

Autores: Eric Flores, Sebastián González, Agustín Cervantes y Miguel Ángel del Río

INDICE 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
C. INGENIERÍA DE CARACTERÍSTICAS.....	5
D. PREPARACIÓN DE DATOS PARA MODELADO.....	5
E. SELECCIÓN Y ENTRENAMIENTO DEL MODELO.....	5
F. EVALUACIÓN DEL MODELO.....	5
G. VALIDACIÓN Y OPTIMIZACIÓN DEL MODELO.....	5
H. DESPLIEGUE Y MONITOREO.....	5
CONCLUSIONES.....	6
CONSIDERACIONES FINALES.....	6
.....	12

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:

1. RapidMiner
2. BigQuery
3. Cloud

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: full_dataset.csv
- b) Tipo: CSV
- c) Observaciones : 6.362.620 observaciones
- d) Atributos: 11 atributos.

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

Limpieza y preprocesamiento: Eliminar datos duplicados, manejar valores faltantes y transformar variables si es necesario (normalización, codificación de variables categóricas).

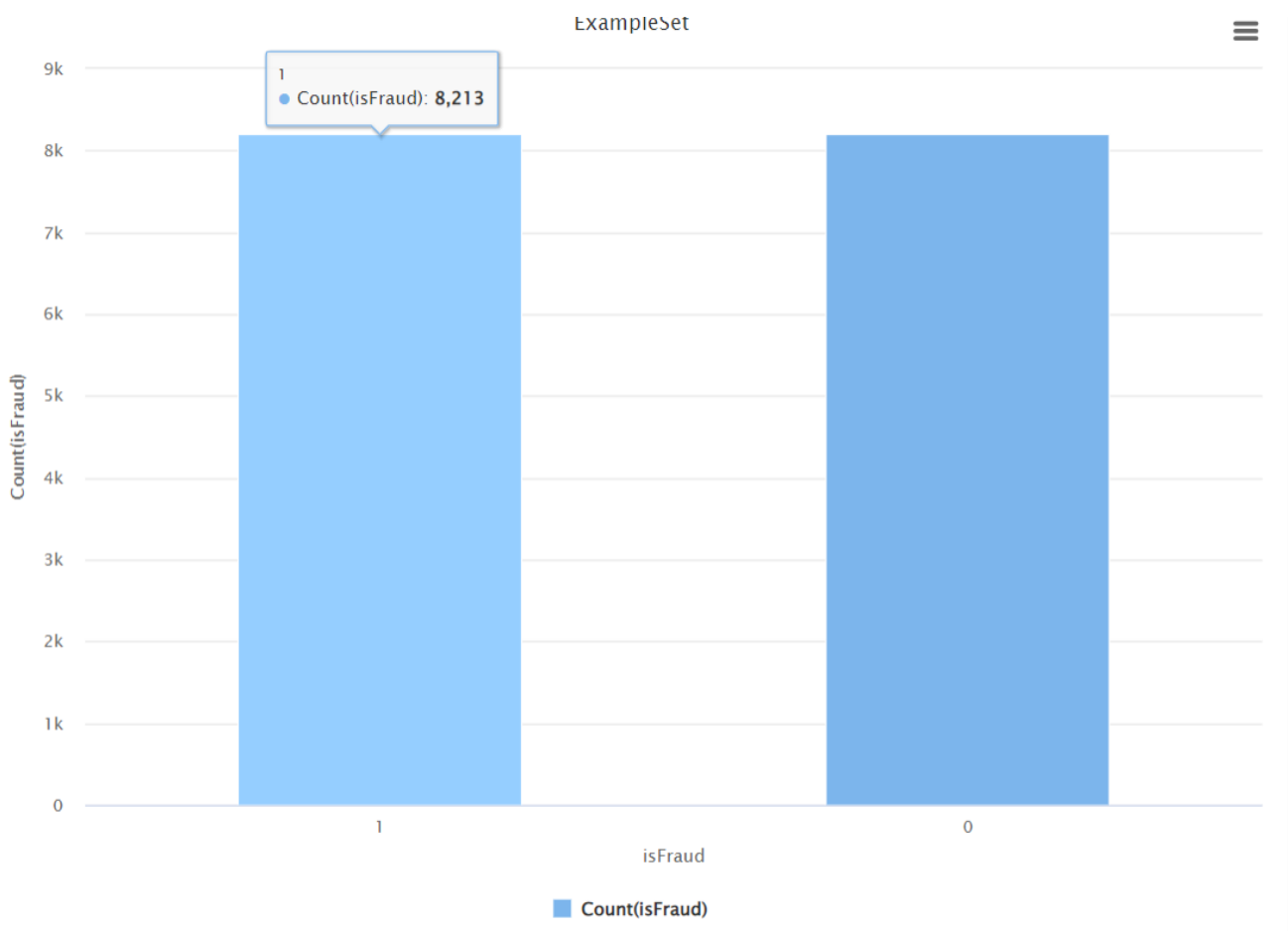
B. Análisis Exploratorio de Datos (EDA)

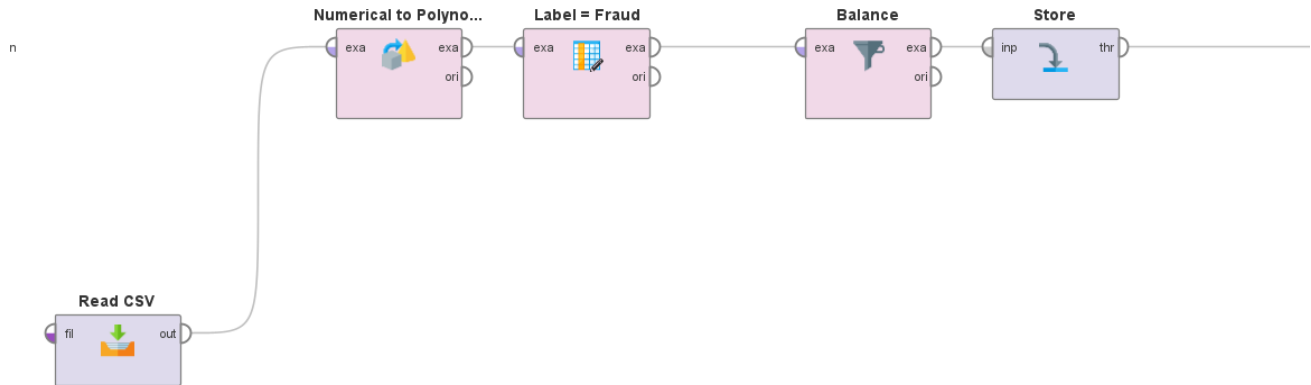
Identificamos los patrones mediante la etiqueta *“isFraud”*, tomado como la etiqueta principal de las “posibles” transacciones fraudulentas. Pueden ocurrir bajo diferentes tipos de movimientos reflejados en las columnas *type* que complementan la información para la composición de los modelos usados.

Una vez teniendo en cuenta que los datos iban a ser nuestros principales, seguimos con un balanceo de datos, los conjuntos de datos de fraude están desequilibrados, con muchos más casos de transacciones no fraudulentas que fraudulentas. Esto nos puede ayudar a identificar este desequilibrio y a aplicar técnicas para manejarlo, como el sobremuestreo o el submuestreo.

A partir de ahí, podemos crear nuevas características que pueden ser más informativas para los modelos, como indicadores de comportamiento anómalo.

Para un mejor entendimiento y aprovechamiento de los resultados utilizamos los gráficos y visualizaciones obtenidas en el balanceo para entender mejor la distribución de transacciones normales y fraudulentas.






Para poder hacer un correcto balanceo de los datos, complementamos toda la muestra “isFraud” con otra muestra con la misma cantidad de datos que no contienen fraude.

Para ello, debemos de tener toda la información en el lenguaje correcto para poder dividirla, introduciendo así el operador *Numerical to Polynomial*, marcando “isFraud” como la label importante y obteniendo así nuestro dataset simplificado.

C. Ingeniería de Características

En este apartado ya teniendo en cuenta las principales características obtenidas en el balanceo procedemos como la asignación de roles para el modelo en este caso un *label* para “isFraud” este rol nos permite tomarlo como el principal y en cual RapidMiner va a generar su estructura de modelado.



Edit Parameter List: **set roles**
This parameter defines new attribute roles.

attribute name	target role
isFraud	label

La asignación de rol y la creación de etiquetas así como el cambio del tipo de dato, son esenciales para un mejor muestreo en los modelos.

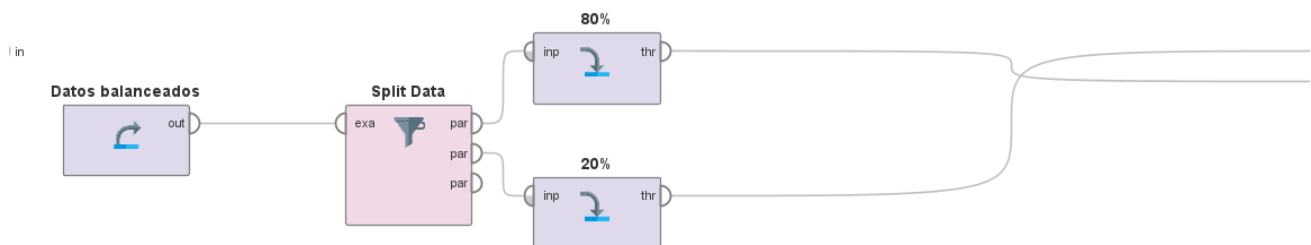
D. Preparación de Datos para Modelado

Una vez obtenida nuestra muestra, se debe de continuar preparando los datos para aplicar los modelos correspondientes.

Por ello, se procede a dividir la muestra en un 80% y 20%. De esta manera, se busca que los modelos entrenen con un alto número de datos (80%) para posteriormente aplicar dicho entrenamiento a l 20% de los datos restantes y poder obtener así resultados más precisos.

Se busca dividir la información, debido a que si el modelo entrena con todo la muestra, no se dispone posteriormente de más datos para poder ver si se ha entrenado el modelo de forma adecuada.

Separación datos 80/20



Explicación:

E. Selección y Entrenamiento del Modelo

UNA vez segmentados los datos y habiendo hecho los cambios necesarios para una mejor lectura de RapidMiner continuamos con las muestras de algoritmo, en este caso RapidMiner ofrece una gran variedad de operadores (Random Forest, Support Vector Machines, Redes Neuronales, etc.).

Para esta muestra hemos decidido partir de las siguientes debido a su facilidad de interpretación y de su capacidad de adaptación y respuesta, *Aunque en estas versiones fueron un preliminar para buscar el mayor número de asertividad posible.

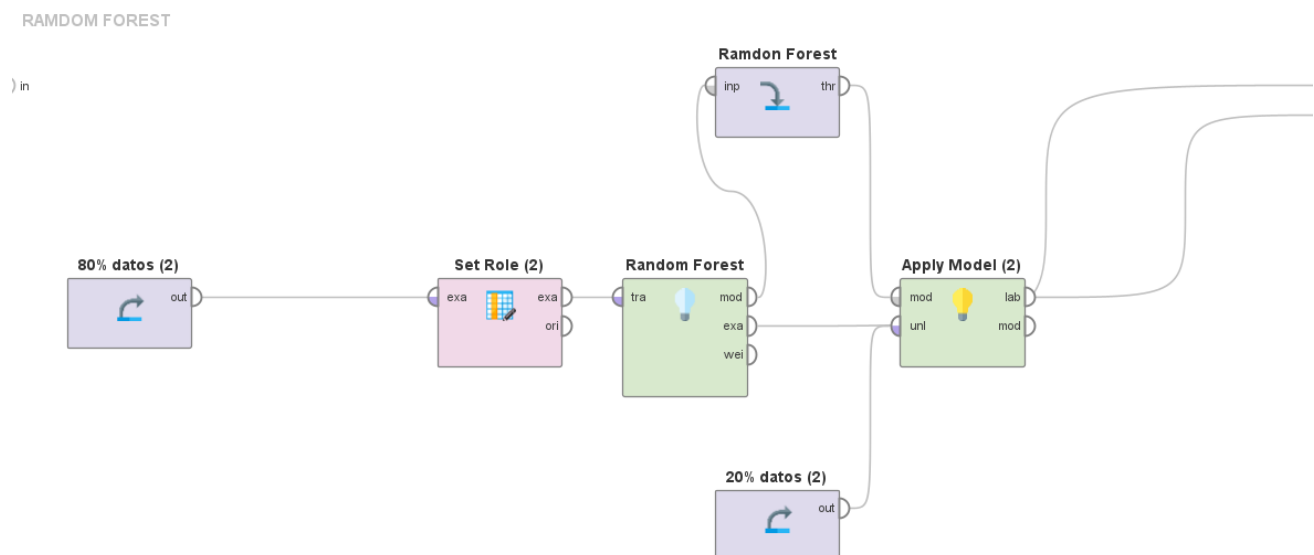
El primer modelo de selección fue *Random Forest* gracias a su capacidad para manejar hasta miles de variables de entrada e identificar las más significativas, reduce la dimensionalidad de los datos, cuenta con métodos efectivos para estimar los valores faltantes, , selecciona los predictores forma automática, como su nombre lo indica, usa varios árboles de decisión usando predictores numéricos como categóricos sin tener que crear tantas variables esto a su vez siendo altamente estable.

El segundo modelo fue *Naive Bayes* que como su nombre lo indica está basado en el teorema de Bayes, es uno de los algoritmos de machine learning más rápidos y sencillos para predecir una clase de conjunto de datos, su manejo de variables de entrada y rápida identificación de las más significativas, su rendimiento de

clasificación arroja buenos resultados incluso teniendo menos datos de entrenamiento en este caso hemos decidido trabajar con 16,426 datos.

El tercero es *Decision Tree* se tomó a consideración gracias a que es fácil de entender e interpretar, y en este caso el árbol lo pudimos visualizar en una de nuestras muestras ejecutables, puede trabajar con variables cuantitativas y cualitativas, no requiere de una preparación de datos exigente simplemente con que no tuviéramos valores nulos, se justifica por su uso de la lógica booleana.

Los tres modelos anteriores también fueron tomados a consideración por la característica de ser menos susceptibles a ser influenciados por valores atípicos.



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.

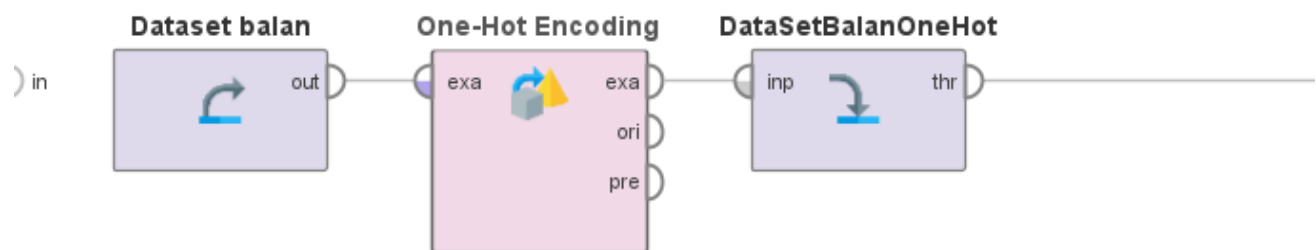
Ajuste de hiper parámetros: Optimizar los hiper parámetros del modelo para mejorar su desempeño.

G. Validación y Optimización del Modelo

Con los resultados obtenidos anteriormente, la AUC se encuentra en valores fuera de los correctos (entre un 50% y un 70%) por lo cual, decidimos optimizar los datos que ofrecemos al modelo.

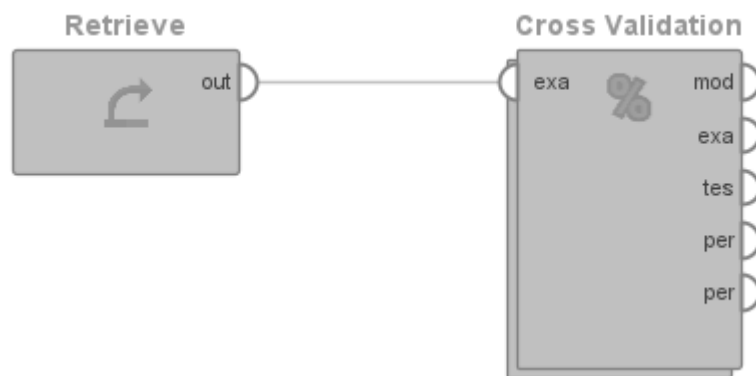
Para ello, transformamos cual campo tipo texto en numérico, favoreciendo así la lectura de los datos para acercarnos más a un grado de AUC sobre el 90%.

DataHot



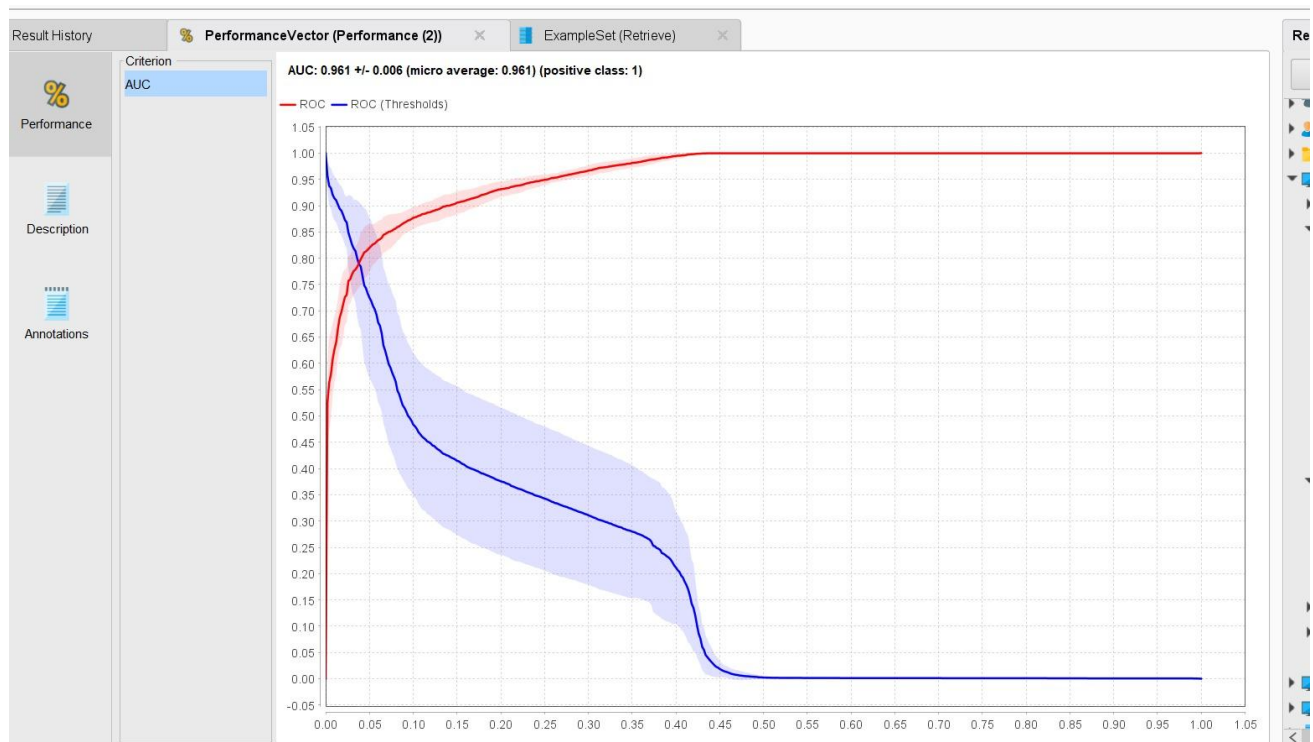
Una vez optimizados los datos, realizamos un cross validation por los siguientes motivos:

- Estimación más robusta del rendimientos del modelo
- Prevención de overfitting
- Optimización de parámetros



Dentro de la cross validation, utilizamos el entrenamiento de neural net, obteniendo así un 96 % de AUC.

Se trata de un valor muy alto con posible overfitting, por ello los siguientes pasos serán cambiar variables y pequeños ajustes en el entrenamiento para ver si se trata de un resultado real o no.



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 únicamente 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.

Demo funcional

[PwCChallengeDataDreamTeam](#)

Repositorio del código

https://github.com/AguCervantes/DesafioPWC_Grupo2

Conclusiones

La predicción de fraude en tarjetas de crédito mediante técnicas de machine learning es fundamental para mitigar riesgos financieros. Los pasos mencionados constituyen un marco sólido para desarrollar un sistema efectivo de detección de fraudes.

Una mayor optimización del modelo, permitiría hacerse a un 100% de efectividad, pudiendo así evitar todas aquellas transacciones fraudulentas, ofreciendo así el cliente final una seguridad por parte de su entidad bancaria como propuesta de valor diferenciadora del resto.

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.