ANALYSIS REPORT

Name: Kumar Raju Bandi

Roll Number: IWM2017502

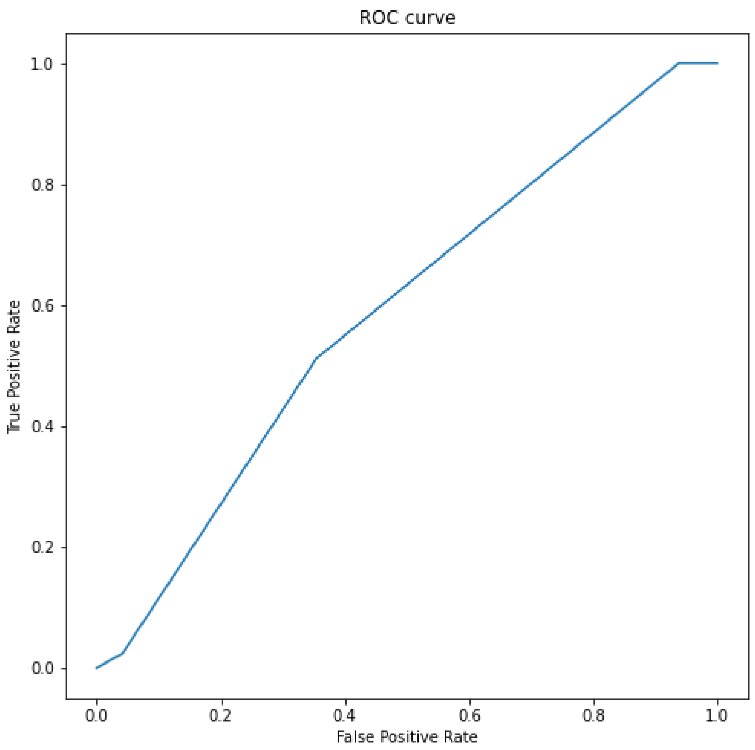
In this assignment, we were asked to implement a soft margin-SVM with RBF Kernel using SMO algorithm.
To train and predict, head disease dataset was used which consisted of 303 samples and 2 features in total were used for this model. The labels were converted to 1 and -1 from 1 and 0 and feature scaling was done for the data samples.

Colab Link :
https://colab.research.google.com/drive/1sbtpcEJpF5fhDWY2BPNsfSBD4pj5h5h9?usp=sharing

The model was implemented as discussed in the class and analysis were drawn during the implementation and testing of the algorithm for the given heart disease dataset.
As given in the problem statement, only two features i.e; column #1 and column #4 were used and for the labels, column #14 was used.
The labels were also converted from {0, 1} to {-1,1}.
Apart from this feature scaling to the range of [-1,1] was applied.
Interestingly, for conversion of labels to -1 and 1, the feature scaling (Normalised [-1,1]) can be applied.

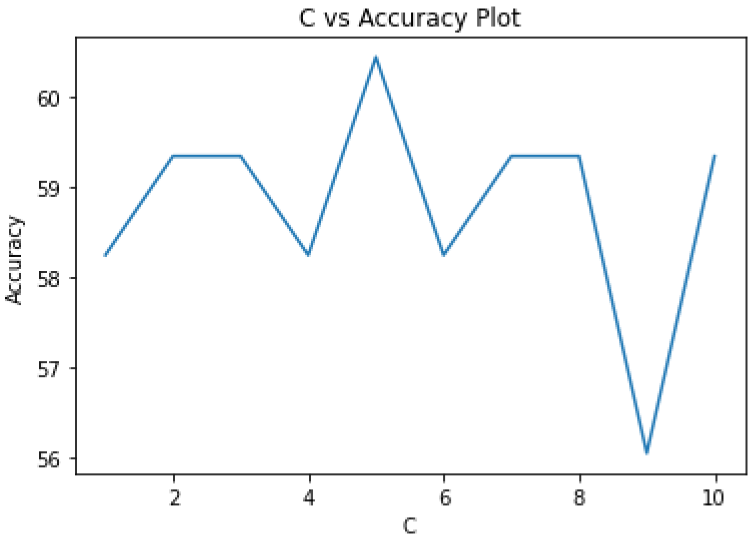
The accuracy achieved on the given dataset was **60.4%**. However, it is to be noted that the accuracy differs with every run of the python notebook as the convergence to the optimal hyperplanes happens differently for each run.

Also, the ROC curve was plotted for the above model and which is as below:



Further analysis was performed to find the optimal values of the hyperparameters.

Firstly, below is the C vs accuracy plot for SVM with SMO model implemented:

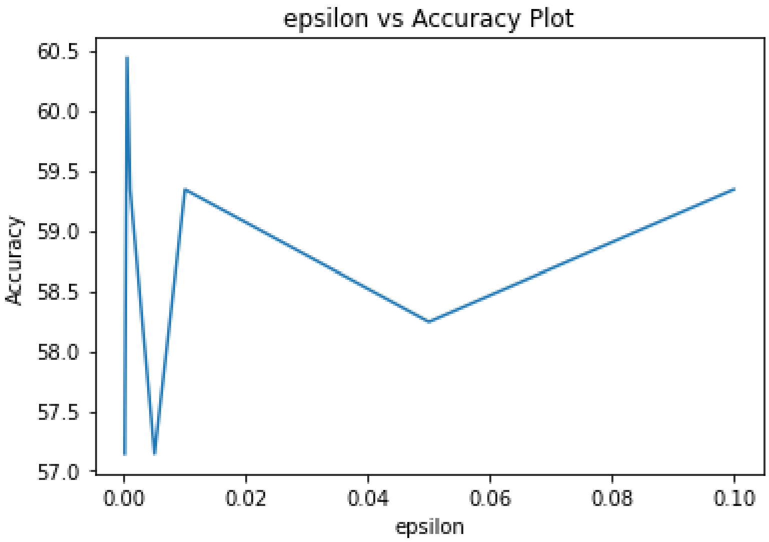


It can be observed from the above plot that for in the range of 0 to 10, the value of 5 was giving better accuracies.

Smaller values of C means the model is going to look for large-margin separating hyperplanes and larger values of C means the model is going to look for small-margin separating hyperplanes.

Furthermore I also observed the accuracy for even higher values of C which are reported below.

Also, I have analysed by changing the value of epsilon.
below is the epsilon vs accuracy plot for SVM with SMO model implemented:

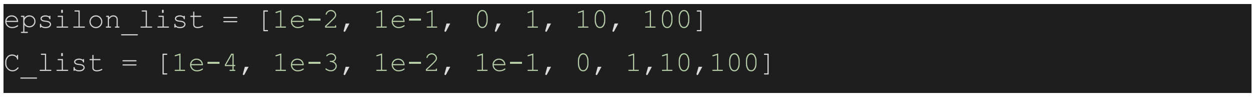


It can be observed from the above plot that for in the range of 0 to 10, the values of epsilon closer to 0 were giving better results.

Smaller values of epsilon means we will be penalizing the samples more and which will thereby lead to more support vectors and on the other hand, larger values of epsilon means larger errors will get admitted in the solution.

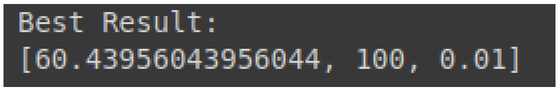
I also did hyperparameter tuning by using the grid method and the results are as follows:

Values used for the grid:



```
epsilon_list = [1e-2, 1e-1, 0, 1, 10, 100]
C_list = [1e-4, 1e-3, 1e-2, 1e-1, 0, 1,10,100]
```

Result:



```
Best Result:
[60.43956043956044, 100, 0.01]
```

All the accuracies for various combinations of C and epsilon can be seen in the notebook.