S-109A Introduction to Data Science: Final Project

Bhandarkar, Mishra

**Scope of Work for Data Science Final Project**

Prepared by Group #5  
Karan Bhandarkar, [karanbhandarkar@gmail.com](mailto:karanbhandarkar@gmail.com)   
Vivek Mishra, [iblpvivek@icloud.com](mailto:iblpvivek@icloud.com)

**Project Statement and Background**

Manipulative robots are spreading fake news online that could be exacerbating volatility in financial markets.

A major study of tens of millions of tweets over two years found that the sheer volume sent by “bots” helped to drive shares in FTSE 100 companies up or down for short periods, effectively moving markets.

Tweets sent by humans typically had a positive impact on stock prices while those sent by robots were more often negative, according to research presented to the Royal Economic Society by economists Oleksandr Talavera, Rui Fan and Vu Tran.

Robot messages typically had an effect for up to half an hour, pulling down prices and increasing volatility.

In 2013, the Twitter feed of the Associated Press [told](http://allthingsd.com/20130423/u-s-stocks-tank-briefly-in-wake-of-associated-press-twitter-account-hack/screenshot_4_23_13_10_18_am-2/) us that [Barack Obama](http://topics.time.com/barack-obama/) had been injured in an explosion at the White House. The tweet was fake — the product of a hack — but given the events in [Boston](http://topics.time.com/boston/) earlier in the week, the news spread like [wildfire](http://topics.time.com/wildfire/), garnering more that 4,000 retweets.

[The AP quickly addressed the situation](http://www.ft.com/intl/cms/s/0/33685e56-ac3d-11e2-a063-00144feabdc0.html#axzz2RLrglMyc), suspending its Twitter account, and alerting readers through associated accounts that the tweet describing an explosion at the White House was the result of a hack.  No harm, no foul, right?

Well, not exactly. [According to the Financial Times](http://www.ft.com/intl/cms/s/0/33685e56-ac3d-11e2-a063-00144feabdc0.html#axzz2RLrglMyc), that one tweet sent shock waves through the stock market — causing the S&P 500 to decline 0.9% — enough to wipe out $130 billion in stock value in a matter of seconds. The market quickly recovered that value, but the breakneck pace at which the stock market tumbled reminded many people of the infamous[2010 “flash crash](http://www.time.com/time/business/article/0,8599,1988201,00.html),” or [2012's crisis at Knight Capital Management](http://business.time.com/2012/08/08/high-frequency-trading-wall-streets-doomsday-machine/), in which a computer glitch cost the firm $440 million and nearly sent it into bankruptcy.

Both of these events were caused by the proliferation of high-frequency trading, or the practice of Wall Street firms using high-powered computers to execute thousands or millions of trades per second, making minuscule profits — that add up in a big way — on each trade.

Though nobody knows for sure what exactly precipitated the volatility, many market watchers blamed high-frequency traders, and more specifically the variety that use algorithms to comb through the internet at lightning-quick speeds, actually “reading” news items and tweets, and making trades based off of that information.

**Goal:** In this project, the goal is to detect Twitter bots using tweets data from the Twitter developer API by utilizing machine learning techniques. This will be done to determine whether a financial tweet is of significance or not.

**Data Resources**

We will collect our own data for this project. We were provided a basic Python script, tweepy\_script.ipynb, that utilizes the tweepy library [4] to access the Twitter API. We have significantly modified this script to fetch user details and user tweets. This process is detailed in the TwitterDataPreProcessing.ipynb submitted.

We have taken 50 accounts that tweet financial information and 50 bot accounts from different categories(to understand bot behavior). Getting hold of more accounts is difficult, especially bot accounts since Twitter has been clamping down, so we have bootstrapped to increase the user details dataset(shown in TwitterDataEDA.ipynb).

Tweet details contain many details, most of which can be eyeballed and considered trivial. We have cleaned up these column to focus our EDA on specific columns (shown in TwitterDataEDA.ipynb).

**High-level project goals**

1. The first step is to create our own dataset. We will mine the data for the project using the Twitter API and utilize feature engineering and pre-processing techniques to prepare the data for analysis.
2. Create several models to determine characteristics of different types of twitter users. Create at least one model that uses natural language processing techniques, such as topic modeling [6], and at least one model that uses a classification algorithm. We may decide to have models that use both. We will provide evidence of success at detecting bots when compared to human users or explain why it wasn’t possible.
3. Perform a comparison of your models. This will include an error analysis and an evaluation of the predictive quality of your models.

**EDA**

There are two json files that we have after the preprocessing, one is tweets and another is the accounts itself of which the tweets belong to.

We are looking at some patterns that we think could be essential for detecting Bot vs Non-Bot accounts

1. One of easiest thing to find was to look at the name column itself and if that has a bot in it and I think that’s part of data collection strategy
2. Looking at the followers count vs friends count from the accounts data and the ideally bots should have more followers less friends. And this shows in our data as well
3. We also look at the default images of every account and do a graphical comparison to see if there is any pattern there. What we see in the data and what we expected was more of anti-pattern and I think more accounts should give us much better clarity
4. We plotted a scattered matrix for both the bot accounts and non bot accounts to find out any correlation and we found that favorites count and status count had strong positive correlation for both the accounts.
5. We generated a word cloud of all the tweets from bot accounts vs non bot accounts and we could see a clear pattern of lot of verbiage from bots vs some meaningful context from non bot accounts

From the EDA so far we have identified that we are going to be looking at favorites count , followers counts ,friends count, status\_counts, name, verified and tweets itself and for that we will use NLP.

Please refer to TwitterDataEDA.ipynb for detailed code and report on data analysis.

**Base Line Model**

1. We will start with Multi linear regression with the columns mentioned above and monitor the error and scores
2. We will then switch to Polynomial regression and records the performance matrx with for the test data.
3. We are also targeting a deep learning technique here which will studied in the class prefrerably a simple neural network with either CNN and RNN.

We will compare the results from all the 3 models and see which one performed better and what could further be tweaked.

**References**

1. Stefan Wojcik, “Bots in the Twittersphere”: http://www.pewinternet.org/2018/04/09/bots-in- the-twittersphere/
2. Chris Baraniuk, “How Twitter Bots Help Fuel Political Feuds”: https://www.scientificamerican.com/article/how-twitter-bots-help-fuel-political-feuds/
3. Chengcheng Shao et al., “The spread of low credibility content by social bots”: <https://arxiv.org/pdf/1707.07592.pdf>
4. The tweepy Python library: <http://www.tweepy.org>
5. Twitter’s developer resources to learn about the API: https://developer.twitter.com
6. Asbjan Ottesen Steinskog et al., ”Twitter Topic Modeling by Tweet Aggregation": http://www.aclweb.org/anthology/W17-0210
7. Time Magazine: <http://business.time.com/2013/04/24/how-does-one-fake-tweet-cause-a-stock-market-crash/>
8. The Telegraph: <https://www.telegraph.co.uk/business/2018/03/31/twitter-bots-manipulating-stock-markets-fake-news-spreads-finance/>
9. Twelve ways to spot a bot: <https://medium.com/dfrlab/botspot-twelve-ways-to-spot-a-bot-aedc7d9c110c>