COL774 Assignment-2

Question1 : Text Classification

Part(a)

Without making lower case and removing punctuations
Size of vocabulary=93929 words
Using Naive Bayes,
Accuracy on training data : 69.996%
Accuracy on testing data : 38.688%

After making lower case and removing punctuations
Size of vocabulary=74891 words
Using Naive Bayes,
Accuracy on training data : **68.488%**
Accuracy on testing data : **38.684%**

The accuracy remains almost the same, making lower case and removing punctuations has very small or no effect on accuracy.

Below in every reference to part(a) making lower case and removing punctuations case is taken.

Part (b)

Using Random class prediction,
Accuracy on testing data : **12.43%**

Using Most Occurring class prediction,
Accuracy on testing data : **20.088%**

Naive Bayes algorithm does a lot of improvement over random,majority baseline. 18.6% increase in accuracy from the majority baseline and 26.258% increase from the random baseline.

Part (c)

Confusion matrix for Part(a) on testing data:

Class	**1**	**2**	**3**	**4**	**7**	**8**	**9**	**10**
1	4348	46	131	240	37	62	12	146
2	1636	37	156	260	54	58	4	97
3	1409	40	199	488	122	120	10	153
4	1081	35	180	657	210	238	24	210
7	427	5	64	248	405	539	50	569
8	441	9	54	156	291	732	90	1077
9	350	3	21	82	131	468	82	1207
10	819	7	36	94	158	553	121	3211

Vertical axis is for actual class and horizontal axis is for predicted class.

Class **1** has the highest value of diagonal entry, this means there are a lot of 1 in the test data and also they were predicted correctly. There are a lot of correct predictions of 1 but there are also a lot of wrong predictions of 1.

There are a lot of 1 and 10 class predictions and the predictions for 2-9 are very less.

Part (d) Stemming and stop word removal

Accuracy with stemming and stop word removal on test set = **38.684%**

Stemming and stop word removal don't affect the accuracy much, accuracy remains almost the same. This may be because stop words don't really differentiate between any 2 classes, but they do act as noise in the data. In our case maybe the data is small and there are less stop word. Stemming also helps when there are a lot of similar words. It also makes computation faster.

Confusion matrix with stemming and stop word removal:

Class	**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

There are a lot of 1 and 10 predictions because in reality also there are a lot of 1 and 10 ratings.

Part(e)

1. Taking mono+bi grams as features
Accuracy on stemmed test data=**36.856%**

Using just bi grams as features instead of mono+bi grams gives ~20% which is almost same as most occurring prediction.

2. Taking mono+tri grams as features
Accuracy on stemmed test data=**35.36%**

3. Taking mono+quad grams as features
Accuracy on stemmed test data=**34.724%**

The above features don't increase the accuracy of the model instead decrease slightly

Question2

Part (a)

Implemented mini-batch version of Pegasos algorithm using the following equations:

$$A \subseteq \{1...m\}, |A| = k$$

A is chosen randomly. k is given 100.

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{i \in A} max(0, 1 - y^{(i)}(w^T x^{(i)} + b))$$

Part (b)

Implemented one-vs-one multi-class SVM using the pegasos algorithm.

For C=1,
Accuracy on train data = **94.135%**
Accuracy on test data = **92.52%**

Part (c)

Implemented multi-class SVM on this dataset using the LIBSVM library.

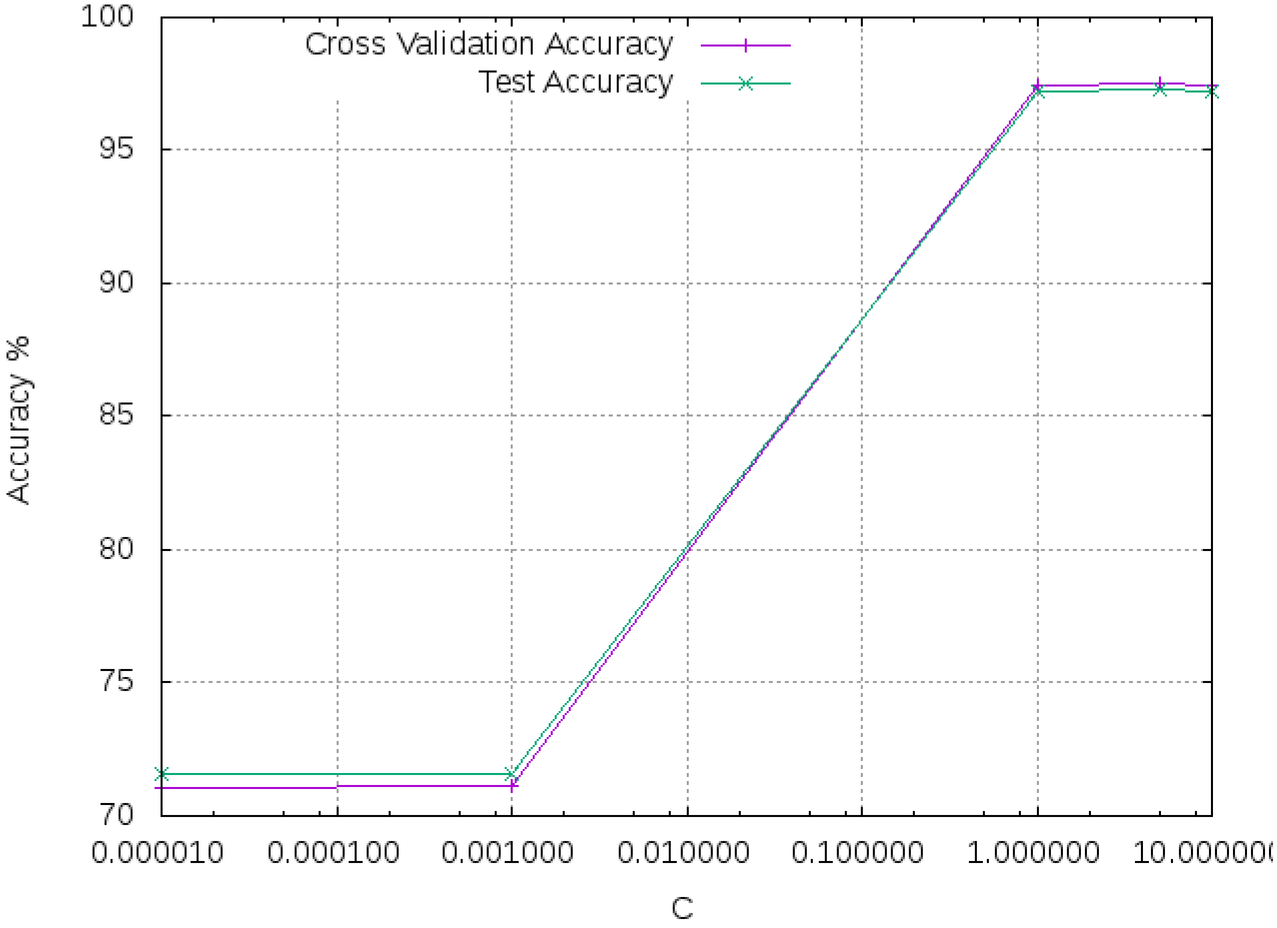
For C=1 in both cases,
Accuracy with Linear kernel on test data = **92.65%**
Accuracy with Gaussian kernel(γ=0.05) on test data = **97.24%**

Accuracy of linear kernel is almost same as in part (b), this is because in (b) the code is for linear kernel, no conversion to higher dimension has been done in (b).

Part (d)

Did 10 fold cross-validation on Gaussian kernel case with γ=0.05 using LIBSVM

Values of C=	[0.00001	0.001	1	5	10]
Cross Validation accuracies =	[71.085	71.12	97.425	97.5	97.43]
Test set accuracies =	[71.58	71.58	97.22	97.29	97.25]



C=5 gives the best validation accuracy it also gives the best test set accuracy.
Cross validation accuracy first increases with C and then decreases slightly, test set accuracy first increases then decreases slightly.

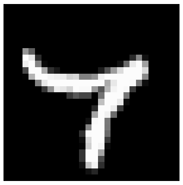
Part(e)
The accuracies for C=1,5,10 are very close.
The best result is for C=5.
Confusion matrix for C=5 is:

label	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Error %**
0	969	0	1	0	0	3	4	1	2	0	1.1224489796
1	0	1123	3	2	0	2	2	0	2	1	1.0572687225
2	4	0	1000	4	2	0	1	6	15	0	3.1007751938
3	0	0	9	985	0	4	0	6	5	1	2.4752475248
4	0	0	5	0	962	0	5	0	2	8	2.0366598778
5	2	0	3	6	1	866	7	1	5	1	2.9147982063
6	5	4	1	0	3	4	939	0	2	0	1.9832985386
7	1	4	20	2	3	0	0	986	2	10	4.0856031128
8	4	0	3	10	1	5	3	3	942	3	3.2854209446
9	4	4	4	8	9	4	0	7	12	957	5.153617443

Class which is similar to many other classes should be most difficult to classify.
Class 9 is the most difficult to classify because it has the highest % of wrong predictions.

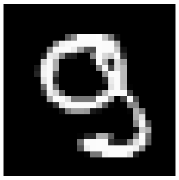
Some miss classified examples:

1)



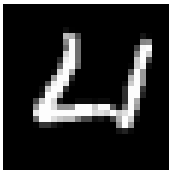
Actual:7
prediction:4

2)



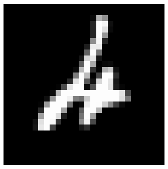
Actual:9
prediction:8

3)



Actual:4
prediction:0

4)



Actual:4
prediction:2

These miss classified examples are not written properly and seem to confuse the classifier, these examples look in between of actual and prediction.