University of Ljubljana
FACULTY OF ECONOMICS

Kardeljeva ploščad 17
SI - 1000 Ljubljana, Slovenia
Tel.: +386 1 5892 400
info@ef.uni-lj.si
www.ef.uni-lj.si

PROJECT PAPER

# PREDICTING STOCK PRICE CHANGES OF TESLA BY ANALYZING TWEETS OF ELON MUSK

**Students:**

Nicolas Bender (19221567)

Lukas Huber (19221284),

Fabian Längle (19220872),

**Professors: Prof. Dr. Jurij Jaklič, Prof. Dr. Aleš Popovič, Prof. Dr. Uroš Godnov**

**Course: Big Data Management and Technologies**

**Study year: 2019/20**

Ljubljana, December 3rd 2019

# TABLE OF CONTENTS

## LIST OF FIGURES



## LIST OF APPENDICES

# INTROCUDCTION

In the context of investment banking, decision makers often face the situation of buying stocks that will bring value to the acquiring company in the long run. Even though strong long-term factors such as seasonality or quarterly statements cannot be influenced, investors can react to short-term changes and adapt the timing of an investment transaction.

The purpose of this project work is therefore to support the timing of an investment decision process. This will be done by comparing the influence of the social media appearance of persons in leading management positions with the stock changes of the associated company, as one can assume that they have the potential to affect the stock performance. In this case the focus will be on the Twitter appearance of Elon Musk and his potential influence towards the stock price development of the respective company (Tesla Inc.).

This specific example is chosen, because it can be assumed that the personal communication of Elon Musk somehow influences the stock performance of the company Tesla Inc. as charges by the U.S. Securities and Exchange Commission (SEC) from September 2018 concerning a tweet of Musk about taking Tesla private at a certain price per share indicates (U.S. Securities and Exchange Commission, 2018).

# 1 EXPLANATION OF THE BUSINESS PROBLEM

For investment managers it is a key skill to somehow anticipate the growth or shrinkage of given stocks. Amongst others, based on this knowledge, far-reaching decisions are made. To anticipate the development of a specific stock, multiple indicators need to be considered. Ever since the increasing popularity of social media plattforms in general, social media appearances of companies or persons can be discussed or even considered to be one of the indicators for stock changes. The reasons for this are mainly platforms like Twitter where nowadays the majority of official communication of enterprises, organizations or even governments with the general public take place. This can be an opportunity for investment bankers by taking advantage of the additional available information in form of announcements of powerful accounts or simply real-time reactions to events. For example, it can be assumed, that the distributed information of an account indicates if the respective company is doing well in terms of financial performance. Since tweets of the right persons can have a very far reach, there is the possibility that this information will influence the development of stock prices. In return, this can be an opportunity for investment bankers to better anticipate respective stock price changes and therefore to gain advantages for timing a potential investment. The underlying question is whether the tweeting behaviours of far reaching accounts actually have a significant impact on the respective stock prices and if it is therefore possible to predict the stock performance (in the form of a classification whether it will develop positively, neutrally or negatively) based on features connected to tweets.

# 2 PROCESS

In the following, the process of data extraction, data cleansing, data preparation and finally model application will be outlined.

## 2.1 Data extraction

Data was extracted from two sources. On the one hand tweets of Elon Musk's Twitter account (@elonmusk) were conducted using RapidMiner by applying the main operator "Get Twitter User Statuses" and saving it to an Excel file. So the first data input counted initially around 3.200 tweets starting from 01/12/2018 and reaching to 16/11/2019, as this is the limit set by the Twitter API (dating one year back in the past). Despite that, data was retrieved on the Tesla stock (TSLA) as well as on the stock index performance where the Tesla stock is listed (NASDAQ) dating on the same time period as the tweets resulting in more than 240 results where each result equals to an operating day on which stocks were traded. This data was also initially saved as an Excel sheet conducted from the official webpage of the NASDAQ stock exchange.

## 2.2 Data cleansing

After retrieving the data, some steps of data cleansing and preprocessing had to be applied.

### 2.2.1 Cleansing and preprocessing of stock data

Before applying statistical methods, the datasets had to be prepared for further use, which included cleansing and preprocessing. Hence, the first task was to simplify the data and remove all unnecessary information. In addition, to address the topic of a classification problem and therefore being able to apply classification methods, the following step included the categorization of the stock market performance of Tesla. Since the sentiment analysis of the tweets categorizes the polarity into the categories positive, neutral and negative, the same scheme will be applied to the stock market data. By subtracting the closing price of the stock from the opening price, the result is the absolute intraday change of the share, which will be saved as *T_Diff* variable. By dividing the *T_Diff* by the opening price, the percentage of said daily change will be retrieved and saved in the column *T_Diff_per*. To transform this number into classes, a structured approach is necessary.

The mean of the *T_Diff_per* is with 0.09 just slightly above 0. For further processing, it will be assumed that the mean actually is 0. This may not be true, but it simplifies the splitting process. The standard deviation of the variable is 2.33, thus signifying that within the range of -2.33 and +2.33, 68% of the changes are included. Assuming, that every change within one standard deviation is neutral, could lead to problems, because there would be an unbalanced dataset with almost 70% of the outcomes being neutral. Instead, the decision was to decrease the neutral range to ¾ of one sd. This measure allows to still consider significant statistical background information, but in the meantime increasing the variety of different outcomes. Therefore, positive changes above 1.75% will be classified as Positive, changes between +1.75% and -1.75% will be classified as Neutral and changes below -1.75 will be classified as Negative. This results in a distribution with 49 days of a positive change, 149 of a neutral change and 44 of a negative change. Lastly, by subtracting the daily high from the daily low the intraday fluctuation can be identified, which will be saved in the column *T_Range*.

After preparing the necessary classes, the unnecessary columns can be deleted, so that only the variables *Date, T_Range, T_Diff, T_Diff_per* and *T_Change* remain in the dataset.

To not distort the dataset, the same process was conducted for the NASDAQ dataset, that will deliever further input variables to help to improve the model accuracy. In the end there are 8 positive, 225 neutral and 9 negative changes. After eliminating the unnecessary columns, the final table contains the columns: *Date, N_Range, N_Diff, N_Diff_per* and *N_Change*. Both tables were merged on the Date column. Finally, with the function “pd.to_datetime()” the Date column was transformed to convert the string Date into a dateformat, which was necessary to enable further calculation. The result was one table with all the necessary stock market data and in total nine different columns and 242 stock days.

### 2.2.2 Cleansing and preprocessing of tweets

For an effective analysis of the extracted Twitter data, several steps of preprocessing had to be applied. A first step of cleaning the raw Twitter data was to drop all answers of Elon Musk towards tweets of other, in this case irrelevant, users. Since the account to whose posts Elon Musk answers most often to is his own, only his public tweets and tweets addressed to himself were considered. This reduced the dataset to the most relevant tweets for the further application. The next step was to select the relevant features by discarding information like e.g. *geo-location*. Thus, the features *Created-At, Retweet-Count* and *Text* remained. From here onwards the preprocessing was mainly focusing on the *Text* feature. Non-value adding parts of the text needed to be removed which includes referenced usernames, URLs, retweet indicators and finally any kind of symbols. Finally, the raw and cleaned text of every tweet was available.

## 2.3 Final data preparation

Besides preprocessing the stock and twitter datasets, further preparation was necessary. Since tweets and stock markets are in an action-reaction relationship, it has to be assumed that the stock change will happen after the tweet was published and not at the same time. This means, that there is some delay between the tweet publishment and the actual change. Connected to this observation, the main assumption had to be made, which time difference should be considered between a tweet being sent and the supposed effect on the stock performance will be noticeable. For this delay first the working hours of the NASDAQ stock exchange (between 09:30 and 16:00 on working days) had to be considered. This led to the assumption that all tweets sent until the middle of the day regarding stock trades (13:00) can be considered to influence the same day. All tweets after 13:00 are then consequently assigned to the next trade day, which can also be multiple days later taking also weekends and public holidays into account. This then leads to the result, that e.g. all tweets between Friday 13:00 and Sunday will consequently be compared to stock performance on the following Monday, so often multiple tweets are assigned to the same trade day. This also becomes clear in the number of rows of the cleaned datasets, where around 1080 rows are included within the cleaned tweets data but only around 240 rows within the stock performance data. So, on average more than 4 tweets must be counted towards one trade day. The final dataset then contains 1080 rows, where tweets and stock market data are merged according to the *Date* column with the rules applied, that are described above.

Very important to mention is also, that the differences between time zones and time formats of the tweets and the stock market data were considered before merging them together. Significant differences between the timestamps of the tweets and the stock data were noticeable, since they were recorded in two different time zones within the United States and also in different formats, they were aligned before merging the datasets.

As a part of data preprocessing some feature engineering in the form of the following two new features was performed:

Once the Twitter data was merged with the stock data, the number of tweets would vary a lot in a range of 1 to 30 tweets per day. Given this fact, a new column was created and the count of tweets (*tweet_count*) as a new feature to display the number of tweets per day was added. This was realized programmatically in Python.

Based on the raw text of the extracted tweets also the number of letters (*num_letters*) was counted per tweet and added as a new feature. Based on this number the tweets were even filtered to only include tweets longer than nine letters. This increases the interpretability of the tweets fed into the sentiment analysis.

In addition to the previous data preparation steps, a sentiment analysis was undertaken using the AYLIEN Text Analysis extension in RapidMiner with the operator "Analyze Sentiment" that was applied to the extracted and preprocessed 1080 tweets. As an output, *polarity* and *subjectivity* were generated for the content of all related tweets.

In total, the preprocessed dataset contains 15 features and 1077 rows. Four features each describe the range, the difference per day, the difference per day in percent and the change in polynomial values of the Tesla stock (*T_Range, T_Diff, T_Diff_per, T_Change*) or in a similar way the whole NASDAQ index performance (*N_Range, N_Diff, N_Diff_per, N_Change*). On top of that, the *Date* and the features related to the tweets such as the number of letters per tweet (*num_letters*), the amount of tweets per day (*tweet_count*), the amount of retweets per tweet (*retweet_count*) and of course the variables related to the analyzed sentiment (*polarity* and *subjectivity*) as well as the original text of the tweets (*text*) are included.

## 2.4 Application of methods

As it cannot be presumed beforehand, which method is most suitable to apply to this data, three different exemplary single methods enabling classification of a polynomial target variable (Random Forest, k-NN and Deep Learning) are used to predict changes in Tesla's stock performance (*T_Change*) that can be either positive, neutral or negative (classification problem) and one method where all three single methods are included in a voting model as an approach to improve overall accuracy. Results are generated using RapidMiner and compared regarding their performance in order to be able to select the most suitable model afterwards.

Generally speaking, for all of the following methods, the data first was retrieved in RapidMiner. After that all potentially relevant attributes were selected (e.g. excluding *Text* for the prediction and other variables related to the Tesla stock except *T_Change*), the "Set Role" operator was applied to set the polynomial target variable *T_Change* as "label" and *Date* as "id". After that the whole dataset is split using the "Multiply" operator and the "Split Validation" operator with a 70% stratified sampling split using 70% of the data for training and the other 30% for testing.

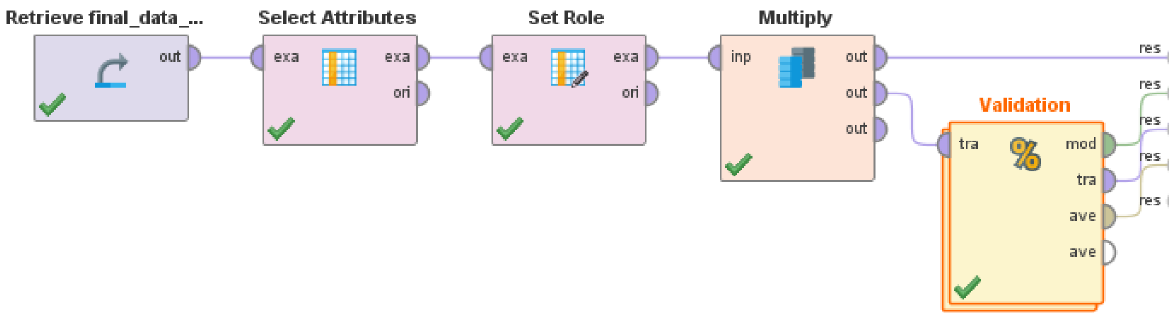


*Figure 1 General process for model application*

### 2.4.1 Random Forest

The first single model Random Forest was chosen, as it is a very common and also powerful method, using default parameters of 100 trees and a maximum depth of 10. It predicts the target variable through classification and is usually a very robust method. As the results depict, a rather high accuracy can be achieved with almost 80% (*figure 4*). Although the accuracy is quite high, the reason for misclassification is that many cases are falsely predicted as neutral although they should be either positive or negative. When having a detailed look at some of the decision trees within the Random Forest model, one can say, that most of the trees almost exclusively rely on NASDAQ variables to predict the performance of the Tesla share. The following correlation matrix (*figure 2*) also shows that those variables are strongly correlated, e.g. *N_Range*, *N_Diff* or *N_Diff_per* with *T_Range*, *T_Diff* etc. So, as it was not part of the approach to research interdependencies between one single stock and the index it is listed in and apparently also correlated with, the NASDAQ related variables needed to be excluded in the adapted model for Random Forest to solely focus on possible effects of tweets on the performance of the Tesla stock.

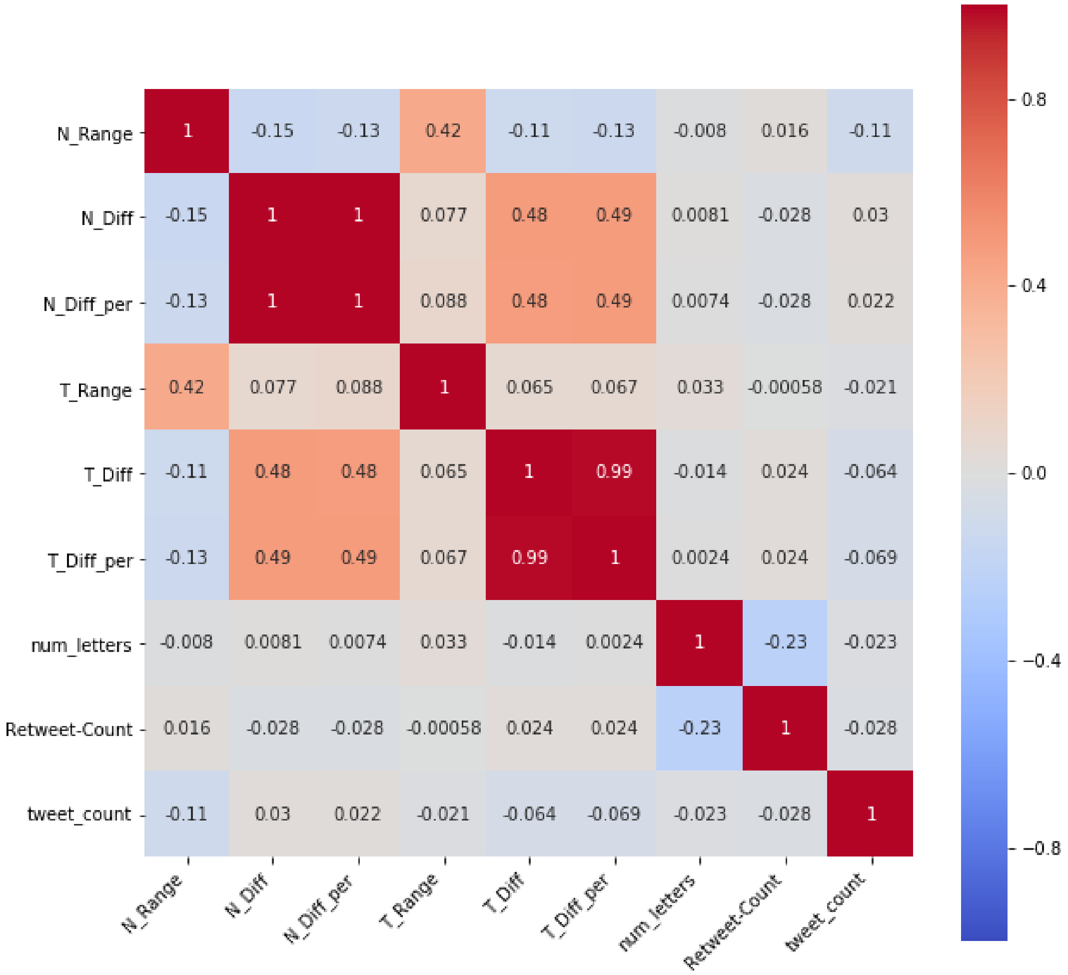


*Figure 2 Correlation matrix of features within final dataset*

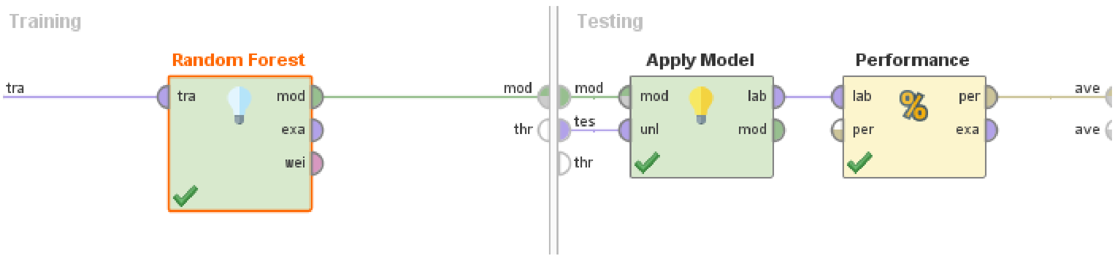


*Figure 3 Random Forest detail view*

## PerformanceVector

```
PerformanceVector:
accuracy: 79.26%
ConfusionMatrix:
True:   Positive        Neutral Negative
Positive:       32      0       0
Neutral:        37      194     29
Negative:       0       1       30
kappa: 0.564
ConfusionMatrix:
True:   Positive        Neutral Negative
Positive:       32      0       0
Neutral:        37      194     29
Negative:       0       1       30
```

*Figure 4 Random Forest performance output including NASDAQ features*

With reduced predictor variables (excluding the NASDAQ features), a lower accuracy has to be expected, because only five predictors (number of tweets per day *tweet_count*, number of letters per tweet *num_letters*, amount of retweets *retweet_count* and sentiment *polarity* and *subjectivity*) are used instead of eight or nine features beforehand. Results show an actual accuracy of around 63% (*figure 5)* which seems to still be acceptable in this case. Now also factors like the number of tweets, retweets and the number of letters per tweet are considered repeatedly in the trees being part of Random Forest. The only variables that do not really seem to have an impact are those related to the sentiment analysis (*polarity* and *subjectivity*).

## PerformanceVector

```
PerformanceVector:
accuracy: 63.16%
ConfusionMatrix:
True:   Positive        Neutral Negative
Positive:       1       0       0
Neutral:        68      195     51
Negative:       0       0       8
kappa: 0.097
ConfusionMatrix:
True:   Positive        Neutral Negative
Positive:       1       0       0
Neutral:        68      195     51
Negative:       0       0       8
```

*Figure 5 Random Forest performance output excluding NASDAQ features*

### 2.4.2 K-NN

k-NN is the next selected method to apply to the data where the output is classified based on the k closest data points (nearest neighbors) in terms of euclidean distance, e.g. the most similar datapoints. Through classification via k-NN with k=29, also excluding NASDAQ variables, an accuracy of around 60% can be achieved (*figure 7*). The main problem here also seems to be misclassification of true positive or true negative values as neutral values, since all of the results are generally classified as neutral and none as positive or negative.

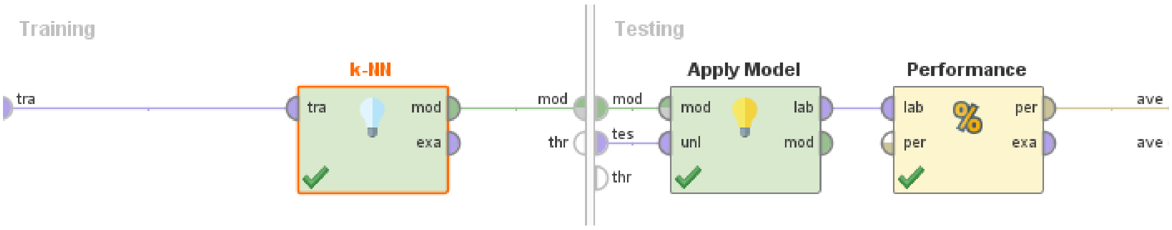


*Figure 6 k-NN detail view*

## PerformanceVector

```
PerformanceVector:
accuracy: 60.37%
ConfusionMatrix:
True:    Positive         Neutral  Negative
Positive:        0        0        0
Neutral:         69       195      59
Negative:        0        0        0
kappa: 0.000
ConfusionMatrix:
True:    Positive         Neutral  Negative
Positive:        0        0        0
Neutral:         69       195      59
Negative:        0        0        0
```

*Figure 7 k-NN performance output excluding NASDAQ features*

### 2.4.3 Deep Learning

The third method is considered to be Deep Learning, based on a multi-layer feed-forward artificial neural network. Here, the default settings to configure the parameters are kept and the Deep Learning operator is applied in the same way as k-NN before (also excluding NASDAQ variables), resulting in an accuracy of around 60% (*figure 9*) with the similar problem of many misclassifications as neutral, that should be either classified as positive or negative.

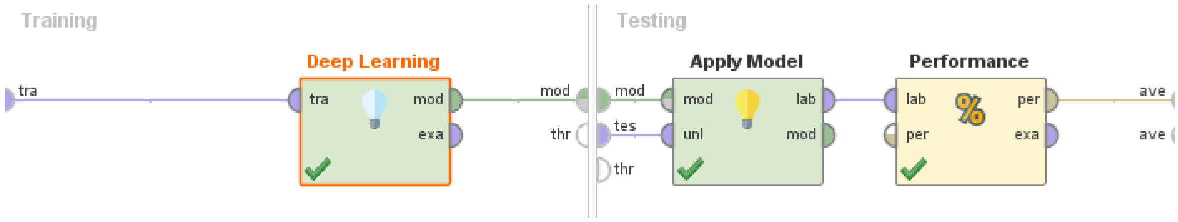


*Figure 8 Deep Learning detail view*

## PerformanceVector

```
PerformanceVector:
accuracy: 60.06%
ConfusionMatrix:
True:    Positive         Neutral Negative
Positive:        0        1       0
Neutral:         69       194     59
Negative:        0        0       0
kappa: -0.005
ConfusionMatrix:
True:    Positive         Neutral Negative
Positive:        0        1       0
Neutral:         69       194     59
Negative:        0        0       0
```

*Figure 9 Deep Learning performance output excluding NASDAQ features*

### 2.4.4 Vote model

As a last option a voting model can be tested, to check whether performance increases if not one single method is being applied, but multiple methods in a voting structure using a majority vote of included methods in case of classification. To realize this, the "Cross Validation" operator is used. Within this operator "Vote" is applied in the training part containing all three methods from before with the same individual settings (k-NN, Deep Learning and Random Forest) generating a majority vote to provide a prediction.

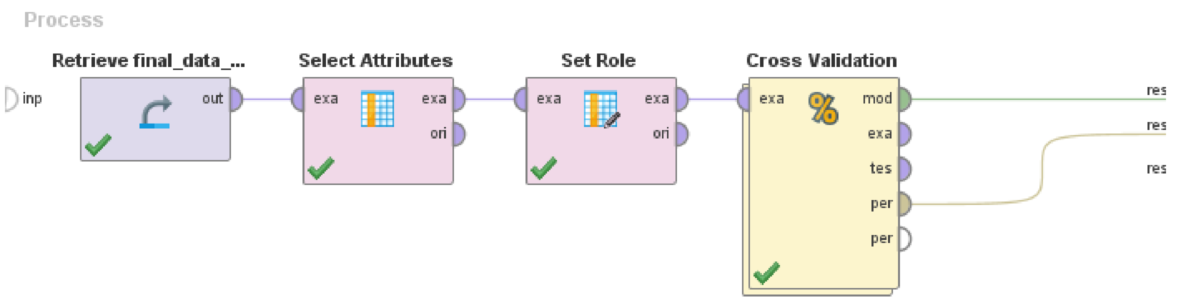


*Figure 10 Vote model process overview*

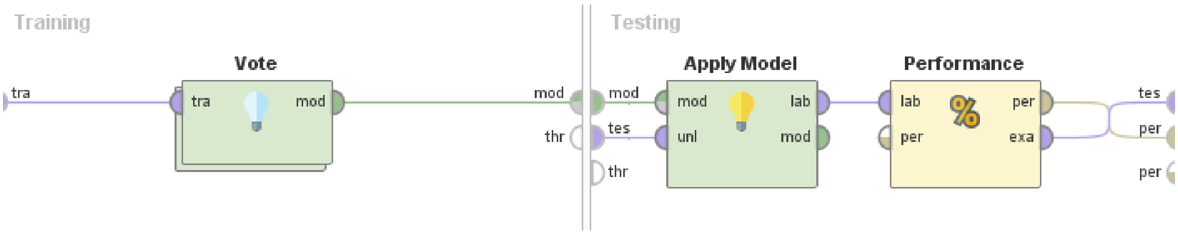


*Figure 11 Vote model detail view*

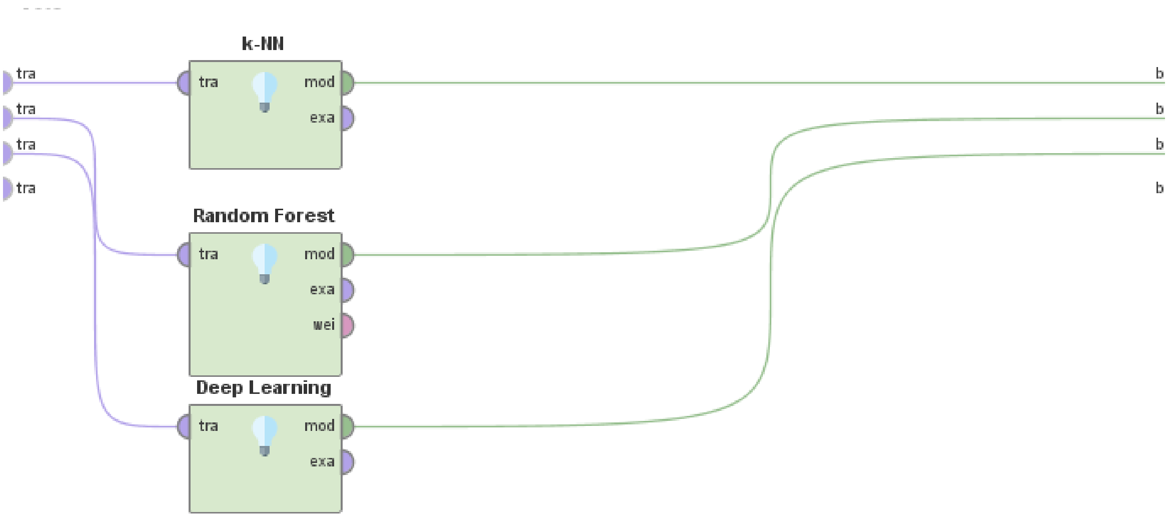


*Figure 12 Vote model applied methods*

## PerformanceVector

```
PerformanceVector:
accuracy: 60.44% +/- 1.99% (micro average: 60.45%)
ConfusionMatrix:
True:    Positive        Neutral Negative
Positive:        3       9       10
Neutral:         226     641     179
Negative:        1       1       7
kappa: 0.027 +/- 0.066 (micro average: 0.028)
ConfusionMatrix:
True:    Positive        Neutral Negative
Positive:        3       9       10
Neutral:         226     641     179
Negative:        1       1       7
```

*Figure 13 Vote model performance output excluding NASDAQ features*

As the performance output shows, the accuracy is still around 60% (*figure 13*) which does not really deliver a significant improvement. Here again, generally a default result value of neutral is being predicted with many misclassifications that should either be positive or negative instead. The last option applied that might improve the model performance was to give penalties and specifying weights of misclassification costs in a so called “cost matrix” for every class prediction using the operator “MetaCost”. Here the previously misclassified classes can be penalized in different ways with different costs (e.g. 2.0) in contrast to the correctly predicted classes, which get a cost of just 1.0. Nevertheless, this also enables only minor improvements resulting in a total accuracy of around 63.5% (*figure 14*) which does not solve the major problem of default prediction of values as “Neutral”.

**PerformanceVector**

```
PerformanceVector:
accuracy: 63.51% +/- 1.48% (micro average: 63.51%)
ConfusionMatrix:
True:   Positive        Neutral Negative
Positive:       1       0       0
Neutral:        228     651     164
Negative:       1       0       32
kappa: 0.106 +/- 0.054 (micro average: 0.107)
ConfusionMatrix:
True:   Positive        Neutral Negative
Positive:       1       0       0
Neutral:        228     651     164
Negative:       1       0       32
```

*Figure 14 Vote model performance using the MetaCost operator excluding NASDAQ features*

# 3 DISCUSSION

In this chapter findings of the data analysis, limitations and the potential future implications are being discussed.

## 3.1 Analysis and findings

The stated results suggest that each model separately as well as the combination of the three operators each have an accuracy of around 60%. Therefore, six out of ten times the change of the Tesla stock was classified correctly in dependence of the tweeting behaviour of Elon Musk. In theory 60% might sound very bad, but in real life implication this seems to be an acceptable result for such a complex topic, that is stock market performance. Generally speaking, this signifies that in two thirds of the occasions, the stock change was classified correctly, which is more than double the probability of a random guess out of the three categories.

Although this seems convincing at first and this outcome also may be desirable, not only the accuracy of the models have to be interpreted but also the content and implications of the results themselves. As it is visible, most of the neutral outcomes have been detected correctly, but almost none of the positive or negative changes. This suggests, that the models oversimplify the underlying data and classify almost every possible result as neutral. Since the majority of the stock price changes is stated as neutral, it can be assumed that by always guessing a neutral outcome, the correct classification is made most of the time. The model therefore does not get 60% accuracy because of its good training and good alignment of variables, but because of the structure of the underlying dataset. This problem of underfitting leads to the condition that the results can´t be interpreted correctly and the models contain very high biases. They furthermore have very high errors and generally very low complexities due to probably a small databasis. This condition suggests that no significant connection between the stock market data and the tweets of Elon Musk can be found within the underlying dataset. Since this problem occurred with every single method, it can be assumed that it was not a specific inaccuracy of an operator, but a general problem of the underlying dataset. Although it may be possible that the tweets can influence stock market performance in general, the results of the study do not support this statement, since no significant connection could be found.

## 3.2 Limitations

To interpret the results comprehensively, not only the results and models should be considered, but also limitations of the underlying data and processes. These could influence the mechanics of the study and lead to distorted results. In general, three types of limitations can be found by analyzing the study design: Limitations due to technological restrictions, limitations due to the procedure and limitations due to the data.

By looking at the technological restrictions, is has to be emphasized that the Twitter API allows users only to extract the number of tweets published in the last twelve months. Therefore, the data that was available ranged just from December 2018 until November 2019. This precondition limits the size of the underlying dataset and with a longer time period, it could have been possible to include more training data and maybe obtain a more profound model. Not only the API was a problem that limited the database but also the use of Twitter as a social platform itself. Twitter offers its users not only to express their opinions via text but also via images and videos. In the study the focus was solely set on the analysis of the respective text statements. The content of the other datatypes was ignored to limit the complexity of the analysis. With this measure, important information could have been disregarded, which suggests that the used data contained less information than what could have been available. One further limitation could have been caused by the used tools. Although RapidMiner and specifically the AYLIEN extension are in general valid methods to perform a profound data analysis, it is very hard to determine their respective performance and especially the accuracy of the AYLIEN sentiment analysis. One study (slant.co, n.d.)

claims an accuracy of roughly 80%, which is generally a good result, but it can still lead to some misclassified values and therefore distort the dataset. Since the accuracy of the used tools cannot be confirmed with absolute certainty, this potential problem has to kept in mind.

Next to the prerequisites, that are set by the used technologies, it may also be possible that the process itself contains flaws. For example, the recalculation of the tweet days may have derived some issues, that could lead to mistakes. Although it is to assume that there is some delay between tweet time and stock impact, just passing the effect to the next day, may be an oversimplification. Especially when it is known to big shareholders or hedge funds managers that Elon Musk for example is the first person to tweet about the company's quarterly results, they may react very fast to what is stated in the tweet. One big problem connected with the recalculation of the next stock operating day, is that the next operation day must not necessarily be the actual next day, but can also be three or four days later, due to weekends or public holidays. In such a long period of time, the connection between a tweet and the stock change would probably not be noticeable anymore. This problem occurs every week, when the tweets of Friday afternoon are assigned to next Monday. The span of three days is very large in business terms and could again distort the assigned influence. Because of this gap, it is possible that almost one fifth of the recalculated dates underly such distorted connections. Furthermore, to simplify the process, an important peculiarity of stocks has been ignored. The prices of a stock can not only vary during the day but can also change after trading hours. Therefore, a tweet can also affect stock prices after the closing, which was ignored too in order to minimize complexity of the analysis.

Lastly, it may also be possible that the underlying dataset contained shortcomings that distorted the results. A data basis with just 242 days of stock operation might be too small, to explain such a complex behaviour. Therefore, it can be assumed that the dataset contained not enough observations to perform a valid model training. With a train-test-split ratio of 70/30, this means that only 175 days e.g. observations, have been taken into consideration when training the models, which may have been not enough to achieve a good performance. What is also problematic is, that there is not a linear connection of one tweet per day but a very big variety of days with multiple tweets and other days with none. The issue is not only the unbalanced distribution itself, but also how to handle the problem, when more than one tweet was posted on one day. During the preparation process it was possible that multiple tweets were designated to influence one day. This occurs for example, when tweets of the previous afternoon were assigned to the next day, along with the tweets of the respective day. *With the tweet_count* a measure was taken to solve this issue, but a problem arises when tweets with different sentiments are expected to influence the same day. This again oversimplifies the complex relationship between the tweets and the stock data and distorts the outcomes. Furthermore, one limitation is the person Elon Musk himself. He is not only CEO of Tesla but also of other companies such as The Boring Company or SpaceX. It is then natural, that not only tweets about one company were published but also about the other ones. To determine if this assumption is true, word frequency and association rules have been applied to the dataset (*figures 15 and 16*).

| Size | Support | Item 1 |
|---|---|---|
| 1 | 0.203 | tesla |
| 1 | 0.113 | model |
| 1 | 0.059 | car |
| 1 | 0.058 | falcon |
| 1 | 0.049 | launch |
| 1 | 0.040 | dragon |
| 1 | 0.039 | starship |
| 1 | 0.037 | year |
| 1 | 0.034 | cars |
| 1 | 0.032 | spacex |

*Figure 15 Word frequency within analyzed tweet sample*

| No. | Premises | Conclusion | Support | Confide... ↓ |
|---|---|---|---|---|
| 17 | selfdriving | full | 0.017 | 0.947 |
| 16 | heavy | falcon | 0.014 | 0.789 |
| 15 | online | order | 0.011 | 0.750 |
| 14 | crew | dragon | 0.014 | 0.652 |
| 12 | selfdriving | tesla | 0.011 | 0.632 |
| 13 | model, car | tesla | 0.011 | 0.632 |
| 11 | autopilot | tesla | 0.014 | 0.625 |
| 10 | electric | tesla | 0.011 | 0.600 |

*Figure 16 Association rules within analyzed tweet sample*

There it is visible, that Tesla may be the main topic of the tweets, but also the company SpaceX for example is very prominent with its rocket launches. The terms “falcon” and “launch” describe the rocket launches of SpaceX and are ranked fourth and fifth in terms of frequency. SpaceX itself is also inside the top 10 of the most used terms. The problem was to assume, that every tweet published by Elon Musk would only contain information somehow related to Tesla, which was not the case. Finally, it also should be stated, that topics as complex as stock market changes are influenced by a huge variety of factors and not just solely by one small set of variables. It can be assumed that the tweets may be used to explain some minor movements or specific daily peaks or lows, but it is not enough to explain the general movement of the stock price. More information has to be taken into consideration to determine the influence of every single input variable.

All in all, it has to be stated, that the interpretation of the results is dependent on the acknowledgment that limitation have occurred during the process and that the data is on some way biased. By considering these preconditions, it is possible to interpret the outcomes comprehensively and make suggestions for an improved analysis.

## 3.3 Outlook

By the analysis of the results and the limitations, recommendations can be given for further studies. By avoiding the stated mistakes and possible distortions, a study could be executed with more profound results. To do so, it is firstly important to obtain a data basis that is large enough for a big data analytics approach. Although data for the stock changes can be retrieved easily, the tweet data also has to be increased. If the tweet history is big enough, a train/test ratio can then be applied that uses enough data to ensure a profound model (e.g. over a time period of two or three years).

Furthermore, the analysis would be more complete, if not only texts but also other types of data could be included. With the inclusion of images for example, posted statistics or company figures value could be added to the whole process. All of these measures would enlarge the test data set and help to perform a better analysis without underfitting.

The recalculation of the stock market changes should also be reconsidered. Although in the study the focus was to achieve a realistic balancing of positive, neutral and negative stock price changes, it may be useful to apply more strict rules. One possibility is the usage of different values to classify the data. For example, change percentages can be used to balance the data, so that one third of the changes is classified as positive change, one third as neutral and one third as negative change. This would be the result when the ranges would be changed to be within ⅓ of the standard deviation. With this operation, there would be no longer a problem, with too many neutral values that tend to let the models underfit.

Another possibility would be increasing the categories to e.g. 5 instead of 3 classes of the target variable. This could help to draw a more diverse picture and allow more variation in the dataset.

Next to changing the preprocessing, one recommendation is also to perform the analysis with a different CEO and a different associated company. As seen by the association rules and especially the term frequency, since there is an increased difficulty imposed by a person that drives more than one business, it is best to choose a character that is responsible for just a single company, for example Tim Cook being responsible for only Apple.

By considering all the proposed changes, a more detailed view can be applied. A bigger database and more direct connections between one tweet and the accompanying stock price change could lead to a significant relationship between the tweeting behaviour of one person and the performance of the respective share, which could them support investment decisions.

# 4 CONCLUSION

A variety of different models have been trained on the extracted data and resulted in a similar score of around 60% accuracy. By taking a closer look into the outcomes of the Random Forest model, the most interpretable model, a straightforward explanation for the model performance can be found. Most of the decision trees do take tweet-related features into account (e.g. num_letters or retweet_count) and predict the stock change respectively, even though the actual sentiment of the tweet never seems to influence the prediction at all. The remarkable observation in the decision trees is that nearly all of the leaves are classified as 'Neutral'. This can be confirmed looking into the resulting confusion matrix of the Random Forest model. Almost all of the data points get classified as 'Neutral' which points out the problem of having an unbalanced distribution of data points in the available classes. Training a model on data of this shape almost certainly results in a highly underfitted model as one can see for the Random Forest and the remaining applied methods.

Given the previous findings it is difficult to specify an interpretation based on the achieved results. All tested models are based on the assumption that tweets from Elon Musk are the only indicators for changes in the stock price. Obviously, this is far from reality but on the other hand considering all possible influencing factors was neither feasible nor desired in the context of this seminar work. In the end a statistically significant correlation between changes in stock prices for Tesla Inc. and the tweeting behaviour of Elon Musk cannot be proven by our approach. Neither can it be disapproved. Therefore, only the possibility of a small correlation between tweets and stock prices can be concluded. For higher evidence more thorough and diverse data needs to be collected and analysed.

Furthermore, features derived from the stock price changes of NASDAQ should not be included into a prediction model for stock price changes of Tesla Inc. This mainly derives from the fact that NASDAQ is a stock index composed of 3000 individual stocks, with Tesla Inc. being one of them, which results in the NASDAQ reflecting the overall mood which is currently present in the American stock market. Consequently, it is very likely that the Tesla Inc. stock approximately follows the development of NASDAQ. As observed in the initially trained Random Forest the features derived from the NASDAQ stock index would dominate the model and therefore need to be excluded.

The findings suggest that analysts can not rely on a generalized model of twitter data as a single source of prediction method to help them place their stock orders. In this study it was not possible to conclude a clear relationship between the tweets of Elon Musk and the Tesla stock performance. Nevertheless, it may be possible, that a comprehensive analysis of selected tweets, among other indicators can contribute to an improved order placement. To conduct if this assumption is the true, further studies with the proposed improvements have to be carried out.

# 5 REFERENCES

Slant.co (n.d.). Aylien Review. Retrieved from https://www.slant.co/options/27165/~aylien-review

U.S. Securities and Exchange Commission (2018, September 29). *Elon Musk Settles SEC Fraud Charges; Tesla Charged With and Resolves Securities Law Charge* [Press release]. Washington D.C. Retrieved from https://www.sec.gov/news/press-release/2018-226

# APPENDICES

**Appendix 1: Python Scripts for data cleansing and preparation**

- 1920_BDM_Preprocessing_stock_data: Cleaning and merging of stock market data
- 1920_BDM_Preprocessing_twitter_data: Date adjustment of tweets and correlation matrix

## Appendix 2: RapidMiner Processes

- Tweet_preprocessing: Cleaning and preprocessing of Twitter data
- Final_data_Project_RF: Process for Random Forest including NASDAQ features
- Final_data_Project_RF2: Process for Random Forest excluding NASDAQ features
- Final_data_Project_KNN: Process for k-NN excluding NASDAQ features
- Final_data_Project_DL: Process for Deep Learning excluding NASDAQ features
- Final_data_Project_VM: Process for Vote model excluding NASDAQ features
- Final_data_Project_VM2: Process for Vote model using the MetaCost operator
- Final_data_Project_AR: Process for word count and Association Rules