Assignment - 1

Twitter hate speech detection using Naive Bayes and Logistic Regression

1 Introduction

This document is used to discuss the results and findings of the assignment 1 of Advanced Natural Language Processing course. The goal was to implement a Naive Bayes and a Logistic Regression classifier for a given twitter data-set classified into 'offensive' and 'nonoffensive' classes. First, results of Naive Bayes classifier are discussed along with feature engineering and then, results of Logistic Regression classifier are discussed.

2 Naive Bayes Classifier

Here, Naive Bayes classifier is implemented by first, calculating the priors for each class and, likelihood of each word occurring in vocabulary for each class. In the evaluation part, each test document (or tweet) is passed through the model, and probability for each class is calculated for that particular document. Then, the class with maximum probability is assigned to it. By comparing the true values of class with the predicted values of class for a particular test document, values of true positive, true negative, false positive and false negative are calculated, which are further used to calculate accuracy and F1 score.

For vanilla Naive Bayes model (k = 1), accuracy of 0.4897 and F1 score of 0.2625 are obtained. As the value of k is decreased, there is improvement in both accuracy and F1 score as can be seen in Figure 1, but when value of k is increased, there is a drop in values of accuracy and F1 score.

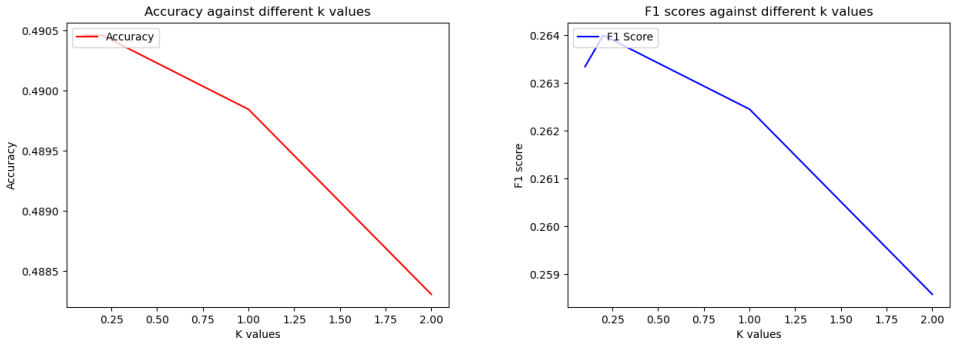

Figure 1: Values of Accuracy and F1 against different values of k

To further improve the model, additional features are implemented. First, all the words in the training data are stemmed to analyze the meaning behind the word. By using this feature, the model can find similarities between the words such as changed and change, for example. Without using stemming, the model would have considered both the words different than each other and assigned them different likelihood. This results in an improvement of accuracy from 0.4898 to 0.4993. Secondly, all the stop words in the training data are removed. By using this feature

all the likelihood assigned to stop words are removed and only the words important to the context of the document are considered. This results in an improvement of accuracy from 0.4898 to 0.4993 as shown in Figure 2.

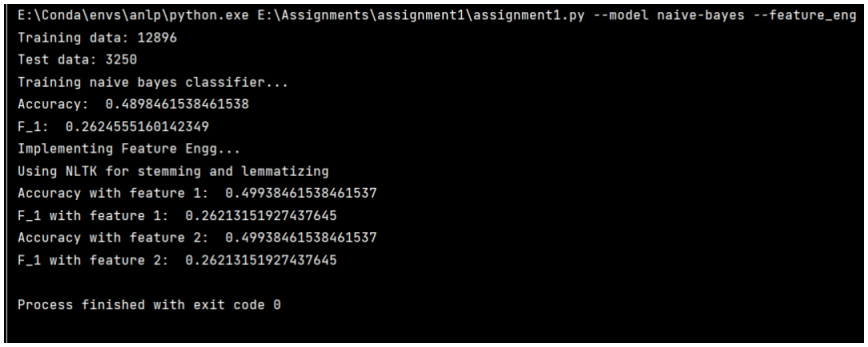

Figure 2: Values of accuracy and f1 score using feature engineering

3 Logistic Regression Classifier

Here, a neural network having a single computing unit is implemented. First, the data is encoded and then passed through this unit. Here the data is multiplied with the weights and then passed through a sigmoid function which gives the predicted class. Loss is calculated by comparing predicted and true values, which is back propagated to update the weights. This is done for a particular number of iterations. After this training process, the test data is passed through this model and using the learned weights, the class is predicted. By using this classifier, accuracy of 0.7443 is achieved which is a significant improvement over Naive Bayes classifier as shown in Figure 3.

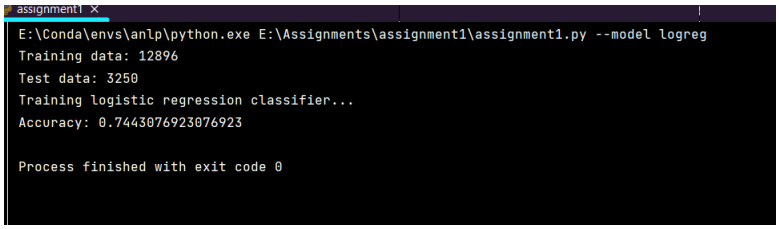

Figure 3: Values of accuracy for Logistic Regression classifier