In [6]:

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

In [7]:

```
dataset=pd.read_excel(r'CRDA.xlsx',sheet_name='2017')
```

In [8]:

```
dataset = dataset.drop(['Total(144)'], axis=1)
```

In [9]:

```
print(dataset)
```

```
    S. No State/UT/District  Homicide/Murder(3,4,15,16)  \
0       1        Ahmednagar                         215
1       2             Akola                         101
2       3          Amravati                         168
3       4         Aurangbad                         220
4       5              Beed                         164
5       6          Bhandara                          39
6       7          Buldhana                         101
7       8        Chandrapur                          74
8       9             Dhule                          96
9      10        Gadchiroli                         106
10     11            Gondia                          59
11     12           Hingoli                          81
12     13           Jalgaon                         121
13     14             Jalna                         121
14     15          Kolhapur                         153
15     16             Latur                          93
16     17            Mumbai                         345
17     18             Nagpur                         261
18     19             Nanded                         120
```

In [10]:

```
X = dataset.iloc[:, 2:18].values
y = dataset.iloc[:, 18:19].values
y2 = dataset.iloc[:, 19:20].values
```

In [11]:

```
1 print(X)
```

```
[[2.1500000e+02 7.1500000e+02 8.8600000e+02 4.1000000e+02 3.0300000e+02
  0.0000000e+00 8.5000000e+01 5.8800000e+02 2.8490000e+03 1.7700000e+02
  2.4530000e+03 2.3500000e+02 4.5431590e+06 9.3900000e+02 7.9050000e-01
  2.6600000e+02]
 [1.0100000e+02 1.4300000e+02 1.1610000e+03 1.4400000e+02 9.6000000e+01
  0.0000000e+00 3.1000000e+01 1.0800000e+02 8.5400000e+02 7.2000000e+01
  2.1060000e+03 2.1000000e+02 1.8139060e+06 9.4600000e+02 8.8050000e-01
  3.2000000e+02]
 [1.6800000e+02 2.7300000e+02 1.5530000e+03 3.4500000e+02 2.1500000e+02
  2.0000000e+00 7.9000000e+01 1.7400000e+02 2.2780000e+03 2.6800000e+02
  3.7650000e+03 2.4500000e+02 2.8884450e+06 9.5100000e+02 8.7380000e-01
  2.3700000e+02]
 [2.2000000e+02 5.1500000e+02 1.6130000e+03 4.7400000e+02 2.6400000e+02
  0.0000000e+00 6.9000000e+01 3.7400000e+02 3.3190000e+03 4.4000000e+02
  2.6670000e+03 3.2600000e+02 3.7012820e+06 9.2300000e+02 7.9020000e-01
  3.6600000e+02]
 [1.6400000e+02 2.9200000e+02 9.2500000e+02 1.9600000e+02 1.3400000e+02
  1.0000000e+00 3.2000000e+01 3.4000000e+02 1.0770000e+03 9.0000000e+01
  1.7390000e+03 1.7200000e+02 2.5850490e+06 9.1600000e+02 7.6990000e-01
```

In [12]:

```
1 print(y)
```

```
[[19.0948287]
 [20.7002159]
 [20.9319821]
 [19.8761653]
 [18.990088 ]
 [21.177658 ]
 [20.5292147]
 [19.9615398]
 [20.9042201]
 [20.184871 ]
 [21.4549477]
 [19.7173703]
 [21.0076578]
 [19.8346659]
 [16.7049873]
 [18.4087934]
 [19.0759837]
 [19.1538231]
 [21.1458004]
 [19.1382514]
 [21.7468548]
 [19.9974533]
 [18.185332 ]
 [19.2608384]
 [18.5204303]
 [18.2376278]
 [16.990215 ]
 [16.8523973]
 [17.6804639]
 [16.3492193]
 [17.6599188]
 [19.2183307]
 [20.745319 ]
 [20.1119123]
 [20.3899385]]
```

In [13]:

```
1 print(y2)
```

```
[[74.7479789]
 [77.0081678]
 [77.7523039]
 [75.3433139]
 [75.7531324]
 [79.6570127]
 [76.1841701]
 [79.2961468]
 [74.7748979]
 [79.9947956]
 [80.1960712]
 [77.1493722]
 [75.5626039]
 [75.8816345]
 [74.2432527]
 [76.5603828]
 [72.8776559]
 [72.8751786]
 [79.0881546]
 [77.3209555]
 [74.123996 ]
 [73.7898023]
 [76.0419642]
 [76.774776 ]
 [73.8567437]
 [73.4445392]
 [73.3120233]
 [74.5814773]
 [74.018261 ]
 [73.5594128]
 [75.9063906]
 [72.9780897]
 [78.6021946]
 [77.1312586]
 [78.1306846]]
```

In [14]:

```
print(dataset)
```

```
    S. No State/UT/District  Homicide/Murder(3,4,15,16)  \
0       1        Ahmednagar                         215
1       2             Akola                         101
2       3          Amravati                         168
3       4         Aurangbad                         220
4       5              Beed                         164
5       6          Bhandara                          39
6       7          Buldhana                         101
7       8        Chandrapur                          74
8       9             Dhule                          96
9      10        Gadchiroli                         106
10     11            Gondia                          59
11     12           Hingoli                          81
12     13           Jalgaon                         121
13     14             Jalna                         121
14     15          Kolhapur                         153
15     16             Latur                          93
16     17            Mumbai                         345
17     18            Nagpur                         261
```

In [15]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state
X_train, X_test, y2_train, y2_test = train_test_split(X, y2, test_size = 0.2, random_st
```

In [16]:

```
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
sc_y = StandardScaler()
sc_y2 = StandardScaler()
X = sc_X.fit_transform(X)
y = sc_y.fit_transform(y)
y2 = sc_y2.fit_transform(y2)
```

In [17]:

```
# Fitting SVR to the dataset
from sklearn.svm import SVR
regressor = SVR(kernel = 'rbf')
#rbf = Gaussian Radial Basis Function Kernel
regressor.fit(X, y)
regressor.fit(X, y2)
```

```
C:\Users\home\AppData\Roaming\Python\Python38\site-packages\sklearn\utils\va
lidation.py:993: DataConversionWarning: A column-vector y was passed when a
1d array was expected. Please change the shape of y to (n_samples, ), for ex
ample using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\home\AppData\Roaming\Python\Python38\site-packages\sklearn\utils\va
lidation.py:993: DataConversionWarning: A column-vector y was passed when a
1d array was expected. Please change the shape of y to (n_samples, ), for ex
ample using ravel().
  y = column_or_1d(y, warn=True)
```

Out[17]:

```
SVR()
```

In [18]:

```
# Predicting a new result

y_pred = regressor.predict(sc_X.transform(np.array([[101,248,1624,277,97,0,29,265,950,1
y_pred2 = regressor.predict(sc_X.transform(np.array([[101,248,1624,277,97,0,29,265,950,
#To transform 6.5 to the scaled X value, we first need to convert it into the array for
#Since the transform method of StandardScaler Library only accepts arrays
```

In [19]:

```
y_pred = sc_y.inverse_transform(np.array(y_pred).reshape(1,-1))
y_pred2 = sc_y2.inverse_transform(np.array(y_pred2).reshape(1,-1))
#Now the prediction gives us the scaled value of y
#Thus we need inverse transformation of the scaled value for the real results
```

In [20]:

```
print(y_pred)
print(y_pred2)
```

```
[[19.96801173]]
[[76.77455569]]
```

In [23]:

```
def district_pred():
    target_district = ""
    min = 999
    num = 0
    for i in range(len(dataset['Latitude'])):
            if (abs(y_pred-dataset['Latitude'][i])+abs(y_pred2-dataset['Longitude'][i])
                min = abs(y_pred-dataset['Latitude'][i])+abs(y_pred2-dataset['Longitude
                target_district = dataset['State/UT/District'][i]
                num = i
    return target_district,num
```

In [25]:

```
str, x = district_pred()
print(str)
#print(dataset['Latitude'][x])
#print(dataset['Longitude'][x])
```

```
Washim
```

In [ ]:

```

```