COL774- MACHINE LEARNING
ASSIGNMENT – 2

UTSAV DEEP
2018CS10396

PART 1 - Text Classification

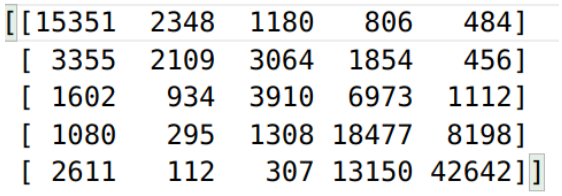
-> This part involved text classification using Naive Bayes model.

(a)

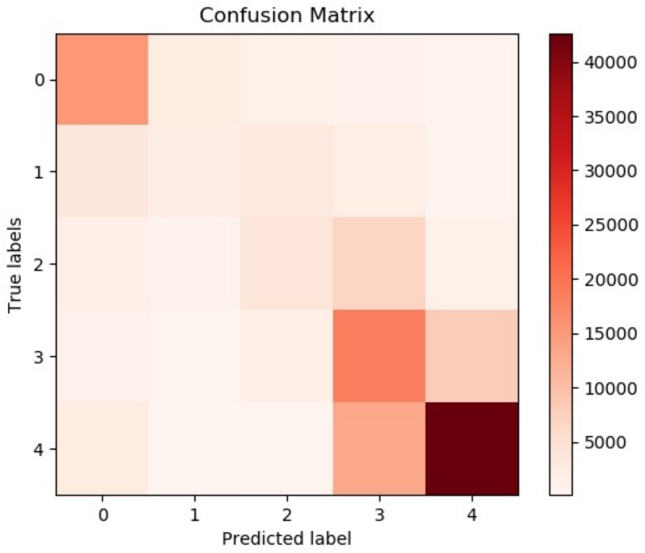
Accuracy over test set = 61.68877787582824 %

Accuracy over training set = 68.6790484452355 %

Confusion Matrix over test data set is:



```
[[15351  2348  1180   806   484]
 [ 3355  2109  3064  1854   456]
 [ 1602   934  3910  6973  1112]
 [ 1080   295  1308 18477  8198]
 [ 2611   112   307 13150 42642]]
```

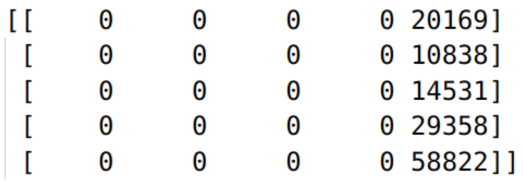


(b)

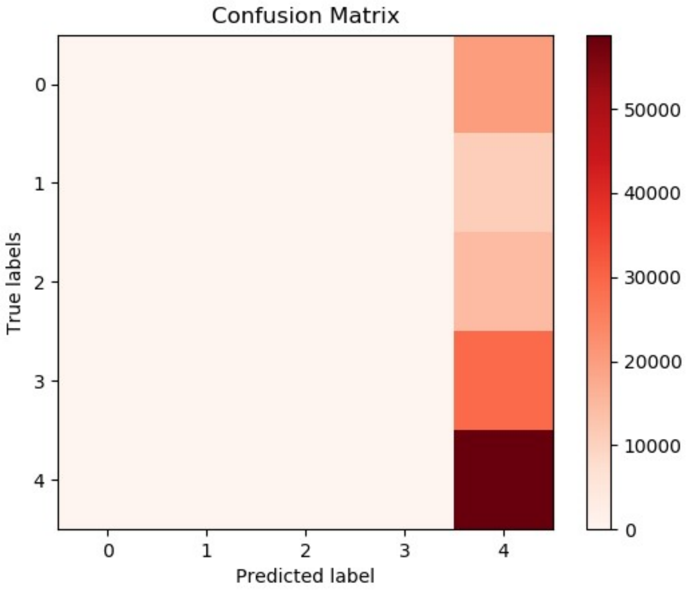
Maximum Prediction:

Accuracy in Test Set= 43.9895900327555 %

Confusion Matrix

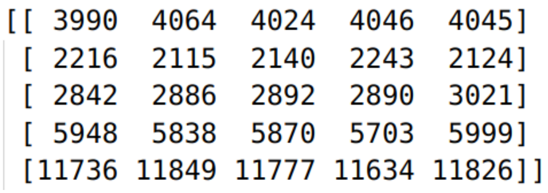


```
[[    0     0     0     0 20169]
 [    0     0     0     0 10838]
 [    0     0     0     0 14531]
 [    0     0     0     0 29358]
 [    0     0     0     0 58822]]
```

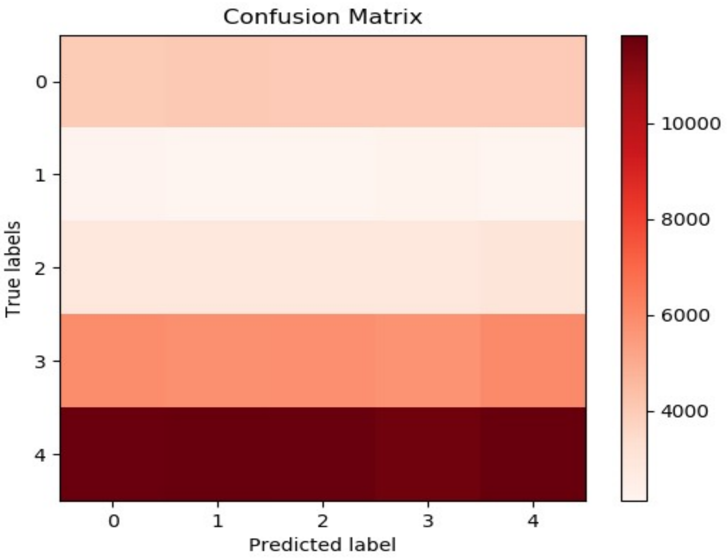


Random Prediction:

Accuracy in Test Set= 19.83726947755725 %
Confusion Matrix



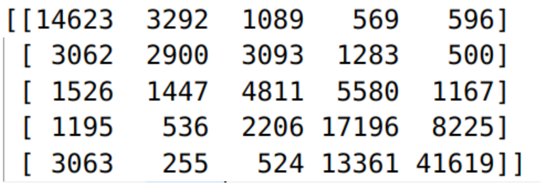
```
[[ 3990  4064  4024  4046  4045]
 [ 2216  2115  2140  2243  2124]
 [ 2842  2886  2892  2890  3021]
 [ 5948  5838  5870  5703  5999]
 [11736 11849 11777 11634 11826]]
```



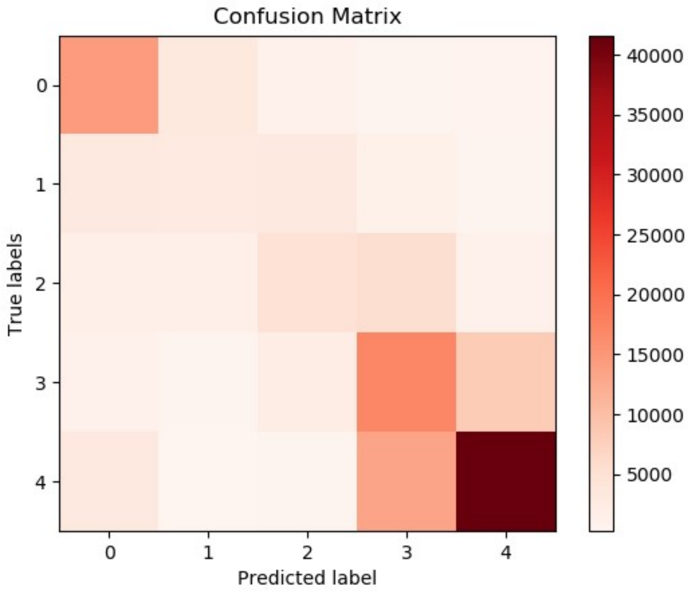
->My naive bayes algo has an improvement of 18% wrt max Prediction and 40% wrt random prediction. This is a huge improvement in terms of accuracy.

(d) Stemming:

Accuracy in Test Set= 60.68666896005025 %
Confusion Matrix



```
[[14623  3292  1089   569   596]
 [ 3062  2900  3093  1283   500]
 [ 1526  1447  4811  5580  1167]
 [ 1195   536  2206 17196  8225]
 [ 3063   255   524 13361 41619]]
```



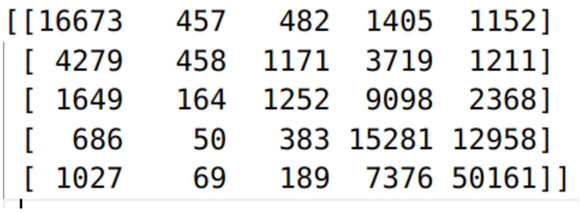
-> The accuracy has decreased slightly upon stemming.
-> Also the time taken to train the model has significantly increased due to stemming.

(e)

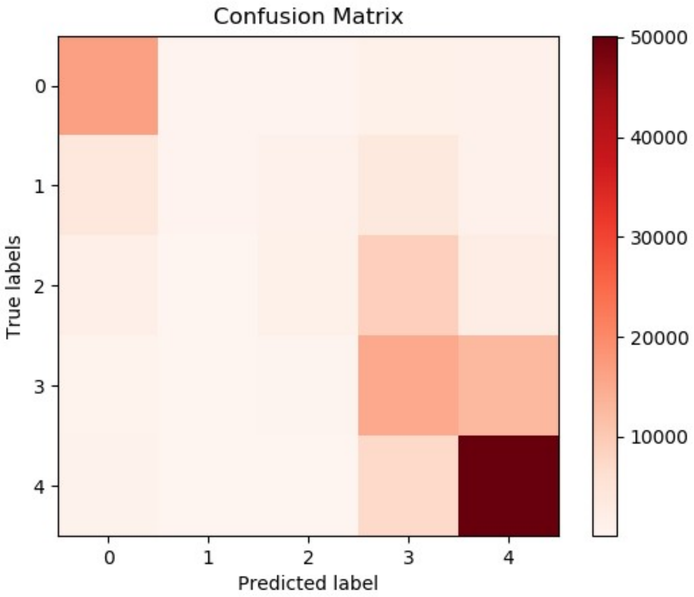
-> I have made predictions using 3 different features. The respective accuracies are noted below:

=> 1. BiGrams:

Accuracy over Test Set = 62.687895421708376 %
Confusion Matrix

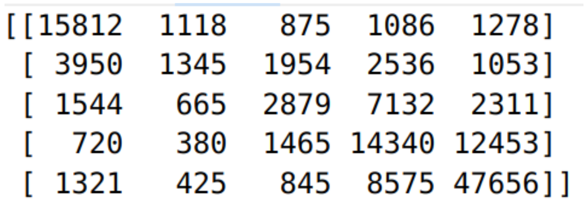


```
[[16673   457   482  1405  1152]
 [ 4279   458  1171  3719  1211]
 [ 1649   164  1252  9098  2368]
 [  686    50   383 15281 12958]
 [ 1027    69   189  7376 50161]]
```

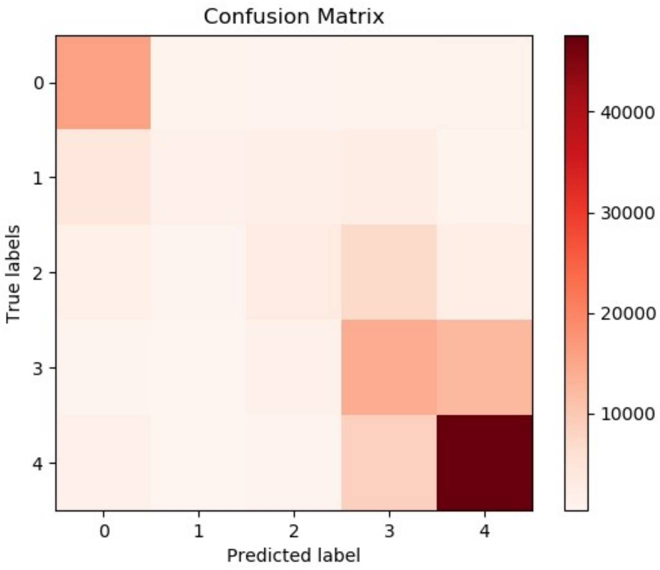


=> 2. TriGrams:

Accuracy over Test Set = 61.347013864999475 %
Confusion Matrix

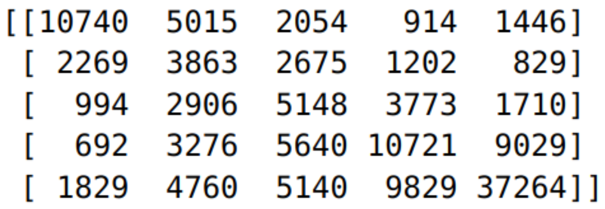


```
[[15812  1118   875  1086  1278]
 [ 3950  1345  1954  2536  1053]
 [ 1544   665  2879  7132  2311]
 [  720   380  1465 14340 12453]
 [ 1321   425   845  8575 47656]]
```

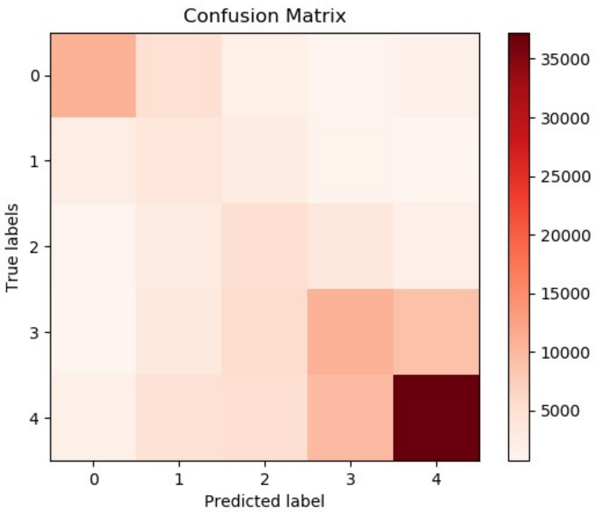


=> 3. Quarternary Grams:

Accuracy over Test Set = 50.65585785010246 %
Confusion Matrix



```
[[10740  5015  2054   914  1446]
 [ 2269  3863  2675  1202   829]
 [  994  2906  5148  3773  1710]
 [  692  3276  5640 10721  9029]
 [ 1829  4760  5140  9829 37264]]
```



-> Out of all , bigrams gives the best accuracy and thus it has been used to give the output.
-> When compared to single word, bigrams give better predictions. This is what has been predicted by the various models.
-> Also we see that when we move from bigrams to trigrams , the accuracy decreases.
-> This accuracy further decreases as we move from trigrams to quarternary grams.

PART 2 - Fashion MNIST Article Classification

-> The MultiClass classification using Sklearn using the gaussian kernel gives the best prediction and has been used for showing output.

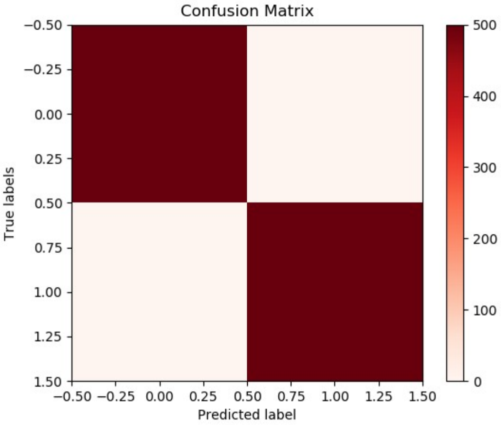
(a) Binary Classification:

The two classes were 6 and 7.
(i)

Number of Support Vectors = 51

Confusion Matrix

```
[[500   0]
 [  0 500]]
```

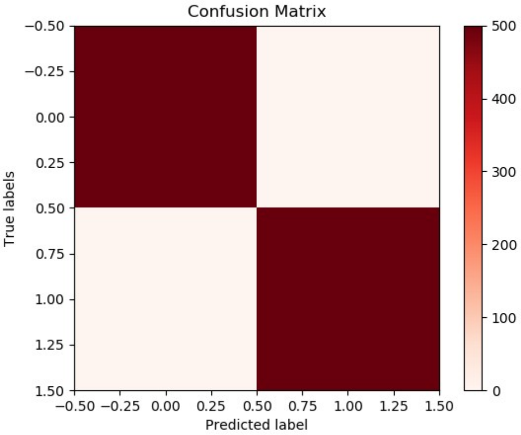


1. -8.95e-03 is the value of b
2. Average Test set accuracy = 1.0
3. Average Validation set accuracy = 1.0

(ii)

1. Number of support vectors = 675
2. Confusioin Matrix:

```
[[499   1]
 [  0 500]]
```



Average Test set accuracy = 0.999

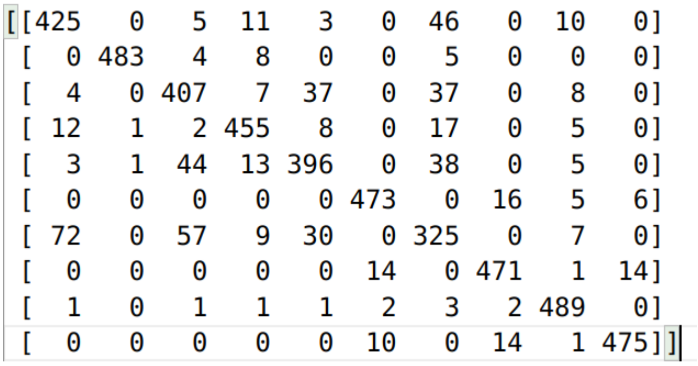
Average Validation set accuracy = 0.992

-> When compared to the linear kernel, the accuracies have decreased for both validation data and test data. Therefore we can say that here linear kernerl is better than gaussian kernel for binary classification between class 6 and 7.

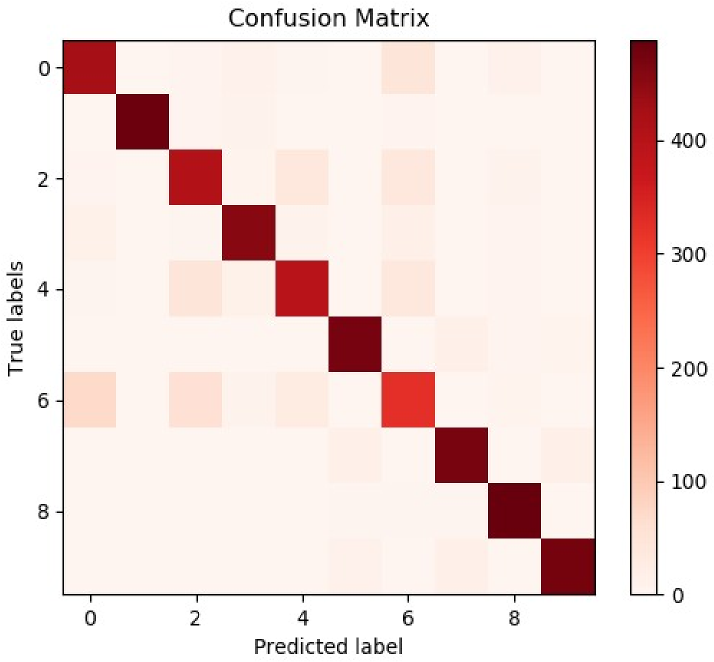
(b) Multi Class Classification:

(i) This model took around 120 minutes to get trained. This is because the program had to make predictions from 45 classifiers.

Confusion Matrix:



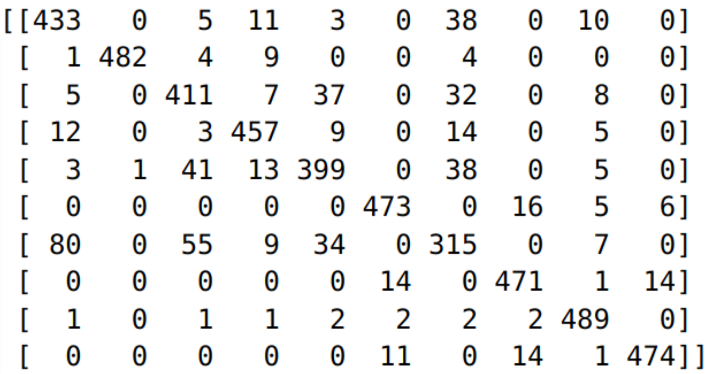
```
[[425   0   5  11   3   0  46   0  10   0]
 [  0 483   4   8   0   0   5   0   0   0]
 [  4   0 407   7  37   0  37   0   8   0]
 [ 12   1   2 455   8   0  17   0   5   0]
 [  3   1  44  13 396   0  38   0   5   0]
 [  0   0   0   0   0 473   0  16   5   6]
 [ 72   0  57   9  30   0 325   0   7   0]
 [  0   0   0   0   0  14   0 471   1  14]
 [  1   0   1   1   1   2   3   2 489   0]
 [  0   0   0   0   0  10   0  14   1 475]]
```



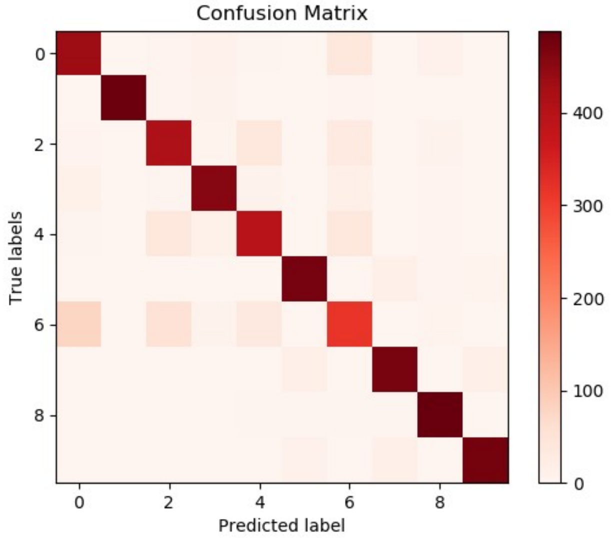
Average Test set accuracy = 0.8798
Average Validation set accuracy = 0.8642

(ii) This model took lesser time than above ~ 15 minutes

Confusion Matrix



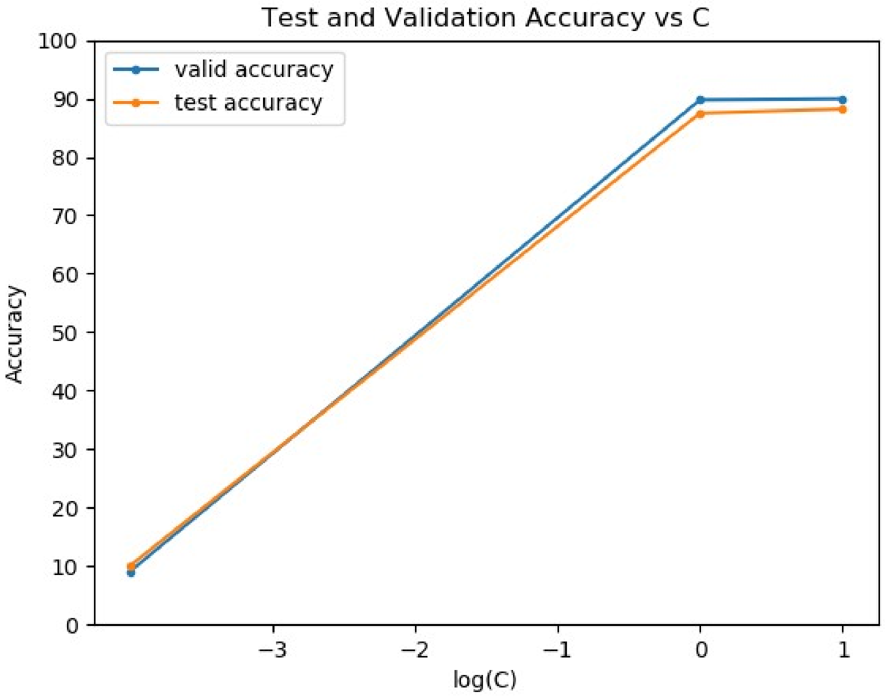
```
[[433   0   5  11   3   0  38   0  10   0]
 [  1 482   4   9   0   0   4   0   0   0]
 [  5   0 411   7  37   0  32   0   8   0]
 [ 12   0   3 457   9   0  14   0   5   0]
 [  3   1  41  13 399   0  38   0   5   0]
 [  0   0   0   0   0 473   0  16   5   6]
 [ 80   0  55   9  34   0 315   0   7   0]
 [  0   0   0   0   0  14   0 471   1  14]
 [  1   0   1   1   2   2   2   2 489   0]
 [  0   0   0   0   0  11   0  14   1 474]]
```



Average Test set accuracy = 0.8808
Average Validation set accuracy = 0.8792

-> When compared to the part(i) model this model is better in terms of accuracy.
-> Not only that, even the time taken to train model in this question is very less when comared to the time taken to train model in above quesitton.
-> The above model took a lot of time becuase it trained 45 different classifiers.

(iv) The values of C taken were 0.001, 1 and 10.
I used a validation set instead of K-fold cross validation and use SciKit.
Gamma was 0.05 in all cases.
The highest accuracy was obtained for C = 10 for both validation and test set.



> From the graph it is seen that, higher the C greater the accuracy. But this need not be the case always.
> Also accuracy at C =1 and 10 are nearly same.
> In my implementation, I have calculated the accuracy for the different C values, then the C which gives the best accuracy is used for prediction puroposes.

PART 3 - Large scale text classification

=> Svm with gaussian kernel gave the best prediction and that has been used to predict data.