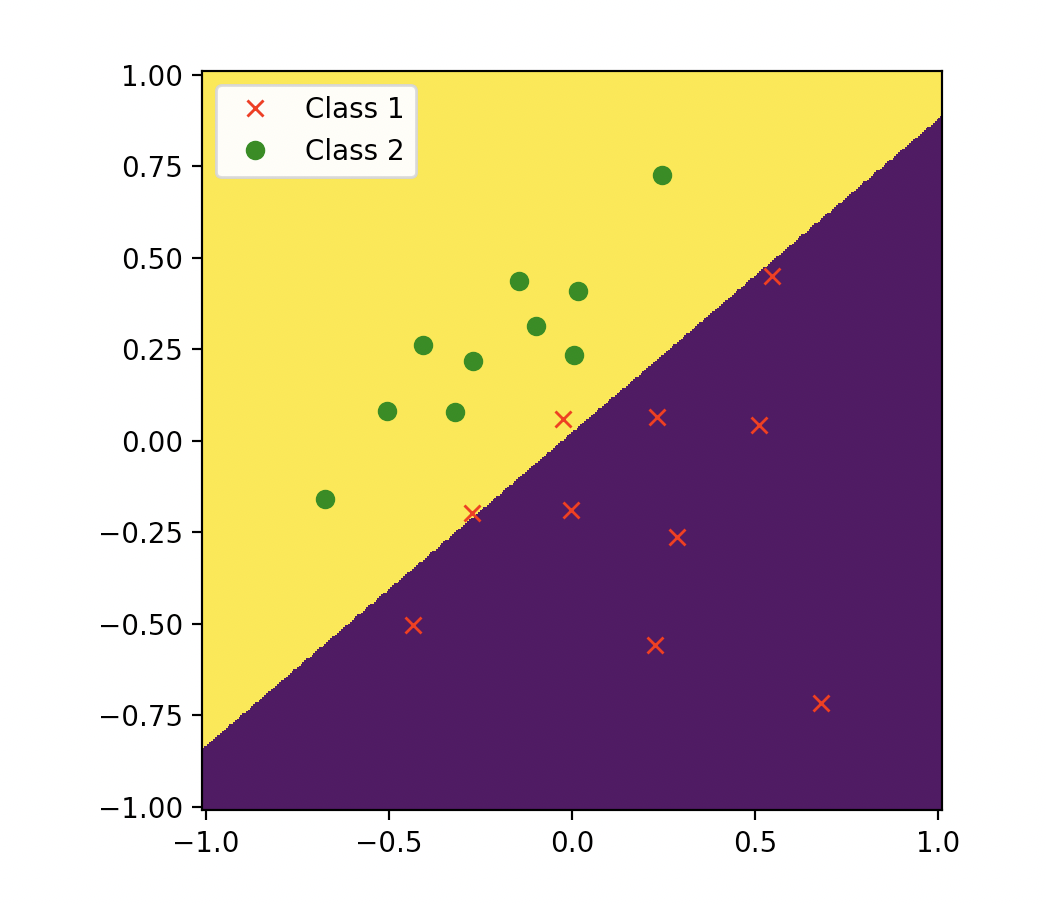
[jxu72364@usc.edu](mailto:jxu72364@usc.edu)

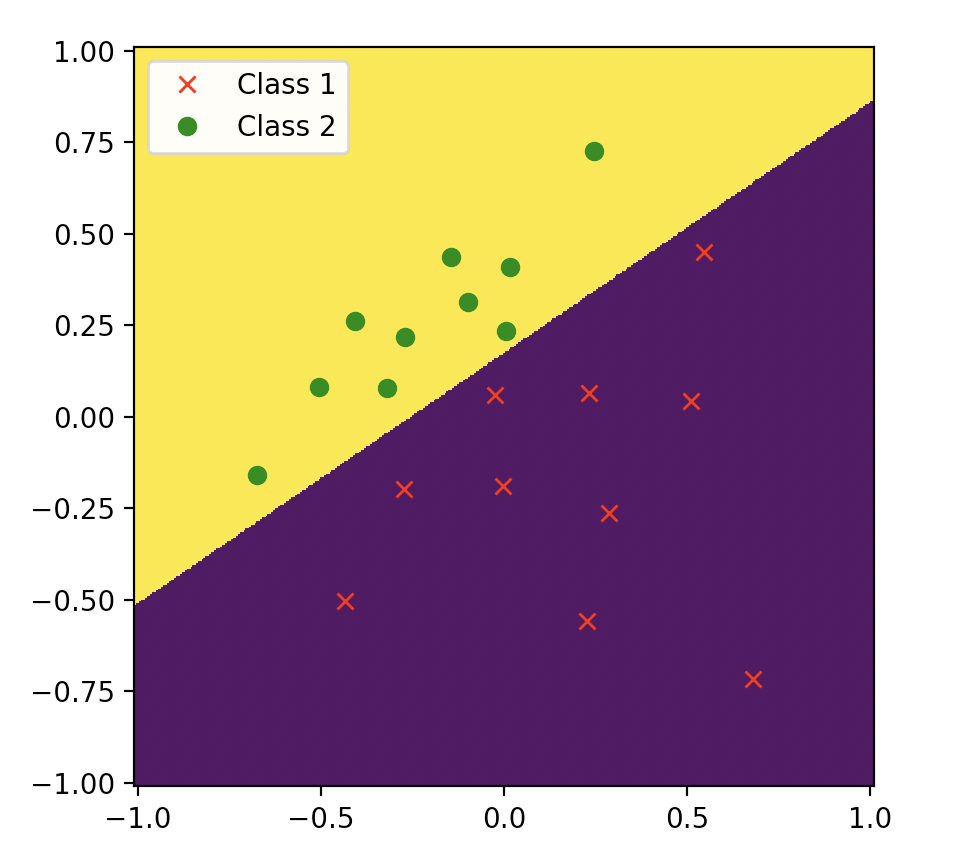
**Problem1 (a)**

Parameter C controls the trade-off between giving a low error on the training data and minimizing the norm of the weights. If parameter c is too big, it means the cost of misclassification is increased and the algorithm will try to reduce |w| as much as possible thus leading the hyperplane trying to classify each training data correctly. Same reason, if c is too small then may leading to large training error





C=1 accuracy is 0.9

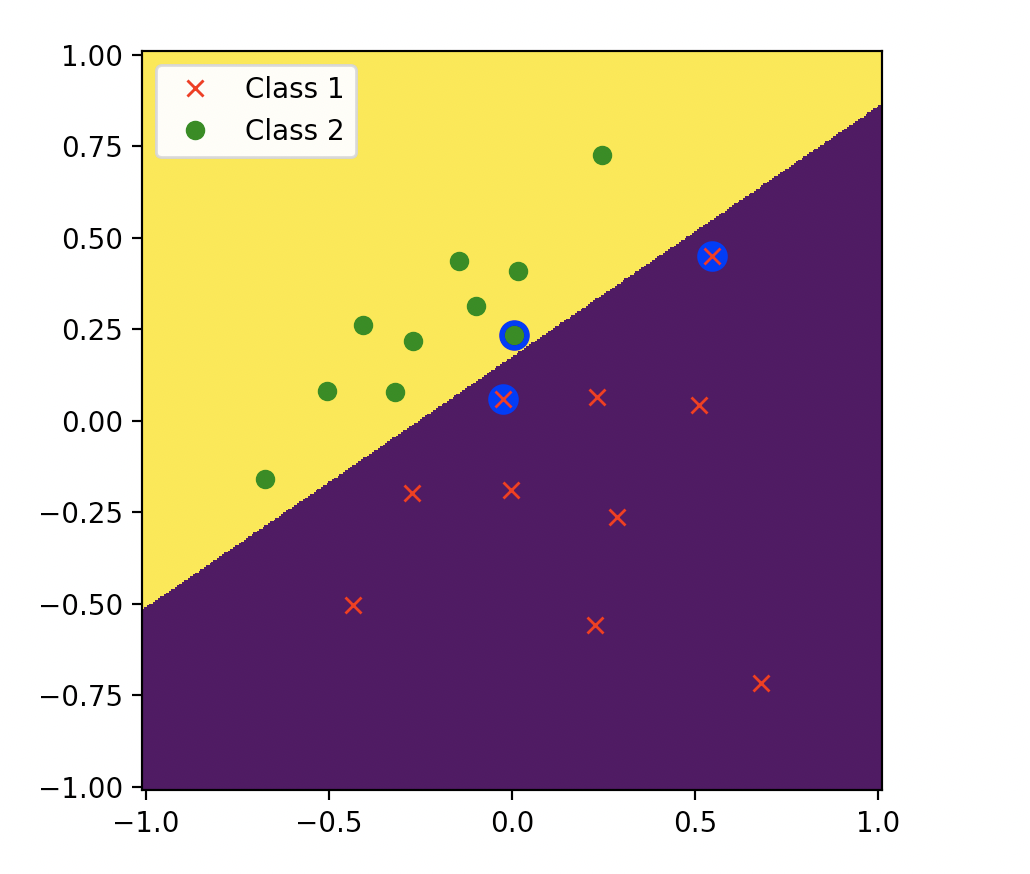


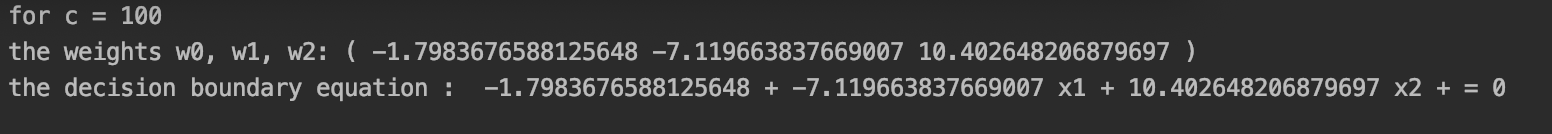


c=100 accuracy is 1.0

**Explanation:** when we increase c, that means we will try to classify all the training data correctly, so we will get a relatively high accuracy. And if C is small, we are more likely to misclassify some training data

**Problem1 (b)**

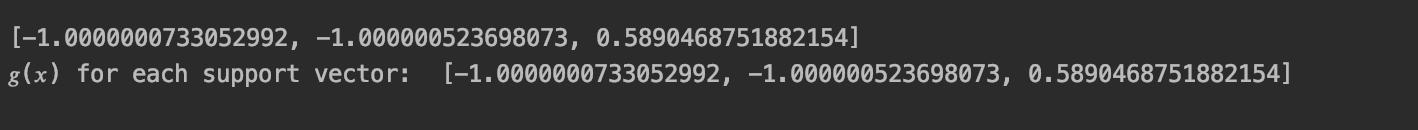




C = 100，the support vectors are circled in the picture

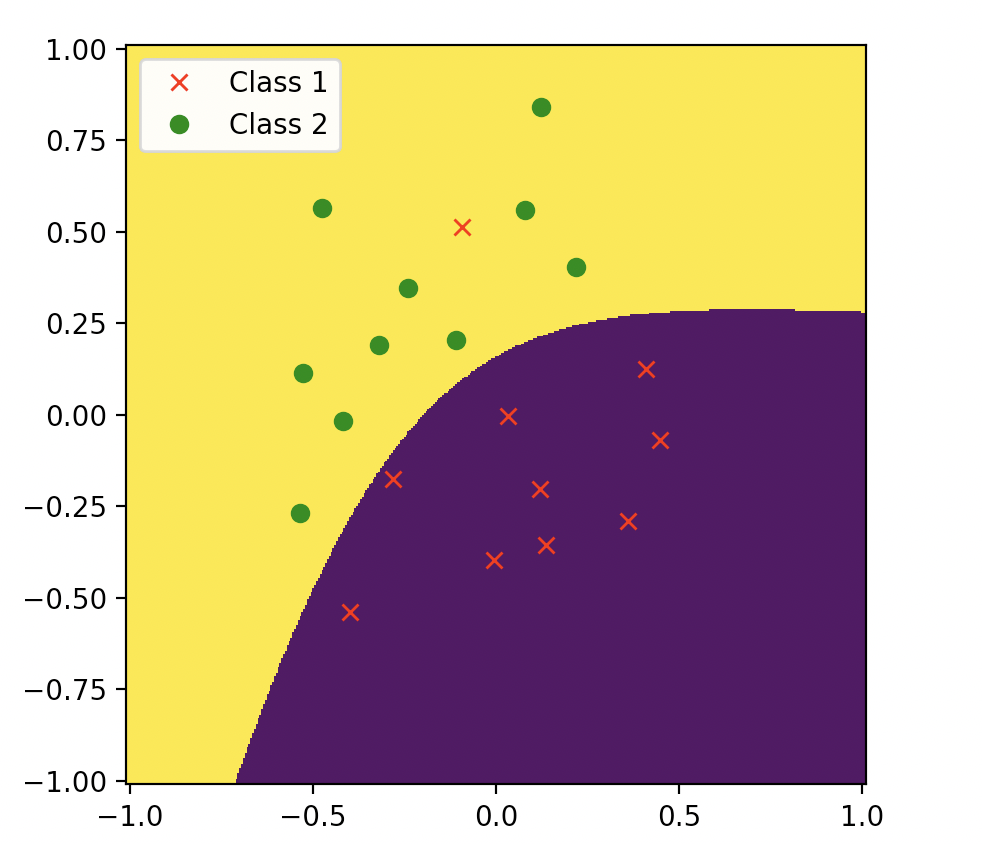
the weights w0, w1, w2 and the decision boundary equation are showed above

**Problem1 (c)**



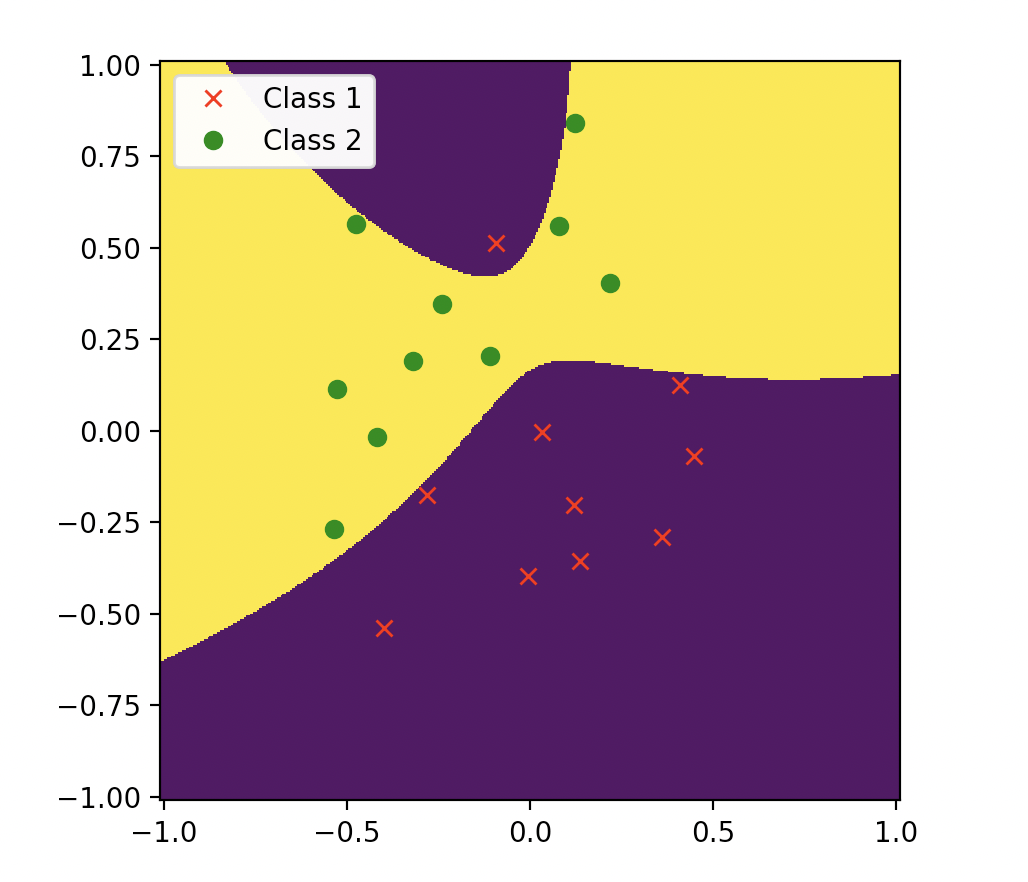
We can see if we put support vectors into equations we can get the value -1,-1 and 0.58905，so the first two support vectors are on the boundary of the margin (corresponding to 𝑔(𝑥) = -1). And the third support vector is in the margin between 𝑔(𝑥) = +1 and 𝑔(𝑥) = -1

**Problem1 (d)**





Use Gaussian (RBF) Kernel c = 50 accuracy is 0.95





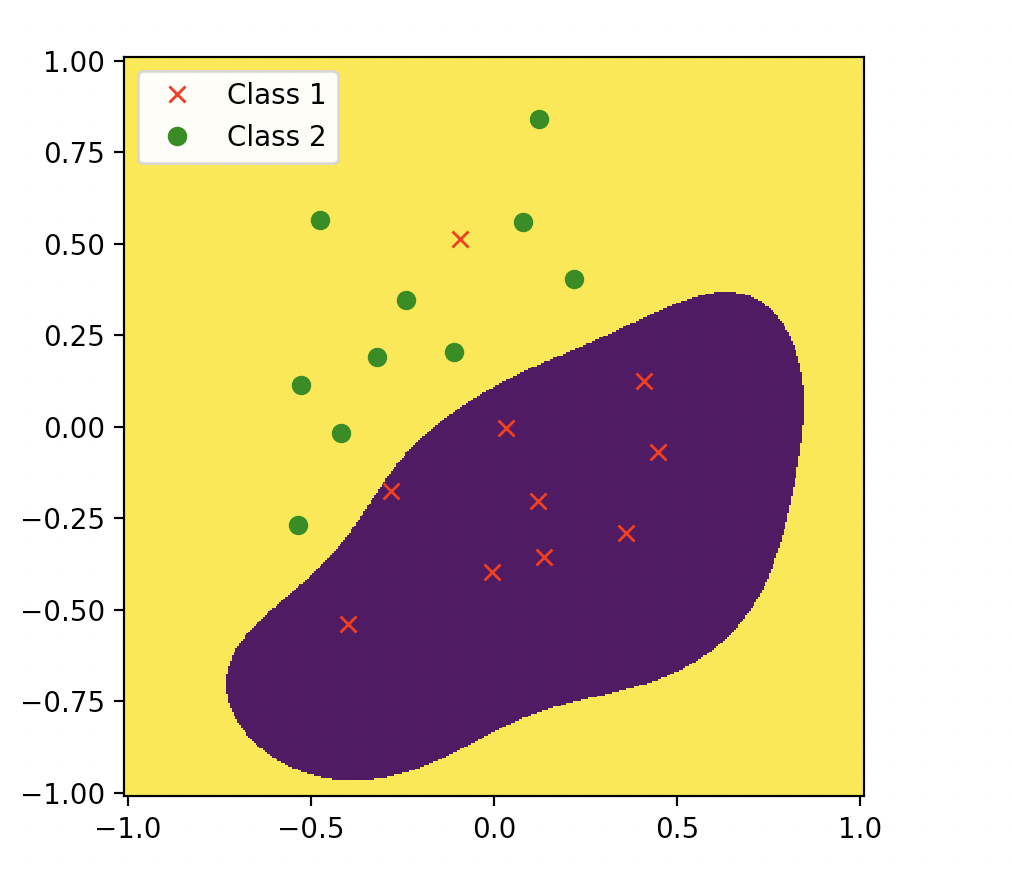
Use Gaussian (RBF) Kernel c = 5000 accuracy is 1.0

Explanation:

The decision boundary is non-linear, because we use Gaussian (RBF) Kernel, and in higher dimensions the boundary is linear, but in 2D dimension it is non-linear.

For the 2 values of C, the same reason as Problem 1(a), when C becomes larger, the boundary will become more squiggly to avoid misclassifying of training data

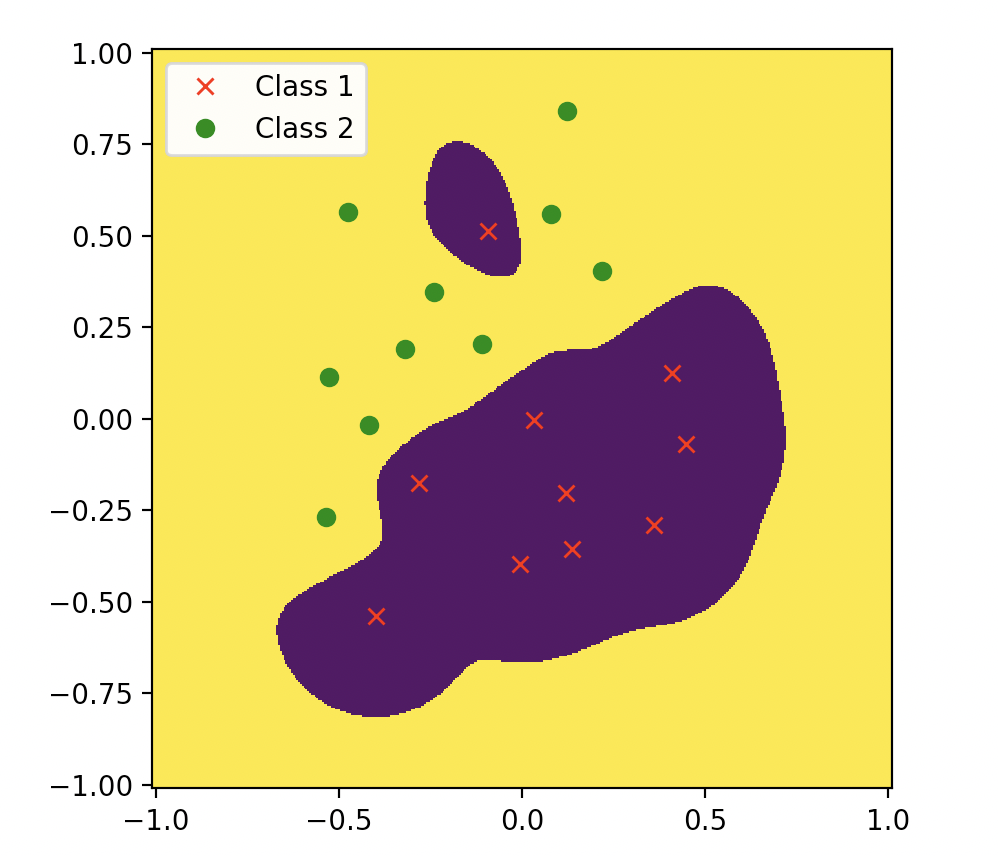
**Problem 1(e)**





Use Gaussian Kernel, default C parameter, gamma = 10

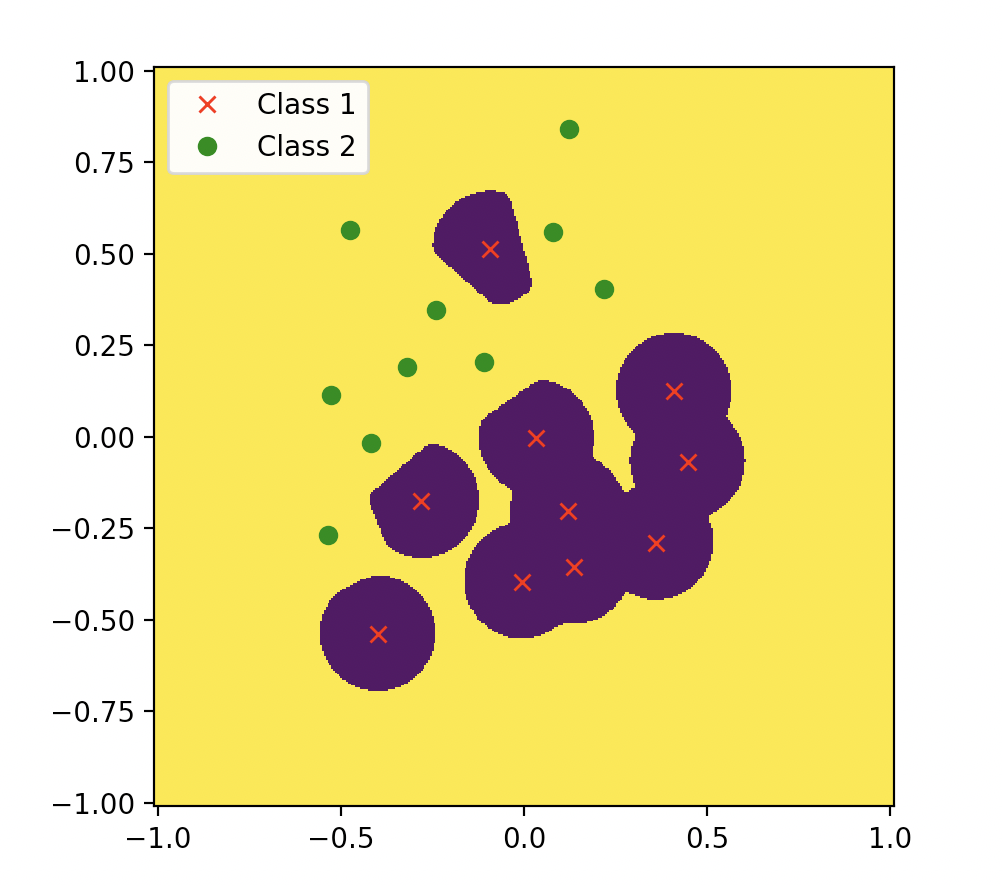
Accuracy is 0.95





Use Gaussian Kernel, default C parameter, gamma = 50

Accuracy is 1





Use Gaussian Kernel, default C parameter, gamma = 5000

Accuracy is 1

Explanation:

the gamma parameter defines how far the influence of a single training point reaches, with low gamma we can get go further. But When gamma is very small, the model will be too constrained, and we cannot classify all the data correctly

So we can see when gamma is 50, we cannot classify all the data correctly, and when gamma is 500.we correctly classify all data and the region becomes smaller. But when gamma is 5000 the region of a single point is too small, and we are overfitting

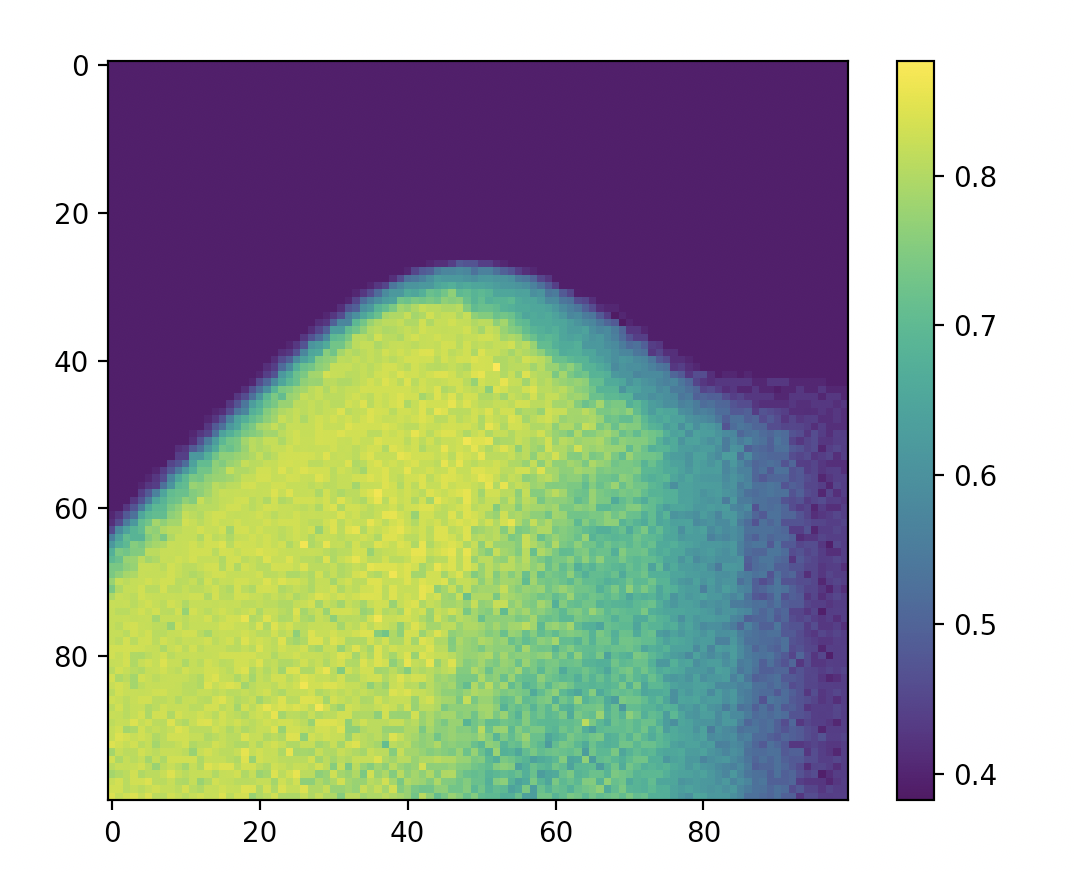
**Problem 2(a)**



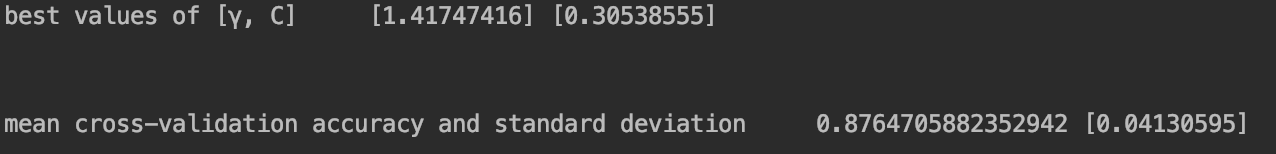
Use Gaussian (RBF) Kernel. Set γ = 1 , C = 1

the average cross- validation accuracy is about 0.84379

**Problem 2(b) (i)**



**Problem 2(b) (ii)**

****

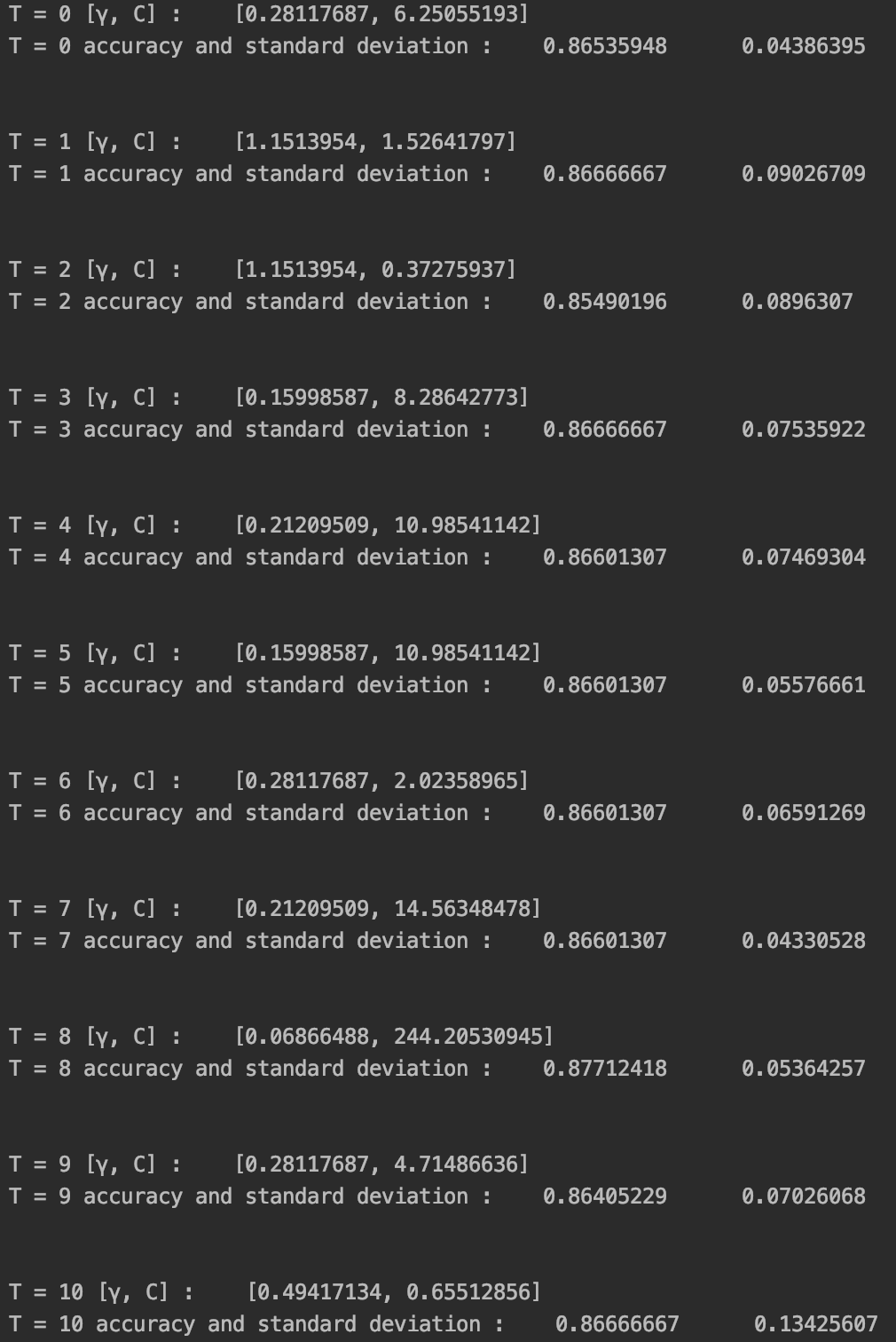
Yes we can see that the clear choice for best values of [γ, C]is

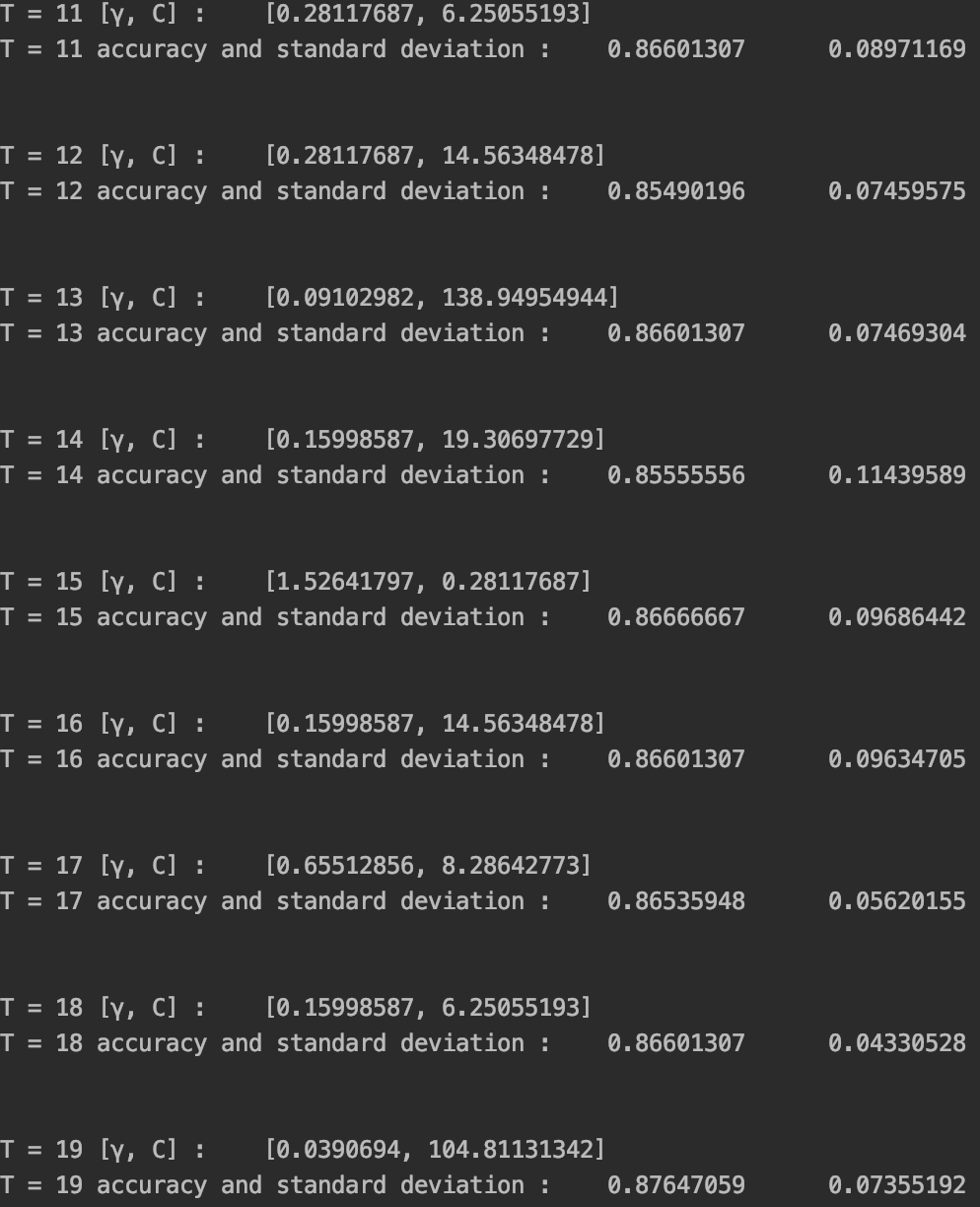
γ = 1.41747 and C = 0.30539

mean cross-validation accuracy is about 0.87647

standard deviation is about 0.0413

**Problem 2(C) (i)**

****

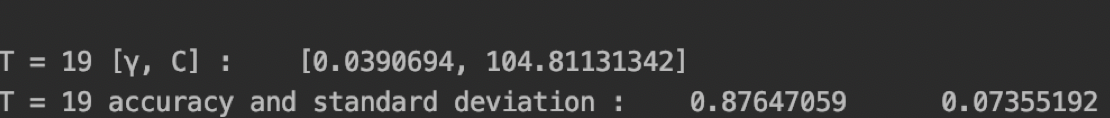
****

20 chosen pairs of [γ, C] and the accuracy and standard deviation are showed above

**Problem2 (c) (ii)**

We can pick from the plot showed above

And the best values for [γ,C] over the T=20 runs more reproducible because we use cross-validation for model selection and the result can be trusted as the result to be used in test data, it will be global optimal, and when we repeat the cross validation procedure 20 times, we can find that the accuracy is almost same, so [γ,C] is more reproducible

****

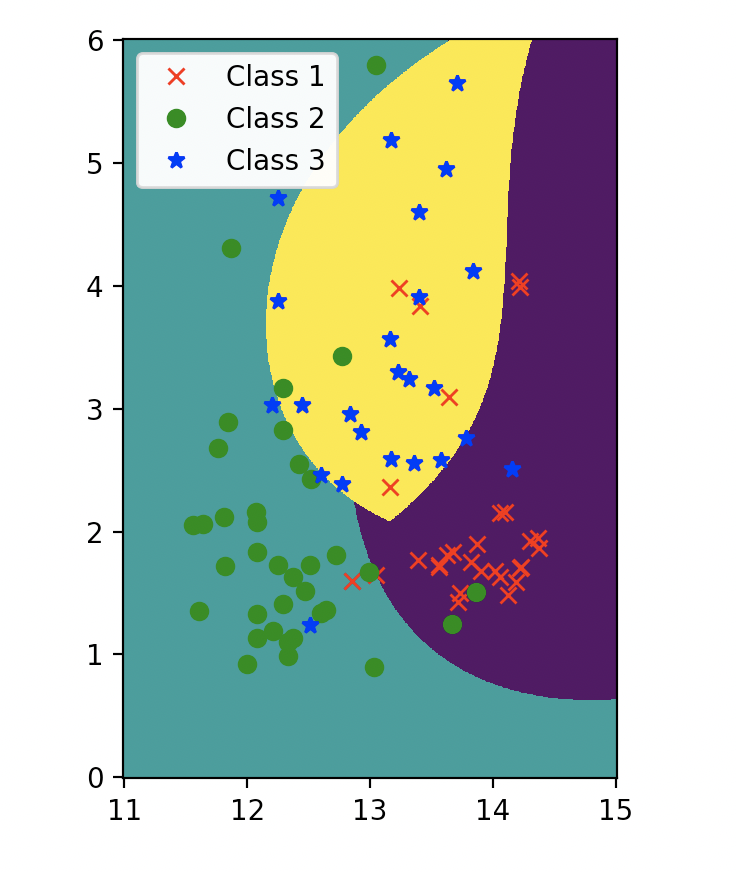
the final chosen best values for [γ, C],are showed above:

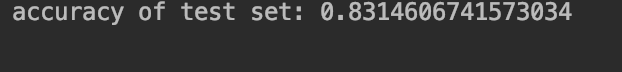
γ=0.0390694 and C = 104.81131342

mean cross-validation accuracy are 0.87647059

standard deviation are 0.07355192

**Problem2 (d)**

****

****

We can see that the accuracy of final classifier on unknowns is about 0.83146067

Because it is within 0.87647059 ± 0.07355192,so it is approximately 1 standard deviation of my mean cross-validation accuracy from (c) (ii).