import pandas as pd

import numpy as np

from sklearn.tree import DecisionTreeClassifier

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

from sklearn.metrics import accuracy\_score

# Load the data from the CSV file

data = pd.read\_csv("/content/drive/MyDrive/ASE/heart.csv")

# Split the data into features and target

X = data.drop("target", axis=1) # Features are all columns except target

y = data["target"] # Target is the column named target

# Split the data into training and testing sets

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

# Create and fit a decision tree classifier

clf = DecisionTreeClassifier()

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

# Make predictions on the test set

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

# Evaluate the accuracy of the model

acc = accuracy\_score(y\_test, y\_pred)

print(f"Accuracy: {acc\*100:.2f}%")

import pandas as pd

import numpy as np

from sklearn.ensemble import RandomForestClassifier

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

from sklearn.metrics import accuracy\_score

# Load the data from the CSV file

data = pd.read\_csv("/content/drive/MyDrive/ASE/heart.csv")

# Split the data into features and target

X = data.drop("target", axis=1) # Features are all columns except target

y = data["target"] # Target is the column named target

# Split the data into training and testing sets

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

# Create and fit a random forest classifier

clf = RandomForestClassifier()

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

# Make predictions on the test set

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

# Evaluate the accuracy of the model

acc = accuracy\_score(y\_test, y\_pred)

print(f"Accuracy: {acc\*100:.2f}%")