Quantifying the Mutual Influence of Cryptocurrency Market Data, News Coverage and Social Media Texts

Wolfgang Thomas Eßl

MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

Interactive Media

in Hagenberg

im Juni 2019

Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere.

Hagenberg, June 25, 2019

Wolfgang Thomas Eßl

Contents

Preface

Writing a preface at the end of a long time working on the present thesis is probably the most rewarding part of the work. Looking back for a moment and thanking those who made this work and the associated Master's degree possible. My first thanks go to my supervisor FH-Prof. Mag. DI Dr. Andreas Stöckl, who improved the work considerably with his very valuable and constructive input.

My deep personal thanks go to my family, before all others my girlfriend Romana Prommer for her unbelievable support, through thick and thin. To my sister Prof. Mag. Christina Eßl, who improved the work linguistically with her excellent knowledge of editing. Many thanks to my parents and my brother for their support in all matters and putting up with me. Special thanks to my dear colleague Florian Weinrich, who supported me with his DevOps skills in setting up the deployment and the server.

Abstract

The market development of crypto currencies is subject to a large influence of posts on social media and reporting on specialized news platforms. In particular, messages from influencers on Twitter seem to influence the market. It seems impossible for people to gain an overview of this influence because of the sheer mass of posts and news. Automatic classification of topics and the measurement of sentiment values are therefore increasingly used in the recent past to measure this influence.

While many of these projects were primarily developed to predict the development of the share price, the present thesis will examine whether and how strong a correlation is between the sentiment value of Twitter messages and the volume of cryptorelevant news and the market development of different cryptocurrencies. In the search for significant correlation values, different techniques for sentiment analysis are compared in their suitability. Both rule-based systems and well-established classification algorithms from machine learning are applied. These techniques are used for sentiment analysis to determine the mood in tweets of the 50 most influential cryptoinfluencers. For the automated recognition of hidden text structures, more precisely topics, probabilistic topic models are distributed in the News. This technology is applied to a data set of news articles consisting of over 80,000 entries from the most influential news platforms on crypto currencies. The results of these calculations are then summed up per day and category and compared with different metrics of the market development of some cryptocurrencies by calculating the correlation coefficient according to Pearson's r.

The results of this calculation suggest that the volume of tweets and the volume of messages in particular show a statistically significant correlation to various market metrics. Although the correlation values of the sentiment values are striking, they differ only slightly from each other and are in total lower than the correlation values of the total volume of tweets per day to market metrics. A special feature of reporting on cryptocurrency is the large common vocabulary, which makes it difficult for the Topic Models used to precisely classify the articles by topic, especially for the topics "Bitcoin" or "Blockchain". In order to improve this analysis possibility, an optimization process is introduced to improve a GuidedLDA algorithm with a list of relevant topic-word assignments.

Crypto currencies and their ecosystem are subject to very rapid change. The technologies used in this master thesis and the approach make it possible to keep track of relevant topics, market sentiment and future developments.

Kurzfassung

Die Marktentwicklung von Kryptowährungen unterliegt einem großem Einfluss von Posts auf Social Media und der Berichterstattung auf spezialisierten Newsplattformen. Besonders scheinen Nachrichten von Influencern auf Twitter den Markt zu beeinflussen. Einen Überblick über diesen Einfluss zu gewinnen scheint ob der schieren Masse an Posts und News für Menschen unmöglich. Automatische Klassifizierung von Themen bzw. die Messung von Sentimentwerten werden deshalb in der jüngsten Vergangenheit immer mehr zur Messung dieses Einflusses herangezogen.

Während viele dieser Projekte vor allem zur vermeintlichen Vorhersage der Kursentwicklung entwickelt wurden, erarbeitet die vorliegende Master Thesis ob und wie stark eine Korrelation zwischen dem Sentimentwert von Twitternachrichten und dem Volumen von kryptorelevanten News und der Marktentwicklung verschiedener Kryptowährungen ist. Auf der Suche nach signifikanten Korrelationswerten werden verschiedene Techniken zur Sentiment Analyse in ihrer Tauglichkeit verglichen. Dabei kommen sowohl regelbasierte Systeme als auch anerkannte Klassifizierungsalgorithmen aus dem Machine Learning zum Einsatz. Diese Techniken werden zur Sentiment Analyse für die Bestimmung der Stimmungslage in Tweets der 50 einflussreichsten Kryptoinfluencer verwendet. Für die automatisierte Erkennung von versteckten Textstrukturen, genauer gesagt Themen, in den News kommen probabilistische Topic Models zum Einsatz. Diese Technolgie wird auf einen Datensatz aus Newsbeiträgen angewandt, der aus über 80.000 Einträgen der einflussreichsten Newsplattformen über Kryptowährungen besteht. Die Ergebnisse dieser Berechnungen werden dann pro Tag in ihrer Kategorie aufsummiert und mittels Berechnung des Korrelationskoeffizienten nach Pearson mit verschiedenen Metriken der Marktentwicklung von einigen Kryptowährungen verglichen.

Die Ergebnisse dieser Berechnung lassen darauf schliessen, das vor allem das Volumen an Tweets und das Volumen an Nachrichten eine statistisch signifikante Korrelation zu verschiedenen Marktmetriken aufweist. Eine Besonderheit der Berichterstattung über Kryptowährung stellt das große gemeinsame Vokabular heraus, welches die verwendeten Topic Models insbesondere bei den Themen „Bitcoin“ oder „Blockchain“ vor Schwierigkeiten in der genauen Themenzuordnung der Artikel stellte. Um diese Analysemöglichkeit zu verbessern, wurde ein Optimierungsprozess eingeführt mit dem ein GuidedLDA Algorithmus mit einer Liste von relevanten Themen-Wörter Zuordnungen verbessert werden kann.

Kryptowährungen und ihr Ökosystem sind einem sehr schnellen Wandel unterworfen. Die in dieser Master Thesis verwendeten Technologien und die Vorgangsweise machen es möglich einen Überblick über relevante Themen, die Stimmung auf dem Markt und die zukünftige Entwicklungen zu behalten.

Chapter 1

Introduction

1.1 Motivation

Cryptocurrencies continue to enjoy the interest of investors and of people who are interested in blockchain technology. Due to the volatility of this high risk investments, the market is a particular area of tension between high profits, high losses and therefore emotional influence on investment behaviour. It is precisely for this reason that the prices of cryptocurrencies seem to be influenced, especially by news coverage and social media texts. Automated analysis tools for market sentiment are already widely used, for example in stock market trading. These tools provide valuable insight into opinion changes that significantly affect the market.

The similarity to the stock market as well as the fear of high losses or the fear of high profits make the cryptocurrency market a particularly interesting field of application for semantic text analysis. There are already projects in the business world, such as BISON's Cryptoradar [43], which use automated semantic text analysis as a basis for their applications. The BISON Cryptoradar project crawls and analyses about 250,000 tweets per day from the crypto community to find out the market sentiment and frequency of mention concerning Bitcoin, Ethereum, Litecoin and Ripple.

In 2015 a research team from the United States found a connection between the number of forum postings on the subject of Bitcoin and its performance on crypto exchanges [25]. Their study shows that more bullish[1] forum posts are associated with higher future Bitcoin values. An interesting finding of their study is that the effect of social media on the performance of Bitcoin is driven by the silent majority. The 95 percent of users who were less active and whose contributions accounted for less than 40 percent of all posts. The team also found that forum posts have a higher impact in comparison to tweets.

Since 2015 much has happended in the field of cryptocurrencies. The number of news sites with specialised crypto posts and crypto entries as well as the number of crypto influencers on social media have increased very steeply. Both parts also have a higher impact on investment behavior than never before. This is due to the greater public access to cryptocurrency and more crypto-affine investors but also has its roots in more interest in crypto currencies or the underlying technology of blockchain in general.

[1] *bullish* – description of the market condition characterised by rising share prices

For these reasons, crypto news and tweets from the cryptosphere are selected for further analysis within this thesis. The present thesis examines the question of whether it is possible to quantify the mutual influence of cryptocurrency trading volume, news coverage and social media texts. The correlation between the volume of news articles or sentiment value of tweets about cryptocurrencies and the market development of this cryptocurrencies is utilized as basis for assessment.

1.2 Research Questions

1.2.1 First Results

First results of the technical analysis show that there is a slight positive correlation between the pure number of tweets on the subject of cryptocurrencies per day to the 24h trading volume of Bitcoin. This circumstance led to further research in probabilistic topic models, which are capable of learning topics from words in a text and furthermore computing which topic each document is about. A research article from cryptocompare.org [49] covers this topic with a focus on a technique called guidedLDA and shows possible correlation to the price development of Bitcoin. This approach seems also appropriate for further analysis of the mutual influence of trading volume and news coverage. Given these results and findings lead the research in probabilistic topic models are ammended to this thesis as well as the accompanying thesis project.

1.2.2 Research Questions

Based on this results two research questions are discussed in this thesis. The first one covers the approach for finding out correlating data with sentiment analysis and probabilistic topic models. The second research question covers the optimization of a guided approach of a Latent Dirichlet Allocation approach.

- How can the mutual influence of cryptocurrency market development, news coverage and social media text be quantified using sentiment analysis as well as probabilistic models techniques?
- How can probabilistic topic models be utilized to find underlying topics and the most important associated words from unlabeled text data in order to obtain the most accurate recognition of the topic of a text?

1.3 Structure

This thesis is structured as follows. Chapter 2 summarizes the fundamental concepts of Machine Learning, Natural Language Processing and Sentiment Analysis and provides a summary of technical terms needed to understand this thesis. Chapter 3 dives into the development of sentiment analysis, presents a selection of previous research in sentiment analysis and probabilistic topic models along with describing the peculiarities and interplay between these two technologies. In the fourth Chapter, the concepts used are presented in depth, compared and explained in terms of their functionality and tasks. Furthermore, there is an insight into the used tools, programming languages and frameworks will be presented. The own solution approach can be found in Chapter 5 which

explains the data mining part, first results of sentiment analysis on twitter data up to the adaptation of the probabilistic topic models and the merging of all used concepts into a functioning real-time analysis web app. Chapter 6 gives a comprehensive insight in the correlation calculation and the optimization process of the models used and explains issues / problems that have emerged. The results are summarized and reflected in Chapter 7. Open issues and future work are also explained in the resuming Chapter.

Chapter 2

Fundamentals

2.1 Cryptocurrencies and Blockchain

The term *cryptocurrency* describes in the narrower sense, a currency that is secured by means of cryptographic methods. The first working cryptocurrency is Bitcoin. Cryptocurrencies consist of three elements: a piece of software that establishes the rules of cryptocurrency, a database (in this case a blockchain), and a decentralized network that operates the database and periodically complements the rules of the computer program. These cryptocurrencies are traded on crypto currency exchanges similar to forex trading [33].

The aforementioned blockchain is a chain of blocks which in turn combines several transactions for administrative reasons. A block is therefore a container in which transactions are stored. These blocks are concatenated by cryptographic techniques, thus providing the blockchain [33].

2.2 Machine Learning

Machine learning belongs to the field of artificial intelligence. The core task of machine learning is learning from data, so this technique is strongly linked to data mining and statistics. These techniques are mostly used to conduct decisions or find classifying features based on training data. Arthur Samuel, a pioneer of the early stages of machine learning, defined the term Machine Learning in 1959 as "field of study that gives computers the ability to learn without being explicitly programmed" [32].

In his book from 2017 Aurélien Géron breaks down the types of machine learning into three major areas [12]:

- whether or not they are trained with human supervision(supervised, unsupervised, semisupervised, and reinforcement learning),
- whether or not they can learn incrementally on the fly (online versus batch learning) and
- whether they work by simply comparing new data points to known data points, or instead detect patterns in the training data and build a predictive model, much like scientists do (instance-based versus model-based learning).

Because this thesis mainly uses techniques from the supervised learning or from the unsupervised learning area, only these two areas will be discussed in this chapter. *Supervised learning* describes the process that a prediction model is fed with input variables, for which the corresponding output variables are known. If the result is a class or category (e.g., positive/negative sentiment) the task is called a *classification task*. If real values are delivered as a result, for example *bitcoins* or *percent*, these values are the result of a regression task. In contrast to supervised learning, the unsupervised method has no predefined results that could be used for training. *Unsupervised learning* describes machine learning techniques that are able to detect underlying patterns in the data.

2.3 Natural Language Processing

Natural Language Processing belongs to the fields of Computer Science, Artificial Intelligence and Computational Linguistics. It is formed from the broad application areas of automatic speech recognition, understanding and interpreting natural language and natural language generation. The advances in NLP in the recent years and the increasing use of NLP in all-day software, are mainly due to advances in machine learning. While early NLP techniques were based on human-made rules, machine learning based algorithms nowadays usually provide low-maintenance and self-learning natural language processing applications.

These applications range from automatic spam detection, over topic-based email filtering, machine translation and smart assistant systems (e.g., Amazon Alexa, Apple Siri, Windows Cordana) to the freely implementable solution as NLP-API (e.g., api.ai).

2.4 Sentiment Analysis

Sentiment analysis or also called *opinion mining* and its techniques belong to the broad field of Natural Language Processing. With this technology it is possible to extract and identify meaning and opinion within unstructured text data. Alike many other problems of the field of *NLP*, opinion mining can be modeled as *classification problem*. This problem can be further subdivided in *subjectivity classification* and *polarity classification*. While polarity classification classifies a sentence in positive, neutral or negative opinion, the term subjectivity classification refers to classifying a sentence as subjective or objective. This classification problem is split in three levels of scope – *Document*, *Sentence* and *sub-sentence* level. Document level analysis sentiment of a complete document or paragraph. Sentence and Sub-sentence Level obtain the sentiment of a sentence or lone expressions. A very well written and received survey book about sentiment analysis was published by Bing Liu in 2012 [23].

2.5 Probabilistic Topic Models

Probabilistic Topic Models are also part of the field of Natural Language Processing and describe statistical models that make it possible to find abstract topics in a collection of documents. This technique is often used to automatically detect hidden semantic struc-

tures in texts. They are among the unsupervised learning techniques and one of their most important implementations is the LDA technique (Latent Dirichlet Allocation) [3].

2.6 Terminology

- *Bitcoin* – refers to the first functioning cryptocurrency as well as the associated payment system. In contrast to e.g., Euro, which designates only the currency, while SWIFT, PayPal or VISA designate payment systems. In the crypto field, there are two common spellings: "bitcoins" denote the currency, "Bitcoin" describes the payment system.
- *Wallet* – is a software that takes over the function of a virtual wallet that stores the associated address and the secret key.
- *Altcoin* – are all crypto currencies other than Bitcoin. There are thousands of them, many of which are very short-lived. The reason for the emergence of Altcoins are projects that want to mend perceived weaknesses of Bitcoin.
- *Token* – refers to a type of cryptocurrency that does not exist as pure means of payment (such as Bitcoin, Litecoin, ...) but stores assets on a blockchain.
- *Corpus* – defines a collection of texts, that are machine readable and stem from natural communicative settings. A corpus can be composed of blog posts, news articles, politic statements, product reviews, etc.
- *Sentiment* – is a point of view or opinion that is held or expressed. Polarity, Subject and the Holder of the view or opinion are the three attributes of such an expression.
- *Precision* – is the unit of measurement of how many texts are categorized correctly of the total number of texts predicted as belonging to this category.
- *Recall* – measures the number of texts which were categorized correctly as belonging to a given category, of all texts that should have been categorized as belonging to the category. The more data is fed to the classifier the better the recall will be.
- *Accuracy* – measures how many texts were predicted correctly (both as belonging to a category and not belonging to the category) out of all of the texts in the dataset.
- *Feature* – is an individual, measurable property or characteristic of an input. Features can be discrete, continuous or categorical values.
- *Feature vector* – is a list of numerical features, extracted from one sample. A feature vector is represented as an one-dimensional matrix.
- *Feature extraction* – describes the process of converting the characteristics of the raw input data into numerical values. This process is needed, because machine learning models cannot utilize raw textual data but rather a matrix with numeric values.
- *Feature engineering* – In order to make features more meaningful within the feature extraction a process called feature engineering is necessary. This detail requires manual intervention and expertise in the topic addressed by the machine learning task.

- *Bag-of-words* – A Bag-of-words model is a simplified, orderless representation of a text or sentence used in *NLP*. For example, a sentence like 'The brown fox jumps over the lazy dog over and over again' would be represented like: "The", "brown", "fox", "jumps", "over", "the", "lazy", "dog", "over", "and", "over", "again". Classifying algorithms often make use of the frequency of words as a feature for training. The Bag-of-words model if often used for this approach.
- *nGram* – The Bag-of-words model can be defined as a nGram, where $n = 1$ – only the word count is relevant. A bigram for example is a nGram, where $n = 2$. A bigram would represent the aforementioned sentence as: "The brown", "fox jumps", "over the", "lazy dog", "over and", "over again".

Chapter 3

State of the Art

Artificial intelligence is currently experiencing a high upswing in public perception. The reason for that is not only the almost unlimited computing power of modern computer systems but also a high number of application possibilities. In addition to systems for autonomous mobility or automation of everyday life, the special focus is on the field of Natural Language Processing. This is mainly due to the large amounts of text data that are produced day after day on social media. With sentiment analysis and Probabilistic Topic Models, this Chapter takes up two important parts of this field, explores their past and examines their current state of research.

3.1 Sentiment analysis

3.1.1 Basic introduction

Sentiment analysis is also often referred to as *opinion mining*. Even though these are two terms, the meaning of both overlaps for the most part. Both terms are used to describe the analysis of people's opinions, sentiment or feelings in relation to things such as products, news, events or companies. In his very well received Book about sentiment analysis [23] Bing Liu states that the currently strong interest in sentiment analysis stems from the strong increase in the presence of different opinions, especially through social media. The author shows that before the year 2000 very little research was done on the matter of sentiment analysis. Mika Mäntylä et al. [26] published a research paper on the development of sentiment analysis in 2018. Thereby he analyzed almost 7,000 papers and found that 99% of the papers on this matter were published after 2004. He also figured out that the focus of the research has started with online product reviews with a shift to social media in recent years.

Alike many other problems of the broad field of *NLP*, opinion mining can be modeled as classification problem. This problem can be further subdivided in *subjectivity classification* and *polarity classification*. While polarity classification classifies a sentence in positive, neutral or negative opinion, the term subjectivity classification refers to classifying a sentence as subjective or objective. This classification problem is split in three levels of scope. Document level analysis sentiment of a complete document or paragraph. Sentence and Sub-sentence Level obtain the sentiment of a sentence or expression. The theoretical and technical focus of the present thesis and approach is a

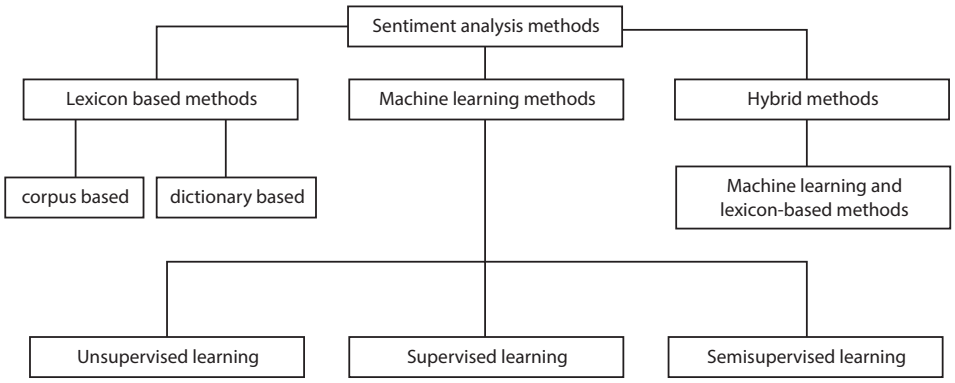


Figure 3.1: Sentiment analysis methods based on their functionality.

polarity classification on document level.

Since there is a considerable amount of techniques for classification tasks in general, this Section will give a short overview about well established classification approaches and algorithms especially suited for sentiment analysis on document level with focus on social media texts. sentiment analysis techniques have been categorized in how they are are executing their task based on their architecture – visible in Figure 3.1. The main categories are rule-based, automatic and hybrid approaches, which represent the structure of this Chapter as well.

3.1.2 Rule-based approaches

Rule-based systems rely on a hand-crafted set of rules in a programming language, which are able to identify and extract polarity, subject and the opinion holder. Such systems use typical *NLP* techniques like part-of-speech tagging, parsing, stemming or other resources, such as lexicons. These lexicons are usually lists of words or expressions with their corresponding sentiment intensity.

One of the most cited and used concepts for rule-based analysis is called *VADER* [16], which stands for *Valence Aware Dictionary for sEntiment Reasoning*. This concept was introduced by C.J.Hutto and Eric Gilbert of the Georgia Institute of Technology in 2014. It is a simple rule-based model for general sentiment analysis. In their paper Hutto and Gilbert are presenting their concept of an empirically validated gold-standard list of lexical features associated by sentiment intensity measures while attuning the features to sentiment in microblog-like contexts. This combination leads to a better performance to social media content. This project does not only outperform human raters on sentiment data but also generalizes better over many domains. Since the publication of this system, many research projects made use of it for the sentiment analysis of social media texts, which will be discussed later in this Chapter. As *VADER* was also used in the thesis project to analyze Twitter data, the technical functionality and structure will be discussed in more detail in the next Chapter.

This VADER approach represents a viable alternative to longer existing solutions

like *LIWC* [38], *General Inquirer* [36] or *SentiWordNet* [1], incorporates some of their used concepts and also outperforms them in tests as Hutto and Gilbert state in their paper. Since these three approaches are the three best known methods of sentiment lexicons and are representatives of both polarity-based and valence-based approaches. They are now briefly described in more detail.

General Inquirer [36] stands out as one of the first automated content analysis dictionaries. The approach was published by Philipp Stone in 1966 and is still regarded today as a state-of-the-art benchmark for sentiment analysis lexicons. Initially planed as tool for content analysis to recognize characteristics in messages and assign messages to specific categories based on these characteristics, the manually constructed lexicon consists of 183 categories with 11,000 corresponding words. Although the Lexicon has nearly 4,200 words (1,915 negative / 2,291 positive) categorized as positive or negative in the Lexicon, the General Inquirer has no way to measure or categorize the intensity of the sentiment, as opposed to the VADER approach.

LIWC [38] is the abbreviation of *Linguistic Inquiry and Word Count*, an approach initially presented in 1994. Since then it was revised in 1997 and 2007. This well-established approach has its roots in psychological research and focuses on emotional, structural and process-oriented components of texts. Because of this specific focus this concept is widely used for sentiment polarity extraction, for example extracting political sentiment from tweets [39] or measuring the happiness of a nation based on facebook statuses [10]. Hutto and Gilbert state that – alike General Inquirer – LIWC has no way to account differences in sentiment intensity of words.

SentiwordNet [1], which was published by Baccianella et al. in 2010, was built on an concept called *WordNet* [8]. WordNet was developed by Fellbaum in 1998 and consists of so-called *synsets*. The word *synset* describes a group of words that are synonyms. *SentiwordNet* consists of 147,306 of those synsets, annotated with their corresponding sentiment score in positive, negative and objectivety. The scores of this collection of *synsets* were computed by a mix of semi-supervised approaches, consisting of classifying algorithms and propagation methods. This is also a major distinguishing feature of this algorithm, because all other methods mentioned beforehand where created and classified by humans. *SentiwordNet* is also available in the reknown Python package NLTK.

Whilst requiring no training data sets and being suitable for fast usage on large data sets, rule-based systems have also considerable downsides. The time-consuming construction and in addition the cumbersome mostly non-automatic maintenance of such a rule-based system is a fundamental challenge for researchers and developers.

3.1.3 Automatic Systems

In contrast to rule-based systems, automatic systems are using established machine learning techniques. These systems are also called corpus-based systems. Classification is one of the first big use cases approached by machine learning, therefore a vast number of different classification techniques is available. After being fed with a text, a classifying algorithm returns a corresponding category (e.g., in sentiment analysis positive, negative, neutral).

Classifiers are often designed as supervised learning architectures consisting of training data (labeled by the corresponding category), various kinds of feature extractors and

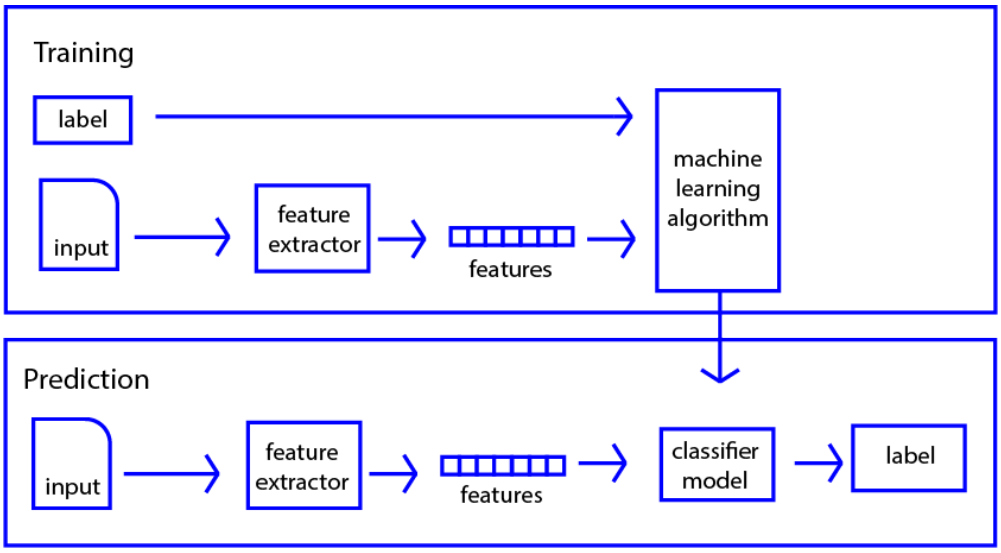


Figure 3.2: Steps and components machine learning classifier [52].

a machine learning algorithm (e.g., Naive Bayes, Logistic or Linear Regression, Support Vector Machines). This architecture learns to link a specific input (e.g., text) to a corresponding label or tag based on the samples in the training data. To prevent the machine learning algorithm from over-fitting the training data is split up in two parts, where one part is used for training and the other part is used for testing. The main features used for training the machine learning models are Bag-of-words, n-gram models (bi- and trigram) as well as polarity. A well structured introduction to automatic systems can be found on the website of MonkeyLearn, a company specialized on sentiment analysis software [52], and is also visible in Figure 3.2.

One of the best known examples of polarity classification on document level is the task of classifying movie reviews. This task proves to be a good testing opportunity for sentiment analysis for more than a decade and also has strong parallels with today's social media texts. Bo Pang and Lillian Lee of the Cornell University proposed a – at the time novel – machine learning approach using *Support Vector Machines* on the subjective parts of the text [28], thereby combining sentence-level subjectivity detection with document-level polarity detection. Further this approach is a method introduced by Kennedy and Inkpen in 2006 [20], which in a first step uses a rule-based approach (General Inquirer) to obtain positive and negative terms as well as negations, intensifiers and diminishers. As a second step their approach also makes use of *Support Vector machines* with unigram features in order to get sentiment polarity. By using both techniques, their algorithm has performed much better than the General Inquirer alone, as well as the Support Vector machine alone. Since then, many projects, similar to the combined attempt of Kennedy and Inkpen, have specialized in textual subtleties such as emojis, slang expressions and other linguistic peculiarities. Since then Naive Bayes and Support Vector machines have proven as the technologies with the best outcomes in numerous projects of document-level sentiment analysis [2].

In recent years the sharp increase in algorithms using deep learning methods has also obtained sentiment analysis. In their relatively new survey on Deep Learning Techniques for sentiment analysis, which has been published in 2018, Bing Liu et al. have compile numerous relevant research projects and add an excellent overview of current projects [41]. Their survey shows, that when it comes to the focus of this thesis on document-level sentiment analysis many projects make use of an *LSTM*[1] model [42] [40] or *Convolutional Neural Networks* [18] [37] (mostly as *BagOfWords-CNN*).

3.1.4 Hybrid systems and other approaches

sentiment analysis systems that use machine-learning and lexicon or rule-based systems in combination are also called Hybrid Models. Using the best of both approaches proves to improve accuracy and prediction of the analysis. Examples of these hybrid techniques include many combinations of well-established techniques but also introduce new methods. Ohana et al. proposed the combination of the *SentiWordNet* [49] as a source of features while in combination with a support vector machine. Their results suggest that the utilization of *SentiWordNet* as feature source for supervised learning approach leads to an improvement over the sole term counting approach [27]. In 2013, Ghiassi et al. published a hybrid approach consisting of n-gram analysis and dynamic neural networks. In this attempt, the authors tried to supplement a special lexicon for Twitter with brand-specific expressions in order to achieve better analysis results concerning consumer sentiment towards a brand. In combination with a *DAN2* – a dynamic architecture for neural networks – they achieved better results than using *SVM* [13]. One of the more recent approaches regarding the technical focus of this thesis is the hybrid cuckoo search method for sentiment analysis of Twitter messages proposed by Pandey et al. in 2017 [4]. The proposed algorithm is a novel metaheuristic method based on k-means and cuckoo search and belongs to the field of clustering based approaches. Their first experimental results outperform existing methods like particle swarm optimization, differential evolution and cuckoo search itself.

3.1.5 Approaches in the cryptocurrency space

The present thesis focuses on a very specific space in the whole sentiment analysis sphere. In the field of cryptocurrenies, sentiment is one of the strongest influencing factor on the investment decision of the acting persons. The fear of high losses and the fear of missing out on great gains drives up the price or lets it fall. Cryptocurrency trading and the established Forex trading are very much alike. Therefore the analytic and forecasting tools resemble each other. The abovementioned approaches in sentiment analysis are widely used in stock market analysis and prediction as well as in the cryptocurrency sphere. This Section explains the most important projects, draws comparisons and highlights findings that have had an influence on the own approach.

One circumstance that almost all sentiment analysis projects in the crypto sector have in common is that they are primarily interested in predicting price developments and market reactions. To a large extent, the projects use classic sentiment analysis techniques to gain interesting insights into market behavior. One of the first projects that

[1] *LSTM* – Long Short Term Memory – an artificial recurrent neural network architecture [14]

analyzed the exchange rate of the leading currency Bitcoin, however, uses an algorithm for pattern recognition. Shah et al. published in 2014 [34], a year when cryptocurrencies was far from the hype of recent days. The authors used the Bayesian regression algorithm [5] developed jointly with Nikolov and Chen. The utilization of this algorithm to predict the price of Bitcoin enabled them to develop a trading strategy that doubled their investment within 60 days. While this result gives interesting insights on market development, investment decisions solely based on pattern recognition do not promise to succeed in the long run as a successful method. Due to the disregard of the high influence of the market sentiment on the investment decision of the masses, the method probably would not perform well in special situations like the hype beginning or the bear market at the end of 2018.

The project of Georgoula et al. from the University of Athens is more suitable for assessing such situations [11]. The authors are introducing an analysis method that incorporates economical and technological factors and sentiment analysis on Bitcoin-related Twitter messages. Besides finding positive correlation between the sentiment on Twitter and the Bitcoin price, they also found out that the number of search queries related to Bitcoin on Wikipedia as well as the hash rate[2] have a positive effect on the price of bitcoins. For sentiment analysis they gathered about 2.2 million tweets over a period of 78 days, focused on tweets with keywords like “Bitcoin”, “BTC”, “Bitcoins” etc. They applied *SVM* on manually labelled tweets and attain an *F-Score* of 0.944 on their classification. This project comes closest to the focus of the present thesis and shows parallels especially in the inclusion of further factors besides sentiment in the analysis and evaluation of the price development.

One of the first projects to use and compare different supervised learning algorithms was published by Colianni et al. in 2015 [6]. The authors used Naive Bayes, Logistic Regression and Support Vector machines to construct an algorithm that predicts whether the Bitcoin price will rise or fall. Over a period of 21 days, they collected over 1 million tweets on Bitcoin, pre-processed them, and evaluated word for word for their sentiment using an API [60]. The positive, negative or neutral label for each tweet was used as feature vectors. Their results show that Logisitic Regression exceeds with accuracy of 98,58% on the hour-to-hour change of Bitcoin price.

Based on the approach of Colianni et al. Stenquist and Lönnö [35] they continued the idea and present a naive prediction algorithm based on sentiment to Bitcoin from Twitter messages in their research project published in 2017. Their project tries to establish a positive correlation between the sentiment and the Bitcoin price and investigates whether a naive prediction algorithm produces better results than random. In their implementation, the authors use the previously mentioned rule-based model *VADER* [16] to identify polarity on about 2.21 million tweets concerning the keywords “Bitcoin”, “BTC”, “XBT”, “Satoshi”. The sentiment values returned by VADER are then grouped into time-series. After the mean sentiment of such an interval is calculated, the shifts in opinion are measured by calculating the difference between neighbouring periods. These shifts in opinion – positive shift = 1, negative shift = 0 – are taken as a prediction for price development. These predictions are then filtered by thresholds, returning the final prediction. An interesting matter on this project is the weaknesses mentioned by

[2] *hash rate* – is the measuring unit of mining performance of bitcoins.

the authors. Besides the static threshold, having no indication of success and lack of sufficient data, they mention that *VADER* does have a multiusable lexicon but the cryptocurrency sphere does have very specific expressions, that are not captured and therefore miscalculated by the rule-based model.

Lamon, Nielsen and Redondo introduced a particularly interesting approach also in 2017. They gathered about 3,600 cryptocurrency-related news article headlines as well as 10,000 tweets for each cryptocurrency (Bitcoin, Ethereum and Litecoin) in their project. Then they utilized traditional classification algorithms for sentiment analysis but used the actual price change of the corresponding coins one day in the future as labels instead of the sentiment polarity values. All texts were tokenized. The feature extraction was done by the scikit-learn *CountVectorizer*. The authors used the classification algorithms Logistic Regression, Linear Support Vector Machine and Naive Bayes. With this approaches the authors are able to predict the days with the largest percent increases and percent decreases in price for Bitcoin and Ethereum. Logistic regression performed best on the Bitcoin and the Litecoin Dataset, wheres Bernoulli Naive Bayes transcended them on the Ethereum dataset.

3.1.6 Challenges

First of all, detection and interpretation of the opinion/sentiment of a text is even for humans a difficult task. The recognition of sentiment polarity is in itself a very subjective task and can be compared to the recognition of sarcasm or humour.

Bing Liu discusses some of the main problems in sentiment analysis in his book and recognizes a very essential point – opinions are subjective [23]. He defines sentence subjectivity as expression of personal feelings, views and believes, whilst stating that an objective sentence presents some factual information about the world. In making this distinction, he points out that subjective statements on the one hand may not contain sentiment, while objective statements on the other hand can contain sentiment.

Alike Bing many researchers have found critical points in their research on sentiment/opinion analysis that make automated sentiment analysis difficult or even influential. Especially in supervised learning approaches, where one is often dependent on manual labelling by people, subjective points of view come into play, since each person evaluates or recognizes the polarity of sentiment for him- or herself differently.

A metric with which this subjectivity can be measured in one form is the Krippendorff Alpha value[3].

According to Saif et al. [31], the Krippendorff Alpha coefficient for sentiment detection on a tweet-level annotation task by humans is 0.765. Saif et al. gathered this data for their research and creation of a new twitter sentiment dataset called "STS-Gold". For their research they asked three graduate students to manually label 3,000 tweets with one of five classes. Krippendorff himself stated in his paper, published in 2004, that social scientists "commonly rely on data with reliabilities $\alpha \geqslant 0.800$, consider data with $0.800 > \alpha \geqslant 0.667$ only to draw tentative conclusions, and discard data whose agreement measures $\alpha < 0.667$." [22].

[3] *Krippendorff's Alpha* is a statistical metric. In matters of sentiment analysis this measure is used for the calculation of inter-annotator agreement which shows how much people agree in their execution of a given annotation task. [21]

Sentiment analysis was therefore naturally a challenging field of *NLP*. It became an even more difficult matter of research with the introduction of emojis, the usage of abbreviations and slang words, especially on social media platforms. MonkeyLearn summarizes the challenges and limitations of sentiment analysis on their introductory site [52] and states the following points as main problems:

- subjectivity and tone,
- context and polarity,
- irony and sarcasm,
- comparisons,
- emojis and
- the definition of what is neutral.

These problems are approached in different ways in various research projects and are not only barriers but further starting points to make sentiment analysis even more applicable and accurate.

This Section has provided insight into the existing and most commonly used techniques for sentiment analysis. Special focus was placed on sentiment analysis on document level. Based on the results of the accompanying concept, which is described in Chapter 5, the next Section deals with topic models in more detail. The technical focus shifts from supervised learning approaches to unsupervised/clustering or semisupervised approaches.

3.2 Probabilistic topic models

One of the key findings of this thesis and project is that not only sentiment plays a vital role in the price development of cryptocurrencies, but above all the number and frequency of news and tweets about a particular cryptocurrency. In order to measure this frequency, suitable technical means are needed to assign the mass of news and tweets to certain topics, in this case cryptocurrencies or topics like mining or forks. Suitable technology can be found in the field of topic modelling. Topic models are algorithms to find topics in an unstructured collection of documents and to structure the collection according to these topics. First of all this Section gives an overview about the important concept in the development of topic models in order to explain basics and backgrounds. After that the Section deals with one of the most used concept *Latent Dirichlet Allocation*, summarises concepts based on this approach and gives insight into a further development of the LDA model to a semisupervised approach called *GuidedLDA*, which is used in the project as well.

3.2.1 Basic introduction to topic models

Topic modeling is often described as a process to identify latent (hidden) topics, which are present in a given text document or to optain information of unstructured data. A topic can be constructed of a specific combination of words. A topic can therefore be defined as “a repeating pattern of co-occurring terms in a corpus”. To give an example: “Satoshi”, “BTC”, “bitcoins”, “XBT”, “Nakamoto” – these words would make up the topic “Bitcoin”. Different from the supervised learning approaches for sentiment analysis

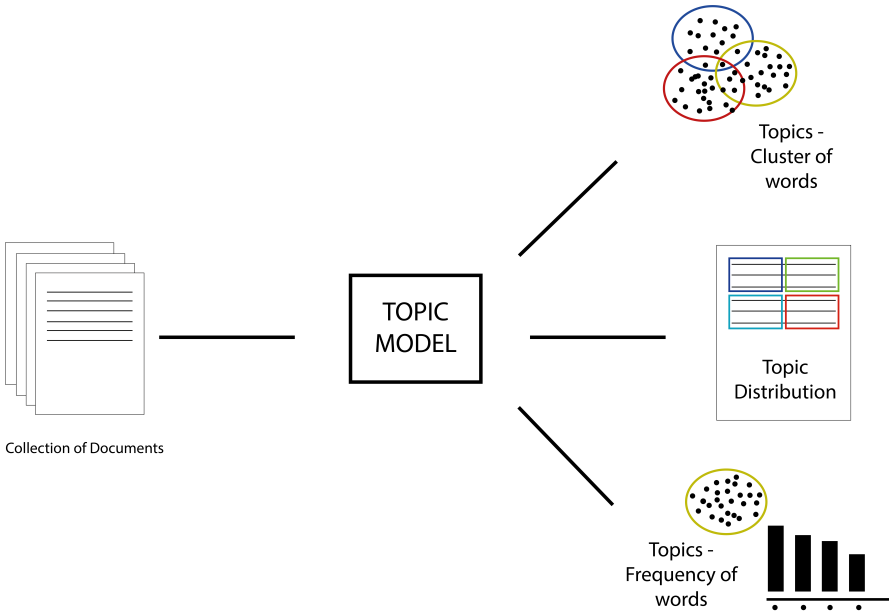


Figure 3.3: Basic sketch of components and functionality of a topic model [64].

which rely on labeled data, this technique is able to find meaningful patterns/topics in text data without additional labeling – also called unsupervised. Topic Models have a wide variety of application, for example feature selection, information retrieval, clustering of document or organizing large textual data. A basic sketch of the functionality of a topic model is depicted in Figure 3.3.

3.2.2 Important projects in information retrieval

The roots of today's state-of-the-art topic models can be found in the field of information retrieval. To provide an insight into the techniques before the groundbreaking publication of the Latent Dirichlet Allocation, three important concepts are explained in short: *tf-idf*, Latent Semantic Indexing (LSI) and probabilistic LSI.

One of the first concepts for hidden structure in textual data is the so called *tf-idf* scheme[4], which was developed over decades by groups of researchers. Term frequency(tf), a notion first introduced by Luhn in 1957 [24], measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear more often in long documents than in shorter ones. Therefore – for the matters of normalization – the term frequency is frequently divided by the length of the document. In a more mathematical notation: $TF(t) = (Numberoftimestermtappearsinadocument)/(Totalnumberoftermsinthedocument)$

Inverse Document Frequency (idf), first introduced by Spörck Jones in 1972 [19] measures the importance of a term in a document. For the calculation of TF all terms are considered as equally important. However, it is known that certain terms, also called

[4] *tf-idf* – term frequency – inverse document frequency weight. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus.

stopwords (e.g., and, is, the), may appear a lot of times but have little importance. It is necessary to weigh down these frequent conditions, as well as to weigh up unusual terms. So one could describe idf as the following formula $IDF(t) = log_e$(Total number of documents / Number of documents with term t in i). While *tf-idf* makes it possible to identify groups of words as distinguishing features of documents in a collection, it can only slightly reduce the length of a document's description and provides little information about other document structures.

One of the projects with the best solution for these problems is *latent semantic indexing* (LSI) introduced by Deerwester et al in 1990 [7]. The approach of this research team is based upon the *tf-idf* approach and achieves significant compression on a large collection of documents and also the identification of basic lingustic features like synonymy. This is achieved by an algorithm that decomposes the singular value of the X matrix in order to identify a linear subspace in the space of *tf-idf* features that captures most of the variance in a document connection as stated by Blei et al. [3].

The introduction of a probabilistic LSI (pLSI) model by Hofmann in 1999 [15] is held in high regard by Blei et al. in their paper. This approach models each word in a document as a sample from mixture model, where the mixture components are multinominal random variables that are representation of topics. Each document is represented as a list of mixing proportions for these mixture components and thereby reduced to a probability distribution on a fixed set of topics.

3.2.3 LDA – Latent Dirichlet Allocation

Latent Dirichlet Allocation is a generative probabilistic model of a corpus for calculating the probability of a word belonging to a topic. The technical background is to be found in the field of generative approaches to statistical classification. Further known concepts that are also classified as generative statistical models are e.g., Hidden Markov Models, Bayesian Networks (e.g., Naive Bayes) and Boltzmann machines. Because of its extensive use as a basis for further ideas, this model is often referred colloquially by the investigator as the "workhorse" of subject models. At the time of writing this thesis, Google Scholar had nearly 26,500 citations in scientific papers for the paper by Blei et al. It is therefore one of the most important and most cited works in the field of machine learning [57].

A similar concept was already proposed in the context of popular genetics in 2000 by Pritchard et al. [30]. Blei et al. introduced their concept in 2003 [3]. Their initial goal was to "find short descriptions of the members of a collection that enable efficient processing of large collections while preserving the essential statistical relationships that are useful for basic tasks such as classification, novelty detection, summarization, and similarity and relevance judgements."

The authors refer to documents as "probabiltity distributions over latent topics" and to topics as "probability distributions over words". In order to implement this idea in an algorithm, the researcher introduces two dirichlet priors (α and β) which are parameters for the per-document topic distribution (α) and the per-topic word distribution (β). With this adaption *LDA* tries to figure out the hidden structure or formula for the creation of a document. Since this technique is used in the thesis project its functionality is further described in detail in the next Chapter. Latent Dirichlet allocation is undoubtedly one of the most elaborate concepts of its kind, but it also

has its shortcomings. An example would be the procession of a corpus with substantial shared vocabulary distributed over topics.

3.2.4 GuidedLDA or SeededLDA

Since its introduction in 2003 the *LDA* approach has been inspiration for a vast number of topic model concepts. One of these concepts – called GuidedLDA – was used in 2018 by Ben Hunter-Craigh [49] in order to implement a topic model for identifying topics in cryptonews. This concept of a *LDA* algorithm seeded with lexical features in order to gain more exact results and personally existing user knowledge about the topic distribution in a certain corpus was first described by Jagarlamudi et al. in 2012 [17]. A more recent and very well documented implementation was introduced by Vikash Singh in 2017 [58]. This approach enables a developer to seed the LDA algorithm with a list of topics in order to narrow the topic distribution down to a fixed number of topics and obtain better results. This approach was used in the thesis project and is described in detail in the next Chapter.

3.3 Summary

In this Chapter gave an overview of known research projects on sentiment analysis and topic models during this Section, looked back briefly on their history and beginnings, and pointed to appropriate initiatives. Having discussed the research history in this Chapter, it is now essential to define the technologies used and explain their functionality in more detail. The next Chapter “Methodology” depicts the technology parts used in detail and describes their functionality.

Chapter 4

Methodology

4.1 Introduction

After the presentation of the state of the art in the last Chapter, this Chapter is dedicated to the selection of suitable techniques, their functioning and the explanation why these techniques have been used. The first research question – mentioned in Section 1.2.2 – discusses the extent to which a statistically relevant correlation can be established between the price development of a cryptocurrency and its reporting on social media and news platforms. For this explanation, it is necessary to classify the statistical procedures used. Furthermore, this Chapter discusses the rule-based sentiment analysis method used, explains various classification algorithms and explains the suitability and the functional principle of the topic model latent dirichlet allocation. In addition, the programming language, librariers and frameworks used are presented.

4.2 Correlation calculation

In order to be able to answer the first research question of this thesis, it has to be clarified what exactly correlation is and what it represents in research case cryptocurrencies. In the process of elaborating research questions, the idea arose that there may be a measurable correlation or correlation between the sentiment on Twitter about certain cryptocurrencies and their price trend, because of the huge influence Twitter has on the crypto world. Based on initial research, the focus was also extended to crypto news, as they also seemed to have an influence on price behaviour.

4.2.1 Correlation

In statistics *correlation* is often defined as a statistical relationship between bivariate data[1]. Prokhorov defines correlation as "a dependence between random variables not necessarily expressed by a rigorous functional relationship" [55]. This dependence can be interpreted as the degree to which two variables are depended in a linear manner. *Correlation* is therefore an interesting metric for this thesis because it can be an indication of a predictive relationship. An example matching the first research question of this

[1] bivariate data – Involving two variables, as opposed to many (multivariate), or one (univariate) [65]

thesis would be that the transaction volume of a cryptocurrency may increase if much is reported about it on news platforms, or particularly many tweets about the currency with positive sentiment are posted. However, there is one important point to mention. Correlation does not imply causation, which means that even if two variables correlate, they do not necessarily causing each other.

4.2.2 Pearson correlation coefficient

The most commonly used method to calculate the correlation is the Pearson Correlation Coefficient, also called Person's r. This mathematical method was developed in the 1880s by Karl Pearson [29], based on an idea by Francis Galton [9]. It measures the extent and direction of linear dependencies between two independent variables. It can be defined as a covariance of the two variables divided by the product of their standard deviations. When applied to a population[2], given a pair of random variables (X, Y) the formula for $\rho_{X,Y}$ looks like

$$\rho_{X,Y} = \frac{\operatorname{cov}(X, Y)}{\sigma_x \sigma_y}, \tag{4.1}$$

where cov is the covariance and σ_x and σ_y are the standard deviations of their respective variable. When applied to a sample[3] the formula would look like:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 (y_i - \overline{y})^2}}, \tag{4.2}$$

where n is the size of the sample, x_i and y_i are the individual sample points with index i and $\overline{x}$ as well as $\overline{y}$ is the sample mean ($\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$). The calculation of Pearson's r returns a value between -1 and $+1$. These maximum values each express a perfect correlation in the positive or negative sense. If the correlation is in the negative range ($-1 < \rho < 0$) one speaks of the negative correlation, if the correlation is in the positive range ($0 < \rho < 1$) one speaks of a positive correlation. While a positive correlation is characterized by the fact that the two variables are directly related in their behavior (x increases – y increases, x decreases, y decreases), a negative correlation behaves exactly the other way round (x decreases – y increases, x increases – y decreases). If the correlation calculation results in the value 0 or against 0, there is no statistically recordable correlation. All this correlations are visible in Figure 4.1. The most frequently used graphical representation for correlations are scatter plots which are also used in this thesis to illustrate the relationships.

4.2.3 p-value – statistical significance

In order to be able to statistically test the calculation of the correlation and the associated hypothesis of a dependency, there is the so-called *p-value*, which measures the statistical significance. Such statistical tests are based on a so called *null hypothesis*, which denotes the claim that is made about a population – in our case the negation of

[2] Population – a set of similar items or events which is of interest for some question or experiment.

[3] A sample is a portion of the elements of a population. A sample is chosen to make inferences about the population by examining or measuring the elements in the sample.

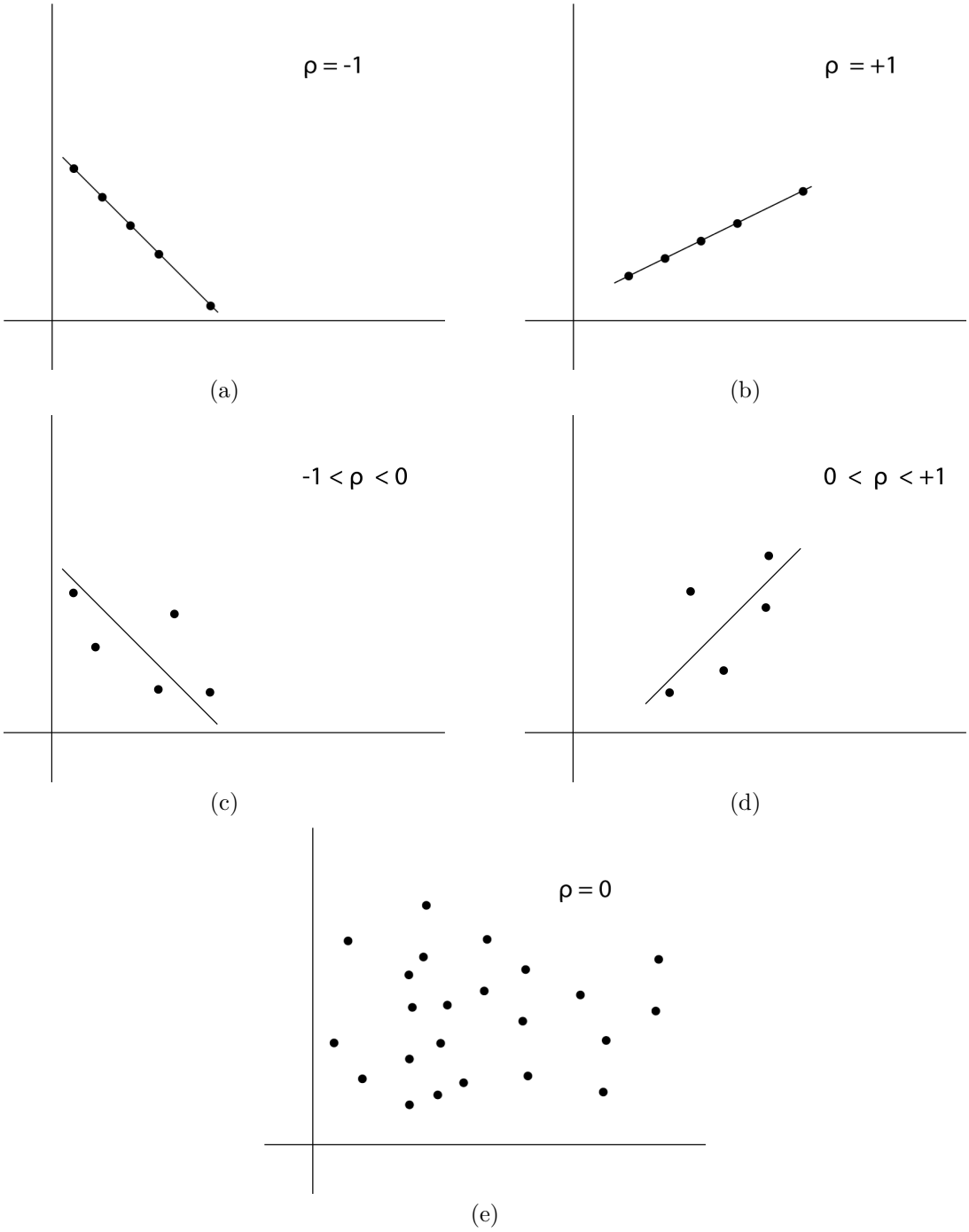


Figure 4.1: Graphical representation of various correlations. Depicting perfect negative correlation (a), perfect positive correlation (b), moderate negative correlation (c), moderate positive correlation (d) and no correlation (e).

the outcome of the correlation calculation. In this case the *alternative hypothesis* is the thesis one would believe if the *null hypothesis* is rejected. For example a *null hypothesis* for a positive correlation after the calculation of Person's r would show no positive correlation at all. In this case the *alternative hypothesis* would be the positive correlation. The *p-value* weighs the strength of evidence for or against the *null hypothesis*. The *p-values* are interpreted in three classes:

- *p-value* $\leqslant 0.05$ – strong statistical significance,
- *p-value* > 0.05 – weak statistical significance and
- *p-value* very close to 0.05 – marginal statistical significance.

If a *p-value* is in the first category, there is a strong evidence against the *null hypothesis*, which means that one can reject it. A weak statistical significance (*p-value* > 0.05) would show a weak evidence against the *null hypothesis*, which makes it not rejectable. Marginal statistical significance could go either way, therefore it is important to always report the *p-value* in addition to the correlation coefficient so conclusions can be drawn.

Due to the frequent use of the Pearson Correlation Coefficient in machine learning projects for further discussion of correlation and causation, Pearson's r was also chosen for this thesis. The results are explained in more depth in Chapter 6.

4.3 Sentiment analysis with a rule-based system

To calculate a Pearson correlation coefficient two values are needed – in this thesis one of these values is the sum of the sentiment value of tweets on a day. In order to know how sentiment of a text can be calculated the following Section describes sentiment analysis with rule-based models. Using the VADER technology already introduced in Section 3.1.2, the functionality of such models is explained and in particular the development process of the model used in the project is discussed. This technology was chosen because of its special suitability for social media texts and its low maintenance and initialization effort. Furthermore, the good calculation results of the sentiment values were convincing.

4.3.1 Introduction to VADER

VADER – which stands for Valence Aware Dictionary for sentiment Reasoning – was introduced by C.J.Hutto and Eric Gilbert of the Georgia Institute of Technology in 2014 [16]. It is a simple rule-based model for general sentiment analysis. This method was used in the thesis project as sentiment analysis tool for the crypto tweets.

4.3.2 Construction and development of this technique

VADER covers a special combination of polarity-based and valence-based methods. As a first step the researchers created a list inspired by established sentiment lexicons like LIWC [38] or GI [36]. Following this construction they incorporated a considerable amount of lexical features common to sentiment expressions in micro-blogs like a list of Western-style emoticons, abbreviations and slang words. A specific and distinguishing feature of VADER is the direct incorporation of human raters in the development of this approach. Already in this step ten human raters rated the sentiment valence of the

collected lexical features on a scale from −4 (extremely negative) to +4 (extremely positive), whilst also allowing a neutral (0). Identifying human-used generalizable heuristics to assess sentiment intensity in text was the next step in order to construct VADER. Also in this step human rating was considered.

Two experts rated about 800 tweets on a scale from −4 to +4. Using a data-driven inductive coding technique. Furthermore they used qualitative analysis to identify five generalizable heuristics based on grammar and syntax. Hutto and Gilbert point out that these heuristics go beyond the capture of established bag-of-word models. These five heuristics are punctuation, capitalization, degree modifiers, the word "but" as shift in sentiment polarity and examination of the tri-gram preceding a sentiment-laden feature for discovering negation flips. With these five heuristics in mind the researchers put together a list of 30 baseline tweets and varied them with the heuristic features. The researchers combined this variation of tweets with first data set of 800 tweets. These list of tweets was again rated by human rates for their sentiment intensity. In the last construction step Hutto and Gilbert let human raters asses the sentiment intensity of four text corpora in the domains of social media text, movie reviews, technical product reviews and opinion news article to create human-validated ground truth of sentiment intensity.

In order to evaluate their approach the researchers assessed the correlation of the computed intensity rating to the ground truth and the classification in positive, negative and neutral as well. They compared VADER to seven other well-established lexicons (LIWC, GI, ANEW, SWN, SCN, WSD and Hu-Liu04). For fairness in comparison all models used the VADER model for processing syntactical and grammatical cues, the only difference was in the lexicons themselves. VADER lexicon show exceptionally good performance in the social media domain and generalizes favorably over the different domains. When it comes to classification tasks and comparison to machine-learning powered approaches, VADER outperforms them on three of four corpora, while being more computationally efficient due to its simplicity.

4.3.3 Functionality

For each text the VADER model analyzes the sentiment and returns four scores (negative, positive, neutral and compound sentiment score). Hutto and Gilbert state in their paper that the compound score is computed by summing the valence scores of each word in the lexicon, adjusted according to the rules, and then normalized between −1 and +1. It is used for uni-dimensional measures of sentiment of a given sentence. When using the compound score, the researchers are setting standardized thresholds for classifying sentences as:

- positive sentiment: compound score >= 0.05,
- neutral sentiment: compound score > −0.05 and compound score < 0.05
- and negative sentiment: compound score <= −0.05.

This compound score was used in the thesis project in order to classify tweets in the three sentiment classes. These results can be found also in Chapter 6.

4.4 Classification Algorithms

Automated sentiment analysis is another widely used technology for sentiment analysis besides the rule-based variants. Automated sentiment analysis projects have already been discussed in Section 3.1.2. These technologies are mostly based on classification algorithms, which have been tested and used in many different areas. For a good analysis result, qualitatively and quantitatively high-quality data sets are required, which must also be labelled with the respective sentiment values. When creating these data sets, special attention has to be given to the intended purpose and focus of use. For example, Sentiment data sets from sports reports are not suitable for the training of sentiment classifiers for the sentiment analysis of texts with a political or financial focus. This labelling has to be done almost solely by hand, which is extremely time-consuming. This is also the reason why there are only very few openly accessible sentiment data collections.

Due to their wide range of applications, classification algorithms were also considered for this thesis project. Owing to missing and not procurable training data sets as well as news data collection that could not be classified according to sentiment, these following classification algorithms were only used in experiments. Because of their overall importance in natural language processing and machine learning projects, the algorithms used are briefly mentioned and described.

4.4.1 Logistic regression

Logistic regression is a very common machine learning method for binary classification. Logistic regression does not predict a variable given a set of features (such as linear regression does), but calculates the probability of a given input variable belonging to a certain class. Therefore, the result of a logistic regression calculation is always 0 and 1. A very important part of the logistic regression algorithm is the usage of the a so called *sigmoid* function or *logistic* function (Figure 4.2), which is

$$g(a) = \frac{1}{1 + e^{-a}}.$$

The variable e defines the so called "exponential function" and equals a value of 2.71828. This function squashes the value and is the reason why it always returns a value between 0 and 1.

A logistic regression algorithm normally consists of these three steps:

- Calculation of the *logit* function – hypothesis of linear regression which is

$$-\theta 0 + \theta 1 * X,$$

- applying the *sigmoid* function to the *logit* function,
- and the error calculation, with a cost function for max. log-Likelihood.

4.4.2 Naive Bayes

Naive Bayes is a simple but powerful probabilistic machine learning algorithm that makes use of the concept of Bayes Theorem and probability theory. For example, when

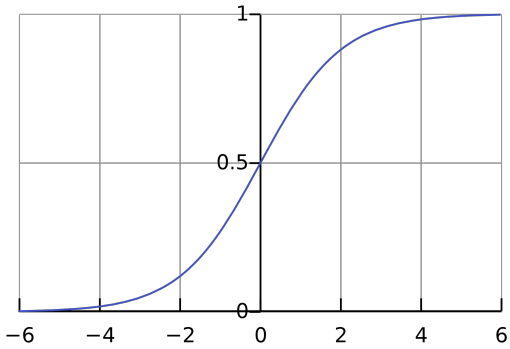


Figure 4.2: The logistic curve.

using Naive Bayes for tagging text pieces, it first calculates the probability of each given tag for the text piece and then outputs the tag with the highest probability of belonging to a text. Bayes' theorem is not only the eponym of the algorithm but also its core function. It describes the probability of an event, already based on previously acquired knowledge about further circumstances that could influence the event. It is stated mathematically as:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}.$$

$P(A|B)$ denotes the possibility of an event A happening given B is true. $P(B|A)$ denotes the possibility of an event B happening given A is true. Both of these mathematical expressions represent conditional probability. $P(A)$ and $P(B)$ represent marginal probability, both are the probability of observing either A or B.

The part of the calculation from which the word "naive" comes is the fact that this algorithm evaluates each word individually and independently of the others in its probability of belonging to a certain class. There are some techniques to improve a Naive Bayes model, especially regarding the words applied. Examples for such techniques are the use of *n-grams* or *tf-idf*, the removal of stopwords or the lemmatization of words.

4.4.3 Support Vector Machines

Support Vector Machine or *SVM* is a very well established classification algorithm. Unlike the two algorithms mentioned so far, which are based on probabilities, this approach is inspired by geometry. The data to be classified is defined by two features (x,y), which are also coordinates in a multidimensional space. SVM tries to establish a discriminator in this space by means of an optimally positioned hyperplane, which is also called "decision boundary", in order to be able to make a classification. The Figure 4.3 illustrates the graphical components of an SVM. The yellow dashed line shows the decision boundary, while the two black outer lines show the support vectors. The distance between the two support vectors is called margin in SVM. The basic idea behind the SVM is to find a classifier whose decision boundary is furthest away from all data points.

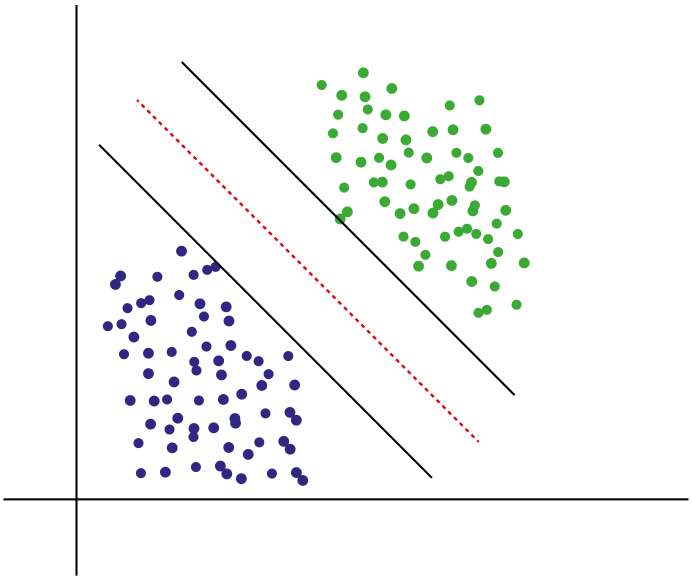


Figure 4.3: High-level view of a SVM plot.

4.5 Probabilistic Topic Models

4.5.1 LDA – Latent Dirichlet Allocation

As already mentioned in Section 3.2.3, LDA is the most cited and used probabilistic topic model today. Its architecture depends heavily on the two dirichlet priors. A good overview of the contributing variables can be found in the Plate Notation, which can be seen in Figure 4.4. It consists of the following parts:

- M – total number of documents in the text corpus,
- N – Number of words in a document,
- Z – denotes each topic -> topic of the n-th word in the document N,
- θ – denotes the topic distribution for document m,
- α – parameter of dirichlet prior on per-document topic distribution
- and β – parameter of dirichlet prior on per-topic word distribution.

After this introduction of the used variables as well as the structure, this part deals now in more detail with the functionality of the LDA Approach. As a first step each word in the corpus is randomly assigned to a topic. This step can be controlled with the so-called Dirichlet priors ((α) and (β)) which have value between 0 and 1. The (α) prior is the parameter responsible for setting the prior for per-document topic distribution. A high (α) value means that every document is likely to contain a mixture of most topics as well as a low (α) value means a document is more likely to be represented by just a few of the topics. Beta prior is in control of per-topic word distribution. So similarly to the (α) prior a high (β) value means the each topic is likely to contain a mixture of most of the words, while a low value mean that a topic contains of just a few words.

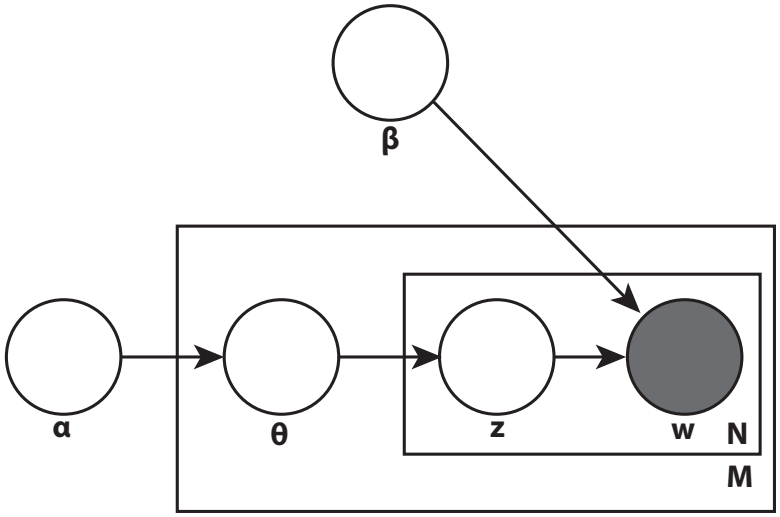


Figure 4.4: Plate Notation of an LDA Model.

A high (α) value results in documents appearing more similar to each others, whilst a high (β) value leads to topics appearing more similar to each other. With this priors developers are able to fine-tune the initialization step of the LDA algorithm for the desired outcome. A high value will lead to a uniform initialization. In contrast to this a low value will lead to a skewed initialization.

The next step is determining which term belongs to which topic – the topic modeling. For example the algorithm starts with the term "Bitcoin" which is first assumed to be assigned to the right topic. Firstly the algorithm computes the terms that "Bitcoin" comes along with in a frequent manner. Secondly it distinguishes the most common topic of those terms. "Bitcoin" is then assigned to that topic. This process is repeated for the next term and also a number of times in general in order to create a converging topic model. The outcome of this process is the probability of "Bitcoin" belonging to a topic. Another influencing factor besides the dirichlet prior is the number of chosen topics, which also refers to the dimensionality of the topic model.

Final Result of a LDA computation is a word topic distribution. This contains all the topics made of all the words with their probability belonging to a topic. Such word-topic distributions can be visualised in a interactive way, using pyLDAvis, a visualisation package for Python, which is also used for this thesis project. An example can be found in Figure 4.5, which depicts a so-called Distance Map. This map depicts distinguishable and related topics in the text corpus.

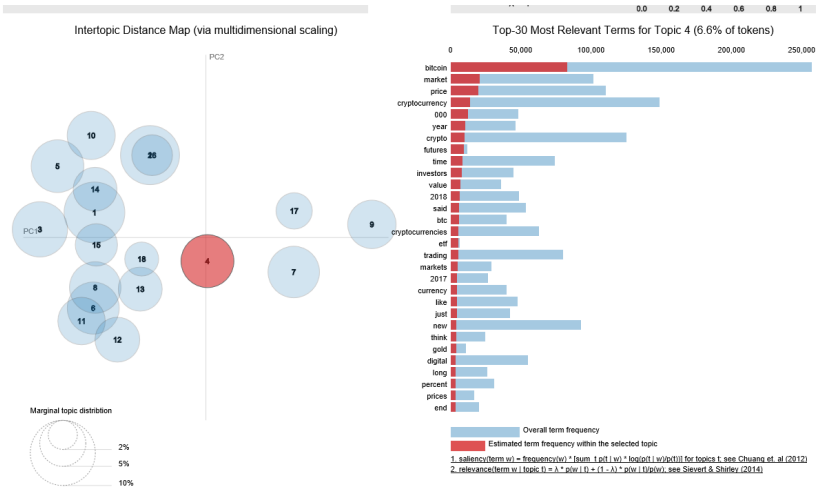


Figure 4.5: Interactive Visualisation using pyLDAvis.

4.5.2 GuidedLDA

LDA is one of the most sophisticated concepts for topic modeling. However, *LDA* does not have the ability to draw on the existing knowledge in the development team. It starts without previous knowledge about the topic distributions available in the text corpus. This is a point where improvement is possible and this is exactly where Guided LDA intervenes. When the topics and the corresponding words are known to the developer and there is a possibility to tell the algorithm what topics to be found including the words that are most frequently connected with this topic, then a better result is clearly reached. With the "guided" version of LDA, it is possible to define that certain words a more likely to appear in particular topics. This technique of "seeding" the LDA algorithms with words and topics was first implemented by Vikash Singh in 2017 GuidedLDA Package [58], also mentioned in Section 3.2.4.

The developers implemented two changes that ultimately lead to better results. First, the possibility was added to assign words to a certain topic (also called "seeding list"). This list is used to make it clear to the algorithm that these words belong to the assigned topic and help to make the algorithm more targeted. Such a list is depicted in Figure 4.6. The second addition is another parameter with a value between 0 and 1. This "confidence" parameter expresses the developer's confidence that the list is accurate and improves topic modeling.

The reason for using the GuidedLDA Approach for this thesis is mainly the special technical terms of the crypto industry, which should be used for good analysis results. These technical terms have been assigned to the respective super-topics in the seeding list and proved their effect in the analysis – further results are presented in Section 6.2.

```
seed_topic_list = [['btc', 'bitcoin','satoshi', 'nakamoto'],
                ['eth', 'ethereum', 'foundation', 'contract'],
                ['ltc', 'litecoin', 'lee'],
                ['ripple', 'xrp',],
                ['bch', 'cash', 'fork',],
                ['eos','monero', 'dash', 'altcoin', 'dash', 'tron', 'stellar', 'altcoins', 'cardano', 'ada', 'trx', 'cla
                ['mining', 'hashing', 'hashrate', 'reward', 'gpu', 'pools', 'electricity'],
                ['exchanges', 'trading', 'coinbase', 'poloniex','huobi', 'kraken', 'gdax', 'mtgox', 'cryptopia', 'binanc
                ['market', 'markets', 'index', 'price', 'analysis'],
                ['china', 'korea', 'japan', 'hong', 'asia', 'tokio', 'beijing', 'taiwan', 'singapore'],
                ['ico', 'offering', 'token', 'sto', 'ito', 'initial', 'tokens'],
                ['regulation', 'legal', 'law', 'tax', 'legislation', 'state', 'sec', 'senate', 'bill', 'taxes', 'treasu
                ['blockchain', 'blocks', 'protocol', 'decentralized'],
                ['bull', 'bear', 'bullish', 'bearish', 'trading', 'bots', 'rally', 'run', 'crash', 'down', 'red', 'gree
                ['tech', 'technology', 'software', 'development', 'artificial', 'ai'],
                ['ledger', 'distributed', 'trezor', 'jaxx', 'myetherwallet', 'keepkey', 'coinomi'],
                ['fiat', 'reserve', 'gold', 'bank', 'dollar', 'usd', 'euro', 'eur', 'bank', 'pound', 'yen'],
                ['business', 'stock', 'startup', 'enterprise', 'ceo', 'profit', 'revenue'],
               ]
topics = ['bitcoin','ethereum', 'litecoin', 'ripple','bch', 'altcoins', 'mining', 'exchanges', 'market', 'asia', 'ico',
```

Figure 4.6: Seeding list for guidedLDA approach.

4.6 Languages and tools

4.6.1 Python

Python [56] is a well-established high-level programming language, which was first introduced in 1990. It was created by Guido van Rossum and first released in 1991. It is dynamically typed, has a garbage collector and can be used for object-oriented as well as functional programming. At the time of writing, the latest stable release has the version number 3.7.3. Its indentation based syntax and the use of significant whitespace are distinguishing features. With the built-in package manager *pip* further software packages can be installed and managed. These dependencies are managed in the requirements.txt file.

Due to the massive upswing in the interest of machine learning and artificial intelligence, Python is also experiencing a massive upswing in popularity, familiarity and user status. The reason for this is the quick introduction by the easy to understand syntax and concepts, the many already existing useful packages (numPy, pandas, etc.) and the free use by the GNU license. Github's latest study on the popularity of programming languages shows that Python has been the undisputed number 3 behind Javascript and Java since 2015. Behind Python are the heavyweights PHP, C++ and C# [50].

Because of its special suitability for machine learning projects, Python was also used for this thesis and the accompanying project. While in the development phase of the prototype normal Python scripts were used, for the realtime application the framework Django was used, which is discussed in the following Section.

4.6.2 Django – a Python Web Framework

Django [59] is a free and open-source web framework based on Python, which follows the model-template-view (MTV) pattern. It is developed and maintained by a non-profit organisation called Django Software Foundation. It was first introduced in 2005.Following the principles of Python, it relies heavily on the concepts of reusability, code generators and the principle of DRY (don't repeat yourself). It can be combined with all kinds of modern webtechnology like bundling software (webpack) and comes with a

built-in admin functionality. Its strength for data-driven applications as well as the fast entry were the main reasons for the creation of the framework for the analysis software, which was developed within the scope of the project.

4.6.3 Useful libraries and tools

As already mentioned the package manager integrated in Python offers access to many different useful libraries and frameworks, which were also used in the accompanying project. Here is a short overview of the packages used:

- FancyURLOpener – for handling screen scraping on web pages,
- Scikit-learn – machine-learning framework for Python,
- NTLK – Natural Language Processing toolkit for Python,
- Pandas – data handling in the code,
- Matplotlib – for visualisation,
- GuidedLDA – python package for guidedLDA approach,
- VADER Sentiment analysis wrapper for rule-based/lexicon-based model,
- Google Cloud NLP API – for interacting with the Google Cloud API and
- Tweepy – framework for easy usage of Twitter API.

Overview is one of the most important factors in machine learning projects with a lot of data. To keep the overview, there are some noteworthy programs that support you in this important part. Here is an overview of the tools used:

- DB Browser – sqlite database browser for database handling,
- sequel Pro – MySql database admin tool,
- Anaconda Navigator – Enviroment Handling for Python,
- Spyder IDE – Scientific Python IDE – for scraping scripts,
- PyCharm – Sophisticated Python IDE with built-in version control,
- Jupyter notebooks for documentation and description of the code and
- Jenkins – a cloud based deployment software.

4.6.4 Scikit-learn

The massive upswing in interest and above all in the mass use of machine learning projects is certainly partly due to machine learning libraries such as Tensorflow and Scikit-learn [53]. They make it possible to get into the subject matter and use familiar functions in a relatively short time and without extensive mathematical or algorithmic knowledge. Scikit-learn itself is a freely usable library that has various machine learning algorithms at its disposal. Different classification, regression and clustering methods are integrated as well as the scientific libraries NumPy and SciPy. The library has been developed since 2007 by David Cournapeau and INRIA (French Institute for Research in Computer Science and Automation). Due to the advantages mentioned above, this library was also used for the thesis accompanying project.

Chapter 5

Solution approach

The theoretical backgrounds and functionalities of the technologies used are the focus of the previous Chapter. This Chapter, on the other hand, focuses on architecture of the analysis software, data structure and the use of the methodologies mentioned in Chapter 4. Furthermore, this Chapter will take a closer look at data mining, takes care of data preparation and explain the first results of the project prototype. After the methods used have been explained, the optimization process will also be discussed in more detail. The final step is to describe the fully automated analysis software, its structure and functionality.

5.1 Architecture

5.1.1 Development process

The development process of the thesis project, which accompanied the present thesis, is divided into two parts. The first part is the creation of a prototype in the form of Jupyter notebooks to try out the technical processes. After the functionality and the correctness of the assumption that there has to be a correlation between cryptocurrency market development, news volume and sentiment on Twitter was confirmed, a fully automated application was developed in the second step. This application crawls, processes and analyzes text data from Twitter and news platforms fully automatically and displays the data. This Section gives a brief overview of the selection criteria of the data sources and the structure of the database.

5.1.2 Selection of data sources

A fundamental decision that had to be well considered was the choice of data source. For the part of the sentiment analysis, short messages from the Twitter platform were selected. The specific language and incorporated emotion makes these messages a perfect fit for sentiment analysis. Another reason is the uniformity (140-280 character limit) and similar text structure of these messages which offer good basic requirements for comparison. An important point that tweets emphasizes is the massive influence on investment decisions and reach, especially in the early days of the cryptohype. In addition to the large number of relevant tweets comes their availability via the Twitter API. In

summary, tweets offer excellent suitability for Semantic Text Analysis. Section 5.2.1 continues with data mining on Twitter in this project, Section 5.3 gives insights into the sentiment analysis process whilst Section 6.1 presents the results.

Due to the first results in the field of sentiment analysis a second data source was considered. A prerequisite was that the text data had to be relevant, accessible and analyzable, and that it seemed to have a similar influence on the market development of crypto currencies as tweets do. In this case, news articles about cryptocurrencies were examined more closely. In the course of the hype on cryptocurrencies starting in 2016 more and more professional news agencies or news sites were founded, which specialized in the crypto market. These news platforms provide investors with further information on news from the world of crypto currencies. Due to their similar structure and length, as well as the large number of available news items, their reach and influence, news items were selected as a second data source for further analysis. Because news data are not suitable for sentiment analysis due to their nature, further semantic text analysis algorithms were considered. In order to identify hidden structures and assign the news to certain currencies or topics, Topic Models (LDA and Guided LDA) were utilized for the news dataset. Section 5.2.2 gives insight into the data mining of news, Section 5.4 explains the usage of topic models and Section 6.2 presents the results.

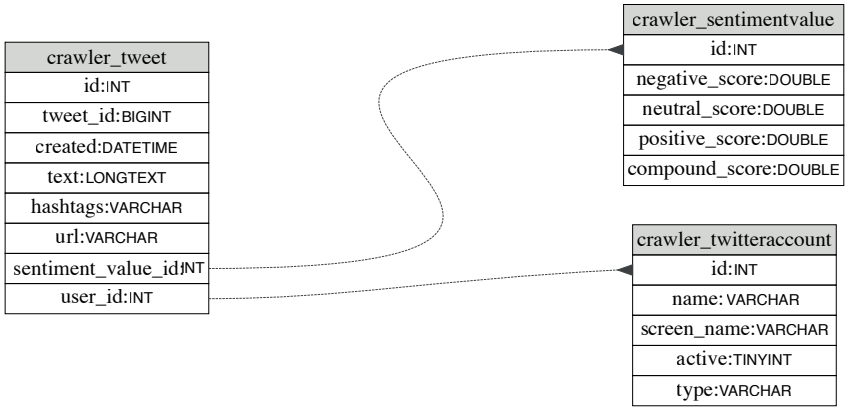
5.1.3 Data structure

In order to keep the news, tweets and their associated sources and analysis data easily accessible in the database, the following database structure was conceived in:

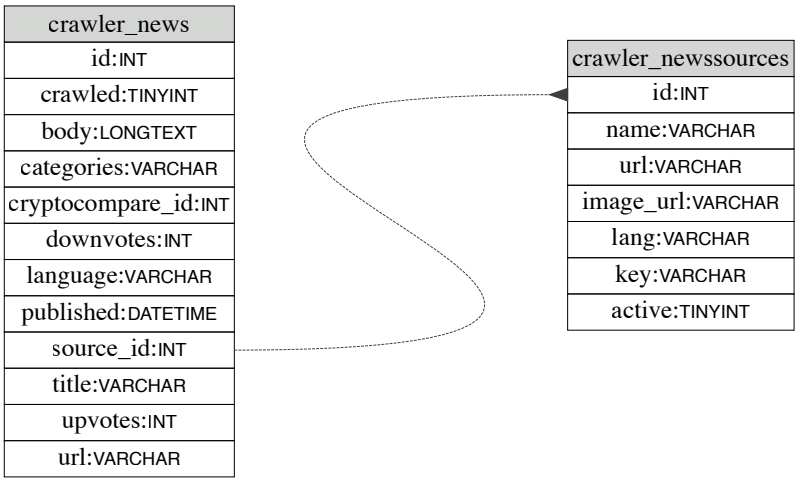
- Database relation for Cryptocurrency Prices, shown in Figure 5.1 consists of "crawler_dailyprice" storing the market data for a currency up-to-date, with a foreign key to "crawler_currency" storing the currency metadata like name, shortname and crawler url.
- Database relation for tweets, Twitter users and sentiment value, shown in Figure 5.1, consists of table "crawler_tweet", storing the tweet itself and relevant metadata. It has two foreign key for the table "crawler_sentimentvalue" storing the calculated sentiment value for each tweet and for the table "crawler_twitteraccount", storing all relevant data for the Twitter account, e.g., the userhandle.
- Database relation for news data and their corresponding sources, shown in Figure 5.1, consists of "crawler_news" storing all newsentries and their metadata like timestamp of when the news entry was published, title, upvotes/downvotes etc. It has a foreign key relation to "crawler_newssources", which stores all news platforms.

5.2 Data Mining

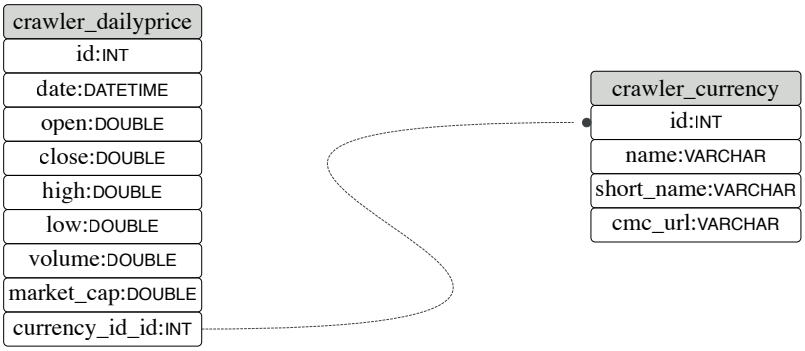
A very fundamental step in the research and development process of this thesis and the accompanying project was to commit to data sources. For this purpose, basic prerequisites were defined. The data that would be considered for the analysis should have following characteristics: data is available in sufficient quantity, data is available through API interfaces or screen scraping and data can be prepared and analyzed meaningful.



(a)



(b)



(c)

Figure 5.1: Graphical representation of database relations. Depicting the database relation for the twitter data (a), the database relation for the news data (b) and the database relation for saving daily prices (c).

Program 5.1: Two functions for cleaning tweets from not relevant text parts

```
1     def strip_links(text):
2         link_regex = re.compile(
3             '((https?):((//)|(\\\\))+([\w\d:#@%/;$()~_?\+-=\\\.&](#!)?)*)',
4             re.DOTALL
5         )
6         links = re.findall(link_regex, text)
7         for link in links:
8             text = text.replace(link[0], '')
9         return text
10
11    def strip_usersAndLines(text):
12        text = re.sub("@(\w)*", "", text)
13        text = "".join([s for s in text.splitlines(True) if s.strip("\r\n")])
14        text = text.lstrip()
15        text = text.rstrip()
16
17        return text
18
```

5.2.1 Twitter

Research on the public Twitter API endpoints resulted in the following findings. Twitter Search API [62] is be useful, because it returns tweets regarding to the topic (e.g., "Bitcoin") from all over Twitter . A limitation of this specific endpoint is that it only returns the tweets of the last 7 days. Further research on available API endpoints led to the Twitter User Timeline endpoint [63], which returns the 3200 most recent tweets for a certain user which is deemed as a fitting number of tweets. In order to acquire tweets with influence and liability, the users to be crawled should have a certain recognition in the crypto sphere. A list of the 50 most influential Twitter users regarding the cryptocurrency world was published on Medium [61] in Mid 2018. This list consists of influencers, entrepreneurs, accounts of crypto exchanges as well as crypto projects (Table A.1 and A.2). This list of users proved as a good fit for the subsequent sentiment analysis of relevant, influencing text content concerning crypto market development. For crawling the Twitter API a package called Tweepy was used, which eases the authentication process to the API [45]. In order to get metadata of the Twitter user also the Twitter User API endpoint was utilized.

Some problems appeared during the closer analysis of the Twitter data. Besides tweeting about cryptocurrencies, most of the chosen users also use Twitter for their personal communication. In order to avert a bias in consequence of crypto-irrelevant tweets, tweets were only considered when incorporating a word from list of crypto-relevant expression. Furthermore, irrelevant parts of tweets such as hyperlinks or Twitter user handles (e.g., "@username") were removed with cleaning functions, – visible in Program 5.1. Tweets over 15 characters remain in the data set due to the elimination of very short tweets. In total about 32,000 usable, cleaned tweets are available in the Twitter data set in the time period from 2009 to 2018.

5.2.2 News Platforms

For the news dataset an API endpoint from cryptocompare.com for cryptonews was used as source of relevant news [48]. This endpoint returns news and metadata of 34 news outlets dedicated to cryptocurrencies from August 2013 till today. Unfortunately only a max. 400 character long subsection of each newsentry is included, besides metadata like URL, newsoutlet, category, etc. This 400 character long body of text is sufficient for showing the principles but not enough for further, more reliable analysis. Therefore, the corresponding URL is used for screen-scraping on every news source webpage and saving the whole body of the news text.

Some problems surfaced within this gathering step. For example, due to the long history since 2013, numerous news entries were missing. Some of the newspages included script tags for dynamic functionality in their texts which had to be excluded. Dropping blank lines, breaking multiple headlines into lines, removing leading or trailing spaces were further steps in cleaning the text data. Another circumstance that caused a considerable amount of work is the structural difference on the news pages. Each news page has its own crawler with its own access and query functions. The cleaning functions are quite similar to the functions for Tweets.

After gathering and cleaning, a data set of about 72,000 news entries with corresponding metadata over the time span from June 2017 to January 2019 is available and usable for further text analysis.

5.2.3 Crawling and preprocessing cryptocurrency market data

History data of the most traded cryptocurrencies were acquired via screen-scraping from coinmarketcap.com [47]. This source is the most reliable cryptocurrency statistics page today because it gets its data from many different marketplaces and calculates it correctly. Being complete and very well structured this data set just needed date time formatting. From this page history data for Bitcoin, Ethereum and Ripple were scraped, covering the period from date of listing on Coinmarketcap until today.

5.3 Sentiment analysis

5.3.1 Rulebased models

The first step was to decide on which time spans tweets are analyzed. For this determination is was interesting to know how many tweets are in the data set per day, which is depicted in Figure 5.2.

The number of tweets increases by the time of December 2016 in a steady manner. This is due to the increase of interest in cryptocurrencies, the establishment of dedicated news sites on Twitter as well as the rise of crypto influencers on twitter. The second cause for the number of tweets going up in the chart is that just the 3,200 recent tweets of each account can be crawled from twitter. This leads to the circumstance that only the 3,200 most recent tweets of Twitter users with many tweets are in the data set, even if they had more messages before.

To cover the most interesting parts of tweets and market development in order to find out if there are any correlating values, tweets from 2016-12-15 to 2018-12-15 are

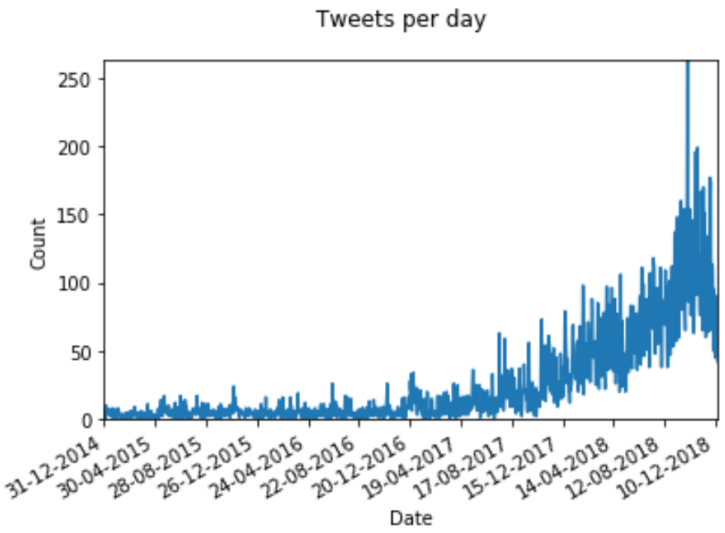


Figure 5.2: Tweets per day.

Figure 5.3: Bitcoin Market Development from April 2015 – January 2019.

taken into account. It not only covers the most interesting part of market development on Bitcoin (shown in Figure 5.3), but also covers the best combination from influencers' tweets and news outlets.

After restricting the data set to a total of 32,654 tweets between 2016-12-15 and 2018-12-15, the dataset is ready for sentiment analysis. For each tweet the VADER analyzes the sentiment and outputs four scores (negative, positive, neutral and compound). Hutto and Gilbert state that the compound score is computed by summing the valence scores of each word in the lexicon, adjusted according to the rules, and then normalized between -1 and $+1$. It is used for uni-dimensional measures of sentiment of a given sentence. The sentiment is computed for each tweet in these four categories and stored to the database. When using the compound score, the researchers are setting standardized thresholds for classifying sentences as positive, negative or neutral.

Program 5.2: Instancing and analyse by VADER package.

```
# import, database connection,  instantiation of VADER Analyzer
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
vader = SentimentIntensityAnalyzer()

# call analyzer on each tweet and saving it to the db
for index, row in chosenDataset.iterrows():
vs = vader.polarity_scores(row['text'])
cur.execute('''INSERT OR IGNORE INTO tweets_analyzed (tweet_id,screen_name,
created,text,account_type, negative_score, neutral_score, positive_score,
compound_score) VALUES ( ?,?,?,?,?,?,?,?,? )''', (row['tweet_id'],row['
screen_name'],row['created'],row['text'],row['account_type'], vs['neg'], vs['neu
'], vs['pos'], vs['compound']) )

conn.commit()

```

After classifying a tweet into positive, negative and neutral due to the thresholds – visible in Program 5.2– the total numbers were summed for each sentiment class. In addition to that also the mean sentiment value was computed in order to analyze if the direct sentiment value is correlating with the trading volume. The results of the Twitter dataset with the sentiment values are discussed in the Section 6.1. An important point can already be anticipated. The volume of positive, negative or neutral tweets correlates markedly with the market development. A point worth mentioning is, in any case, that the general volume of tweets correlates most distinctively. This circumstance and the mostly neutral texts of news (like mentioned in 5.3.2 led to change, where the project examined the news data set by means of Topic Models on the volume of occurring topics and their correlation to the corresponding cryptocurrency.

5.3.2 API Usage and Classification algorithms

Regarding the use of Sentiment Analysis APIs it can be stated that there is a vast number of projects that cover all kinds of analysis techniques. Some of them use rule-based model [54], some of use classification algorithms [60] and some of them offer no information about their analysis algorithms at all [51]. Despite often being referred to as a "blackbox", those API's come in handy for a fast proof-of-concept. Google Cloud API for example has a highly valued reputation and can be viewed as a state-of-the-art sentiment analysis tool.

When structuring the project at start, the news dataset was selected for applying machine-learning classification algorithms for sentiment analysis. A decision that proved to be false. In order to use supervised classifiers for sentiment analysis, there must be training data in sufficient size and fitting to the topic. Hence no fitting sentiment test/training dataset on news entries was available, a test dataset is generated using the crawled news, which were labeled by the Google NLP Api. The sentiment analysis using the Google Cloud API, which proves to be state-of-art and generalizes well over numerous domains labeled about 12,500 news entries from the crawled data. These

news texts are randomly chosen to cover all writing styles, news outlets and times of publishing. The results, which can be reviewed in Section 6.1.2, led to the elimination of the news dataset in means of sentiment analysis and subsequently led to the selection of topic models for further analysis of the news data set as mentioned in Section 6.1.3.

5.4 LDA und Guided LDA

As with sentiment analysis, the first decision regarding the timing and relevance of the data set was also important for the topic modeling part of the project. Sufficient amounts of news are available in September 2017. Therefore, the analysis period was set from September 2017 to January 2019. A period that is not only suitable due to the data but also includes the time of the most striking market development of crypto currencies. In the selected period 72500 news contributions can be accessed.

5.4.1 Latent Dirichlet Allocation

After defining the appropriate time series, the first step was to use an LDA algorithm. To ease the use of the algorithm as well as the data handling and vectorisation, the implementation in Scikit Learn was used in the implementation visible in Program 5.3. An important step before fitting the text data to the model is the vectorisation and removing stopwords. This was done by the *CountVectorizer*, which also removes stop words, which are likely to lead to false positives. Finding the right configuration (chosen values of topics, alpha and beta dirichlet priors) is time consuming, because the calculation of an LDA Topic Model, especially for such a big dataset takes its time. The meaning of α and β dirichlet prior is mentioned in the corresponding Section 4.5.1 of the Chapter 4. For the news data set, a topic number of 25 was set, because it is assumed that a relatively large number of topics are hidden in this large dataset. The results of the LDA passage are discussed in more detail in Section 6.2; however, there was a frequent overlapping of topics or a wrong allocation of words to certain topics.

5.4.2 Guided/Seeded Latent Dirichlet Allocation

Based on the LDA algorithm a further developed version called GuidedLDA was published for the first time in 2017. It offers the possibility to configure the algorithm using a topic-words assignment list or also called seeding list. This distance map of the LDA approach in Figure 6.3 depicts that there are distinguishable and related topics but it also shows how difficult it is to find or define the right topics. Broadly shared terms like “crypto”, “cryptocurrency” or “bitcoin”, which have a high frequency in news texts in this sphere are the main reason for this circumstance. This result is not satisfying and not good enough for further usage on this dataset. Because of this circumstance guidedLDA is used for seeding topics and assigned words. In addition, we have access to a tried and tested assignment of typical words to topics from the crypto industry [48] and also have enough expertise ourselves to further improve this list. This list consists of 19 topics from the crypto industry which are assigned between 2 and 10 words and are fed to the algorithm by means of the seeding possibility. The list is already visible in the Chapter 4 – Figure 4.6. With this list it is important that all seeding words also

Program 5.3: Usage of a LDA Algorithm.

```
1  from sklearn.decomposition import LatentDirichletAllocation
2  from sklearn.feature_extraction.text import CountVectorizer
3  import pyLDAvis
4  import pyLDAvis.sklearn
5
6  # vectorizing
7  count_vectoriser = CountVectorizer(stop_words = 'english', max_features = 1000)
8  document = count_vectoriser.fit_transform(df['news_document'].values)
9  features = count_vectoriser.get_feature_names()
10
11 # LDA modeling
12 topics = 25
13 alpha = 0.01
14 beta = 0.2
15 LDA = LatentDirichletAllocation(n_components = topics, doc_topic_prior= alpha,
   topic_word_prior = beta)
16 news_lda = LDA.fit(document)
17
```

occur in the text corpus, therefore this circumstance is checked before use and unused words are deleted from the list. By using this seeding option, a more precise allocation to topics was possible and can be mentioned as a positive effect. The results of this algorithm can be found in the Section 6.2.2.

5.5 Optimization process

5.5.1 Problem statement

The GuidedLDA Algorithm provides the developer with an opportunity to facilitate the topic finding process of the LDA algorithm with a Topic-Word assignment, known to the developer. However, this circumstance requires specialist and industry knowledge with regard to the interrelationships of the technologies and the special terms used. This knowledge is not always directly available to the developer and must first be obtained from specialists or clients through a proprietary process. To ease this problem, we introduce a process that allows the developer to get to Topic-Word contexts without contacting an expert in the field first.

5.5.2 Introduction of the process

Here we introduce a multi-stage process consisting of the acquisition of relevant data, the use of an LDA model and validation by experts. The result is a seeding list that is suitable for use with Guided LDA. This process can not only be used for the generation of a Seeding List, but it is also suitable for continuous improvement or adaptation to changed contexts in the specialist topics. For example, if a guidedLDA shows overlapping topic-word distribution due to a large shared vocabulary etc. The core steps of the optimization process are defined as:

- research for relevant and categorizable text sources,
- crawling and preprocessing of this text data,
- application of an LDA algorithm to this data,
- validation of the final Topic Distribution by several experts and
- use the seeding list in a GuidedLDA Approach.

5.5.3 An example using a cryptocurrency topic

The Topic-Word Distribution of the online comparison platform cryptocompare.com – which is already used in the thesis project – will also be applied to this example. This platform uses this distribution as a categorization for its own news database. Unfortunately there are only very few words for certain topics in this seeding list. An example would be the topic "Ripple", which refers to a cryptocurrency with only two words assigned ("Ripple", "XRP") – see Figure 4.6. This already leads the guidedLDA algorithm in the right direction, but unfortunately two terms are not enough to define the topic well enough.

For the first step of the optimization process, various data sources were examined for their suitability. The most important factors were the daily update, but also the exact assignment to the topic. Another important reason for the decision was the coverage of as many news platforms as possible in order to find out the most frequently used terms of all platforms. All these points together resulted in the decision to use the News API endpoint of Cryptocompare, which by means of parameters includes all news about Ripple but excludes all other news from other crypto currencies from the result. After finishing crawling and pre-processing a dataset consisting of 2,500 news about "Ripple" is ready for the next step.

An already proven concept of this thesis is the LDA algorithm to identify hidden structures and coherent expressions. The prepared data set is now analyzed by the LDA algorithm and visualized by pyLDAvis. The results are presented and analyzed in Section 6.3. The last two steps of the optimization process are also discussed there.

5.6 Fully Automated Analysis Application

5.6.1 Introduction

The main idea behind this application is the visualization, automation and further description of the thesis project. In the research part of the project in the beginning the data acquisition and the analysis steps were implemented in simple Python scripts or Jupyter notebooks with a corresponding documentation. This kind of development is cumbersome in administration and not suitable for broader usage and further experiments. Therefore the decision was made to merge the functionality of all scripts and notebooks into one automated application as a second part of the project. Thus, it is possible to call up daily updated analyses at any time via the browser and to expand the functionalities further without having to spend a lot of time administrating the data.

Another focus of this step was on acquiring a new technology stack on the one hand and on being able to continue using proven concepts on the other. Therefore the decision was made to use the python-based web framework Django [59]. It was necessary to adapt

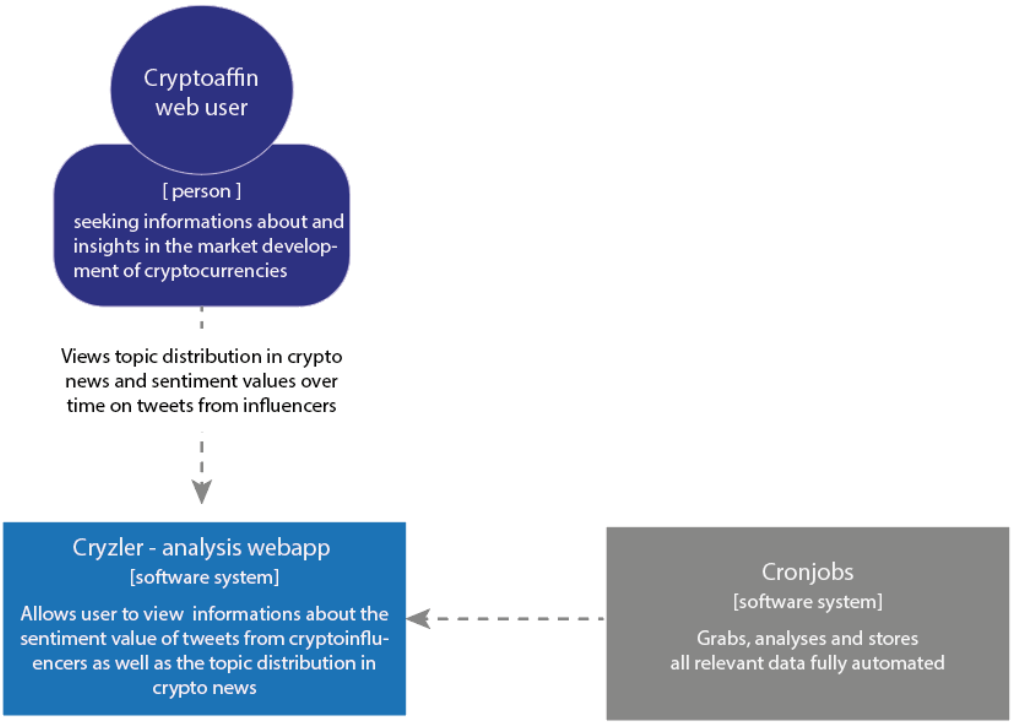


Figure 5.4: The architecture of the thesis project – a system context diagram according to the rules of the C4 model [46].

the jupyter notebooks into *commands* – Django's name for scheduled – and services as well as to convert the database into Django's own ORM model. Furthermore the cleaning and pre-processing processes were improved and refined and a new data visualization was considered. The architecture of this application can be viewed in Figure 5.4, and is described in detail in the following Chapter.

5.6.2 Architecture

The architecture of the application can be roughly divided into three different areas. The data collection and analysis part is decoupled from the other two parts and only responsible for crawling, pre-processing, and analysis of the text data. It consists mainly of scheduled tasks, which are responsible for handling the API calls and the screenscraping process, which is further subscribed in the following Chapter.

The externally visible application is divided into a backend and a frontend. Django is a MVT-Framework, which stands for *Model-View-Template* – visible in Figure 5.5. Therefore *models* and *views* are representing the backend – or the server side part of the software. This deviation from otherwise known MVC frameworks was an interesting point. For program parts that do not send a response back to the view, a separate service layer (service.py) was added for improved clarity so that the views (views.py)

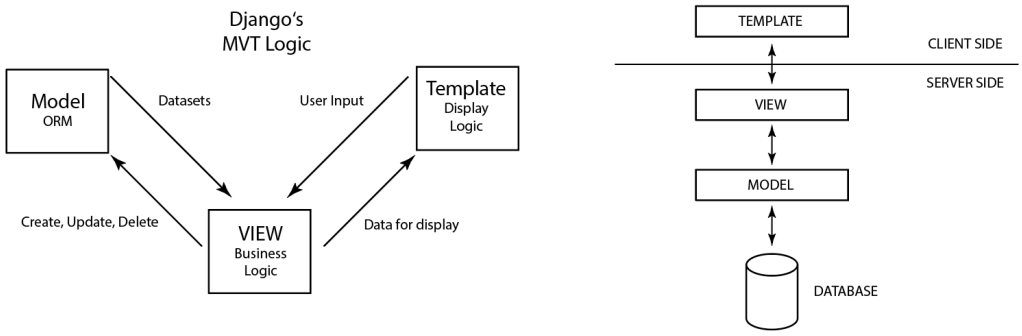


Figure 5.5: Django's own MVT concept.

were not cluttered. The necessary data is aggregated and the plotting library Bokeh [44] is configured in the views file. The frontend – respectively the *templates* – are responsible for visualisation and user handling of the website.

5.6.3 Automated data crawling

A main part of the webapp deals with the automated crawling of tweets via the Twitter API and news via the Cryptocompare API. In addition, current news texts are crawled from the respective news platforms using screen scrapers that are adapted to each website. The data used is currently crawled from 55 Twitter crypto influencer accounts and 34 news platforms. The data is automatically crawled, processed and analyzed by time-based, scheduled tasks every day from 23:00 to 24:00. These scheduled tasks are divided into pure crawling commands, which have pure crawling and preprocessing tasks, and analysis commands, which control sentiment analysis and the use of topic models. To avoid overlapping and overloading the database, the tasks are called in 5 minutes intervals. After these tasks have finished, the current status of the graphics and overviews is displayed on the website for the day. Total available and analyzed data (at the time of writing) is consisting of:

- 54,826 crawled and analyzed tweets,
- 88,450 crawled and analyzed news entries and
- 9,150 crawled market metrics data.

5.6.4 Analysis step

Two additional scheduled tasks have been set up to analyze the pre-processed tweets and news. The first job is only responsible for the Twitter data. Using the rule-based sentiment analysis framework VADER, it calculates the sentiment values of the respective tweets (negativity, positivity, neutral and compound score). Furthermore, he adds up all negative, positive, neutral tweets per day and calculates the mean sentiment for this day. This 0 then stores the calculated data in the database so that it can be used for visualization. The second scheduled job applies a guidedLDA with the already men-

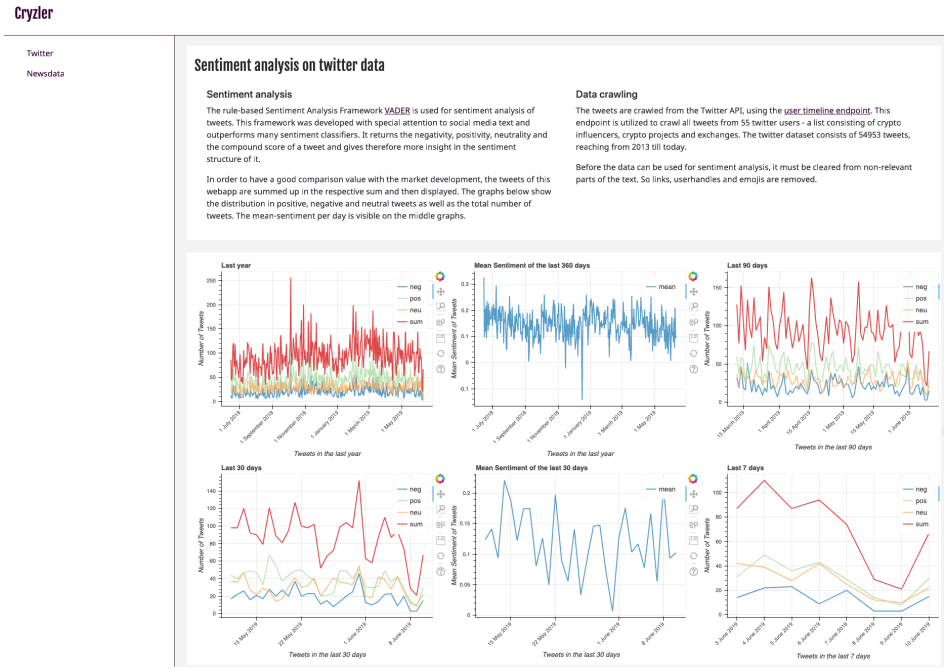


Figure 5.6: Subpage for the Twitter data and the sentiment analysis part.

tioned seeding list with the topic-word assignments to the news dataset and stores the model by pickle to make it available for visualization. Furthermore, this scheduled task generates and stores the visualization of the Intertopic Distance Map, which is very computation intensive and therefore done asynchronously in the backend.

5.6.5 Frontend

The frontend of the webapp is split into three pages. The homepage gives an overview of the project and its individual parts. The Twitter subpage, which provides information about the endpoints used and the algorithm used for sentiment analysis is illustrated in Figure 5.6. Below are four graphs showing the course of the sentiment over time. This course is represented by summing the tweets in the respective sentiment class.

The subpage for the news dataset, depicted in Figure 5.7) gives a brief insight into the data sources used, followed by a brief explanation of the topic model used. On the right side there is a pie chart which gives an overview of the distribution of the topics in the text corpus. Below is the already mentioned visualization of the GuidedLDA algorithm. For this the Intertopic Distance Visualization is used, which is created daily in the backend. Apart from the GuidedLDA visualization, the Python Plotting Library *Bokeh* [44] is always used, which enables a more interactive display in contrast to the standard library *matplotlib*.

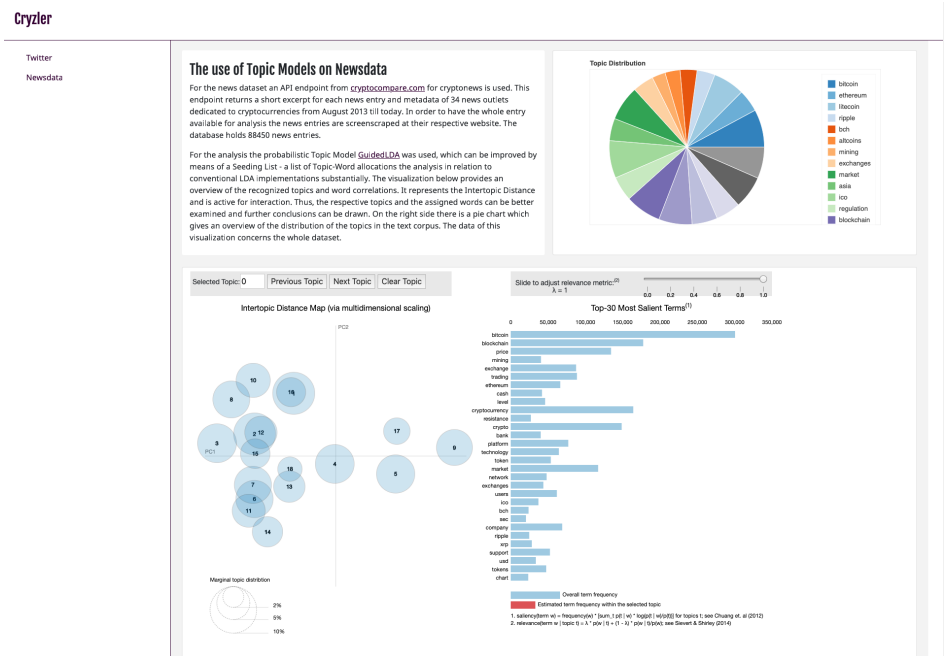


Figure 5.7: Subpage for the news dataset and the topic models.

Chapter 6

Solution evaluation

While in Chapter 4 the methodologies used are explored theoretically in their mode of operation and in Chapter 5 their application in this thesis is discussed in more detail, this Chapter is dedicated solely to the analysis of the results. First, the results of the sentiment analysis are analyzed and correlation values to the market development are calculated and conclusions are drawn. The next step is to explore the use of an API to calculate sentiment and discuss the results. Then the results of the Topic Models LDA and Guided LDA are examined and correlation values to the market development are calculated. As a final step, the improvements made by the optimization process and the results of the analysis software are discussed. The challenges as well as the lessons learned are declared at the end.

6.1 Evaluation of sentiment analysis

6.1.1 VADER Sentiment

After classifying tweet into positive, negative and neutral due to the thresholds, the total numbers were summed for each sentiment class. In addition to that also the mean sentiment value was computed in order to analyze if the direct sentiment value is correlating with the trading volume. From 32,654 tweets there are:

- 14,311 Tweets with positive sentiment,
- 7,234 Tweets with negative sentiment and
- 11,109 Tweets with neutral sentiment.

As a last step it was determined if it is possible to establish a correlation of the values, which were mentioned above, and the 24h trading volume of Bitcoin. The pearson correlation coefficient was used to compute these values. In fact a moderate positive correlation can be established from the total numbers of a three sentiment classes and also for the total number of tweets. These correlations are also statistically significant. Therefore, it can be concluded, that there is a positive correlation between the level of tweets about cryptocurrency topics and the trading volume. This circumstance was also a turning point and led to the amendment of probabilistic topic models to the project in order to measure the level of interest using the news dataset. This is represented in the Section 6.2.

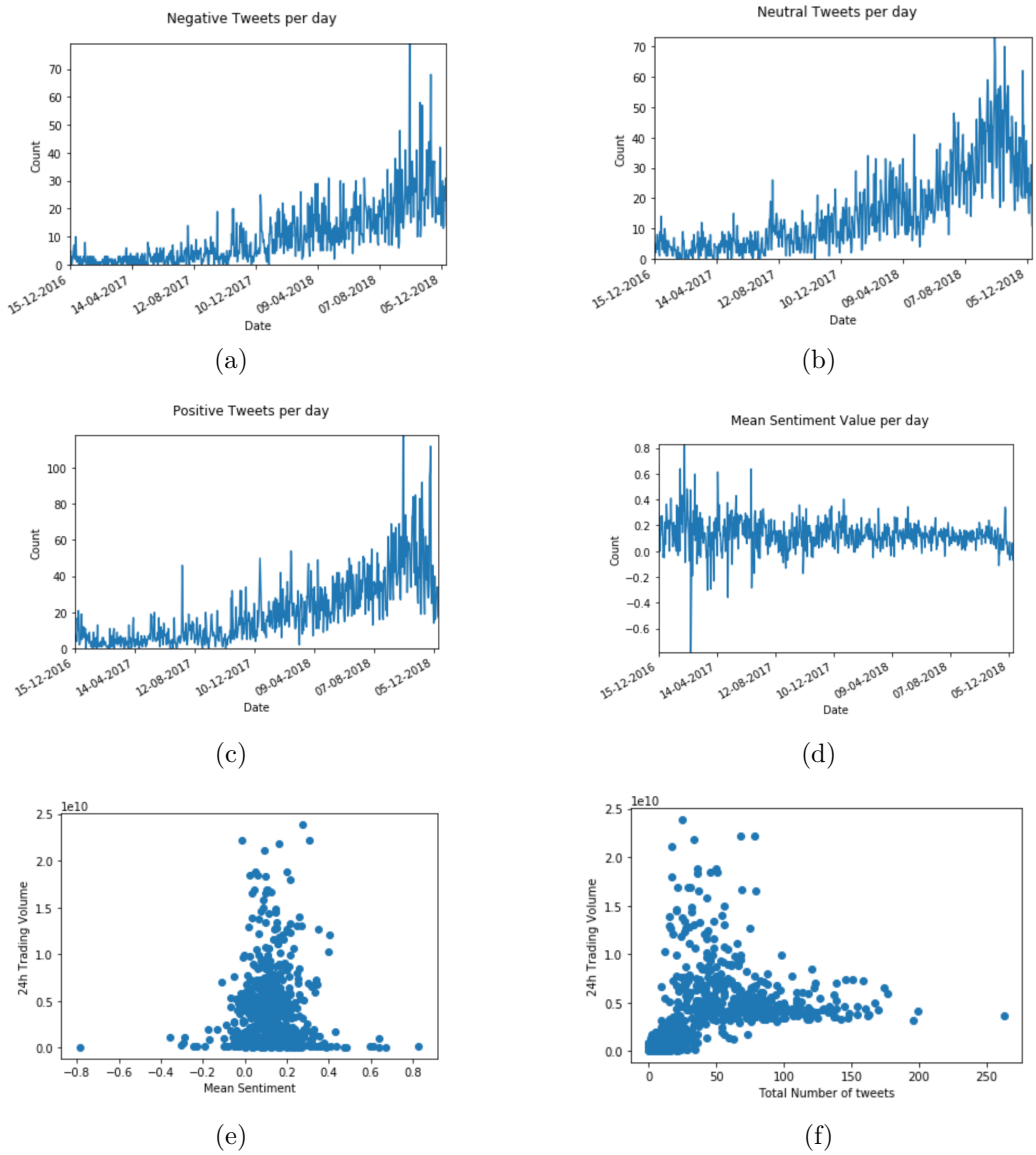


Figure 6.1: Graphical representation of the sum of tweets ordered by sentiment category and the mean sentiment value. negative Tweets per day (a), neutral Tweets per day (b), positive Tweets per day (c), mean sentiment per day (d), mean Sentiment correlation (e), Total number correlation (f).

If these results are examined more closely, two things stand out particularly. In contrast to the promising results on total numbers of tweets, the mean sentiment per day showed a very low, not statistically significant negative correlation. Therefore, we conclude that sentiment in itself does not have the great influence as thought, but rather much more the volume of the tweets.

- Total number of positive tweets:
  - Person coefficient: 0.304278

  - P-Value: 3.805245e−17
- Total number of negative tweets:
  - Person coefficient: 0.293326
  - P-Value: 5.432770e−16
- Total number of neutral tweets:
  - Person coefficient: 0.283444
  - P-Value: 5.431898e−15
- Total number of tweets:
  - Person coefficient: 0.311883
  - P-Value: 5.613105e−18
- Mean sentiment value of tweets per day:
  - Person coefficient: −0.036001
  - P-Value: 0.330708

6.1.2 Sentiment API

As a next step the test news were classified by the *Google Cloud NLP API* [51], which returns sentiment and magnitude value. The sentiment parameter reaches from −1 (negative) to +1 (positive) and depicts the positivity/negativity of the text. The magnitude parameter states how emotional the text is and represents a further calibration value, when comparing documents of variable length. As news entries are mostly in a neutral style of writing and all used news entries have a similar length this parameter is not taken into consideration for this analysis. Due to their sentiment value the texts are classified into positive, negative and neutral. News entries above a sentiment value of 0.25 are rated as positive, beneath −0.25 news texts are classified as negative. Between 0.25 and −0.25 news entries are categorized as neutral, according to the documentation. The API labeled a part of the news dataset as follows into:

- 1,087 News with positive sentiment,
- 1,026 News with negative sentiment and
- 10,395 News with neutral sentiment.

This very naive approach led to a very interesting finding. The newsdata consists predominantly of neutral texts which makes the dataset unusable for training classification algorithms – visible in Figure 6.2. This also may be the reason why there are no open datasets for training sentiment classifiers on newsdata.

6.1.3 Discussing and discarding the news dataset from sentiment analysis

News coverage should be perceived as neutral by showing all sides of a topic and must not be opinion-forming. Therefore, a very neutral and non-emotional writing style is used by journalists which makes it very difficult to distinguish if a topic is meant in a positive or negative manner - even for humans. Due to the insight that hardly usable values could be found on the basis of sentiment analysis, it was decided to continue using Sentiment API and to classify using classification algorithms. This circumstance

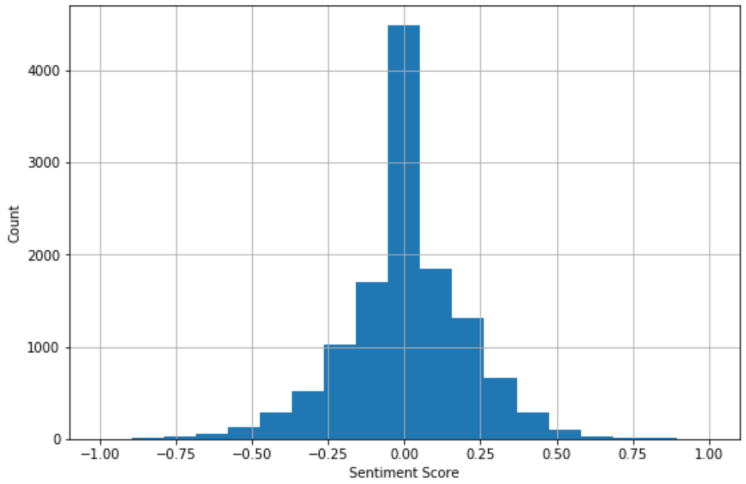


Figure 6.2: Distribution of sentiment values to number of news.

further consolidates the approach of this project to the usage of twitter messages for sentiment analysis and topic modeling for news coverage. This consideration extends the first approach of this project only using sentiment analysis with a second level of analysis and could add valuable insights into the public perception of cryptocurrencies and the mutual influence of trading volume, news and social media texts.

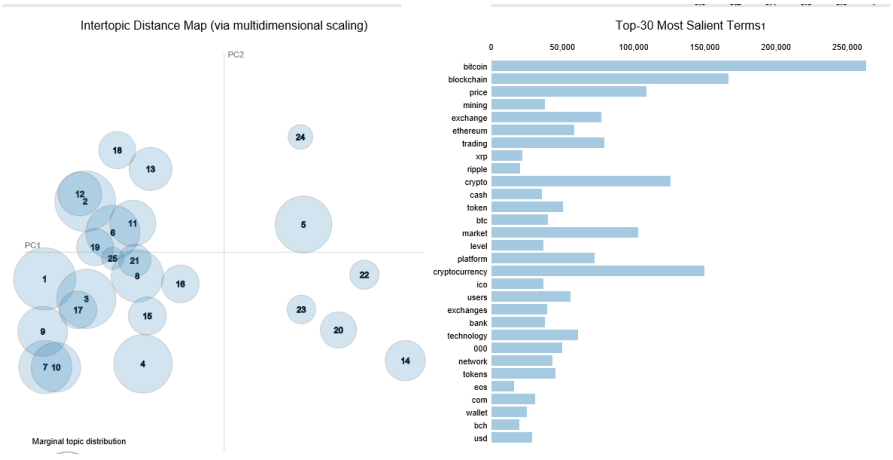
6.2 Results LDA/Guided LDA

6.2.1 LDA

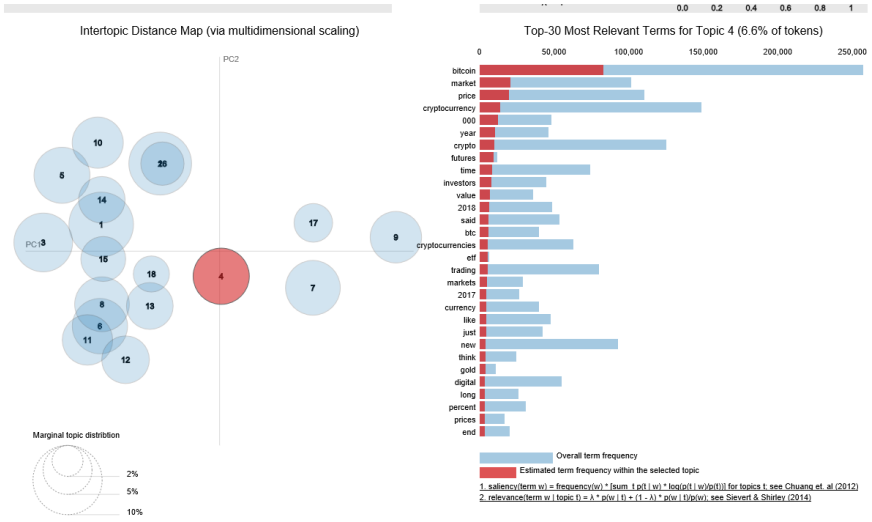
A recommendable way of plotting a computed topic model is to visualize the Top-30 most salient terms and the intertopic distance map. This can be easily done with a package called pyLDAvis and is depicted in Figure 4.5. Notable circumstances are visible in this visualization. On the one hand there are well separable topics (e.g., 4, 5, 14, 20) but it also shows topics in the left part with a widely shared vocabulary, which makes it difficult to determine topics. A further interesting finding is the high frequency of the words “bitcoin”, “blockchain”, “cryptocurrency” and “crypto”. These insights are of course fundamentally interesting and can help to understand the process, the content of the corpus and the existing topics. The “unguided” approach is too unreliable for a clear allocation of the found topics and would have to be calculated first. Due to this circumstance as well as the existing expertise about the content of the corpus and the topics to be found, the guidedLDA approach was used and a seeding list was created.

6.2.2 Guided LDA

To find fitting topics for this seeding list in order to “guide” the LDA, the news categorization from cryptocompare.com is used as a starting point, with little adaptions to categorize further altcoins with high trading volume like Ripple or Bitcoin Cash. Following the initial approach of using this technology to quantify the level-of-interest in



(a)



(b)

Figure 6.3: Visualisation of the computed LDA model (a) and visualisation of the guidedLDA approach (b).

a certain topic and computing possible correlations the next step covered aggregating total numbers of news articles of the topics “Bitcoin” and “cryptocurrency” and the computation part. Afterwards the results were visualised and discussed.

After analyzing Figure 6.3(a) and the Figure 6.3(b) it becomes clear, that there is a number of well distinguishable topics like “bitcoin”, “mining”, “ethereum”, “blockchain” but its also clear that many of the topics share a certain vocabulary and are hardly separatable. Furthermore a deeper research into topics used in the cryptocurrency sphere and their corresponding words will lead to a better seeding of the LDA and therefore to better results in clustering the text data. This was the main goal of the optimisation process. For this optimisation process it is necessary to understand how concentrated or sparce the seeded topics are distributed over the corpus, which is visible in Figure 6.4.

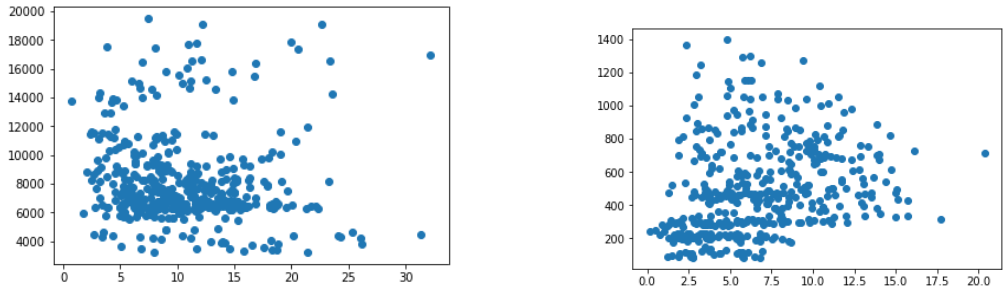
Figure 6.4: Topic distribution of topics found by the GuidedLDA algorithm.

The pearson correlation coefficient calculation returned very interesting results. For the topic "Bitcoin" there is little to no sufficient correlation between the number of news entries and trading volume. Nearly equal results were calculated for price development and market capitalization.

- Correlation between total number of entries with topic "Bitcoin" to its trading Volume: very low positive correlation - no statistical significance ($p\text{-}value > 0.05$)
  - Person coefficient: 0.028130
  - P-Value: 0.556191
- Correlation between total number of entries with topic "Bitcoin" to its price development: low negative correlation - statistically significant ($p\text{-}value < 0.05$)
  - Person coefficient: -0.105366
  - P-Value: 0.027101
- Correlation between total number of entries with topic "Bitcoin" to its market capitalization: low negative correlation - statistically significant ($p\text{-}value < 0.05$)
  - Person coefficient: -0.099154
  - P-Value: 0.037609

This result may occur because of the very high frequency of Bitcoin in the newsentries. As depicted in Figure 6.3(b) its the most salient term by far. This could be the reason for a kind of noise in the data concerning Bitcoin and therefore a biased topic model for this specific topic. In spite of this fact the total number of news entries on the topic of Ethereum show correlation on all market data values.

- Correlation between total number of entries with topic "Ethereum" to its trading Volume: low positive correlation - statistical significance ($p\text{-}value < 0.05$)
  - Person coefficient: 0.120931
  - P-Value: 0.011123



(a) (b)

Figure 6.5: Correlation values for Bitcoin total numbers of entries to close price (a) and the correlation values for Ethereum total numbers of entries to close price (b).

- Correlation between total number of entries with topic "Ethereum" to its price development: moderate positive correlation - statistically highly significant (p-$value < 0.05$)
  - Person coefficient:0.280192
  - P-Value: 2.219418e−9
- Correlation between total number of entries with topic "Ethereum" to its market capitalization: moderate positive correlation - statistically highly significant (p-$value < 0.05$)
  - Person coefficient: 0.271933
  - P-Value: 6.731043e−9

Ethereum is the second-in-line on the cryptocurrency market, is based on new concept of smart contracts and has very little shared vocabulary with other cryptocurrency news. Therefore this topic is well-distinguishable using a topic model. The promising results of positive correlation between the total number of Ethereum news and its corresponding market development states that the initial assumption of this project was right. It also shows that there is much room of improvement in the seeding list, in order to distinguish topics like bitcoin in more detail. Another interesting approach will be covered in the next semester, when verifying if there some sort of time lag. If news/tweets from today are affecting the market development of tomorrow or vice-versa.

6.3 Result of the optimization process

6.3.1 Analysis of the LDA process

The first overview of the Topic Distribution gives an insight into the most used terms and their connections - see Figure 6.6. Topic 1 shows an interesting connection between the word "Ripple" and the words "bank", "payments" and "technology". Further research using traditional methods (Google, Wikipedia, project website) found that Ripple is a technology providing a network for international transactions between banks. These are distinguishing words for the topic Ripple and should be included in the seeding

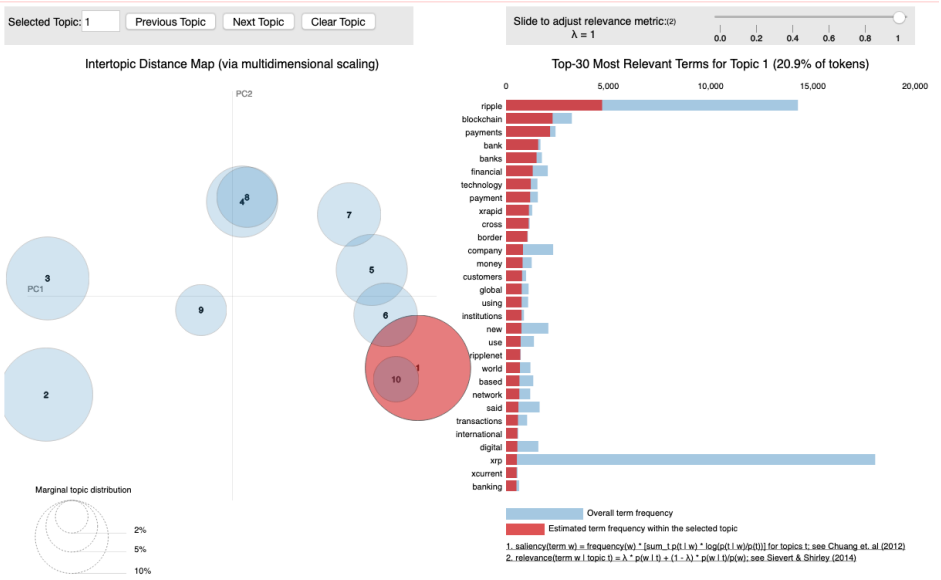


Figure 6.6: Result of the LDA analysis of the "Ripple"- Newsdataset.

list. Furthermore "xRapid", which describes a technical solution by Ripple, as well as "RippleNet", a blockchain made by Ripple, are added to the team. "xRapid" should therefore also be included. The more accurate the seeding list is, the more targeted the GuidedLDA algorithm's seeding process becomes.

6.3.2 Validation through experts and usage

If a seeding list has been compiled in the course of the project, which should consist of 6-8 words for each word and an access to industry experts is available, the list should be validated and evaluated by them. Such experts know most special expressions or the language in which the specific topic is reported and can provide valuable input to the list and the process. For this project the experts of blockpit.io from Linz were consulted and they validated the list of the Ripple seeding list. The words "garlinghouse" - the name of the CEO of Ripple, "centralistic" as Ripple Labs holds the majority of XRP tokens and is therefore a centralistic system and "SWIFT" because of the usecase of Ripple as possible successor of the banking technology "SWIFT", are added to the seeding list as a result of the validation process.

After the validation the new topic-word combination for the topic "Ripple" was added to the seeding list for the guidedLDA algorithm of this project. First results suggest that the topic Ripple can be better recognized and news better assigned. However, the changes to the seeding list only become apparent over a longer period of time. So the next few months will only show the full impact of the measure. It will then be extended to other topics in the project.

6.4 Challenges and Learnings

One of the biggest challenges of this project was the requisition of suitable data sets. A further challenge was to find one's way in the field of NLP, to find the suitable concept in the large mass of projects, techniques and theories and above all to understand the technical background of the algorithms. Especially interesting was the approach of not predicting the price but to look at the market development and to find connections. The resulting links to statistics and stochastics were demanding.

An interesting learning was that sentiment detection is even for humans a difficult task, which is shown by the score of 0.655 value of Krippendorff's Alpha according to Saif et al. [31] This score states the inter-annotator agreement (how humans agree on classification tasks). Social scientists normally take 0.8 as lower limit for valuable data. Especially the factory subjectivity, tone, context, polarity, irony and sarcasm are very difficult to distinguish in text data.

The most important learning is that a relevant and large dataset which is as clean as possible constitutes the success of a machine learning project. If the data is wrong, "dirty" or even irrelevant, a result can be falsified or no result can be obtained at all. An in-depth data research is the key to success. If this circumstance is not considered, the work can multiply after recognizing an error.

Concerning the data analysis it has been learned that sentiment analysis alone is not a powerful metric to perform a proper market research analysis. Rather, several analytical methods would have to complement each other in order to carry out valid analyses.

Chapter 7

Conclusion and Future Work

This thesis explored the mutual influence of sentiment and volume of tweets and news about cryptocurrencies and their respective market development. In order to measure this influence, rule-based sentiment analyses were performed on a Twitter dataset and probabilistic topic models (LDA and GuidedLDA) were applied to a news dataset. These datasets include about 50,000 tweets of cryptoinfluencers and over 80,000 news from the newsplattform dedicated to cryptocurrencies. While a statistically relevant correlation and thus the assumption of the research questions was confirmed, challenges were identified especially with the topic models, due to shared vocabulary. Therefore an optimization process was developed to make the used topic todels even more accurate. The technical components of the individual project parts were implemented in a fully automated webapp which reduces the administration effort and helps to ensure a good use of the data for future utilisation.

7.1 Best correlation values

On closer examination of the sentiment values of tweets, it was found that the general sentiment of the news does not correlate with the market development, but much more with the volume. The conclusion to be drawn: If a lot is tweeted about crypto, a lot is traded. The correlation calculation between the total amount of tweets per day and the market volume of the leading currency Bitcoin achieved the highest value of Pearson's r (Person coefficient: 0.311888, p-value: 5.613105e−18). It shows a moderate positive correlation which is statistically highly significant (p-value < 0.05).

The use of topic models in this thesis is grounded on finding and assigning topics subject to the mass of news data. This turned out to be especially difficult with certain topics within the news. The amount of common vocabulary especially with the topic "Bitcoin" or "Blockchain" makes it difficult for the probabilistic topic models to assign the news to certain topics. Even if the seeding technology of the guidedLDA algorithm can ease this problem, certain topics remain difficult, which is visible in the results of the correlation calculation. There is little to very little correlation between the volume of news assigned to the topic "Bitcoin" and each market metric used, although it is also statistically significant (Pearson's r between -0.105366 and 0.028130). On the opposite side, the topic models assigned the topic "Ethereum" very clearly and the

correlation values show a moderate to strong positive correlation between the volume of reporting on the topic Ethereum and the market metrics of the crypto currency Ethereum (Pearson's r between 0.120931 and 0.280192), all while beeing statistically moderate to highly significant (p-value from 0.011123 to 6.731043e−9).

7.2 Implications

The main benefit of automated data analysis, besides new insights and a deeper understanding of the matter, is the possibility to draw conclusions. The gut feeling, which the reporting on Twitter and news platforms influences the price development or vice versa, could be confirmed by this work. On the basis of this result, further things can be inferred or assumed. An arguable assumption could be that the price development influences the reporting, and that the correlation between the price development and the reporting on the following day is even stronger and thus a certain timelag is in the current version of the analysis. But this time related issue is not the only implication that surfaced in the processs of writing this thesis. Another challenging fact is the fast pace of the crypto world and also the shift in reporting about it. Thus, at the beginning of the research on this project the topics "ICO" or "Token Offering" were very popular, while at the time of writing of this work only little or hardly any coverage was noticeable. Due to this fast pace, the analysis software has to maintain a certain administrability and agility in order to quickly and correctly identify new topics over time that might influence the market and to include them in the analysis.

7.3 Outlook - Future Work

Given these implications, the possible next steps will focus mainly on refining the analysis algorithm. This includes the processing of the already mentioned possible timelag - i.e. the effect of the course on the reporting on the next day and vice versa - as well as the possibility to react quickly to new topics and to incorporate them into the analysis in a weighted way. Another important step would be to assign the Twitter messages to a certain topic or a certain crypto currency in order to enable more specific analyses and to add sentiment values of the individual tweets to the analysis in a meaningful way.

Apart from these purely technical possible steps in the future, the general situation on the cryptomarket should also find its way into future considerations. Thus the influence of Twitter is changing more and more in the direction of news platforms. While two years ago a single tweet could still trigger a "bull run" on a certain currency and the market jumped. Nowadays it is quite unlikely to happen as the market becomes more and more professional and approaches the behaviour comparable to the stock market. Future studies could fruitfully explore this issue further by reconsidering professional news convergence concerning the data sources and may find meaningful correlations.

Appendix A

Twitter accounts

Table A.1: All Twitter user used for crawling

Name	Twitterhandle	Category	on Twitter since
Charlie Lee	@SatoshiLite	Influencer	April 2008
Brock Pierce	@brockpierce	Influencer	April 2009
Joseph Young	@iamjosephyoung	Influencer	January 2014
Balaji S. Srinivasan	@balajis	Influencer	November 2013
Cryptobull	@CryptoBull	Influencer	May 2014
CryptoYoda	@CryptoYoda1338	Influencer	April 2017
Tone Vays	@ToneVays	Influencer	June 2014
Luke Martin	@VentureCoinist	Influencer	June 2017
Justin Untron	@justinsuntron	Influencer	August 2017
Nick Szabo	@NickSzabo4	Influencer	June 2014
Erik Voorhees	@ErikVoorhees	Influencer	July 2009
Dr. Anatoly Radchenko	@aradchenko1	Influencer	Mai 2011
Ian Balina	@DiaryofaMadeMan	Influencer	January 2010
Michael Suppo	@MichaelSuppo	Influencer	January 2014
Jake aka Korean Jew Trading	@koreanjewcrypto	Influencer	June 2017
Brian Armstrong	@brian_armstrong	Influencer	April 2008
Tim Draper	@TimDraper	Influencer	January 2009
Barry Silbert	@barrysilbert	Influencer	October 2011
Bobby Lee	@bobbyclee	Influencer	June 2011
Vinny Lingham	@vinnyLingham	Influencer	March 2007
Adam Back	@adam3us	Influencer	November 2010
Wheatpond	@wheatpond	Influencer	Juli 2011
Peter Todd	@peterktodd	Influencer	May 2013
Andreas M. Antonopoulos	@aantonop	Influencer	May 2013
Pavel Durov	@durov	Influencer	September 2008
Miko Matsumura	@mikojava	Influencer	November 2007
Ari Paul	@ariDavidPaul	Influencer	March 2017
Jihan Wu	@JihanWu	Influencer	February 2015
John McAfee	@officialmcafee	Influencer	November 2012
Roger Ver	@rogerkver	Influencer	August 2010

Table A.2: continuation of Table A.1

Name	Twitterhandle	Category	on Twitter since
Coindesk	@coindesk	Newspage	April 2013
Cointelegraph	@cointelegraph	Newspage	November 2013
Bitstamp	@Bitstamp	Exchange	August 2011
Bitmex	@BitMEXdotcom	Exchange	April 2014
Binance	@binance	Exchange	June 2017
Idexio	@idexio	Exchange	July 2017
Bithumb Official	BitthumbOfficial	Exchange	September 2017
Bittrex	BittrexExchange	Exchange	January 2014
Kraken	krakenfx	Exchange	May 2013
Bitfinex	bitfinex	Exchange	October 2012
Coinbase	coinbase	Exchange	May 2012
Bloomberg Crypto	crypto	Newspage	November 2017
Bloomberg Business	business	Newspage	April 2009
Vitalik Buterin	@VitalikButerin	Influencer	May 2011
Laura Shin	@laurashin	Influencer	March 2009
Chris Burniske	@cburniske	Influencer	March 2016
Coinmarketcap	@CoinmarketCap	Newspage	November 2013
Bitcoin Magazine	@BitcoinMagazine	Newspage	August 2011

Appendix B

Content of the CD-ROM

Format: CD-ROM, Single Layer, ISO9660-Format

B.1 PDF-File

Path: /

Essl_Wolfgang_2019.pdf Masterthesis

B.2 Additional content

Path: /project

project.zip . . . . . Zip file of all project files

read_me.pdf . . . . . Read me file for the project

Path: /images

*.pdf . . . . . vector images

*.jpg, *.png . . . . . raster images

References

Literature

[1] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. "SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining". In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. (Valletta, Malta). Ed. by Nicoletta Calzolari (Conference Chair) et al. Paris: European Language Resources Association (ELRA), May 2010, pp. 2200–2204 (cit. on p. 10).

[2] S. Behdenna, F. Barigou, and G. Belalem. "Document Level Sentiment Analysis: A survey". *EAI Endorsed Transactions on Context-aware Systems and Applications* 4.13 (Mar. 2018), pp. 1–7 (cit. on p. 11).

[3] Blei. David M., Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". *Journal of Machine Learning Research* (2003), pp. 993–1022 (cit. on pp. 6, 17).

[4] Avinash Chandra Pandey, Dharmveer Singh Rajpoot, and Mukesh Saraswat. "Twitter sentiment analysis using hybrid cuckoo search method". *Information Processing and Management* 53.4 (July 2017), pp. 764–779 (cit. on p. 12).

[5] George H. Chen, Stanislav Nikolov, and Devavrat Shah. "A Latent Source Model for Nonparametric Time Series Classification". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*. (Lake Tahoe, Nevada). NIPS'13. Red Hook, NY: Curran Associates Inc., Dec. 2013, pp. 1088–1096 (cit. on p. 13).

[6] Stuart Colianni, Stephanie Rosales, and Michael Signorotti. *Algorithmic Trading of Cryptocurrency Based on Twitter Sentiment Analysis*. Tech. rep. Stanford, CA: Stanford University, 2015. URL: http://cs229.stanford.edu/proj2015/029_report.pdf (cit. on p. 13).

[7] Scott Deerwester et al. "Indexing by latent semantic analysis". *Journal of the American Society for Information Science* 41.6 (1990), pp. 391–407 (cit. on p. 17).

[8] Christiane Fellbaum. "A Semantic Network of English: The Mother of All WordNets". *Computers and the Humanities* 32.2 (Mar. 1998), pp. 209–220 (cit. on p. 10).

[9] Francis Galton. "Regression towards mediocrity in hereditary stature". *The Journal of the Anthropological Institute of Great Britain and Ireland* 15 (1886), pp. 246–263 (cit. on p. 20).

[10] Michael Gamon et al. “Predicting Depression via Social Media”. In: *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media*. (Boston, USA). Cambridge, Massachusetts: Association for the Advancement of Artificial Intelligence, July 2013, pp. 128–137 (cit. on p. 10).

[11] Ifigeneia Georgoula et al. “Using Time-Series and Sentiment Analysis to Detect the Determinants of Bitcoin Prices”. In: *MCIS 2015 Proceedings of the Mediterrean Conference on Information Systems*. (Samos, Greece). Associations for Information Systems, Oct. 2015, pp. 815–821 (cit. on p. 13).

[12] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA: O’Reilly Media, 2017 (cit. on p. 4).

[13] M. Ghiassi, J. Skinner, and D. Zimbra. “Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network”. *Expert Systems with Applications* 40.16 (2013), pp. 6266–6282 (cit. on p. 12).

[14] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-term Memory”. *Neural computation* 9.8 (Nov. 1997), pp. 1735–80 (cit. on p. 12).

[15] Thomas Hofmann. “Probabilistic Latent Semantic Indexing”. In: *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. (Berkeley, California, USA). SIGIR ’99. New York, NY, USA: ACM, 1999, pp. 50–57 (cit. on p. 17).

[16] C. J. Hutto and Eric Gilbert. “VADER: A parsimonious rule-based model for sentiment analysis of social media text”. In: *Proceedings of the 8th International AAAI Conference on Weblogs and Social Media (ICWSM-2014)*. (Ann Arbor, MI). Cambridge, Massachusetts: Association for the Advancement of Artificial Intelligence, June 2014, pp. 216–225 (cit. on pp. 9, 13, 22).

[17] Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. “Incorporating Lexical Priors into Topic Models”. In: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. (Avignon, France). Stroudsburg, PA: Association for Computational Linguistics, Apr. 2012, pp. 204–213 (cit. on p. 18).

[18] Rie Johnson and Tong Zhang. “Semi-supervised Convolutional Neural Networks for Text Categorization via Region Embedding.” *Advances in neural information processing systems* 28 (2015), pp. 919–927 (cit. on p. 12).

[19] Karen Sparck Jones. “A statistical interpretation of term specificity and its application in retrieval”. *Journal of Documentation* 28.1 (1972), pp. 11–21 (cit. on p. 16).

[20] Alistair Kennedy and Diana Inkpen. “Sentiment Classification of movie reviews using contextual valence shifters”. *Computational Intelligence* 22.2 (May 2006), pp. 110–125 (cit. on p. 11).

[21] Klaus Krippendorff. “Estimating the reliability, systematic error and random error of interval data.” *Educational and Psychological Measurement* 30.1 (1970), pp. 61–70 (cit. on p. 14).

[22] Klaus Krippendorff. “Reliability in Content Analysis”. *Human Communication Research* 30.3 (2004), pp. 411–433 (cit. on p. 14).

[23] Bing Liu. *Sentiment Analysis and Opinion Mining*. 1st ed. Williston, VT: Morgan & Claypool, May 2012 (cit. on pp. 5, 8, 14).

[24] H. P. Luhn. “A Statistical Approach to Mechanized Encoding and Searching of Literary Information”. *IBM Journal of Research and Development* 1.4 (1957), pp. 309–317 (cit. on p. 16).

[25] Feng Mai et al. “How Does Social Media Impact Bitcoin Value? A Test of the Silent Majority Hypothesis”. *Journal of Management Information Systems* 35 (Jan. 2018), pp. 19–52 (cit. on p. 1).

[26] Mika V Mäntylä et al. “The evolution of sentiment analysis – A review of research topics, venues, and top cited papers”. *Computer Science Review* 27.2 (2018), pp. 16–32 (cit. on p. 8).

[27] Bruno Ohana and Brendan Tierney. “Sentiment classification of reviews using SentiWordNet”. In: *School of Computing 9th. IT & T Conference*. (Dublin, Ireland). Dublin, Ireland: Dublin Institute of Technology, Oct. 2009, pp. 13–21 (cit. on p. 12).

[28] Bo Pang and Lillian Lee. “A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts”. In: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics ACL-04*. (Barcelona, Spain). ACL, July 2004, pp. 271–278 (cit. on p. 11).

[29] Karl Pearson. “Notes on regression and inheritance in the case of two parents”. In: *Proceedings of the Royal Society of London*. (London). v. 58. Taylor & Francis, 1895, pp. 240–242 (cit. on p. 20).

[30] Jonathan K. Pritchard, Matthew Stephens, and Peter Donnelly. “Inference of Population Structure Using Multilocus Genotype Data”. *Genetics* 155.2 (2000), pp. 945–959 (cit. on p. 17).

[31] Hassan Saif et al. “Evaluation datasets for Twitter sentiment analysis: a survey and a new dataset, the STS-Gold”. In: *1st Interantional Workshop on Emotion and Sentiment in Social and Expressive Media: Approaches and Perspectives from AI (ESSEM 2013)*. (Turin, Italy). Turin, Italy: University of Turin, Department of Computer Science, 2013, pp. 9–21 (cit. on pp. 14, 53).

[32] Arthur L. Samuel. “Some Studies in Machine Learning Using the Game of Checkers”. *IBM Journal of Research and Development* 3 (1959), pp. 210–229 (cit. on p. 4).

[33] MMag. Dr. Niklas Schmidt. *Kryptowährungen und Blockchains*. 1st ed. Wien: Linde Verlag, 2019 (cit. on p. 4).

[34] Devavrat Shah and Kang Zhang. “Bayesian regression and Bitcoin”. In: (Urbana-Champaign). Vol. 2. 1. Urbana, IL: Coordinated Science Laboratory GRAINGER COLLEGE OF ENGINEERING, 2014, pp. 409–414 (cit. on p. 13).

[35] Evita Stenqvist and Jacob Lönnö. *Predicting Bitcoin price fluctuation with Twitter sentiment analysis*. Tech. rep. Stockholm, Sweden: KTH, School of Computer Science and Communication (CSC), 2017. URL: http://www.diva-portal.org/smash/get/diva2:1110776/FULLTEXT01.pdf (cit. on p. 13).

[36] Philip J Stone et al. "The general inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information". *Behavioral Science* 7.4 (1962), pp. 484–498 (cit. on pp. 10, 22).

[37] Duyu Tang, Bing Qin, and Ting Liu. "Document Modeling with Gated Recurrent Neural Network for Sentiment Classification". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. (Lisbon, Portugal). Stroudsburg, PA: Association for Computational Linguistics, Sept. 2015, pp. 1422–1432 (cit. on p. 12).

[38] Yla R. Tausczik and James W. Pennebaker. "The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods". *Journal of Language and Social Psychology* 29.1 (2010), pp. 24–54 (cit. on pp. 10, 22).

[39] A. Tumasjan et al. "Predicting elections with twitter: What 140 characters reveal about political sentiment". In: *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*. (Washington, DC). Cambridge, Massachusetts: Association for the Advancement of Artificial Intelligence, May 2010, pp. 178–185 (cit. on p. 10).

[40] Jiacheng Xu et al. "Cached Long Short-Term Memory Neural Networks for Document-Level Sentiment Classification". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. (Austin, Texas). Stroudsburg, PA: Association for Computational Linguistics, Nov. 2016, pp. 1660–1669 (cit. on p. 12).

[41] Lei Zhang, Shuai Wang, and Bing Liu. "Deep learning for sentiment analysis: A survey". *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8.4 (2018) (cit. on p. 12).

[42] Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. "Attention-based LSTM Network for Cross-Lingual Sentiment Classification". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. (Austin, Texas). Stroudsburg, PA: Association for Computational Linguistics, Nov. 2016, pp. 247–256 (cit. on p. 12).

Online sources

[43] EUWAX Aktiengesellschaft. *Cryptoradar by BISON*. 201. URL: https://bisonapp.de/en/radar/ (visited on 03/11/2019) (cit. on p. 1).

[44] Anaconda and Bokeh Contributors. *Bokeh Plotting Library for Python and Django*. 2019. URL: https://bokeh.pydata.org/en/latest/ (visited on 05/18/2019) (cit. on pp. 42, 43).

[45] Tweepy API. *API Wrapper for the TwitterAPI*. 2018. URL: http://docs.tweepy.org/en/v3.5.0/api.html (visited on 01/12/2019) (cit. on p. 34).

[46] Simon Brown. *System Context Diagram*. 2019. URL: https://c4model.com/ (visited on 05/18/2019) (cit. on p. 41).

[47] Coinmarketcap. *Coinmarketcap Historical Data Pages for Cryptocurrencies*. 2018. URL: https://coinmarketcap.com/currencies/bitcoin/historical-data/ (visited on 01/29/2019) (cit. on p. 35).

[48] Cryptocompare. *Cryptocompare News API*. 2018. URL: https://min-api.cryptocompare.com/documentation?key=News&cat=latestNewsArticlesEndpoint (visited on 01/29/2019) (cit. on pp. 35, 38).

[49] cryptocompare.com. *Sentiment analysis on cryptocurrency news*. 2018. URL: https://blog.cryptocompare.com/sentiment-analysis-on-cryptocurrency-news-d28923b356be (visited on 01/12/2019) (cit. on pp. 2, 12, 18).

[50] Github.com. *Top languages over time*. 2018. URL: https://octoverse.github.com/projects#languages (visited on 05/16/2019) (cit. on p. 29).

[51] Google Inc. *Google Cloud API for Natural Language Processing*. 2018. URL: https://console.cloud.google.com/apis/library/language.googleapis.com?project=news-text-sentim-1548515499669&folder&organizationId (visited on 01/29/2019) (cit. on pp. 37, 47).

[52] MonkeyLearn Inc. *Nearly everything you need to know about sentiment analysis*. 2018. URL: https://monkeylearn.com/sentiment-analysis (visited on 01/08/2019) (cit. on pp. 11, 15).

[53] INRIA. *Scikit-learn machine learning library*. 2019. URL: https://scikit-learn.org/stable/ (visited on 05/16/2019) (cit. on p. 30).

[54] LIWCOnline. *LIWC Online Sentiment Analyse*. 2018. URL: http://liwc.wpengine.com/ (visited on 01/29/2019) (cit. on p. 37).

[55] A.V. Prokhorov (originator). *Correlation (in statistics)*. 2011. URL: http://www.encyclopediaofmath.org/index.php?title=Correlation_(in_statistics)&oldid=11629 (visited on 05/16/2019) (cit. on p. 19).

[56] Python. *Python programming language*. 2019. URL: https://www.python.org/ (visited on 05/16/2019) (cit. on p. 29).

[57] Google Scholar. *Citation data for the paper "Latent dirichlet allocation"*. 2019. URL: https://scholar.google.at/scholar?hl=de&as_sdt=0%5C%2C5&q=latent+dirichlet+allocation&btnG=&oq=Latent+ (visited on 04/22/2019) (cit. on p. 17).

[58] Vikash Singh. *GuidedLDA*. 2018. URL: https://guidedlda.readthedocs.io/en/latest/ (visited on 01/12/2019) (cit. on pp. 18, 28).

[59] django Softwarefoundation. *Djanog Python Web Framework*. 2019. URL: https://www.djangoproject.com/ (visited on 05/16/2019) (cit. on pp. 29, 40).

[60] text-processing.com. *Text Processing API*. 2018. URL: http://text-processing.com/demo/sentiment/ (visited on 01/29/2019) (cit. on pp. 13, 37).

[61] United Traders. *Top 50 Blockchain and Crypto Twitter Accounts to Follow*. 2018. URL: https://medium.com/@Uttoken.io/top-50-blockchain-and-crypto-twitter-accounts-to-follow-3a343948d176 (visited on 01/29/2019) (cit. on p. 34).

[62] Twitter. *Twitter Search API Documentation*. 2018. URL: https://developer.twitter.com/en/docs/tweets/search/overview (visited on 01/29/2019) (cit. on p. 34).

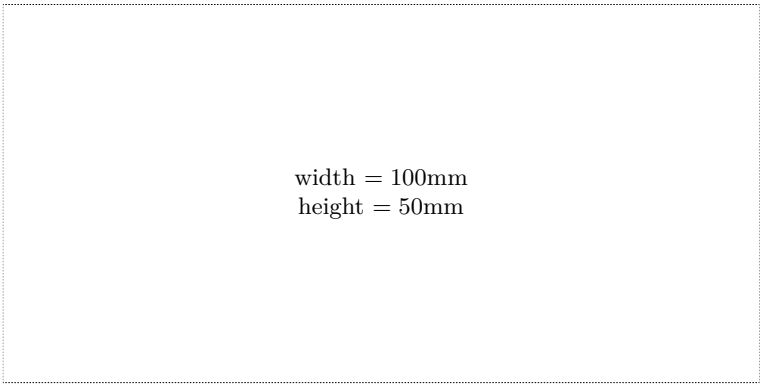
[63] Twitter. *Twitter User Timeline API Documentation*. 2018. URL: https://developer.twitter.com/en/docs/tweets/timelines/overview (visited on 01/29/2019) (cit. on p. 34).

[64] Analytics Vidhya. *A Stepwise Introduction to Topic Modeling using Latent Semantic Analysis*. 2018. URL: https://www.analyticsvidhya.com/blog/2018/10/stepwise-guide-topic-modeling-latent-semantic-analysis/ (visited on 04/20/2019) (cit. on p. 16).

[65] Eric W. Weisstein. *"Bivariate."* 2019. URL: http://mathworld.wolfram.com/Bivariate.html (visited on 05/16/2019) (cit. on p. 19).

Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —