Ryan Martin

Dr. Everaldo Aguiar

CS 7180

07/18/19

"Sentiment Analysis Study: Comparing Classification Models with Random Forest"

1) What problem are you going to be tackling on your project?

The problem I will be solving in my project is how to create the most accurate and feasible machine learning model for sentiment analysis. Sentiment analysis is a relatively new topic that has formed with the evolution of social networking, and strives to bridge the understanding between public mood and current events. Many scientists have recognized the possibility of gaining a profit from sentiment analysis, and there have been many studies that have shown a significant correlation between public mood and the DOW & Jones Industrial Index (Pagolu, Reddy, Panda, & Majhi 2016). In my project, I will train various models and use the best model to be implemented in my app that will simply allow a user to access data from Twitter's API and gain insight on the general public's mood on a particular topic and/or people.

2) Why is that an interesting/useful application of data science?

There has been almost exhaustive use of Twitter data to predict the stock market in recent years, but scientists have found many other useful applications for sentiment analysis, such as Google's flu tracker that can predict disease outbreaks around the world, which is arguably more important than financial applications (Pagolu, Reddy, Panda, & Majhi 2016). Because twitter users collectively produce over 140,000,000 messages on its micro-blogging service (known as

"tweets"), there is a seemingly endless supply of data to study the public mood to make predictions. I believe that one could possibly predict the outcomes of elections with sentiment analysis, which would make this app a poll device for the 21st century.

3) What models are you envisioning training to address that (e.g., classification, regression, clustering)?

In order to explain the models that I will be implementing in this project, I will first explain the steps that will allow the models to work effectively, and the workflow that I envision for this assignment, which consists of data collection, data understanding, data preprocessing, feature extraction, and model training and evaluation. The data that I will be using contains an excellent case study of sentiment analysis, consisting of over 14,000 instances of user opinions of airlines. I will prepare this data through tokenization and stopword removal, and then use regex matching for special character removal, which will essentially clean the text for feature extraction. Next, I will vectorize and feed these resulting features into a classifier such as a random forest algorithm. For completeness, I will evaluate my random forest algorithm and measure its accuracy against other models (AdaBoost, GradientBoost, Logistic Regression, *et cetera*) in order to ensure that my model is the most accurate. If my Random Forest Classifier fails against the other models, I will investigate why that happened and include it in my abstract. I will also explain why some models worked better than others. Studies have shown that people have between 70 -79% sentiment agreement on a text, so the accuracy of a model usually will not exceed that threshold (Pagolu, Reddy, Panda, & Majhi 2016). I will personally test this statement to see if it is true in my model.

Although much slower than n-gram analysis, it would be of some interest to train this data with a neural language model that uses a Naive Bayes classifier to see if it can beat the threshold of sentiment agreement in a text. This type of model is accomplished in a feed forward neural net that " is an instance of supervised machine learning in which we know the correct output *y* for each observation *x*. What the system produces, via Eq. 7.12, is *y*ˆ, the system's estimate of the true *y*" (Jurafsky & Martin 2009). My hypothesis is that the neural net could beat the threshold of 79% but it might not be feasible to implement in an app because it is slower than random forest. However, this is an experiment I am willing to pursue.

4) What will a user-facing service that packages your model(s) look like and how will you make it user-friendly for someone to leverage your work?

Since my project will essentially allow users to search any topic available on twitter, I want to make my product easily accessible so that people will be able to access information about the world around them and make predictions. I will accomplish this task by converting my ML model with Apple's CoreMLTools, so that it can be used in xCode to create an iOS app on sentiment analysis. The app will allow the user to search for a topic, and the model will display the public mood on that topic with a score such as overall positive, negative or neutral. I believe that this app could be very useful to users in some instances. For instance, someone could search for the public mood concerning the flu simply by searching for the hashtag of the flu ("#flu") in the app. Another person could check on their favorite or least favorite politicians popularity ratings. And lastly, someone could compare a candlestick analysis of a stock with the sentiment analysis of that company. In some ways, sentiment analysis will be more effective than the news

because an algorithm can display facts, like a recent flu or measel's outbreak, faster than a human can write, edit, and report on the news about the same topic.

Works Cited:

Jurafsky, Dan, and James H. Martin. *Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Prentice Hall, 2009.

Pagolu, Venkata Sasank, et al. "Sentiment Analysis of Twitter Data for Predicting Stock Market Movements." *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 2016, doi:10.1109/scopes.2016.7955659.