Stockholms universitet

Sentiment Analysis for Stock Price Prediction

Willie Langenberg

Kandidatuppsats i matematisk statistik
Bachelor Thesis in Mathematical Statistics

Kandidatuppsats 2021:4
Matematisk statistik
Juni 2021

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Mathematical Statistics
Stockholm University
Bachelor Thesis **2021:4**
http://www.math.su.se

Sentiment Analysis for Stock Price Prediction

Willie Langenberg*

June 2021

Abstract

Predicting the price of a stock is a task that could be highly profitable. There are several techniques and methods to do this, while my goal is to look into how well this could be achieved using sentiment analysis. Four stocks have been analyzed. First calculating the daily sentiment for a given stock, and then using these sentiment values to try predicting the future stock returns. When using vector autoregression models the lagged sentiments were insignificant. However, the instantaneous sentiment showed significance.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: willielangenberg@gmail.com. Supervisor: Taras Bodnar, Tony Johansson.

Contents

1 Acknowledgements

I wish to express my sincere appreciation to my supervisors, Taras Bodnar and Tony Johansson, for their guidance and encouragement along the way of this work.

2 Introduction

What would happen if you knew the future price of some financial asset. You would probably become very rich. In reality, we would never be able to predict the future price with complete certainty, we would have to make a calculated guess. There are various methods to make this guess more accurate, using different models and indicators. In this thesis, we shall further analyse the predictive power of social media sentiment. The sentiment is in short the feeling one might express in a text about some entity. This could for example be split into positive/negative. For a few stocks, we will gather social media data from a site called StockTwits. Using autoregressive models we will test if we can make a better guess using these sentiments, than without.

In the first section, we are going over the theory being used. Most of the methods used in this thesis are explained in this section. However, I will assume that the reader possesses some basic statistical knowledge. In the next section, we are doing a summary of the data. In the section that follows we are doing the analysis. This section is divided into two subsections whereas we first analyse the sentiment data, and then the modeling. We then discuss the results and present some improvement opportunities.

3 Theory

The time series theory and notation are based on the book *Analysis of time series* written by Ruey S. Tsay.[12]

3.1 Time series

Time series data contains information about a subject over time. For example, we consider the GDP of a country to be a time series. This type of data is different in comparison to the more standard cross-sectional data where we consider multiple subjects over different variables at one specific time. The order of a time series has apparent importance, which cross-sectional data lacks. Thus they also differ when analysing. With cross-sectional data, one seeks to explain or predict one response variable using other explanatory variables. With time series we want to predict or explain the series using lagged (i.e, past) values. Note that we will introduce theory for both univariate and multivariate time series. Multivariate time series is just the extension when we study multiple time series over time, whereas we not only consider relationships with lagged values but also among the series.

3.2 Stationarity

In time series analysis stationarity is the basis of many models. This could be seen visually but also tested using various statistical tests. If a series is non-stationary one would often have to transform the series into a stationary one. The transformation usually consists of differencing the series, meaning we take the difference of consecutive values. First of all, there is strict stationarity, with a strong condition. A time series is said to be strictly stationary if the joint distribution of a sample of the series is identical to the joint distribution of another sample, shifted in time. I.e, if the joint distribution of

$$(x_{t_1}, \cdots, x_{t_k})$$

is identical to the joint distribution of

$$(x_{t_1+t}, \cdots, x_{t_k+t}),$$

for all $t, t_1, \cdots, t_k$, and k is a positive integer, $\{x_t\}$ is a strictly stationary time series. This means the joint distribution of $(x_t, \cdots, x_{t+k})$ is time-invariant. However, this condition is not easy to fulfill in practice, we often consider weak stationarity instead. For a series to be weakly stationary its mean and autocovariance should be time-invariant. This means that the series has a constant mean and that the covariance between any two time points, t and $t+k$ only depends on the lag k and the difference between the

two times. We also indirectly assumes the first two moments to be finite. Visually we would thus want the series to fluctuate equally random around a fixed mean (i.e. without showing a trend). If a series is concluded to be strictly-, or weakly stationary, we can then for example apply models to forecast future values.

3.2.1 Augmented Dickey-Fuller

A non-stationary series that have a systematic pattern, a trend, is often called a random walk with drift. If such a trend exists in a series we say that we have a present unit root. To test the null hypothesis of a present unit root one can use the augmented Dickey-Fuller test (ADF), an augmented version of the Dickey-Fuller test, modified to handle more complex models. The null hypothesis is that we have a unit root and the alternative hypothesis is stationarity or trend-stationarity. Suppose that we want to test if there is a unit root in an AR(p) (see section 3.5) process, for the series x_t. Then we may test $H_0 : \beta = 1$ against $H_1 : \beta < 1$ using

$$x_t = c_t + \beta x_{t-1} + \sum_{i=1}^{p-1} \phi_i \Delta x_{t-i} + e_t,$$

where c_t is a deterministic function that could be zero, a constant or $c_t = w_0 + w_1 t$. Also, Δx_j is the first difference of the series x_t where $\Delta x_j = x_j - x_{j-1}$. We estimate the β with the least-squares estimate $\hat{\beta}$. The augmented Dickey-Fuller test statistic is then given by

$$\text{ADF} = \frac{\hat{\beta} - 1}{std(\hat{\beta})},$$

where we reject the null hypothesis with some level of certainty if the t-ratio statistic exceeds a critical value.

3.2.2 KPSS

To complement the ADF test, one could also use the Kwiatkowski-Phillips-Schmidt-Shin [5] (KPSS) test. On the contrary, this tests the null hypothesis of stationarity against the alternative hypothesis of a present unit root. Suppose we want to test for stationarity in a time series y_t. The test involves decomposing the series into the sum of a deterministic trend, a random walk and a stationary error

$$y_t = \xi t + r_t + \epsilon_t,$$

where r_t is the random walk defined as

$$r_t = r_{t-1} + u_t.$$

Furthermore, u_t are iid distributed with zero mean and variance σ_ϵ^2. Under the null hypothesis that $\sigma_\epsilon^2 = 0$ the series is considered to be trend stationary. If we also use the special case that the series is instead decomposed using $\xi = 0$ the null hypothesis would imply level stationarity. To test this we would go on to use the statistic

$$\hat{\eta} = T^{-2} \sum S_t^2 / s^2(l)$$

for the hypothesis $\sigma_\epsilon^2 = 0$. We define the partial sum process of the residuals S_t as

$$S_t = \sum_{i=1}^{t} e_i, \quad t = 1, \cdots, T.$$

Also, $s^2(l)$ is a consistent estimator of σ_ϵ^2. It follows that

$$\hat{\eta} \longrightarrow \int_0^1 V_2(r)^2 dr,$$

where $V_2(r)$ is a second-level standard Brownian bridge. For further notes see [5].

3.3 Return

When dealing with financial data, and in particular stock price data, one might consider the price as a time series. However, this is not a good idea. The reason being is that the price of a stock is not stationary in its foundation. The price varies in trends, having peaks and lows that contradict the definition of stationarity. To solve this problem it is common to instead consider the return of a stock. The return has better statistical properties than the price, one being that it often holds weak stationarity. Thus a better fit for time series analysis. There are different ways of defining return. In all cases though, we are not including any dividend, as one might think of returns.

Simple return

We let P_t denote the price at time t of a stock. The simple return for holding the stock one period is then defined as

$$R_t = \frac{P_t}{P_{t-1}} - 1.$$

If we buy one stock at time $t-1$ and sell it at time t, we would get the return of R_t on our investment. The simple gross return is then defined as $R_t + 1$. If we would consider to buy and hold our stock for more than one

period, i.e. buying the stock at $t-k$ and selling at t, the simple gross return would be

$$1 + R_t[k] = \frac{P_t}{P_{t-k}} = \prod_{j=0}^{k-1}(1 + R_{t-j}).$$

Log return

The natural logarithm of the simple gross return is called the log return and is defined as

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right). \tag{1}$$

The log return hold some pleasant statistical properties. When holding a stock for multiple time periods the log return is the sum of every one-period log return. If we hold the stock for k periods, from $t-k$ to t the log return is

$$r_t[k] = r_t + r_{t_1} + \cdots + r_{t-k+1}.$$

3.4 Correlation

Correlation coefficient

Correlation is the statistical relationship between two random variables. The correlation coefficient is a measure of this relationship. It is a value between -1 and 1, for the strength of the linear dependence. The coefficient being zero equals zero correlation. For two random variables X and Y, the correlation coefficient is

$$\rho_{x,y} = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}} = \frac{E[(X-\mu_x)(Y-\mu_y)]}{\sqrt{E(X-\mu_x)^2 E(Y-\mu_y)^2}},$$

with $\mu_x = E(X)$, $\mu_y = E(Y)$ and we assume that the variances exists.

Autocorrelation function

When dealing with time series the correlation coefficient is extended to the autocorrelation function. Here we instead study the linear dependence between a time series and its lagged values. Consider $\{r_t\}$ to be a time series of log returns from a stock, we assume the series r_t to be weakly stationary. The correlation coefficient between r_t and the lagged value $r_{t-\ell}$ is the lag-ℓ autocorrelation or ρ_ℓ of r_t. Similarly to the correlation coefficient we then

define the lag-ℓ autocorrelation

$$\rho_\ell = \frac{\mathrm{Cov}(r_t, r_{t-\ell})}{\sqrt{\mathrm{Var}(r_t)\mathrm{Var}(r_{t-\ell})}} = \frac{\mathrm{Cov}(r_t, r_{t-\ell})}{\mathrm{Var}(r_t)},$$

with simplifying the equation by using the properties of weakly stationarity. Further, we conclude that r_t is not serially correlated if and only if $\rho_\ell = 0$ for all $\rho > 0$. In practice we would estimate ρ_ℓ by

$$\hat{\rho}_\ell = \frac{\sum_{t=\ell+1}^{T}(r_t - \bar{r})(r_{t-\ell} - \bar{r})}{\sum_{t=1}^{T}(r_t - \bar{r})^2}, \quad 0 \leq \ell \leq T-1,$$

where T is the sample size. Under some conditions this estimate is a consistent estimate of ρ_ℓ. Further we can test the hypothesis that individual ρ_ℓ equals zero.

Portmanteau Test

More importantly, we can also jointly test the hypothesis that multiple lag-ℓ autocorrelations ρ_ℓ equals zero. This is done with the Portmanteau test, first stated by Box and Pierce (1970). This statistic was later modified by Ljung and Box (1978), to be more accurate in finite samples. Suppose that we state the hypothesis $H_0 : \rho_1 = \cdots = \rho_m = 0$ against the alternative hypothesis $H_1 : \rho_i \neq 0$ for some $i \in \{1, \cdots, m\}$. The Ljung-Box statistic is defined by

$$Q(m) = T(T+2)\sum_{\ell=1}^{m}\frac{\hat{\rho}_\ell^2}{T-\ell}.$$

Under the null hypothesis and some regularity conditions, $Q(m)$ follows a chi-squared distribution with m degrees of freedom. We then reject the null hypothesis if the statistic $Q(m)$ is greater than some significant percentile of the $\chi^2(m)$ distribution.

3.5 AR

An AR *(autoregressive)* model is very similar to a linear regression model. In a linear regression model, we want to explain or predict the value of a response variable with the use of various explanatory variables. The AR model instead uses lagged values as explanatory variables, in its attempt

to explain or predict the time series. The hyperparameter to tune in such model is the amount of lagged values, p.

AR(1)

We start by defining the simplest form of an AR model, the AR(1) model. Consider $\{r_t\}$ to be the time series of log returns from a stock, then

$$r_t = \phi_0 + \phi_1 r_{t-1} + a_t$$

is an AR(1) model, where $\{a_t\}$ is assumed to be a white noise series with zero mean and variance σ_a^2.

AR(p)

Directly from the AR(1) model we can proceed to define the AR(p) model as

$$r_t = \phi_0 + \phi_1 r_{t-1} + \cdots + \phi_p r_{t-p} + a_t, \tag{2}$$

where p is defined as the positive integer for the amount of lagged values to be considered, and $\{a_t\}$ a white noise series. To estimate the parameters ϕ we could use the fact that Eq.(2) is similar to a linear regression model. Thus we can estimate the parameters using the least squares method.

Estimation

In short the least squares method is based on the idea that we want to minimize the residuals of the actual time series r_t and the fitted $\hat{r}_t$. We fit $\hat{r}_t$ by estimating ϕ as $\hat{\phi}$. The fitted model is given by

$$\hat{r}_t = \hat{\phi}_0 + \hat{\phi}_1 r_{t-1} + \cdots + \hat{\phi}_p r_{t-p},$$

with the residuals

$$\hat{a}_t = r_t - \hat{r}_t.$$

The least squares method finds the estimates for ϕ so that the residuals are minimized. We rewrite Eq.(2) as

$$r_t = XB + a_t,$$

where we define X to be the design matrix containing lagged values of the series and B the coefficients. The time series is of length n, and we will have the equation

$$\begin{pmatrix} r_t \\ r_{t-1} \\ \cdots \\ r_{t-n} \end{pmatrix} = \begin{pmatrix} 1 & r_{t-1} & r_{t-2} & \cdots & r_{t-p} \\ 1 & r_{t-2} & r_{t-3} & \cdots & r_{t-p-1} \\ & & \cdots & & \\ 1 & r_{t-n-1} & r_{t-n-2} & \cdots & r_{t-n-p} \end{pmatrix} \begin{pmatrix} \phi_0 \\ \phi_1 \\ \cdots \\ \phi_p \end{pmatrix} + \begin{pmatrix} a_t \\ a_{t-1} \\ \cdots \\ a_{t-n} \end{pmatrix}, \quad (3)$$

where r_t, a_t are vectors of length $n-p$, X a $(n-p) \times (p+1)$ matrix and B a vector of length $p+1$. Furthermore the least squares estimate for B is given by

$$\hat{B} = (X^T X)^{-1} X^T r_t.$$

Forecasting

Forecasting is the basis of our study. The goal is to find a model that could forecast future stock prices, given the information we have at time t. It is possible to forecast different horizons, meaning we can forecast the return for a stock one day, two days or even a week from now. Longer horizons are followed with higher uncertainty. It can be shown that for a stationary series the forecast will converge to the mean $E(r_t)$ when the forecast horizon increases. In this study we will focus on the one step ahead forecast, forecasting only one day in the future. With an AR(p) model the forecasting is straightforward using the fitted model. Consider the situation that we are at time index h and want to forecast one period ahead $h+1$. Now let $\hat{r}_{th}(\ell)$ be the ℓ-step ahead forecast of $r_{h+\ell}$. The one-step-ahead forecast is then given by

$$\hat{r}_h(1) = \hat{\phi}_0 + \sum_{i=1}^{p} \hat{\phi}_i r_{h+1-i}.$$

3.6 Order determination

When fitting an AR(p) model on real data we have to come up with a way to pick the amount of lagged values p that best fit our data. This task is referred to as order determination. There is essentially two main ways to go. The first is to use the partial autocorrelation function (PACF) and the other to use an information criterion.

3.6.1 PACF

Suppose that we want to fit a series r_t to an AR(p) model. We start by considering an AR(1) model to fit to our data. We add one more lag to create an secondary AR(2) model, and uses a F-test to see if the new contribution is

significant. The idea is to keep doing this until we see that every contribution is getting less significant for every added lag to some limit $j > k$. We then choose $p = j$ for our model. Typically we would use a plot where we can see the p-values of each added lag.

3.6.2 Information Criterion

The Akaike information criterion (AIC) is a common information criterion based on likelihood. It is defined as

$$\text{AIC} = \frac{-2}{n}\ln(\hat{L}_{\max}) + \frac{2}{n} \times (k),$$

with $\hat{L}_{\max}$ being the maximum value of the likelihood function for the model and k the amount of estimated parameters in the model. Given a set of models we would choose the model with the lowest AIC. As similar information criteria, AIC encourages goodness of fit by the likelihood function, but also punishes overfit by the penalty of $+\frac{2}{n}$ for every parameter included. Usually the goodness of fit is increased by more estimated parameters, while we also increase the risk of overfitting, what AIC tries to counter.

3.7 Multivariate time series

With multivariate time series we are extending the theory of univariate time series to the analysis of multiple time series. In finance we can often see dependencies between various markets, opening up new ideas for us to model the world. In our paper this will enable us to analyse the relationship between the return of a given stock and some other time series. In most aspects the theory can be generalized directly from the univariate case.

Multivariate time series consists of multiple univariate time series, or components. The notation of these series makes use of vectors and matrices. So for an example, lets assume we have two series r_t and s_t. Whereas r_t is the log returns of a stock and s_t the daily sentiment, where we assume that they have the same length. For now we will not define the daily sentiment further, think of them as some indicator of sentiment for each day t. We can then write them together in a more concise manner as $\mathbf{r}_t = (r_t, s_t)'$. The bold font indicates that we are dealing with a vector. Also, $\mathbf{r}_t'$ is the transpose of $\mathbf{r}_t$.

3.7.1 Stationarity

The vector series $\mathbf{r}_t$ is weakly stationary if its first two moments are time invariant, i.e. constant over time. The first two moments being the mean vector and the covariance matrix. For such 2-dimensional time series $\mathbf{r}_t$ we defined its mean vector and covariance matrix as

$$\boldsymbol{\mu} = E(\mathbf{r}_t), \quad \boldsymbol{\Gamma}_0 = E[(\boldsymbol{r}_t - \boldsymbol{\mu})(\boldsymbol{r}_t - \boldsymbol{\mu})'].$$

Further we define the lag-ℓ cross-covariance matrix of $\mathbf{r}_t$ as

$$\boldsymbol{\Gamma}_\ell = E[(\boldsymbol{r}_t - \boldsymbol{\mu})(\boldsymbol{r}_{t-\ell} - \boldsymbol{\mu})'],$$

where the matrix $\boldsymbol{\Gamma}_\ell$ is a measure of the lead-lag relationship between the two component series in $\mathbf{r}_t$. For a weakly stationary series $\boldsymbol{\Gamma}_\ell$ is only a function of ℓ and not the time t.

3.8 VAR

An extension of the auto regressive model in multivariate time series is the vector autoregressive (VAR) model. Here we want to model one series using its lagged values and the lagged values of other series. When using the other series we might also want to include the instantaneous values i.e. not only the lagged values but also the present values of those series. We could use multiple series, but in our thesis we will only consider the case of using two series. Suppose we have the mentioned series r_t and s_t, then we define the VAR(p) model as

$$\begin{aligned} r_t &= \sum_{j=1}^{p} \alpha_{t-j} r_{t-j} + \sum_{j=1}^{p} \beta_{t-j} s_{t-j} + u_{1,t}, \\ s_t &= \sum_{j=1}^{p} \lambda_{t-j} r_{t-j} + \sum_{j=1}^{p} \gamma_{t-j} s_{t-j} + u_{2,t}. \end{aligned} \tag{4}$$

where r_{t-j} is the $t-j$ lagged return and s_{t-j} the lagged sentiment values. For both equations in the VAR model, we will estimate the coefficients similarly to how we did for the AR model. We can treat each equation separately as an AR model. We will use the least squares method to estimate the coefficients, by including the necessary lagged values in the design matrix.

3.9 Efficient Market Hypothesis

The efficient market hypothesis[8] states that the price of a financial asset (i.e. stocks), reflects all available information at that time. Thus implying there is no way to consistently predict future prices with historical data.

The market would price the asset with all information available and would therefore always reflect the correct valuation. If this is true, we would not be able to fit any models to forecast future stock prices. There is both a large number of supporters and critics of this thesis. Several reports have concluded that the thesis is correct. Critics tend to use human behavior as a counterargument, for example, that overconfidence or fear might lead to inaccurate prices that could be exploited. Depending on our results, we would either have evidence to reject the hypothesis, or not.

4 Data

We are to fetch data during the period 2020-01-01 to 2021-02-01. To get reliable estimates of the sentiment it is important to use a large dataset. The number of messages posted about a certain stock varies and is dependent on the popularity among investors. I chose to randomly pick four stocks from the Dow Jones Index. This index consists of 30 American companies, all having a high market value. I reasoned that these companies would have high popularity as well. The randomly chosen stocks turned out to be American Express, Boeing, McDonald's, and Walt Disney.

4.1 Stocktwits

In this paper, we are using social media content from Stocktwits, a microblogging platform for stock-related content. Stocktwits was created in 2008 and quickly became a popular platform for discussions regarding the stock market. The Stocktwits users can discuss and post their ideas about certain stocks. In every post, you include a "cashtag", the symbol of the company or companies that are being discussed. It is a popular platform, now getting a large amount of traffic every day, with users responding to real-time events constantly. With that being said, it suits our purpose to have plenty of data that are up-to-date, and relatable to the stock market. One might also speculate that its users have a greater understanding of the stock market, even though there are no requirements needed to create an account or post.

The data is retrieved using a written software in R that interacts with Stocktwits API. We are restricted to only make 200 request calls per hour, with every request giving us 30 posts. The total amount of posts retrieved is 426978. The distribution of these message is seen in *Table.1*. There is a big difference, with Walt Disney and Boeing having the most amount of observations.

Every post contains a lot of valuable, and unnecessary information. We have in total 86 columns with only 4 being interesting for us. These are

Distributions of messages

American Express	McDonald	Walt Disney	Boeing	**Total**
6180	14400	107757	298641	**426978**

Table 1: Summary of the amount of observations distributed over the stocks.

the messages, the date of which the post was created, its entities, and the basic sentiments. The first two are self-explanatory. The entities give us information about which companies the author is writing about. This will prove to be useful later when we want to classify the unclassified sentiments. The basic sentiments are the pre-classified sentiments. Every author has the option to include if they are bullish or bearish when posting a message. Of our data 49% is pre-classified, so our task is to classify the other 51%. We will later analyze how the models will perform using only the pre-classified sentiments, and the combined sentiments.

It is important to know that the messages of the pre-classified sentiments hold low quality. The authors can express their sentiment by changing the "status", which means that they tend to not express any, or little, sentiment in the text. A large portion of these messages are impossible to classify, even for a human.

4.1.1 Data preprocessing

Text data is a messy format, containing lots of noise, but also a great deal of information. When analysing text data it is crucial to preprocess the data, clean it from unnecessary information. If this is not done properly it could affect the accuracy of the classification model. The basic and most obvious actions include removing links, additional spaces, punctuations, special characters, and numbers. We want to perform these simplifications mainly for one reason; to lower the number of words considered, making it faster to fit a model, but also to remove noise that could lead to misunderstandings in the classification.

However, while preproccessing is absolutely necessary, too complex methods does not only have a very slight improvement, it could also lower the performance shown in some reports. Citing from [6]

> "Some linguistic modifications using WordNet, stemming, negation, and collocation were tested too. However, these were not helpful and actually degraded the classification accuracy".

4.2 Stock price data

We have retrieved the stock price data from yahoo finance, using the R package tidyquant. For each company chosen we have daily stock data for the low, high, open, close and adjusted close price. We are using the adjusted close price which is adjusted for all applicable splits and dividend distributions. See *Figure.1* for plots of each stock price. In each plot the title corresponds to the stock symbol for each company with AXP being American Express, BA for Boeing, DIS for Walt Disney and MCD for McDonald's.

Figure 1: Adjusted price for each stock, over our given time period.

One problem with financial data, is the missing values on weekends and holidays. This will create problems when we eventually deal with sentiment data, that exists for every day. To fix this we apply a rolling mean function to extend the data. Visually this would look like connecting the ends where data is missing, see *Figure.2* for an example.

Furthermore, looking at *Figure.1* these time series do certainly not look stationary. We will instead consider the log return of the adjusted price, given by Eq.(1) in section 3.3. For each company we will have a series $r_{t,j}$ that corresponds to its log returns, where $j \in \{1, 2, 3, 4\}$. See *Figure.3* for a plot of each series. The price fall in *Figure.1* and the large values in *Figure.3* during late February 2020 is the financial market's reaction to the coronavirus outbreak.

Figure 2: Example of the extended stock prices for American Express during January 2021.

5 Analysis

The analysis will consist of two parts. First we will analyse the sentiment of our text data, and then try to model any relationship between the sentiment and log return.

5.1 Sentiment analysis

This section will explain how to analyze the sentiment for user-created content. Sentiment analysis is the practice of analyzing text to conclude what sentiment, attitude, or emotion the writer intent [6]. It could be seen as a subfield of natural language processing (NLP). NLP is the more general field, containing methods for machines to interact with human languages. When analysing sentiment regarding the stock market, our purpose is to see if the writer has a "bullish", "bearish" or neutral sentiment. These are standard lingo's being used in the industry. Bullish means that the writer is optimistic about the stock price, hoping it to increase, and bearish is the opposite. What we want to accomplish is to automatically classify tens of thousands of posts as bearish, bullish or neutral.

When classifying text several problems could occur. A simple sentence as "I love this company" is considered to be an easy text to classify. However, it gets more complicated when the writer is being sarcastic or writing in a

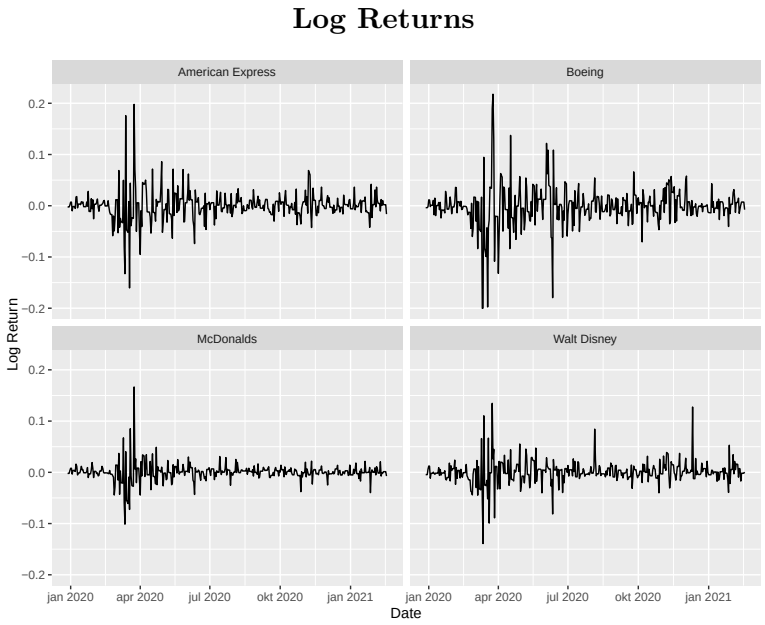

Figure 3: Log return of the adjusted price.

manner where the order is important. For example, simpler models could have problems classifying "I don't love this company". In this paper, we are not going to use the most advanced methods, but a suitable method that can classify the messages with decent accuracy and speed. With more complex models we would see significantly larger computing times.

Our goal is to analyse the sentiment for posts that we assume are created by one user, and who expresses their opinion on a single entity. These assumptions generally hold for all posts, with some exceptions. In some posts, the user expresses comments regarding multiple entities (companies). With the format of Stocktwits, the users have tags for the intended companies they are commenting upon. This simplifies things, making it easy for us to filter so the posts only contain opinions of one particular stock. If not, we remove the observation from our dataset. With these assumptions in place, we can treat sentiment classification as a classical text classification problem. [6] We could then use any supervised learning algorithm to solve this problem. In other reports, they have found Naïve Bayes classification, and Support vector machines to be especially successful. [9] [1].

When applying supervised learning algorithms it is key to find useful features. In the context of text classification, one typical approach is to use a bag-of-words approach for every sentence, that contains each word and its frequency. We gather every "possible" word as an element in a vector. For

example, one could expect to have a 1000 long vector with every position reflecting a word. This vector contains 1's for every word that is in a given post, and zero else were. This approach would ignore the order of the words, but it can be expanded to consider doublets or more. To get a more complex model we would have a longer list, and hence considering the sentiment values of even more words, with the cost of computing power. There are also other features containing information of the part of speech for each word, sentiment shifters, and so on. The crucial part of classifying the posts using a supervised model would be to have labeled data. With our data, we have some observations that are pre-classified as bullish/bearish. We would have to extend this by labeling thousands of posts as neutral as well. This would be a highly time-consuming task. Additionally, the pre-classified posts hold low quality and are not suited for training a model. For these reasons, we move on to consider an unsupervised approach.

A popular unsupervised method is the lexicon-based approach. This method involves using a pre-made lexicon containing words combined with a sentiment value. These lexicons are often made with more sophisticated machine learning models. There are multiple lexicons available, and some are specifically made for financial content. In the most basic way, one would classify a post by considering the sum of every sentiment value each word holds. If the sum is positive, we classify the post as positive. There are however extensions that deal with contextual valence shifters. Valence shifters being words that alter, intensify, or diminish the sentiment value of a polarized word. We will use this method to classify our text data.

To implement the lexicon-based approach in an optimized manner, we will use the R package "sentimentr"[11]. This package utilizes valence shifters when calculating the sentiment for a text. Usually, this leads to higher accuracy, with the cost of speed. The balance between accuracy and speed is what the author had in mind when creating the package. The equation for calculating the sentiment is best explained by the author in the given reference, but in short:

> Every sentence is broken down into an ordered bag-of-words, where every word is compared with a given sentiment lexicon. If there is a match we consider that word, two words before and two words after to be a sentiment "cluster". We give the cluster a value of $+1$ if the said word is positive and -1 if negative. Further, the cluster value can alter, increase and decrease depending on other words in the cluster. All these other words in the cluster are compared to a valence shifter lexicon. Finally, we calculate the sentiment as the sum of all clusters divided by

the square root of the word count in that sentence.

When using the lexicon-based approach, our performance will be dependent on which sentiment lexicon we are using. There is a variety of lexicons available, with different accuracy. We will try to analyze our data using three different lexicons; Augmented Jockers & Rinker's sentiment lexicon, Loughran-McDonald sentiment lexicon [7] and NTUSD-Fin sentiment lexicon [3]. The augmented Jockers-Rinker and Loughran-Mcdonald lexicons are both retrieved using the R package "lexicon"[10]. Out of these three, we want to pick the lexicon that has the highest accuracy of classification. To evaluate this we will use the pre-classified sentiments and consider the accuracy of how many correct classifications we can obtain using the lexicons. It is not beneficial for us to evaluate the lexicons using the whole data set of pre-classified sentiments, this would take too much time. Therefore we will instead consider a sample of 20000 posts.

Lexicon Evaluation

Lexicon	Accuracy	Non-neutral	Size
Jockers-Rinker	0.6008557	13088/20,000	11710
Loughran-McDonald	0.5932174	5691/20,000	2702
NTUSD-Fin	0.5746058	9577/20,000	8331

Table 2: The accuracy of the non-neutral classifications, on our sample of 20000 observations. Non-neutral is the amount of classifications that are classified as bullish or bearish to the total amount.

See *Table.2* for the evaluation of all three lexicons. Recall from section 4.1 that the pre-classified sentiments only consist of bullish or bearish values, while we classify messages as neutral as well. When using the sentiments in the prediction modeling we will ignore all neutral sentiments. Thus we only calculate the accuracy of the bullish or bearish classifications.

Often a post is classified as neutral if there is no polarized word in the post, matching a word in the given lexicon. Additionally, this could come from either the post or the lexicon being short. This is in line with *Table.2*, where we can see that the amount of non-neutral classification increase with the lexicon size. The accuracy is similar for all three lexicons, but they have apparent differences regarding the amount of non-neutral classifications. Since we are not taking the neutral sentiments into consideration, we want to keep the non-neutral as high as possible. Especially when this evaluation data only consists of non-neutral sentiments. From this evaluation, it is evident to pick the Jockers-Rinker lexicon. See *Table.8* and *Table.9* in Appendix A, for examples of the used lexicons. Note that the accuracy can not be

directly compared with other reports, because of the mentioned inaccurate labeling of our pre-classified sentiments. The accuracy is only a measure to evaluate the best fitting lexicon for our data.

We continue by further evaluating the results of using the Jockers-Rinker lexicon. See *Table.3* for a confusion matrix for the results. From this, we can calculate the accuracy of bullish sentiments as $5735/9474 \approx 60.5\%$ and bearish sentiments as $2129/3614 \approx 58.9\%$. We have similar accuracy for both classifications.

Confusion Matrix

	Predicted - Bullish	Predicted - Bearish	
Actual - Bullish	5735	3739	9474
Actual - Bearish	1485	2129	3614
	7220	5868	**13088**

Table 3: Confusion matrix from the evaluation of Jockers-Rinker lexicon.

We proceed by calculating sentiment for all unclassified posts using the lexicon-based classifier and Jockers-Rinker lexicon. We now have access to the sentiment of every post in our given time period. With this we can do some data analysis.

Using our sentiment data we can now do some basic data analysis to get some grasp of our contents. For all classifications, there are 65.7% bullish posts and 34.3% bearish. This suggests that there is an overall optimistic view of the stock market during our time period. Also, we can analyse when most of the posts are published in *Figure.4*. Most of the posts happen after the market has closed. Specifically, 13.6% are published before the market opens, 26.2% during market hours, and 60% after.

5.1.1 Aggregated Sentiment

Related reports suggests using a bullishness index [2][1] to aggregate the sentiment value for every day. We define this as

$$s_{t,j} = \ln\left[\frac{1 + BULL_{t,j}}{1 + BEAR_{t,j}}\right], \quad j \in \{1, 2, 3, 4\} \tag{5}$$

where $s_{t,j}$ is the bullishness index for stock j, $BULL_{t,j}$ the number of bullish messages at time t and $BEAR_{t,j}$ the bearish messages. The time series $s_{t,j}$ is then a index for the daily sentiment.

Figure 4: Histogram that shows the distribution of messages for all sentiment data. The blue highlighted bars corresponds to when the market is open for trade.

5.2 Modeling

We will start by fitting AR(p) models to our return series. These will be our base models. While the base models find linear relationships in the log returns, we will then see the effect of adding the daily sentiment in the following models. We denote the base models as $\text{MODEL}_{1,j}$ for $j \in \{1, 2, 3, 4\}$ where j corresponds to one of the stocks.

5.2.1 Stationarity

Before fitting any model we will control the assumption of stationarity in our time series. With an Augmented Dickey-Fuller test we will test the null hypothesis that a unit root is present in the series, with the alternative hypothesis of stationarity. From *Table.4* we can conclude that the null hypothesis of a present unit root can safely be rejected, with all p-values being less than 0.05, for both the log returns and the daily sentiment index. Note that we are using the series $s_{t,j}^{\text{combined}}$ containing both classified and pre-classified sentiments. For a test on the daily sentiments based on pre-classified text exclusively see *Table.10* in Appendix A.

Further, we complement this test by also using the KPSS-test, with the null

Augmented Dickey-Fuller Tests

Stock j	Statistic($r_{t,j}$)	P-value($r_{t,j}$)	Statistic($s_{t,j}$)	P-value($s_{t,j}$)
American E.	-10.1397	<0.01	-5.8445	<0.01
Boeing	-8.4095	<0.01	-5.7042	<0.01
Walt Disney	-8.0787	<0.01	-3.4258	0.0498
McDonald	-8.1772	<0.01	-5.6682	<0.01

Table 4: Augmented Dickey-Fuller tests for each stock on the lag order 7.

hypothesis of stationarity against the alternative hypothesis of a unit root. In *Table.5* we can see that the null hypothesis will not be rejected for $r_{t,j}$ since all p-values > 0.05, but not for the daily sentiments $s_{t,j}$. To fix this we will instead consider the first difference of $s_{t,j}$. This would mean that we instead consider the series

$$s^*_{t,j} = s_{t,j} - s_{t-1,j}$$

as the daily sentiment. However, we will keep the notation of $s_{t,j}$ to keep the language consistent. After differencing the sentiments the KPSS test was successful for the sentiments as well. For all series $s_{t,j}$ the p-values were larger than 0.1.

KPSS Test for Level Stationarity

Stock j	Statistic $r_{t,j}$	P-value $r_{t,j}$	Statistic $s_{t,j}$	P-value $s_{t,j}$
American E.	0.1552	>0.1	0.5002	0.0416
Boeing	0.1807	>0.1	3.6944	<0.01
Walt Disney	0.4243	0.067	1.8099	<0.01
McDonald	0.0585	>0.1	1.0365	<0.01

Table 5: KPSS tests on each stock, with truncation lag parameter being 5.

5.2.2 Base models

Having the stationarity checked we move on to fit an AR model on the log returns. First of all we will split the data into training sets and a test sets. We will set aside the first 75% observations to train our models, and the other to test.

To choose appropriate amount of lags we will first consider the partial autocorrelation functions seen in *Figure.5*. For American Express and Boeing we can see that 11 lags seem to be the best choice. For McDonald 12 lags, and for Walt Disney there is some tendency that points to using 15 lags as

well as 11.

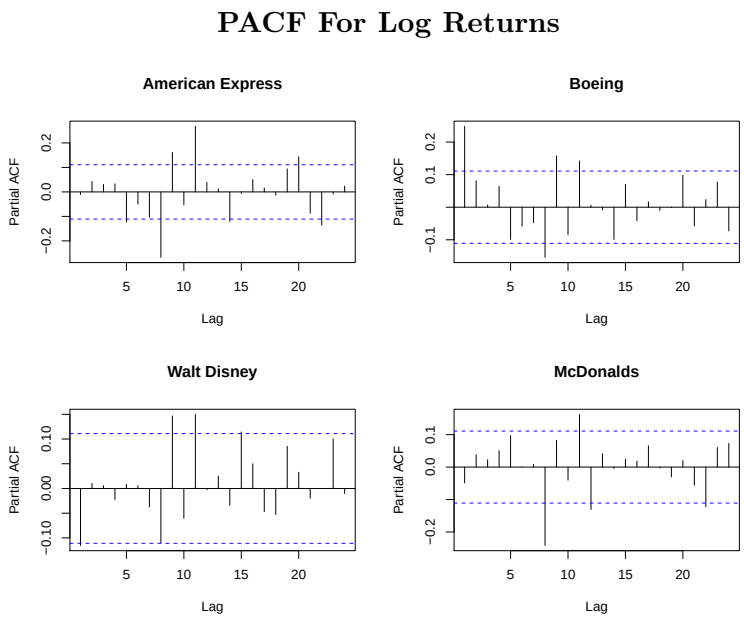

Figure 5: Partial autocorrelation function for the log returns $r_{t,j}$.

We can complement the PACF by also considering AIC. To do this we fit AR(p) models for each $r_{t,j}$ series with $p \in \{1, \cdots, 26\}$. We would then go on to choose the model with lowest AIC. In *Table.11* in Appendix A we can see the corresponding AIC to each value of the parameter p. The AIC points in the same way as what we concluded from the PACF. For Walt Disney the second lowest AIC was 11. We use $p = 11$ for all series except Walt Disney where $p = 12$. Our base models are then

$$\begin{aligned} \text{(American Express) MODEL}_{1,1} &: \text{AR(11)} \\ \text{(Boeing) MODEL}_{1,2} &: \text{AR(11)} \\ \text{(Walt Disney) MODEL}_{1,3} &: \text{AR(12)} \\ \text{(McDonalds) MODEL}_{1,4} &: \text{AR(11)}. \end{aligned}$$

We proceed by checking the significance of all lags being used. The models is estimated using maximum likelihood, thus the coefficients are going to be asymptotically normal distributed. We then calculate the corresponding z-statistics by dividing the coefficients for each model by their standard errors (estimation error), and then calculate the p-values. We successively set the insignificant ($p > 0.05$) estimates to zero and get the final base models

$$
\begin{aligned}
\text{MODEL}_{1,1} : \hat{r}_{t,1} &= -0.100481 r_{t-5,1} - 0.118500 r_{t-7,1} - 0.264070 r_{t-8,1} \\
&\quad + 0.135863 r_{t-9,1} + 0.230631 r_{t-11,1} \\
\text{MODEL}_{1,2} : \hat{r}_{t,2} &= 0.271069 r_{t-1,2} + 0.09993 r_{t-4,2} - 0.191571 r_{t-8,2} \\
&\quad + 0.153586 r_{t-9,2} - 0.118568 r_{t-10,2} + 0.144093 r_{t-11,2} \\
\text{MODEL}_{1,3} : \hat{r}_{t,3} &= 0.160780 r_{t-9,3} + 0.160062 r_{t-11,3} \\
\text{MODEL}_{1,4} : \hat{r}_{t,4} &= 0.095862 r_{t-5,4} - 0.234231 r_{t-8,4} + 0.146311 r_{t-11,4}.
\end{aligned}
$$

Additionally we check if there is any serial correlation in the residuals of the base models. This is done using a portmanteau test. See *Table.12* in Appendix A where we have used the Box-Ljung statistic $Q(m)$ for $m \in \{10, 15, 20\}$. For each stock we are unable to reject the null hypothesis on all levels of m. We assume the residuals to not be serially correlated, and thus the base models seems to be adequate.

5.2.3 Alternative Models

We go on to further create models that include the daily sentiment series $s_{t,j}$. To test if the sentiment have a significant effect on the return we will use a Granger's causality test[4]. This is a method first mentioned in 1969 by Clive Granger. Suppose we have two series X and Y. We then say that X "Granger Causes" Y if we are better off to predict Y with the information from X than without. Note that this is not equivalent with real causality, but a rather weaker statement. Formally we are going to fit a vector autoregression (VAR) model, and use a F-test to see if the sentiment estimates are significant. From the VAR model we are also able to test if the relationship is reversed, with return predicting the sentiment. The VAR model is evidently a linear model, but could also interpret non-linear relations if we perform non-linear transformations of the data. This is something we will skip. We will do these tests using both the pre-classified sentiments and the combined sentiments.

Now we will fit VAR(p) models to our data. The equation of the model is given by

$$
\begin{aligned}
r_t &= \sum_{j=1}^{p} \alpha_{t-j} r_{t-j} + \sum_{j=1}^{p} \beta_{t-j} s_{t-j} + u_{1t}, \\
s_t &= \sum_{j=1}^{p} \lambda_{t-j} r_{t-j} + \sum_{j=1}^{p} \gamma_{t-j} s_{t-j} + u_{1t}.
\end{aligned}
\tag{6}
$$

as we presented in section 3.8. In this first step we will tune the parameter p by choosing the VAR(p) model with lowest AIC. We got similar values of

p as we used in the base models, but slightly higher in some of the stocks. After we fit each VAR(p) model, we will do a F-test. The purpose is to test the null hypothesis that all coefficients for the lagged sentiments is equal to zero. I.e. $H_0 : \beta_t = 0$ for all $t \in \{1, \cdots, p\}$. The alternative hypothesis is $H_1 : \beta_t \neq 0$ for at least one $t \in 1, \cdots, p$. See *Table.6*, the daily sentiment turned out to show low significance when explaining the log returns. Only the pre-classified sentiments were significant for one stock, Walt Disney. The fact that the pre-classified sentiments were significant, and not the combined are signs of inaccuracy in our sentiment classification. For the other stocks the p-value was also slightly lower for the pre-classified sentiments. Further we can conclude that the sentiments Granger causes the log returns for Walt Disney. In the other stocks we consider the base model to be superior.

Granger's Causality Test (F-test)

Stock	Lags	P-values
American E.	11	0.6324
Boeing	12	0.6356
Walt Disney	7	0.0739
McDonalds	13	0.9830

(a) Combined Sentiments

Stock	Lags	P-values
American E.	13	0.989300
Boeing	12	0.588800
Walt Disney	5	**0.009929**
McDonalds	13	0.531200

(b) Pre-Classified Sentiments

Table 6: F-test for all lagged sentiment coefficients to equal zero.

We also tested the reverse relationship, i.e the second equation in Eq.(6). The null hypothesis that all coefficients for the lagged log returns were zero for explaining the sentiment series. We could not reject the null hypothesis for any stock, using either combined or pre-classified sentiments. For now, we continue by further analysing the VAR model for Walt Disney using pre-classified sentiments. First of all, we checked the adequacy of the model by using a portmanteau test on the residuals. For each value of $m \in \{10, 15, 20\}$ the p-values exceeded 0.15 which is above any critical value. Thus, we can not reject the null hypothesis of no serial correlation in the residuals. We continue by now comparing the prediction from the VAR(5) model and the base model ($\text{MODEL}_{1,3}$) on the test set. See *Figure.6* for the one-step-ahead forecasts. It is hard to reveal any result from the plot. The VAR model seems to predict the negative spikes better than the base model, whereas both underperform on the large positive spikes.

To simplify the result we can instead consider the classifications up/down. We classify each value to "Up" if it is positive and "Down" if negative. In this way we can create a confusion matrix again, to further analyse the models. Consider the confusion matrix in *Table.7*, we can calculate the accuracy

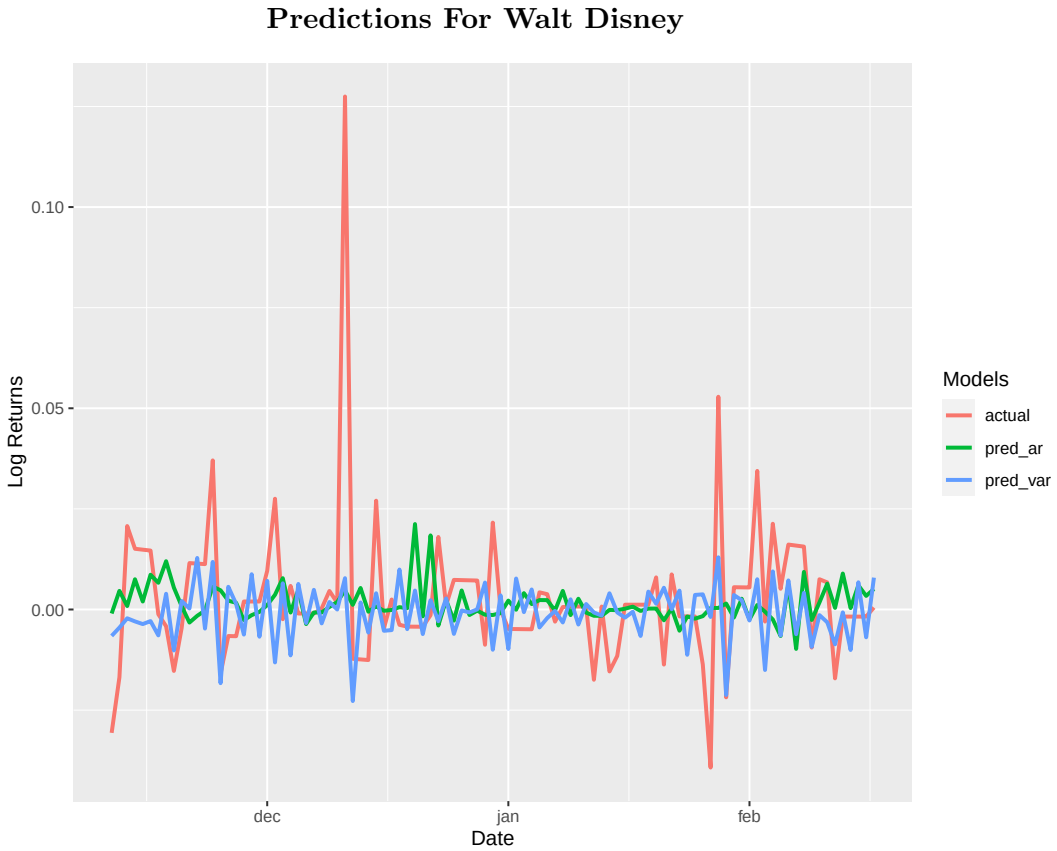

Figure 6: Predictions on the Walt Disney test set using the fitted AR and VAR model, compared with the actual log returns.

for each model. The VAR model has $49/99 \approx 49.5\%$ accuracy and the AR model also $\approx 49.5\%$. There is really no difference, and we are actually not better off using the daily sentiment compared with the base model. If were to guess the future return by only predicting that the future log return would be positive we would probably be better off, since there is generally more "Up" than "Down", $(54/99 \approx 54.5\%)$. This suggests that we can not consider the sentiment to Granger cause the log returns for Walt Disney either, when using classifications up/down.

Considering the other series we can also test if the returns can be explained by the instantaneous sentiment value. If we were to have the sentiment value s_t at time t, when predicting the r_t at time t, and their respective lagged values as well. When adding the instantaneous sentiment value to any existing model, its estimate was significant (p-value < 0.05) for all stocks and for using both combined and pre-classified sentiments. In this case, the relationship between the returns and sentiments was significant in

Prediction Confusion Matrix

Actual\Predicted	Up	Down
Up	24	30
Down	20	25

(a) VAR model

Actual\Predicted	Up	Down
Up	30	24
Down	26	19

(b) AR model

Table 7: Confusion matrix for predictions made by the VAR and AR model. The actual values are on the horizontal, and predicted on the vertical.

both directions. This suggests that we are able to explain each component r_t, s_t if we assume to have instant information.

6 Discussion

Our results goes in line with the efficient market hypothesis. For three of our stocks, the lagged sentiments proved to be insignificant when explaining the one-step-ahead log returns. For Walt Disney, the estimates were significant if we used the pre-classified sentiments. We also got significant estimates for the lagged log return, which certainly goes against the efficient market hypothesis. Even though the estimates were significant, it turned out to not be enough to give any potential profit, since the predictions of the test set were $\approx 50\%$. According to the efficient market hypothesis, the current price of a stock reflects all available information. I would argue that more complex information that requires substantial work to be done is not per se "available" for everyone. A more complex model, that also grasps nonlinear relations would be interesting to further analyse. Either way, our results suggest that we are not better off using the daily sentiment to predict future stock prices.

While the lagged sentiments showed weak/non significance, the instantaneous sentiments were significant in all stocks. This mean that the latest information possible is preferable. In our model we are using sentiments for the past day while we could also use the sentiments for all data before the market opens in each day. Too illustrate this idea further consider the *Figure.4* again, where we see the hourly message distribution. All information prior to the market open is actionable. As we concluded, this data consists of 13.6% messages of the whole day. To get access to this information we could simply consider the sentiment series s_t to be daily sentiments for a modified period of time. Where we instead consider a day to being when the market open. Thus we would get all information prior to market open in the lag-ℓ of the daily sentiment s_t. Essentially we would then get access to the most recent sentiment information, and might be better off predicting the future log returns.

Furthermore, the pre-classified sentiments turned out to be better than the combined sentiments. This indicates that our sentiment classifications are inaccurate. As I mentioned earlier our lexicon-based method is simple. For further analysis one would probably have to consider choosing a more complex method, to get more accurate classifications.

Our models are a very simplified view of reality. I did not expect the models to accurately predict the future. To get a more accurate prediction one would probably have to consider a model that can capture nonlinear relationships. It would also be a good idea to include more explanatory variables. Although, the goal was to see if daily sentiment could increase the predictability. The results indicated that we are not better off with adding the sentiment in general. However, for Walt Disney, it at least got significant estimates. This suggests that it could be useful in predicting the price of some type of company. Further research could be done to find what elements such companies share. I would say that the social media sentiment in general, is the voice of normal living people. Especially the authors at Stocktwits. For a stock to be influenced by people, it might have to be a smaller company. Now in retrospect, I regret choosing such similar sized companies, all from the Dow Jones. There is not evidence enough for us to claim that sentiment does not increase the predictability for any stock. All we can say is that in some cases it does not have a significant effect in predicting the log returns.

A Appendix

Jockers-Rinker Lexicon

word	sentiment
deflate	-0.50
criminals	-1.00
legacy	0.60
rollicking	0.60
haughty	-0.75
selfhumiliation	-0.25
unavailable	-0.50
grievous	-0.50
vociferous	-0.25
clueless	-0.50

Table 8: Example of Jockers-Rinker sentiment lexicon.

Valence Shifter Lexicon

word	y
acute	2
hardly	3
barely	3
only	3
kind of	3
sure	2
but	4
no	1
mightn't	1
severely	2

Table 9: Example of the valence shifter lexicon. Here y corresponds to a number 1-4, with 1 =Negator, 2 =Amplifier, 3 =De-amplifier and 4 = Adversative Conjunction.

Augmented Dickey-Fuller Tests

Stock j	Statistic ($s_{t,j}^{\text{pre-classified}}$)	P-value ($s_{t,j}^{\text{pre-classified}}$)
American E.	-5.4392	<0.01
Boeing	-5.5442	<0.01
Walt Disney	-3.1089	0.1089
McDonald	-4.3959	<0.01

Table 10: Augmented Dickey-Fuller tests for each stock on the lag order 7.

AIC

Stock	Lags
American Express	11
Boeing	11
Walt Disney	1
McDonald	12

Table 11: Amount of lags that corresponds to the lowest AIC for each log return series.

Portmanteau Test Of Residuals

Stock	m	Statistic	p-value
American Express	10	8.841	0.5473
	15	18.624	0.2313
	20	29.805	0.07307
Boeing	10	4.1231	0.9416
	15	11.125	0.7437
	20	14.466	0.8061
Walt Disney	10	7.0536	0.7204
	15	13.435	0.5687
	20	17.123	0.645
McDonalds	10	5.3181	0.8689
	15	11.196	0.7386
	20	14.939	0.7799

Table 12: Test of serial correlation in residuals for each base model, using the Box-Ljung test.

References

[1] Werner Antweiler and Murray Z. Frank. "Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards". In: *The Journal of Finance* 59.3 (2004), pp. 1259–1294. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2004.00662.x.

[2] Johan Bollen, Huina Mao, and Xiao-Jun Zeng. "Twitter Mood Predicts the Stock Market". In: *Journal of Computational Science* 2 (Oct. 2010). DOI: 10.1016/j.jocs.2010.12.007.

[3] Hen-Hsen Huang Chung-Chi Chen and Hsin-Hsi Chen. "NTUSD-Fin: A Market Sentiment Dictionary for Financial Social Media Data Applications." In: (May 2018). URL: http://nlg.csie.ntu.edu.tw/nlpresource/NTUSD-Fin/.

[4] C. W. J. Granger. "Investigating Causal Relations by Econometric Models and Cross-spectral Methods". In: *Econometrica* 37.3 (1969), pp. 424–438. ISSN: 00129682, 14680262. URL: http://www.jstor.org/stable/1912791.

[5] Denis Kwiatkowski et al. "Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?" In: *Journal of Econometrics* 54.1 (1992), pp. 159–178. ISSN: 0304-4076. DOI: https://doi.org/10.1016/0304-4076(92)90104-Y. URL: https://www.sciencedirect.com/science/article/pii/030440769290104Y.

[6] Bing Liu. "Sentiment Analysis: Mining Opinions, Sentiments, and Emotions." In: (2015), pp. 1–70. DOI: 10.1017/CBO9781139084789.

[7] Tim Loughran and Bill McDonald. "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks". In: *The Journal of Finance* 66.1 (2011), pp. 35–65. DOI: 10.1111/j.1540-6261.2010.01625.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.2010.01625.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2010.01625.x.

[8] Burton G. Malkiel and Eugene F. Fama. "EFFICIENT CAPITAL MARKETS: A REVIEW OF THEORY AND EMPIRICAL WORK". In: *The Journal of Finance* 25.2 (1970), pp. 383–417. DOI: https://doi.org/10.1111/j.1540-6261.1970.tb00518.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1970.tb00518.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1970.tb00518.x.

[9] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up? Sentiment Classification Using Machine Learning Techniques". In: *EMNLP* 10 (June 2002). DOI: 10.3115/1118693.1118704.

[10] Tyler Rinker. “Lexicons for Text Analysis”. In: (Mar. 2019). URL: https://cran.r-project.org/web/packages/lexicon/lexicon.pdf.

[11] Tyler Rinker. *sentimentr: Calculate Text Polarity Sentiment*. version 2.7.1. Buffalo, New York, 2019. URL: http://github.com/trinker/sentimentr.

[12] Ruey S. Tsay. *Analysis of financial time series*. 3rd ed. Hoboken: John Wiley & Sons, Inc., 2010, pp. 1–56, 389–416. ISBN: 9780470644553.