ASSIGNMENT 2
MACHINE LEARNING

SAKET DINGLIWAL 2015CS10254

PART 1
NAIVE BAYES

Question 1

Naive Bayes was implemented without using any cleaning of the text data (simple string.split()) was used to tokenize the data.
The training accuracy obtained was - **68.844**
The test accuracy obtained was - **37.388**

Question 2

Test set accuracy for random selection of class, with each class being equally likely was obtained as - **11.188**
Accuracy for choosing the most frequent class was obtained as – **20.088**
The test set accuracy for the naive bayes was 37.388 which is almost double than that by majority selection and triple than that by random selection.Although this is not much ahead of these simple baseline, but good enough for the simplistic naive bayes model.Since there are 8 classes, the random one was expected to be close to 1/8. Since there are uneven number of documents with each classes, the majority is around 20%

Question 3

The confusion matrix obtained is as shown on the next page. The largest diagonal value is of the first class ie **4726**. This value signifies that there are 4726 documents in the test data that are marked with label 1 and Naive Bayes classifies them to be 1.

pred-> Sys _|	1	2	3	4	7	8	9	10
1	4726	0	4	34	9	33	0	216
2	2028	2	4	56	13	44	0	155
3	1928	1	28	140	24	120	0	270
4	1674	2	4	241	62	218	1	433
7	656	0	0	55	91	436	0	1069
8	616	0	2	33	34	469	2	1694
9	435	0	0	9	16	210	0	1674
10	975	0	2	8	13	210	1	3790

We can see that the 1st column of the table has very high values. This column represents number of examples that were predicted as '1'. We can see the decreasing trend in these values because as the rating increases, the Naive bayes loses evidence for it being in class '1'. A similar trend can be seen for last column where an increasing trend is visible. For naive bayes it's difficult to classify closeby classes eg 1,2,3 and 8,9,10. Since there is class imbalance in the training data, these are mostly classified mostly as 1 and 10 respectively.

Question 4

New accuracy obtained on test data -> **38.684**
Since now the words like "run","runs" etc will be the same and hence all the instances in different example correspond to same form. Also the words like "the", "of" which do not tell much about the class probabilities are ignored. However, there is a very little increase in the accuracy. This is due to the data has a large class imbalance and most of the loss of accuracy arises from the classifier not able to learn between closeby classes. The impact of the stop words and various forms of the same word have a little impact relatively. Although the size of the vocabulary

decreases considerably but the probabilities learned do not bring a bigger impact. The diagonal elements of classes other than class 1 and 10 increases significantly.

pred-> Sys _|	1	2	3	4	7	8	9	10
1	4236	89	160	240	42	72	24	159
2	1559	54	175	278	61	45	9	121
3	1331	72	237	498	106	111	18	168
4	980	53	259	651	205	218	32	237
7	360	19	93	286	378	516	85	570
8	375	20	87	198	303	713	146	1008
9	295	9	39	110	156	433	114	1188
10	630	19	52	130	191	514	175	3288

Question 5

(i) **Negation tokens->**
Many times in the review there is use of the term “not” which essentially negates the adjective used after that. However using bag of words model, such a case is ignored. Therefore I added the feature of “not word” to account for this. After trial and runs I found that the inverting of two words ahead of word “not”,brings up better accuracy. Many words got inverted in the process and some of the significant words were like “not_like”, “not_great”. This helps in improving accuracy as it removes words like “great” from the examples of lower classes.
Accuracy achieved on test data- > **38.776**
(ii) **Bigrams ->**
To tackle the independence assumption of the bag of words model, bigrams were added as features to the document. Various bigrams like [{ good stori, around great, like complet, immedi recogn, shown screen, arguabl one , especi end , movi support , anim anim }]
Were present more than 7 times in the training data. Therefore adding these to the dataset improved the accuracy of the training and the test

set by a large amount. So final model was learned on having bigrams count in the corpus being more than 7 which was obtained by multiple iterations over the hyperparameter threshold.
Test accuracy achieved -> **41.344**
(iii) **Inverse Document Frequency** ->
There are certain common words which are domain specific like “movie”, “play” that are not removed by the stop word removal and also do not have any contribution towards the classification task. Therefore such words can be weighed down by using idf.
Idf of a word = Log(N/nk) + 1 where nk are the number of documents having the word. For very rare word the weight is very high while for those which are almost present in every document the weight is close to 1. However using these weighted features do not increase the accuracy of the test set. Accuracy obtained on training -> **76.768**
While that on text -> **36.252**

PART 2

SVM

Question 1

Pegasos algorithm was implemented. For the intercept term, gradient descent was used as well.
The following accuracies were obtained for C = 1.0 and T = 10000
Training -> **0.8867**
Test -> **0.901**
However for the unscaled data the accuracy are higher eg test accuracy for T=10000 is 0.92

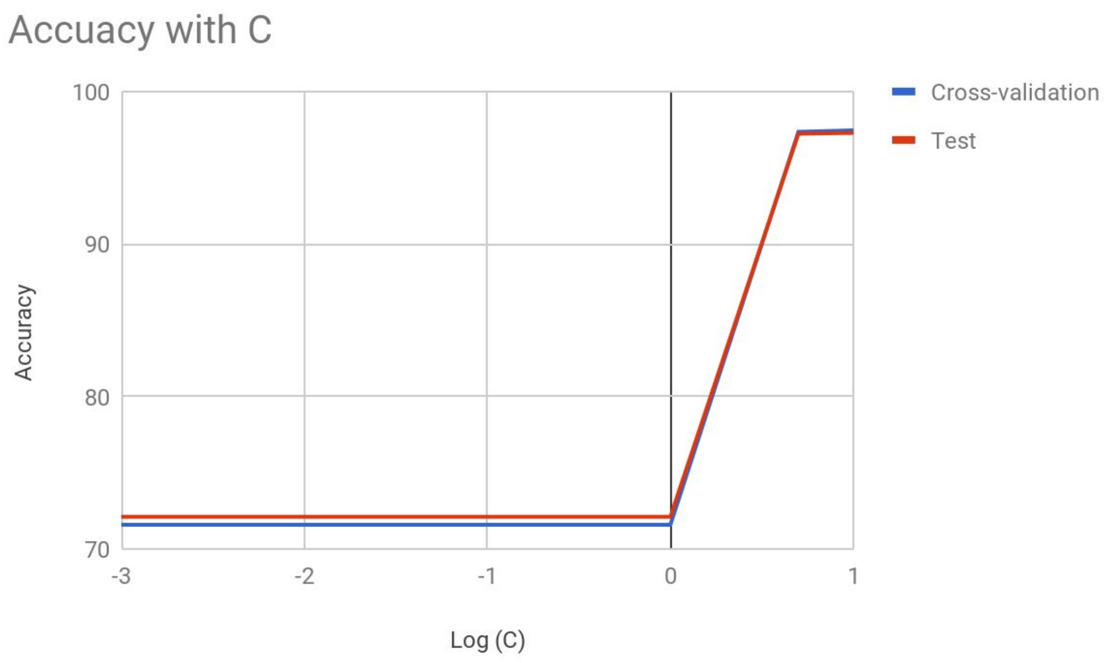
Question 2

When LibSvm is used the accuracies come out to be as follows
The C = 1.0 and gamma - 0.05 are used.
Linear-accuracy on scaled test data -> **92.78%**
Gaussian accuracy on scaled test data -> **97.23%**

The accuracy of linear svm is greater than our implementation. This may be due to larger number of iterations and maybe better handling of the intercept term. Use of gradient for the intercept, decreases the convergence rate. Also since gaussian kernel includes other features as well and hence provide a better fit and higher accuracy than our implementation.

Question 3

C	Cross-validation Accuracy	Test Accuracy
0.00001	71.59	72.11
0.001	71.59	72.11
1	97.355	97.23
5	97.455	97.29
10	97.575	97.29



We observe that the best cross validation as well as test set accuracy is achieved for C = 10. This may be due to making larger penalty for points lying inside the margin.

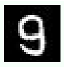
We can see that the test accuracies follow exactly same trend as cross validation accuracies. This can be very helpful if we have lesser amount of data at hand. Thus hyper parameters can be set using training data only.

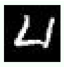
Question 4

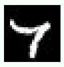
The following confusion matrix gets formed for the best model which is C = 10, gamma = 0.05

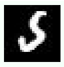
```
[[ 969    0    1    0    0    3    4    1    2    0]
 [   0 1122    3    2    1    2    2    0    2    1]
 [   4    0 1000    4    2    0    1    6   15    0]
 [   0    0    8  985    0    4    0    7    5    1]
 [   1    0    4    0  962    0    5    0    2    8]
 [   2    0    3    6    1  866    7    1    5    1]
 [   5    4    0    0    3    4  940    0    2    0]
 [   1    4   20    2    3    0    0  986    2   10]
 [   4    0    3   10    1    5    3    3  942    3]
 [   4    4    3    8    9    4    0    9   11  957]]
```

The most difficult class to classify is 9. It has the lowest recall ie 94%. There are 11 cases where it has been classified as 8, and 9 of them as 7 or 4. Clearly due to its similarity with different numbers, it becomes most difficult to classify.

 gets classified as 8

 classified as 0

 classified as 4

 classified as 6

This happens due to similarity in the image. 9 has a great chance of being classified as 8 because there is very small difference in 2 closed loops (in 8) and 1 closed loop and other open(in 9).