

SVM

Instructed by Dr.Azimifar

# SVM

A supervised machine learning model called a support vector machine (SVM) employs classification techniques to solve two-group classification problems. An SVM model can classify new text after being given sets of labeled training data for each category.

They offer two key advantages over more recent algorithms like neural networks: incredible speed and improved performance with fewer samples (in the thousands). As a result, the approach is excellent for text classification issues, where it's typical to only have access to a dataset with a few thousand tags on each sample.

This part aims to classify two datasets with SVM with linear and non-linearly.

## Impelemt linear SVM

To implement linear svm, *SVMclassifier* class has been created. This class has a constructor method to initiate the “ C, W, and b “ parameters. After that, we have the *Calcloss* method to calculate the amount of our loss or error.

To train our model, we have a *fit* method, which gets X, Y, size, alpha, and iterations as arguments and will introduce our linear SVM model based on mathematic formulas.

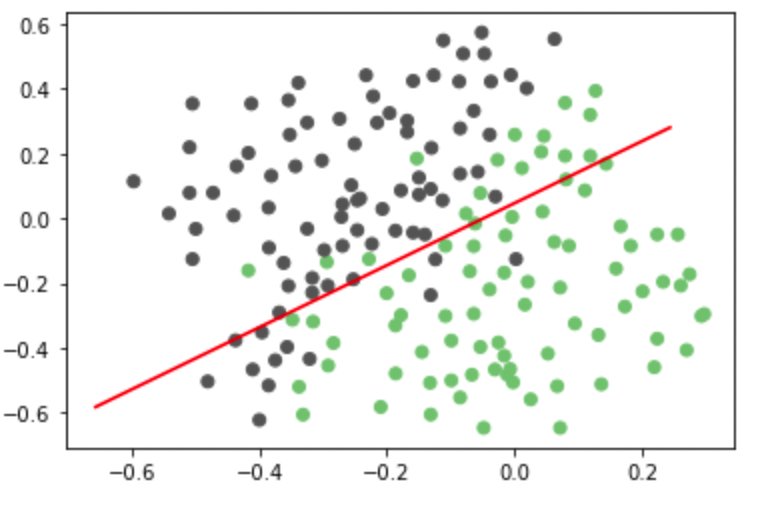
In the end, it returns weights and bias, and losses.

We use the *hypothesis* function to predict the labels of the data; this function uses the sigmoid method to predict.

We used numbers 1, 100, and 1000 for C hyperparameters to model our train data.

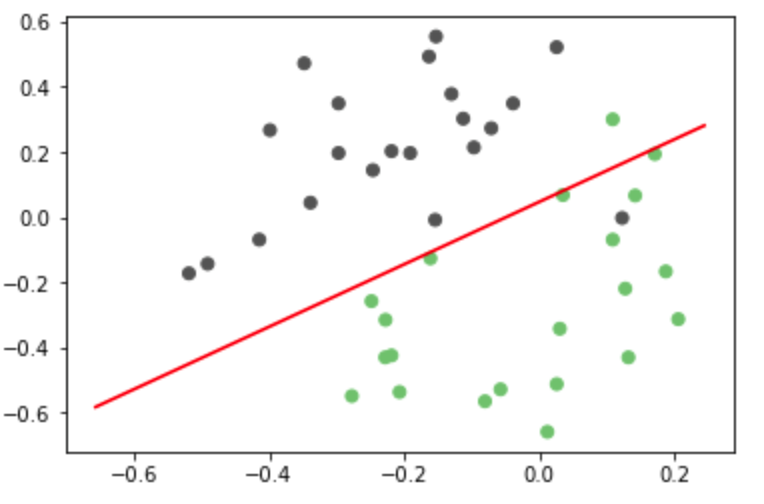
The results are as follows

### C = 1



C=1 train plot

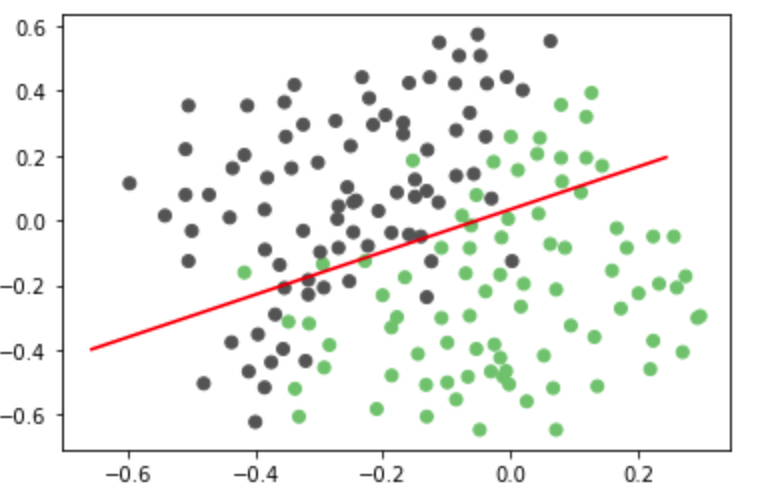
The accuracy for c1 in training data is 83.43 %



C=1 test plot

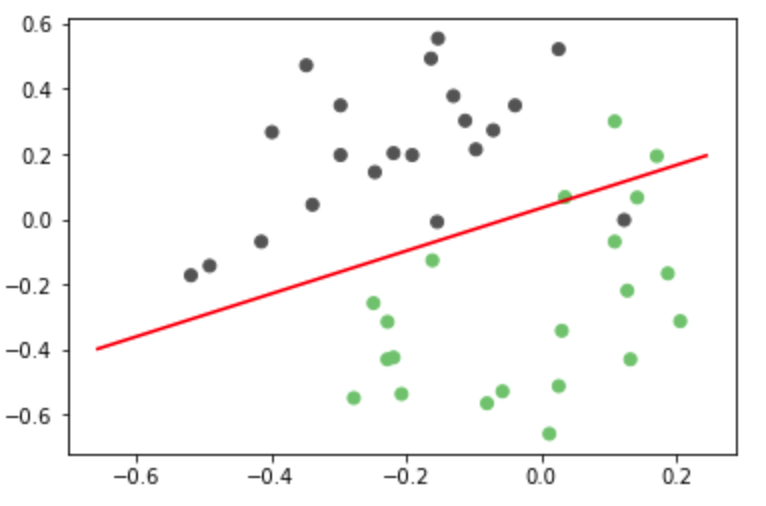
The accuracy for c=1 in test data is 95.23% %

### C = 10



C=10 train plot

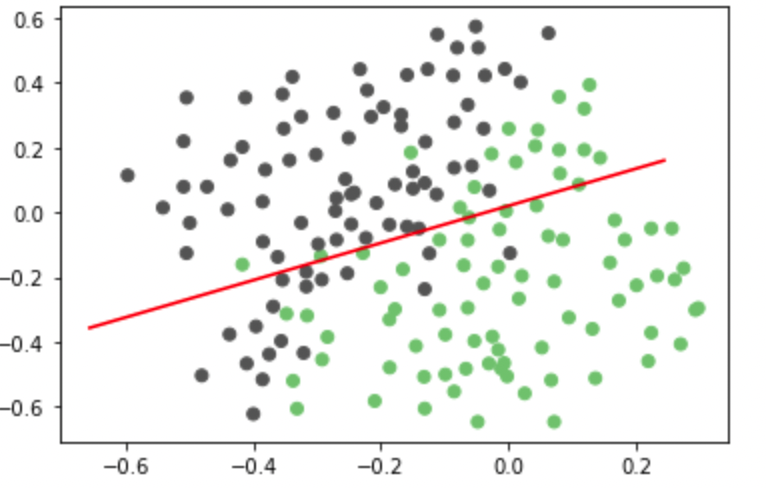
The accuracy for c=10 in training data is 79.28 %



C=10 test plot

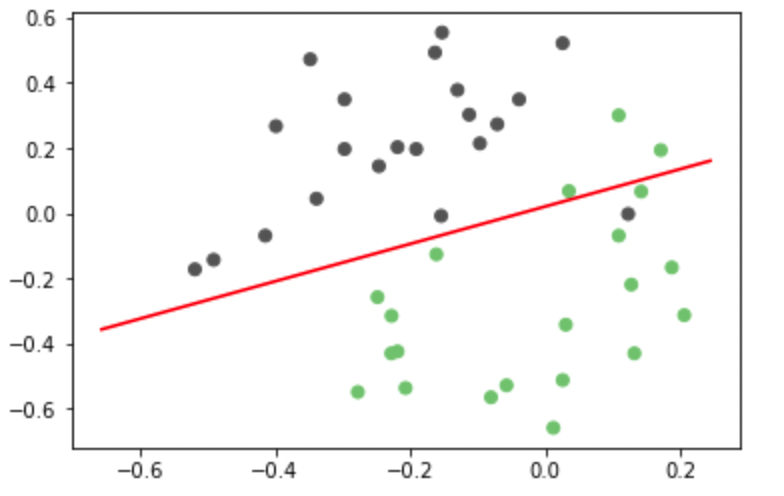
The accuracy for c=10 in test data is 90.47 %

### C = 100



C=100 train plot

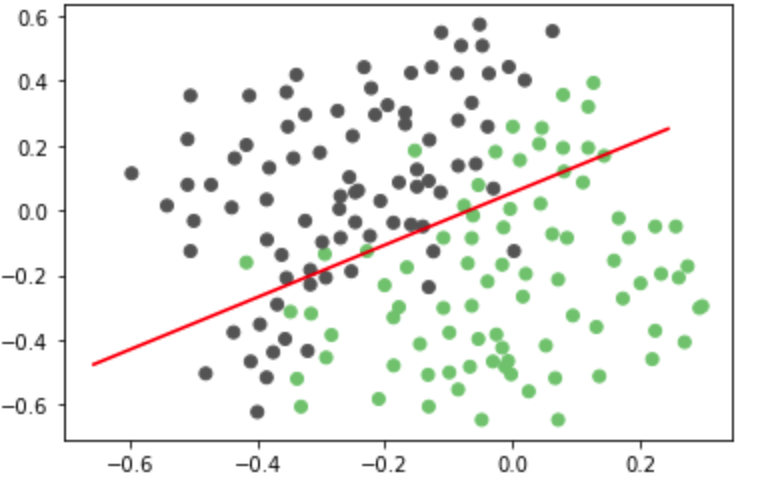
The accuracy for c=100 in training data is 78.69 %



C=100 test plot

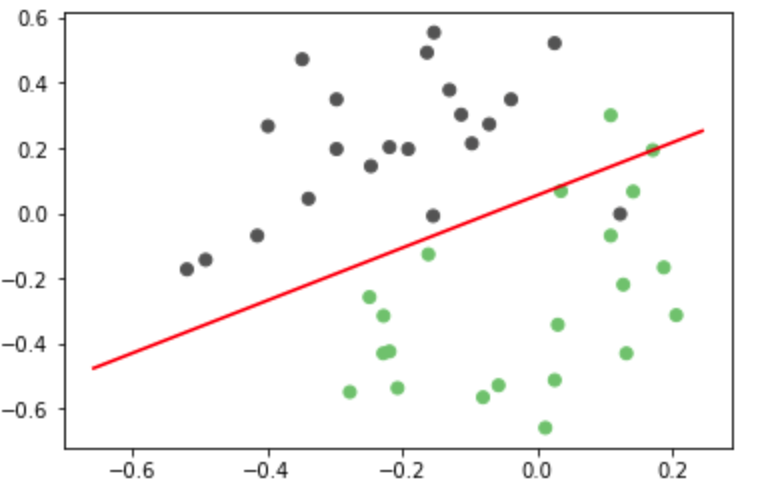
The accuracy for c=100 in test data is 90.4 %

### C = 1000



C=1000 train plot

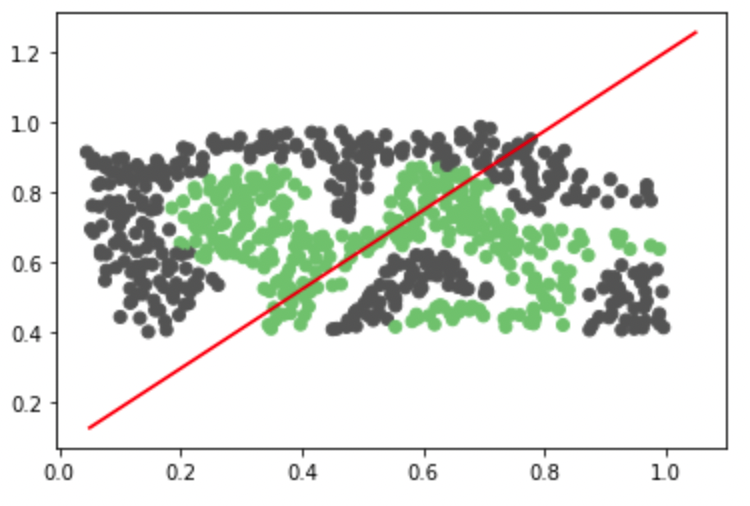
The accuracy for c=1000 in training data is 81.06 %



C=1000 test plot

The accuracy for c=1000 in test data is 92.85 %

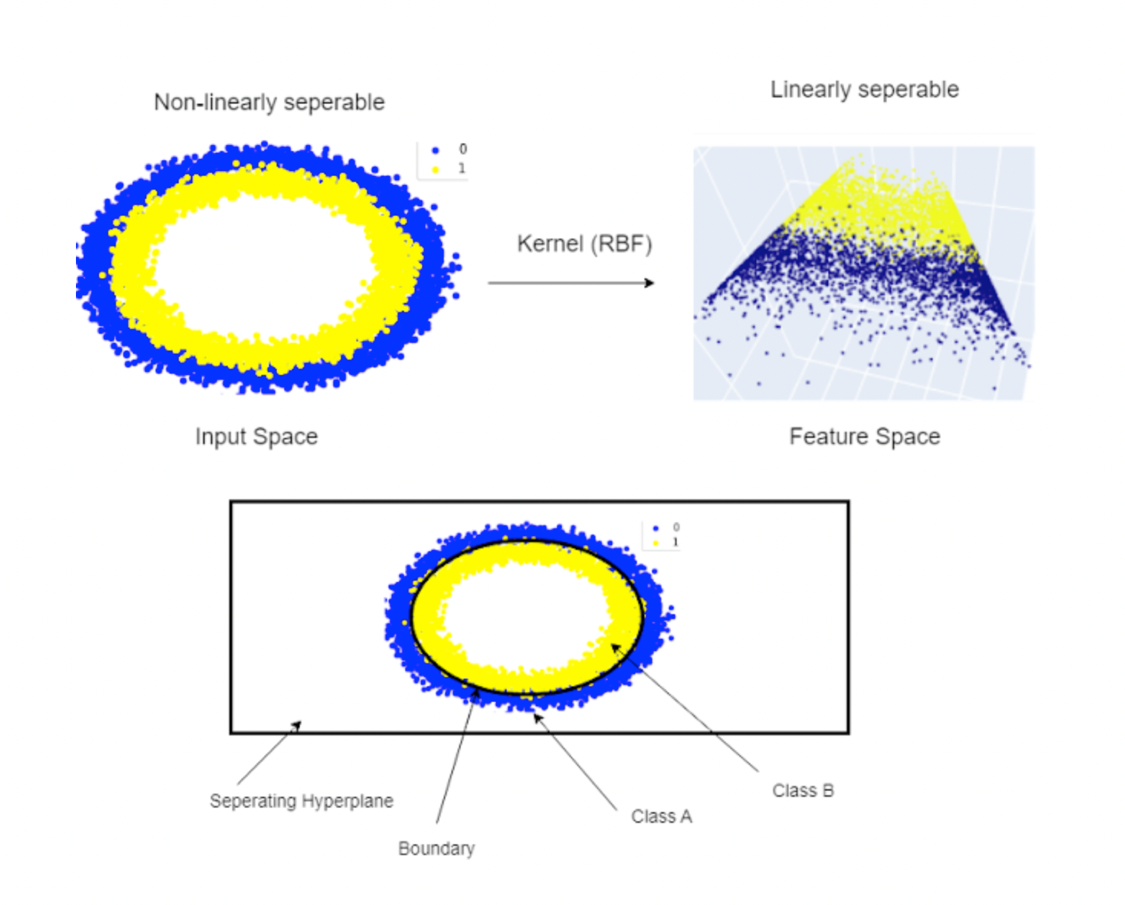
### Linear SVM for Dataset2



Linear SVM is not an excellent way to use our data, so we use non-linear SVM.

## Implement non-linear SVM

RBF, short for **Radial Basis Function Kernel,** is a very powerful kernel used in SVM. Unlike linear or polynomial kernels, RBF is more complex and efficient because it can combine multiple polynomial kernels multiple times of different degrees to project the non-linearly separable data into higher dimensional space so that it can be separable using a hyperplane.



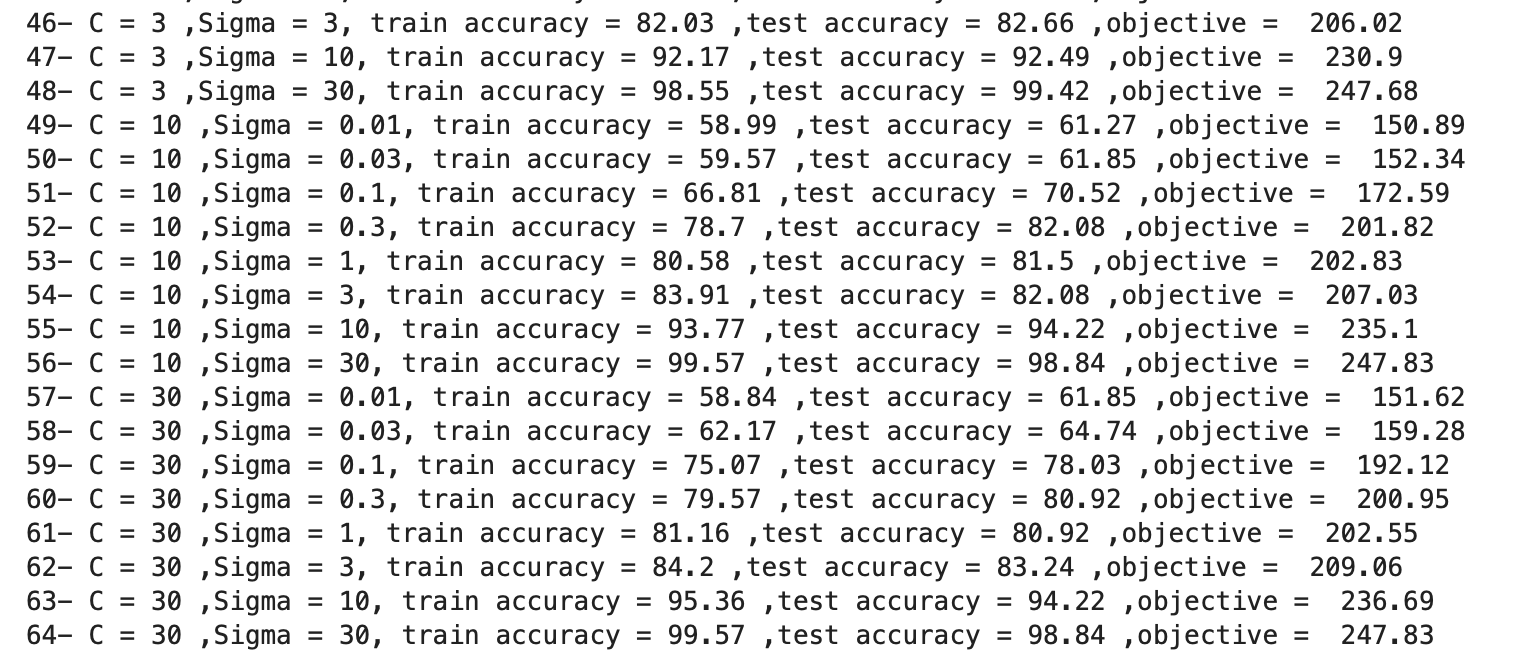
To implement the RBF kernel SVM, the *SVM*  class has been created. This class has a constructor method to initialize “C, kernel, sigma, iterations number.”.

The class uses the CalcLoss method to calculate the loss in our model. And like any other model, it has a *fit* method to train our model, and it will calculate the weights, bias, and loss.

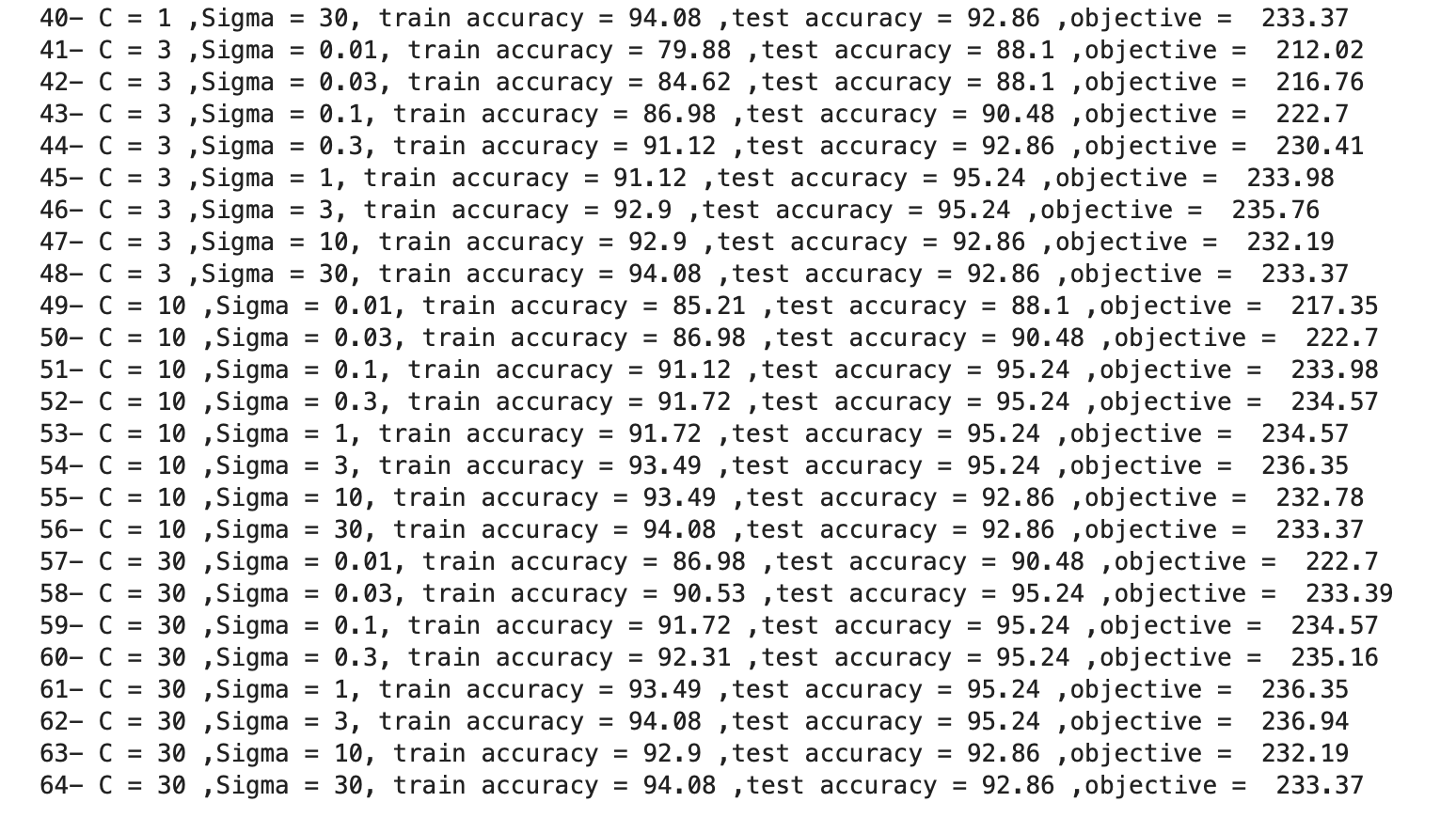
The prediction method predicts the labels of data and uses the *decide* model.

### Get parameters

We used numbers “0.01,0.03,0.1,0.3,1,3,10,30” for our C and sigma, and we used our data to model on each c and sigma to get the best number for c and sigma for each dataset.



Example of calculating the best sigma and C for Dataset2



Example of calculating the best sigma and C for Dataset1

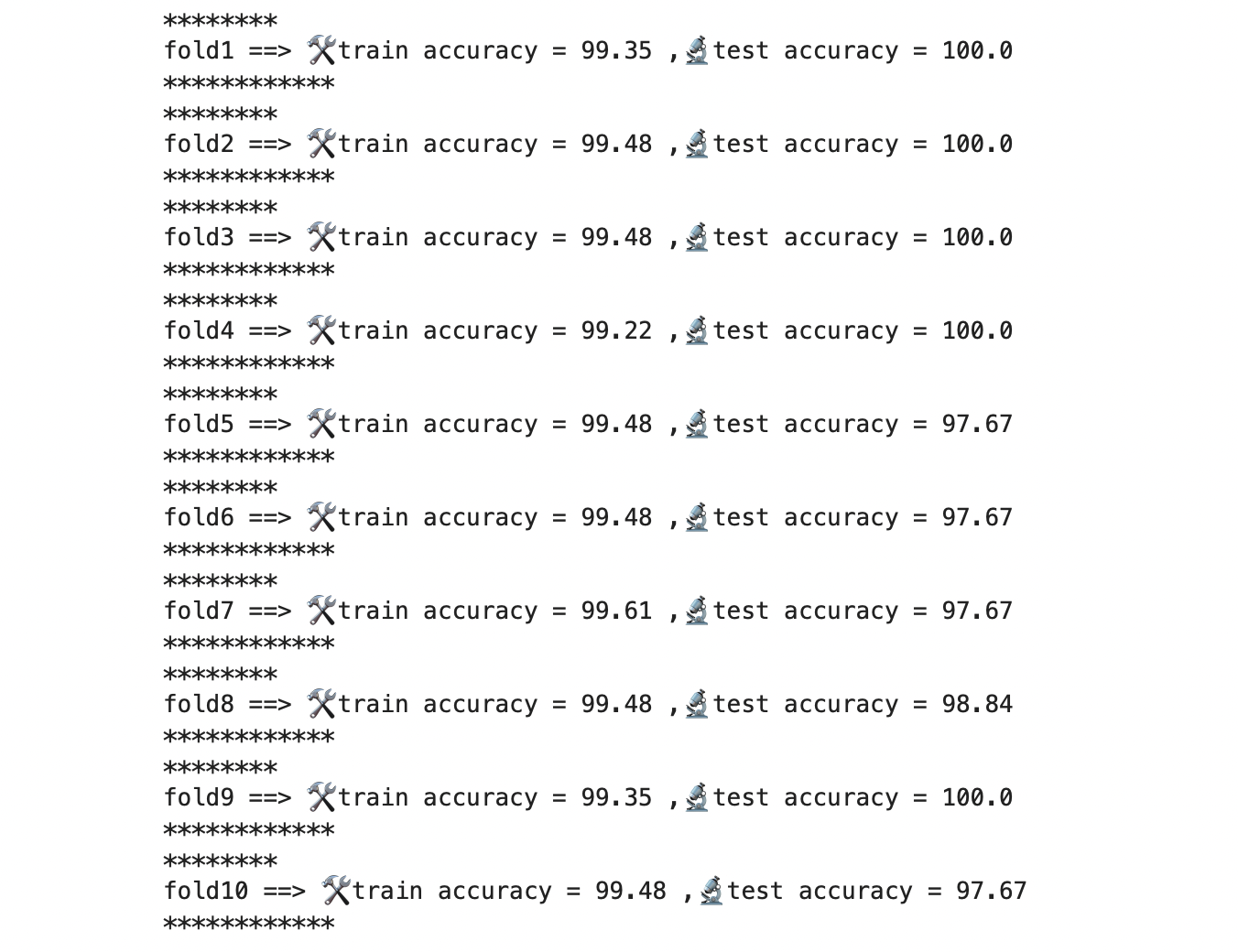
The best C and sigma have been used to run a ten fold cross-validation on both datasets. The accuracies are as follows:

Dataset one, best sigma and C k fold :



Dataset1, best sigma, and C 10 folds

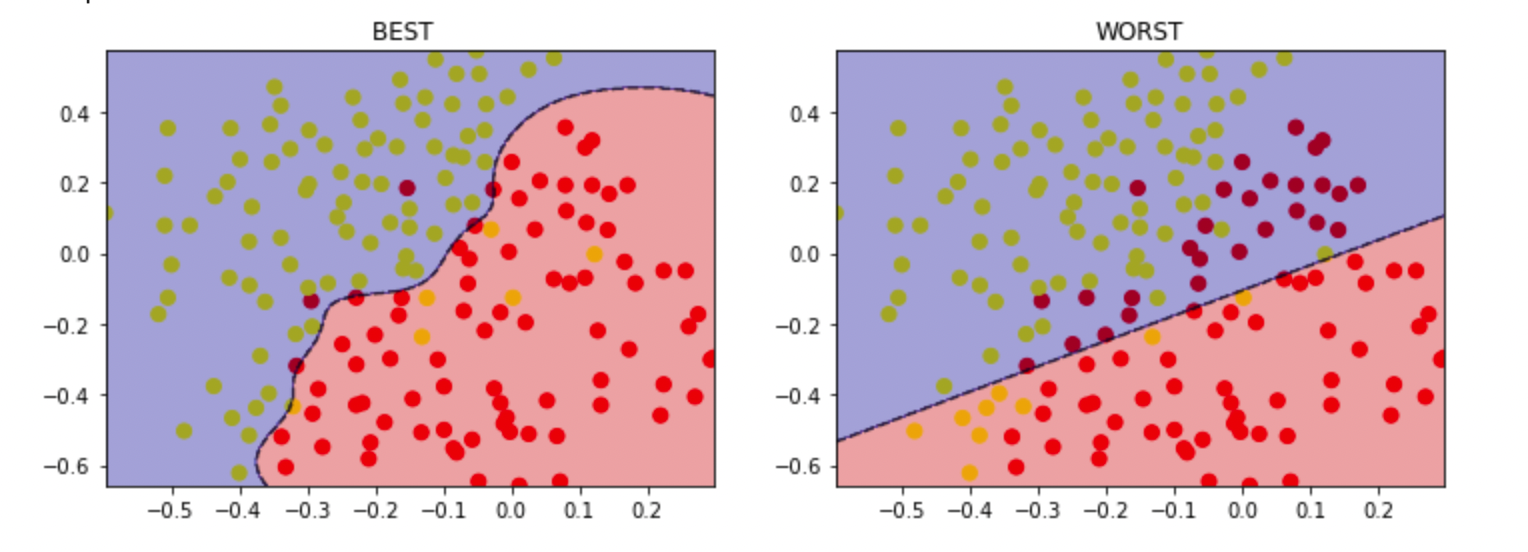
Dataset two with the best sigma and C k fold :



Dataset2, best sigma, and C 10 folds

### Plot

Best and worst model ( C, sigma ) for Dataset1:



Best and worst model ( C, sigma ) for Dataset2:

