Name: Mukka Teja

Net Id: TXM162230

Machine Learning Assignment (CS 7301.006)

(Naive Bayes and logistic Regression) (Note: Runs in python 2.7) (Note: File Path should be full path)

Zip File name: TXM162230_ML_Assignment2.zip

Folder Name: TXM162230_ML_Assignment2

Folder Structure:

I have kept the folder on my desktop with folder name as TXM162230_ML_Assignment2

I am using python 2.7 version for my code.

Results:

Naive Bayes with stop words	Accuracy: 94.9790794979
Naïve Bayes without stop words	Accuracy: 94.3514644351
LR with stop words (For eta =0.01 and lambda = 0.5) # num of iterations =100	Accuracy = 91.004
LR without stop words (For eta =0.01 and lambda = 0.5) # num of iterations =100	Accuracy = 93.72
LR with stop words (For eta =0.01 and lambda = 5) # num of iterations =100	Accuracy = 92.35
LR without stop words (For eta =0.01 and lambda = 5) # num of iterations =100	Accuracy = 93.3
LR with stop words (For eta =0.01 and lambda = 10) # num of iterations =100	Accuracy = 86.19
LR without stop words (For eta =0.01 and lambda = 10) # num of iterations =100	Accuracy = 93.09
NB with better feature extraction	Accuracy =96.44
LR with better feature extraction	Accuracy = 95.81

Reasoning:

Why accuracy does not change much for NB:?

1. There is not so significant difference in the accuracies of NB with and without stopwords. There can be two reasons for this:
    1. Stop words were not collected properly and hence removal does not affect the accuracy.
    2. Stop words are equal likely to be present in both ham and spam frequently and does not really help in classifying one from other. Most of the stop words are present equal likely in both spam and ham, hence removing them do not have much impact.

Feature extraction on NB which improve the accuracy:

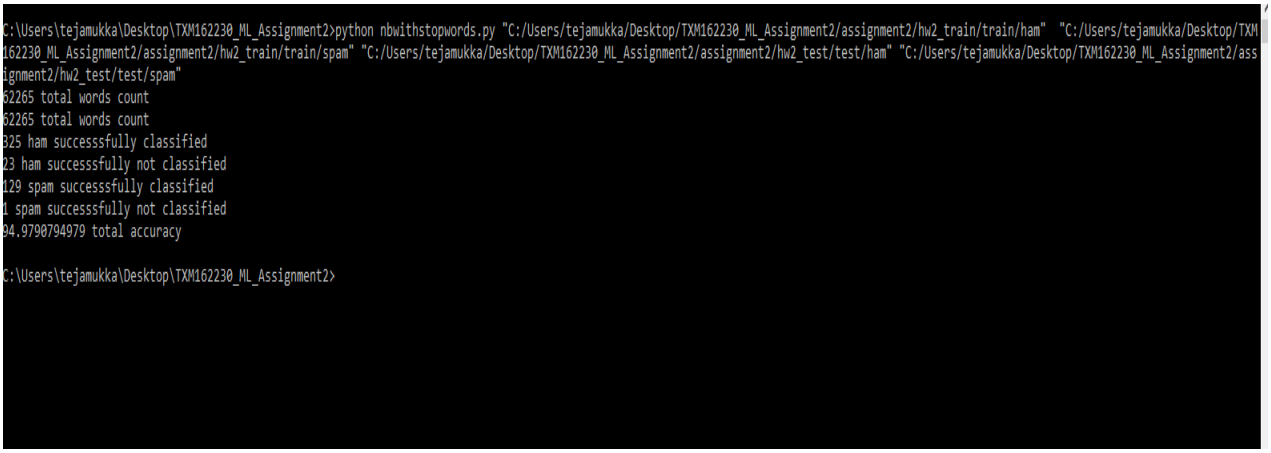
3. To strengthen my argument on the first point, that the stop words are not collected properly. I have observed most of the stop words were collected in the sense, combined with /n, two words combined, some special characters next to it. So I changed my logic to parse the text by splitting based on anything that are not words, and hence the accuracy has been increased to 96.44 from 94.7 in NB.(Extra credit implementation question)
   Effect of lambda on the accuracy and on the smoothing of the curve.
4. For logistic regression, as the lambda increases the curve becomes more smoother and reaches the optimal point quickly. The curve becomes more smoother.
5. With the increase in the lambda, the accuracy decreases as it dominates the weights evaluations and causes incorrect weights.
   Why removal of stop words increased accuracy for LR:
6. There is a 2 percent increase in the LR accuracy when the stop words has been removed. As mentioned above, just based on the stop words you cannot distinguish spam from ham. Hence removing these weights, the dimensions(features) of the problem decreases and hence increases the accuracy.

Feature extraction to improve the accuracy:

7. Better feature extraction has been done using the split with any character that is not word and improved the accuracy by 2 % in NB.
8. Computed the tf-idf (cal frequency of each token and weights accordingly)which finds the weights of each token and removes few tokens which helps us in concentrating more on the important features.

1) Go to ‘command prompt’

1. Naïve Bayes including stop words :



Input:

python nbwithstopwords.py “trainhampath” “trainspampath” “testhampath” “testspampath”

python nbwithstopwords.py
"C:/Users/tejamukka/Desktop/TXM162230_ML_Assignment2/assignment2/hw2_train/train/ham"
"C:/Users/tejamukka/Desktop/TXM162230_ML_Assignment2/assignment2/hw2_train/train/spam"
"C:/Users/tejamukka/Desktop/TXM162230_ML_Assignment2/assignment2/hw2_test/test/ham"
"C:/Users/tejamukka/Desktop/TXM162230_ML_Assignment2/assignment2/hw2_test/test/spam"

Displays the total number of tokens, correct and incorrect classification of both spam and ham and total accuracy along with it.

Output: (94.9790794979 total accuracy)

62265 total words count

62265 total words count

325 ham successfully classified

23 ham successfully not classified

129 spam successfully classified

1 spam successfully not classified

94.9790794979 total accuracy

2. Naïve Bayes without stop words:

Input: Please note extra argument in the input here (stopwordspath):

python nbwithstopwords.py “trainhampath” “trainspampath” “testhampath” “testspampath” “stopwordspath”

python nbwithoutstopwords.py
"C:/Users/tejamukka/Desktop/TXM162230_ML_Assignment2/assignment2/hw2_train/train/ham"
"C:/Users/tejamukka/Desktop/TXM162230_ML_Assignment2/assignment2/hw2_train/train/spam"
"C:/Users/tejamukka/Desktop/TXM162230_ML_Assignment2/assignment2/hw2_test/test/ham"
"C:/Users/tejamukka/Desktop/TXM162230_ML_Assignment2/assignment2/hw2_test/test/spam"
"C:/Users/tejamukka/Desktop/TXM162230_ML_Assignment2/assignment2/stopwords.txt"

Displays the total number of tokens, correct and incorrect classification of both spam and ham and total accuracy along with it.

Output (94.3514644351 total accuracy)

49524 total words count

49524 total words count

322 ham successfully classified

26 ham successfully not classified

129 spam successfully classified

1 spam successfully not classified

94.3514644351 total accuracy

3.Naïve Bayes with feature extraction: (Bonus points question☺)

Input:

python nbwithstopwords.py “trainhampath” “trainspampath” “testhampath” “testspampath”

python nbfeatureextraction.py
"C:/Users/tejamukka/Desktop/TXM162230_ML_Assignment2/assignment2/hw2_train/train/ham"
"C:/Users/tejamukka/Desktop/TXM162230_ML_Assignment2/assignment2/hw2_train/train/spam"
"C:/Users/tejamukka/Desktop/TXM162230_ML_Assignment2/assignment2/hw2_test/test/ham"
"C:/Users/tejamukka/Desktop/TXM162230_ML_Assignment2/assignment2/hw2_test/test/spam"

Displays the total number of tokens, correct and incorrect classification of both spam and ham and total accuracy along with it.

Output:(Accuracy = 96.44)

333 ham successfully classified

15 ham successfully not classified

128 spam successfully classified

2 spam successfully not classified

96.4435146444 total accuracy

4. Logistic Regression including stopwords :

Note: For LR I have used Numpy but could not get it installed on windows machines. Hence tested this with bash on windows, hence only the relative paths are given. On windows absolute may be required like in the earlier files.

Input:

Python lrwithstopwords.py "trainpathonlytilltrainfolder" "testpathonlytilltestfolder" "

Tested on Bash on windows: (for windows cmd, absolute might be needed☺)

python lrwithstopwords.py assignment2/hw2_train/train assignment2/hw2_test/test assignment2/stopwords.txt

For eta =0.01 and lambda = 0.5

Displays the total number of tokens, correct and incorrect classification of both spam and ham and total accuracy along with it.

Output: 91.14 total accuracy

327 ham successfully classified

21 ham successfully not classified

98 spam successfully classified

32 spam successfully not classified

435

478

91.14 total accuracy

5. Logistic Regression without stop words :

Note: For LR I have used Numpy but could not get it installed on windows machines. Hence tested this **with bash on windows**, hence only the relative paths are given. On windows absolutepath, may be required like in the earlier files.

Input:

Python lrwithstopwords.py “trainpathonlytilltrainfolder” “testpathonlytilltestfolder” “

Tested on Bash on windows: (for windows cmd, absolute might be needed☺)

python lrwithoutstopwords.py assignment2/hw2_train/train assignment2/hw2_test/test assignment2/stopwords.txt

Displays the total number of tokens, correct and incorrect classification of both spam and ham and total accuracy along with it.

For eta =0.01 and lambda = 0.5

Output: (93 total accuracy)

330 ham successsfully classified

18 ham successsfully not classified

118 spam successsfully classified

12 spam successsfully not classified

448

478

93 total accuracy

5. Logistic Regression feature extraction :

Note: For LR I have used Numpy but could not get it installed on windows machines. Hence tested this **with bash on windows**, hence only the relative paths are given. On windows absolutepath, may be required like in the earlier files.

Input:

Python lrwithstopwords.py “trainpathonlytilltrainfolder” “testpathonlytilltestfolder” “

Tested on Bash on windows: (for windows cmd, absolute might be needed☺)

python lrwithoutstopwords.py assignment2/hw2_train/train assignment2/hw2_test/test assignment2/stopwords.txt

Displays the total number of tokens, correct and incorrect classification of both spam and ham and total accuracy along with it.

For eta =0.01 and lambda = 0.5

Output: (93.72 total accuracy)

330 ham successsfully classified

18 ham successsfully not classified

118 spam successsfully classified

12 spam successsfully not classified

448

478

93.72 total accuracy

Conclusion:

It was a very interesting project. Learnt how to implement NB and LR. Learnt how the removal of stop words affect the accuracy. Learnt how the changes in the lambda affects the accuracy in the LR, Methods of improving the accuracy.