untitled12

July 25, 2023

1 import libraries

[ ]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

2 Load the dataset

[ ]:
```
df = sns.load_dataset("iris")
df.head()
```

[ ]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

3 Select input and output

[ ]:
```
X = df.iloc[:,:-1]
Y = df.iloc[:,-1:]
```

4 Training and fitting the modle

[ ]:
```
from sklearn.naive_bayes import GaussianNB
model=GaussianNB().fit(X,Y)
model
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

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

5 Train trst split and checking accuracy of modle

[ ]:
```
from scipy.sparse import random
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.
↪2,random_state=0)
```

6 Training the model on training data

[ ]:
```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB().fit(X_train,Y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

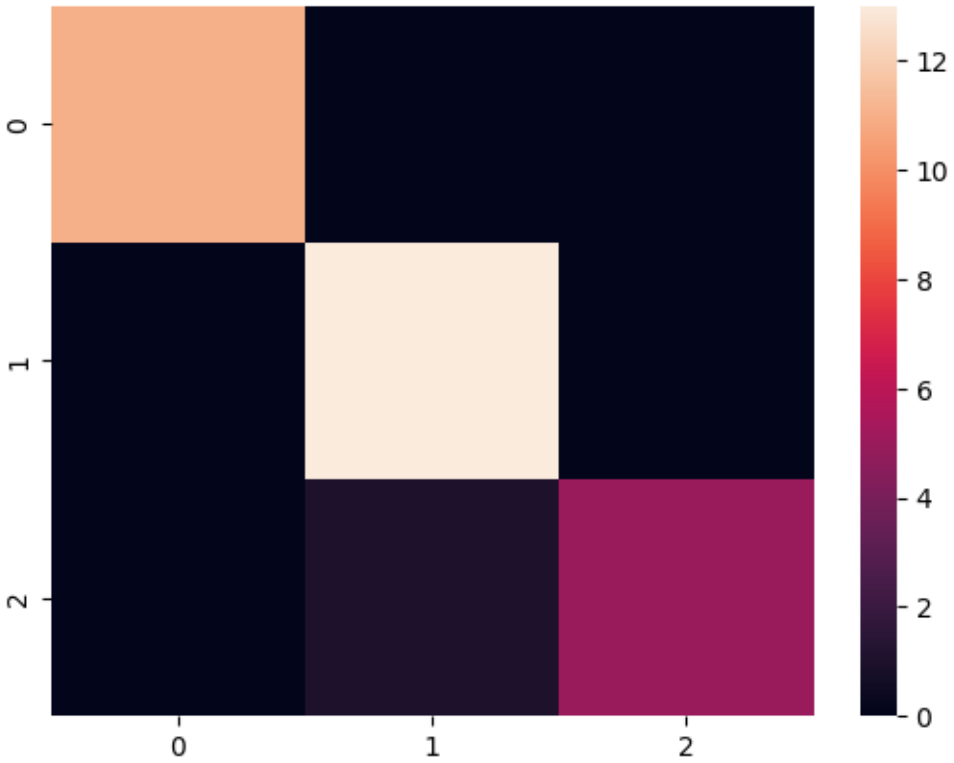
7 Making prediction on testing data

[ ]:
```
Y_pred=model.predict(X_test)
Y_pred
```

[ ]:
```
array(['virginica', 'versicolor', 'setosa', 'virginica', 'setosa',
       'virginica', 'setosa', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'versicolor', 'versicolor', 'versicolor',
       'versicolor', 'setosa', 'versicolor', 'versicolor', 'setosa',
       'setosa', 'virginica', 'versicolor', 'setosa', 'setosa',
       'virginica', 'setosa', 'setosa', 'versicolor', 'versicolor',
       'setosa'], dtype='<U10')
```

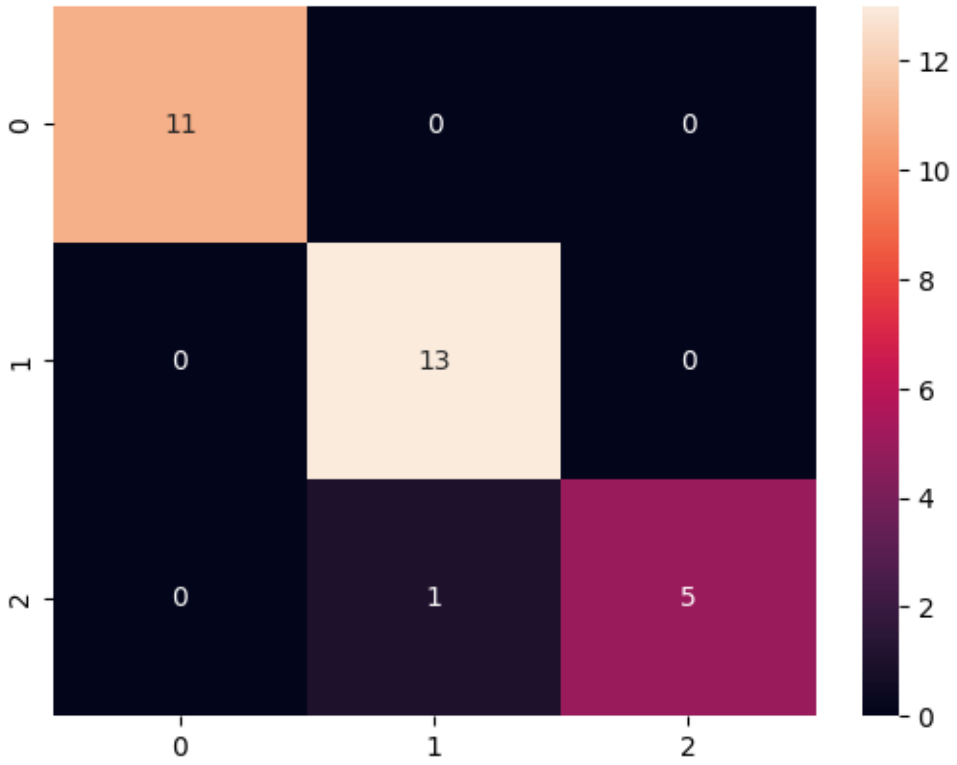
[ ]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test, Y_pred)
sns.heatmap(cm)
```

[ ]:
```
<Axes: >
```



[ ]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test, Y_pred)
sns.heatmap(cm,annot=True)
```

[ ]: <Axes: >



[ ]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test, Y_pred)
#sns.heatmap(cm,annot=True)
cm
```

[ ]:
```
array([[11,  0,  0],
       [ 0, 13,  0],
       [ 0,  1,  5]])
```

[ ]:
```

```