Assignment 2

Akshat Khare, 2016CS10315

March 13, 2019

1 Text Classification: Naive Bayes

1.1 a

Accuracy over test set: 60.9%
Accuracy over training set: 64.4%

1.2 b

1.2.1 Random Accuracy

Test data: 16.82%
Train data: 16.69%

1.2.2 Most occuring based classification accurace

Test data: 43.98%
Train data: 43.88%

1.2.3 Improvement over Random based

Test data: 44.08%
Train data: 47.71%

1.2.4 Improvement over Most occuring based

Test data: 16.92%
Train data: 20.52%

1.3 c

Test data confusion matrix:

```
[[14494. 2867. 1371. 1094. 3081.]
 [ 3609. 3201. 1640.  655.  280.]
 [ 1149. 3368. 5283. 2459.  548.]
 [  496. 1039. 5344. 17712. 14067.]
 [  421.  363.  893. 7438. 40846.]]
```

Five stars rating has the most value (40846) of diagonal entry.
This means that five star is class which has most number of reviews which have been predicted correctly, i.e., it has most number of true positives.
We can see that one stars and five stars review have been classified most successfully as their diagonal entries

are much more than corresponding non-diagonal entries.
Two stars and three stars have been poorly classified as their true positives can't overpower the false positives and true negatives. Four stars have mediocre performance.
So we can observe that five star reviews occur in majority and hence the previous observation of high most occurring based accuracy of 43.98% occurs due to this reason. We can also observe that people vote majorly on extremes and the classification works good on extremes. Confusion matrix is hence a good measure of judging performance of the classifier.

1.4 d

Test data accuracy: 60.76%
We see that applying stop word removal and stemming we in fact see a decrease in observed accuracy over test data.
Time taken significantly increased to 1.5 hour.
The confusion matrix obtained was:

```
[[14522. 2980. 1484. 1197. 3063.]
 [ 3512. 3100. 1569.  613.  269.]
 [ 1073. 3121. 5089. 2406.  589.]
 [  514. 1157. 5265. 17110. 13474.]
 [  548.  480. 1124. 8032. 41427.]]
```

So we in fact observed a decrease in accuracy due reasons like presence of many words which were important to classify by stop words and changing of important words by stemming.

1.5 e

Applying bi-grams helping in increasing accuracy.
Test data accuracy: 63.97%
Improvent over part a is 3.07%
Improvement over part d is 3.21%
So we can say that bigrams helped in improving overall accuracy.
The obtained confusion matrix was:

```
[[16716. 4037. 1627.  736. 1201.]
 [  907.  956.  289.   63.   65.]
 [  750. 1935. 2115.  518.  160.]
 [ 1218. 3329. 9049. 17625. 9268.]
 [  578.  581. 1451. 10416. 48128.]]
```

So we can see that one star, and five star performance increased significantly because around 2000 increase in one star's true positives and 7000 increase in five star's true positives.
Four stars performance didn't increase as much but increased.
Whats strange is performance of two stars and three stars decreased substantially with decrements of 2000 in two stars true positive and 3000 in five star's true positive.
So we can safely say that extremes are being classified nicely but the improvement took over a toll on middle cases.

Applying lemmatization also helping in increasing accuracy.
Test data accuracy: 61.05%
Improvent over part a is 1%
Improvement over part d is 1%
So we can say that lemmatization helped in improving overall accuracy but not much.

The obtained confusion matrix was:

```
[[14612. 2986. 1489. 1138. 2867.]
 [ 3464. 3052. 1500.  544.  243.]
 [ 1076. 3174. 5055. 2325.  534.]
 [  500. 1153. 5387. 17286. 13542.]
 [  517.  473. 1100. 8065. 41636.]]
```

So ngrams is a good choice and we will use that in next part

1.6 f

$f1$ for 1 stars is
0.7515173312952389
$f1$ for 2 stars is
0.1457539259033389
$f1$ for 3 stars is
0.21140486780948572
$f1$ for 4 stars is
0.5046745028419259
$f1$ for 5 stars is
0.8022937920917517
$F1$ array is

```
[0.7515173312952389, 0.1457539259033389, 0.21140486780948572, 0.5046745028419259,
    0.8022937920917517]
```

Macro $F1$ score is: 0.48312888398834825
In my opinion $F1$ score is better metric for the data set given. This is because the accuracy metric may be biased over a specific input which in our test data happens to be the five stars. Our model has accuracy on one and five star cases but not for any general case. It is also evident from the fact that when accuracy is calculated with most occuring review which is five the accuracy comes up to be whopping 43% but this might not be true for any general test set. Therefore accuracy is misleading. But $f1$ score compares true positives, false positives and true negatives and give a more informed metric. With accuracy we might not be able to correctly compare each class specific performance but with $F1$ we can see that $F1$ score of stars is very less than five stars and that brings macro $F1$ score down which should be the case. So in the current dataset $F1$ score is a better metric.

1.7 g

The best performing algorithm was of bigrams with stop word removal and stemming which is part e.
Running on train full and test of previous parts with almost 2 hours of computing.:
Test set accuracy: 74.4% Confusion matrix:

```
[[17023. 2856. 1360.  728. 1078.]
 [ 1523. 4418.  350.   93.   52.]
 [  919. 2001. 7370.  397.  116.]
 [  485. 1308. 4700. 21738. 8690.]
 [  219.  255.  751. 6402. 48886.]]
```

$f1$ for 1
0.7878465312167353
$f1$ for 2
0.5115202037744587
$f1$ for 3
0.581826794031736 $f1$ for 4
0.6559543746888155
$f1$ for 5
0.8477218537304374
$F1$ array is:

```
[0.7878465312167353, 0.5115202037744587, 0.581826794031736, 0.6559543746888155, 0.8477218537304374]
```

Average F1 score is: 0.6769739514884365
On training on full dataset we see many improvements:

1. A increase of about 11% in accuracy

2. A whopping increase of about 20% in macro $f1$ score

3. A whopping increase of about 40% in $f1$ score of 2 stars and 3 stars which were not being predicted nicely before.

4. Fair distribution two stars and three stars in confusion matrix as opposed to previous models.

So we see many noteworty improvements stated above.

2 MNIST Handwritten digit Classification

2.1 Binary Classification

My entry number ends with 5, so I classified for 5 and 6.

2.1.1 a

Number of support vectors: 233 out of all 4000
Support vectors are in the file *svmbinarysupportvector.txt*
w is in the file *svmbinaryw.txt*
b was found to be -2.407894871839998
Test set accuracy was found to be: 96.75675675675676%

2.1.2 b

Number of support vectors: 1496 out of all 4000
Support vectors are in the file *svmbinarysupportgaussian.txt*
b was found to be -0.13989252271787814
Test set accuracy was found to be: 99.18918918918919
As compared to linear model the accuracy of gaussian kernel model is 2.43% more which is a good thing.

2.1.3 c

Test set accuracy for linear kernel is: 97.2973% (1800/1850) (classification)
Test set accuracy for gaussian kernel is: 99.1892% (1835/1850) (classification)
Value of bias for linear kernel is -1.62 which is less than that of cvxopt linear kernel
Value of bias for gaussian kernel is -0.14 which is same as that of cvxopt gaussian kernel
Number of support vector for linear kernel is 233 which is same as of cvxopt linear kernel.
Number of support vector for gaussian kernel is 1478 which is slightly less than that of cvxopt gaussian kernel.
Training time of linear kernel with cvxopt was: 28.27 seconds
Training time of gaussian kernel with cvxopt was: 26.48 seconds
Training time of linear kernel with libsvm was: 1.26 seconds
Trainng time of gaussian kernel with libsvm was: 5.12 seconds

2.2 Multi-Class Classification

2.2.1 a

Test set accuracy obtained was: 97.23% (9723/10000)
Train set accuracy obtained was: 99.92% (9992/10000)

2.2.2 b

Test set accuracy obtained was: 97.23% (9723/10000) which is same as obtained in part a.
Train set accuracy obtained was: 99.92% (19984/20000) which is same as obtained in part a.
Training time with cvxopt was 1132 seconds
Training time with libsvm was 223 seconds
So we can see the accuracy are same but time taken by libsvm is 1/5th of taken by cvxopt.

2.2.3 c

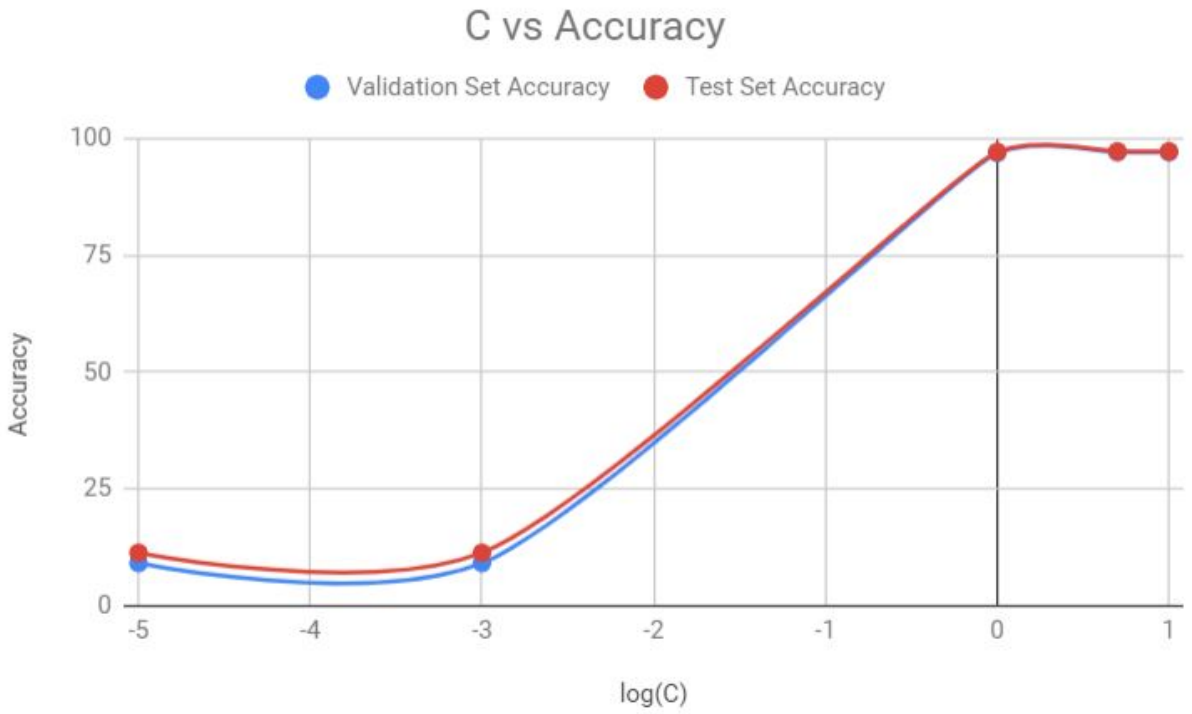
Confusion matrix obtained by testing over test set is:

```
[[ 969    0    4    0    0    2    6    1    4    5]
 [   0 1121    0    0    0    0    3    4    0    4]
 [   1    3 1000    8    4    3    0   19    3    3]
 [   0    2    4  984    0    6    0    2   10    8]
 [   0    1    2    0  962    1    4    4    2   13]
 [   3    2    0    4    0  866    4    0    5    3]
 [   4    2    1    0    6    7  939    0    1    0]
 [   1    0    6    6    0    1    0  987    3    7]
 [   2    3   15    5    2    5    2    2  943   14]
 [   0    1    0    3    8    1    0    9    3  952]]
```

We observe that 2,7 and 9 are being identified incorrectly the most. The digit 2 and 8 are predicted wrongly by the classifier often.
The digit 9 is being classified into other digits the most.
Yes the results make sense. The digits 2 and 7 are confused the most. False positive of 7 as 2 and 2 as 7 are high which is possible because 2 and 7 are quite similar when handdrawn. Resemblance of 9 to other numbers is also displayed by the confusion matrix.

2.2.4 d

C	Validation Set Accuracy	Test Set Accuracy
10^{-5}	9.15	11.35
10^{-3}	9.15	11.35
1	96.85	97.08
5	97	97.2
10	97	97.2



C equal to 5 or 10 gives the best validation set accuracy.
Yes they also give the best test set accuracy.

We observe that value of C 10^{-5} and 10^{-3} give very bad validation and test set accuracy. Value of C as 1 gives better accuracy. But value of C as 5 or 10 give the best validation and test set accuracy.
Setting value of C = 5 and training on full data set and running on test set resulted in the test accuracy of 97.31% which is more than 97.23% obtained with C=1.
Thus Validation set algorithm helped us in finding better value of C.