NMDA 2024 Assignment-3

Abdullah Al Asif

2022521460130

Code and output screenshort:

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import CategoricalNB
from sklearn.preprocessing import LabelEncoder
import numpy as np

# Step 1: Define the dataset based on the problem description
# Replace this with the actual data from the assignment.
data = {
    "Humidity": ["High", "Low", "High", "High", "Low"],
    "Cloud_Cover": ["Cloudy", "Sunny", "Sunny", "Cloudy", "Cloudy"],
    "Wind_Strength": ["Medium", "Strong", "Weak", "Medium", "Strong"],
    "Sky_Condition": ["Bright", "Dull", "Bright", "Dull", "Bright"],
    "Rain_Status": ["Rain", "No Rain", "Rain", "Rain", "No Rain"]
}
# Step 2: Convert the data into a DataFrame
df = pd.DataFrame(data)
# Step 3: Encode categorical variables
# Use a separate LabelEncoder for each column
encoders = {}
for column in df.columns:
    encoders[column] = LabelEncoder()
    df[column] = encoders[column].fit_transform(df[column])
# Step 4: Split features and target
X = df.drop("Rain_Status", axis=1)
y = df["Rain_Status"]

# Step 5: Split 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)

# Step 6: Train a Naive Bayes classifier
model = CategoricalNB()
```

```python
model.fit(X_train, y_train)
# Step 7: Define the input conditions for prediction
# High Humidity, Cloudy Skies, Medium Wind, Bright Sky
input_conditions = pd.DataFrame([{
    "Humidity": encoders["Humidity"].transform(["High"])[0],
    "Cloud_Cover": encoders["Cloud_Cover"].transform(["Cloudy"])[0],
    "Wind_Strength": encoders["Wind_Strength"].transform(["Medium"])[0],
    "Sky_Condition": encoders["Sky_Condition"].transform(["Bright"])[0]
}])
# Step 8: Predict the likelihood of rain
predicted_class = model.predict(input_conditions)[0]
predicted_prob = model.predict_proba(input_conditions)[0]

# Step 9: Decode the results back to labels
rain_status =
encoders["Rain_Status"].inverse_transform([predicted_class])[0]
rain_likelihood =
predicted_prob[encoders["Rain_Status"].transform(["Rain"])[0]] * 100
no_rain_likelihood = predicted_prob[encoders["Rain_Status"].transform(["No
Rain"])[0]] * 100

# Step 10: Display the results in a detailed format
print("Prediction Details:")
print(f"Conditions Given:")
print(f"- Humidity: High")
print(f"- Cloud Cover: Cloudy")
print(f"- Wind Strength: Medium")
print(f"- Sky Condition: Bright")
print("\nResults:")
print(f"- Predicted Rain Status: {rain_status}")
print(f"- Likelihood of Rain: {rain_likelihood:.2f}%")
print(f"- Likelihood of No Rain: {no_rain_likelihood:.2f}%")
```

```
PS C:\Web Development\5th Semester\NMDA> python -u "c:\Web Development\5th Semester\NMDA\labeledDataRain.py"
Prediction Details:
Conditions Given:
- Humidity: High
- Cloud Cover: Cloudy
- Wind Strength: Medium
- Sky Condition: Bright

Results:
- Predicted Rain Status: Rain
- Likelihood of Rain: 92.10%
- Likelihood of No Rain: 7.90%
PS C:\Web Development\5th Semester\NMDA>
```

This Python script predicts the likelihood of rain given specific weather conditions using a Naive Bayes Classifier. It processes categorical weather data and determines the probabilities of rain and no rain under specified conditions.

Problem Description

The task is to predict the likelihood of rain given:

High Humidity

Cloudy Skies

Medium Wind

Bright Sky

The classifier uses labeled weather data with features such as humidity, cloud cover, wind strength, and sky condition to make the prediction.

Implementation Steps

1. Dataset Preparation: The dataset is represented as a table where each feature (e.g., humidity, cloud cover) and the target variable (rain status) are categorical.
2. Encoding: Each categorical feature is converted into numerical values using LabelEncoder to make the data compatible with the Naive Bayes classifier.
3. Training: The data is split into training and testing sets, and the CategoricalNB model is trained on the training data.
4. Prediction: Given the conditions (High Humidity, Cloudy Skies, Medium Wind, Bright Sky), the model predicts:

   The most probable rain status.

   The likelihood percentages for both rain and no rain.

5. Output:

   The script displays:

   Input conditions in human-readable format.

   Predicted rain status.

   Likelihood of rain and no rain.