# Customer Churn Prediction Using Logistic Regression

##  Step 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 LabelEncoder

from sklearn.linear\_model import LogisticRegression

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

##  Step 2: Load and Inspect Dataset

df = pd.read\_excel("Telco\_customer\_churn.xlsx")

df.columns = df.columns.str.strip().str.replace(' ', '').str.lower()

df.head()

##  Step 3: Explore the Data

print(df.info())

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

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

# Visualize churn distribution

sns.countplot(data=df, x='churn')

plt.title("Customer Churn Distribution")

plt.show()

##  Step 4: Data Cleaning & Preprocessing

# Drop 'customerid'

if 'customerid' in df.columns:

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

# Convert 'totalcharges' to numeric

df['totalcharges'] = pd.to\_numeric(df['totalcharges'], errors='coerce')

df['totalcharges'].fillna(df['totalcharges'].median(), inplace=True)

# Encode 'churn'

df['churn'] = df['churn'].map({'Yes': 1, 'No': 0})

# Label encode remaining object columns

for col in df.select\_dtypes(include='object').columns:

    df[col] = LabelEncoder().fit\_transform(df[col])

df.head()

##  Step 5: Split Data

X = df.drop('churn', axis=1)

y = df['churn']

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

##  Step 6: Train Model

model = LogisticRegression(max\_iter=1000)

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

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

##  Step 7: Evaluate Model

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

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

# Confusion Matrix

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

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

plt.title("Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.show()