ASSIGNMENT 4

Course : Foundations of Machine Learning (**CS564**)
Department of CSE, IIT Patna

Deadline for Submission: November 02, 2018

Marks:- 20 Marks

Instructions:

1. The assignment should be completed and uploaded by (**date of submission**)
2. Make a group of 3 for it(group as Assignment 1 and 2) .
3. Markings will be based on the correctness and soundness of the outputs. Marks will be ***deducted in case of plagiarism***.
4. Be precise for your explanations in the report. Unnecessary verbosity will be penalized. Prepare a Detailed report of the assignment.
5. You should zip all the required files and name the zip file as ***Group_no*.zip**.
6. Upload your assignment (**the zip file**) in the following link: https://www.dropbox.com/request/gXuZGoBQMixIox9FplbH

Dataset: Twitter Sentiment analysis dataset

Dataset: Attached
Dataset Description: No of Tweets: 7086
No of classes for prediction: 0 (Negative Sentiment), 1 (Positive Sentiment)
Train the model using 75 % data
Test using 25 %

Preprocessing

(Tweets should brought in usable format before feeding it to classifier hence we have the following preprocessing to be done)

Step1 : Remove URLs, Hashtag (i.e #happy) and mentions (i.e @BarrackObama) from the tweets.
Step2 : All Punctuation should be removed except **apexes (i.e can't, isn't, don't etc these add negative sentiment hence should not be removed.)**

Step 3: All apexes or **Negative construct** like can’t, wouldn’t, wasn’t, hadn’t, never etc should be replaced by **not** (**ex wasn’t playing** → **not playing**).

Step 4: Reduce the number of Emoticons by categorizing it into two classes
smile_positive and smile_negative. Here is the list of emoticons and the word they should be replaced and emoticon outside it can be ignored and removed.

smile_positive	:-) :-] :-3 :-> 8-) :-} :o) :c) :^) =] =) :) :] :3 :> 8) :} :-D :D 8-D 8D x-D xD X-D XD =D =3 B^D :-)) :'-) :') :-O :O :-o :o :-0 8-0 >:O :-* :* :× ;-) ;) *-) *) ;-] ;] ;^) :-, ;D :-P :P X-P XP x-p xp :-p :p :-Þ :-Þ :-þ :þ :Þ :Þ :-b :b d: =p >:P O:-) O:) 0:-3 0:3 0:-) 0:) 0;^) |;-) :-J #-) %-) %) <3 @};- @}->-- @}-;-'--- @>-->--
smile_negative	:-(:(:-c :c :-< :< :-[:[:-|| >:[:{ :@ >:(:'-(:'(D-': D:< D: D8 D; D= DX :-/ :/ :-. >:\ :L =L :S :-| :| :-X :X :-# :# :-& :& >:-) >:) }:-) }:) 3:-) 3:) >;) ',:-l ',:-| >_> <_< <\3 </3

Step 5: Apply Stemming
Stemming techniques put word variations like “great”, “greatly”, “greatest”, and “greater” all into one bucket, effectively decreasing entropy and increasing the relevance of the concept of “great”. In other words, Stemming allows us to consider in the same way nouns, verbs and adverbs that have the same radix

Step 6: Remove Stopword like article and pronoun only.

Step 7: Finally, all the text is converted to lower case, and extra blank spaces are removed.

Note: For more information refer preprocessing part of paper in the link.
link : http://ceur-ws.org/Vol-1748/paper-06.pdf

Classification Algorithms: Implement Naive Bayes classifier (Java/Python) without using any external libraries . Use the multinomial model for implementation.

Resources/documents to be submitted:

1) Codes with proper documentation
2) Report Accuracy, Precision, Recall and F-measure for the classification model.