Analyzing Relations between Financial Indicators and the Public Debate Using Knowledge Extraction and Machine Learning Technologies

Master Thesis submitted in fulfillment

of the Degree Master of Business Administration

in Entrepreneurship, Innovation & Leadership

Submitted to Prof. DDr. Arno Scharl

Mag. DI Dr. Wolfgang Radinger-Peer

1702017

Vienna, June 18, 2019

AFFIDAVIT

I hereby affirm that this Master's Thesis represents my own written work and that I have used no sources and aids other than those indicated. All passages quoted from publications or paraphrased from these sources are properly cited and attributed.

The thesis was not submitted in the same or in a substantially similar version, not even partially, to another examination board and was not published elsewhere.

Date	Signature

ABSTRACT

The finance domain is very sensitive to messages in the public debate communicated through different channels. A short message can have a tremendous impact on the strategies employed by investors. If many investors react to a given message, it might have a major impact on an entire market. The aim of this master thesis is to examine dependencies between financial indicators and messages in the public debate by using knowledge extraction and visualization technologies.

Related work to this research area can be divided into three different approaches. The first attempts to predict stock market trends based only on market data and technical indicators like Relative Strength Index, Momentum, or Simple Moving Average. Other approaches employ sentiment analysis to extract meaning, with applications becoming ever more focused on social media, and especially Twitter, in recent years. The information gathered on social media are analyzed in order to measure public mood and to predict stock market trends. The third approach uses a combination of technical indicators and news sentiment analysis. This thesis uses the third approach and defines seven case studies covering multiple languages and sectors, including two companies from the US, the UK, and Austria, as well as Bitcoin as a global virtual currency.

The historical time series data used to calculate the technical indicators were imported from market data providers and the news sentiment collected from the webLyzard Web Intelligence platform of MODUL University Vienna's Department of New Media Technology, which enriches the data with sentiment analysis and statistical information. These two datasets combined to form the feature set which serves as input for a Machine Learning algorithm. XGBoost was selected for the Machine Learning as it is a highly acclaimed and award-winning classification algorithm. The labeling for the classification was made based on daily returns to predict a "rise" or "fall". The prototype, implemented in R-Language, showed that the accuracy of the prediction is generally improved by adding news features to the technical indicators. To evaluate the effect of the news features, three scenarios have been defined. The first scenario uses only stock quotes, the second adds the technical indicators and the third adds the news features.

It can be summarized that if there is a lively public debate, the news features influence the model and are designated as important features. For the Austrian case studies, it turned out that the low number of messages and their rather positive sentiment meant that they do not constitute an improvement over the technical indicators only model, although it should be considered that the algorithm was optimized for a US case study. The neutral sentiment of the UK news is mostly due to the dominance of social media messages, which leads to lower accuracy but to a higher resulting Area Under Curve metric (expected true positive rate, averaged over all false positive rates) value.

ACKNOWLEDGMENTS

First I want to thank MODUL University Vienna for giving me the opportunity to study in the Master of Business Administration program. In particular I would like to thank my thesis advisor Arno Scharl for offering me this interesting and challenging topic and giving me the opportunity to discover a new field of research.

Thanks also to my fellow students, with whom it was a pleasure to spend the weekends on Kahlenberg. I also want to thank Richard who tried hard to make the complexity of mathematics understandable to me. Also I want to thank Michael for helping me bringing the words in the right order.

A special thanks are due to my wife, Verena, for her continuous support and understanding, for being a counterpart in discussions, and also for motivating me to complete this work. Words cannot express how grateful I am to my kids who had several weekends without me.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Equations

List of Abbreviations

ADX	Average Directional Movement Index
AI	Artificial Intelligence
AUC	Area Under The Curve
ANN	Artificial Neural Networks
API	Application Programming Interface
APO	Absolute Price Oscillator
ATS	Automated Trading System
CAPM	Capital Asset Pricing Model
CSV	Comma Separated Values
CV	Cross-Validation
EMA	Exponential Moving Averages
EMH	Efficient Market Hypothesis
EOD	End Of Day
ETF	Exchange Traded Funds
HFT	High-Frequency Trading
KNN	k-Nearest Neighbors
KPI	Key Performance Indicators
LSE	London Stock Exchange
ML	Machine Learning
MWCC	Media Watch on Climate Change
OBV	On Balance Volume
NOAA	National Oceanic and Atmospheric Administration
RL	Reinforcement Learning
RMSE	Root Mean Squared Error
RoC	Rate of Change
ROC	Receiver Operating Characteristics
ROI	Return On Investment
RSI	Relative Strength Index
SMA	Simple Moving Average
SVM	Support Vector Machine
WYSDOM	webLyzard Stakeholder Dialog and Opinion Model
XGB	eXtreme Gradient Boosting

1. Introduction

"Forecasts may tell you a great deal about the forecaster; they tell you nothing about the future."

Warren Buffett

Financial markets have always attracted the interest of business and academia: the former to foster business ventures and make money, while the latter use this complex environment for their studies. Figures and their statistical interpretation play a dual role in financial market environments, providing on the one hand information about current and past events, and on the other hand, acting as a key driver of decisions which will shape the future. The range and quality of tools for statistical analysis have improved consistently over time, such that human decisions are today supported by countless statistical tools designed to improve both the speed and accuracy of decisions. Even more dramatic have been changes in the availability of information, which has increased tremendously in terms of speed and quantity over the last two decades. As this information has to be aggregated and structured in order to serve as support for decisions, it is no longer possible to process this data without technological support. For a better understanding of this developing field it is necessary to trace its long history and analyse the dramatic changes which have occurred within the last decades.

The year 1460 saw the foundation of the world's first bourse in Antwerp, which was the first building exclusively built for exchange trading. In this time of increasing speculation, new professions like broker came into existence. London and Amsterdam followed in the ensuing decades, with the first stock - a share of the Dutch East India Company - being traded in Amsterdam in the year 1602. 1971 was another milestone in the history of exchange trading, when the computer age started for stock exchanging in New York. On the 8th of February 1971 the NASDAQ was founded as the world's first electronic stock exchange. This new kind of trading had a huge impact on the trades themselves, as the bid-ask spread was reduced and worldwide trading became possible.

In Urstadt (2009, p. 44) explains in a MIT technical report "By the end of the day, his computers will have bought and sold about 60 million to 80 million shares". Trading has already moved from humans to machines which are able to process huge amounts of information and make decisions based on algorithms, hence the name Algorithm Trading. "The profits go to the company with the fastest hardware and the best algorithms", continues

Urstadt (2009, p. 44). Computer algorithms can place orders based on a pre-defined strategy, either more conservative or speculative. The main advantage is that these algorithms combine historical market data and real-time market data and can place orders within milliseconds. A dominant type of Algorithm Trading is High Frequency Trading (HFT), where negotiation orders are placed in a high frequency with a low latency. Gerig (2012) determined that 55% of trading volume in U.S. stock markets and 40% of European stock markets volumes are executed with HFT.

Quantitative analysis aims to understand the market and valuate financial instruments to predict behaviors and events using financial economic techniques, mathematical measurements, statistics, predictive modeling, and computing (Merton, 1973). Hence quantitative trading involves the application of automated strategies based on such analyses. It is also known as black box trading, because some systems make use of proprietary and undisclosed algorithms.

At the same time as these developments, changes have taken place affecting the speed and types of news distribution in general, which have also influenced the character of financial news. The impact of news on the stock market was already verified by Mitchell and Mulherin (1994). They analyzed news activities concerning the Dow Jones and demonstrate that they are directly related but of a weak character. Since that time the information landscape has changed, especially with the emergence of social media. Research by Bollen and Mao (2011) concerning the effect of public mood of Twitter messages also showed an impact related to behavioral economics.

1.1. Motivation

The motivation for this thesis arises from the author's profession in the information management sector, in which he has been working for more than 20 years with a focus on financial data management. Hence the opportunity to combine time series statistical analysis with new Artificial Intelligence methods was the main motivation for this thesis. As outlined by Nassirtoussi, Aghabozorgi, Wah, and Ngo (2014) the challenge is the interplay of three separate fields:

1. linguistics: to understand the nature of language
2. machine-learning: to enable computational modeling and pattern recognition
3. behavioral-economics: to establish economic sense

1.2. Research Aims and Objectives

The aim of this thesis is to investigate the research question:

> **Can the prediction of finance market time series data be improved by combining sentiment news features with technical indicators using a Machine Learning approach?**

In doing so the existing webLyzard Web Intelligence platform of MODUL University Vienna's Department of New Media Technology is used a s source for news and social media data. webLyzard continuously gathers data from news feeds and social media platforms. This news data is further enriched through the collection of financial market data information. The case studies are of stock companies and a cryptocurrency, with a focus on the markets of Austria, the United Kingdom and the United States. The United Kingdom is currently interesting example because of the BREXIT scenario which is mainly driven by the public opinion and has been discussed with much controversy. Austria is characterized by a small stock market and a financial news information culture with a low international range. The US market has a leading position in the financial sector, as most of the largest stock companies issue their shares on Wall Street and also most of the big financial news channels like Reuters and Bloomberg are also located in the US.

The following steps will be used to elaborate on the research question:

- Review literature focusing on work related to stock price prediction based on time series analysis and in combination with news and social media data.
- Analyse technical indicators for time series statistics to serve as features for a prediction model.
- Define case studies for the evaluation of the model.
- Define an approach based on Machine Learning to predict the movement of stock prices or currency.
- Implement a prototype which can be integrated in the webLyzard platform.

1.3. Structure of the Thesis

In chapter 2 a detailed literature review presents the state of the art. This begins by summarizing the evolution of stock price predictions, before explaining in detail the different technologies used in this field. Based on this theoretical background, chapter 3 describes the methods used within this thesis to address the previously defined objectives. The chosen approach is compared with other approaches published in various papers. Chapter 4 describes the implementation of the prototype and how the prediction model is realized. The results are discussed and evaluated in chapter 5. The thesis concludes with a summary in chapter 6 and suggestions for further research opportunities.

2. Literature Review

“If we knew what it was we were doing, it would not be called research, would it?”

Albert Einstein

There is a long history of trying to predict stock market trends. Many different approaches have been published, whereby the first methods relied on technical analysis of the historical stock quotes and used different statistical methods. In the last 30 years the correlation between stock market trends and the public opinion (i.e. news, social media) has been analyzed in more detail. Particularly through the emergence of social media, this field of research has gained major importance. For news, the micro-blogging service Twitter, has been used to measure the public mood and investigate correlation to various topics. One example is to predict the winners of the popular Oscar film academy awards (Zauzmer, 2018), where Zauzmer was able to predict 20 out of 21 correctly. Hence it was obvious that the sensitive financial sector would also make use of social media data to improve the prediction of stock prices or rather market trends.

2.1. History of Stock Market Predictions

Fama (1965) published in “The Behavior of Stock-Market Prices” the Efficient Market Hypothesis (EMH) stating that the price of an asset is informationally efficient, which means that the price completely reflects all information available. Based on this hypothesis, a prediction of prices is not possible because all the information is already included and all agents have the same information. This theory was later amended by Fama (1970), whereby he defined three levels of efficiency: strong, semi-strong and weak. This Random Walk Theory was the subject of lively discussions throughout the 1960s and 1970s. It is based on the signal theory, where the price of a stock is modeled as the sum of last price X_{t-1} and a drift constant μ, and a random disturbance ϵ_t (Cootner, 1964). On the other hand, Behavioral Economics analyses the effects of social, cognitive and emotional perspectives to evaluate economic decisions. An effect of public information on stock market prices has been outlined in several publications. Cutler, Poterba, and Summers (1988) found that one third of the variance of the return of a stock can be ascribed to public information. Years later Mitchell and Mulherin

(1994) studied the relation between news stories of the Dow Jones and the trading volume sum of the stock returns. They conclude a direct, robust relation based on a day-of-the-week pattern.

2.2. Analysis Techniques

Categorically, two types of analysis can be differentiated. *Fundamental analysis* uses a company's financial conditions and operations data, as well as the macroeconomic situation, to predict the development of the company. *Technical analysis*, in comparison, relies on historical price analysis to make predictions, in accordance with the belief that history tends to repeat itself. Frequently used technical indicators in the literature are collected in the next section.

2.2.1. Fundamental Analysis

Fundamental analysis investigates the various economic sectors related to a given company, which could also include the sector of operation like financial, automotive or health care (macroeconomic factors). The analysts collect both quantitative (e.g. earnings) and qualitative (e.g. competitors) data with the aim of calculating a representative figure which indicates if a company is overvalued or undervalued. According to Nassirtoussi et al., 2014 fundamental data can have the following sources:

1. financial data of a company like data in its balance sheet or financial data about a currency in the FOREX market,
2. financial data about a market like its index,
3. financial data about government activities and banks,
4. political circumstances,
5. geographical and meteorological circumstances like natural or unnatural disasters.

Financial indicators for fundamental analyses include company figures like Return On Investment (ROI) which changes at least monthly. They also include country level data, which is represented by financial indicators like foreign exchange rates with a high rate of change and other economic factors with a lower rate of change like the inflation rate. As today's economies are delineated not only by borders, financial indicators on a sector level like banking or the automobile sector are also relevant. A problem in a given sector normally affects all enterprises operating in this sector, independent of the country.

2.2.2. Technical Analysis

The technical analysts, in contrast,s believe in repetition of historic market movements. Their search for patterns is normally supported by charts, which is why they are often referred to as chartists. They only trust in historic time series and do not consider any fundamental data. More and more such analysis techniques are supported by mathematical models because of the amount of data that can easily be processed by computers and the ability to define special algorithms for pattern recognition (see section 2.4).

2.3. Technical Indicators

In this subsection financial indicators based on historic market data time series are discussed. There exists a long list of technical indicators, indicating different perspectives on the price trend. Some of these indicators are simple to calculate, like return or volatility, but the point of technical analysis is the interpretation and prediction of possible future trends. In Berkin and Swedroe (2016) a multi-step approach for factor based modeling is explained which uses different technical indicators

Beta factor: The simplest model is the *One-Factor-Model* which uses the market beta. This model is related to the Capital Asset Pricing Model (CAPM). The beta defines the variation of a stock compared to the overall market. The interpretation of beta is the following:

- Beta > 1 indicates more risk than the market
- Beta < 1 indicates less risk than the market
- Beta of 1 means that the price moves proportionally to the market
- Beta of 0 means that the price is independent from the market

Berkin and Swedroe (2016) refer to the 2011 study by Dimson, Marsh, and Staunton entitled “Equity premiums around the world”, where the authors determined that since 1900 market beta has been positive in almost every country and region around of the world.

Size factor: The second factor Berkin and Swedroe (2016) mentioned is size. Based on the 1981 study by W. Banz “The relationship between return and market value of common stocks”, the market beta does not fully explain the higher average return of smaller stocks. The size factor is calculated by taking the annual average return of small-cap stocks and subtracting the annual average return of large-cap stocks (Small Minus Big). The meaning of this factor is that investing in small caps has a higher risk than large caps.

Value factor: is used to identify stocks which have low stock prices in relation to their fundamental data like cash flow, price earning ratio, dividend yield. This factor is used in addition to size and the market factor to define the “Fama and French Three Factor Model” Jegadeesh and Titman (2011). Based on the combination of these three factors more than 90 percent of portfolio returns can be explained.

Momentum Factor: The momentum factor shows the tendency for a stock that had a good performance in the recent past to continue performing well (or vice versa). It measures the speed of a price change for a given time period and is an oscillator which is simple to calculate. The momentum factor is calculated as difference of the price from today(P_t) to the price in the past (P_{t-10} time period 10 days).

$$MOM = P_t - P_{t-10} \tag{2.1}$$

An extension of the momentum factor is used by Carhart’s “Four-factor model”, as mentioned in Berkin and Swedroe (2016). This type is also called cross-sectional momentum by Jegadeesh and Titman (2011). Cross-sectional momentum measures relative performance by comparing the return of an asset relative to the returns of other assets within the same asset class. The cross-sectional momentum factor can be seen as a buy indicator for an asset class because it indicates the best relative performance (long position). On the other hand a bad relative performance indicates a signal to sell the position (short position). Even if all assets of an asset class trend to rise, the cross-sectional momentum show the best and worst performer in an asset class. The other type of momentum is called time-series momentum (Jegadeesh and Titman, 2011) or trend-following momentum, which measures absolute performance relative to the trend of its own performance. In contrast to the previous absolute performance relative to the trend of its own performance no signal to sell.

The factors described so far are based on *Your Complete Guide to Factor-Based Investing: The Way Smart Money Invests Today* by Berkin and Swedroe (2016). In addition, there are more statistical factors which use historical time series. The following briefly describes the most useful indicators based mainly on “FM Labs”[1] and “Investopedia”[2]. An in depth overview is provided by Fang, Qin, and Jacobsen (2014).

Simple Moving Average (SMA): represents the average price for a period of time, whereby each price is equally weighted. The time period depends on the strategy of the investor, but often uses time periods of 20, 50, 100 and 200 days, and then compares the results

[1] https://www.fmlabs.com/reference/

[2] https://www.investopedia.com/dictionary/

which is less responsive to recent changes.

$$SMA = \frac{\sum_{i=1}^{n} price}{n} \tag{2.2}$$

A typical strategy for using SMA could be

- SMA (20) > SMA(250) signal for buy
- SMA (20) < SMA(250) signal for sell

Williams %R: is a momentum based oscillator used to identify overbought and oversold conditions, as defined by Larry Williams. The %R is based on a comparison between the current close and the highest high for a user defined look back period. %R is between 0 and -100.

$$\%R = 100 * \frac{HighestHigh(lastnperiods) - close}{HighestHigh(lastnperiods) - LowestLow(lastnperiods)} \tag{2.3}$$

The interpretation is as

- %R close to 0: overbought
- %R close to -100: oversold

Relative Strength Index (RSI): is an oscillator which measures the speed and change of the price fluctuations and ranges between 0 and 100.

$$RSI = 100 - \frac{100}{(100 + \frac{AverageGain}{AverageLoss})} \tag{2.4}$$

The interpretation is as

- RSI > 70: overbought
- RSI < 30: oversold

Rate of Change (RoC): is a indicator of change relative to previous periods used to determine how fast the quote is changing. Usually the factor is 100 and is used only to ease interpretation of the numbers. Actually the function can be used for any data series and is not only used in the financial field. If it is used in the financial sector it is often referred to as the Price Rate Of Change (PROC).

$$\begin{aligned} &ROC = (\frac{P_t}{P_{t-1}} - 1) * 100 \\ &P_t = Price\,today \\ &P_{t-1} = Price\,yesterday \end{aligned} \tag{2.5}$$

Disparity Index Indicator (DII): is a technical indicator to measures the relative position of an asset's most recent closing price to a selected moving average and reports the value as a percentage.

- DI > 0: rising price and suggests a gaining upward momentum
- DI = 0: price is exactly consistent with its moving average
- DI < 0: a sign that selling pressure is increasing, forcing the price to drop

An Interpretation of the DII is if the value crosses the Zero line, this is an early signal of an imminent rapid change in the trend. High values, whether positive or negative, may indicate an upcoming price correction.

$$DisparityIndex = \frac{price_{close} - average_n}{average_n} * 100 \tag{2.6}$$

Typical values are 5 and 10.

Commodity Channel Index (CCI): is an indicator to identify beginning and ending market trends. The value is normally between -100 and 100. Higher or lower values are a signal for overbought or oversold market conditions. If the price rises but the CCI does not, a price correction can be indicated. The Commodity Channel Index was introduced by Donald Lambert in 1980.

$$\begin{aligned}
&CCI = \frac{TP - MA_{TP}}{0.015 * MD_{TP}} \\
&TP = \frac{high_{-n} + low_{-n} + close}{3} \\
&TP = Typical\ Price \\
&high_{-n} = Highest\ high\ in\ the\ last\ n\ time\ periods \\
&low_{-n} = Lowest\ low\ in\ the\ last\ n\ time\ periods \\
&MA = \ Moving\ Average \\
&MD = Mean\ Deviation
\end{aligned} \tag{2.7}$$

Stochastic %K and %D: are oscillators to measure the closing price in relation to the recent trading range. The indicator was introduced by George C. Lane and values lie between 0 and 100. The following terminology is used to differentiate different types:

- Fast Stochastic: Refers to both %K and %D where %K is un-smoothed
- Slow Stochastic: Refers to both %K and %D where %K is smoothed
- Raw %K: Un-smoothed %K
- Fast %K: Un-smoothed %K
- Slow %K: Smoothed %K
- Fast %D: Moving average of an un-smoothed %K

- Slow %D: Moving average of a smoothed %K, in effect: a double smoothed %K
- %D: Always refers to a smoothed %K (whether or not the %K itself is smoothed)

The numerical interpretation is

- %D values over 75 indicate an overbought condition
- %D values under 25 indicate an oversold condition
- When the Fast %D crosses above the Slow %D, it is a buy signal
- When it crosses below, it is a sell signal
- The Raw %K is generally considered too erratic to use for crossover signals.

$$\%K = 100 * \frac{close - LowestLow_{[lastnperiods]}}{HighstHigh_{[lastnperiods]} - LowestLow_{[lastnperiods]}} \tag{2.8}$$

$$\%D = MovingAverage(\%K) \tag{2.9}$$

On Balance Volume (OBV): is a momentum based indicator to measures a volume flow to determine the direction of the trend. Volume and price rise are directly proportional. A rising price is shown in a rising OBV, while a falling OBV stands for a falling price. If the OBV indicator rises in the same pattern as the prices this is a positive signal. In contrast it is a negative sign if the OBV follows a falling price in the same pattern. The raising and falling patterns can be deduced that the price trend is sustainable. However, if the OBV shows a decline while prices rise, this could signal a turnaround.

$$OBV = OBV_{t-1} + \begin{cases} volume, & if close > close_{t-1} \\ 0, & if close = close_{t-1} \\ -volume, & if close < close_{t-1} \end{cases} \tag{2.10}$$

Average Directional Movement Index (ADX): is a indicator to determine if the price trend is lasting. Hence it is also named the ultimate trend indicator. ADX uses the moving average of price range expansion over a given period of time, which is often 14 days. The values are between 0 and 100 and shows normally in graphs the trend of the price.

- ADX 0-25: absent or weak trend
- ADX 25-50: strong trend
- ADX 50-75: very strong trend
- ADX 75-100: extremely strong trend

The formula for Directional Movement is

$$ADX_{-t} = \frac{((ADX_{t-1} * (n-1)) + DX_t)}{n} \tag{2.11}$$

Absolute Price Oscillator (APO): displays the difference between two exponential moving averages of a price and is expressed as an absolute value.

- APO > 0: shows bullish conditions and indicates an upward movement
- APO < 0: shows bearish conditions and indicates a downward trend.

Deviations occur when a new high or low price is not confirmed by the Absolute Price Oscillator. A bullish divergence occurs when the price reaches a lower low, but the APO reaches a higher low. This indicates a lesser downward movement that could anticipate a bullish turn. A bearish divergence occurs when the price reaches a higher level, but the APO is lower. This shows a lower upward movement that could predict a downward movement[3].

$$APO = ShorterPeriodEMA - LongerPeriodEMA \tag{2.12}$$

In the next section, Machine Learning is explained in detail. It becomes recognized as a useful tool for combining different features either from quantitative or from qualitative data to predict stock price movements.

2.4. Machine Learning

Machine Learning (ML) is one approach of Artificial Intelligence (AI) which has become more and more popular in the last years, although the underlying techniques are older. One driver for ML in the last decade was the amount of information which comes with Big Data and the increasing computer power to process data within a reasonable time. The term was first mentioned by Samuel in 1959:

> *“... the field of study that gives computers the ability to learn without being explicitly programmed."*

The term *Machine Learning* is mostly related to Data Science, whereas statisticians use the term *Statistical Learning*, which is defined as (James, Witten, Hastie, and Tibshirani, 2014):

> *“... formalization of relationships between variables in the form of mathematical equations."*

The common baseline between both is *Learning from Data*. ML is structured in three different types:

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

In the next section the different types are outlined more in detail.

[3] https://www.fidelity.com

2.4.1. Supervised Learning

This ML method uses an external supervisor with additional knowledge of the data available. The data becomes a set of labels or a numeric target variable; for example each set of stock market data belongs to either the bullish or bearish set. Given a set of features such as the technical indicators, the supervised machine learning algorithm can be used to predict whether the price will rise or not (James et al., 2014). Based on the learning with supervision, the system should be able to predict the label for future data based on the learning process. This method is the most commonly used machine learning method, although most data sets are not labeled. Supervised Learning uses classification or regression methods to predict an output variable. The prediction is based on a data set which is called training data used to learn the model. Each input data set is mapped to an output label of class. Hence this step is known as classification. The output variable can either be a binary classification (e.g. buy/sell or rise/fall) or multi-label classification (e.g. news category like finance, politics, sports). Widely used classification algorithms are logistic regression, Support Vector Machines (SVM), Neural Networks like random forests and gradient boosting (see section 2.4.5), K-nearest neighbors or decision trees.

2.4.2. Unsupervised Learning

In contrast to supervised learning, unsupervised learning tries to find patterns in the data without any pre-labeling. Hence the algorithm cannot use a linear regression model to predict a response variable, because this variable does not exist. Unsupervised Learning uses a metric such as distance to group the data and tries to learn inherent latent structures, patterns and relationships from given data without any supervision. The statistical method behind is called clustering (James et al., 2014). Based on multiple variables the data is clustered, which depends on the quality of the measurement of data and the selection of the variables. For example Unsupervised Learning is used to suggest possible labels for tweet sentiment by using English vocabulary.

2.4.3. Reinforcement Learning

Reinforcement Learning (RL) is the third group of ML approaches and is close to the human way of learning. RL is based on the principle of “Trial and Error” where actions are observed and evaluated. The feedback is based on rewards or penalties.

As shown in Figure 2.1 the action is triggered by an algorithm with the aim to maximize the rewards, which leads to the best solution of the problem Sutton and Barto (2018). The rewards or penalties can be mapped to numerical values, which serve as input parameters in the next step of the evaluation.

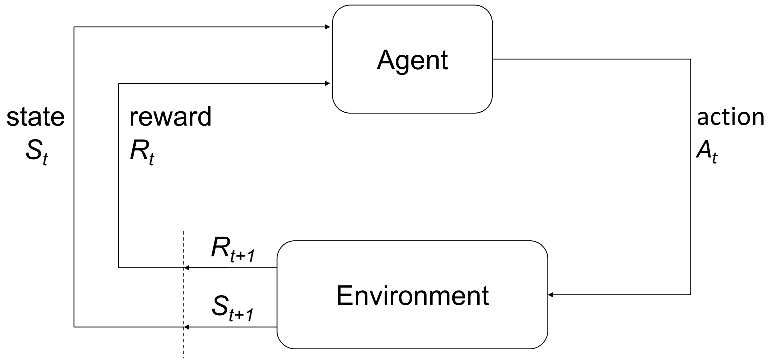


Figure 2.1.: Markov decision process for RL based on Sutton and Barto, 2018

2.4.4. Learning Algorithms

Supervised learning algorithms are used to solve both classification and regression problems. Popular methods are:

Linear Regression: which is used to predict continuous numeric outcomes such as stock prices (James et al., 2014).

Logistic Regression: is a popular classification algorithm that is especially used in the credit industry in order to predict loan defaults.

k-Nearest Neighbors: The k-NN algorithm is a classification algorithm that is used to classify data into two or more categories, and is widely used for classification.

Tree-Based Algorithms: Tree-based algorithms such as decision trees, Random Forests, and Boosted trees are used to solve both classification and regression problems.

Naïve Bayes: is an algorithm that uses the mathematical model of probability to solve classification problems.

Support Vector Machine: The Support Vector Machine (SVM) algorithm is a supervised learning technique for classification and regression tasks introduce by Vapnik and Lerner in 1963. It is also used in applications such as image and face detection and for handwriting recognition. The aim of the algorithm is to find the maximum margin hyperplane which separates the values into two different classes. This maximum vector is then called a support vector. The advantages of SVM are the small set of control parameters, which makes over-fitting very unlikely.

In the field of stock price prediction, Support Vector Machine (SVM) algorithms have become the preferred method (Nassirtoussi et al., 2014) and are used in several related works (Huang, Nakamori, and Wang, 2005; Kim, 2003; Tay and Cao, 2001; and Zhai, Hsu, and Halgamuge, 2007). Beckmann (2017) provides an overview of the different approaches used

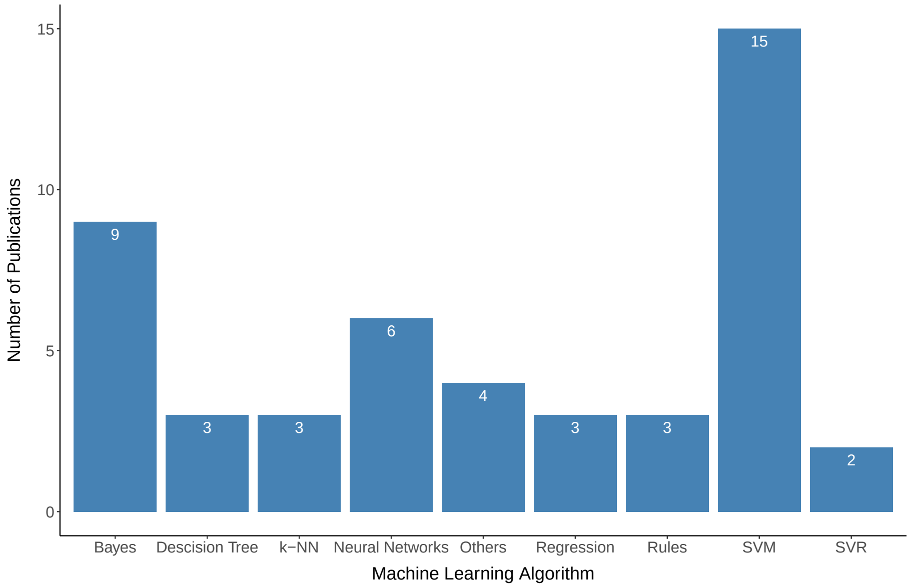


Figure 2.2.: Comparison of the ML approaches based on Beckmann, 2017

in the literature. The most used method is SVM, with Naïve Bayes placing second. In this overview no Gradient Boosting Algorithm was listed. In recent years, however, XGBoost has turned out to be a highly effective algorithm. This is outlined by Chen and Guestrin (2016) through the successes at the machine learning competition offered by Kaggle[4]. From 29 submitted solutions 17 used XGBoost, which are listed and compared on github (https://github.com/dmlc/xgboost/tree/master/demo). Hence, XGBoost is the selected approach for this work and it is therefore described in detail below.

2.4.5. eXtreme Gradient Boosting

Gradient boosting was proposed in the article “Greedy Function Approximation: A Gradient Boosting Machine” by Friedman (2000). The first boosting algorithm was AdaBoost: short for Adaptive Boosting. eXtreme gradient boosting (XGBoost or XGB) is a library for boosted tree algorithms with the aim to provide a scalable, portable and accurate machine learning algorithm for large scale tree boosting. The project is hosted at https://xgboost.ai/ and the library is available on github https://github.com/dmlc/xgboost. XGBoost is used for supervised learning problems, where training data with multiple features x_i are used to predict a target variable y_i.

Boosting is an approach for improving the results of decision trees. Boosting copies the decision trees multiple times, whereby each tree is created sequentially and includes the

[4] https://www.kaggle.com/

results of the previous tree. Unlike fitting a single large decision tree to the data, which amounts to a hard fitting of the data and potentially to overfitting the boosting approach instead learns slowly.

The full mathematical background is provided in Chen and Guestrin (2016) and Friedman (2000). For this thesis the reduced formula from Paradkar (2017) is sufficient.

$$Obj = L + \Omega \tag{2.13}$$

Loss function L: models the predictive power. L needs to be optimized e.g. by means of Root Mean Squared Error (RMSE) or Logloss by using binary classification

Regularization component function Ω: controls simplicity and overfitting and depends on the number of leaves and their prediction score

The algorithm of XGB is explained in Natekin and Knoll (2013) based on the work of Friedman (2000)

Algorithm 2.1: Friedman‘s Gradient Boost algorithm

Inputs:

- input data $(x, y)_{i=1}^{N}$
- number of iterations M
- choice of the loss-function $\Psi(y, f)$
- choice of the base-learner model $h(x, \theta)$

Algorithm:

```
initialize $\widehat{f}_0$ with a constant
for t = 1 to M do
    compute the negative gradient $g_t(x)$
    fit a new base-learner function $h(x, \theta_t)$
    find the best gradient descent step-size $\rho_t$:
      $\rho_t = arg\ min_\rho \sum_{i=1}^{N} \Psi[y_i, \widehat{f}_{t-1}(x_i) + \rho_t h(x_i, \theta_t)]$
    update the function estimate:
      $\widehat{f}_t \leftarrow \widehat{f}_{t-1} + \rho_t h(x, \theta_t)$
end for
```

The advantage of the algorithm 2.1 is that it supports various different loss functions like Gaussian L2, Laplace L1 or the Huber loss function. For this work a categorical response ($y \in 0, 1$) in form of a binomial classifier is used.

The XGBoost implementation developed by Chen is designed for computational speed and model performance and includes the following characteristics (Brownlee, 2018, p. 15):

1. Parallelization of tree construction using all of your CPU cores during training.

2. Distributed Computing for training very large models using a cluster of machines.
3. Out-of-Core Computing for very large datasets that don't fit into memory.
4. Cache Optimization of data structures and algorithms to make best use of hardware

According to James et al. (2014) boosting has three tuning parameters:

1. The number of trees B (B separate training sets), whereby boosting can lead to over-fit if B is too large. To test for overfitting, cross-validation (cv) can be used to find a suitable B.
2. The shrinkage parameter λ, to control the rate at which boosting learns (values are 0.01 or 0.001).
3. The number of splits in each tree, d, to control the complexity of the boosted ensemble (normal d = 1)

Tuning these parameters is essential to avoid overfitting. Tuning is an iterative process and depends on the domain. According to Brownlee (2018)) the best way is to iterate with varying parameters and plot the different behavior to find the optimal parameters.

Figure 2.3 provides an example of an XGBoost tree for one of the following case studies. This tree is a multiple tree visualization whereby several trees are shown in one graph. The features used for this decision tree are shown in the node. In the root node only technical indicators are used, whereas on the next level additional news features are added. This sample tree has only a depth of two and already has leaves on the first and the second levels.

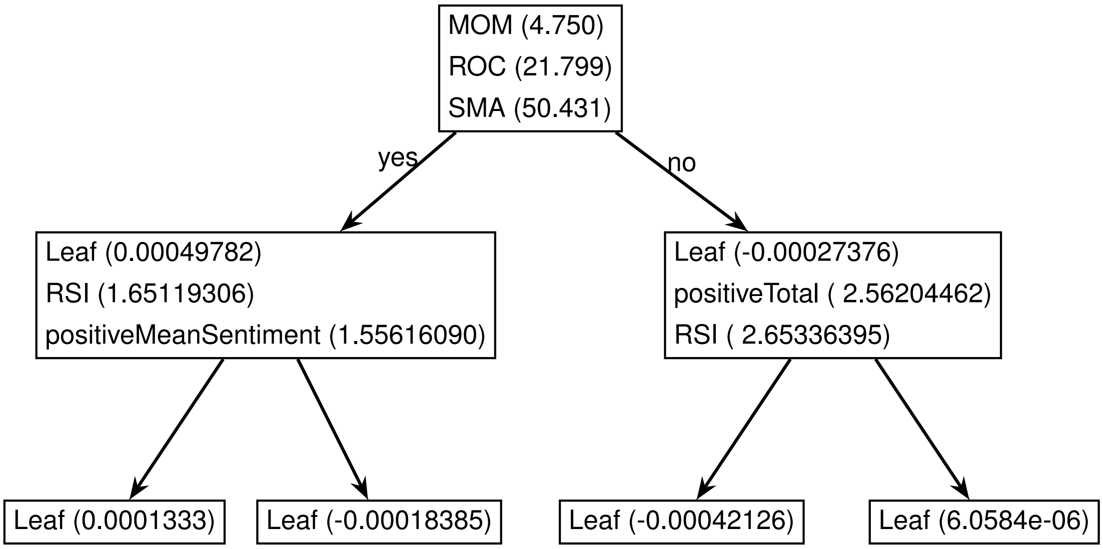


Figure 2.3.: Example for a XGBoost Tree based on the case study

In the node provides the value gain, which is used to classify the features. Gain is the relative contribution of the feature to the model (see section 3.4.2)

2.5. Summary

This chapter has presented the theoretical background for stock market prediction, which is popular topic in the scientific community. In the last two decades ML approaches have widely been used. While traditional ML algorithms are well discussed in the literature, the implementation of boosted trees XGBoost is a rather new and promising phenomenon in this field. Hence this thesis makes use of the XGBoost as a ML approach. The approach is explained in the next chapter.

3. Methodology

"Any sufficiently advanced technology is equivalent to magic."
Sir Arthur C. Clarke

The selected methodological approach is explained in this chapter, which requires topic-related literature to be discussed in further detail. Nassirtoussi et al. (2014) studied approaches of text mining for market data prediction and define a generic framework to provide a basis for comparing these different approaches. Figure 3.1 shows the generic common system components diagram.

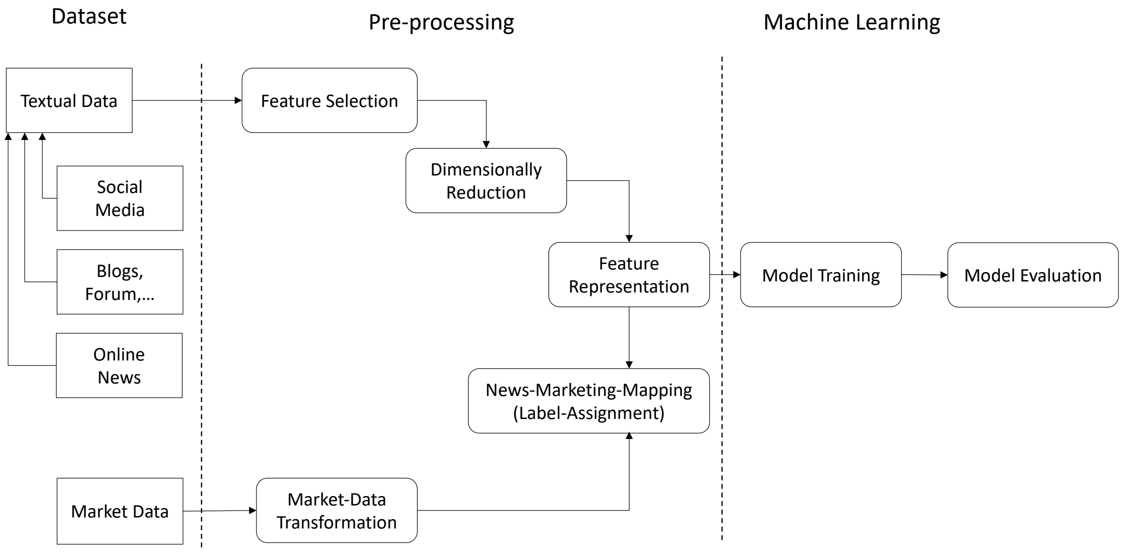


Figure 3.1.: Generic Common System Components Diagram based on Nassirtoussi et al. (2014)

The comparison framework consists of three parts:

1. *Data Sets*: this part defines the sources of market data and news data. Some approaches listed use either market or news data and few use both as features.
2. *Pre-Processing:* this step is used to transform the data into a suitable format and to select and reduce the features. The following sub-tasks are essential:

   a) Feature selection
   b) Dimensionality reduction
   c) Feature representation
   d) Combining news and technical data or signals

3. *Machine Learning:* the third part is the machine learning process step where different machine learning algorithms are used. This includes training and evaluation of the model.

In the next section different approaches are described to show related work in this area.

3.1. Related Work

The literature reveals various approaches to market data prediction with the use of ML algorithms. These approaches differ by the type of data (features) they use and the machine learning algorithm. For this thesis the comparison based on the features is of greatest significance, because the aim of the thesis is to combine technical indicators and public mood data for prediction. Categorization can be made by

- Market Data-Based Approaches
- Information-Based Approaches
- Combined Approaches

Market Data Based approaches use in most cases time series of stock data from different markets, but mostly US markets. Only few approaches use FOREX data as an input source. Besides the type of data, also the time frame, the period and the forecasting type are of interest. Table 2 from Nassirtoussi et al. (2014) shows that the majority of the compared approaches employ a daily time frame and only few try to predict on an intraday level using time frames of 5 min, 10 min or 20 min. This complex issue has a huge impact on the processing of the because market data and news data have to be synchronized and evaluated on a time-based level (Beckmann, 2017.) The period of input data has an influence on the ML algorithm. On average the approaches use several months for training data.

Information based approaches differ mainly in terms of the source of the news. Older approaches made use of traditional news sources, mainly financial news (Wuthrich et al., 1998), whereas newer approaches rely on social media data (Bollen and Mao, 2011; Zhang, Fuehres, and Gloor, 2011).

3.1.1. Market Data Based Approaches

Kim (2003) represents one of the first approaches using SVM for stock market prediction to be published, and uses twelve technical indicators like MOM, ROC, CCI, and RSI to predict the trend of the Korea Composite Stock Price Index (KOSPI). For the classification they used a binary logic whereby “1” indicated that the index is tomorrow higher and “0” that the index is tomorrow lower than today. The data set covered ten years starting with January 1989. The results showed that SVM can predict the direction of the market if the tuning parameters are correctly chosen. Therefore, they analyzed their model with various sets of parameters to optimize the prediction. Basak, Kar, Saha, Khaidem, and Dey (2019, pp. 552–567) use XGBoost as a ML engine with technical indicators and achieved a “remarkably high accuracy of prediction”. An approach to recommend buying or selling the cryptocurrency Bitcoin is described in Guo and Antulov-Fantulin (2018). They defined a model mainly based on the volatility to detect regimes by using a gate weighting function.

3.1.2. Information Based Approaches

One of the first approaches to predict the stock market was published by Wuthrich et al. (1998). The authors implemented a system to predict the trend of five major stock market indexes (DJIA, Nikkei, FTSE, HSE, STI) by collecting news data during the night from the web pages of the Financial Times, Wall Street Journal, Reuters, and some others. The sentiment analysis was performed using a manually defined dictionary and Neural Network Classifier to label the article with up, steady or down. The forecast was calculated overnight and they concluded: “...accuracy is significantly above random guessing and is absolutely comparable to what can be expected from human expert predictions.” Wuthrich et al. (1998, p. 6)

Bollen and Mao (2011) used the public mood by revealed by Twitter messages for prediction the authors collected 9,853,498 Twitter messages posted by approximately 2.7 million users. These messages were analyzed with a tool called OpinionFinder to measure positive or negative mood. The results were clustered across by using the six dimensions of the Google-Profile of Mood States (GPOMS), which are “Calm” “Alert”, “Sure”, “Vital”, “Kind”, and “Happy”. They assume that public mood can influence the stock market, and the results showed that a change in the public mood measured with the GPOMS is reflected in the Dow Jones Industrial Average (DJIA) 3 to 4 days later.

Zhang et al. (2011) also analyses Twitter messages and give them an emotional tag like “fear”, “worry” or “hope”, which are categorized in two groups: positive or negative. They find a negative correlation of the mood sentiment of tweets and the Dow Jones, NASDAQ and S&P500, and conclude that “...when the emotions on Twitter fly high, that is when people express a lot of hope, fear, and worry, the Dow goes down the next day. When people have less hope, fear, and worry, the Dow goes up.” Zhang et al. (2011, p. 61).

These publications show that social media news are a useful tool to measure the public mood and to use it for predictions. Further approaches are Mittermayer (2004) and Schumaker and Chen (2009a) whereby the latter suggests an approach where the prediction is within 20 minutes after the news message was published.

3.1.3. Combined Approaches

Zhai et al. (2007) describe a combined approach of news and technical indicators for daily stock prediction. They used eight technical indicators (Momentum, ROC, Williams %R, A/D Oscillator, Disparity 5, Stochastic %K and Stochastic %D) and the Australian Financial Review to predict the direction of the stock BHP Billiton Ltd. For the ML they used a SVM. Based on the results they showed that the combined approach enhanced the predictive ability of the system.

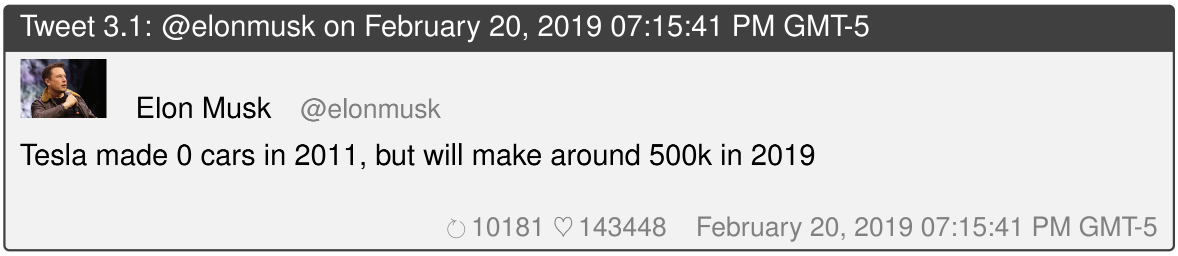
3.2. Web Intelligence Knowlege Base

webLyzard is a media intelligence platform developed at MODUL University Vienna. The focus is on collecting and analyzing information regarding various domains like politics, sports, and climate change (Scharl et al., 2017). Even the series Game of Thrones (Scharl et al., 2016) is used as an example to analyze social media news data (Scharl and Herring, 2013). Moreover, the framework is able to identify events and sub-events, as well as to classify topics (Scharl and Fischl, 2015). The tracking covers information from a range of different media channels in selected countries in three languages (English, German and French). The main tracked channels are social media including Twitter, Facebook and YouTube, and the web portals of news organizations, companies, municipalities, and environmental NGOs (Scharl and Fischl, 2015).

There are several successful examples of the usage of the webLyzard platform. In Scharl and Herring (2013) the web intelligence platform was used to monitor news concerning climate change. The project initiated by the National Oceanic and Atmospheric Administration was named *Media Watch on Climate Change* and is available as a public platform[1]. The searched keywords are enriched by metadata and are categorized by document sentiment as positive, neutral, or negative. For its visual presentation in a dashboard (user interface) the sentiment is mapped via a color coding (Scharl and Fischl, 2015).

A recent example of the reaction of stock market prices to Twitter messages involves statements published by Elon Musk, CEO of Tesla, in mid February 2019. Musk's tweet is shown in tweet 3.1, and suggests that the planned production of 400k vehicles per year would strongly increase to "around 500k".

Tweet 3.1: @elonmusk on February 20, 2019 07:15:41 PM GMT-5



Elon Musk @elonmusk

Tesla made 0 cars in 2011, but will make around 500k in 2019

↻ 10181 ♡ 143448 February 20, 2019 07:15:41 PM GMT-5

[1] https://www.weblyzard.com/showcases/

Just a few hours later, Musk sought to clarify this statement by issuing a second tweet which expressed what he "meant to say" (see tweet 3.2). With this tweet Elon Musk violated his agreement with the Securities and Exchange Commission (SEC) by tweeting inaccurate information. United States SEC wrote a few days later in the court filing "Musk did not seek or receive pre-approval prior to publishing this tweet, which was inaccurate and disseminated to over 24 million people" (Securities and Commission, 2019, p. 1.)

Tweet 3.2: @elonmusk on February 20, 2019 11:41:03 PM GMT-5

Elon Musk @elonmusk

Meant to say annualized production rate at end of 2019 probably around 500k, ie 10k cars/week. Deliveries for year Deliveries for year still estimated to be about 400k.

⟳ 1820 ♡ 51674 February 20, 2019 11:41:03 PM GMT-5

The reaction of the stock price reflected this a few days later (see chart in figure 3.2; left gray line reflects the tweets, the right gray line the SEC announcement). Tesla shares fell about 4% hours after of this announcement was published and heavily discussed in the news.

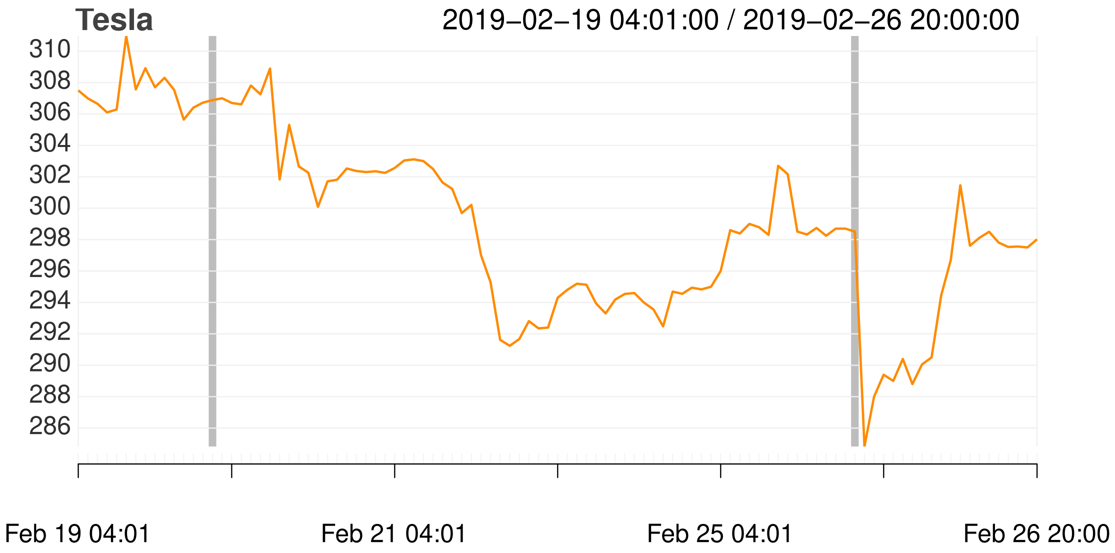


Figure 3.2.: Tesla stock chart movement after the SEC announcement

3.2.1. Sentiment Analysis

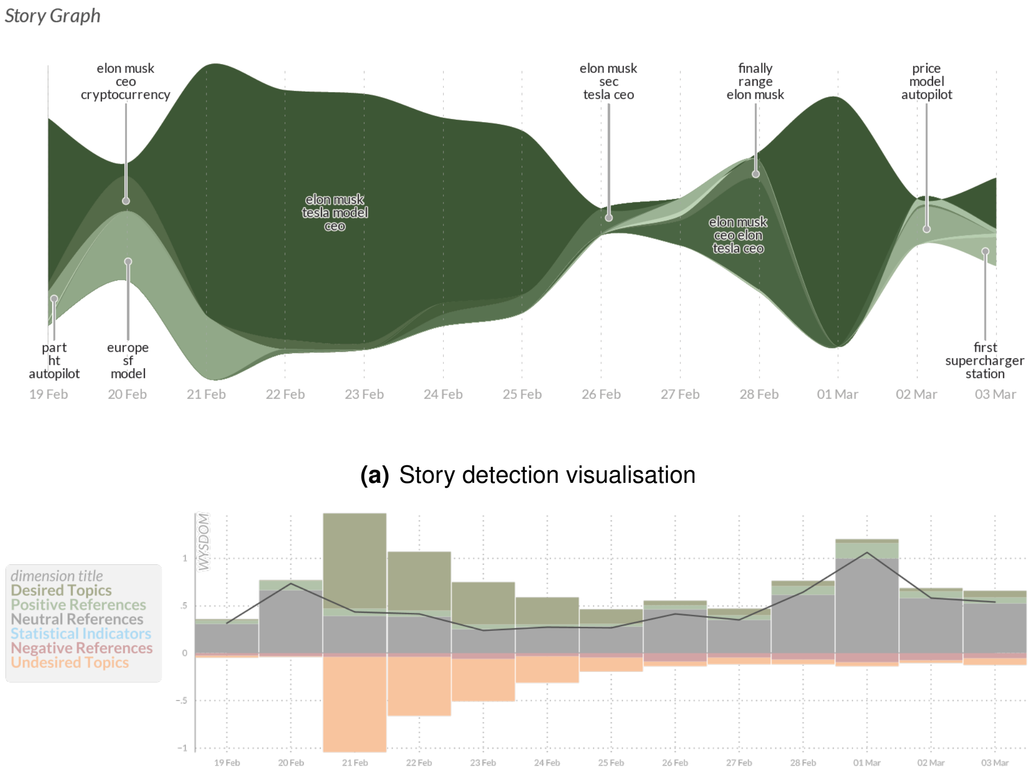
Sentiment analysis is used to measure the public mood. This opinion mining task is done by the webLyzard system, with a defined sentiment lexicon and techniques for extracting and interpreting the news from online news media and social media. Nassirtoussi et al. (2014) also compares different approaches for detecting sentiment and the types of news which are used. As mentioned in section 3.1.2 Bollen and Mao (2011) uses a categorization for the analysis of the Twitter messages to measure the public mood and use this as a feature to enhance prediction. Some approaches try to use a specific ontology for specific domains. Ruiz-Martìnez, Valencia-Garcìa, and Garcìa-Sànchez (2012), for example, define an ontology for terms related to the financial domain (like 'asset', 'stock market') and use three entity lists, semantic (like 'Dividend', 'Income Tax'), sentiment (like 'growth','earning','to cut') and modifier (like 'extremely', 'this morning') to have a better accuracy. The webLyzard system enables sentiment analysis yielding values from -1 (entirely negative) to 1 (entirely positive), yet this functionality is not tailored to the financial sector. Such modification would be possible by extending the underlying lexicon with polarity terms and phrases.

3.2.2. WYSDOM Metric

The webLyzard framework defines a success metric named webLyzard Stakeholder Dialog and Opinion Model (WYSDOM), which "goes beyond the bipolar assessment of sentiment (...) and allows real-time insights into the success of public outreach activities" Scharl et al. (2017, p. 767). The main focus is to measure if the communication targets have been fulfilled and if the selected communication strategy was successful, which makes it a useful tool for marketing departments seeking to monitor their success in positioning their brands.

The WYSDOM metric evaluates the degree of association between an organization and various topics, whether desirable or undesirable. Such associations have the potential to influence the stock price of an organization listed on the stock market. Even topics which are only indirectly related to the organization such as sector-wide news, financial events, or industry-specific regulations could be also reflected in the metrics. News and other external resources like Google Analytics are used for the calculation of the metrics, which are reflected in the KPIs. A visual representation for the Elon Musk case is shown in 3.3(b), where the gray line indicates the overall WYSDOM score. Scharl et al. (2017) provide the following further description of the success metrics:

1. Dark green the desired topics and the number of positive references in light green;
2. Gray area the number of neutral references;
3. Red areas below the axis the number of negative references and
4. Orange the association with undesired topics.



(a) Story detection visualisation

(b) WYSDOM metric

Figure 3.3.: Story detection by webLyzard about the Eleon Musk Tweets and the Securities and Exchange Commission announcement

3.3. Defined System Architecture

The system architecture for this work is shown in figure 3.4 and outlines the concept of combining market data (quantitative data) and news data (qualitative data) as a foundation for prediction.

Referring to the comparison framework of Nassirtoussi et al. (2014) the proposed approach used two sources of input data. On the market data side, any market data provider can be used which delivers quotes and volumes with time stamps. Most of the technical indicators can be calculated based on these values. For some indicators like beta additional data and further configuration would be needed. For example to calculate a beta an appropriate index has to be defined and the time series of the index has to be loaded in addition. Some providers like Alpha Vantage (2019) also offer pre-calculated technical indicators. Some providers offer an API which allows direct processing of the market data in real-time.

The second part of the dataset input is covered by the Web Intelligence platform webLyzard, which also offers an API for integration into the system architecture. The main advantage of the usage of webLyzard system is the well-proven sentiment analysis functionality and the flexible query engine (including regular expressions: Scharl and Fischl, 2015). This allows

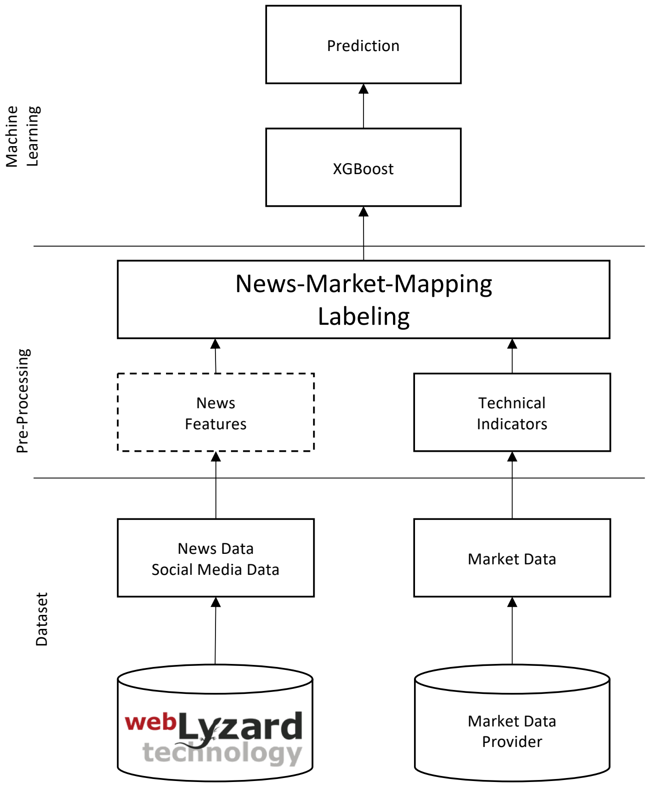


Figure 3.4.: Defined system architecture for the prediction with webLyzard-powered public mood data

for the adaption of news extraction parameters for each case and even the combination of company news with sector news or country news. This is important for the banking sector because public authorities like the European Central Bank announce money market decisions which also influence the strategies of banks.

The pre-processing stage maintains the separation between the market data and the news data. For the market data pre-processing mainly involves the calculation of technical indicators. It is important that the calculation of technical indicators reduce the length of the time period, as the full period cannot be used for the ML algorithm. For example, the calculation of the daily return needs the quote from a particular day and the preceding day, hence this value is not available on the first day. For the technical indicator ADX, if the parameter n = 20 days, a gap of 20 days results. Hence if the desired time period for the analysis is six months, least seven months have to be loaded to have a complete data set for the calculation.

The features provided by webLyzard do not need any further pre-processing, as the necessary tasks are already performed by webLyzard. It calculates the needed statistical figures and makes these available through an export function. This means that, in contrast to the market data, the news information is available in a disaggregated format, and more detailed analyses have to be made within the webLyzard system through defined analysis reports like tag cloud and story detection (Scharl and Fischl, 2015).

The step of combining market data with news data is named “News-Market-Mapping”. By choosing a daily time frame, the data sets can be joined according to the date. For real-time predictions or for time frames of minutes or hours, a synchronization mechanism between news and market data is necessary. After data of two sources are joined, the labeling has to be added. As shown in section 3.1 most publications use a binary classification as forecast type.

The forecast type for this work uses the *daily return* for the prediction. This means the outcome is of binary character and predicts the return for the next day.

forecast	meaning
rise	the return from today to tomorrow will be positive or 0
fall	the return from today to tomorrow will be negative

Table 3.1.: Forecast types for the defined model

Each day of the data set is enriched with a label (classification): “0” for negative daily returns (fall), and “1” otherwise (rise).

Algorithm 3.1: Labeling of the features

```
for t = 1 to days do
    calculate daily return
    determine sign of return
    add binary label to the features
    shift the label to the next day
end for
```

The complete dataset consists of market data and news features, with technical indicators and labels as the respective inputs for the Machine Learning. Until here the feature set it is independent from the actual ML method used. The input data and the parameters for the ML model configuration determines the results generated by the model. The input data has to be split into training, validation and test data (see figure 3.5).

Training dataset: used to fit the model

Validation dataset: used to evaluate the model fit with the training dataset and the model hyperparameters.

Test dataset: used to evaluate the final model fitted on the training dataset.

The standard split rate has been discussed in detail in the community, with the normal division between training and test data lying somewhere between 70/30 and 80/20. The data for the

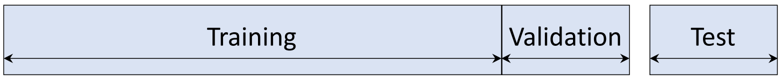


Figure 3.5.: Input data is split into three parts

validation is afterwards taken from the training data set to verify the model. A split rate of 75/25 was defined for this work.

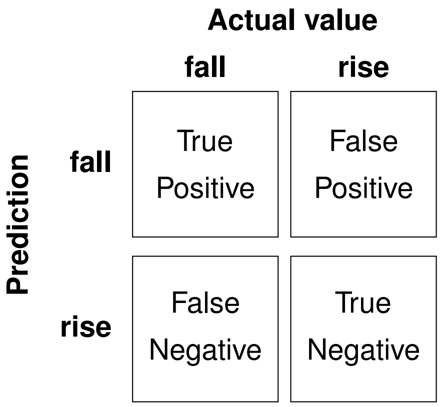
The next section gives an overview of how to evaluate the results of the model.

3.4. Evaluation

The evaluation of the model is based on figures generated from the results of the machine learning algorithm and additional metadata of the calculation. Based on these results the model can be improved to avoid overfitting.

3.4.1. Confusion Matrix

The confusion matrix is an appropriate method to describe the performance of a classification model. Figure 3.6 shows an example of a binary classifier as used in the proposed model.



		Actual value	
		fall	**rise**
Prediction	**fall**	True Positive	False Positive
	rise	False Negative	True Negative

Figure 3.6.: Confusion matrix to evaluate the model based on James et al. (2014, p. 148)

The model defines two possible values “rise” and “fall”, whereby according to table 3.1 rise means a positive return and fall means a negative return. In the lower row all the prediction for rise are listed and in the upper row all the prediction for fall are listed. In total this is the sample rate used for the calculation. The combination of predictions (rise/fall) and actual values (rise/fall) generates four possible outcomes in the confusion matrix:

True Positive (TP): Cases the model predicted fall and prediction is correct.

True Negative (TN): Cases the model predicted rise and prediction is correct.

False Positive (FP): Cases the model predicted fall and the prediction is incorrect (type I error).

False Negative (FN): Cases the model predicted rise and the prediction is incorrect (Type II error).

3.4.2. Key Metrics

Based on James et al. (2014) and "Towards Data Science" (2019) some key metrics like accuracy, precision and recall and also f-score for the model evaluation and their interpretation are introduced.

Accuracy is one metric to evaluate classification models, but it has to be used with caution (accuracy paradox). It shows how accurate predictions are on average and can be calculated with

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{3.1}$$

Accuracy paradox indicates that in some cases a model with a lower Accuracy can have a higher predictive significance. Hence two additional factors, Recall and Precision, have to be evaluated.

Precision looks at the right column and calculates the accuracy of the rise predictions:

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

False positives potentially imply costs, because the model suggests that returns would rise when actually they fall, which could cause the loss of money.

Recall, also named the True Positive Rate or sensitivity, looks at the first row of the confusion matrix. The value expresses the relation between all correctly positive results and all actually positive results.

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

Hence recall shows the fraction of all actually positive outcomes (actual rises) which are classified correctly as positive. This gives the proportion of increased returns which are correctly predicted by the model. Cases where actually positive results are not correctly predicted represent missed opportunities to make financial gains if the investor follows the suggestions provided by the model.

The F-score is used to determine the balance between Precision and Recall, which provides a measure of how accurate a model is.

$$F = \frac{2 * (Recall * Precision)}{(Recall + Precision)} \tag{3.4}$$

The Receiver Operating Characteristics (ROC) defines a curve visualizing the performance of the classification model at all classification thresholds. The ROC curve uses FPR on the x-axis and TPR on the y-axis (figure 3.7 shows an example).

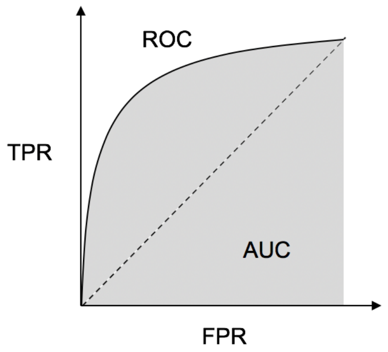


Figure 3.7.: Example for a ROC chart with visualization of AUC

Hence ROC is a probability curve and a measurement of the quality of the model. Based on the ROC curve, the Area Under The Curve (AUC) provides a measure of separability, which reflects how well the model is able to distinguish between classes. For a random binary classifier the AUC is 0.5%, hence the aim is to push the AUC towards to 1, which would mean a perfect rate between FPR and TPR.

In a boosting tree three parameters characterize the importance of features and are used build the tree based on the weak learner principle (Scott, 2019):

Gain: The average training loss reduction gained when using a feature for splitting.

Cover: The number of times a feature is used to split the data across all trees weighted by the number of training data points that go through those splits.

Weight: The number of times a feature is used to split the data across all trees.

There are two further values for the model evaluation, the training error and test error.

Training error: is calculated with the same data on which the model was trained.

Test error: is calculated with a different dataset. If the training error is low and test error high, it is an indicator for overfitting.

In Rasekhschaffe and Jones (2019), the gap between training and test error is discussed (in-sample and out-of-sample error rates). In-sample error rates are always lower than out-of-sample error rates. In their example they increased the number of iterations (400 boosting iterations), but the model begins to overfit beginning from the fiftieth iteration.

3.5. Summary

In this chapter, the selected methodological approach was defined and compared to the related works of the scientific community. To sum up, the advantages and distinctive features are:

- XGBoost offers more parameter for easier tuning than SVM does.
- XGBoost as machine learning algorithm for a combined approach, which uses market data, news data and social media data as features for the machine learning algorithm
- In the sientific literature is no evidence for the usage of XGBoost in a combined approach.
- In contrast to all other approaches, the usage of webLyzard platform offers the integration of an established tool for news channel data analysis. Hence the processing of information is a well established process, which nevertheless needs some adjustments. Additional specific news feeds for financial data are a useful extension.
- The market data part can be seen as an enhancement of the webLyzard and also the WYSDOM metrics. The stock quote of a company can also be used as a KPI and reflects the current situation on the market.
- webLyzard collects news media content in nearly real-time. Some other providers also offer stock quotes in real-time. From a long term perspective a near real-time prediction is desirable.
- The defined technical indicators are comparable to Kim (2003) and Zhai et al. (2007).
- Most of the previous models have been applied to US markets, or specific Asian markets. This thesis will compare the predictability of companies from Austria, the United Kingdom and the United States.
- The approach uses six months of data for training and one month of data for testing.
- This approach uses news data and social media data, whereas most other approaches use either one or the other.

The next chapter explains the implementation of the prototype in detail.

4. Implementation of the Prototype

"Truth can only be found in one place: the code."

Robert C. Martin

In this chapter the case studies are defined and the implementation of the prototype is described in detail. As outlined in chapter 3, the process takes several steps to calculate and then visualize the results. First of all the case studies are defined. Afterwards the prototype is implemented using the statistical computing and graphics language R. The code base can be found on github https://github.com/wolferl42195/XGBoost4StockPrices.

4.1. Definition of the Case Studies

In order to be suitable as a case study in the current master thesis project, various requirements had to be fulfilled:

- the company is listed on the stock market,
- availability of stock quotes and news data, and
- active public debate that reflects the mood of market participants.

Furthermore, different market sizes and local conditions are taken into account in the present research. Therefore two companies from the US, two companies from Austria and two companies from the UK are selected. In the US the average size of the companies is larger and the media discussion is much more intense than in Europe, which likely affects the quality of the prediction. In contrast, the Austrian market is smaller than the US market and the public debate is not so intense. The UK case studies are interesting because the Brexit debate may influence global and local acting companies. This example is used to analyze effects of the market to these stock companies. Finally, the cryptocurrency Bitcoin is used as an example of a relatively new currency and a highly controversial topic. Salisu, Isah, and Akanni (2018) focused on the daily returns of Bitcoin and provides in- and out-of-sample forecasts. They conclude that in periods with high trading volume the predictive results are better. These considerations lead to the following list:

Tesla Inc.: is a US company that manufactures and sells electric cars, power storage and photovoltaic systems (Ticker symbol TSLA on the Nasdaq stock market).

Apple Inc.: is a US high tech company developing hardware and software and is the most valuable company in the world as of the end of 2018 (Statista (2019)) (Ticker symbol AAPL on the Nasdaq stock market).

Erste Group Bank AG: is the first Austrian savings bank, founded in 1819 in association form and the oldest existing bank in Austria (Ticker Symbol EBS.VI on the Vienna Stock Exchange).

OMV: is an Austrian integrated oil and gas company with upstream and downstream activities (Ticker symbol OMV on the Vienna Stock Exchange).

HSBC Holdings plc: is an international British bank based in London and the largest European bank and ranked by Forbes as the world's 9th largest bank (Ticker symbol HSBA.L on LSE).

Royal Dutch Shell plc: is one of the world's largest oil and gas companies and is a Dutch company headquartered in London and listed at the LSE (Ticker symbol RDSB.L on LSE).

Bitcoin: is the first digital currency based on a decentralized technology Blockchain.

The timeline for the analysis was defined as the period between *1st September 2018 until 31st March 2019*, whereby the time series data is loaded with a lead time to have all technical indicators calculated with the beginning of September 2018.

To simplify the model for a first implementation, all data is aggregated on a daily basis. This means that the model works with end of day quotes (closing prices) and all news events are aggregated to the day they are published. This simplifies the approach because quotes and news do not need to be synchronized by the time of day. In comparison, Beckmann (2017) uses an approach where real-time market data and real-time news data are synchronized. Schumaker and Chen (2009b) showed an approach where the prediction is calculated 20 minutes after the news article was released.

To evaluate the performance of the prediction three scenarios are defined (see table 4.1). The first scenario uses only the historic time series to predict if the ROC for tomorrow is positive or negative (rise or fall). Hence this model works with one feature only. The second scenario still uses only market data but adds the technical indicators as features. These are calculated based on the closing price (see chapter 2) and the volume (only for formula 2.10). Finally, the third model uses the sentiment analysis from the news export of webLyzard in addition to the market data including technical indicators.

Scenario	Features used
Scenario 1	Market data
Scenario 2	Market data, Technical indicators
Scenario 3	Market data, Technical indicators, News and social media data

Table 4.1.: Definition of three scenarios

4.2. Prototype

For the implementation of the prototype the statistical programming language R ("The R-Project", 2019) is applied. R is one of the most popular programming environments for ML because it offers numerous free available packages which can easily be integrated and offer various possibilities to analyze the results. In the scientific community R and Python are the most common tools to implement ML approaches. Many commercial tools implement interfaces to R to use the strength of the analysis methods. The prototype uses several R-packages. The complete list of the packages with explicit version is listed in the appendix (see section Prototype Setup in the appendix).

caret: is package (short for Classification And REgression Training) with a set of functions that attempt to streamline the process for creating predictive models (see http://topepo.github.io/caret/index.html).

DiagrammeR: is a package used by XGboostExplainer to draw decision trees (see http://rich-iannone.github.io/DiagrammeR/index.html)

dplyr: offers data specific data structures called tibble for data organization and transformation (see https://dplyr.tidyverse.org/).

e1071: is a package for class analysis, short time Fourier transformation, fuzzy clustering, support vector machines, shortest path computation, bagged clustering, Naive Bayes classifier (see https://www.rdocumentation.org/packages/e1071/versions/1.7-1).

quantmod: offers access to market data providers to load time series data (see https://www.quantmod.com/).

PerformanceAnalytics: offers functions for analyzing portfolio performance (see https://github.com/braverock/PerformanceAnalytics).

ROCR: is a package for performance measures and visualization (see https://rocr.bioinf.mpi-sb.mpg.de/)

tidyquant: is a collection of R packages for business financial analysis (see https://business-science.github.io/tidyquant/).

tidyverse: is a collection of R packages for data science, with the aim to ensure the data is *tidy*, like each variable has its own column, each observation is a row and each value is a cell (see https://tidyr.tidyverse.org/)

xgboost: is a package that includes efficient linear model solver and tree learning algorithms including parallel computation on a single machine. It offers functions for regression, classification and ranking (see https://xgboost.readthedocs.io/en/latest/R-package/index.html). The R package xgboost won the 2016 John M. Chambers Statistical Software Award.

xts: stands for eXtensible time series and is a package used to work with time series data (see http://joshuaulrich.github.io/xts/).

xgboostExplainer: is a package that allows the predictions from an XGBoost model (see https://github.com/AppliedDataSciencePartners/xgboostExplainer) to be split into the impact of each feature, making the model as transparent as a linear regression or decision tree.

4.3. Processing

Figure 4.1 shows the process implemented in the prototype. For the prototype implementation, market and news data were downloaded as excel or csv (Comma Separated Values) files. This is done for two reasons. First, the results are reproducible and the development can also be done offline. Second, the market data providers and the webLyzard platform offer an Application Programming Interface (API) that can be used to load the data directly into a productive system.

In the next sections, the most relevant parts of the processing pipeline are explained with reference to the particular code base.

4.3.1. Load Market Data

The market data for the seven case studies was downloaded as csv file from the “Finance Yahoo Platform” (2019). Yahoo provides a free download of near-real-time data and a manual download of historic time series as csv. The resulting data is equivalent in structure, hence this part can easily be changed to an API call. The code is shown in listing 4.1. The output of this code section is a list of closing prices and volumes per day.

In the first line a constant (`constant <-42`)[1] is defined which will be used for the `seed` function. This makes the results reproducible because random functions always generate.

[1] 42 was chosen because it is “The answer to life, the universe and everything" Adams, 1980

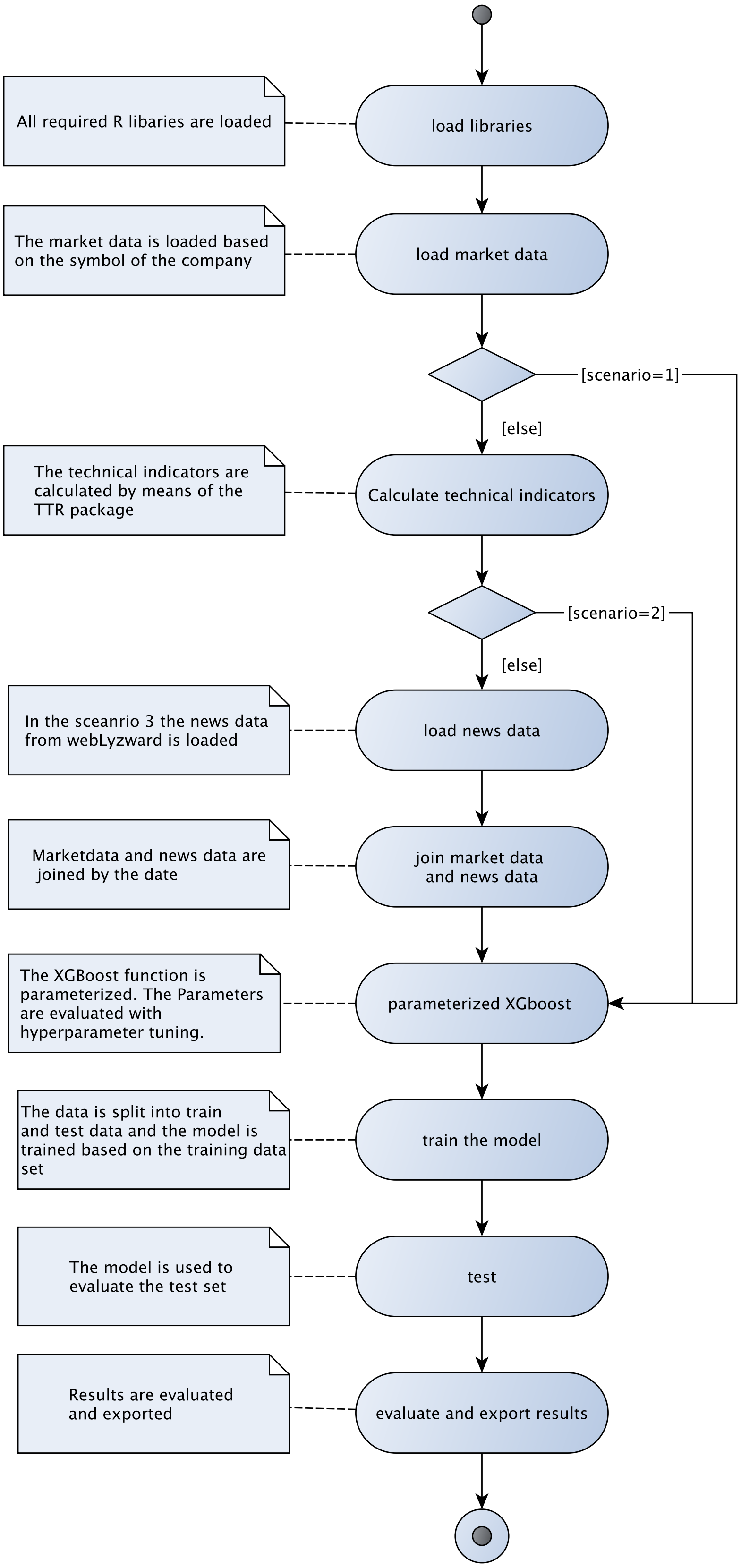

All required R libaries are loaded
load libraries
The market data is loaded based on the symbol of the company
load market data
[scenario=1]
[else]
The technical indicators are calculated by means of the TTR package
Calculate technical indicators
[scenario=2]
[else]
In the sceanrio 3 the news data from webLyzward is loaded
load news data
Marketdata and news data are joined by the date
join market data and news data
The XGBoost function is parameterized. The Parameters are evaluated with hyperparameter tuning.
parameterized XGboost
The data is split into train and test data and the model is trained based on the training data set
train the model
The model is used to evaluate the test set
test
Results are evaluated and exported
evaluate and export results

Figure 4.1.: Processing steps of the calculation for the prediction model

```
constant <- 42
DATA_DIR   = "data/"

symbol = "TSLA"

#load market data
filename <- paste(DATA_DIR, symbol, ".csv", sep = '')
quotes  = read.zoo(file=filename,header=TRUE,  sep = ",") %>% as.xts

# prepare marketdata
closePrices <- Cl(quotes)
volumes     <- Vo(quotes)
```

Listing 4.1: Code Snippet “loading market data”

4.3.2. Calculate Technical Indicators

The technical indicators are calculated by using the TTR package. Most of the indicators have a corresponding function, which has to be parameterized. The length of the period `n` is defined as `n = 14` days. All calculations are based on closing prices (end of day prices).

The technical indicators calculated are based on the literature review (see section 2.3). In total eleven indicators are calculated.

```
# calcuulate technical indicators
set.seed(constant)
ret  <- Return.calculate(closePrices)
roc  <- ROC(closePrices)              # ROC Rate of Change
sma  <- SMA(closePrices, nDays)       # SMA Simple Moving Average
rsi  <- RSI(closePrices, nDays)       # RSI Relative Strength Index
wpr  <- WPR(closePrices, nDays)       # WPR William's % R.
mom  <- momentum(closePrices, nDays) # MOM Momentum
cci  <- CCI(closePrices, n=nDays)     # CCI Commodity Change Index
obv  <- OBV(closePrices, volumes)     # OBV On Balance Volume
dpo  <- DPO(closePrices, n=nDays)     # DPO De-Trended Price Oscillator
adx  <- ADX(HLC(quotes), n=nDays)     # ADX Average Directional Movment
smi  <- SMI(closePrices,              # Stocastik %K %D
            n=nDays,
            nFast=12,
            nSlow=26,
            nSig=9,
            maType=SMA)

macd <- MACD(closePrices,             # MACD Moving Average Convergence/Divergence
             nFast=12,
             nSlow=26,
             nSig=9,
             maType=SMA)
```

Listing 4.2: Code Snippet “Calculation technical indicators”

Not all of these indicators have been used in the final model. During the evaluation it turned out that some have a more significant influence than others. A correlation matrix of the technical indicators feature together with the news features outlines this later.

4.3.2.1. Load News Data

The news data is exported from webLyzard as xlsx file with the news statistics aggregated on a daily basis. The query used to select the news data is a combination of the company name and the stock market ticker. It turned out that filtering only on the ticker leads on the one hand to more stock related information, but on the other hand to a low amount of news. In contrast to the market data, which is only available on dates when the respective stock market is open, news are collected every day. For every news report a sentiment analysis is performed and a value between -1 and +1 is assigned, which:

- -1 negative message,
- +1 positive message
- 0 for neutral message

For every category and day the statistics of total, mean, and standard deviation are calculated by webLyzard (see table 4.2), which leads to nine features in total for the news.

Feature	Description
total	sum of all messages occur
mean	Average
stdDev	Standard deviation, how polarized the reporting is
neutral	Proportion of messages recognized as neutral
pos	Proportion of messages recognized as positive
neg	Proportion of messages recognized as negative

Table 4.2.: Features of the news data provided by webLyzard

The data is converted into csv format and joined to the previously created technical indicator features list by the date (see listing 4.3).

```
#load news data
filename <- paste(DATA_DIR, symbol,"-news",".csv", sep = '')
news = read_csv(file = filename)
news = news %>% mutate(Date = dmy(Date))

#merge market data, technical indicators with news
data = quote_data %>% left_join(news, by = c("qDate" = "Date"))
```

Listing 4.3: Code Snippet “Load news data”

The import file has the following format:

Date	positive Total	positive Mean Sentiment	positive Sentiment StdDev	negative Total	negative Mean Sentiment	negative Sentiment StdDev	neutral Total	neutral Mean Sentiment	neutral Sentiment StdDev
01.09.18	2.00	0.20	0.31	3.00	-0.10	0.32	3.00	-0.03	0.35
02.09.18	7.00	0.16	0.35	2.00	-0.10	0.25	0.00	0.00	0.00
03.09.18	32.00	0.13	0.29	9.00	-0.16	0.28	2.00	0.02	0.12
04.09.18	83.00	0.17	0.26	19.00	-0.13	0.28	4.00	0.00	0.24
05.09.18	64.00	0.21	0.25	43.00	-0.20	0.31	7.00	-0.01	0.25
06.09.18	43.00	0.17	0.26	30.00	-0.16	0.33	1.00	-0.03	0.23
07.09.18	109.00	0.12	0.28	157.00	-0.15	0.31	26.00	0.00	0.25
08.09.18	44.00	0.09	0.25	38.00	-0.10	0.33	14.00	0.01	0.24
09.09.18	24.00	0.16	0.29	11.00	-0.09	0.27	6.00	0.01	0.28

Table 4.3.: Example for the news data file, exported from webLyzard

At the end of this step the market data and the news data are joined together (only for the scenario 3). Due to the fact that market data (except BTC) is only available on days when the stock market open, weekend days and public holidays, as well as any news occuring on those days, are lost in this process. This may could effect the model performance because if crucial events (either positive or negative) occur on these days this information is not reflected in the data set (see section Future Research).

4.3.2.2. Feature Selection and Labeling

XGBoost is a classification model that required data labeling. This label is also the base for the prediction. The model uses positive or negative daily return (ROC) as labels. If a return is negative the data row is labeled with 0 if the return is 0 or higher the data row is labeled with 1 (see algorithm 3.1). The code for the labeling is shown in listing 4.4.

```
# labeling
return_lead       = lead(data$return, n=1)              # shift by 1 day
sign_return_lead = sign(return_lead)
lead_target       = if_else(sign_return_lead==1, 1, 0)

data <- data %>% mutate(return_lead, sign_return_lead, lead_target)
data <- data %>% na.omit
```

Listing 4.4: Code Snippet “Labeling”

At this point, all features, depending on the scenario and label are in one data set. A correlation visualizes the relation between the particular features (see figure 4.2). Obviously the market data features have a high positive or negative correlation to the other market data features. The news and the market data, on the other hand, have only low correlation which means that additional independent information is added to the model by using the news features. The final list of features for all case studies and scenarios is summarized in table 4.4.

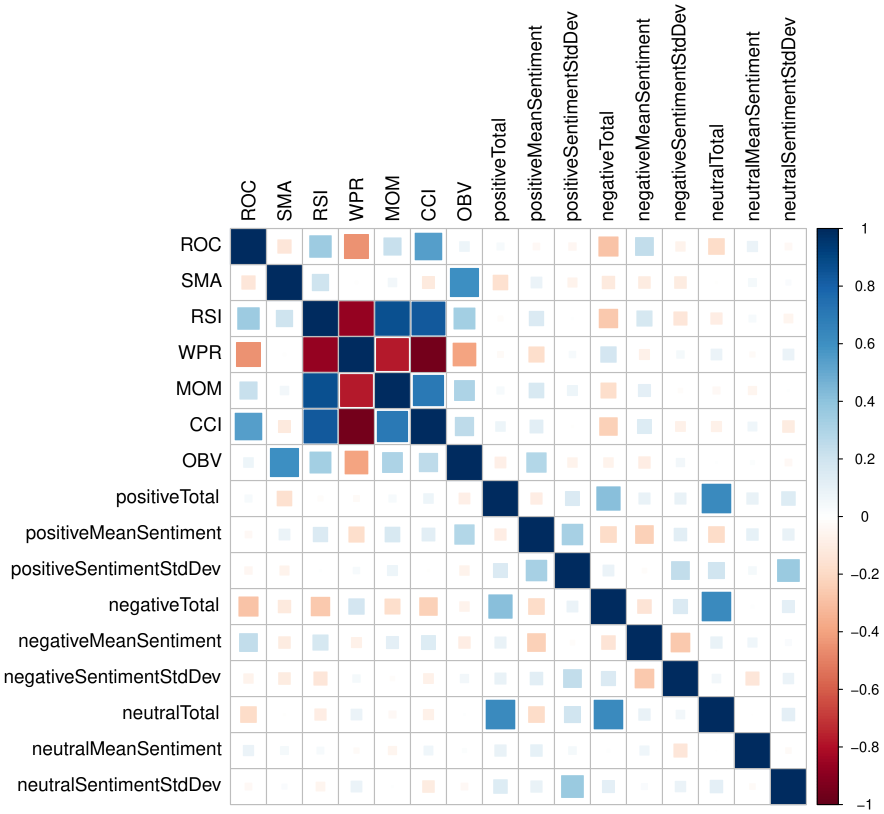


Figure 4.2.: Correlation of the features for the Tesla case study

Feature	Description	source
f0	ROC	market data
f1	SMA	market data
f2	RSI	market data
f3	WPR	market data
f4	MOM	market data
f5	CCI	market data
f6	OBV	market data
f7	positiveTotal	news data
f8	positiveMeanSentiment	news data
f9	positiveSentimentStdDev	news data
f10	negativeTotal	news data
f11	negativeMeanSentiment	news data
f12	negativeSentimentStdDev	news data
f13	neutralTotal	news data
f14	neutralMeanSentiment	news data
f15	neutralSentimentStdDev	news data
f16	lead_target (label)	news data

Table 4.4.: Final feature list used in the model

4.3.3. XGBoost

XGBoost uses several parameters which have to be adapted to the specific model. In this case the special situation is that the data is time series based, which means the order has an impact on the prediction. To find the right parameter a script was used which varies the values and then plots the results. This approach is called hyperparameter tuning.

4.3.3.1. Hyperparameter Tuning

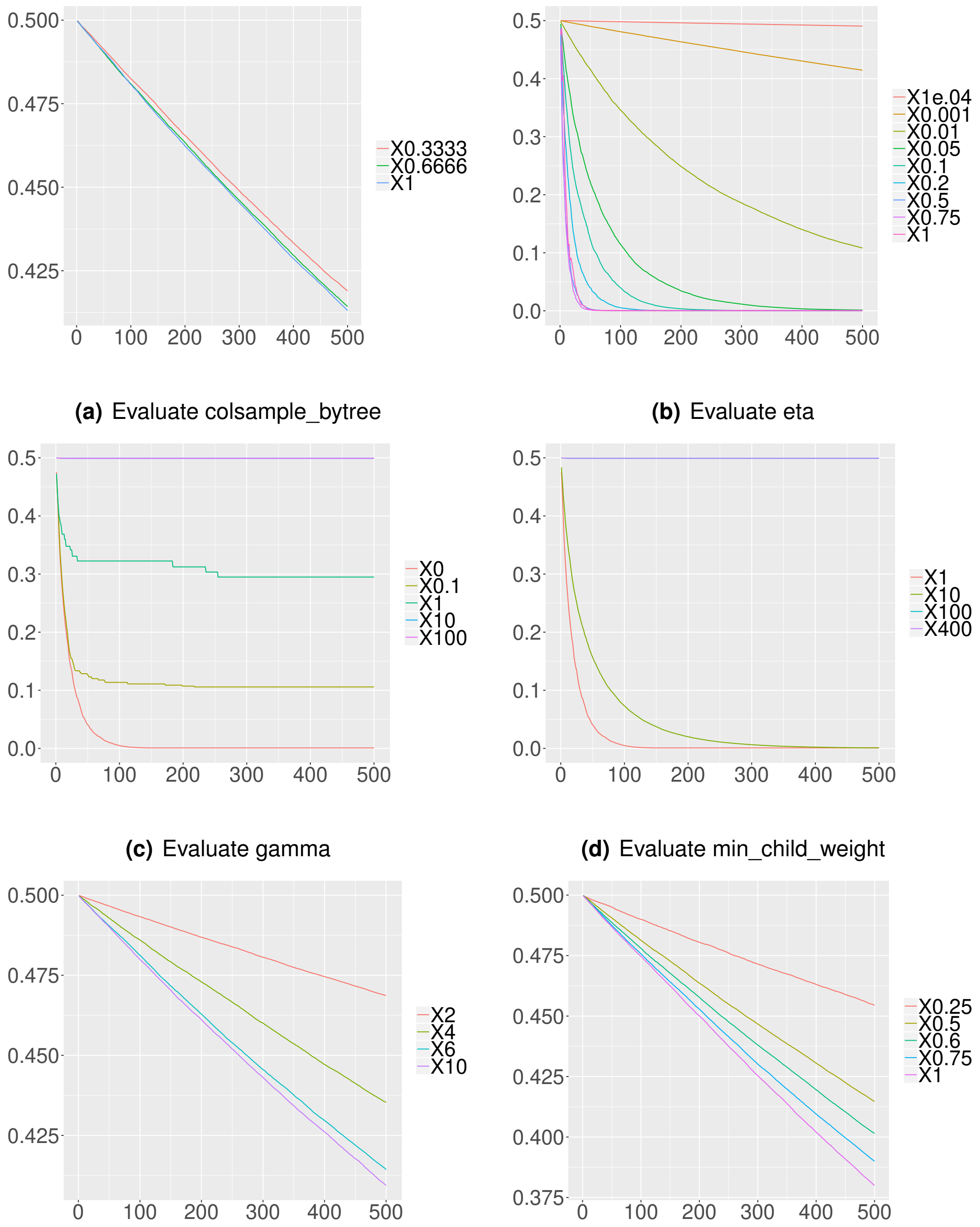
Hyperparameter Tuning was applied for a total of six parameters. The results are shown in figure 4.3. For the parameter `colsample_bytree` figure 4.3(a) shows the difference between all curves is marginal. Hence this parameter is set to the default value 1. An important parameter is `eta`, the learning rate. The resulting plot (see figure 4.3(b)) shows that if the value for `eta` is low (0.0001) the resulting value stays high over all iterations. The same applies for the `gamma` (see figure 4.3(c)), hence `gamma` is set to 0. The parameter `max_depth_canditates` is set to the default value of 1. The curve indicates for a low number of trees similar results. Next the parameter `min_child_weights` is also set to 1 the because the rate is constant independent of the iterations. And finally the `sub_sample_candidates` is set to 0.6. (see figure 4.3(c)). Even though the tuning algorithm would suggest a lower value (e.g. 0.25), tests show better results with the value 0.6. Table 4.5 summarizes the parameters and listing 4.6 shows the parameterization of the R-script.

Parameter	Value
`colsample_bytree`	1
`eta`	0.0001
`gamma`	0.0
`min_child_weight`	1
`max_depth`	7
`subsample`	0.6

Table 4.5.: Final XGBoost parameters used for all scenarios and case studies

```
tree.params = list(
  booster          = "gbtree",
  eta              = eta,
  max_depth        = max_depth,
  min_child_weight = 1,
  subsample        = 0.6,
  colsample_bytree = 1,
  gamma            = 0.0,
  objective        = "binary:logistic")
```

Listing 4.5: Code Snippet “Parameterizing the XGBoost tree”



(a) Evaluate colsample_bytree

(b) Evaluate eta

(c) Evaluate gamma

(d) Evaluate min_child_weight

(e) Evaluate max_depth

(f) Evaluate subsample

Figure 4.3.: Results of the Hyperparameter tuning based on the Tesla features (x axis iterations and y axis the corresponding value)

4.3.3.2. Splitting Dataset

The variable `data` contains the features and has to be split into a training and a test. The split factor was set to 75%. The split factor can influence the result, as more training data can either lead to a better prediction or to oversampling. While the selected factor is very common, future research could experiment with this value in order to further optimize the model. The code fragment for the data splitting is shown in listing 4.6.

```
total_n      =  data %>% nrow()
length_train =  floor(total_n * 0.75)
length_test  =  total_n - length_train

train_data = data %>% top_n( -length_train, qDate)
test_data  = data %>% top_n(  length_test,  qDate)

var_ex = c("qDate", "Price", "return", "return_lead", "sign_return_lead", "lead_target")
X_train <- train_data %>% select(-one_of(var_ex)) %>% as.matrix()
Y_train <- train_data$lead_target

X_test  <- test_data %>% select(-one_of(var_ex)) %>% as.matrix()
Y_test  <- test_data$lead_target
```

Listing 4.6: Code Snippet “Splitting data into training and test data sets”

4.3.3.3. Train and Test

XGBoost defines a variable of the type `DMatrix`, which has to be initialized with the training data, whereby `X_train` are all the features and `Y_train` is the column with the label information. The same structure is needed for the test data. First the tree parameters have to be defined (see listing 4.5). Additional to the parameters evaluated in the section before two further parameters have to be set. The type of the tree is set to `booster = gbtree` and the `objective` is set to `objective = binary:logistic` because the tree is used to solve a binary problem.

The train and test data is now used to generate the XGBoost `DMatrix` object.

```
# xgboost
set.seed(constant)
xgb.train.data <- xgb.DMatrix(data = X_train, label = Y_train)
xgb.test.data  <- xgb.DMatrix(data = X_test , label = Y_test )
```

Listing 4.7: Code Snippet “Generate the XGBoost DMatrix objects”

These two variables are the input for the training and the prediction function. First the model is trained with the DMatrix (function train) of the training data (`xgb.train.data`) and then the function predict is used on the training and test set. This allows one to see the prediction in-sample (training data) and out-of-sample (test data).

```
xgb.model.tree = xgb.train(
  data          = xgb.train.data,
  weight        = NULL,
  watchlist     = watchlist,
  params        = tree.params,
  nrounds       = nrounds,
  verbose       = 2,
  print_every_n = 10L,
  eval_metric   = eval_metric,
  maximize      = TRUE,
  early_stopping_rounds = 10)
train_preds = predict(xgb.model.tree, X_train)
test_preds  = predict(xgb.model.tree, X_test)
```

Listing 4.8: Code Snippet “Process Prediction”

4.3.3.4. Cross-Validation

Cross-validation is used to optimize the prediction. As outlined in figure 3.5, a part of the training set is used for cross-validation, which is a separate function in the XGBoost package. The main parameter for the cross-validation is `nFolds`, which splits the train data into `nFolds` equal size subsamples. One of the nFold subsamples is used for the validation of the model and the remaining `nFolds` are used for training. This is done `nrounds` with each subsample. The listing for the cross-validation including the plot of ROC and AUC is shown in listing 4.9.

```
# cv cross validation
xgb_params    = list(
  objective   = "binary:logistic",
  eta         = eta,
  max.depth   = max_depth,
  eval_metric = eval_metric)

xgb_cv = xgb.cv(
  params        = xgb_params,
  data          = X_train,
  label         = Y_train,
  nrounds       = nrounds,
  prediction    = TRUE,
  nfold         = 3,
  print_every_n = 1,
  early_stopping_rounds = 10)

plot(pROC::roc(response = Y_train, predictor = xgb_cv$pred,levels=c(0, 1)), lwd=1.5)

print(xgb_cv$evaluation_log[which.min(xgb_cv$evaluation_log$test_auc_mean)])
nrounds <- xgb_cv$best_iteration
```

Listing 4.9: Code Snippet “Cross-Validation”

The resulting curve for the Tesla use case is shown in 4.4. Through the optimization the AUC has been optimized and shows the typical steepness character.

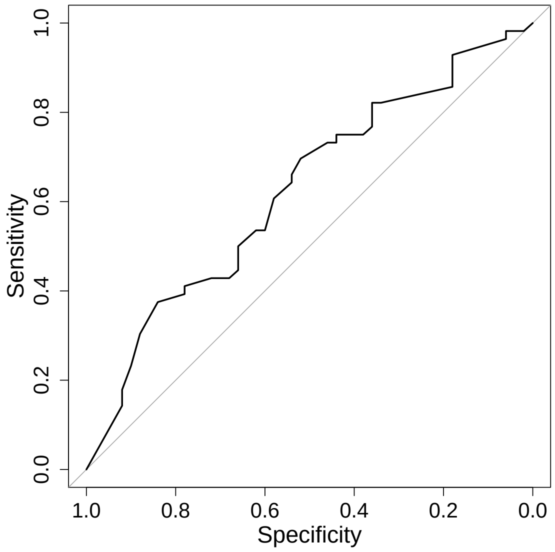


Figure 4.4.: Cross-Validation for the Tesla case

4.3.4. Data Evaluation

The results are evaluated mainly based on the confusion matrix. This matrix is generated for both the training (in-sample) and for the test data (out-of-sample). These results are used to see how well the models are learning and what differences exist between the scenarios. The code fragment for the generation of the Confusion Matrix is in listing 4.10.

```
#evaluate results
confMatrixTest  <- caret::confusionMatrix(Y_test  %>% as.factor, ifelse(test_preds  >
    0.5, 1, 0) %>% as.factor)
confMatrixTrain <- caret::confusionMatrix(Y_train %>% as.factor, ifelse(train_preds >
    0.5, 1, 0) %>% as.factor)
```

Listing 4.10: Code Snippet “Generate Confusion Matrix”

5. Results and Discussion

“Success is a science; if you have the conditions, you get the result.”

Oscar Wilde

This chapter discusses the results of the case studies defined in chapter 4. First a general overview of the data and a comparison between the case studies and scenarios is made. Afterwards the results of each case study are discussed in detail. For the first case study (Tesla) all graphics are presented in this chapter, whereas for all other case studies the results are found in the appendix and only a summary is presented here.

5.1. Market Data Overview

In table 5.1 an overview of the market data is shown. All market data time series starts with the 1st of September, 2018. The differences in numbers of data points result from differing numbers of holidays in the various countries. Bitcoin, on the other hand, does not belong to a country and has quotes even on the weekends. The columns ‘neg’ and ‘pos’ show the counts of the label in the time period. These columns do not differ markedly for any of the case studies, which means that there is an approximate balance between the number of days with positive and negative returns.

Case Study	Region	high	low	neg	pos	Start	days count
TSLA	US	379.57	250.56	69	73	01.09.18	142
APPL	US	232.07	142.19	67	76	01.09.18	142
BTC	Global	8,395.82	3,232.51	104	105	01.09.18	209
OMV	AT	51.00	37.65	72	72	01.09.18	143
EBS	AT	37.76	28.10	75	68	01.09.18	143
RDSB.L	UK	2,780.50	2,227.00	72	75	01.09.18	147
HSBA.L	UK	734.60	600.80	75	72	01.09.18	147

Table 5.1.: Overview of the market data for all case studies

5.2. News Data Overview

In table 5.2 an overview of the news data is provided. It is important to notice that the absolute values differ a lot, which depends on the news sources uses by webLyzard. During the data gathering process additional data sources were added and webLyzard was reconfigured for the UK cases. That is the reason why the period for news data for RDSB.L and HSBA.L starts only on the 1st February 2019. New sources were also added for BTC, such that much more data became available for BTC starting on the 1st February 2019. The comparison shows that Bitcoin is highly reflected in the news, with half of these messages being marked as neutral and only 12.8% as negative.

For the Austrian cases (OMV and Erste Group Bank) the amount of messages is rather low, yet 70.3% of the messages for OMV and almost 75.1% for Erste Group Bank are positive.

In contrast, the UK cases show a higher proportion of neutral messages, while positive and negative messages are roughly equivalent. HSBA is the only case where the proportion of negative messages exceeds the proportion of positive messages.

Case Study	pos Total	neg Total	neutral Total	pos %	neg %	neutral %	max news	Start	days
TSLA	12,044	5,882	1,845	60.9%	29.8%	9.3%	410	01.09.18	211
APPL	12,047	5,034	3,156	59.5%	24.9%	15.6%	701	01.09.18	211
BTC	386,357	142,967	589,566	34.5%	12.8%	52.7%	25.501	01.09.18	211
OMV	5,291	1,282	954	70.3%	17.0%	12.7%	87	01.09.18	211
EBS	2,792	557	369	75.1%	15.0%	9.9%	40	01.09.18	211
RDSB.L	817	743	3,367	16.6%	15.1%	68.3%	639	01.02.19	59
HSBA.L	19,709	26,568	50,434	20.4%	27.5%	52.1%	17.636	01.02.19	59

Table 5.2.: Overview of the news data for all case studies

5.3. Case Studies

The case studies and the results are analyzed in detail to evaluate the hypotheses stated in chapter 1. At the end of this chapter the results of all case studies are summarized.

5.3.1. Tesla

Tesla was the first case used for optimizing the model. Tesla is a very innovative company and all its actions are heavily discussed. First the market data and the news collected are analyzed and afterwards the results are interpreted.

5.3.1.1. Technical Analysis Results,

Figure 5.1 shows the histograms of the technical indicators for Tesla between 1st of September 2018 and 31st March 2019. The advantage of the histogram plot is that it depicts the distribution of the values in the selected time period and highlights extreme values in the data set. Discussion of the technical indicators of Tesla:

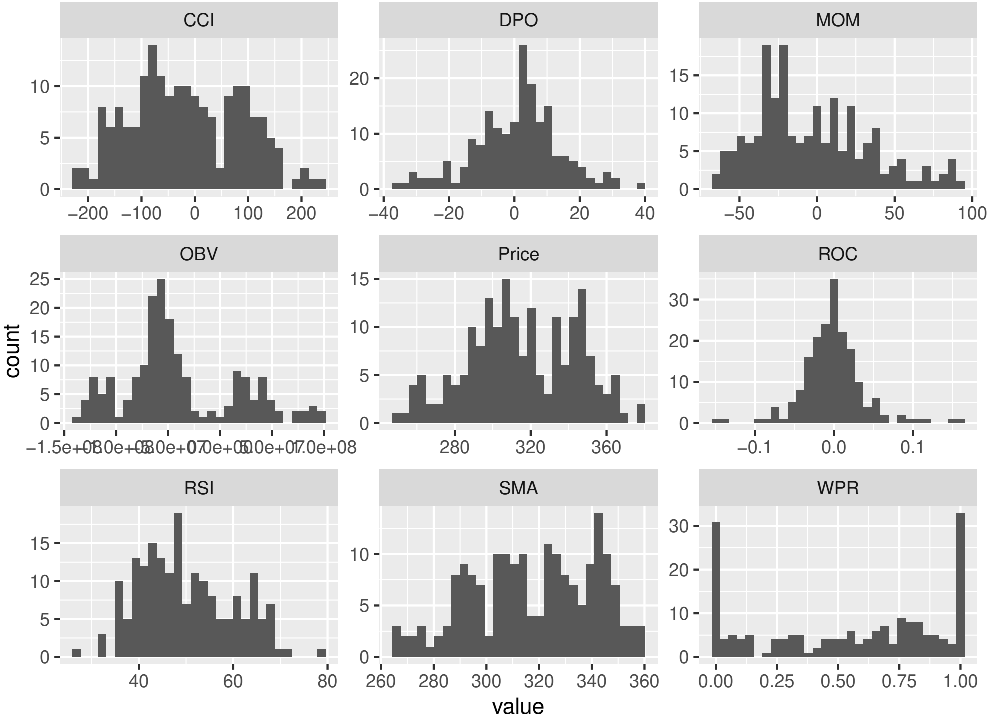


Figure 5.1.: Histogram of the technical indicators of the Tesla case study

CCI: values over 100 indicate a rise and under -100 signal a loss. The chart shows that the trend in the period was more in the direction of loss.

DPO: values are little above zero, which means that the price was more often above the SMA than below.

MOM: Also the momentum shows a distribution left of 0 and would also indicate a loss over the time period.

OBV: indicates value change based on the closing price. The OBV shows no trend for a sell buy signal.

Price: reflects the fluctuation of the quotes whereby the level was always passed quickly by 330.

ROC: is distributed around the 0 with a few outliers positive and negative.

RSI: does not indicate overbought or oversold because the values are mainly between 30 and 70.

SMA: is the result of smoothing out price fluctuation and shows a value tending above 300.

WPR: a value near one indicates oversold, hence WPR indicates that Tesla was more oversold than overbought during the time period.

5.3.1.2. News Analysis Result

Figure 5.2 shows the histograms of the news indicators for Tesla between 1st of September 2018 and 31st March 2019. Discussion of the news features:

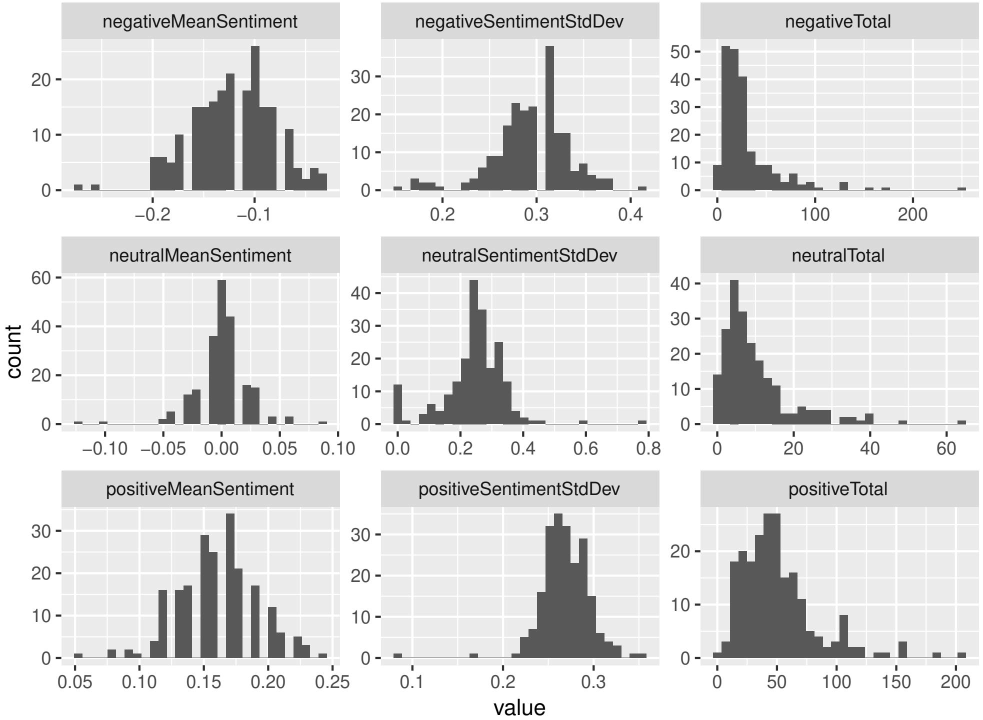


Figure 5.2.: Histogram of the news features of the Tesla case study

negative Total: Most of the days have a quantity of around 30 negative messages. There are six days in total with more than 100 negative messages,

negative Mean Sentiment: Overall the messages are not particularly negative, only three days have a mean below -0.2.

positiv Total: In contrast to the negative messages, the distribution is rather more spread, resulting in a higher average number of positive messages per day.

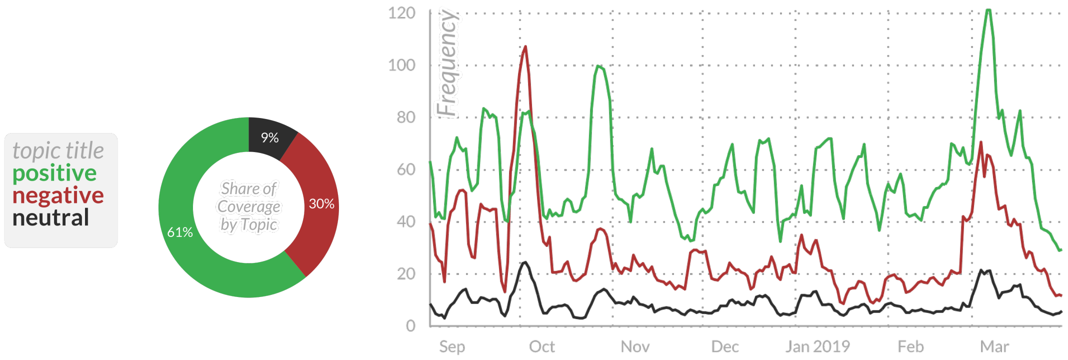
positiv Mean Sentiment Also the mean sentiment low, and only 6 days are above 0.22.

neutral Total: There are only 23 days with more than 20 messages.

neutral Mean Sentiment: For the neutral messages the mean is logically around 0.

The middle column shows the standard deviation and shows how far the individual numbers are distributed. The graphical analysis provided by webLyzard is shown in figure 5.3.1.2.

The chart shows that positive news outweighs negative news, but also that peaks in positive messages tend to be accompanied by peaks in negative messages.



(a) Sentiment distribution

(b) Trend chart

Figure 5.3.: Tesla sentiment distribution and trend chart

5.3.1.3. Predictions Results

In figure 5.4 the confusion matrix for the test data set is shown. In total the data set contains 36 observations. The statistical figures based on this matrix are listed in 5.4

		Actual	
		fall	rise
Prediction	fall	7	12
	rise	1	16
	Total	8	28

Figure 5.4.: Confusion Matrix for Tesla for out-of-sample evaluation

The results show a high value for accuracy on the training data (91.5%) and a also high accuracy on the test data (63.9%) for scenario 3, but scenario 2 reaches exactly the same accuracy. The high accuracy results from the fact that the data set of Tesla was used to optimize the model. Also, the F-Value is lower for the prediction with 51.9% and also a bit lower than for scenario 2. The high F-Value score is reached by a high recall and at the cost of the specificity, which is lower with 57.1% and means that the prediction of negative results is worse than the prediction of positive results. Table 5.4 the feature importance

Scenario	Accuracy	Precision	Recall	Specificity	F Value
		in-sample			
1	69.8%	72.0%	66.7%	73.1%	69.2%
2	86.9%	82.4%	89.4%	85.0%	85.7%
3	91.5%	86.0%	95.6%	88.5%	90.5%
		out-of-sample			
1	55.6%	57.9%	57.9%	52.9%	57.9%
2	63.9%	47.4%	75.0%	58.3%	58.1%
3	63.9%	36.8%	87.5%	57.1%	51.9%

Table 5.3.: Overview of the results of the Tesla case study

for scenario 2 and 3, which reveals which of the features have the greatest impact on the results. Scenario 2 does not include news features, and is therefore dominated by ROC (`gain = 0.24`). Scenario 3 has no such dominating feature, but the addition of news features in this model displaces two of the three most influential technical indicators from scenario 2.

Rank	Feature	Gain	Cover	Frequency
		Scenario 2		
1	ROC	0.24613	0.27292	0.22794
2	RSI	0.18328	0.17287	0.18382
3	OBV	0.14733	0.14667	0.14706
		Scenario 3		
1	OBV	0.14669	0.16568	0.10280
2	positiveTotal	0.12971	0.12204	0.10280
3	neutralSentimentStdDev	0.11662	0.10660	0.12150

Table 5.4.: Feature Importance for scenario 2 and 3 of the Tesla case study

As outlined in section 3.4.2 the AUC is a measurement of the quality of the prediction. For this case study the result is `AUC = 0.6672` and the graphical representation is shown in figure 5.5. An XGBoost tree (a multiple tree) for the Tesla case study is shown in A.5, to visualize

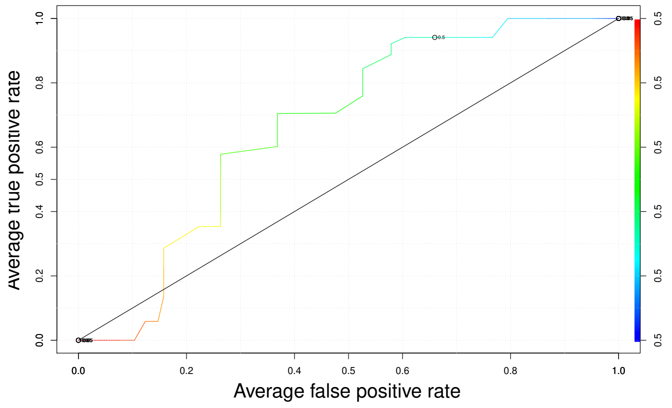
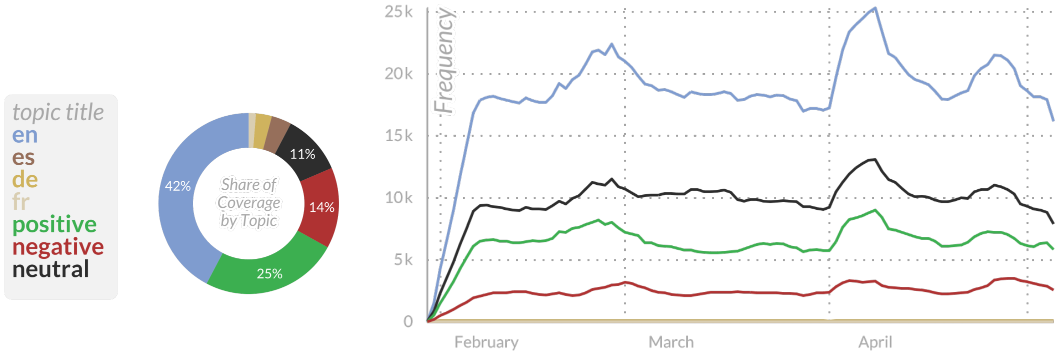


Figure 5.5.: Tesla ROC and AUC visualization

how a decision tree based on the features looks like. For the upcoming case studies only the essential findings are outlined.

5.3.2. Apple

The detailed results for the Apple case study can be found in appendix A.2.2. Apple shows a similar behavior to Tesla in terms of News, where positive news dominates. By comparing sentiment between news and tweets, the tweets show a more neutral character. There are three days where more than 400 positive messages occur but only one day with more than 300 negative message. Among the technical indicators (see figure A.6), WPR reveals an overbought characteristic, RSI shows no trend to locate (values mainly between 30 and 70), and ROC centers around 0 with a few outliers below -0.1.



(a) Sentiment distribution

(b) Trend chart

Figure 5.6.: Apple sentiment distribution and trend chart

The simulation results show an out-of-sample accuracy of 61.1% with an F-Value 50.0%. In contrast to Tesla the F-Value is more balanced and there is no gap between recall and specificity. The accuracy of scenario 2 is lower than that of scenario 1, which means that the addition of technical indicators worsened the predictive ability. Similar to the Tesla case,

Scenario	Accuracy	Precision	Recall	Specificity	F Value
		in-sample			
1	68.9%	69.8%	68.5%	69.2%	69.2%
2	86.0%	88.7%	83.9%	88.2%	86.2%
3	89.6%	92.5%	87.5%	92.0%	89.9%
		out-of-sample			
1	55.6%	57.1%	44.4%	66.7%	50.0%
2	47.2%	50.0%	36.8%	58.8%	42.4%
3	61.1%	50.0%	50.0%	68.2%	50.0%

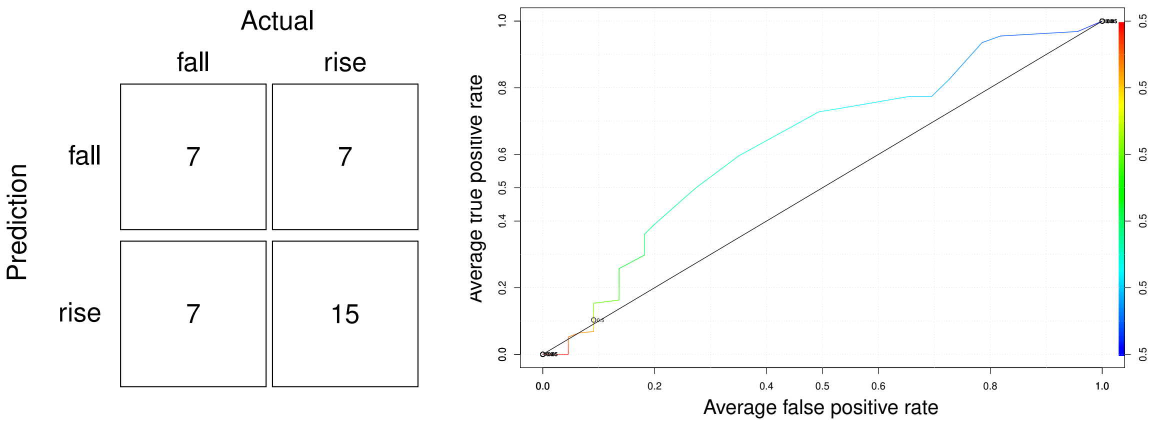
Table 5.5.: Overview of simulation results of the Apple case study

ROC was the dominant feature in scenario 2, while two news features were among the most influential features in scenario 3.

Rank	Feature	Gain	Cover	Frequency
		Scenario 2		
1	ROC	0.20893	0.20915	0.23129
2	OBV	0.18177	0.20776	0.17687
3	SMA	0.18142	0.16988	0.15646
		Scenario 3		
1	ROC	0.10642	0.07754	0.11724
2	neutralSentimentStdDev	0.09792	0.09171	0.09655
3	positiveMeanSentiment	0.09307	0.08878	0.08966

Table 5.6.: Feature Importance ranking for scenario 2 and 3 of the Apple case study

The Area under the Curve metric is `AUC = 0.5747` based on the confusion matrix depicted in figure 5.7. The value of only marginally above 0.5 results from a bad true positive vs. false positive rate.



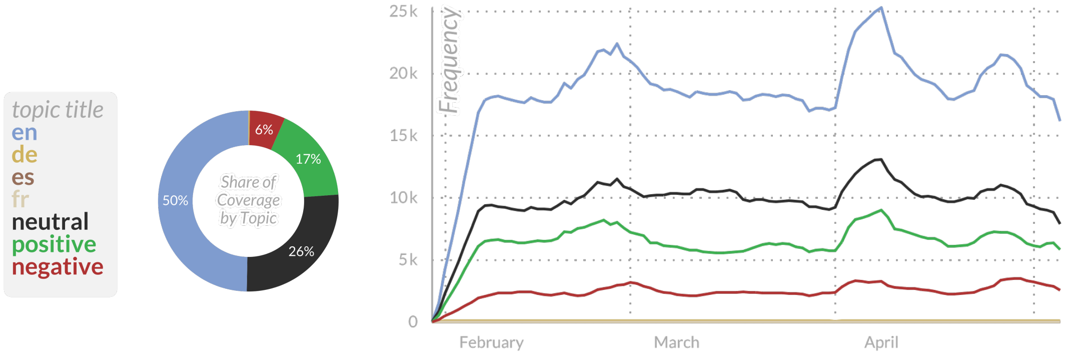
		Actual	
		fall	rise
Prediction	fall	7	7
	rise	7	15

Figure 5.7.: Confusion Matrix and ROC curve of the Apple case study

5.3.3. Bitcoin

The detailed results for the Bitcoin study can be found in appendix A.2.3. Bitcoin serves as an example of a cryptocurrency and is a very controversial topic in the public debate. During the analysis period additional news sources were added to webLyzard, which lead to a domination neutral messages (see 5.3.3). The market data (see histogram A.11) visualizes a downtick. The plot of the prices has two peaks with a gap in between, which is also reflected by the SMA.

The simulation results show a high accuracy in-sample with 91.0% and also a high F-value of 91.7%. However, the out-of-sample prediction was worse for scenario 3 than for the



(a) Sentiment distribution

(b) Trend chart

Figure 5.8.: Bitcoin sentiment distribution and trend chart

other 2 specifications, which was likely caused by the change of news sources. Less news information was incorporated in the training data than the test data, which shows that a change in the sources needs to be done on the whole data otherwise the prediction results are poor. An interesting finding is that scenario 1 outperforms the other two scenarios in this case study. The recall result for scenario 2 and 3 are below 50%, which means that actual "rises" would only be predicted in every second case.

Scenario	Accuracy	Precision	Recall	Specificity	F Value
		in-sample			
1	67.3%	73.2%	67.4%	67.2%	70.2%
2	78.3%	87.8%	75.0%	83.6%	80.9%
3	91.0%	93.9%	89.5%	92.9%	91.7%
		out-of-sample			
1	58.5%	77.3%	50.0%	73.7%	60.7%
2	49.2%	96.7%	49.2%	50.0%	65.2%
3	43.4%	90.9%	41.7%	60.0%	57.1%

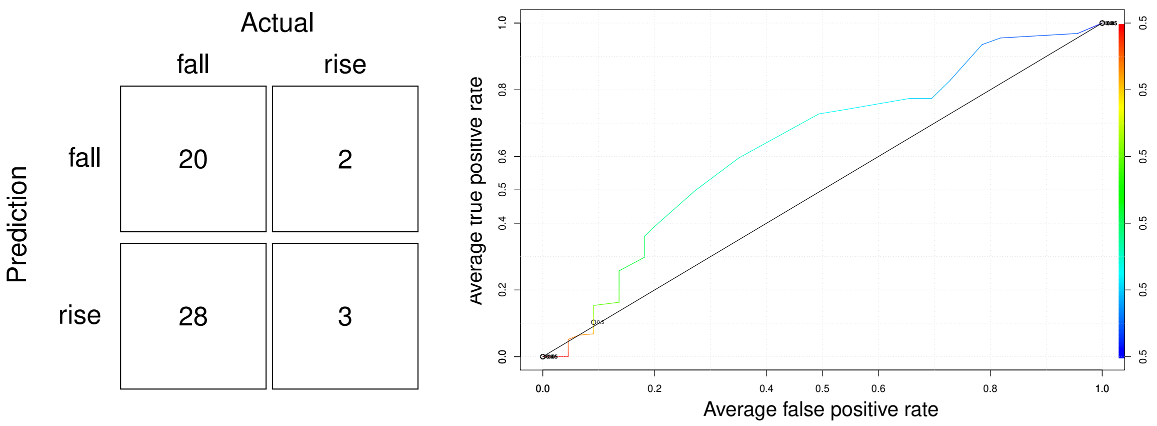
Table 5.7.: Overview of results of the Bitcoin case study

The feature importance shows that only technical figures fall among the three most influential features and these results are similar between scenarios 2 and 3.

The Area Under Curve metric is `AUC = 0.6312` based on the confusion matrix (visualized in figure 5.9). Although the accuracy is worse than in the Apple case study AUC is higher because of the better TPR vs. FPR ratio.

Rank	Feature	Gain	Cover	Frequency
		Scenario 2		
1	OBV	0.26985	0.26290	0.22727
2	ROC	0.23819	0.24190	0.23484
3	RSI	0.17028	0.14004	0.12121
		Scenario 3		
1	OBV	0.16472	0.15304	0.13826
2	RSI	0.12085	0.11896	0.06752
3	ROC	0.11823	0.13064	0.09324

Table 5.8.: Feature Importance ranking for scenario 2 and 3 of the Bitcoin case study

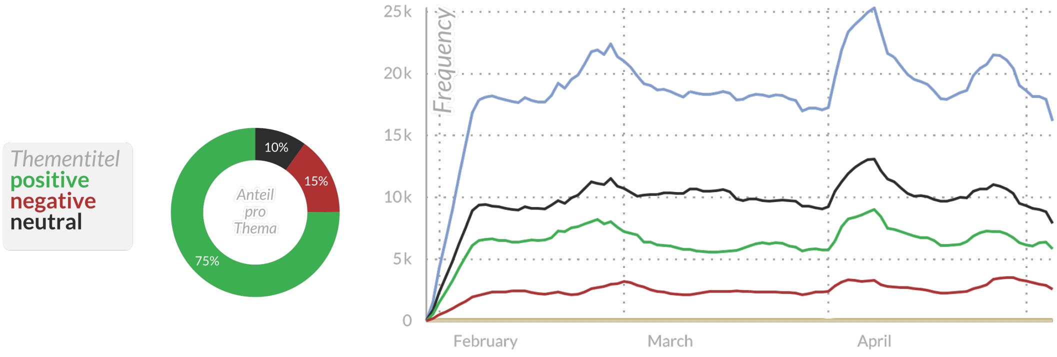


Prediction	Actual: fall	Actual: rise
fall	20	2
rise	28	3

Figure 5.9.: Confusion Matrix and ROC curve for Bitcoin case study

5.3.4. Erste Group Bank

The detailed results for the Erste Group Bank case study can be found in appendix A.2.4. The market data histogram (see figure A.16) reflects the negative price trend. Returns are mainly low between -0.04 and 0.04 and RSI does not indicate overbought or oversold.



(a) Sentiment distribution

(b) Trend chart

Figure 5.10.: Erste Group Bank sentiment distribution and trend chart

On the news side, three-quarters of the messages are identified as positive. This is an effect of the German language and the different sentiment analysis of webLyzard for the German language. There is only one day with more than 20 negative but the other hand several days with positive recognized messages. Further, the positivity of the positive messages exceeds the negativity of the negative ones.

Accuracy in-sample is slightly better in scenario 3 than in scenario 2, whereby scenario 3 has a better F-Value. Out-of-sample scenarios 1 and 2 have the best accuracy which means that the technical indicators do not improve the model. In this case study scenario 3 has the worst out-of-sample accuracy, which is a result of the lower precision and recall.

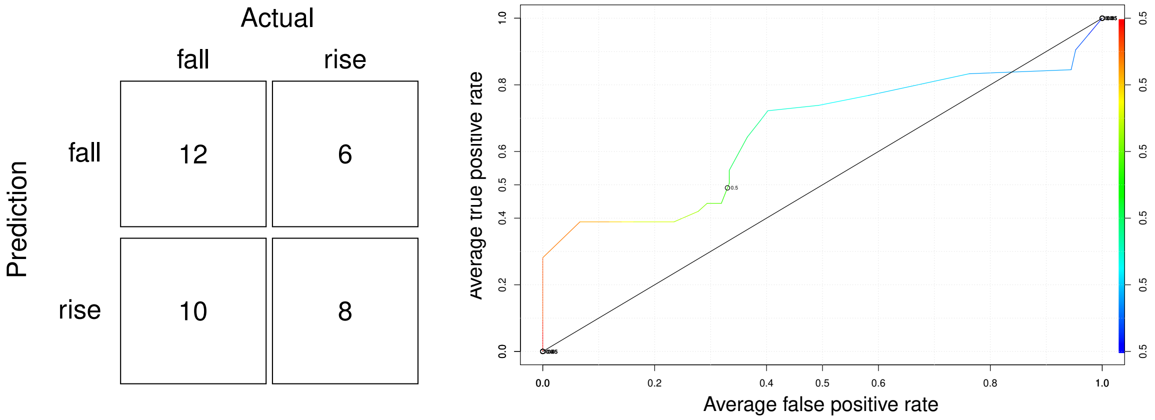
Scenario	Accuracy	Precision	Recall	Specificity	F Value
		in-sample			
1	65.1%	87.7%	62.5%	73.1%	73.0%
2	81.3%	82.5%	82.5%	80.0%	82.5%
3	82.1%	89.5%	79.7%	85.7%	84.3%
		out-of-sample			
1	63.9%	88.9%	59.3%	77.8%	71.1%
2	63.9%	83.3%	60.0%	72.7%	69.8%
3	55.6%	66.7%	54.5%	57.1%	60.0%

The feature list for scenario 3 ranks `negativeSentimentStdtDev` with the highest importance (`gain = 0.235`). Again the ROC is in both scenarios a feature with high relevance.

Rank	Feature	Gain	Cover	Frequency
		Scenario 2		
1	ROC	0.19986	0.194954	0.19
2	MOM	0.18279	0.194595	0.18
3	RSI	0.15558	0.168288	0.16
		Scenario 3		
1	negativeSentimentStdDev	0.23022	0.23507	0.13187
2	ROC	0.09739	0.08634	0.07692
3	OBV	0.08415	0.06847	0.07692

Table 5.9.: Feature Importance ranking for scenario 2 and 3 of the Erste Group Bank case study

The Area Under Curve metric is `AUC = 0.7143` based on the confusion matrix is visualized in figure 5.11. Although the scenario 3 is only third-ranked, the resulting AUC shows an acceptable rate for AUC.

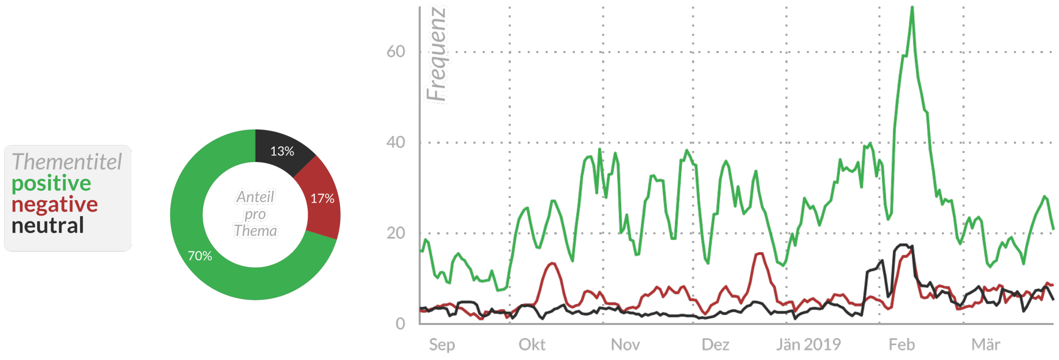


		Actual	
		fall	rise
Prediction	fall	12	6
	rise	10	8

Figure 5.11.: Confusion Matrix and ROC curve for Erste Group Bank case study

5.3.5. OMV

The detailed results for the OMV case study can be found in appendix A.2.5. The chart for OMV has the same appearance as Erste Group Bank. The returns have a light trend to low negative but the returns are lower than in the previous example. RSI is also between 30 and 70, which means no clear buy or sell signal. The news characteristic shows a majority of positive news (70%) and an approximate balance between neutral and negative news. There are no days with more than 30 negative news items, but there are days with more than 100 positive news messages. The sentiment for the negative is primarily above -0.2 and positive messages are on some days over 0.2, which indicates that messages are more strongly positive than they are negative.



(a) Sentiment distribution

(b) Trend chart

Figure 5.12.: OMV sentiment distribution and trend chart

For OMV the accuracy of scenario 3 is a bit better than scenario 2 and also the F-value. Out-of-sample, scenario 2 performs better in accuracy and F-Value (54.5% vs 50.0%). Again, the weakness of scenario 3 was because of the low recall value.

Scenario	Accuracy	Precision	Recall	Specificity	F Value
		in-sample			
1	71.7%	81.0%	71.2%	72.5%	75.8%
2	81.3%	82.8%	82.8%	79.6%	82.8%
3	82.1%	86.2%	82.0%	82.2%	84.0%
		out-of-sample			
1	47.2%	71.4%	40.0%	63.6%	51.3%
2	58.3%	64.3%	47.4%	70.6%	54.5%
3	55.6%	57.1%	44.4%	66.7%	50.0%

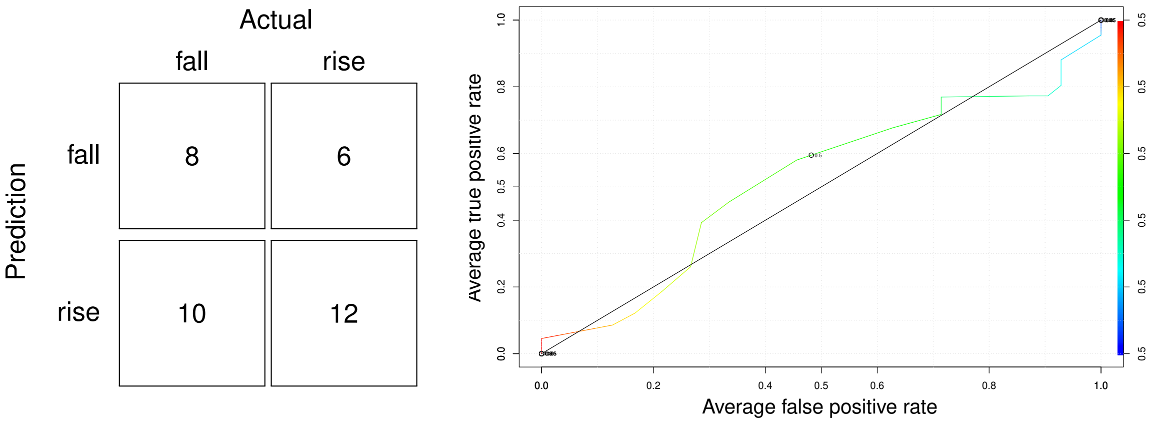
Table 5.10.: Overview of the results of the OMV case study

The `positiveMeanSentiment` is a news feature which was the second most influential feature in scenario 3. The importance feature list ranked ROC first in both scenarios.

Rank	Feature	Gain	Cover	Frequency
		Scenario 2		
1	ROC	0.38336	0.37027	0.30370
2	OBV	0.13842	0.15588	0.14814
3	MOM	0.13504	0.14343	0.16296
		Scenario 3		
1	ROC	0.11177	0.10391	0.096774
2	positiveMeanSentiment	0.10167	0.09920	0.096774
3	MOM	0.10021	0.093772	0.10753

Table 5.11.: Feature Importance ranking for scenario 2 and 3 of the OMV case study

The Area Under Curve metric is `AUC = 0.5244` based on the confusion matrix visualized in figure 5.13. The value is low and close to the near random line (dotted line).

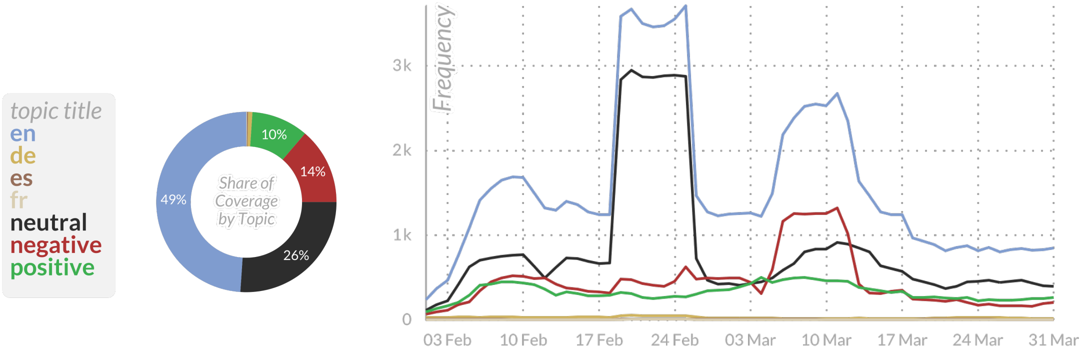


		Actual	
		fall	rise
Prediction	fall	8	6
	rise	10	12

Figure 5.13.: Confusion Matrix and ROC curve for OMV case study

5.3.6. HSBC Holdings

The detailed results for the HSBC case study can be found in appendix A.2.6. The market data characteristics show an oversold characteristic according to R% and RSI. The returns are evenly distributed between -0.25 and 0.25. The chart for the closing price shows a steady gradient during the whole period. In contrast to the other case studies the news period starts on the 1st of February. The chart shows to shorter time periods where neutral first and negative message afterwards rise unproportional. In this case the neutral message overweight, which is effected by social media messages. The results in this case can only be compared partly, because scenario 1 and 2 used the complete time series of the stock quotes and technical indicators, respectively. In-sample the performance is high with an accuracy of 86.7% and an F-value is 81.8%, which is a bit below scenario 2. Out-of-sample the result for accuracy is low with 40% and an also low F-value. This can be explained by the shorter time period and the two peaks in the training data, which negatively influence the quality of the results. Interesting is the rate of 100% specificity, which means that all positive results were detected correctly. The feature importance for scenario 3 ranks the two statistical features for



(a) Sentiment distribution

(b) Trend chart

Figure 5.14.: HSBC trend chart and sentiment distribution

Scenario	Accuracy	Precision	Recall	Specificity	F Value
		in-sample			
1	70.6%	79.0%	72.1%	68.3%	75.4%
2	80.0%	88.7%	78.6%	82.5%	83.3%
3	86.7%	81.8%	81.8%	89.5%	81.8%
		out-of-sample			
1	59.5%	76.9%	45.5%	80.0%	57.1%
2	64.9%	61.5%	50.0%	76.2%	55.2%
3	40.0%	100.0%	33.3%	100.0%	50.0%

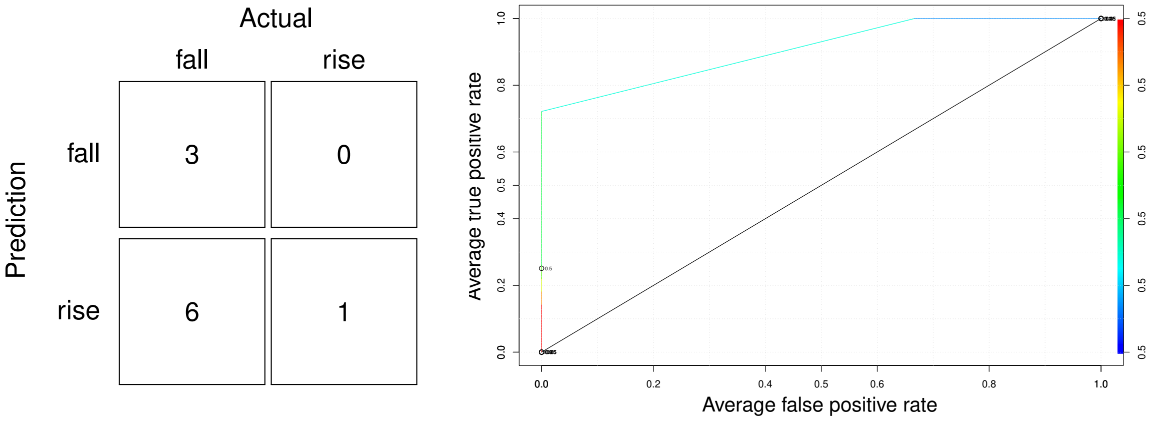
Table 5.12.: Overview of the results of the HSBC case study

the negative news as most influential, and SMA as technical feature third. ROC is again in the list of the important features for scenario 2.

Rank	Feature	Gain	Cover	Frequency
		Scenario 2		
1	RSI	0.25051	0.22847	0.21296
2	OBV	0.22170	0.23702	0.19444
3	ROC	0.20602	0.18296	0.16667
		Scenario 3		
1	negativeSentimentStdDev	0.28411	0.26186	0.24242
2	negativeMeanSentiment	0.23401	0.20304	0.18181
3	SMA	0.09600	0.08349	0.09091

Table 5.13.: Feature Importance ranking for scenario 2 and 3 of the HSBC case study

The Area under Curve metric is `AUC = 0.9762` based on the confusion matrix visualized in figure 5.15. The poor accuracy rate result in a high value for Area under Curve because of the good relation of TPR and FPR.



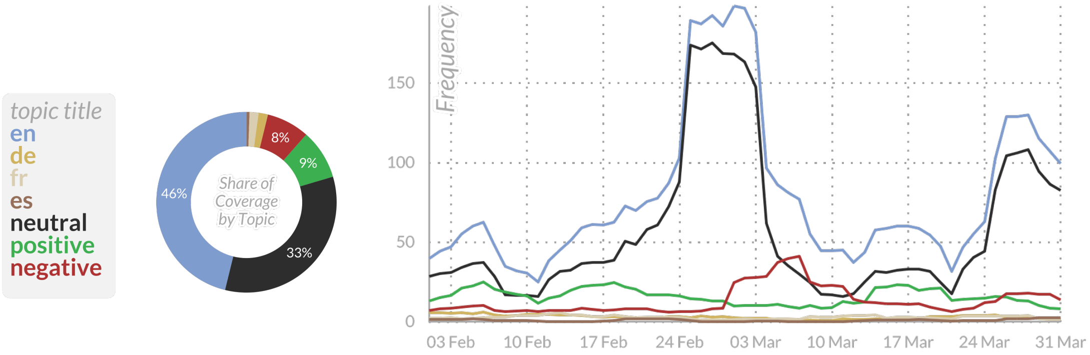
Prediction \ Actual	fall	rise
fall	3	0
rise	6	1

Figure 5.15.: Confusion Matrix and ROC curve for HSBC case study

5.3.7. Royal Dutch Shell

The detailed results for the Royal Dutch Shell case study can be found in appendix A.2.7. The curve of the closing prices has the same characteristics as the one from HSBC, with a constant downwards trend. R% indicates the stock is oversold, whereas RSI does not indicate any clear signals. For the news also Royal Dutch Shell has two peaks for the neutral news, which each last few days. Most messages are classified as neutral. There are two days with over 400 neutral messages, whereas the positive and negative messages are below 40 messages per day.

The simulation result is better than for HSBC. For the in-sample results the accuracy is 86.7% and has a recall value of 100%. In the out-of-sample simulation scenario 1 is better for



(a) Sentiment distribution

(b) Trend chart

Figure 5.16.: Royal Dutch Shell trend chart and sentiment distribution

accuracy, but comparing the F-value scenario 1 has a better overall result (60.6. vs 62.2.% Acc and in the F-value 53.3 vs 33.3%). Also in this case study the recall and precision are again low in comparison with specificity. In this simulation no news feature was ranked in the top 3 most influential features. SMA is the dominating feature in both scenarios.

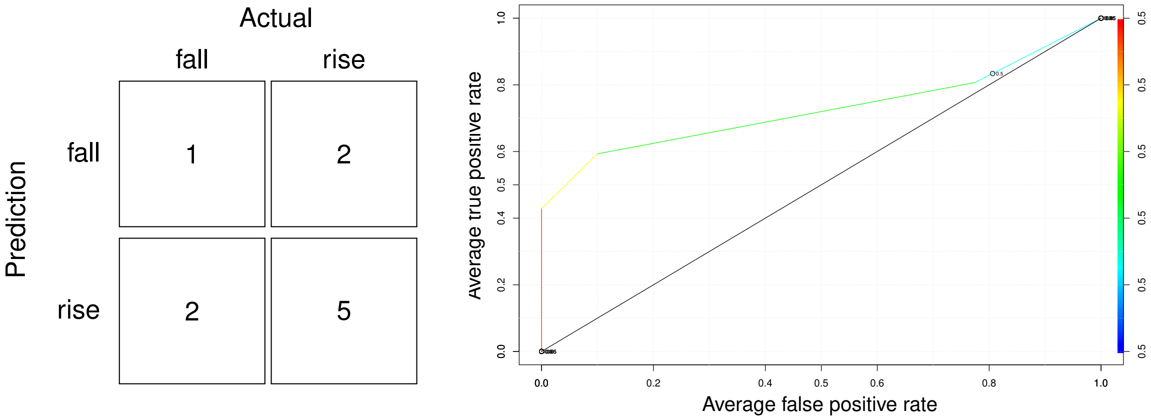
Scenario	Accuracy	Precision	Recall	Specificity	F Value
			in-sample		
1	60.6%	52.8%	60.9%	60.3%	56.6%
2	80.9%	71.7%	86.4%	77.3%	78.4%
3	86.7%	75.0%	100.0%	77.8%	87.5%
			out-of-sample		
1	62.2%	42.1%	72.7%	57.7%	53.3%
2	54.1%	57.9%	55.0%	52.9%	56.4%
3	60.0%	33.3%	33.3%	71.4%	33.3%

Table 5.14.: Overview of the results of the Royal Dutch Shell case study

Rank	Feature	Gain	Cover	Frequency
		Scenario 2		
1	SMA	0.22225	0.24470	0.24460
2	OBV	0.20314	0.20805	0.22302
3	RSI	0.19292	0.17567	0.17266
		Scenario 3		
1	SMA	0.56020	0.42817	0.34783
2	ROC	0.23946	0.20563	0.17391
3	MOM	0.05218	0.03380	0.04348

Table 5.15.: Feature Importance ranking for scenario 2 and 3 of the Royal Dutch Shell case study

The Area under Curve metric is `AUC = 0.7143` based on the confusion matrix visualized in figure 5.17.



		Actual	
		fall	rise
Prediction	fall	1	2
	rise	2	5

Figure 5.17.: Confusion Matrix and ROC curve for RDSB case study

5.4. Synthesis

The seven case studies show the impact of the news features on the prediction of stock market fluctuations. Starting with the Tesla case study, the three different models were each run with consistent parameter sets across all case studies. Both US case studies have similar results with an F-Value of about 50%. The prediction for Bitcoin is lowest (accuracy = 43,4%), which can be explained by the change of the news data sources used by webLyzard. This is, of course, a valid “real-world” scenario and has to be kept in mind for a system which constantly calculates predictions. The Austrian case studies differ in the sense that they have fewer messages and almost three-quarters of the messages are positive. This could be explained by a different sentiment analysis between the English and the German news messages. Finally the UK case studies, which have the shortest timeline (two months of news data) have entirely different results. The majority of the messages are neutral, and for Shell only technical indicators are among the most important features. HSBC on the other hand has only negative news features in the importance ranking and is the only case study where more negative news was collected than positive. Table 5.16 provides the findings of the previous sections.

The results show the utility of analyzing the results with several performance indicators, as AUC shows in some cases a different picture of the results. From the point of view of the AUC metric (relation of TPR and FPR) the model has to be improved. A ratio close to 0.5 means that the prediction of correct results is only achieved in one of two tries.

	Messages			Simulation results		
Case Study	pos %	neg %	neutral %	F-Value	AUC	Feature Importance
Tesla	60.918%	29.751%	9.332%	51.90%	0.6672	OBV, positiveTotal, neutralSentimentStdDev
Apple	59.530%	24.875%	15.595%	50.00%	0.5747	ROC, neutralSentimentStdDev, positiveMeanSentiment
Bitcoin	34.530%	12.778%	52.692%	57.10%	0.6312	OBV, RSI, ROC
OMV	70.294%	17.032%	12.674%	50.00%	0.5244	MOM, ROC, positiveMeanSentiment
Erste	75.094%	14.981%	9.925%	60.00%	0.6559	neutralSentimentStdDev, SMA, negativeTotal
Shell	16.582%	15.080%	68.338%	33.30%	0.7143	SMA, ROC, MOM
HSBC	20.379%	27.472%	52.149%	50.00%	0.9762	negativeSentimentStdDev, negativeMeanSentiment, neutralTotal

Table 5.16.: Summary of the case studies

6. Conclusion

Predicting the future isn't magic, it's artificial intelligence.

Dave Waters

Machine Learning turned out to be an adequate method to predict the trend of financial market data time series. The difference between the approaches in related work is on the one hand the type of data used for the features and on the other hand the Machine Learning algorithm. In this thesis, an approach was applied which combined quantitative data in form of market data time series and qualitative data in form of sentiment analyzed news information from news platforms and social media channels provided by webLyzard. This data serves as input for the classification ML algorithm XGBoost.

6.1. Summary

Prediction of stock market movements has a long history. In scientific research, it has been a topic for more than 70 years, whereby different models have been applied. First models use only time series and work in the tradition of the technical analysts. In the last decades, the improved performance of the hardware allows processing big data in an acceptable time frame, which allows for the processing of more and more data. Hence new models make not only use of the quotes but also additional information. This thesis uses historic time series data of a period of more than six months to have a complete feature matrix of technical indicators for six months. The selected technical indicators are chosen based on the literature and their power to predict a trend of a time series. This is the way in which technical analysts use historical data and indicators for their decisions. However, this does not reflect any public opinion. Today's communication is heavily based on online news and social media channels. As outlined in chapter 3 related work has already shown an influence of the public mood on stock market movements.

Based on the scientific question from chapter 1 *"Can the prediction of finance market data time series be improved by combining sentiment news features with technical indicators using a Machine Learning approach?"*, seven case studies had been defined to verify an XGBoost prototype implemented in the script language R. XGBoost is a classification Machine

Learning algorithm which became popular after winning several Kaggle challenges. Overall, the results for the US case show that the model including news features achieved greater accuracy than those which did not. For the Austrian case studies, the results across models are almost equal, regardless of the inclusion of news data.

6.2. Contribution to Knowledge

The literature reveals only a few approaches using XGBoost, and always with models based exclusively. This thesis combines historic market data and sentiment news provided by webLyzard as training and test data for XGBoost as a Machine Learning algorithm. The case studies serve to verify the model and optimize the prediction results. The model was optimized by means of Hyperparameter Tuning and Cross-Validation for the case study Tesla. With the optimized model, XGBoost outperformed the scenario incorporating only technical indicators based on F-score comparison. It turned out that the amount of data and the trend of the sentiment influenced the results. For the two Austrian case studies, where the amount of messages is lower and the sentiment is much more positive, the prediction model is equal to the scenario using only technical indicators. For three case studies (two in the UK and Bitcoin) the news data was rather variable. The data had unnatural peaks and new data sources were added, which changed the quality of the prediction because training and test data differed.

6.3. Implications for Relevant Stakeholders

This method allows the stakeholder to integrate sentiment news information in their predictions. As market behavior is driven by public mood, this approach can support decision making by providing a more comprehensive overview picture. Today's news and social media landscape is complex and difficult to monitor. Hence tools like webLyzard are an optimal support tool to aggregate news and detect the prevailing public mood behind the debate. This thesis concludes with some suggestions for future work.

6.4. Future Research

The prototype implementation provided a first verification of the hypothesis. Here is a list of possible future works which can be divided into improvement of the model as well as enhancement of implementation itself.

Hyperparameter Tuning: The tuning can be made individually for each case separately. This would lead to better prediction of each case, but is on the other hand effort which has to be done manually.

Equity Model: The current approach models each stock or currency individually. A different approach would be to collect the data of several stocks and or indices and develop a model to predict the trend of stocks based on a set of stock technical features and news features. This would lead to an increased number of features, which can be handled by XGBoost. Therefore a parallelization of calculation could be useful.

Real Time: The market data can be loaded near real-time. Also, webLyzard provides an API and the possibility to load news in a shorter time frame. This would allow for the calculation of predictions in “near real-time”.

Sliding Window: Currently the training data and test data is fixed. A possible way for better predictions would be the implementation of sliding windows. In this scenario the training data is split, for example in weeks. A window of four weeks is used to train the model and make predictions on the test set. Subsequent iterations shift the training data sample forwards in time to include what was previously test data.

Weekend news: Analyze the effect of the news of weekends or holidays, which was omitted in the current research.

Split factor: The split factor was defined with 75%. A further investigation could analyze the response of the model by changing the split factor by using more or fewer data to teach.

Sentiment Analyses: Currently the model uses data from news channels and Twitter. This data could be selected in a more specific manner. The sentiment analysis could be improved to be more sensitive to financial news specifics.

Specific information: The news extraction from webLyzard could be extended by country and sector information and this could serve as additional features.

Data Loading: The data could be loaded via the API of the market data providers and webLyzard. This would allow calculating on a daily basis and observing the quality of predictions over a longer time frame.

Bibliography

Adams, 1.-2., Douglas. (1980). *The hitchhiker's guide to the galaxy*. First American edition. New York : Harmony Books, 1980. Retrieved from https://search.library.wisc.edu/catalog/999547338802121

Alpha Vantage. (2019). Retrieved from https://www.alphavantage.co/

Awesome XGBoost. (2019). Retrieved from https://github.com/dmlc/xgboost/tree/master/demo

Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (2019). Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance*, *47*, 552–567. doi:https://doi.org/10.1016/j.najef.2018.06.013

Beckmann, M. (2017). *Stock price change prediction using news text mining* (Doctoral dissertation, Federal University of Rio de Janeiro).

Berkin, A., & Swedroe, L. (2016). *Your complete guide to factor-based investing: The way smart money invests today*. Buckingham. Retrieved from https://books.google.at/books?id=iSNBvgAACAAJ

Bollen, J., & Mao, H. (2011). Twitter mood as a stock market predictor. *Computer*, *44*(10), 91–94. doi:10.1109/mc.2011.323

Brownlee, J. (2018). *Xgboost with python*. https://machinelearningmastery.com/xgboost-with-python/.

Chen, T., & Guestrin, C. (2016). Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*. doi:10.1145/2939672.2939785

Cootner, P. H. (1964). *The random character of stock market prices*. M.I.T. Press.

Cutler, D. M., Poterba, J. M., & Summers, L. H. (1988). What moves stock prices? *Journal of Portfolio Management*. Working Paper Series, (2538). doi:10.3386/w2538

Dimson, E., Marsh, P., & Staunton, M. (2011). Equity premiums around the world. *Ch. 4 of Rethinking the Equity Risk Premium*, 32–52. Retrieved from https://dx.doi.org/10.2139/ssrn.1940165

Fama, E. F. (1965). The behavior of stock-market prices. *The Journal of Business*, *38*(1), 34–105. Retrieved from http://www.jstor.org/stable/2350752

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, *25*(2), 383–417. Retrieved from http://www.jstor.org/stable/2325486

Fang, J., Qin, Y., & Jacobsen, B. (2014). Technical market indicators: An overview. *Journal of Behavioral and Experimental Finance*, *4*, 25–56. doi:https://doi.org/10.1016/j.jbef.2014.09.001

Fidelity. (2019). Retrieved from https://www.fidelity.com

Finance Yahoo Platform. (2019). Retrieved from https://finance.yahoo.com/

FM Labs. (2019). Retrieved from https://www.fmlabs.com/reference/

Friedman, J. H. (2000). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*, 1189–1232.

Gerig, A. (2012). High-frequency trading synchronizes prices in financial markets. *SSRN Electronic Journal*. doi:10.2139/ssrn.2173247

Github Project of the Thesis. (2019). Retrieved from https://github.com/wolferl42195/XGBoost4StockPrices

Guo, T., & Antulov-Fantulin, N. (2018). Predicting short-term bitcoin price fluctuations from buy and sell orders.

Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, *32*(10), 2513–2522. doi:10.1016/j.cor.2004.03.016

Iannone, R. (2019). R-package diagrammer. Retrieved from http://rich-iannone.github.io/DiagrammeR/index.html

Investopedia. (2019). Retrieved from https://www.investopedia.com/dictionary/

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2014). *An introduction to statistical learning: With applications in r*. Springer Publishing Company, Incorporated.

Jegadeesh, N., & Titman, S. (2011). Momentum. *SSRN Electronic Journal*. doi:10.2139/ssrn.1919226

Kaggle. (2019). Retrieved from https://www.kaggle.com/

Kim, K. J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, *55*(1-2), 307–319. doi:10.1016/s0925-2312(03)00372-2

Merton, R. (1973). The theory of rational option pricing. *Bell J Econ Manage Sci*, *4*, 141–183. doi:10.1142/9789812701022_0008

Mitchell, M. L., & Mulherin, J. H. (1994). The impact of public information on the stock market. *The Journal of Finance*, *49*(3), 923–950. doi:10.1111/j.1540-6261.1994.tb00083.x

Mittermayer, M. .-.-. (2004). Forecasting intraday stock price trends with text mining techniques. In *37th annual hawaii international conference on system sciences, 2004. proceedings of the* (p. 10). doi:10.1109/HICSS.2004.1265201

Nassirtoussi, A. K., Aghabozorgi, S., Wah, T. Y., & Ngo, D. C. L. (2014). Text mining for market prediction: A systematic review. *Expert Systems with Applications*, *41*(16), 7653–7670. doi:https://doi.org/10.1016/j.eswa.2014.06.009

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, *7*, 21. doi:10.3389/fnbot.2013.00021

Paradkar, M. (2017). *Forecasting markets using extreme gradient boosting (xgboost)*. Pioneer Institute for Algo Trading.

Project side xgboost.ai. (2019). Retrieved from https://xgboost.ai/

R-Package caret. (2019). Retrieved from http://topepo.github.io/caret/index.html

R-Package dplyr. (2019). Retrieved from https://dplyr.tidyverse.org/

R-Package e1071. (2019). Retrieved from https://www.rdocumentation.org/packages/e1071/versions/1.7-1

R-Package PerformanceAnalytics. (2019). Retrieved from https://github.com/braverock/PerformanceAnalytics

R-Package quantmod. (2019). Retrieved from https://www.quantmod.com/

R-Package ROCR. (2019). Retrieved from https://rocr.bioinf.mpi-sb.mpg.de/

R-Package tidyquant. (2019). Retrieved from https://business-science.github.io/tidyquant/

R-Package tidyverse. (2019). Retrieved from https://tidyr.tidyverse.org/

R-Package xgboost. (2019). Retrieved from https://xgboost.readthedocs.io/en/latest/R-package/index.html

R-Package xts. (2019). Retrieved from http://joshuaulrich.github.io/xts/

Rasekhschaffe, K. C., & Jones, R. C. (2019). Machine learning for stock selection. *Financial Analysts Journal*, *0*(0), 1. doi:10.1080/0015198X.2019.1596678. eprint: https://doi.org/10.1080/0015198X.2019.1596678

Ruiz-Martìnez, J., Valencia-Garcìa, R., & Garcìa-Sànchez, F. (2012). Semantic-based sentiment analysis in financial news. In *Ceur workshop proceedings* (Vol. 862, pp. 38–51).

Salisu, A., Isah, K., & Akanni, L. (2018). Improving the predictability of stock returns with bitcoin prices. *The North American Journal of Economics and Finance*. doi:10.1016/j.najef.2018.08.010

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *3*(3), 210–229.

Scharl, A., & Fischl, D. (2015). Topic wizard: Interactive visual tool for defining and disambiguating topics via regular expressions. In *Ihc '15 proceedings of the 14th brazilian symposium on human factors in computing systems* (pp. 1–4). doi:10.1145/3148456.3148517

Scharl, A., & Herring, D. D. (2013). Extracting knowledge from the web and social media for progress monitoring in public outreach and science communication. In *19th brazilian* *symposium on multimedia and the web, webmedia '13, salvador, brazil, november 5-8, 2013* (pp. 121–124). doi:10.1145/2526188.2526219

Scharl, A., Herring, D. D., Rafelsberger, W., Hubmann-Haidvogel, A., Kamolov, R., Fischl, D., Föls, M., & Weichselbraun, A. (2017). Semantic systems and visual tools to support environmental communication. *IEEE Systems Journal*, *11*(2), 762–771. doi:10.1109/JSYST.2015.2466439

Scharl, A., Hubmann-Haidvogel, A., Jones, A., Fischl, D., Kamolov, R., Weichselbraun, A., & Rafelsberger, W. (2016). Analyzing the public discourse on works of fiction - detection and visualization of emotion in online coverage about hbo's game of thrones. *Inf. Process. Manage. 52*(1), 129–138. doi:10.1016/j.ipm.2015.02.003

Schumaker, R. P., & Chen, H. (2009a). A quantitative stock prediction system based on financial news. *Information Processing & Management*, *45*(5), 571–583. doi:10.1016/j.ipm.2009.05.001

Schumaker, R. P., & Chen, H. (2009b). Textual analysis of stock market prediction using breaking financial news. *ACM Transactions on Information Systems*, *27*(2), 1–19. doi:10.1145/1462198.1462204

Science, A. D. (2019). R-package xgboostexplainer. Retrieved from https://github.com/AppliedDataSciencePartners/xgboostExplainer

Scott, L. (2019). Interpretable machine learning with xgboost. Retrieved from https://towardsdatascience.com/interpretable-machine-learning-with-xgboost-9ec80d148d27

Securities, U. S., & Commission, E. (2019). Retrieved from https://www.scribd.com/document/400499093/

Statista. (2019). The 100 largest companies in the world by market value in 2018. Retrieved from https://www.statista.com/statistics/263264/top-companies-in-the-world-by-market-value/

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (Second). The MIT Press. Retrieved from http://incompleteideas.net/book/the-book-2nd.html

Tay, F. E., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega*, *29*(4), 309–317. doi:10.1016/s0305-0483(01)00026-3

The R-Project. (2019). Retrieved from https://www.r-project.org/

Towards Data Science. (2019). Retrieved from https://towardsdatascience.com/

Urstadt, B. (2009). *Trading shares in milliseconds*. MIT Technical Report. New York.

Vapnik, V., & Lerner, A. (1963). Pattern recognition using generalized portrait method. *Automation and Remote Control*, *24*.

W. Banz, R. (1981). The relationship between return and market value of common stocks. *Journal of Financial Economics*, *9*, 3–18. doi:10.1016/0304-405X(81)90018-0

Wuthrich, B., Cho, V., Leung, S., Permunetilleke, D., Sankaran, K., & Zhang, J. (1998). Daily stock market forecast from textual web data. (Vol. 3, 2720–2725 vol.3). doi:10.1109/ICSMC.1998.725072

XGBoost Github Project home. (2019). Retrieved from https://github.com/dmlc/xgboost

Zauzmer, B. (2018). The best motion picture of 2017. Retrieved from https://twitter.com/BensOscarMath

Zhai, Y., Hsu, A., & Halgamuge, S. K. (2007). Combining news and technical indicators in daily stock price trends prediction. In D. Liu, S. Fei, Z. Hou, H. Zhang, & C. Sun (Eds.), *Advances in neural networks – isnn 2007* (pp. 1087–1096). Berlin, Heidelberg: Springer Berlin Heidelberg.

Zhang, X., Fuehres, H., & Gloor, P. A. (2011). Predicting stock market indicators through twitter “i hope it is not as bad as i fear”. *Procedia - Social and Behavioral Sciences*, *26*, 55–62. The 2nd Collaborative Innovation Networks Conference - COINs2010. doi:https://doi.org/10.1016/j.sbspro.2011.10.562

A. APPENDICES

A.1. Prototype Setup

The prototype was implemented with R Version 3.5.2 in R-Studio Version 1.1.463. Table A.1 lists the version of all R packages used.

Package	Version
bindrcpp	0.2.2
devtools	2.0.2
caret	6.0-81
corrplot	0.84
DiagrammeRsvg	0.1
DiagrammeR	1.0.0
e1071	1.7-0.1
dplyr	0.7.8
forcats	0.3.0
FSelector	0.31
gplots	3.0.1.1
ggplot2	3.1.0
lattice	0.20-38
quantmod	0.4-13
pacman	0.5.0
PerformanceAnalytics	1.5.2
purrr	0.3.0
readr	1.3.1
ROCR	1.0-7
rsvg	1.3
usethis	1.4.0
tidyr	0.8.2
tibble	2.0.1
tidyverse	1.2.1
stringr	1.3.1
TTR	0.23-4
ubridate	1.7.4
xgboost	0.82.1
xgboostExplainer	0.1
xts	0.11-2
zoo	1.8-4

Table A.1.: Complete configuration of the prototype environment

A.2. Case Study Result Details

In this section of detailed results of the case studies are collected.

A.2.1. Tesla

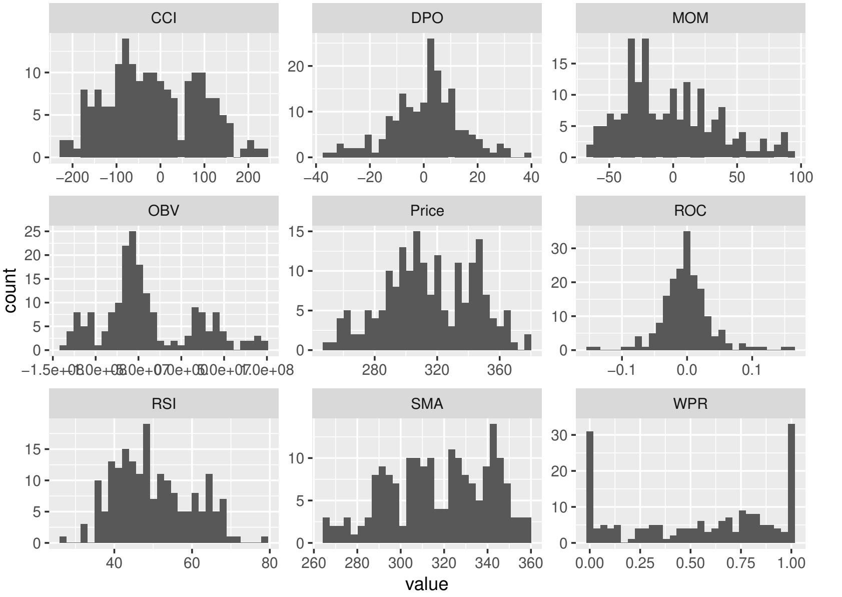


Figure A.1.: Histogram of the technical indicators of the Tesla case study

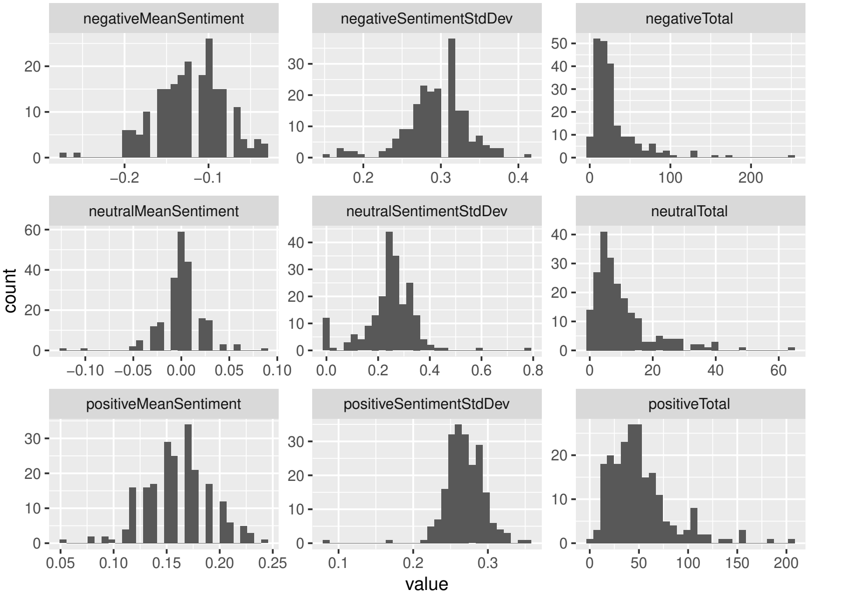


Figure A.2.: Histogram of the news features of the Tesla case study

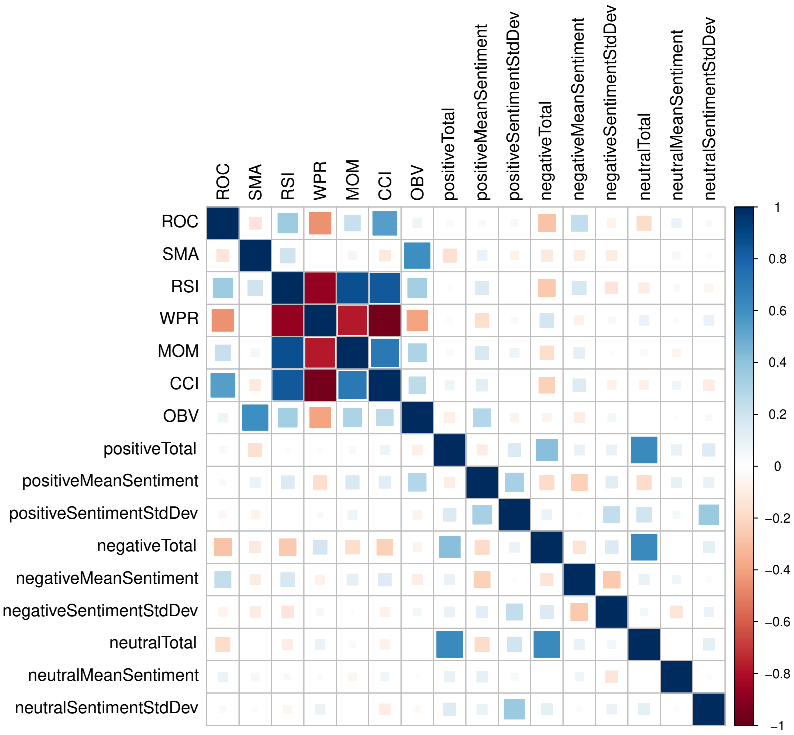


Figure A.3.: Correlation matrix of all features of the Tesla case study

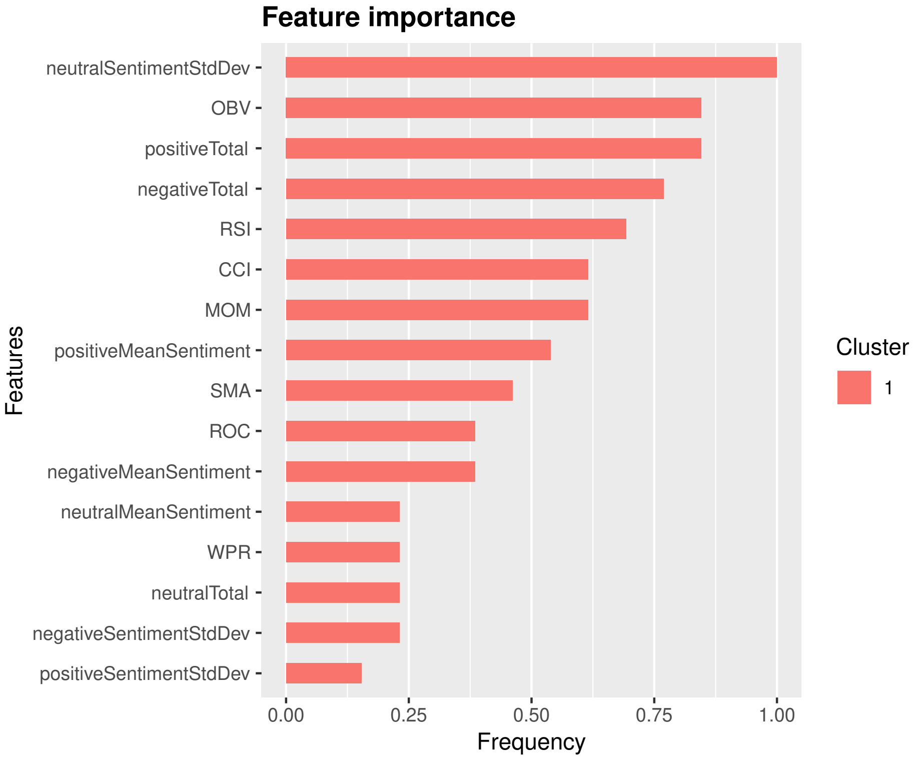


Figure A.4.: Feature Importance Ranking for Scenario 3 of the Tesla case study

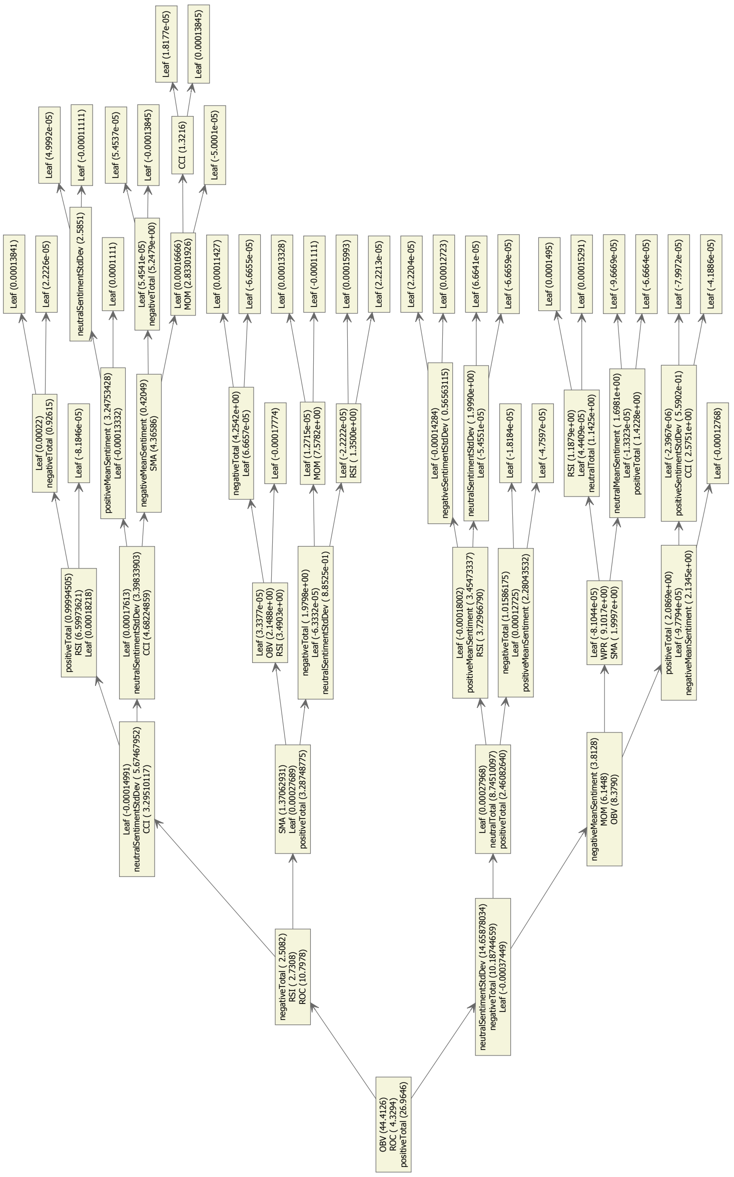


Figure A.5.: Example tree of the Tesla case study

A.2.2. Apple

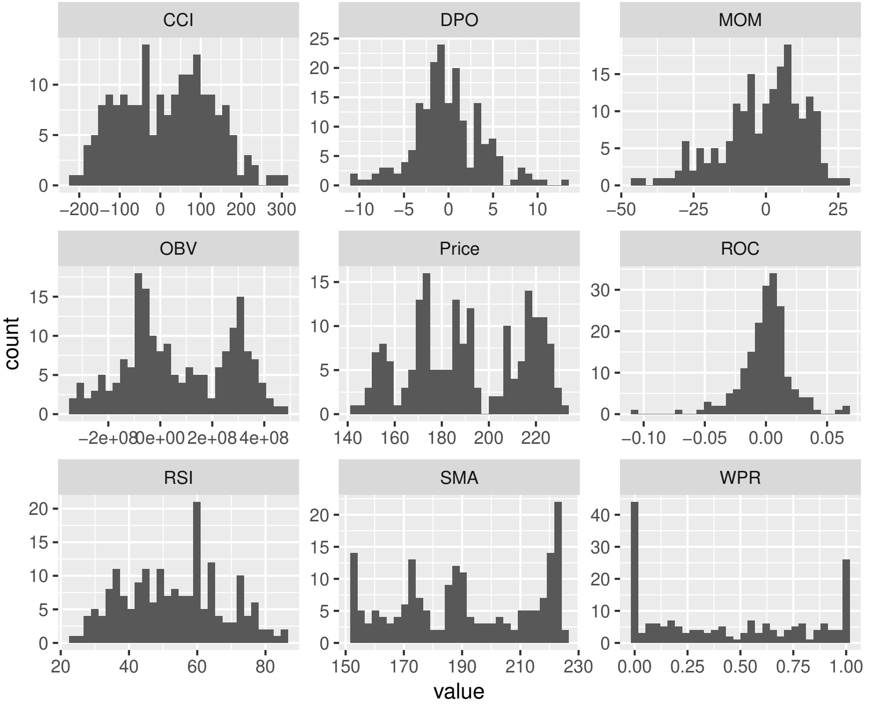


Figure A.6.: Histogram of the technical indicators of the Apple case study

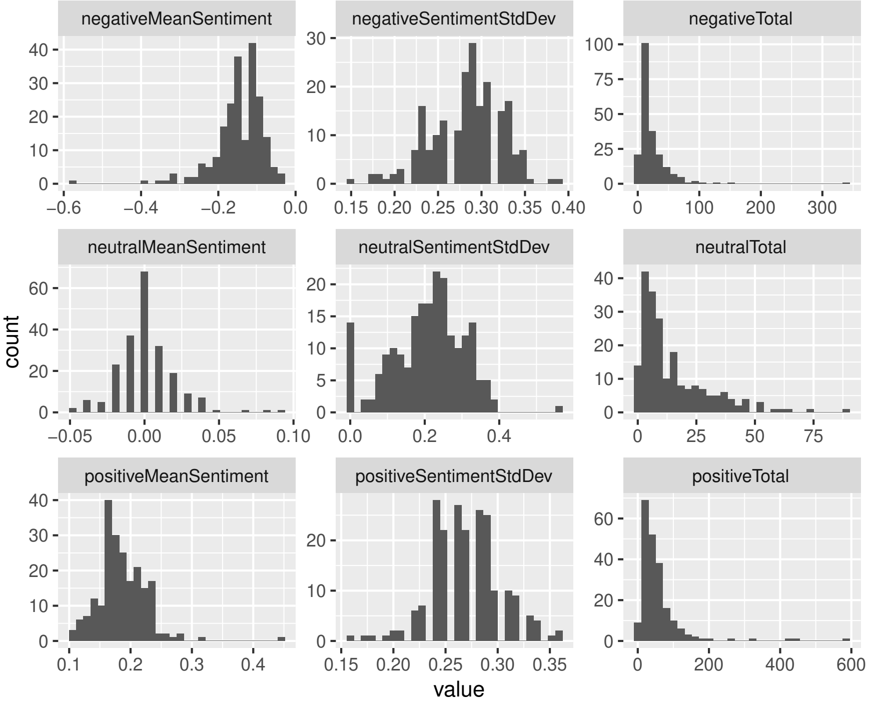


Figure A.7.: Histogram of the news features of the Apple case study

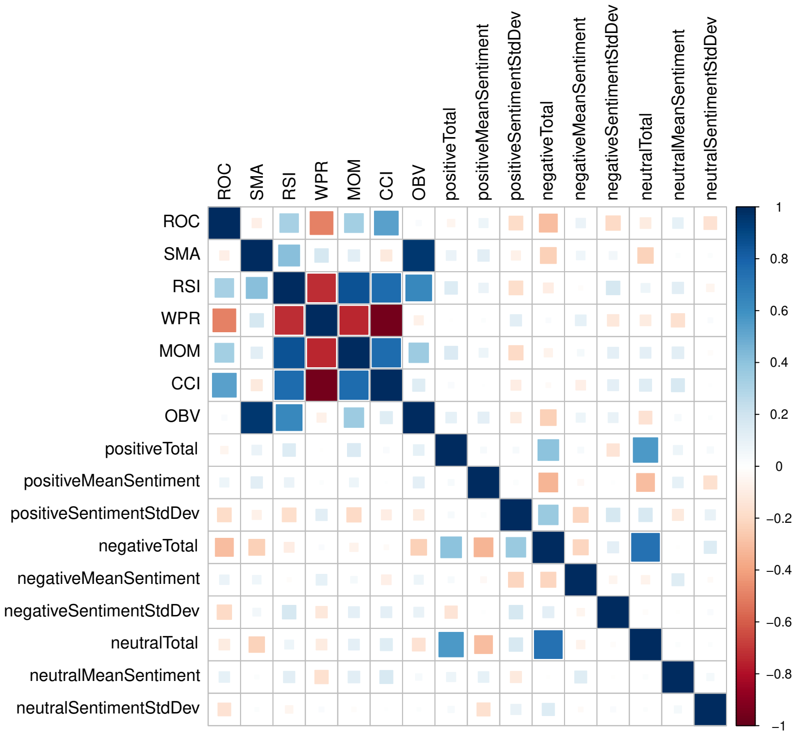


Figure A.8.: Correlation matrix of all features of the Apple case study

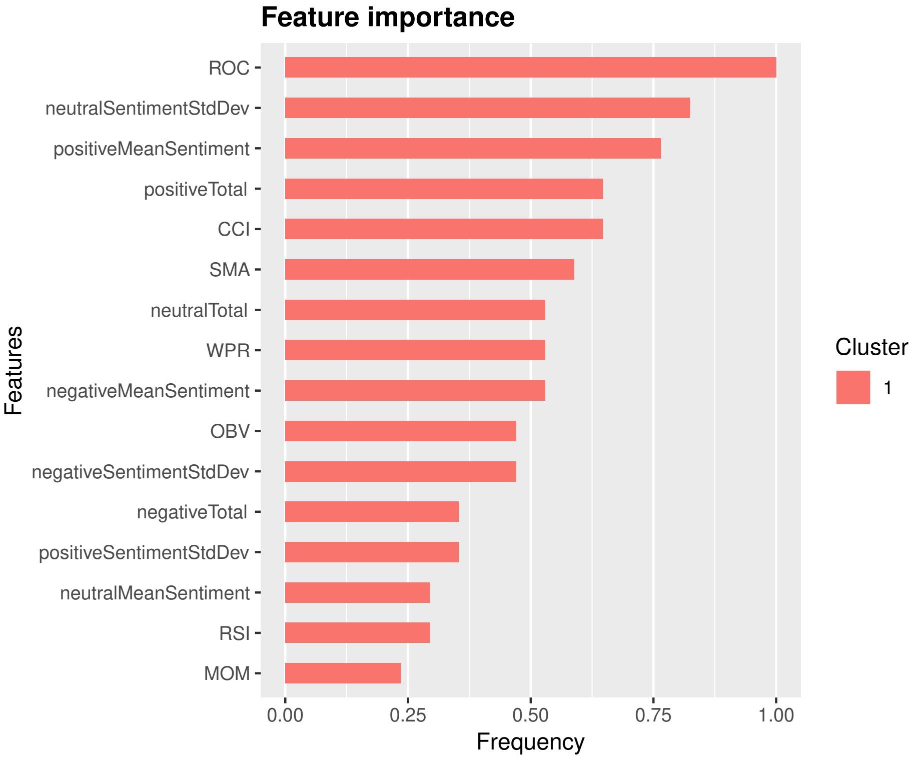


Figure A.9.: Feature Importance Ranking for Scenario 3 of the Apple case study

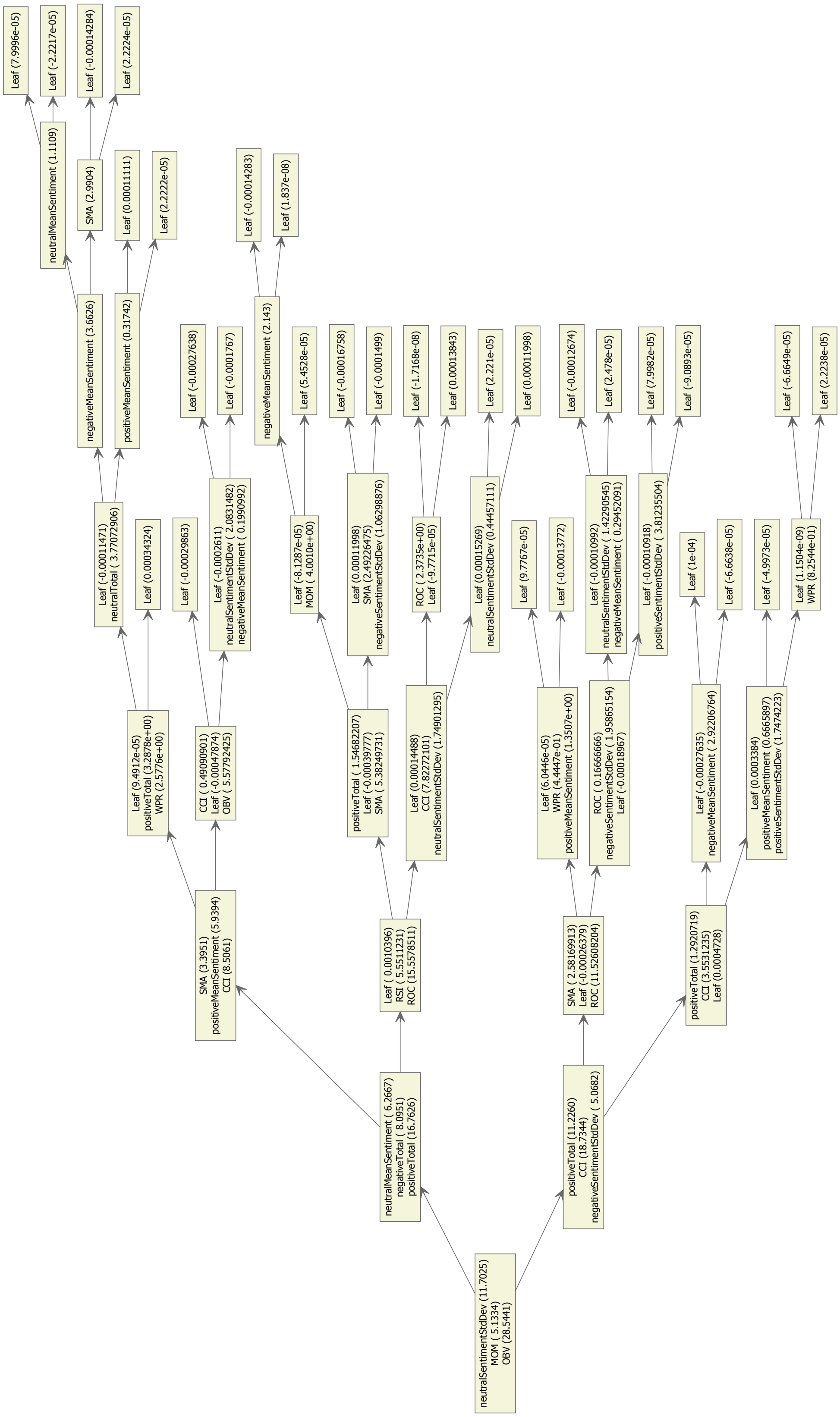


Figure A.10.: Example tree of the Apple case study

A.2.3. Bitcoin

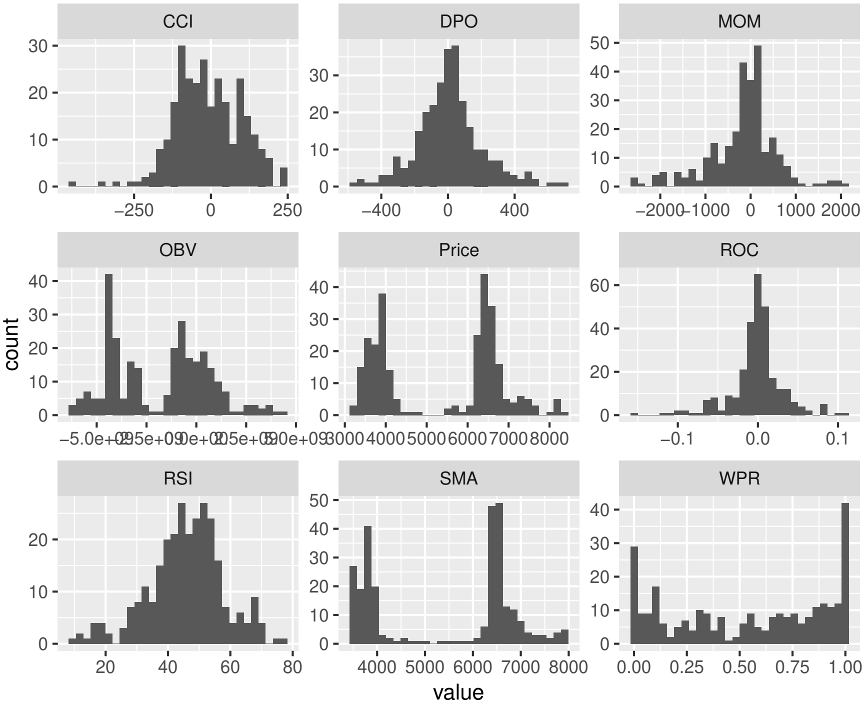


Figure A.11.: Histogram of the technical indicators of the Bitcoin case study

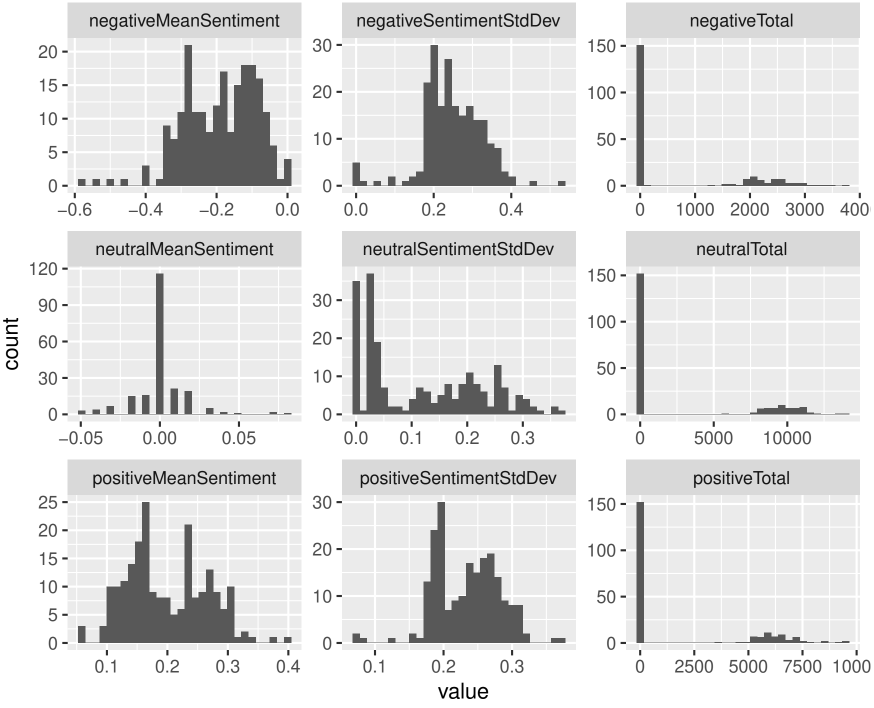


Figure A.12.: Histogram of the news features of the Bitcoin case study

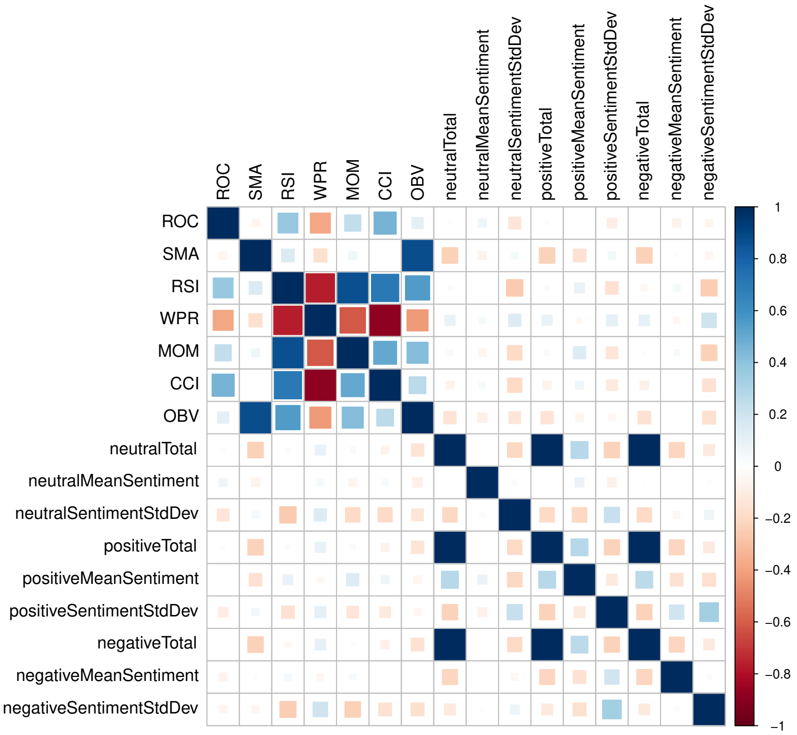


Figure A.13.: Correlation matrix of all features of the Bitcoin case study

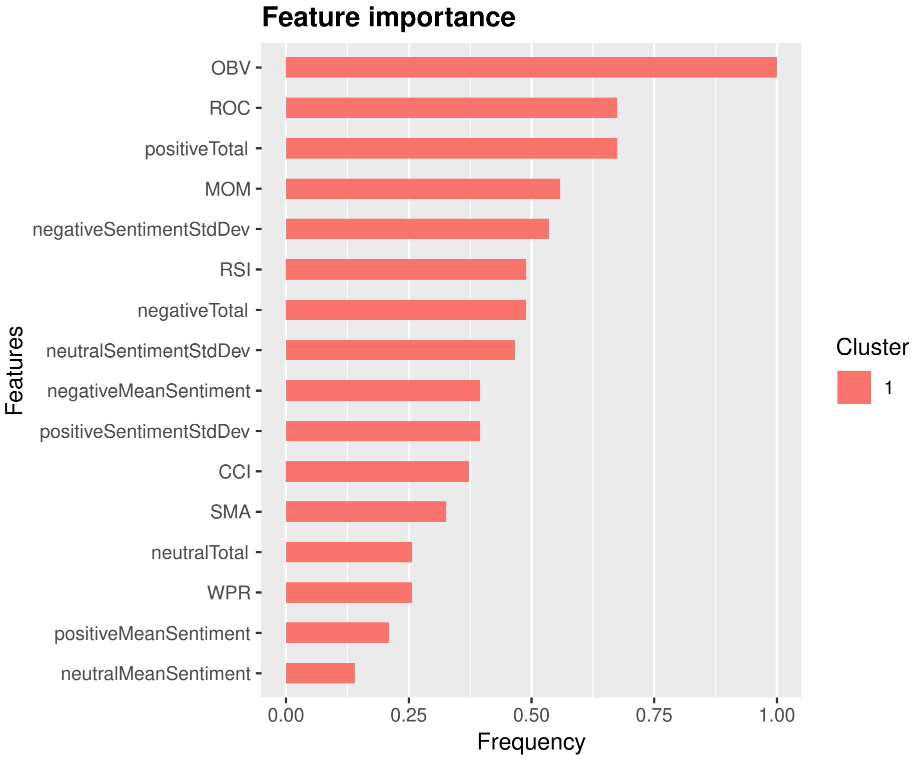


Figure A.14.: BTC Feature Importance Ranking for Scenario 3

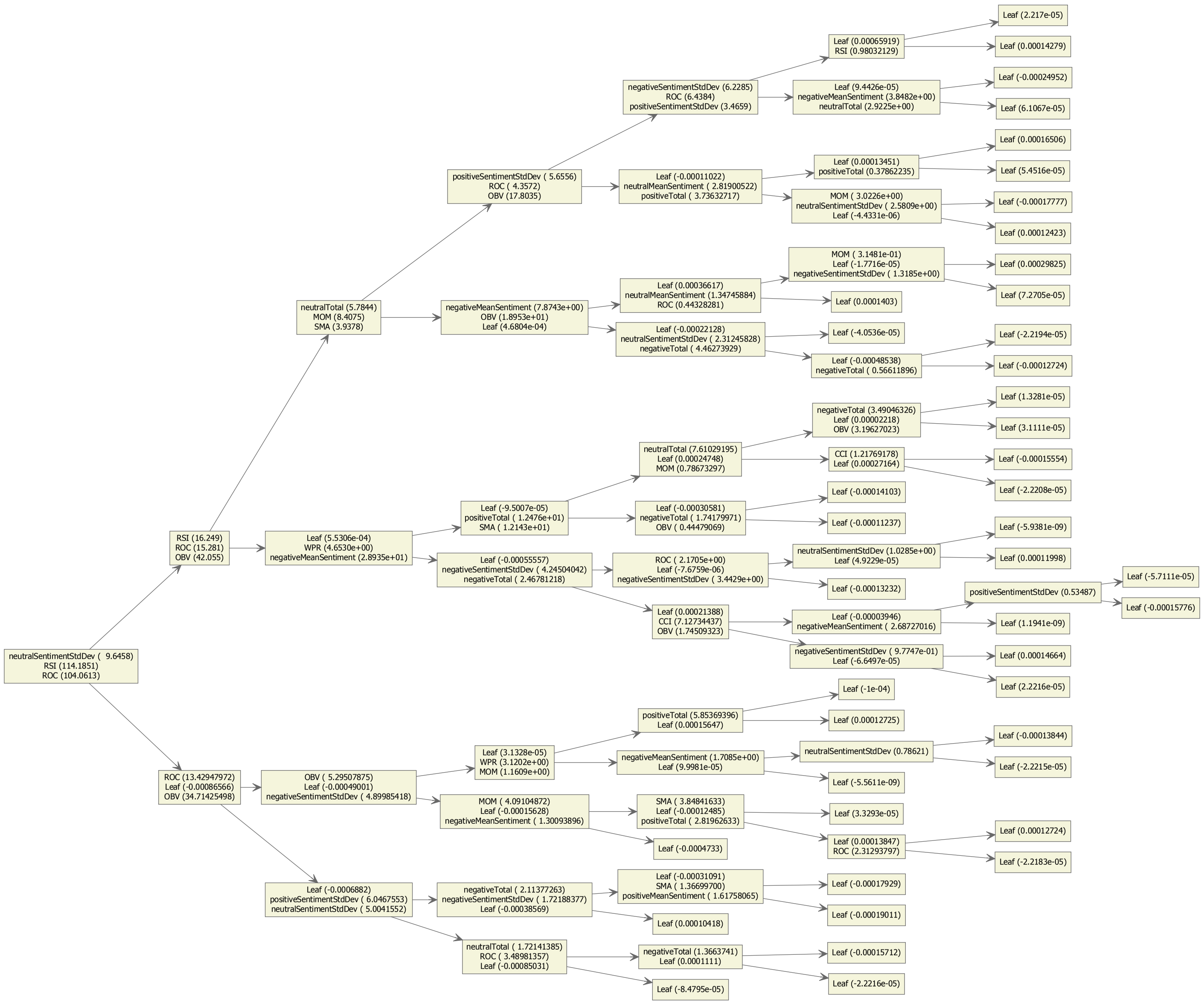


Figure A.15.: Example tree of the Bitcoin case study

A.2.4. Erste Group Bank

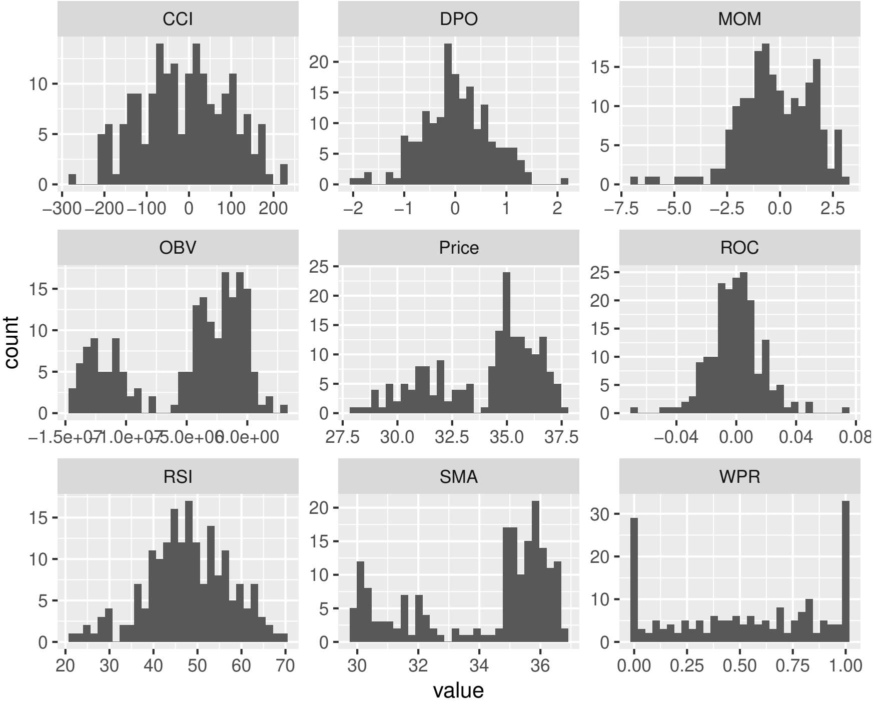


Figure A.16.: Histogram of the technical indicators of the Erste Group Bank case study

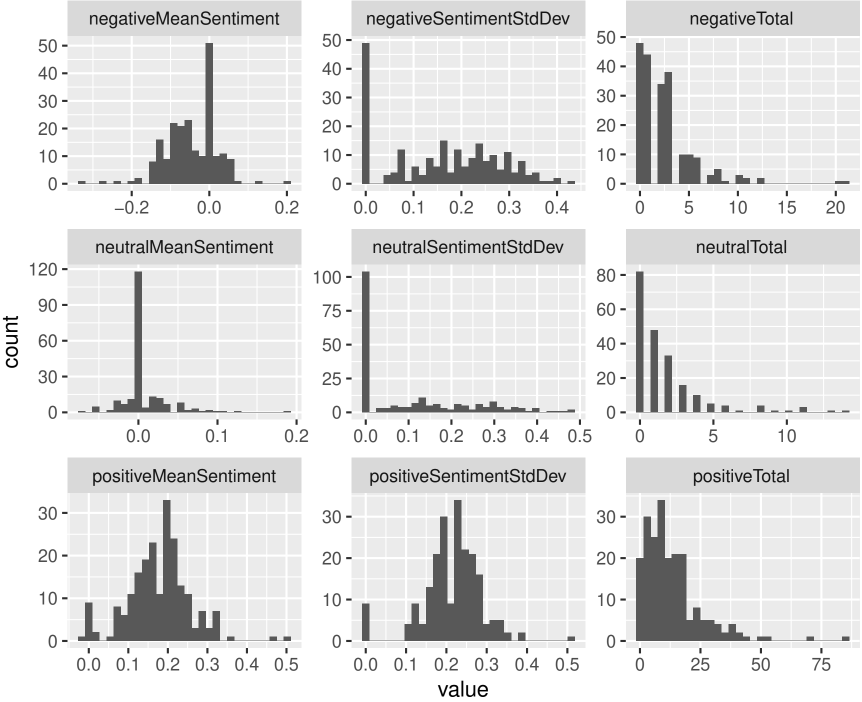


Figure A.17.: Histogram of the news features of the Erste Group Bank case study

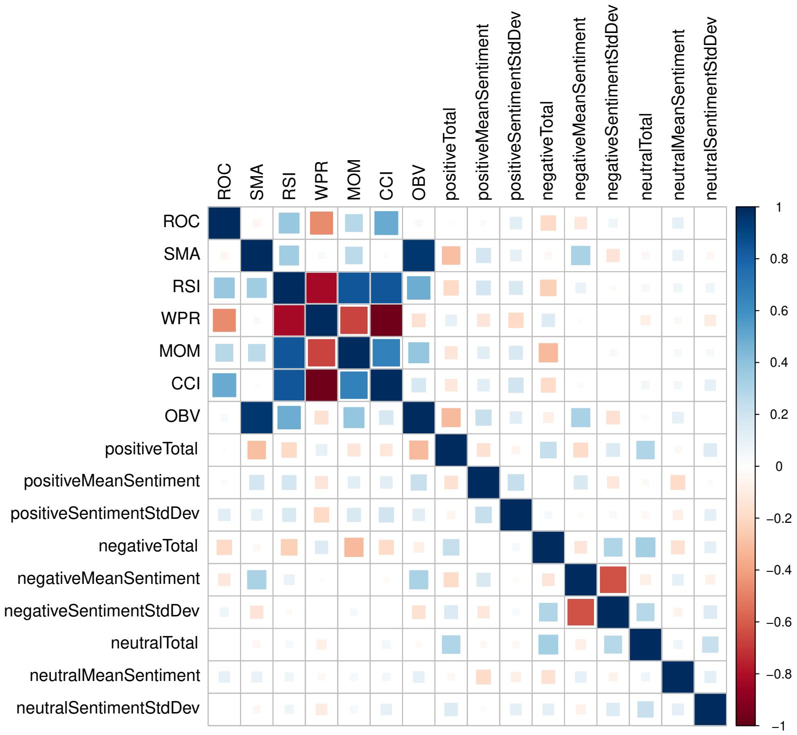


Figure A.18.: Correlation matrix of all features of the Erste Group Bank case study

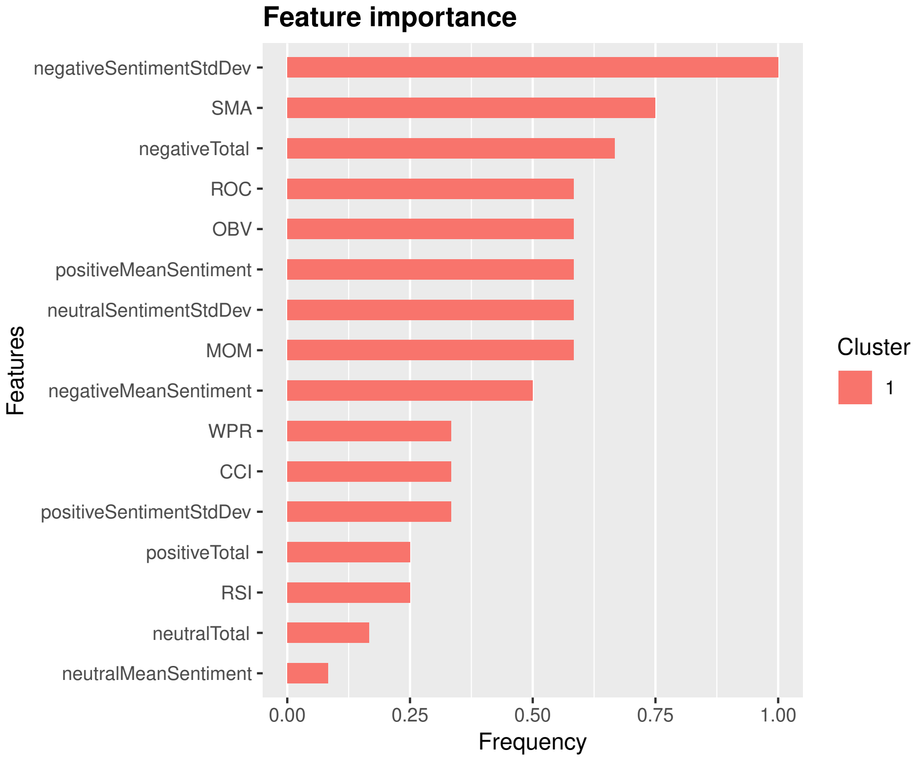


Figure A.19.: Feature Importance Ranking for Scenario 3 of the Erste Group Bank case study

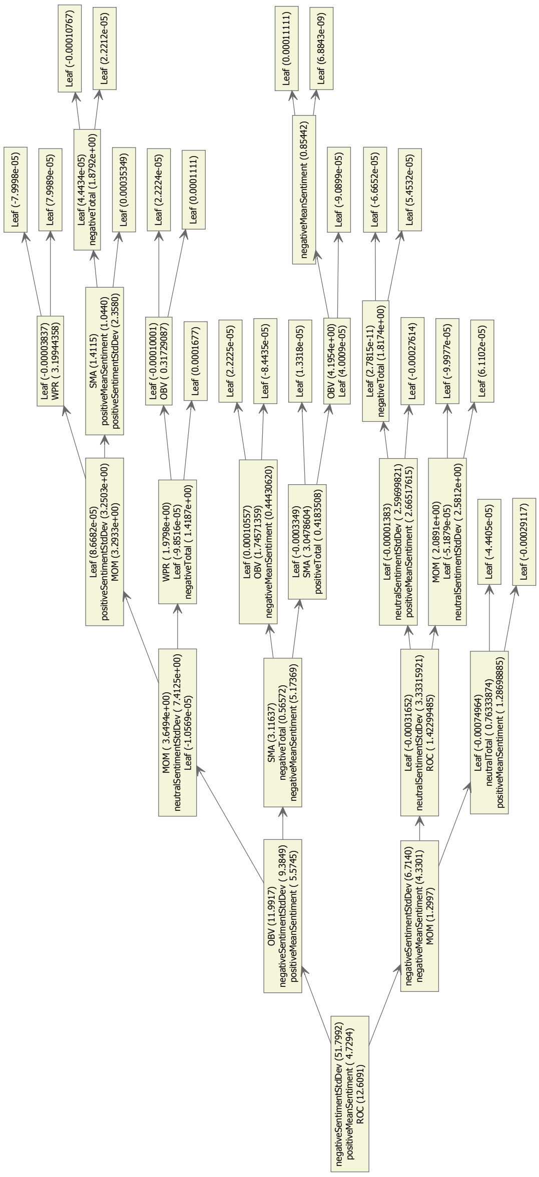


Figure A.20.: Example tree of the Erste Group Bank case study

A.2.5. OMV

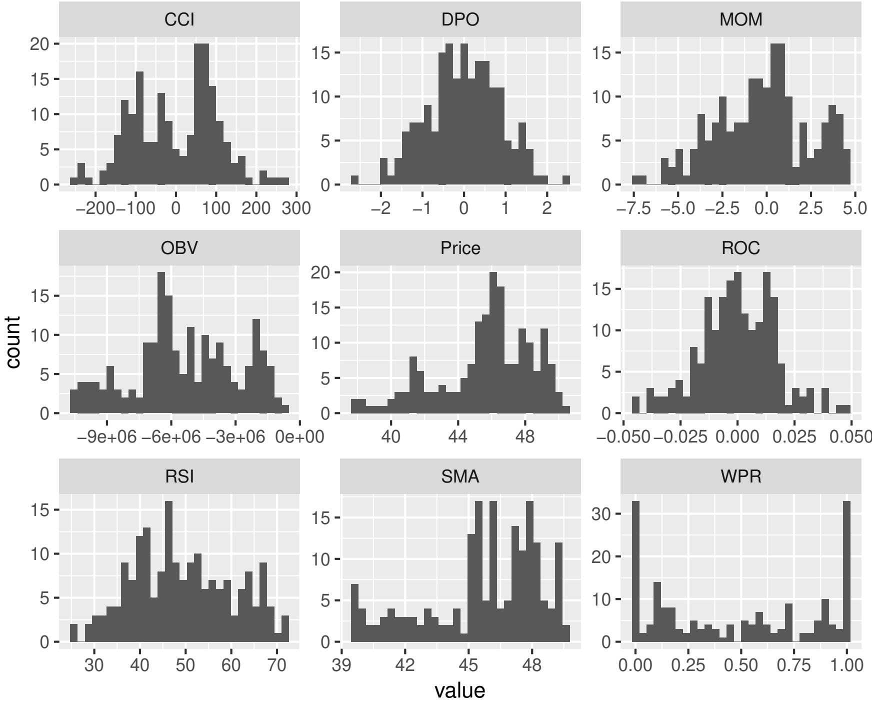


Figure A.21.: Histogram of the technical indicators of the OMV case study

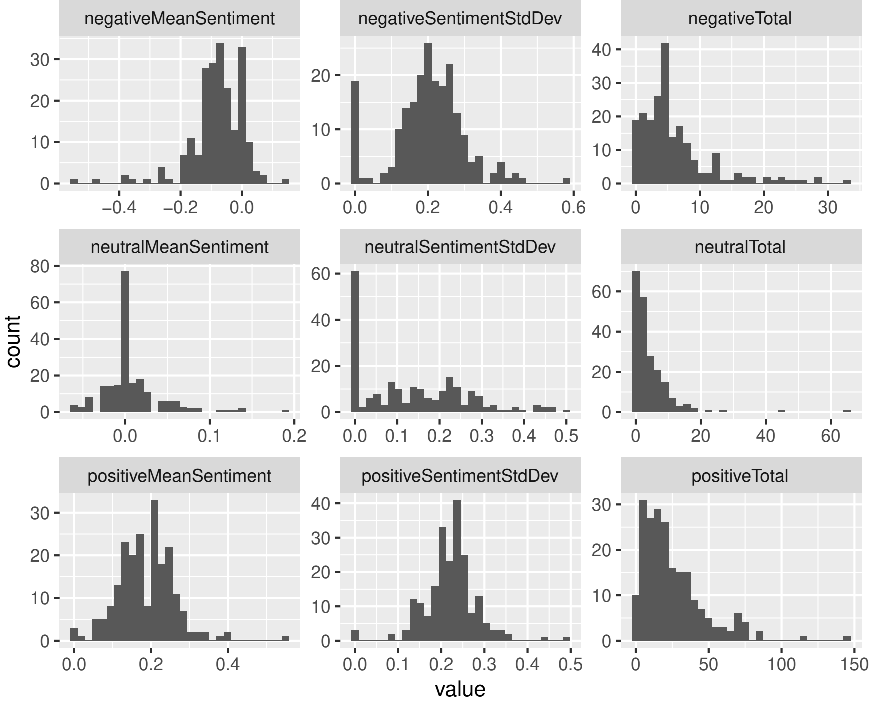


Figure A.22.: Histogram of the news features of the OMV case study

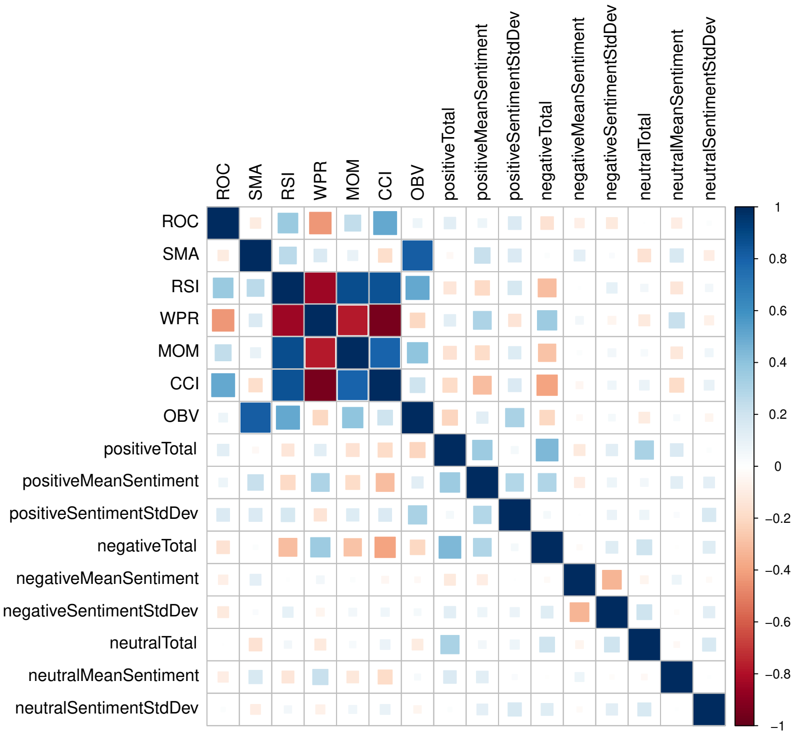


Figure A.23.: Correlation matrix of all features of the OMV case study

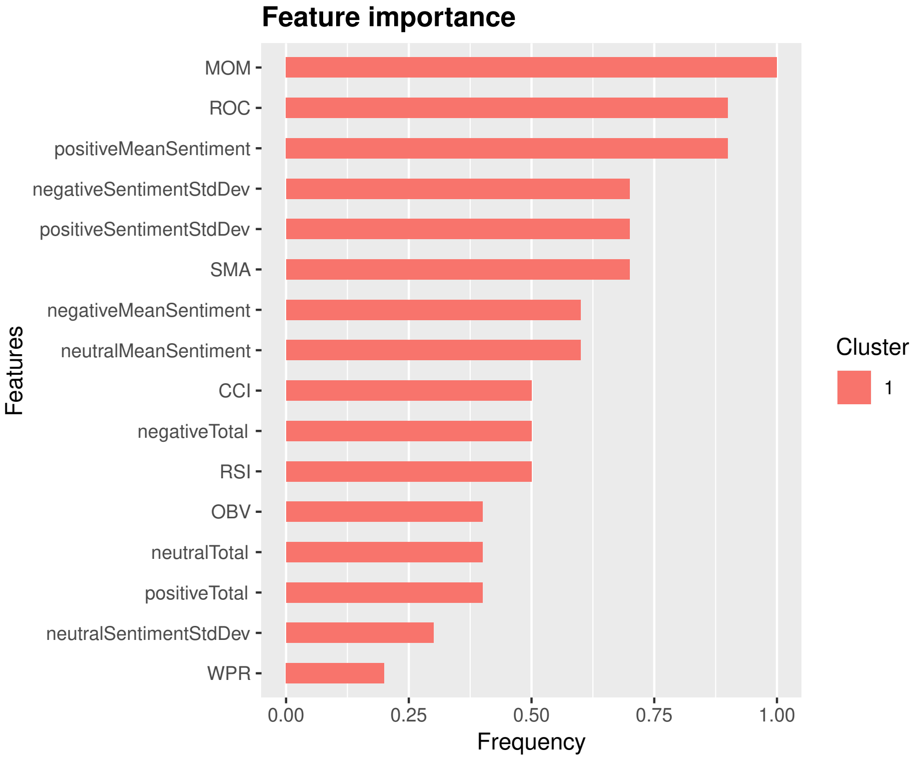


Figure A.24.: Feature Importance Ranking for Scenario 3 of the OMV case study

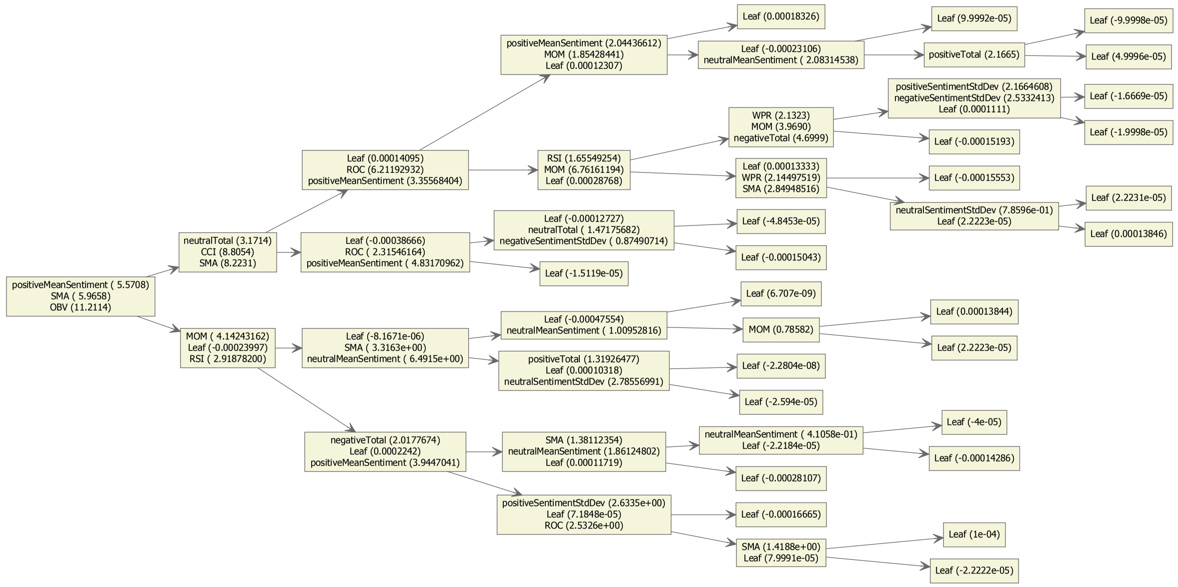


Figure A.25.: Example tree of the OMV case study

A.2.6. HSBC Holdings

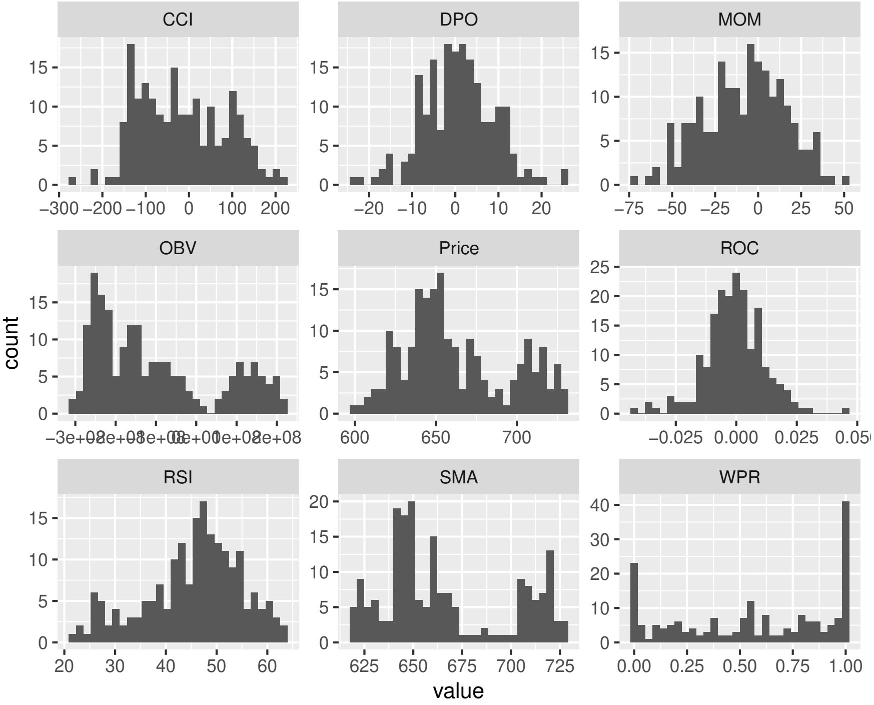


Figure A.26.: Histogram of the technical indicators of the HSBC case study

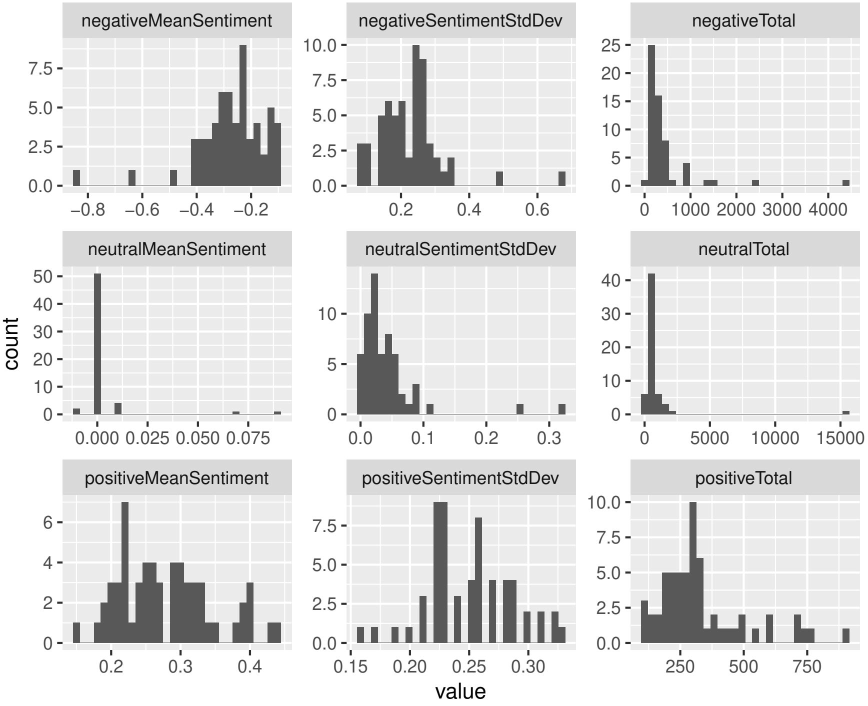


Figure A.27.: Histogram of the news features of the HSBC case study

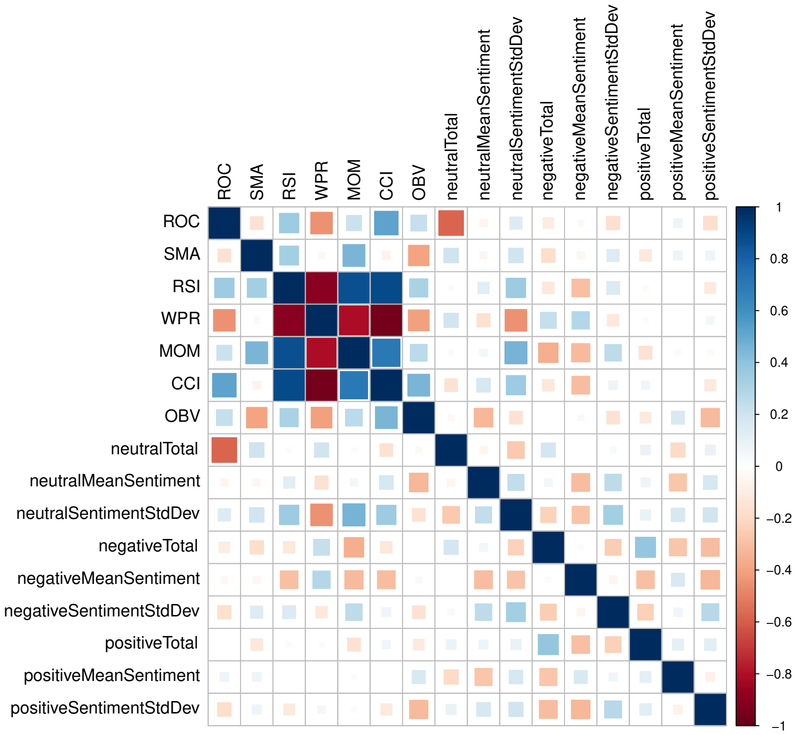


Figure A.28.: Correlation matrix of all features of the HSBC case study

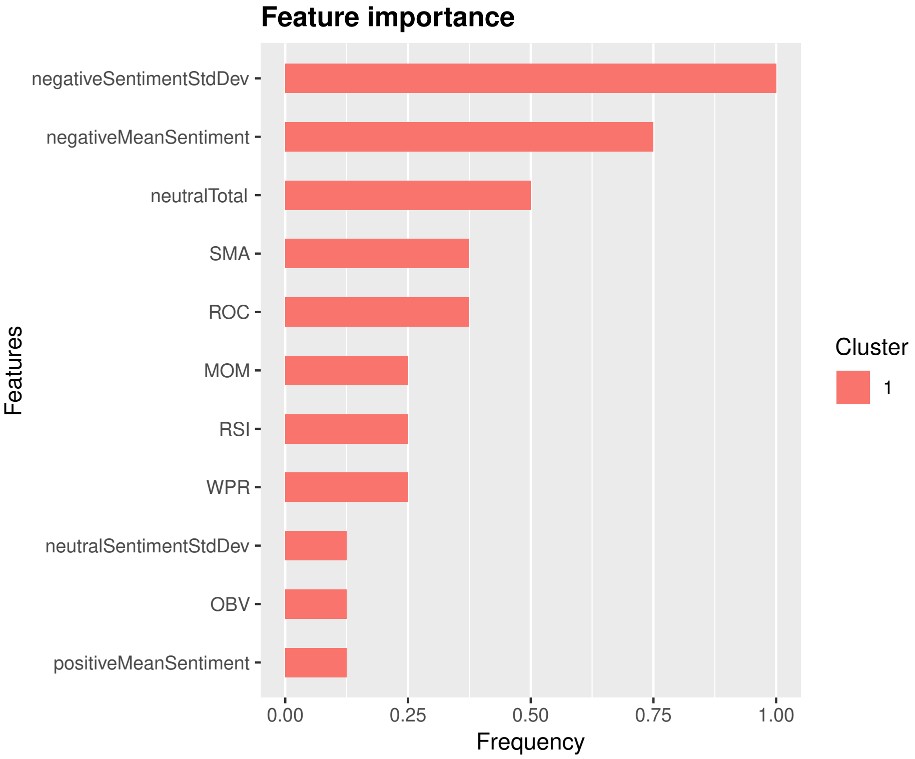


Figure A.29.: Feature Importance Ranking for Scenario 3 of the HSBC case study

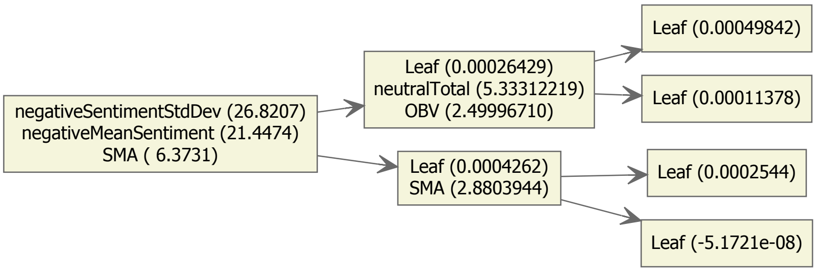


Figure A.30.: Example tree of the HSBC case study

A.2.7. Royal Dutch Shell

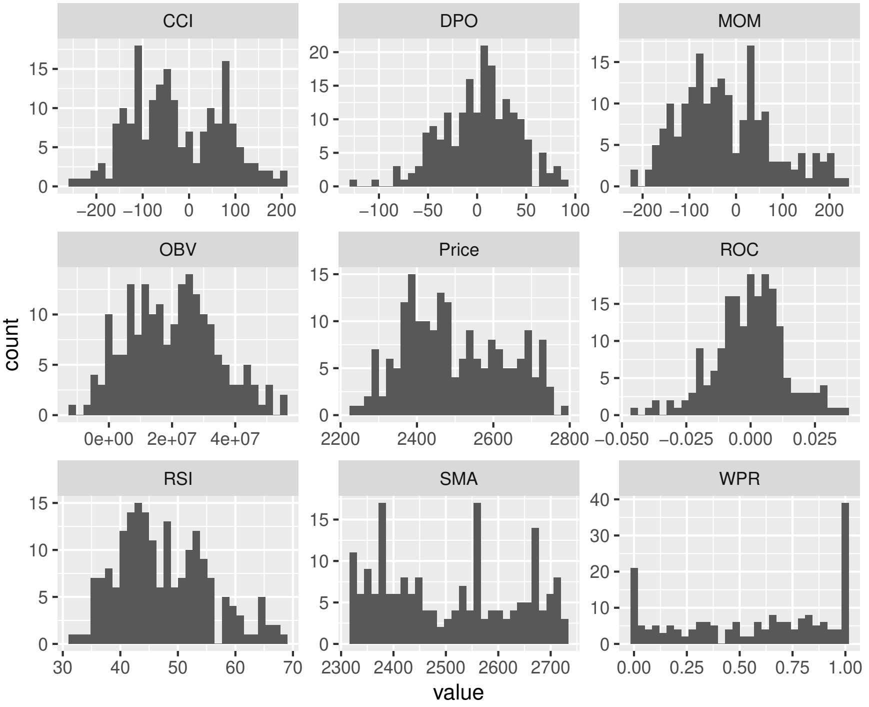


Figure A.31.: Histogram of the technical indicators of the Royal Dutch Shell

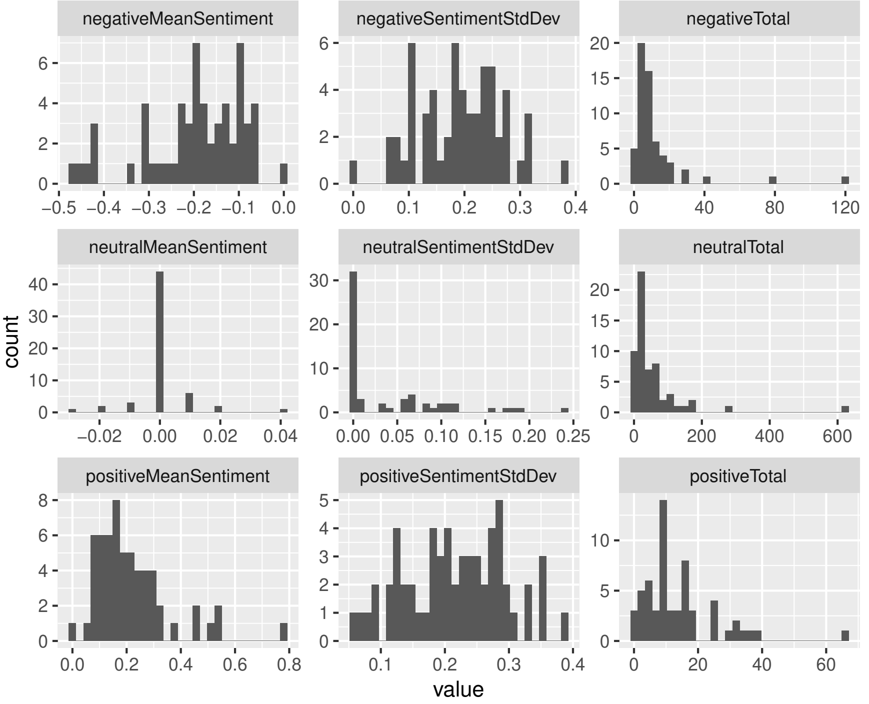


Figure A.32.: Histogram of the news features of the Royal Dutch Shell

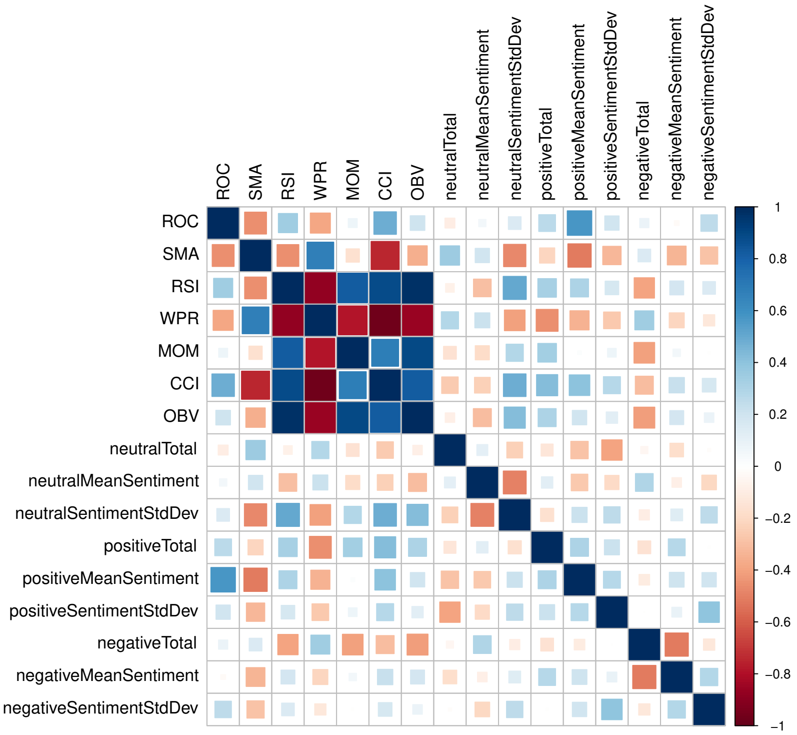


Figure A.33.: Correlation matrix of all features of the Royal Dutch Shell case study

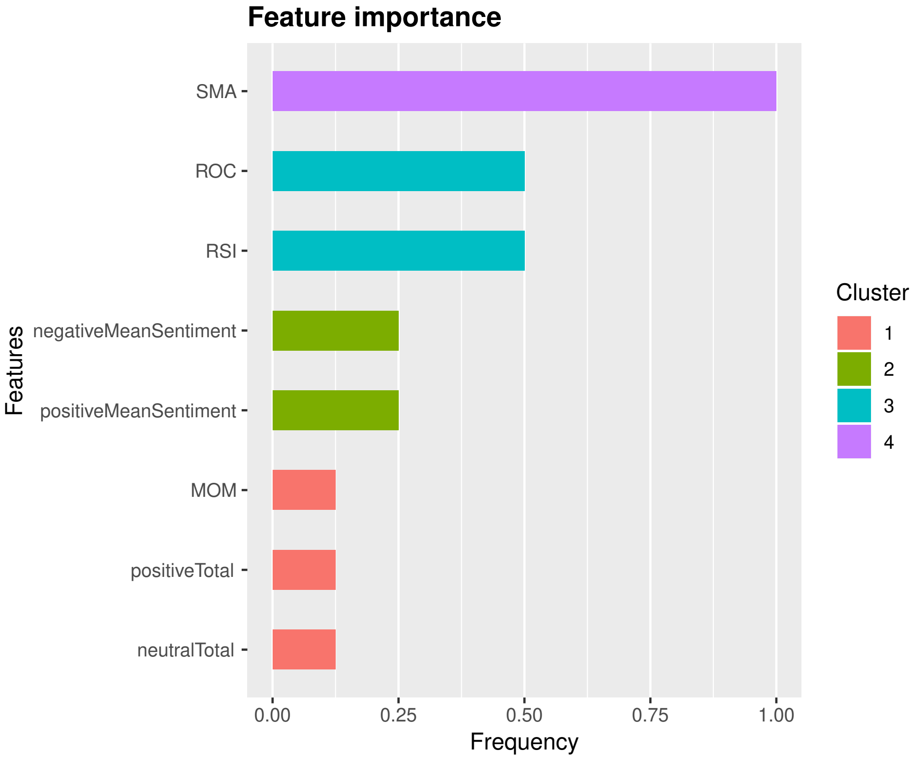


Figure A.34.: Feature Importance Ranking for Scenario 3 of the RDSB case study

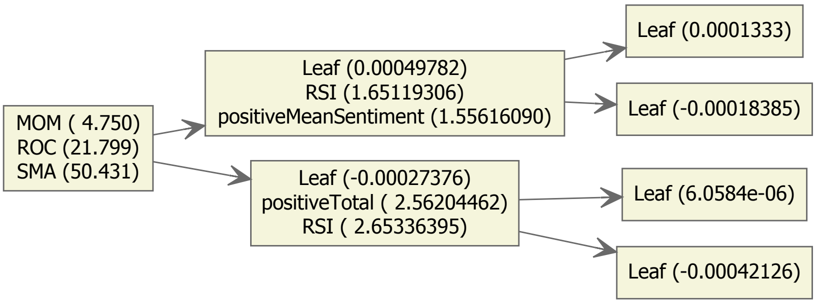


Figure A.35.: Example tree of the RDSB case study