assignment-6

April 13, 2024

[1]:
```
#imports
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
import warnings
warnings.filterwarnings('ignore')

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, classification_report
```

[2]:
```
iris = load_iris()
data = pd.DataFrame(iris.data, columns=iris.feature_names)
data['target'] = iris.target
data.head()
```

[2]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

[3]:
```
data.sample(5)
```

[3]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	\
88	5.6	3.0	4.1	1.3	

```
72                6.3               2.5                4.9               1.5
145               6.7               3.0                5.2               2.3
137               6.4               3.1                5.5               1.8
111               6.4               2.7                5.3               1.9
```

```
     target
88        1
72        1
145       2
137       2
111       2
```

[4]:
```
set(iris.target), iris.target_names
```

[4]:
```
({0, 1, 2}, array(['setosa', 'versicolor', 'virginica'], dtype='<U10'))
```

target : target_name 0 : setosa 1 : versicolor 2 : virginica

[6]:
```
X_train, X_test, y_train, y_test = train_test_split(data.drop('target',␣
↪axis=1), data['target'], test_size=0.2, random_state=42)
```

0.0.1 Model

[7]:
```
model = GaussianNB()
model.fit(X_train, y_train)
```

[7]:
```
GaussianNB()
```

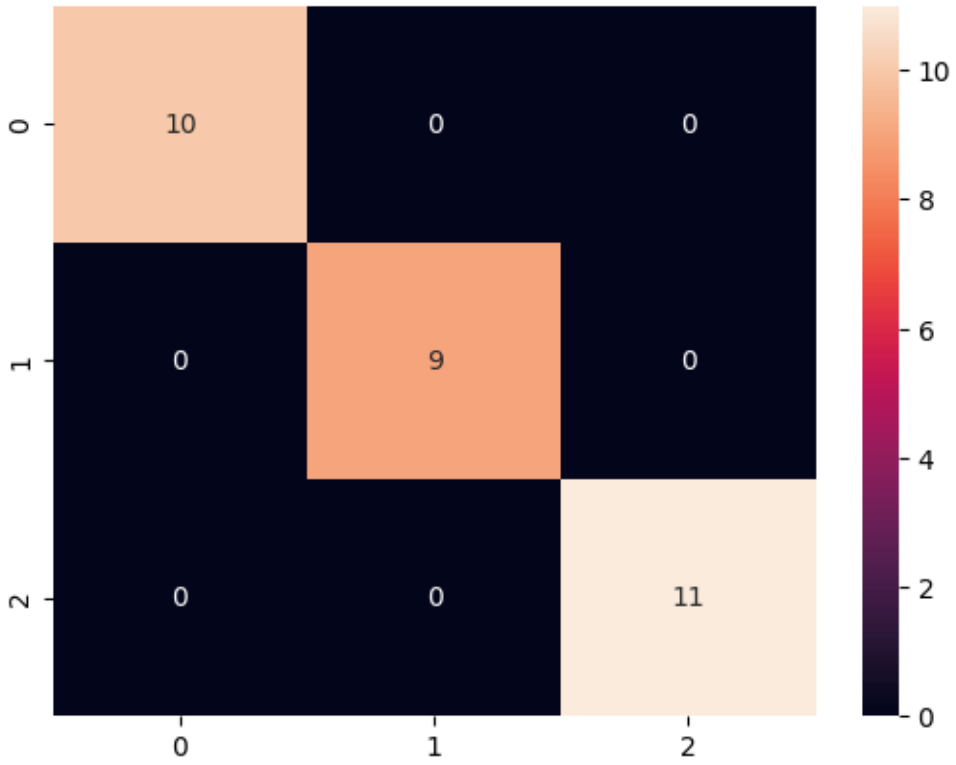
0.0.2 Prediction

[8]:
```
y_pred = model.predict(X_test)
y_pred
```

[8]:
```
array([1, 0, 2, 1, 1, 0, 1, 2, 1, 1, 2, 0, 0, 0, 0, 1, 2, 1, 1, 2, 0, 2,
       0, 2, 2, 2, 2, 2, 0, 0])
```

0.0.3 Evaluation

[9]:
```
sns.heatmap(confusion_matrix(y_test, y_pred), annot = True);
```



[10]:
```
print(classification_report(y_test, y_pred))
```

```
              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
```