Machine Learning Kaggle Submission

Rohan Patel (rp442) and Alexander Li (afl59)

December 5, 2018

1 Submission:

Team Name: WineBurger
Public Score: 0.84166

2 Introduction:

In recent years, the tweets of the President of the United States (@realDonaldTrump) have become increasingly prominent in American society and politics. They have become a common method for disseminating important news, developments, reactions, and opinions about national and global affairs. However, when the public sees a tweet, it is not known whether the President himself has written the tweet or whether a member of his staff has done so on his behalf. The true author behind a tweet is significant because some tweets from the account can be viewed as divisive or "un-presidential." A development in this question was made several months ago when media reports noted that members of the President's staff utilize iPhones but the President himself uses an Android. This was consequential as Twitter's API provides information about the type of phone that posted the tweet. We assume that tweets coming from an iPhone are from the President's staff while tweets coming from an Android came from the President himself. In this project we developed a Machine Learning algorithm that would utilize Twitter data in order to classify which tweets came from the President or his staff. This was framed as a binary classification problem with -1 corresponding to an iPhone tweet and +1 corresponding to an Android tweet.

3 Feature Extraction:

The Twitter data consists of the tweet text, whether it was favorited, favorite count, time-date created, whether it was truncated, retweet count, whether it was retweeted, and whether it was a retweet. In our final implementation, we used the tweet text, time-date created, retweet count, and favorite count as features. Our feature extraction worked as follows:

1. **Tweet Text:** To process the text in a tweet, we use a bag-of-words approach. For every word in a tweet, we hash the word to a 1000 element vector using Python's built-in hash function. To clarify, we hash words in the tweet text to numerical values that we then mapped to indices in the feature array. The feature has value 1 if the word is in the tweet, and 0 otherwise.

2. **Time-Date Created:** We found it more likely that the President's staff would tweet during business hours. So we added a feature in our vector to determine if a tweet was created before 7am. We also added a feature for if a tweet was sent after 6pm. These features were represented as 1 or 0.

3. **Retweet Count:** We found it more likely that tweets that came from the President himself would relate content that would be retweeted. And so we added a feature on whether the tweet was retweeted more than 10000 times. This feature could take on value 1 or 0.

4. **Favorited Count:** We also hypothesize that tweets that come from the President himself would relate content that would be favorited. And so we added a feature on whether the tweet was favorited more than 30000 times.

5. **Commonly Used Words in Tweet Text:** After manually examining the training data and doing research online, we found the following words/symbols to to be especially representative of a true tweets author: ”I”, ”#”, ”https” (link exists or not), @, ”media”, Hillary, Trump, ”...”, ”Thank.” Technical things like @ or ”#” likely come from a staffer, while mentions of ”media” or ”I” come from Trump. We represent the existence of each of these words in a tweet with a feature that takes on value 1 or 0.

4 Model Creation and Validation Error:

Using the Sci-Kit Learn Python Package, we implemented a Random Forest algorithm and an RBF-Kernalized SVM. Splitting our training data into both training (80 percent) and validation sets (20 percent), we found Random Forest to work better against the validation set with an accuracy of 81%. This is likely because, as learned in class, Random Forests have low-bias, low-variance, and do not need to be parameterized. Our best out-of-bag accuracy is 88%. We hope to further improve this project after the deadline and perhaps implement a web-app to classify tweets live.