Rachit Kumar

2017CS10364

# COL774 Assignment2

## Q1. Naive Bayes: (Text Classification)

(a) Implemented with smoothing constant=1 and using logarithms to avoid underflow.

Over training set: Accuracy = 82.8669375

Over test set: Accuracy = 79.94428969359332

(b) Random and Majority Accuracy = 50

Hence, the improvement is around of 29.944

(c) Confusion Matrices:

Training Data:

[[ 742996 217125 ]

[ 57004 582875 ]]

Test Data:

[[ 152 47 ]

[ 25 135 ]]

The negative sentiment has the highest diagonal entry. Hence, we see that it is easier to identify negative sentiments as its correct entries are more and incorrect ones are less. It also means that our model favours negative sentiment as it predicts more negative sentiments than positive.

(d) Over training set: Accuracy = 82.4226875

Confusion Matrix:

[[ 674177 155414 ]

[ 125823 644586 ]]

Over test set: Accuracy = 81.8941504178273

Confusion Matrix:

[[ 144 32 ]

[ 33 150 ]]

The accuracy increased over the test set as we removed the noise and clubbed/considered similar words together.

(e) I tried using bigrams and trigrams using both with stemming and without stemming.

Bigrams without stemming:

Accuracy=83.56545961002786

Confusion Matrix:

[[ 157 39 ]

[ 20 143 ]]

Bigrams with stemming:

Accuracy=79.66573816155989

Confusion Matrix:

[[ 141 37 ]

[ 36 145 ]]

Trigrams without stemming:

Accuracy=76.32311977715878

Confusion Matrix:

[[ 133 41 ]

[ 44 141 ]]

Trigrams with stemming:

Accuracy=67.40947075208913

Confusion Matrix:

[[ 99 39 ]

[ 78 143 ]]

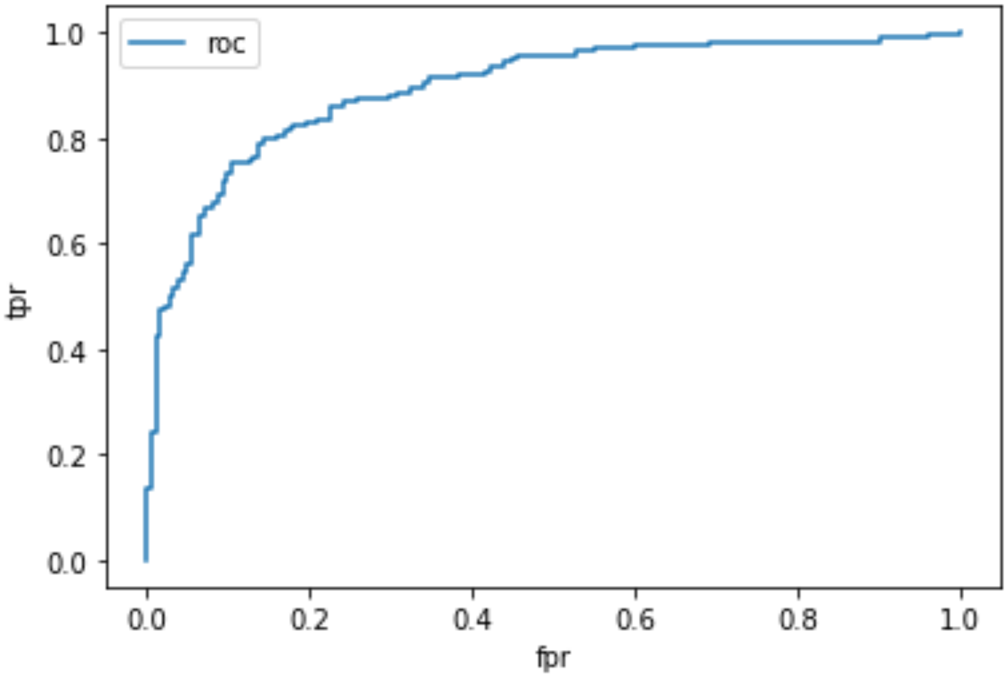
Hence, bigrams proved to be the best model(without stemming).

(f) The accuracy obtained was 50.69637883008357 and it took 6hrs (5hrs 56mins) to train the whole model using partial fit.

When I took 1/10th of the examples, the accuracy was 57.10306406685237 and it took 11 mins. Hence, it was concluded that gaussian distribution is not suitable for naive Bayes (and probably overfitting of the data is happening).

The accuracy obtained when I took 1/1000th of the features was 51.81058495821727 and it took 4mins 20sec. Hence we see that many features are useless and removing them will not affect the performance much, but will reduce the computation time significantly.

(g) The ROC curve is as given below(taking negative to be true). We observe that negative sentiment is predicted more and more correctly from the graph and this can be confirmed from the confusion matrix we made and observed earlier.



## Q2-1. SVM: (Binary Classification)

(a) Training Time=52sec

Test Accuracy=99.8

Validation Accuracy=100

b: [0.49688294]

no of support vectors: [16  57]

(b) Training Time=47sec

Test Accuracy=99.6

Validation Accuracy=100

b: [0.325361651389229]

no of support vectors: [391 617]

(c) Linear Kernel:

Training Time=0.34sec

Test Accuracy=99.8

Validation Accuracy=100

b: [0.49697655]

no of support vectors: [16  57]

Gaussian Kernel:

Training Time=4.2sec

Test Accuracy=99.6

Validation Accuracy=100

b: [0.32536799]

no of support vectors: [388  594]

w for the linear kernels for both our implementation and scikit's is the same and has been mentioned in the comments in the code(as it is too big).

## Q2-2. SVM: (Multi-Class Classification)

(a)  Test Accuracy=85.48

Validation Accuracy=94.74

The ties have been resolved using the sigmoid of the scores.

(b)  Test Accuracy=88.08

Validation Accuracy=96.92

Time for training= 4 mins

Time for training of our model(part (a)) =33 mins

Hence, our model is very slow compared to Scikit SVM's. Also, our accuracy was poorer which was probably due to difference in tie breaking conditions.

(c) Confusion matrix for part (a):

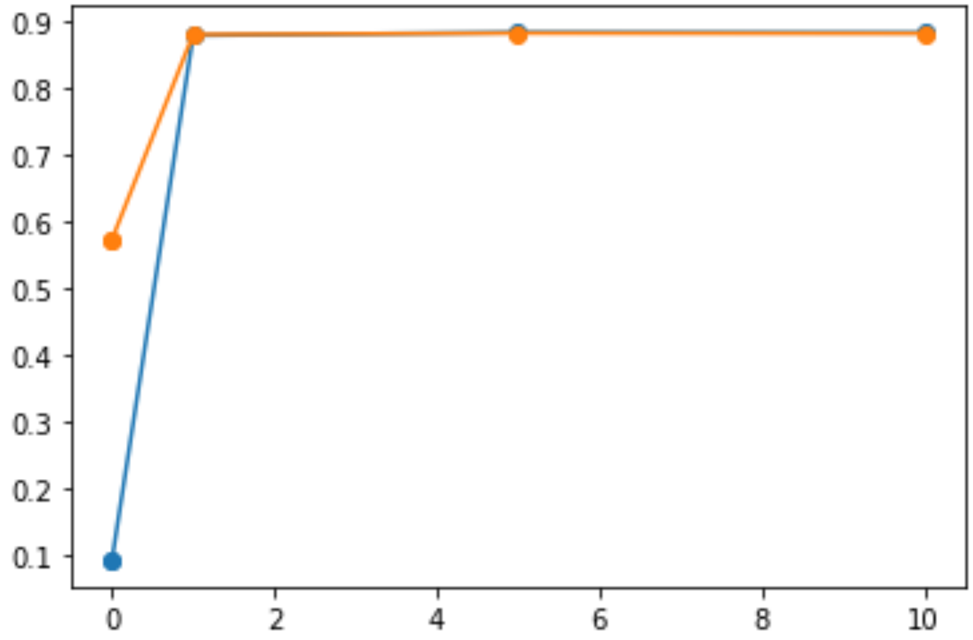
```
[[463   0   7   8   1   0  11   0  10   0]
 [  2 484   4   9   0   0   1   0   0   0]
 [ 13   0 423   7  24   0  22   0  11   0]
 [ 25   2   3 457   3   0   5   0   5   0]
 [  6   1  74  63 308   0  40   0   8   0]
 [  0   0   0   0   0 474   0  16   5   5]
 [162   0  69  10  18   0 231   0  10   0]
 [  0   0   0   0   0  14   0 471   1  14]
 [  1   0   1   2   0   2   3   2 489   0]
 [  0   0   0   0   0  11   0  14   1 474]]
```

Confusion matrix for part (b):

```
[[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]]
```

The similar looking items like t-shirt and shirt are often misclassified whereas the very distinct items like shirts and sandals are almost never misclassified.

(d)



Validation accuracies=

[0.09288888888888888, 0.09288888888888888, 0.8792888888888889, 0.8839555555555556, 0.8836444444444445]

Test Accuracies=

[0.5736, 0.5736, 0.8808, 0.8828, 0.8824]

The best value of C is 5, which gave best accuracy on both validation and test data.