Name: Mehboob Ali

Roll Number: 17L-4316

Section: A

Classifying Iris Dataset using Support Vector Machines (SVMs)

I. Introduction

Support Vector Machines are one of the most commonly used supervised learning algorithms used for classification task. SVMs perform classification in such a way that the margin between two data points, closest to the decision boundary, is maximum. The magnitude of the margin could be changed from soft to hard using the parameter ‘C’. The value of C indicates how much we want our model to penalize the misclassification. If the value of C is too large, then the model would heavily penalize misclassifications and separate data using hard margins which might result in over-fitting (high-variance). On the other hand, if the value of C is too small, then the model would separate data using soft margins. Thus, the model would allow misclassifications and might be under-fitting (high-bias). ‘C’ is a hyper-parameter whose value could be optimized using cross-validation.

There exist non-linear SVMs as well which could be used to classify non-linear datasets. There are two hyper-parameters for these models: ‘C’ and gamma. Value of gamma determines the complexity of the model. The smaller the gamma value, the simpler the model would be. Consequently, it could lead to under-fitting and a lower training and test accuracy. Conversely, if the value of gamma is too large, the model would try to memorize patterns and it would result in overfitting. Hence, it would not generalize well on unseen data. The hyper-parameters, C and gamma, can be optimized using cross-validation as well.

II. Methodology

We have used SVM with linear and non-linear kernels for the classification task. Firstly, SVM with linear kernel was used. It only requires optimization of the ‘C’ value. The best ‘C’ value was obtained after k-fold cross validation with k=5 and k=10. The best mean test cross-validation scores were obtained and based on that the final ‘C’ was chosen. Finally, the model was trained using it and tested on unseen data.

There are different non-linear kernels which could be used for SVM. We used the Radial Basis Function (rbf) kernel for the classification task as mentioned in the assignment document. It has two hyper-parameters which can be optimized. These include ‘C’ and gamma values. The best ‘C’ and gamma values were obtained using k-fold cross validation with k=5 and k=10 as well. Two different k values were tested to find the appropriate number of folds for this task.

III. Dataset

The Iris plants dataset will be used to train, test and evaluate the two different classifiers. The dataset cont ains around 150 samples. There are three different labels/classes in the dataset including Iris-Setosa, Iris-V ersicolor and Iris-Virginica. Each class has 50 labels so the data is evenly spread as shown in table 1.

Table 1: Number of sample in each class

Class	Count
Iris-Setosa,	50
Iris-Versicolour	50
Iris-Virginica	50

IV. Evaluation and Results

For both, linear and non-linear SVMs, the dataset was split on 80:20 ratio to obtain the training and testing datasets. The test dataset was preserved for the final evaluation while the training dataset was used for training and cross-validation.

A. Linear SVM

a. K-Fold Cross-Validation Results

A range of ‘C’ values, from 0 to 100, were defined to be used during the K-fold cross validation. First, the value of k was chosen to be 5, i.e. the number of folds. The cross-validation results are shown in Fig. 1. The best mean test accuracy obtained through cross-validation was 0.9769 corresponding to C value of 0.75 and k = 5.

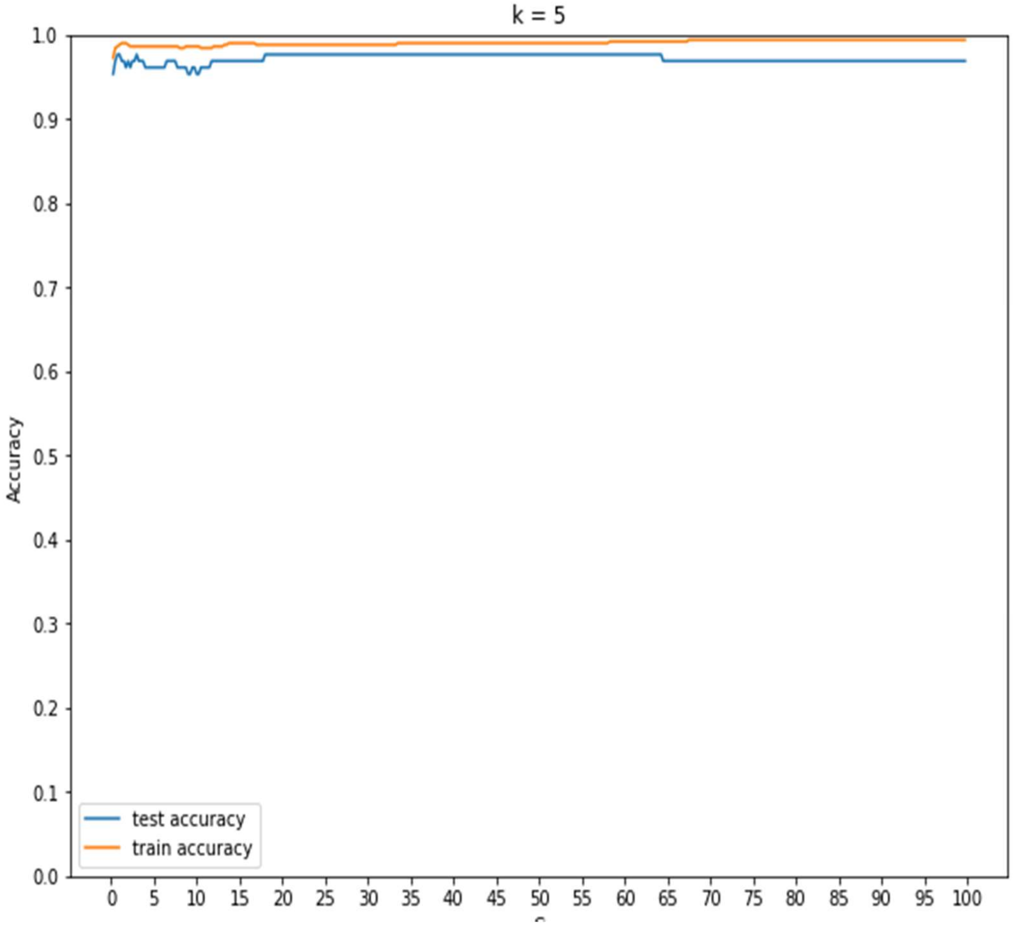


Figure 1: Cross-validation accuracies using k=5

Afterwards, the K-fold cross validation was performed with k = 10. The cross-validation results are shown in Fig. 2. The best mean test accuracy obtained through cross-validation was 0.99230 corresponding to C value of 0.5 and k = 10.

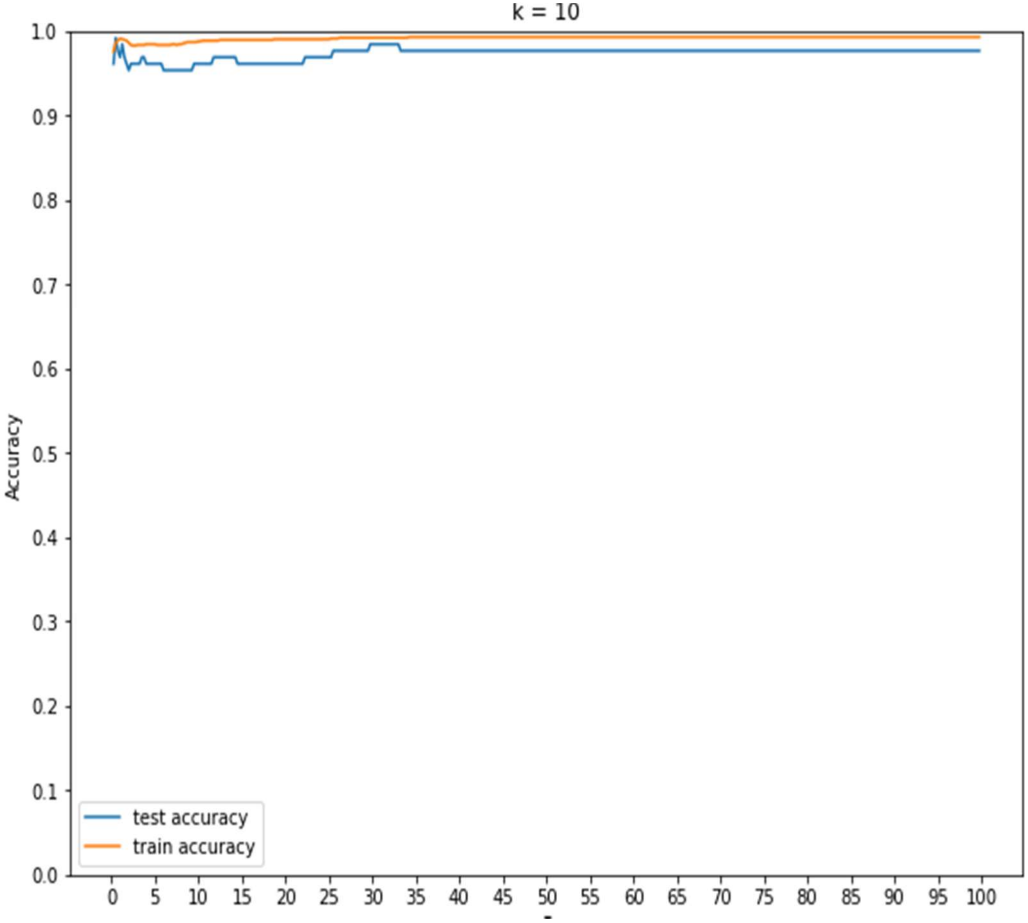


Figure 2: Cross-validation accuracies using k=10

b. Test Dataset Results

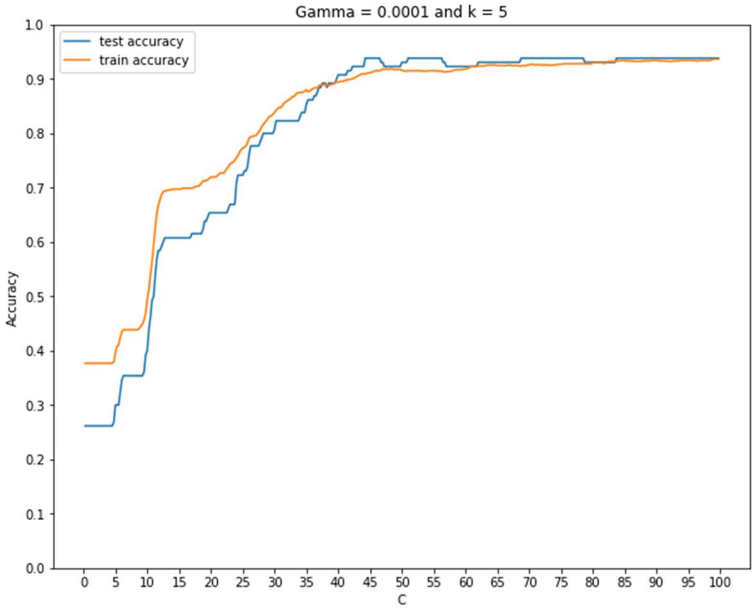
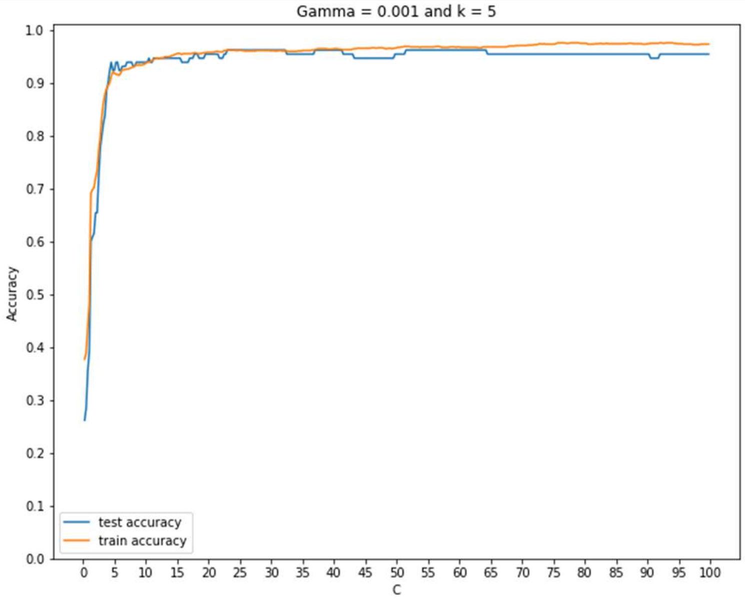
The models were then re-trained using the optimized 'C' values. The model which was cross-validated using k=5 gave the final accuracy of 100% on the test dataset. However, the model which was cross-validated with k=10 had one misclassification. Hence, its final accuracy was 95% on the final unseen dataset.

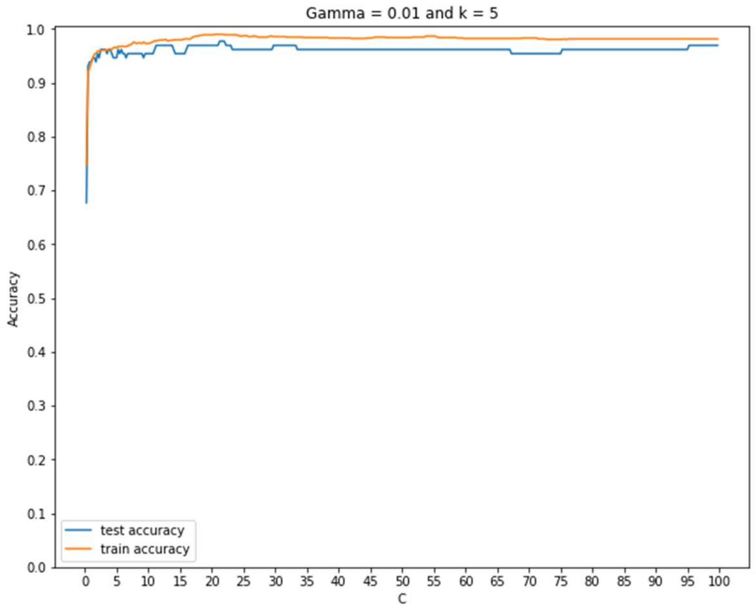
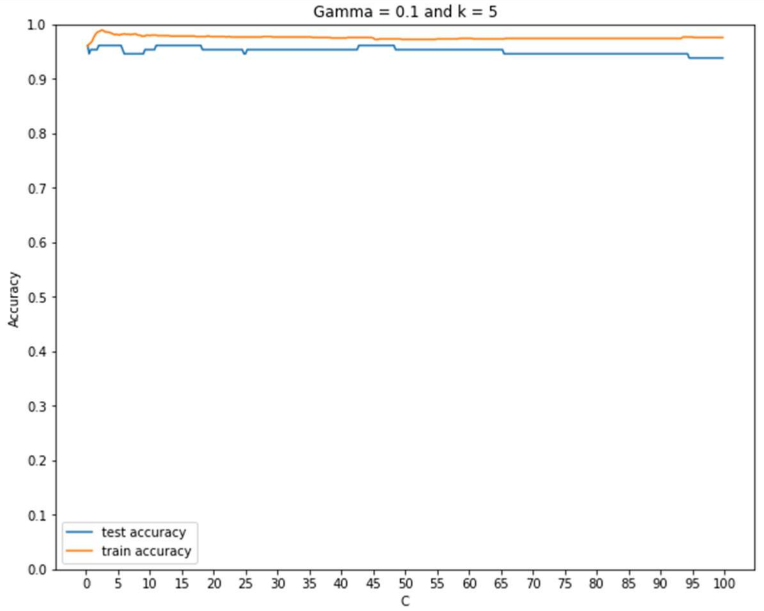
B. *Non-Linear SVM using 'rbf' kernel*

a. *K-Fold Cross Validation Results*

A range of 'C' and gamma values, were defined to be used during the K-fold cross validation. First, the value of k was chosen to be 5, i.e. the number of folds. Several cross-validation results were obtained using different values of gamma. These results are shown in table 2. The best mean test accuracy obtained through cross-validation was 0.9769 corresponding to C value of 7.5, gamma value of 0.1 and k = 5.

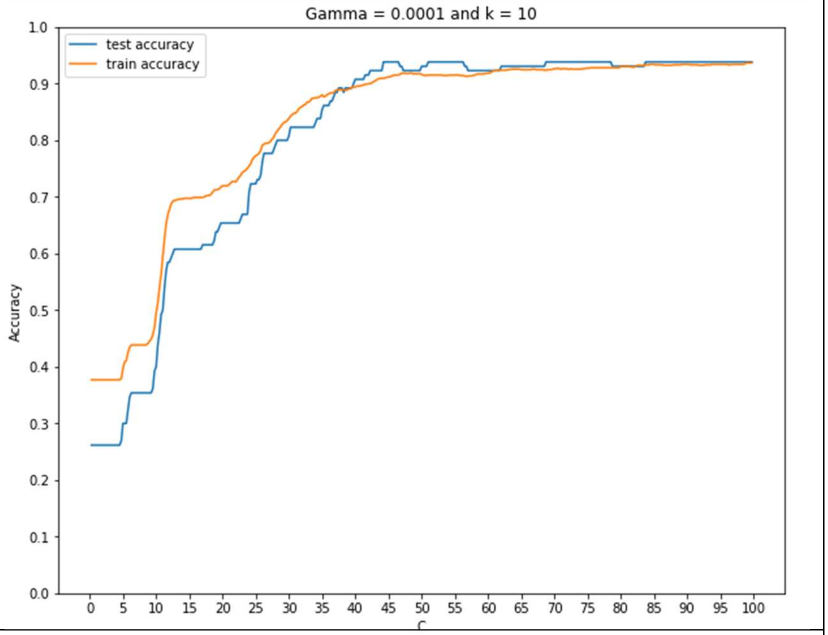
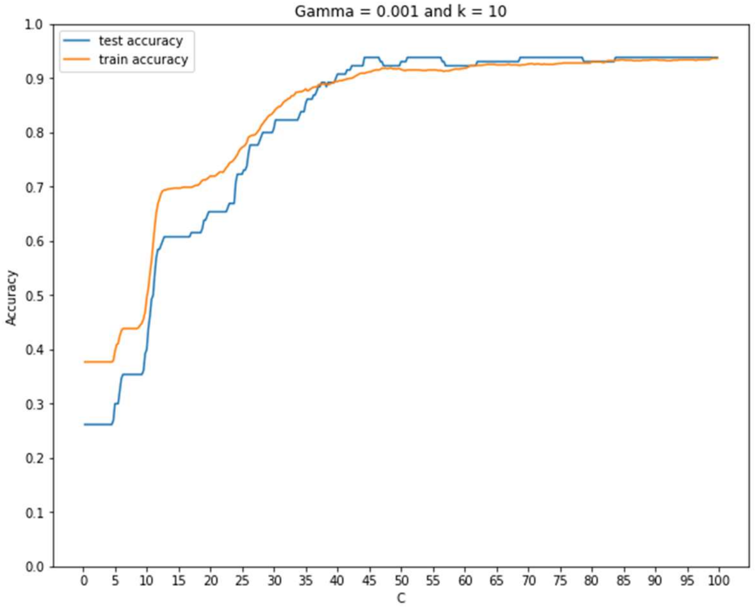
Table 2: Cross-validation results with k=5 using non-linear SVM

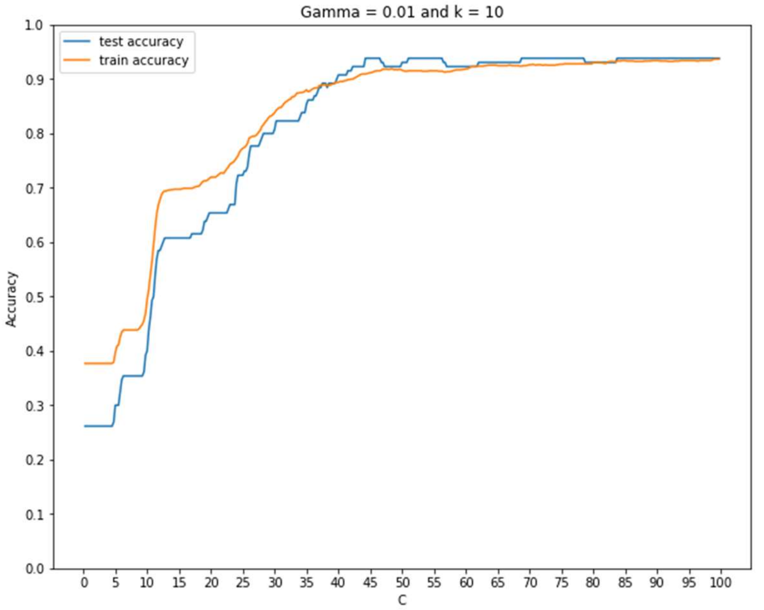
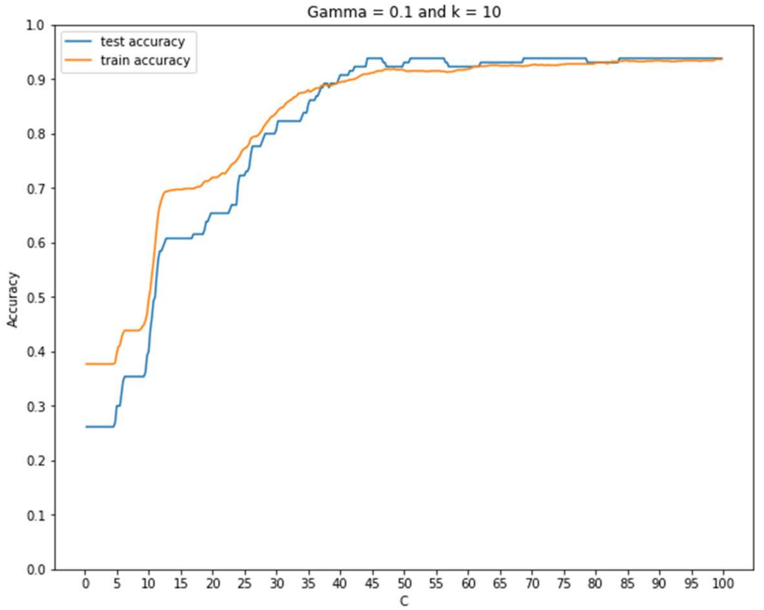
Experimental Setting	**Cross-Validation Results**
Gamma = 0.0001 and K = 5. Value of C ranges from 0 to 100.	
Gamma = 0.001 and K = 5. Value of C ranges from 0 to 100.	

Gamma = 0.01 and K = 5. Value of C ranges from 0 to 100.	
Gamma = 0.1 and K = 5. Value of C ranges from 0 to 100.	

Afterwards, the K-fold cross validation was performed with k = 10. The cross-validation training and test results are shown in table 3. The best mean test accuracy obtained through cross-validation was 0.9846 corresponding to C value of 23.5, gamma value of 0.01 and k = 10.

Table 3: Cross-Validation results using k=10 and non-linear SVM

Experimental Setting	**Cross-Validation Results**
Gamma = 0.0001 and K = 10. Value of C ranges from 0 to 100.	
Gamma = 0.001 and K = 10. Value of C ranges from 0 to 100.	

Gamma = 0.01 and K = 10. Value of C ranges from 0 to 100.	
Gamma = 0.1 and K = 10. Value of C ranges from 0 to 100.	

b. *Test Dataset Results*

The models were then re-trained using the optimized ‘C’ and gamma values. The model which was cross-validated using k=5 gave the final accuracy of 90% on the test dataset due two misclassifications. The model which was cross-validated with k=10 had two misclassifications as well. Hence, its final accuracy was 90% on the final unseen dataset.

V. Visualizing the effect of ‘gamma’ on SVM decision boundary

The best value of gamma has been determined using cross-validation as described in the previous sections of the report. The decision boundary using the optimized C and gamma is shown in Fig. 3.

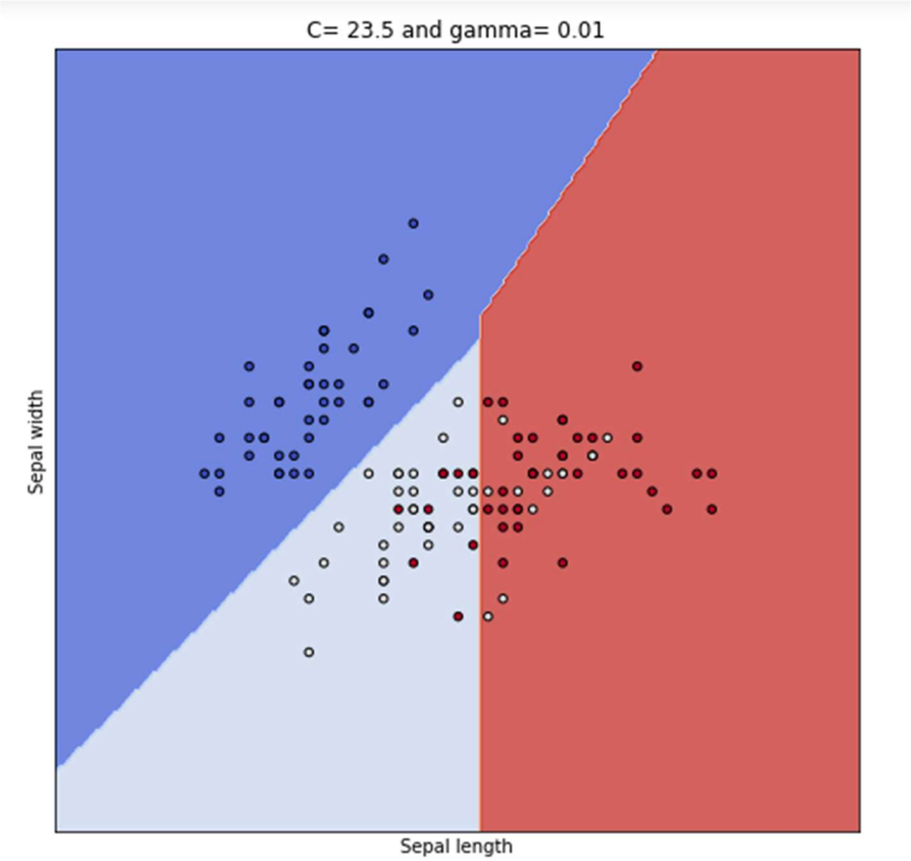


Figure 3: Decision Boundary using optimized C and Gamma value

Now, we will see how the decision boundary of the non-linear SVM classifier changes due to change in gamma values while keeping C same. Firstly, smallest gamma value will be chosen. If the gamma value is small, then the model will be simple and would allow misclassifications. Thus, it could lead to poor training and test accuracies. The decision boundary drawn due to smallest gamma value is shown in Fig. 4 below.

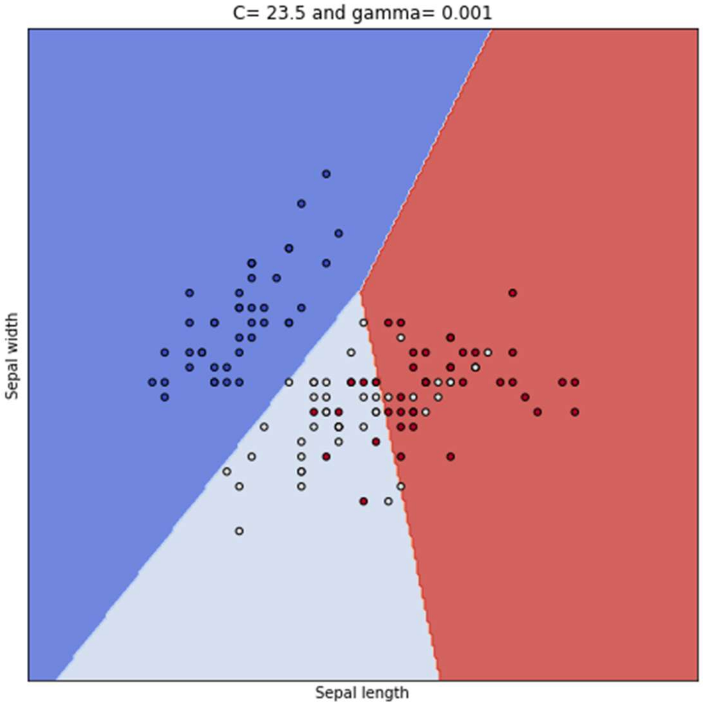


Figure 4: Decision Boundary using smallest gamma value

Then, the model was trained with the largest gamma value to determine its decision boundary. As shown in Fig. 5, it results in over-fitting as model becomes very complex and tries to separate data using hard margins. The consequence of over-fitting is that the model will have high training accuracy but low test accuracy. Hence, it will not generalize really well.

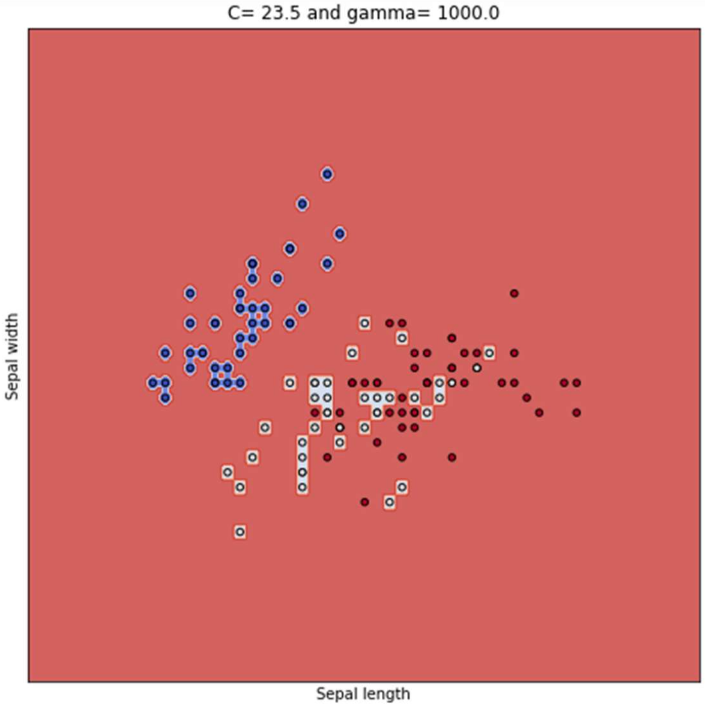


Figure 5: Decision Boundary using largest gamma value

VI. Conclusion

Linear and non-Linear SVM models were used on the Iris plan dataset. K-Fold cross-validation was performed on both models for optimization of hyper-parameters and to make sure that the model is trained on all parts of data. After going through evaluation and results provided in the previous section, it could be concluded that for linear SVM, the best value for k-fold cross-validation 5 as it gave an accuracy of 100% on test dataset with C=0.75. For non-linear SVM, k-fold cross validation with k=5 and k=10 had the same accuracy on the test dataset, i.e. 90 percent. However, the cross-validation results using k=10 were much better as compared to k=5. Finally, the dataset was quite small, i.e. only 150 samples, so it might be interesting to see how the models will perform on a larger dataset.