*#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

In [48]:

iris **=** load\_iris()

data **=** pd**.**DataFrame(iris**.**data, columns**=**iris**.**feature\_names)

data['target'] **=** iris**.**target

data**.**head()

Out[48]:

|  | **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 |

In [49]:

data**.**sample(5)

Out[49]:

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** |
| --- | --- | --- | --- | --- | --- |
| **10** | 5.4 | 3.7 | 1.5 | 0.2 | 0 |
| **19** | 5.1 | 3.8 | 1.5 | 0.3 | 0 |
| **100** | 6.3 | 3.3 | 6.0 | 2.5 | 2 |
| **79** | 5.7 | 2.6 | 3.5 | 1.0 | 1 |
| **69** | 5.6 | 2.5 | 3.9 | 1.1 | 1 |

In [36]:

set(iris**.**target), iris**.**target\_names

Out[36]:

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

target : target\_name  
0 : setosa  
1 : versicolor  
2 : virginica

In [50]:

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(df**.**drop('target', axis**=**1), data['target'], test\_size**=**0.2, random\_state**=**42)

**Model**

In [51]:

model **=** GaussianNB()

model**.**fit(X\_train, y\_train)

Out[51]:

GaussianNB()

**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.  
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

**Prediction**

In [52]:

y\_pred **=** model**.**predict(X\_test)

y\_pred

Out[52]:

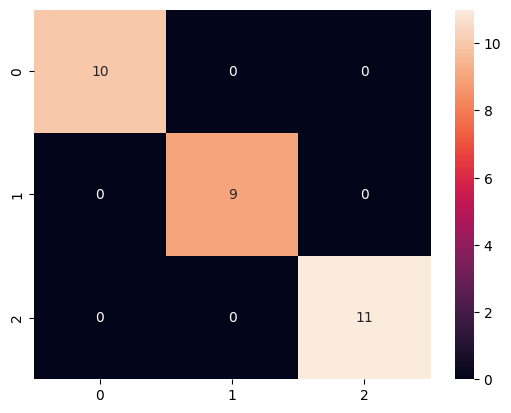
array([1, 0, 2, 1, 1, 0, 1, 2, 1, 1, 2, 0, 0, 0, 0, 1, 2, 1, 1, 2, 0, 2,

0, 2, 2, 2, 2, 2, 0, 0])

**Evaluation**

In [53]:

sns**.**heatmap(confusion\_matrix(y\_test, y\_pred), annot **=** **True**);



In [54]:

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