In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
from sklearn.datasets import load_wine
```

In [4]:
```
wine = load_wine()
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 [5]:
```
wine.target
```

Out[5]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2])
```

In [6]:
```
wine.target_names
```

Out[6]:
```
array(['class_0', 'class_1', 'class_2'], dtype='<U7')
```

In [10]:
```
from sklearn.model_selection import train_test_split

xtrain,xtest, ytrain, ytest = train_test_split(wine.data,wine.target, test_size=0.3, random_state=1)
```

In [11]:
```
from sklearn.svm import SVC

svm = SVC()
```

In [12]:
```
svm.fit(xtrain, ytrain)
print(svm.score(xtrain,ytrain))
print(svm.score(xtest,ytest))
```

```
0.7258064516129032
0.6851851851851852
```

In [13]:

```
kernels = ['linear', 'poly', 'rbf', 'sigmoid']

for kernel in kernels:
    print(kernel)
    svm = SVC(kernel=kernel)
    svm.fit(xtrain, ytrain)
    ypred = svm.predict(xtest)
    print(svm.score(xtrain,ytrain))
    print(svm.score(xtest,ytest))
```

```
linear
1.0
0.9629629629629629
poly
0.7096774193548387
0.6111111111111112
rbf
0.7258064516129032
0.6851851851851852
sigmoid
0.41935483870967744
0.35185185185185186
```

In [14]:

```
svm = SVC(kernel='linear')
svm.fit(xtrain, ytrain)
ypred = svm.predict(xtest)
print(svm.score(xtrain,ytrain))
print(svm.score(xtest,ytest))
```

```
1.0
0.9629629629629629
```

In [32]:

```
for i in range(23,24):
    xtrain, xtest, ytrain, ytest = train_test_split(wine.data,wine.target, test_size=0.3, random_state=i)
    print('Random State', i)
    svm = SVC(kernel='linear')
    svm.fit(xtrain, ytrain)
    ypred = svm.predict(xtest)
    if svm.score(xtrain,ytrain) > 0.99 and svm.score(xtest,ytest) > 0.98:
        print('\n\tTraining Score',svm.score(xtrain,ytrain))
        print('\tTesting Score',svm.score(xtest,ytest))
        break
```

```
Random State 23

        Training Score 0.9919354838709677
        Testing Score 1.0
```

In [20]:

```
0.9922480620155039 - 1.0
```

Out[20]:

```
-0.007751937984496138
```

In [33]:

```
xtrain, xtest, ytrain, ytest = train_test_split(wine.data,wine.target, test_size=0.3, random_state=23)
svm = SVC(kernel='linear')
svm.fit(xtrain, ytrain)
ypred = svm.predict(xtest)
print(svm.score(xtrain,ytrain))
print(svm.score(xtest,ytest))
```

```
0.9919354838709677
1.0
```