THE TWEETING 500: CAN WE PREDICT THE IMPACT ON THE STOCK MARKET?

Researching the impact of tweets written by fortune 500 CEOs and companies on the price of stocks using sentiment analysis

Student details

Author: Tyra Monsch
Student number: u898424
E-mail: t.s.monsch@tilburguniversity.edu

Thesis committee

Supervisor: Prof. Pieter Spronck
Second reader: Prof. Eric Postma

Tilburg University
School of Humanities & Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands

June 25th, 2021

Table of Contents

Abstract

Background. Stock market prediction has been an active research area for both practitioners and academia. Stock prices are notoriously sensitive to investor sentiment, and with the rise of social media networks, there has been an increasing trend to incorporate several sources of both textual and numerical data in forecasting models. Previous research has investigated how public sentiment correlates with stock prices and to what extent. However, recent stock market rallies partly fueled by Tesla's CEO Elon Musk have raised the question of how much influence a company's or CEO's tweets have on the price of their stocks. With social media being used as a source by many for investment strategies, those who have an incentive to drive the price of a certain stock up are easier to do so when they have a large enough followers base. The Security and Exchange Commission poses strict regulations to prevent possible market manipulation by companies and forbids companies to spread false information in an attempt to influence their stock price. However, it is not illegal for a company or CEO to tweet positively about their company. Thus, researching whether the sentiment of CEOs and companies' tweets influences their stock price could add value to those involved in stock market regulation, and academia.

Objectives. In our research, we aim to investigate to what extent CEOs and companies can influence the price of their stocks based on the sentiment of their tweets.

Methods. We collected the tweets of 339 companies on the fortune 500 and 7 CEOs, and their respective stock prices from July 5th 2019 until January 30th 2020. We assigned polarity scores to their tweets to measure the sentiment and evaluated the influence of the sentiment on the stock price immediately after posting the tweets, and up till two hours later by making use of regression. XGBoost, SVM, and Random Forest were used.

Results. Our results indicate that neither CEOs nor companies seem to influence the price of their stock immediately after posting a tweet, and up till two hours later.

Conclusions. Training a model on our datasets has proven to be difficult, as there is a lack of negative tweets posted and little correlation between the polarity values and percentage changes in stock prices.

Keywords: Sentiment analysis, Fortune 500, CEOs, Stock prices, SVM, XGBoost, Random Forest.

1. Introduction

Multiple factors can influence the price of stocks. Examples are company announcements, financial news, but also public sentiment on social media. Negative news about a company often results in a drop in stock prices, whilst positive news can lead to a surge in the stock price. With this increase in available news through different news outlets, and social media being used as a tool to not only share news but also to express opinions, researchers have been investigating the relationship between public sentiment and stock prices for a few years now. However, an individual's influence on stock prices has not been investigated yet. Some individual social media accounts have more impact and engagement than others, which can be used to influence or incite an action among followers. In our research, we aim to investigate if certain individuals can also impact the price of stocks. The reason for this is the recurrent phenomenon that Elon Musk can influence capital markets through his social media following, whether intentionally or unintentionally.

On January 29th 2021, Elon Musk, Tesla Inc.'s CEO, caused the cryptocurrency Bitcoin to rise by over 20% after he added the hashtag 'Bitcoin' to his Twitter biography. The most recent surge in stock prices fueled by Elon Musk was with the stock GameStop on January 26th, 2021. The stock was already rising because users on a platform named 'Reddit' were elaborately planning to buy the stock to drive its price up. After Elon Musk tweeted "GameStonk", where Stonk is another word for stock, many believed that Elon meant to invest in the stock GameStop, causing the stock price to rise even further. The stock increased with such unnatural high percentages that multiple stock trading platforms had to pause trading for that day, giving rise to the launch of a federal investigation by the Department of Justice and Securities and Exchange Commission in the USA for possible market manipulation (Business Insider, 2021).

Elon Musk's tweets also impact the stocks of his own company, Tesla. For example, the tweet "funding secured" in 2018, which led Tesla's stock price to reach a record-high level. However, sometimes the

causal relationship between a tweet and a stock price is more difficult to discover. The most prominent example was when Elon Musk tweeted 'Use Signal', which led many to believe Elon meant the stock of a company named 'Signal Advance', causing the price to rise by over 400% overnight.
Utilizing social media platforms to influence people's behavior has been done extensively for years. However, market regulations prevent anyone from spreading inaccurate information on social media platforms about a stock or company to instigate pump-and-dump schemes, in which traders urge others to buy a stock to increase the value of their current position, allowing them to sell the stock at the higher price. The problem that arises is that influential individuals such as Elon Musk can instigate such pump-and-dump schemes by simply stating their opinion about a company, or even by posting a tweet that at first glance is difficult to interpret what is meant by it.

The widespread media coverage of Elon Musk's tweets and their impact on the price of several stocks, including his stock, Tesla, gives rise to the question of whether other CEOs' or their company's tweets also have this impact. CEOs and companies' potential ability to cause herding amongst their followers, – e.g. massively buying or selling a stock - can be disruptive to financial markets, as it causes asset bubbles and irrational asset prices. A prime example of the impact of herding and information overflow is the dotcom bubble between 1998 and 2001, in which different internet companies that weren't generating any revenue were traded on the stock exchange and valued too high. This caused many other investors to believe these companies were a good investment – causing the bubble to bloat even further. Eventually, this dotcom bubble burst, causing large losses to investors. Thus, being able to anticipate abnormal behavior of investors in the stock market is of high importance to policymakers, as it allows for immediate intervention. This leads to the following research questions:

R1: "To what extent do CEOs influence the stock price depending on the sentiment of their tweets?"
R2: "To what extent do corporates influence their stock price depending on the sentiment of their tweets?"

2. Related work

The goal of this chapter is to survey previous research on stock market prediction using sentiment analysis. There has been significant research conducted in sentiment analysis and how it can be used for prediction purposes, including stock market prediction. Research to date in that area has mainly focused on the correlation between stock market indexes and overall public sentiment. In this study, we try to build on previous work that has been done in this area to establish a correlation between individual tweets of CEOs and the price of their stock, and the correlation between the company's tweets and the stock price.

2.1 Disinformation and stock market manipulation

The Securities and Exchange Commission oversees all the financial markets in the United States, aims to ensure fair markets and to protect investors. Market manipulation and one-sided information can greatly distort market fairness, which implies the importance of governments to support greater competition in information and stricter regulation to discourage market manipulation, as stated in a study conducted by Aggarwal and Wu (2013). They researched how those who seek out information about a stock's value often pave the way for certain individuals and organizations to manipulate the stock market. They find that in a market without manipulators, information seekers improve market efficiency, but in a market with manipulators, information seekers play a more ambiguous role; more information seekers equals a greater competition and demand for shares, which makes it easier for manipulators to distort market efficiency.

Uncovering which parties are more likely to be market manipulators is vital to take a more active approach towards protecting investors. Some studies have found that especially informed parties, such as corporate insiders, are likely to be the market manipulators (Aggarwal & Wu, 2013). As social media is increasingly being used by companies as a way to express their voice, but also for company announcements, those who seek information about a stock will consider several news outlets to form their opinion and to make their decision whether to buy or sell a stock. As companies are in control of what type of news they want to share with the world through their often large social media platforms, it is challenging to verify all the social media posts on accuracy. This highlights the importance of implementing a reform policy on social media to combat disinformation (Nicoli, 2020). Disinformation is defined as ‘false, incomplete or misleading information that is passed, fed, or confirmed to a targeted individual, group, or country’ (Nicoli, 2020). If CEOs and companies can actually influence the stock prices of their company through their social media posts, it is all the more important for policymakers to ensure that there is no false information shared to manipulate the stock market. By establishing whether there is an indication that companies and CEOs influence their stock prices based on the sentiment of their tweets, this may give rise to further research to investigate this correlation further.

2.2 Sentiment-based investment strategies

The opinions of stockholders have proven to be a critical indicator of the future value of a stock. With the rise of social media, opinions about stocks are widely available through several social media networks. Although having a diversified pool of information and opinions about stocks available can be a positive development, it can also be problematic when individuals do not conduct proper research to validate certain claims made on social media by particular individuals. This vulnerability of certain investors and the need for protection of those who are financially less literate is highlighted in a study conducted by Chousa et al. (2017). They analyzed investors’ social media activity through a platform named StockTwits and how these messages influenced the Chicago Board Options Exchange Volatility Index (VIX) by using a logit model. StockTwits has similarities to Twitter, except that it focuses solely on the stock market and users can share their opinions about stocks and the capital market. StockTwits works through a subscription-based model, where the audience can follow different investors for advice and tips. The information sharing is rather quick, which could lead to an immediate effect on stock markets as some users have thousands of followers. They find that the social media activity deployed by investors on StockTwits leads to a variation in market risk. They also found that investors can be classified into different groups, those who rely on technical analysis and those who do not, and that these different investors are influenced differently by the content they consume through platforms such as StockTwits. For nontechnical followers, there was a significant effect for message sentiment on the variation of the VIX, whereas for technical users message sentiment does not seem to have any significant effect on the variation of the VIX. Their findings are important, as it show cast that some individuals are mainly guided by the sentiment, rather than by technical analyses. This further suggests that stock prices can indeed rise or fall based on news sentiment. Additionally, it could explain abnormal high returns following a tweet, if no other cause for the spike in stock price was found other than the tweet itself. The findings of Chousa at el. (2017) also highlight the importance for policymakers to protect the less financially literate individuals, as acting purely upon the sentiment of a message is not always a rational investment strategy.

2.3 Time lags for stock market prediction

In stock market prediction, it is crucial to uncover the time lag between a tweet and its effect on a stock. Previous research about sentiment analysis in stock market prediction has consistently incorporated granger causality analysis. One study conducted by Narges et al (2018) found a statistically significant causal relationship between the tweets and their sentiments in different lags and the stock prices. For example, Apple inc. had a lag of 2 days on impact of social media on stock market return. Similar results are found for the companies Netflix and Microsoft by a study conducted earlier by Smailović et al. (2013). Other researchers have uncovered a correlation between sentiment and stock price movements in the short term, for example the study by Rao and Srivastava (2015). They analyzed over 4 million tweets and their effect on stock market indexes. Their results showed a high correlation between Twitter sentiments and stock prices. More interestingly, they established through Granger's Causality that Twitter forums influence stock price movements especially in the short term.

Granger causality analysis is not a suitable method for our dataset. Our data is not a time series, as tweets are not posted with a regular time interval. We could artificially introduce a lag and make our data a time series, but this is not ethical data science practice. However, we are able to establish whether there is a correlation between a tweet's sentiment and the stock price immediately after the tweet is posted and up till two hours later. Unlike the results of Narges et al (2018) who found an impact of the sentiment of tweets up till 2 days later, we suspect that tweets only have an impact on the closest opening price and one hour later, and its impact will eventually decrease after one hour. Thus, we expect to find that tweets influence stock price movements especially in the short term. The rationale behind this assumption is that with Elon Musk, we have seen immediate effects of the tweets on the stock price, often followed by a correction of the stock price after a few days. An additional reason for solely analyzing the impact of the first 2 hours is that as more time progresses, additional factors could have influenced the changes in stock prices. Despite the fact that we will not analyze the influence of tweets on stock prices through granger causality, the performance of our different models will establish whether we are able to predict the stock price immediately after a tweet, and a few hours later.

However, it is also possible that there is a low correlation between Twitter sentiment and daily stock price, but a high correlation between Twitter sentiment and abnormal returns. Elon Musk recently proved that influential individuals can also cause abnormal stock returns, as seen in the cases of

Gamestop and even Tesla. A study by Gabriele et al. (2015) researched the effects of Twitter Sentiment on Stock price returns, using the 30 companies that form the Dow Jones Industrial Average index. They found a low Pearson correlation between the time series over the 15 months they investigated, but a high dependence between abnormal returns and Twitter sentiment. Research by Borovkova and Xiaobo (2015) shows similar results, with a strong relationship between news sentiment and abnormal returns of S&P 500 stocks.

2.4 Machine learning techniques and results from previous research

SVM is often used in previous research to predict stock market movements, as the study conducted by Mittal and Goel (2011). They used four different learning algorithms, one will be used in our research namely SVM. The SVM has an accuracy of 59%. Similarly, Ren and Liu (2019) also used SVM to predict the stock market movements using sentiment analysis. They achieved a result as high as 89.9%. However, these results are not representative of our dataset as they analyzed investor sentiment, which is known to play an important role in stock market forecasting. Similarly, Khedr et al. (2017) predicted the influence of financial news on the stock market using sentiment analysis. Their prediction accuracy for their SVM model was 58% when predicting stock prices using data from yahoo and nearly 69% when using Facebook data. As financial news outlets are one of the main sources for investors to decide their investment strategy, these results are not representative for this research.

Research is increasingly suggesting that Random forests, e.g. RFs, perform well on predicting stock prices compared to other machine learning methods (Lohrmann and Luukka (2019); Basak et al., (2019); Khan et al., (2020)). Random forests are an ensemble learning algorithm and are relatively easy to understand. Ballings et al. (2015) argued that SVMs are more commonly used in academic research for stock price prediction and that up until the year 2015 only 9% of the papers used RFs in their research on stock price prediction. However, their research on stock price direction using SVMs, RFs, K-nearest neighbor, and logistic regression suggests that RFs performs better in terms of prediction accuracy , especially over a one-year period. This suggests that RFs could in fact outperform SVM for our task.

Additional research also supports the use of RFs for stock market prediction. It is even evident that RFs could outperform SVMs, as found by Basak et al. (2019). They predicted the stock price of ten technology and social media companies. They found that RFs indeed outperform SVMs. Khan et al. (2020) used a combination of twelve learning algorithms to predict the stock price on financial data.

They also argued that RFs performed the best amongst their learning algorithms. Lorhmann and Luukka (2019) used RFs to predict stocks for the S&P 500 index. Their findings were that RFs were more reliable for traders than conventional buy and hold strategies. Additionally, it is argued that tree-based ensemble models in general perform well on stock price prediction. A study by Ampomah et al. (2020) compared the performance of different tree-based models namely Bagging, AdaBoost, XGBoost, and Random forests.

This previous research suggests that it is worthwhile to investigate whether RF and XGBoost outperform SVM.

2.5 How sentiment influences stock prices

It is vital to note that it can not be expected that all companies show the same correlation between sentiment and stock price. A study conducted on sentiment analysis of investors' opinions by Dickinson and Hu (2015) aimed to distinguish companies who showcase a correlation between stock price and sentiment and those who do not on commonalities. They used a random forest model for classification. Their models achieved an accuracy of 68%. Their findings were that the correlation between stock price and sentiment depends on the company, and that consumer-facing companies show different correlations. These findings are interesting for our research as well, as we may find that certain companies have a stronger correlation between stock price and sentiment, depending on for example their industry or market.

Another potential cause for a high correlation between stock price and sentiment for some companies is media coverage. Previous research has investigated the relationship between financial news and the stock market. Alanyali et al (2013) established a positive correlation between the daily number of mentioning's of a company in the Financial Times and the daily transaction volume of a company's stock on the same day that the news is published and the day before. Thus, they conclude that a greater interest in a particular company in the news is related to a greater interest in its stock. Another study conducted by Mohan et al. (2019) suggests a strong relationship between stock prices and media news. When a correlation is found between sentiment and stock price, it may be worthwhile to investigate whether the companies and CEOs who show a high correlation also have a higher media exposure.

3. Methods

In this chapter, we will discuss the methods used to perform the analysis. We will discuss how we collected the data, the techniques used for extracting sentiment from tweets, along with the algorithms and machine learning techniques used to answer the research questions: *"To what extent do CEOs influence the stock price depending on the sentiment of their tweets?"* and *"To what extent do corporates influence their stock price depending on the sentiment of their tweets?"*.

3.1 Data collection

In this section, we will discuss the methods used for collecting our data. We have two final datasets, one for the CEOs and one for the companies. For these datasets, Twitter and stock data were collected separately. We used similar methods for collecting the data for both datasets.

3.1.1 Twitter data

The tweets are gathered from CEOs and companies on the Fortune's 500 list of 'Worlds most admired companies' (hereinafter referred to as 'Fortune 500'), which is based on a study that surveys executives, directors, and financial analysts to identify the companies that have the strongest reputation within and across their industries (Fortune, n.d.). The tweets are collected from July 5th 2019 until January 30th 2020. For this research, we will study companies that are on the Fortune 500 and have a CEO who is active on Twitter.

An "active Twitter user" is an arbitrary definition, of which the literature so far does not provide a definition. As this research aims to determine the extent to which CEOs and companies influence the price of their stocks, it is important for CEOs and companies to tweet regularly to determine a relationship between a tweet's sentiment and the stock price. Thus, an active Twitter is defined as one who tweets at least 52 times a year. It is also important that the account is verified by Twitter or the company, which indicates that the account is linked to the company and CEO.

The Twitter data could be collected through the Twitter API, manually, or through Twint. The Twitter API is quite cumbersome in its set up and imposes several restrictions in its use. Twint is an advanced web-scraping tool for Twitter that allows one to fetch all tweets posted meeting specific criteria such as certain usernames, hashtags or timeframes. We decided to use Twint to collect the tweets from the companies and CEOs, as it was found to be the most convenient in its set up.

3.1.2 Stock data

Historical stock prices were gathered through the Python library yfinance. Yfinance is the API from Yahoofinance, which is a well-known platform for financial news and stock prices. To gain access to the stock data, the library requires you to specify a beginning date, end date, interval, and company ticker. The interval specifies whether you want the stock price per day, per hour, or even per minute. The company ticker specifies the stock symbol ticker, which is the abbreviation used for the company on the stock market.

There were in total 252 trading days in 2019. We only used July 5th 2019 until January 30th 2020, and were only able to access the stock prices per one-hour interval, e.g. 9.30-10.30, 10.30-11.30, 11.30-12.30, etc. until 5 PM. The historical stock price information includes the opening price, high, low, closing price, and adjusted closing price. The opening price is the price of the stock when trading began that day, whilst the closing price is the price of the stock when trading ended that day. The adjusted closing price takes into account corporate actions. The normal closing price is the cash value of a stock, while the adjusted closing price reflects the overall value of a stock better. Low and high refer to the lowest and highest price that the stock reached during a given trading day. As we use 1-hour intervals for the stock prices, the opening prices refer to the opening time at that given one-hour timeslot, closing price to the closing price for that given timeslot, high for maximum price during that timeslot, and low to the minimum price during that timeslot.

3.1.3 Matching tweets to stock prices

Matching the tweets to stock prices is a crucial step to perform our analysis and subsequently answer our research questions. CEOs and companies post tweets on an irregular time interval, which requires appropriate matching of the tweets to the stock prices. The stock market is open from 9.30 AM until 17.00 PM. It should be noted that the sentiment of investors on social media can influence the stock prices while the stock market is closed. Companies often publish important news after the stock market closes, which could result in a different opening price the next day. This is due to the fact that limited trading occurs outside of the opening hours of the stock market by a specific set of individuals and corporations. This is the reason that the opening price of the next day is also included in the research, if a tweet is posted outside of the opening hours of the stock market.

Tweets posted outside of opening hours are assumed to have their first effect on the stock market at the opening hour of the next day, which equals the opening price at 9.30 AM. As we analyze only the short-term effects of the tweets on the stock prices, the opening prices at 10.30 AM and 11.30 AM are also included for tweets posted outside of the opening hours of the stock market. To determine whether there is a change in stock price, the relative change is calculated from one hour to the next. Thus, when a tweet is posted outside the opening hours, the opening price one hour before the stock market closes will be compared to the three first opening hours of the following day. This means that the closing price of 17.00 PM of the previous day will be compared to the opening prices 9.30 AM, 10.30 AM, and 11.30 AM of the following day. Tweets posted while the stock market is still open are matched to their closest opening price. Thus, a tweet posted at 13.00 PM will be matched to the opening price at 13.30 PM, and subsequently to the opening prices at 14.30 PM and 15.30 PM. The relative changes are calculated from one hour to the next. Thus, the change between the opening price of 12.30 PM and the three consequential opening hours 13.30 PM, 14.30 PM, and 15.30 PM will be calculated for a tweet that was posted at 13.00 PM. Tweets posted on weekends are matched to the opening hours on Monday, and tweets posted on holidays are matched to the day the stock market opens up again.

It is crucial to ensure that the appropriate time zones are used, which are America/New York for all the stock prices. These should be matched to the time zones in which the Tweets are posted, which sometimes differ. The python library pytz has a function time zone, which can be used to adapt and match the time zones used.

3.2 Sentiment analysis

In this section 3.2, we will discuss the steps taken to perform the sentiment analysis. We will discuss the steps involved in pre-processing data, and the sentiment classifier used to perform the sentiment analysis.

3.2.1 Pre-processing

Pre-processing textual data involves cleaning the data to prepare it for sentiment classification (Haddi, Liu, & Shi, 2013). Social media data usually contains noise and parts that are uninformative such as HTTPS tags and white spaces. Additionally, some words do not have an impact on its orientation. Not removing certain words or noise increases the dimensionality of the problem, hence also increasing the difficulty in classification, as every single word is considered a separate dimension (Haddi, Liu, & Shi, 2013). Thus, cleaning data can improve the performance of the classifier and even speed up the classification. Text cleaning, stop words removal, whitespace removal, stemming and lemmatization belong to the data transformation process.

In our research, we use the lexicon tool Textblob. Textblob does not require a substantial amount of pre-processing. Removing certain (stop)words may actually intervene with its functionality. Textblob can analyze non-conventional text and automatically detect and remove stop words that can be ambiguous. For example, words such as “but” and “very” are useful for identifying the sentiment after those words (Zhou, 2019). Additionally, lemmatization techniques often used in Natural Language Processing could potentially be problematic when using Textblob. When words have the same base root such as “good” and “better”, they will be ignored when doing lemmatization (Zhou, 2019). As lemmatization and stemming are argued to remove valuable information when using sentiment analysis tools, we will not follow these approaches in our research.

Thus, the bare minimum of pre-processing should suffice for using Textblob, such as removing HTTPS tags and unnecessary white spaces. An additional argument for the minimum amount of pre-processing needed during this particular research, is the nature of our data. Despite the fact that social media data is often considered messy, our data is relatively clean as it is posted by renowned companies and CEOs. Thus, the use of for example slang is not prevalent in our data, and misspellings rarely occur.

3.2.2 Sentiment classification using Textblob

After the pre-processing, a sentiment classifier is applied to the data to determine the sentiment of the tweets for CEOs and companies. The Python library Textblob is used, as this is considered the most appropriate sentiment classifier for formal language.

Textblob classifies the sentiment of tweets using the term ‘polarity’ with a scale from -1 to 1, where -1 means extremely negative, and 1 extremely positive, and 0.0 neutral. A polarity score of smaller than -0.5 translates to a “Very negative” tweet, 0.0 translates to a “Neutral tweet”, larger than -0.5 but smaller than 0.0 translates to a “Slightly negative” tweet, larger than 0.0 but smaller than 0.5 translates to a “Slightly positive” tweet. Finally, a polarity score larger than 0.5 translates to a “Very positive” tweet. In the table below we illustrate this labelling.

Label	Very negative	Slightly negative	Neutral	Slightly positive	Very positive
Polarity score	< -0.5	> -0.5 and <0.0	0.0	>0.0 and <0.5	>0.5

Table 1: Sentiment labels Textblob Polarity

3.3 Machine learning techniques

In this section, we will discuss the machine learning techniques used. These are support vector machine, random forest, and XGBoost. We used these machine learning techniques for regression purposes to determine the extent to which both CEOs and companies influence the stock price based on the sentiment of their tweets. The models will each yield 3 different predictions, one for the relative stock price change immediately after a tweet is posted, up till 2 hours later.

It should be noted that the default values of the hyperparameters were used for all our machine learning techniques. The libraries we used employ default values which are deemed to be sufficient to cater to most use cases (Lee, 2019). Research has shown that it is challenging to improve the performance significantly when further tuning these parameters, and that default values often result in non-inferior performance compared to tuning the hyperparameters (Weerts & Vanschoren, 2020).

3.3.1 Support vector machine

Support vector machine, hereinafter mentioned as SVM, has become one of the most widely used machine learning algorithms used to estimate future stock prices (Henrique, Sobreiro, & Kimura, 2018). We used the supervised learning algorithm SVM to determine the extent to which CEOs and companies can influence their stock price based on the polarity of their tweets. By training the SVM on our dataset, we can determine whether stock prices increase as polarity values increase, and decrease when polarity values decrease.

SVM makes use of kernel functions which allows it to solve nonlinear problems by projecting it into the high-dimensional feature space (Ren & Liu, 2019). SVM is a dynamic approach, which works well with forecasting stock prices, as the stock market is dynamic and nonlinear. Another advantage of SVM is that it can reduce overfitting as it selects a maximal margin hyperplane in the feature space (Ren & Liu, 2019). SVM implements a structural risk minimization principle which aims to minimize an upper bound of the generalization error instead of minimizing the training error (Bao, Lu, & Zhang, 2004). This is in stark contrast with most traditional neural network models which implement an empirical risk minimization principle. SVMs achieve an optimum network structure as it balances the empirical error and the VC-confidence interval. This leads to better generalization compared to other traditional neural network models (Bao, Lu, & Zhang, 2004).

The python library scikit-learn provides an implementation for SVM for machine learning, namely svm.SVR, which is specifically for regression purposes. The four main parameters to tune for SVM are kernel, degree, gamma and C. We will shortly discuss them below (Saini, 2020).

Kernel specifies the kernel type which the algorithm will use. The default value is set to RBF. RBF kernels are considered the most generalized and used form of kernelization as it is similar to the Gaussian distribution (Sreenivasa, 2020). RBF kernel computes the similarity between points and how close they are to each other. RBF has as its main advantage that it only stores the support vectors during training instead of the entire dataset, thereby overcoming space complexity problems (Sreenivasa, 2020).

Degree is the degree of the polynomial kernel function and is ignored by the other kernels (scikit-learn, 2021). The default of this value is 3. Research has shown that the polynomial kernel function of 3 is indeed optimal in several cases (Liu & Xu, 2013). However, it should be noted that there is limited research available that could support the default value of 3. One can try different values for this parameter to aim for better results.

C is the regularization parameter. The strength of the regularization is inversely proportional to C. C must be strictly positive, with a default value set to 1.0 (scikit-learn, 2021). The function of C is to balance the trade-off between the complexity of the model and the empirical error (Achsan, 2019). A value of C that is too large will often lead to overfitting, whereas a value of C that is too small tends to lead to underfitting.

Gamma is the kernel coefficient for RBF. Gamma is mathematically denoted as γ. The default value of γ is set to scale, which means it uses 1/ (n_features * X.var)) as value of γ (scikit-learn, 2021). The γ parameter defines how far the influence of a single training example reaches. Low values for γ indicate 'far' influence, and high values indicate 'close' (scikit-learn, 2021). Overall, when γ is too small, the model is unable to capture the complexity of the data. If γ is too large, the radius of the area of influence will only include support vector itself, which means that regularization with C can not prevent overfitting (scikit-learn, 2021). Thus, when γ is too large, SVM tends to overfit. In contrast, when γ is too small, SVM tends to underfit (Achsan, 2019).

3.3.2 Random forest

A random forest, hereinafter mentioned as RF, is a meta estimator which aggregates several decision trees and fits classifying decision trees on sub-samples of the dataset (scikit-learn, 2020). Random forest makes use of the bagging technique, which implicates that the trees run parallel and that there is no interaction between the trees when they are built (Chakure, 2019). The predictive accuracy is improved and overfitting is controlled by using averaging.

The python library scikit-learn provides an implementation of RF for machine learning, namely the RandomForestRegressor (scikit-learn, 2020). The evaluation technique used for RF is RMSE, which we will discuss in section 3.5. The RandomForestRegressor of scikit-learn has several parameters. We will briefly discuss the most relevant ones with a short description of what they entail.

“N_estimators” indicates the number of trees in the forest, which has the default of 100 (scikit-learn, 2020). Research has shown that the optimal number of trees lies in a range of 64 to 128. Beyond this, a larger number of trees most likely does not lead to a significant performance gain, but increases the computational cost substantially (Oshiro, Pere, & Baranauskas, 2012). However, it should be mentioned that research to date does not reach an unambiguous conclusion on the optimal number of trees. Thus, one can tune this parameter to achieve better results.

“Criterion” specifies the function to measure the quality of a split, which is set to the mean squared error as default. We discuss more about the mean squared error in section 3.5, as this is our evaluation method for all our machine learning techniques.

“Max_depth” specifies the maximum depth of a tree, which is set to none as default. This means that the nodes are expanded until all leaves are pure or until the leaves contain less than the minimum number of samples required to split an internal node. As a general rule, the deeper one allows the tree to grow, the more complex the model will become which makes it prone to overfitting. On the other hand, setting this parameter to a low value can cause underfitting. Thus, it is suggested to run the model at the default and change this parameter when overfitting or underfitting is detected.

“Min_samples_split” is the minimum number of samples required to split an internal node, which is set to 2 for the default value (scikit-learn, 2020). This implies that when a terminal node has more than 2 observations and is not a pure node, it gets split further into sub-nodes (Saxena, 2020). Setting this parameter too high may result in underfitting, as the minimum requirement of splitting a node leads to no significant splits to be observed, resulting in a decrease in both training and test scores (Saxena, 2020).

“Min_samples_leaf” is the minimum number of samples in newly created leaves, which is set to value 1 as default. Splits are discarded if one of the leaves would contain less than the minimum number of samples in the newly created leaf. When this parameter is too low, the risk of overfitting emerges. Once the parameter increases too much, the model easier underfits. An empirical study states that this parameter is optimal between 1 to 20 for the CART algorithm that scikit-learn is employing (Mantovani R. G., et al., 2018).

“Max_features” is the number of features that are considered when looking for the best split. The default is automatic, which means the max features are equal to the number of features (scikit-learn, 2020). This parameter limits overfitting, as it increases the trees’ stability and reduces variance (Mithrakumar Mukesh, 2019).

“Bootstrap” indicates whether bootstrap samples are used to build the trees. A bootstrap sample is a smaller sample that is bootstrapped from a larger sample (Statistics how to, 2020). This is set to a default of true. The “max_samples” parameter determines the fraction of the original dataset given to individual trees (Saxena, 2020). The default value is set to none.

3.3.3 XGBoost

Several scholars have argued that tree-based models can outperform other machine learning methods such as logistic regression and SVM for stock price prediction, as discussed in chapter 2. Additionally, it is argued that tree-based ensemble models, in general, perform well on stock price prediction. A study by Ampomah et al. (2020) compared the performance of different tree-based models namely Bagging, AdaBoost, XGBoost, and Random forests. They found that tree-based models outperformed other models such as SVM and logistic regression.

We will use XGBoost as a regression technique. XGBoost is an ensemble tree method and was introduced for better speed and performance. XGBoost uses regularization, which helps to reduce overfitting as it discourages learning a more complex model (Edureka, 2020). Additionally, XGBoost has an in-built routine for handling missing values in the data. Employing XGBoost for stock market prediction is still in its infancy compared to other tree-based algorithms. However, its potential is deemed promising. A study conducted by Gumelar et al. (2020) about stock market prediction using XGBoost and Long Short-Term Memory (LSTM) found that XGBoost had the best performance.

The Python package used is XGBoost. To evaluate the performance of XGBoost, RMSE is used as this is a standard evaluation metric used in literature. It should be noted that XGBoost has a substantial amount of hyperparameters compared to other machine learning techniques. We will discuss the most commonly configured hyperparameters.

"N_estimators" represents the number of trees in the ensemble, which can be increased until no further improvements are seen. The default is set to 100. Most gradient boosting techniques in python are configured with a relatively small default setting for the number of trees. The rationale behind this is that in most cases, adding trees beyond a certain limit does not improve the performance of the model. This is caused by the way the boosted tree is constructed, which is sequential, where each new tree attempts the model and corrects for the errors made by the sequence of previous trees (Brownlee, 2016). This causes the model to quickly reach a point of diminishing returns. Thus, the default setting of 100 is considered sufficient in most cases.

"Max_depth" is the maximum depth of each tree, which is set to 6 for default. Generally, shallow trees perform poorly as they capture fewer details of the problem. Consequently, these are referred to as weak learners (Brownlee, 2016). Deeper trees run the risk of capturing too many details of the problem, which could easily lead to overfitting on the training set. This limits its ability to make good predictions on newly seen data.

"Eta" is the learning rate used to weight each model, which is often set to small values such as 0.3, 0.1, or smaller. The default value is set to 0.3. Eta controls the magnitude of change permitted from one tree to the next (XGBoost, 2021). Generally, the lower eta is set, the easier the optimum is reached. The downside of a lower eta is that it makes computation slower, as more input rounds are needed.

"Subsample" is the number of samples (rows) used in each tree, set to a value between 0 and 1. A value of 1.0 is common, as this ensures all samples are used. The default value is set to 1. Subsample denotes the fraction of observations that have to be randomly sampled for each tree. Generally, lower values prevent overfitting as it makes the algorithm more conservative. However, the risk of underfitting emerges when the values are set too small (XGBoost, 2021). In contrast, "Colsample_bytree" is the number of features (columns) used in each tree, set to a value between 0 and 1. This value is often set to 1.0 to ensure all samples are used. The default value is 1. This parameter denotes what percentage of the features, e.g. columns, will be used for building each tree (Analytics Vidhya, 2020).

3.4 Baseline

A baseline is often described as a simple model that provides reasonable results and does not require much expertise to be built (Li, 2020). In machine learning, we want our models to outperform the baselines we selected. The baseline that we used is for the RF, SVM and XGBoost is "mean". This means that we are always predicting the mean of the training set.

3.5 Evaluation metrics

Evaluation metrics are a crucial part of machine learning. The performance of a trained model is important, as it show cast how well the model can generalize to unseen data. This performance is measured through evaluation metrics. Improper evaluation of the trained model can lead to non-adaptive machine learning models.

We used RMSE as evaluation technique. RMSE give relatively high weight to large errors, which is preferred in this study (Medium, 2016). Additionally, RMSE reflects performance well when dealing with large error values, and RMSE is useful when lower residual values are preferred.

We will calculate the RMSE for the different models trained on the company and CEO dataset. Every dataset will yield 3 different RMSE scores per model, one for the closest stock price after the tweet is posted, up until 2 hours later. This allows for easier comparison across models and across the 3 different hours.

4. Experimental setup

In this chapter, we will describe our dataset in detail and discuss the experimental procedure that we followed to derive at our final results.

4.1 Data limitations

In this section, we will discuss the limitations in our data. We will first discuss the limitations with data gathering, and subsequently the limitations for the tweets.

4.1.1 Limitations with data gathering

Tweets are gathered from July 5th 2019 until January 30th 2020. As the corona pandemic resulted in a higher fluctuation in stock prices for several stock market indexes (Ngwakwe, 2020), it may become a confounding factor in our research if not accounted for. The corona crisis was officially declared as a pandemic on January 30th, 2020. Thus, half of the year 2019 is analyzed, as this is the year before the corona pandemic and may provide more accurate and generalizable results. The Python library yfinance only allows one to access the stock prices for 1-hour intervals up till 730 days ago. When choosing to analyze the stock prices per minute, this can only be done for the last 7 days. Consequently, we only analyze the data from July 5th 2019 until January 30th 2020.

4.1.2 Limitations for the tweets

There we solely 7 CEOs who met the requirements of having a Twitter account that could be verified through either Twitter or the company's website, and met the requirement of being a regular Twitte r user. Of the 500 companies, 339 had a verified Twitter account or an account we could verify throu gh the company's website, and met the definition of being a regular Twitter user. This resulted in a la rger dataset than the CEOs dataset. See Appendix A for the list of companies and CEOs that were incl uded in this research.

4.2 Data description

In this section, we will describe our final datasets for our study.

4.2.1 Twitter data

The replies are filtered out, as these often do not contain relevant information. After fetching the tweets, a data frame with 36 columns is created. These were the time zone, conversation id, user id, date, username, tweet content, language, like count, retweet count, replies count etc. To perform the analysis and match the tweets to the right stock prices later on, we only need the username, time zone, date, time, and the tweet content. The other information is deemed irrelevant. There are a total of 76.015 tweets for the companies, and 1262 for the CEOs.

4.2.2 Stock data

The stock prices are gathered through yfinance. There are in total 7 one-hour time slots for every trading day. There were a total of 144 trading days from July 5th 2019 until January 30th 2020, and 1002 one-hour time slots. There are in total 8 columns when fetching the data through yfinance, which are: datetime, open, high, low, close, volume, dividends, and stock splits. Only the 'datetime', 'close', and 'open' are relevant to match the stock data to the tweets. Additionally, we added the columns 'stockprice_before' which refers to the nearest closing price before a tweet was posted when posted outside of opening hours of the stock market. In contrast, the 'stockprice_before' refers to the nearest opening price before a tweet is posted when the tweet is posted during opening hours of the stock market. This is the stock price to which we will compare the next stock prices to. The column 'label_open' is the relative change between the stock price of the closest opening hour after a tweet is posted, and the 'stockprice_before'. The column 'label_1hr' refers to the relative change between the stock price of the second closest opening hour after a tweet is posted, and the 'stockprice_before', and label_2hr to the third.

4.2.3 Sentiment scores

In this section, we will discuss the steps taken to retrieve the sentiment scores for the tweets. We will start with the basic pre-processing of the tweets, followed by a short explanation of how we represented the classified tweets.

4.2.3a Pre-processing of the tweets

We discussed this in section 3.2.1 'pre-processing'. We removed the HTTPS tags and white spaces by applying a function in Python over the tweets in our datasets.

4.2.3b Sentiment classification

We applied a function over the tweets in the dataset to get their polarity score, which ranges from -1 to 1. We created an extra column 'Polarity', which includes all the polarity scores of each tweet. This column is necessary to perform the regression. Table 2 below illustrates an example of how the tweets are classified.

tweet	Polarity
Congratulations to the team in India for its number 2 ranking. You continue to set the bar in excellence for #PoweringProsperity around the world.	0.000000
Continued education is critical whether just starting out or at the top of your game. Check out how 2 of our engineers were introduced to the programming language #Kotlin and how they are now leading the Learning Community at .	0.025000
I'm inspired every day by the women in my life. I'm committed to supporting not only women at but believe in creating a culture where everyone is welcome and can bring their best self to work.	0.512500
Honored to be recognized by as the Best CEO for Diversity. I'm proud to advance #diversityandinclusion because at , diversity isn't something we do, it's part of who we are.	0.900000
At , we bring our whole self to work. Inclusivity is more than just a seat at the table – it's about integrating & valuing everyone's voice. Understanding these 5 traits can help you play a bigger role in fostering an inclusive workplace.	0.233333
We are so excited to welcome summer interns to this week! I can't wait to learn with them and see all they accomplish over the next few months.	0.293750
Like she said. Best marketing team on the planet	1.000000
Innovation is at the core of everything we do at . Not only does #innovation drive better outcomes for customers, but it also drives engagement and performance of employees.	0.250000
We are excited to welcome to the family. This acquisition helps us get us closer to our goals of advancing our #AI driven expert platform strategy by enabling us to help customers manage finances with more confidence, ease & convenience. #PoweringProsperity	0.558333
Thank you to all our employees and customers for achieving this exciting milestone. Having the privilege to be part of the lives of this many small businesses is a true honor	0.225000

Table 2: Example of classification of tweets

4.3 Experimental procedure

In this section, we will discuss the experimental procedure for the regression tasks for SVM, RF and XGBoost. The experimental procedure is the similar for both the CEOs dataset and the companies dataset.

4.3.1 Machine learning implementation

The Python library sklearn was used to perform the analyses for SVM and RF. Sklearn has an implementation for both SVM and RF to perform regression tasks, namely SVR and RandomForestRegressor. For XGBoost, the Python library XGBoost was used with the implementation XGBRegressor. For all the models, 80% of the data was reserved for training, and 20% for testing. The 3 different labels label_open, label_1hr and label_2hr were regressed on the polarity values. The dependent variables are label_open, label_1hr and label_2hr. The independent variable is polarity.

Tuning of the parameters was deemed unnecessary after an analysis of the results, which will be discussed in chapter 5. For the baseline model, the error and prediction is also calculated. For the models, we calculated the RMSE to evaluate the performance, the mean to evaluate which values the model predicts, and the standard deviation to evaluate to what extent the predictions deviate from the mean.

5. Results

In this chapter, we will report our results for RF, SVM and the XGBoost from immediately after a tweet is posted up till two hours later. We will start with the results for the CEOs, and then we will discuss the results for the companies.

5.1 CEOs

In this section, we will report and analyze the results for the RF, SVM and XGBoost on our dataset of the CEOs for immediately after a tweet is posted, up till two hours later. This will allow us to answer the first research question: *"To what extent do CEOs influence the stock price depending on the sentiment of their tweets?".*

5.1.1 Baseline predictions and errors

We use the mean as our baseline for the stock change immediately after a tweet is posted, up till two hours later. We name them our baseline predictions, as these are the values that the baseline predicts. We labelled them with label_open, label_1hr, label_2hr for clarity. The baseline predictions are reported in table 3 below.

Label	Baseline prediction
Open	0.001
1hr	0.0004
2hr	0.001

Table 3

These values imply that the baseline predicts 0.10% as the mean change for the stock price immediately after a tweet is posted, 0.04% as the mean change for the stock price one hour after a tweet is posted, and 0.10% as the mean change for the stock price two hours after a tweet is posted.

In table 4 below we see the RMSE for the baseline and our three machine learning models. The goal in machine learning is for our models to perform better than the baseline.

	Baseline error	RMSE random forest	RMSE SVM	RMSE XGBoost
Label_open	0.017	0.022	0.037	0.022
Label_1hr	0.016	0.019	0.018	0.019
Label_2hr	0.019	0.021	0.020	0.020

Table 4

5.2.2 Analysis of the results

In table 4 it is illustrated that SVM has almost exactly the same error as the baseline for label_1hr and label_2hr. Subsequently, RF and XGBoost have an error that is very close to the baseline error, especially for label_1hr and label_2hr. We can safely conclude that our models did not learn anything from our data. To establish whether it is possible to train any model on our dataset, we need to look at if we can find any correlations in our data. Figure 1 and 2 present the relationships between our polarity values and relative change of the stock price for immediately after a tweet is posted, and one hour later. The x-axis represents the polarity values, and the y-axis the relative changes in stock price.

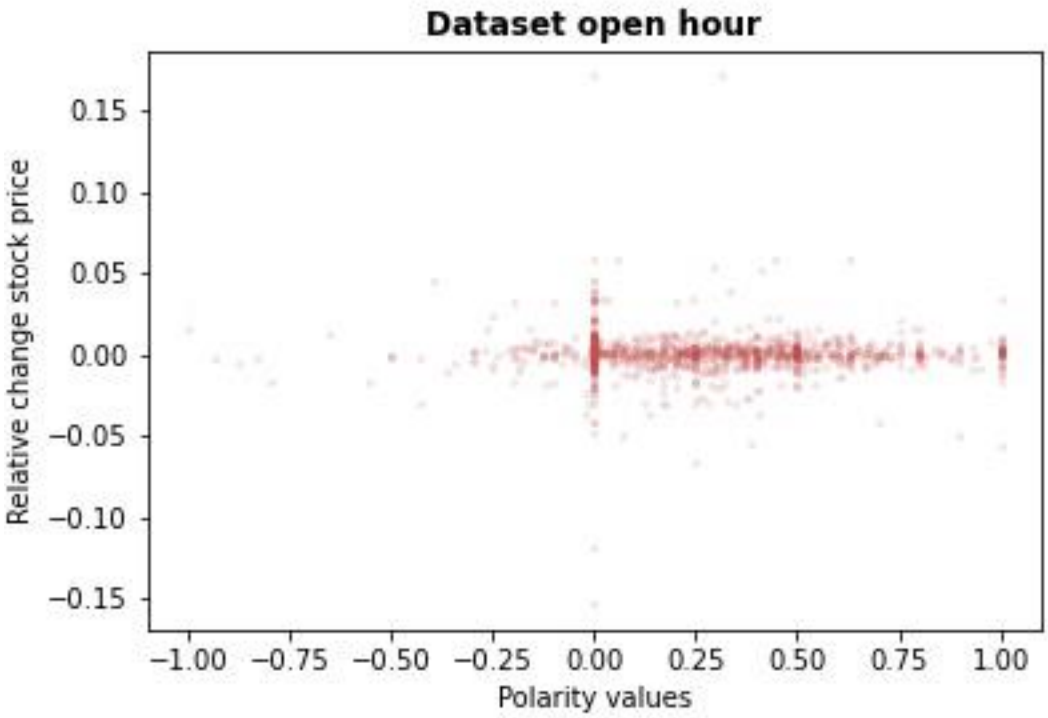


Figure 1: dataset for label_open

We can see a very small correlation between the polarity values and relative changes in stock prices immediately after a tweet is posted. We can see relatively many data values for polarity 0.

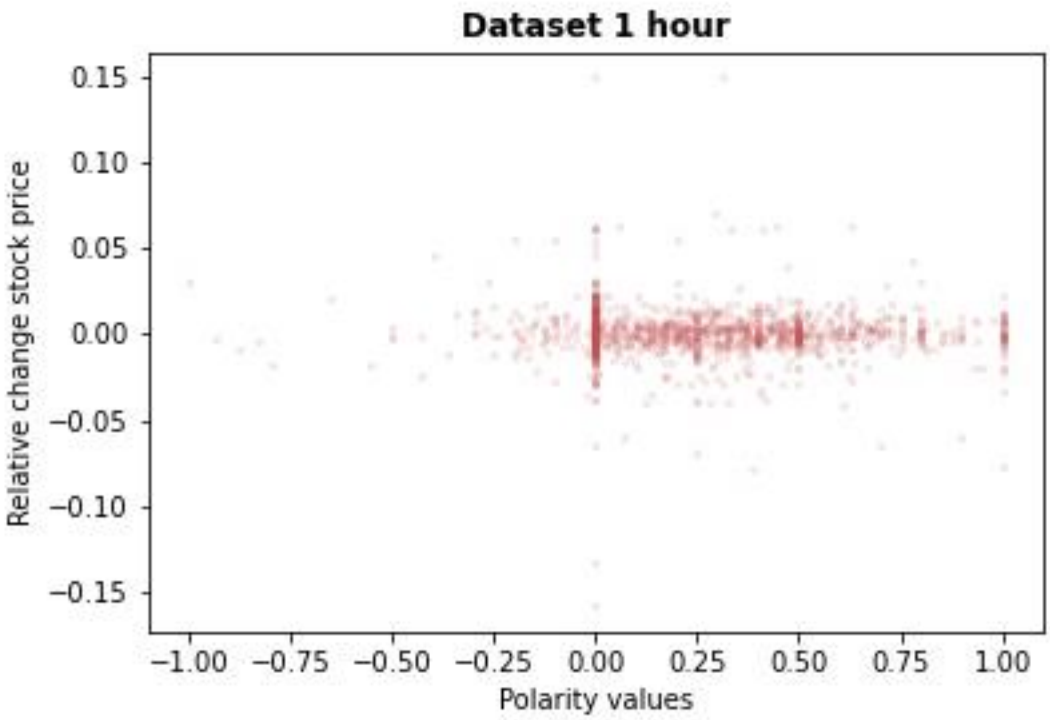


Figure 2: dataset for label_1hr

Subsequently, for the dataset of 1 hour after a tweet is posted we see little correlation, but many polarity values with a value of 0. The same pattern can be found for our dataset of 2 hours after a tweet is posted, see appendix B. Based on the scatterplot analysis we can state that there is no correlation between polarity and changes in stock price for any of our three labels. Hence, we do not expect to find a better model than the baseline.

Another observation from figure 1 and 2 is that there are relatively many values around 0, but these values do not indicate a clear increase or decrease and are spread between 5% increase in stock price and -5% decrease in stock price. The relatively high density around 0 polarity and no clear direction in stock price change for this polarity value of 0 indicates that it will be relatively difficult to train any model on this data. Additionally, we can see that the dataset of the CEOs has almost no negative tweets, which adds to the difficulty of training a model on this data.

To gain a deeper understanding of the performance of our model compared to the baseline, table 5 gives an overview of the error, mean, and standard deviation of our 3 machine learning models.

	Error	Mean	Standard deviation
Random forest			
Label_open	0.022	0.001	0.004
Label_1hr	0.019	0.001	0.006
Label_2hr	0.021	0.001	0.007

	Error	Mean	Standard deviation
SVM			
Label_open	0.037	0.030	0.008
Label_1hr	0.018	-0.004	8.673
Label_2hr	0.020	-0.004	8.674

	Error	Mean	Standard deviation
XGBoost			
Label_open	0.022	0.001	0.003
Label_1hr	0.019	0.001	0.007
Label_2hr	0.020	0.001	0.007

Table 5

Table 5 illustrates that the RF and XGBoost have an error close to the baseline for label_1hr and label_2hr. However, the standard deviations are not zero, which implies that there is some variation from the mean, thus our model is not an exact copy of the baseline.

In figure 3,4 and 5 below we illustrate the results for the predictions of our model RF, SVM, and XGBoost for the label_open.

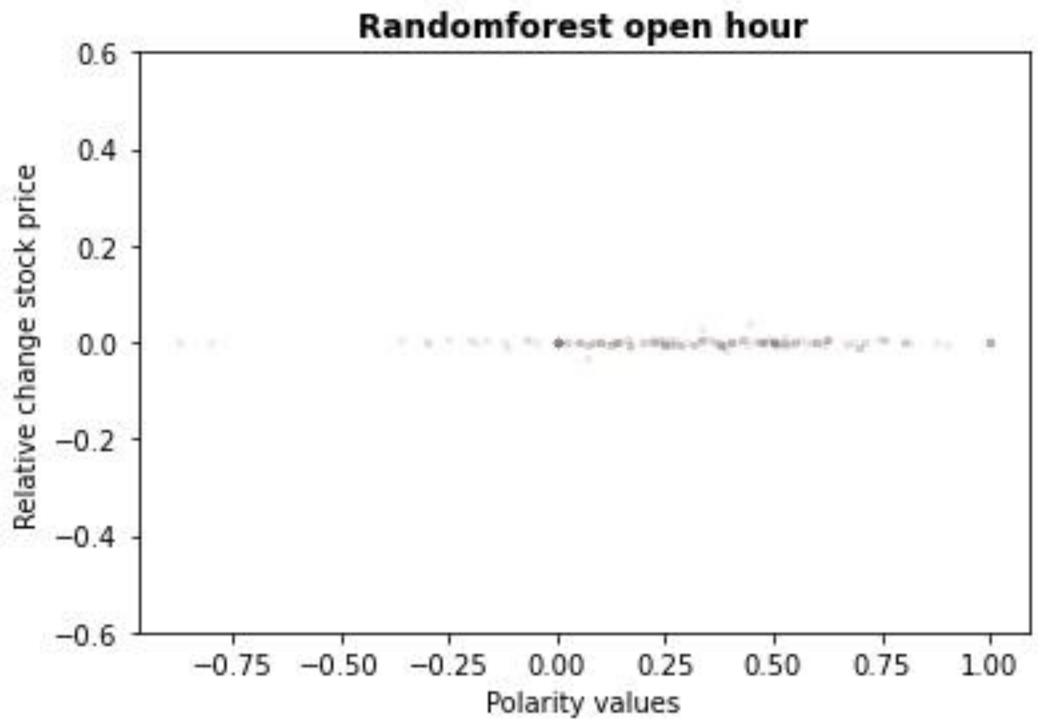


Figure 3: Random forest label_open

We see that the model is almost exactly predicting the baseline, and there seems to be very little to no variation. This means we can safely conclude that the RF did not learn anything from our data. The results for label_1hr and label_2hr are almost identical and can be found in the appendix B.

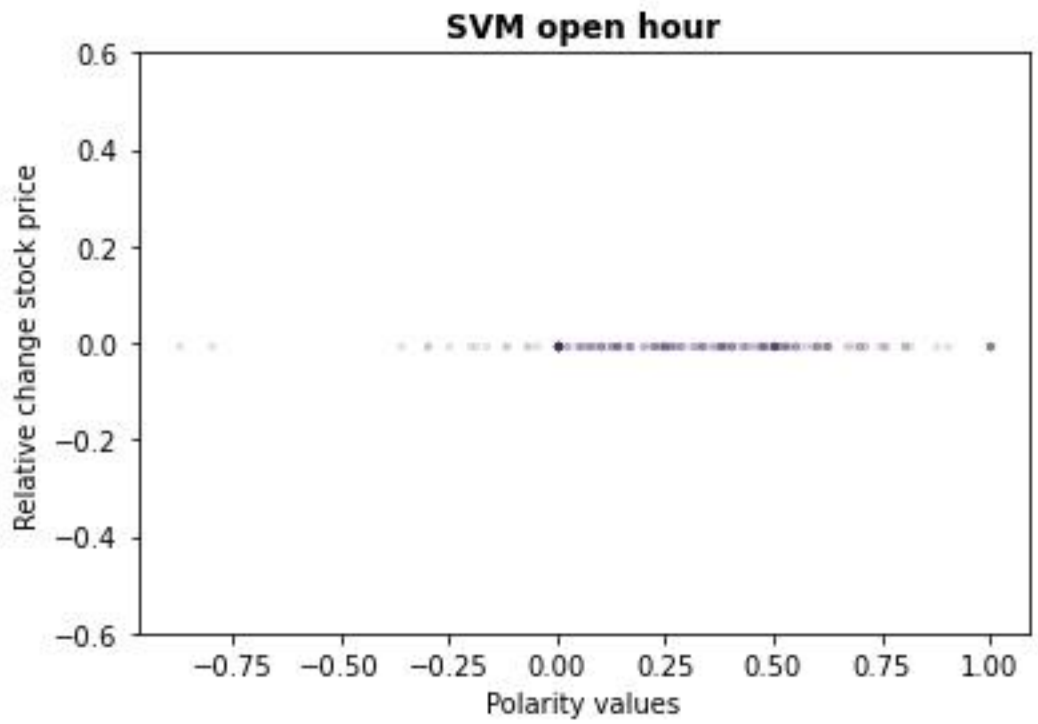


Figure 4: SVM label_open

Figure 4 shows that SVM is a complete straight line, which means we are almost exactly predicting the baseline. Our SVM also did not learn anything. The results for label_1hr and label_2hr are almost identical and can be found in the appendix B.

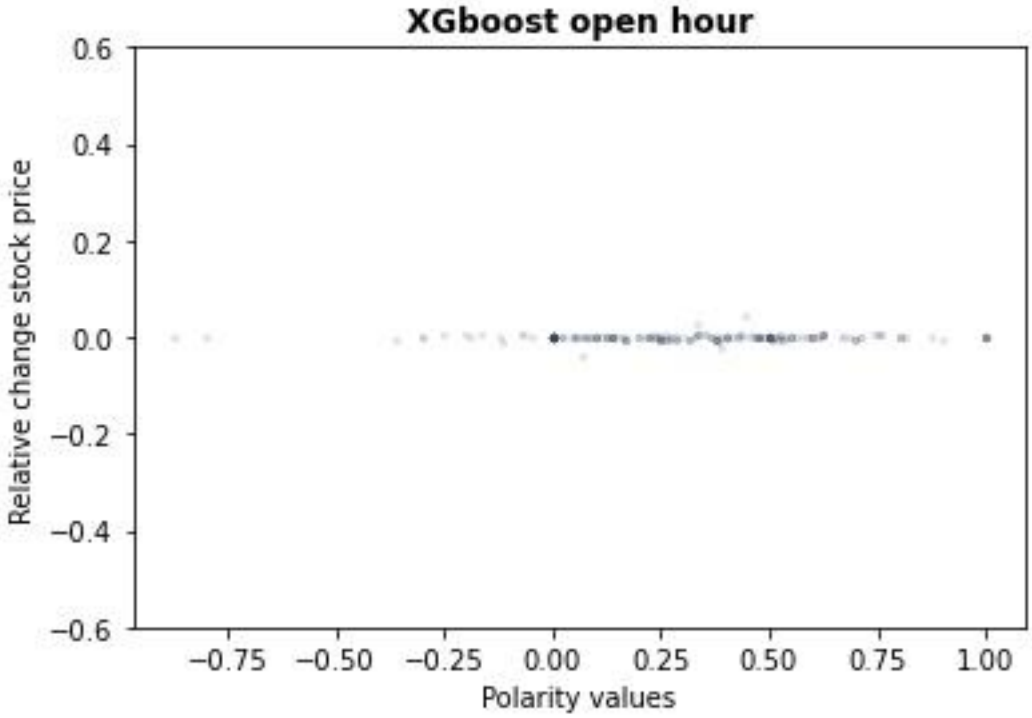


Figure 5: XGBoost label_open

Finally, figure 5 displays the XGBoost. We see slightly more variation compared to the SVM, but we are again almost exactly predicting the baseline. XGBoost also did not learn anything. The results for label_1hr and label_2hr are almost identical and can be found in the appendix B.

The high standard deviation values for the SVM for label_1hr and label_2hr indicate that there are large deviations from the mean values the model predicts.

5.2.3 Conclusion

Our models do not perform much better than the baseline, which indicates that neither of our three models has learned anything from the data.

5.2 Companies

In this section, we will report and analyze the results for the RF, SVM and XGBoost on our dataset of the companies for immediately after a tweet is posted, up till two hours later. This will allow us to answer the second research question: *"To what extent do corporates influence their stock price depending on the sentiment of their tweets?"*.

5.2.1 Baseline predictions and errors

We use the mean as our baseline for the stock change immediately after a tweet is posted, up till two hours later. We name them our baseline predictions, as these are the values that the baseline predicts. We labelled them with label_open, label_1hr, label_2hr for clarity. The baseline predictions are reported in table 6 below.

Label	**Baseline prediction**
Open	0.0011
1hr	0.0011
2hr	0.0014

Table 6

These values imply that the baseline predicts 0.11% as the mean change for the stock price immediately after a tweet is posted, 0.11% as the mean change for the stock price one hour after a tweet is posted, and 0.14% as the mean change for the stock price two hours after a tweet is posted.

In table 7 below we see the RMSE for the baseline and our three machine learning models. The goal in machine learning is for our models to perform better than the baseline.

	Baseline error	RMSE random forest	RMSE SVM	RMSE XGBoost
Label_open	0.018	0.018	0.025	0.018
Label_1hr	0.019	0.019	0.023	0.019
Label_2hr	0.021	0.022	0.027	0.021

Table 7

5.2.2 Analysis of the results

The error results in table 7 indicate that random forest and XGBoost have the same error as the baseline. This implies that they both constantly predict the mean for every value, which means the model performance is not better than the baseline and subsequently did not learn anything from our data. To determine whether this result is caused by improper models used or the way our data is constructed, we need to take a closer look at our data to see if we can find correlations between the polarity values and the changes in stock prices for our three different labels. Figure 6 and 7 present the relationships between our polarity values and relative change of the stock price for our label_open and label_1hr. The x-axis represent the polarity values, and the y-axis the relative changes in stock price.

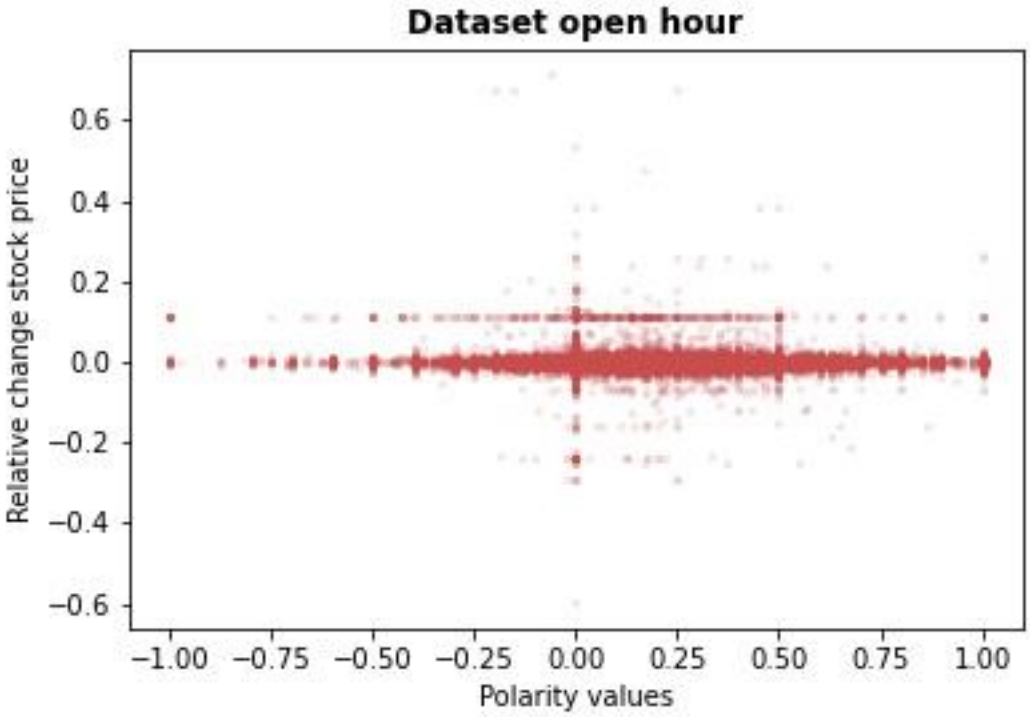


Figure 6: dataset for label_open

We can see that there is very little correlation between polarity and the relative changes in the stock prices for the opening price immediately after a tweet is posted. We can also see that most of the data is around the 0.0 point, which shows most of the data is close to the baseline prediction of 0.11%. We do have a few outliers that reach 60% relative change, but these are rare.

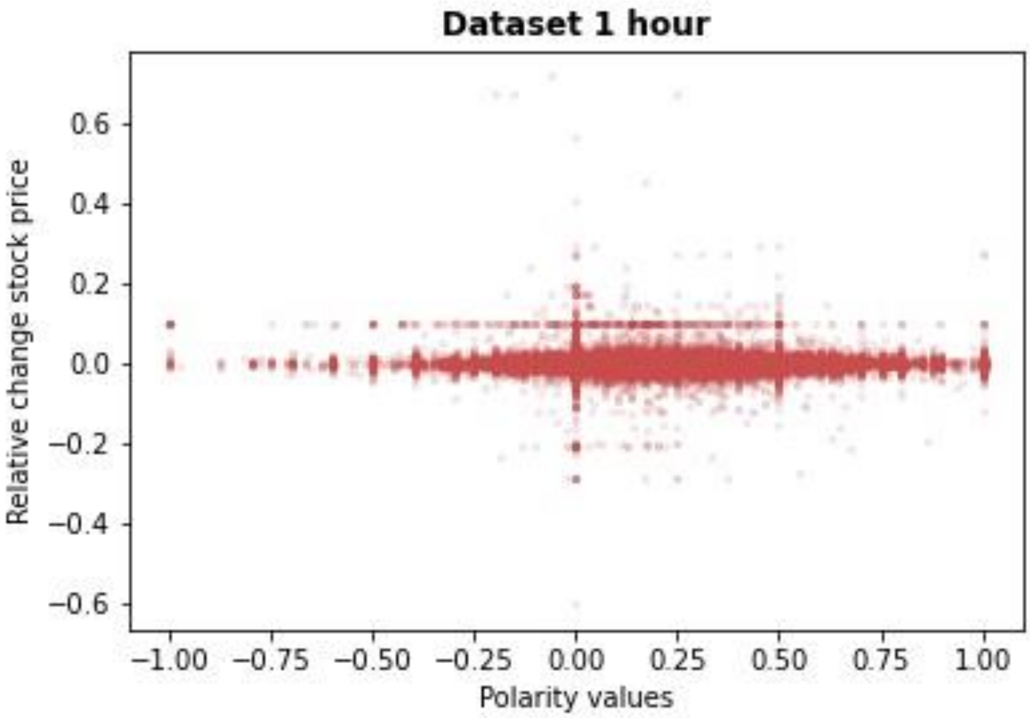


Figure 7: data set for label_1hr

In figure 7 we see an almost identical scatterplot as for the label_open. We see very little to no correlation between the polarity and changes in stock prices for one hour after a tweet is posted. For label_2hr {see appendix B}, we see again no correlation between percentage changes of stock prices and the polarity values. Based on our scatterplot analysis, we can confidently state that we see no correlation between polarity and changes in stock price for any of our three labels. Hence, we do not expect to get a better model than the baseline. Hyperparameter tuning will likely result in a model that predicts exactly the baseline.

	Error	Mean	Standard deviation
Random forest			
Label_open	0.018	0.0011	0.004
Label_1hr	0.0198	0.001	0.004
Label_2hr	0.022	0.001	0.004

	Error	Mean	Standard deviation
SVM			
Label_open	0.025	0.017	0.008
Label_1hr	0.023	0.006	0.011
Label_2hr	0.027	0.030	0.008

	Error	Mean	Standard deviation
XGBoost			
Label_open	0.18	0.001	0.003
Label_1hr	0.019	0.001	0.003
Label_2hr	0.022	0.001	0.003

Table 8

Our model is not exactly predicting the baseline, as we still have some variation in our data, which is illustrated in table 8. Our means for XGBoost and RF are exactly the same as the baseline, but the standard deviation is not exactly zero, which implies that there is some variation from the mean, thus our model is not exactly a copy of the baseline. For example, the baseline predicts 0.0011 for our label_open, but our model does not always predict 0.0011. Figure 8 and 9 illustrate this.

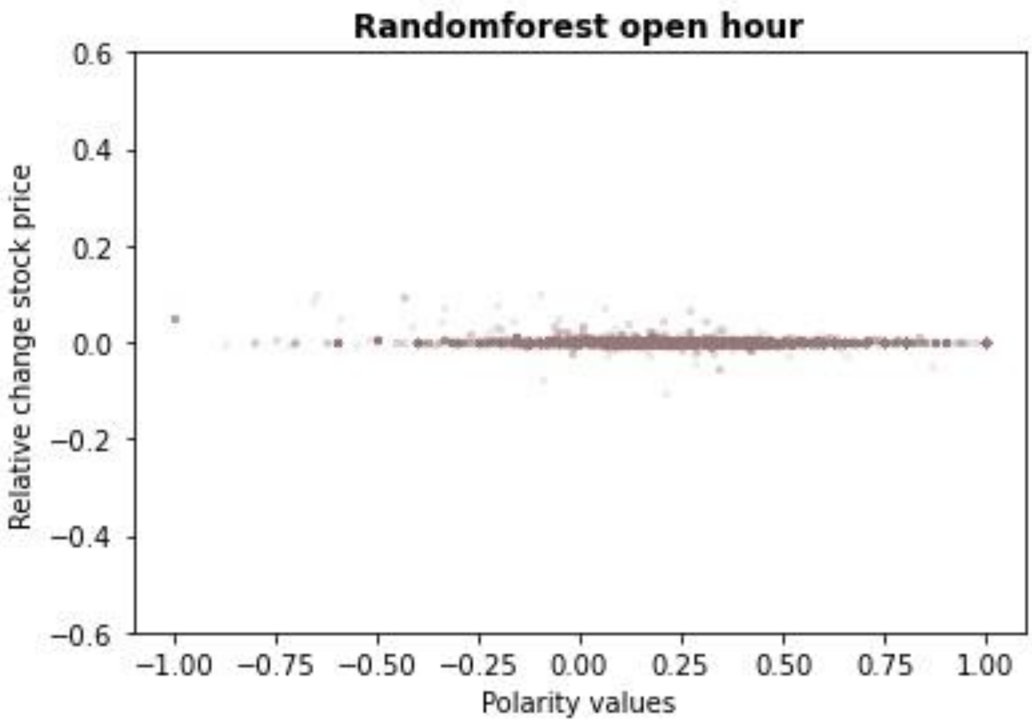


Figure 8: randomforest label_open

We see that the predictions of the random forest are very close to the baseline of 0.0011, but not exactly. There are some deviations from the baseline.

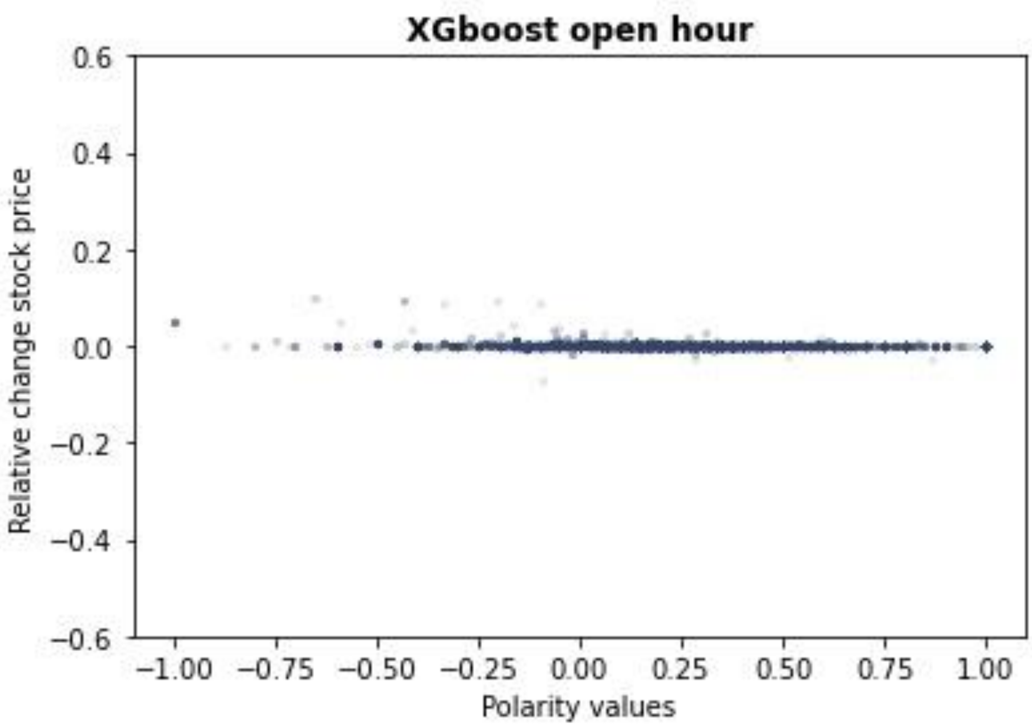


Figure 9: XGBoost label_open

For the XGBoost we see a similar result as for the random forest. The predictions are close to the baseline of 0.0011, but we do see some deviations. In figure 10 below, we see the predictions of the SVm for the label_open. Again, we see predictions close to the baseline of 0.0011 but with deviations from it.

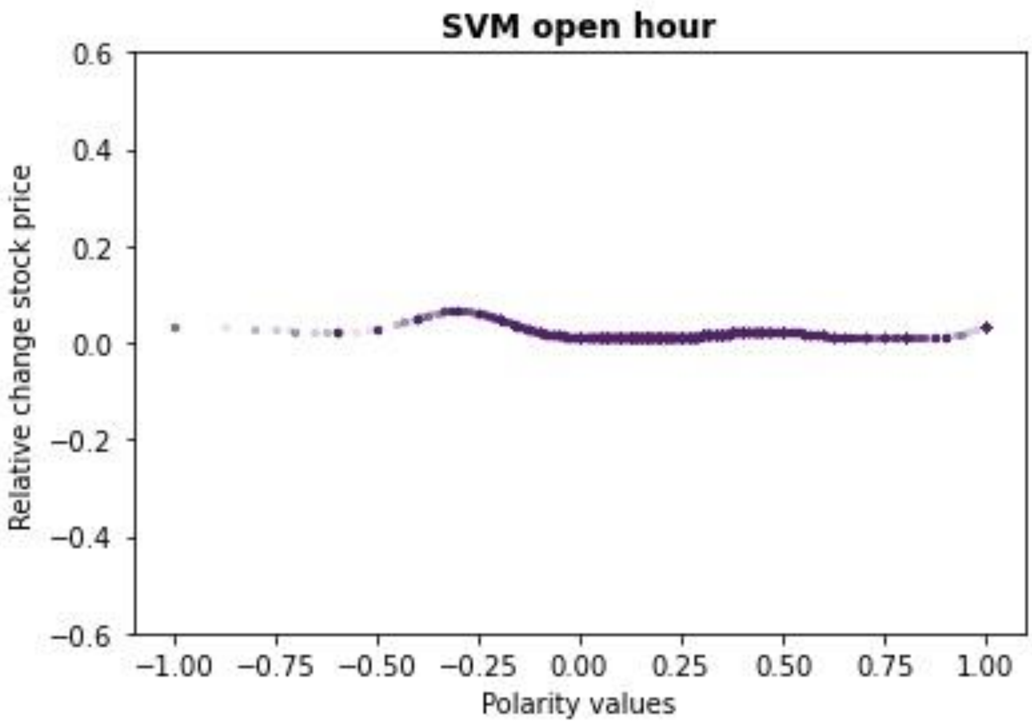


Figure 10: SVM label_open

For SVM we do see a different mean for test set output. For example, label_open for SVM has a mean of 0.017 while the baseline has a mean of 0.0011. However, SVM also gave us the highest error compared to RF and XGBoost so this is the worst model. We could potentially tune SVM, but it would then do just as worse as XGBoost and RF because it is simply going to give us the mean scores.

5.2.3 Conclusion

The predictions on our test set for the different models leads us to the following conclusion. The models deviate slightly from the baseline because we do not see a completely straight line in most graphs. However, we do find that the errors especially for XGBoost are exactly the same as the baseline. The slight variation between the predictions of the model and the baseline is likely solely due to the fact that the model tries to get the same prediction as the training data, but we do not expect to find better results for individual companies.

6. Discussion

The main goal of this thesis is to examine the extent to which CEOs and companies can influence the price of their stocks based on the sentiment of their tweets immediately after a tweet is posted, up till 2 hours later. The datasets used, one for the CEOs and one for the companies, contain the tweets and stock information from the period July 5th 2019 until January 30th 2020. We performed regression on both datasets, using the machine learning algorithms SVM, XGBoost, and Random Forest. Prior research suggested that tree-based models can outperform traditional regression techniques used for stock market prediction as SVM. In our research, XGBoost and RF generally had a lower error compared to SVM for both the companies dataset and the CEOs dataset. Thus, our research confirms that tree-based models indeed perform slightly better. However, neither of the 3 models used can predict the stock price based on the sentiment of tweets. There are different explanations for this. Firstly, there does not seem to be any correlation between polarity and percentage change in stock prices. This indicates that it will be challenging to train any model on this data. Secondly, the datasets are highly imbalanced especially for the CEOs, as the tweets are classified as either neutral or positive; negative polarity values rarely occur. A possible explanation for this imbalance is that companies and CEOs may always publish neutral or positive news, as publishing negative news could negatively impact their stock prices. Thus, posting highly negative tweets may in fact negatively impact the stock price, but these tweets are uncommon so we are unable to prove that. The result of this imbalance is that it makes it challenging to train a regression model on the data.

The results of this study seem to deviate from those obtained by other studies which investigated the correlation between stock prices and Twitter sentiment. A study by Kordonis et al. (2016) achieved a 87% accuracy concerning correct stock movement prediction and found a correlation between stock price and polarity scores. Other studies report similar results. There are several explanations for our contrasting results. Previous studies have focused mainly on public sentiment towards specific companies. This leads to a larger pool of data and more variety in the sentiment. The tweets posted by CEOs and companies are mainly positive, which explains the challenge in training a machine learning model. In contrast, tweets posted by the public contain a variety of sentiments as the majority of individuals on Twitter do not have an incentive to solely post positive tweets about a company or stock.

A limitation of this research is the short time frame. Yfinance does not allow one to access tweets longer than 730 days ago for the one-hour interval, so integrating the stock data differently or solely analyzing the opening and closing prices per 24 hours gives access to a larger pool of stock data, which also allows one to extend this research. The corona pandemic also limited the availability of data for our research, as the stock market was relatively more volatile during the pandemic, which could lead to biased results.

7. Conclusion

We can conclude that neither CEOs nor companies influence their stock price based on the sentiment of their tweets for the closest hour after a tweet is posted, up till two hours later based on the data of July 5th 2019 until January 30th 2020. Our models XGBoost and RF tend to predict close to the baseline for the company dataset. The SVM performs the worst of all our models on the company dataset as it has the highest error. For the CEOs dataset, XGBoost and RF tend to predict close to the baseline for label_open and label_2hr. SVM performs the worst of all our models on the CEOs dataset. Based on the results, we can confidently state that it will be challenging to find any model that can predict the stock price based on the sentiment of the tweets for CEOs and companies. Thus, we can answer our research questions as follows:

R1: "To what extent do CEOs influence the stock price depending on the sentiment of their tweets?"

The results indicate that the sentiment of a tweet posted by CEOs does not influence the 3 nearest stock prices.

R2: "To what extent do corporates influence their stock price depending on the sentiment of their tweets?"

The results indicate that the sentiment of a tweet posted by corporates does not influence the 3 nearest stock prices.

The results of this study indicate that CEOs and companies on the Fortune 500 do not seem to influence their stock prices, regardless of how positive their tweets are. This implicates that companies and CEOs are unlikely to manipulate the stock market through the sentiment of their tweets. This could be valuable for regulators investigating stock market manipulation by companies. Due to the time frame limitations encountered in this study, it is potentially worthwhile for future research to investigate a longer time frame. Additionally, investigating the influence of the tweets/posts of stock-related Twitter or Instagram accounts on the price of stocks could also be promising, as there are a variety of accounts on Twitter that give investment advice or their opinion on stocks.

Subsequently, future research could explore whether adding an additional feature, such as subjectivity, which indicates how much factual information versus personal opinion is expressed in the tweet, will improve the performance of the models. Additionally, adding polarity values for CEOs' and companies' other social media platforms such as Instagram and Facebook may yield different results. Regression often performs better when an additional feature is added, so it is worthwhile to add additional features to determine whether the models perform better. Thus, our study paves the way for future research in this study area.

Bibliography

Business Insider. (2021, February 11). *The Reddit-fueled GameStop rally is reportedly under federal investigation for possible market manipulation - and Robinhood has been subpoenaed* . Retrieved from Business Insider: https://markets.businessinsider.com/news/stocks/reddit-gamestop-stock-rally-investigation-market-manipulation-robinhood-regulation-gme-2021-2-1030074397

Ranco , G., Aleksovski , D., Calda, G., Grčar, M., & Mozetič, I. (2015). *The Effects of Twitter Sentiment on Stock Price Returns.* IMT Institute for Advanced Studies.

Tabari, N., Seyeditabari, A., Praneeth, B., Hadzikadic, M., Biswas, P., & Zadrozny, W. (2018). *Causality Analysis of Twitter Sentiments and Stock Market Returns.* Melbourne: Association for Computational Linguistics.

Dickinson, B., & Hu, W. (2015). Sentiment Analysis of Investor Opinions onTwitter. *Scientific Research Publishing*.

Rao, T., & Srivastava, S. (2015). Twitter Sentiment Analysis: How To Hedge Your Bets In The Stock Markets. *Netaji Subhas Institute of Technology*.

Mohan, S., Mullapudi, S., Sammeta, S., Vijayvergia, P., & Anastasiu, D. C. (2019). Stock Price Prediction Using News Sentiment Analysis. *IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*.

Fortune. (n.d.). *World's Most Admired Companies*. Retrieved from https://fortune.com/worlds-most-admired-companies/

Ngwakwe, C. C. (2020). *Effect of COVID-19 Pandemic on Global Stock Market Values: A Differential Analysis.* ŒCONOMICA.

LIWC. (n.d.). *Discover LIWC2015*. Retrieved from http://liwc.wpengine.com/

Phienthrakul, T., Kijsirikul, B., Takamura, H., & Okumura, M. (2009). *C.S. Leung, M. Lee, and J.H. Chan (Eds.): ICONIP 2009, Part II, LNCS 5864, pp. 583–592, 2009. © Springer-Verlag Berlin Heidelberg 2009 Sentiment Classification with Support Vector Machines and Multiple Kernel Functions.* Department of Computer Engineering, Faculty of Engineering, Mahidol University.

Wang, S., & Manning, C. D. (2012). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification.

Jurafsky, D., & Martin, J. H. (2020). *Speech and Language Processing.*

Twitter. (n.d.). *Inactive account policy* . Retrieved from https://help.twitter.com/en/rules-and-policies/inactive-twitter-accounts

Borovkova, S., & Xiaobo, D. (2015). News Sentiment, Factor Models and Abnormal Stock Returns . *Vrije Universiteit Amsterdam*.

Alanyali, M., Moat , H., & Preis , T. (2013). Quantifying the Relationship Between Financial News and the Stock Market. *Scientific Reports*.

Smailović, J., Grčar, M., Lavrač, N., & Žnidaršič, M. (2013). A. Holzinger and G. Pasi (Eds.): HCI-KDD 2013, LNCS 7947, pp. 77–88, 2013. © Springer-Verlag Berlin Heidelberg 2013 Predictive Sentiment Analysis of Tweets: A Stock Market Application . *Springer-Verlag Berlin Heidelberg*.

Chahine, S., & Malhotra, N. K. (2018). Impact of social media strategies on stock price: the case of Twitter . *European Journal of Marketing* .

Aggarwal, R. K., & Wu, G. (2013).

Nicoli, N. (2020). The battle to end fake news: A qualitative content analysis of Facebook announcements on how it combats disinformation.

Pineiro-Chousa, J., Gonzalez, M., & Pico, A. (2017). Influence of Social Media over the Stock Market.
Mittal, A., & Goel, A. (2011). Stock Prediction Using Twitter Sentiment Analysis. *Stanford University*.
Marin, V.-C. (2018). The Predictive Power of Social Sentiment Over Cryptocurrencies' Price Fluctuations. Causal Effects and Forecasts. *UNIVERSITY OF AMSTERDAM*.
Ren, R., & Liu, T. (2019). Forecasting Stock Market Movement Direction Using Sentiment Analysis and Support Vector Machine. *IEEE SYSTEMS JOURNAL*.
Khedr, A. E., Salama, S. E., & Yaseen, N. (2017). Predicting Stock Market Behavior using Data Mining Technique and News Sentiment Analysis.
Liagkouras, K., & Metaxiotis, K. (2020). Stock Market Forecasting by Using Support Vector Machines. In *Learning and Analytics in Intelligent Systems.*
Shah, P. (2020, June 27). *Sentiment Analysis using TextBlob* . Retrieved from https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524#:~:text=TextBlob%20is%20a%20simple%20library,classifying%20negative%20and%20positive%20words.
Swarnkar, N. (2020, May 21). *VADER Sentiment Analysis in Algorithmic Trading* . Retrieved from https://blog.quantinsti.com/vader-sentiment/
Hutto, C., & Gilbert, E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Association for the Advancement of Artificial Intelligence*.
Haddi, E., Liu, X., & Shi, Y. (2013). The Role of Text Pre-processing in Sentiment Analysis. *Procedia Computer Science*.
Zhou, M. (2019, July 23). *StopWords and Lexicon Normalization for Sentiment Analysis* . Retrieved from https://medium.com/data-science-blogs/stopwords-and-lexicon-normalization-for-sentiment-analysis-f9f10f0d4108
Twinword. (n.d.). *What Is Lemmatization?* Retrieved from https://www.twinword.com/blog/what-is-lemmatization/
Techslang. (n.d.). *What is Lemmatization?* . Retrieved from Techslang: https://www.techslang.com/definition/what-is-lemmatization/
Investigate ai. (n.d.). *Sentiment analysis tools*. Retrieved from Investigate ai: https://investigate.ai/investigating-sentiment-analysis/comparing-sentiment-analysis-tools/
Loria, S. (2020). *textblob Documentation.* Textblob.
Ahmad, M., Aftab, S., & Ali, I. (2017). Sentiment Analysis of Tweets using SVM. *International Journal of Computer Applications*.
Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. (2019). Predicting the Direction of Stock Market Prices Using Tree-Based Classifiers. *The North American Journal of Economics and Finance*.
Khan, W., Ghazanfar, M., Azam, M. A., Karami, A., Alyoubi, K. H., & Alfakeeh, A. (2020). Stock market prediction using machine learning classifiers and social media, news. *Journal of Ambient Intelligence and Humanized Computing* .
Lohrmann, C., & Luukka, P. (2019). Classification of intraday S&P500 returns with a Random Forest. *International Journal of Forecasting*.
Ampomah, E. K., Qin, Z., & Nyame, G. (2020). Evaluation of Tree-Based Ensemble Machine Learning Models in Predicting Stock Price Direction of Movement. *Information (Switzerland)*.
Ballings, M., & Van den Poel , D. (2015). Evaluating multiple classifiers for stock price direction prediction. *EXPERT SYSTEMS WITH APPLICATIONS* .

Mokoteli, T., Ramsumar, S., & Vadapalli, H. (2019). THE EFFICIENCY OF ENSEMBLE CLASSIFIERS IN PREDICTING THE JOHANNESBURG STOCK EXCHANGE ALL-SHARE INDEX DIRECTION. *Journal of Financial Management Markets and Institutions*.

Nti, I. K., Adekoya, A. F., & Weyori , B. A. (2020). A comprehensive evaluation of ensemble learning for stock-market prediction. *Journal of Big Data*.

Weng, B., Lu, L., Wang, X., Martinez, W., & Megahed, F. M. (2018). Predicting Short-Term Stock Prices using Ensemble Methods and Online Data Sources. *Expert Systems with Applications*.

Khaidem, L., Saha, S., & Dey, S. (2016). Predicting the direction of stock market prices using random forest. *Applied Mathematical Finance*.

Aznar, P. (2020). *Decision Trees: Gini vs Entropy*. Retrieved from Quantdare: https://quantdare.com/decision-trees-gini-vs-entropy/

Mantovani, R., Horváth, T., Cerri, R., Junior, S. B., Vanschoren, J., & de Carvalho, A. (2018). An empirical study on hyperparameter tuning of decision trees. *Cornell University*.

Mithrakumar, M. (2019, November 11). *How to tune a Decision Tree?* Retrieved from Towards Data Science: https://towardsdatascience.com/how-to-tune-a-decision-tree-f03721801680

Pant, A. (2019, January 22). *Introduction to Logistic Regression* . Retrieved from Towards Data Science: https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148#:~:text=Logistic%20regression%20is%20a%20classification,a%20discrete%20set%20of%20classes.&text=Logistic%20regression%20transforms%20its%20output,to%20return%20a%20probability

Brownlee, J. (2021, January 1). *Multinomial Logistic Regression With Python* . Retrieved from Machine learning mastery : https://machinelearningmastery.com/multinomial-logistic-regression-with-python/

sklearn. (2020). *Baselines sklearn*. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html

Brownlee, J. (2014, March 21). *Classification Accuracy is Not Enough: More Performance Measures You Can Use* . Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/

Machine learning mastery. (2014, March 21). *Classification Accuracy is Not Enough: More Performance Measures You Can Use*. Retrieved from machinelearningmastery: https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/

Young, J. (2021, May 21). *Market Index* . Retrieved from Investopedia: https://www.investopedia.com/terms/m/marketindex.asp

Corporate finance institute. (2021, January 1). *What is a Stock Market Index?* Retrieved from Corporate finance institute: https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/stock-market-index/

Erika. (2019, December 15). *Scraping Tweets off Twitter with TWINT*. Retrieved from Medium: https://medium.com/@erika.dauria/scraping-tweets-off-twitter-with-twint-a7e9d78415bf

Srivastava, T. (2019, August 6). *11 Important Model Evaluation Metrics for Machine Learning Everyone should know* . Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/

Medium. (2016, March 23). *MAE and RMSE — Which Metric is Better?* . Retrieved from Medium: https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

Li, S. (2020, August 31). *3 – Baselines*. Retrieved from ML CMU: https://blog.ml.cmu.edu/2020/08/31/3-baselines/

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2018). Stock price prediction using support vector regression on daily and up to the minute prices. *KE AI*.

Chakure, A. (2019, June 25). *Random Forest Regression* . Retrieved from Medium: https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f

Lee, A. (2019, April 21). *Why you should do Feature Engineering first, Hyperparameter Tuning second as a Data Scientist* . Retrieved from towardsdatascience: https://towardsdatascience.com/why-you-should-do-feature-engineering-first-hyperparameter-tuning-second-as-a-data-scientist-334be5eb276c

Weerts, H., & Vanschoren, J. (2020). Importance of Tuning Hyperparameters of Machine Learning Algorithms. *ResearchGate*.

Bao, Y., Lu, Y., & Zhang, J. (2004). Forecasting Stock Price by SVMs Regression. *Springer-Verlag Berlin Heidelberg*.

Statistics how to. (2020, March 1). *Bootstrap sample: definition, example*. Retrieved from Statistics how to : https://www.statisticshowto.com/bootstrap-sample/

Oshiro, T. M., Pere, P. S., & Baranauskas, J. A. (2012). How Many Trees in a Random Forest? *Department of Computer Science and Mathematics*.

scikit-learn. (2020, January 1). *scikit-learn*. Retrieved from scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

Saxena, S. (2020, March 12). *A Beginner's Guide to Random Forest Hyperparameter Tuning*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2020/03/beginners-guide-random-forest-hyperparameter-tuning/

Mantovani, R. G., Horváth, T., Cerri, R., Junior, S. B., Vanschoren, J., de Leon, A., & de Carvalho, F. (2018). An empirical study on hyperparameter tuning of decision trees . *Cornell University*.

Mithrakumar Mukesh. (2019, November 11). *How to tune a Decision Tree?* Retrieved from towardsdatascience: https://towardsdatascience.com/how-to-tune-a-decision-tree-f03721801680

Saini, B. (2020, December 26). *Hyperparameter Tuning of Support Vector Machine Using GridSearchCV* . Retrieved from Medium: https://medium.com/swlh/hyperparameter-tuning-of-support-vector-machine-using-gridsearchcv-4d17671d1ed2

Sreenivasa, S. (2020, October 12). *Radial Basis Function (RBF) Kernel: The Go-To Kernel* . Retrieved from towardsdatascience: https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a

scikit-learn. (2021, January 1). *sklearn.svm.SVR*. Retrieved from scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

Liu, Z., & Xu, H. (2013). Kernel Parameter Selection for Support Vector Machine Classification. *Journal of Algorithms & Computational Technology*.

scikit-learn. (2021, January 1). *RBF SVM parameters*. Retrieved from scikit-learn: https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

Achsan, B. M. (2019, December 10). *Support Vector Machine: Regression* . Retrieved from medium: https://medium.com/it-paragon/support-vector-machine-regression-cf65348b6345

Edureka. (2020, April 24). *How To Use Regularization in Machine Learning?* . Retrieved from Edureka: https://www.edureka.co/blog/regularization-in-machine-

learning/#:~:text=In%20the%20context%20of%20machine,flexible%20model%2C%20to%20prevent%20overfitting.
Gumelar, A. B., Nilowardono , S., Wibowo , A. T., Setyorini , H., Adi , D. P., & Widodo, A. (2020). Boosting the Accuracy of Stock Market Prediction using XGBoost and Long Short-Term Memory. *International Seminar on Application for Technology of Information and Communication* .
Brownlee. (2016, September 7). *How to Tune the Number and Size of Decision Trees with XGBoost in Python*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/tune-number-size-decision-trees-xgboost-python/
XGBoost. (2021, March 1). *XGBoost Parameters*. Retrieved from XGBoost: https://xgboost.readthedocs.io/en/latest/parameter.html
Analytics Vidhya. (2020, January 17). *XGBoost colsample_by* hyperparameters explained* . Retrieved from Medium: https://medium.com/analytics-vidhya/xgboost-colsample-by-hyperparameters-explained-6c0bac1bdc1
Kordonis, J., Arampatzis, A., & Symeonidis , S. (2016). Stock Price Forecasting via Sentiment Analysis on Twitter.

Appendix

A. List of companies and CEOs included in this research

Companies:

@Walmart	@CharterNewsroom	@MarriottIntl	@DominionEnergy	@Thrivent	@HDSupply
@exxonmobil	@AmericanExpress	@westerndigital	@ADP	@FrontierCorp	@Intuit
@amazon	@Nationwide	@Visa	@salesforce	@NOVGlobal	@NetApp
@UnitedHealthGrp	@BestBuy	@Lennar	@HenrySchein	@EversourceCorp	@Ingredion
@McKesson	@Merck	@Kohls	@newell_brands	@DICKS	@Zoetis
@CVSHealth	@Honeywell	@AECOM	@BJsWholesale	@Genworth	@Fiserv
@ATT	@united	@SYNNEX	@StateStreet	@FISGlobal	@roberthalf
@Healthcare_ABC	@TIAA	@PNCBank	@ViacomCBS	@yumbrands	@harleydavidson
@Chevron	@TysonFoods	@TheHartford	@ameriprise	@CaesarsEnt	@yumbrands
@Ford	@Oracle	@AltriaNews	@VFCorp	@AlaskaAir	@WilliamsSonoma
@GM	@Allstate	@BNYMellon	@Discover	@HIIndustries	@Navient
@Costco	@wfscorp	@FluorCorp	@edisonintl	@EXPD_Official	@WesternUnion
@WBA_Global	@tjmaxx	@Avnet	@ONEOK	@darden	@peabodyenergy
@jpmorgan	@conocophillips	@molinahealth	@MurphyUSA	@UnitedRentals	@LeviStraussCo
@Verizon	JohnDeere	@KCCorp	@BedBathBeyond	@libertyglobal	
@kroger	@Tech_Data	@tenethealth	@ConEdison	@erie_insurance	
@generalelectric	@Nike	@synchrony	@CSX	@FifthThird	
@FannieMae	@generaldynamics	@CarMax	@LKQCorp	@footlocker	
@Phillips66Co	@Exelon	@SherwinWilliams	@firstenergycorp	@ConagraBrands	
@ValeroEnergy	@3M	@Emerson_News	@lithiamotors	@TractorSupply	
@BankofAmerica	@abbvie	@XPOLogistics	@MGMResortsIntl	@AllianceData	
@Microsoft	@CapitalOne	@Applied4Tech	@nvidia	@Hersheys	
@HomeDepot	@progressive	@PGE4Me	@SempraEnergy	@JetBlue	
@Boeing	@CocaCola	@nexteraenergy	@xcelenergy	@cbrands	
@WellsFargo	@HPE	@CHRobinson	@nscorp	@QuestDX	
@Citi	@AbbottNews	@Gap	@Corning	@Activision	
@MarathonPetroCo	@MicronTech	@lincolnfingroup	@ExpediaGroup	@Weyerhaeuser	
@comcast	@Travelers	@DaVita	@autozone	@RaymondJames	
@anthemgame	@riteaid	@JLL	@grainger	@caseysgenstore	
@DellTech	@northropgrumman	@WestRock	@officedepot	@keybank	
@DuPont_News	@ArrowGlobal	@CDWCorp	@baxter_intl	@CitizensBank	
@StateFarm	@InsidePMI	@AEPnews	@LamResearch	@MotoSolutions	
@JNJNews	@StoneX_Official	@Cognizant	@Entergy	@MagellanHealth	
@IBM	RaytheonTech	@DRHorton	@CharlesSchwab	@NewmontCorp	

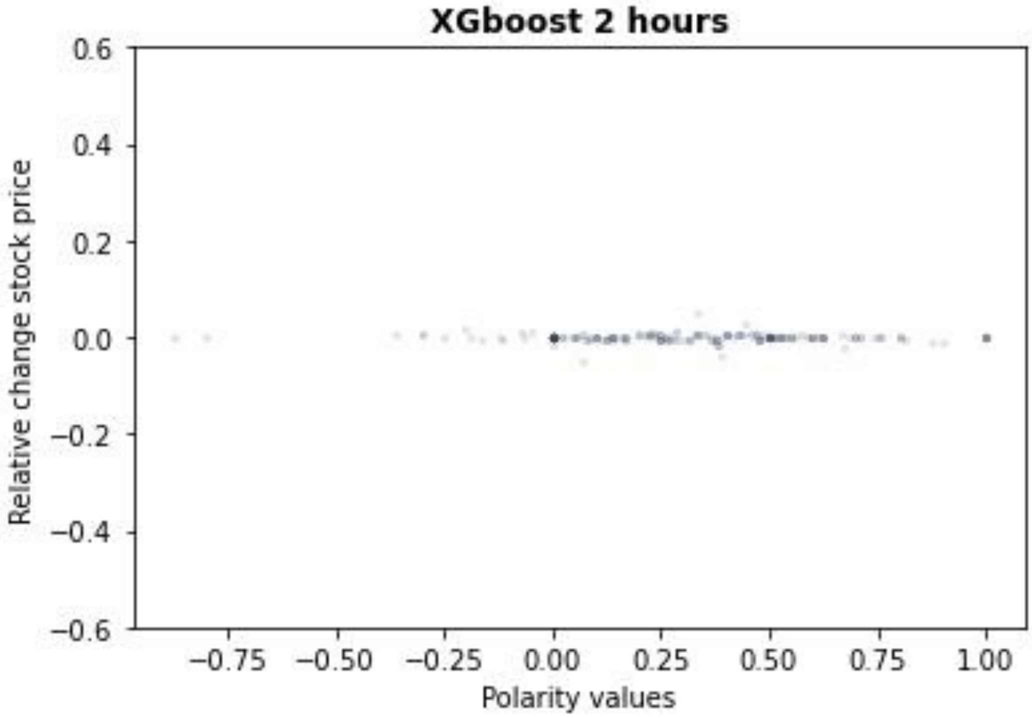
@Target	@KraftHeinzCo	@BDandCo	@nrgenergy	@SpiritAero
@FreddieMac	@MDLZ	@Nordstrom	@LiveNation	@AveryDennison
@UPS	@usbank	@netflix	@MolsonCoors	@celanese
@Lowes	@Macys	@Aramark	@eBay	@tollbrothers
@intel	@DollarGeneral	@TXInstruments	@DevonEnergy	@SanminaCorp
@MetLife	@Starbucks	@GeneralMills	@Discovery	@InsightEnt
@ProcterGamble	@DXCTechnology	@CP_News	@Ally	@OwensCorning
@UTC	@LillyPad	@goodyear	@IQVIA_global	@TATravelCenters
@FedEx	@thermofisher	@PayPal	@UNFI	@ConsumersEnergy
@PepsiCo	@USFoods	@PPG	@EastmanChemCo	@blackstone
ADMupdates	@DukeEnergy	@OmnicomMediaGrp	@RepublicService	@Wayfair
@Prudential	@Halliburton	@Mastercard	@SonicAutomotive	@askRegions
@Centene	@Cummins	@WasteManagement	@Xerox	@ultabeauty
@Albertsons	@Amgen	@Ecolab	@bostonsci	@Regeneron
@WaltDisneyCo	@CenturyLink	@BookingHoldings	@InterpublicIPG	@Burlington
@Sysco	@IntlPaperCo	@CBS	@PSEGNews	@ROKAutomation
@HP	@UnionPacific	@ParkerHannifin	@PVHCorp	@TheNTGolf
@Humana	DollarTree	@principal	@AdvanceAuto	@chemours
@Facebook	@Qualcomm	@DTE_Energy	@HormelFoods	@MarathonOil
@CaterpillarInc	@bmsnews	@blackrock	@oreillyauto	@Dillards
@EnergyTransfer	@GileadSciences	@Kinder_Morgan	@Hertz	@AMD
@LockheedMartin	@Jabil	@Loews_Hotels	@COTYInc	@MandT_Bank
@pfizer	@ManpowerGroup	@arconic	@AGCOcorp	@NCRCorporation
@GoldmanSachs	@SouthwestAir	@StanleyBlkDeckr	@Avis	@iHeartMedia
@MorganStanley	@aflac	@Textron	@Adobe	@AmerenCorp
@Cisco	@Tesla	@LasVegasSands	@BrighthouseFin	@SPGlobal
@Cigna	AutoNation	@EsteeLauder	@Voya	@ADI_News
@AIGinsurance	@CBRE	@dish	@airproducts	@RalphLauren
@HCAhealthcare	@WhirlpoolCorp	@KelloggCompany	@HiltonHotels	@BoozAllen
@AmericanAir	@McDonalds	@biogen	@GameStop	@Clorox
@Delta	@Broadcom	@Alcoa	@CampbellSoupCo	@Realogy

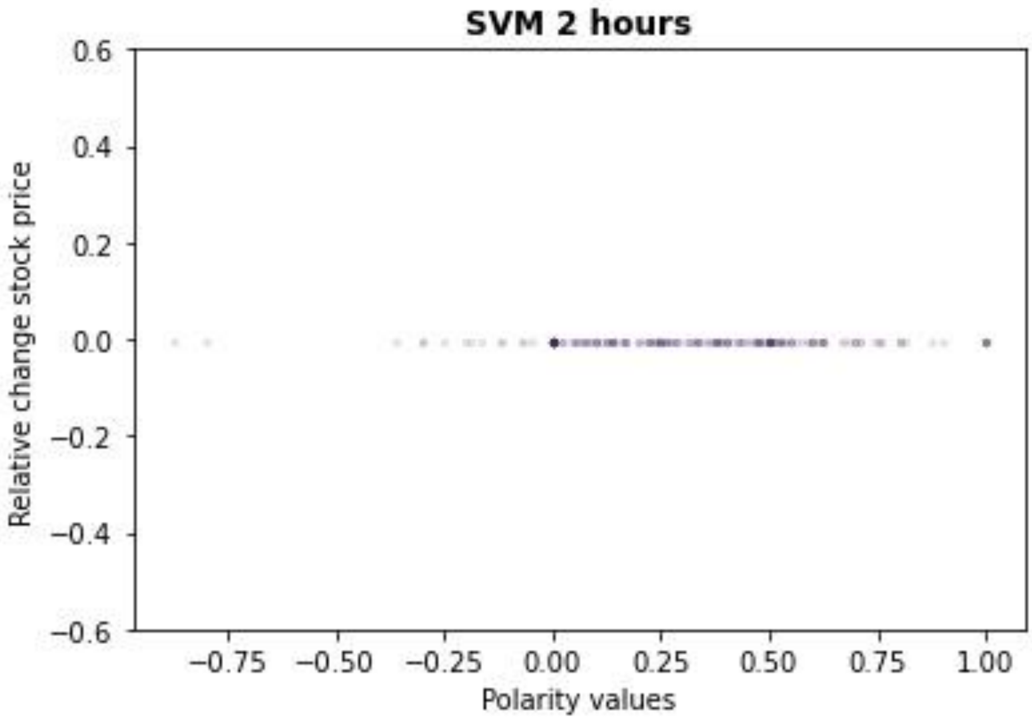
CEOs:

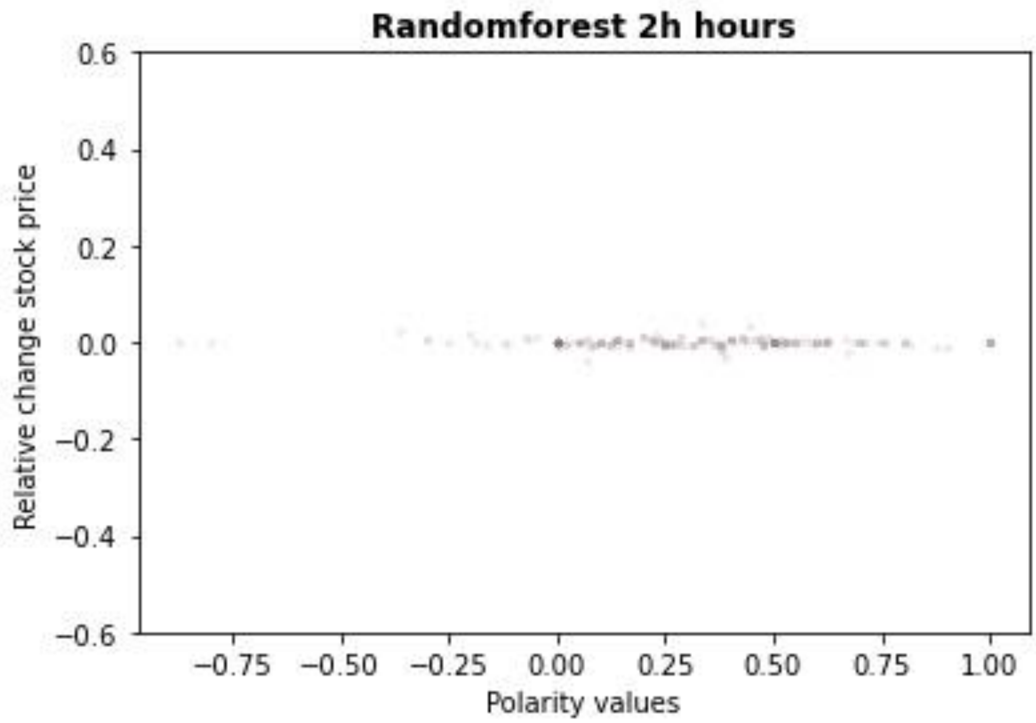
elonmusk
tim_cook
ChuckRobbins
AntonioNeri_HPE
Kevin_Johnson
Benioff
sasan_goodarzi

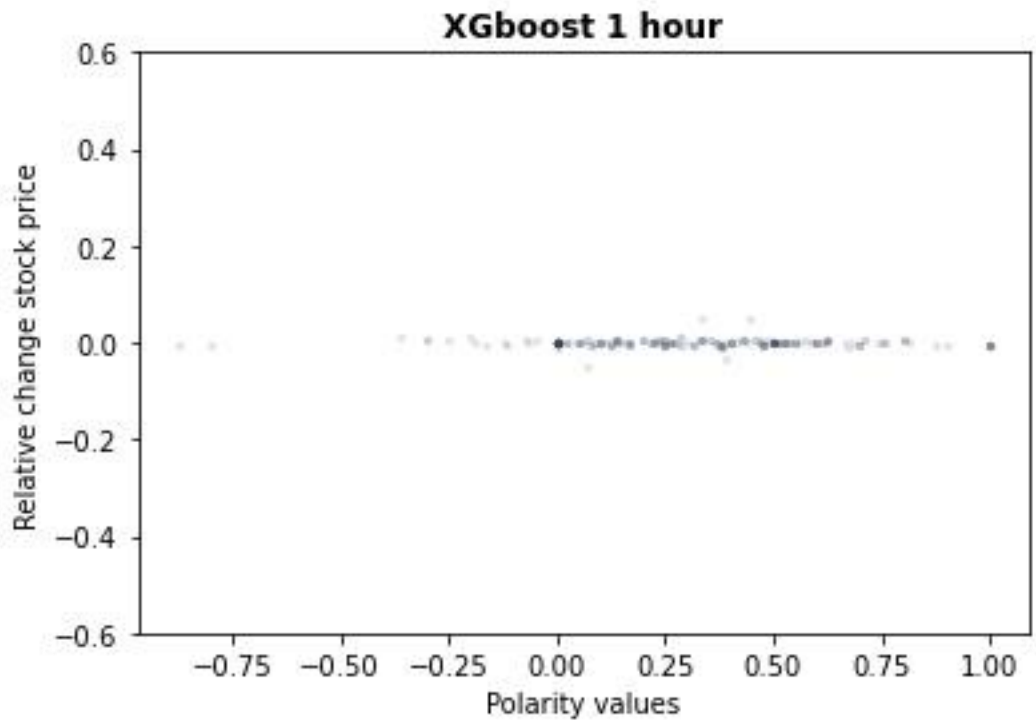
B. Figures for CEOs and companies (SVM, RF, XGBoost)

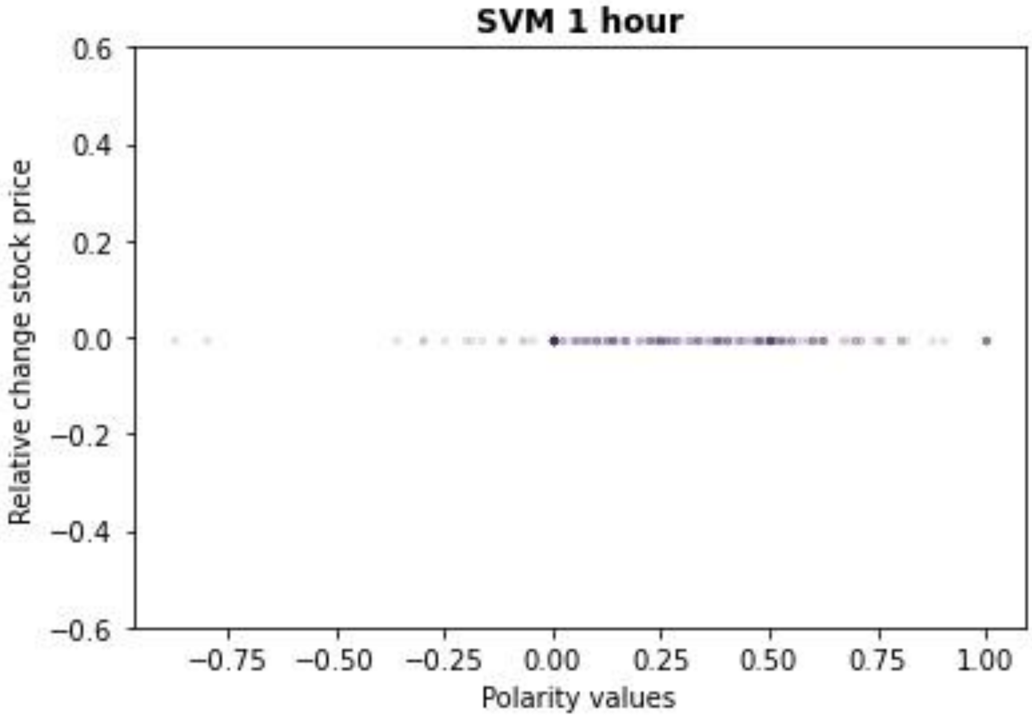
CEOs:

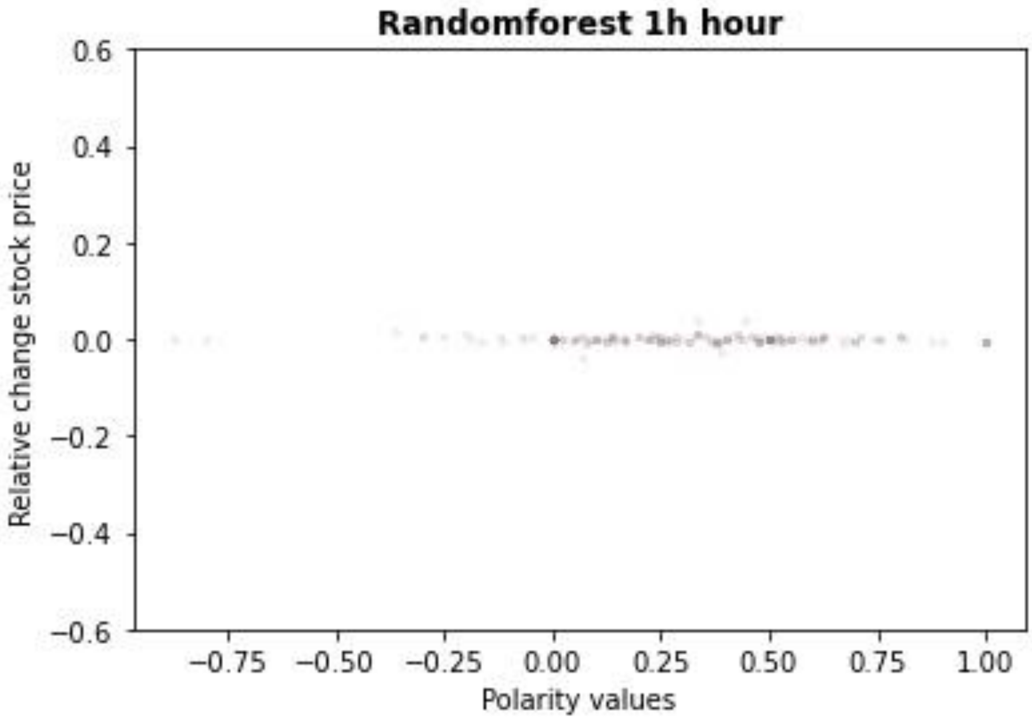

XGboost 2 hours
Relative change stock price
Polarity values
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

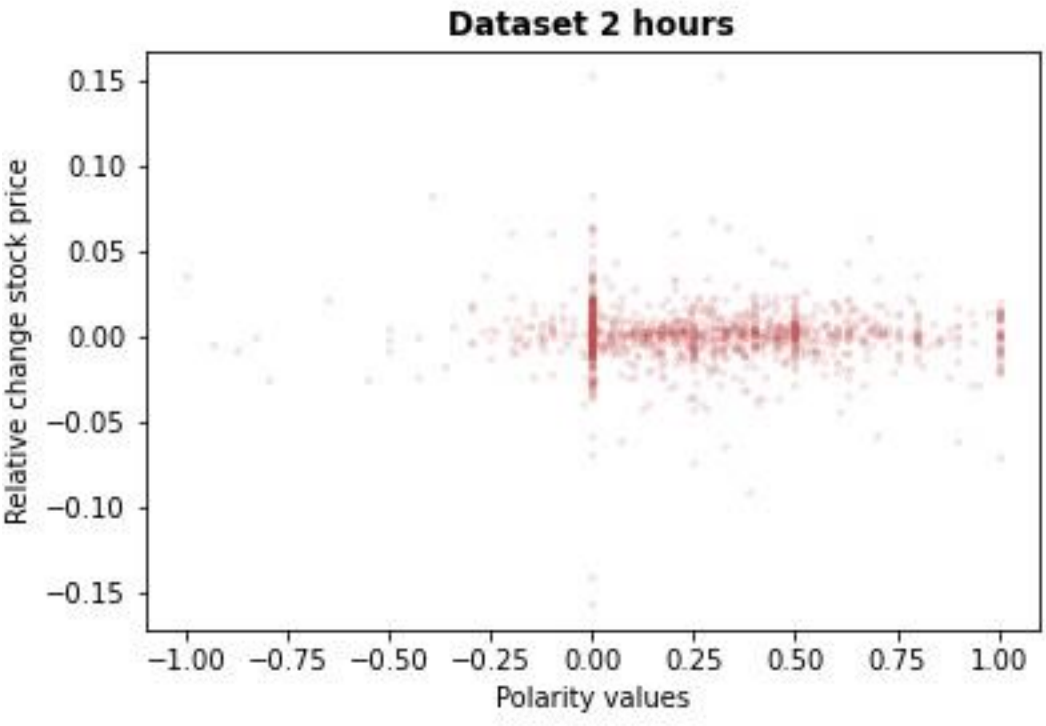

SVM 2 hours
Relative change stock price
Polarity values
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


Randomforest 2h hours
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Relative change stock price
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
Polarity values

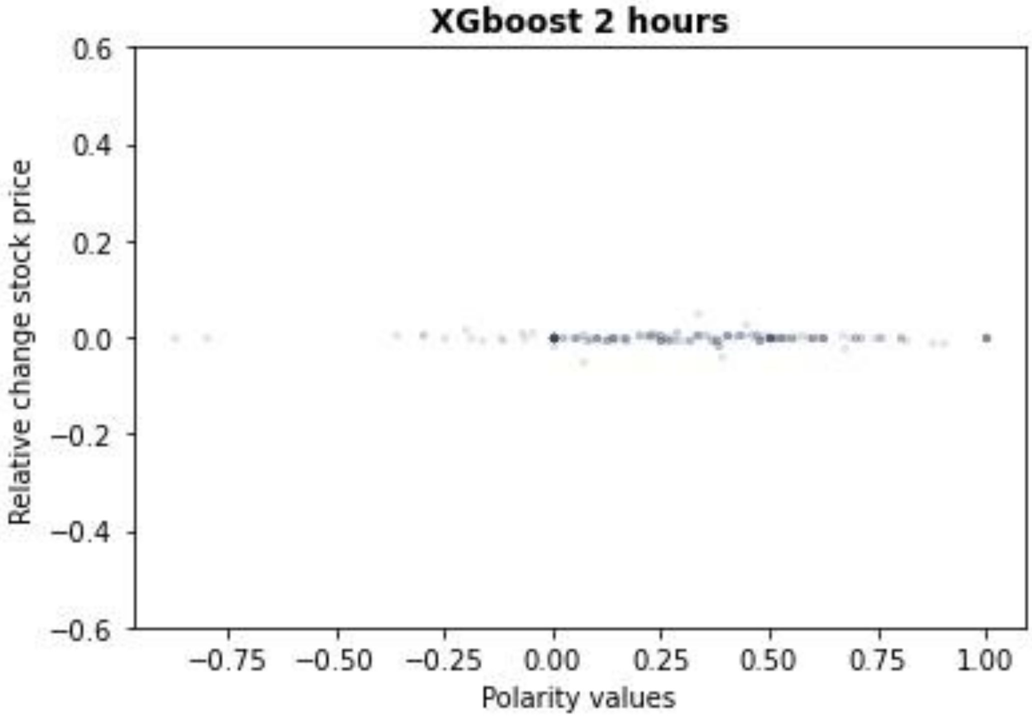

XGboost 1 hour
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Relative change stock price
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
Polarity values

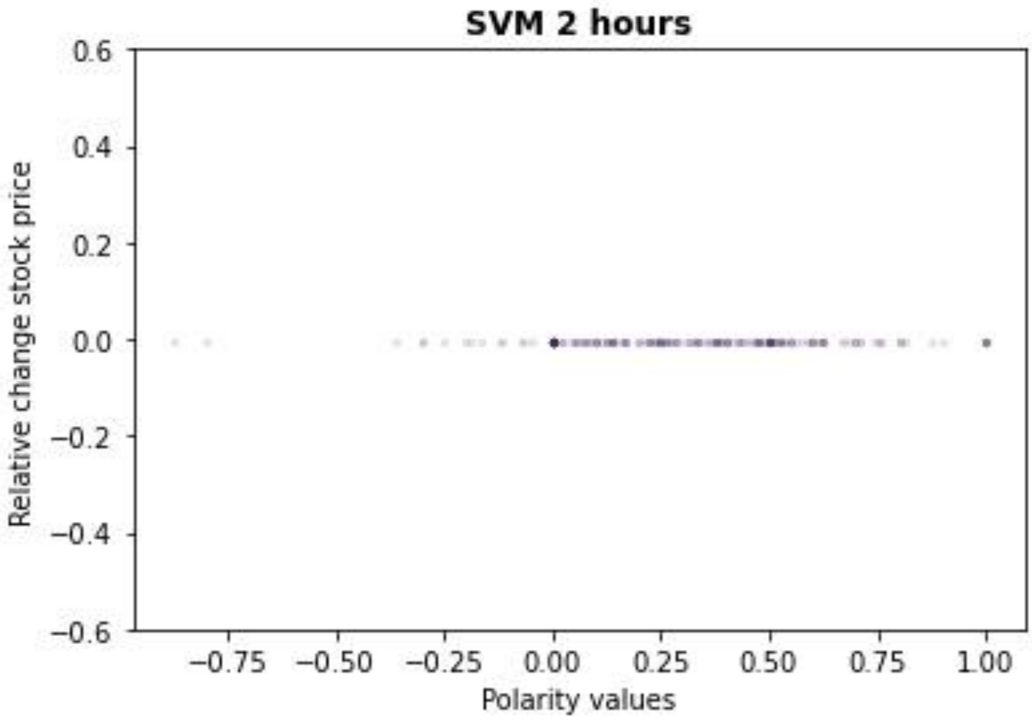

SVM 1 hour
Relative change stock price
Polarity values
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

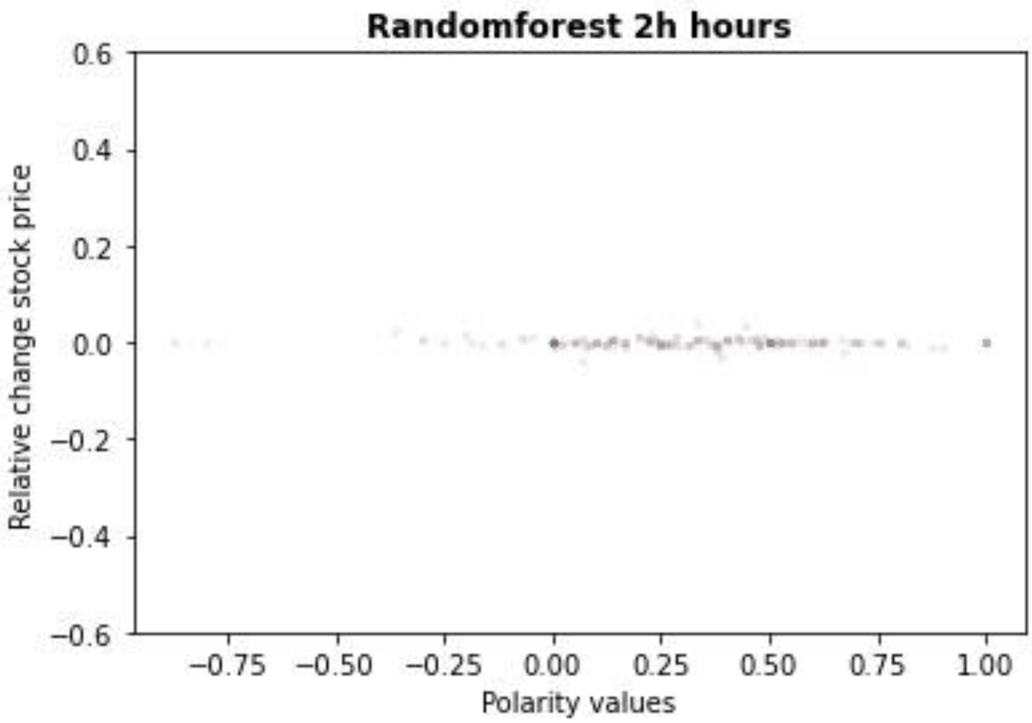

Randomforest 1h hour
Relative change stock price
Polarity values
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

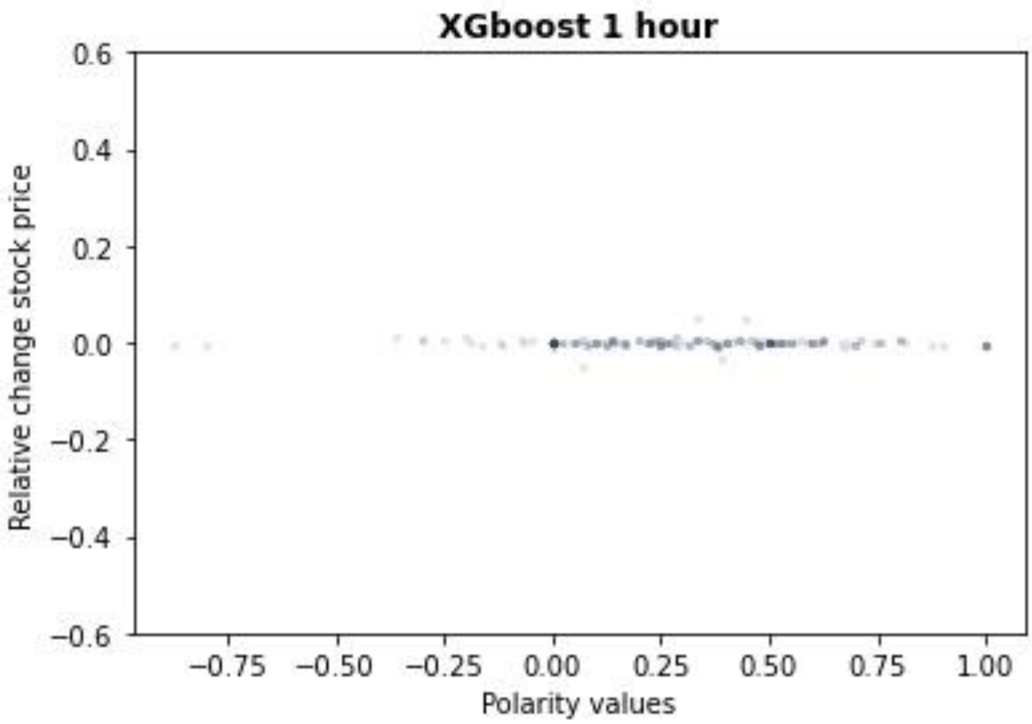

Dataset 2 hours
Relative change stock price
Polarity values
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

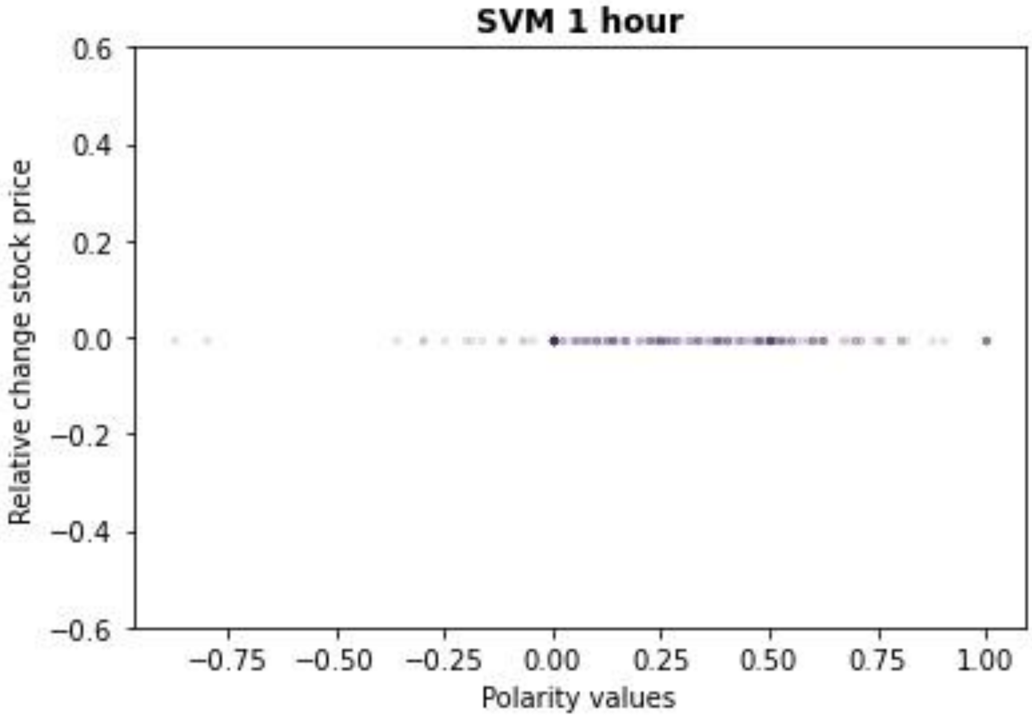
Companies:

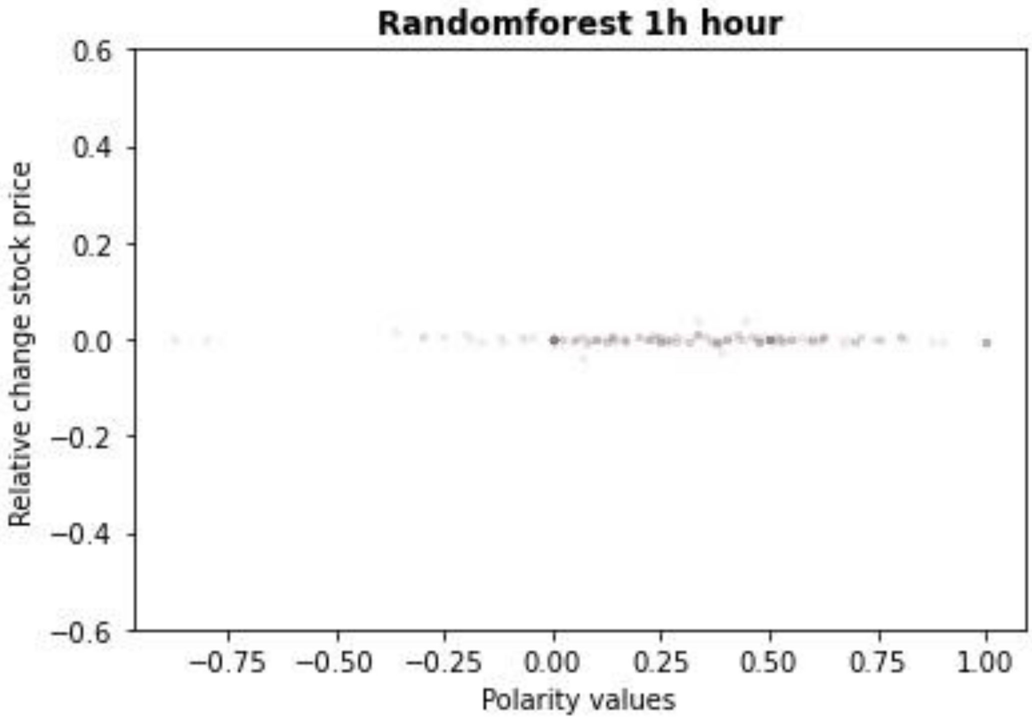

XGboost 2 hours
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Relative change stock price
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
Polarity values

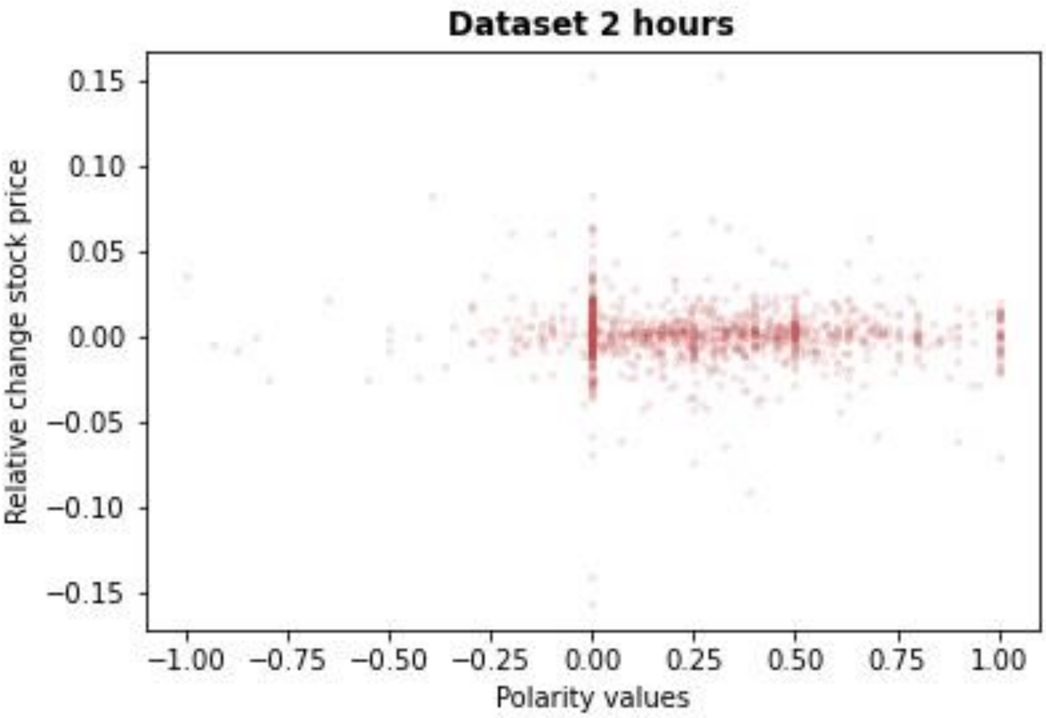

SVM 2 hours
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Relative change stock price
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
Polarity values

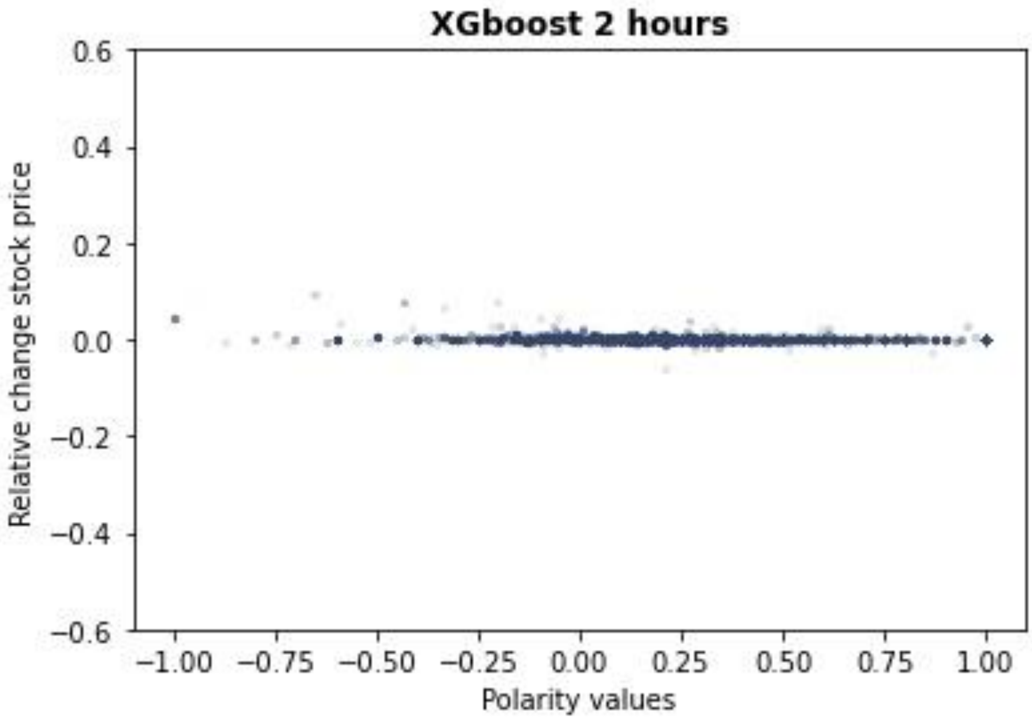

Randomforest 2h hours
Relative change stock price
Polarity values
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

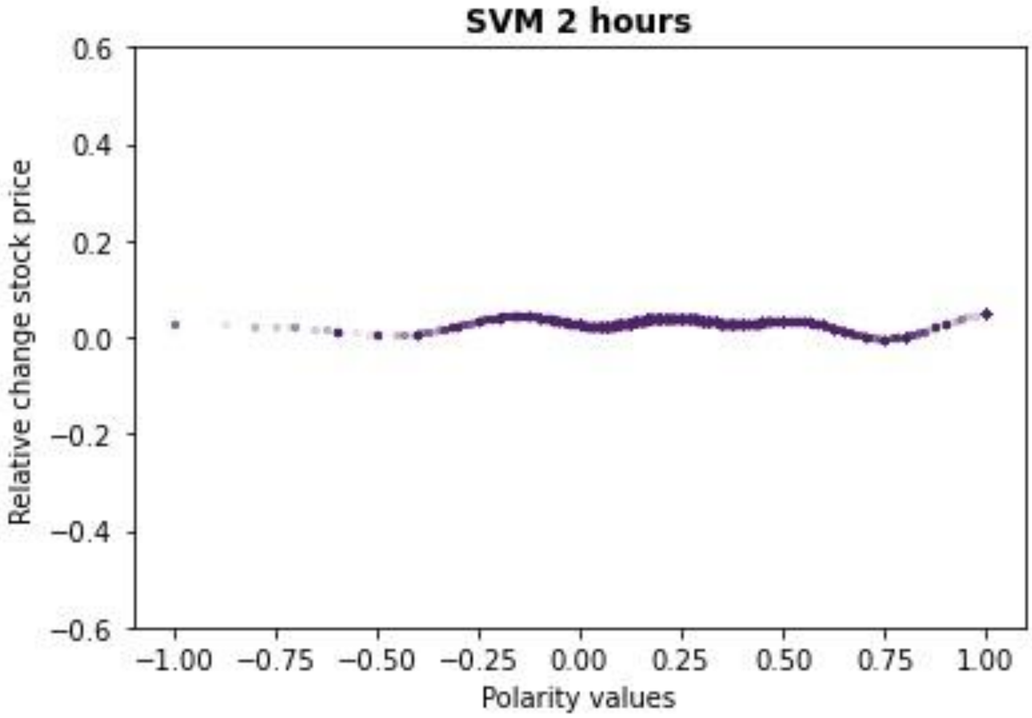

XGboost 1 hour
Relative change stock price
Polarity values
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

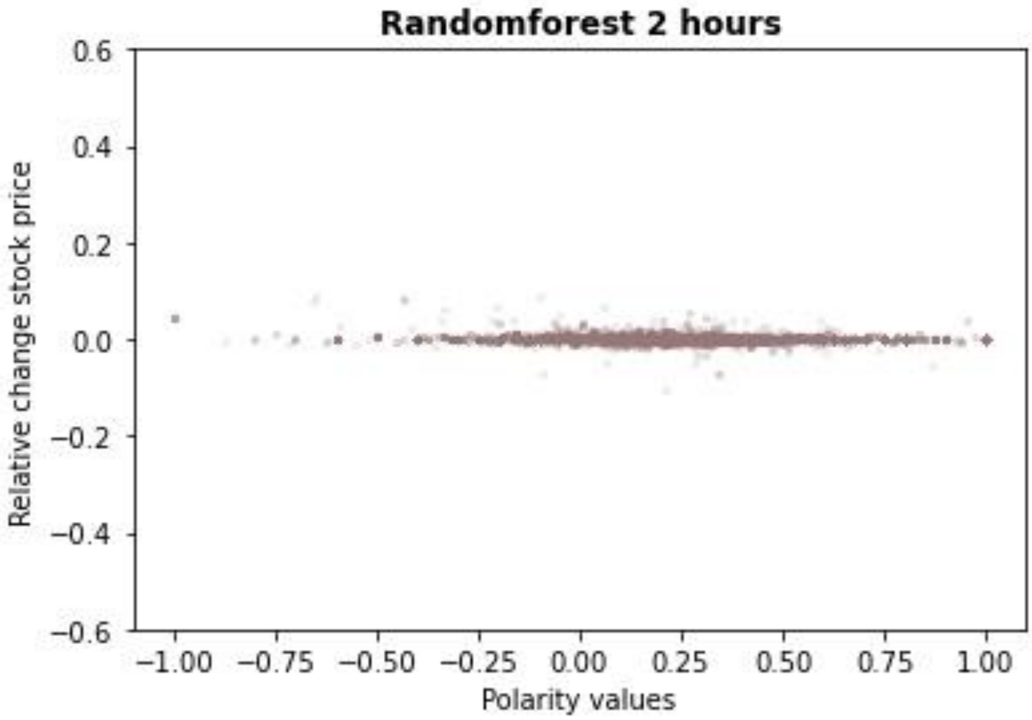

SVM 1 hour
Relative change stock price
Polarity values
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

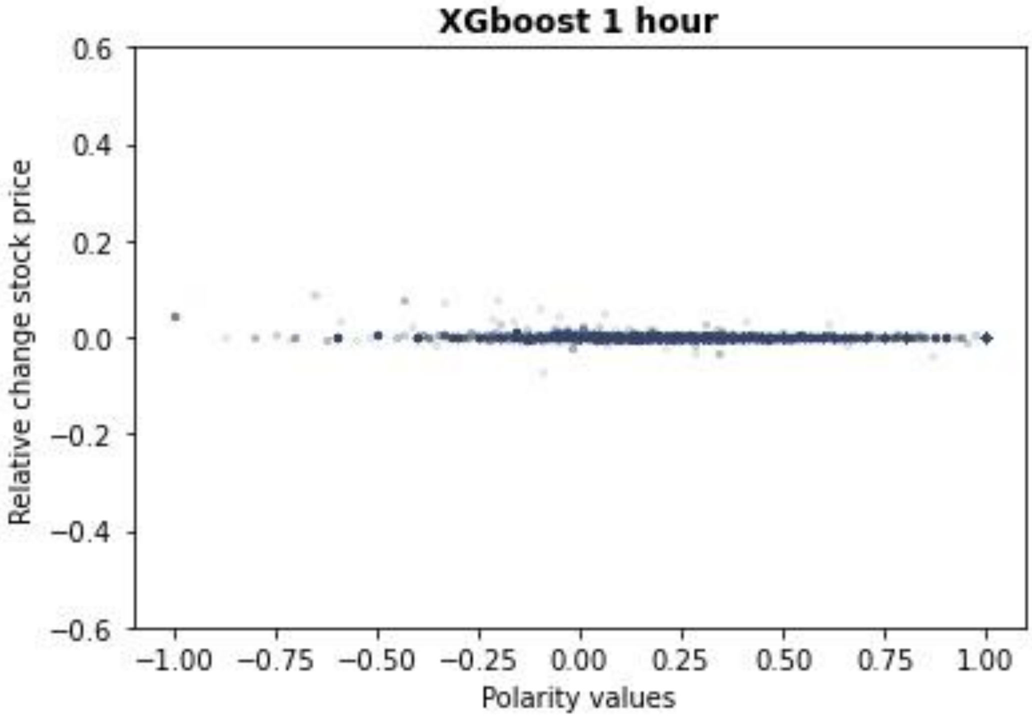

Randomforest 1h hour
Relative change stock price
Polarity values
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

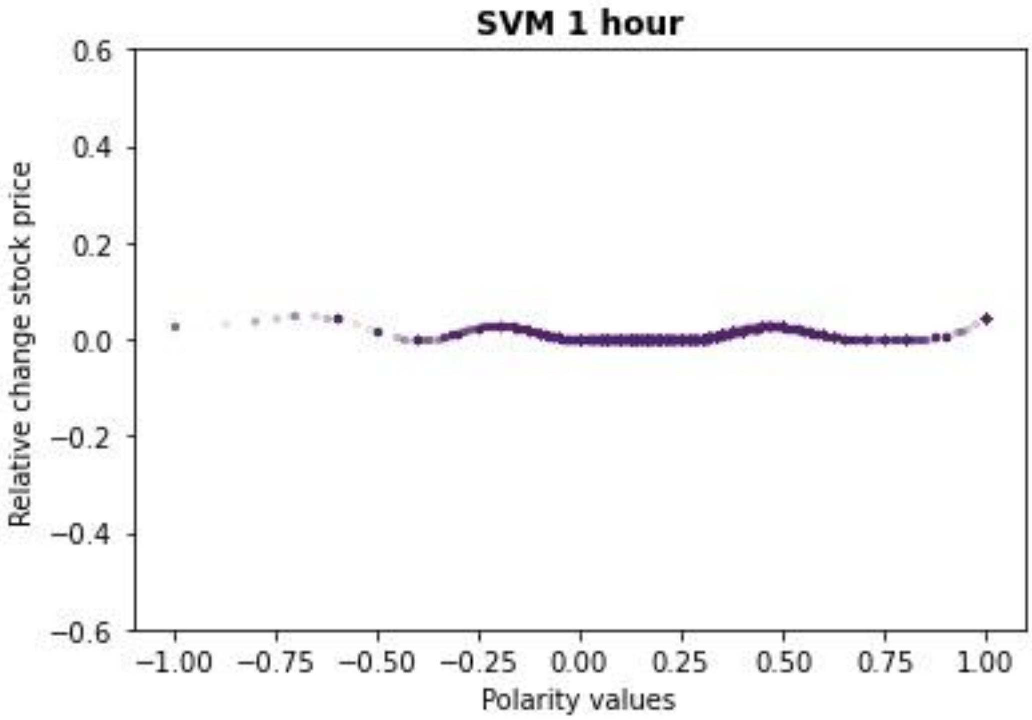

Dataset 2 hours
Relative change stock price
Polarity values
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

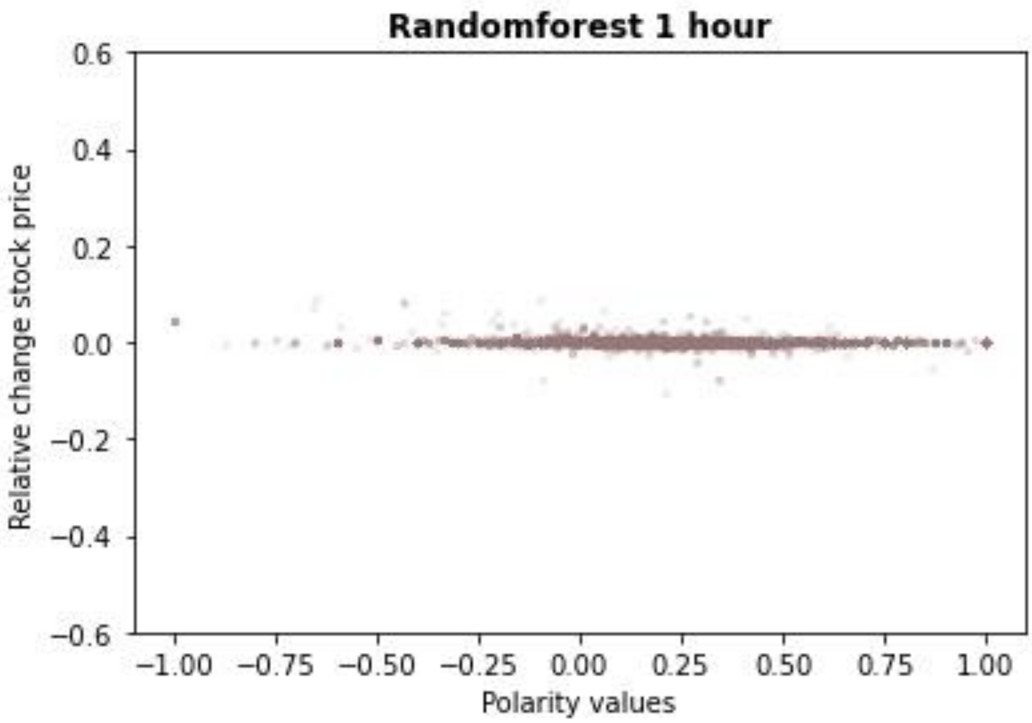

XGboost 2 hours
Relative change stock price
Polarity values
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

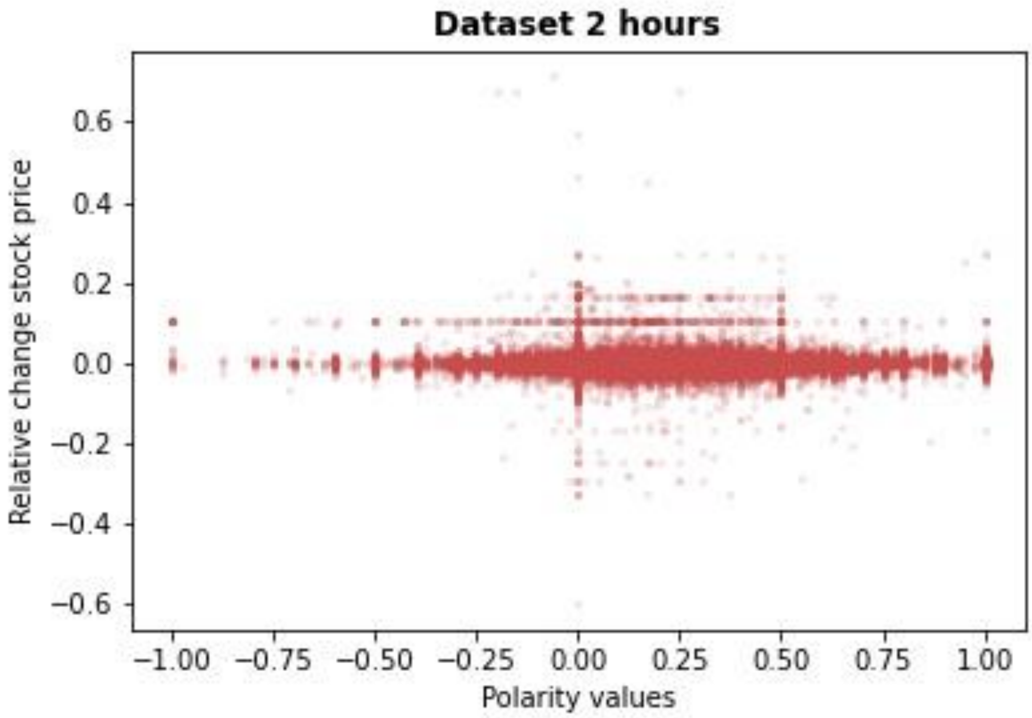

SVM 2 hours
Relative change stock price
Polarity values
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


Randomforest 2 hours
Relative change stock price
Polarity values
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


XGboost 1 hour
Relative change stock price
Polarity values
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


SVM 1 hour
Relative change stock price
Polarity values
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


Randomforest 1 hour
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Relative change stock price
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
Polarity values


Dataset 2 hours
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Relative change stock price
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
Polarity values