In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
df = pd.read_csv(r"C:\Users\ABHISHEK\Desktop\winequality-red.csv")
# Exploratory Data Analysis
df.shape
```

Out[2]:

(1599, 12)

In [3]:

```
df.head()
```

Out[3]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcoh
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9

In [8]:

```
# Divide the data into attributes and labels:
X = df.drop('quality', axis=1)
y = df['quality']
X.head()
```

Out[8]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcoh
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9

In [9]:

```
y.tail()
```

Out[9]:

```
1594    5
1595    6
1596    6
1597    5
1598    6
Name: quality, dtype: int64
```

In [10]:

```
# Train Test Split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)
```

In [11]:

```
# Training the Algorithm
from sklearn.svm import SVC
svclassifier = SVC(kernel = 'linear')
svclassifier.fit(X_train, y_train)
```

Out[11]:

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='linear', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
```

In [12]:

```
# Making Predictions
y_pred = svclassifier.predict(X_test)
# Evaluating the Algorithm
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

```
[[  0   0   2   0   0   0]
 [  0   0   7   3   0   0]
 [  0   0 108  29   0   0]
 [  0   0  50  79   0   0]
 [  0   0   3  37   0   0]
 [  0   0   0   2   0   0]]
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         2
           4       0.00      0.00      0.00        10
           5       0.64      0.79      0.70       137
           6       0.53      0.61      0.57       129
           7       0.00      0.00      0.00        40
           8       0.00      0.00      0.00         2

   micro avg       0.58      0.58      0.58       320
   macro avg       0.19      0.23      0.21       320
weighted avg       0.48      0.58      0.53       320


C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:1143: Un
definedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels
with no predicted samples.
  'precision', 'predicted', average, warn_for)
```

2. Kernel SVM

In [14]:

```
# Importing the Dataset
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
# Assign colum names to the dataset
colnames = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']
# Read dataset to pandas dataframe
irisdata = pd.read_csv(url, names=colnames)
# Divide the data into attributes and labels:
X = irisdata.drop('Class', axis=1)
y = irisdata['Class']
# Train Test Split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)
irisdata.head()
```

Out[14]:

	sepal-length	sepal-width	petal-length	petal-width	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

A. Polynomial Kernel

In [16]:

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='poly', degree=4)
svclassifier.fit(X_train, y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: FutureWarning:
The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account b
etter for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this warnin
g.
  "avoid this warning.", FutureWarning)
```

Out[16]:

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=4, gamma='auto_deprecated',
  kernel='poly', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
```

In [17]:

```
# Making Predictions
y_pred = svclassifier.predict(X_test)
# Evaluating the Algorithm
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[ 9  0  0]
 [ 0  8  1]
 [ 0  2 10]]
              precision    recall  f1-score   support

   Iris-setosa       1.00      1.00      1.00         9
Iris-versicolor       0.80      0.89      0.84         9
 Iris-virginica       0.91      0.83      0.87        12

      micro avg       0.90      0.90      0.90        30
      macro avg       0.90      0.91      0.90        30
   weighted avg       0.90      0.90      0.90        30
```

B. Gaussian Kernel

In [18]:

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='rbf')
svclassifier.fit(X_train, y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: FutureWarning:
The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account b
etter for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this warnin
g.
  "avoid this warning.", FutureWarning)
```

Out[18]:

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='rbf', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
```

In [19]:

```
# Prediction and Evaluation
y_pred = svclassifier.predict(X_test)
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[ 9  0  0]
 [ 0  8  1]
 [ 0  0 12]]
              precision    recall  f1-score   support

   Iris-setosa       1.00      1.00      1.00         9
Iris-versicolor       1.00      0.89      0.94         9
 Iris-virginica       0.92      1.00      0.96        12

      micro avg       0.97      0.97      0.97        30
      macro avg       0.97      0.96      0.97        30
   weighted avg       0.97      0.97      0.97        30
```

C. Sigmoid Kernel

In [20]:

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='sigmoid')
svclassifier.fit(X_train, y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: FutureWarning:
The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account b
etter for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this warnin
g.
  "avoid this warning.", FutureWarning)
```

Out[20]:

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='sigmoid', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
```

In [23]:

```
# Prediction and Evaluation
y_pred = svclassifier.predict(X_test)
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[ 4  5  0]
 [ 8  1  0]
 [12  0  0]]
              precision    recall  f1-score   support

   Iris-setosa       0.17      0.44      0.24         9
Iris-versicolor       0.17      0.11      0.13         9
 Iris-virginica       0.00      0.00      0.00        12

      micro avg       0.17      0.17      0.17        30
      macro avg       0.11      0.19      0.13        30
   weighted avg       0.10      0.17      0.11        30


C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:1143: Un
definedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels
with no predicted samples.
  'precision', 'predicted', average, warn_for)
```