In [ ]:

```
'''

sklearn library contains so many modules and built in ready made features.

load_iris dataset is already preprocessed and transformed into standard form and ready

no need to use pandas or numpy calculations
or standardscaler to transform data

we are all set to go directly to our practical.

Naive Bayes classifier uses conditional probability

P(A|B)    = ( P(B|A) x     P(A)    ) / P(B)

Posterior = ( Prior  x Likelihood ) / Evidence

'''
```

In [1]:

```
from sklearn.datasets import load_iris

X,y = load_iris(return_X_y=True) # splitting input and output
```

In [34]:

```
from sklearn.model_selection import train_test_split

xtrain,xtest,ytrain,ytest = train_test_split(X,y,test_size=0.3,random_state=0)
# splitting training and testing data into 70% for training and 30% for testing
```

In [35]:

```
from sklearn.naive_bayes import GaussianNB # for continous values in gaussian distribut
                                           # i.e normal distribution
clf = GaussianNB()

clf.fit(xtrain,ytrain)
```

Out[35]:

```
GaussianNB()
```

In [36]:

```
ypred = clf.predict(xtest)

ypred
```

Out[36]:

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

In [37]:

```
ypred = clf.predict(xtest)
```

In [38]:

```
ypred
```

Out[38]:

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

In [39]:

```
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(ytest,ypred)
cm
```

Out[39]:

```
array([[16,  0,  0],
       [ 0, 18,  0],
       [ 0,  0, 11]], dtype=int64)
```

In [40]:

```
from sklearn.metrics import accuracy_score

ac = accuracy_score(ytest,ypred)
ac = ac*100
ac
```

Out[40]:

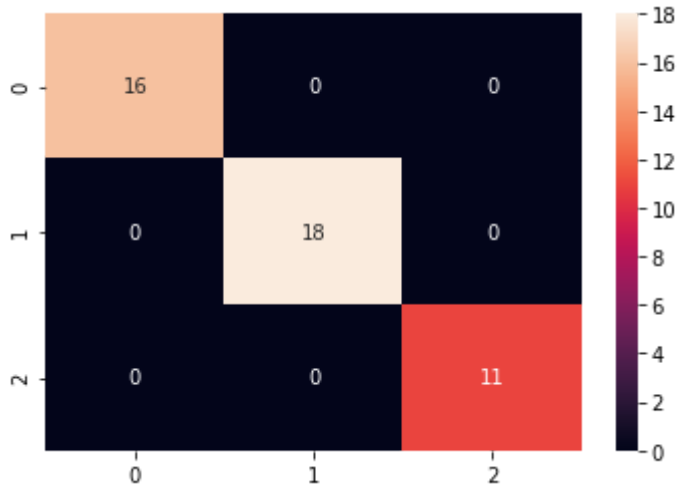
```
100.0
```

In [41]:

```
import seaborn as sns
sns.heatmap(cm,annot=True)
```

Out[41]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1fcc48c9310>
```



In [ ]:

```
1
```