Pattern Recognition & Machine Learning

Lab-5 :- *"Text based Bayes Classification"*

Objectives

1. Preprocess and Clean the Text-Based Dataset, according to the sensitivity of Bayesian Decision Equation.

2. Compute the Term Document Matrix for Whole Dataset as well as for both Classes.

3. Find out the frequency of words belonging to class 0 and 1. Use Laplace Smoothing and Predict the target class for Validation Data Portion.

4. Generate the Classification scores like f1 score, precision, recall. Also, find out the confusion matrix for predictions.

Dataset

Tweets Dataset :- ***train.csv*** , with columns as 'id','location','keyword','text' and 'target'.

Dependencies

Following were the Dependencies in order to fulfill the requirements of Lab-5 :-

1. Numpy and Pandas (For Data Manipulation)
2. Sklearn (For Model Score Purposes)
3. Seaborn , wordcloud and Matplotlib (For Visualisation of Graphs and Plots)
4. Re, pyspellchecker, string, Nltk and spacy (For Text based operations)

Preprocessing Methods

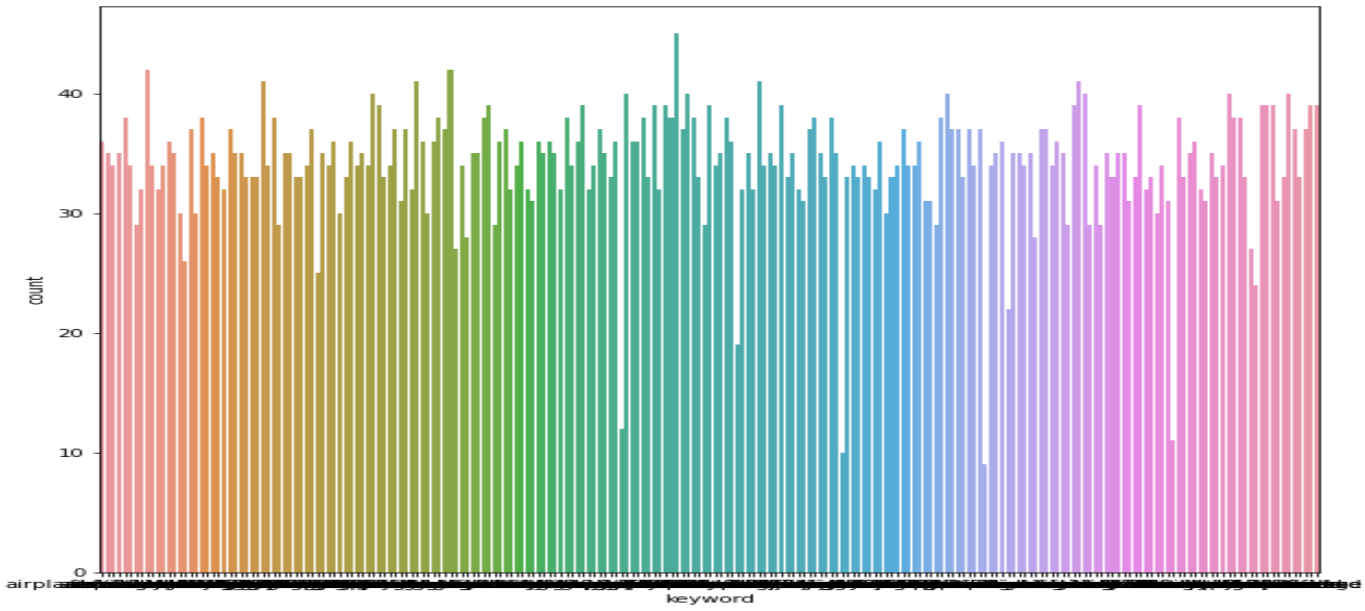
Following is the Sequential pipeline, which was incorporated for Dataset :-

1. Dropped NaN values from the Dataset.
2. Removed URLs and Converted Emoji & Emoticons into words.
3. Removed Punctuation Marks and Underscores.
4. Converted Chat Short Form Words into Normal Full Form Words.
5. Converted every alphabet's case into lowercase and removed numeric digits.
6. Removed Multiple Spaces, again Converted Residual Chat Short Words into Normal Words (for those words, which have their full forms in lower case).
7. Lowered each alphabet's case and used Spell Checker to correct the spellings of wrong words.

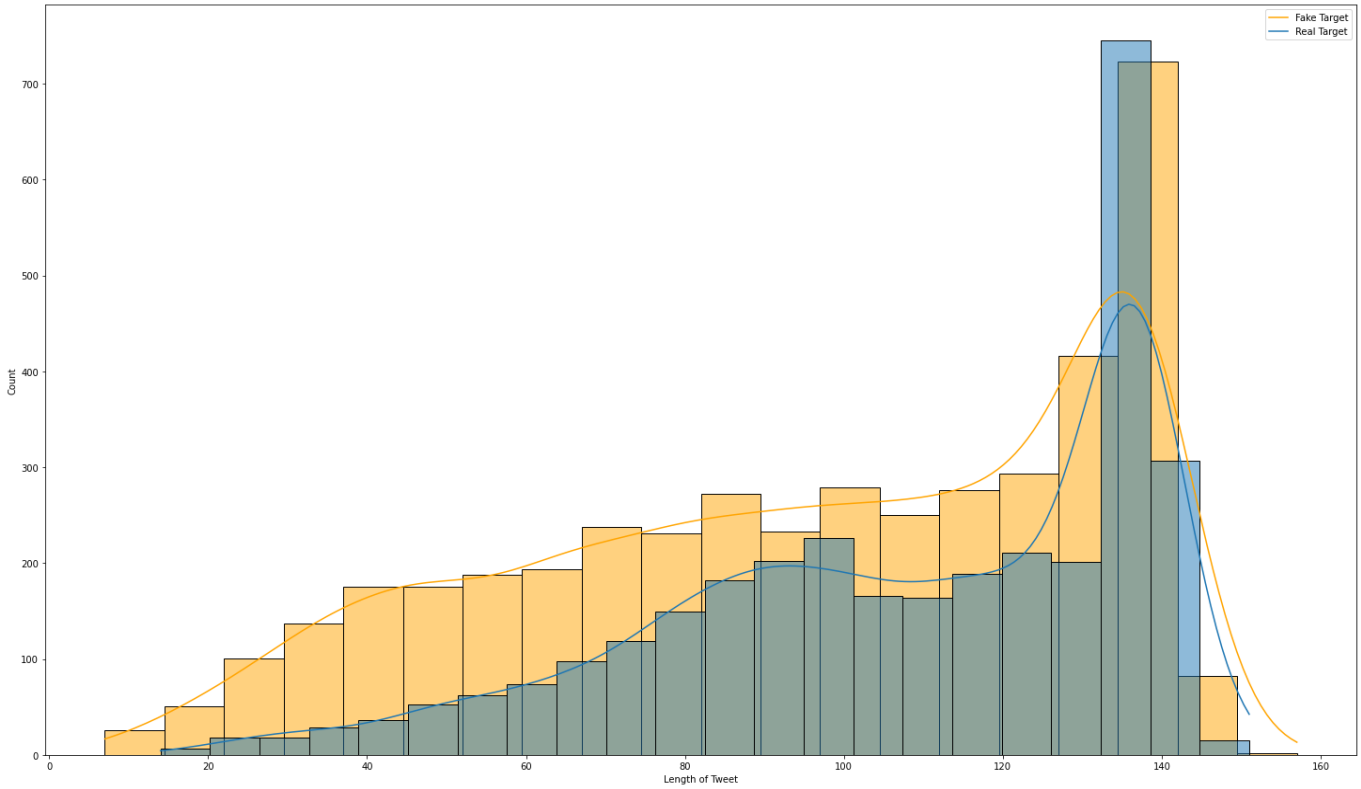
Graphs and Plots

Following are some useful Plots and Graphs, related to Dataset and Texts :-

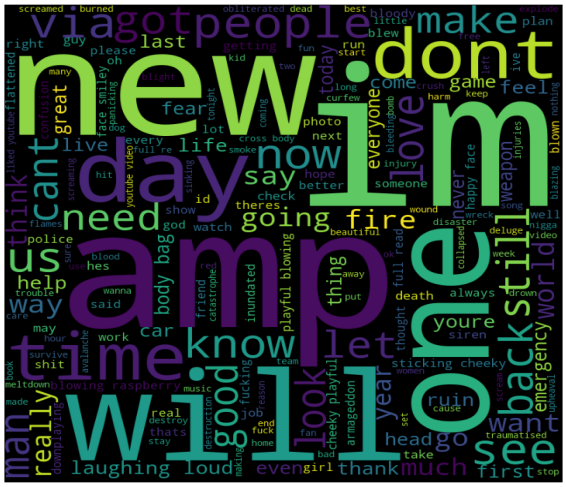
Count of each Keyword in Dataset



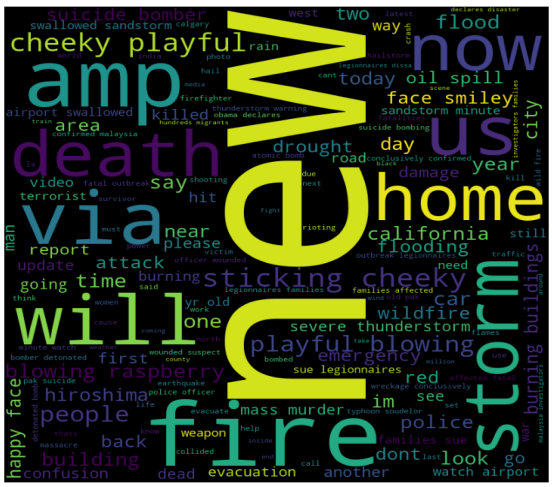
Distribution of Real and Fake Tweets with the Length of Tweets



Word Cloud for Class-0



Word Cloud for Class-1



Term Document Matrix

The Term Document Matrix consists of Columns, which represent every Unique word, occurring in the given Dataset Portion. On the other hand, Rows represent the frequency of words occurring in a sentence / piece of document.

For implementing Laplacian Smoothing, Text Document Matrix was used, to determine the number of sentences/tweets in which the unique word occurred, alongside considering the Total Number of Sentences belonging to Class for which likelihood would be found.

Bayesian Decision Equations

Here are some of the Equations, followed in Bayesian Decision Theory :-

$$posterior = \frac{likelihood \times prior}{evidence}.$$

In the Comparison, between Class 0 and Class 1 posterior probabilities, evidence for a validation sentence will remain the same, so in this Lab, there will be no requirement of calculating Evidence.

For Posterior Probability in Multi-Feature Dataset with each feature acting independently, the formula goes as following :-

$$p(C_k|x_1, \ldots, x_n) = p(C_k) \prod_{i=1}^{n} \frac{p(x_i|C_k)}{p(x_i)}$$

The requirement will only be to calculate the ratio of Class 0 and Class 1 Posterior Probabilities and decide accordingly, that if the ratio is greater than or equal to 1, then predict 0 as target class otherwise predict 1 as target class.

Laplacian Smoothing

It comes into picture, when a certain test data feature doesn't contain as a training example in a Model. Because of this, the Posterior Probability can result into 0. So, in order to avoid this, the formula comes, as following :-

$$P(w'|positive) = \frac{\text{number of reviews with w' and y = positive } + \ \alpha}{\text{N } + \ \alpha * K}$$

Where :-

w' -----> A datapoint for which likelihood was being calculated.

Alpha ---- smoothing parameter (Set as 1)

N ------> Total number of samples belonging to positive class (Specific to this example)

K ------> Number of Feature Columns (In this Lab's case, it was 1)

This formula was applied to each word for which the likelihood probability was calculated, whether belonging to training dataset or not. More the value of smoothing parameter, more will be the likelihood probability of words.

Classification Task Report

Following are the details, related to Classification Task :-

Class	Precision	Recall	F1-Score
0	0.66	0	0.01
1	0.40	1	0.61

With the Train-Test Split Ratio as 7:3 and smoothing parameter as 1, Accuracy was found out to be 43.7% . The Confusion Matrix formed, was:-

3	856
2	663

Conclusion

The reasons behind the biasness of Bayes Classification Model towards Class-1 , were as follows :-

1. Bayes Decision Equation and its corresponding Probabilities depend heavily on quantitative Distribution of features according to Split ratio and Class Segregation.

2. As Model is Data Sensitive, the intermediate results show unexpectedly higher dependency on the Type of Text Cleaning Techniques, with their order of implementation.

3. The Techniques varied the results of Bayesian Classification, with variation in presence of Numeric Values, Non-English Encoding Errors and Underscores. Also, Emoji and Emoticons affect the quality of information, as removing them reduces the essence of tweet and including them in the form of words, increases the Likelihood of certain emotion words like happy, smile, cry, etc.

4. While Calculating the Posterior Probabilities of each Validation Sentence/Tweet, the range was found out to be from 10**(-118) to 10**(-05).

5. The number of class-0 and class-1 unique words, when combined together, turned out to be not equal to unique words of the whole dataset. Infact, it was coming out to be greater than the total unique word count, because some of the unique words, which occurred both in class-0 and class-1, were counted as different entities., not same