COL341 - Assignment 2
Linear Regression

Simran Mahawar- 2020CS10387

1. Datasets

2. Kernels

Implemented the following kernels for using kernel tricks along with SVMs in the kernel.py.

- Linear Kernel
- Polynomial Kernel
- Radial Basis Function (RBF) Kernel
- Sigmoid Kernel
- Laplacian Kernel

3. Binary SVM

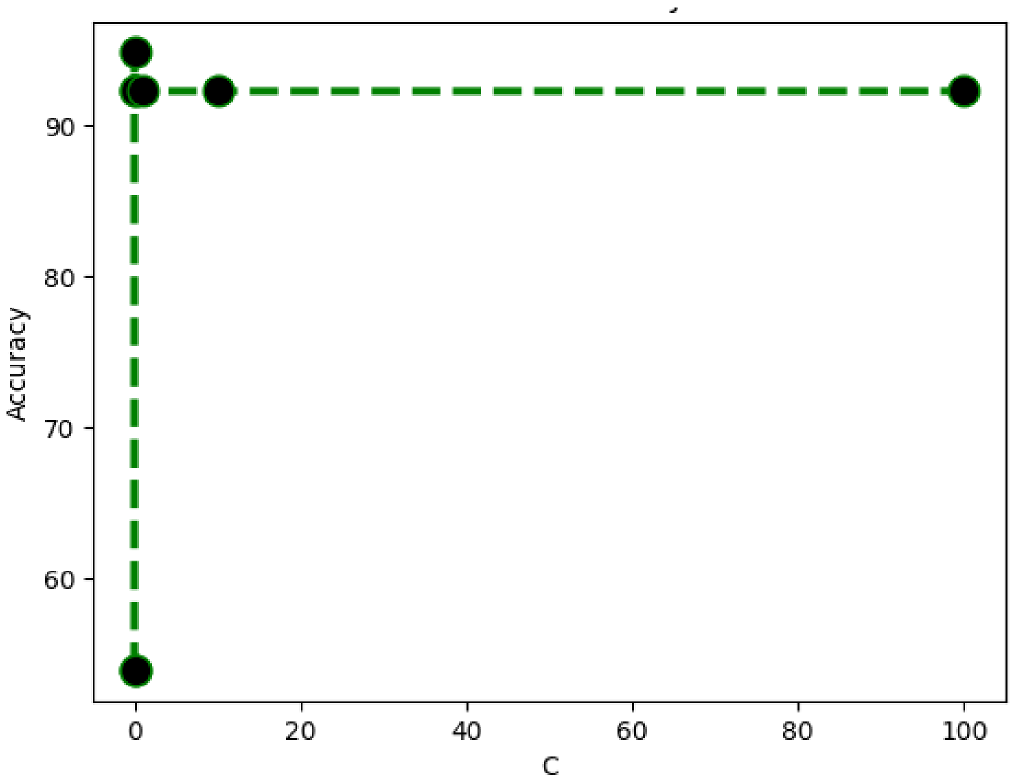
Implementation

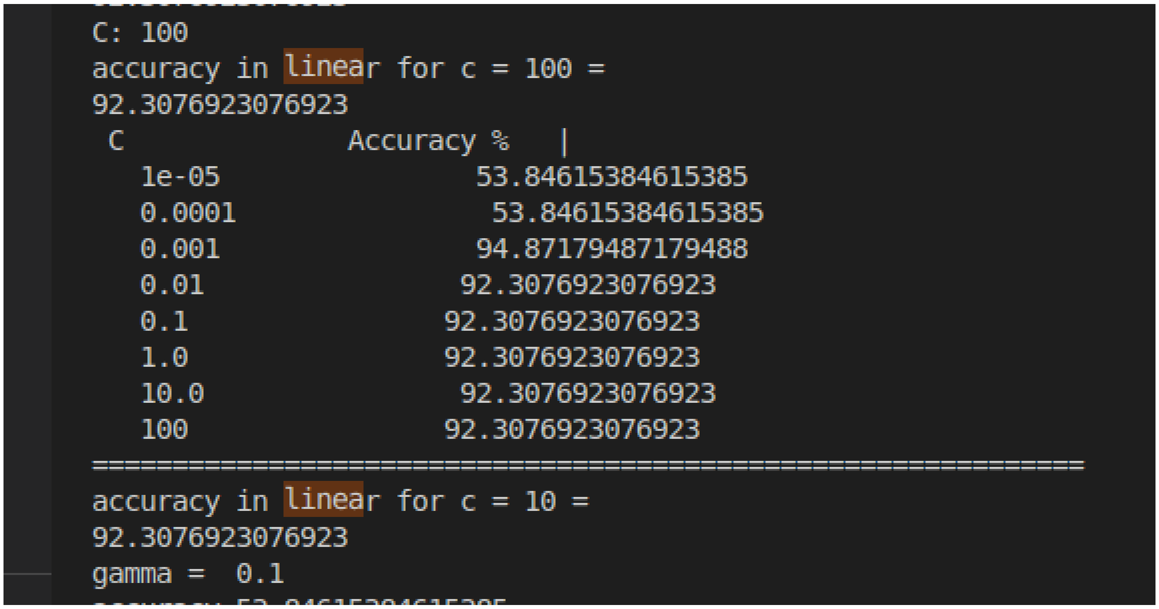
Implemented the standard soft margin support vector machine from scratch. Completed the Trainer.fit() and Trainer.predict() functions in svm sm.py.

Analysis

In SVM, the regularization parameter C controls the trade-off between maximizing the margin and minimizing the classification error on the training set. Smaller values of C result in wider margins but may allow more misclassifications, while larger values of C result in narrower margins and fewer misclassifications.

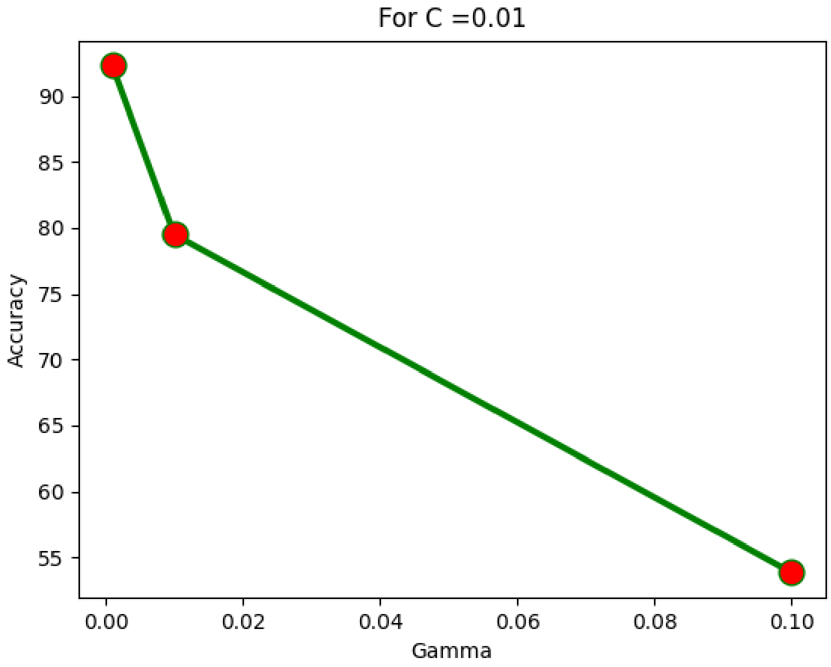
In linear Classification
Best accuracy is 94.87179487179488 on c = 0.001

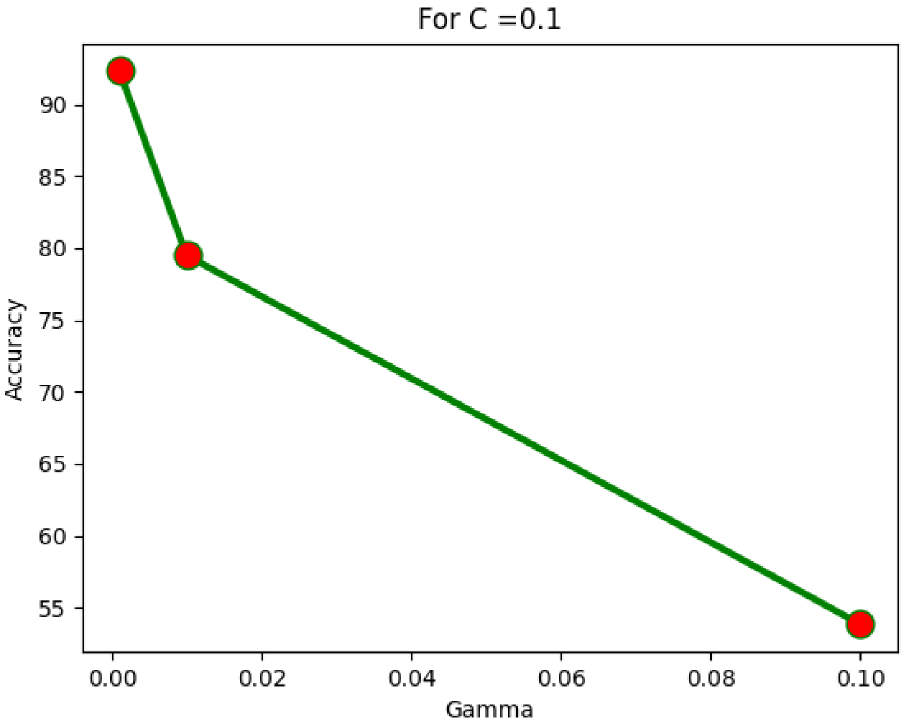


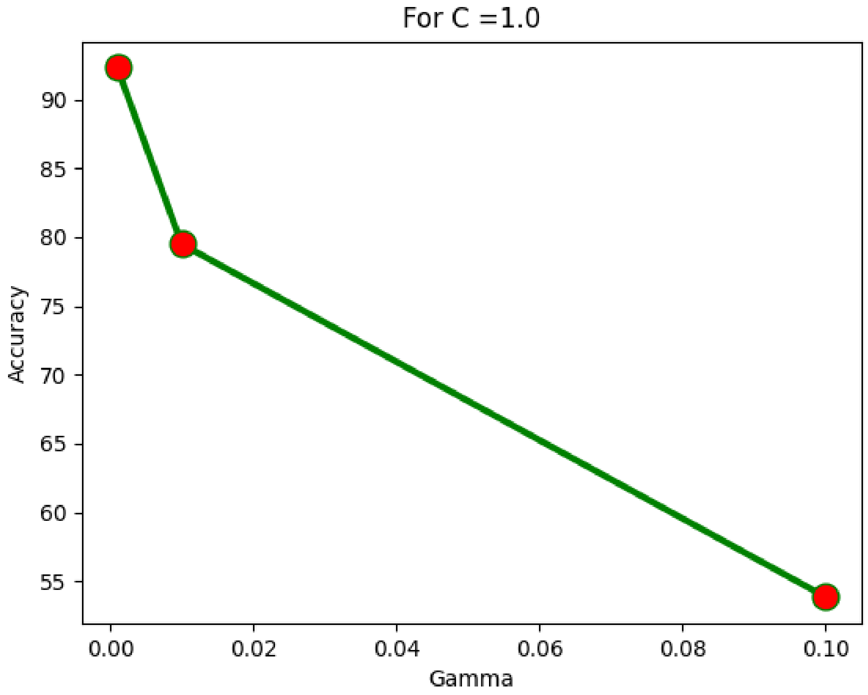


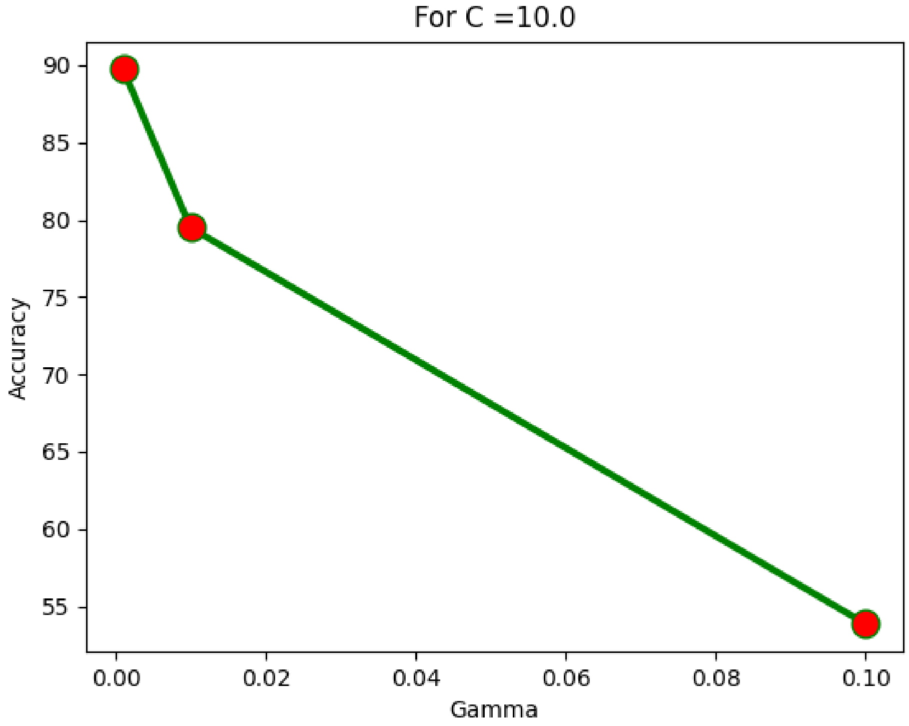
```
C: 100
accuracy in linear for c = 100 =
92.3076923076923
 C               Accuracy %   |
   1e-05                 53.84615384615385
   0.0001                 53.84615384615385
   0.001                 94.87179487179488
   0.01                 92.3076923076923
   0.1                 92.3076923076923
   1.0                 92.3076923076923
   10.0                 92.3076923076923
   100                 92.3076923076923
==============================================================
accuracy in linear for c = 10 =
92.3076923076923
gamma =  0.1
accuracy 53.84615384615385
```

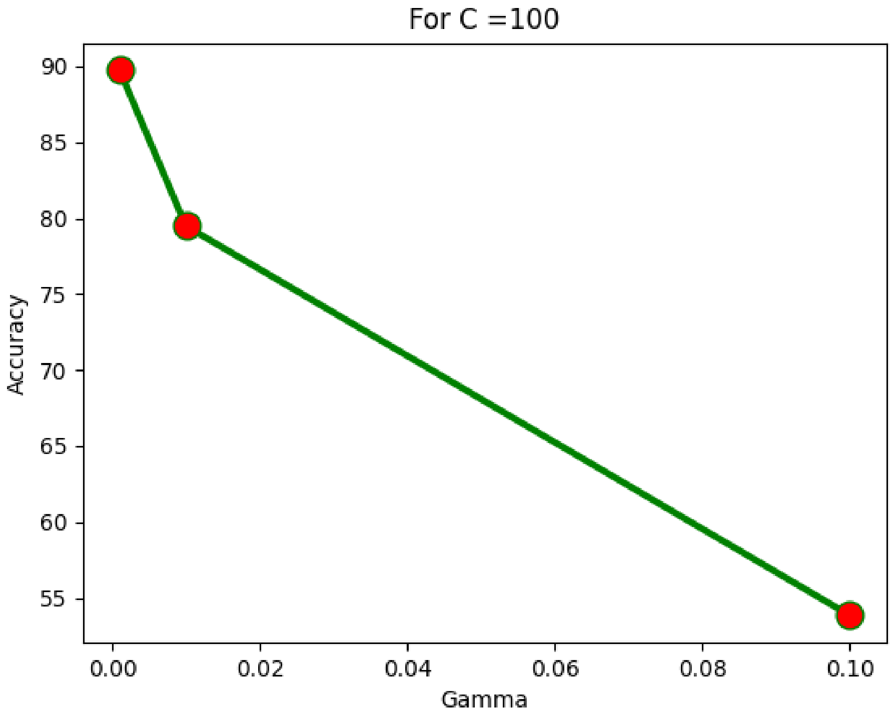
In RBF, here is the analysis
Accuracy is greater when gamma value is less, accuracy decreases with increasing gamma. In SVM, the gamma parameter determines the influence of a single training example. A smaller value of gamma will make the decision boundary smoother, while a larger value will make the decision boundary more complex, potentially overfitting the training data.

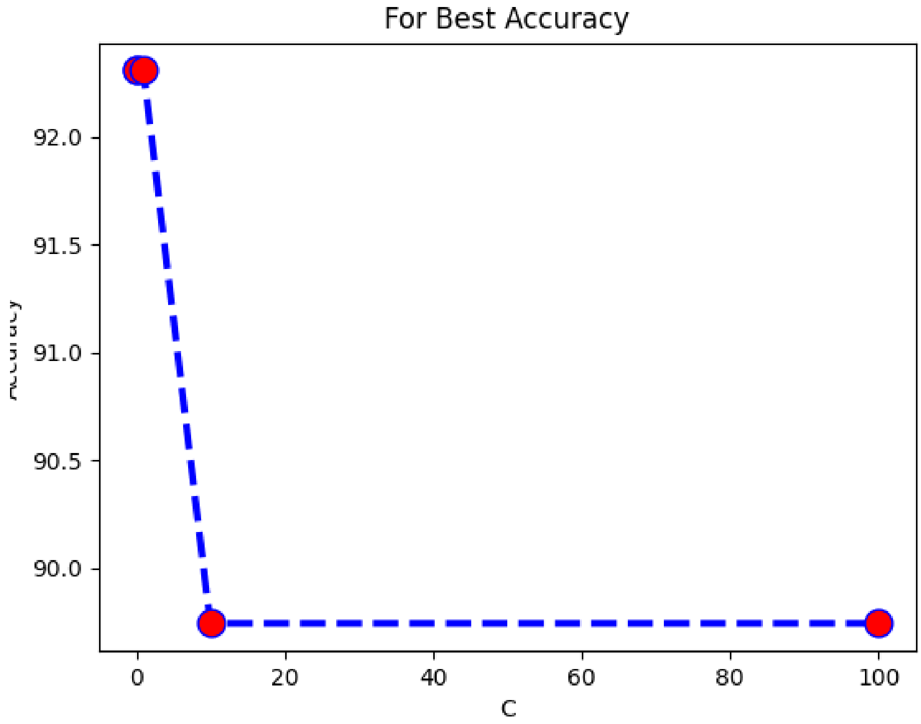

For C =0.01
Accuracy
90
85
80
75
70
65
60
55
0.00
0.02
0.04
0.06
0.08
0.10
Gamma

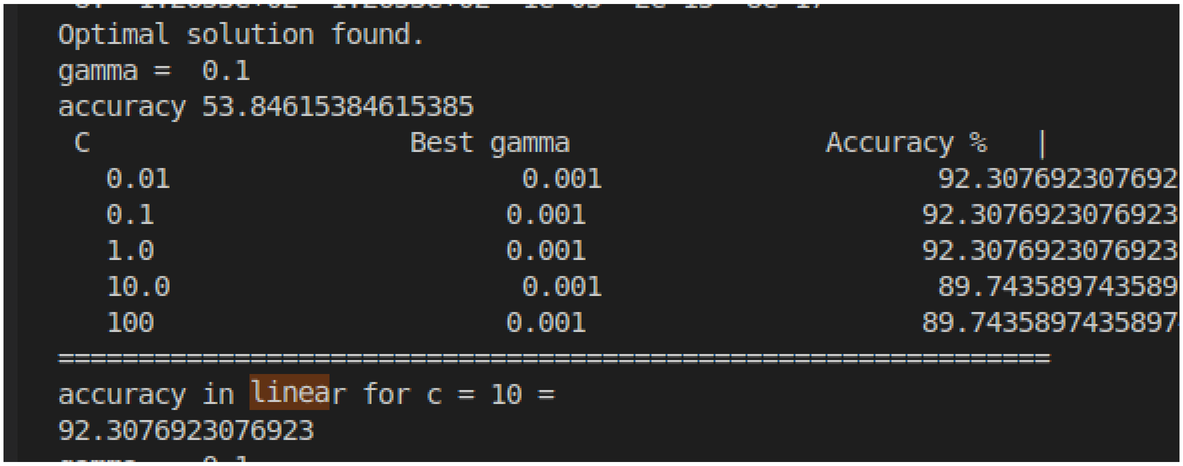

For C =0.1
Accuracy
90
85
80
75
70
65
60
55
0.00
0.02
0.04
0.06
0.08
0.10
Gamma


For C =1.0


For C =10.0


For C =100


For Best Accuracy



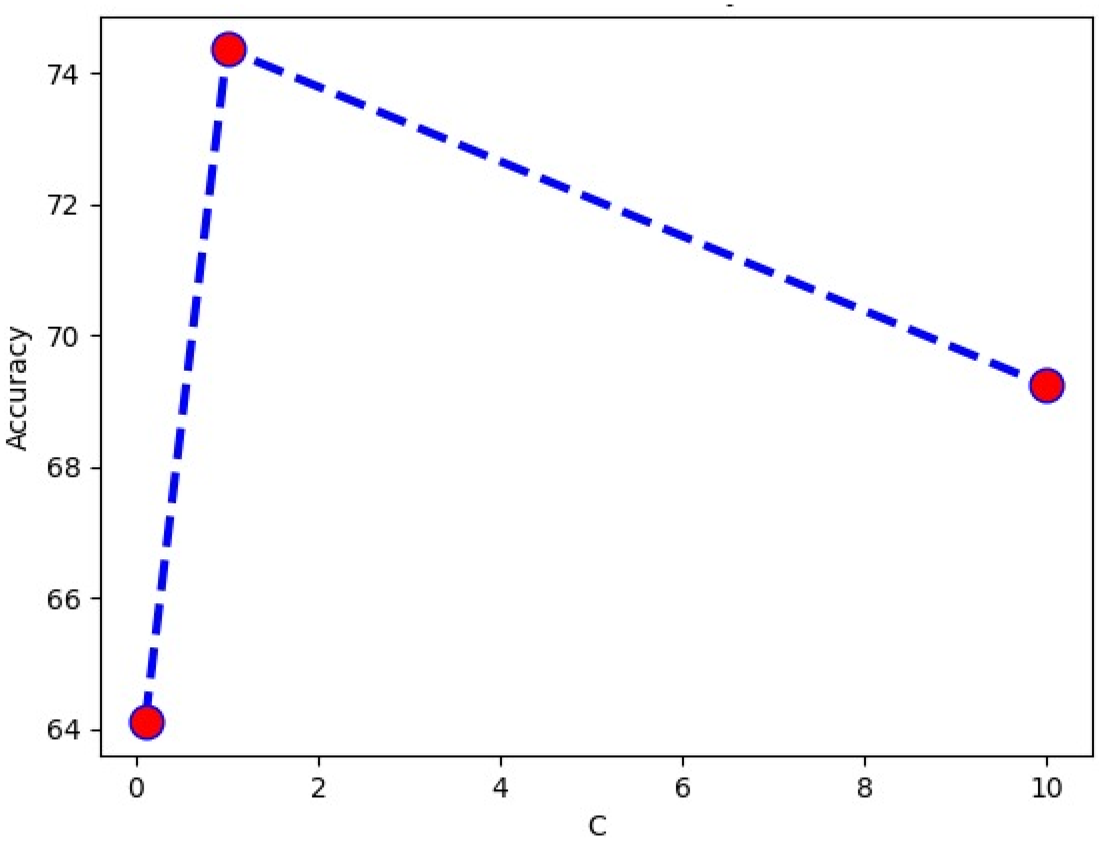
```
Optimal solution found.
gamma =  0.1
accuracy 53.84615384615385
 C                     Best gamma                    Accuracy %   |
   0.01                       0.001                         92.307692307692
   0.1                       0.001                         92.3076923076923
   1.0                       0.001                         92.3076923076923
   10.0                       0.001                         89.743589743589
   100                       0.001                         89.7435897435897
==============================================================
accuracy in linear for c = 10 =
92.3076923076923
```

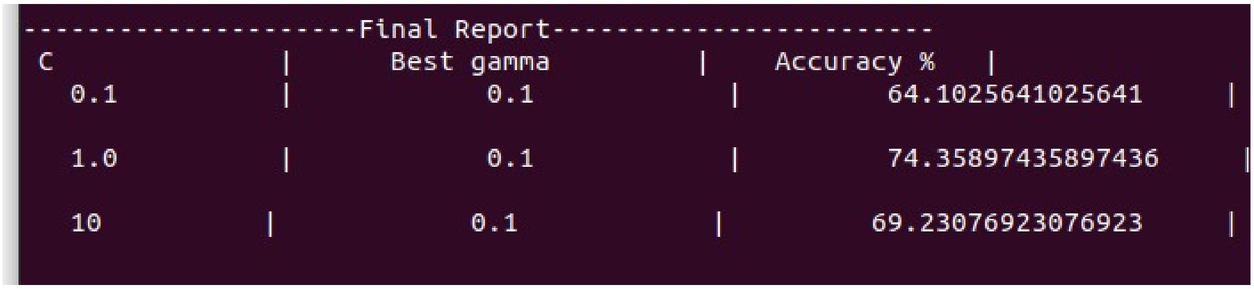
4. Multi-class Classification

Assume we have N different classes. One vs all will train one classifier per class in total N classifiers. For class i, it will assume i -labels as positive and the rest as negative. This often leads to imbalanced datasets meaning generic SVM might not work, but still, there are some workarounds.

In one vs one we have to train a separate classifier for each different pair of labels. This leads to N(N−1) / 2 classifiers. This is much less sensitive to the problems of imbalanced datasets but is much more computationally expensive.

In rbf



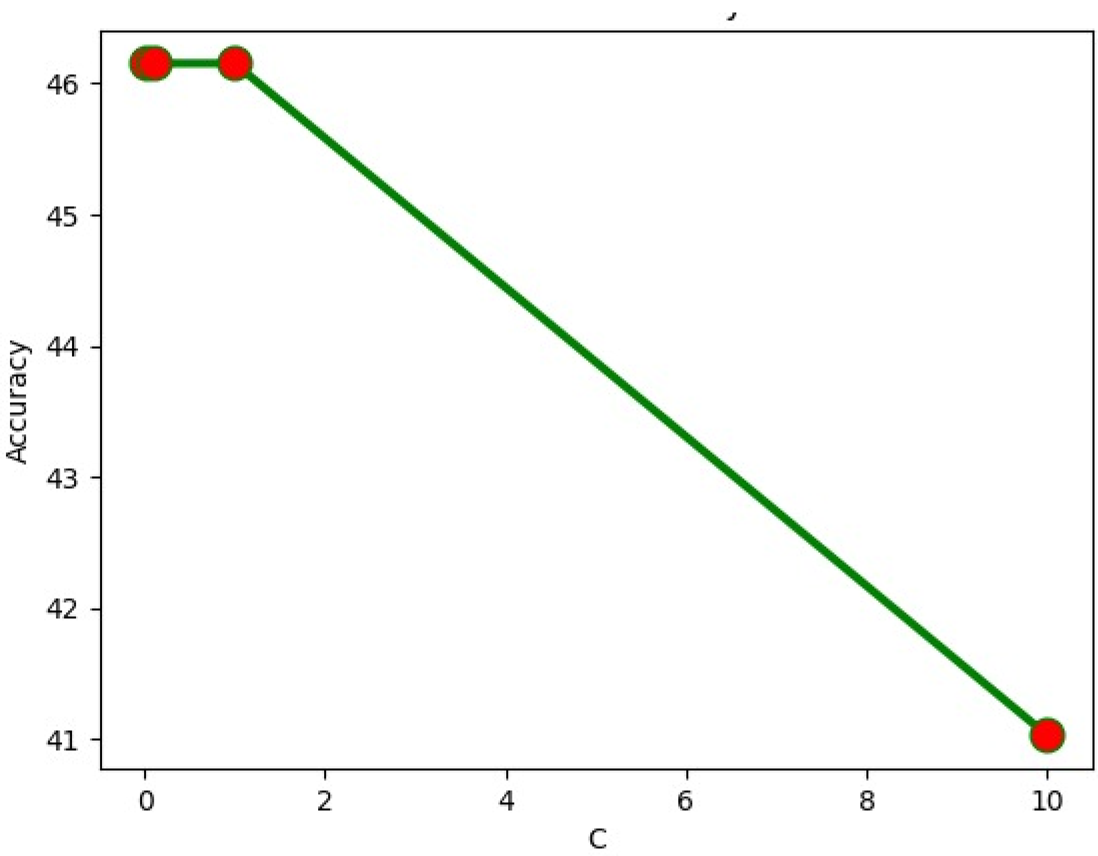


```
---------------------Final Report------------------------
 C               |       Best gamma          |    Accuracy %    |
   0.1           |            0.1              |          64.1025641025641      |

   1.0           |            0.1              |          74.35897435897436      |

   10           |            0.1              |          69.23076923076923      |
```

In OVA
when c = 0.1 accuracy is   64.102564102561
When c = 1.0 accuracy is 74.3589743587436

In OVO
when c = 0.1 accuracy is 76.92307692307692
When c = 1.0 accuracy is 76.92307692307692

In linear



I am also getting confusion matrix in OVO and OVA