



Universidad Internacional San Isidro Labrador

Programa de Ciencia de Datos

Proyecto Final - Parte I

**Preparación y Limpieza de Datos para el Análisis de Deserción de Clientes en
una Institución Bancaria Costarricense**

Profesor: Dr. Samuel Saldaña Valenzuela

Estudiante: Byron Bolaños Zamora

Fecha de entrega: 11 de noviembre del 2024

2024

Índice

Introducción	2
Objetivos	3
Marco Teórico	4 - 39
Conclusiones	40
Recomendaciones	41
Bibliografía	42

Introducción

Con el aumento de la competencia en el sector financiero, la retención de clientes ha pasado a ser una prioridad estratégica para los bancos. En Costa Rica, una institución bancaria busca comprender mejor las razones por las cuales sus clientes deciden abandonar sus servicios, con el propósito de reducir esta deserción y fortalecer la fidelización.

La pérdida de clientes no solo implica una disminución de ingresos, sino que también eleva significativamente los costos de adquisición de nuevos clientes, lo cual impacta la estabilidad financiera y la sostenibilidad a largo plazo de la entidad. Este proyecto se enfoca en organizar y depurar datos, utilizando un conjunto de información diseñado para anticipar los patrones de cambio de banco de los clientes. El presente proyecto, se centra en revisar y corregir los datos para eliminar cualquier inconsistencia o sesgo que pudiera afectar el rendimiento del modelo de aprendizaje automático en una etapa posterior del proyecto. Siguiendo el proceso CRISP-DM, según IBM (2021) “CRISP-DM, que son las siglas de Cross-Industry Standard Process for Data Mining, es un método probado para orientar sus trabajos de minería de datos.”

El análisis inicia con la comprensión del entorno bancario y las características del conjunto de datos, y luego avanza hacia la limpieza, transformación de variables y normalización. Esta estrategia asegura que el conjunto de datos final esté en condiciones óptimas para construir modelos predictivos que apoyen la planificación estratégica de la institución financiera, permitiéndole responder de forma más efectiva a las necesidades de sus clientes y a los desafíos del mercado actual.

Objetivos

Objetivo general

Preparar un conjunto de datos sobre deserción de clientes en una institución bancaria costarricense, realizando un análisis exploratorio y un proceso de limpieza exhaustivo, con el fin de garantizar que los datos estén en condiciones óptimas para el entrenamiento de un modelo de machine learning en la siguiente etapa del proyecto.

Objetivos específicos

1. Identificar y analizar las características principales de los datos mediante un análisis exploratorio de datos (EDA), determinando patrones y posibles relaciones entre variables que puedan impactar la deserción de clientes.
2. Detectar y corregir problemas de calidad en el conjunto de datos, tales como valores nulos, datos atípicos, y variables inconsistentes, para mejorar la integridad y confiabilidad del conjunto de datos.
3. Transformar las variables categóricas y normalizar los datos relevantes, preparando el conjunto de datos para el entrenamiento efectivo de un modelo predictivo en la fase siguiente.

Marco Teórico

La deserción de clientes es uno de los problemas más significativos en el sector bancario, ya que implica la pérdida de ingresos y la necesidad de realizar mayores esfuerzos en la retención de nuevos clientes. Con el aumento de la competencia y la disponibilidad de opciones financieras, es crucial que las instituciones bancarias comprendan mejor las razones detrás de la deserción de sus clientes. En Costa Rica, como en otros países de la región, los bancos están recurriendo a análisis avanzados de datos para predecir y mitigar este fenómeno, buscando no solo mejorar la retención de clientes, sino también optimizar sus estrategias de marketing y servicio al cliente.

En este contexto, el análisis de datos se ha convertido en una herramienta esencial para las instituciones financieras. El proceso de minería de datos permite extraer patrones y relaciones ocultas dentro de grandes volúmenes de información, lo que facilita la toma de decisiones basadas en datos precisos y confiables. Sin embargo, antes de aplicar cualquier técnica predictiva o analítica avanzada, es fundamental realizar un proceso de limpieza y preparación de los datos, el cual asegure que la información esté en las condiciones adecuadas para ser utilizada en los modelos de machine learning, esto se logra gracias a ejecutar la metodología EDA.

Según IBM (2021) “CRISP-DM, que son las siglas de Cross-Industry Standard Process for Data Mining, es un método probado para orientar sus trabajos de minería de datos.” Ampliamente utilizada en el campo de la minería de datos y el análisis predictivo, que proporciona una estructura flexible y estandarizada para abordar proyectos de análisis de datos. Esta metodología se compone de seis fases principales:

1. Comprensión del negocio.
2. Comprensión de los datos.
3. Preparación de los datos.
4. Modelado.
5. Evaluación.
6. Despliegue.

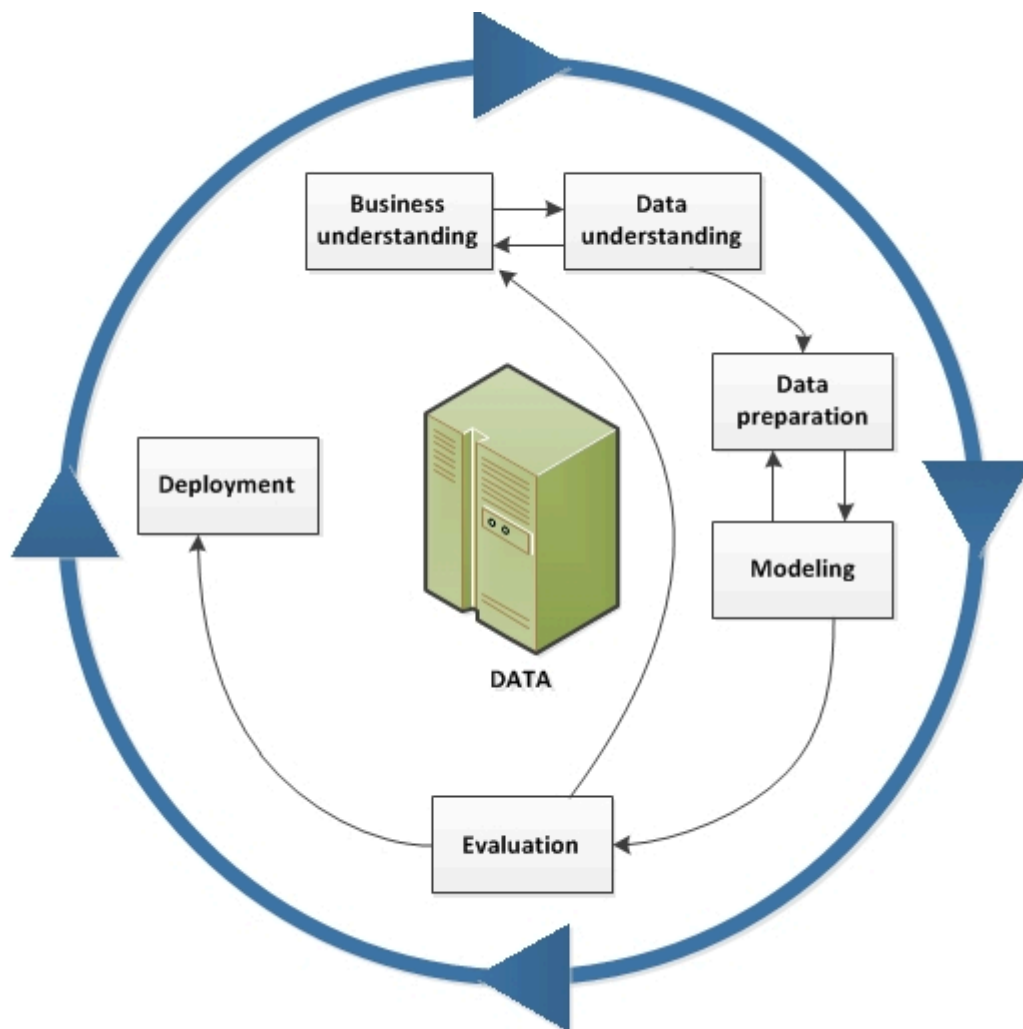


Figura 1: Ciclo de vida de minería de datos. Tomada de: <https://www.ibm.com>

En el contexto de la deserción de clientes en una institución bancaria, CRISP-DM ayuda a guiar el proceso de análisis desde la formulación del problema hasta la implementación de soluciones basadas en los resultados obtenidos.

El proceso comienza con la comprensión del negocio, donde se identifican los objetivos específicos, como la predicción de la deserción de clientes, y se establece el enfoque que se debe tomar para alcanzar estos objetivos. Luego, la comprensión de los datos consiste en recopilar y explorar el conjunto de datos disponible, donde el análisis exploratorio de datos (EDA) se convierte en una herramienta esencial para obtener una visión clara de las características del conjunto de datos. Este análisis es crucial para identificar patrones relevantes y posibles relaciones entre las variables que podrían influir en la deserción de clientes.

Una vez que se comprenden los datos, se inicia la fase de preparación de los datos, que implica la limpieza, transformación y normalización de los datos para asegurarse de que estén listos para el modelado predictivo. Este es un paso fundamental dentro de CRISP-DM, ya que garantiza que los datos sean consistentes, precisos y adecuados para construir modelos predictivos de calidad. La ejecución de EDA juega un papel central en esta fase, ya que permite detectar y corregir problemas en los datos, como valores nulos, datos atípicos e inconsistencias que podrían afectar la eficacia del modelo en etapas posteriores.

Según IBM (2024), "EDA se utiliza principalmente para ver qué datos se pueden revelar más allá de la tarea formal de modelado o de la prueba de hipótesis y proporciona una mejor comprensión de las variables del conjunto de datos y las relaciones entre ellas. También puede ayudar a determinar si las técnicas estadísticas que usted está considerando para el análisis de datos son adecuadas.

Desarrolladas originalmente por el matemático estadounidense John Tukey en la década de 1970, las técnicas EDA siguen siendo un método ampliamente utilizado en el proceso de descubrimiento de datos en la actualidad.”

El análisis exploratorio de datos (EDA) es una de las primeras etapas en cualquier proyecto de análisis de datos. A través de técnicas como la visualización y las estadísticas descriptivas, el EDA permite comprender las características de los datos y detectar patrones que podrían ser relevantes para la deserción de clientes. A lo largo de este marco teórico, se explorarán los conceptos clave relacionados con la minería de datos, el análisis exploratorio de datos, la limpieza y la transformación de variables, todos esenciales para preparar un conjunto de datos de calidad para su posterior uso en la fase de modelado predictivo.

La finalidad de este capítulo es proporcionar los fundamentos teóricos necesarios para entender cómo se deben preparar y transformar los datos antes de ser utilizados en un modelo de machine learning, con el objetivo de anticipar la deserción de clientes. A través de esta base conceptual, se pretende establecer las mejores prácticas para garantizar que el proceso de limpieza y preparación de datos sea exhaustivo y adecuado para asegurar la eficacia del modelo predictivo en etapas posteriores del proyecto.

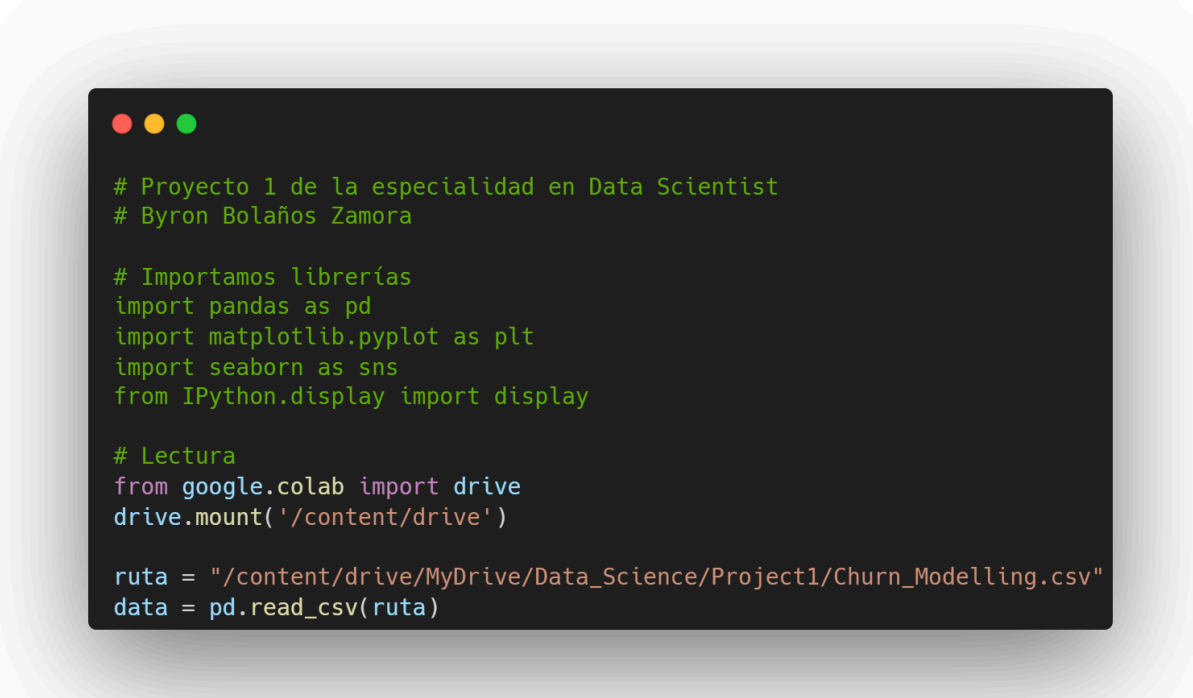
Basado en lo anterior, se extiende una explicación ampliada del aplicar CRISP-DM al dataset llamado Churn_Modelling.csv

1. Comprensión del negocio:

El primer paso en CRISP-DM es entender el contexto del negocio, los objetivos y los requerimientos del proyecto. En este caso, el objetivo es prever la deserción de clientes en una institución bancaria costarricense. Este análisis es fundamental para la institución, ya que permite identificar patrones en el comportamiento de los clientes y tomar decisiones informadas para retenerlos. Esto involucra estudiar las características de los clientes (edad, balance, productos contratados, etc.) y cómo estas se relacionan con la deserción.

2. Comprensión de los datos:

Aquí es crucial entender qué contiene el dataset, las variables disponibles y sus significados, así como la calidad de los datos.



```
# Proyecto 1 de la especialidad en Data Scientist
# Byron Bolaños Zamora

# Importamos librerías
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.display import display

# Lectura
from google.colab import drive
drive.mount('/content/drive')

ruta = "/content/drive/MyDrive/Data_Science/Project1/Churn_Modelling.csv"
data = pd.read_csv(ruta)
```

Figura 2 - Importación de librerías y montaje de drive. Elaboración propia.

En la figura 2, este código importa las librerías necesarias para el análisis de datos y la visualización, como `pandas`, `matplotlib.pyplot`, `seaborn`, y `IPython.display`. Luego, monta Google Drive en el entorno de Google Colab para poder acceder a los archivos almacenados allí. A continuación, especifica la ruta del dataset CSV llamado `Churn_Modelling.csv` en el Drive y lo carga en un DataFrame llamado `data` usando `pandas.read_csv()`, lo que permite trabajar con los datos del archivo para su análisis posterior.



Figura 3 - Visualización de data. Elaboración propia.

La figura 3, muestra dos operaciones principales: primero, utiliza `data.shape` para imprimir las dimensiones del DataFrame `data`, es decir, el número de filas y columnas que contiene el conjunto de datos. Luego, utiliza `data.head()` para visualizar las primeras 5 filas del DataFrame, lo que permite obtener una visión rápida de los primeros registros del dataset. Esto es útil para entender la estructura de los datos y verificar su contenido antes de realizar análisis más detallados.

Posteriormente genera el siguiente output:

RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

Figura 4 - OutPut del código de la figura 3. Elaboración propia.

Seguidamente, se ejecuta lo siguiente:



```
# Visualizamos las variables categóricas y numéricas
data.info()
```

Figura 5 - Visualización de variables. Elaboración propia.

En la figura 5, el código `data.info()` muestra un resumen del DataFrame, incluyendo el número de entradas, las columnas, el tipo de datos de cada columna (categóricas o numéricas) y la cantidad de valores no nulos en cada columna. Es útil para obtener una visión general de la estructura de los datos y detectar posibles valores faltantes.

Y el output generado, es el siguiente:

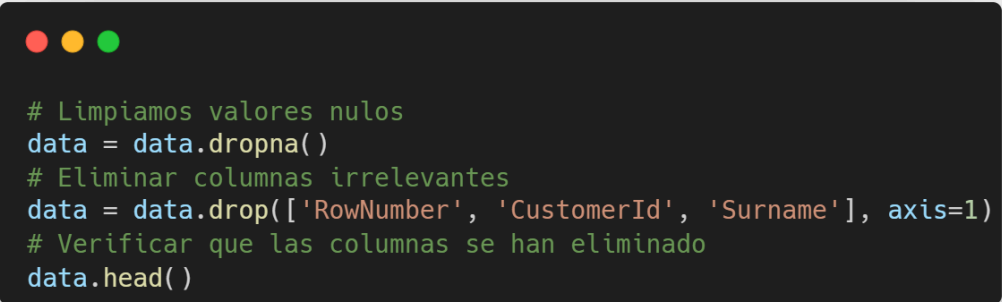
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   RowNumber             10000 non-null  int64
1   CustomerId            10000 non-null  int64
2   Surname               10000 non-null  object
3   CreditScore           10000 non-null  int64
4   Geography             10000 non-null  object
5   Gender               10000 non-null  object
6   Age                  10000 non-null  int64
7   Tenure               10000 non-null  int64
8   Balance              10000 non-null  float64
9   NumOfProducts        10000 non-null  int64
10  HasCrCard            10000 non-null  int64
11  IsActiveMember       10000 non-null  int64
12  EstimatedSalary       10000 non-null  float64
13  Exited               10000 non-null  int64
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
```

Figura 6 - Output del código de Figura 5. Elaboración propia.

En la figura 6, se muestra que el conjunto de datos no tiene valores nulos, ya que todas las columnas tienen 10,000 entradas no nulas, lo que garantiza la integridad de los datos. El dataset consta de 14 columnas, que incluyen variables numéricas como **CreditScore**, **Age**, **Balance**, **Tenure** y **EstimatedSalary**, así como variables categóricas como **Geography**, **Gender** y **Surname**. Además, hay variables binarias como **HasCrCard**, **IsActiveMember** y **Exited**, que indican la presencia de ciertos atributos o estados del cliente.

3. Preparación de los datos:

Esta fase es crucial porque los datos deben ser limpiados y transformados para ser aptos para el modelado. En tu código, esto se aborda de manera exhaustiva:



```
# Limpiamos valores nulos
data = data.dropna()
# Eliminar columnas irrelevantes
data = data.drop(['RowNumber', 'CustomerId', 'Surname'], axis=1)
# Verificar que las columnas se han eliminado
data.head()
```

Figura 7 - Limpieza de datos. Elaboración propia.

El código de la figura 7, elimina filas con valores nulos (aunque no se detectaron en el análisis previo) y elimina las columnas **RowNumber**, **CustomerId** y **Surname**, ya que no son relevantes para el análisis o la predicción de la variable **Exited**, que indica si un cliente ha abandonado o no el servicio. Las columnas eliminadas no

proporcionan información útil para predecir si un cliente se va (Exited), ya que `RowNumber` es solo un índice, `CustomerId` es un identificador único sin valor predictivo, y `Surname` es un dato personal que no influye directamente en la decisión de abandonar el servicio. Eliminar estas columnas optimiza el dataset, dejándolo con variables más relevantes para el análisis. Al final, se vuelve a mostrar el output del código, generando lo siguiente:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

Figura 8 - Visualización de datos sin variables innecesarias. Elaboración propia.



Figura 9 - Estadística descriptiva. Elaboración propia.

El código de la figura 9, mediante `data.describe()` genera un resumen estadístico descriptivo de las variables numéricas en el DataFrame, proporcionando

información clave como el conteo de valores, la media, la desviación estándar, los valores mínimo y máximo, y los percentiles (25%, 50%, 75%). Esta estadística es útil para entender la distribución y características de los datos, identificar posibles valores atípicos y obtener una visión general de la variabilidad y el rango de las variables numéricas antes de realizar un análisis más detallado o crear modelos predictivos.

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881	0.203700
std	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818	0.402769
min	350.000000	18.000000	0.000000	0.000000	1.000000	0.000000	0.000000	11.580000	0.000000
25%	584.000000	32.000000	3.000000	0.000000	1.000000	0.000000	0.000000	51002.110000	0.000000
50%	652.000000	37.000000	5.000000	97198.540000	1.000000	1.000000	1.000000	100193.915000	0.000000
75%	718.000000	44.000000	7.000000	127644.240000	2.000000	1.000000	1.000000	149388.247500	0.000000
max	850.000000	92.000000	10.000000	250898.090000	4.000000	1.000000	1.000000	199992.480000	1.000000

Figura 10 - Visualización de estadísticas. Elaboración propia.

El resumen estadístico generado que se muestra en la figura 10, mediante `data.describe()` ofrece información detallada sobre las variables numéricas en el conjunto de datos. A continuación, te explico cada columna y cómo se relaciona con la predicción de la variable `Exited` (abandono del servicio):

1. **CreditScore:**

- Rango: 350 a 850, con un promedio de 650.5.
- Este valor refleja la puntuación crediticia de los clientes, que puede influir en su decisión de abandonar el servicio. Por ejemplo, clientes con una puntuación baja pueden estar más inclinados a abandonar debido a dificultades económicas, lo que podría ser un factor clave en la predicción de `Exited`.

2. **Age:**

- Rango: 18 a 92 años, con una media de 38.9.
- La edad de los clientes también puede ser un factor relevante. Es posible que ciertos grupos etarios tengan mayor probabilidad de abandonar el servicio, ya sea por cambios en sus necesidades o comportamiento financiero.

3. **Tenure:**

- Rango: 0 a 10 años, con una media de 5.01.
- El tiempo de permanencia del cliente en el servicio (**Tenure**) es un dato clave. Los clientes con menor tiempo en el servicio (cerca de 0 años) pueden tener más probabilidades de abandonar, mientras que aquellos con mayor tenencia podrían ser menos propensos a irse.

4. **Balance:**

- Rango: 0 a 250,898.09, con una media de 76,485.89.
- El balance refleja la cantidad de dinero que tiene el cliente en su cuenta. Los clientes con balances más bajos o nulos podrían tener más probabilidades de abandonar, especialmente si sienten que no pueden aprovechar los beneficios del servicio.

5. **NumOfProducts:**

- Rango: 1 a 4 productos, con un promedio de 1.53.
- La cantidad de productos que un cliente tiene con la empresa también podría ser importante. Los clientes que usan más productos pueden estar más comprometidos con la empresa, mientras que los que tienen solo uno podrían ser más propensos a abandonar.

6. **HasCrCard (Tiene tarjeta de crédito):**

- Es una variable binaria (0 o 1), con un promedio de 0.7055, lo que significa que el 70.55% de los clientes tienen una tarjeta de crédito.

- Los clientes con tarjeta de crédito pueden tener una mayor fidelidad al servicio, por lo que esta variable podría ser importante para predecir si un cliente abandona o no.

7. **IsActiveMember (Es miembro activo):**

- También es binaria (0 o 1), con una media de 0.5151, indicando que el 51.51% de los clientes son miembros activos.
- Los clientes activos, que interactúan regularmente con la empresa, probablemente tengan menos probabilidades de abandonar el servicio, lo que hace que esta variable sea relevante en la predicción de **Exited**.

8. **EstimatedSalary:**

- Rango: 11.58 a 199,992.48, con una media de 100,090.24.
- El salario estimado de los clientes puede influir en su decisión de abandonar el servicio. Aquellos con salarios más bajos podrían tener una mayor probabilidad de abandonar debido a razones financieras.

9. **Exited:**

- Esta es la variable objetivo, con valores 0 (no ha abandonado) y 1 (ha abandonado).
- El análisis de las estadísticas de las otras variables ayuda a identificar patrones y tendencias que pueden predecir si un cliente está más o menos propenso a abandonar el servicio. Por ejemplo, si los clientes con un balance bajo, una edad más avanzada o una baja puntuación crediticia tienen más probabilidades de abandonar, estas variables se convierten en características importantes para predecir **Exited**.

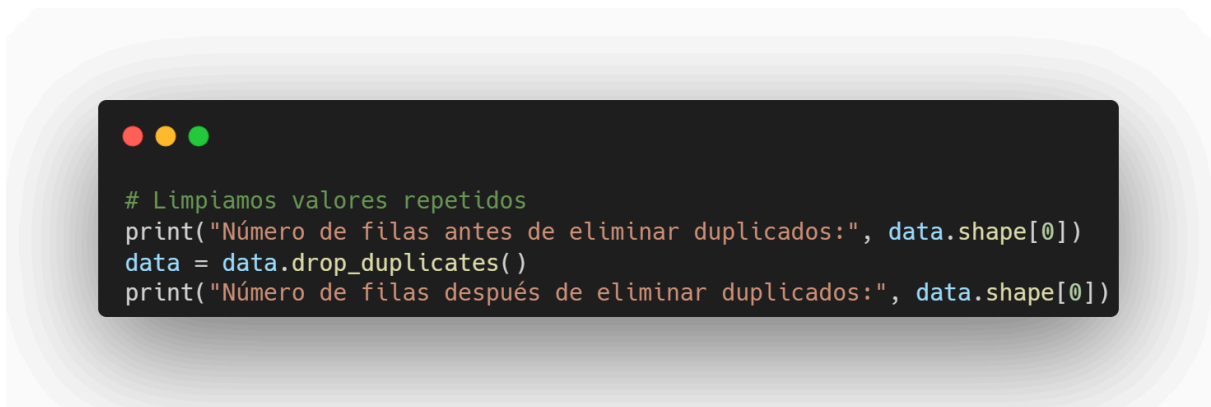


Figura 11 - Limpieza de valores repetidos. Elaboración propia.

El código elimina las filas duplicadas en el DataFrame, primero mostrando el número de filas antes y después de la operación. Esto asegura que cada fila sea única, evitando que los datos repetidos afecten el análisis o modelo de predicción. Generando el output de ello:

```
Número de filas antes de eliminar duplicados: 10000
Número de filas después de eliminar duplicados: 10000
```

Figura 12 - Output del código de Figura 11. Elaboración propia.

Basado en el resultado de la figura 12, es evidente que no habían valores repetidos en el dataset.

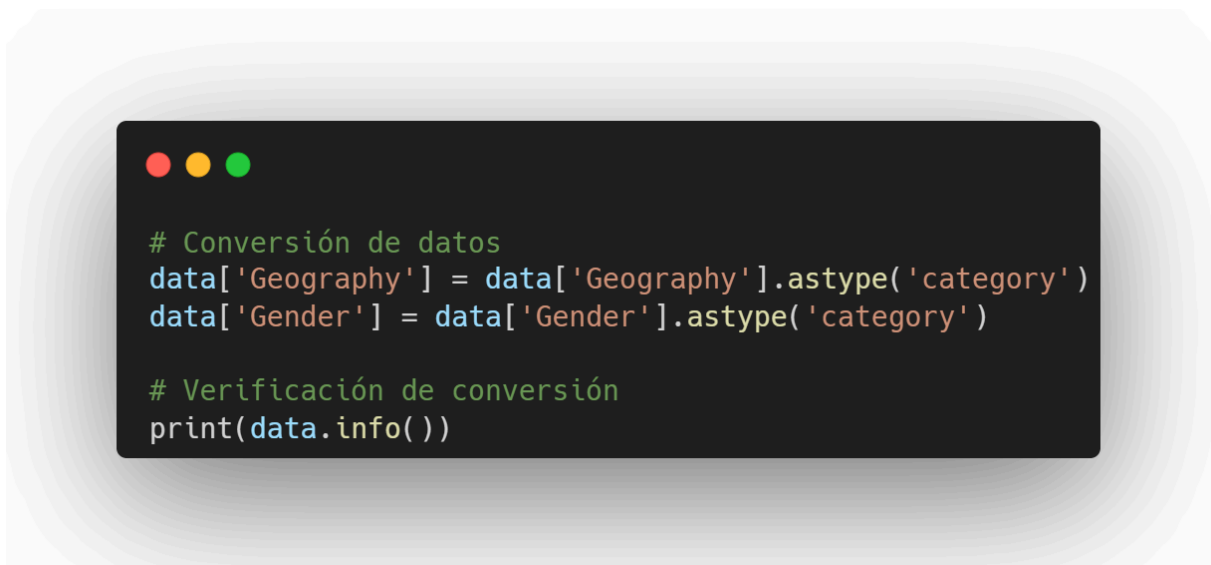


Figura 13 - Conversión de datos. Elaboración propia.

La conversión de las variables `Geography` y `Gender` a tipo `category` se prepara para aplicar la técnica de "dummy variables" o "variables ficticias". Las variables categóricas como `Geography` y `Gender` tienen valores limitados (por ejemplo, `Geography` puede tener valores como 'France', 'Germany' y 'Spain', y `Gender` puede ser 'Male' o 'Female'). Para usar estas variables en modelos de predicción, es necesario transformarlas en un formato numérico.

Las dummy variables convierten cada valor categórico en una columna binaria (0 o 1). Por ejemplo, para `Geography`, se crearían tres columnas separadas (`France`, `Germany`, `Spain`), donde se asigna un 1 en la columna correspondiente al país de cada cliente y 0 en las otras columnas. Esto permite que los modelos de machine learning interpreten correctamente las variables categóricas.

En el código, la conversión a `category` es un paso previo para facilitar esta transformación a dummy variables, que se puede realizar usando

`pd.get_dummies()`. Esto es importante porque permite que el modelo utilice la información de las variables categóricas de manera más eficiente, optimizando el rendimiento y la precisión del modelo predictivo en la variable `Exited`.



Figura 14 - Codificación de las variables Dummies. Elaboración propia.

Para la figura 14, se visualiza que se utiliza la función `pd.get_dummies()` para aplicar la técnica de codificación de variables categóricas a las columnas `Geography` y `Gender`. Al hacerlo, se crean nuevas columnas binarias (dummy variables) para cada valor único de estas variables, representando la presencia de cada categoría con un 1 y la ausencia con un 0. El parámetro `drop_first=True` elimina la primera categoría de cada columna, lo que previene la multicolinealidad, ya que las columnas generadas por `get_dummies()` son linealmente dependientes. Al eliminar la primera categoría, se asegura que el modelo no reciba información redundante.

Por ejemplo, para la columna `Geography`, si tenemos tres valores posibles (`France`, `Germany`, `Spain`), se generarán dos columnas adicionales: `Geography_Germany` y `Geography_Spain`, donde un valor de 1 indicará que el

cliente pertenece a esa categoría y 0 en caso contrario. La columna

`Geography_France` se eliminará para evitar colinealidad.

De todo ello, se obtiene lo siguiente:

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_Germany	Geography_Spain	Gender_Male
0	619	42	2	0.00	1	1	1	101348.88	1	False	False	False
1	608	41	1	83807.86	1	0	1	112542.58	0	False	True	False
2	502	42	8	159660.80	3	1	0	113931.57	1	False	False	False
3	699	39	1	0.00	2	0	0	93826.63	0	False	False	False
4	850	43	2	125510.82	1	1	1	79084.10	0	False	True	False

Figura 15 - Output de figura 14. Elaboración propia.

```
# Generamos gráficos tipo boxplot para identificar outliers

# Lista de columnas numéricas para analizar
numeric_columns = ['CreditScore', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'EstimatedSalary',
'Exited']

# Configurar el tamaño de las gráficas
plt.figure(figsize=(15, 10))

# Crear boxplots para cada columna numérica
for i, column in enumerate(numeric_columns, 1):
    plt.subplot(3, 3, i) # 3 filas, 3 columnas
    sns.boxplot(y=data_encoded[column])
    plt.title(f'Boxplot de {column}')
    plt.xlabel(column)

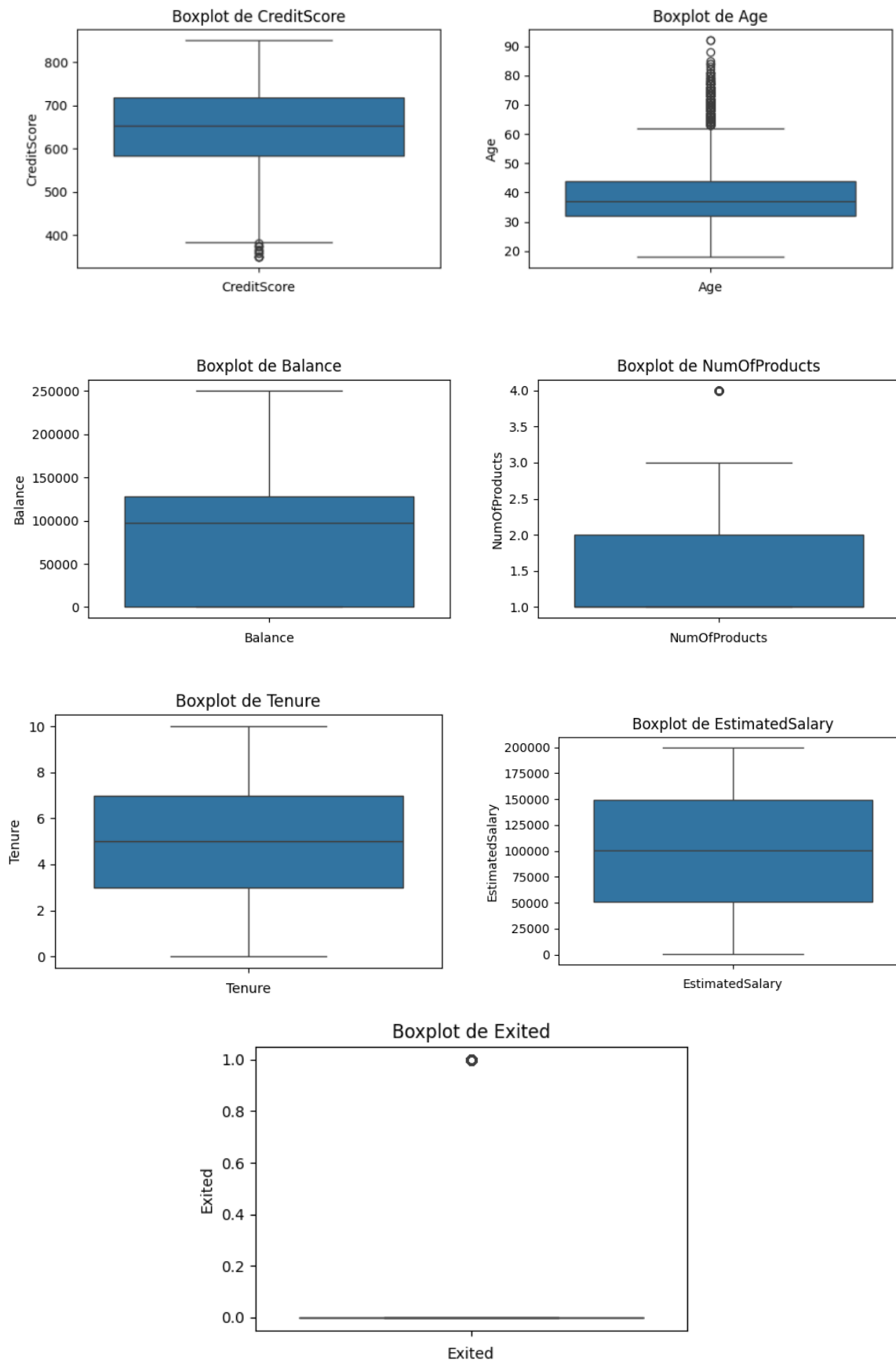
plt.tight_layout()
plt.show()
```

Figura 16 - Búsqueda de outliers. Elaboración propia.

En la figura 16, el código genera gráficos tipo boxplot para cada columna numérica del DataFrame `data_encoded`, con el fin de identificar valores atípicos (outliers).

Utiliza la librería `seaborn` para crear los boxplots y `matplotlib` para configurar el tamaño y disposición de las gráficas, mostrando un gráfico para cada columna en una cuadrícula de 3x3. El parámetro `tight_layout()` ajusta el espaciado para que las gráficas no se superpongan, y `plt.show()` las muestra en pantalla.

Basado en lo anterior, se generan gráficas tipo boxplot que vemos a continuación:



Las gráficas tipo boxplot, poseen:

Caja (box): Representa el rango intercuartílico (IQR), que abarca desde el primer cuartil (Q1, el 25% de los datos) hasta el tercer cuartil (Q3, el 75% de los datos). La caja muestra el 50% central de los datos.

Línea de la mediana (mediana): Dentro de la caja, hay una línea que indica la mediana (el 50% de los datos), que divide la caja en dos partes. Esta línea proporciona una medida de tendencia central.

Bigotes (whiskers): Los bigotes son líneas que se extienden desde los cuartiles hasta el valor máximo o mínimo dentro de un rango definido por 1.5 veces el IQR. Los valores fuera de este rango son considerados outliers.

Outliers (valores atípicos): Son puntos que se encuentran fuera de los bigotes, más allá de 1.5 veces el rango intercuartílico. Estos valores son considerados atípicos porque se alejan significativamente del resto de los datos.

Por ende, según sea necesidad, se deben eliminar aquellos valores atípicos que se consideren nocivos para el futuro modelo de machine learning que se desarrollará.

Por tal, se procede con la eliminación de aquellos outliers que se consideran que pueden afectar el desarrollo, específicamente son aquellos de CrediScore, Edad y número de productos. A continuación se visualiza en código:

```

# Calcular el primer cuartil (Q1) y el tercer cuartil (Q3) para CreditScore
original_size = data.shape[0]
Q1 = data['CreditScore'].quantile(0.25)
Q3 = data['CreditScore'].quantile(0.75)
IQR = Q3 - Q1

# Calcular los límites inferior y superior
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Filtrar el DataFrame para eliminar outliers de CreditScore
data = data[(data['CreditScore'] >= lower_bound) & (data['CreditScore'] <= upper_bound)]

# Verificar el tamaño del nuevo DataFrame
print(f"Tamaño original: {original_size}") # Tamaño original antes de la eliminación
print(f"Tamaño sin outliers: {data.shape[0]}")

# Verificamos luego de borrar outliers
plt.figure(figsize=(10, 6))
sns.boxplot(x=data['CreditScore'])
plt.title('Boxplot de CreditScore sin Outliers')
plt.show()

```

Figura 17 - Eliminación de outliers en CreditScore. Elaboración propia.

La figura 17, posee el código que calcula el primer cuartil (Q1) y el tercer cuartil (Q3) de la columna **CreditScore**, luego calcula el rango intercuartílico (IQR), que es la diferencia entre Q3 y Q1. Con el IQR, se determinan los límites inferior y superior para identificar los outliers, utilizando un factor de 1.5, que es un estándar común para detectar valores atípicos: valores fuera del rango $Q1 - 1.5 * IQR$ a $Q3 + 1.5 * IQR$ son considerados outliers. Luego, el código filtra los datos eliminando estos outliers y muestra el tamaño del DataFrame antes y después de la eliminación. Finalmente, genera un boxplot del **CreditScore** sin outliers para visualizar cómo ha cambiado la distribución. El factor 1.5 es una convención estadística que balancea bien la identificación de outliers sin eliminar demasiados datos.

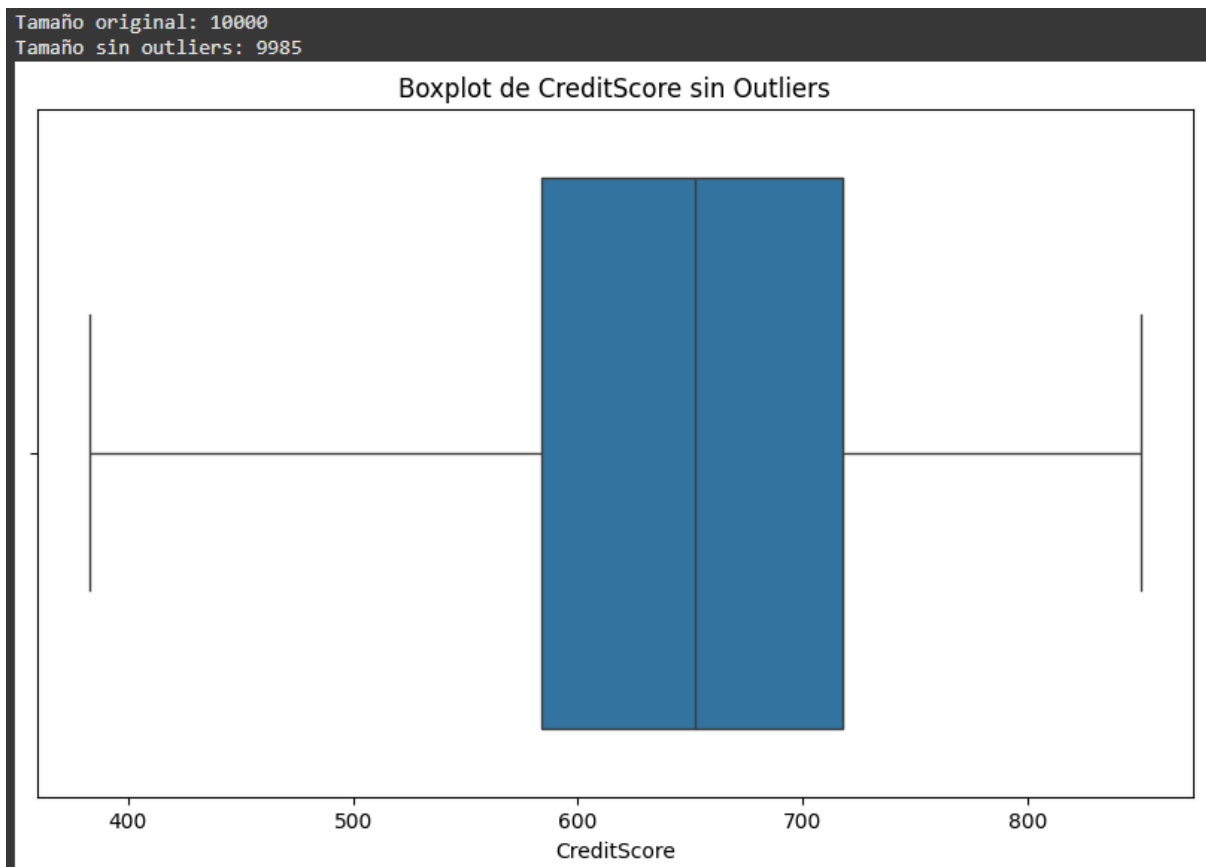


Figura 18 - CreditScore sin outliers. Elaboración propia.

En la figura 18, se presenta la gráfica tipo boxplot correspondiente a la variable **CreditScore**, donde se observa una reducción en el número de datos de 10,000 a 9,985. Esto indica que se han eliminado 15 valores atípicos (outliers) del conjunto de datos, los cuales se encontraban fuera del rango definido por el rango intercuartílico (IQR). Este proceso de eliminación de outliers es crucial para mejorar la calidad del modelo predictivo, al asegurar que los datos sean representativos y no estén sesgados por valores extremos que podrían afectar la precisión del análisis.

```

# Guardamos el tamaño original antes de eliminar outliers
original_size = data.shape[0]

# Calcular el primer cuartil (Q1) y el tercer cuartil (Q3) para Age
Q1 = data['Age'].quantile(0.25)
Q3 = data['Age'].quantile(0.75)
IQR = Q3 - Q1

# Calcular los límites inferior y superior usando el factor original de 1.2
lower_bound = Q1 - 1.2 * IQR
upper_bound = Q3 + 1.2 * IQR

# Filtrar el DataFrame para eliminar outliers de Age
data = data[(data['Age'] >= lower_bound) & (data['Age'] <= upper_bound)]

# Verificar el tamaño del nuevo DataFrame
print(f"Tamaño original: {original_size}") # Tamaño original antes de la eliminación
print(f"Tamaño sin outliers: {data.shape[0]}") # Tamaño después de eliminar outliers

# Verificamos luego de borrar outliers
plt.figure(figsize=(10, 6))
sns.boxplot(x=data['Age'])
plt.title('Boxplot de Edad sin Outliers')
plt.show()

```

Figura 19 - Eliminación de outliers en edad. Elaboración propia.

El código de la figura 19, realiza el mismo procedimiento para eliminar outliers, pero esta vez lo aplica a la columna **Age** (edad). Primero, se guarda el tamaño original del DataFrame antes de la eliminación de outliers. Luego, se calcula el primer cuartil (Q1) y el tercer cuartil (Q3) de la edad y se obtiene el rango intercuartílico (IQR). Con el IQR, se calculan los límites inferior y superior para definir los outliers. A diferencia del caso anterior, en este caso se usa un factor de 1.2 en lugar de 1.5 para los límites, lo que hace que se consideren más valores como dentro del rango, permitiendo una mayor flexibilidad en la detección de outliers. Después de filtrar los datos, se muestra el tamaño original y el nuevo tamaño del DataFrame. Finalmente, se genera un boxplot de la columna **Age** sin outliers para visualizar cómo cambió la

distribución de los datos. El uso del factor 1.2 sugiere una aproximación más estricta o conservadora, reduciendo más los valores del conjunto original.

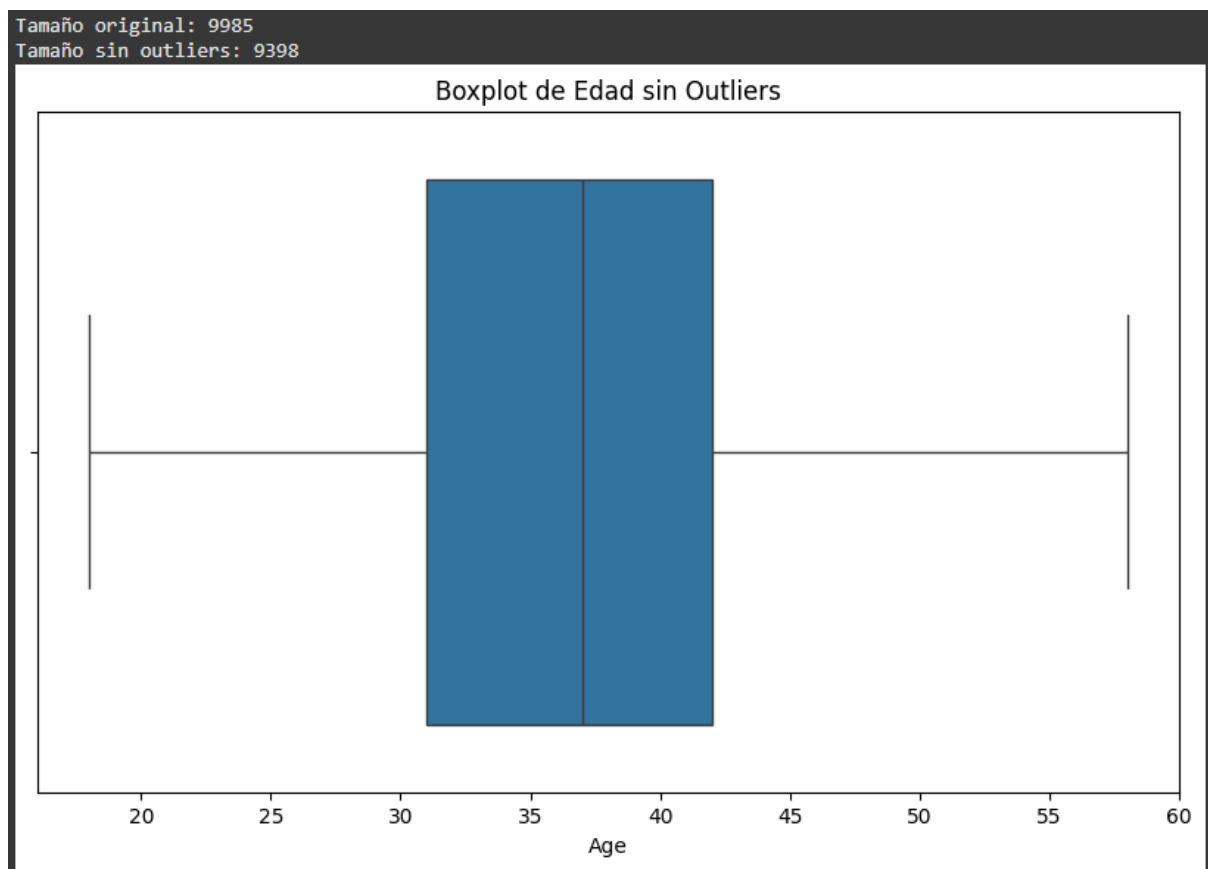


Figura 20 - Age sin outliers. Elaboración propia.

En la figura 20, se muestra la gráfica tipo boxplot de la variable **Age**, donde se observa una disminución en el número de datos de 9,985 a 9,398. Esto indica que se han eliminado 587 valores atípicos (outliers) de la columna de edad, los cuales se encontraban fuera del rango determinado por el rango intercuartílico (IQR) ajustado con un factor de 1.2.

```

# Guardamos el tamaño original antes de eliminar outliers
original_size = data.shape[0]

# Calcular el primer cuartil (Q1) y el tercer cuartil (Q3) para NumOfProducts
Q1 = data['NumOfProducts'].quantile(0.25)
Q3 = data['NumOfProducts'].quantile(0.75)
IQR = Q3 - Q1

# Calcular los límites inferior y superior usando un factor ajustable de 1.5
factor = 1.5
lower_bound = Q1 - factor * IQR
upper_bound = Q3 + factor * IQR

# Filtrar el DataFrame para eliminar outliers de NumOfProducts
data = data[(data['NumOfProducts'] >= lower_bound) & (data['NumOfProducts'] <= upper_bound)]

# Verificar el tamaño del nuevo DataFrame
print(f"Tamaño original: {original_size}") # Tamaño original antes de la eliminación
print(f"Tamaño sin outliers para NumOfProducts: {data.shape[0]}") # Tamaño después de eliminar outliers

# Verificamos luego de borrar outliers
plt.figure(figsize=(10, 6))
sns.boxplot(x=data['NumOfProducts'])
plt.title('Boxplot de NumOfProducts sin Outliers')
plt.show()

```

Figura 21 - Eliminación de outliers en NumOfProducts. Elaboración propia.

La figura 21 representa el código que tiene como objetivo eliminar los valores atípicos (outliers) de la variable **NumOfProducts** utilizando el rango intercuartílico (IQR) con un factor ajustable de 1.5. El proceso comienza calculando el primer cuartil (Q1) y el tercer cuartil (Q3) de la variable **NumOfProducts** para luego determinar el IQR (la diferencia entre Q3 y Q1). Usando el factor de 1.5, se calculan los límites inferior y superior, fuera de los cuales se consideran outliers. Posteriormente, se filtran los datos para eliminar aquellos que están fuera de estos límites, y se muestra el tamaño del conjunto de datos antes y después de la eliminación de outliers. Finalmente, se visualiza un boxplot de la variable **NumOfProducts** sin los valores atípicos, permitiendo verificar gráficamente el impacto de la eliminación de estos datos extremos en la distribución de la variable.

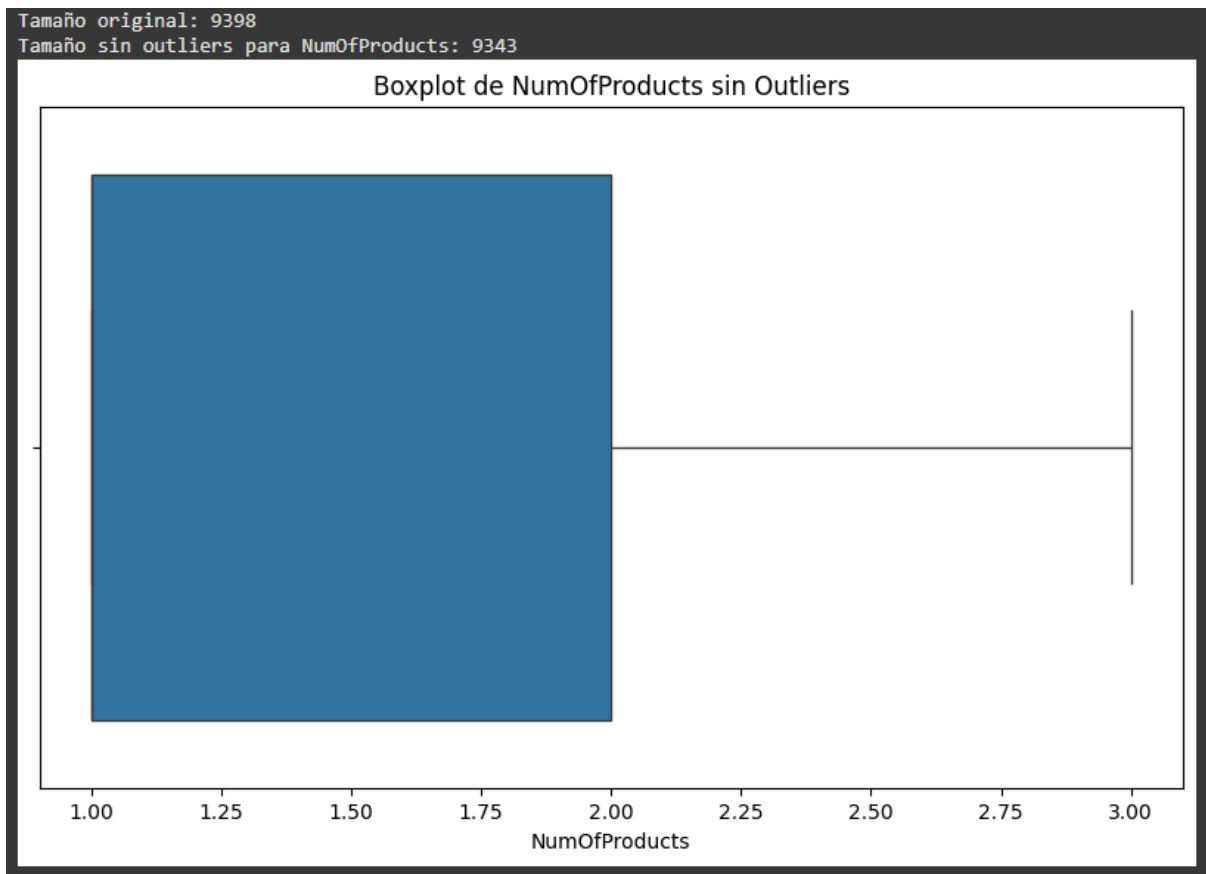


Figura 22 - NumOfProducts sin outliers. Elaboración propia.

En la figura 22, se presenta la gráfica tipo boxplot de la variable **NumOfProducts**, donde se observa una disminución en el número de datos de 9,398 a 9,343. Esto indica que se han eliminado 55 valores atípicos (outliers) de la columna **NumOfProducts**, los cuales se encontraban fuera del rango determinado por el rango intercuartílico (IQR) ajustado con un factor de 1.5.

```

# Vamos a verificar a detalle los datos para asegurar de una calidad excelente

# Verificar tipos de datos
print("Tipos de Datos:\n")
for column, dtype in data.dtypes.items():
    print(f"{column}: {dtype}")
print("\n") # Salto de línea

# Identificar valores faltantes
missing_values = data.isnull().sum()
if missing_values.sum() == 0:
    print("Valores Faltantes: No hay valores faltantes.\n")
else:
    print("Valores Faltantes:\n")
    display(missing_values[missing_values > 0])

# Buscar duplicados
duplicates = data.duplicated().sum()
if duplicates == 0:
    print("Valores Duplicados: No hay duplicados.\n")
else:
    print(f"Valores Duplicados: {duplicates} duplicados encontrados.\n")

# Verificar rango de valores para Age
age_outliers = data[(data['Age'] < 0) | (data['Age'] > 100)]
if age_outliers.empty:
    print("Valores de Edad Fuera de Rango: No hay valores fuera de rango.\n")
else:
    print("Valores de Edad Fuera de Rango:\n")
    display(age_outliers)

# Estadísticas descriptivas
statistics = data.describe()
print("Estadísticas Descriptivas:\n")
display(statistics)

```

Figura 23 - Verificación de limpieza. Elaboración propia.

El proceso de verificación de la calidad de los datos se llevó a cabo de manera exhaustiva, abordando diversos aspectos fundamentales para garantizar la integridad y fiabilidad de los datos antes de continuar con el análisis y la modelización. Primero, se revisaron los tipos de datos de todas las columnas del conjunto para asegurarse de que cada una estuviera correctamente definida y

adecuada para su análisis. Posteriormente, se realizó una revisión en busca de valores faltantes, y se confirmó que no existían valores nulos en ninguna de las variables, lo que asegura que no sea necesario ningún proceso de imputación o tratamiento adicional para estos casos.

Además, se identificaron posibles duplicados en los registros, pero el análisis reveló que no había registros duplicados en el conjunto de datos, lo cual es crucial para evitar sesgos o distorsiones en los resultados del análisis. En cuanto a la variable **Age**, se verificó que no existieran valores fuera de su rango lógico (0-100 años), y efectivamente, no se encontraron registros inconsistentes en esa columna.

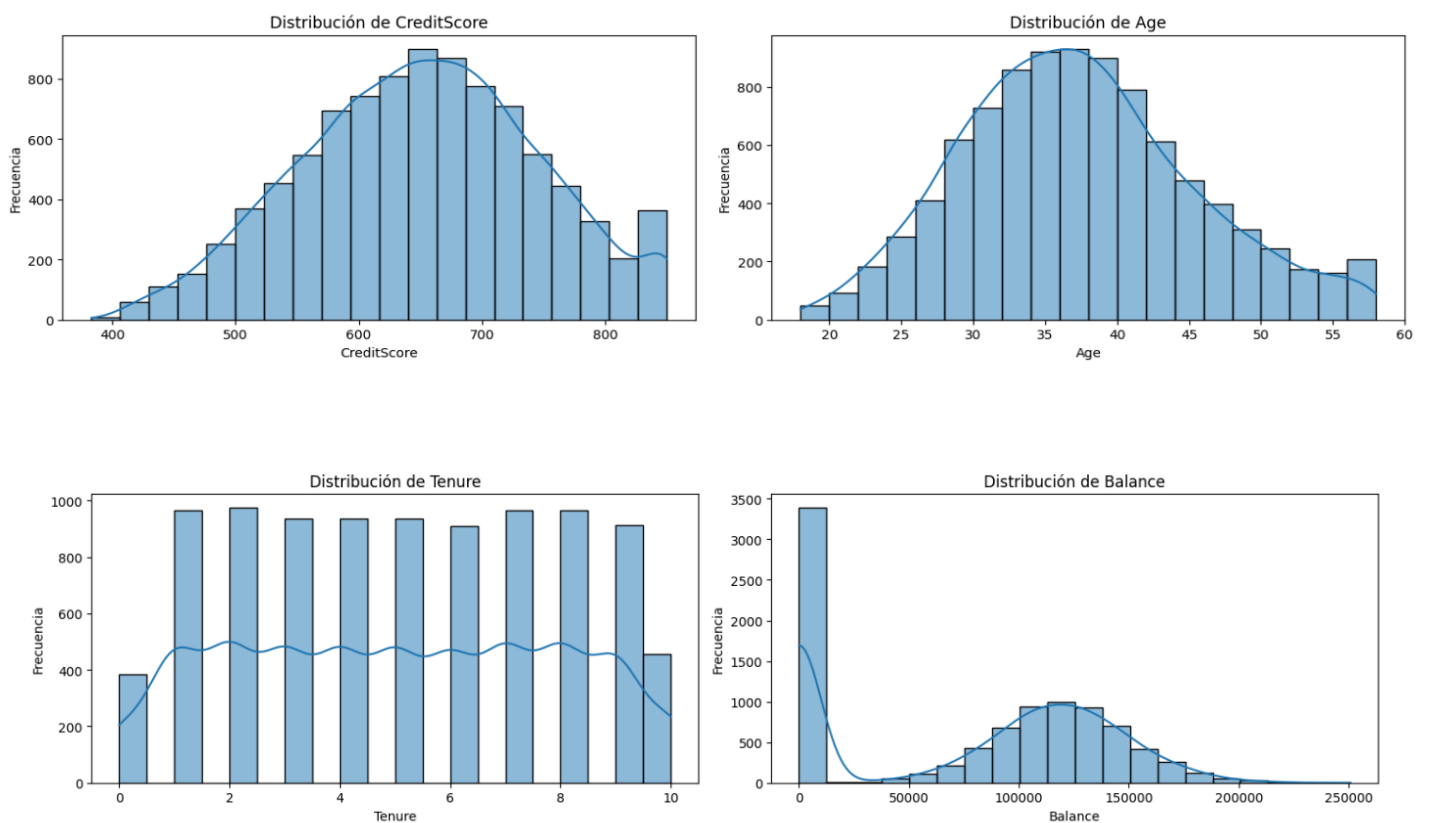
Finalmente, se presentaron las estadísticas descriptivas, las cuales proporcionaron un panorama claro sobre la distribución de los datos, como las medias, las desviaciones estándar, los valores mínimo y máximo, y los percentiles, lo que permitió validar que las características de los datos son coherentes y dentro de los rangos esperados. Según lo anterior, visualizamos:

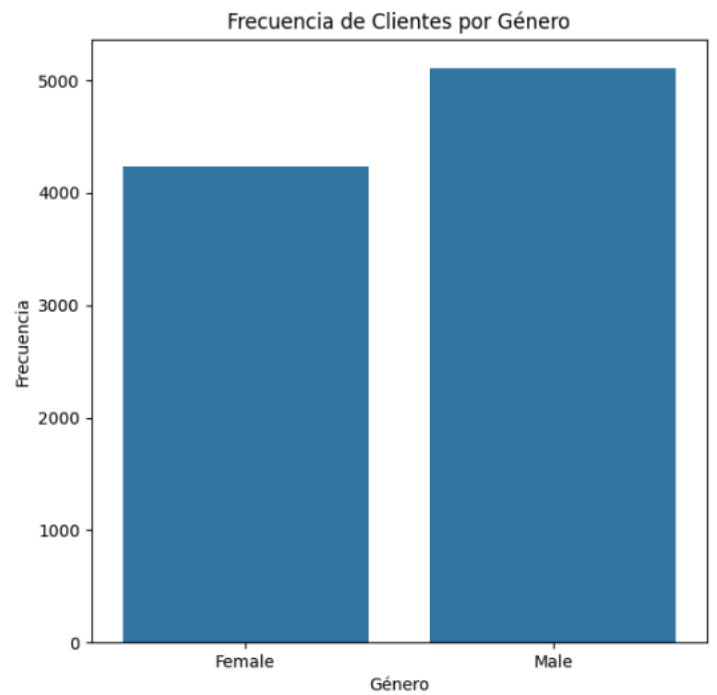
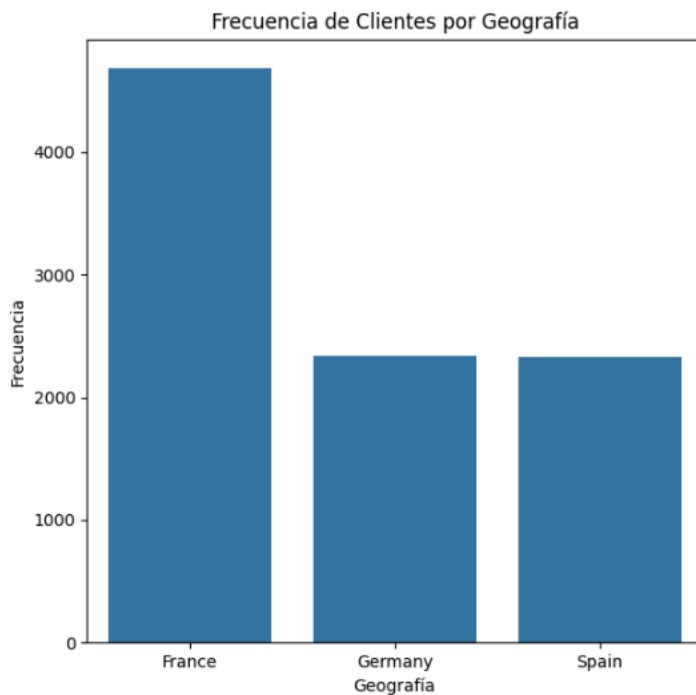
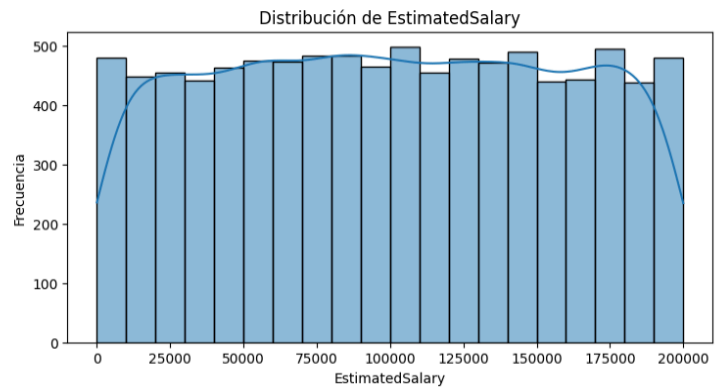
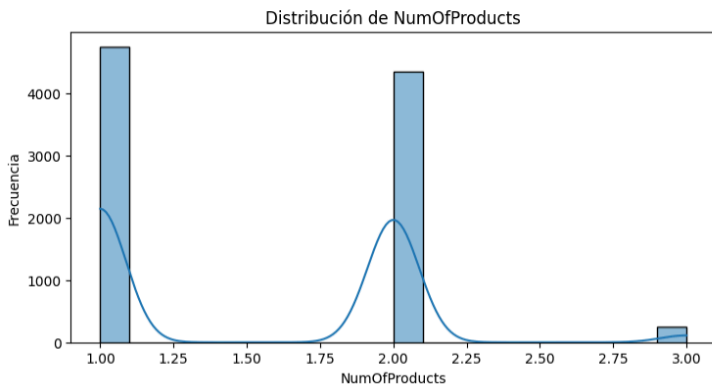
Valores Faltantes: No hay valores faltantes.									
Valores Duplicados: No hay duplicados.									
Valores de Edad Fuera de Rango: No hay valores fuera de rango.									
Estadísticas Descriptivas:									
	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	9343.000000	9343.000000	9343.000000	9343.000000	9343.000000	9343.000000	9343.000000	9343.000000	9343.000000
mean	650.799957	37.189554	5.016911	76353.798203	1.518142	0.705127	0.499732	100144.032720	0.191694
std	95.952078	8.141197	2.888518	62444.180280	0.549681	0.456010	0.500027	57483.691575	0.393655
min	383.000000	18.000000	0.000000	0.000000	1.000000	0.000000	0.000000	11.580000	0.000000
25%	584.000000	31.000000	3.000000	0.000000	1.000000	0.000000	0.000000	51257.330000	0.000000
50%	652.000000	37.000000	5.000000	97041.160000	1.000000	1.000000	0.000000	100236.020000	0.000000
75%	717.000000	42.000000	7.000000	127597.785000	2.000000	1.000000	1.000000	149392.065000	0.000000
max	850.000000	58.000000	10.000000	250898.090000	3.000000	1.000000	1.000000	199992.480000	1.000000

Figura 24 - Resultados de limpieza. Elaboración propia.

En la próxima fase del proyecto, se emplearán las visualizaciones generadas para facilitar la identificación del modelo más adecuado según los resultados obtenidos durante el análisis. A través de diversas representaciones gráficas, se podrá evaluar de manera efectiva el desempeño de los distintos modelos y tomar decisiones fundamentadas sobre cuál ofrece el mejor rendimiento en términos de precisión, recall, F1-score, entre otros indicadores clave.

A continuación, se presentan las visualizaciones correspondientes, las cuales serán clave para seleccionar el modelo más eficaz para abordar el problema de deserción de clientes en la institución bancaria.





Basado en las gráficas anteriores, se generan hipótesis de hacia dónde está tendiendo cada variable.

1. CreditScore:

- La distribución es aproximadamente normal con un sesgo ligero
- El rango principal está entre 400-850 puntos
- La mayoría de los clientes tienen puntajes entre 600-700
- Hay un pequeño grupo con puntajes muy altos (800+)

Hipótesis: Clientes con puntajes muy bajos podrían tener mayor probabilidad de abandono

2. Age:

- Distribución normal con un sesgo positivo
- La mayoría de los clientes están entre 30-45 años
- Hay una cola larga hacia la derecha (clientes mayores)
- Se observa una menor frecuencia en clientes jóvenes (<25 años)

Hipótesis: Los clientes en los extremos de edad podrían tener patrones diferentes de abandono

3. Tenure:

- Distribución bastante uniforme entre 1-10 años
- Menor frecuencia en clientes nuevos (0-1 año)
- La distribución plana sugiere estabilidad en la retención

Hipótesis: Clientes muy nuevos podrían tener mayor riesgo de abandono

4. Balance:

- Distribución bimodal muy marcada
- Un gran pico cerca de 0 (clientes con poco saldo)
- Un segundo pico alrededor de 125,000-150,000
- Cola larga hacia la derecha

Hipótesis: Los extremos en el balance podrían ser predictores importantes de abandono

5. NumOfProducts:

- Distribución discreta con tres picos principales:
 - Un pico alto en 1 producto (aproximadamente 4500 clientes)

- Un pico similar en 2 productos (aproximadamente 4200 clientes)
- Un pico muy pequeño en 3 productos (menos de 500 clientes)

Insights importantes:

- La mayoría de los clientes tienen 1 o 2 productos
- Es muy poco común tener 3 productos
- No hay clientes con más de 3 productos

Hipótesis:

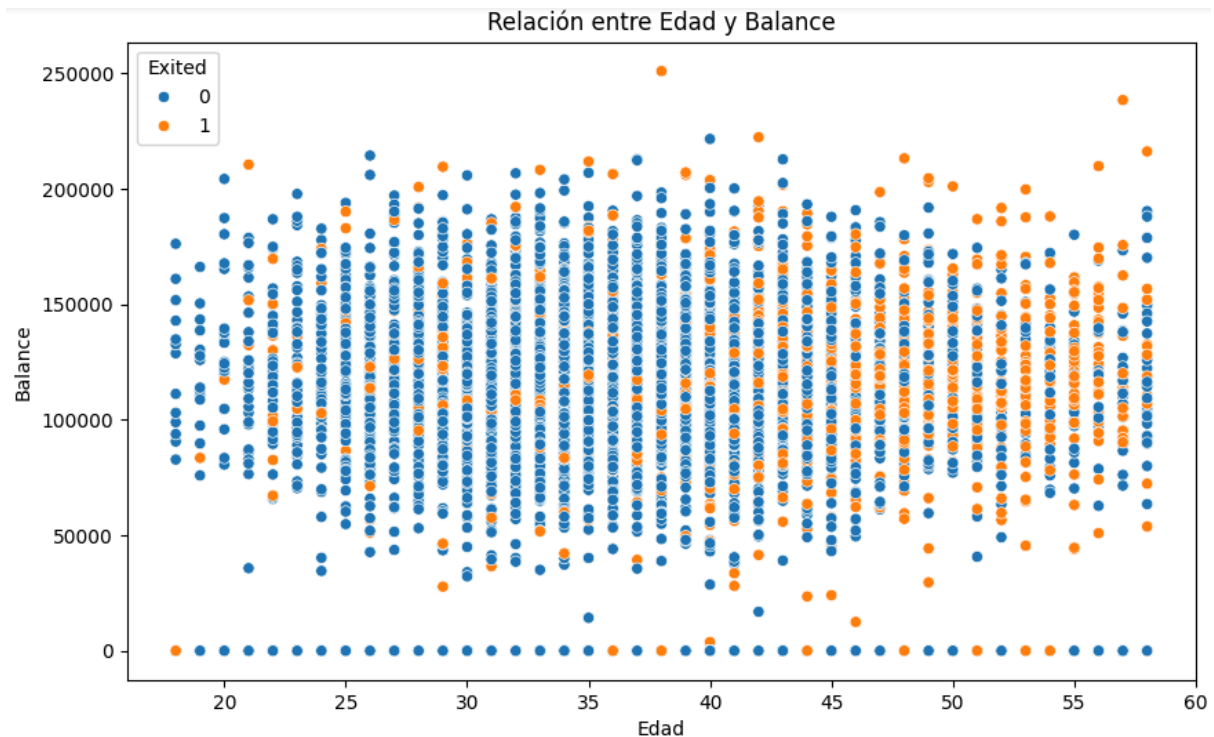
- Los clientes con 3 productos podrían tener un comportamiento diferente en términos de abandono
- Un solo producto podría indicar menor compromiso con el banco
- 2 productos podría ser el punto óptimo de vinculación

2. EstimatedSalary:

- Distribución bastante uniforme entre 0 y 200,000
- Ligeras fluctuaciones pero sin tendencias marcadas
- No hay sesgos significativos
- La frecuencia se mantiene relativamente constante alrededor de 450-500 clientes por rango

Insights importantes:

- No hay concentración en ningún rango salarial específico
- La distribución uniforme sugiere una base de clientes diversificada
- Los extremos (muy bajo o muy alto) muestran una ligera disminución



Basado en la gráfica relacional:

1. Patrones Generales:

- Se observa una dispersión amplia de balances en todas las edades
- Hay dos grupos claros de balance:
 - Un grupo concentrado cerca de 0 (clientes con bajo balance)
 - Un grupo distribuido entre 50,000 y 200,000

2. Relación con Abandono (Exited):

- Los puntos naranjas (Exited=1) parecen tener mayor presencia en:
 - Edades más avanzadas (>45 años)
 - Balances más altos
- Los puntos azules (Exited=0) tienen una distribución más uniforme

3. Observaciones específicas:

- Mayor densidad de abandonos en:
 - Clientes mayores con balances altos

- Algunos clusters de edad media con balances medios-altos
- Menor tasa de abandono en:
 - Clientes jóvenes (20-35 años)
 - Clientes con balances muy bajos

En el proyecto de deserción de clientes en una institución bancaria, la fase de limpieza y preparación de datos fue fundamental para asegurar que el conjunto de datos estuviera en condiciones óptimas para el modelado predictivo. Este proceso incluyó la detección y corrección de valores nulos, la eliminación de columnas irrelevantes, el manejo de valores atípicos y la transformación de variables categóricas.

Eliminación de valores nulos y duplicados

La eliminación de valores nulos y duplicados fue el primer paso en el proceso de limpieza, un aspecto crítico para evitar sesgos en los resultados y mejorar la fiabilidad de los modelos de aprendizaje automático (Kotsiantis et al., 2006). Estos valores pueden introducir ruido y reducir la precisión del modelo, por lo que su detección y eliminación fueron esenciales para la integridad del conjunto de datos (Rahm & Do, 2000).

Transformación y codificación de variables categóricas

Las variables categóricas, como 'Geography' y 'Gender', se convirtieron a un formato adecuado mediante codificación “one-hot” (Bishop, 2006). Esta técnica es esencial cuando se emplean algoritmos de machine learning que requieren variables numéricas, ya que mejora la capacidad del modelo para interpretar correctamente la información categórica (Pedregosa et al., 2011).

Detección y tratamiento de valores atípicos

Los valores atípicos o “outliers” fueron identificados y tratados utilizando el rango intercuartílico (IQR) para variables como 'CreditScore', 'Age' y 'NumOfProducts'. Esta técnica es ampliamente recomendada para reducir el impacto de datos extremos en los modelos, garantizando así una mejor generalización (Rousseeuw &

Croux, 1993). Los valores fuera de los límites calculados fueron eliminados o ajustados para mejorar la representatividad del conjunto de datos (Tukey, 1977).

Visualización de datos y análisis exploratorio (EDA)

La visualización de datos permitió identificar patrones y distribuciones en el conjunto de datos, facilitando la comprensión de las relaciones entre variables y de la deserción de clientes. Las herramientas de EDA, como histogramas y boxplots, son claves para obtener información valiosa antes del modelado, además de identificar posibles problemas en los datos que podrían afectar el desempeño del modelo (Cleveland, 1985).

Una vez completamos todos los pasos que consideré necesarios para poder proseguir con el entrenamiento del modelo, realicé el paso final que consistió en exportar el nuevo dataset:

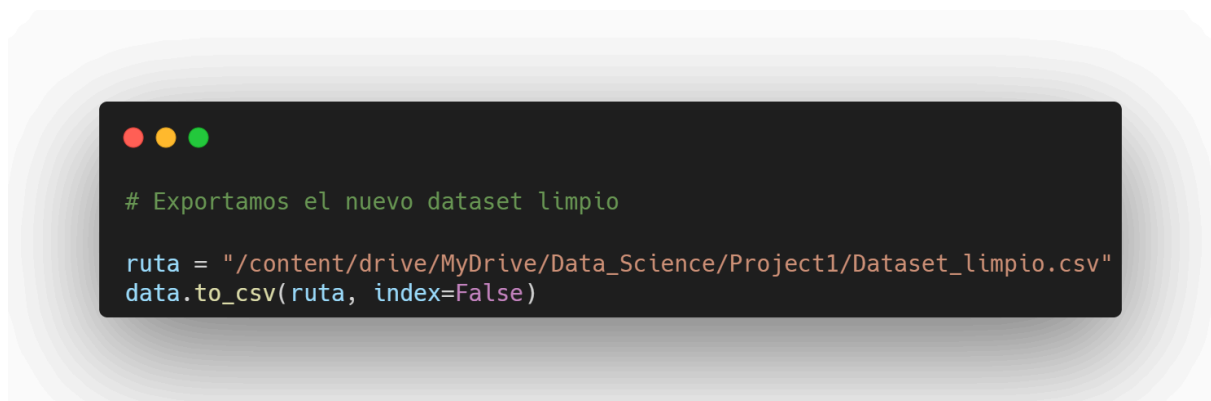


Figura 25 - Exportar dataset. Elaboración propia.

Basado en todo lo anterior, el presente trabajo se enmarca dentro del campo de la minería de datos predictiva, ya que se centra en el análisis y transformación de datos históricos para identificar patrones y factores que puedan anticipar el comportamiento futuro de los clientes. Mediante técnicas de limpieza,

procesamiento de datos y análisis visual, el objetivo principal es desarrollar modelos predictivos que estimen la probabilidad de abandono o retención de clientes. Este enfoque predictivo permite aprovechar grandes volúmenes de datos para obtener información relevante y orientada a la toma de decisiones estratégicas, lo cual resulta crucial para la optimización de recursos y la mejora de estrategias de retención en la institución.

Conclusiones:

1. **Optimización de la calidad del conjunto de datos:** El proceso exhaustivo de limpieza y preparación de datos ha permitido identificar y corregir valores nulos, datos atípicos y variables inconsistentes, incrementando la confiabilidad del conjunto de datos. Esta mejora en la calidad de los datos es fundamental, ya que facilita un entrenamiento más efectivo y preciso del modelo de machine learning, reduciendo el riesgo de sesgos o errores en las predicciones.
2. **Detección de patrones significativos en el comportamiento de los clientes:** A través del análisis exploratorio de datos (EDA), se identificaron patrones y correlaciones entre variables que impactan en la deserción de clientes, como la relación entre la edad y el balance de cuenta. Estos hallazgos proporcionan una base sólida para definir las variables de entrada más relevantes en el modelo de predicción de deserción, mejorando su capacidad para capturar las características clave que influyen en el comportamiento del cliente.
3. **Preparación estructurada para la etapa de modelado:** La transformación de variables categóricas de los datos han sido pasos críticos en la preparación del conjunto de datos para su uso en un modelo predictivo. Este trabajo asegura que el conjunto de datos esté en condiciones óptimas, facilitando la implementación de algoritmos de machine learning que requieran datos consistentes y normalizados para lograr una mayor precisión en la predicción de la deserción.

Recomendaciones para el futuro modelo de machine learning:

1. **Evaluar diferentes algoritmos de clasificación:** Dado el objetivo de predecir la deserción de clientes, se recomienda evaluar modelos de clasificación como árboles de decisión, bosques aleatorios (random forests) y algoritmos de boosting. Estos modelos son efectivos para identificar relaciones complejas y no lineales en los datos, lo que puede mejorar la precisión en la predicción de la deserción.
2. **Implementar un enfoque de validación cruzada y ajustar hiperparámetros:** Para asegurar la robustez del modelo, es fundamental emplear técnicas de validación cruzada, lo cual permitirá evaluar la capacidad del modelo para generalizar a nuevos datos. Asimismo, ajustar los hiperparámetros de los modelos seleccionados puede optimizar el rendimiento predictivo, maximizando su efectividad en la identificación de clientes propensos a abandonar la institución.

Bibliografía

IBM (2021) IBM - Conceptos básicos de ayuda de CRISP-DM

<https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview>

IBM (2024) IBM - ¿Qué es el análisis exploratorio de datos?

<https://www.ibm.com/mx-es/topics/exploratory-data-analysis>

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Cleveland, W. S. (1985). *The Elements of Graphing Data*. Wadsworth Advanced Books and Software.

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). *Data preprocessing for supervised learning*. International Journal of Computer Science, 1(2), 111-117.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). *Scikit-learn: Machine learning in Python*. Journal of machine learning research, 12, 2825-2830.

Rahm, E., & Do, H. H. (2000). *Data cleaning: Problems and current approaches*. IEEE Data Eng. Bull., 23(4), 3-13.

Rousseeuw, P. J., & Croux, C. (1993). *Alternatives to the median absolute deviation*. Journal of the American Statistical Association, 88(424), 1273-1283.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.