Text Mining – Cannon Fall 2017

HW4 - Tatum

Question 1

For this assignment I am using a data set that contains almost 5,000 tweets from political representatives. The tweets are labeled as having an audience of either constituency or national. My goal is to create a predictive model that can take a new tweet and determine if it is intended for a constituency or national audience without reading it. At first glance of the data, there are many common words that do not add much meaning, called stop words, which are very prevalent. I have removed 131 stop words from the data set like the, and, to and of. There are also many two word phrases that are very prevalent in the data that are being considered as individual words. I have joined many of these combinations together to be treated as one sentiment instead of two like high school, town hall, and last night.

To predict whether a tweet is intended for a constituency or national audience, I have used four classification models; Knn, decision tree, naïve bayes, and logistic regression. In all of these models, I have randomly divided the data into a training set with 3,464 tweets and a validation set with 1,485 tweets. The training set is used to develop the model and the validation set is used to measure how well the model performs. The better a model performs; the better it is at predicting which audience a new tweet is intended for.

There are several performance indicators that can be used to determine how well model performs. If we look strictly at accuracy (how often the model predictions the correct outcome, constituency or national), Knn and Logistic Regression performed the best with 79% accuracy rates compared to the decision tree with 74%, and Naïve Bayes with 76%. However, if we look at specificity (how often the model predictions constituency when in fact it is constituency) these models were the worst; Knn 6% and Logistic Regression 24%. Because there is a smaller number of constituency versus national tweets in the data set, I would place greater value on the model that is bester at predicting constituency tweets.

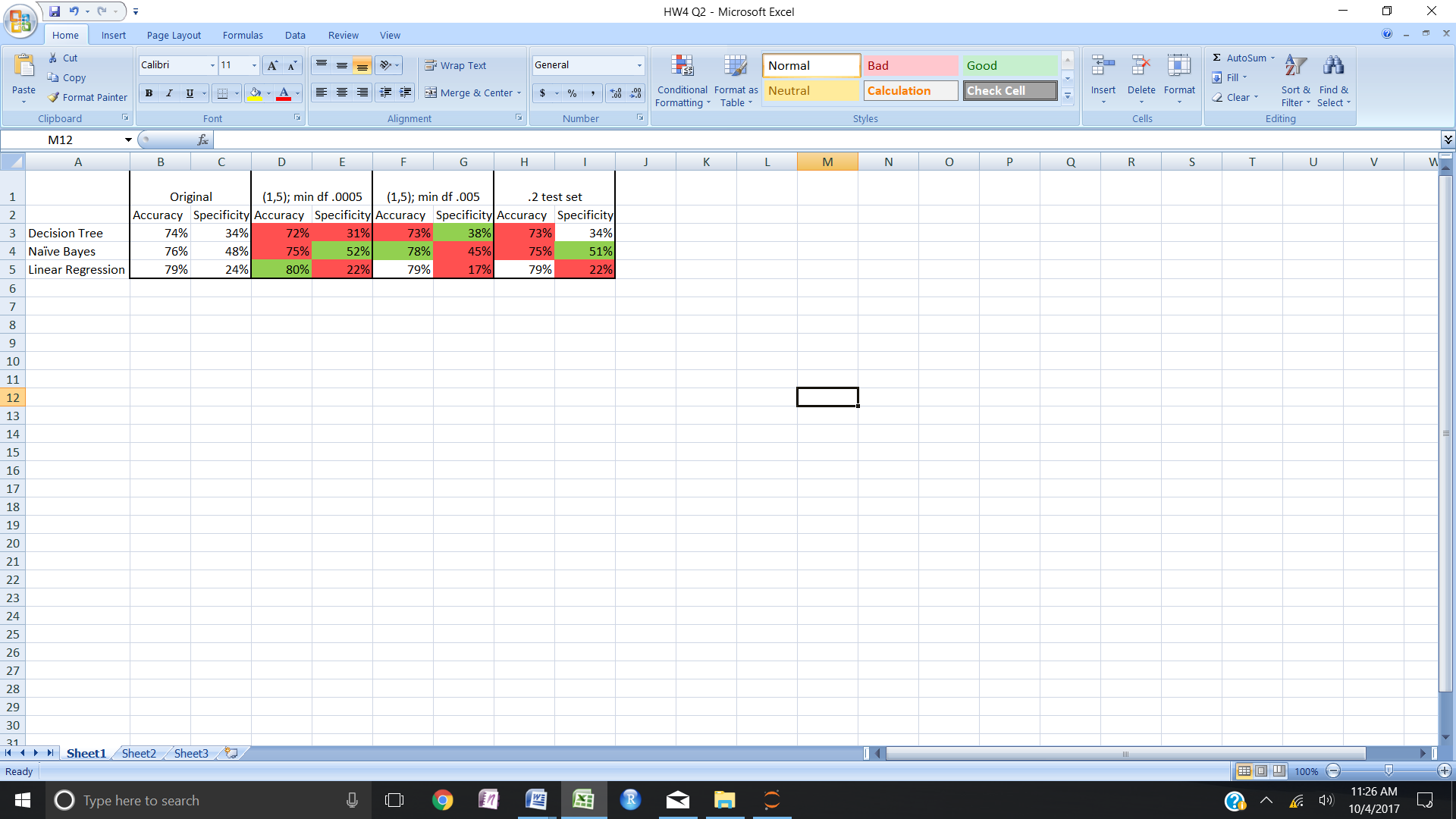
Therefore I believe the best performing model was Naïve Bayes with the second highest accuracy rate of 76% and the highest specificity rate of 48%. Although this is the best performing model in this test case, I do not believe these performance measures are strong enough to be truly reliable.

Question 2

To increase the performance of my models, I first looked at the feature space. In the original model, no ngram\_range was set. I tweaked the ngram\_range until there was no more added value and ended up with (1,5). Because this increased the size of the feature space to over 300,000 columns, I set the min\_df to .0005 which is much easier commutated. These two simple changes turned out to have the most significant improvements to the performance of the Naïve Bayes model. Although it lost 1% in accuracy, it gained 3% in specificity, which I determined to be the measure of most significance.

Because reducing the size of the feature space seemed to have an effect on the results, I reduced it even further by setting the min\_df to .005. This however did not have the same effect as before, results in table below. Then in an effort to increase the performance of the Decision Tree model, I adjusted many of the parameter settings; criterion = entropy, min\_samples\_split = 4 and 8, and min\_samples\_leaf = 2 and 4, all with negligible effect (not included below). And finally adjusted the size of the training and testing set. This did increase the specificity by 2% for the Naïve Bayes model, but that was not as much of an increase as the first adjustments made.

Overall I am still not very happy with these results. A specificity rate of 52% is hardly better than a flip of a coin. I do not believe these models perform well enough to predict whether a tweet is for a constituency or national audience.



Question 3

In conducting the K-means analysis on my data set, I first started without any pre-processing. This resulted in a very large data frame and was computationally intensive. Removing the stop words seemed to speed things up a little and yielded the same results. I ran a K-means analysis for 1 – 20 clusters and the number of clusters that incrementally explained the most variance was 4.

From here a changed the data frame to include tokens that where between 1 and 5 words long and included in a minimum of .0005 of the tweets. This reduced the data frame size significantly but did not explain nearly the same amount of the variance. I ended up selecting four clusters that explain about 40% of the variance from the original data set minus the stop words.

After reading several of the tweets per cluster, there is not a general theme that can be found to describe the various clusters. Having only 40% of the variance explained, does not reveal a very good fit for this cluster analysis.