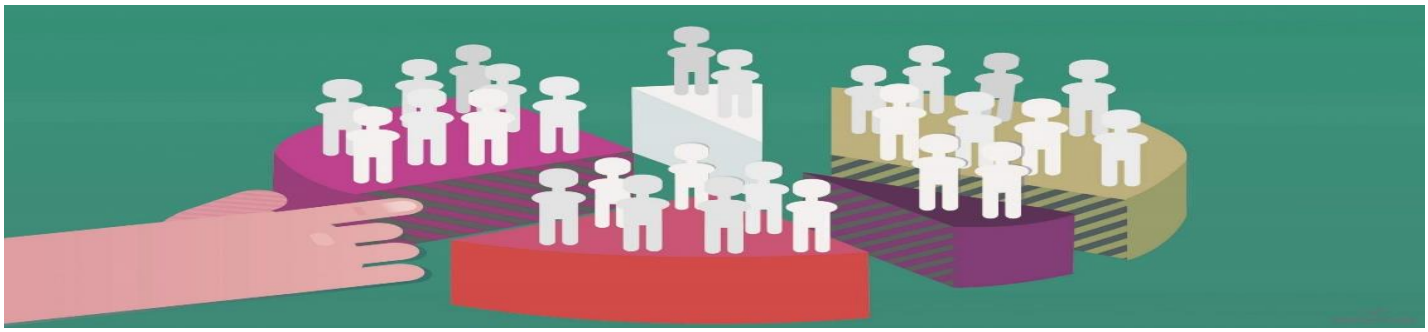


Proyecto de Optimización de Estrategias de Fidelización con Machine Learning

En este proyecto, el objetivo fue desarrollar un modelo predictivo que ayudara a identificar y comprender mejor a los clientes con alto riesgo de deserción (churn) en una empresa, con la intención de diseñar estrategias de fidelización efectivas y optimizar la retención. Aprovechando un conjunto de datos ficticio que simula comportamientos y características relevantes de clientes, se buscó descubrir patrones que servirían para proponer soluciones al churn.

Para alcanzar estos objetivos, se emplearon técnicas avanzadas de Machine Learning, como el análisis de segmentación con K-Means y la regresión logística para predicciones de churn. La metodología también incluyó un análisis exploratorio detallado (EDA) para visualizar la relación entre distintas variables clave y la tasa de deserción, apoyándose en librerías como pandas, scikit-learn, matplotlib y seaborn para el procesamiento de datos, modelado y visualización de resultados.

Este proyecto refleja mis habilidades en el análisis de datos y el diseño de modelos predictivos, así como mi capacidad para aplicar conocimientos de Machine Learning en la solución de problemas de negocios reales. La combinación de análisis exploratorio, segmentación y modelado predictivo permite crear un enfoque integral, aplicable en el desarrollo de estrategias de retención personalizadas y en la toma de decisiones informadas basadas en datos.

Contaré como trabaje el Proyecto paso a paso

Paso 1: Exploración Inicial de Datos
Se inició el análisis con una muestra de las primeras cinco filas del dataset mediante sample(5) y una revisión rápida de la estructura y el tamaño con shape e info. Esto ayudó a conocer el tipo de datos y detectar valores faltantes o inconsistencias iniciales.

Paso 2: Análisis Univariado
Realicé un análisis univariado para comprender la distribución de las variables clave: Edad, Gasto Mensual, Antigüedad, y Satisfacción del Cliente. Se usaron histogramas para cada variable y un gráfico de torta para la variable Churn, lo que brindó un panorama general de las características de los clientes.

Paso 3: Análisis Bivariado
El análisis bivariado exploró relaciones entre pares de variables usando boxplots, incluyendo: Satisfacción del Cliente por Gasto Mensual, Gasto Mensual por Edad, y Gasto Mensual por Antigüedad.

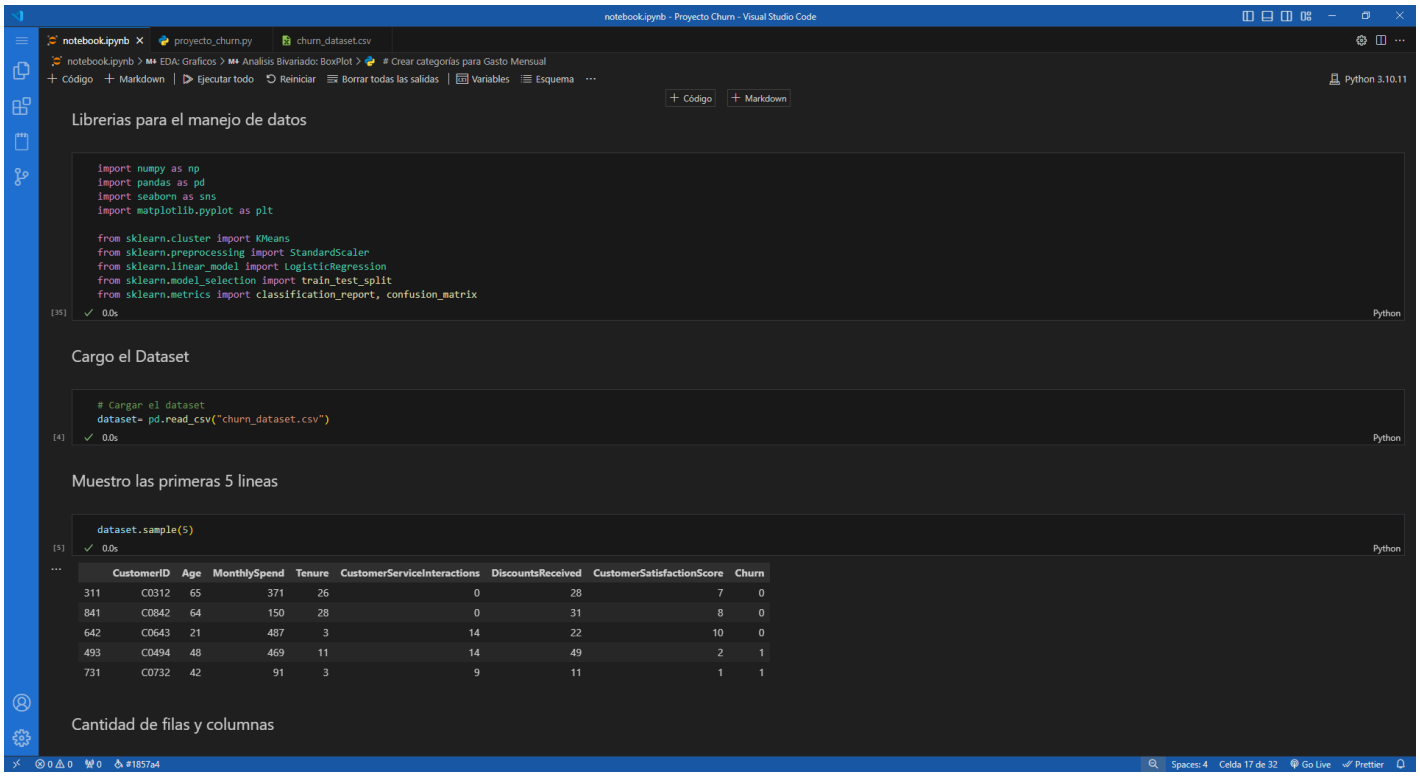
Paso 4: Preprocesamiento de Datos y Método del Codo
Los datos se estandarizaron mediante StandardScaler para asegurar uniformidad en la escala de las variables. Luego, utilicé el método del codo para determinar el número óptimo de clústeres, evaluando la inercia para elegir el valor que mejor segmentara a los clientes.

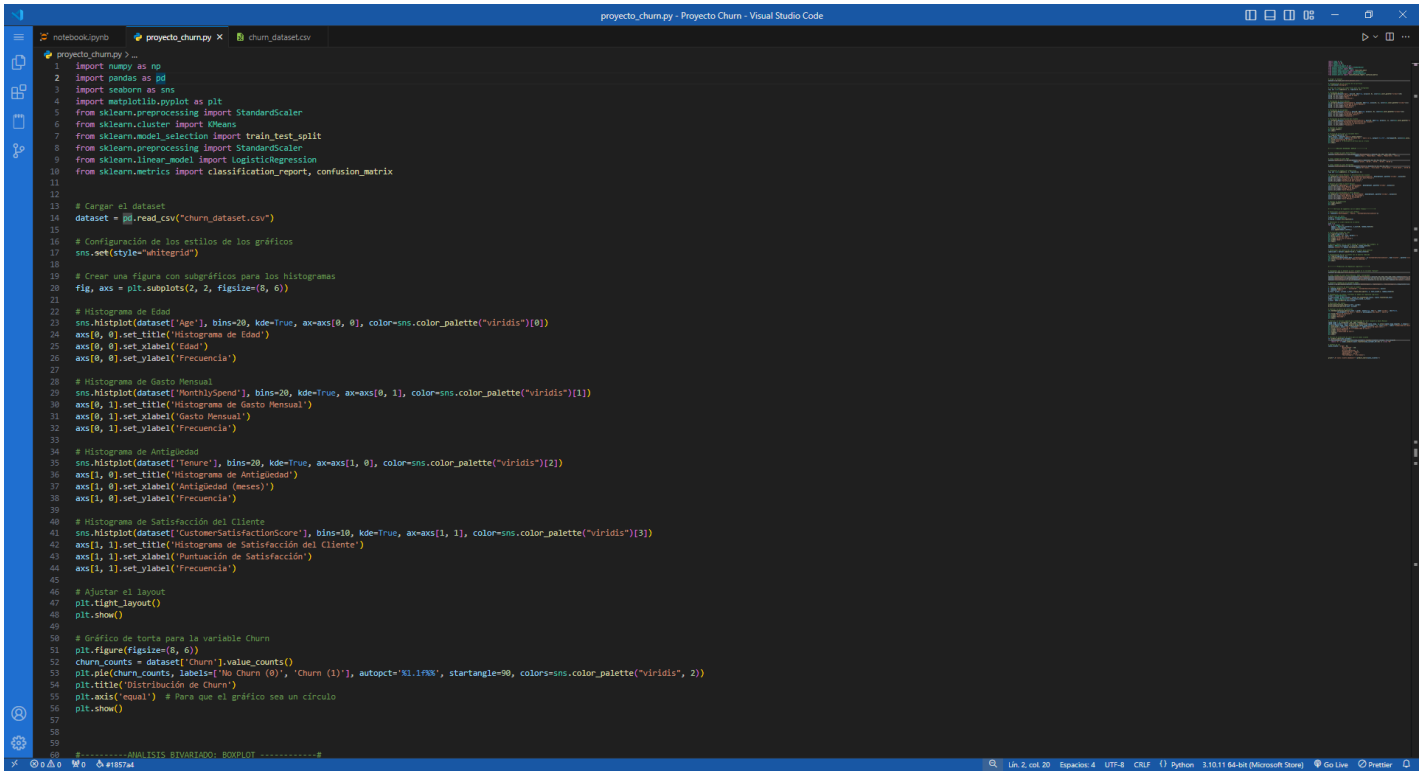
Paso 5: Segmentación de Clientes con K-Means
Con el número óptimo de clústeres determinado, segmenté los datos en tres grupos usando K-Means. Esta segmentación permitió identificar perfiles de clientes con características similares, esenciales para diseñar estrategias de retención específicas.

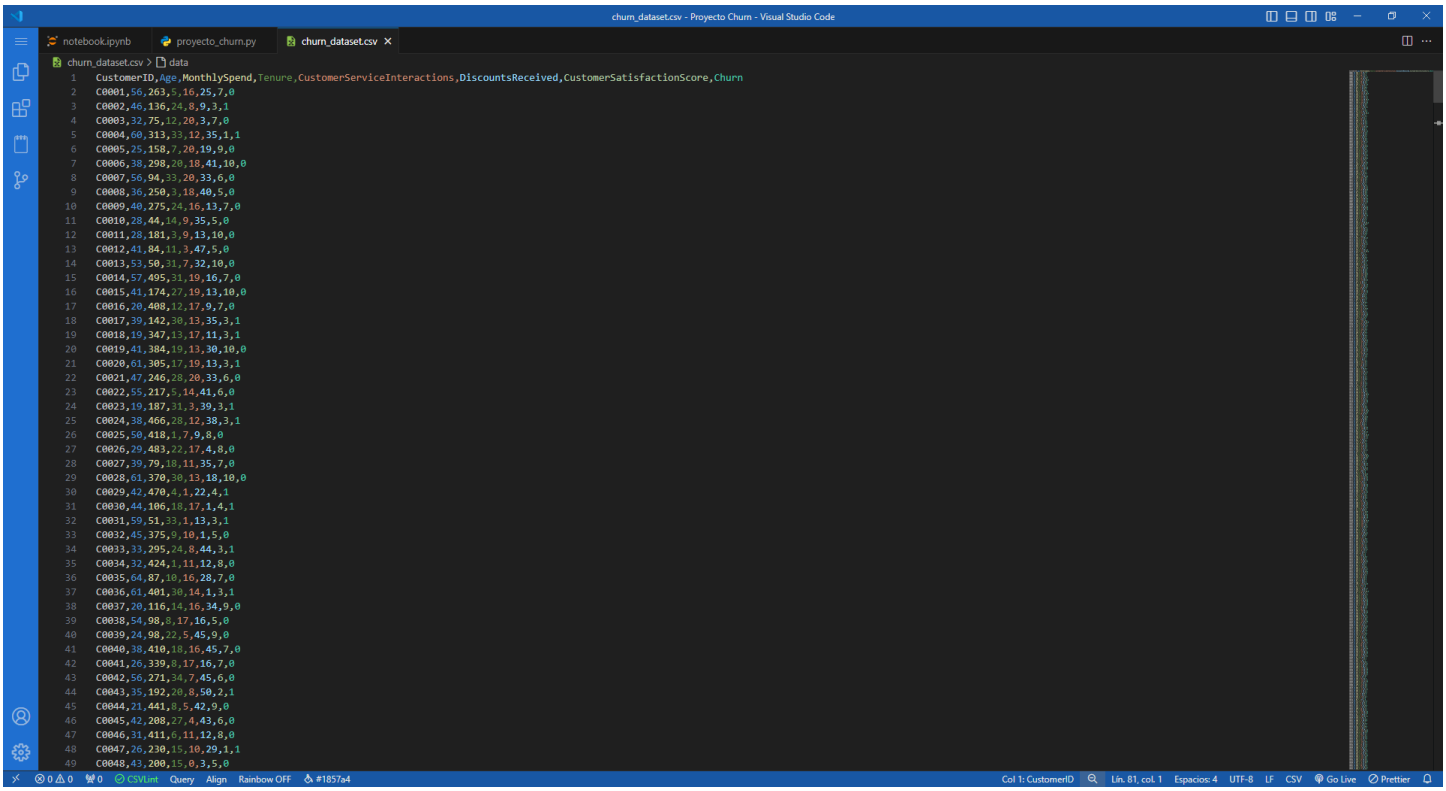
Paso 6: Preprocesamiento Adicional para Regresión Logística
Preprocesé los datos para la etapa de modelado predictivo, con el fin de preparar las variables adecuadamente para la regresión logística. Esta preparación aseguraba la consistencia en las predicciones de probabilidad de churn.

Paso 7: Modelo de Regresión Logística y Evaluación
Entrené una regresión logística y validé el modelo mediante una matriz de confusión, evaluando la precisión en la predicción de churn. Este paso confirmó la efectividad del modelo en diferenciar entre clientes propensos al abandono.

Paso 8: Predicción de Churn para Nuevos Clientes
Finalmente, se probó el modelo al ingresar los datos de un cliente nuevo, prediciendo si abandonaría. Esta predicción sirvió para demostrar la aplicabilidad del modelo en escenarios prácticos de retención de clientes.


notebook.ipynb - Proyecto Churn - Visual Studio Code
notebook.ipynb
proyecto_churn.py
churn_dataset.csv
Librerias para el manejo de datos
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
Cargo el Dataset
# Cargar el dataset
dataset= pd.read_csv("churn_dataset.csv")
Muestro las primeras 5 lineas
dataset.sample(5)
Cantidad de filas y columnas


proyecto_churn.py - Proyecto Churn - Visual Studio Code
proyecto_churn.py
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
# Cargar el dataset
dataset = pd.read_csv("churn_dataset.csv")
# Configuración de los estilos de los gráficos
sns.set(style="whitegrid")
# Crear una figura con subgráficos para los histogramas
fig, axs = plt.subplots(2, 2, figsize=(8, 6))
# Histograma de Edad
sns.histplot(dataset['Age'], bins=20, kde=True, ax=axs[0, 0], color=sns.color_palette("viridis")[0])
axs[0, 0].set_title('Histograma de Edad')
axs[0, 0].set_xlabel('Edad')
axs[0, 0].set_ylabel('Frecuencia')
# Histograma de Gasto Mensual
sns.histplot(dataset['MonthlySpend'], bins=20, kde=True, ax=axs[0, 1], color=sns.color_palette("viridis")[1])
axs[0, 1].set_title('Histograma de Gasto Mensual')
axs[0, 1].set_xlabel('Gasto Mensual')
axs[0, 1].set_ylabel('Frecuencia')
# Histograma de Antigüedad
sns.histplot(dataset['Tenure'], bins=20, kde=True, ax=axs[1, 0], color=sns.color_palette("viridis")[2])
axs[1, 0].set_title('Histograma de Antigüedad')
axs[1, 0].set_xlabel('Antigüedad (meses)')
axs[1, 0].set_ylabel('Frecuencia')
# Histograma de Satisfacción del Cliente
sns.histplot(dataset['CustomerSatisfactionScore'], bins=10, kde=True, ax=axs[1, 1], color=sns.color_palette("viridis")[3])
axs[1, 1].set_title('Histograma de Satisfacción del Cliente')
axs[1, 1].set_xlabel('Puntuación de Satisfacción')
axs[1, 1].set_ylabel('Frecuencia')
# Ajustar el layout
plt.tight_layout()
plt.show()
# Gráfico de torta para la variable Churn
plt.figure(figsize=(8, 6))
churn_counts = dataset['Churn'].value_counts()
plt.pie(churn_counts, labels=['No Churn (0)', 'Churn (1)'], autopct='%1.1f%%', startangle=90, colors=sns.color_palette("viridis", 2))
plt.title('Distribución de Churn')
plt.axis('equal') # Para que el gráfico sea un círculo
plt.show()
#----------ANALISIS BIVARIADO: BOXPLOT ----------#


churn_dataset.csv - Proyecto Churn - Visual Studio Code
notebook.ipynb
proyecto_churn.py
churn_dataset.csv
churn_dataset.csv > data
CustomerID,Age,MonthlySpend,Tenure,CustomerServiceInteractions,DiscountsReceived,CustomerSatisfactionScore,Churn
C0001,56,263,5,16,25,7,0
C0002,46,136,24,8,9,3,1
C0003,32,75,12,20,3,7,0
C0004,60,313,33,12,35,1,1
C0005,25,158,7,20,19,9,0
C0006,38,298,20,18,41,10,0
C0007,56,94,33,20,33,6,0
C0008,36,250,3,18,40,5,0
C0009,40,275,24,16,13,7,0
C0010,28,44,14,9,35,5,0
C0011,28,181,3,9,13,10,0
C0012,41,84,11,3,47,5,0
C0013,53,50,31,7,32,10,0
C0014,57,495,31,19,16,7,0
C0015,41,174,27,19,13,10,0
C0016,20,408,12,17,9,7,0
C0017,39,142,30,13,35,3,1
C0018,19,347,13,17,11,3,1
C0019,41,384,19,13,30,10,0
C0020,61,305,17,19,13,3,1
C0021,47,246,28,20,33,6,0
C0022,55,217,5,14,41,6,0
C0023,19,187,31,3,39,3,1
C0024,38,466,28,12,38,3,1
C0025,50,418,1,7,9,8,0
C0026,29,483,22,17,4,8,0
C0027,39,79,18,11,35,7,0
C0028,61,370,30,13,18,10,0
C0029,42,470,4,1,22,4,1
C0030,44,106,18,17,1,4,1
C0031,59,51,33,1,13,3,1
C0032,45,375,0,10,1,5,0
C0033,33,295,24,8,44,3,1
C0034,32,424,1,11,12,8,0
C0035,64,87,10,16,28,7,0
C0036,61,401,30,14,1,3,1
C0037,20,116,14,16,34,9,0
C0038,54,98,8,17,16,5,0
C0039,24,98,22,5,45,9,0
C0040,38,410,18,16,45,7,0
C0041,26,339,8,17,16,7,0
C0042,56,271,34,7,45,6,0
C0043,35,192,20,8,50,2,1
C0044,21,441,8,5,42,9,0
C0045,42,208,27,4,43,6,0
C0046,31,411,6,11,12,8,0
C0047,26,230,15,10,29,1,1
C0048,43,200,15,0,3,5,0
Col 1: CustomerID
Lín. 81, col. 1
Espacios: 4
UTF-8
LF
CSV
Go Live
Prettier