



Technische Universiteit Delft  
Faculteit Elektrotechniek, Wiskunde en Informatica  
Delft Institute of Applied Mathematics

Analyse van de impact van nieuws op de financiële  
markt

(Engelse titel: Analysis of the impact of news on the  
financial market)

Verslag ten behoeve van het  
Delft Institute of Applied Mathematics  
als onderdeel ter verkrijging

van de graad van

**BACHELOR OF SCIENCE**  
in  
**TECHNISCHE WISKUNDE**

door

**Sjoerd Dijkstra**

**Delft, Nederland**  
**Juli 2019**

Copyright © 2019 door Sjoerd Dijkstra. Alle rechten voorbehouden.





**BSc verslag TECHNISCHE WISKUNDE**

**“Analyse van de impact van nieuws op de financiële markt”  
(Engelse titel: “Analysis of the impact of news on the financial market)”**

Sjoerd Dijkstra

**Technische Universiteit Delft**

**Begeleider**

Dr.ir. J.H.M. Anderluh

**Overige commissieleden**

Drs. E.M. van Elderen

Prof.dr.ir. A.W. Heemink

Juli, 2019

Delft



# Contents

<b>1</b>	<b>Summary</b>	<b>5</b>
<b>2</b>	<b>Introduction</b>	<b>6</b>
2.1	Goal of research . . . . .	6
2.2	The stock market . . . . .	6
<b>3</b>	<b>Analyzing the news</b>	<b>7</b>
3.1	Financial news . . . . .	7
3.2	8-K reports . . . . .	7
<b>4</b>	<b>Text analysis to quantify sentiment in news articles</b>	<b>9</b>
4.1	Rule-based text analysis . . . . .	9
4.2	Automatic text analysis . . . . .	10
<b>5</b>	<b>Random forests</b>	<b>11</b>
5.1	Decision trees . . . . .	11
5.2	Over-fitting . . . . .	12
5.3	Random forests . . . . .	15
<b>6</b>	<b>Our approach</b>	<b>17</b>
6.1	Datamining . . . . .	17
6.2	Creating scores . . . . .	23
6.3	Text analysis . . . . .	24
<b>7</b>	<b>Results</b>	<b>25</b>
7.1	Justifying the split . . . . .	25
7.2	Success rates . . . . .	26
7.3	Analysis of results . . . . .	27
7.3.1	General trend . . . . .	27
7.3.2	The baseline . . . . .	27
7.3.3	Inconsistencies . . . . .	28
<b>8</b>	<b>Conclusion</b>	<b>28</b>
<b>9</b>	<b>Discussion</b>	<b>29</b>

# 1 Summary

In this work we set out to determine the impact, if any, of the analysis of news on stock price prediction, that is, are we able to predict stock movements more accurately on a consistent basis than a proposed baseline or random guessing on the basis of news' text analysis.

We considered a methodology to be more accurate if its success rate is greater than that of a baseline or random guess.

We considered a methodology to be consistently more accurate if the average of the success rates over a specified number of runs, say one hundred, is greater than that of a baseline or random guess.

As a result, we discovered that the analysis of news, though readily available with modern day technological advancements, does come paired with some problems.

1. The widespread availability of news has made it more difficult to find that news which is of importance to us, news can cover anything and everything.
2. The content of news can discuss events happening anywhere from far past to the far future, making consistent analysis difficult.
3. Most financial news sources tend to block any mass datamining attempts.

These problems can mostly be solved by making use of so-called 8-K reports. These reports only cover major events of companies sorted into nine different categories. The 8-K reports reduce the time interval the news impacts from the far past and far future to an interval of five business days, as the reports ought to be published within four business days. Finally, since companies are obligated to publish these reports by the U.S. securities and exchange commission, the reports are readily available and easily accessible through the U.S. securities and exchange commission website.

We can then use these texts and analyze them using a rule-based or automatic text analysis approach. However, the rule-based text approach, using lists of positive and negative words for the analysis, tends to be unreliable as text contains a plethora of challenging cases. This problem is solved by using an automatic text analysis, using predetermined scores for texts.

The form of automatic text analysis used, is a decision tree approach. Though single decision trees we construct have the characteristic to over-fit, we can construct random forests of decision trees on subsets of our input data to solve this problem.

For our analysis we looked at the stocks prices of Tesla, Microsoft, EA and Amazon, due to their varying  $\beta$  values. We gave scores to the texts of the 8-K reports using the stock price movement of the day of publishing. We did this for up to 4 business days prior to publishing as well. We also compensated for the market movements using the variable  $\beta$  for days zero to four. We gave scores from -1, 0 or 1 dependent on the price movement.

This generally resulted in success rates greater than our considered baseline of 33.33% of random guessing. The highest success rate for Tesla, Microsoft, EA and Amazon were in order: 73.09%, 100.00%, 88.64%, 84.95%.

From this, we can conclude that we can predict stock movements more accurately on a consistent basis than basic speculation or guessing based on the analysis of news' text.

## 2 Introduction

With news becoming more and more readily available with the rise of the internet and the concept of being able to make vast amounts of money, the stock markets start to peek our interests. Though people may hold back on taking part as this quick way to profit does come with its risks. If we were to eliminate this risk then the path to money is wide open. That is, with the right model.

If we are to do some research into the topic of predicting stock prices, we come across endless models analyzing every bit of numerical data stocks come accompanied with. We can ask ourselves, in this landscape filled with competition, how it is even possible to compete.

One way to stay ahead of the rest is considering the analysis of a new input or variable. One such relatively new input, is text. Not just any text though: news. If we could find a way to analyze the texts of news and use this information to predict stock prices or at least stock price movements, then this could just give us the edge on the more classical models for stock price prediction.

We should mention the Markowitz property of price models, which states that the stock price is a representation of every single bit of public knowledge with respect to that stock. As such, if we consider news as an input parameter, it would mean that the impact of news is represented in the stock price and as such if we see that new news becomes public, the market should react with a fluctuation in the stock price as a result. This then eventually ought to result in a stock price that now also represents the impact this news had. It is this movement in stock price that we wish analyze. We shall analyze this movement in the span of trading days.

As such we formulate our goal.

### 2.1 Goal of research

We wish to determine the impact, if any, of the analysis of news on stock price prediction, that is, are we able to predict stock movements more accurately on a consistent basis than a proposed baseline or random guessing on the basis of news' text analysis. We consider a methodology to be more accurate if its success rate is greater than that of a baseline or random guess. We consider a methodology to be consistently more accurate if the average of the success rates over a specified number of runs, say one hundred, is greater than that of a baseline or random guess.

### 2.2 The stock market

It is the American dream to become rich. What better gateway to this heaven does there exist than the stock market. The stock market is the term we use for the collection of markets and exchanges where we are able to namely buy and sell shares of publicly-held companies. As such, we can uncover the main way for us to make money making use of the stock market. If we are to buy a share for a certain price now and we see that when we sell the share at a later time for a greater price than we had bought it for, then we make money. This is called taking a long position in a stock. If, on the other hand, we are to buy a share for a certain price and we see that when we sell the share at a later time for a lesser price than we had bought it for, we see that we lose money. In order to counteract this and be able to also profit from these scenarios, the stock market allows for a so-called short position. To take a short position in a stock, is to sell the share at this moment, without actually having it in possession, then later on when the price of the share is lower we buy it back. As such we also profit from this scenario, if we predict it correctly.

### 3 Analyzing the news

In wanting to make predictions of stock price movements, we ought to first determine what source of news we wish to use as an input.

#### 3.1 Financial news

In order to determine what news source we would use for our analysis, we decided to look into a variety of factors. To be able to use our news in models, we need a readily and easily available source, we also need this source to be reliable.

The first sources that may come to mind are sources such as Bloomberg, Reuters, The Financial Times, etc. It is true that these sources are commonly seen as relevant and trustworthy sources for financial news. They indeed are credibly better sources than those such as Twitter, Facebook or any other social media, as these sources tend to focus on financial news. However, these sources come with some problems.

A problem we face with these news sources, is that the content of the news differs greatly. The news can cover anything and everything to do with the stocks and companies. This would have us first need to filter through this news to just use articles of importance to us and sort the news we use in certain categories.

Another problem we would have to cope with is the fact that news in the articles can cover facts happened in the past, happening in the present and even going to happen in the future. This then makes it significantly more difficult to determine whether or not the news article can be deemed positive, neutral or negative. This is due to the fact we cannot reliably analyze stock prices or other aspects of the stock between or on certain dates.

The most significant problem we face, is that these websites do not allow easy access to their content, so an approach using datamining would be near to impossible, as they block easy automation. That is, unless we are willing to mine the texts in mass by hand

Personal bias of writers of these articles can also form a problem. That is, if we wish to follow a rule-based text approach. This would mean that a growth in stock price for the article could be linked with a negative bias and word usage and we would thus always predict these changes wrongly.

#### 3.2 8-K reports

It is for this reason that we consider Form 8-Ks. These forms, following the definition as stated on the U.S. Securities and Exchange Commission website (SEC), are current reports that companies must file with the SEC to announce major events that shareholders should know about. The reports need to be published within 4 business days of the event taking place. The contents of the forms can be sorted into nine main categories, with each several sub categories. The contents of each of these categories is summarized:



Section 1	Registrant's Business and Operations
Item 1.01	Entry into a Material Definitive Agreement
Item 1.02	Termination of a Material Definitive Agreement
Item 1.03	Bankruptcy or Receivership
Item 1.04	Mine Safety - Reporting of Shutdowns and Patterns of Violations
Section 2	Financial Information
Item 2.01	Completion of Acquisition or Disposition of Assets
Item 2.02	Results of Operations and Financial Condition
Item 2.03	Creation of a Direct Financial Obligation or an Obligation under an Off-Balance Sheet Arrangement of a Registrant
Item 2.04	Triggering Events That Accelerate or Increase a Direct Financial Obligation or an Obligation under an Off-Balance Sheet Arrangement
Item 2.05	Costs Associated with Exit or Disposal Activities
Item 2.06	Material Impairments
Section 3	Securities and Trading Markets
Item 3.01	Notice of Delisting or Failure to Satisfy a Continued Listing Rule or Standard; Transfer of Listing
Item 3.02	Unregistered Sales of Equity Securities
Item 3.03	Material Modification to Rights of Security Holders
Section 4	Matters Related to Accountants and Financial Statements
Item 4.01	Changes in Registrant's Certifying Accountant
Item 4.02	Non-Reliance on Previously Issued Financial Statements or a Related Audit Report or Completed Interim Review
Section 5	Corporate Governance and Management
Item 5.01	Changes in Control of Registrant
Item 5.02	Departure of Directors or Certain Officers; Election of Directors; Appointment of Certain Officers; Compensatory Arrangements of Certain Officers
Item 5.03	Amendments to Articles of Incorporation or Bylaws; Change in Fiscal Year
Item 5.04	Temporary Suspension of Trading Under Registrant's Employee Benefit Plans
Item 5.05	Amendment to Registrant's Code of Ethics, or Waiver of a Provision of the Code of Ethics
Item 5.06	Change in Shell Company Status
Item 5.07	Submission of Matters to a Vote of Security Holders
Item 5.08	Shareholder Director Nominations

Table 1: Contents of 8-K reports sorted in categories (1)

Note. The table is copied from: "E-File Form 8-K" U.S. Securities and Exchange Commission, 10 Aug. 2012, [www.sec.gov/fast-answers/answersform8k.htm](http://www.sec.gov/fast-answers/answersform8k.htm).

Section 6	Asset-Backed Securities
Item 6.01	ABS Informational and Computational Material
Item 6.02	Change of Servicer or Trustee
Item 6.03	Change in Credit Enhancement or Other External Support
Item 6.04	Failure to Make a Required Distribution
Item 6.05	Securities Act Updating Disclosure
Section 7	Regulation FD
Item 7.01	Regulation FD Disclosure
Section 8	Other Events
Item 8.01	Other Events (The registrant can use this Item to report events that are not specifically called for by Form 8-K, that the registrant considers to be of importance to security holders.)
Section 9	Financial Statements and Exhibits
Item 9.01	Financial Statements and Exhibits

Table 2: Contents of 8-K reports sorted in categories (2)

Note. The table is copied from: "Form 8-K" U.S. Securities and Exchange Commission, 10 Aug. 2012, [www.sec.gov/fast-answers/answersform8k.htm](http://www.sec.gov/fast-answers/answersform8k.htm).

## 4 Text analysis to quantify sentiment in news articles

Though we have now determined a credible, usable and consistent news source, we need to analyze these reports. This is first done through a method we call parsing. This method takes the input text file of the 8-K report. We then split the review into sentences. After this we look at each sentence. We then first remove any non-letter characters from our text. After which we convert all upper case letters to lower case letters and split them. Then we have the option to remove stop words. Stop words are words usually filtered out during natural language processing, these are often the most common words in a language. Finally, we end up with a list of lists, with each list belonging to a sentence, where the contents of the list are the separate words in that list. We are now able to start processing this output.

### 4.1 Rule-based text analysis

The first method of processing this output is a rule-based text analysis. This analysis requires us to first give lists of words in certain categories. This analysis then uses the output, consisting of the lists of words, and determines in what frequency words of each category appear in the input text. Though, this method allows the use of multiple categories, it would be most practical for our purposes to use a mere two lists of words. Namely, we need only require a list of positive and a list of negative words. As using merely these two lists of words would then result us a score between -1 and 1. Where -1 indicates that the word usage in the text, based on the positive and negative word lists, to be completely negative, whereas a score of 1 indicates that the word usage in the text is completely positive. As such a score of 0 for the text would deem it to be neutral in word usage. We would then expect to see a positive score be related to a positive stock price movement and a negative score to negative stock price movement.

This method comes with its flaws, however. It is true that for the more simple cases, this form of text analysis, is capable of determining the sentiment. In the more challenging cases, such as negations, adverbials, sarcasm, etc, this model shows its true colours. For example, the sentence:

‘I do not dislike apples’, would be deemed negative, as it contains the negative words: ‘not’ and ‘dislike’. However, as we know the combination of the two words results in a positive sentiment. Furthermore, when analyzing financial texts, the lists of positive and negative words ought to contain such a vast amount of vocabulary, that this methodology seems unattainable for our purposes.

## 4.2 Automatic text analysis

The second method of processing this output we discuss, is the automatic text analysis. In this approach we would add predetermined scores to our 8-K reports. A text with a predetermined score of 1 would be considered a positive text, a text with a score of 0 would be considered neutral and finally a predetermined score of -1 would result in our text to be considered negative. Then this approach requires us to first throw out our ‘junk words’, words such as ‘the’, ‘a’, ‘with’, etc. Then using the remaining list of ‘important’ words in the text, we determine their frequencies. For each text we now construct a list of lists where each list contains a word and the frequency of that word. Then based on this list accompanied by the score, this algorithm constructs multiple decision trees into a so-called forest. This then concludes the training of the model. Then for testing, this model again receives the lists of words of the text we want to test. Then we again throw out the ‘junk words’ and make a list of the words and their corresponding frequencies. Finally, based on the constructed list, the algorithm follows the decision trees in the forest down, to determine whether the text scores are positive, negative or neutral.

Though this method does still suffer under some less frequent challenging language cases. It does not have to cope with the majority of the challenging cases the rule-based text approach had to deal with. As such it can be argued that for our purposes an automatic text analysis approach would be better.

## 5 Random forests

### 5.1 Decision trees

Decision trees are a method with which we can take a given input and, following a series of questions or conditions, follow each of the decisions we take down the tree-like structure. We continue doing so until we reach a result. An example is given in the image below:

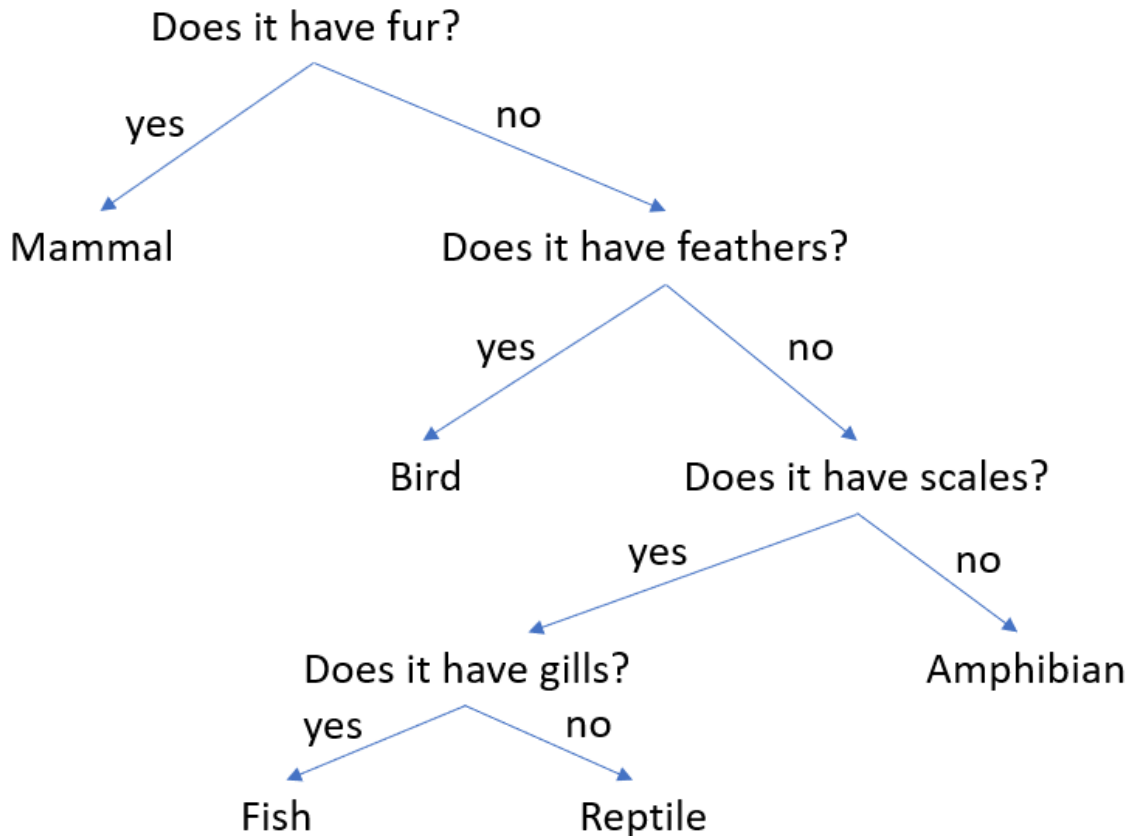


Figure 1: Example of a simple decision tree to determine animal type

As is seen in the figure, at each node where we are asked to make a decision, the node splits into different paths. In the ideal scenario we long for our decision to split the possible options into two. This binary splitting makes for an extremely effective and efficient decision making, even if we have large data to choose from. So, in wanting to achieve this idealistic binary splitting, we need to determine those questions at each node. Generally, the questions tend to take the form of axis-aligned splits in the data. At each node our decision tree would then split the data into two groups using a cutoff value within one of the input features.

## 5.2 Over-fitting

We shall run through an example. First we create a two dimensional data set consisting of four different clusters of points each belonging to a different colour:

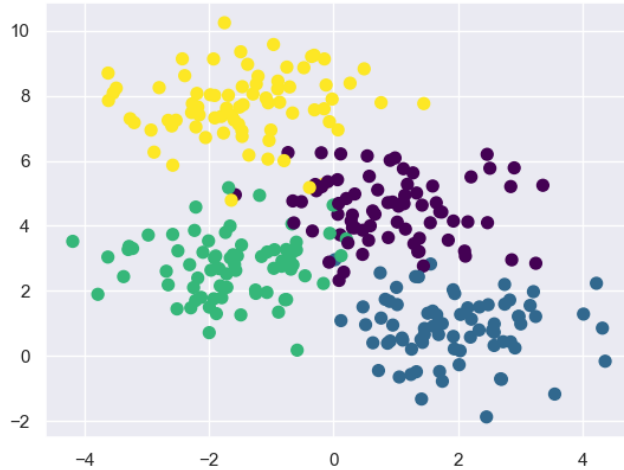


Figure 2: Our two dimensional data set consisting of points of four different colours

We now wish to use a decision tree to determine to which colour a new random point would belong. Our decision tree starts by splitting the data into two different sets. This split is chosen by means of the classification criteria. This criteria is defined as follows from '<https://scikit-learn.org/stable/modules/tree.html>':

Given training vectors  $x_i \in R^n$ ,  $i=1, \dots, l$  and a label vector  $y \in R^l$ , a decision tree recursively partitions the space such that the samples with the same label are grouped together.

If a target is a classification outcome taking on values of 0, 1, ..., K-1, for a node  $m$ , representing a region  $R_m$  with  $N_m$  observations, let

$$p_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

be the proportion of class k observations in node  $m$ .

Then measure of impurity used, is that of Gini:

$$H(X_m) = \sum_k p_{mk}(1 - p_{mk})$$

where  $X_k$  is the training data in node  $m$ .

The decision on where to split is then taken as follows:

Let the data at node  $m$  be represented by  $Q$ . For each candidate split  $\theta = (j, t_m)$  consisting of a feature  $j$  and threshold  $t_m$ , partition the data into  $Q_{left}(\theta)$  and  $Q_{right}(\theta)$  subsets

$$Q_{left}(\theta) = (x, y) | x_j \leq t_m$$

$$Q_{right}(\theta) = Q \setminus Q_{left}(\theta)$$

The impurity at  $m$  is computed using the Gini impurity function  $H(\cdot)$ :

$$G(Q, \theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta))$$

Then select the parameters that minimizes the impurity:

$$\theta^* = \operatorname{argmin}_{\theta} (G(Q, \theta))$$

Recurse for subsets  $Q_{left}(\theta^*)$  and  $Q_{right}(\theta^*)$  until the maximum allowable depth is reached,  $N_m < \min_{samples}$  or  $N_m = 1$ .

(scikit-learn developers. '1.10. Decision Trees'. Retrieved from <https://scikit-learn.org/stable/modules>,

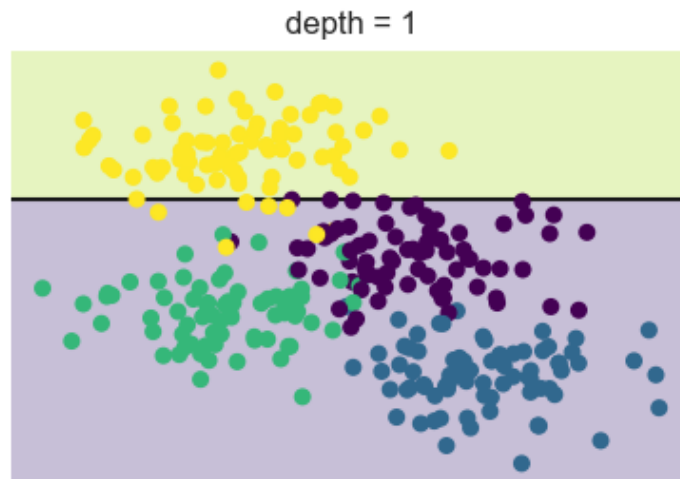


Figure 3: Depth 1 of decision tree for two dimensional data set

Then in depth two we wish to, if necessary, split the two different data sets into to more data sets:

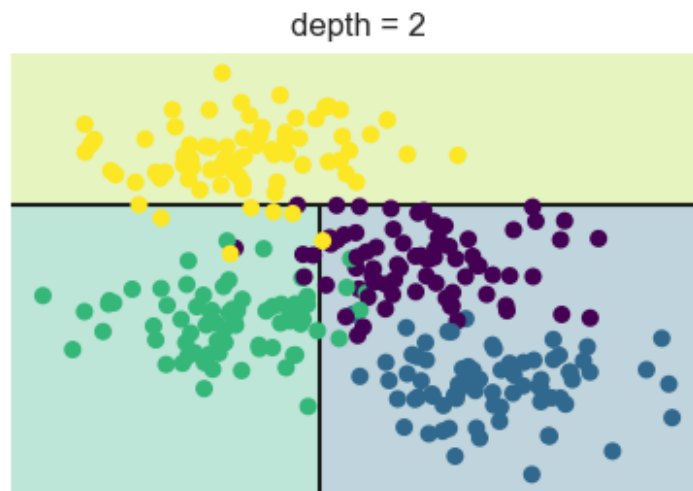


Figure 4: Depth 2 of decision tree for two dimensional data set

As we can see from the figure of the second depth of our decision tree construction, the bottom data set is split into two, while the top data set remains as one. This is due to the fact that in the top split of our data in depth 1, the area bordered off only contains points of the same colour class and as such it is not necessary to split this data any further. On the other hand, every single area containing more than one point of a class is again split into two. If we now are to continue this process further, we come across the following scenario:

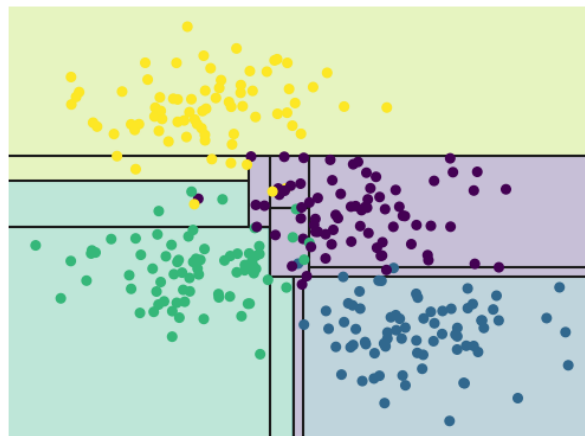


Figure 5: Depth 5 of decision tree for two dimensional data set

In this figure we come across some odd looking shapes and structures for the classification of the data points. This is not due to characteristics of the underlying data distribution and more of the noise properties of the data. Since we are fitting our model to the noise of our data, it is clear this methodology has the tendency to over-fit the given data. This is a general property of decision trees: decision trees tend to go deep and as such they start looking more into the details of the data rather than overall properties. This phenomenon becomes even clearer when

we look at two different decision trees constructed to subsets of the data:

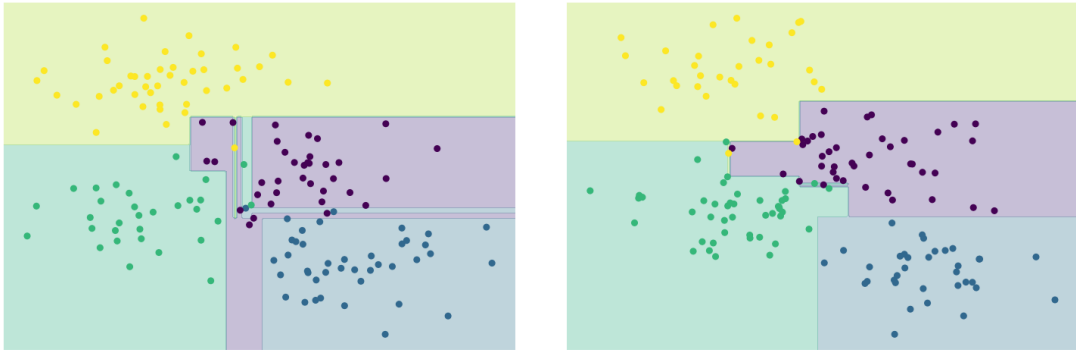


Figure 6: Decision trees constructed on two different subsets of the data

It is clear from the images of both decision trees, that the trees tend to differ in result in the borders between different clusters. This is due to the model being uncertain in these regions. In the more broader decisions, however, we see the two trees tend to overlap. The model is much more certain here.

From this result we may assume that making use of both of these trees, we can construct an even better result and if we can do so with two decision trees, then we should also be capable of doing so with a whole bunch of different trees to make the result even more accurate.

### 5.3 Random forests

We start the construction of 100 different decision trees, each one of which we construct on 80% of the data, this is called a random forest. Then to get the desired result, we average over each of these trees. The averaging is done through the averaging of the probabilistic predictions of the classifiers. So, a vector belongs to a certain class whenever the predicted class probability of this class is greatest of all classes. This method is often referred to as "soft voting". This then results in the following decision tree construction:



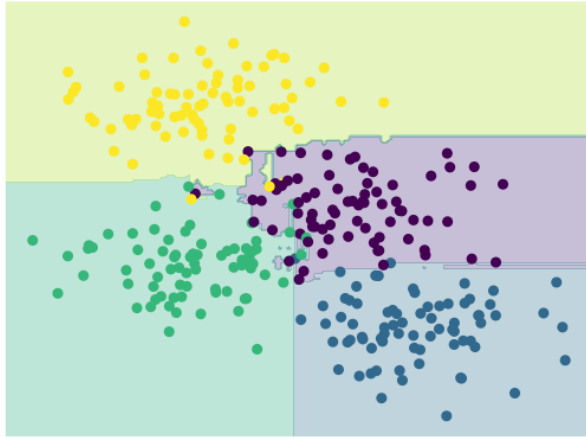


Figure 7: Average of 100 random decision trees

The resulting plot of the average of the one hundred random decision trees, shows us an intuitively more accurate model for the prediction of the classes of points, than the separate over-fitting decision trees.

## 6 Our approach

### 6.1 Datamining

Though we have now determined a credible, usable and consistent news source, which we are going to use, we need to analyze these reports. Doing this by hand would be near to impossible if we wish to have enough data to train and test our model with. It is for this reason we make an appeal to datamining. In our form of datamining, our goal is to get access to the texts of 8-K reports through the SEC website.

The first thing we ought to do, is to determine the exact URLs to webpages we use to base our datamining program on. The example we will be using, is that of the Tesla, Inc. company. We search the SEC for Tesla, Inc. and search up their 8-K reports. This yields us the following webpage:

The screenshot shows the SEC Edgar Search Results page for Tesla, Inc. (CIK# 0001318605). The page includes a navigation bar with 'Home | Latest Filings | Previous Page' and the SEC logo. The main heading is 'EDGAR Search Results' with a 'Search Results BETA View' button. Below the heading is a breadcrumb trail: 'SEC Home » Search the Next-Generation EDGAR System » Company Search » Current Page'. The company information section displays 'Tesla, Inc. CIK#: 0001318605 (see all company filings)' and provides details on SIC (3711), state location (CA), and fiscal year end (1231). It also lists business and mailing addresses in Palo Alto, CA. A filter section allows users to refine results by filing type (8-K), ownership, and limit results per page (100 entries). Below the filters, there is an RSS feed link and a 'Next 100' button. The main content is a table of filings with columns for Filing Date and File/Film Number.

Filings	Format	Description	Filing Date	File/Film Number
8-K	Documents	Current report, items 8.01 and 9.01 Acc-no: 0001564590-19-018876 (34 Act) Size: 71 KB	2019-05-13	001-34756 19819153
8-K	Documents	Current report, items 1.01, 2.03, 3.02, 8.01, and 9.01 Acc-no: 0001564590-19-016764 (34 Act) Size: 995 KB	2019-05-08	001-34756 19804919
8-K	Documents	Current report, items 1.01, 3.03, 8.01, and 9.01 Acc-no: 0001193125-19-135910 (34 Act) Size: 750 KB	2019-05-03	001-34756 19794126
8-K	Documents	Current report, items 2.02 and 9.01 Acc-no: 0001564590-19-012758 (34 Act) Size: 1 MB	2019-04-24	001-34756 19764582
8-K	Documents	Current report, items 5.02 and 8.01 Acc-no: 0001564590-19-012122 (34 Act) Size: 34 KB	2019-04-19	001-34756 19758340
8-K	Documents	Current report, items 2.02, 7.01, and 9.01 Acc-no: 0001564590-19-010743 (34 Act) Size: 41 KB	2019-04-04	001-34756 19730905
8-K	Documents	Current report, item 5.02 Acc-no: 0001564590-19-007669 (34 Act) Size: 25 KB	2019-03-14	001-34756 19679804
8-K	Documents	Current report, items 7.01 and 9.01 Acc-no: 0001564590-19-007080 (34 Act) Size: 30 KB	2019-03-11	001-34756 19670808

Figure 8: SEC search for Tesla, Inc. 8-K reports

As we can see from the screenshot taken from the website, we acquire a list of the hundred most recent published 8-K reports of Tesla accompanied by their filing date. We now first want to get

a hold of every single link to each filing in this link before we continue. In order to do so we inspect the pages elements, resulting in the following:

The image shows a screenshot of the U.S. Securities and Exchange Commission (SEC) website. The main content area displays search results for Tesla, Inc. (CIK# 0001318605). The results table lists two 8-K filings. The first filing is dated 2019-05-13 and has a file number of 001-34756. The second filing is dated 2019-05-08 and has a file number of 001-34756. The browser's developer tools window is open on the right, showing the HTML structure of the search results table. The table is rendered as a series of rows with alternating blue and white backgrounds. The first row is highlighted in blue. The HTML structure shows a table with columns for Filing Date, File/Film Number, and Description. The first row contains the following data: 2019-05-13, 001-34756, and Current report, items 8.01 and 9.01. The second row contains the following data: 2019-05-08, 001-34756, and Current report, items 1.01, 2.03, 3.02, 8.01, and 9.01.

Figure 9: Website inspection of SEC

We discover that the hyperlink extension to the first 8-K filing is saved in a html element, as we can see in the right hand side of the image. This element is saved in an element containing the '`<td>`' tag. Now we first use the link to this webpage to convert the website to html, with which we search for every '`<td>`' tag. This gives us an output as such:

```

[<td>Filter Results:</td>, <td><label for="type">Filing Type:</label><br><input id="type" name="type" size="10" tabindex="1" value="0-K"/></td>, <td><label
for="prior_to">Prior to:</label> (YYYYMMDD)<br><input id="prior_to" name="dateb" size="10" tabindex="2" value=""/></td>, <td><label for="include"
name="owner" tabindex="3" type="radio" value="include"/><label for="include">include</label><input checked="" id="include" name="owner" tabindex="4" type="radio"
value="exclude"/><label for="exclude">exclude</label><input id="only" name="owner" tabindex="5" type="radio" value="only"><label for="only">only</label>
<td><label for="count">Limit Results Per Page:</label><br>
<select id="count" name="count" tabindex="6">
<option value="10">10 Entries
    <option value="20">20 Entries
    <option value="40">40 Entries
    <option value="80">80 Entries
    <option selected="" value="100">100 Entries
</select>
</td>
<td style="text-align: middle;"><input type="submit" value="Search"/><br><input onclick="this.form.type.value=''" type="submit" value="Show All"/></td>
</td><td><label for="count">Limit Results Per Page:</label><br>
<select id="count" name="count" tabindex="6">
<option value="10">10 Entries
    <option value="20">20 Entries
    <option value="40">40 Entries
    <option value="80">80 Entries
    <option selected="" value="100">100 Entries
</select>
</td>
<td style="text-align: middle;"><input type="submit" value="Search"/><br><input onclick="this.form.type.value=''" type="submit" value="Show All"/></td>
<td>Items 1 - 100 <a href="/cgi-bin/browse-edgar?action=getcompany&CIK=0001318605&type=8-K
%25&dateb=&owner=exclude&count=100&output=atom">img alt="EDGAR USA Search" src="/images/edgar_button_usasearch.png" style="border-style: none"/></a>, <a href="/cgi-bin/browse-edgar?action=getcompany&CIK=0001318605 Filings" border="0" src="/images/rss-feed-icon-14x14.png"/>
RSS Feed</a></td>, <td style="text-align: right;"><input onclick="parent.location='/cgi-bin/browse-edgar?action=getcompany&CIK=0001318605&type=8-K
%25&dateb=&owner=exclude&start=100&count=100'" type="button" value="Next 100"/></td>, <td nowrap="nowrap"><a href="/
Archives/edgar/data/1318605/000156459019018876/0001564590-19-018876-index.htm" id="documentsbutton"> Documents</a></td>, <td class="small">Current report, items 8.01 and
9.01
<br>Acc-no: 0001564590-19-018876 (34 Act) Size: 71 KB </td>, <td nowrap="nowrap"><a href="/cgi-bin/browse-edgar?
action=getcompany&filenum=001-34756&owner=exclude&count=100">001-34756</a><br>19819153 </td>, <td nowrap="nowrap"><a href="/
Archives/edgar/data/1318605/000156459019018876/0001564590-19-018876-index.htm" id="documentsbutton"> Documents</a></td>, <td class="small">Current report, items
1.01, 2.03, 3.02, 8.01, and 9.01

```

Figure 10: Output containing all '`<td >`' tags

As we can see, the list indeed contains our desired link extensions, however, the list also contains a lot of tags that are of no use to us. Thus we need to make our search more specific by searching for another defining tag for our link extension.

When looking back at the website inspection, we can see that the tag in which the link is directly saved, has the '`<a >`' tag. We now need to search our list of tags for elements with the '`<a >`' tag. This results in the following:


```

[<a href="/index.htm">Home</a>, <a href="/cgi-bin/browse-edgar?action=getcurrent">Latest Filings</a>, <a href="javascript:history.back()">Previous Page</a>, <a href="/
index.htm">img alt="SEC Seal" border="0" src="/images/sealTop.gif"/></a>, <a href="/cgi-bin/browse-edgar?action=getcompany&CIK=0001318605&type=8-K
%25&dateb=&owner=exclude&count=100&output=xml">img alt="EDGAR USA Search" src="/images/edgar_button_usasearch.png" style="border-style: none"/></a>, <a
href="/index.htm">SEC Home</a>, <a href="/edgar/searchedgar/webusers.htm">Search the Next-Generation EDGAR System</a>, <a href="/edgar/searchedgar/
companysearch.html">Company Search</a>, <a href="/cgi-bin/browse-edgar?action=getcompany&CIK=0001318605&owner=exclude&count=100">0001318605 (see all company
filings)</a>, <a href="/cgi-bin/browse-edgar?action=getcompany&SIC=3711&owner=exclude&count=100">3711</a>, <a href="/cgi-bin/browse-edgar?
action=getcompany&State=CA&owner=exclude&count=100">CA</a>, <a href="/cgi-bin/own-disp?action=getissuer&CIK=0001318605">insider transactions</a></a>,
<a href="/cgi-bin/browse-edgar?action=getcompany&CIK=0001318605&type=8-K%25&dateb=&owner=exclude&start=100&count=100&output=atom">img
align="top" alt="0001318605 Filings" border="0" src="/images/rss-feed-icon-14x14.png"/> RSS Feed</a>, <a href="/Archives/edgar/data/
1318605/000156459019018876/0001564590-19-018876-index.htm" id="documentsbutton"> Documents</a>, <a href="/cgi-bin/browse-edgar?
action=getcompany&filenum=001-34756&owner=exclude&count=100">001-34756</a>, <a href="/Archives/edgar/data/1318605/000156459019018876/0001564590-19-018876-
index.htm" id="documentsbutton"> Documents</a>, <a href="/cgi-bin/browse-edgar?action=getcompany&filenum=001-34756&owner=exclude&count=100">001-34756</a>, <a href="/Archives/edgar/data/1318605/000156459019012758/0001564590-19-012758-
index.htm" id="documentsbutton"> Documents</a>, <a href="/cgi-bin/browse-edgar?action=getcompany&filenum=001-34756&owner=exclude&count=100">001-34756</a>, <a href="/Archives/edgar/data/1318605/000156459019012122/0001564590-19-012122-index.htm" id="documentsbutton"> Documents</a>, <a href="/cgi-bin/browse-edgar?
action=getcompany&filenum=001-34756&owner=exclude&count=100">001-34756</a>, <a href="/Archives/edgar/data/1318605/000156459019010743/0001564590-19-010743-
index.htm" id="documentsbutton"> Documents</a>, <a href="/cgi-bin/browse-edgar?action=getcompany&filenum=001-34756&owner=exclude&count=100">001-34756</a>, <a href="/Archives/edgar/data/1318605/000156459019007669/0001564590-19-007669-index.htm" id="documentsbutton"> Documents</a>, <a href="/cgi-bin/browse-edgar?
action=getcompany&filenum=001-34756&owner=exclude&count=100">001-34756</a>, <a href="/Archives/edgar/data/1318605/000156459019007080/0001564590-19-007080-
index.htm" id="documentsbutton"> Documents</a>, <a href="/cgi-bin/browse-edgar?action=getcompany&filenum=001-34756&owner=exclude&count=100">001-34756</a>, <a href="/Archives/edgar/data/1318605/000156459019006788/0001564590-19-006788-index.htm" id="documentsbutton"> Documents</a>, <a href="/cgi-bin/browse-edgar?
action=getcompany&filenum=001-34756&owner=exclude&count=100">001-34756</a>, <a href="/Archives/edgar/data/1318605/000156459019005560/0001564590-19-005560-
index.htm" id="documentsbutton"> Documents</a>, <a href="/cgi-bin/browse-edgar?action=getcompany&filenum=001-34756&owner=exclude&count=100">001-34756</a>, <a href="/Archives/edgar/data/1318605/000156459019001597/0001564590-19-001597-index.htm" id="documentsbutton"> Documents</a>, <a href="/cgi-bin/browse-edgar?
action=getcompany&filenum=001-34756&owner=exclude&count=100">001-34756</a>, <a href="/Archives/edgar/data/1318605/000156459019001399/0001564590-19-001399-
index.htm" id="documentsbutton"> Documents</a>, <a href="/cgi-bin/browse-edgar?action=getcompany&filenum=001-34756&owner=exclude&count=100">001-34756</a>, <a href="/Archives/edgar/data/1318605/000156459019000740/0001564590-19-000740-index.htm" id="documentsbutton"> Documents</a>, <a href="/cgi-bin/browse-edgar?
action=getcompany&filenum=001-34756&owner=exclude&count=100">001-34756</a>, <a href="/Archives/edgar/data/1318605/000156459019000190/0001564590-19-000190-
index.htm" id="documentsbutton"> Documents</a>, <a href="/cgi-bin/browse-edgar?action=getcompany&filenum=001-34756&owner=exclude&count=100">001-34756</a>, <a href="/Archives/edgar/data/1318605/000156459019000013/0001564590-19-000013-index.htm" id="documentsbutton"> Documents</a>, <a href="/cgi-bin/browse-edgar?

```

Figure 11: Output containing all '`<a >`' tags

It is here we can start to see our desired result. We see a pattern appear in the list, containing a lot of tags with very similar structure. These tags continue on to be the hundred link extensions we need. So, we can now acquire our extensions by iterating from the 14th element up until the 114th element. However, to keep this program working for lists not consisting of 100 elements, we set the program to iterate up until the four to last element as this is consistent for this website. Having acquired the link extensions we add these to the base link of 'https://www.sec.gov/', which results in the link to the filing. This link then takes us to the following website:



Home | Latest Filings | Previous Page

## U.S. Securities and Exchange Commission

### Filing Detail

Search the Next-Generation EDGAR System

SEC Home » Search the Next-Generation EDGAR System » Company Search » Current Page

**Form 8-K - Current report:** **SEC Accession No.** 0001564590-19-018876

Filing Date	Period of Report	Items
2019-05-13	2019-05-13	Item 8.01: Other Events Item 9.01: Financial Statements and Exhibits
<b>Accepted</b> 2019-05-13 16:59:18		
<b>Documents</b> 4		

Document Format Files

Seq	Description	Document	Type	Size
1	8-K	<a href="#">tsla-8k_20190513.htm</a>	8-K	36632
2	EX-8.1	<a href="#">tsla-ex81_14.htm</a>	EX-8.1	13782
3	EX-8.2	<a href="#">tsla-ex82_55.htm</a>	EX-8.2	13374
4	EX-99.1	<a href="#">tsla-ex991_15.htm</a>	EX-99.1	6848
	Complete submission text file	<a href="#">0001564590-19-018876.txt</a>		71782

**Tesla, Inc. (Filer) CIK: 0001318605 (see all company filings)**

IRS No.: 912197729   State of Incorp.: DE   Fiscal Year End: 1231 Type: 8-K   Act: 34   File No.: 001-34756   Film No.: 19819153 SIC: 3711 Motor Vehicles & Passenger Car Bodies Assistant Director 5	Business Address 3500 DEER CREEK RD PALO ALTO CA 94304 650-681-5000	Mailing Address 3500 DEER CREEK RD PALO ALTO CA 94304
--	--	---

Figure 12: First 8-K report filing of Tesla, Inc. on the SEC website

From this website we now desire three elements. We desire the date of the filing, the time of the filing and the .txt submission file. So, again we inspect website elements to find the date under the following inspection:

The screenshot shows the SEC website's 'Filing Detail' page for Tesla, Inc. (Form 8-K, SEC Accession No. 0001564590-19-018876). The page displays filing information, document format files, and company details. The browser's developer tools are open, showing the HTML structure. The 'contentDiv' contains a 'formDiv' with a 'formContent' class. Inside 'formContent', there is an 'info' element containing the filing date and time: '2019-05-13 16:59:18'.

Seq	Description	Document	Type	Size
1	8-K	tsla-8k_20190513.htm	8-K	36632
2	EX-8.1	tsla-ex81_14.htm	EX-8.1	13782
3	EX-8.2	tsla-ex82_55.htm	EX-8.2	13374
4	EX-99.1	tsla-ex991_15.htm	EX-99.1	6848
		Complete submission text file	0001564590-19-018876.txt	71782

Figure 13: SEC website inspection of date and time of filing element

As we can see in the figure, the dates are saved under the '`<div>`' tag, but we ought to also take note of the fact that in the tag the date is saved under 'class="info"'. So, we now iterate over all of the websites live and save all the strings contained under the combination of those two tags in a list. An example of this is given:

```
[ '<div class="info">2019-05-13 16:59:18</div>' ]
```

Figure 14: Example of date and time of filing element in list

Now in order to acquire the third and final element we require from this website, we again inspect the page for the link to the .txt-file of the 8-K filing:

The screenshot shows the SEC Filing Detail page for Tesla, Inc. (Form 8-K) filed on 2019-05-13. The page includes a table of document format files and a section for the filer's information. An element inspector on the right highlights a link to a .txt file within a table cell.

**Form 8-K - Current report:** SEC Accession No. 0001564590-19-018876

Filing Date	Period of Report	Items
2019-05-13	2019-05-13	Item 8.01: Other Events
Accepted 2019-05-13 16:59:18		Item 9.01: Financial Statements and Exhibits
Documents 4		

**Document Format Files**

Seq	Description	Document	Type	Size
1	8-K	tsla-8k_20190513.htm	8-K	36632
2	EX-8.1	tsla-ex81_14.htm	EX-8.1	13782
3	EX-8.2	tsla-ex82_55.htm	EX-8.2	13374
4	EX-99.1	tsla-ex991_15.htm	EX-99.1	6848
Complete submission text file				71782
		0001564590-19-018876.txt		

**Tesla, Inc. (Filer) CIK: 0001318605 (see all company filings)**

Business Address: 3500 DEER CREEK RD, PALO ALTO CA 94304  
 Mailing Address: 3500 DEER CREEK RD, PALO ALTO CA 94304

JRS No.: 912197729 | State of Incorp.: DE | Fiscal Year End: 1231  
 Type: 8-K | Act: 34 | File No.: 001-34756 | Film No.: 94304  
 SIC: 3711 Motor Vehicles & Passenger Car Bodies Assistant Director 5

The element inspector shows the following HTML structure for the highlighted link:

```

<td scope="row">
  &nbsp;&nbsp;&nbsp;</td>
<td scope="row">
  Complete submission text file</td>
<td scope="row">
  <a href="/Archives/edgar/data/1318605/000156459019018876/0001564590-19-018876.txt">
    0001564590-19-018876.txt</a> == $
  </td>
<td scope="row">
  &nbsp;&nbsp;&nbsp;</td>
<td scope="row">
  71782</td>
</tr>
</tbody>
</table>
</div>
</div>
<!-- END DOCUMENT DIV -->
<!-- START FILER DIV -->
<div id="filerDiv">...</div>
<!-- END FILER DIV -->
</div>
<script type="text/javascript" id="...</script>
  
```

Figure 15: Inspection of link to .txt-file

We now see a similar scenario to when we first acquired the link to this website; the link extension to the .txt-file is saved in a ' < a > ' tag within a ' < td > ' tag as the 'href'. This scenario does differ slightly though, as we do not need a whole list of links, we only require the one. Thus now we again search for the ' < a > ' tags in the list of ' < td > ' tags and in this list we need only find the link that contains the extension '.txt'. Then we convert the website to a html and save the text accompanied by its corresponding date. In short, we now have an output of a list containing lists of a filing date and time paired with the text of the 8-K report.

## 6.2 Creating scores

The next step in our input processing, is to acquire the scores necessary for us to start using an automatic text analysis algorithm. However, there is no database containing the sentiment analysis scores of these reports, so we need to determine our own methodology to give these texts a score of -1, 0 and 1. We have decided to start with the first and simplest of the ways to make a profit as discussed earlier. We shall give the texts scores based on the movement in the stocks price.

In order to achieve such a feat, we need to download and get access to the historical stock market prices of Tesla, Inc. We do this by downloading all the necessary data from '<https://finance.yahoo.com/>'. Following this we start analyzing our date and time input. We want our score to be determined by the difference as a result of the 8-K report. As such we desire the difference to be determined between the stock price before and after the publishing of the report.

In the financial world the so called trading day starts at 9:30 a.m. and ends at 4:00 p.m. Then the price of the stock at 9:30 a.m. is called the opening price and the price at 4:00 p.m. is called the closing price.

If a report is published on the trading day itself, we consider the difference between the closing price and the opening price divided by the opening price of that day to determine the percentual difference. If a report is published before the trading day starts, we take the difference between the opening price of that day and the closing price of the previous day. If a report is published after the trading day ends, we take the difference between the closing price of that day and the opening price of the following trading day. In the financial world, however, we have certain days in the year, such as American national holidays and weekends, on which the stock market does not open. If a report is published in such a 'gap', we just take the difference between the latest closing price of the stock and the first opening price. The price change is determined using the following formulas, depending on the time of publishing:

$$\Delta\text{Price (\%)} = \frac{\text{Open Price} - \text{Close Price}}{\text{Open Price}} * 100\%$$

$$\Delta\text{Price (\%)} = \frac{\text{Close Price} - \text{Open Price}}{\text{Close Price}} * 100\%$$

Using the determined percentual differences in stock price for each 8-K report, we give the text accompanying that difference a score of -1 if the difference is less than -1.00%, a score of 1 if the difference is greater than 1.00% and finally a score of 0 if the difference is between -1.00% and 1.00%. The neutral score is given due to the fact, that in order to be profitable, we need the stock price to change substantially due to transaction costs and in general, a change of less than a percent is not of much significance.

Since 8-K reports need to be published within 4 days of the event taking place, we also ought to analyze the stock price movements prior to the publishing date itself. Thus we summarize as follows:

$$\text{8-K report score} = \begin{cases} 1, & \text{if price change} > 1.00\% \\ 0, & \text{if } -1.00\% \leq \text{price change} \leq 1.00\% \\ -1, & \text{if price change} < -1.00\% \end{cases}$$

It can be argued that since the entire stock market also moves in value that we need to compensate for these shifts in the entire market. In order to compare these results, we also construct a data set that does compensate for this change.



We download a data set again from `https://finance.yahoo.com/`, now containing the index prices of the S&P500. The S&P500 is stock index in the U.S. consisting of the 500 biggest American companies according to their market capitalization. It is generally considered to be a reliable source in determining the status of the entire American stock market. So, in compensating for the changes in the entire stock market, we can use the price changes in the S&P 500 index. In a similar manner to determining the differences in price of the single stock, we take the difference between closing and opening prices before and after the publishing of the 8-K report. Having now acquired two sets of percentage, we multiply the difference of the S&P500 index with the beta between the stock and the market from that of the stock. The introduction of the variable beta is due to different stocks reacting in different manners to price movements. If a stock has a beta of 1.50, then, in general, the asset tends to move up and down with the market, meaning if the market sees a growth of 10%, then this stock tends to move up with 15%. The beta is calculated with the following formula:

$$\beta = \frac{Cov(r_M, r_a)}{Var(r_M)}$$

In this formula,  $r_M$  indicates the return of the market,  $r_a$  the return of the asset,  $Cov(.,.)$  being the covariance function and  $Var(.)$  is the variance. After having determined the value of beta, we subtract this percentage from the movement of the stock itself. So, say if the Tesla, Inc. stock has a growth of 1.00% and the S&P500 index changes with -0.50% and the beta value is 0.75, then the "real" difference is determined to be 1.38%. To put it in formula form:

$$\text{Real difference (\%)} = \text{Stock difference (\%)} - \beta * \text{Market difference (\%)}$$

In the given example, the score for that 8-K report would be a 1 instead of a 0 as a result.

### 6.3 Text analysis

Having come to the conclusion that automatic text analysis is the more desired approach, because of its nature to not succumb to as many challenging things in natural language. We now wish to start the construction of decision trees following the methodology discussed in chapter 5.

We take the input, now consisting of a list containing every important word and the count of that word in the 8-K report, and start the construction of random decision trees. The classes we want to distinguish between are 1, 0 and -1.

So following the methodology in chapter 5, we now construct a decision tree using a random subset of 80% of the input data as the training vectors. These vectors have a maximum of 5000 dimensions, where each dimension represents a specific word and the number in this dimension represents the frequency of this word in the text. As such an example of a possible threshold for the decision tree could be if a certain word appears less than five times in the text.

We construct 2000 different decision trees in our random forest, based on 80% of the input data, to not succumb to over-fitting. The resulting tree we then use to determine our accuracy on the other 20% of the data. Due to the inherent randomness in the construction of our forest, we take the average over 2000 different runs to determine our accuracy.

## 7 Results

When making use of our constructed programs for first analyzing the 8-K reports and then running this analysis through our automatic text analysis program, we can start analyzing our results. These results are all achieved with a random forest consisting of 2000 decision trees.

### 7.1 Justifying the split

In the final section on our model, we discuss the use of 80% of the input data as our training set, where we use 20% of the input data to test on. We shall justify this by running our model for varying sizes of the training set against the success rate:



Figure 16: Success rates of Amazon against the training set size when compensating for the market change on the 4 day look-back covering 199 reports

From the figure it is made clear that our success rate for the prediction of the price movements of the Amazon stock is greatest when we decide to split our data into a training set on 80% of the data and 20% of the data as a test set. We shall use this as justification to only consider this split of the data for all the following results.

## 7.2 Success rates

We analyzed the 8-K reports of Tesla, Microsoft, EA and Amazon due to their varying values for  $\beta$ . In the tables the "Success rate without" tag indicates that these are the success rates without the compensation for the market changes, where "Success rate with" tag contains the success rates with compensation for the market changes.

Days prior to publishing	Success rate without	Success rate with
-2	52.03%	37.47%
-1	77.06%	74.50%
0	33.50%	33.50%
1	62.69%	62.81%
2	54.38%	52.88%
3	73.09%	73.06%
4	65.63%	65.63%
5	74.09%	74.16%
6	84.34%	84.31%

Table 3: Success rates of analysis of 8-K reports belonging to Tesla, Inc. ( $\beta = 0.03$  and  $\#reports = 150$ )

Days prior to publishing	Success rate without	Success rate with
-2	94.20%	96.67%
-1	80.00%	83.33%
0	56.67%	63.17%
1	73.33%	96.67%
2	93.33%	100.0%
3	90.00%	93.33%
4	91.43%	100.00%
5	87.80%	89.97%
6	68.80%	70.00%

Table 4: Success rates of analysis of 8-K reports belonging to Microsoft ( $\beta = 1.05$  and  $\#reports = 150$ )

Days prior to publishing	Success rate without	Success rate with
-2	79.55%	90.91%
-1	77.59%	77.86%
0	56.91%	59.68%
1	71.30%	77.84%
2	77.82%	79.55%
3	78.61%	88.64%
4	74.30%	80.89%
5	88.43%	95.45%
6	82.91%	88.50%

Table 5: Success rates of analysis of 8-K reports belonging to EA ( $\beta = 1.23$  and  $\#reports = 220$ )

Days prior to publishing	Success rate without	Success rate with
-2	70.35%	71.78%
-1	65.13%	61.30%
0	47.18%	47.40%
1	67.50%	70.00%
2	73.50%	72.08%
3	77.75%	73.43%
4	84.95%	82.55%
5	72.10%	73.60%
6	75.00%	70.00%

Table 6: Success rates of analysis of 8-K reports belonging to Amazon ( $\beta = 1.73$  and  $\#reports = 199$ )

## 7.3 Analysis of results

### 7.3.1 General trend

In general, we see a trend. On the day of publishing our success rates are substantially lacking in comparison to the success rates of prior day assessment. From this it is made clear that the effect these events in the 8-K reports have on the stock price, takes place before the publishing of the reports. So the market tends to have the information published in the reports earlier and as such reacts earlier. As such, if we were to have access to this information when the event actually takes place or when the information on the event goes public, then we can consider the analysis of the days prior to be realistic. If this were to be the case and we are to take 33.33% as the baseline to a random guess if the change in stock price results in a -1, 0 or 1, then we acquire a significantly better result than the baseline. In fact, most analysis of the 8-K reports resulted in a success rate greater than this baseline with the exception of the 0-day look-back of the Tesla stock.

With this insight, one could assume that the event taking place in the 8-K report has an effect on the stock price movements for a period of 4 business days prior to the publication. This effect would be greatest 4 days prior, assuming the event takes place at this time.

In theory, taking into consideration that the 8-K reports ought to be published within four business days of the event taking place, we would expect our predictions on say a 5 or 6 day look-back to be significantly lower than that of the predictions prior. From our results, however, it is made clear that these predictions do show high success rates, some of them even exceeding the predictions within the given time interval of 8-K reports.

### 7.3.2 The baseline

In the previous section we discussed a baseline of a random guess of 33.33%. Though this indeed forms a baseline for basic guessing, it may not seem very satisfactory. One could justify predicting better than random guessing if they have knowledge of the stock. If the stock shows more upwards movements than the other two movements, one would guess this option more often than the other two. As such we consider a better baseline for the comparison with our results. One could consider a baseline for our predictions to be a prediction using our model on the price movements 1 or 2 days after the report is published. If the market indeed is efficient, then our model ought to show a considerable baseline for our predictions as the event should have already taken its effect on the stock price movements. When analyzing the results, however, we see our success rates spike up when comparing them to that of the 0 day look-back of the stock price movements. As such we adhere to our original baseline of 33.33%

### 7.3.3 Inconsistencies

Though our results are promising in regards to the goal of our research, they do differ greatly between companies. Though there can be a plethora of reasons for these occurrences, we discuss a few. Firstly, these reports are all written by different companies, only the base template of the report is consistent. As such, the possibility exists that one companies word usage by the organ responsible for their reports is more consistent than that of a different company. This could explain why it is easier to predict stock price movements through text analysis for say Microsoft than Tesla.

Secondly, the time window, though shrunk down to an interval of five business days, does allow for inconsistencies. This allows the event discussed in the report to take place on any of these days. If a company publishes these three days after the event takes place, the forest of these would score higher on the third day look-back. Generally, one could assume that the highest success rates are achieved when the event takes place or when the information regarding event is first made public.

Thirdly, the events that take place or the information released as a result of these events can have a longer effect on the stock price movements than just a day. This could explain the gradual decrease in success rates as we approach the day of publishing.

## 8 Conclusion

In determining the impact of the analysis of news on stock price prediction, we sought to predict stock movements more accurately on a consistent basis than basic speculation or guessing based on news' text analysis. We analyzed 8-K reports due to their coverage of only major events within companies, their content discussing events taking place within a period of four business days and their accessibility. Through the analysis of these reports using an automated text analysis approach with random forests to overcome over-fitting, we got our results.

The results do differ substantially between the analyzed stocks. We can explain these differences occurring between stocks through their consistencies in word usage and publication of the report with regard to the event discussed taking place or the information becoming public. The difference between analysis of the different look-backs can again be designated to the event taking place with respect to the event occurring, however, one can also assume we owe this to the event taking place having a longer effect on the stock price movements than just the day it takes place on.

The highest average success rate for Tesla, Microsoft, EA and Amazon over one hundred runs, were in order: 73.09%, 100.00%, 88.64%, 84.95%. Though, if we wish to be consistent and take just one methodology to predict the price movements, we choose to look-back four days and compensate for market movements. This resulted in average success rates for Tesla, Microsoft, EA and Amazon over one hundred runs, in order: 65.63%, 100.0%, 80.89%, 82.55%. Following our definition of 'consistently more accurate', where we wish our average success rate to be greater than that of our baseline of random guessing, we found that it is very well possible to predict stock price movement significantly better than our considered baseline of random guessing of 33.33% over a hundred runs, using only the text of 8-K reports as a predictor. However, if we wish to acquire the best result, we ought to analyze each stock separately.

## 9 Discussion

Having found our approach to text analysis to be a significant improvement to random guessing using company specific 8-K reports, one may consider a more global approach. Instead of using only the texts of 8-K reports discussing company specific events, one could analyze a whole plethora of companies' 8-K reports with their corresponding scores for the construction of a random forest that then can be used globally on any stock. This method could make the use of 8-K reports easier and save time, as now with the model construction, we need to change up the program each time we wish to analyze a different companies' 8-K reports.

Another aspect one could elaborate on further, is the extension of our current random forest construction. In our approach these decision trees in the random forest only make use of words and their occurrence rate in the reports. However, this method can be extended to also consider different aspects of stocks such as: volatility, trading volume, earnings per share, etc. If done correctly, one can presume a model trained on these aspects with the word usages in the reports, would achieve better predictions for the stock price movements.

In our model we only considered the use of 8-K reports, these reports have to be published within four business days prior to one of the listed events taking place. If we would be able to find a source that can give us a report as soon as the event takes place or first becomes public, then we could propose an actual trading strategy based on our model. We would no longer lag behind reported events and as such could actually consider the real time prediction of stock price movements.

Finally, in our results we came to the conclusion that the success rates for the predictions of stock price movements on the 2 days prior to and after the 4 business day interval, showed success rates much greater than expected. Analysis of these results could definitely yield interesting results as it goes against our intuition. For example, if our predictions prior to the four business days are greater market wide, then it could be a possibility that the information of the event discussed in the 8-K report does actually go public earlier than is permitted by the U.S. Securities and Exchange Commission.

## References

- [1] *On the importance of text analysis for stock price prediction*. Lee, H. et al.; 2014.
- [2] *Sentiment polarity identification in financial news: a cohesion-based approach*. Devitt, A. & Ahmad, K.; Proceedings of the 45th annual meeting of the association of computational linguistics (pp. 984-991); 2007.
- [3] *Measuring and testing the impact of news on volatility*. Engle, R.F. & Ng, V.K.; The journal of finance, 48(5), 1749-1778; 1993.
- [4] *Mining financial news for major events and their impacts on the market*. Mahajan, A., Dey, L., & Haque, S. M.; In 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (Vol. 1, pp. 423-426). IEEE.; 2008.
- [5] *News which Moves the Market: Assessing the Impact of Published Financial News on the Stock Market*. Soon, Y.C.; 2010.
- [6] Yuz, T., & Yuz, T. (2018, April 28). *A Sentiment Analysis Approach to Predicting Stock Returns*. Retrieved from <https://medium.com/@tomyuz/a-sentiment-analysis-approach-to-predicting-stock-returns-d5ca8b75a42>.
- [7] Scikit-learn developers 1.10. *Decision Trees*. Retrieved from <https://scikit-learn.org/stable/modules/tree.html>
- [8] Vanderplas, J. T. (2017). *Python data science handbook: Essential tools for working with data*. Beijing: OReilly.