**Comparison of Movie Genre Classifiers Based on Plots**

**Theron Spencer, Spring 2019**

**Abstract:**

Natural language processing (NLP) is an ever expanding area of research. Many tasks are involved in NLP, including text classification. This report evaluates the performance of three popular text classification algorithms. The problem of interest is determining movie genres based off of their movie plot summaries. The data was scraped from Wikipedia and consists of 20,070 movies. One of the most popular algorithms, due to its consistency in tackling a wide range of classification problems, is Random Forests. This algorithm will be compared to two other classifiers seen more often in text classification, namely Naïve Bayes and logistic regression.

**Background and Objectives:**

Turning the underlying structure and meaning of text into something quantifiable continues to be a difficult task. However, doing so isn’t impossible and there are many applications including sentiment analysis, spam filtering, topic labeling and many more. The algorithms of interest implemented in this report are based on a bag-of-words approach. This approach considers the text being analyzed as exactly that, a bag of words, which disregards the structure of sentences and only focuses on word frequency. These word frequencies are then used as features that are fed into the classifiers. In our implementation, we chose to use a modified approach, and instead of using word frequency we used word binary instances, i.e., we only paid attention to whether specific words were present or absent in the plots. The goal is to see how each algorithm preforms without any proper parameter tuning.

**Methods/ Results**

To begin, the genre category had to be cleaned. Using someone else’s code found on Kaggle, the genre category was cleaned up using Python. [1] Then in R, we further cleaned the data by dropping the movies that had genre frequencies lower than 2,000. This left four genres to be classified: action (4064), comedy (4507), drama (6265), and thriller (2298) for a total of 17,134 movie plots to be analyzed. A binary document term matrix was then generated with sparse words removed, where the final matrix consisted of the top 208 most prevalent words. We then split the data into training and testing sets that consisted of 70% and 30% of the randomized original data, respectively. The following are results of the implemented algorithms.

**Method 1: Random Forests**

Random forests is an ensemble method that has no distributional assumptions and preforms extremely well in many situations. The algorithm generates many decision trees and the class is determined based off of the majority vote. The following are the overall results using the “randomForest” package without any parameter tuning.

Accuracy Kappa AccuracyLower AccuracyUpper AccuracyNull

0.55331 0.36147 0.53959 0.56696 0.37082

AccuracyPValue McnemarPValue

0.00000 0.00000

**Method 2: Multinomial Naïve Bayes**

Naïve Bayes is a probabilistic classifier that is based off of Bayes’ theorem, and assumes independence between features, which is rarely the case, thus the term naïve. The following are the overall results using the “fastNaiveBayes” package without any tuning.

Accuracy Kappa AccuracyLower AccuracyUpper AccuracyNull

0.54922 0.36867 0.53550 0.56289 0.37082

AccuracyPValue McnemarPValue

0.00000 0.00000

**Method 3: Multinomial Logistic Regression**

Logistic regression is a form of regression that outputs probabilities of a certain class that are restricted to the interval [0,1]. Probabilities greater than 0.5 result in a positive classification and vice versa. The following are the overall results using the “nnet” package without any proper model selection/ tuning.

Accuracy Kappa AccuracyLower AccuracyUpper AccuracyNull

0.5575 0.3734 0.5323 0.5825 0.3613

AccuracyPValue McnemarPValue

0.0000 0.0000

**Discussion:**

Surprisingly, all three models, without any proper tuning preformed about the same. If proper tuning was done, we can almost be certain we would see more significant differences in classification accuracies. The model assumptions weren’t exactly investigated in depth and, just in general, a lot of work still remains in assessing the performance of these algorithms in this particular context. What would be interesting to explore is how the models perform as a function of word sparsity. It may be the case that a particular model performs significantly better with more or less words as features. It may also be the case that a particular model is more or less sensitive to word sparsity. Additionally, a more thorough data cleaning could possibly yield better results.

**References**

[1] <https://www.kaggle.com/aminejallouli/genre-classification-based-on-wiki-movies-plots>