# Import necessary 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

# Load the dataset (use a CSV file path or URL)

df = pd.read\_csv('https://raw.githubusercontent.com/datasciencedojo/datasets/master/Telco-Customer-Churn.csv')

# Drop customerID as it's not useful

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

# Convert 'TotalCharges' to numeric

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

df.dropna(inplace=True) # Drop missing values

# Encode categorical variables

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

if col != 'Churn':

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

# Encode target column

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

# Split features and target

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

y = df['Churn']

# Scale the features

scaler = StandardScaler()

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

# Split data into training and test sets

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

# Train a Random Forest Classifier

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

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

# Make predictions

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

# Evaluate the model

print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

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

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