**SAMUEL ATULE**

**SENTIMENT ANALYSIS REPORT ON LOGISTIC AND NAIVEBAYES MODEL**

**11/12/2018**

**Introduction**

This report compares the Naïve Bayes model and Logistic Regression model as sentiment classification models to determine which is the best and most appropriate for sentiment analysis. The data employed in this report contains binary class. That is, the model determines whether a review document contains a positive sentiment or a negative sentiment.

We will experiment and then discuss the accuracy of the two models on the same dataset. For each model, we will also compare its accuracy, given a normalized and an un-normalized version of the same dataset. Finally, we will compare how each classifier is evaluated and what may account for differences in output and accuracy of the models.

**Libraries and Distributions of Model Employed**

The Scikit-Learn, NumPy and Pandas where the libraries employed to build both the Naïve-Bayes and the Logistic Regression Models. The NumPy and Pandas was largely used to manipulate the dataset and perform calculation. Specifically, Pandas was used to read in dataset, in other to generate a well formatted data frame as a python dictionary object. This could then allow me to retrieve aspects of the dataset such as only labels or the review sentences. The NumPy library was used to calculate the accuracy of my model by returning the percentage of predicted models that where the model predicted correctly.

The Scikit-Learn provided functions to build and perform both the Naïve Bayes and logistic Regression. The “*CountVectorizer*” package was used to convert document reviews into word features, a requirement sentiment analysis. Using the *CountVectorizer*, I converted my reviews into a sparse matrix form (that a matrix containing extracted word features and their occurrences in). In this sparse matrix form, I could easily call *MultiNBomialNB* or *LogisticRegression* packages to perform my sentiment analysis.

To avoid frequency discrepancies between shorter sentences and longer sentences, the *TfidfTransformer* package was used. That is, since the *CountVectorizer* only gives the occurrences of word features in a review, regardless of the length of the review, longer documents reviews will have higher average count values than shorter documents, despite the fact that they may belong to the same category. Therefore, the *TfidfTransformer* was used to divide the occurrences of each word feature in a document by the total number of words in the document. This will normalize word frequency distribution in a category.

For the Naïve Bayes model, the *MultinomialNB* distribution model was used. And the sklearn *LogisticRegression* package was used in building the logistic regression classifier. These distibutuions was used because of the volumes of documentation that are available to guide me in building these model with a very short time. I also use the sklearn because of its integrity as a well structure framework for machine learning algorithms.

**Evaluation of the model**

For both the Naïve Bayes and the Logistic Regression I used the confusion matrix for evaluation. A confusion matrix was generated for each model, normalized and un-normalized version of the Naïve Bayes classifier as well as the normalized and un-normalized Logistic Regression classifier. This confusion matrix is easily generated using the *classification*\_*report* function in Scikit-learn. Using the confusion matrix, a simple accuracy of each model is then computed. . from the function, the Precision, Recall and F-measure can then be generated

**Results and Inference**

The *normalized* Naive Bayes classifier had a *precision of 0.802, Recall of 0.8 and F1-measure 0.80.* *The unnormalized model version of Naïve Bayes had a precision is 0.82, Recall 0.82 and F1-measure of 0.84.* The normalized Logistic Regression had a *Precision 0.80, Recall 0.80 and F1-measure of 0.8*. For the unnormalized version, the Logistic Classifier had a *Precision 0.83, 0.83 Recall and 0.83 F1-measure.* The observation is that, Logistic Regression turns to perform barely the same as Naïve Bayes model for Normalized dataset. However, Naïve Bayes performs better than Logistic Regression when the dataset is un-normalized.