Predicting customer churn using machine learning to uncover hidden patterns

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

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

from sklearn.linear\_model import LogisticRegression

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

from sklearn.preprocessing import StandardScaler

# --- 1. Generate Synthetic Customer Churn Data ---

np.random.seed(42) # for reproducibility

n\_customers = 1000

# Features: Monthly Charges, Tenure (months), Total Charges (calculated)

monthly\_charges = np.random.normal(loc=60, scale=20, size=n\_customers)

tenure = np.random.randint(1, 72, size=n\_customers) # 1 to 72 months

# Introduce some noise and dependency for churn

# Churn is more likely with higher monthly charges, lower tenure

churn\_probability = (

    0.1  # base churn rate

    + (monthly\_charges - 40) \* 0.005

    - (tenure - 30) \* 0.003

    + np.random.normal(loc=0, scale=0.1, size=n\_customers)

)

churn\_probability = np.clip(churn\_probability, 0, 1) # Ensure probabilities are between 0 and 1

churn = (np.random.rand(n\_customers) < churn\_probability).astype(int) # 1 for churn, 0 for no churn

# Create a DataFrame

data = pd.DataFrame({

    'MonthlyCharges': monthly\_charges,

    'Tenure': tenure,

    'Churn': churn

})

print("--- Synthetic Data Head ---")

print(data.head())

print("\n--- Churn Distribution ---")

print(data['Churn'].value\_counts(normalize=True))

# --- 2. Prepare Data for Machine Learning ---

X = data[['MonthlyCharges', 'Tenure']]

y = data['Churn']

# Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42, stratify=y)

# Scale numerical features (important for many ML models, especially Logistic Regression)

scaler = StandardScaler()

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

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

# --- 3. Train a Machine Learning Model (Logistic Regression) ---

model = LogisticRegression(random\_state=42)

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

# --- 4. Make Predictions and Evaluate ---

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

y\_pred\_proba = model.predict\_proba(X\_test\_scaled)[:, 1] # Probability of churn

print("\n--- Model Evaluation ---")

print(f"Accuracy: {accuracy\_score(y\_test, y\_pred):.2f}")

print("\nClassification Report:")

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

# --- 5. Graphical Output ---

plt.style.use('seaborn-v0\_8-darkgrid')

# Plot 1: Feature Distributions by Churn Status

plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)

sns.histplot(data=data, x='MonthlyCharges', hue='Churn', kde=True, palette='viridis', bins=30)

plt.title('Monthly Charges Distribution by Churn')

plt.xlabel('Monthly Charges ($)')

plt.ylabel('Number of Customers')

plt.subplot(1, 2, 2)

sns.histplot(data=data, x='Tenure', hue='Churn', kde=True, palette='viridis', bins=30)

plt.title('Tenure Distribution by Churn')

plt.xlabel('Tenure (Months)')

plt.ylabel('Number of Customers')

plt.tight\_layout()

plt.show()

# Plot 2: Confusion Matrix

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

plt.figure(figsize=(7, 6))

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

            xticklabels=['Predicted No Churn', 'Predicted Churn'],

            yticklabels=['Actual No Churn', 'Actual Churn'])

plt.title('Confusion Matrix for Churn Prediction')

plt.xlabel('Predicted Label')

plt.ylabel('True Label')

plt.show()

# Plot 3: ROC Curve

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

roc\_auc = auc(fpr, tpr)

plt.figure(figsize=(7, 6))

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

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) Curve')

plt.legend(loc="lower right")

plt.show()

print("\n--- Interpretation of Graphical Output ---")

print("1. Feature Distributions: Observe how 'MonthlyCharges' and 'Tenure' differ for churned vs. non-churned customers. You might see that churned customers tend to have higher monthly charges and lower tenure.")

print("2. Confusion Matrix: This shows the number of correct and incorrect predictions.")

print("   - Top-left (True Negatives): Correctly predicted non-churners.")

print("   - Bottom-right (True Positives): Correctly predicted churners.")

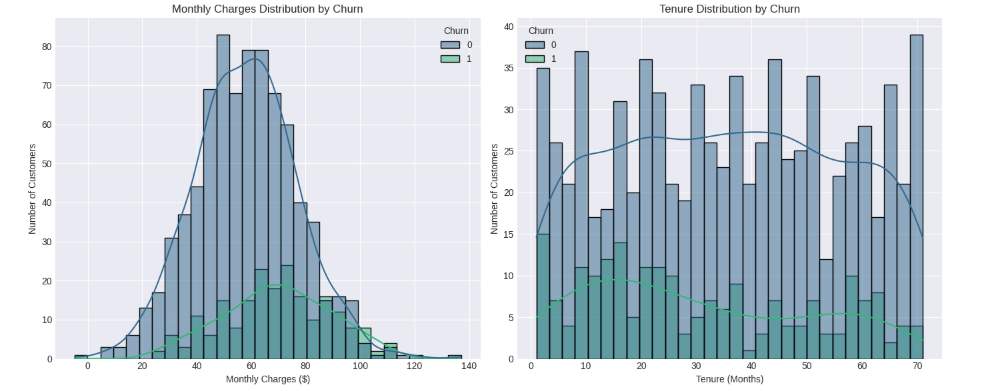
print("   - Top-right (False Positives): Predicted churn, but actually didn't churn (Type I error).")

print("   - Bottom-left (False Negatives): Predicted no churn, but actually churned (Type II error - a costly error for businesses).")

print("3. ROC Curve: This plot illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.")

print("   - A curve closer to the top-left corner indicates a better performing model.")

print("   - The Area Under the Curve (AUC) provides a single measure of overall performance (0.5 is random, 1.0 is perfect).")

**OUTPUT**