COL 774: Machine Learning. Assignment 2

Shreya Sharma - 2017CS50493

PART- 1: Text Classification :

a) Implement the Naı¨ve Bayes algorithm to classify each of the tweets into one of the given categories :

Test accuracy = 80.78%
Training accuracy = 84.61%

b) What is the test set accuracy that you would obtain by randomly guessing one of the categories as the target class for each of the review (random prediction).
Random Predictor Accuracy :
On test data = 50.975%

What accuracy would you obtain if you simply predicted the class which occurs most of the time in the training data (majority prediction)?
Majority Predictor Accuracy :
On test data = 50.696%

How much improvement does your algorithm give over the random/majority baseline?
My algorithm gives a very significant improvement in predicting the tweet sentiments compared to the random or majority baseline predictor (at least 30-31% more accurate).

c) Draw the confusion matrix for your results in the part (a) above (for the test data only). Which category has the highest value of the diagonal entry? What does that mean? What other observations can you draw from the confusion matrix? Include the confusion matrix in your submission and explain your observations.
For testing data: Confusion Matrix

```
[157.,  49.],
[ 20., 133.]
```

The entry (0,0) of the confusion matrix which is class1 actual and class1 predicted has the maximum value which is Y = 0 (Negative sentiments)

For a 100% accuracy model, the confusion matrix is a diagonal matrix. We see that for a model with significant accuracy on testing data we get high diagonal values and low nondiagonal values, also true in this case. But the entry for class1 actual but class2 predicted is also significant.

For training data:

[751826., 198010.],
[48174., 601990.]

d) Perform stemming and remove the stop-words in the training as well as the test data. Also, remove Twitter username handles from the tweets. Learn a new model on the transformed data. Again, report the accuracy. How does your accuracy change over the test set? Comment on your observations.

Test Accuracy = 82.173%

Test accuracy increased from 80.78% to 82.173%. This is due to the pre-processing of raw tweet data by removing redundant stopwords and twitter handles which give no information on the tweet sentiment and only increase the number of redundant features. Also doing stemming helps reduce features as it converts words to their root, hence giving better classification.

e) Come up with at least two alternative features and learn a new model based on those features. Add them on top of your model obtained in part (d) above. Compare with the test set accuracy that you obtained in parts (a) and parts (d). Which features help you improve your overall accuracy? Comment on your observations.

Feature 1:

Bigrams + Monograms Accuracy = 83.008%

Using bigrams in addition to monograms helps augment the feature space and so improves the accuracy of the model on testing data.

Feature 2:

Accuracy = 81.058%

First categorized each word in the vocabulary as being in positive sentiment set or negative one based on the weightage of that word in the vocabulary set of tweets belonging to a particular class. Added the logarithm of the ratio of #postive word and #negative words in a tweet to get the probability for it to belong to a class.

f)

i) Use TF-IDF features with Gaussian Naive Bayes Model to predict the sentiments of the given tweets. How does your accuracy change over the test set?

Test Accuracy = 77.716%

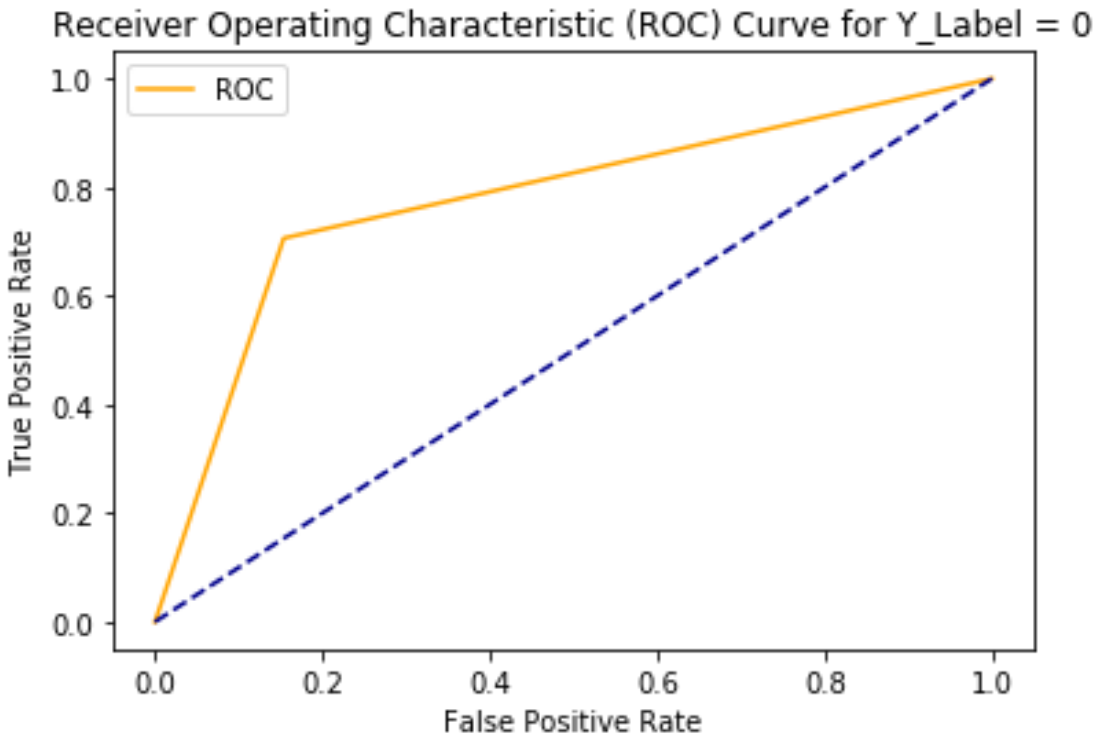
ii) Use Scikit-learn's SelectPercentile to choose features with the highest scores. Report your accuracy over the test set. Does selecting a smaller set of features improve your

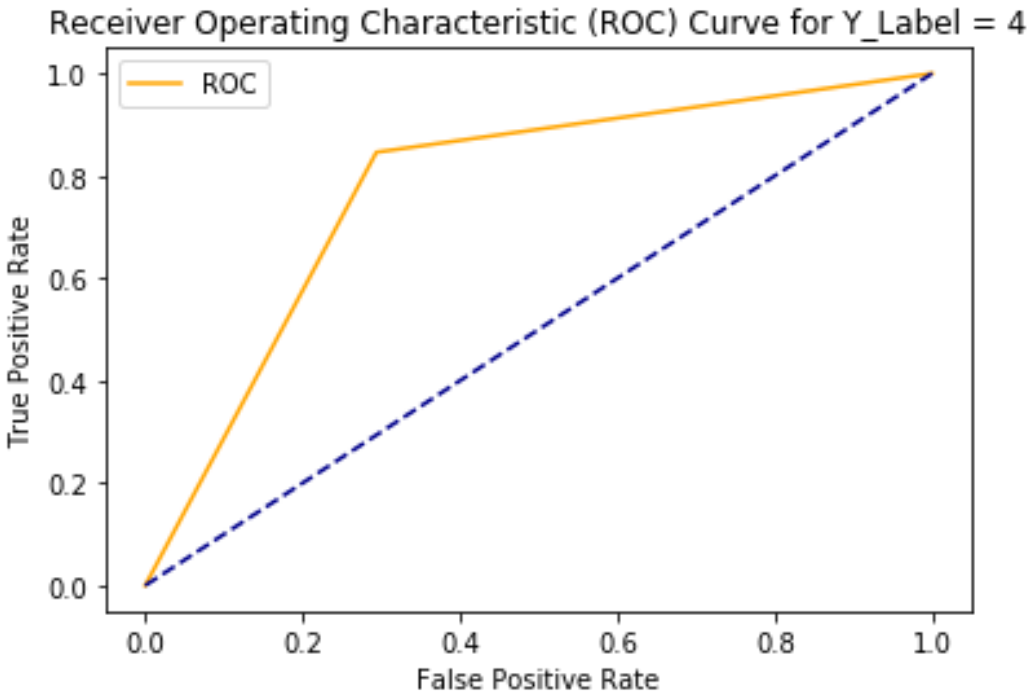
time taken to train the model? YES but it is a trade-off between training time and accuracy on the test data.

Compare the time taken in both cases:

Percentile (%)	Accuracy (in %)	Time Taken (in sec)
10	64.9	1.863
20	73.3	4.02
50	75.48	9.713
100	77.7	18.6

g) Plot the suitable ROC curve as per the problem requirement and comment on your observation.





PART- 1: Fashion MNIST Article Classification :

1. Binary Classification -

a) Linear SVM Classifier: Calculate the weight vector ‘w’ and the intercept term ‘b’ and classify each of the examples in the validation and test file into one of the two labels. Report the validation and test set accuracy obtained.
Test Accuracy = 94.5%
Validation Accuracy = 93.6%

b) SVM Gaussian Classifier: Use your learned model to classify the validation and test examples and report the accuracy obtained.
Test Accuracy = 96.1%
Validation Accuracy = 96.0%

c) Using Scikit SVM Package report accuracy on validation and test set for both linear and Gaussian kernels. Furthermore, compare weight (w), bias (b) and nSV (# of Support Vectors) with your implementation in

(a) for linear kernel

Support vectors in Linear SVM Model = 495
Support vectors in Linear sklearn.svm= 493
Difference in w = [0.0067307]
Difference in b = [0.21715703]
Accuracy Linear SVM Model = 93.6%
Accuracy Linear sklearn.svm = 93.4%
Training time case 1: 164.318s
Training time case2: 3.201s

(b) for Gaussian kernel

Support vectors in Linear SVM Model = 1238
Support vectors in Linear sklearn.svm= 1259
Difference in b = [0.01568324]
Accuracy Linear SVM Model = 96%
Accuracy Linear sklearn.svm = 96.4%
Training time case 1: 200.459s
Training time case2: 6.114s

2. Multi-Classification -

a) Using your solver from the previous section, implement one-vs-one multi-class SVM.Use a Gaussian Kernel withC= 1.0 andγ= 0.05. Classify the given dataset and report validation and test set accuracy. In case of ties, choose the label with the highest score.

Test Accuracy = 83.68%
Validation Accuracy = 84.2%

b) Now train a multi-class SVM on this dataset using the Scikit SVM library. Repeat part(a) using a Gaussian kernel withγ= 0.05. UseC= 1.0 as earlier. Report the validation as well as test set accuracy. How do your results compare with those obtained in part (a) above? As earlier, compare the computational cost (training time) of the two implementations? Comment.

Accuracy test = 88.08%
Accuracy validation = 82.16%

Support vectors = 11665
Support vectors = 1952

c) Draw the confusion matrix as done in the first Part (Naive Bayes) of this assignment for both of the above parts (2(a) and 2(b)). What do you observe? Which articles are misclassified into which ones most often? Do the results make sense?

Confusion Matrix:

2a) Test Data :

```
[[405.  0.  8. 10.  0.  0. 58.  0. 19.  0.]
 [ 1. 481.  5.  4.  0.  0.  6.  0.  3.  0.]
 [ 2.  0. 413.  5. 28.  0. 38.  0. 14.  0.]
 [ 20.  7.  1. 412.  6.  0. 40.  0. 14.  0.]
 [ 1.  1. 75. 12. 335.  0. 61.  0. 15.  0.]
 [ 0.  0.  0.  0.  0. 401.  0. 12. 77. 10.]
 [ 64.  1. 63.  4. 20.  0. 330.  0. 18.  0.]
 [ 0.  0.  0.  0.  0. 25.  0. 432.  9. 34.]
 [ 0.  0.  1.  0.  0.  1.  5.  0. 493.  0.]
 [ 0.  0.  0.  0.  0.  6.  0.  7.  5. 482.]]
```

Validation Data:

```
[[208.  0.  1.  4.  0.  0. 32.  0.  5.  0.]
 [ 1. 235.  2.  6.  0.  0.  3.  0.  3.  0.]
 [ 3.  0. 210.  0. 12.  0. 11.  0. 14.  0.]
 [ 13.  5.  0. 196.  8.  0. 20.  0.  8.  0.]
 [ 0.  1. 40.  5. 174.  0. 22.  0.  8.  0.]
 [ 0.  0.  0.  1.  0. 212.  0.  3. 27.  7.]
 [ 23.  0. 31.  2. 11.  0. 176.  0.  7.  0.]
 [ 0.  0.  0.  0.  0. 11.  0. 211.  9. 19.]
 [ 0.  0.  1.  0.  0.  0.  2.  2. 245.  0.]
 [ 0.  0.  0.  0.  0.  1.  0.  6.  5. 238.]]
```

2b) Test Data:

```
([[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]])

Validation Data:

([[382, 0, 10, 25, 1, 1, 63, 0, 18, 0],
[2, 467, 10, 14, 2, 0, 5, 0, 0, 0],
[4, 0, 370, 3, 66, 0, 45, 0, 12, 0],
[26, 2, 1, 432, 10, 0, 23, 0, 6, 0],
[1, 1, 60, 24, 362, 0, 41, 0, 11, 0],
[0, 0, 0, 0, 0, 443, 0, 26, 24, 7],
[78, 0, 78, 16, 46, 0, 263, 0, 19, 0],
[0, 0, 0, 0, 0, 23, 0, 447, 1, 29],
[0, 0, 1, 3, 0, 10, 4, 4, 477, 1],
[0, 0, 0, 0, 0, 17, 0, 12, 6, 465]])

Class 6 is often classified as class 0. This makes sense as there are high chances for a shirt (class 6) to be classified as a T-shirt (class 0) as they are highly similar and difficult to differentiate.

d) For this problem, we will do 5-fold cross-validation to estimate the best value of the parameter for the Gaussian kernel case. Test data should not be touched. Fixγas 0.05 and vary the value of C in the set$\{10^{-5},10^{-3},1,5,10\}$and compute the 5-fold cross-validation accuracy for each value of C. Also, compute the corresponding accuracy on the test set. Now, plot both the 5-fold cross-validation accuracy as well as the test set accuracy on a graph as you vary the value of C on the x-axis (you may use log scale on the x-axis). What do you observe? Which value of C gives the best 5-fold cross-validation accuracy? Does this value of the C also give the best test set accuracy? Comment on your observations.

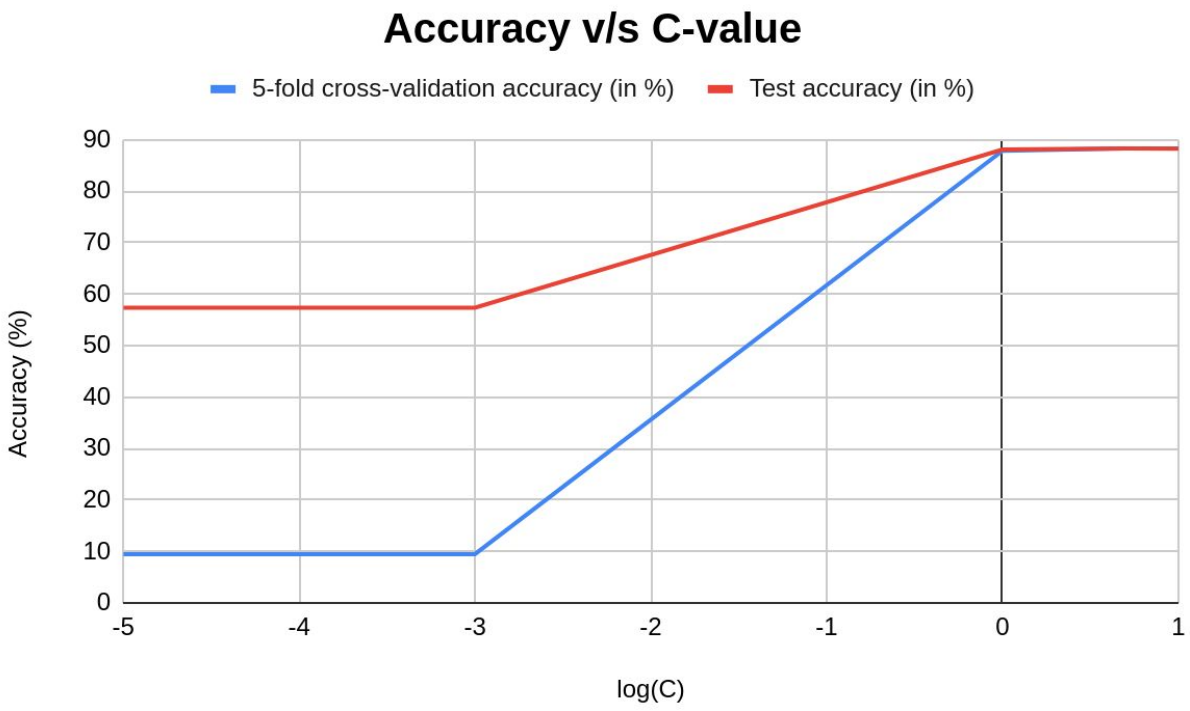
C = 5 and 10 give the best 5-fold cross-validation accuracy.

Value of C	5-fold cross-validation accuracy (in %)
0.00001	9.467
0.001	9.467

1	87.853
5	88.324
10	88.324

C = 5 gives the best test accuracy.

Value of C	Test accuracy (in %)
0.00001	57.36
0.001	57.36
1	88.08
5	88.28
10	88.24



Since the same value of C gives the best cross-validation accuracy and test accuracy, we can say that cross-validation helps us find the parameters which generalize the model the most.