LAB NO: 09

NAIVE BAYES CLASSIFIER

LAB TASK

Q1. Load Iris dataset.

```
[1]
✓ 1s
#Load Iris dataset
from sklearn.datasets import load_iris
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report
```

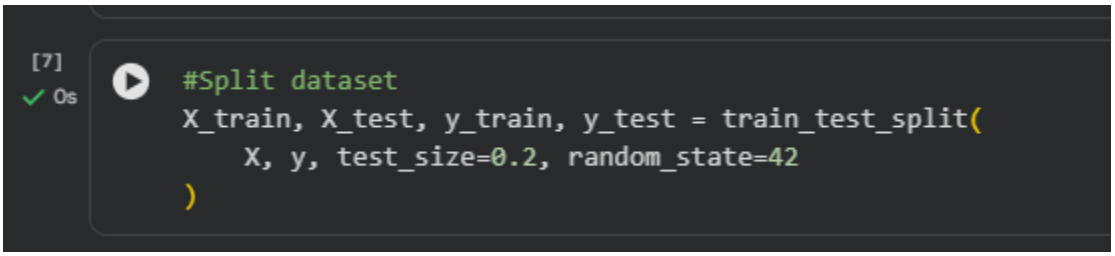
```
[5]
✓ 0s
#Load dataset
iris = load_iris()
X = iris.data
y = iris.target
df = pd.DataFrame(X, columns=iris.feature_names)
df["target"] = y
```

Q2. Apply data preprocessing (handle missing values, encode categorical data if needed).

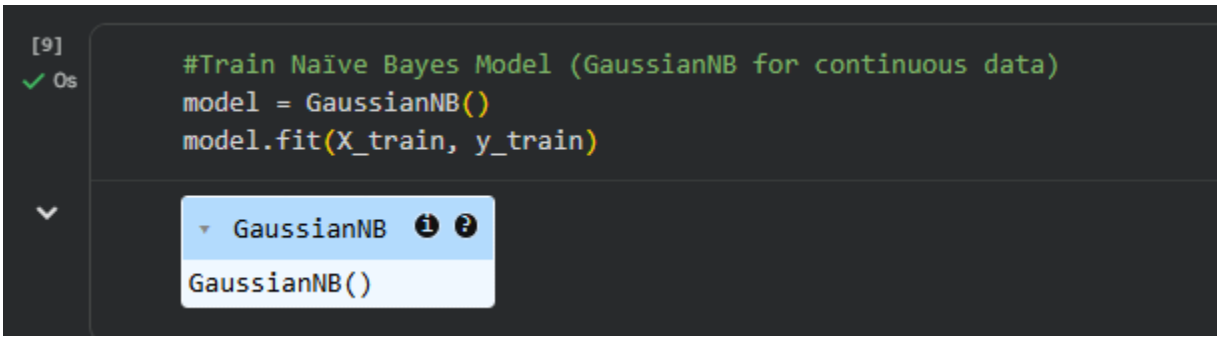
```
[6]
✓ 0s
#Data preprocessing

# Check missing values
print(df.isnull().sum())
```

```
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
target               0
dtype: int64
```

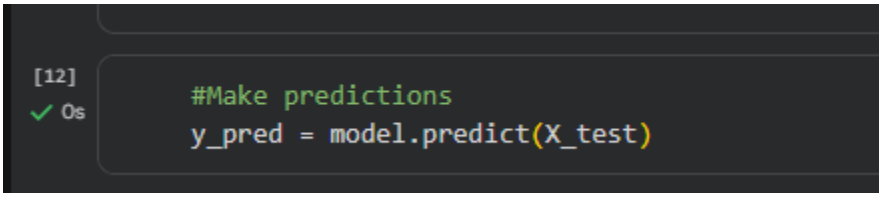
Q3. Split the dataset into training and testing sets.



Q4. Train the Naïve Bayes Model using Gaussian Naïve Bayes since features are continuous.



Q5. Make predictions on the test set.



Q6. Evaluate performance.

```
[11]
✓ 0s
#Evaluate performance
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 1.0

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

Q7. Test the model on new input/unseen data.

```
[14]
✓ 0s
#Test model on new/unseen input
new_data = [[5.1, 3.5, 1.4, 0.2]]  # Example measurement
prediction = model.predict(new_data)

print("\nPredicted class for new input:", iris.target_names[prediction][0])
```

```
Predicted class for new input: setosa
```