In [1]:
```
# Author : Amir Shokri
# github link : https://github.com/amirshnll/Wine
# dataset link : http://archive.ics.uci.edu/ml/datasets/Wine
# email : amirsh.nll@gmail.com
```

In [34]:
```
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import datasets
from pandas.plotting import scatter_matrix
from sklearn.model_selection import train_test_split
```

In [35]:
```
col_names = ['class', 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Ma
gnesium', 'Total phenols', 'Flavanoids', ' Nonflavanoid phenols','Proanthocyan
ins','Color intensity','Hue','OD280/OD315 of diluted wines','Proline']
wine =pd.read_csv("wine.csv",header=None, names=col_names)
```

In [36]:
```
wine.head()
```

Out[36]:

	class	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proan
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	

In [32]:
```
inputs =wine.drop('class',axis='columns')
target = wine['class']
```

In [38]:
```
inputs
```

Out[38]:

	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proanthocy
0	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	
...	...	...	...	...	...	...	...	...	
173	13.71	5.65	2.45	20.5	95	1.68	0.61	0.52	
174	13.40	3.91	2.48	23.0	102	1.80	0.75	0.43	
175	13.27	4.28	2.26	20.0	120	1.59	0.69	0.43	
176	13.17	2.59	2.37	20.0	120	1.65	0.68	0.53	
177	14.13	4.10	2.74	24.5	96	2.05	0.76	0.56	

178 rows × 13 columns

In [39]:
```
input_train,input_test,target_train,target_test=train_test_split(inputs,target
,test_size=0.3,random_state=1)
```

In [40]:
```
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(hidden_layer_sizes=(6,7), max_iter=1000)
mlp.fit(input_train, target_train)
```

Out[40]:
```
MLPClassifier(hidden_layer_sizes=(6, 7), max_iter=1000)
```

In [31]:
```
from sklearn.metrics import accuracy_score
predictions_train = mlp.predict(input_train)

print("accuracy for train data: ", accuracy_score(predictions_train, target_tr
ain))
predictions_test = mlp.predict(input_test)
print(target_test)
print(predictions_test)
print("accuracy for test data: ", accuracy_score(predictions_test, target_test
))
```

```
accuracy for train data:  0.7741935483870968
161    3
117    2
19     1
69     2
53     1
138    3
112    2
14     1
160    3
107    2
11     1
4      1
108    2
42     1
84     2
113    2
152    3
35     1
105    2
31     1
51     1
126    2
130    3
73     2
40     1
162    3
47     1
29     1
16     1
147    3
97     2
159    3
151    3
5      1
120    2
94     2
91     2
81     2
114    2
48     1
54     1
59     2
165    3
39     1
56     1
44     1
78     2
33     1
18     1
58     1
127    2
172    3
148    3
12     1
Name: class, dtype: int64
[2 2 3 2 1 3 2 1 2 2 1 2 2 1 2 2 2 1 2 1 1 2 2 2 2 2 1 1 1 3 2 3 2 1 2 2 2
```

```
 3 2 1 1 2 2 3 1 1 2 1 1 1 2 3 3 1]
accuracy for test data:  0.7777777777777778
```

In [ ]: