#Import Libraries

import pandas as pd

import numpy as np

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

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.ensemble import RandomForestClassifier

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

import seaborn as sns

import matplotlib.pyplot as plt

#Load Dataset

# You can replace this with your own dataset

df = pd.read\_csv('https://raw.githubusercontent.com/blastchar/telco-customer-churn/master/WA\_Fn-UseC\_-Telco-Customer-Churn.csv')

#Data Preprocessing

df.drop(['customerID'], axis=1, inplace=True)

df.replace(" ", np.nan, inplace=True)

df.dropna(inplace=True)

# Convert categorical columns

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

if col != 'Churn':

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

# Encode target variable

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

# Feature Scaling

scaler = StandardScaler()

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

y = df['Churn']

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

#Train-Test Split

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

#Model Training

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

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

#Model Evaluation

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

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

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred))

# Confusion Matrix

sns.heatmap(confusion\_matrix(y\_test, y\_pred), annot=True, fmt='d', cmap='Blues')

plt.title('Confusion Matrix')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()