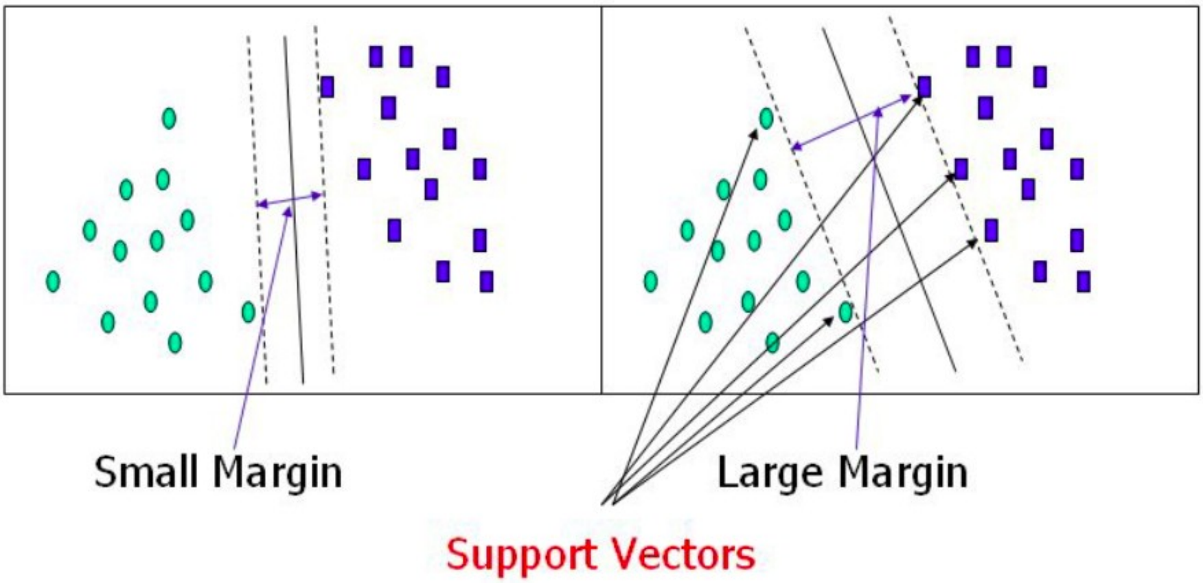
Lab 10
Abu Shahid
B20CS003

SVM and its methodology

SVM is based on the idea of finding a hyperplane that best separates the features into different domains. As SVM is very sensitive to scales; we normalized our data using StandardScaler( ).
To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Dimension of hyperplane depends upon number of features.



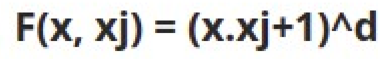
Kernels help us to deal with **high dimensional** data in a very efficient manner.

SVM algorithms use a group of mathematical functions that are known as kernels. The function of a kernel is to require data as input and transform it into the desired form.

- **Linear**
  - It is the most basic type of kernel, usually one dimensional in nature. It proves to be the best function when there are lots of features.

$$F(x, xj) = sum(x.xj)$$

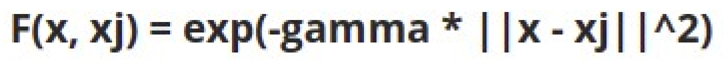
- **Poly**
    - It is a more generalized representation of the linear kernel. It is not as preferred as other kernel functions as it is less efficient and accurate.



F(x, xj) = (x.xj+1)^d

, where for quadratic d equals 2.

- **RBF**
    - It is one of the most preferred and used kernel functions in SVM. It is usually chosen for non-linear data. It helps to make proper separation when there is no prior knowledge of data.



F(x, xj) = exp(-gamma * ||x - xj||^2)

Significance of C parameter

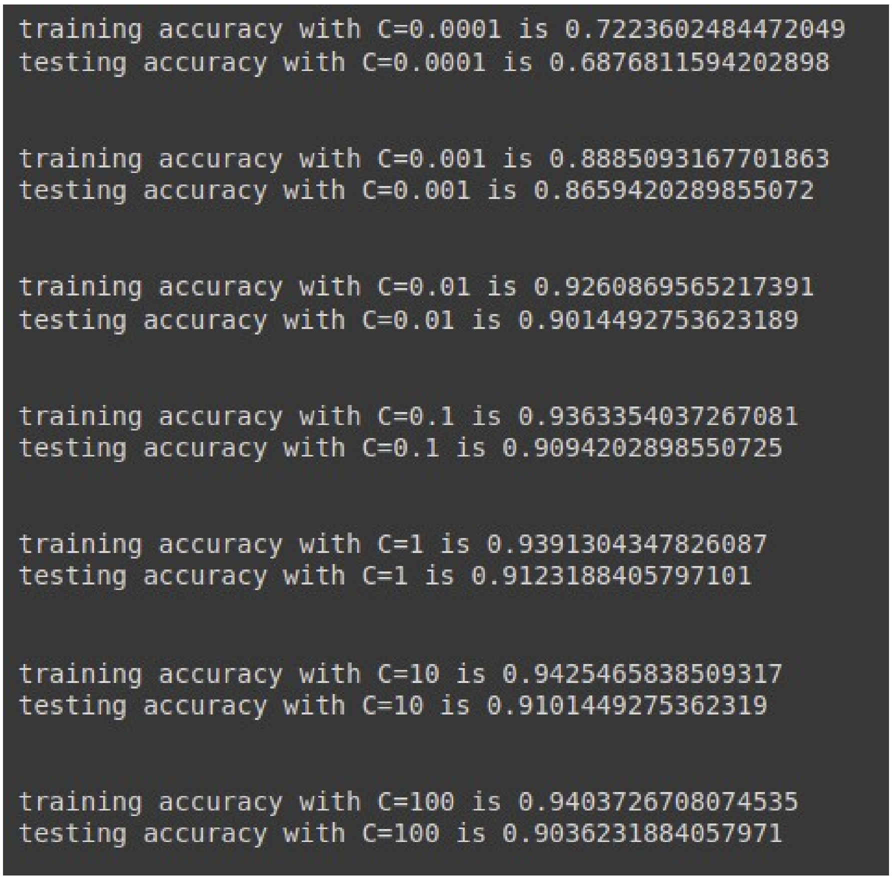
C parameter in SVM is Penalty parameter of the error term. It can be considered as the degree of correct classification that the algorithm has to meet or the degree of optimization the the SVM has to meet. For greater values of C, there is no way that SVM optimizer can misclassify any single point. Futher, higher the value of C, more computationally expensive is the implemntation.

Analysis

Kernel used	Training accuracy	Testing accuracy	Optimal value of C
Linear	93.913%	91.232%	1
Quadratic	95.093%	89.927%	10
RBF	96.925%	92.246%	10
Sigmoid	87.857%	87.681%	1

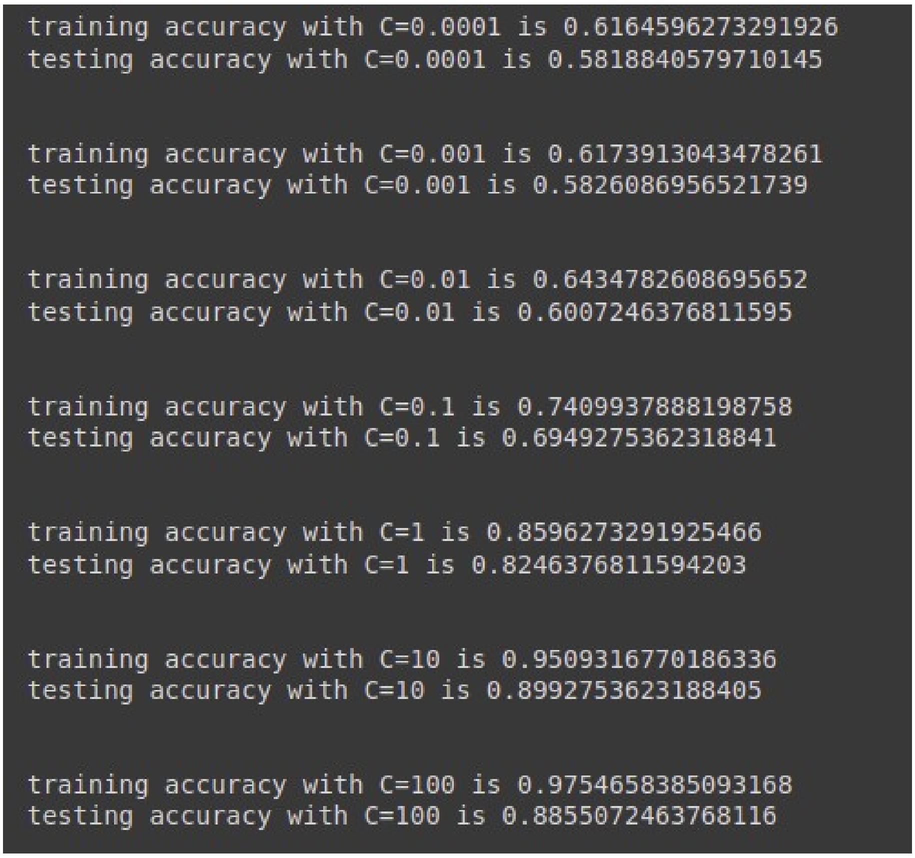
PS: extensive analysis can be found below

For linear:



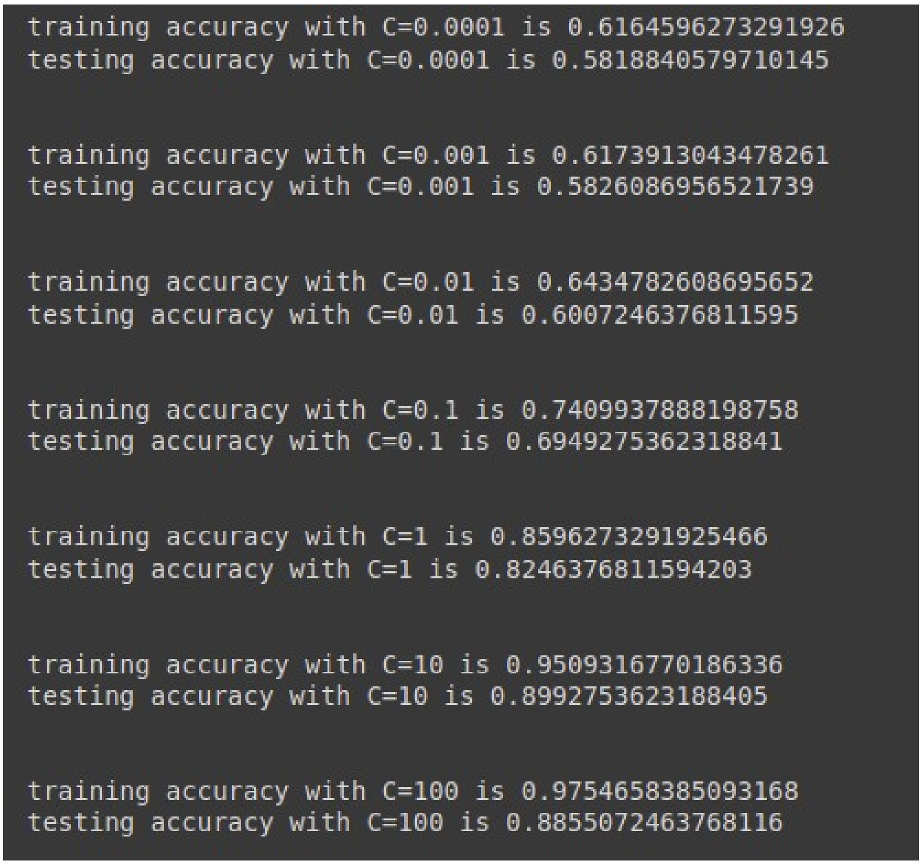
```
training accuracy with C=0.0001 is 0.7223602484472049
testing accuracy with C=0.0001 is 0.6876811594202898


training accuracy with C=0.001 is 0.8885093167701863
testing accuracy with C=0.001 is 0.8659420289855072


training accuracy with C=0.01 is 0.9260869565217391
testing accuracy with C=0.01 is 0.9014492753623189


training accuracy with C=0.1 is 0.9363354037267081
testing accuracy with C=0.1 is 0.9094202898550725


training accuracy with C=1 is 0.9391304347826087
testing accuracy with C=1 is 0.9123188405797101


training accuracy with C=10 is 0.9425465838509317
testing accuracy with C=10 is 0.9101449275362319


training accuracy with C=100 is 0.9403726708074535
testing accuracy with C=100 is 0.9036231884057971
```

For Quadratic:



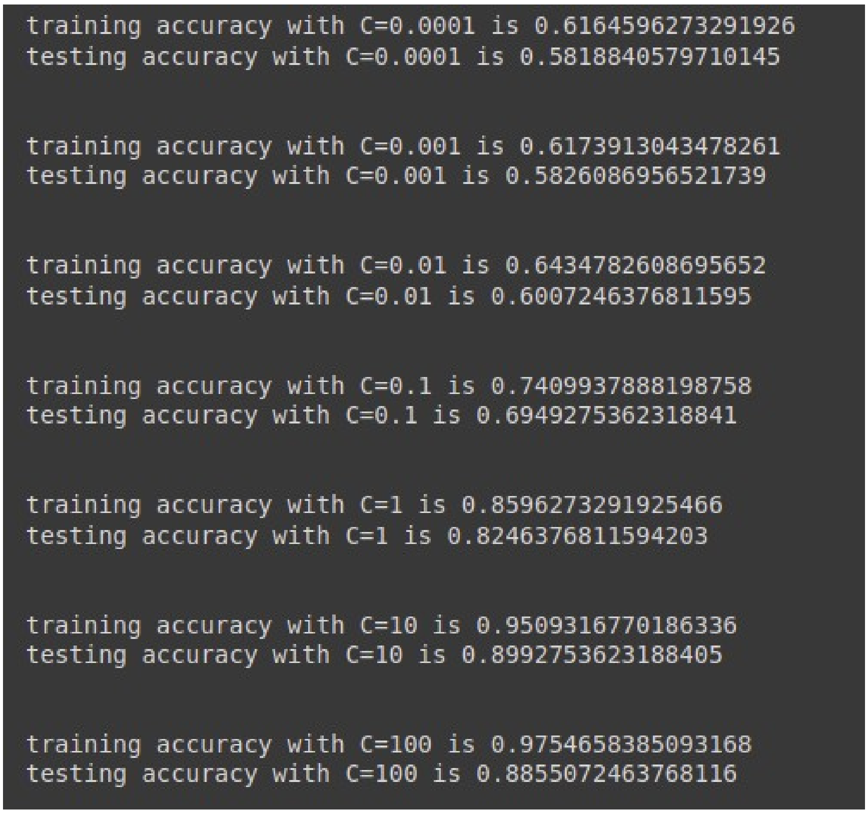
```
training accuracy with C=0.0001 is 0.6164596273291926
testing accuracy with C=0.0001 is 0.5818840579710145


training accuracy with C=0.001 is 0.6173913043478261
testing accuracy with C=0.001 is 0.5826086956521739


training accuracy with C=0.01 is 0.6434782608695652
testing accuracy with C=0.01 is 0.6007246376811595


training accuracy with C=0.1 is 0.7409937888198758
testing accuracy with C=0.1 is 0.6949275362318841


training accuracy with C=1 is 0.8596273291925466
testing accuracy with C=1 is 0.8246376811594203


training accuracy with C=10 is 0.9509316770186336
testing accuracy with C=10 is 0.8992753623188405


training accuracy with C=100 is 0.9754658385093168
testing accuracy with C=100 is 0.8855072463768116
```

For RBF:



```
training accuracy with C=0.0001 is 0.6164596273291926
testing accuracy with C=0.0001 is 0.5818840579710145


training accuracy with C=0.001 is 0.6173913043478261
testing accuracy with C=0.001 is 0.5826086956521739


training accuracy with C=0.01 is 0.6434782608695652
testing accuracy with C=0.01 is 0.6007246376811595


training accuracy with C=0.1 is 0.7409937888198758
testing accuracy with C=0.1 is 0.6949275362318841


training accuracy with C=1 is 0.8596273291925466
testing accuracy with C=1 is 0.8246376811594203


training accuracy with C=10 is 0.9509316770186336
testing accuracy with C=10 is 0.8992753623188405


training accuracy with C=100 is 0.9754658385093168
testing accuracy with C=100 is 0.8855072463768116
```

For Sigmoid:



```
training accuracy with C=0.0001 is 0.6164596273291926
testing accuracy with C=0.0001 is 0.5818840579710145


training accuracy with C=0.001 is 0.6173913043478261
testing accuracy with C=0.001 is 0.5826086956521739


training accuracy with C=0.01 is 0.6434782608695652
testing accuracy with C=0.01 is 0.6007246376811595


training accuracy with C=0.1 is 0.7409937888198758
testing accuracy with C=0.1 is 0.6949275362318841


training accuracy with C=1 is 0.8596273291925466
testing accuracy with C=1 is 0.8246376811594203


training accuracy with C=10 is 0.9509316770186336
testing accuracy with C=10 is 0.8992753623188405


training accuracy with C=100 is 0.9754658385093168
testing accuracy with C=100 is 0.8855072463768116
```