CCPS 844 Data Mining (Lab 7) Submit your solution as a pdf file

- Q-1 Select a dataset/datasets of your choice. Apply SVM Classification Evaluate the results
- Q-2 Select a dataset/datasets of your choice. Apply SVM Regression Evaluate the results

In [118]:
```
% matplotlib inline

from IPython.core.display import display, HTML
display(HTML("<style>.container { width:90% !important; }</style
>"))
```

In [119]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
pd.set_option('display.max_columns', 100)
```

In [120]:
```
from sklearn import svm

df = pd.read_csv('winequality-red.csv')
df['overfive'] = (df.quality > 5).astype(int)
```

In [121]:
```
df.sample(5)
```

Out[121]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH
1318	7.5	0.63	0.27	2.0	0.083	17.0	91.0	0.99616	3.26
1147	10.0	0.41	0.45	6.2	0.071	6.0	14.0	0.99702	3.21
200	9.6	0.32	0.47	1.4	0.056	9.0	24.0	0.99695	3.22
1372	8.7	0.78	0.51	1.7	0.415	12.0	66.0	0.99623	3.00
907	6.1	0.56	0.00	2.2	0.079	6.0	9.0	0.99480	3.59

In [122]:
```
X=df.iloc[:,0:3]
y=df.overfive
```

In [123]:
```
clf = svm.SVC()
clf.fit(X, y)
```

Out[123]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kerne
l='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking
=True,
  tol=0.001, verbose=False)
```

In [124]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_s
ize = 0.25, random_state = 0)
```

In [125]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
```

In [126]:
```
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [127]:
```
clf2 = svm.SVC(kernel = 'linear', random_state = 0)
clf2.fit(X_train, y_train)
```

Out[127]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kerne
l='linear',
  max_iter=-1, probability=False, random_state=0, shrinking=Tr
ue,
  tol=0.001, verbose=False)
```

In [128]:
```
y_pred = clf2.predict(X_test)
```

In [129]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)
```

Out[129]:
```
array([[106,  79],
       [ 62, 153]])
```