**Python code**

**Load Libraries and Data**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

import xgboost as xgb

# Load the dataset

data = pd.read\_csv('Bank\_Customer\_Churn.csv') # Replace with your dataset path

**Exploratory Data Analysis (EDA)**

print(data.head())

print(data.describe())

print(data.isnull().sum())

sns.countplot(x='Exited', data=data)

plt.title('Distribution of Exited (Churn)')

plt.show()

**Data Preprocessing**

# Encode categorical variables

le\_gender = LabelEncoder()

data['Gender'] = le\_gender.fit\_transform(data['Gender'])

# One-hot encode categorical features

data = pd.get\_dummies(data, columns=['Geography'], drop\_first=True)

# Split the data into features and target

X = data.drop(['Exited', 'RowNumber', 'CustomerId', 'Surname'], axis=1)

y = data['Exited']

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Scale the features

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

**Build and Train the Model**

# Initialize the XGBoost model

model = xgb.XGBClassifier(use\_label\_encoder=False, eval\_metric='mlogloss')

# Train the model

model.fit(X\_train, y\_train)

**Evaluate the Model**

# Make predictions

y\_pred = model.predict(X\_test)

# Evaluate the model

print(f'Accuracy: {accuracy\_score(y\_test, y\_pred)}')

print('Confusion Matrix:')

print(confusion\_matrix(y\_test, y\_pred))

print('Classification Report:')

print(classification\_report(y\_test, y\_pred))

**Feature Importance**

# Plot feature importance

xgb.plot\_importance(model)

plt.show()

**Bash**

pip install pandas numpy matplotlib seaborn scikit-learn xgboost