**Customer Churn Prediction using Logistic Regression**

**Introduction**

Customer churn refers to the phenomenon where customers stop using a company's products or services. Predicting churn helps businesses retain customers and optimize marketing strategies. In this project, we use a logistic regression model to predict whether a customer will churn based on various customer attributes such as gender, subscription type, contract length, and payment delays.

**1. Import Libraries**

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

from sklearn.linear\_model import LogisticRegression

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

**2. Load Dataset**

df = pd.read\_csv("customer\_churn\_dataset-testing-master.csv")

df.drop('CustomerID', axis=1, inplace=True)

**3. Data Exploration**

# Check for missing values

print(df.isnull().sum())

# Gender distribution

print(df['Gender'].value\_counts())

**Encode Gender**

df['Gender'] = df['Gender'].apply(lambda x: 0 if x=='Female' else 1)

**4. Encode Categorical Variables**

**Subscription Type**

dummy\_pdf = pd.get\_dummies(df['Subscription Type'], dtype=int, drop\_first=True)

df2 = pd.concat([df, dummy\_pdf], axis=1).drop('Subscription Type', axis=1)

**Contract Length**

dummy\_pdf\_2 = pd.get\_dummies(df2['Contract Length'], dtype=int, drop\_first=True)

df3 = pd.concat([df2, dummy\_pdf\_2], axis=1).drop('Contract Length', axis=1)

**5. Outlier Detection and Capping**

plt.boxplot(df3["Payment Delay"])

plt.title("Payment Delay Boxplot")

plt.show()

# Calculate IQR and cap outliers

Q1 = df3["Payment Delay"].quantile(0.25)

Q3 = df3["Payment Delay"].quantile(0.75)

IQR = Q3 - Q1

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

df3["Payment Delay\_capped"] = np.where(

df3["Payment Delay"] < lower\_bound, lower\_bound,

np.where(df3["Payment Delay"] > upper\_bound, upper\_bound, df3["Payment Delay"])

)

**6. Feature Scaling and Train-Test Split**

x = df3.drop('Churn', axis=1)

y = df3['Churn']

scaler = StandardScaler()

x\_scaled = scaler.fit\_transform(x)

x\_train, x\_test, y\_train, y\_test = train\_test\_split(

x\_scaled, y, test\_size=0.2, random\_state=42, stratify=y

)

**7. Logistic Regression Model Training**

logi = LogisticRegression()

logi.fit(x\_train, y\_train)

y\_pred = logi.predict(x\_test)

**8. Model Evaluation**

**Confusion Matrix and Accuracy**

cm = confusion\_matrix(y\_test, y\_pred)

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

plt.figure(figsize=(6,4))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.title("Confusion Matrix")

plt.show()

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

**9. ROC Curve and AUC Score**

y\_pred\_prob = logi.predict\_proba(x\_test)[:, 1]

fpr, tpr, thresholds = roc\_curve(y\_test, y\_pred\_prob)

roc\_auc = auc(fpr, tpr)

print("AUC Score:", roc\_auc)

plt.figure(figsize=(8, 6))

plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (AUC = %0.2f)' % roc\_auc)

plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic (ROC)')

plt.legend(loc="lower right")

plt.show()