Summerschool: Social Data Science

Participants: 96, 220, 41 and 127

The Trump Twitter Effect

Analyzing the effects of President Trump's tweets on financial markets

Handed in 30th of August 2019. Supervisors: David Dreyer Lassen, Andreas Bjerre-Nielsen and Snorre Ralund

Copenhagen University, Department of Economics

Contents

[1]

[1]Responsible party in (·).

1 Introduction

1.1 Research Area (127)

This paper studies whether Donald Trump's sentiment can be used to predict American stock market movements. We conduct a *sentiment analysis* of his tweets, and use *supervised machine learning* to try and predict the Dow Jones and S&P 500 stock movements based on his sentiment. We present an analysis based on respectively all of his tweets and the tweets relevant to our subject to attempt to increase the accuracy of our models.

Introduction Donald J. Trump – television personality, businessman and alleged multi-billionaire – became the 45th president of the United States of America in January 2017. Trump had for decades before taking the oval office been known for his strong populist opinions. Before and while campaigning for the presidency, he had several outlets for his many points of view – rallies, tv-debates, and even his own reality program. After moving into the White House, he found himself having only one direct and rather uncontrolled line to the public: Twitter. But the social media seems to suit the President well. Trump sent his first tweet on May 4th, 2009, and has since then fired off more than 47.7 thousand tweets. While being the most powerful man in the world, he uses this line of communication to praise his friends and shame his enemies including the companies that do not act as he thinks they should, as an example below.

Figure 1.1: Example of tweet about Apple

Note: Link to tweet can be found here.

Twitter was founded in 2006 and is becoming an increasingly important source to describe and explain financial dynamics[2] The social media works as a real-time information channel that both includes major news events but also small niche occurrences, before the mainstream medias take it.

[2] See e.g. "Twitter Sentiment Analysis Applied to Finance: A Case Study in the Retail Industry" (2015).

We conduct a sentiment analysis based on text mining of all Trump's tweets in the period 2015 to present day well as only his market relevant tweets and compare it to the Dow Jones Industrial Average Index (DJIA), which is the sum of the stock prices of the 30 largest American companies, and the Standard & Poor's 500 (S&P 500), which is a stock index for 500 big American companies, respectively.

The first part of this paper contains a literature review, which lay out the framework of our subject. Next, we go through our methods and theory used in this paper. This is followed by descriptive analysis of our data, an analysis and discussion of our results including what we could have done differently. Finally, we conclude on our results and suggest how one could further expand the subject.

1.2 Literature Review (127)

Several studies have investigated the effect of Trump's tweets on the financial markets. Otani et al. (2017) have made a *Trump Target Index*, which examines the long term share prices of 12 companies mentioned negatively by Trump on Twitter or in other public remarks. They find that stock prices of the 12 companies have generally climbed since Trump won the election in 2016, slightly outperforming the broader American stock market.

The same idea was used by Jeffrey Born et al. (2017) in their study which examined whether Trump's tweets had an effect on the stock price of the mentioned firm, thereby testing the semi-strong Efficient Market Hypothesis. Their findings revealed that positive tweets would elicit positive abnormal returns and vice versa, although the abnormal returns would be statistically insignificant within five days. Their study relied solely on event study methods, which allows us to possibly expand on their findings using machine learning. Mittal et al. (n.d.) apply sentiment analysis and machine learning to find correlation between piblic and market sentiment using Twitter data with 75 pct. accuracy.

We have learned from behavioral economics that emotions can effect individual decision-making and behaviour. Johan Bollen (2011) study whether this also applies to society at large. They derive collective mood states from large-scale Twitter feeds to see if they are correlated with or predictive of the DJIA as an economic indicator. They analyse the Twitter mood using two mood tracking tools - OpinionFinder and Google-Profile of Mood States and investigate whether the mood can predict the DJIA closing values. Their results indicate that the accuracy of DJIA predictions can be significantly improved when including specific public mood dimensions. They find an accuracy of 86.7 pct. in predicting the daily changes in the closing values of the DJIA and a reduction of the *Mean Average Percentage Error* by more than 6 pct.

A study by Colonescu (2018) examines the effect of the daily flow of Donald Trump's tweets on the US financial and foreign exchange markets. It finds some evidence of short term and lasting effects on US-Canada and US composite exchange rates. The study also investigates other exchange rates, but finds no significant effects. The only lasting effects found are in the case of the US dollar composite exchange rate.

In the following section we present the theory used in the project.

2 Theory

2.1 Sentiment Analysis (96)

Sentiment analysis is a type of *natural language processing* that tries to quantify a sentiment, or opinion, of a given text. With the rise of social media the act of conducting sentiment analysis, or opinion mining, is becoming more frequent in both research and business. Opinion mining can provide valuable insight into user behaviour, but this is not without controversies. One of the more notable examples is the Facebook-Cambridge Analytica scandal, where it was revealed that the british consulting firm, Cambridge Analytica, had scraped millions of facebook users' data and used it to do a psychographic profile to aid certain candidates in the 2016 US presidential election.[3] Our consideration of ethics are presented in Section 3.3. Nevertheless, automating the process of being able to extract sentiment and opinions from text can be exceedingly useful in the fields of sociology, psychology or marketing. Even in the fields of finance it may be used to predict behaviour and trends in the stock market.[4] As previously mentioned, this project seeks to conduct a sentiment analysis on President Trump's tweets and see whether or not there is correlation with particular elements in the stock market. Before we conduct our sentiment analysis on our data, we must first decide on which method we use to calculate the sentiment. Two frequently used methods are AFINN and VADER.

AFINN versus VADER AFINN is a word-list and *polarity-based* approach to sentiment analysis.[5] This means it categorizes each word into either a positive or negative class and gives the overall text a score ranging from -5 to 5 with 5 being the most positive. AFINN is a lexical based approach meaning it simply matches the positive words and does not take into account negation. E.g. the sentence 'I do not love ice cream' and 'I love ice cream' will both have a score of 3, while the first sentence obviously bears a negative opinion.

VADER categorizes each word from a score from -1 to 1 with 1 being the most positive. It is a *valence-based* approach, meaning it takes into consideration the intensity of the sentiment. For example the word 'excellent' will provide a higher positive score than the word 'good'. VADER is a combination of a lexical and rule-based approach. It adopts hard coded rules to take into account simple negations, in contrast to the AFINN package. Using VADER the sentence 'I do not love ice cream' is scored at -0.52 while 'I do love ice cream' gets a

[3] See e.g. Hern (2018).
[4] See e.g. Otani et al. (2017).
[5] See e.g. Nielsen (2011).

score of 0.64. Some of the advantages of VADER is that it doesn't require training data and is already curated by humans and it does not suffer from a considerable speed-performance trade-off. In the context of tweets it outperforms even individual human raters[6] while in the context of Social Media Networks VADER outperforms AFINN in a 3-class experiment (positive, neutral and negative).[7] Based on these characteristics we continue our analysis with VADER as our sentiment analyzer of choice.

2.2 Efficient Market Hypothesis (220)

The basic premise of the *efficient market hypothesis* (EMH) is that in a liquid market the price of a security reflects all the available information. EMH follows the logic of the Random Walk theory, i.e. price changes follow a random walk and as such prices are independent of each other.[8] There are three forms of the EMH; weak, semi-strong and strong form. The semi-strong form is considered the most applicable to real life scenarios. In its semi-strong the EMH is that price movements are a reflection of information that is publicly available. This implies that fundamental and technical analysis is rendered useless as it is only non-public available information that can give investors an edge over the general market and thereby profit. Even if the market somehow reacts disproportionately to new information it is considered an anomaly and the effect will quickly dissipate.[9] As we try to predict the price movements of stock from Trump's tweets, imply a correlation between the market sentiment and Trump's sentiment. The assumption being that Trump's tweets can be interpreted as an exogenous shock to the stock prices and will be reflected in the price movements by affecting market sentiment.[10]

2.3 Machine Learning (220)

2.3.1 Train-test-split

Machine learning (ML) is "the application and science of algorithms that make sense of data" (Raschka et al. (2017), p. 1). One method of machine learning is *supervised learning* where a model is created that takes in labeled *training data* and uses it to predict something about future data and measures the accuracy of the predictions on a designated set of *test data*. The general split of training and test data follows a random state distribution that assumes

[6] See e.g. Hutto et al. (2015).
[7] See e.g. Ribeiro et al. (2016).
[8] See e.g. Malkiel (2003).
[9] See e.g. Fama (1970).
[10] See e.g. Mittal et al. (n.d.).

independent observations. This is referred to as `train_test_split` in Python. When creating a machine learn model it is extremely important the the training data and test data are not mixed as it would lead to dishonest results. As this project is using time series data we cannot split the data in the conventional way as the observations are not independently distributed. However, since we use a first difference transformation that removes the trend this is less of a concern. To be safe we will use `TimeSeriesSplit` so we can divide our time series data into chronological segments. It should be noted that unlike standard cross-validation methods, consecutive training sets are supersets of the previous sets.[11] Supersets allow us to enhance the performance of our model without needing exceedingly amount of data. In this project we focus on both a regression model and a classification model.

2.3.2 Ordinary Least Squares

The Ordinary Least Squares (OLS) approach is a type of linear regression, where the the goal is to estimate the unknown parameter by minimizing the the residual sum of squares between the observed observations and the one predicted from the model. This can be written in matrix form as

$$Y = X\beta + \epsilon \tag{2.1}$$

$$\widehat{Y} = X\widehat{\beta} \tag{2.2}$$

$$\widehat{\beta} = \left(X^T X\right)^{-1} X^T Y \tag{2.3}$$

$$RSS = \frac{1}{\mathcal{N}} \|Y - \widehat{Y}\|^2, \tag{2.4}$$

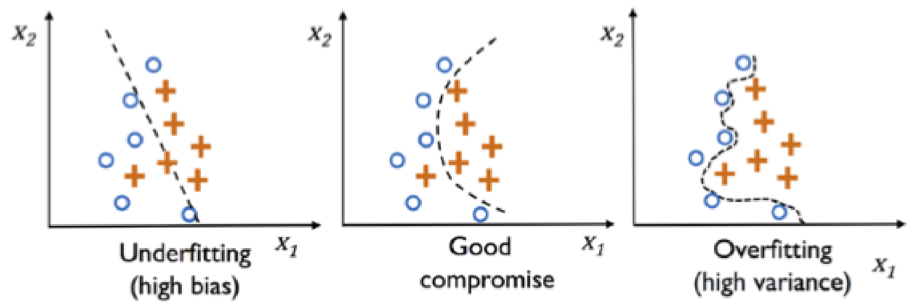
where Y is the target value, X is a $n \times k$ matrix with k explanatory variables, or features in the context of ML. β is a vector of k coefficients for the features, $\mathcal{N}$ the number of observations, and RSS the residual sum of squares.[12] While one can perform machine learning by using non-linear regression the OLS approach is one of the simpler methods. However, it comes with its downsides. OLS generally does not work well when the dataset contains outliers. Of course, if the relationship between variables is not linear in nature the OLS will not be able to capture it but it will still be the best linear approximation to the underlying conditional expectation function [13]. Another pitfall is that by increasing the numbers of features R^2 will increase in the training set and will hurt the prediction results.

[11] See the documentation (2019) for sklearn.
[12] See e.g. Verbeek (2017).
[13] See e.g. Angrist et al. (2009)

2.3.3 Regularization

Within machine learning there are two common problems when fitting the model: overfitting and underfitting. Overfitting occurs when the model does not generalize well from the training data to the test data which implies a high variance within the model. This is illustrated in Figure 2.1. Conversely, underfitting is when the model suffers from a high bias. This implies that the model is **not** complex enough to capture the underlying data.

Figure 2.1: Fitting the model



Note: Raschka et al. (2017), Chapter 3.

To combat the issues with over- and underfitting we use regularization as a way to tune the complexity of the model by inserting a penalty function. There are two common types of regularization.[14]

Ridge The advantage of the Ridge model is dealing with high degrees of multicollinearity. In the Ridge model regression we add the squared sum of weights to the cost function, this can be expressed as,

$$\mathcal{J}(w)_{Ridge} = \lambda \sum_{j=1}^{m} w_j^2 + \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)}\right)^2, \tag{2.5}$$

where λ is the hyperparameter, which we optimize over. As λ increases the weights of the model shrinks and regularization strengthens.

LASSO An alternative to Ridge, LASSO focusing on reducing the underfitting of the model by removing features that provide a high degree of bias. This makes it a valuable tool in feature selection. It can be expressed as,

$$\mathcal{J}(w)_{LASSO} = \lambda \sum_{j=1}^{m} |w_j| + \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)}\right)^2. \tag{2.6}$$

[14] See e.g. Raschka et al. (2017).

2.3.4 Classification

Besides the regression model we can use machine learning to solve classification problems. Using a classification model we seek to predict the qualitative movement of the stock price by introducing a dummy variable determining whether the stock price goes up or down. This can be expressed as the indicator function $\mathbf{1}_A(x) = \begin{cases} 1 \text{ if } x \in A, \\ 0 \text{ if } x \notin A. \end{cases}$, where x is the subset of observations with positive daily percentage changes. We use a k-fold cross validation while preserving the chronological order of the splits to strengthen the robustness of the classifier tree. This is done by performing random splits between the training data set into k number of folds, where $k-1$ folds are used for the model training, and the remainder for performance evaluation.

3 Data Collection and Ethics

3.1 Data Collection (41)

To compile the tweets of Donald Trump, we used the Trump Twitter Archive, a database consisting of all of his tweets. The database do have some limitations due to the fact that it is missing around 4000 tweets before 2016. However, it is updating in real time since January 2017. Since our project revolves around the period of his presidency, the archive does prove to be a reliable data source and a more efficient way of collecting the tweets than if we were to use the Twitter API.

In an attempt to focus on more market specific and thus relevant tweets we obtained Bloomberg's collection of Trump tweets, which is a collection of economically related tweets he has made since his inauguration in January 2017.[15] We tried to scrape it directly from the website but were blocked. We therefore manually downloaded the data and saved it as a `.json` format. Later we removed outliers from the data set before feeding it into our model.

The Down Jones Industrial Index and S&P 500 data is collected through the Alpha Vantage API. We wanted intra-day data, but that was not accessible. Instead, we use daily changes between the open and adjusted close price. For both the Bloomberg tweets and our collected tweets, we filter out tweets with a sentiment score of 0.00 to improve our model as they are non-informative and often consists only of links. For the Trump Twitter Archived tweets we also removed retweets as they don't directly tell us anything about Trump's own sentiment.

3.2 Fuzzy String Matching (96)

Since the literature finds either small or insignificant effects of Trump's twitter sentiment on the Dow Jones Industrial Average,[16] we attempt to go a step further into detail to trace possible effects.

Intuitively, if we consider his general sentiment in a day and the effect it has on the stock markets average, we could have many situations where the effects of a tweet in one company are cancelled out by the opposite effect in its competitor and therefore measuring against an index might prove futile.

[15] See Bloomberg (2019).
[16] See e.g. Colonescu (2018).

To trace the direct effect of a tweet and its sentiment we try and find the tweets that are relevant to specific companies. This we pair with data on their respective stock prices before and after the tweet.

To find the relevant tweets we use a fuzzy string matching design by implementing the package `Fuzzywuzzy`. Because a company name may be misspelled, written in genitive, appearing with @ or something similar, we can not simply loop through the tweets searching for the substring 'Company Name'. Instead, `Fuzzywuzzy` uses the *Levenshtein Distance* to determine the difference between sequences of strings. The Levenshtein is a metric to measure how far apart words are by calculating the minimum edits required to change one into the other.[17] The `Fuzzywuzzy` also offers alternate methods for comparison. Since we do not know the structure of a tweet and the order of the words we use `token_set_ratio`. This ignores word order and duplicates when comparing sequences.

When determining the score required from the `token_set_ratio`, we have tried different benchmark scenarios of what parts of company names might be included in a 140 character tweet and arrived at 60 pct. This varies a bit by the length of the company name, so a firm with a longer name will have an easier time obtaining a match if their name is in the tweet. At 60 pct. we will capture tweets even though a company will be mentioned as "Amazon" rather than "Amazon.com inc". But the risk of setting the requirement so low is the amount of false positives. However, we are probably not able to set the perfect cutoff in a complex case like this with 505 official company names. Add to the problem, that there are many companies that are not known as their official names i.e. Google vs. Alphabet, companies that will be mentioned by abbreviations, i.e. AIG vs. American International Group. This would probably be an entire study in itself, so we use our benchmark test cases result of 60 pct.

We find the official company names from S&P500 by scraping their names and tickers from Wikipedia[18]. We add to this list company names where we do not expect their official names to be used, i.e. Google. We have not exhausted this possible problem completely as it was too time consuming. This could be one way of improving the design. We end with a list of 509 names. Searching through 19,663 tweets for a fuzzy score for 509 different company names is computationally intensive so we have kept the design basic.

[17] See Aries (2019).

[18] See *Natural Language Processing for Fuzzy String Matching with Python* (2019).

Since we use a simple design, we are alert of type 1 errors, i.e. finding tweets relevant for a company without this being true. On the other hand, it is much more difficult for us to inspect type 2 errors since these will not be observed.

After the first fuzzy string matching search we drop the companies that have zero matches. This might be questionable since our manipulations in the section that follows could have lead to new matches when their names were changed. Next, we identify multiple problems that might lead to type 1 errors. We inspect our results using `relevant_tweets` in order to see how many times the companies are mentioned. This leads to the conclusion that firm names with inherent meaning are problematic. This could be names which include energy, resources, national, technology, international, but also financial, companies, corporation. These names give matches when Trump for example tweets about international relations or the financial situation. We therefore remove these parts of the company names hoping to search for ”Ameriprise” rather than ”Ameriprise Financial” for example. This may pose some new problems with company names becoming very short increasing the risk of their entire name being in the tweet at random, for example will ’International Paper Company’ become ’Paper’ which will have much higher chance of giving false positives. This is why we are not worried with having removed the companies that give zero matches before we manipulate the names.

After the second search we go through the output removing problematic companies entirely. This may remove a few relevant tweets, but we do not have the time and resources to look through 205 tweets for American Express and 331 tweets for Bank of America Corp., so we assume that these have been identified in tweets where Trump mentions America and remove companies with names that includes the following exceptions: ”America”, ”American”, ”Anthem”, ”united”. We have also identified a problem with AT&T, which has been identified in 695 tweets. They are identified in short tweets with ’at’, so we remove all of these as well.

Finally we go through the tweets manually for the following companies: ’Unum Group’, ’Public Storage’, ’Dollar Tree’, ’Best Buy, ’Robert Half’, ’Avalon’, ’International Paper’, ’Waters’, ’Dish Network’, ’The Bank of New York Mellon’, ’Ball Corp’, ’E*Trade’, ’General Electric’, ’General Dynamics’, ’General Motors’, ’General Mills’, ’HCA Healthcare’, ’Southern Co.’, ’Chubb Limited’, ’CVS Health’, ’Juniper Networks’, ’Fox Class A’, and ’Lam Research’. We go through these because they had an improbable number of matches. Going through the tweets revealed that all but 10 pct. of the General Motors (GM) tweets and most of Fox

Class A tweets were false positives. E.g. the 'Unum Group' matches were from tweets where 'group' is featured.

Given this conclusion we identify the relevant GM tweets by searching them for keywords and we remove all other inspected tweets than these GM and the Fox Class A tweets. This leaves us with 723 tweets. Given how easily we could find more than a thousand irrelevant tweets in our initial fuzzy results there is bound to still be a lot of type 1 errors. Though we have not managed to automate the process fully it provides a good iterative process that could be repeated to improve the results.

The resulting 723 tweets are transformed into a data frame with time stamp of the tweet, company name, and the company ticker. We use the tickers to construct the relevant link to request from Alpha Vantage. The 723 tweets involves 115 companies. We request daily stock value, volume, close price, adjusted close price, high, and low for all 115 companies according to the time stamps in the 723 tweets. This process requires 723 calls to the Alpha Vantage API, which has a limit of 500 requests per day and only 5 requests per minute. Therefore we split the relevant tweets in 4 parts and run it on separate computers. When doing this it is paramount that you change the API-key each of the computers run with, otherwise there will be missing values (errors) when the sleep time of 13 seconds has not been respected. We pickle the results and combine them into one data frame.

Despite complying with the 5 requests per minute and using four different API-keys we still fail 49.5 pct of our calls. Unable to fix this problem we proceed with a sample of 366 tweets out of the initial 723. Out of the 366 tweets we also exclude those made on days where the market was closed, however, this might not be a disadvantage as new relevant public information and other exogenous shocks between opening hours would muddy the effects of the tweets we seek to measure. In the end we have a sample of 239 tweets for which we have company specific Δ stock and pct. change for a given tweet that is relevant for the company according to our fuzzy search approach.

3.3 Ethics (41)

In this section we consider the ethics of our data collection process. We find that all of our data is accessible for everyone with internet access and does not contain any sensitive information. Since the President wishes to reach as many readers as possible, we think it is in order to study his tweets as if they were categorized as public information.

In Salganik (2018) four principles of good data research are presented. Especially the first principle which is *respect for persons* is relevant for our project. The principle is about informed consent, which means "the participants should be presented with relevant information in a comprehensible format and they should voluntarily agree to participate." We have not informed Trump, that we are doing this project. Although Trump has not given us direct permission to use his tweets via the database, he has agreed to a set of terms and conditions, thereby acknowledging that his tweets will be seen as public information. Which could be interpreted as a way of Donald Trump giving informed consent.

3.4 Log (96)

Our log from collecting the final data has no errors as shown in Table 3.1. Collecting the data has been successful in the calls reported in the table and there does not appear to be missing observations or standard responses from the API. This is confirmed by visual inspections of our data.

Table 3.1: Overview of connection requests

id	project	connector_type	t	delta_t	response_size	response_code	success	error
0	Tweets	requests	1.567147e+09	-1.543273	2567864	200	True	NaN
1	Tweets	requests	1.567147e+09	-1.425995	1457152	200	True	NaN
2	Tweets	requests	1.567147e+09	-1.280184	949464	200	True	NaN
3	Tweets	requests	1.567147e+09	-1.533299	1482466	200	True	NaN
4	Tweets	requests	1.567147e+09	-1.711999	1592154	200	True	NaN
5	Bloomberg - Tweets	requests	1.567147e+09	-0.464530	10940	200	True	NaN
6	dji_data_collection	requests	1.567147e+09	-4.727906	1761268	200	True	NaN
7	sp_data_collection	requests	1.567147e+09	-4.174791	1743545	200	True	NaN
8	wikilist	requests	1.567147e+09	-0.256015	602888	200	True	NaN
9	wikilist	requests	1.567147e+09	-0.079007	602888	200	True	NaN

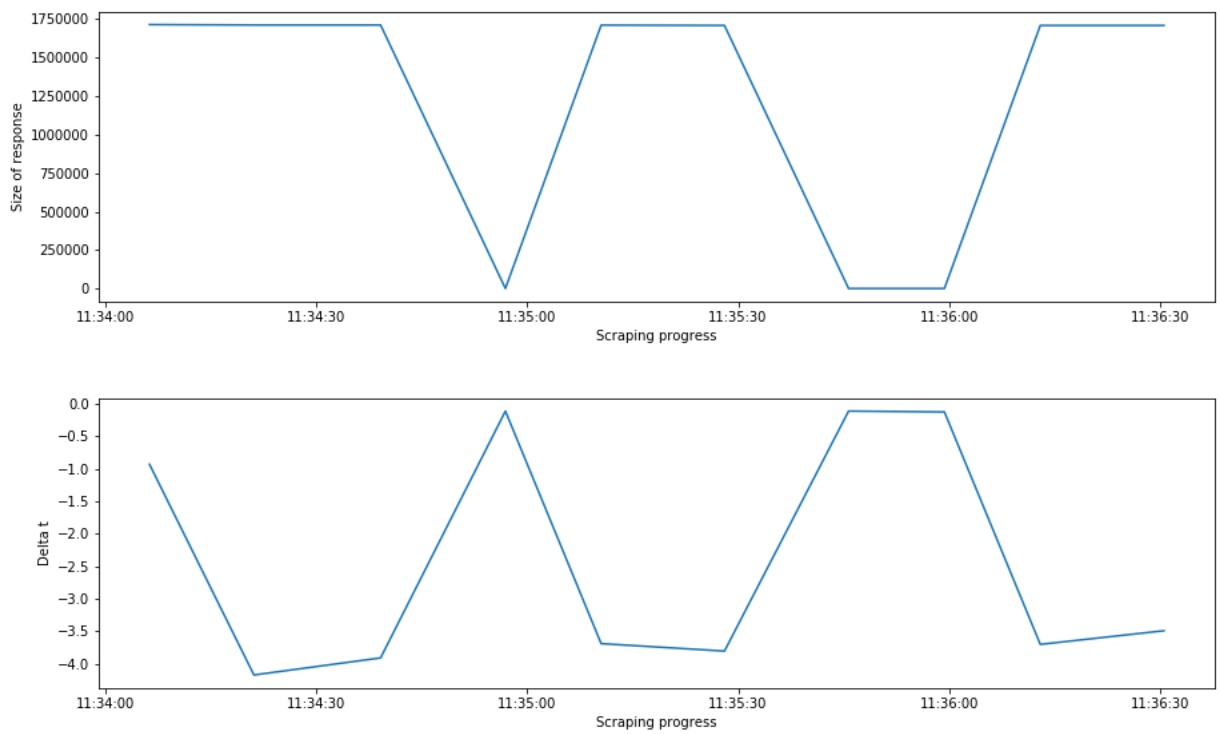
Our data collection process had problems two times. When scraping Bloomberg for `json` data on the market relevant tweets we were instead given the `html` text for the page: Are you a robot?. This does not show in the log but was apparent when inspecting the results. We circumvented this blocking mechanism by copying the `json` response directly.

When collecting financial data from Alpha Vantage for the individual firms we had a scraping issue. Even though we got no errors, there were missing observations with the following message in our data frame rather than the data we requested:

"Thank you for using Alpha Vantage! Our standard API call frequency is 5 calls per minute and 500 calls per day. Please visit https://www.alphavantage.co/premium/ if you would like to target a higher API call frequency."

We tried changing our or time-sleep and API-keys to comply with their API rules but got worse results in our second attempt. A plot of the logged response time and size of response is shown in Figure 3.1. These indicate that there has been spikes in response time with a corresponding drop in size of response. Unable to remedy this issue of missing observations left us with no alternative but to remove the observations with missing values and continue with a limited data set.The implications of this are discussed in Section 6.

Figure 3.1: Visualizing data calls to Alpha Vantage



Note: Data calls logged by the provided package ***scraping_class*** *and saved in a CSV format. Test calls not included.*

4 Description of Data

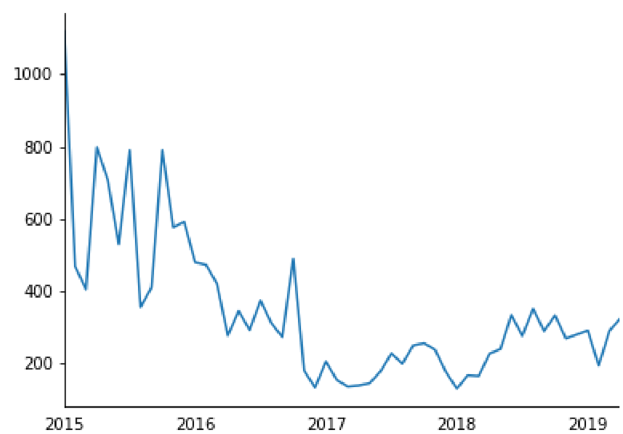
This section contains an outline of our data through descriptive statistics and figures, that aim to visualize the development of the American stock markets and the essence of Trump's tweeting; how many, most popular words, the sentiment, etc.

4.1 Trump and Twitter (127)

Figure 4.1 shows the amount of tweets that Donald Trump has written over the years. In (a) we see how he was most active around the time where he announced his candidature for presidency in 2015. Since then, the frequency of his tweeting has been descending towards his inauguration in the beginning of 2017, whereupon the curve shifts to increase over time. (b) shows the Bloomberg filtered tweets since he became president in January 2017. The curve is relatively volatile short-term but relatively steady long-term until March 2019, whereupon the curve is rising the last period.

Figure 4.1: The development in number of Trump's tweets per month

(a) All tweets (2015-2019)



(b) Bloomberg filtered tweets (2017-2019)

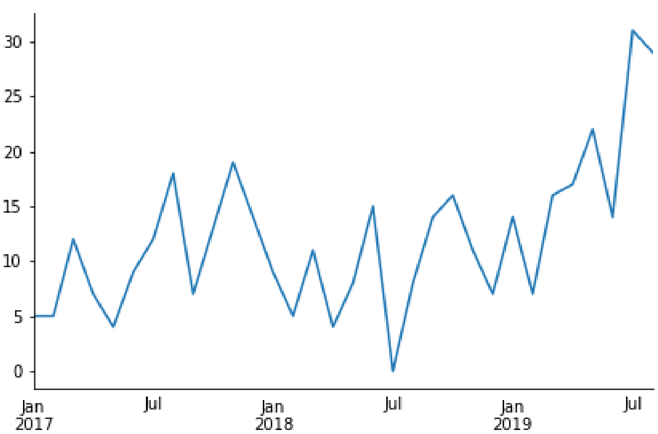


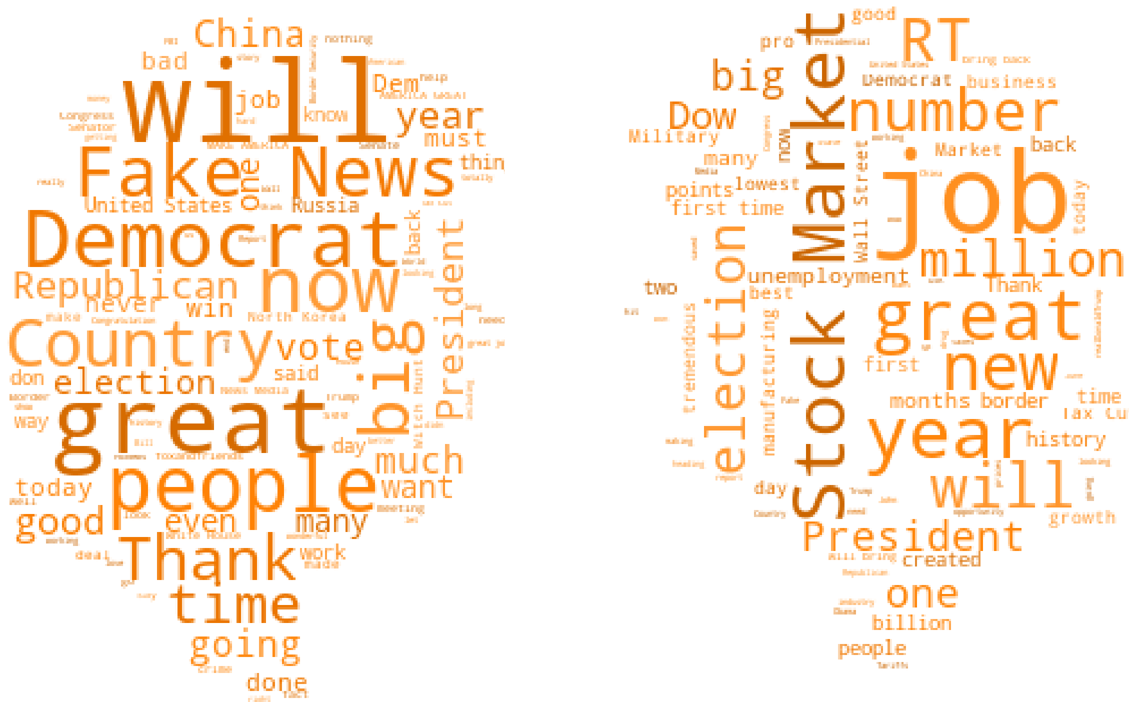
Figure 4.2 shows two clouds of words where the size of the words is proportional to the frequency of which the words appear in Trump's tweets. (a) shows that if you look at Trump's tweets in general, he is focused on what he "will" do, the "fake news", his opponents - the "democrats", and what is/could be "great". Words like "people", "now", "country", and "thank" is frequently used. If we look at his financially related tweets, (b) shows how "job",

"stock market", "great", "new", "year" and "will" are most frequently used. The caveat to this is that we do not know how Bloomberg filtered the tweets.

Figure 4.2: Wordcloud of Trump's tweets

(a) All tweets

(b) Financially related tweets



Note: Includes all tweets since the 20th of January 2017.

Note: The tweets have been filtered by Bloomberg and contains retweets. Tweets selected since the 20th of January 2017.

Table 9.1 in appendix shows the summary statistics for the sentiment in Trump's tweets. If we look at all tweets he has sent since 2015, the average sentiment of the 19,622 tweets is 0.22, where the scale goes from -1 (negative) to 1 (positive). His tweets are on average most positively loaded during his campaign in 2015 and have been least positive in 2019. For each year, the sample contains both very positive and very negative tweets, which is seen by the maximum and minimum scores in the table. Looking only at tweets concerning the financial market, the sentiment average on 0.32 of the 238 tweets is slightly more positive than what we saw when looking at all the tweets. The tweets from 2018 are the ones that on average are the most positively loaded, while the tweets from 2019 are the most negative, which might be due to the ongoing trade war with China. The sample also contains very positive and negative tweets.

Figure 9.1 in the Appendix shows the sentiment distribution in four cases, respectively all tweets and tweets selected by Bloomberg[19] both with and without sentiment scores of 0.00. The two graphs (a) and (c) contain the score 0.00, which we have decided to omit, c.f. Section 3.1. The tweets with a score of 0.00 are typically links or other formats, which can not be given a sentiment score and are also irrelevant for our analysis. Looking at (b) and (d), we see that both average distributions are skewed to the right, which means Trump has more tweets being positive than negative.

4.2 The Stock Market Development (41)

In Table 9.1 in the Appendix we see that Trump's average sentiment has been somewhat stable over the years. As one can see in Figure 4.3 that is not the case with the Dow Jones Index in the same period of time. From Trump's inauguration in the beginning of 2007, it is clear from (a) that the Dow Jones Index has increased over time. (b) shows the first differences from which it is clear how the stock prices have been quite volatile and shows signs of volatile clustering in the beginning of 2018 and 2019.

Figure 4.3: Development in the Dow Jones Industrial Average (2017-2019)

(a) Absolute values **(b)** First differences

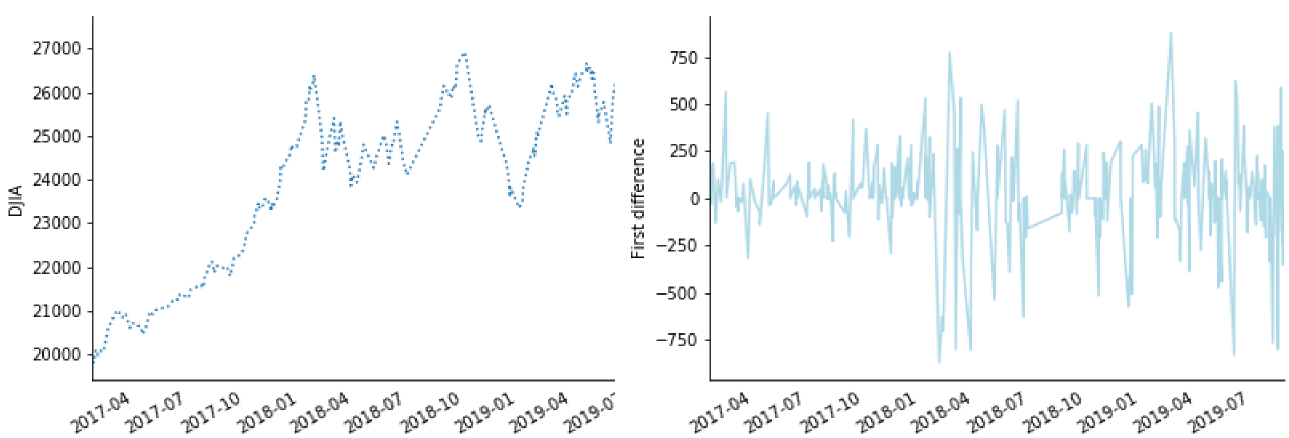
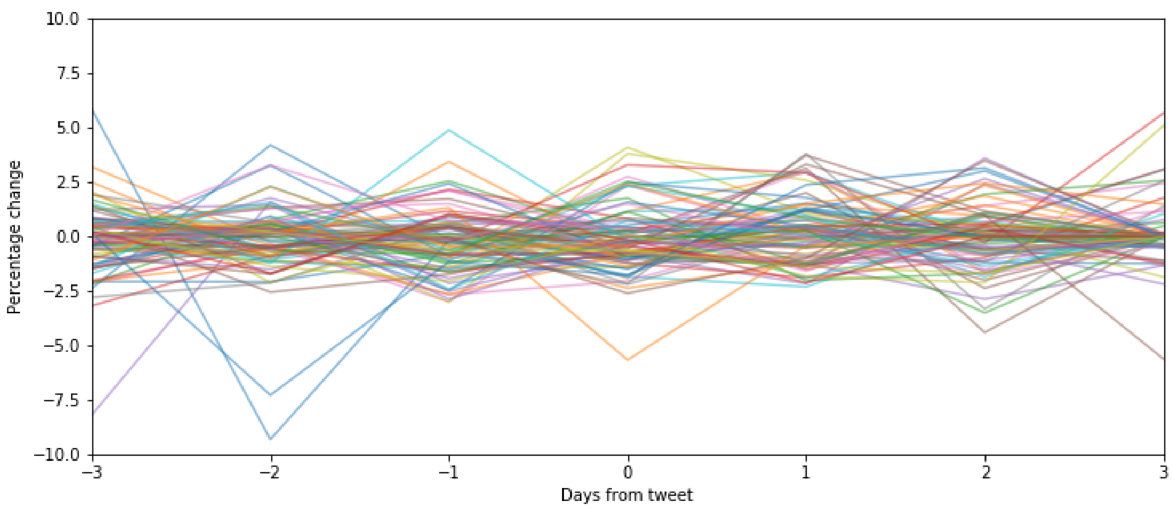


Figure 9.1 beneath displays the percentage change in the closing values for 89 companies that Trump has mentioned on Twitter. The figure includes the daily changes from three days before and three days after the tweet was made in order to compare. We see that Trump

[19]See e.g. Bloomberg (2019)

mentioning a company on Twitter has no obvious effect on the closing values. The figure provides a quick overview but it is also slightly too general, thereby motivating the further use of machine learning in order to perhaps gain new insight.

Figure 4.4: Percentage change in closing values for companies mentioned by Trump



5 Results

5.1 Regression Model (220)

In this section we present our results from our machine learning models using both regression (Table 5.1) and classification (Table 5.2) models. We tune our regression using both Ridge and LASSO regularization and find that the LASSO model outperforms both the linear regression and the Ridge. It should be noted that we have not proven any causality in this analysis and might therefore be subject to the usual pitfalls of reverse causality and omitted variable bias. Overall, we see that the smallest root mean squared error (RMSE) is obtained from using LASSO regularization on the model consisting of all tweets. Based on the data we see that the LASSO model removes sentiment as a feature. This is not surprising as all tweets encompass everything related and unrelated to the stock market and Trump's general sentiment could be considered irrelevant to the market movements. When we focus on his sentiment on tweets related to the economy we see that the LASSO model still performs the best, and that the sentiment does have a slight positive coefficient.

Table 5.1: Regression results

	Constant	**Time**	**Sentiment**	**Volume**	**Open**	**RMSE**
			All tweets			
OLS	1.9911 (0.108)	$6.322e-05$ ($3.78e{-}06$)	0.0018 (0.013)	$-2.183e-10$ ($1.18e{-}11$)	−0.0007 ($4.61e{-}05$)	1.271
Ridge	0	6.761e-05	2.555e-03	-1.853e-10	-7.349e-04	1.284
LASSO	0	6.752e-05	0	-1.852e-10	-7.338e-04	**1.168**
			Bloomberg tweets			
OLS	−1.7965 (1.064)	−0.0034 (0.001)	0.1132 (0.087)	$-1.457e-09$ ($5.84e{-}10$)	0.0001 ($4.9e{-}05$)	2.559
Ridge	0	-3.560e-04	7.125e-02	-3.144e-10	7.163e-06	2.019
LASSO	0	-3.557e-04	7.088e-02	-3.143e-10	7.151e-06	**1.239**
			Fuzzyword tweets			
OLS	−0.4833 (0.215)	−0.0006 (0.001)	0.2640 (0.171)	$3.09e-08$ ($7.55e{-}09$)	0.0008 (0.00)	3.291
Ridge	0	-5.021e-04	4.027e-02	2.878e-08	6.72e-04	2.292
LASSO	0	-5.018e-04	3.992e-02	2.878e-08	6.718e-04	**2.282**

Note: Standard errors repported in (·). The data from Bloomberg tweets is compared against DJIA data, 'All tweets' against S&P 500 and 'Fuzzyword' against the individual company stock price.

In Figure 5.1 we see the preceding learning curve plots for respectively all Trump's tweets (a) and the market relevant tweets only (b) - showing training set sizes and the mean squared errors as an evaluation metric. In both (a) and (b) we see, how the mean squared errors of the training data are reduced when the sample sizes increase, which means the more observations the better will our model predict. Fewer observations widen the gab between the train - and test curves, which is an indicator of an increasing degree of over-fitting. We also see, how the mean squared errors in (b) with selected tweets are smaller than in (a). That points to our model performing better on the data set in (b).

5.2 Classification Model (220)

For the classification we perform a k-fold cross validation, where we set $k = 10$, which is considered a good standard value.[20] As with any classification problem, this model is subject to Type 1 and Type 2 errors (false positives and true negatives). To evaluate the performance of our model we can either investigate the accuracy or the precision: i.e. whether the model is able to accurately predict the value of certain observations or alternatively how well it is at predicting rare cases. Since we have a balanced data set, i.e. the dummy values are fairly evenly split, we find that accuracy is a better measure of performance. The results are presented in Table 5.2 below.

Table 5.2: Classification accuracy

	All tweets	Bloomberg	Fuzzyword
Accuracy	47.9 %	67.65 %	31.82 %
Share of dummy = 1	50.0 %	58.64 %	44.35%

Note: Using k-fold cross validation with 10 folds. The indicator function returns 1 when the daily percentage change is positive, 0 otherwise. The data from Bloomberg tweets is compared against DJIA data, while the rest against S&P500 data.

When it comes to predicting the qualitative movement of the stock prices, we see that our model performs poorly when feeding it all Trump's tweets. With an accuracy of 47.9 pct. it is akin to flipping a coin. If we were to guess the stock market to increase every day of our observation set, we would have a higher accuracy given the share is 50 pct. When modelling on the Bloomberg filtered tweets we do find an increase in our accuracy at 67.65 pct. with a share of positive daily changes of 58.64 pct. Looking at the tweets filtered by our `Fuzzyword`

[20] See e.g. Raschka et al. (2017).

process we unfortunately at not able to predict the movement at all and with a score of 31.82 pct. Our model is even worse at predicting it compared to our original data set of all his tweets. This has numerous implications, one is that with 239 observations it is not enough to properly train the model, which is also indicated by Figure 5.1c where the convergence is not complete. As described in Section 3.2 we were hindered in expanding our data set by the API we used to collect data and relevant tweets might have been omitted, which could have a large impact on the model's predictive ability. Another implication is that the approach still needs refinement, which will be discussed later in Section 6.

Overall, the results points to that it is possible to predict the movements of the stock market to some degree based on Trump's sentiment. The prediction quality naturally is largely affected by the quality of the tweets. All models are best tuned using the LASSO regularization, meaning that the model does get improved by removing irrelevant features thus reducing underfitting. In the following section we discuss our results and the methodological considerations.

6 Discussion *(shared)*

We tried to improve existing research by focusing on the sentiment in only market specific tweets when building our model and by looking at the short term stock price for the companies that Trump mentions. By focusing on market relevant tweets, we limited our data quantity significantly. We could have obtained more relevant data by including other medias than Twitter. By way of example, Trump is also on Instagram, but it is obvious how his profile on this social media is more formal and maybe written by a communications employee as the president is often mentioned in third person in the description texts as well as many of the posts just cite him. Alternatively, one could also include his speeches, which Otani et al. (2017) puts emphasis on. Due to time constraints we decided to focus on Twitter data, because it gives a better impression of Trump's sentiment than Instagram and it is easier to access and analyse in Python than his public appearances.

We used `fuzzywuzzy` to pick out different combinations of words to make sure we got as many relevant tweets as possible. We did this because we can not assume, that Trump always mentions the names of the companies in the exact same way. This is a quite time consuming process, and if we had had more time we could make this process more fine-tuned and maybe end up with more and better data for our analysis.

Since fuzzystring matching was an entirely new field for us, our approach is very novice and flawed. We end up with a data set that performs worse than all tweets and worse than the Bloomberg market tweets. The simplest way of refining our data would be to continue the process of going through problematic companies and removing them. But there may be options for designing the search better fundamentally. One could consider using `sort_ratio` from fuzzy instead so the order of parts of the company names mattered for a second attempt.
Another idea would be to lean on the work of others who have tracked which companies Trump has mentioned and start from a shorter more relevant list of companies than the entire S&P 500 list. The ambition by starting there was to capture everything in a systematic way, but made it difficult to get an overview of which company names would be problematic and what remedies might be needed.

Considering the machine learning model, we could have expanded our analysis to include ElasticNet, a mixture of the Ridge and LASSO regularization to try and capture both the underfitting and overfitting problem in the same model. In this course, we have been introduced to linear models, but one could expand on this project by comparing to include non-linear regressions, but this was outside the scope of this project. In regards to the performance evaluation it might be interesting to construct a confusion matrix to analyse different error types.

In regards to the semi-strong form of the Efficient Market Hypothesis, our final results could serve as evidence either for or against the hypothesis depending on how you choose to interpret it. Our results for the general market concluded that no real pattern between the stock values and Trump's tweets could be detected. This would serve as further evidence against the semi-strong form of the EMH.

However, when using the tweets provided by Bloomberg, our model were able to predict the movement on the stock market with an accuracy of 67.65 pct. Those results would suggest that a new tweet regarding the stock market made by Trump would serve as new public information. In Mittal et al. (n.d.), where they tested a hypothesis of whether public sentiment would have an effect on market sentiment, their model achieved an accuracy of 75.56 pct., serving as evidence for a correlation between public and market sentiment and thus working as evidence for the semi-strong EMH. If Trump's sentiment could serve as a proxy for public sentiment, our report too would serve as evidence for the semi-string EMH due to the fact that our model are able to somewhat predict patterns on the market with an accuracy similar to the referenced one.

One could also question the research area: is Trump's sentiment even relevant to market price movements? It is possible - and proved[21] - that the market reacted nervously the first times Trump mentioned companies, but the effects often only lasted a short term. If the market by now have figured out that Trump is a man of BIG WORDS and expressions but rather bound in what he is able to actually do, then the market should be able to call the effect as only short term and not react.

[21] See e.g. Otani et al. (2017) and Colonescu (2018).

7 Conclusion (127)

In this project we have analyzed Trump's tweets effect on stock market data. We use machine learning both to try and predict the relative changes given the sentiment of his tweets and predict the qualitative movements of the prices. We tried to improve on the literature by focusing on company and economic related tweets. The former part proved to be quite difficult as it required a delicate type of fuzzy string matching in order to try and gauge relevant tweets. Modelling all tweets we found that the sentiment was removed in the regression based on a LASSO regularization, which indicates that Trump's sentiment had no bearing on the stock market movements. Filtering on Bloomberg tweets we see that the sentiment has a positive coefficient. In our classification model we obtain an accuracy of 47.9 pct. and 67.65 pct. for all tweets and Bloomberg tweets, respectively.

Overall, by selecting tweets more relevant to the stock market we were able to increase the performance of our models. However, for the purpose of real life applications the performance of our models are still insufficient.

7.1 Further Work

Further work in this field could consist of building a more thorough and specific dictionary for the sentiment analysis. A lot of words are neutral isolated, but e.g. "China" could in context of the present trade war mean a lot for the stock prices. Better results might have been obtained if we had built a model that analyses events or foreign politics - and relations rather than the American market and company mentions. An idea could be to create a data set of classified relevant tweets that could allow one to do machine learning on the categorization of tweets in relevant and irrelevant. Ideally, one would also present a confusion matrix to present the multiple different types of errors to further evaluate the performance of the model(s).

8 References

Books

Angrist, Joshua D. and Jörn-Steffen Pischke (2009). *Mostly Harmless Econometrics: An Empiricist's Companion*. First edition. Princeton University Press.

Raschka, Sebastian and Vahid Mirjalili (2017). *Python Machine Learning*. Second Edition. 35 Livery Street Birmingham B3 2PB, UK.: Packt Publishing.

Verbeek, Marno (2017). *A Guide to Modern Econometrics*. Fifth Edition. Frederiksberg: John Wiley & Sons Ltd.

Salganik, Matthew J. (2018). *Bit by Bit: Social Research in the Digital Age*. First Edition. Oxford Street, Woodstock, Oxfordshire: Princeton University Press.

Articles

Fama, Eugene F. (1970). "Efficient Capital Markets: A Review of Theory and Empirical Work". In: *The Journal of Finance* 25, pp. 383–417.

Malkiel, Burton G. (2003). "The efficient market hypothesis and its critics". In: *Journal of Economic Perspectives* 17, pp. 59–82.

Johan Bollen Huina Maoa, Xiaojun Zengb (2011). "Twitter mood predicts the stock market". In: *Journal of Computational Science*. URL: `https://www.sciencedirect.com/science/article/pii/S187775031100007X`.

Nielsen, Finn Årup (2011). "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs". In: *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages* 718, pp. 93–98.

Hutto, C.J. and Eric Gilbert (2015). "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text". In: *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*.

"Twitter Sentiment Analysis Applied to Finance: A Case Study in the Retail Industry" (2015). In: URL: `https://www.researchgate.net/publication/279864932_Twitter_Sentiment_Analysis_Applied_to_Finance_A_Case_Study_in_the_Retail_Industry`.

Ribeiro, Filipe N. et al. (2016). "SentiBench - a benchmark comparison of state-of-the-practice sentiment analysis methods". In: *Springer Berlin Heidelberg*.

Jeffrey Born, David H. Myers and William J. Clark (2017). "Trump tweets and the efficient Market Hypothesis". In: *Algorithmic Finance*. URL: https://content.iospress.com/articles/algorithmic-finance/af211.

Otani, Akani and Shane Shifflett (2017). "Think a Negative Tweet From Trump Crushes a Stock? Think Again". In: URL: https://www.wsj.com/graphics/trump-market-tweets.

Colonescu, Constantin (2018). "The Effects of Donald Trump's Tweets on US Financial and Foreign Exchange Markets". In: *Athens Journal of Business and Economics*, pp. 375–388.

Hern, Alex (2018). "Cambridge Analytica: how did it turn clicks into votes?" In: *The Guardian*. URL: https://www.theguardian.com/news/2018/may/06/cambridge-analytica-how-turn-clicks-into-votes-christopher-wylie.

Websites

Aries, Francisco Javier Carrera (2019). *Fuzzy String Matching in Python*. URL: https://www.datacamp.com/community/tutorials/fuzzy-string-python (visited on 08/29/2019).

Bloomberg (2019). *Trump Tweets Market*. URL: https://www.bloomberg.com/features/trump-tweets-market (visited on 08/25/2019).

documentation, sklearn (2019). *sklearn.TimeSeriesSplit Documentation*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html (visited on 08/25/2019).

Natural Language Processing for Fuzzy String Matching with Python (2019). URL: http://en.wikipedia.org/wiki/List_of_S%26P_500_companies (visited on 08/29/2019).

Mittal, Anshul and Arpit Goel (n.d.). *Stock Prediction Using Twitter Sentiment Analysis*.

9 Appendix

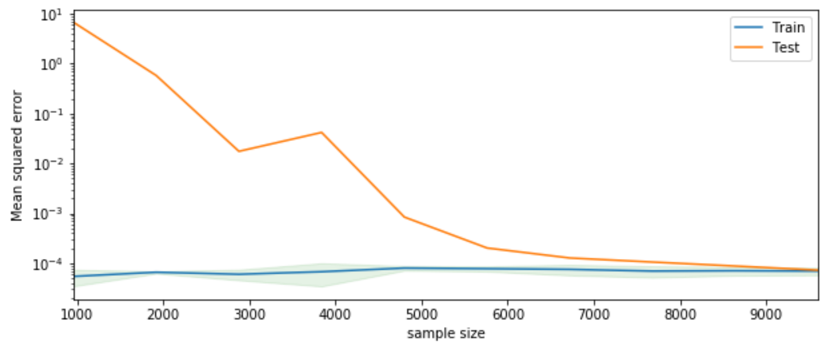
Table 9.1: Summary data for sentiment in Trump's tweets

period	count	mean	std	min	25%	50%	75%	max
				Selected Bloomberg tweets				
All periods	383	0.32	0.54	-0.97	0.00	0.53	0.79	0.90
2017	125	0.31	0.49	-0.97	0.00	0.43	0.73	0.94
2018	108	0.36	0.53	-0.90	0.00	0.54	0.81	0.98
2019	150	0.30	0.59	-0.97	-0.14	0.54	0.79	0.96
				All tweets				
All periods	19,622	0.22	0.55	-0.99	-0.10	0.36	0.70	0.99
2015	7536	0.26	0.48	-0.97	0.00	0.36	0.67	0.98
2016	4037	0.17	0.54	-0.96	-0.26	0.34	0.62	0.97
2017	2292	0.20	0.58	-0.97	-0.25	0.36	0.72	0.98
2018	3049	0.23	0.63	-0.99	-0.34	0.42	0.80	0.99
2019	2708	0.16	0.63	-0.98	-0.44	0.29	0.74	0.98

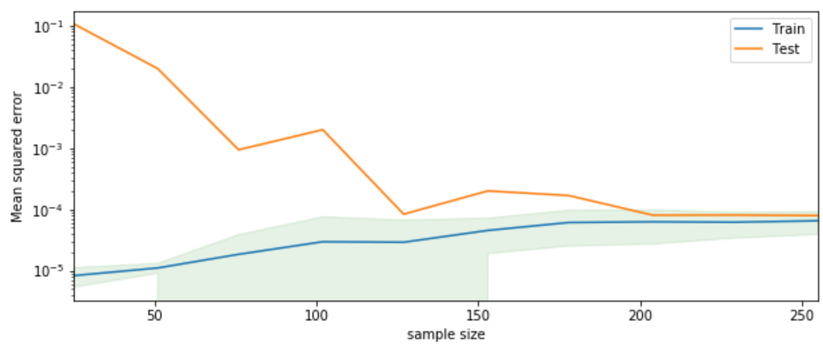
Note: All tweets encompass all Donald Trumps' tweets since 01/01/2015. Selected tweets are tweets selected from Bloomberg since 20/01/2017.

Figure 5.1: Learning curves

(a) All tweets



(b) Bloomberg filtered tweets



(c) Fuzzyword filtered tweets

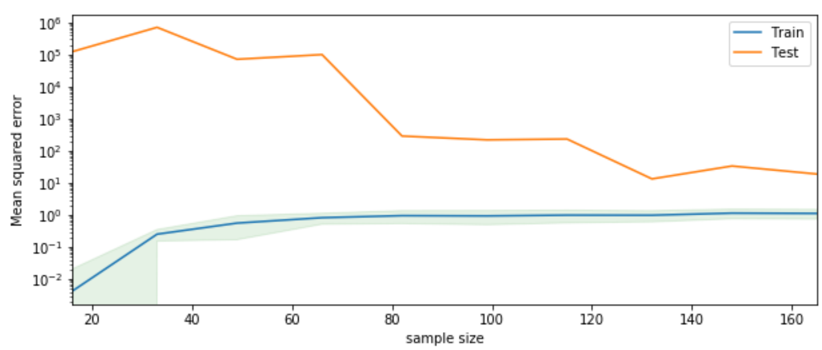
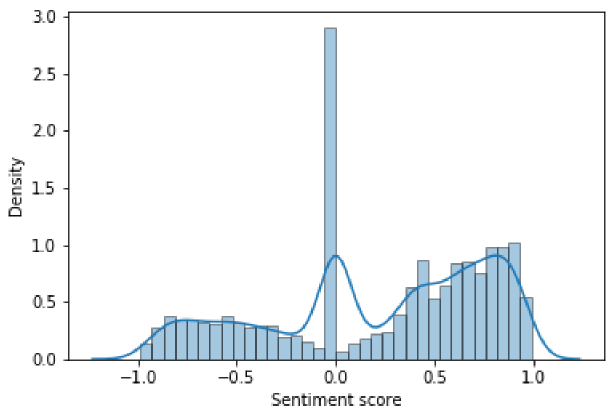


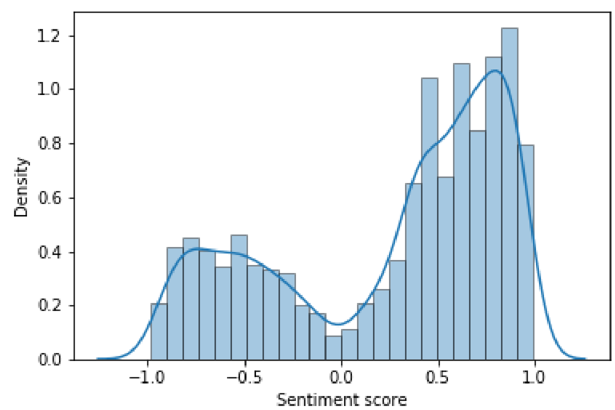
Figure 9.1: Sentiment distribution of Trump's tweets

All tweets

(a) Initial distribution

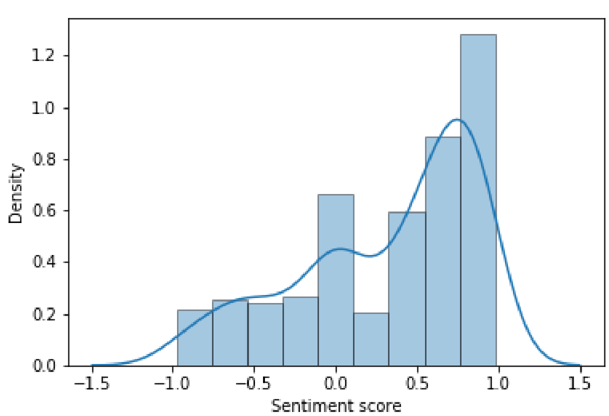


(b) Distribution without sentiment score of 0.00



Tweets selected by Bloomberg

(c) Initial distribution



(d) Distribution without sentiment score of 0.00