ASSIGNMENT 1 REPORT

*ANKIT SINGHAL (2017A7PS0100H)*

*AVANISH GHOSH (2017A71393H)*

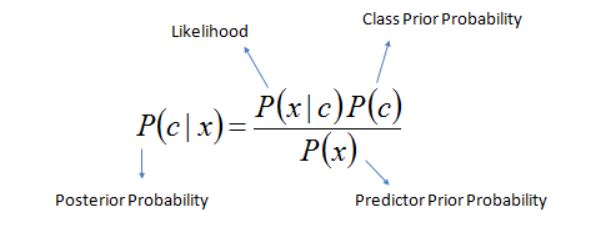
*NAMAN KHUNTETA (2017A7PS0162H*)

**Naïve Bayes for Sentiment Analysis:**

1. ***Introduction:***

* Language : Python
* Libraries Used : Numpy, nltk.tokenize (for words tokenization)
* Dataset : a1\_d3.txt

The crux of the classifier is based on the Bayes theorem :



Which gives us the probability of event c happening given event x has occurred.

1. ***Approach:***

Created 3 new files (good.txt, bad.txt, test.txt) for simple implementation.

good.txt – contains words that led to ‘good’ or ‘1’ sentiment.

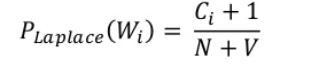
bad.txt – contains words that led to ‘bad’ or ‘0’ sentiment.

test.txt – contains test query or test lines.

Next, two dictionaries were created freq\_good and freq\_bad that held frequencies of all words present in good.txt and bad.txt respectively.

1. ***Pre-processing:***
2. Removed special characters mainly brackets ({,[,( ) , symbols such as (. , ‘ / $ #) and whitespaces from end of lines before adding a line to good.txt or bad.txt.
3. Converted all words in the file “a1\_d3.txt” to lowercase.
4. ***Handling zero frequency:***

To handle test query words which had zero frequency Laplace smoothing was performed for each word according to the formula :

where ,

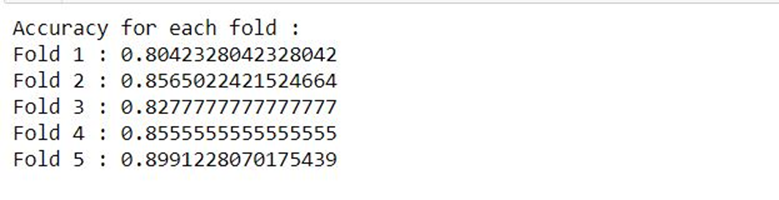
Ci = count of word[i]

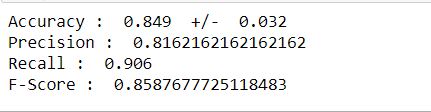
N = number of word tokens

V = number of words in specific dictionary

1. ***Results:***

Achieved an overall F-Score of 0.8587 and accuracy of 0.849 +/- 0.032





1. ***Conclusion:***

The classifier achieved an accuracy of almost 0.85 which is pretty satisfactory. We have assumed that the features are independent which may not be true in real life. This hinders the performance of the classifier.