

In [21]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import
mean_squared_error, accuracy_score, confusion_matrix, classification_report
from sklearn.datasets import load_wine
```

In [4]:

```
data = load_wine()
```

In [6]:

```
df = pd.DataFrame(data = data.data, columns = data.feature_names)
```

In [11]:

```
df['target'] = data.target
```

In [14]:

```
x_train, x_test, y_train, y_test = train_test_split(data.data, data.target, test_size =
0.3, stratify = df.target)
```

In [15]:

```
gnb = GaussianNB()
gnb.fit(x_train, y_train)
```

Out[15]:

```
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.**
GaussianNB

```
GaussianNB()
```

In [18]:

```
predicted = gnb.predict(x_test)
```

In [19]:

```
predicted
```

Out[19]:

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

In [24]:

```
confusionM = confusion_matrix(y_test,predicted)
accuracy = accuracy_score(y_test,predicted)
total = classification_report(y_test,predicted)
```

In [25]:

```
confusionM
```

Out[25]:

```
array([[17,  1,  0],
       [ 0, 21,  0],
       [ 0,  0, 15]], dtype=int64)
```

In [27]:

```
accuracy
```

Out[27]:

```
0.9814814814814815
```

In [28]:

```
total
```

Out[28]:

```
'          precision    recall  f1-score   support\n\n 0.94      0.97      18\n 1      0.95      1.00      0.98      21\n 2      1.00      1.00      1.00      15\n 54\n 54\n 0.98      0.98      54\n'
```

	precision	recall	f1-score	support	accuracy	
0	1.00					
1	0.98	1.00	0.99	21		
2	1.00	1.00	1.00	15		
54	0.98	0.98	0.98	54		
macro avg	0.98	0.98	0.98			
weighted avg	0.98	0.98	0.98			0.98

In []: