Queen Mary
University of London

Specific Learning Difficulties Cover Note

Student ID: 150397568

Advice for assessors and examiners

Guidelines for markers assessing coursework and examinations of students diagnosed with Specific Learning Difficulties (SpLDs) –
As far as the learning outcomes for the module allow, examiners are asked to mark exam scripts sympathetically, ignoring the types of errors that students with SpLDs make and to focus on content and the student's understanding of the subject. Specific learning difficulties such as Attention Deficit Disorders, dyslexia and or dyspraxia may affect student performance in the following ways:

- The candidate's spelling, grammar and punctuation may be less accurate than expected
- The candidate's organisation of ideas may be confused, affecting the overall structure of written work
- The candidate's proof reading may be weak with some errors undetected, particularly homophones and homonyms which can avoid spell checkers

Under examination conditions, these difficulties are likely to be exacerbated. Errors are likely to become more marked towards the end of scripts.

Useful approaches can include:

- Reading the passage quickly for content
- Including positive/constructive comments amongst the feedback so that students can work with specialist study skills tutors on developing new coping strategies
- Using clear English and when correcting; explain what is wrong and give examples
- Using non-red coloured pens for comments/corrections

Colleagues in schools are asked to ensure that students with specific learning difficulties access the support provided by the Disability and Dyslexia Service.

For more information regarding marking guidelines see DDS webpage http://www.dds.qmul.ac.uk/staffinfo/index.html and the Institutional Marking Practices for Dyslexic Students

Disability and Dyslexia Service
Student Services
Room 2.06 Francis Bancroft Building
www.dds.qmul.ac.uk
Tel: 020 7882 2756 Email: dds@qmul.ac.uk

Final Year Undergraduate Project 2018/19

BSc(Eng) Information Technology Management for Business (ITMB)

School of Electronic Engineering and Computer Science

Can Twitter be a Herald for the Stock Market? The Predictive Power of Tweet Volume on Stock Volatility

Project Report

David Sint

d.sint@se15.qmul.ac.uk

150397568

Supervised by

Dr. Arman Khouzani

arman.khouzani@qmul.ac.uk

23rd April 2019

Queen Mary University of London

Disclaimer

This report, with any accompanying documentation and/or implementation, is submitted as part requirement for the degree of BSc(Eng) Information Technology Management for Business (ITMB) at Queen Mary, University of London. It is the product of my own labour except where indicated in the text. The report may be freely copied and distributed provided the source is acknowledged.

Acknowledgements

The author would like to thank the couple of financial experts for their help explaining some of the more abstract financial concepts and their support in the project. Immense gratitude and thanks would also like to be extended to the author's supervisor, Dr Arman Khouzani, for his tremendous help and time teaching, advising and sharing his wisdom.

Abstract

Traders on stock markets have used traditional sources of data to inform their decisions, like company quarterly reports, government fiscal policy and the news. Yet other, less conventional, sources of data have become an interesting source of information. One such unconventional resource is microblogging websites like Twitter, which has over 300 million monthly users, as of early 2019. Users can write about almost anything they like, including about stock markets.

Predicting trends, like stock volatility are notoriously difficult tasks. The question of this report is whether the conversation on Twitter has any predictive power on stock volatility, or whether it is simply noise. The central premise is that an increase in the volume of tweets regarding a stock, could be an early symptom of turbulence to come. I have put this hypothesis to the test by scraping and analysing financially focused tweets, through their use of Twitter's *cashtag*, to assess whether a stock's hourly volume of tweets can somehow predict its next day's price volatility. After gathering the tweets and stock prices for the S&P 100 index, correlation between these elements was tested by computing their Pearson Correlation Coefficients (with p-values). This analysis showed a low, yet statistically significant, correlation between hourly tweet volumes and the next-day volatility for a range of stocks.

Subsequently, four supervised machine learning regression models were utilised to see whether improvement could be found by including the volume of tweets as features. Of the stocks analysed, 93% saw some improvement at predicting the next day's volatility when the hourly volume of tweets was included as features. Linear SVM Regression was found to be the most effective model of the four analysed with 58% of stocks finding their greatest improvement with it.

Keywords: Twitter, Stock Market, Volatility, Machine Learning, Regression

Contents

1 Introduction

1.1 Project Aims and Objectives

Predicting the market for stocks and shares is a contested subject in terms of the best methods and whether it is even possible. Traditional approaches include examining company financial reports, following market momentum and making bets based on intuition and belief.

Non-traditional methods examine unconventional data sources that may signal that a stock will make an adjustment, for example, tracking a company jet (Nair et al., 2017), analysing a CEOs body language or scraping Twitter. The social media and micro-blogging site Twitter is one of the largest and most well-known and popular of these relatively new social-media platforms. In addition, most tweets are public and usually textually based with a 140, and from 2017, 280-character limit. Since anyone can post almost anything on Twitter, one may argue that tweets cannot be used to make trading decisions. However, I will be analysing the crowd en-masse, to see if signals within the noise will begin to appear.

Volatility of a stock price can be seen in its standard deviation. High volatility increases the risk profile of a stock as its price rises or falls drastically. Through trading options, it is possible to buy or sell stock at a specified price over an agreed period of time. During a period of high volatility in a stock's price, profit can be made through well timed options trades independent of the up or down direction of the stock price using options trading. This is one reason why knowledge about when volatility will occur is valuable information to a trader. The volume of tweets about a stock could indicate volatility as significant changes to a stock price are often reactions to external factors that people may be tweeting about before the stock adjusts. Therefore, the volume of tweets over time about a specific company could indicate period of volatility. It is important to note that volatility does not specify a specific trajectory, rather fluctuations in general. Pertinent to this project, the root cause of the volatility, whether external factors or the tweets themselves, are not relevant, only that the volatility can be predicted from

the tweets. This project will attempt to find out whether it is possible to use machine learning analysis to examine financial themed tweets in order to predict the volatility of stocks on the market.

1.2 Motivations and Challenges

Primarily this project, as with any project aimed at forecasting the stock market, aims to be able to generate advice to traders that can be used to influence investing decisions. Some traders may choose to exit a market or sell shares in volatile times, others may take on the challenge and use more sophisticated techniques which favour higher volatility. It is even possible to directly profit from volatility through trading on the Chicago Board Options Exchange's Volatility Index (VIX) in an Exchange Traded Note (ETN) like the VXXB (CBOE, 2019; iPath, 2019)

Furthermore, with social media taking an increasingly prominent role in people's lives, this project will continue the examination of whether social media posts can collectively lead to investment decision suggestions as some literature has already investigated (Bollen et al., 2011; Chen et al., 2014; Li et al., 2018; Tsui, 2016).

The micro-blogging website Twitter, is the largest social media platform that is predominantly public and makes use of tagged content. In particular, the site makes use of *cashtags* which are used to refer to stocks by their ticker value (Twitter, 2018a). It is possible that someone might use the cashtag, for example $F, in a non-financial manner, however, predominantly, this tweet will be referring to the Ford Motor Company stock or organisation in some way. These factors makes it a suitable platform for obtaining large quantities of public financial-focused sentiment about stocks.

As for challenges, I realise that users of Twitter would often tweet about companies in the public consciousness, usually those that consumers interact with in everyday life. Therefore, it is expected that consumer-facing businesses and large enterprises would have the most number of tweets about them. This limits the scope of the solution, because if there is limited interest on Twitter about a company, the output of this research will unlikely be effective, due to a lack of data, to make a

prediction.

In addition, volatility is often affected by external factors. For example, during the time of analysis (October 2018 – April 2019), many British stocks are currently affected by the developing 'Brexit' situation. This issue will then remain invisible to the algorithm if Twitter users do not tweet using a *cashtag* identifier for a British stock in tweets about 'Brexit' occurrences that affect the market as a whole. These challenges must be addressed in the design of this project.

1.3 Literature Review

Before designing the experiment, a review of the literature was conducted to find the ways Twitter was already being used to predict the stock market and to examine these experiments methodologies to inform my own. I will now discuss these experiment's results and methodologies.

1.3.1 Results

In the past decade since 2010, the web has become increasingly accessible, particularly with the adoption of mobile computing. This has meant that people are now communicating publicly in a recorded format that can be analysed and used to make decisions.

Often these recorded formats can be through search engine queries, news media articles or social media posts, which Mao et al. (2011) used to analyse their predicative power on stock market returns. The authors found that Google search volumes of their financial search terms "is indeed predictive of financial indicators" including stock and index returns, volatility and trading volume (ibid., p. 9). This was later confirmed by Dimpfl and Jank (2016, p. 171), in their finding that "a heightened number of searches today is followed by an increase in volatility tomorrow".

In terms of negative news sentiment, Mao et al. (2011) found it to be statistically significant as a predictor, however, Twitter sentiment and the tweet volume of their

financial search terms were better at predicting the log return (compounded rate of return for a stock in a specified period). Although the authors did not examine whether Twitter volumes were predicative of stock volatility, they did find that "before the highly downward movement of DJIA in the end of July and August 2011, Tweet volumes of financial terms started to increase several weeks earlier than Google volumes did. This indicates a potential efficiency gain of Twitter" (p. 9).

From this research it is apparent that Twitter sentiment and volume of tweets have some predictive power in at least some financial indicators. Therefore, for the research of predictive power and stock market volatility, Twitter seems to be a worthy medium to examine.

Indeed, Twitter has been the source of research in recent times, particularly in terms of sentiment analysis. Nofer and Hinz (2015) found that in their period of research, aggregate Twitter sentiment alone did not correlate with market value, although, when they also considered the follower count (perhaps an indicator of influence), the authors were able to design a trading strategy that beat their benchmark index by nearly 166% in a six month trading period, even after accounting for transaction costs.

Even by examining the sentiment of Twitter as a whole, Bollen et al. (2011) found that by isolating some mood dimensions of tweets, in particular 'calmness', it was possible to predict shifts that occurred in the Dow Jones Industrial Average (DJIA) 3-4 days later. However, they acknowledge that this may be because "the particular period under observation Twitter.com users were *de facto* predominantly English speaking and located in the US" (ibid., p. 7). Zhang et al. (2011) further found in preliminary results that when any emotions on Twitter "fly high", whether "hope, fear and worry", the DJIA goes down the next day. Conversely, when there is less emotion, the DJIA will more often go up the next day.

Some papers have also examined the volume of tweets, but not always as an independent variable. Ranco et al. (2015) use tweet volume to identify 'events' for their 'event study' methodology but then use the sentiment of tweets during the event as the features, rather than the volume of tweets. Others have used the tweet

volume data in their analysis but did not make conclusions about its effectiveness as a predictor such as Rao and Srivastava in their 2012 conference proceedings (Rao and Srivastava, 2012).

Perhaps the most relevant paper from Oliveira et al. (2017) attempted to forecast the returns, volatility and trading volume of a range of indices and portfolios including the S&P 500 using a range of machine learning regression techniques. The authors measured volatility through the VIX and annualised realised volatility of the indices. The results showed that "The inclusion of the number of tweets do not seems to benefit the forecasting of volatility. There are no models utilizing posting volume that significantly outperform the respective baseline models" (ibid., p. 137). Oliveira et al. do not examine the short term effects of tweet volume on the volatility of individual stocks, particularly those frequently tweeted about like the most valuable.

Overall, the literature shows that there is some relationship between social media and the markets which warrants further examination. As Ranco et al. (2015, p. 18) say, "there is a signal worth investigating which connects social media and market behavior". In particular and to the best of my knowledge, it has not yet been investigated whether the hourly volume of tweets from a day n can predict the volatility of the day $n + 1$ for a stock.

1.3.2 Methodologies

In terms of the methodology of the studies mentioned, I have collected key details from each paper and summarised them in Table 1. The key details consist of the research's time period of examination, data sources they used in addition to the stock market, volume of tweets collected, analysis model type, features analysed, and the target stock market indicator (or multiple targets) the authors attempted to predict.

	Period (months)	Data Sources	Tweet Volume	Model type	Features	Target(s)
Mao et al. (2011)	15	Survey, Google queries, Twitter, News headlines	N/A	Multiple Regression	Survey sentiment, negative news sentiment, google search volumes of financial terms, Twitter sentiment, tweet volumes of financial terms	DJIA, VIX & gold price and DJIA trading volume
Nofer and Hinz (2015)	35	Twitter	≈ 100,000,000	Ordinary Least Squares	Twitter sentiment in Germany, user's follower count	DAX return
Bollen et al. (2011)	11	Twitter	9,853,498	Neural Networks	Twitter sentiment	DJIA price
Zhang et al. (2011)	6	Twitter	4,791,038	Correlation	Twitter sentiment, retweet count, followers count	DJIA, NASDAQ, S&P 500 and VIX price
Ranco et al. (2015)	15	Twitter	1,555,770	Correlation	DJIA cashtagged Twitter Sentiment	DJIA return
Rao and Srivastava (2012)	13	Twitter	4,025,595	Correlation	DJIA, NASDAQ 100 and 13 tech stocks Twitter sentiment	DJIA, NASDAQ 100 and 13 tech stock's close, return and volatility
Oliveira et al. (2017)	35	Survey, Twitter	≈ 31,000,000	Multiple Regression, Neural Networks, Support Vector Machine, Random Forest and Ensemble Averaging	US stock cashtagged Twitter sentiment, first difference volume, survey sentiment,	S&P 500, RSL, DJIA, NASDAQ 100 and various portfolios daily returns, trading volume and volatility
Sint (2019)	5	Twitter	2,765,230	Random Forrest, Linear, Linear SVR and K-Nearest Neighbor	S&P 100 cashtagged Tweet volumes	S&P 100 individual stock volatility

Table 1: Summary of methodologies from literature reviewed

2 Design, Requirements and Management

For this report, it was pertinent to first perform an analysis to devise the design criteria of the project. These design criteria are the requirements for the project. This section will outline the design decisions chosen with explanations before stating the requirements I produced from these decisions. Finally, I will briefly comment on some of the tools I chose to use to implement the experiment.

2.1 Design

Fundamentally, this project needed to be designed so as to find whether there is evidence to the hypothesis that the volume of tweets about a stock can predict its future volatility. To carry this out, I followed a methodology similar to prior research on the topic of using Twitter to predict the stock market. The design was the result of some analysis, which will be explained in this section.

2.1.1 Concept

In order to verify the validity of the claim that the volume of tweets about a stock can predict its future volatility, I decided to examine whether the volume of tweets on a given day n can predict the volatility of the stock's next day, $n+1$. Choosing parameters such as what period of time to use when counting the tweets, whether this period of time should be split into smaller subsets of time (bins), how long the delay is between this window and the period when volatility is measured, and how long the period of volatility is measured need to be decided. However, I am not looking to tune these parameters to such a degree that the prediction can be as accurate as possible. Instead, first it is necessary to identify whether there is any substance to the claim that the volume of tweets a stock receives can positively affect the prediction of stock volatility.

I needed to choose some stocks to analyse and I settled on Standard and Poor's 100 (S&P100) index. This is a subset of the top hundred stocks of the larger S&P500

index. A hundred industry balanced blue-chip American companies constitute the index (S&P Dow Jones Indices, 2019). The assumption was that companies in this index like Apple ($APPL), Facebook ($FB) and Microsoft ($MSFT) would be the most tweeted about due to their size and being well known organisations. Further, volatility is often affected by external influences like politics. As a response to this analysis, I opted to choose U.S. index rather than a FTSE based index to limit the volatility that 'Brexit' may have had over the period of research. The list of S&P 100 stocks that I will be tracking for this project are based on the 100 stocks as on the 2018/10/29 when the stocks were selected. They are as follows (iShares, 2018):

AAPL	WFC	WMT	HON	SBUX	MS	BK
MSFT	PG	PM	LLY	COP	SPG	EMR
AMZN	T	MCD	ACN	LOW	AGN	COF
BRKB	INTC	ORCL	COST	GS	CHTR	MET
JNJ	CVX	NFLX	GE	UPS	FDX	F
JPM	CSCO	ABBV	PYPL	CVS	OXY	AIG
FB	HD	AMGN	NKE	AXP	CL	KHC
XOM	MRK	MO	GILD	SLB	CELG	ALL
GOOG	BA	MDT	UTX	LMT	RTN	KMI
GOOGL	KO	DWDP	QCOM	CAT	BLK	HAL
PFE	MA	ABT	TXN	WBA	SO	FOX
BAC	DIS	NVDA	BKNG	BIIB	GD	
UNH	CMCSA	IBM	NEE	DHR	TGT	
V	C	MMM	BMY	DUK	GM	
VZ	PEP	UNP	USB	MDLZ	EXC	

List of stock tickers to be used for my research

To predict the future stock volatility for these tickers, I planned to use machine learning. This will require data about the stocks to train and test to the machine learning models.

I therefore started to collect data about these stocks from 2018/10/29 until the

2019/01/01. I later extended this period to gather more data until 2019/03/25, at which point I began to analyse what had been collected. This data consisted of tweets about the company using the *cashtag* as the identifier, and stock market prices.

2.1.2 Data

For this project, minute-by-minute stock data will be required so that the stock's volatility can be analysed. Minute-by-minute data will allow periods within the day to be utilised.

2.1.2.1 Datasets and features

Following the methodology of Oliveira et al. (2017, p. 128) who trained their machine learning regression on a "baseline model (without microblog features) and microblog based (with such features)", I decided to follow them and develop two datasets to train the models allowing me to compare them and their differences. This methodology would let me see whether the volume of tweets made a difference to the accuracy of predictions.

A dataset's features are input variables to an algorithm. They would need to be engineered in such a way that the machine learning algorithms could take them as inputs to predict the outputs. Therefore to summarise a day's stock market value I opted to choose the mean and standard deviation during the day n. However, to provide more granular data, I split the trading day n into 6 equal parts of 65 minutes. This time was chosen because it is was a period of time close to an hour that could divide into 390 minutes (in a trading day) without a remainder, leading me to settle on 65 minutes. The tweets volume periods could be divided into hours. This was because any less time might mean there is too little data for some of the less tweeted stocks, while any more time might be too much for some of the more tweeted about stocks. This parameter could almost certainly do with more fine tuning, however, this was not the primary intention of this project, and as such was not carried out. This could be adapted or developed in future works.

The target for each day in a stock's dataset needs to be the next day's volatility. This is the last column in each row of the datasets.

2.1.3 Correlation

For machine learning to work, the features of a day must correlate with the targets. Therefore, before starting with the machine learning, I first needed to determine whether there was any correlation between the features and targets. If there was none, then this experiment would likely not yield any positive results.

As the data is on a continuous scale (i.e. time), we should use a parametric correlation, the most popular of which is 'Pearson' correlation (Boslaugh, 2012). If there appears to be some correlation for a stock's volume of tweets and the next day's stock volatility, then this indicates that the machine learning regressors may be able to predict future stock volatility.

2.1.4 Machine Learning Testing

To predict the future volatility using supervised machine learning, one can use two main types of machine learning models. The first is a classification model (called a classifier) and the second is a regression model (called a regressor). A classifier will endeavour to predict a category from given features. For example, given the volume of tweets we may want to predict if the next day has high volatility or low volatility. A regressor on the other hand, will try to predict a quantity from the given features. This will mean that it will try to predict the numerical value of volatility. For this project, we are trying to predict the numerical volatility and therefore regression will be used rather than classification.

To measure the effectiveness of the regression models, I plan to identify the 'Mean Absolute Percentage Error' (MAPE), similar to previous literature. As Oliveira et al. (2017, p. 133) noted, "other related works also have adopted absolute error metrics, such as: Mean Absolute Error (MAE) (Deng et al., 2011) and Mean Absolute Percentage Error (MAPE) (Bollen et al., 2011; Deng et al., 2011; Mao

et al., 2011)". By adopting the percentage error, this normalises the results to a degree between the results from the different stocks.

Oliveira et al. (2017) used four machine learning models and so have I, three of which are the same and one that is different. I planned to keep Random Forrest Regression, Linear Regression, and Linear SVM Regression (also called Linear SVR). I also planned to use K-Nearest Neighbor. For the regressors parameters, I left the parameters to their defaults as recommended by the library used. I only changed the defaults when it was necessary due to the nature of this project and the data captured. For the times that this occurred, I will endeavour to explain why those decisions were made in this report. The high-level explanation of each regressor will be explained.

2.1.4.1 Random Forrest

The way that Random Forest Regression works, is that it will pick random rows from the dataset. Using these, it will build a 'decision tree' which is a structure that holds various diverging paths and probabilities for the paths, each path ends in an outcome. This will be repeated for a given number of times to produce many decision trees from the dataset – a forest. When a prediction needs to be made, each decision tree will use the given features to follow their paths to predict the outcome. These outcomes are then averaged to produce the Random Forest prediction. (Breiman, 2001)

2.1.4.2 Linear Regression

With Linear Regression, each of the features are assigned a standardised weighting, called a beta coefficient. These coefficients estimate the degree to which changes in their feature affects change to the target. If the volume of tweets in the final hour of a day is a strong indicator of the next day's volatility, it will be assigned a higher beta coefficient in Linear Regression. It can be written as:

$$y_n = + \sum_{i=0}^{k} \beta_i x_{ni} + \epsilon_n \tag{1}$$

Where x_i are the k features and y is the target. For each sample n, the value of $y(n)$. The β coefficients are found by minimising the error of prediction. The mean of ϵ should be 0, as it is a random error component that measures the distance of y from the True Regression Line.

2.1.4.3 K-Nearest Neighbor

To predict an outcome when given features, a K-Nearest Neighbor regression model will estimate the answer based on the k most similar entries from the training data. It will work out the most similar entries by calculating the distance of the training features to the features being input. The average of the training data will be the outcome estimate for this model.

2.1.4.4 Linear SVR

Instead of minimising the error of prediction as with Linear Regression, Linear Support Vector Machine (SVM) Regression, also known as Linear SVR, attempts to keep the error of prediction within a certain boundary. Features that fall out of this boundary will not be utilised. The fitting line for future predictions will fall in the middle of this area.

2.2 Requirements

Based on the aforementioned design decisions, the requirements were made for the experiment.

- The experiment should determine whether the volume of tweets on a given day n can predict the volatility of the stock's next day $n + 1$, which is the claim.

- The volume of tweets on a given day n should be split up into multiple bins of time (i.e. how many tweets in each hour).
- The stocks analysed should be from the S&P100 as of 2018/10/29.
- Supervised machine learning should be used to test the claim.
- The baseline features for a stock should consist of its mean and standard deviation of each 65 minute trading period during a trading day. There will be 12 features, two for each 65 minute period of the trading day.
- The baseline targets for a stock should be its next day's volatility.
- The baseline dataset for a stock should consist of the baseline features and targets for that stock.
- The tweets features for a stock should be each hour of the day's volume of tweets for the given stock. There will be 24 features, one for each hour of the day.
- The tweets dataset for a stock should consist of its tweets features as well as its baseline features. The targets are the same as the baseline's.
- Pearson correlation should be calculated for the stocks between each tweet volume feature and the target to ensure that some correlation exists.
- Parameters should be their defaults, unless the nature of the experiment dictates otherwise.
- The experiment should test four types of regression to see which is the most effective at limiting the error in predictions.
- The four types of regression are: Random Forest, Linear Regression, K-Nearest Neighbor, and Linear SVR.
- The effectiveness measure of these models should be Mean Average Percentage Error (MAPE).
- The difference between each stock's baseline's MAPE and tweets MAPE should be calculated for each regressor to identify the predictive power of

tweet volume on stock volatility.

2.3 Research Tools and Resources

I used various languages, libraries and other tools in this project. Some of the most prominent and noteworthy of the project will be briefly touched on here.

2.3.1 Python

`Python 3.7` was chosen as this project's scripting language as I have some experience with it in the past. In addition, the `twitterscraper` library is made for Python and it is a language that has powerful additions like the `Pandas` and `NumPy` libraries for analysing data, as well as the `scikit-learn` library for machine learning.

2.3.2 Pandas

The `Pandas` library is a Python library of simple, yet powerful data structures and data analysis tools (McKinney et al., 2010). I extensively used the `Pandas dataframe` to hold and manipulate data in memory, such as in the engineering of features. Furthermore, `Pandas dataframes` are able to be given as arguments to the `scikit-learn` models as both features and targets.

2.3.3 Twitterscraper

`Twitterscraper` is "a simple script to scrape for Tweets using the Python package requests to retrieve the content and Beautifullsoup4 to parse the retrieved content" (Taspinar, 2018). It was employed to collect tweets containing *cashtags* of stocks in the S&P100 index in the given period.

2.3.4 Alpha Vantage and IEX Finance

Initially, I used the Alpha Vantage API (Alpha Vantage, 2019) to collect minute-by-minute data about the stocks in the S&P100. However, this API has a limited history and after a break in data collection it became necessary to get older data than AlphaVantage could provide. At this point, the IEX Finance API was used (Lynch, 2019) via the IEX Finance library.

2.3.5 MongoDB

To store the tweets and stocks a database became necessary. After briefly researching databases, I settled on MongoDB as it is widely supported, simple to use and interrogate, and has high performance and automatic scaling (MongoDB, 2018). In addition, Python has the `PyMongo` library which made using MongoDB even more seamless. Furthermore, the data I receive from the API and by scraping is in a *JSON* format which is a very similar format to MongoDB's documents, this makes it easier to use than alternatives like MariaDB.

2.4 Management Tools

Some tools were also used in the project management of this experiment. They included the 'GitHub' version control system, 'Conda' environment manager and 'Trello' task tracker.

2.4.1 Version Control System

As a "code hosting platform for version control and collaboration, [GitHub] lets you and others work together on projects from anywhere" (GitHub, 2018). In addition, I also used my GitHub repository as a cloud hosting service for my code in case of local loss. In addition, I have shared my private project repository with my project supervisor to allow oversight and monitoring of progress. GitHub provides students with free premium accounts which I have taken advantage of. This is

another reason why I chose GitHub, because it allows me to keep all my projects in one location, as opposed to using alternatives like GitLab.

2.4.2 Environment Manager

To isolate my project from other projects on my system, I am using Conda as an environment manager (Conda, 2018). This avoids conflicts in my versioning and dependencies with other projects on my systems. In addition, it is also used to create the `environment.yml` from the supporting materials necessary for the scripts to run in.

2.4.3 Organisational Tools

2.4.3.1 Trello

Trello is a *webapp* project management tool that allows me to record and track my tasks as they are completed. I use Trello as the service that hosts my Kanban board. A Kanban board is a way to visualise the workload (Benson and Barry, 2011). This tool is free and allows me to keep by the agile Kanban principle of limiting my concurrent work in progress to ensure productivity.

3 Method

The methodology of the research explains the process used in collecting and cleaning the data, engineering features and targets for machine learning, calculating the correlation between features and targets, and machine learning training and testing. The method is summarised in a schematic at the end of this section Fig. 2.

3.1 Data Collection

The necessary data required for this project were the tweets and stock close prices. Any tweets where a *cashtag* for a stock in the S&P100 index was used, within the period under examination, needed to be captured and stored. In addition, I wanted as much detail about the stock pricing as possible and therefore I needed access to the minute-by-minute value of each stock. Finally, I needed to store this in a 'MongoDB' database.

3.1.1 Tweets

The tweets needed to be collected somehow from the social media platform 'Twitter'. There are two main ways to retrieve data from Twitter in such a way that the data can be stored in a database for future processing, using the Twitter API, and scraping web responses from the Twitter server.

To collect the tweets, I first attempted to use the Twitter API, however, despite requesting permission for access in September 2018, at the time of writing in April 2019, I have still not been reviewed and granted or denied access. This API would have given me an interface to which I could retrieve rate limited data from Twitter's database of tweets (Twitter, 2018b). Instead, I attempted to scrape the data from the Twitter website.

Twitter allows you to carry out advanced search functions (2019), including searching for a *cashtag* between specified dates on a web page, before retrieving the tweets from their database and placing them on a web page to be read. It is technically

possible to manually collect tweets this way using the advanced search web page and copy and pasting the tweets into a database. Because it is possible to do this manually, it is also possible to do this in an automated fashion using a scraper. There was no point in writing my own HTML parser for Twitter, because a library had already been created which did that, called `twitterscraper` (Taspinar, 2018).

I wrote a script, called `tweetgetter.py`, found in the supporting material, that would run daily using the `twitterscraper` library to get the tweets for each stock in the S&P100, store them in the local MongoDB database and alert me via email when it was done.

To do this I would only run the *tweetgetter* on days in the past up until the last completed day (i.e. yesterday from the day the script was run). This was done by finding the date of the last tweet in the database, and setting this as the starting day while setting the finish date to be yesterday (from runtime). It is important to note that the tweets were collected with the timezone of London (i.e. BST or GMT). These would later need to be converted to Eastern Time to match the index's time zone.

In order to prevent potentially limiting myself in future, I captured a wide variety of the metadata of a tweet including the timestamp, username, full name, tweet text, number of replies, number of retweets and number of likes at the time of capture. Most of this was not used in this project, as I was only interested in the volume of tweets.

This was then set to run daily on a laptop that was left powered on to scrape Twitter and the stock market for the duration of the period. This meant that it was important that this script would not crash, I did not check the system every day, however if a script did not complete, I would not receive the script completion email. This did occur multiple times, usually due to the internet connection failing. In these events I did not have to do anything as the script was built to run again the next day. The visualisation of the average tweets per day captured can be seen in Fig. 1

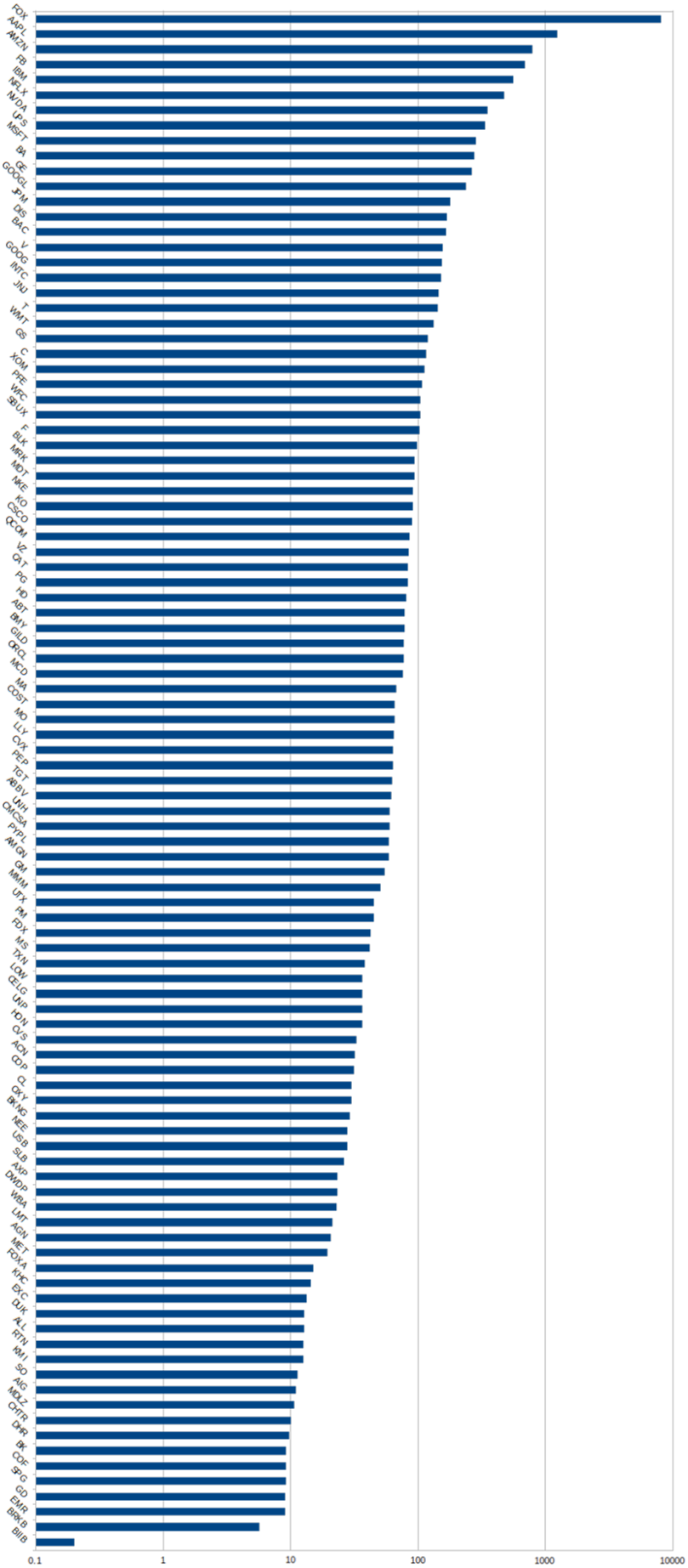

Figure 1: Average volume of tweets per day by stock ticker. Volume axis is logarithmic.

3.1.2 Stocks

To collect the stock data for every stock in the S&P100 index, I researched the various API offerings for stocks being sold on the U.S. exchanges and at the time of research in October 2018, Alpha Vantage appeared to be the best free API that would provide the minute-by-minute 'close' prices of the index's underlying stocks. A close price is the cost a share is trading at a given time i.e. at the end of a minute or end of a day.

I wrote a script called `stockgetter.py`, found in the supporting material, that would use this API to download the minute-by-minute closes of the stocks in the index. The Alpha Vantage database that this API accessed only stores the stock data for about a few days. Therefore, this script would need to be run often in order to ensure that stock data was not missed.

The API allowed 6 calls a minute, therefore I included a sleep timer in the script to ensure that this rate was never exceeded. The script would run nightly and would go through the list of stocks and for each one, would get all the minute-by-minute data in the Alpha Vantage database. It would then compare this data with with the data held in my local MongoDB database and if anything was missing from my database (i.e. anything new), then it would insert it as a new document. For each minute of the stock market, I collected the close and the volume of trades in that minute. The volume of trades was extra data that was not used in this project.

The Alpha Vantage database was not consistent and sometimes had missing minutes. Sometimes a minute's data would not be in the Alpha Vantage database, and later would be – which is why I needed to continuously compare the databases for maximum data collection.

In January 2019, I stopped the running of the *tweetgetter* and the *stockgetter* because I believed I had enough data with millions of tweets overall and planned to start the analysis. However, I had failed to realise that these tweets would be summarised as volumes of tweets and therefore it was the number of days that counted and not the number of tweets. When I realised that I did not have

enough data, it was necessary to obtain more days worth of tweets and stock closes. However, Alpha Vantage had already scrubbed the necessary dates from its backlog.

Luckily, another minute-by-minute stock API had been released in the months after I had done my initial research that held stock data that goes back three months historically. This API is from IEX Finance and it even came with a Python library. The script's process was very similar to the original *stockgetter* however, IEX Finance referred to classes of shares slightly different than my list of S&P100 shares. For example, Berkshire Hathaway Class B's ticker is normally '$BRKB', however in IEX Finance it is referred to as '$BRK.B'. This was trivial to solve as I would simply try the proper ticker and if that failed I would insert a dot before the final character. Nevertheless, IEX Finance was not perfect, and some stocks minutes were lost, similar to Alpha Vantage, but also some days were completely missing too. Once the stocks had been collected, I stored each minute for each stock in a new database in the MongoDB. This would later be merged with the original database to form one source of truth of a stock's values. All these problems and inconsistencies had to be accounted for in the cleaning and feature engineering steps.

3.2 Cleaning Data

Once the data was collected, preliminary analysis found inconsistencies making it increasingly apparent that it would need to be cleaned to ensure that it could be properly processed. Data cleaning "deals with detecting and removing errors and inconsistencies from data in order to improve the quality of data" (Rahm and Do, 2000). An example of issues with the data were the difference in timezone between the tweets and the stocks timestamps. This is vital to fix because we can only analyse tweets before the period in which volatility is examined - otherwise the entire experiment would be flawed.

It was discovered that some stocks received more tweets than others. For example, the Biogen *cashtag* ($BIIB) was only used 29 times in tweets in the period

examined. This averaged out to 0.2 tweets per day. Although the purpose of this project is not to understand how many tweets about a company is necessary before the volatility can be predicted, it was believed that 0.2 tweets a day was not significant enough and therefore, Biogen was removed from the index examined.

The `get` function in `functions.py` creates two Comma Separated Values (CSV) files which can be later imported to Pandas dataframes for analysis McKinney et al. (2010). The CSV files were also manually examined in spreadsheet software for quick reference without the need for writing a script. The two files per stock were the databases of stocks and tweets called `sdb.csv` and `tdb.csv` respectively. This process merged the two separate stocks databases into the one `sdb.csv` file for each stock.

For the tweets, the timezone was converted from London time to Eastern time, matching the stock's timezone.

For the stocks, the cleaning process made the timestamp the table index of a stock entry in the sdb and sorted on this element to ensure that the minutes were in the correct order. The volume of trades was not needed and therefore was removed.

3.3 Feature Engineering

First a baseline dataset had to be made from the stocks. To do this, a day n was split into 6 periods of 65 minutes. This was because a trading day is from 09:30–16:00 (390 minutes), not divisible into hours. Instead, I elected to split it into a period close to an hour that could divide 390 without a remainder, which led me to settle on 65 minutes. For each 65 minute period, the mean and standard deviation of the minute-by-minute closes was calculated. This meant that even if a minute was missed out due to issues with the underlying data source (Alpha Vantage or IEX Finance), then that would be averaged out. Also, it meant that close values could be succinctly summarised into fewer features.

The volatility was calculated for day $n+1$ by taking all the closes that trading day and finding their standard deviation. This was then added to the datasets as a

new column. This column was shifted back one row, so that the entry for day n had the volatility of day *n+1*. Therefore a row in the baseline would appear as:

$$\text{row}_n = m(p_0), \sigma(p_0), m(p_1), \sigma(p_1), ..., m(p_6), \sigma(p_6), \sigma(d_{n+1})$$

Where p_0 and p_1 are the closes of the first 65 minute period and second period respectively, d_x is the closes of day x, m is the mean, and σ is the standard deviation.

Once this was made the second dataset needed to be engineered so that it contained the hourly volume of tweets, as well as the same data in the baseline. The method to do this was to take the `tdb.csv` for a stock and count the number of tweets in each hour. Therefore a row would look like this:

$$\text{row}_n = t(h_0), t(h_1), ..., t(h_{23})$$

Where h_0 and h_1 is the tweets from 00:00:00–00:59:59 and 01:00:00–01:59:59 respectively, and t is the volume of tweets.

After this row was created, the baseline dataset was appended to the end of the dataset so that it looked like this:

$$\text{row}_n = t(h_0), t(h_1), ..., t(h_{23}), m(p_0), \sigma(p_0), m(p_1), \sigma(p_1), ..., m(p_6), \sigma(p_6), \sigma(d_{n+1})$$

For the period examined, I created a preset list of U.S. public holidays and half public holidays, including the sudden national day of mourning for George H.W. Bush's death. The holidays were: Thanksgiving, Black Friday (markets close at 13:00), George H.W. Bush's day of mourning, Christmas Eve (market closes at 13:00), Christmas Day, New Years Day, Martin Luther King's birthday and Presidents' Day. The days that had the target of these day's volatility needed to be removed from the dataset of features and targets for both the baseline dataset and the tweets dataset. If a target was discovered missing, for example if the API database did not have data for that day for whatever reason, then the entire entry was deemed irrelevant and was removed.

After this process, the two features datasets per stock were ready with the features and labels (targets) saved in new CSVs called `stocks ftlbl.csv` and `tweets ftlbl.csv`. Henceforth, the baseline dataset will be referred to as the baseline

or the baseline features and the tweets dataset will be referred to as the tweets features.

3.4 Correlation

To identify the relationship between the baseline features and targets, and tweets features and targets, I have used 'Pearson' correlation. For the purposes of the correlation only, the baseline was not added to the tweets features.

I used the `scipy` stats module which has the `pearsonr` function (Jones et al., 2019). The parameters that the function needs are the feature for each date, and the target for each date. This correlation is calculated for each feature in the datasets, so 12 times for the baseline and 24 times for the tweets.

The `pearsonr` function returns the Pearson correlation coefficient and the 2 tailed p-value. Although this calculation will be performed for all the stocks datasets, for the results in this report, I will be summarise them by selecting a handful of stocks. I will choose the most valuable stock as well as the stocks with the highest, median and lowest absolute mean correlation coefficients as examples.

3.5 Machine Learning Training and Testing

The machine learning training process will involve giving four different machine learning regressors the baseline features and targets and the tweets features and targets. For each of these I will calculate the Mean Average Percentage Error (MAPE). This will then allow me to find the difference between the baseline and the tweets. If there is an improvement, this could indicate that the volume of tweets do indeed hold some predictive power for the next day's stock volatility. If there is a degradation and deterioration from the baseline or the difference is not significant or consistent, then this may indicate that the volume of tweets do not hold some predictive power for the next day's stock volatility.

I employed the `scikitlearn` library for the machine learning functionality (Pedregosa et al., 2011). Each time a regressor was used on a dataset, I would split the

data into a training data and testing data. By default, the testing data size was 0.1. All of the `scikitlearn` defaults were used and for Random Forest regression, the number of estimators was set at 1000.

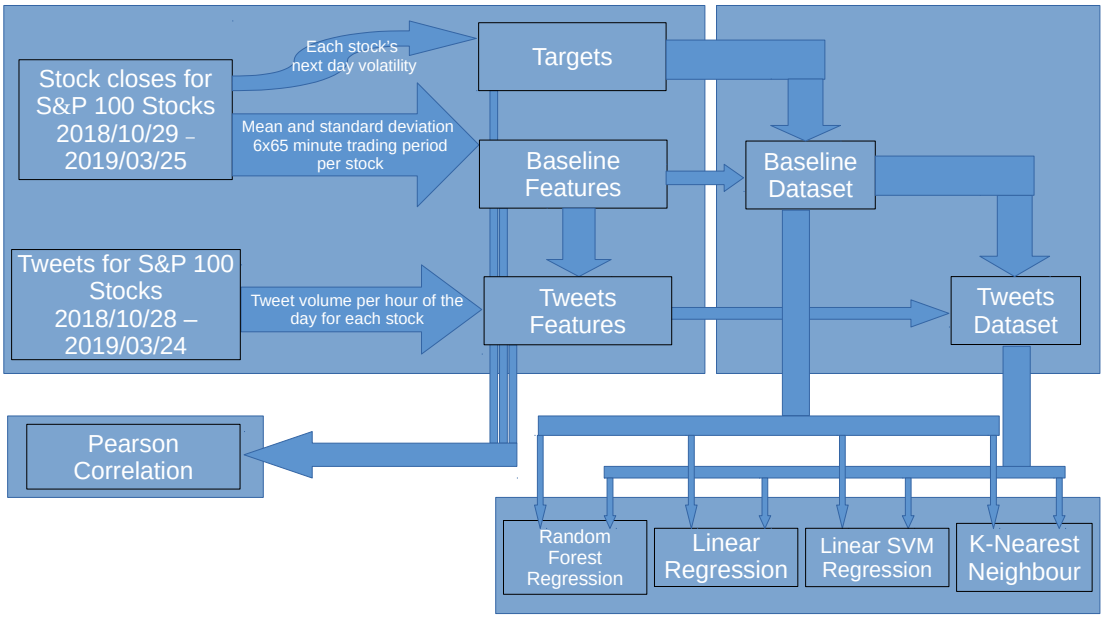

Figure 2: Schematic of the method

The process I used was to split the dataset into training features, testing features, training targets, and testing targets. This way, I could use the training data to fit the model and test out its effectiveness on the testing data. If I trained and tested the model on the same data without a split, then I would not be able to test out the model on any new data to find its effectiveness. Because I was limited in the amount of data that I had, I set the testing size to be smaller than `scikitlearn`'s default.

In addition, I adjusted the K-Nearest Neighbors default number of neighbors down to 3 from the default of 5. This was due to the dataset being limited in size. Although parameter tuning, as already mentioned, was not the point of this experiment, for Random Forest regression, I did try several different values for the `n_estimators` that the model used. I found little difference in the results, unless the value was particularly low. Therefore, I opted to choose the value of 1000 as

it is acceptable in terms of results, computational power and time.

Once a model was trained with a stock's baseline or tweets training data, I could then use the model to predict the targets of the testing data by giving the model the testing features. Once the model had done this it was possible to calculate the MAPE using the true targets and the predictions.

This was performed for each stock's baseline and tweets data for each of the models. The difference between the two MAPEs were calculated for each stock and regressor. For this report, I will include the complete results as well as the mean and median differences for each regressor before evaluating each regressors performance.

4 Results

In this analysis, I will report on both the results of the correlation and machine learning models. Firstly, I will record the extent to which the volume of tweets about a stock on day n are correlated with the stock market volatility of day $n+1$. Secondly, I will note the extent that the machine learning models can use the the volume of tweets about a stock on day n to predict its stock market volatility on day $n+1$. Finally, I will compare and discuss the correlation and the model results.

4.1 Correlation

For the volume of tweets to have any predictive power on the next day's volatility, there should exist some kind of correlation between them. To assess this, I used 'Pearson' correlation which is a measure of the linear correlation between two datasets. With Pearson, a coefficient value of 0 means that there is no correlation i.e. the two datasets are uncorrelated, while a coefficient of $+1$ or -1 means a direct linear positive or negative relationship between the two datasets.

Specifically, the Pearson Correlation Coefficient PCC is calculated as:

$$PCC = \frac{\sum_{i=1}^{n}(x_i - m_x)(y_i - m_y)}{\sqrt{\sum_{i=1}^{n}(x_i - m_x)^2 \sum_{i=1}^{n}(y_i - m_y)^2}} \tag{2}$$

where x is the feature and y is the target stock's next day's volatility. m_x and m_y is the mean of the feature and the mean of the next day's volatility respectively (Britain), 1895).

The p–value is the probability that the correlation is at least the computed coefficient (it could be higher) if the null hypothesis were true (i.e. the tweets and targets are uncorrelated). It is the probability that due to limited samples, the coefficient was calculated as it was, despite there being no correlation. Therefore, the lower the p-value, the higher the probability that the identified correlation is

correct. The convention is that if the p–value is less than 0.05, then it is considered statistically significant. In effect, the p–value explains the significance of the calculated correlation. Although, it is prudent to notice that the p–values cannot be considered completely reliable as the dataset is not large enough - it would need to have at least 500 samples (days) or so. (Jones et al., 2019).

As Pearson correlation is only bivariate, it is not possible to calculate the correlation between multiple bins of tweets (i.e. a whole day of bins) to the next day's volatility. Therefore, I have assessed each bin of tweets (the tweets of each hour of the day) individually against the volatility. Further, I have selected a representative handful of stocks to analyse in these results including the most valuable stock in the S&P100, Apple ($AAPL), the stock with the highest absolute mean correlation, Booking Holdings ($BKNG), the stock with the median absolute mean correlation, Mastercard ($MA) and the stock with the lowest absolute mean correlation, Southern Company ($SO). The results can be found in Table 2.

	Apple		Booking Holdings		Mastercard		Southern Company	
hour	PCC	p–value	PCC	p–value	PCC	p–value	PCC	p–value
0	0.102632681	0.317138526	0.05566595	0.590103519	-0.115243273	0.260991607	0.017344733	0.866092438
1	0.055346654	0.590267921	-0.081763772	0.428389199	0.055237337	0.591003306	-0.020162964	0.844588463
2	0.0275799	0.788576012	-0.043443331	0.674281938	-0.094585918	0.356760716	-0.040818805	0.691387742
3	0.17706563	0.082736466	0.125869052	0.221718473	0.052890122	0.606892128	0.117640454	0.251136188
4	0.131489884	0.199205279	-0.026814815	0.795376438	-0.068082891	0.507576813	-0.267864693	0.007986932
5	-0.09103286	0.375191963	0.440237558	0.000007177	-0.092862928	0.365627407	0.059530011	0.562443406
6	0.035300225	0.731397698	0.135974677	0.186512518	-0.043999467	0.668699092	0.034478242	0.737422234
7	0.117977709	0.249770435	-0.059355172	0.565664527	0.121093765	0.237393008	-0.079701628	0.437731519
8	0.147157963	0.15032262	-0.045679187	0.658539315	-0.055369058	0.59011726	-0.11408395	0.265851073
9	0.115922366	0.258173345	0.098716677	0.338621956	-0.045400578	0.658796752	-0.13225986	0.196563833
10	0.100645932	0.32664664	0.172831075	0.092199768	0.039151257	0.703394343	0.010038464	0.922259321
11	0.22007145	0.030312471	0.158824015	0.122202535	-0.042995232	0.675831695	-0.090713763	0.376875111
12	0.245291273	0.015450653	0.010716599	0.91746433	-0.108673902	0.289335495	-0.013495704	0.895616986
13	0.258885829	0.010452579	0.16850392	0.100771514	-0.048584283	0.636515832	0.045011503	0.661540729
14	0.363338398	0.000254269	0.196231946	0.055344641	0.164101919	0.108235848	-0.024375211	0.812663747
15	0.345775373	0.00052241	0.255124746	0.012120472	-0.057222415	0.577714707	0.185426233	0.069008305
16	0.191312427	0.060496041	0.683079679	0.000000001	-0.081349732	0.428285146	-0.146088164	0.153343103
17	0.175996814	0.0846385	0.65864488	0.000000001	0.008019385	0.937860565	0.012724331	0.901552408
18	0.174532791	0.08730015	0.681625243	0.000000001	0.005882797	0.954395262	0.023415469	0.819912745
19	0.160631591	0.116007699	0.456677519	0.000002915	-0.081203982	0.429115802	0.138167338	0.17713387
20	0.127030638	0.215005787	0.410350103	0.000032909	0.139439921	0.173139298	-0.054585412	0.595397426
21	0.148648643	0.146188557	0.572319034	0.000000001	0.13236348	0.196210297	0.001363147	0.989427222
22	0.113745017	0.267283264	0.374613338	0.000169738	0.054016155	0.599246297	0.127516405	0.21324266
23	0.141811669	0.165872024	0.442413513	0.000006388	0.065227316	0.525582987	-0.001820717	0.985878552

Table 2: The chosen stock's Pearson Correlation Coefficient (PCC) for each hour of the day (0-23) with p–value rounded to 9 decimal places

4.1.1 Apple

Apple's correlation data in Table 2 has been visualised in Fig. 3 showing the PCC for each hour of the day with the colour intensity being $1 - \text{p–value}$. It shows

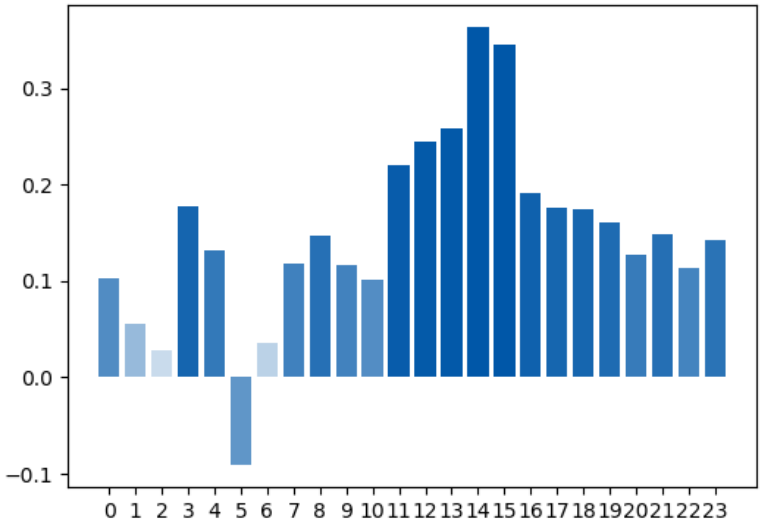

Figure 3: Apple's Pearson Correlation Coefficient for each hour of the day (0-23) with strength of colour signifying p–value (from Table 2)

greater correlation with the volume of tweets from about midday until the close of the markets at 16:00. There also seems to be a relationship with the absolute correlation coefficient and the p–value where the greater the absolute coefficient, the smaller the p–value.

4.1.2 Booking Holdings

The volumes of tweets sent after the close of the markets at 16:00, as shown in Fig. 4, appear to have the greatest correlation on the next day's volatility for Booking Holdings. The p–values of these correlations are also very strong indicating a high confidence in the correlations.

4.1.3 Mastercard

The correlation seen in Fig. 5 for Mastercard appears quite erratic. The absolute Pearson Correlation Coefficients are not particularly high, with the highest being

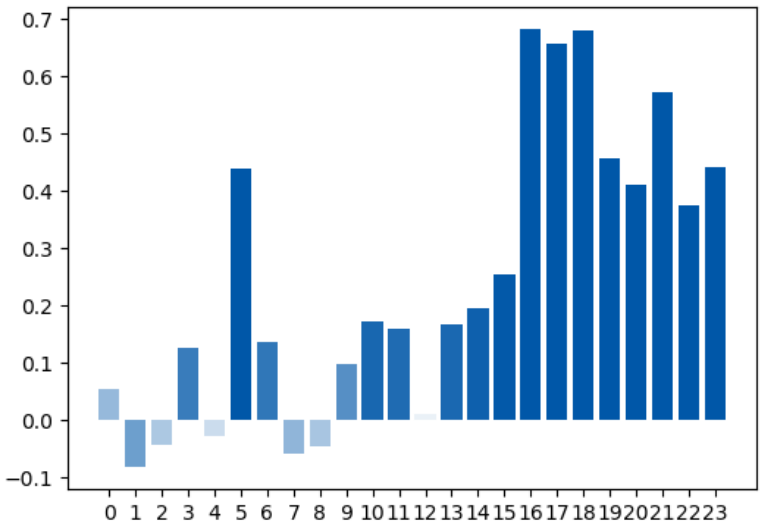

Figure 4: Booking Holding's Pearson Correlation Coefficient for each hour of the day (0-23) with strength of colour signifying p–value (from Table 2)

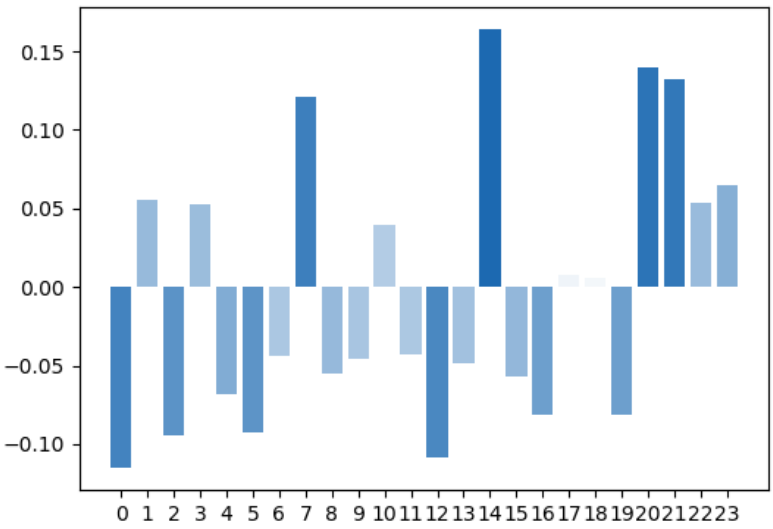

Figure 5: Mastercard's Pearson Correlation Coefficient for each hour of the day (0-23) with strength of colour signifying p–value (from Table 2)

0.16, meaning that there appears to not be a considerable amount of correlation between the volume of tweets and the next day's volatility.

4.1.4 Southern Company

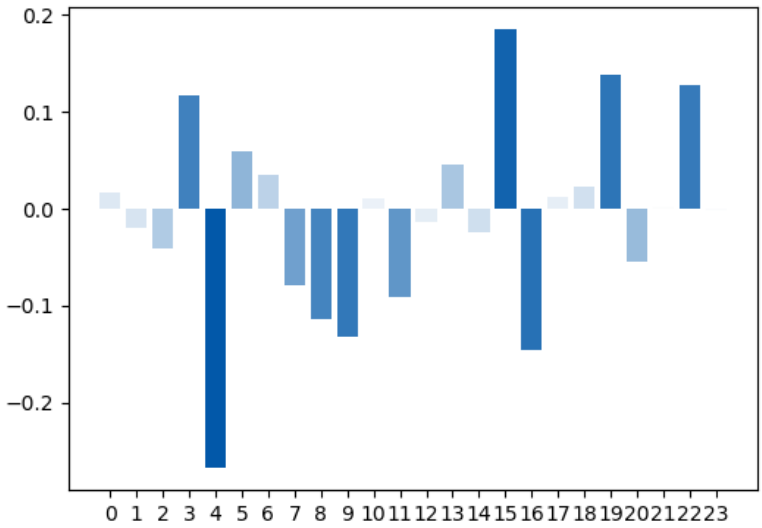

Figure 6: Southern Company's Pearson Correlation Coefficient for each hour of the day (0-23) with strength of colour signifying p–value (from Table 2)

Similar to Mastercard, Southern Company is also erratic with no clear period of correlation of tweet volumes to the next day's volatility. There are a few spikes of confident correlation, the highest having an absolute coefficient of 0.27. It is potentially interesting to note the difference between the difference in correlation from 15:00-16:00 during the market hours and from 16:00-17:00 once they have closed.

4.1.5 Discussion: Correlation of Tweet Volumes and Stock Volatility

The results of the correlation data raise interesting questions about the reason for the variability in correlating power for the different times of the day, for why there

is sometimes erratic swings in the coefficient, and why there appears to be a link between coefficient value and p-value.

- For many of the stocks, the greatest correlation appeared later in the day, with the highest around the market closing times. While it is not completely clear why this is the case, I can propose two potential reasons. Firstly, perhaps it is because the bins at the end of the day are closer to the next day, where the volatility of the stock is calculated. Secondly, perhaps traders or analysts that are busy working during the trading day become more free to tweet after the market has closed about their predictions for next day.
- Some of the stocks correlation may be incorrect due to a lack of data. For example, Southern Company ($SO) only had 11 tweets a day on average in the period examined. This may mean that the correlation is incorrect because there are not enough tweets in each bin for there to be significant enough difference between the bins to make accurate calculations from.
- If the results with high p–value (those greater than 0.05) are ignored, then we find fewer erratic results, and greater correlations. However, for transparency, I have not removed these values from the data in this report, as it is important to note that the correlation is not always consistent across the time or across the stocks.
- There appears to be a link between the coefficient and the p-value, where the higher the correlation coefficient, the lower the p-value. This may be because if one is measuring a high correlation, then the chances of it being due to an accident and the null hypothesis being true (and there being no correlation) goes down.

Overall, from the Pearson Correlation Coefficient of many of the stocks, it appears that there is some correlation between the volume of tweets and the next days volatility. This is encouraging for the machine learning component, since if there was no correlation, then it would indicate that the machine learning models would not be accurate in predicting the future volatility.

4.2 Machine Learning Model

Following the process outlined in the Method section of this report, the machine learning models were trained and run on the testing data. This section will report, analyse and discuss these results.

The differences in average error of the different fitted machine learning models between the baseline (consisting of just the mean and standard deviation of stock values in each 65 minute period of the trading day) and tweets (consisting of both the baseline and the tweet volumes for each hour in the day) datasets was varied. To calculate the differences I first needed to calculate the 'Mean Average Percentage Error' (MAPE) which is defined as:

$$MAPE = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{A_t - P_t}{A_t} \right| \tag{3}$$

where A_t is the actual results from the testing data and P_t is the predicted result from the fitted machine learning model, t is the day that is being predicted, and n is the number of days fitted by the model. I then rounded the MAPE to 2 decimal places for brevity.

This then allowed me to calculate the difference between the baseline average error and the tweets average error to show whether any improvement had been made by the machine learning model (a positive difference) or not (a negative difference).

To ascertain which model was the best, I compared the models to each other by calculating the mean and median of the differences. All of these results are listed in the appendix in Table 4. I have selected 10 stocks which had over 100 tweets on average per day and had a significant improvement with the Linear SVR model as a subset of the larger table in the appendix. This selection is in Table 3.

4.2.1 Random Forest

For 10 stocks, the Random Forest regression model had the greatest differences between the baseline features and the tweet features, meaning that for 10% of the

	Random Forest			Linear			Linear SVR			K-Nearest Neighbor		
Stocks	Baseline	Tweets	Difference	Baseline	Tweets	Difference	Baseline	Tweets	Difference	Baseline	Tweets	Difference
FB	69.31%	63.12%	6.19%	79.94%	37.86%	42.08%	184.96%	74.09%	110.87%	62.62%	40.63%	21.99%
UPS	58.92%	58.93%	-0.01%	58.86%	56.07%	2.79%	142.86%	50.25%	92.61%	64.25%	68.1%	-3.85%
JNJ	99.53%	103.91%	-4.38%	57.61%	86.06%	-28.45%	208.96%	125.24%	83.72%	73.57%	114.78%	-41.21%
AAPL	42.01%	40.86%	1.15%	57.56%	70.57%	-13.01%	249.18%	165.78%	83.4%	61.99%	45.24%	16.75%
BAC	45.17%	38.69%	6.48%	38.65%	48.65%	-10%	163.58%	80.34%	83.24%	50.58%	43.47%	7.11%
JPM	63.25%	57.01%	6.24%	64.78%	69.67%	-4.89%	155.17%	92.29%	62.88%	70.24%	48.71%	21.53%
XOM	50.82%	47.29%	3.53%	59.61%	59.35%	0.26%	166.2%	104.37%	61.83%	49%	47.14%	1.86%
INTC	46.95%	53.96%	-7.01%	48.1%	53.3%	-5.2%	107.98%	49.15%	58.83%	42.89%	50.45%	-7.56%
V	43.69%	34.69%	9%	35.08%	32.62%	2.46%	159.79%	114.43%	45.36%	44.4%	40.12%	4.28%
PFE	59.12%	51.66%	7.46%	59.42%	57.54%	1.88%	106.72%	64.55%	42.17%	68.84%	43.57%	25.27%

Table 3: MAPE of machine learning models on baseline and tweets datasets and calculated difference (improvement) sorted descending on 'Linear SVR Difference' for the specified subset of stocks

stocks analysed, the Random Forest regressor performed the best. However, half of these greatest differences were still negative, meaning that despite the Random Forest regressor performing the best for these 5 stocks, the baseline was still a better predictor than the model. The stock whose most effective regressor still led to the worst result when compared to the other greatest results was Burger King ($BK) using Random Forest regression. The average improvement was 0.29% and the median improvement was 0.085% – only very slight increases.

The distribution of differences between the baseline Random Forest regression model and the tweets features Random Forest regression model for each stock is depicted in Fig. 7.

4.2.2 Linear Regression

Linear Regression had the worst performance of the four models with only 3 stocks out of the 100 analysed finding their highest average error improvement with this model. For these 3 stocks, all of the differences between their baseline and tweets features fitted models were positive, meaning that Linear Regression improved the average error over the baseline for them. Despite this, for the majority of stocks, Linear Regression did not improve from the model using baseline features. The average deterioration in improvement using the tweets features compared to the baseline features using linear regression was 16.268% and the median deterioration was 11.625%. No other model deteriorated on average at all, let alone to this extent.

The distribution of differences between the baseline Linear Regression model and the tweets features Linear Regression model for each stock is depicted in Fig. 8. It can clearly be seen from the left-sided skew on this graph that the model is poor on average.

4.2.3 K-Nearest Neighbors

Regression using the K-Nearest Neighbors model performed reasonably well. Of the stocks analysed, 29 found it to be the most effective model at improving over the baseline features using the tweets features. Moreover, none of these results were negative, meaning the tweets features only improved on the baseline features using this regression model. On average, the model found a 1.062% improvement from the baseline to the tweets features average error of predicting power and a median improvement of average error of 1.51%.

The distribution of differences between the baseline K-Nearest Neighbors Regression model and the tweets features K-Nearest Neighbors Regression model for each stock is depicted in Fig. 9.

4.2.4 Linear SVR

Linear SVR was the best performing of the four tested regressors. Of the stocks analysed, 58 found it to lead to the greatest decrease in prediction average error from the baseline to the tweets features. Two of these stocks were still negative differences, meaning that the baseline was better than all of the regressors predictive power – despite Linear SVR performing the best out of the four models. The greatest improvement in average error was seen by Linear SVR on Celgene ($CELG) with an improved average error of 191% from the baseline fitted model to the tweets fitted model. The average improvement was 18.355% and the median improvement was 10.625%.

The distribution of differences between the baseline Linear SVR model and the tweets features Linear SVR model for each stock is depicted in Fig. 10.

Since Linear SVR was the best performing regressor, I wanted to delve in further to the model to see how the training was better for the tweets over the baseline. I took Apple ($AAPL) as an example stock and examined the training curve for training score and cross-validation testing score. If the testing score is performing well, it will converge onto the training curve the more training examples there are. The baseline Linear SVR training curve for Apple can be seen in Fig. 11 and the tweets Linear SVR training curve for Apple can be seen in Fig. 12, where the convergence is clearly better.

4.2.5 Discussion: Predictability of Stock Volatility from Tweet Volume

On average, all the machine learning models performed better than the baseline (positive difference) except for Linear Regression. For Linear Regression, both the average and median differences between the model using the baseline features

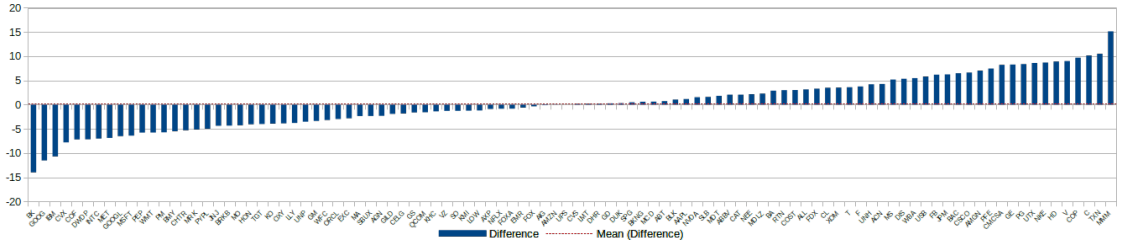

Figure 7: MAPE difference between baseline and tweets features trained Random Forest models vs Stock with a mean line

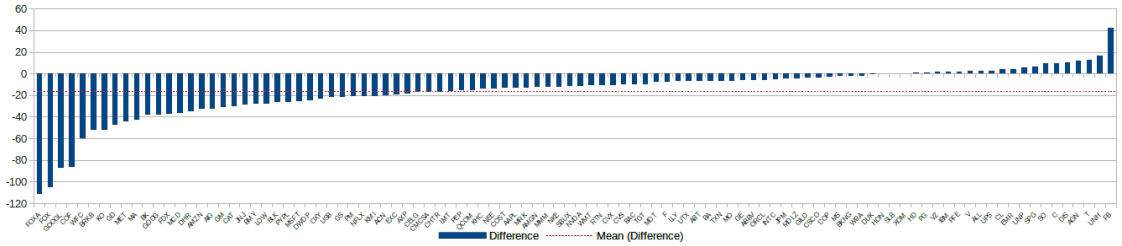

Figure 8: MAPE difference between baseline and tweets features trained Linear Regression models vs Stock with a mean line

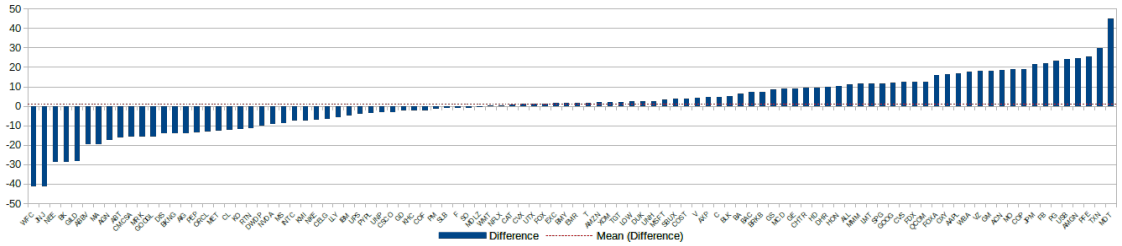

Figure 9: MAPE difference between baseline and tweets features trained K-Nearest Neighbors models vs Stock with a mean line

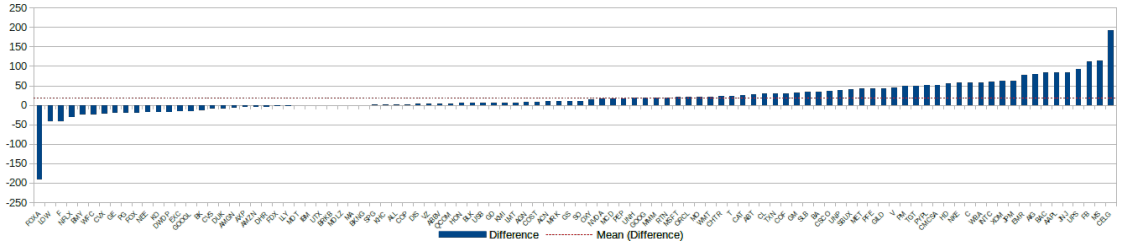

Figure 10: MAPE difference between baseline and tweets features trained Linear SVR models vs Stock with a mean line

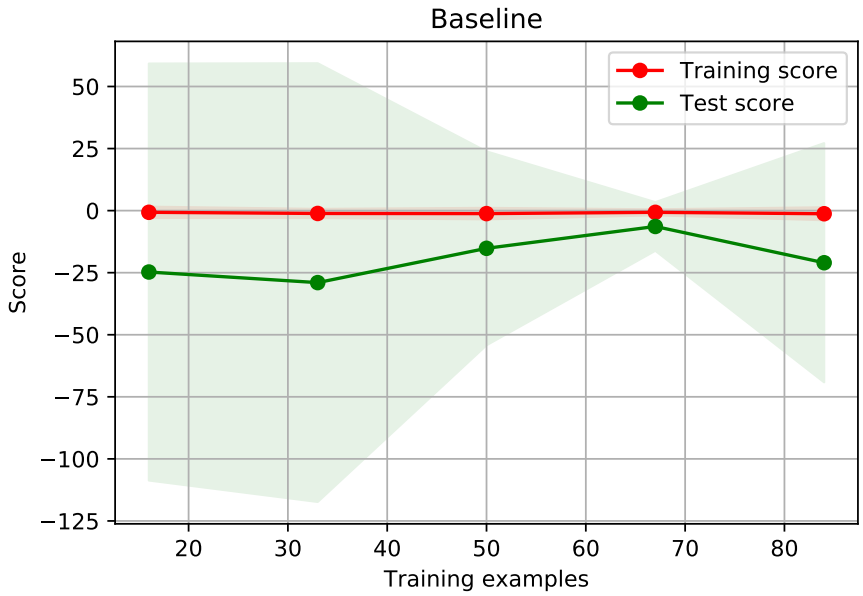

Figure 11: Baseline dataset's Linear SVR score per number of training examples for Apple

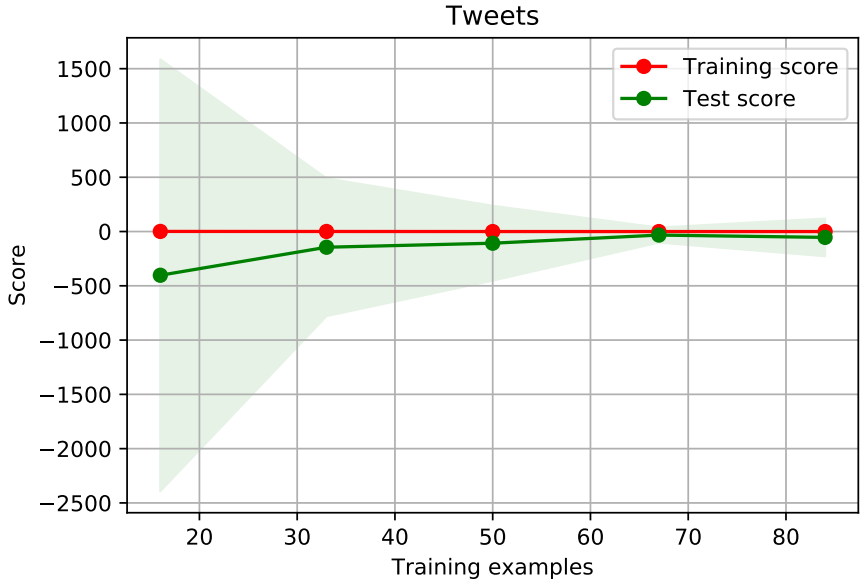

Figure 12: Tweets dataset's Linear SVR score per number of training examples for Apple

and the model using the tweets features were very negative at -16.268 and -11.625 respectively. On average, the best model was the Linear SVM regression which achieved an average decrease in average error and improvement of 18.355%.

Furthermore, there was no clear winner that outperformed all the other models all of the time. Even Linear Regression had some stocks that appeared suited to it. Nevertheless, there were 7 stocks that did not see any improvement in average error over the baseline by using tweets with any of the regression models analysed in this project.

It appears, from the Pearson Correlation Coefficient, that a stock's volume of tweets on day n does have a relationship with day $n + 1$'s volatility. In addition, it was possible to improve the predicting power of the next day's volatility using tweet volumes compared to baseline with different regressors for 93% of stocks analysed. This seems to re-affirm the comment from Ranco et al. (2015, p. 18) saying that "there is a signal worth investigating which connects social media and market behavior".

Each of the machine learning models were able to maximise the improvement in predictability from the baseline when the volume of tweets per hour was included for at least a number of stocks. Specifically, Linear SVR was the best machine learning model with the greatest improvement for 58% of the stocks over the baseline, the greatest individual increase in improvement over the baseline, and the greatest average and median improvements over the baseline.

Some of the regressors got negative results because the datasets were split so that the models were trained on one dataset and were tested on another. If the regressors were applied to the same dataset as it was trained on, the values would always be positive.

The reason why 7% of stocks did not experience an improvement over the baseline when tweets were added as features could have been due to multiple reasons. Firstly, it could be possible that there is simply not enough data from the number of days analysed. When this happens overfitting occurs, which is when the machine learning model recognises noise in the training data as a relevant signal such that

the model's performance decreases when presented with testing data (Dietterich, 1995). If this is the case, supplementary days need to be captured to average out the noise more effectively. A second reason could be because these stocks did not have enough tweets about them. For example, the stock whose difference between the baseline and tweets features average error increased the most (even with its best performing regressor) was Burger King ($BK). On average, Burger King was tagged on only 9 tweets approximately a day during the period analysed. This lack of data within the hours of the days may have resulted in a poor performance.

Moreover, the stocks that had negative results were usually negative across the board. This lends support that there is something negative about the data itself, and not an issue with the analysis of the data. Whilst the stocks that did well with the regression models, tended to do better, on average, with the other regression models.

5 Conclusion

5.1 Summary

This report has shown the extent to which the volume of tweets per hour on a given day n can predict the volatility of the stock's next day $n + 1$. I have gathered minute-by-minute stock data for the chosen index, the S&P100, as well as the tweets that used the stock's *cashtag* between 2018/10/28 and 2019/03/25.

After cleaning and engineering this data into features and targets, I calculated the correlation between the volume of tweets per hour and the target i.e. the next day's volatility. Despite some of the stocks' correlation not being completely reliable because of a lack of data, many of the stocks had the greatest correlation occurring later in the day, with the highest often occurring around the market closing times. This could potentially be because the bins of tweets at the end of the day are closer to the next day – where the volatility of the stock is calculated – or perhaps traders that are busy working during the trading day become more free to tweet after the market close.

Since it appeared from the correlation analysis that for many of the stocks, that there exists some correlation between the volume of tweets and the next days volatility, I proceeded to perform machine learning on the baseline (without volume of tweets features) and tweets (with volume of tweets features) datasets.

This analysis revealed that with the exception of Linear Regression, the machine learning models performed better than the baseline with a decrease to the Mean Average Percentage Error (MAPE) – an improvement. On average, the best model was the Linear SVM regression which achieved an average improvement to the MAPE by about 18%.

Linear SVR found the greatest improvement for 58% of the stocks over the baseline as well as the greatest individual increase in improvement over the baseline for a stock and the greatest average and median improvements over the baseline. This made Linear SVR the best performing of the models.

Each of the models utilised in this experiment, even Linear Regression, improved the MAPE of at least some of the stocks. However, there were 7 stocks that did not record an improvement to the MAPE from the baseline to the tweets with any of the regression models analysed in this project. This could be because due to the relatively few days analysed, there was not enough data causing overfitting. Alternatively, another reason could be because the stocks did not have enough people using the *cashtag* in tweets and therefore there is not enough for the models to use.

It was possible to use the machine learning models to improve the MAPE of the other 93% of stocks analysed by including the volume of tweets as features. The stocks that had poorer results were usually poorer across the board, while the stocks that do well, on average tend to perform better with the other models too. This lends credence to the claim that, at least in some analyses, the volume of tweets can help predict the volatility of the stock market.

5.2 Further Investigation

The area covered in this report, the hourly tweets from day n and its affect on the day $n + 1$'s volatility whilst promising, is a narrow examination of the larger topic area of tweets and volatility. Further research is needed on this topic to reach greater understanding of the effects of tweets and stock market volatility. Indeed, there are a multitude of experiments that are needed before the effect can be confirmed or denied.

Firstly, data from more trading days should be gathered. This will allow a larger study to be more confident in their results with a larger sample size and greater power. It is arguable that this research had too few samples for a statistically significant conclusion. In addition, in the event that accidental overfitting did occur with some stocks in this experiment, larger sample sizes will help mitigate it.

Secondly, this report, similar to other research has focused on tagged tweets that identify the stocks that the Twitter user is tweeting about. However, it could

be that many or even most tweets about a stock are not tagged with a *cashtag*. Therefore, a methodology should be created in order to identify all the tweets about a stock (or at least most of them) and not just those with *cashtags*.

Additionally, this project was interested in the volatility of a stock and not the direction that the stock would go in. For this reason, it was not believed that the content of the tweet was significant. This assumption should be tested, using Natural Language Processing of the tweets to ascertain whether it has any affect on volatility.

Furthermore, future research can investigate the best possible parameters for the learning of a machine learning model to be maximised. This hyperparameter optimisation will then be able to tune the models in such a way to decrease their error, leading to better results.

Another area of interest is how easy it is to game this system. If someone wants to manipulate these results, it is possible to use automated Twitter bots to artificially affect the volume of tweets. It needs to find out the extent that it is feasible carry this out, and to develop a strategy to limit the effects to the tweet volume analysis.

After this has transpired, it will be possible to test the experiment in the real world by designing a trading strategy to see whether a greater return can be achieved by trading volatility, for example on an index like the VIX (CBOE, 2019) compared to a benchmark index like the S&P500.

6 References

Alpha Vantage (2019). *API Documentation | Alpha Vantage*. URL: https://www.alphavantage.co/documentation/#intraday (visited on 20/03/2019).

Benson, Jim and Tonianne DeMaria Barry (2011). *Personal Kanban: Mapping Work, Navigating Life*. Vol. 218. Modus Cooperandi Press Seattle.

Bollen, Johan, Huina Mao and Xiaojun Zeng (2011). 'Twitter mood predicts the stock market'. In: *Journal of computational science* 2.1, pp. 1–8.

Boslaugh, Sarah (2012). *Statistics in a Nutshell, 2nd edition*. O'Reilly Media, Inc.

Breiman, Leo (2001). 'Random forests'. In: *Machine learning* 45.1, pp. 5–32.

Britain), Royal Society (Great (1895). *Proceedings of the Royal Society of London*. v. 58. Taylor & Francis.

CBOE (2019). *Vix-Index*. URL: http://www.cboe.com/vix (visited on 08/04/2019).

Chen, Hailiang et al. (2014). 'Wisdom of crowds: The value of stock opinions transmitted through social media'. In: *The Review of Financial Studies* 27.5, pp. 1367–1403.

Conda (2018). *Overview – Conda documentation*. URL: https://conda.io/docs/user-guide/overview.html (visited on 05/12/2018).

Deng, Shangkun et al. (2011). 'Combining technical analysis with sentiment analysis for stock price prediction'. English. In: *Proceedings - IEEE 9th International Conference on Dependable, Autonomic and Secure Computing, DASC 2011*, pp. 800–807. ISBN: 9780769546124. DOI: 10.1109/DASC.2011.138.

Dietterich, Tom (1995). 'Overfitting and undercomputing in machine learning'. In: *ACM computing surveys* 27.3, pp. 326–327.

Dimpfl, Thomas and Stephan Jank (2016). 'Can internet search queries help to predict stock market volatility?' In: *European Financial Management* 22.2, pp. 171–192.

GitHub (2018). *Hello World – GitHub Guides*. URL: https://guides.github.com/activities/hello-world/ (visited on 05/12/2018).

iPath (2019). *iPath Exchange-Traded Notes (ETNs)*. URL: http://www.ipathetn.com/US/16/en/details.app?instrumentId=341408 (visited on 08/04/2019).

iShares (2018). *iShares S&P 100 ETF | OEF | US Class*. URL: https://www.ishares.com/us/products/239723/ishares-sp-100-etf (visited on 29/10/2018).

Jones, Eric, Travis Oliphant, Pearu Peterson et al. (2019). *SciPy: Open source scientific tools for Python, version 1.2.1*. URL: https://docs.scipy.org/doc/scipy/reference/index.html (visited on 08/04/2019).

Li, Ting, Jan van Dalen and Pieter Jan van Rees (2018). 'More than just noise? Examining the information content of stock microblogs on financial markets'. In: *Journal of Information Technology* 33.1, pp. 50–69.

Lynch, Addison (2019). *GitHub - addisonlynch/iexfinance: Python SDK for IEX Cloud and the Legacy Version 1.0 Investor's Exchange (IEX) Developer API*. URL: https://github.com/addisonlynch/iexfinance (visited on 20/03/2019).

Mao, Huina, Scott Counts and Johan Bollen (2011). 'Predicting financial markets: Comparing survey, news, Twitter and search engine data'. In: *arXiv preprint arXiv:1112.1051*.

McKinney, Wes et al. (2010). 'Data structures for statistical computing in python'. In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. Austin, TX, pp. 51–56.

MongoDB (2018). *Introduction to MongoDB – MongoDB Manual*. URL: https://docs.mongodb.com/manual/introduction/ (visited on 05/12/2018).

Nair, Dinesh, Nishant Kumar and Manuel Baigorri (2017). 'Hedge Funds Track J&J Private Jet for Edge on Actelion Score'. In: *Bloomberg Quint*. URL: https://www.bloombergquint.com/business/hedge-funds-track-j-j-private-jet-for-an-edge-on-actelion-score (visited on 12/11/2018).

Nofer, Michael and Oliver Hinz (2015). 'Using Twitter to predict the stock market'. In: *Business & Information Systems Engineering* 57.4, pp. 229–242.

Oliveira, Nuno, Paulo Cortez and Nelson Areal (2017). 'The impact of microblogging data for stock market prediction: Using Twitter to predict returns, volatility, trading volume and survey sentiment indices'. In: *Expert Systems with Applications* 73, pp. 125–144.

Pedregosa, F. et al. (2011). 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Rahm, Erhard and Hong Hai Do (2000). 'Data cleaning: Problems and current approaches'. In: *IEEE Data Eng. Bull.* 23.4, pp. 3–13.

Ranco, Gabriele et al. (2015). 'The effects of Twitter sentiment on stock price returns'. In: *PloS one* 10.9, e0138441.

Rao, Tushar and Saket Srivastava (2012). 'Analyzing Stock Market Movements Using Twitter Sentiment Analysis'. In: *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*. ASONAM '12. Washington, DC, USA: IEEE Computer Society, pp. 119–123. ISBN: 978-0-7695-4799-2. DOI: 10.1109/ASONAM.2012.30. URL: http://dx.doi.org/10.1109/ASONAM.2012.30.

S&P Dow Jones Indices (2019). *S&P 100 Fact Sheet*. URL: http://us.spindices.com/idsenhancedfactsheet/file.pdf?calcFrequency=M&force_download=true&hostIdentifier=48190c8c-42c4-46af-8d1a-0cd5db894797&indexId=2431 (visited on 08/04/2019).

Taspinar, Ahmet (2018). *GitHub - taspinar/twitterscraper: Scrape Twitter for Tweets*. URL: https://github.com/taspinar/twitterscraper (visited on 25/10/2018).

Tsui, Derek (2016). *Predicting Stock Price Movement Using Social Media Analysis*. Tech. rep. Stanford University, Technical Report.

Twitter (2018a). *Glossary*. URL: https://help.twitter.com/en/glossary (visited on 06/12/2018).

– (2018b). *Rate limits – Twitter Developers*. URL: https://developer.twitter.com/en/docs/basics/rate-limits (visited on 06/12/2018).

– (2019). *Twitter Advanced Search*. URL: https://twitter.com/search-advanced (visited on 17/04/2019).

Zhang, Xue, Hauke Fuehres and Peter A Gloor (2011). 'Predicting stock market indicators through Twitter "I hope it is not as bad as I fear"'. In: *Procedia-Social and Behavioral Sciences* 26, pp. 55–62.

7 Appendix

	Random Forest			Linear			Linear SVR			K-Nearest Neighbor		
Stocks	Baseline	Tweets	Difference	Baseline	Tweets	Difference	Baseline	Tweets	Difference	Baseline	Tweets	Difference
CELG	57.01%	58.86%	-1.85%	65.71%	82.97%	-17.26%	218.11%	27.11%	191%	50.48%	57.11%	-6.63%
MS	42.23%	37.05%	5.18%	38.99%	41.5%	-2.51%	157.31%	44.26%	113.05%	46.7%	55.54%	-8.84%
FB	69.31%	63.12%	6.19%	79.94%	37.86%	42.08%	184.96%	74.09%	110.87%	62.62%	40.63%	21.99%
UPS	58.92%	58.93%	-0.01%	58.86%	56.07%	2.79%	142.86%	50.25%	92.61%	64.25%	68.1%	-3.85%
JNJ	99.53%	103.91%	-4.38%	57.61%	86.06%	-28.45%	208.96%	125.24%	83.72%	73.57%	114.78%	-41.21%
AAPL	42.01%	40.86%	1.15%	57.56%	70.57%	-13.01%	249.18%	165.78%	83.4%	61.99%	45.24%	16.75%
BAC	45.17%	38.69%	6.48%	38.65%	48.65%	-10%	163.58%	80.34%	83.24%	50.58%	43.47%	7.11%
AIG	46.52%	46.64%	-0.12%	47%	79.47%	-32.47%	163.95%	84.62%	79.33%	38.42%	52.45%	-14.03%
EMR	50.99%	51.61%	-0.62%	53.34%	48.98%	4.36%	184.07%	106.41%	77.66%	43.98%	42.46%	1.52%
JPM	63.25%	57.01%	6.24%	64.78%	69.67%	-4.89%	155.17%	92.29%	62.88%	70.24%	48.71%	21.53%
XOM	50.82%	47.29%	3.53%	59.61%	59.35%	0.26%	166.2%	104.37%	61.83%	49%	47.14%	1.86%
INTC	46.95%	53.96%	-7.01%	48.1%	53.3%	-5.2%	107.98%	49.15%	58.83%	42.89%	50.45%	-7.56%
WBA	69.9%	64.43%	5.47%	72.49%	74.37%	-1.88%	193.74%	135.3%	58.44%	59.37%	41.9%	17.47%
C	47.19%	37.04%	10.15%	41.91%	32.23%	9.68%	175.36%	117.58%	57.78%	50.91%	46.29%	4.62%
NKE	52.67%	43.99%	8.68%	44.23%	56.46%	-12.23%	283.9%	226.82%	57.08%	40.02%	47.17%	-7.15%
HD	48.2%	39.29%	8.91%	47.16%	46.58%	0.58%	180.64%	125.17%	55.47%	45.74%	36.46%	9.28%
CMCSA	50.91%	42.68%	8.23%	45.91%	63.02%	-17.11%	313.83%	262.47%	51.36%	53.26%	69.13%	-15.87%
PYPL	37.79%	42.77%	-4.98%	38.24%	64.57%	-26.33%	199.78%	148.76%	51.02%	52.45%	56.21%	-3.76%
TGT	28.57%	32.57%	-4%	38.6%	48.38%	-9.78%	123.37%	73.5%	49.87%	28.91%	26.94%	1.97%
PM	82.63%	88.35%	-5.72%	102.81%	123.71%	-20.9%	121.2%	71.44%	49.76%	76.23%	77.65%	-1.42%
V	43.69%	34.69%	9%	35.08%	32.62%	2.46%	159.79%	114.43%	45.36%	44.4%	40.12%	4.28%

Table 4 continued from previous page

	Random Forest			Linear			Linear SVR			K-Nearest Neighbor		
Stocks	Baseline	Tweets	Difference	Baseline	Tweets	Difference	Baseline	Tweets	Difference	Baseline	Tweets	Difference
GILD	45.81%	47.73%	-1.92%	52.45%	56.58%	-4.13%	166.23%	123.85%	42.38%	29.3%	57.61%	-28.31%
PFE	59.12%	51.66%	7.46%	59.42%	57.54%	1.88%	106.72%	64.55%	42.17%	68.84%	43.57%	25.27%
MET	48.35%	55.23%	-6.88%	29.7%	74.26%	-44.56%	194.8%	153.25%	41.55%	55.49%	68.31%	-12.82%
SBUX	34.82%	37.16%	-2.34%	37.65%	49.47%	-11.82%	138.51%	98.89%	39.62%	39.46%	35.9%	3.56%
UNP	43.72%	47.25%	-3.53%	48.86%	43.31%	5.55%	179.87%	141.53%	38.34%	52.94%	56.07%	-3.13%
CSCO	44.09%	37.46%	6.63%	32.65%	36.17%	-3.52%	235.5%	198.85%	36.65%	31.09%	34.2%	-3.11%
BA	61.37%	58.48%	2.89%	62.66%	69.54%	-6.88%	100.27%	66.74%	33.53%	64.23%	58.04%	6.19%
SLB	35.54%	33.93%	1.61%	32.05%	32.22%	-0.17%	65.77%	32.75%	33.02%	35.14%	36.33%	-1.19%
GM	47.42%	50.79%	-3.37%	44.15%	75.35%	-31.2%	123.1%	91.9%	31.2%	51.8%	33.59%	18.21%
COF	54.52%	61.71%	-7.19%	52.03%	138.61%	-86.58%	83.87%	53.49%	30.38%	48.42%	50.57%	-2.15%
TXN	47.84%	37.33%	10.51%	34.72%	41.37%	-6.65%	217.66%	188.29%	29.37%	60.05%	30.45%	29.6%
CL	31.73%	28.24%	3.49%	35.66%	31.35%	4.31%	88.64%	59.66%	28.98%	25.83%	38.21%	-12.38%
ABT	32.72%	31.99%	0.73%	34.35%	41.26%	-6.91%	103.12%	75.64%	27.48%	28.35%	44.68%	-16.33%
CAT	52.21%	50.15%	2.06%	67.35%	97.41%	-30.06%	222.34%	196.09%	26.25%	43.06%	42.17%	0.89%
T	82.38%	78.79%	3.59%	96%	83.02%	12.98%	88.01%	64.27%	23.74%	91.93%	90.17%	1.76%
CHTR	24.3%	29.6%	-5.3%	18.23%	34.94%	-16.71%	123.46%	100.74%	22.72%	39.21%	30.03%	9.18%
WMT	44.29%	50.06%	-5.77%	52.54%	63.63%	-11.09%	69.91%	48.58%	21.33%	60.35%	60.26%	0.09%
MO	52.38%	56.65%	-4.27%	61.4%	68.01%	-6.61%	74.45%	54%	20.45%	55.9%	36.96%	18.94%
ORCL	38.45%	41.42%	-2.97%	49.97%	56.25%	-6.28%	81.42%	61.07%	20.35%	33.58%	46.81%	-13.23%
MSFT	19.28%	25.68%	-6.4%	23.13%	48.63%	-25.5%	162.95%	142.67%	20.28%	37.08%	33.93%	3.15%
RTN	64.9%	61.9%	3%	56.74%	67.65%	-10.91%	104.56%	85.36%	19.2%	58.17%	69.46%	-11.29%
MMM	69.63%	54.51%	15.12%	53.35%	65.9%	-12.55%	171.03%	151.96%	19.07%	55.17%	43.85%	11.32%
GOOG	41.59%	53.1%	-11.51%	40.8%	79.1%	-38.3%	168%	149.75%	18.25%	40.12%	28%	12.12%

Table 4 continued from previous page

	Random Forest			Linear			Linear SVR			K-Nearest Neighbor		
Stocks	Baseline	Tweets	Difference	Baseline	Tweets	Difference	Baseline	Tweets	Difference	Baseline	Tweets	Difference
UNH	41.03%	36.83%	4.2%	60.69%	44.56%	16.13%	76.89%	58.7%	18.19%	40.81%	38.17%	2.64%
PEP	49.47%	55.24%	-5.77%	51.4%	67.11%	-15.71%	122.71%	106.02%	16.69%	44.79%	58.34%	-13.55%
MCD	34.02%	33.4%	0.62%	33.51%	70.2%	-36.69%	63.96%	47.4%	16.56%	35.98%	27.01%	8.97%
NVDA	57.37%	55.84%	1.53%	51.99%	63.42%	-11.43%	114.58%	98.95%	15.63%	62.99%	72.2%	-9.21%
OXY	54.1%	57.97%	-3.87%	43.58%	66.83%	-23.25%	81%	65.81%	15.19%	77.44%	61.05%	16.39%
SO	53.07%	54.35%	-1.28%	59.24%	49.92%	9.32%	67.9%	57.18%	10.72%	34.97%	35.93%	-0.96%
GS	44.78%	46.4%	-1.62%	39.12%	61.17%	-22.05%	71.06%	60.53%	10.53%	53.61%	45.27%	8.34%
MRK	25.11%	30.26%	-5.15%	36.48%	49.31%	-12.83%	62.56%	52.71%	9.85%	18.85%	34.68%	-15.83%
ACN	50.06%	45.79%	4.27%	55.6%	75.51%	-19.91%	41.69%	31.97%	9.72%	52.38%	33.88%	18.5%
COST	27.18%	24.17%	3.01%	30.13%	43.51%	-13.38%	75.64%	67.53%	8.11%	25.17%	21.35%	3.82%
AGN	48.95%	51.26%	-2.31%	53.12%	41.53%	11.59%	110.73%	103.68%	7.05%	30.67%	48.27%	-17.6%
LMT	48.02%	47.83%	0.19%	55.07%	71.34%	-16.27%	102.52%	96.35%	6.17%	56.97%	45.62%	11.35%
KMI	44.46%	45.71%	-1.25%	49.18%	69.74%	-20.56%	128.88%	122.74%	6.14%	37.85%	45.4%	-7.55%
GD	58.66%	58.4%	0.26%	74.27%	122.08%	-47.81%	52.09%	46.24%	5.85%	68.33%	70.65%	-2.32%
USB	54.93%	49.12%	5.81%	56.5%	78.57%	-22.07%	68.97%	63.46%	5.51%	65.53%	41.57%	23.96%
BLK	46.58%	45.52%	1.06%	58.79%	85.54%	-26.75%	194.2%	188.89%	5.31%	50.63%	45.55%	5.08%
HON	27.75%	31.82%	-4.07%	35.37%	35.62%	-0.25%	88.31%	83.5%	4.81%	51.96%	41.91%	10.05%
QCOM	34.82%	36.4%	-1.58%	34.37%	49.98%	-15.61%	84.31%	79.78%	4.53%	38.04%	25.49%	12.55%
ABBV	40.25%	38.19%	2.06%	40.58%	46.9%	-6.32%	54.45%	50.88%	3.57%	37.64%	57.38%	-19.74%
VZ	46.25%	47.54%	-1.29%	51.82%	50.42%	1.4%	68.48%	65.36%	3.12%	39.28%	21.23%	18.05%
DIS	55.48%	50.14%	5.34%	38.14%	27.91%	10.23%	148.06%	145.29%	2.77%	60.52%	74.66%	-14.14%
COP	70.72%	61.03%	9.69%	57.61%	60.35%	-2.74%	123.14%	121.01%	2.13%	88.53%	69.45%	19.08%
ALL	33.65%	30.5%	3.15%	34.83%	32.3%	2.53%	46.21%	44.67%	1.54%	55.2%	44.04%	11.16%

Table 4 continued from previous page

	Random Forest			Linear			Linear SVR			K-Nearest Neighbor		
Stocks	Baseline	Tweets	Difference	Baseline	Tweets	Difference	Baseline	Tweets	Difference	Baseline	Tweets	Difference
KHC	44.6%	45.99%	-1.39%	55.32%	69.47%	-14.15%	83.36%	81.98%	1.38%	49.97%	52.17%	-2.2%
SPG	56.49%	56.01%	0.48%	58.69%	52%	6.69%	49.69%	49.27%	0.42%	51.48%	40.06%	11.42%
BKNG	50.85%	50.26%	0.59%	48.51%	50.52%	-2.01%	95.04%	94.78%	0.26%	43.41%	57.53%	-14.12%
MA	52.78%	55.14%	-2.36%	44.76%	87.23%	-42.47%	142.3%	142.44%	-0.14%	29.16%	48.88%	-19.72%
MDLZ	69.21%	66.92%	2.29%	51.84%	56.09%	-4.25%	46.73%	47.02%	-0.29%	54.46%	55.19%	-0.73%
BRKB	43.99%	48.33%	-4.34%	64.69%	117.01%	-52.32%	40.46%	41.33%	-0.87%	46.72%	39.48%	7.24%
UTX	54.48%	45.88%	8.6%	51.93%	58.95%	-7.02%	97.07%	98.2%	-1.13%	48.68%	47.72%	0.96%
IBM	29.7%	40.41%	-10.71%	36.73%	34.97%	1.76%	122.34%	124.06%	-1.72%	36.18%	41.06%	-4.88%
MDT	51.28%	49.46%	1.82%	56.25%	64.28%	-8.03%	65.01%	66.78%	-1.77%	91.48%	46.61%	44.87%
LLY	59.05%	62.83%	-3.78%	49%	56.08%	-7.08%	76.58%	79.7%	-3.12%	60.86%	66.62%	-5.76%
FDX	79.73%	76.43%	3.3%	78.31%	115.27%	-36.96%	42.3%	45.53%	-3.23%	57.09%	44.76%	12.33%
DHR	38.24%	38.05%	0.19%	43.39%	78.6%	-35.21%	77.79%	82.78%	-4.99%	53.01%	43.33%	9.68%
AMZN	18.77%	18.81%	-0.04%	24.53%	57.22%	-32.69%	121.53%	127.3%	-5.77%	25.44%	23.61%	1.83%
AXP	32.74%	33.65%	-0.91%	29.1%	47.53%	-18.43%	78.16%	84%	-5.84%	46.62%	42.06%	4.56%
AMGN	62.69%	55.67%	7.02%	47.18%	59.85%	-12.67%	127.58%	135.64%	-8.06%	63.91%	39.32%	24.59%
DUK	39.85%	39.54%	0.31%	40.37%	41.03%	-0.66%	15.46%	25.59%	-10.13%	39.65%	37.08%	2.57%
CVS	42.5%	42.33%	0.17%	41.46%	51.77%	-10.31%	39.31%	49.83%	-10.52%	51.86%	39.58%	12.28%
BK	45.61%	59.6%	-13.99%	43.08%	81.56%	-38.48%	82%	96.73%	-14.73%	41.1%	69.87%	-28.77%
GOOGL	38%	44.52%	-6.52%	38.52%	125.56%	-87.04%	172.57%	188.43%	-15.86%	25.8%	41.6%	-15.8%
EXC	65.67%	68.49%	-2.82%	57.67%	77.35%	-19.68%	43.62%	59.52%	-15.9%	62.97%	61.52%	1.45%
DWDP	48.92%	56.09%	-7.17%	55.56%	80.6%	-25.04%	125.31%	142.39%	-17.08%	45.11%	55.29%	-10.18%
KO	43.78%	47.71%	-3.93%	47.66%	99.52%	-51.86%	86.37%	103.84%	-17.47%	51.27%	63.12%	-11.85%
NEE	48.9%	46.74%	2.16%	75.88%	89.75%	-13.87%	46.78%	64.95%	-18.17%	45.03%	73.96%	-28.93%

Table 4 continued from previous page

	Random Forest			Linear			Linear SVR			K-Nearest Neighbor		
Stocks	Baseline	Tweets	Difference	Baseline	Tweets	Difference	Baseline	Tweets	Difference	Baseline	Tweets	Difference
FOX	142.33%	142.69%	-0.36%	201.03%	306.42%	-105.39%	53.32%	72.71%	-19.39%	113.16%	112.16%	1%
PG	59.28%	50.92%	8.36%	44.2%	43.51%	0.69%	117.33%	138.33%	-21%	77.51%	54.09%	23.42%
GE	38.83%	30.55%	8.28%	30.45%	36.8%	-6.35%	33.74%	54.93%	-21.19%	36.45%	27.44%	9.01%
CVX	61.56%	69.36%	-7.8%	66.39%	77.22%	-10.83%	169.61%	192.65%	-23.04%	54.58%	53.64%	0.94%
WFC	59.71%	62.9%	-3.19%	52.96%	112.68%	-59.72%	197.76%	222.48%	-24.72%	52.78%	94.28%	-41.5%
BMY	66.37%	71.87%	-5.5%	58.17%	86.13%	-27.96%	129.88%	155.22%	-25.34%	67.16%	65.66%	1.5%
NFLX	27.39%	28.24%	-0.85%	30.28%	51.16%	-20.88%	99.68%	130.56%	-30.88%	43.24%	42.85%	0.39%
F	51.58%	47.83%	3.75%	37.9%	45.44%	-7.54%	83.94%	125.09%	-41.15%	47.71%	48.75%	-1.04%
LOW	48.82%	50.02%	-1.2%	32.53%	60.15%	-27.62%	72.2%	113.89%	-41.69%	43.49%	41.11%	2.38%
FOXA	47.26%	48.09%	-0.83%	126.38%	237.85%	-111.47%	58.17%	250.09%	-191.92%	62%	46.17%	15.83%
Average			0.29%			-16.268%			18.355%			1.062%
Median			0.085%			-11.625%			10.625%			1.51%

Table 4: MAPE of machine learning models on baseline and tweets datasets and calculated difference (improvement) with mean and median of the difference sorted descending on 'Linear SVR Difference'