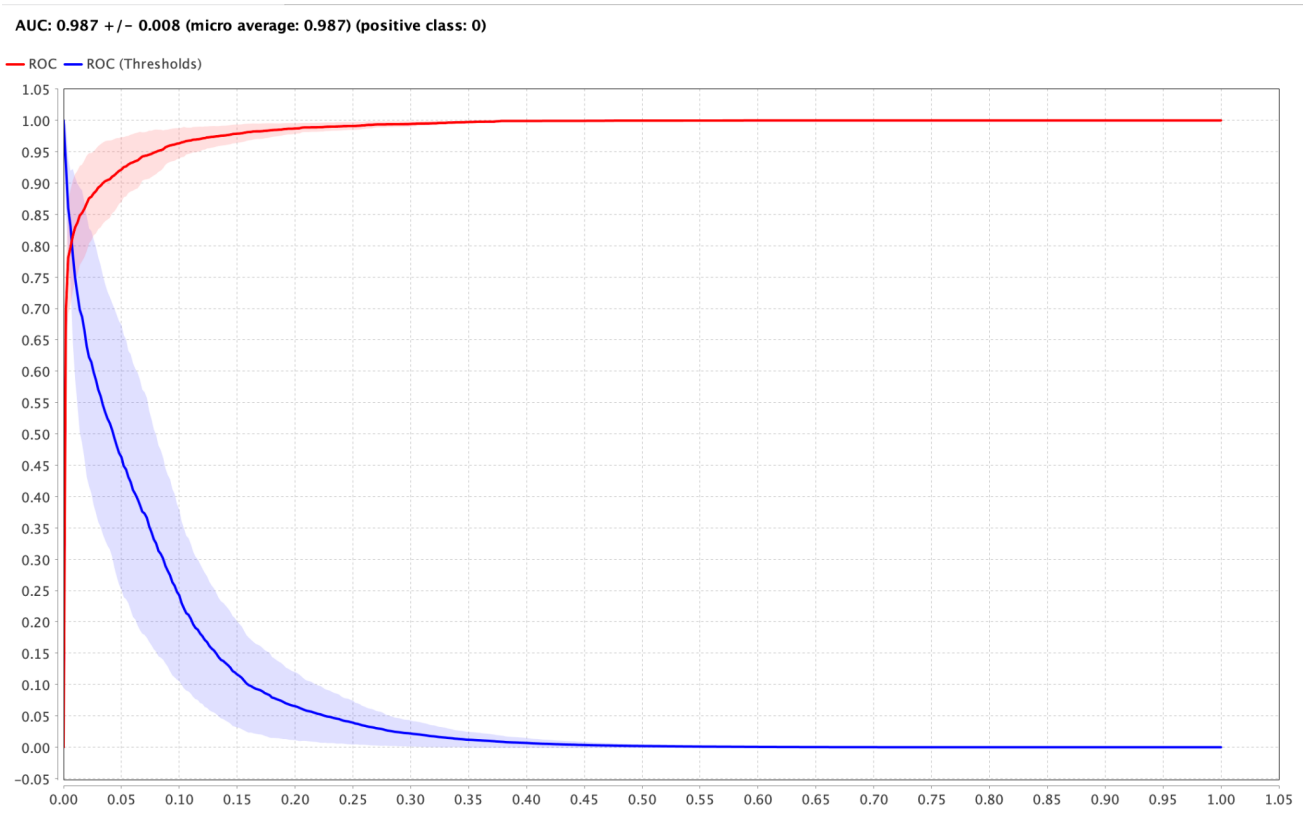

Figura 24. AUC del modelo Redes Neuronales

En cuanto al AUC se presenta poca diferencia pasando de de 0.986 a 0.987. Sin embargo, esto continua representando que hay una buena capacidad de discriminación en el modelo.

H. Despliegue y Monitoreo

Implementación del modelo: Integrar el modelo en un entorno de producción para monitorear transacciones en tiempo real, ya que en la actualidad sólo podrá predecir entorno simulados o contratos, no en tiempo real.

Monitoreo continuo: Supervisar el rendimiento del modelo y realizar actualizaciones según sea necesario para mantener su efectividad.

Repositorio del código

Github

Conclusiones

Podemos concluir que con un dataset tan pequeño, como el que obtuvimos luego del undersampling para hacerle un balanceo, se pueden utilizar las técnicas tradicionales como Naive Bayes o Random Forest para obtener un modelo de clasificación que ofrezca una buena precisión y estabilidad. Sin embargo, lo óptimo e ideal sería realizarle al dataset un oversampling utilizando los recursos de la nube, como por ejemplo Google Cloud Platform, ya que los recursos de nuestros ordenadores no tienen suficiente capacidad para dichos procesos computacionales. Con mayor se puede sacar provecho de técnicas más modernas como Deep Learning y Redes Neuronales para obtener el modelo de clasificación más óptimo, preciso y confiable.

Consideraciones Finales

La actualización constante del modelo es crucial para adaptarse a nuevos patrones de fraude.

La colaboración con expertos en seguridad financiera es esencial para mejorar la precisión y eficacia del modelo.