MÓDULO TÉCNICO

RETO 2

DETECCIÓN DE FRAUDE EN TARJETAS DE CRÉDITO

NICOLÁS BAZTAN YOLDI

VÍCTOR MONTILLA CASTILLA

ALEJANDRO PALANCAR DEL ESTAL

ÍNDICE DEL INFORME PROPUESTA SOLUCION RETO PwC

PREDICCIÓN DEL FRAUDE EN TARJETAS DE CRÉDITO – RETO PWC 2

INTRODUCCIÓN 2
OBJETIVO 2
PASOS PARA PREDECIR FRAUDE EN TARJETAS DE CRÉDITO 2
A. ADQUISICIÓN DE DATOS 2
B. ANÁLISIS EXPLORATORIO DE DATOS (EDA) 4
C. INGENIERÍA DE CARACTERÍSTICAS 5
D. PREPARACIÓN DE DATOS PARA MODELADO 5
E. SELECCIÓN Y ENTRENAMIENTO DEL MODELO 7
F. EVALUACIÓN DEL MODELO 8
G. VALIDACIÓN Y OPTIMIZACIÓN DEL MODELO 9
CONCLUSIONES 10
CONSIDERACIONES FINALES 10

Predicción del Fraude en Tarjetas de Crédito – Reto PwC

Introducción

En un mundo en constante evolución, donde las transacciones digitales están desplazando gradualmente al dinero en efectivo, la capacidad de discernir movimientos financieros fraudulentos se convierte en un elemento crucial.

En el pasado, la detección de tales movimientos se basaba principalmente en motores de reglas estáticas; sin embargo, con la revolución de la inteligencia artificial, se han abierto nuevas y más precisas posibilidades.

En este contexto, se plantea un reto interesante: identificar movimientos fraudulentos dentro de un conjunto de datos proporcionado. Este desafío no solo representa una oportunidad para aplicar técnicas avanzadas de machine learning, incluyendo enfoques clásicos y deep learning, sino que también invita a considerar metodologías alternativas como el uso de grafos.

Este reto no solo busca detectar anomalías financieras, sino también explorar y optimizar el potencial de diversas herramientas de vanguardia en la lucha contra el fraude en un mundo cada vez más digitalizado y complejo.

Objetivo

Este documento detalla los procedimientos fundamentales que se llevarán a cabo en relación con el conjunto de datos proporcionado (consultar punto A), con el fin de anticipar actividades fraudulentas en transacciones que involucran tarjetas de crédito. Este análisis se realizará empleando diversas técnicas de machine learning respaldadas por una variedad de herramientas. La elección de estas herramientas se fundamenta en la necesidad de entregar este informe en un plazo ajustado, entre otros factores:

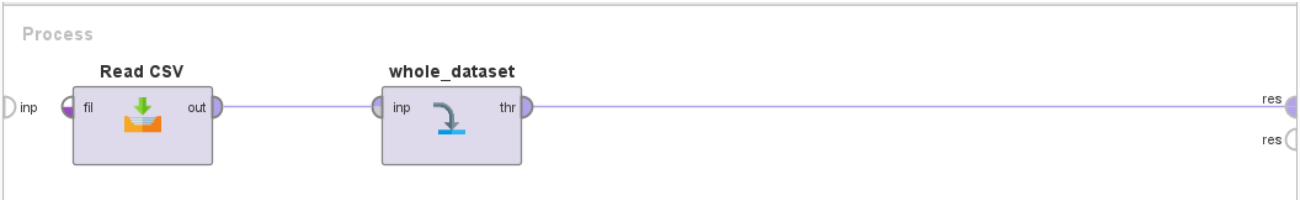
1. RapidMiner
2. BigQuery
3. Cloud

Pasos para Predecir Fraude en Tarjetas de Crédito

A. Adquisición de Datos

Obtención de datos:

Se nos proporciona un conjunto de datos que comprende transacciones de tarjetas de crédito. Este dataset carece de información de fechas en las transacciones; sin embargo, incluye atributos como la cantidad involucrada en la transacción, el tipo de transacción y otros detalles relevantes.



Descripción del dataset original del reto PwC

a) Nombre: whole_dataset.csv
b) Tipo: CSV
c) Observaciones : 6.362.620 observaciones
d) Atributos: 11 atributos.

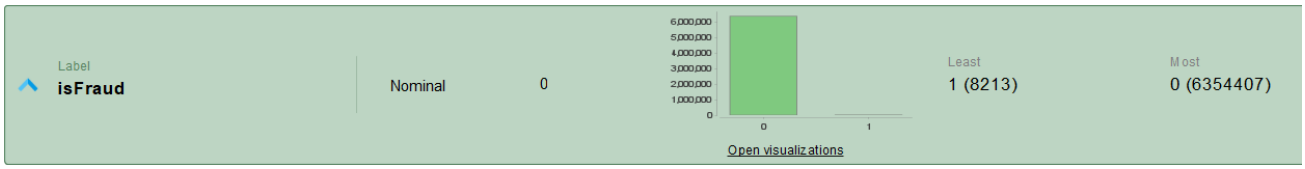
Name	Type	Missing
amount	Real	0
oldbalanceOrg	Real	0
newbalanceOrig	Real	0
oldbalanceDest	Real	0
newbalanceDest	Real	0
step	Integer	0
isFlaggedFraud	Integer	0
Label isFraud	Nominal	0
type	Nominal	0
nameOrig	Nominal	0
nameDest	Nominal	0

En un primer proceso de limpieza y preprocesamiento de los datos, se llevó a cabo una exhaustiva revisión para eliminar posibles datos duplicados y se implementó un enfoque riguroso para el manejo de valores faltantes, aunque en este caso particular, el dataset no presentaba ausencia de valores.

Además, se consideró la posibilidad de transformar variables si fuese necesario, aplicando técnicas como la normalización o la codificación de variables categóricas. Cabe destacar que, en esta primera instancia, no se realizó ninguna transformación ni normalización en el conjunto de datos con el que se trabajó.

B. Análisis Exploratorio de Datos (EDA)

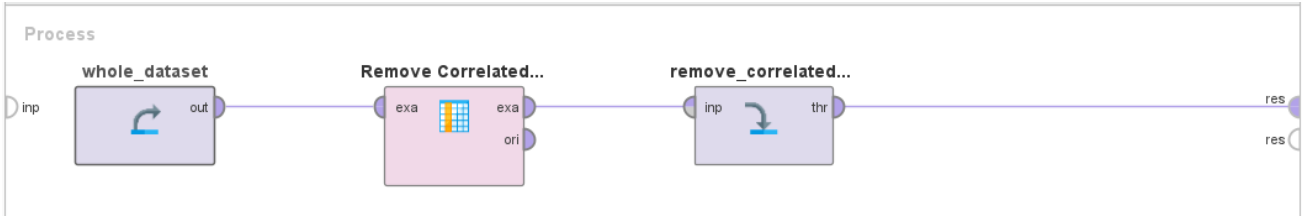
Tras una primera revisión del dataset se puede observar que, en relación con la variable isFraud, la cual indica si un movimiento es fraude o no, la muestra está completamente desbalanceada. Hay más de 6 millones de observaciones que no son fraude (isFraud = 0), frente a pocas más de 8.000 observaciones que sí lo son (isFraud = 1). Es decir, de la muestra total, tan solo el 0.1% de las observaciones son fraude.



Este planteamiento representa un desafío debido a que, al entrenar y aplicar un modelo para detectar fraudes utilizando el conjunto de datos completo, el modelo podría tender a clasificar todas las transacciones como fraudulentas, mostrando una confianza excesiva, lo cual sería un error.

Por este motivo, será necesario equilibrar el conjunto de datos tomando en consideración la variable "isFraud", sin eliminar ninguna observación donde esta sea igual a 1 (fraude), ya que esta es la clase minoritaria en el dataset. Este enfoque buscará evitar problemas potenciales de ajuste excesivo (overfitting) o insuficiente (underfitting) durante la etapa de construcción del modelo. Dicho procedimiento se detallará más adelante en la fase de Preparación de Datos para Modelado.

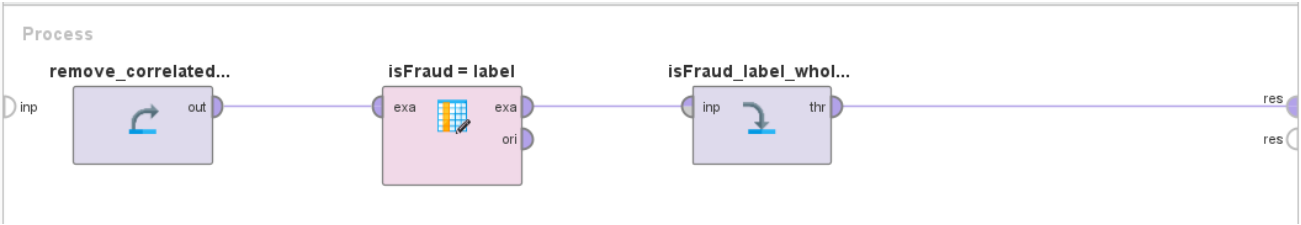
Durante esta fase de exploración de características, se llevó a cabo un análisis exhaustivo de la distribución de variables, así como la identificación de posibles correlaciones entre ellas. En este proceso, se utilizó el operador 'Remove Correlated Attributes' en RapidMiner para eliminar cualquier relación lineal fuerte entre las variables, lo que permitió mitigar posibles redundancias en el dataset. Además, se exploraron detalladamente los datos en busca de patrones o anomalías; sin embargo, no se identificaron irregularidades significativas ni patrones destacables en la información analizada, lo que sugiere una relativa ausencia de comportamientos atípicos o estructuras particulares en los datos.



C. Ingeniería de Características

En el contexto de preparación de datos para la construcción de modelos predictivos, se llevó a cabo un proceso de selección y creación de características fundamental. Se estableció la variable "isFraud" como la etiqueta principal mediante el operador 'Set Role' en RapidMiner, con el objetivo de ajustar la muestra de datos para futuros modelos. Esta variable será el punto central alrededor del cual girarán los modelos, ya que serán responsables de determinar si una transacción es considerada como fraude o no.

Además, en este proceso se realizó la identificación de las características más relevantes que podrían influir en la capacidad predictiva del modelo. Asimismo, se exploró la posibilidad de generar nuevas características derivadas de las existentes, con el fin de potenciar aún más la capacidad predictiva y mejorar el rendimiento de los modelos a construir.



D. Preparación de Datos para Modelado

Como ya se apreció y mencionó en la fase del EDA, la muestra estaba completamente desbalanceada en relación con la variable isFraud, la cual indica si una transacción es fraude o no, y que será el eje principal de los modelos a entrenar y aplicar más adelante.

Por ello, el primer paso a realizar será el balanceo de la muestra total mediante una técnica de undersampling. La técnica de undersampling es un método común utilizado en el campo del aprendizaje automático para abordar desequilibrios en conjuntos de datos, especialmente cuando se trata de problemas de clasificación binaria donde una clase es significativamente más predominante que la otra, como ocurre en este caso. Este procedimiento es útil para mejorar el rendimiento de los modelos utilizados.

En un conjunto de datos desequilibrado donde una de las clases está subrepresentada en comparación con la otra, el modelo de machine learning puede sesgarse hacia la clase mayoritaria y tener dificultades para aprender patrones o características de la clase minoritaria. El undersampling o submuestreo es una estrategia que busca equilibrar las proporciones entre las clases, reduciendo aleatoriamente el número de instancias de la clase mayoritaria para que se asemeje más a la cantidad de instancias de la clase minoritaria.

En este caso, se reduce aleatoriamente el número de observaciones de la variable isFraud cuando esta es igual a 0 (no fraude) hasta quedar 8.213. Este número es exactamente igual al número de instancias totales de fraude (isFraud = 1). Por tanto, queda una muestra final balanceada de 16.426 observaciones, donde la mitad son transacciones donde hay fraude y la otra mitad donde no lo hay.

El siguiente paso es el One-Hot Encoding, una técnica de transformación utilizada para convertir variables categóricas en representaciones numéricas, lo que facilita su uso en algoritmos de aprendizaje automático.

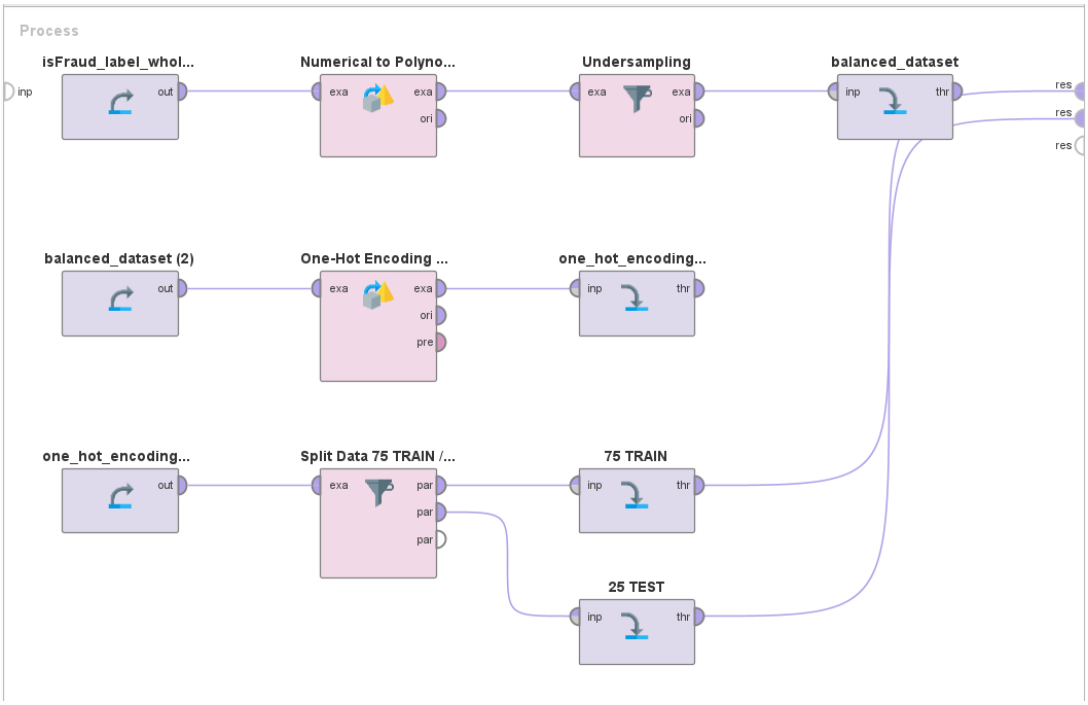
En el contexto de la preparación de datos para el modelado, el One-Hot Encoding es fundamental cuando se trabaja con variables categóricas o nominales que no son directamente interpretables por muchos modelos de machine learning, ya que estos suelen requerir datos numéricos para su procesamiento.

Consiste en convertir una columna de datos categóricos en varias columnas binarias (0 o 1) correspondientes a las diferentes categorías presentes en la variable original. Cada categoría se convierte en una columna separada, donde un "1" indica la presencia de esa categoría en una observación específica y un "0" en las demás columnas. Esto permite que el modelo pueda interpretar y utilizar estas variables categóricas de manera más efectiva al tratarlas como características numéricas independientes. En este caso, se ha realizado concretamente para las categorías de la variable "type".

El One-Hot Encoding es crucial para modelos de machine learning que no pueden manejar directamente variables categóricas, ya que suelen requerir que todas las entradas sean numéricas. Además, incluso algunos algoritmos que pueden manejar datos categóricos de forma directa pueden beneficiarse de este paso al evitar interpretaciones erróneas de las categorías como si tuvieran un orden inherente.

En este caso se van a aplicar los modelos de Naive Bayes y Redes Neuronales, los cuales requieren de esta transformación de variables previa, y que se detallan en la siguiente fase de Selección y Entrenamiento del Modelo. Con la aplicación de esta técnica se asegura que la información contenida en las variables se incluya de manera adecuada en el proceso de modelado.

Por último, se divide la base de datos balanceada en datos de entrenamiento y en un conjunto de prueba, con el propósito de evaluar la capacidad predictiva del modelo entrenado en datos que no ha visto durante el proceso de entrenamiento. Este enfoque es fundamental en la aplicación de modelos de machine learning para estimar su rendimiento y generalización a nuevos datos. La proporción del 75% se utiliza para enseñar al modelo a reconocer patrones y relaciones en los datos, mientras que el 25% restante se reserva para evaluar qué tan bien el modelo generaliza sobre datos no utilizados previamente, permitiendo así estimar su desempeño en situaciones del mundo real. Esta división ayuda a evitar el sobreajuste (overfitting) al evaluar la capacidad del modelo para hacer predicciones precisas en nuevos conjuntos de datos.

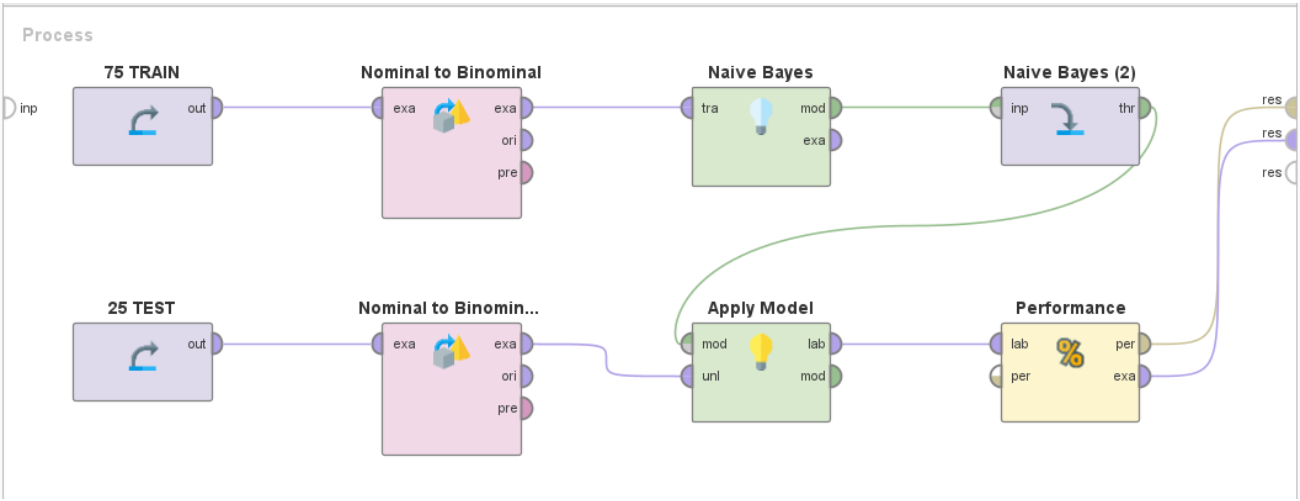


E. Selección y Entrenamiento del Modelo

En cuanto a la elección y entrenamiento de los modelos de predicción, se han elegido dos diferentes, ya mencionados en la fase anterior.

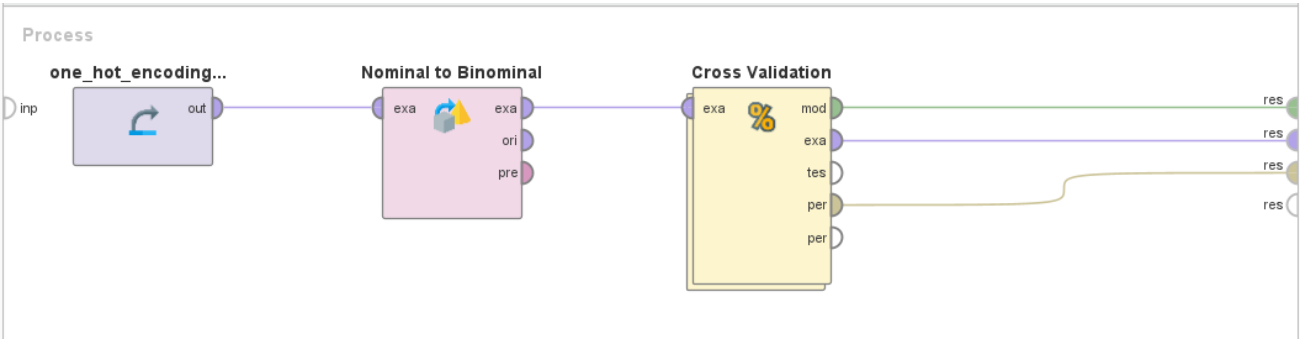
En primer lugar, se ha aplicado un modelo Naive Bayes, ya que es un modelo que conocíamos previamente, y que se caracteriza por una fácil comprensión y un gran rendimiento. Una de las posibles desventajas de este modelo es que asume independencia entre características, lo que no era un problema en nuestro caso ya que previamente habíamos eliminado las posibles correlaciones en la base de datos.

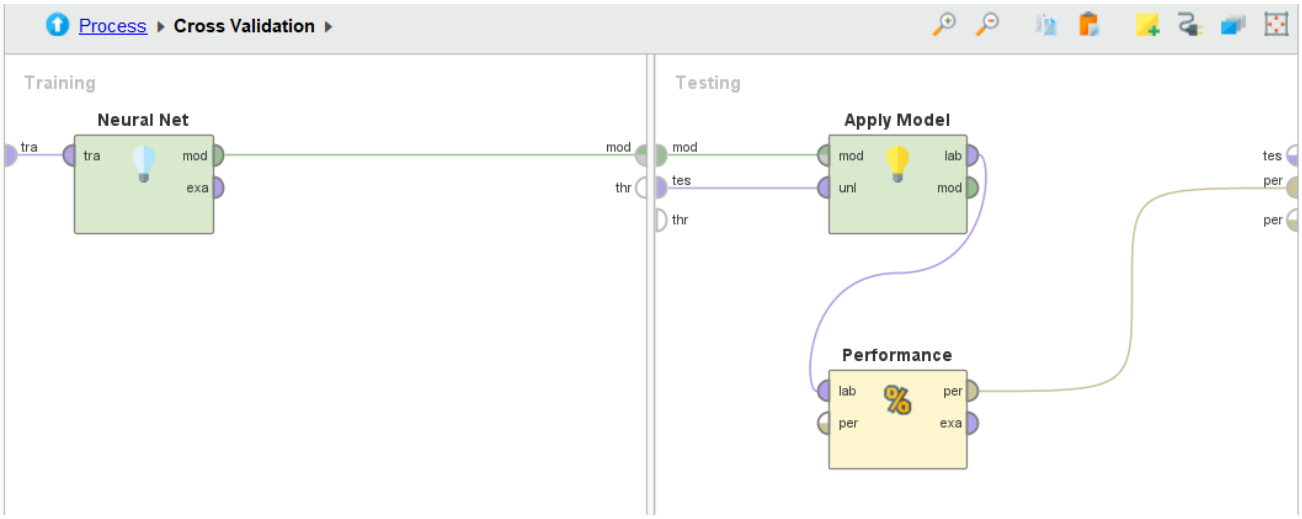
Este modelo, al igual que con el resto de los procedimientos, se ha entrenado en el programa RapidMiner, utilizando los datos de entrenamiento (75% de la base de datos balanceada) para entrenar el modelo y el conjunto de prueba (25%) para posteriormente evaluarlo. Además, antes de nada, se ha cambiado el atributo label 'IsFraud' a binomial para adaptarlo a las necesidades del modelo. Por último, se ha añadido un operador Performance con el objetivo de conocer el AUC del modelo.



Por otro lado, en cuanto a la utilización del modelo de Redes Neuronales, se ha elegido por su mayor complejidad, profundidad y adaptabilidad a los diferentes datos. Además, para optimizar aún más el modelo se ha utilizado un Cross Validation, que proporciona una estimación más confiable del rendimiento del modelo al considerar múltiples configuraciones de datos de entrenamiento y prueba. Al utilizar este modelo, parecido al DeepLearning, se esperaba lograr una mejora sustancial en la métrica AUC (Área bajo la curva ROC) al realizar ciertos ajustes o cambios en el modelo.

En RapidMiner, se añadide un operador Nominal to Binominal y posteriormente el operador Cross Validation; dentro, en la parte del Training se añade el operador Neural Net, y en la parte de Testing el operador Apply Model y el operador Performance para conocer el AUC del modelo.





F. Evaluación del Modelo

En cuanto a la evaluación de los modelos, una vez entrenados y probados ambos modelos se incluyó el operador Performance para conocer el AUC de cada uno de ellos. El AUC es una métrica que evalúa los modelos (en este caso predictivos), donde un AUC más alto indica un mejor rendimiento global del modelo. En el caso del Naive Bayes el AUC resultante del modelo fue de 92,1%.

En el caso del modelo de Redes Neuronales, el cual se realizó mediante un operador Cross Validation, como se ha explicado anteriormente; se probó con los Training Cycles, Learning Rate y Momentum predeterminados. Se añadió también al final el operador Performance con el objetivo de conocer el AUC de este modelo, que en este caso fue de 95,29%.

Es por esto por lo que se optó por el modelo de Redes Neuronales mediante un Cross Validation, dando lugar al siguiente paso: optimización de los Training Cycles, Learning Rate y Momentum de este modelo.

G. Validación y Optimización del Modelo

En cuanto a la optimización del modelo, una vez que opta definitivamente por utilizar un modelo predictivo de Redes Neuronales, se debía determinar el 'Learning Rate' y 'Momentum' óptimos para este modelo.

Por un lado, una tasa de aprendizaje más alta puede acelerar el proceso de entrenamiento, pero también puede hacer que el modelo sea menos preciso. Por otro lado, una tasa de aprendizaje más baja puede mejorar la precisión, pero el entrenamiento puede ser más lento. Es por esto esencial encontrar un equilibrio para garantizar una convergencia efectiva del modelo.

En cuanto al 'Momentum', ayuda a acelerar el entrenamiento al agregar un término proporcional a la velocidad anterior de los parámetros. Esto ayuda a superar los mínimos locales y converge de manera más eficiente hacia un mínimo global. Un valor de Momentum más alto implica una mayor persistencia en la dirección del movimiento.

Como se puede ver en la siguiente tabla, se ha aplicado al modelo cada una de las posibles combinaciones de Learning Rate y Momentum, cuyos valores están comprendidos entre 0.1 y 0.5, con el objetivo de encontrar el AUC más alto. Esto se ha realizado con tan sólo 50 ciclos de entrenamiento (Training Cycles) para agilizar el proceso.

		LEARNING RATE				
	AUC (%)	0.1	0.2	0.3	0.4	0.5
MOMENTUM	0.1	95,2	95.8	95,7	95,8	95,7
	0.2	95,4	95.8	95,9	95.7	95,7
	0.3	95,5	95.8	95,8	95.7	95,6
	0.4	95,5	95.8	95,7	95,8	95,2
	0.5	95,7	95,8	95,7	95,3	95,3

Una vez se ha comprobado que el AUC máximo se obtiene con un 'Learning Rate' = 0.3 y un 'Momentum' = 0.2, obteniendo un AUC de 95.9%, se procede a optimizar el número de Training Cycles. Esto es importante para evitar el sobreajuste, causado por utilizar demasiados ciclos (overfitting); o el subajuste (underfitting) del modelo, causado por utilizar muy pocos ciclos.

Se empieza usando 500 ciclos de entrenamiento, ya que el número recomendado es entre 500 y 10000. Empezando con 500 ciclos, se obtiene un AUC del 97.7%. Posteriormente, con 1000 ciclos se obtiene un AUC del 98.4%, y con 1500 ciclos se obtiene un AUC de 98.7%. Por último, con 2000 ciclos se obtiene un AUC de 98.8%. Observando estos resultados, se decide optar por el de 1500 ciclos de entrenamiento, ya que a partir de ahí el AUC deja de aumentar significativamente al incrementar los ciclos, y se podría incurrir en un problema de overfitting mencionado anteriormente.

TRAINING CYCLE 500	97,7
TRAINING CYCLE 1000	98,4
TRAINING CYCLE 1500	**98,7**
TRAINING CYCLE 2000	98,8

Repositorio del código GITHUB

https://github.com/victormontilla/PwC-Challenge

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.

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.