EE559- Mathematical Pattern Recognition

Homework #8

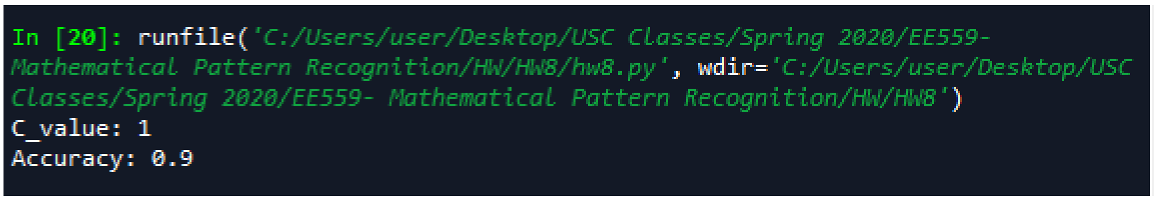
Sourabh Tirodkar
3589406164

Results:

Problem 1:

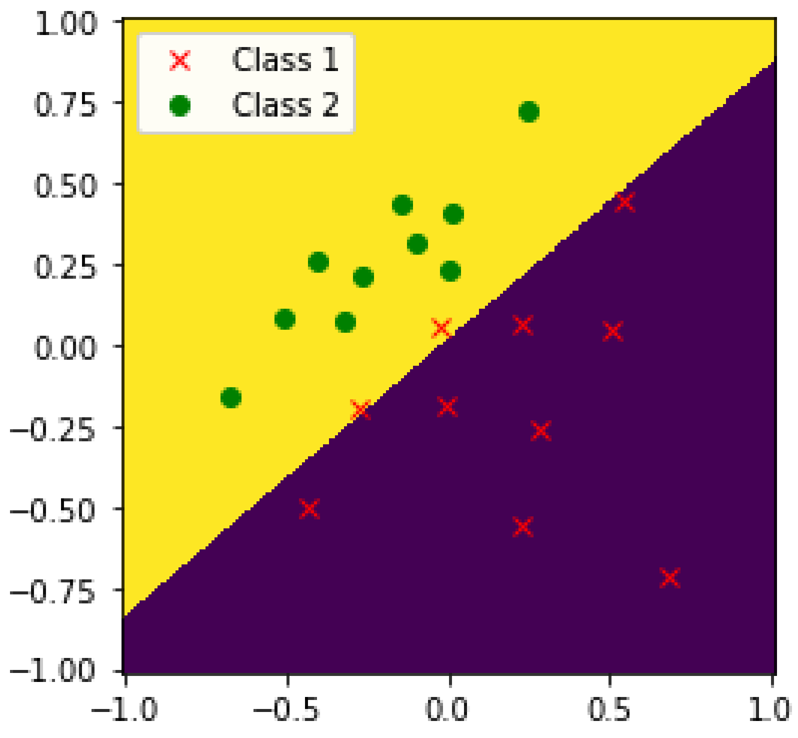
1.a

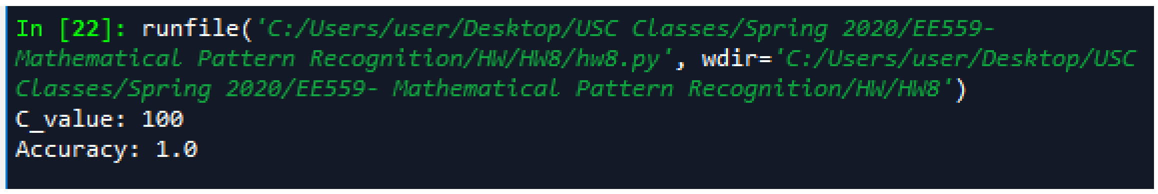
Data: HW8_1



```
In [20]: runfile('C:/Users/user/Desktop/USC Classes/Spring 2020/EE559-
Mathematical Pattern Recognition/HW/HW8/hw8.py', wdir='C:/Users/user/Desktop/USC
Classes/Spring 2020/EE559- Mathematical Pattern Recognition/HW/HW8')
C_value: 1
Accuracy: 0.9
```

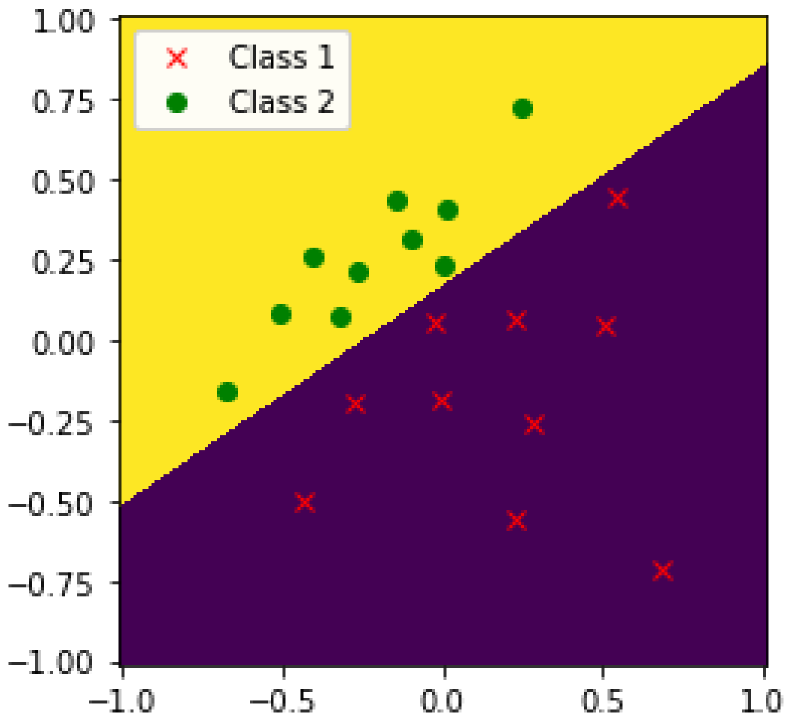
C=1
Accuracy=90%





```
In [22]: runfile('C:/Users/user/Desktop/USC Classes/Spring 2020/EE559-
Mathematical Pattern Recognition/HW/HW8/hw8.py', wdir='C:/Users/user/Desktop/USC
Classes/Spring 2020/EE559- Mathematical Pattern Recognition/HW/HW8')
C_value: 100
Accuracy: 1.0
```

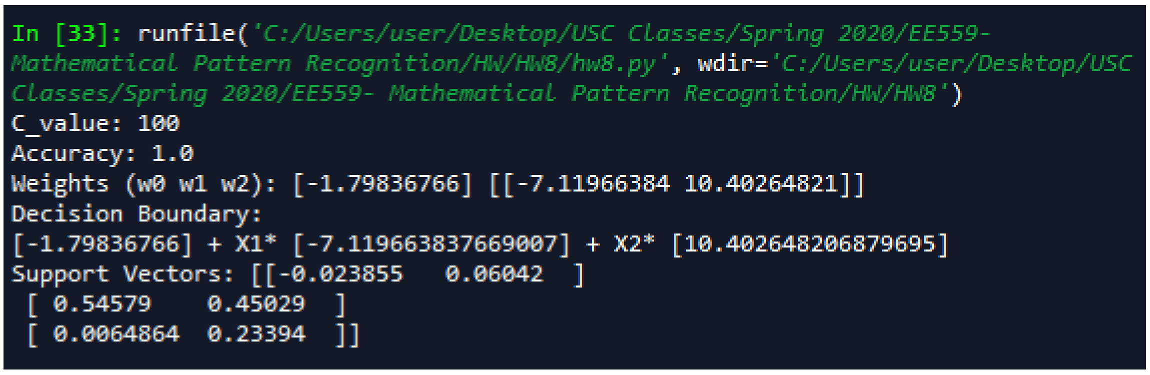
C=100
Accuracy=100%



Comments:

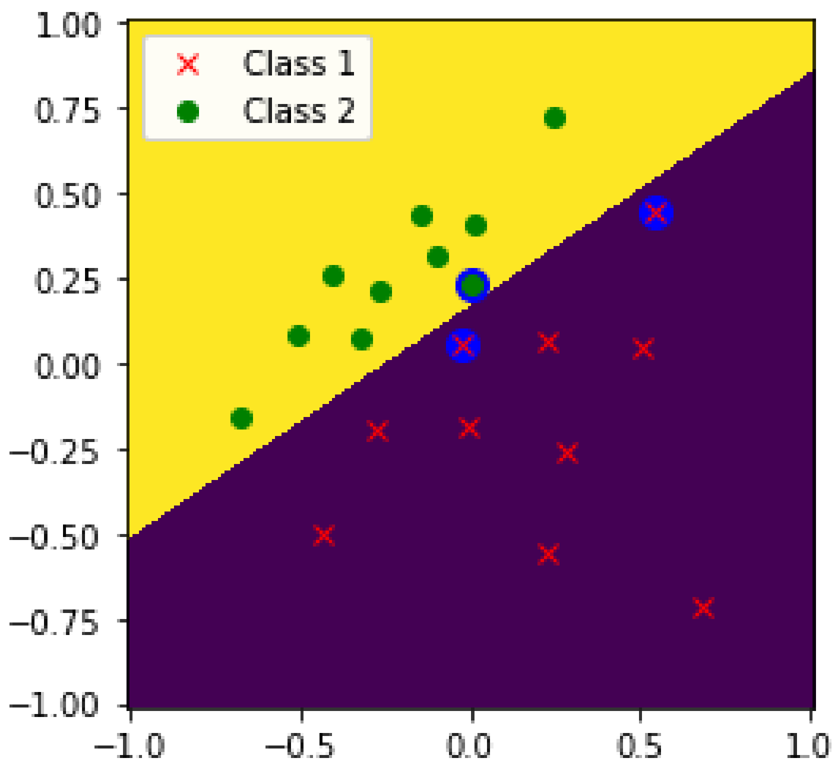
C is a regularization parameter in SVM classifier. It is also a penalty for the misclassifying data points. When C is small, the misclassified data points areacceptable and thus making the accuracy to decrease. When C is increase (here C=100), no misclassified data points are allowed and hence the decision boundary perfectly fits the data and gives best performance.

1.b

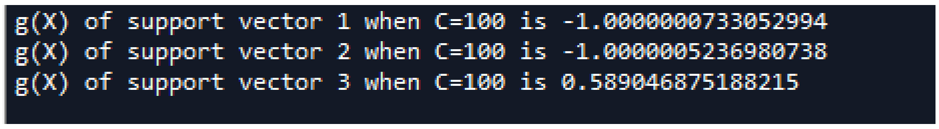


```
In [33]: runfile('C:/Users/user/Desktop/USC Classes/Spring 2020/EE559-
Mathematical Pattern Recognition/HW/HW8/hw8.py', wdir='C:/Users/user/Desktop/USC
Classes/Spring 2020/EE559- Mathematical Pattern Recognition/HW/HW8')
C_value: 100
Accuracy: 1.0
Weights (w0 w1 w2): [-1.79836766] [[-7.11966384 10.40264821]]
Decision Boundary:
[-1.79836766] + X1* [-7.119663837669007] + X2* [10.402648206879695]
Support Vectors: [[-0.023855   0.06042  ]
 [ 0.54579    0.45029  ]
 [ 0.0064864  0.23394  ]]
```

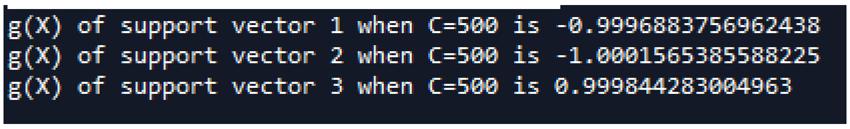
Accuracy: 100.0
Weights (w0 w1 w2): [-1.79836766] [[-7.11966384 10.40264821]]



1.c



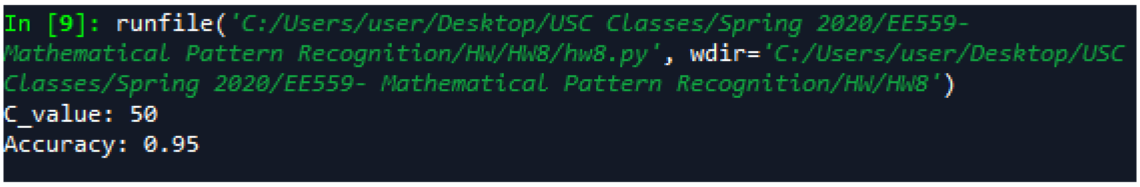
```
g(X) of support vector 1 when C=100 is -1.0000000733052994
g(X) of support vector 2 when C=100 is -1.0000005236980738
g(X) of support vector 3 when C=100 is 0.589046875188215
```



```
g(X) of support vector 1 when C=500 is -0.9996883756962438
g(X) of support vector 2 when C=500 is -1.0001565385588225
g(X) of support vector 3 when C=500 is 0.999844283004963
```

The 1st and 2nd support vector values are in (+-1) range. So it lies on the decision boundary. But the 3rd vector does not. As discussed, when C is small, misclassified points are allowed and hence support vector 3 does not lie.
When increasing the C value to 500, we get all 3 support vectors on the decision booundary. This cam be seen by the value above.

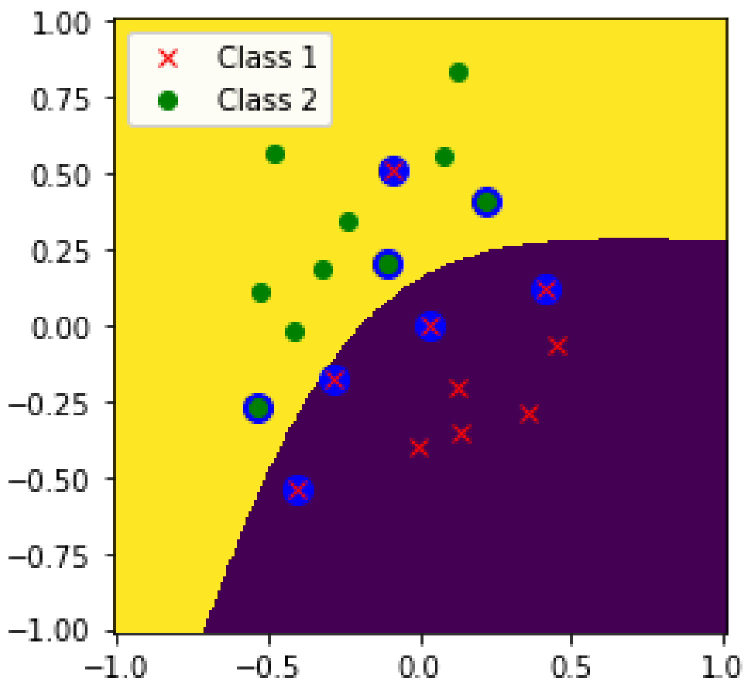
1.d

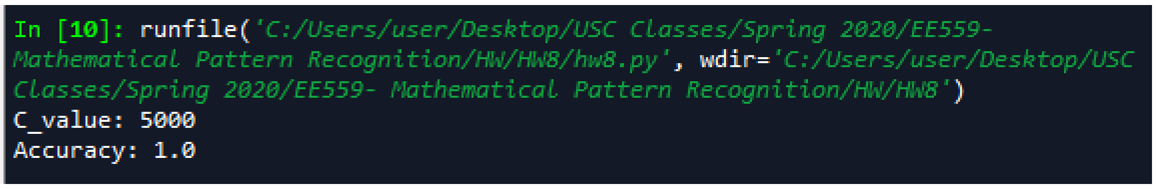


```
In [9]: runfile('C:/Users/user/Desktop/USC Classes/Spring 2020/EE559-
Mathematical Pattern Recognition/HW/HW8/hw8.py', wdir='C:/Users/user/Desktop/USC
Classes/Spring 2020/EE559- Mathematical Pattern Recognition/HW/HW8')
C_value: 50
Accuracy: 0.95
```

C=50

Accuracy: 95.0

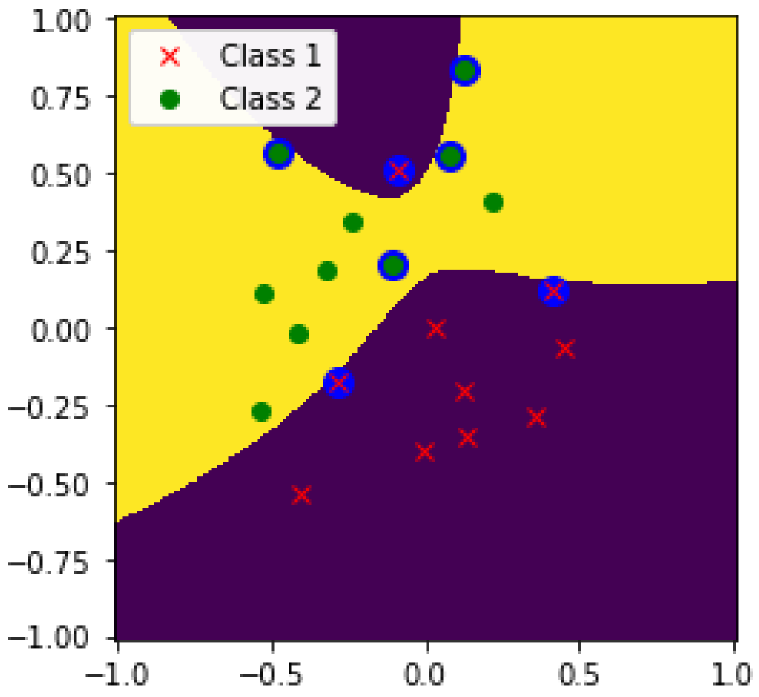




```
In [10]: runfile('C:/Users/user/Desktop/USC Classes/Spring 2020/EE559-
Mathematical Pattern Recognition/HW/HW8/hw8.py', wdir='C:/Users/user/Desktop/USC
Classes/Spring 2020/EE559- Mathematical Pattern Recognition/HW/HW8')
C_value: 5000
Accuracy: 1.0
```

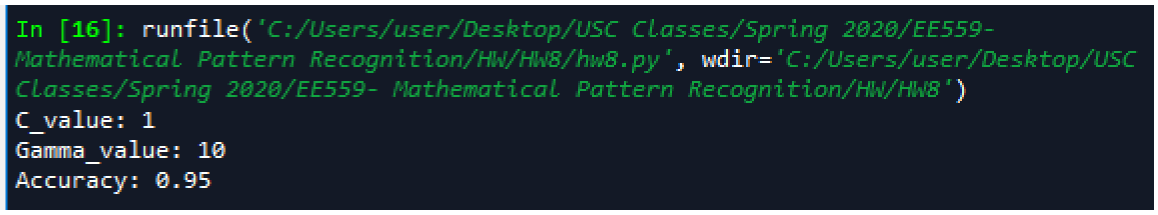
C=5000

Accuracy: 100.0



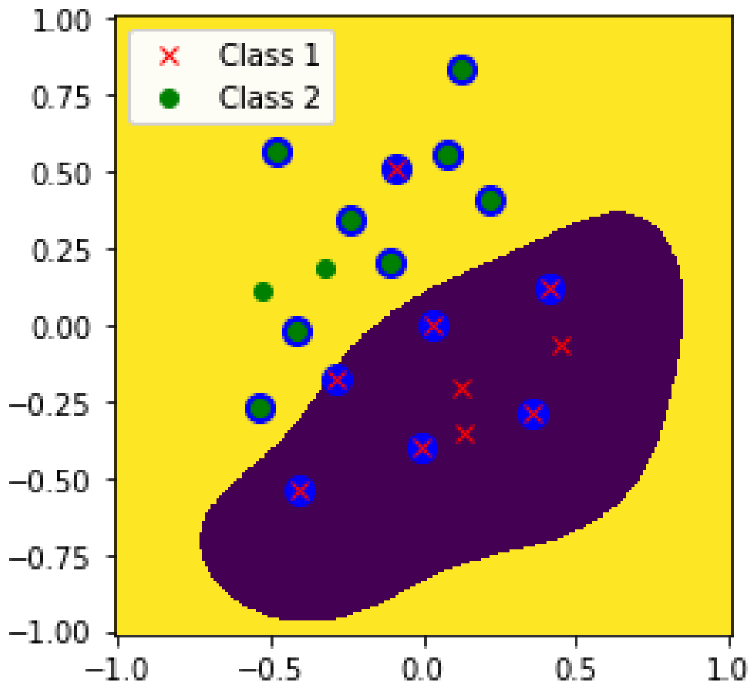
Comments:
The data in original feature space are non linear, so we have to used Gaussian Kernel (RBF) as it classifies the data points. RBF makes this feature space into higher dimesnion which makes the decision boundary non linear.
It is observed that increasing the value of C (Regularization Parameter) gives us the higher result, meaning it correctly classifies all the data points. The strength of regularization is inversely propotional to C. The penalty is squared L2 distance. When C is smalll, unclassified data points are considered for the evalution in the classifier and so decision boundaray is not so precise (here C=50). When C is increased, unclassified data points are more penalized in the classifier and so slack variables becomes smaller to adjust. Thus making the decision boundary more accurate.

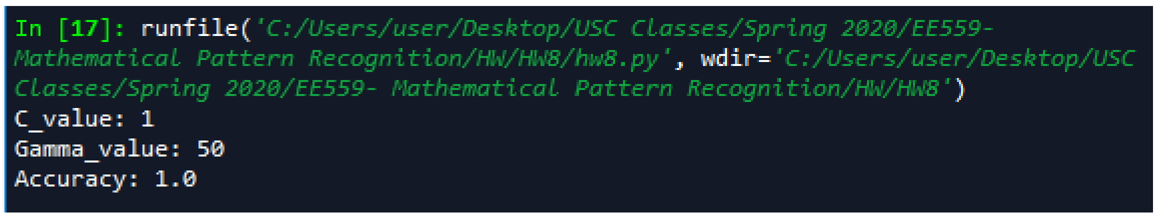
1.e



```
In [16]: runfile('C:/Users/user/Desktop/USC Classes/Spring 2020/EE559-
Mathematical Pattern Recognition/HW/HW8/hw8.py', wdir='C:/Users/user/Desktop/USC
Classes/Spring 2020/EE559- Mathematical Pattern Recognition/HW/HW8')
C_value: 1
Gamma_value: 10
Accuracy: 0.95
```

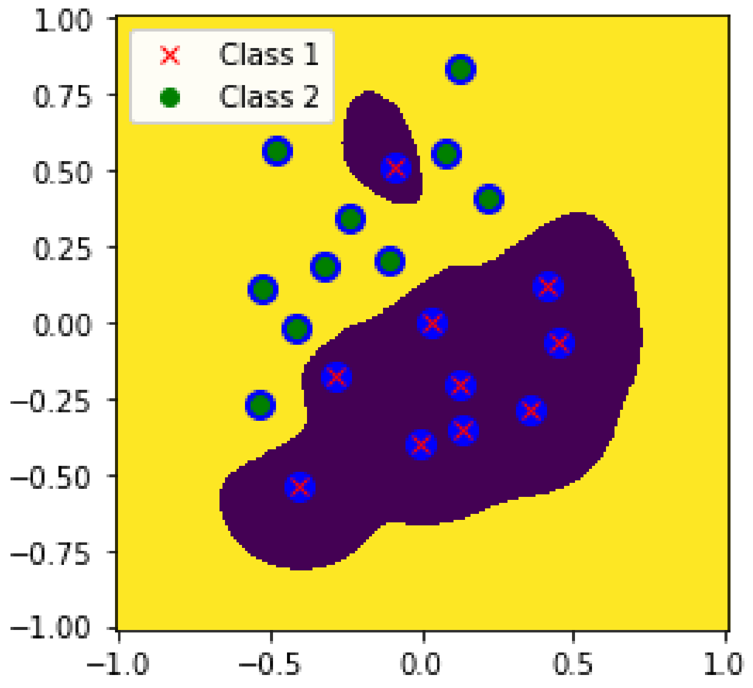
C=1 (default value)
Gamma=10
Accuracy: 95.0

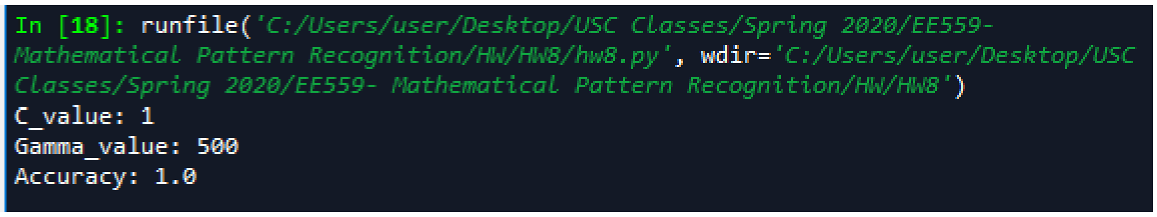




```
In [17]: runfile('C:/Users/user/Desktop/USC Classes/Spring 2020/EE559-
Mathematical Pattern Recognition/HW/HW8/hw8.py', wdir='C:/Users/user/Desktop/USC
Classes/Spring 2020/EE559- Mathematical Pattern Recognition/HW/HW8')
C_value: 1
Gamma_value: 50
Accuracy: 1.0
```

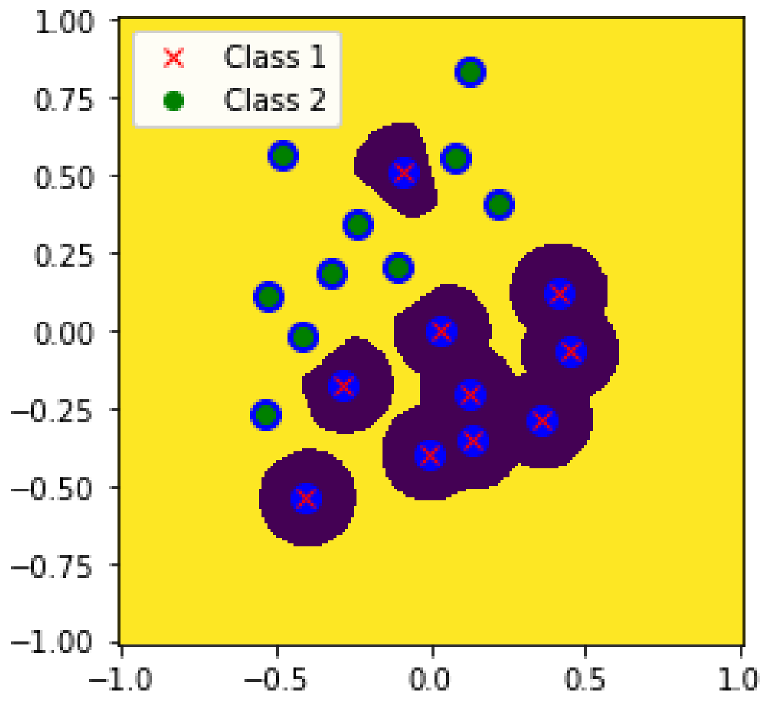
C=1 (Default)
Gamma=50
Accuracy: 100.0





```
In [18]: runfile('C:/Users/user/Desktop/USC Classes/Spring 2020/EE559-
Mathematical Pattern Recognition/HW/HW8/hw8.py', wdir='C:/Users/user/Desktop/USC
Classes/Spring 2020/EE559- Mathematical Pattern Recognition/HW/HW8')
C_value: 1
Gamma_value: 500
Accuracy: 1.0
```

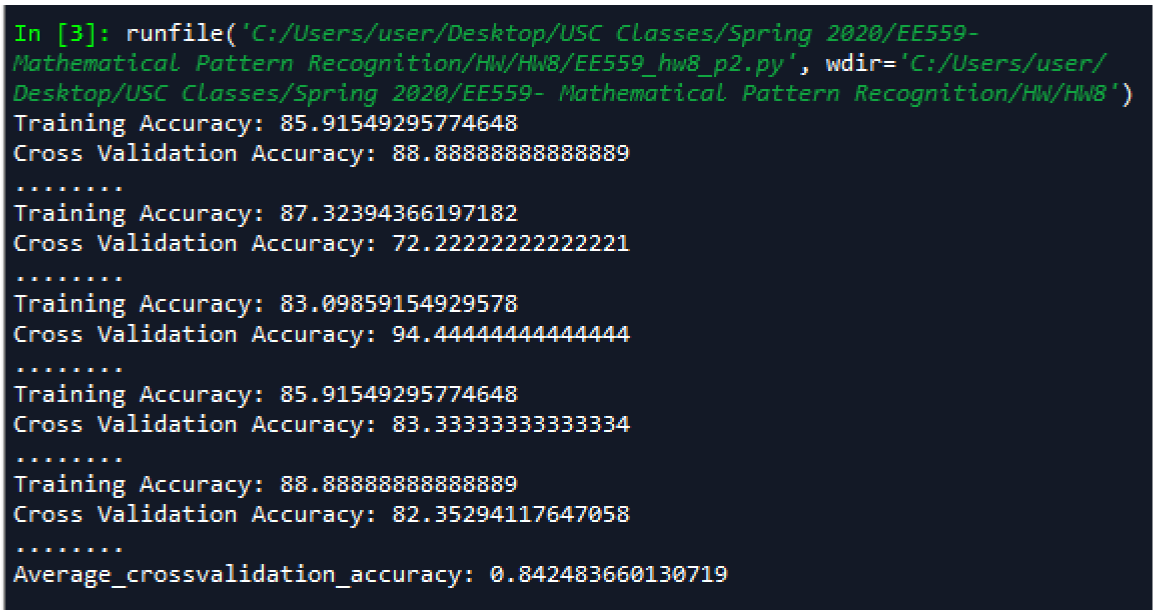
C=1(default)
Gamma=500
Accuracy: 100.0



<u>Comments</u>:
Gamma is the kernel parameter for the RBF kernel. It can viewed as a spread of RGF kernel which affects the decision boundary and hence the accuracy. It is observed that when we increase the gamma value, the decision boundary fits the data accurately. But after increasing the value more, it does overfit the data. This can be observed in gamma=500 as the plot shows circle of each data point clearly showing overfitting. So, we are getting 100% accuracy on gamma=500 but the test data which is never seen will give poor performance. The gamma value should be choosen accordingly in the training data set for the same reason.
The decision boundaries plot also shows the same as more fiiting occurs, so less misclassfied points as gamma increases.

Problem 2

2.a



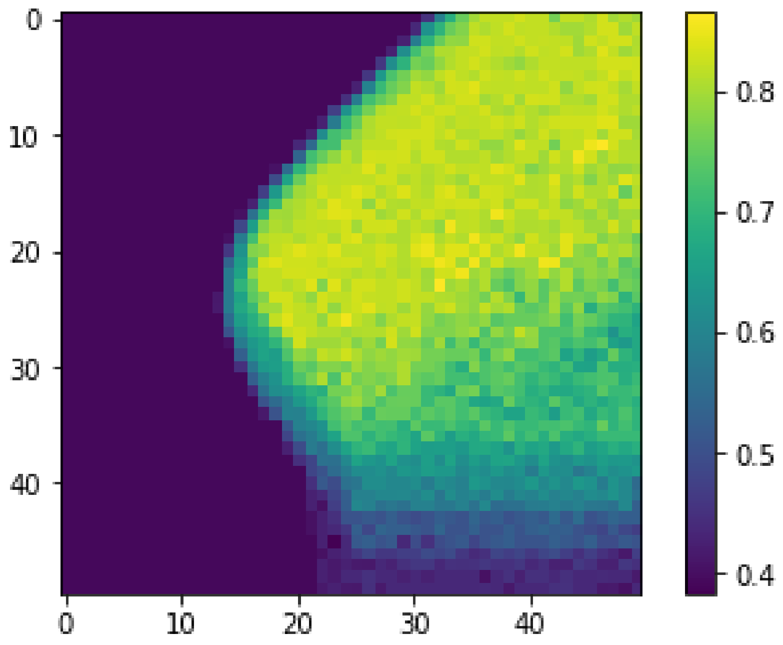
```
In [3]: runfile('C:/Users/user/Desktop/USC Classes/Spring 2020/EE559-
Mathematical Pattern Recognition/HW/HW8/EE559_hw8_p2.py', wdir='C:/Users/user/
Desktop/USC Classes/Spring 2020/EE559- Mathematical Pattern Recognition/HW/HW8')
Training Accuracy: 85.91549295774648
Cross Validation Accuracy: 88.88888888888889
........
Training Accuracy: 87.32394366197182
Cross Validation Accuracy: 72.22222222222221
........
Training Accuracy: 83.09859154929578
Cross Validation Accuracy: 94.44444444444444
........
Training Accuracy: 85.91549295774648
Cross Validation Accuracy: 83.33333333333334
........
Training Accuracy: 88.88888888888889
Cross Validation Accuracy: 82.35294117647058
........
Average_crossvalidation_accuracy: 0.842483660130719
```

The average cross validation is 84.24% when C=1, gamma=1.

2.b

In this part, gamma and C values are choosen from the range given [10^-3 to 10^3]. 50 points are randomly choosen from the range. ACC matrix is used for storing the average accuracy while DEV is used for storing standard deviation.

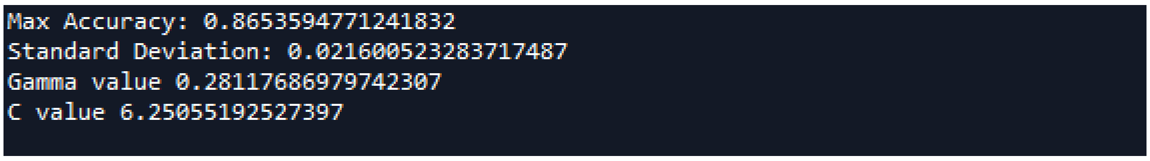
2.b.i



The plot shows the visulaization of ACC matrix which is used to strore the average accuracy in the range for gamma and C.

2.b.ii

The average of max accuracy is taken. However it is observed, the max accuarcy can be repeated many times in ACC matrix. For those, we take accuracy with lowest standard deviation and report it. My values for the same are as below:

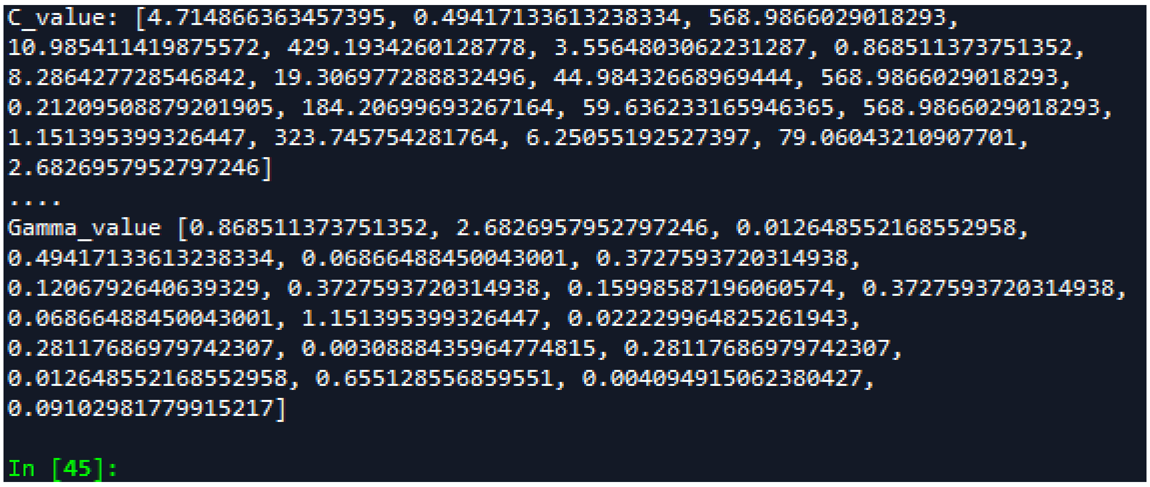


```
Max Accuracy: 0.8653594771241832
Standard Deviation: 0.021600523283717487
Gamma value 0.28117686979742307
C value 6.25055192527397
```

2.c

2.c.i

The same part in b is exceuted for 20 epochs and every time the ACC and DEV matrix are stored. So we get 20 ACC and 20 DEV matrix. From this, we get 20 pairs of gamma and C value which are as follows:



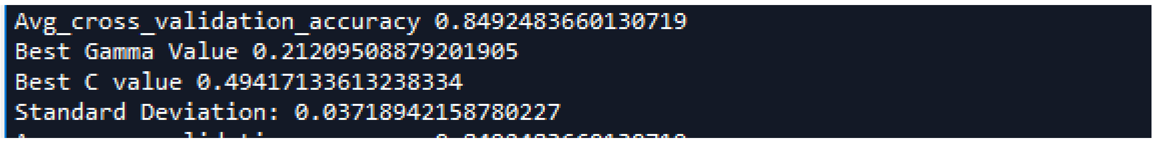
```
C_value: [4.714866363457395, 0.49417133613238334, 568.9866029018293,
10.985411419875572, 429.1934260128778, 3.5564803062231287, 0.868511373751352,
8.286427728546842, 19.306977288832496, 44.98432668969444, 568.9866029018293,
0.21209508879201905, 184.20699693267164, 59.636233165946365, 568.9866029018293,
1.151395399326447, 323.745754281764, 6.25055192527397, 79.06043210907701,
2.6826957952797246]
....
Gamma_value [0.868511373751352, 2.6826957952797246, 0.012648552168552958,
0.49417133613238334, 0.06866488450043001, 0.3727593720314938,
0.1206792640639329, 0.3727593720314938, 0.15998587196060574, 0.3727593720314938,
0.06866488450043001, 1.151395399326447, 0.022229964825261943,
0.28117686979742307, 0.0030888435964774815, 0.28117686979742307,
0.012648552168552958, 0.655128556859551, 0.004094915062380427,
0.09102981779915217]

In [45]:
```

The C_value and Gamma_value are key value pairs here in the above figure.

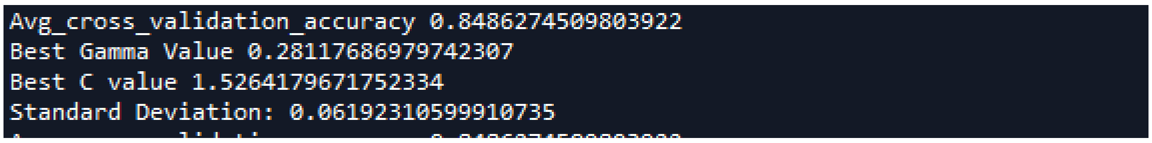
2.c.ii

Random state=k+5



```
Avg_cross_validation_accuracy 0.8492483660130719
Best Gamma Value 0.21209508879201905
Best C value 0.49417133613238334
Standard Deviation: 0.03718942158780227
```

Random state=k+7



```
Avg_cross_validation_accuracy 0.8486274509803922
Best Gamma Value 0.28117686979742307
Best C value 1.5264179671752334
Standard Deviation: 0.06192310599910735
```

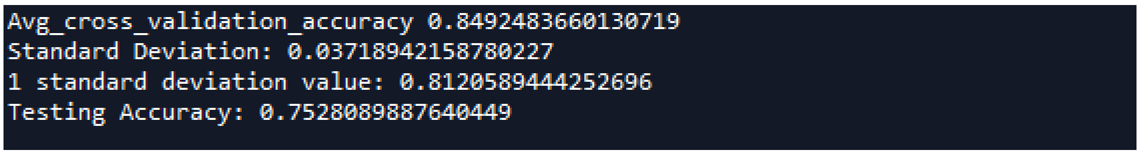
Running over 20 epochs give more stability to the choose the parameters. So averaging the ACC and DEV matrix which is calculated for 20 epochs can help us find the best C and gamma value. Picking the best [gamma, C] is more well defined now as the it is the average of all those.

2.d

The best C, gamma pair is chosen from the c part. This C and gamma value is choosen for the training of all the dataset to make amodel. This model is then applied to the testing dataset and accuracy is reported.

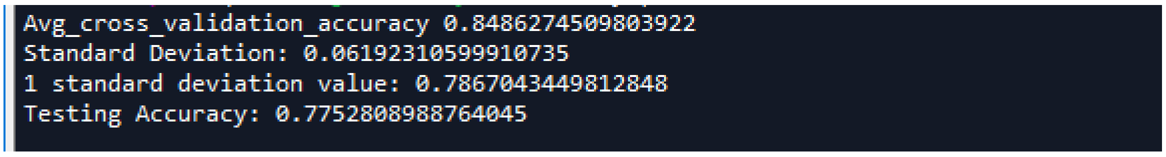
The testing accuracy is not in the one standard deviation, it is more than it. This is because we have chosen only 2 features out of 13. For better accuracy, we need to consider all 13 features. Thus the classifier is more than 1 standard deviation.

For random state=k+5



```
Avg_cross_validation_accuracy 0.8492483660130719
Standard Deviation: 0.03718942158780227
1 standard deviation value: 0.8120589444252696
Testing Accuracy: 0.7528089887640449
```

For random state=k+7



```
Avg_cross_validation_accuracy 0.8486274509803922
Standard Deviation: 0.06192310599910735
1 standard deviation value: 0.7867043449812848
Testing Accuracy: 0.7752808988764045
```