import pandas as pd

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

from sklearn.preprocessing import LabelEncoder

from sklearn.ensemble import RandomForestClassifier

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

# Create a dummy dataset

data = pd.DataFrame({

'Weight (grams)': [120, 150, 130, 140, 180, 200, 160, 190, 170, 110],

'Color (1=Red, 2=Yellow, 3=Orange)': [2, 1, 2, 1, 3, 2, 3, 2, 3, 1],

'Fruit': ['Apple', 'Banana', 'Apple', 'Banana', 'Orange', 'Banana', 'Orange', 'Apple', 'Banana', 'Apple']

})

# Split the data into features (X) and the target variable (y)

X = data[['Weight (grams)', 'Color (1=Red, 2=Yellow, 3=Orange)']]

y = data['Fruit']

# Encode the target variable into numerical labels

label\_encoder = LabelEncoder()

y = label\_encoder.fit\_transform(y)

# Split the data into training and testing sets (80% training, 20% testing)

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

# Build a Random Forest classifier and train it on the training data

model = RandomForestClassifier(random\_state=42)

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

# Make predictions on the test data

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

# Decode the numerical labels back to fruit names

y\_test = label\_encoder.inverse\_transform(y\_test)

y\_pred = label\_encoder.inverse\_transform(y\_pred)

# Evaluate the model's accuracy and provide a classification report

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

print("Accuracy:", accuracy)

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

print("Confusion Matrix:\n", confusion)

classification\_rep = classification\_report(y\_test, y\_pred)

print("Classification Report:\n", classification\_rep)