practical6

May 4, 2024

```python
[5]: import numpy as np
     import pandas as pd
     from sklearn.model_selection import train_test_split
     from sklearn.metrics import confusion_matrix, precision_score, recall_score,␣
      ↪accuracy_score, classification_report
     from sklearn.naive_bayes import GaussianNB
     import warnings
     warnings.filterwarnings('ignore')
```

```python
[6]: data = pd.read_csv('Iris.csv')
     data
```

```
[6]:       Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
     0      1            5.1           3.5            1.4           0.2
     1      2            4.9           3.0            1.4           0.2
     2      3            4.7           3.2            1.3           0.2
     3      4            4.6           3.1            1.5           0.2
     4      5            5.0           3.6            1.4           0.2
     ..   ...            ...           ...            ...           ...
     145  146            6.7           3.0            5.2           2.3
     146  147            6.3           2.5            5.0           1.9
     147  148            6.5           3.0            5.2           2.0
     148  149            6.2           3.4            5.4           2.3
     149  150            5.9           3.0            5.1           1.8

                 Species
     0       Iris-setosa
     1       Iris-setosa
     2       Iris-setosa
     3       Iris-setosa
     4       Iris-setosa
     ..              ...
     145  Iris-virginica
     146  Iris-virginica
     147  Iris-virginica
     148  Iris-virginica
     149  Iris-virginica
```

```
[150 rows x 6 columns]
```

[4]: `data.columns`

[4]: Index(['sepal.length', 'sepal.width', 'petal.length', 'petal.width',
           'variety'],
          dtype='object')

[8]:
```python
X = data.drop('Species', axis=1)
y = data['Species']
```

[9]:
```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
     random_state=42)
```

[10]:
```python
nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)
y_pred = nb_classifier.predict(X_test)
```

[11]:
```python
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print("Confusion Matrix:")
print(conf_matrix)
print("\nClassification Report:")
print(class_report)
```

```
Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]

Classification Report:
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00         9
 Iris-virginica       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
```