# six-tcod74

April 20, 2024

Assignment 6 TCOD74 - Bendre Anushka A.

```python
[17]: import pandas as pd
      from sklearn.model_selection import train_test_split
      from sklearn.naive_bayes import GaussianNB
      from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,␣
       ↪recall_score
```

```python
[18]: # Load the dataset
      data = pd.read_csv('iris.data.csv')
```

```python
[19]: # Display the first few rows of the dataset
      print("First few rows of the dataset:")
      print(data.head())
```

```
First few rows of the dataset:
   5.1  3.5  1.4  0.2  Iris-setosa
0  4.9  3.0  1.4  0.2  Iris-setosa
1  4.7  3.2  1.3  0.2  Iris-setosa
2  4.6  3.1  1.5  0.2  Iris-setosa
3  5.0  3.6  1.4  0.2  Iris-setosa
4  5.4  3.9  1.7  0.4  Iris-setosa
```

```python
[20]: # Separate features and target variable
      X = data.iloc[:, :-1].values
      y = data.iloc[:, -1].values
```

```python
[21]: # Splitting the dataset into the Training set and Test set
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,␣
       ↪random_state=0)
```

```python
[22]: # Naïve Bayes classifier
      classifier = GaussianNB()
      classifier.fit(X_train, y_train)
```

```
[22]: GaussianNB()
```

```
[23]: # Predicting the Test set results
      y_pred = classifier.predict(X_test)
```

```
[24]: # Compute Confusion Matrix
      conf_matrix = confusion_matrix(y_test, y_pred)
      print("\nConfusion Matrix:")
      print(conf_matrix)
```

```
Confusion Matrix:
[[14  0  0]
 [ 0 13  1]
 [ 0  4  6]]
```

```
[25]: # Calculate TP, FP, TN, FN
      TP = conf_matrix.diagonal()
      FP = conf_matrix.sum(axis=0) - TP
      FN = conf_matrix.sum(axis=1) - TP
      TN = conf_matrix.sum() - (TP + FP + FN)
```

```
[26]: # Compute Accuracy
      accuracy = accuracy_score(y_test, y_pred)
```

```
[27]: # Compute Error Rate
      error_rate = 1 - accuracy
```

```
[28]: # Compute Precision
      precision = precision_score(y_test, y_pred, average='macro')
```

```
[29]: # Compute Recall
      recall = recall_score(y_test, y_pred, average='macro')
```

```
[30]: # Print Metrics
      print("\nAccuracy:", accuracy)
      print("Error Rate:", error_rate)
      print("Precision:", precision)
      print("Recall:", recall)
```

```
Accuracy: 0.868421052631579
Error Rate: 0.13157894736842102
Precision: 0.8739495798319328
Recall: 0.8428571428571429
```

```
[ ]:
```