In [2]:

```
from sklearn.datasets import load_wine
wine = load_wine()
dir(wine)
```

Out[2]:

```
['DESCR', 'data', 'feature_names', 'target', 'target_names']
```

In [5]:

```
wine.feature_names
```

Out[5]:

```
['alcohol',
 'malic_acid',
 'ash',
 'alcalinity_of_ash',
 'magnesium',
 'total_phenols',
 'flavanoids',
 'nonflavanoid_phenols',
 'proanthocyanins',
 'color_intensity',
 'hue',
 'od280/od315_of_diluted_wines',
 'proline']
```

In [4]:

```
wine.data
```

Out[4]:

```
array([[1.423e+01, 1.710e+00, 2.430e+00, ..., 1.040e+00, 3.920e+00,
        1.065e+03],
       [1.320e+01, 1.780e+00, 2.140e+00, ..., 1.050e+00, 3.400e+00,
        1.050e+03],
       [1.316e+01, 2.360e+00, 2.670e+00, ..., 1.030e+00, 3.170e+00,
        1.185e+03],
       ...,
       [1.327e+01, 4.280e+00, 2.260e+00, ..., 5.900e-01, 1.560e+00,
        8.350e+02],
       [1.317e+01, 2.590e+00, 2.370e+00, ..., 6.000e-01, 1.620e+00,
        8.400e+02],
       [1.413e+01, 4.100e+00, 2.740e+00, ..., 6.100e-01, 1.600e+00,
        5.600e+02]])
```

In [11]:

```
from sklearn.naive_bayes import MultinomialNB, GaussianNB
from sklearn.model_selection import KFold
```

In [39]:

```
kf = KFold(n_splits = 10) # performing 10 fold cross validation
```

In [18]:

```
def get_score(model,xtrain,x_test,y_train,y_test):
    model.fit(x_train, y_train)
    return model.score(x_test, y_test)
```

In [40]:

```
multiNB_score = []
gaussianNB_score = []
```

```
for train_index, test_index in kf.split(wine.data):
    x_train,x_test,y_train,y_test = wine.data[train_index], wine.data[test_index], wine.
target[train_index], wine.target[test_index]
    multiNB_score.append(get_score(MultinomialNB(),xtrain,x_test,y_train,y_test))
    gaussianNB_score.append(get_score(GaussianNB(),xtrain,x_test,y_train,y_test))
```

In [41]:

```
import numpy as np
print(np.mean(gaussianNB_score))
print(np.mean(multiNB_score))

# we can see that Gaussian Naive Bayes is performing better than Multinomial Naive Bayes
```

```
0.961111111111111
0.826797385620915
```

In [46]:

```
# using Gaussian Naive Bayes to make predictions on test data
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(wine.data, wine.target, test_size = 0.2
)
```

In [47]:

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

Out[47]:

```
GaussianNB(priors=None, var_smoothing=1e-09)
```

In [48]:

```
gnb.predict(x_test)
```

Out[48]:

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

In [49]:

```
gnb.score(x_test, y_test)
```

Out[49]:

```
0.9722222222222222
```