Universidad San Francisco de Quito

UNIVERSIDAD SAN FRANCISCO

Nombre: Cesar Carrera – 00344613

Fundamentos de Ciencia de Datos

Proyecto Final

Para este proyecto se ha decidido usar el dataset de Fraud Detection. A continuación, se detallarán los pasos a seguir usando la metodología CRISP-DM para tener una base limpia y útil para el desarrollo del proyecto.

Entendimiento del negocio, contexto y alcance

En el contexto del negocio, tomaremos el papel de un banco el cual tiene registros de diferentes transacciones bancarias recopiladas de diferentes ciudades. Esta información indica si la operación realizada es fraudulenta o no. El hecho de que una operación sea fraudulenta hace que el banco incurra en costos operacionales como lo son las emisiones de nuevas tarjetas de crédito, débito, chequeras, etc. También las consecuencias de estas operaciones hacen que los clientes pierdan confianza en el banco lo cual puede ocasionar pérdida de estos y puede ocasionar una mala reputación en general. Poder determinar si una operación será fraudulenta o no ayudará al banco a mitigar estos costos asociados y tener una mejor retención de socios.

Para esta etapa inicial se utilizará una base de datos con información limitada para tener una prueba inicial del modelo y analizar su viabilidad. Se tienen registros de las ciudades principales donde se realizan la mayoría de las transacciones, el alcance de tiempo será de únicamente 6 meses. En cuanto al tipo de transacciones, se eligieron solamente aquellas que son de carácter virtual ya que si se incluyen las presenciales pueden incurrir otro tipo de errores de carácter humano.

Entendimiento de los datos

Los datos tienen un registro único dado por la columna ‘transaction_id’ por lo que no hay información duplicada. Para cada uno de los registros hay un id de cliente único para saber cualquier tipo de información relevante. El objetivo de este proyecto es lograr predecir si una

operación va a ser fraudulenta en el futuro analizando los datos actuales. Se usarán algoritmos de clasificación para poder saber este resultado. La información se ha obtenido de una base disponible en línea. Se encuentran diferentes tipos de datos tanto categóricos como numéricos los cuales serán descritos en el archivo de EDA. A continuación, se presentan los hallazgos iniciales acerca de la información disponible.

Información faltante:

```
RangeIndex: 51000 entries, 0 to 50999
Data columns (total 12 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   transaction_id                  51000 non-null  object
 1   user_id                         51000 non-null  int64
 2   transaction_amount              48480 non-null  float64
 3   transaction_type                51000 non-null  object
 4   time_of_transaction             48448 non-null  float64
 5   device_used                     48527 non-null  object
 6   location                        48453 non-null  object
 7   previous_fraudulent_transactions  51000 non-null  int64
 8   account_age                     51000 non-null  int64
 9   number_of_transactions_last_24h  51000 non-null  int64
 10  payment_method                  48531 non-null  object
 11  fraudulent                      51000 non-null  int64
dtypes: float64(2), int64(5), object(5)
memory usage: 4.7+ MB
```

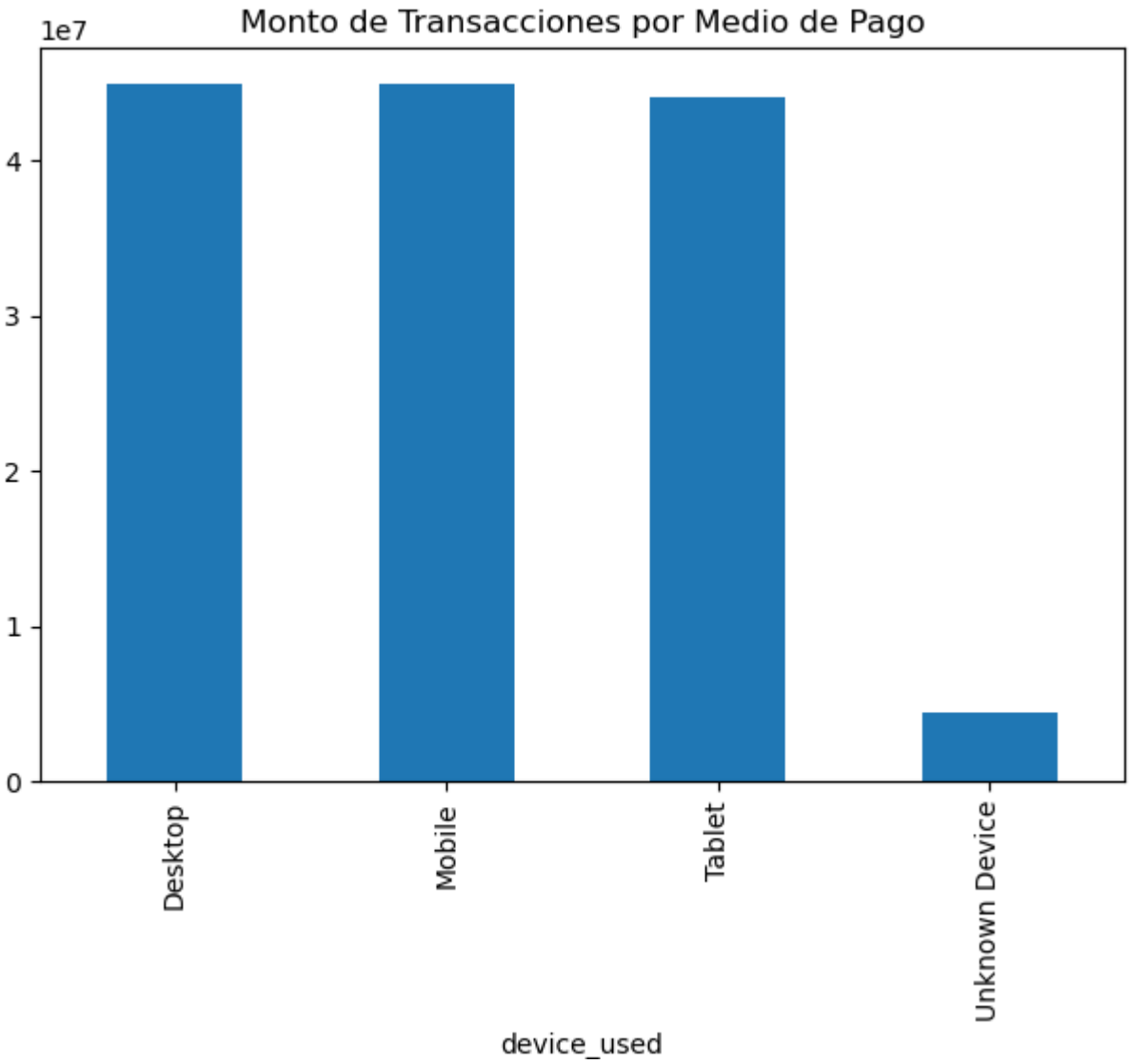
Conteo de nulos:

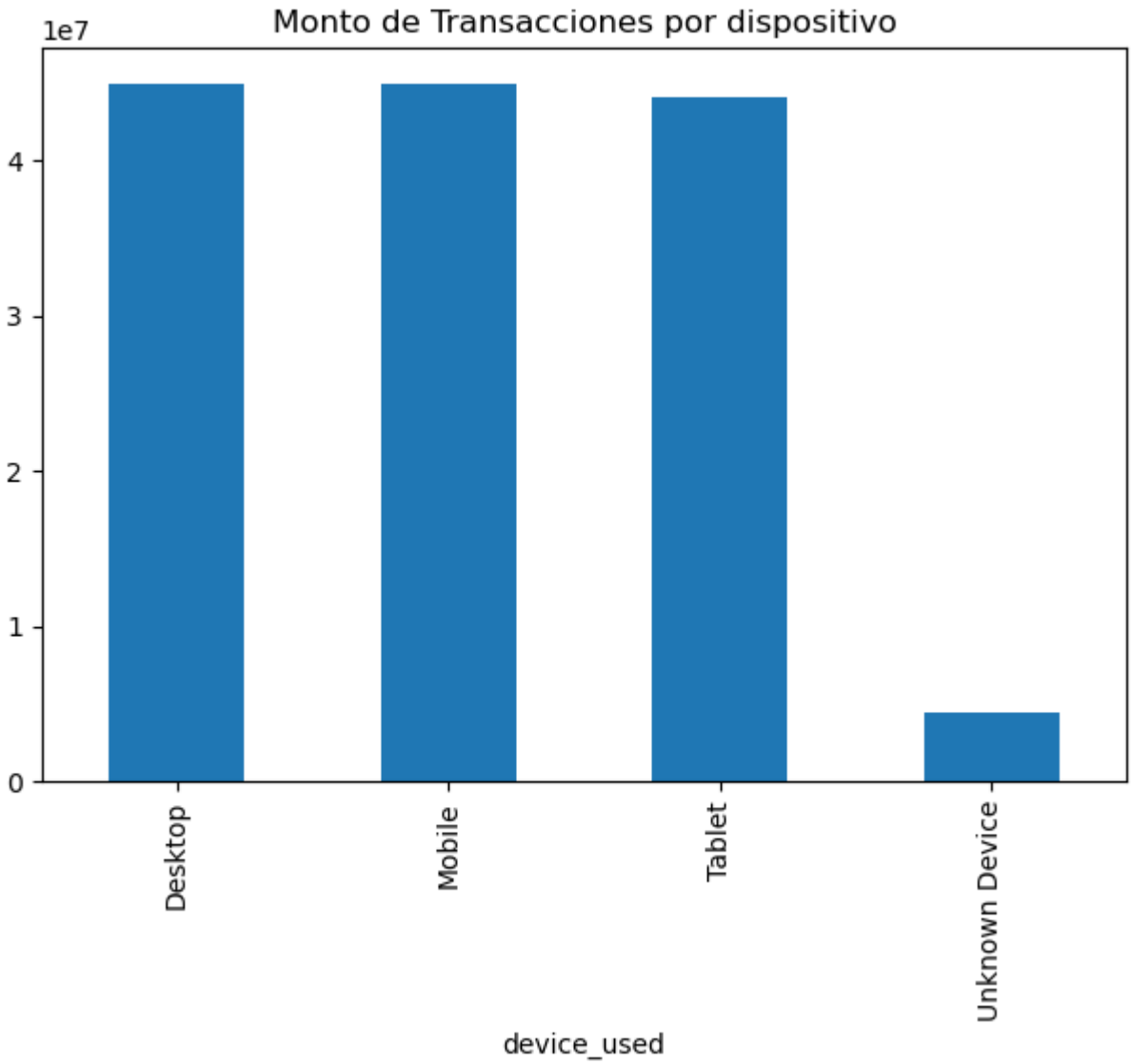
```
Valores nulos:
 transaction_id                     0
user_id                             0
transaction_amount                2520
transaction_type                    0
time_of_transaction               2552
device_used                       2473
location                          2547
previous_fraudulent_transactions     0
account_age                         0
number_of_transactions_last_24h     0
payment_method                    2469
fraudulent                          0
dtype: int64
```

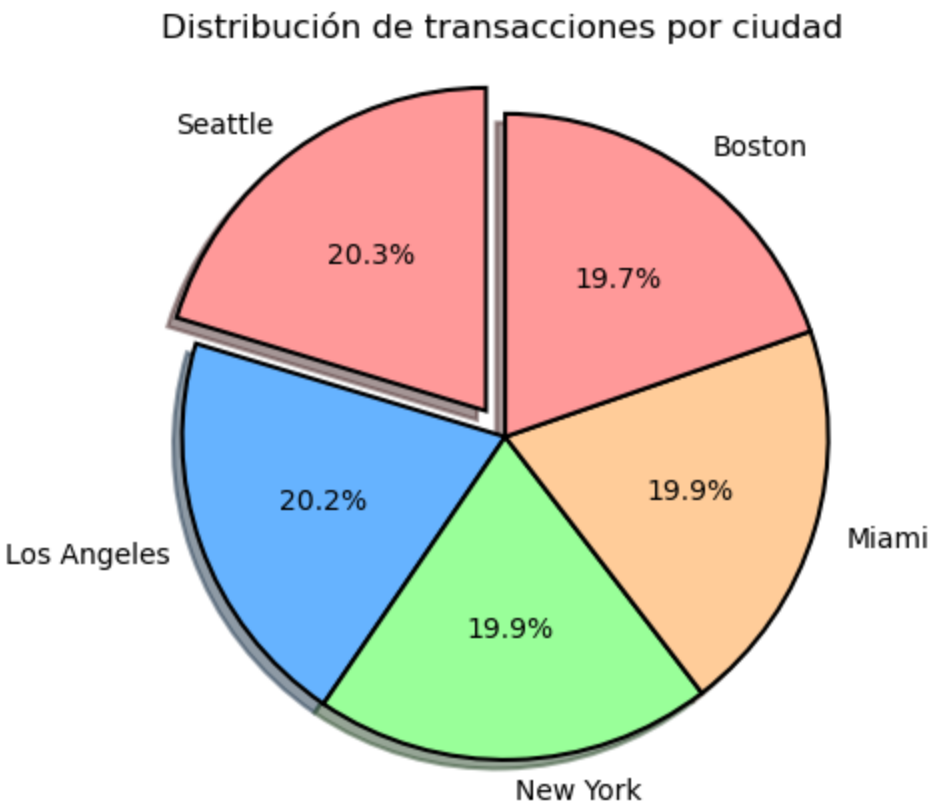
Estadísticos descriptivos generales:

```
Estadisticos descriptivos:
            user_id  transaction_amount  time_of_transaction  \
count  51000.000000        48480.000000         48448.000000
mean    3005.110176         2996.249784            11.488400
std     1153.121107         5043.932555             6.922954
min     1000.000000            5.030000             0.000000
25%     2007.000000         1270.552500             5.000000
50%     2996.000000         2524.100000            12.000000
75%     4006.000000         3787.240000            17.000000
max     4999.000000        49997.800000            23.000000

       previous_fraudulent_transactions   account_age  \
count                      51000.000000  51000.000000
mean                           1.995725     60.033902
std                            1.415150     34.384131
min                            0.000000      1.000000
25%                            1.000000     30.000000
50%                            2.000000     60.000000
75%                            3.000000     90.000000
max                            4.000000    119.000000

       number_of_transactions_last_24h    fraudulent
count                     51000.000000  51000.000000
mean                          7.495588      0.049216
...
25%                           4.000000      0.000000
50%                           7.000000      0.000000
75%                          11.000000      0.000000
max                          14.000000      1.000000
```

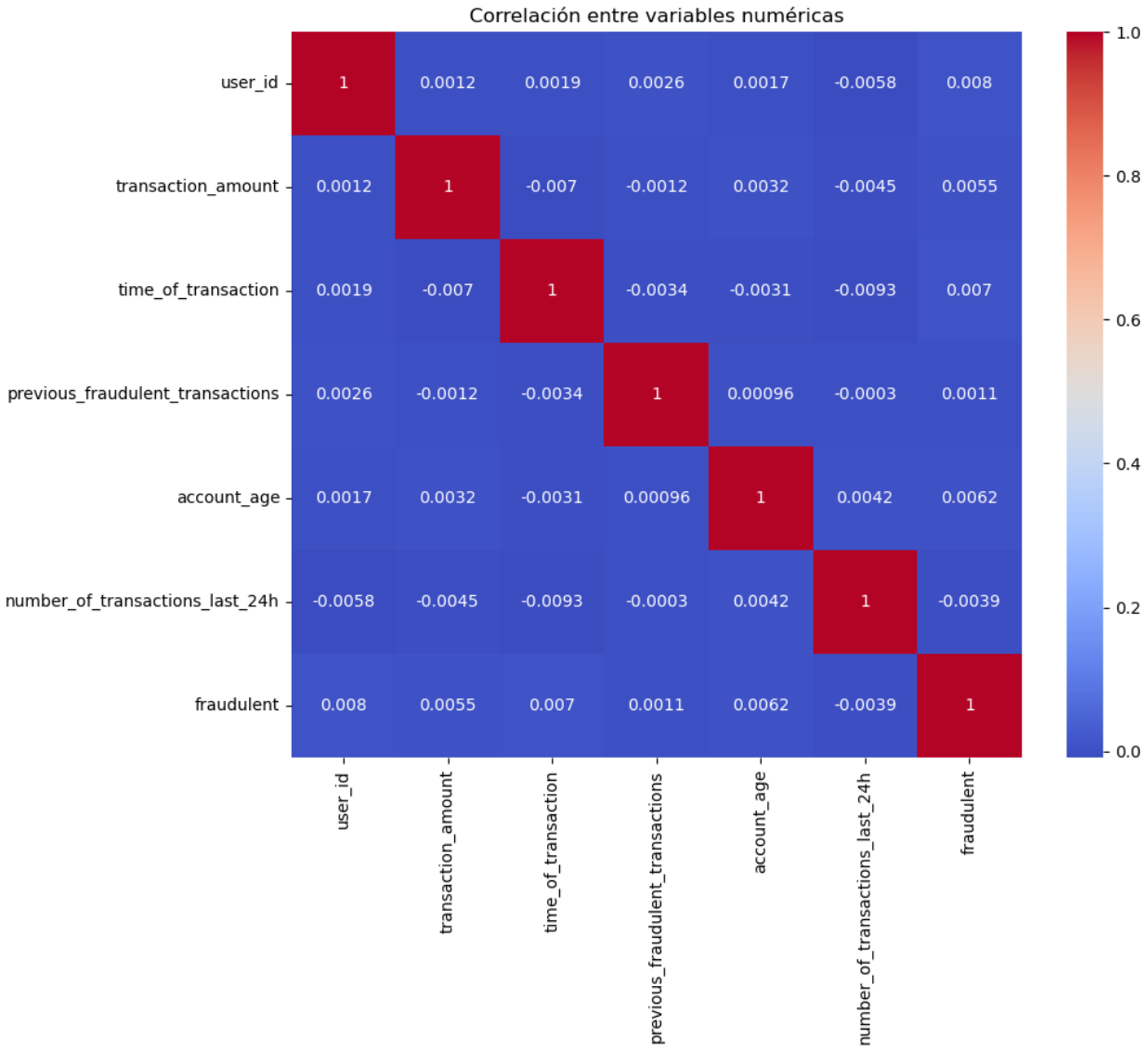
Información general de variables:


Monto de Transacciones por Medio de Pago
1e7
4
3
2
1
0
Desktop
Mobile
Tablet
Unknown Device
device_used


Monto de Transacciones por dispositivo
1e7
4
3
2
1
0
Desktop
Mobile
Tablet
Unknown Device
device_used



Matriz de correlación inicial:



Preparación de datos

Para la preparación de datos se realizarán diferentes técnicas de limpieza de datos, se identificarán duplicados, valores nulos y tipos de datos. Esta tarea se realizará en el archivo de data_wrangling. La tarea más importante en este paso fue la imputación de valores nulos. Se tuvo dos casos diferentes:

- Variables categóricas: para estas variables lo que se hizo fue llenar la información vacía con ‘unknown’ ya que no hay otra característica que defina a esos valores nulos.
- Variables numéricas: Para estas variables se tuvo dos casos diferentes. Para la variable ‘transaction_amount’ se decidió llenar los valores nulos con la media por ciudad ya que nuestro objetivo es intentar replicar el comportamiento de consumo demográfico. Por otra parte, para la variable ‘time_of_transaction’ se reemplazaron los valores nulos con

la media de tiempo por dispositivo. Se realizó de esta manera porque el tiempo de uso por dispositivo es lo más cercano a la realidad que se tiene para imputar este dato.

Feature Engineering

Para esta sección la cual se encuentra en el archivo de feature_engineering.ipynb, lo que se hizo fue eliminar la columna 'transaction_id' ya que, si bien es informativa, no agrega valor a los datos finales ya que un valor único de transacción no aporta a que una operación sea fraudulenta o no.

Luego de eso, se creó una nueva variable llamada 'fraud_per_month' la cual es el resultado de la relación entre la antigüedad de la cuenta (meses) para el número de transacciones fraudulentas de la persona. Esto quiere decir que mientras más bajo sea el valor, más transacciones fraudulentas ha hecho esa persona.

Finalmente, se usó la técnica de one hot encoding para poder codificar las variables categóricas y poder usarlas en el modelo.

Modelado

Para la parte de modelado, se decidió usar los siguientes algoritmos:

- Regresión logística: Se usa esta técnica como un punto de partida ya que es la más simple para interpretar resultados, principalmente para relaciones entre variables independientes lo cual se tiene como hipótesis para el desarrollo de este proyecto.
- Árboles de decisión: Este modelo ayuda principalmente cuando se cree que se tiene relaciones no lineales entre las variables, la desventaja es que puede a tender a un sobreajuste así que se deberá tener en cuenta eso.
- Random Forest: Usaremos este modelo para dar un poco más de análisis a los árboles de decisión ya que ayuda a reducir el overfitting de los otros modelos y se pueden ajustar los hiper parámetros para encontrar la mejor solución. Si bien es cierto que este modelo es más complejo y su interpretabilidad es más difícil, la ventaja es que es más robusto al momento de encontrarse con diferentes tipos de datos.

La partición de datos se la hizo a través de la metodología train/test split. Esto se decidió ya que es un modelo de prueba inicial y nos ayuda a generalizar la información. En cuanto a las métricas de evaluación, se usaron las siguientes:

- Recall: Es la más importante para medir el desempeño del modelo ya que esta medida nos ayuda a saber todos los fraudes reales positivos que encuentra el modelo. En este contexto, nos sirve saber la mayor cantidad de valores positivos porque el costo de no encontrar estos valores es alto en términos de dinero. Como negocio, el banco prefiere 'molestar' a un cliente por un falso caso positivo que no identificar un caso real de fraude.

- Precisión: Por otra parte, esta medida nos ayuda a saber con exactitud los valores reales que si son ciertos. Por lo que, si este valor es alto, podré saber que la mayoría de los casos que identifique serán correctos.
- F1-score: Esto es una medida general que nos ayuda a saber la relación entre nuestras métricas anteriores, lo cual es útil porque podremos saber qué tan bien se desempeña el modelo no solo en entrenamientos, también con el set de prueba.

Paso previo antes de modelar

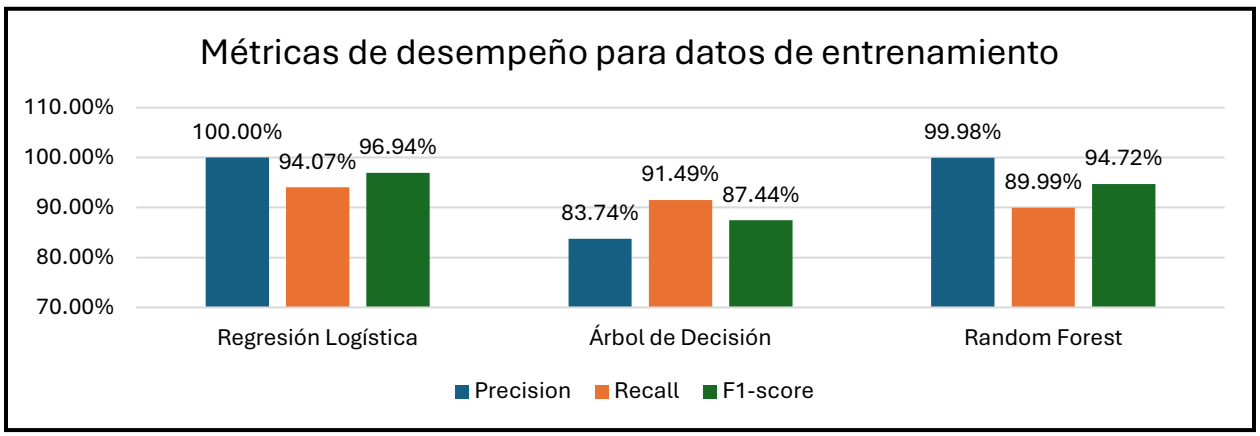
Antes de empezar a modelar, hay que tener en cuenta la distribución de los datos.

```
ml_data['fraudulent'].value_counts()
✓ 0.0s
0    48490
1     2510
Name: fraudulent, dtype: int64
```

Como se puede observar, la variable que etiqueta a las operaciones fraudulentas está desbalanceada lo cual puede sesgar a cualquiera de los modelos elegidos. Por esta razón, lo primero que se realizará es un sobre muestreo usando SMOTE, se eligió esta técnica porque nos ayuda a crear registros sintéticos en lugar de registros al azar para evitar duplicidad de los valores.

Resultados y comparación de modelos

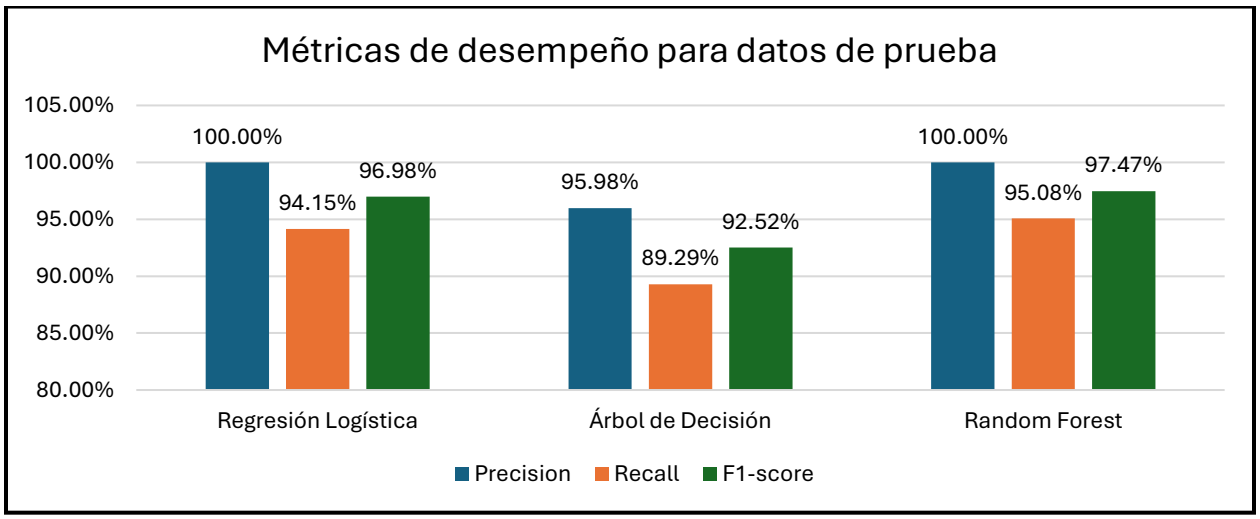
Luego de haber elegido los modelos, se puede empezar a comparar las métricas para poder determinar cuál sería el idóneo para seguir usándolo. A continuación, se muestran los resultados de desempeño para el conjunto de datos de entrenamiento:



Como se puede observar en la imagen de arriba, el modelo de regresión logística parece ser el mejor ya que cuenta con una precisión del 100% y su recall es el más alto de todos. Sin embargo,

es importante medir estos resultados con el set de prueba para evaluar el desempeño con nueva información.

A continuación, se muestran los resultados de los modelos con el set de prueba:



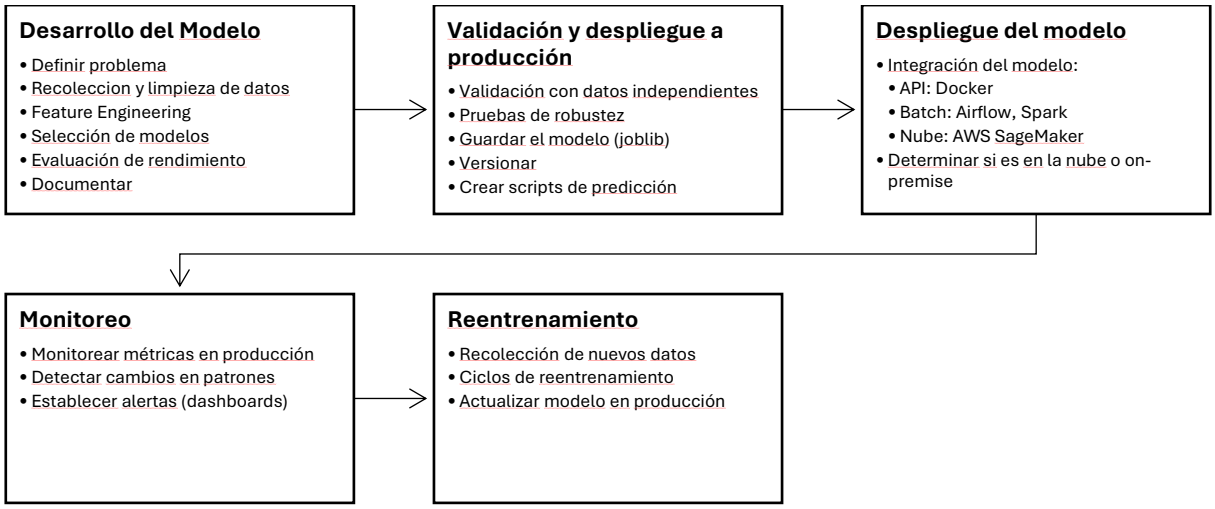
Como se puede observar, hay un cambio bastante significativo en las métricas de desempeño ya que, si bien es cierto, el modelo de regresión logística se mantiene casi igual, el modelo de random forest mejora significativamente sus valores incluso superando el desempeño de la regresión logística.

Dado que el modelo de random forest es más robusto porque sirve para diferentes tipos de datos, tiene una probabilidad menor de sobreajuste y ha presentado un mejor desempeño en el set de prueba, este será el modelo elegido.

Como apartado adicional, es importante saber que el resultado obtenido es con un proyecto de prueba por lo que lo más recomendable es tener un dataset más grande y si es posible con más features para saber si de verdad es el mejor modelo.

Plan de implementación

A continuación, se presentará un plan de despliegue a nivel general del proyecto con las instancias necesarias para poder tener un mapeo adecuado del proceso de implementación:



Este plan de despliegue sigue una metodología general la cual está diseñada para tener una estandarización de proceso y, a su vez, diferentes opciones de las herramientas necesarias para un óptimo despliegue. Para este contexto, la forma adecuada de hacer un despliegue correcto es usar técnicas de MLOps con el objetivo de saber en dónde se puede mejorar al modelo a futuro. El tipo de algoritmo usado puede cambiar, así como la cantidad de features del modelo incluso el tiempo de reentrenamiento es importante registrarlo; normalmente para estos proyectos, un tiempo estándar de 6 meses es recomendable para hacer un reentrenamiento.

Conclusiones y recomendaciones

Se logró encontrar el mejor modelo de clasificación siendo este un random forest el cual nos da las mejores métricas de desempeño. Los datos usados son una prueba de factibilidad del proyecto por lo cual es recomendable usar más información y variables para ver la viabilidad del proyecto usando este algoritmo ya que puede haber limitaciones computacionales.

Por otra parte, se recomienda realizar otras técnicas de modelado, ya sea ensamblaje o stacking ya que se pudo observar que el algoritmo de regresión logística es bastante adecuado. Se pueden unir estos dos modelos para sacar un mejor rendimiento del que se tiene en este momento.

Repositorio en GitHub

cesarcarrera96/Proyecto-Final