Universidad Internacional San Isidro Labrador

Curso: Data Science

Tema:

Modelo predictivo acerca de la deserción de clientes

(*Bank Customer Churn Prediction*)

PROYECTO II

Profesor: Samuel Saldaña Valenzuela

Estudiante:

Melanie Joana Moreira Sánchez

402450025

25 de Noviembre, 2024

Contenido

Introducción:

Como continuidad al "Proyecto I", en esta ocasión se desarrollará un modelo predictivo aplicando el uso de la tecnología moderna con herramientas digitales, tal como el uso de la Inteligencia Artificial, para lograr nuestro objetivo se tomará el conjunto de datos limpio del primer proyecto, y se aplicarán metodologías y técnicas para reconocer los principales patrones y características específicas que influyen en cuanto a la deserción de los clientes.

Tal y como se ha mencionado, este es un desafío significativo que impacta directamente en la rentabilidad y la sostenibilidad de las instituciones financieras. En un entorno cada vez más competitivo, la capacidad de predecir y prevenir la pérdida de clientes es una prioridad estratégica, este proyecto mediante el uso de métodos tales como el análisis de componentes y el uso de clustering, teniendo en cuenta que con K-Means (Clustering), se podrán agrupar a los clientes en segmentos basados en su riesgo de deserción, lo que permitirá enfoques de retención específicos, esta etapa incluye la selección y ajuste de los técnicas para optimizar su utilidad y asegurar la precisión más alta en las predicciones señaladas.

Finalmente, se espera como resultado un sistema inteligente que pueda predecir el "churn" y los distintos factores importantes que le competen, de manera que le permita al banco la toma de decisiones sobre como retener a los usuarios con probalidades de abandonar los servicios o bien mantener la fidelidad de los clientes actuales sin que estos lleguen a pensar en desertar, debido a que el hecho de actuar antes de, le permite a la institución ahorrar dinero ya que no tendrán que atraer nuevos clientes y se ahorrarán dichos costos, además los clientes actuales estarán satisfechos con lo que el banco les ofrece, dándole credibilidad y posición en el mercado.

Objetivo General:

El objetivo general de este proyecto, es crear un sistema inteligente que utilice técnicas digitales -AI- para predecir cuando un cliente podría dejar la institución financiera. Este sistema ayudará al banco a comprender mejor lo que quieren y necesitan sus usuarios, así como los beneficios que desean obtener sus clientes y las diferencias en cuanto a otros bancos, mismas que les podrían brindar mayor competitividad ante otras instituciones.

Objetivos Específicos:

1. Identificar la principal razón de deserción de los usuarios, mediante el uso de herramientas de ML para analizar los datos de los clientes y descubrir cuáles son los factores que hacen que los mismos decidan abandonar la institución financiera.
2. Ejecutar un modelo de soporte que le permita al banco tomar decisiones inteligentes y por consecuencia, según los resultados actuar.
3. Diseñar estrategias personalizadas de retención basadas en los segmentos de clientes obtenidos, permitiendo al banco optimizar recursos y reducir costos asociados en cuanto a "captar" nuevos clientes.

Marco Teórico:

La deserción de clientes en las entidades financieras es una de las principales complicaciones en dicho mercado actualmente, esto debido a la pérdida de ingresos que este tema representa y por consecuencia, la necesidad de realizar un esfuerzo mayor para retener a los usuarios. Debido al incremento de la competencia y a la gran cantidad de opciones existentes, es vital que las instituciones bancarias entiendan las razones existentes tras la deserción de sus clientes.

Interiorizando el tema focal, se menciona que en general, las instituciones, como ya sabemos, enfrentan un reto importante, aprovechar los datos, años atrás estos eran solamente información que se acumulaba en las distintas bases y registros, sin embargo, hoy en día, ya no es así, sino que se convirtieron en un recurso valioso que, cuando se analiza de manera correcta permite la obtención de recursos esenciales para el desarrollo óptimo del negocio.

Importante mencionar que, el focus no está en la cantidad de datos que se obtienen sino en cómo se procesan estos, de modo que, en cierto punto, pasan a ser utiles para actuar sobre ellos.

El modelo busca proporcionar información clave sobre los diferentes factores que influyen, con estrategias fundamentadas, de manera que se logre disminuir significativamente la tasa de deserción, mejorando también, la satisfacción de los clientes y fortaleciendo la posición del banco en el mercado.

Para ir ejecutando lo anteriormente mencionado dicho proyecto hace énfasis en los datos, ya que acá se pretende transformar los mismos en herramientas de predicción, es decir la etapa de modelado, justo aquí después de organizar y analizar los datos recopilados se crea el

sistema inteligente, el cual permite hacer predicciones o dar información relevante la cual es la que ayuda a la toma de decisiones importantes para la empresa.

Además, para entender esta etapa es necesario conocer el ciclo de vida de los datos, este ciclo se podría definir en palabras sencillas como un "conjunto de pasos que explican que se hace con los datos desde que se recopilan hasta que se convierten en algo útil" (ESIC, 2024). Se menciona además que, todas estas etapas están conectadas entre sí y que por supuesto, trabajan juntas para la obtención de buenos resultados.

Para abordar lo anterior tenemos y requerimos de lo que se explicará y desarrollará a continuación:

Desarrollo Teórico:

- Ciclo de vida de los datos:

El ciclo de vida de los datos es una secuencia de etapas por las que pasan los datos a lo largo de toda su vida útil. Los datos se separan en fases en función de diferentes criterios, y pasan por estas etapas a medida que completan diferentes tareas o cumplen ciertos requisitos.

Adicionalmente, se menciona que el ciclo de vida de los datos abarca todo el periodo de tiempo que los datos existen en una organización, desde la generación de los datos hasta su eliminación o reutilización a través de diferentes tipos de repositorios de investigación.

Se considera que es un ciclo porque los conocimientos obtenidos de un proyecto de datos suelen servir de base para el siguiente. De este modo, la última etapa del proceso retroalimenta la primera.

-Datos obtenidos de: (Narvaez, 2018)

En este ciclo, tenemos las siguientes sub-etapas:

- Recopilación
- Almacenamiento
- Procesamiento
- Análisis
- Modelado
- Implementación
- Monitoreo
- Mantenimiento

Recopilación de los datos:

De acuerdo con (Morales & Flores, 2023), y basándonos en el informe de Inteligencia Artificial Aplicada en el Tecnológico de Monterrey, entendemos que actualmente todo archivo genera datos, independientemente del formato, ya sea un documento escrito, un video, podcast y hasta un comentario en redes sociales, etc.

Es entonces que podemos señalar que la recolección de datos es la primera etapa del ciclo y se entiende como “el proceso de búsqueda, recolección y medición de datos de diferentes

fuentes para obtener información sobre los procesos, servicios y productos de tu empresa o negocio y poder evaluar dichos resultados y así tú puedas tomar mejores decisiones."

Y que sirve para mejorar los procesos de mejora continua, pero se debe entender que también depende en gran medida del problema que esté atacando u objetivo planteado por el cual se está realizando dicha recolección.

Además, ellos nos muestran algunos de los usos para dicha recolección, dentro de las cuales podemos tomar como propias según el tema elegido las siguientes:

- Almacenar los datos de acuerdo a las características de un público determinado para apoyar los esfuerzos de tu área de marketing.
- Comprender mejor los comportamientos de tus clientes, usuarios y leads.

Es entonces que, en palabras sencillas, los datos recopilados para este apartado son esenciales ya que de ellos se obtiene la calidad del modelado.

Almacenamiento de los datos:

Básicamente, se puede decir que el almacenamiento de datos se basa en cómo están organizados los datos y en qué base se guardan.

Como segunda fase, tenemos que los datos recopilados se guardan en sistemas especiales, de modo que sean más fáciles para poder trabajar con ellos. Para hacer esto, se pueden elegir herramientas como bases de datos, mismas que son capaces de manejar muchísima información, asegurándose a su paso que los datos se mantengan protegidos y como se mencionó al inicio de la redacción, tenerlos "a mano" para trabajar y accesar fácilmente a ellos cuando se necesite.

Procesamiento de los datos:

Muchas compañías o personas naturales reúnen conjuntos de datos, de diferente índole, que los pueden ayudar a desarrollar sus proyectos y a obtener conocimiento acerca de lo útiles que son estos datos y cómo están afectando el desarrollo. No obstante, debe existir una forma en la que se puedan recolectar, analizar, describir y detallar de estos datos. Esto es el procesamiento de datos.

En palabras más simples, tenemos que es la práctica en la que recolectar y transformar un conjunto de datos (que puede ser un grupo pequeño o mucho más extenso de data). El objetivo del procesamiento de datos es sacarle el mayor provecho a lo extraído de los datos, para aportarle al desarrollo de un programa y al cumplimiento de los objetivos.

Los datos originales, antes de ser procesados, no pueden decirle mucho al usuario, ni tampoco son de libre acceso, ya que son muy difícil de decodificar. Por lo tanto, necesitan un agente que los haga accesibles y claros.

El procesamiento de datos se encarga de recolectar, filtrar, sortear, analizar y hasta almacenar todos los conjuntos de datos, para que puedan ser utilizados por los departamentos de marketing, ventas o los administrativos de una compañía. Después del procesamiento de datos y para cumplir con el mismo objetivo, los analistas de datos se encargan de registrar los hallazgos en forma de gráficos, cuadros y otros documentos que puedan servir para cumplir lo presupuestado.

-Datos obtenidos de: (Navarro, 2022)

Análisis de los datos:

El análisis de datos es importante porque ayuda a entender los datos antes de usar modelos más avanzados que predicen lo que podría suceder en un futuro. Se puede definir también, como el proceso de investigar y hacer preguntas para conocer mejor los datos antes de hacer predicciones con ellos

Es entonces que se sobre entiende que en esta fase se exploran los datos para encontrar patrones que nos puedan brindar información útil…

Por ejemplo, "¿Cuánto tardan los clientes en abandonar el banco?", según lo investigado podemos comprender que, para obtener resultados en dicha fase, se pueden utilizar una o varias técnicas según lo que se requiera, entre ellas:

- Análisis descriptivo
- Inferencial
- Correlacional

Es justo por ello, que el análisis de los datos es fundamental ya que ayuda a tomar decisiones más acertadas basadas en lo que los datos realmente nos están diciendo.

Modelado:

Esta fase se enfoca en usar los datos que ya han sido procesados y analizados para crear modelos predictivos, acá se "aplican" algoritmos de Machine Learning (ML) supervisados y no, con el fin de generar predicciones sobre lo que podría acontecer, en dicho proyecto "la posibilidad de que un cliente deserte a la financiera"

Se considera esta, una de las etapas más importantes ya que convierte los datos en información util que puede afectar de manera directa las decisiones que tome la compañía.

Implementación:

En cuanto a esta etapa, se menciona que es la etapa en la que se implementan modelos con una canalización de datos en un entorno de producción o similar, también se considera la fase en la que los modelos predictivos y las soluciones desarrolladas a lo largo del ciclo se ponen a prueba en un entorno real.

Es decir, después de la creación y de la prueba del modelo, la implementación consiste en aplicar de manera práctica para que nos ayude a la toma de decisiones, ya que, en este momento, se integran los modelos dentro de los sistemas, permitiendo así que las distintas compañías utilicen dichas predicciones para aplicarlas en las distintas estrategias.

En resumen, tenemos que implementación es igual a "poner en práctica" lo que se ha logrado desarrollar para que realmente e pueda utilizar en cuanto a la resolución de problemas o bien, a la mejora continua de los procesos.

En resumen, la implementación es el paso crucial donde los modelos desarrollados se ponen en práctica dentro de los procesos de la empresa. Esta fase asegura que las soluciones predictivas sean accesibles y útiles para la toma de decisiones reales, permitiendo que la empresa actúe sobre la información de manera efectiva.

Además, es el momento donde se realizan ajustes para optimizar el rendimiento del modelo y garantizar que aporte valor de forma continua.

Monitoreo y Mantenimiento:

El monitoreo y mantenimiento de datos son procesos que permiten garantizar que los datos estén accesibles y utilizables, y que los sistemas funcionen de manera óptima, para ello deben ser supervisados de modo que se garantice su eficacia.

- ***Monitoreo de datos:***

Es un proceso continuo de supervisión y análisis del rendimiento y estado de un sistema de datos. Este monitoreo permite identificar y abordar problemas como lentitud en las consultas, cuellos de botella de recursos y fallos potenciales.

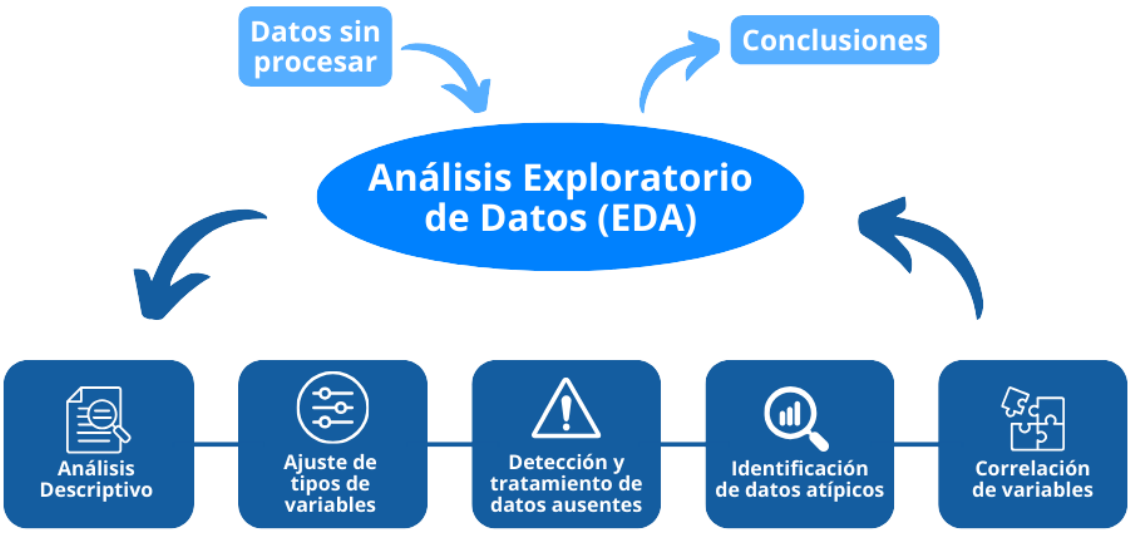
- ***Mantenimiento de datos:***

Es el proceso de organizar y conservar los datos de acuerdo con las necesidades. El mantenimiento de datos implica tareas como la limpieza, las copias de seguridad y la optimización de los índices.

-Datos obtenidos de: (Universidad Oberta)

- Análisis Exploratorio de Datos:

-Exploratory Data Analysis



El EDA permite una comprensión profunda del comportamiento de los datos y su relación con la deserción de clientes. Técnicas como gráficos de dispersión, histogramas y diagramas de caja ayudan a visualizar las características principales del conjunto de datos.

Estas herramientas facilitan:

- Detectar patrones inesperados.
- Identificar relaciones significativas entre las variables independientes y la deserción como variable objetivo.
- Evaluar posibles sesgos o desequilibrios en los datos.

Su importancia radica en que, como se puede observar a lo largo de la ejecución de dicho proyecto el -Análisis Exploratorio de Datos- (EDA) no solo revela patrones ocultos en el conjunto de datos, sino que también proporciona una base sólida para seleccionar las técnicas estadísticas y los modelos más adecuados.

Es decir, en palabras sencillas, no solo ayuda a entender los datos, sino que también, guía el enfoque hacia las técnicas de modelado convenientes y asegura que el análisis se realice sobre datos confiables y bien preparados.

Dentro de este ciclo, esta etapa resulta imprescindible ya que, tal y como su definición lo menciona, su objetivo principal es el de descubrir los patrones y las inconsistencias presentes, además de comprender los datos, de modo que se logre evaluar la estructura, tamaño, características y generalidades del Dataset con esto se logran detectar patrones iniciales, identificar las tendencias y las relaciones entre las variables, además de los datos incongruentes que están fuera de lo esperado y que por supuesto podrían afectar el análisis general.

"Predecir el abandono en una institución bancaria", un análisis correctamente ejecutado podría:

- Analizar las variables y todo lo que en relación compete
- Estudiar la relación entre variables e identificar si existen valores incongruentes.
- Visualizar la distribución de ingresos para identificar si los datos están bien diferenciados.

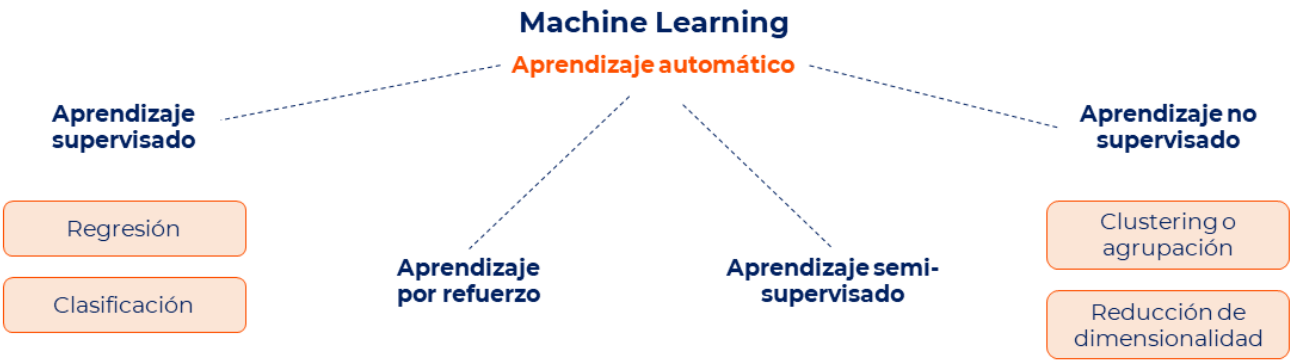
- Modelado Predictivo:

Este es un proceso en el que se utilizan datos históricos y técnicas de Machine Learning (ML) para la creación de modelos que puedan predecir eventos o comportamientos futuros, el objetivo principal es usar la información anterior para anticipar lo que va a suceder (tal y como se ha venido estudiando y mencionando en el documento).

Existen diferentes tipos de modelos predictivos, dependiendo de la naturaleza del problema. Pueden ser:

- Modelos supervisados: Se entrenan con datos etiquetados (es decir, sabemos el resultado de los datos de entrada, como si un cliente se fue o no). Ejemplos incluyen regresión y clasificación.

- Modelos no supervisados: No tienen etiquetas y se utilizan para encontrar patrones o grupos en los datos sin saber el resultado de antemano. Un ejemplo común es el clustering, que agrupa clientes similares.



Claves:

- Los modelos predictivos utilizan datos históricos para hacer predicciones sobre eventos futuros. Cuanto más relevantes y completos sean los datos, más precisas pueden ser las predicciones.
- Los Algoritmos de Machine Learning, se utilizan para predecir un valor continuo.
- La clasificación se utiliza para predecir categorías, como si un cliente abandonará el banco o no.

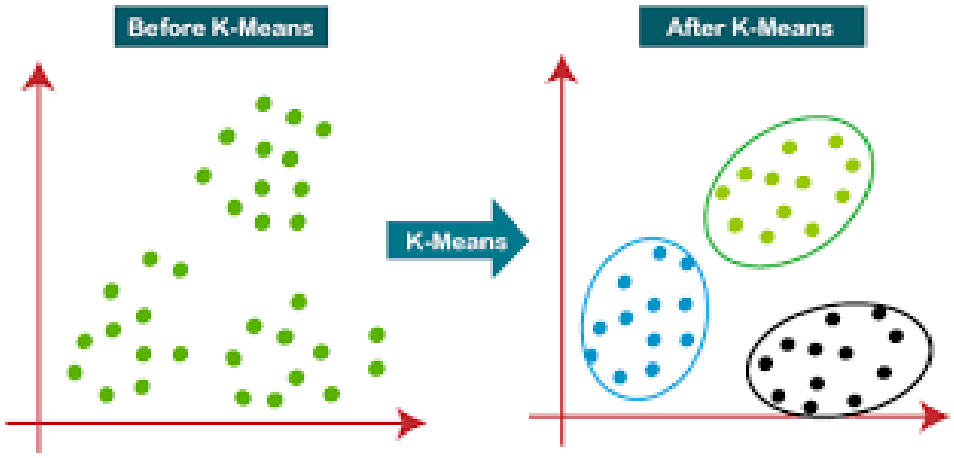
- Clustering: Agrupa a los clientes en segmentos similares según sus características.
- Los modelos predictivos se entrenan utilizando un conjunto de datos llamado conjunto de entrenamiento, que enseña al modelo a reconocer patrones. Una vez entrenado, el modelo se prueba con otro conjunto de datos, el conjunto de prueba, para evaluar qué tan bien predice resultados.
- La evaluación del Modelo se utiliza para saber qué tan bueno es un modelo, para ello se utilizan métricas de evaluación. Algunas de las métricas comunes son:
    - Precisión: Qué tan exactas son las predicciones.
    - Recall: Cuántos de los casos importantes (como los clientes que realmente se van) fueron identificados correctamente.
    - AUC-ROC: Mide el rendimiento del modelo en términos de su capacidad para discriminar entre las categorías.
- La validación cruzada se usa para evaluar la precisión del modelo. En lugar de usar un solo conjunto de datos de entrenamiento y prueba, se divide el conjunto de datos en múltiples partes y se entrena y prueba el modelo varias veces en diferentes combinaciones. Esto ayuda a obtener una evaluación más robusta del modelo.

Finalmente, el objetivo de un modelo predictivo es usar las predicciones generadas para tomar decisiones informadas.

Por ejemplo, un banco puede decidir ofrecer promociones o servicios especiales a clientes con alta probabilidad de abandonar la institución.

Dentro de los factores más importantes tenemos el de:

- K-means:



Como se ha mencionado en reiteradas ocasiones a lo largo de este escrito, tenemos que **es** un método de agrupamiento que divide un conjunto de datos en k grupos o clusters. Los datos se agrupan de tal manera que los puntos en el mismo clúster sean más similares entre sí que los puntos en otros clústeres.

Del universo de algoritmos de aprendizaje no supervisado, K-means sigue siendo uno de los algoritmos más conocidos para el aprendizaje no supervisado, aunque alternativas más avanzadas como DBSCAN o algoritmos basados en clustering espectral han ganado popularidad en ciertos escenarios debido a su capacidad para manejar conjuntos de datos más complejos y de mayor dimensión.

Datos obtenidos de: (Pita, s.f.)

Teniendo como punto focal el concepto de “deserción”, la técnica anterior se nos ajusta a la resolución del problema, ya que segmenta a los usuarios según los comportamientos respectivos.

Esta segmentación permite diseñar estrategias personalizadas para cada grupo, enfocándose en retener a los clientes en riesgo mediante ofertas específicas, además de crear lazos y vínculos con los clientes fieles al servicio, e incentivar a los usuarios inactivos a interactuar en mayor nivel con los servicios, mejorando así la fidelidad general.

Desarrollo Práctico:

- **Explicación textual del código:**

Para dar inicio vamos a mencionar nuevamente que el código desarrollado evalúa un modelo que permite predecir cuando un cliente podría desertar o bien dejar un banco y sus servicios. Como primera fase, tenemos que preparar los datos y seguidamente, ajustarlos para su respectivo análisis, luego de, se utilizarán herramientas como PCA esto para simplificara la información obtenida y por consiguiente, haremos uso de K-Means para agrupar los clientes según las similitudes que estos tengan.

Después, se observará la implementación de un modelo llamado Gradient Boosting el cuál se entiende como un tipo de algoritmo de aprendizaje automático supervisado por conjuntos que combina múltiples aprendices débiles para crear un modelo final.

Entrena secuencialmente estos modelos colocando más pesos en las instancias con predicciones erróneas, minimizando gradualmente una función de pérdida. Las predicciones de los aprendices débiles se comparan con los valores reales y la diferencia representa la tasa de error del modelo.

Esta tasa de error se utiliza para calcular el gradiente, que se utiliza para encontrar la dirección del ajuste de parámetros del modelo en la siguiente ronda de entrenamiento.

A diferencia de un modelo de red neuronal, donde se utiliza un solo modelo, el aumento de gradiente combina las predicciones de múltiples modelos para minimizar el error.

-Datos obtenidos de: (Machine Learning Guide Oil & Gas, 2021)

Ahora bien, finalmente se evaluará que tan exacto es este modelo con medidas como la precisión, la matriz de confusión, la curva ROC y la importancia de los datos utilizados…

Todo esto lo podremos observar en la explicación práctica y ejemplificada que se mostrará a continuación:

Explicación práctica:

Primeramente, damos inicio a la creación del proyecto en Google Colab, en primera instancia se importan las librerías, y seguidamente se procede a la lectura del dataset (el cual fue previamente cargado a un archivo Drive nombrado “PROYECTO II”, para así facilitar la ubicación del mismo al momento de cargar los datos), esta visualización nos permite tener un contexto claro y amplio del tipo de información con el cual estamos trabajando.

```python
# Importamos el dataset limpio
import pandas as pd
ruta_archivo = '/content/drive/MyDrive/PROYECTOII/Dataset_Nuevo.csv'
df = pd.read_csv(ruta_archivo)

# Verificamos que se cargó correctamente
print("Dimensiones del dataset:", df.shape)
print("\nPrimeras 5 filas:")
df.head()
```

Una vez corrido el código, nos desplegó la siguiente tabla de información con parte de los datos utilizados para dicha ejecución

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
Dimensiones del dataset: (10000, 12)

Primeras 5 filas:
```

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_Germany	Geography_Spain	Gender_Male
0	-0.326221	0.293517	2	-1.225848	-0.911583	1	1	0.021886	1	False	False	False
1	-0.440036	0.198164	1	0.117350	-0.911583	0	1	0.216534	0	False	True	False
2	-1.536794	0.293517	8	1.333053	2.527057	1	0	0.240687	1	False	False	False
3	0.501521	0.007457	1	-1.225848	0.807737	0	0	-0.108918	0	False	False	False
4	2.063884	0.388871	2	0.785728	-0.911583	1	1	-0.365276	0	False	True	False

El código comienza importando las herramientas necesarias para trabajar con los datos. Algunas de estas son utilizadas para reducir la cantidad de información que se analiza sin perder lo realmente importante (PCA) o para formar grupos de datos que son parecidos entre sí, función de K-Means.

```python
# Importar librerías adicionales para técnicas no supervisadas
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import roc_curve, auc, classification_report, confusion_matrix
import numpy as np
import joblib
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from imblearn.over_sampling import SMOTE
from sklearn.preprocessing import StandardScaler
from imblearn.pipeline import Pipeline
```

También se incluyen herramientas para la creación de representaciones visuales tales como gráficos, los cuales nos ayudan a entender mejor los datos, usando matplotlib y seaborn.

Además, permite la división de los datos, ya sea para entrenarlos y probarlos (train_test_split), y otra para transformar los números para que estén en la misma escala (StandardScaler), y con ello arreglar problemas cuando existe más de un tipo de dato (SMOTE).

Una vez, teniendo en cuenta lo anterior el código generado se encarga básicamente de cargar el Dataset que fue previamente guardado en formato CSV para que se pueda usar dentro del programa.

```
# Cargar dataset
dataset_path = '/content/drive/MyDrive/PROYECTOII/Dataset_Nuevo.csv'  # Ruta de tu dataset
df = pd.read_csv(dataset_path)
```

- **Define la ubicación del archivo**: En la variable dataset_path se guarda la dirección donde está guardado el archivo llamado Dataset_Nuevo.csv. Este archivo está en una carpeta específica dentro de Google Drive.
- **Carga el archivo**: Con la función pd.read_csv(), que es de la librería pandas, se lee el archivo que está en la ruta mencionada y se convierte en una estructura llamada **DataFrame**. (Se podría decir que esto es como una tabla en Excel, con filas y columnas, donde se organizan los datos para analizarlos más fácilmente.)

```
# Separar características (X) y variable objetivo (y)
X = df.drop(columns=['Exited'])
y = df['Exited']

# Preprocesamiento y balanceo de clases
smote = SMOTE(random_state=42)
scaler = StandardScaler()
```

Seguidamente damos inicio a la segmentación de características, la cual observamos tiene dos partes importantes:

Separar características (X) y variable objetivo (y):

- X = df.drop(columns=['Exited']): Aquí, se está separando el DataFrame df en dos partes.

En X se guardan las características o variables de entrada (todas las columnas excepto la columna Exited). Esto significa que se quiere usar todo el conjunto de datos para predecir algo, pero sin la columna Exited.

- y = df['Exited']:

En y se guarda la variable objetivo (en este caso, la columna Exited), que es lo que se quiere predecir. Usualmente, esta columna es binaria, donde 1 indica que el cliente salió (churn) y 0 indica que no lo hizo.

Este paso es clave para la preparación de los datos antes de entrenar el modelo, ya que las características de las variables son separadas según los objetivos, además de que el modelo como tal necesita saber que variables va a utilizar para poder así realizar las predicciones.

Por otra parte, tenemos el pre - procesamiento y el balanceo, pasos importantes ya que aquí se pueden tener un conjunto de datos que no poseen la misma escala, es decir, “desbalanceados”, y al aplicar SMOTE podemos intentar buscar un balance para así mejorar el rendimiento de los modelos, de modo que aseguramos que las características tengan una escala comparable.

Entonces, tenemos por resultado que este fragmento del código prepara y organiza los datos de manera adecuada para el entrenamiento del modelo asegurando que se puedan manejar correctamente los datos y que las características tengan una escala adecuada para su ejecución en Machine Learning.

```
# Modelo base ajustado manualmente
rf = RandomForestClassifier(
    n_estimators=300,           # Más árboles para estabilidad
    max_depth=15,               # Reducir complejidad para generalizar mejor
    min_samples_split=10,       # Divisiones mínimas más altas
    min_samples_leaf=5,         # Hojas mínimas más grandes
    class_weight='balanced',    # Balance automático de clases
    bootstrap=True,
    random_state=42
)
```

Por otro lado, este código crea el modelo de clasificación mediante el uso del algoritmo llamado “RANDOM FOREST”. Lo que hace, es básicamente analizar los datos dados y predecir una categoría para cada caso, por ejemplo, “un cliente se va o se quedan en el banco”. Las posibles opciones que se configuran en el código ayudan a mejorar la precisión del modelo, bajo los siguientes puntos:

- Se usan más árboles para hacerlo más estable.
- Se ajustan las divisiones dentro de los datos para que el modelo no se sobreajuste (es decir, para que no aprenda demasiado de los datos de entrenamiento).
- Se asegura de que las clases estén balanceadas, lo que significa que, si hay muchas más instancias de una clase que de otra, se ajusta para que no se vea afectado.
- Se añaden otros detalles para mejorar el comportamiento general del modelo.

Por consecuencias tenemos que este código ajusta el modelo para que sea más efectivo y preciso a la hora de realizar las predicciones.

```
# División del dataset
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y
)

# Entrenar el modelo
pipeline.fit(X_train, y_train)

# Evaluar en el conjunto de entrenamiento
train_accuracy = pipeline.score(X_train, y_train)
print(f"Precisión en entrenamiento: {train_accuracy:.2%}")

# Evaluar en el conjunto de prueba
test_accuracy = pipeline.score(X_test, y_test)
print(f"Precisión en prueba: {test_accuracy:.2%}")

# Promedio de precisiones
average_accuracy = (train_accuracy + test_accuracy) / 2
print(f"Promedio de precisión: {average_accuracy:.2%}")

# Predicciones y probabilidades
y_pred = pipeline.predict(X_test)
y_proba = pipeline.predict_proba(X_test)[:, 1]  # Probabilidad para clase positiva
```

Este fragmento del código está entrenando y evaluando un modelo de aprendizaje automático, y se divide en varias partes clave:

1. **División del dataset**:
    - El conjunto de datos (X y y) se divide en dos partes: entrenamiento y prueba.
    - **X_train** y **y_train** se usan para entrenar el modelo, mientras que **X_test** y **y_test** se usan para evaluar qué tan bien está funcionando el modelo.

- test_size=0.3 significa que el 30% de los datos se utilizarán para pruebas y el 70% para entrenamiento.
- stratify=y asegura que la división mantiene la proporción original de las clases en ambas partes.

2. **Entrenar el modelo**:
   - El modelo (representado por pipeline) se entrena con los datos de entrenamiento (X_train, y_train).
3. **Evaluar precisión**:
   - Se calcula y se imprime la precisión del modelo tanto en los datos de entrenamiento como en los de prueba. La precisión indica cuántas veces el modelo hizo una predicción correcta.
4. **Promedio de precisión**:
   - Se calcula el promedio de la precisión en los conjuntos de entrenamiento y prueba para tener una idea de cómo está funcionando el modelo en general.
5. **Predicciones y probabilidades**:
   - Se hacen predicciones sobre el conjunto de prueba con y_pred. y_proba devuelve las probabilidades de que cada predicción pertenezca a la clase positiva (por ejemplo, "sí" en una clasificación binaria).
6. **Reporte de clasificación**:
   - Finalmente, se imprime un reporte detallado sobre el rendimiento del modelo, como la precisión, recall, f1-score, entre otros, para cada clase.

En resumen y en palabras claves, el código divide los datos, entrena un modelo, evalúa su rendimiento en diferentes conjuntos y genera un reporte detallado para entender su efectividad.

```
# Reporte de clasificación
print("Reporte de Clasificación:\n", classification_report(y_test, y_pred))

# Matriz de Confusión (sin modificaciones adicionales)
plt.figure(figsize=(8, 6))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues', cbar=False,
            annot_kws={"size": 16}, linewidths=0.5, linecolor='black', square=True)
```

Este código está generando un reporte de clasificación y visualizando la matriz de confusión para evaluar el rendimiento de un modelo de clasificación. Aquí está el desglose:

Reporte de clasificación:

- classification_report(y_test, y_pred) genera un resumen con las métricas clave del modelo: precisión, recall, f1-score, y soporte (número de muestras) para cada clase.
- Se imprime el reporte para evaluar qué tan bien está funcionando el modelo en la predicción de las clases, en este caso, "Exited" (se fue) y "No Exited" (no se fue).

Matriz de Confusión: Como concepto, tenemos que la matriz de confusión muestra el número de predicciones correctas e incorrectas que hace el modelo, organizadas en una tabla.

- sns.heatmap genera una representación visual de la matriz de confusión usando colores para indicar la frecuencia de las predicciones:

- Los valores son anotados en la celda (annot=True).
- Se usa un mapa de colores Blues para facilitar la interpretación visual.
- Los valores de la matriz (cuentas de predicciones) están formateados como enteros (fmt='d').

Configuraciones de la visualización:

Se ajusta el tamaño de la figura a 8x6 para mejor visualización (plt.figure(figsize=(8, 6))).

Se añaden títulos y etiquetas a los ejes X e Y con un tamaño de fuente apropiado (plt.title, plt.xlabel, plt.ylabel).

Se personalizan los valores de los ejes para mostrar las clases específicas ("No Exited" y "Exited") y el tamaño de fuente de las etiquetas de los ejes.

Se elimina la cuadrícula extra con plt.grid(False). plt.tight_layout() se utiliza para ajustar automáticamente el diseño y evitar que el contenido se sobreponga.

Teniendo entonces como resultado que este código evalúa el rendimiento del modelo mostrando las métricas clave en un reporte de clasificación y visualizando la matriz de confusión para entender cómo clasifica correctamente o no las instancias del conjunto de prueba.

```
# Mostrar gráfico
plt.show()

# Curva ROC y AUC
fpr, tpr, thresholds = roc_curve(y_test, y_proba)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, label=f'Curva ROC (AUC = {roc_auc:.2f})', color='darkorange', lw=2)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')  # Línea diagonal de referencia
plt.title('Curva ROC')
plt.xlabel('Tasa de Falsos Positivos (FPR)')
plt.ylabel('Tasa de Verdaderos Positivos (TPR)')
plt.legend(loc='lower right')
plt.grid(True)
plt.show()
```

En primera instancia, tenemos que se utiliza el código para mostrar los gráficos generados por Matplotlib, en este caso, las gráficas de la matriz de confusión y la curva ROC que se han definido antes.

Y que "# Curva ROC y AUC" calculan los valores necesarios para trazar la curva ROC, que muestra la relación entre la Tasa de Falsos Positivos (FPR) y la Tasa de Verdaderos Positivos (TPR) para diferentes umbrales de clasificación.

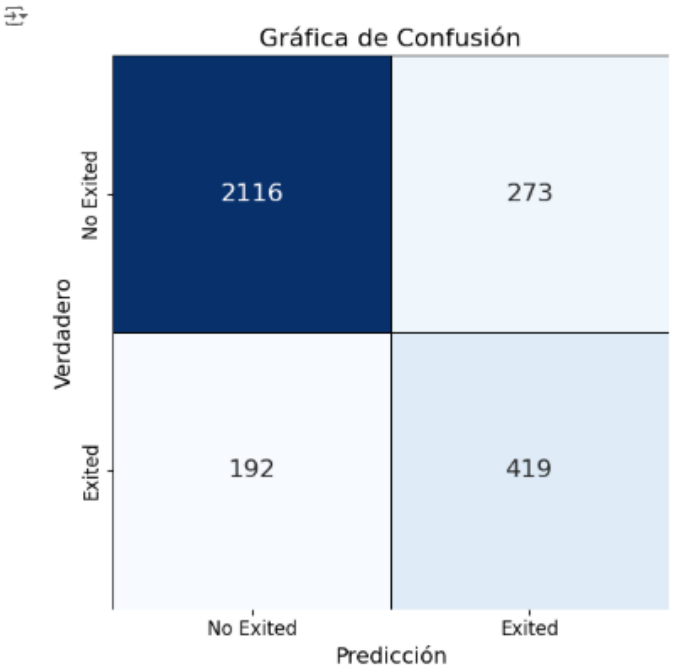
En resumen, este código genera y visualiza la curva ROC, que es una herramienta útil para evaluar cómo un modelo clasifica los datos en términos de falsos positivos y verdaderos positivos. El AUC asociado indica cuán eficaz es el modelo para distinguir entre las clases.

Ahora bien, teniendo en cuenta la explicación previa se da nuevamente la exportación del nuevo dataset, el cual nos arrojará la precision del mismo.

```
# Exportar modelo
joblib.dump(pipeline, '/content/drive/MyDrive/PROYECTOII/MODELO.pkl')  # Guardar el modelo
print("Modelo exportado correctamente.")
```

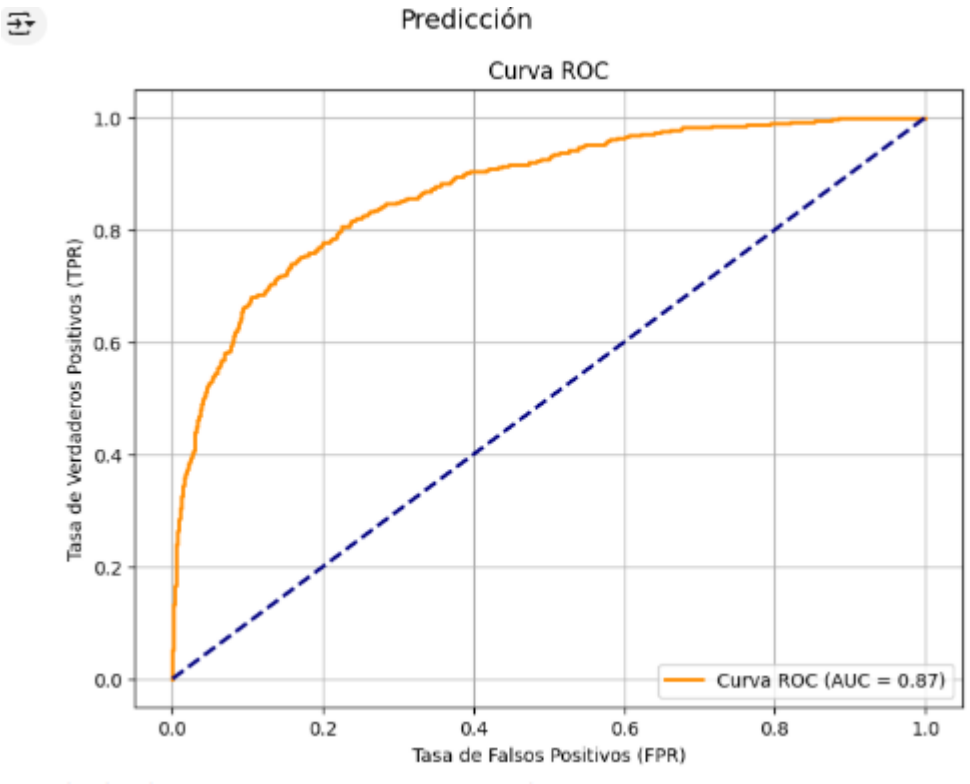
```
Precisión en entrenamiento: 92.09%
Precisión en prueba: 84.50%
Promedio de precisión: 88.29%
Reporte de Clasificación:
              precision    recall  f1-score   support

           0       0.92      0.89      0.90      2389
           1       0.61      0.69      0.64       611

    accuracy                           0.84      3000
   macro avg       0.76      0.79      0.77      3000
weighted avg       0.85      0.84      0.85      3000
```

Una vez realizado esto, tenemos como resultado, se muestran en las gráficas a continuación, que vienen siendo los resultados de los códigos explicados anteriormente:



Lo que observamos acá es el resultado del análisis de si los usuarios iban a permanecer o desertar de los servicios bancarios.

El modelo fue bastante bueno prediciendo cuando la gente se queda, aunque todavía tiene margen de mejora con las predicciones de quienes se van. Esto nos puede ayudar a enfocarnos en mejorar cómo detectamos las señales de que alguien podría abandonar el servicio.



Esta gráfica nos muestra qué tan bueno es nuestro modelo para hacer predicciones.

Se puede observar que el valor obtenido de “0.87” (AUC = 0.87) es como una calificación de nuestro modelo, donde:

- 1.0 sería una calificación perfecta
- 0.5 sería como adivinar al azar
- Cualquier valor arriba de 0.8 se considera bastante bueno

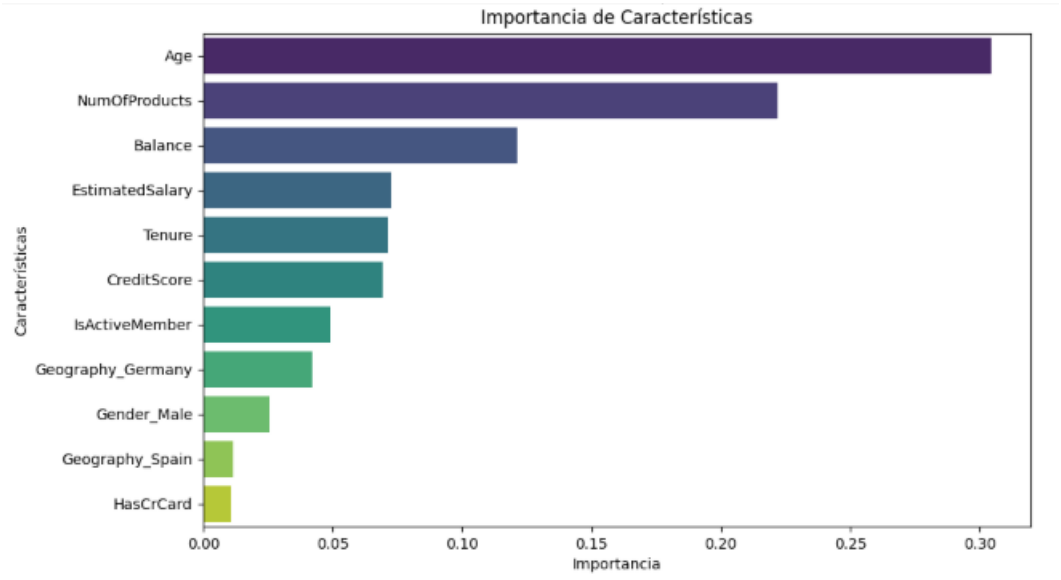
-Datos obtenidos de: (UNIVERSIDAD CARDENAL HERRERA, 2014)

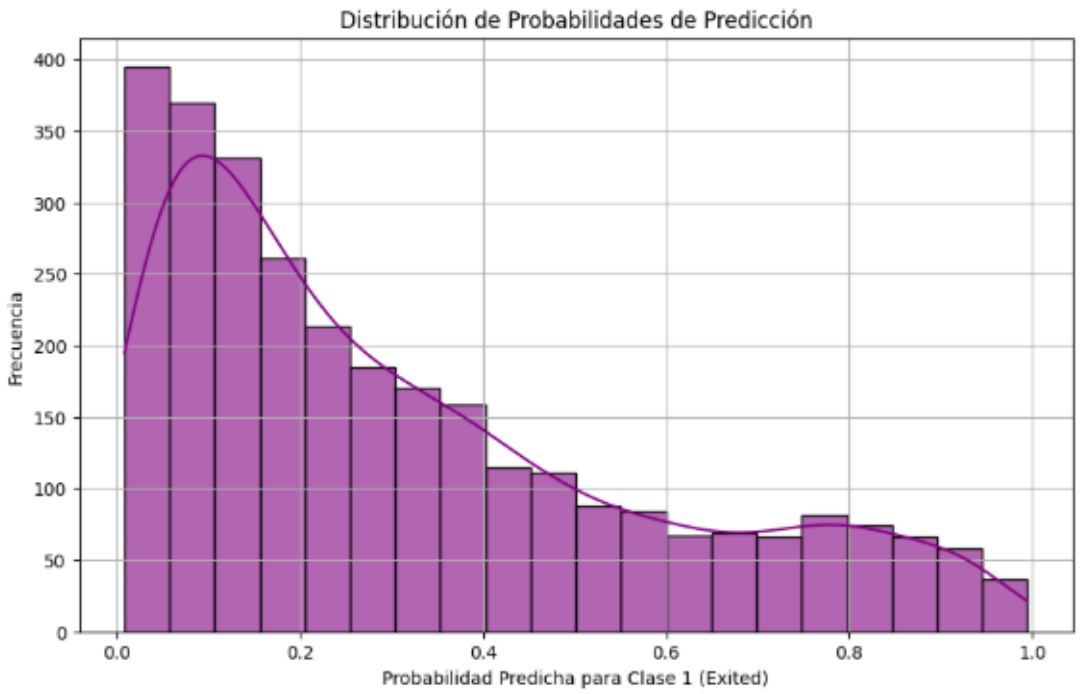
En este caso, tenemos como resultado que nuestro modelo obtuvo un 0.87, lo que en términos simples significa que:

- Es mucho mejor que adivinar al azar
- Acierta aproximadamente 87% de las veces
- Es bastante confiable para predecir si un usuario se quedará o se irá

Según la siguiente gráfica en cuanto a la importancia de características, tenemos que, en este análisis propiamente, podemos ver que la edad del cliente es el factor más decisivo para predecir su comportamiento de churn, seguido por el número de productos o servicios que tiene y seguidamente, influye en tercera posición su balance bancario.

Curiosamente, factores que podríamos pensar son importantes, como el salario estimado o el puntaje crediticio, tienen un impacto moderado en las predicciones. Es interesante notar que características como la ubicación geográfica, el género o tener una tarjeta de crédito tienen una influencia mínima en el comportamiento del cliente, lo que sugiere que el banco tiene una base de clientes bastante uniforme en términos de estos aspectos.

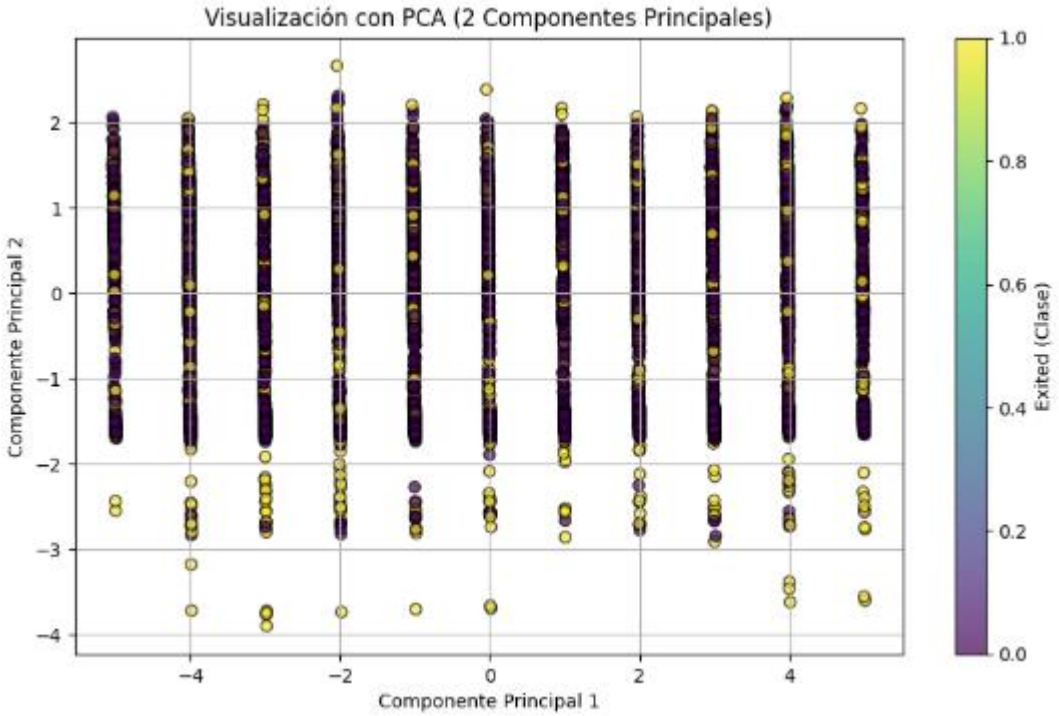




La gráfica de distribución de probabilidades de predicción nos muestra cómo el modelo distribuye sus predicciones sobre la probabilidad de que un cliente se vaya del banco. La mayor concentración de predicciones está en el lado izquierdo como se puede observar, esto nos brinda como resultado que, en teoría, el modelo predice que la mayoría de los clientes se quedarán.

Hay un descenso gradual hacia la derecha, con algunos picos pequeños, indicando que hay menos casos donde el modelo está muy seguro de que los clientes abandonarán el servicio.

Como resultado tenemos que, el modelo parece ser mejor para predecir que los clientes se quedan, pero tiene algunas dificultades para predecir quienes se irán, aunque bien, en caso hipotético, este dato se podría obtener por descarte.

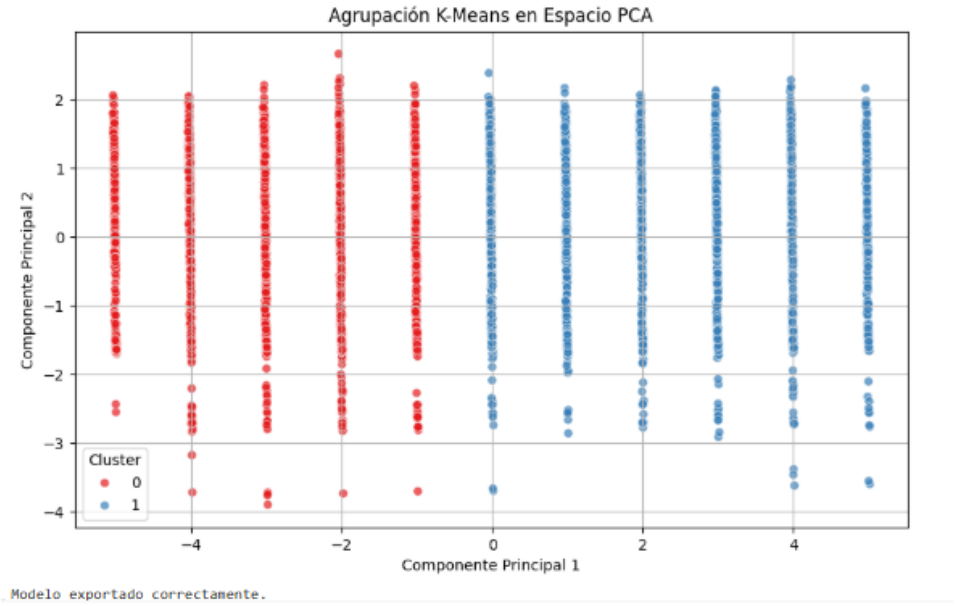


Esta gráfica de visualización con PCA (Análisis de Componentes Principales) tiene como función intentar visualizar patrones en los datos del banco en 2 dimensiones. Lo que vemos son como "columnas" de puntos que se distribuyen verticalmente, donde cada punto representa un cliente y los colores indican la probabilidad de que se vayan (siendo los puntos amarillos los más probables de hacer abandono).

Finalmente, Al observar esta gráfica, podemos ver cómo el análisis K-means ha logrado separar a los clientes del banco en dos grupos muy claros, representados en rojo y azul.

Estos grupos aparecen también como "columnas" bien definidas, casi como un código de barras, donde cada columna parece representar diferentes perfiles de clientes. La separación entre la zona roja (izquierda) y azul (derecha) es bastante clara, sugiriendo que el banco tiene dos tipos principales de clientes con características muy distintas.

Es interesante notar cómo algunos puntos se alejan de sus grupos, como pequeños casos excepcionales, pero en general, los clientes mantienen patrones muy consistentes dentro de sus respectivos grupos. Esta visualización nos ayuda a entender que, aunque cada cliente es único, existen patrones claros de comportamiento que permiten al banco anticipar mejor sus necesidades y riesgos.



Para dar cierre al documento, tenemos que en términos generales el análisis de las predicciones del modelo muestra una clara tendencia hacia la identificación de clientes leales, con una mayor concentración de probabilidades bajas de deserción.

También, siendo objetivos logramos observar como el modelo tiene un buen desempeño al predecir que los clientes se quedarán, sin embargo, su capacidad para identificar a aquellos que abandonarán el banco cuenta con algún margen de error… Mejorar esta capacidad permitirá tomar decisiones más informadas y efectivas en la retención de clientes, lo cual es fundamental para el éxito a largo plazo de la institución financiera.

Conclusiones:

1. Los clientes que mantienen edad baja y que tienen una relación corta con el banco muestran una mayor tendencia a desertar. Esto sugiere que es fundamental implementar estrategias de fidelización específicas para este segmento, que ayuden a fortalecer la relación desde el principio. Al centrar los esfuerzos en estos clientes, el banco podría reducir significativamente la pérdida de clientes nuevos que todavía no han alcanzado una fase de lealtad. Esto podría incluir estrategias como ofertas exclusivas o incentivos por permanencia en los primeros meses de relación con el banco.
2. Al implementar el modelo de soporte, se logra identificar con precisión los usuarios con mayor riesgo de deserción, lo que permitió priorizar estrategias personalizadas de retención, de modo que se generó una mejora en la relación costo / beneficios con respecto a las iniciativas de fidelización.
3. La segmentación de clientes utilizando técnicas de agrupación (clústeres) ha demostrado ser útil para identificar y agrupar a los clientes según el riesgo de deserción. Esto permite que el banco adopte una estrategia de retención más personalizada y eficiente, ya que cada segmento puede recibir acciones diseñadas específicamente para sus características y nivel de riesgo. Por ejemplo, para clientes de alto riesgo, las intervenciones pueden ser más inmediatas e intensivas, mientras que para clientes de menor riesgo se pueden implementar estrategias de retención más generales.

Recomendaciones:

1. Se recomienda diseñar estrategias específicas para los nuevos clientes, de modo que se genere un interés por parte del usuario y que conforme el paso del tiempo garantice la permanencia dentro la institución financiera, esto fortalecerá la relación desde el inicio y reducirá el riesgo de deserción en los primeros años de servicio.
2. Adicionalmente, se recomienda también implementar estrategias de retención personalizadas basadas en el modelo predictivo, al hacer uso de ellas, se priorizan los clientes en riesgo, ofrecerles soluciones específicas, como, por ejemplo, asesoramiento financiero, beneficios, paquetes personalizados etc. de acuerdo a sus necesidades, asegurará que los recursos de retención se asignen de manera eficiente…
3. Utilizar técnicas de clustering para agrupar a los usuarios por su nivel de riesgo de deserción y características comunes, permite un enfoque intensivo para los clientes de alto riego y, por el contrario, un enfoque general en los clientes de bajo riesgo.

Anexos:

Google Colab:

https://colab.research.google.com/drive/12ETQuRmGah3OUdEvcH8XH__1_WoRAcFC?usp=sharing

Video:
https://www.canva.com/design/DAGXaMONDrU/VAUKo1UQ1B2Hbv9N-BPviw/edit?utm_content=DAGXaMONDrU&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Bibliografía

ESIC. (MARZO de 2024). *Business Analytics*. (E. B. School, Editor) Obtenido de https://www.esic.edu/rethink/tecnologia/ciclo-vida-datos-c

Machine Learning Guide Oil & Gas. (2021). *Science Direct.* WEB: Python. Obtenido de https://www.sciencedirect.com/topics/computer-science/gradient-boosting#:~:text=Gradient%20boosting%20is%20a%20type,gradually%20minimizing%20a%20loss%20function.

Morales, L. E., & Flores, N. (2023). *Recopilación de datos para bases análiticas.* Monterrey: Educ.Continua.

Narvaez, M. (2018). *Ciclo de vida de los datos: Qué es y qué etapas tiene.* Leader Middle West.

Navarro, S. (2022). Business Intelligence & Big Data Advisor & Coordinadora del Bootcamp en Data Science, Big Data & Machine Learning. *KeepCoding*.

Pita, A. (s.f.). *Youtube*. Obtenido de #FutureofData21: Departamento de Marketing y Comunicación de IEBS Business School

UNIVERSIDAD CARDENAL HERRERA. (2014). *MODELOS PREDICTIVOS.* Valencia: AECS.

Universidad Oberta . (s.f.). *Ciencia de datos.* Cataluña - España: UOC.