import pandas as pd

from sklearn.naive\_bayes import GaussianNB

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

from sklearn.metrics import accuracy\_score

# Step 1: Create the dataset

# Creating a DataFrame for the provided dataset

data = {

'Weather': ['Sunny', 'Sunny', 'Sunny', 'Overcast', 'Overcast', 'Overcast', 'Rainy', 'Rainy', 'Rainy'],

'Temperature': ['Hot', 'Mild', 'Cool', 'Hot', 'Mild', 'Cool', 'Hot', 'Mild', 'Cool'],

'Play': ['No', 'No', 'Yes', 'Yes', 'Yes', 'Yes', 'No', 'No', 'Yes']

}

df = pd.DataFrame(data)

# Step 2: Encoding categorical variables (Weather, Temperature, Play)

# Encoding 'Play' as 1 (Yes) and 0 (No)

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

# Encoding 'Weather' and 'Temperature' as integers

df['Weather'] = df['Weather'].map({'Sunny': 0, 'Overcast': 1, 'Rainy': 2})

df['Temperature'] = df['Temperature'].map({'Hot': 0, 'Mild': 1, 'Cool': 2})

# Step 3: Define features and target

X = df[['Weather', 'Temperature']] # Features: Weather and Temperature

y = df['Play'] # Target: Play (Yes/No)

# Step 4: Train a Naive Bayes classifier

model = GaussianNB()

model.fit(X, y)

# Step 5: Make a prediction for the scenario where Weather is Overcast and Temperature is Mild

prediction = model.predict([[1, 1]]) # Overcast (1) and Mild (1)

# Step 6: Output the prediction

if prediction == 1:

print("Prediction: The players can play.")

else:

print("Prediction: The players cannot play.")

# Step 7: Evaluate the model (optional)

y\_pred = model.predict(X)

print("Accuracy of the model: ", accuracy\_score(y, y\_pred))