Parte 1

Paso 1: Adquisición de Datos

En la vida real, nos habríamos conectado a la base de datos del banco para extraer información sobre las transacciones de tarjetas de crédito. Sin embargo, debido a la sensibilidad de los datos y para los fines de este proyecto, hemos utilizado un conjunto de datos de Kaggle (Enlace al conjunto de datos).

Descargamos el archivo CSV que contiene las festividades nacionales para el país correspondiente, en nuestro caso, los Estados Unidos (Enlace de Fuente, Enlace de Fuente).

Recopilamos datos de desempleo a nivel estatal.
[https://www.kaggle.com/datasets/justin2028/unemployment-in-america-per-us-state]

Obtenemos tasas de criminalidad por estado.

Reunimos datos del mercado de valores para calcular el tipo de cambio entre el dólar y el euro y la evolución del S&P.
[https://data.ecb.europa.eu/data/datasets/EXR/EXR.D.USD.EUR.SP00.A]
[https://www.marketwatch.com/investing/index/spx/download-data]

Guardamos todos estos conjuntos de datos en Google Drive.

Paso 2: Análisis Exploratorio de Datos

Realizamos un análisis exploratorio de los datos combinados para comprender sus características, distribuciones, valores atípicos, correlaciones, etc.
En esta etapa, dividimos el conjunto de datos en conjuntos de entrenamiento y pruebas utilizando la técnica "train-test split".

Paso 3: Preprocesamiento de Datos

Aplicamos las transformaciones necesarias en ambos conjuntos (train y test).

Codificamos las variables no numéricas y guardamos el conjunto de entrenamiento procesado con todas las variables en formato numérico. Aquí separamos 10.000 filas del dataset original y las guardamos un csv aparte (last_10000_rows.csv) para hacer predicciones más adelante.

Paso 4: Evaluación y Optimización del Modelo Inicial

Evaluamos el rendimiento del modelo inicial.

Ajustamos los hiper parámetros y optimizamos el modelo en Google Colab.

Guardamos el modelo ajustado, listo para su implementación, en Google Colab.

Paso 5: Subida a GitHub

Subimos los cuadernos y los conjuntos de datos de train y test a GitHub.

Parte 2

Paso 6: Descarga desde GitHub y Almacenamiento en Google Cloud

Descargamos los cuadernos desde GitHub junto con los archivos CSV necesarios que contienen datos.

Luego, almacenamos estos archivos en el depósito de Google Cloud Storage (bucket).

Paso 7: Preparación para la Implementación

Utilizamos el servicio en la nube de Google, Google Cloud Storage, para almacenar el modelo serializado o convertido en un bucket. Esto facilita la implementación y gestión eficiente del modelo.

Paso 8: Implementación del Modelo

Utilizamos el servicio en la nube de Google, Vertex AI, para implementar el modelo de manera eficiente y escalable.

Antes de implementar el modelo en Vertex AI, debemos tenerlo almacenado en un bucket. Una vez guardado, registramos el modelo con Vertex y lo lanzamos en Vertex AI, que crea un endpoint para verificar si una transacción es fraudulenta o no.

Paso 8.1: Notificaciones

En caso de que una transacción resulte fraudulenta, enviamos una notificación por correo electrónico al usuario a través del servidor SMTP (Protocolo Simple de Transferencia de Correo) de Gmail.

Paso 8.2: Automatización

Para automatizar los pasos de adquisición, preparación, implementación e inferencia, utilizamos Cloud Scheduler, Pub/Sub y Cloud Functions.

Cloud Scheduler se utiliza para programar un trabajo, cada X tiempo, que descarga los datos y los almacena en Google Cloud Storage.

Pub/Sub se utiliza para enviar los datos desde Google Cloud Storage a un Cloud Function.

La Cloud Function se utiliza para preprocesar los datos, entrenar el modelo y registrarlo en Vertex AI.

Vertex AI se utiliza para implementar el modelo y enviar notificaciones por correo electrónico en caso de fraude.

Cloud Scheduler se utiliza para programar un trabajo que realiza inferencia por lotes utilizando el modelo implementado en Google Cloud.

Paso 9: Inferencia por Lotes

Por otro lado, también a través de Vertex AI, utilizamos el servicio en la nube de Google, Google AI Platform (Batch Inference), para realizar inferencia por lotes utilizando el modelo implementado en Google Cloud. Esto se hace con conjuntos de datos almacenados en un depósito de almacenamiento en la nube de Google y lleva los resultados al bucket de nuevo.