import pandas as pd

import numpy as np

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

from sklearn.ensemble import RandomForestClassifier

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

import seaborn as sns

import matplotlib.pyplot as plt

# Load sample dataset (Pima Indians Diabetes Dataset)

def load\_data():

url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"

columns = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',

'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

data = pd.read\_csv(url, names=columns)

return data

# Train the disease prediction model

def train\_model(data):

X = data.drop('Outcome', axis=1)

y = data['Outcome']

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

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

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

predictions = model.predict(X\_test)

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

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

return model

# Predict disease outcome for a new patient

def predict\_new\_patient(model, patient\_data):

patient\_df = pd.DataFrame([patient\_data])

prediction = model.predict(patient\_df)

result = "Likely Diabetic" if prediction[0] == 1 else "Not Diabetic"

print(f"Prediction for new patient: {result}")

# Main execution

if \_\_name\_\_ == "\_\_main\_\_":

data = load\_data()

model = train\_model(data)

# Example patient data: Change values to test

example\_patient = {

'Pregnancies': 2,

'Glucose': 120,

'BloodPressure': 70,

'SkinThickness': 20,

'Insulin': 85,

'BMI': 28.0,

'DiabetesPedigreeFunction': 0.5,

'Age': 33

}

predict\_new\_patient(model, example\_patient)