IST736 Text Mining - Homework 1

Martin Alonso

## Introduction

For this exercise, we will take a set of 92 reviews, run them through a Multinomial Naïve Bayes classifier using 10-fold cross validation. The objective is to determine whether the model can detect whether a review is positive or negative, and if the review is true or false.

The data will first be run through Weka to check the probabilities of each word appearing in each class. The Weka results will then be run through Python using feature weighting in order to detect the 20 most frequent words for each class. The results should tell us whether the classifier is successful at identifying the sentiment and veracity of each review.

## Methodology

The file deception\_data\_converted\_arff.arff was first loaded into Weka. For the first part, the sentiment attribute was dropped so it doesn’t interfere with the Multinomial Naïve Bayes classifier. After this, I used the FilterClassifier, to convert strings to vectors and run the classifier.

For a first run at the veracity model, the settings were kept at default, using no stemmers and not filtering stop words. The model was run using 10-fold cross validation, giving a correct classification rate of 47.83 percent.

The model was then tuned to account for stemming, using the SnowballStemmer, and stop words, using Rainbow stop words. All words were also converted to lower case, and we required a minimum word frequency of two (figure 1). The model was run, and correct classification improved to 60.87 percent.

The results were exported to a tabular-separated values file and set aside.

Using these same parameters, the data set was reloaded and, this time, the lie attribute was removed. The model was ran without further parameter tuning, resulting in a correct classification of 78.26 percent. If the parameters were set to default, the correct classification rate would be 83.70 percent. However, to keep the results comparable, we’ll opt for the former parametrization.

The files were loaded to Python, along with the Pandas, Numpy, and Math libraries. For the veracity set, the logarithm of the fake column was subtracted from that of the true column. For the sentiment set, while the negative column was subtracted from the positive column. We took the top 20 words for both sets to check whether the model was doing a good job classifying (table 1).

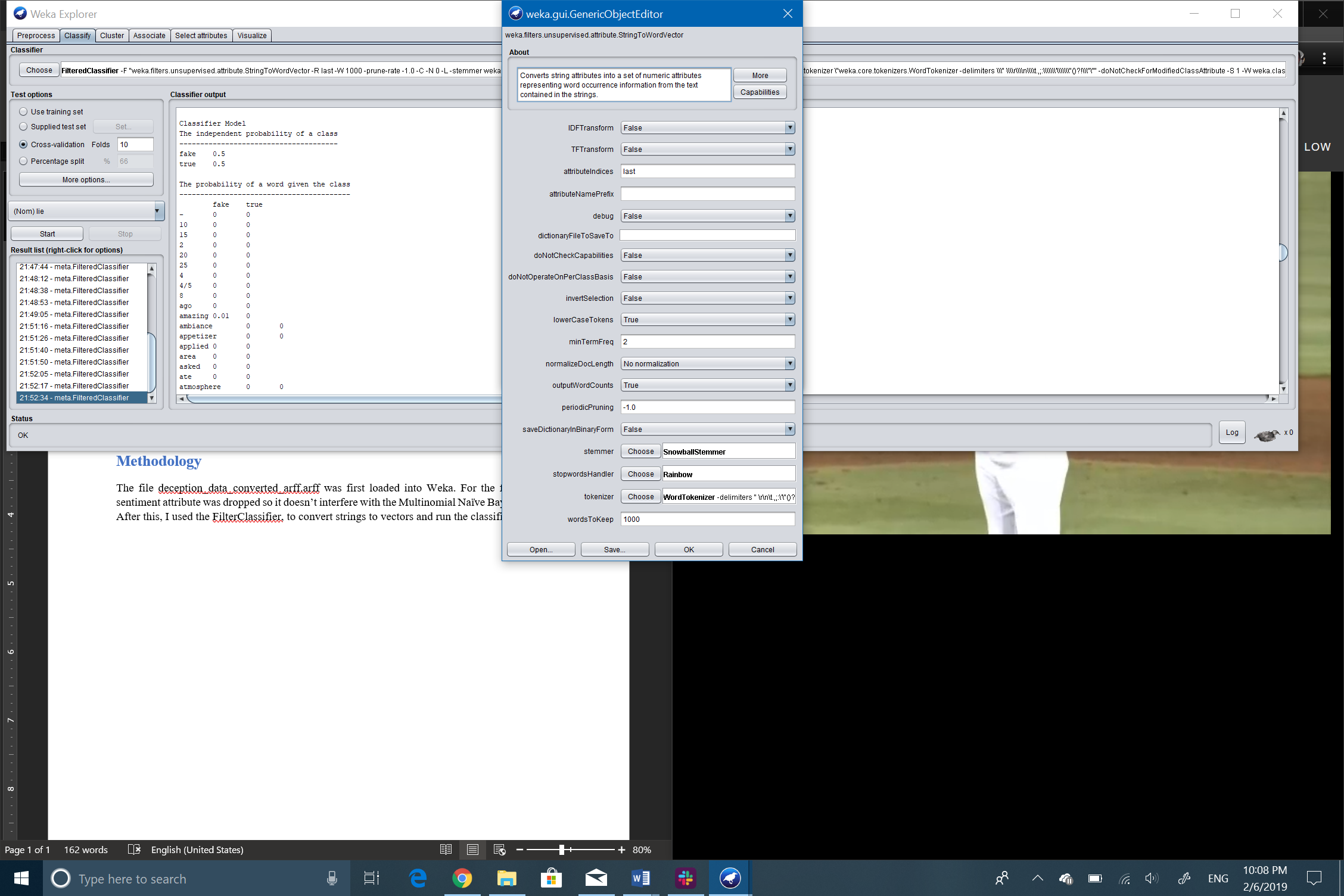


Figure 1

Table 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veracity Token | Veracity Log Ratios |  | Sentiment Token | Sentiment Log Ratio |
| Dirty | 1.490 | Sister | 1.610 |
| Eggs | 1.490 | Birthday | 1.610 |
| Environment | 1.357 | Feel | 1.610 |
| People | 1.308 | Town | 1.610 |
| Pretty | 1.203 | Café | 1.610 |
| Bar | 1.203 | Hot | 1.610 |
| Cheese | 1.203 | View | 1.610 |
| Isn | 1.203 | Soft | 1.610 |
| Inside | 1.203 | Delicious | 1.427 |
| Jersey | 1.203 | Nice | 1.427 |
| Idea | 1.203 | Sauce | 1.322 |
| Huge | 1.203 | Fresh | 1.322 |
| Joe | 1.203 | Fish | 1.322 |
| Kinda | 1.203 | Full | 1.322 |
| House | 1.203 | Hibachi | 1.322 |
| Large | 1.203 | Ice | 1.322 |
| Lemongrass | 1.203 | Huge | 1.322 |
| Brown | 1.203 | Hudson | 1.322 |
| Hotel | 1.203 | House | 1.322 |
| Entree | 1.203 | Hotel | 1.322 |

## Results

From the results in table 1, it is obvious that the model does a much better job classifying sentiment than veracity because the top 20 words for the sentiment classifier feature words with a positive connotation. The veracity classifier has a much tougher time differentiating between truths and lies because there is no clear pattern to what words are used.

Given these results, I believe computers have an easier time detecting sentiment than detecting lies. Lies can be either positive or negative, but there is no clear word usage whatsoever that can clearly indicate that a review is a lie. This is not the case for sentiment, as the classifier can easily detect which words are associated to positive reviews and which ones to negative reviews.

## Conclusions

The model, at least with this data set, isn’t able to clearly detect a false review from a true review. Perhaps this is due to the size of the data set, but it’s more likely that this is because words have no clear indication as to whether the author is lying or not. If the model is basing its ability to detect lies from truths based solely on word usage, then it will have a hard time classifying texts. If the model had more input data and were using a more specialized algorithm, such as NLP, then it might have an easier time classifying the texts.

But, with the current set, it is clear that a model, and a computer by association, can readily detect whether a review and a text have a positive connotation or a negative one. I would add, nevertheless, that if the model were to also incorporate NLP, its success rate could probably be higher than it currently sits at.