In [135]:

```
import numpy as np
import matplotlib.pyplot as plt
```

Importing Dataset 1

In [197]:

```
import pandas as pd
data1=pd.read_csv('d1_new.csv')
data1.head()
```

Out[197]:

	x1	x2	y
0	34.623660	78.024693	0
1	30.286711	43.894998	0
2	35.847409	72.902198	0
3	45.083277	56.316372	0
4	95.861555	38.225278	0

In [198]:

```
#39
#data1.iloc[39]['y']
data1.head()
```

Out[198]:

	x1	x2	y
0	34.623660	78.024693	0
1	30.286711	43.894998	0
2	35.847409	72.902198	0
3	45.083277	56.316372	0
4	95.861555	38.225278	0

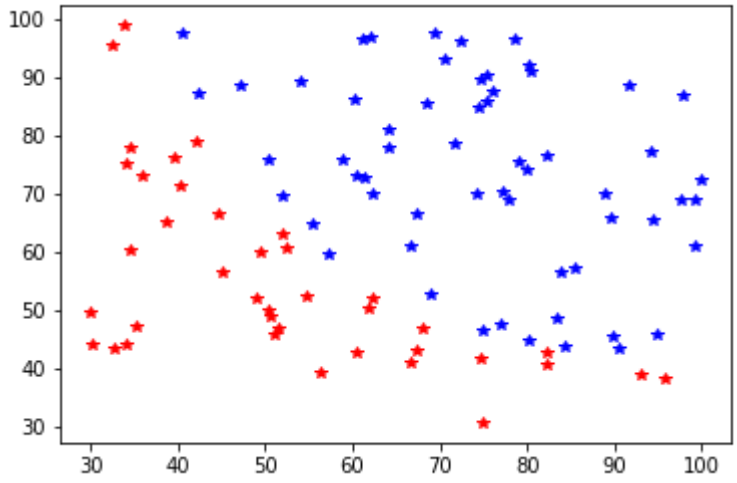
Plotting Data

In [199]:

```
plt.plot(data1['x1'][0:39],data1['x2'][0:39],'*',c='r')
plt.plot(data1['x1'][40:99],data1['x2'][40:99],'*',c='b')
```

Out[199]:

```
[<matplotlib.lines.Line2D at 0x288f29dc518>]
```



In [200]:

```
X = data1.drop('y', axis=1)
```

In [201]:

```
X.shape
```

Out[201]:

```
(100, 2)
```

In [202]:

```
Y=data1['y']
```

In [203]:

```
Y.head()
```

Out[203]:

```
0    0
1    0
2    0
3    0
4    0
Name: y, dtype: int64
```

In [204]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.25)
```

In [205]:

```
X_train.shape
```

Out[205]:

```
(75, 2)
```

In [206]:

```
X_test.shape
```

Out[206]:

```
(25, 2)
```

In [207]:

```
y_train.shape
```

Out[207]:

```
(75,)
```

In [208]:

```
y_test.shape
```

Out[208]:

```
(25,)
```

SVM using Linear Kernel

In [209]:

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

Out[209]:

```
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 [210]:

```
y_pred = svclassifier.predict(X_test)
```

In [211]:

```
y_pred
```

Out[211]:

```
array([0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1,
       1, 1, 1], dtype=int64)
```

In [212]:

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

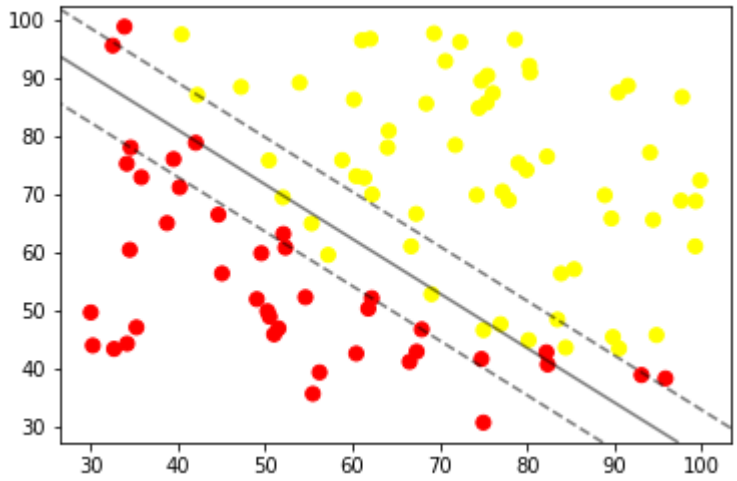
```
[[ 9  2]
 [ 2 12]]
              precision    recall  f1-score   support

           0       0.82      0.82      0.82        11
           1       0.86      0.86      0.86        14

   micro avg       0.84      0.84      0.84        25
   macro avg       0.84      0.84      0.84        25
weighted avg       0.84      0.84      0.84        25
```

In [213]:

```
def plot_svc_decision_function(model, ax=None, plot_support=True):
    """Plot the decision function for a 2D SVC"""
    if ax is None:
        ax = plt.gca()
    xlim = ax.get_xlim()
    ylim = ax.get_ylim()

    # create grid to evaluate model
    x = np.linspace(xlim[0], xlim[1], 30)
    y = np.linspace(ylim[0], ylim[1], 30)
    Y, X = np.meshgrid(y, x)
    xy = np.vstack([X.ravel(), Y.ravel()]).T
    P = model.decision_function(xy).reshape(X.shape)

    # plot decision boundary and margins
    ax.contour(X, Y, P, colors='k',
               levels=[-1, 0, 1], alpha=0.5,
               linestyles=['--', '-', '--'])

    # plot support vectors
    if plot_support:
        ax.scatter(model.support_vectors_[:, 0],
                   model.support_vectors_[:, 1],
                   s=300, linewidth=1, facecolors='none');
    ax.set_xlim(xlim)
    ax.set_ylim(ylim)
```

In [214]:

```
plt.scatter(X['x1'], X['x2'], c=Y, s=50, cmap='autumn')
plot_svc_decision_function(model);
```



In [215]:

```
clf = SVC(kernel='poly', C=1E6)
clf.fit(X, Y)
```

```
C:\Users\Rajat_PC\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: Fut
ureWarning: The default value of gamma will change from 'auto' to 'scale'
in version 0.22 to account better for unscaled features. Set gamma explici
tly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

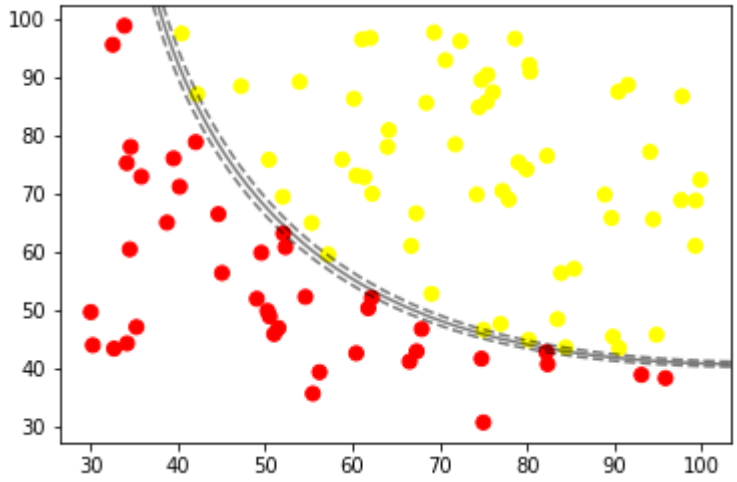
Out[215]:

```
SVC(C=1000000.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='poly', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
```

SVM using Polynomial Kernel

In [216]:

```
plt.scatter(X['x1'], X['x2'], c=Y, s=50, cmap='autumn')
plot_svc_decision_function(clf)
plt.scatter(clf.support_vectors_[:, 0], clf.support_vectors_[:, 1],
            s=300, lw=1, facecolors='none');
```



Loading Dataset 2

In [176]:

```
data2=pd.read_csv('d2.csv')
data2.head()
```

Out[176]:

	x1	x2	y
0	0.051267	0.69956	1
1	-0.092742	0.68494	1
2	-0.213710	0.69225	1
3	-0.375000	0.50219	1
4	-0.513250	0.46564	1

In [177]:

```
data2['y'][117]
```

Out[177]:

```
0
```

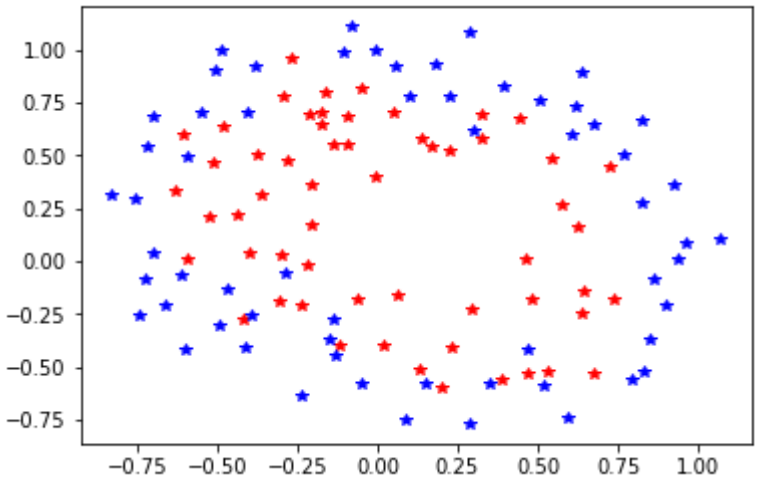
Plotting Data

In [217]:

```
plt.plot(data2['x1'][0:57],data2['x2'][0:57],'*',c='r')
plt.plot(data2['x1'][57:117],data2['x2'][57:117],'*',c='b')
```

Out[217]:

```
[<matplotlib.lines.Line2D at 0x288f2b0ff60>]
```



In [179]:

```
X = data2.drop('y', axis=1)
```

In [180]:

```
X.shape
```

Out[180]:

```
(118, 2)
```

In [181]:

```
Y=data2['y']
```

In [182]:

```
Y.shape
```

Out[182]:

```
(118,)
```

In [183]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.25)
```

In [188]:

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

```
C:\Users\Rajat_PC\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: Fut
ureWarning: The default value of gamma will change from 'auto' to 'scale'
in version 0.22 to account better for unscaled features. Set gamma explici
tly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

Out[188]:

```
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 [189]:

```
y_pred = svclassifier.predict(X_test)
```

In [190]:

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

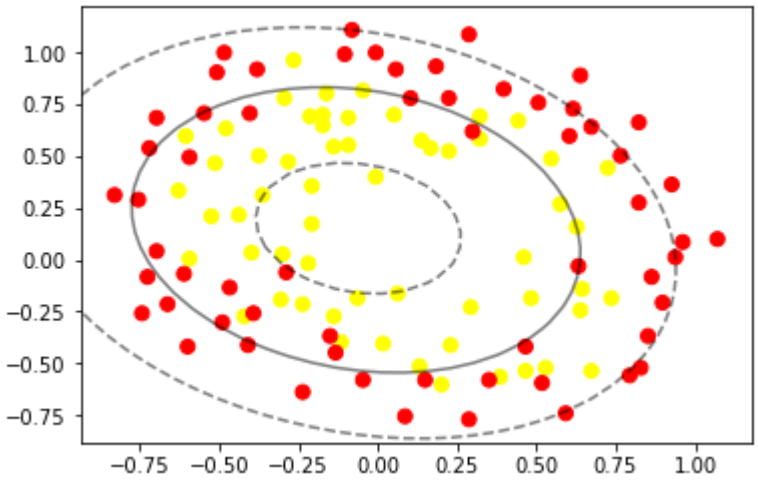
```
[[14  0]
 [16  0]]
              precision    recall  f1-score   support

           0       0.47      1.00      0.64        14
           1       0.00      0.00      0.00        16

   micro avg       0.47      0.47      0.47        30
   macro avg       0.23      0.50      0.32        30
weighted avg       0.22      0.47      0.30        30
```

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

Using RBF kernel

In [193]:

```
plt.scatter(X['x1'], X['x2'], c=Y, s=50, cmap='autumn')
plot_svc_decision_function(model);
```



In [ ]:

```

```