AALBORG UNIVERSITY
COPENHAGEN

Semester: 4th Semester

Title: Twitter Data Mining

Project Period: Spring 2016

Aalborg University Copenhagen
A.C. Meyers Vænge 15
2450 København SV

Semester Coordinator: Henning Olesen

Secretary: Maiken Keller

Semester Theme:
Master Thesis

Supervisor(s): Samant Khajuria

Project group no.: 4.1

Members
(do not write CPR.nr.):

Marek Holub

Abstract:

Social media generates massive amounts of data every minute, which is caused by its mainstream adoption over the past years. Innovations in the industry have enabled new ways of communications between people and created many business opportunities. Big Data in social media require effective and advanced processing technologies. Purpose of data mining analyses is to find valuable patterns and insights from Twitter data.

Aim of this project is to develop a concept for sentiment analyses that will be able to classify streamed tweets by sentiment polarity. Focus will be also on Big Data architecture design that is capable to process Twitter stream on clusters. Data mining techniques, together with natural language processing are implemented in order to find context behind data. Moreover, sentiment score will be visualized on graph.

Pages: 95
Finished: 09/06/2016

Acknowledgements

It has been an amazing experience studying at the Aalborg University Copenhagen. From the first day of my master's degree in Innovative Communication Technologies and Entrepreneurship I was motivated and excited about study, which completely fulfilled my expectations.

I would like to especially thank my supervisor, Samant Khajuria for his advices, comments and discussions during this thesis period. Moreover, I would like to thank CMI department for interesting courses, teaching and friendly environment. Furthermore, I would like to thank to my family for their support during my studies.

Table of Contents

Reading Guide

Chapter 1 – Introduction

This chapter deals with the introductory area of this research, where the data mining connects with Big Data, social networks and sentiment analyses. Moreover, Twitter is researched as a service that may create many opportunities for data mining. Problem formulation is defined, together with project delimitations that limit the projects scope.

Chapter 2 – Methodology

Methodology chapter defines the research approaches in order to solve the problem formulation and reach the conclusion. It includes sub-chapters that define various steps that begin with project planning, desktop research, Twitter data analysis, experimental approaches and concept development.

Chapter 3 – State of the Art

Discusses the technologies used nowadays for applying data mining techniques on the datasets. Big Data from Apache are researched as well as third party platforms and databases. It also discusses various machine-learning algorithms that can be implemented on Twitter data. Moreover, this chapter contains research on the related work from the academy and industry.

Chapter 4 – Twitter and Apache Spark

Chapter discusses the Twitter sentiment implementation with Apache Spark. Spark Streaming process is described with discrete stream. Moreover, architecture design for sentiment analyses is included.

Chapter 5 – Twitter Classification

Chapter takes a look on the Naïve Bayes classification with Bayes theorem. Bernoulli model of Naïve Bayes is used as example to classify Twitter words. Posterior probability predicts the sentiment polarity for given words.

Chapter 6 – Twitter Data Analysis

Connection to Twitter Streaming API and Twitter application setup are discussed first in this chapter. Later, structure and format of Twitter data are analysed and visualized from data frame. Moreover, natural language processing analyses and tweets normalization are also included.

Chapter 7 – Concept Development

Concept development is discussed within this chapter. Supervised machine learning approach and implementation with natural language processing are deployed first. Classification part begins with Naïve Bayes classifier training and its accuracy. Concept also discusses approach with explicitly labelled tweets. Outcome is visualized on graph. Moreover, graphical user interface is presented.

Chapter 8 – Conclusion and Future Work

Chapter concludes the outcomes in relation to research question. Moreover, suggestions for future work are discussed.

1 Introduction

Over the past years, the amounts of data generated from the Internet services have increased significantly. Innovations in the field of ICT have enabled new business opportunities for creating services capable of handling vast data volumes. Technology reached the level, where people are interconnected with social media on daily basis and are able to share their life over social networks.

Social networking over Internet has become popular in the last years, which is also justified with the increased data volumes. New challenges appeared in relation to data storage architectures with scalability features and effective processing algorithms.

Data mining analysis has a great potential in finding meaningful insights within social networks data. Twitter social network is a service developed in order to enable communication between people by sending short messages. [1]

According to research [2], Twitter as the second largest social media platform, right behind Facebook, generates around 350, 000 tweets each minute, or 21 million per hour. Such volumes of data present challenges for engineers to develop innovative solution for effective data architecture and processing capabilities in order to apply data mining. The importance of Big Data implementations within enterprises in various sectors, such as health industry, retail, telecom or social networks plays crucial scenario in optimizing business processes and creating new value propositions for revenue streams.

According to Accenture research [3], 87% of enterprises believe that Big Data will reshape the industry in the next years. Therefore, the early adoption of this trend may create additional value to enterprises in sense of data mining of customer's opinions about company in the Twitter use case. Knowing what customers think about the services or how services can be innovated in the future gives companies insights and strategies for development. Furthermore, survey also found that 89% of respondents said that companies who do not adapt to this Big Data trend would risk losing market share. [3]

Twitter has big potential for data mining as its users produce Big Data that can be processed. In addition, there are requirements for architecture development that can scale to continuous new-streamed tweets and also ability to integrate with

advanced machine learning algorithms. Knowing what users think or how they feel about products is valuable proposition for companies.

Sentiment analyses are part of data mining, which monitors public perceptions about various topics. It can analyse what people think about business products and quality, brands, pricing strategies or worldwide trends. Moreover, it can identify business opportunities and thus become an effective factor for companies to innovate their services. [4]

Twitter as micro blogging platform backed with its active users create opportunities for data mining and more particular sentiment analyses based on tweets. Twitter users often express their opinions about various topics within their posted tweets. And so by applying text-processing data mining technique can serve companies as feedback or for brand management.

On the other hand, since Twitter generates massive volumes of data every day, sentiment analyses can help with marketing related campaigns to research public opinions about newly released product, for example blockbuster movie and analyse sentiment about users satisfaction, whether they felt positive or negative about movie. According to [4] consumers are willing to pay from 20% to 99% more for movie rated with 5/5 stars. This research discusses that positive comments or reviews on product are great influencers and will indicate success among consumers.

1.1 Motivation and Background

Twitter is an online social network, where users can share short messages, called tweets. Service was launched in 2006 with headquarters in California, USA. During the years of its deployment gained worldwide popularity with 320 million active users. [5] Users signed in to service can share their moments with friends, include links to pictures or videos in tweets, make comments and re-tweet other tweets. Twitter also supports functionality of following other users.

Apart from creating personal accounts, many enterprises create their own in order to engage with customers and get feedback on products, promote company activities, discounts or newsletters. Therefore, Twitter became interesting platform for branding and marketing.

Twitter represents the effective way of communication between friends, users or customers. Hence, data mining on top of Twitter can result in interesting insights about users and create value for companies.

Based on the article [6], Twitter data strategy chief Chris Moody talks about cases when Twitter data mining can took place in real-world scenarios. For example, thanks to Twitter, Aircraft Company could surprise one of their customers travelling on board with little present, when they discovered she was traveling to see her new grandchild. Chris arguing that such scenarios can happen on daily basis. Moreover, he says that Twitter gives unique opportunity to understand people in context like never before.

Another use case took departure in gender prediction by using machine learning over Twitter. Algorithm can learn and determine if Twitter accounts belongs to man or woman. [7]

1.2 Twitter

Twitter allows people communicate with short messages called tweets. Each tweet can contain a maximum of 140 characters of text. According to Twitter website [1] 140 characters presents perfect length for sending status updates via text messages. Furthermore, 20 more characters are reserved for people's names. Once users sign up into service and register for free account, members can send tweets or follow other members to be updated about the latest news. These short messages are posted in users own profile. Moreover, they can be sent to followers and be searchable on Twitter search. [1]

Social networking service is not limited only to website access, but users can engage and interact with service through applications developed for smart devices. In fact, according to Twitter usage figures [5], there are 80% of active users on mobile. Popularity over micro blogging gained success across the world, which is supported by fact that Twitter has around 320 million active users who engage with the service on the monthly basis.

User generated content can create many opportunities for marketing and advertising cases in which the data mining techniques are used. Twitter is useful for reading and finding interesting topics that catches user's attention. People can discover real time news about what's happening in the world or stay in touch with

friends. On the other hand, many companies use Twitter to keep customers updated about offers and deals. Textual context of tweets has a relationship with two additional metadata that are divided into entities and places. Tweet entities are user mentions, which represent way of mentioning other users in own tweets by including @ sign, followed by their username. [8] Furthermore, tweet entities may contain also hashtags and URLs. On contrary, tweet places represent real world locations that may be integrated to a tweet. [9]

Terminology is an important part of the Twitter, because it teaches users about the service functionality and features. Moreover, it defines aspects of Twitter and various possibilities how to use it. In order to understand the Twitter terminology, a brief overview is presented. [8] Twitter use at (@) sign in order to call someone username in the tweet or send user a message. Moreover, this sign is used whenever user wants to create connection with other user and link to his Twitter profile. Username uniquely identifies each user and are generally used with @ sign, for example, Andy Murray is @andy_murray. [11]

Another popular symbol used on Twitter is called hashtag. It helps users to categorize messages. In fact, the structure of it comes with the # sign followed by the relevant keyword in connection to tweet message. Essentially, it helps categorize the tweets based on its context and enables better search results by Twitter Search. [10] Hashtags can be located anywhere in the tweet. When users click on it, they will be directed to category that groups all tweets from the Twitter users within the same topic.

In tweet: *“Great start for @JamesWardtennis and @Evo151216 here at aussie open in brutal conditions. Over 40 degrees today. #1down2togo”*, [11] famous tennis player Andy Murray is writing about success of his British colleagues James Ward (@JamesWardtennis) and Dan Evans (@Evo151216). They advanced to the next round of the qualification for tennis tournament played in Australia with hashtag (#1down2togo). Tweet contains positive sentiment statement.

Apart from the above mentioned most used symbols, Twitter uses other expressions in its terminology. One of them is “bio”, which is dedicated to short description of the user profile of 160 characters.

Twitter also supports sending direct private messages between users, hence is implement basic messaging functionality but with added value of micro blogging service as mentioned before. This fact is the main advantage in comparison with standalone messaging services, such as WhatsApp or Viber.

Ability to subscribe to a different Twitter accounts is known under term “following”. When user decides to follow other account, Twitter will update his profile with the latest tweets. Such user becomes in the terms of Twitter “follower”. Number of followers is shown in the user profile overview.

Geotag is used to inform users about the location where the creator of the tweet was located during posting it.

Button “Like” shows the positive reaction towards the tweets likeness. User profile enables users to create dedicated lists that will group all topics into categories. Better navigation and coherence between points of interest matters.

Recommender algorithms can suggest users “Who to Follow”. Information took place in the profile page and usually it consists of promoted Twitter accounts, trends and tweets by advertisement.

Reply option comes with each posted tweet. It gives possibility to react and send response.

Moreover, retweet option is part of every message. It has functionality of sharing ones tweet with own “followers”. If the tweet consists of more then one @ sign, it is called “mention”. [8]

1.3 Big Data Characteristics

Big Data are characterized as data that overcome the capabilities of traditional processing by performance and requirements, which can be categorized into 5 groups. Hence, the Big Data [9] are determined by 5 “V”s: volume, velocity, variety, veracity and value.

Volume characterizes the total amount of data generated by services. Adoption of social media, as Twitter lead to increased demands for advanced database systems.

Velocity describes the speed in which data are created Twitter users. Such demands on the data architecture within enterprises require powerful and scalable processing.

Variety implies to data formats, such as video or audio, but also there is difference between unstructured and structured formats.

Veracity deals with data relevancy, which is necessary to reach before any data analyses. Redundant information and noise has to be removed from data by normalization tasks.

Lastly, value if data concerns about finding valuable insights, e.g. with data mining techniques. Value proposition can be reached by sentiment analyses on Twitter by finding data patterns and initializing machine learning algorithm. [56] Summary of Big Data characteristics depict Figure 1.

Big Data	
Volume	Terabytes, records, tables, files
Velocity	Batch, Real/near-time, streams
Variety	Structured, unstructured, probabilistic
Veracity	Authenticity, availability, trustworthy
Value	Correlations, statistical, insights

Figure 1: Big Data characteristics [56]

1.4 Sentiment Analyses Overview

Sentiment analyses deals with the opinion mining over user perception about particular reality. It is behavioural analyses that evaluate users information in order to recognize and find. Information availability concerns with the privacy and security of personal credentials. However, the technologies evolved and solutions for developers enabled access to service with application programming interface (API).

People share ideas, opinions via blogs and social networks, which bring new opportunities for developing innovative service solutions. Sentiment describes analyses in which opinions from data are extracted with different technologies and data mining techniques. Users on social networks can express their opinions on particular product, service, brand, and various events such as sport championships or political elections. Twitter profiles has advantage to other social networks such as Facebook, because the information (tweets) are publically available, if private

account settings are not set. Hence, one can view Twitter feed even without owning Twitter account. [4]

Insights from the sentiment analyses can be valuable resource for marketers, pooling, and more. Moreover, sentiment analyses can result as feedback information to track users product preferences and capitalize it to stay ahead from competition on the market.

1.5 Problem Formulation

Social media generates Big Data volumes, which bring new challenges for the services. Storing, analysing and processing of such massive datasets become relevant for developers to handle and offered opportunities to create new innovation solutions. Streaming Twitter data to client presents opportunities for applying data mining techniques and finding valuable insights, which can help marketing departments and companies learn about their current customers or attract new ones with interesting product offers.

Moreover, pooling method, which is used to track public opinions on specific event offer wide opportunists for concept development.

Furthermore, machine learning has been to great development over the years. Its implementation in the concept can lead into classification algorithm that will process tweets and predict their sentiment.

Based on the above study, the research question for this project is defined as:

How to apply data mining techniques on Twitter social network and find valuable insights about its users with sentiment analyses?

Data mining in relation with machine learning and Big Data can create competitive advantage for companies. Approaches to sentiment analyses have many challenges. Therefore, the research must study various techniques from this domain. In addition, research question supports analyses on technologies, algorithms, Big Data architecture and output visualization.

1.6 Project Delimitations

The aim of this thesis is focused on data mining within social networks that will analyse Twitter data stream with algorithms in order to reveal insights behind tweets. Goal will be reached by using state of the art information technologies. Discussing the delimitations in this chapter enables project to specify outline and scope of the research. Data mining in the field of social networks and Big Data has many challenges that have to be researched. Hence, the timeframe for the thesis, software and hardware capabilities can limit the projects scope.

Limitations will help to keep focus on the aspects of the project that are vital for reaching the problem formulation. They can be categorised as follows:

- Twitter Streaming API is limited to stream rates that allow to stream only small part of the total volume of tweets
- Hardware capabilities – approach and implementation of sentiment analyses for Big Data with Apache Spark would require powerful computing and clusters, therefore, natural language processing prioritized
- Available dataset with pre-labelled tweets with sentiment polarity – classification would need political oriented tweets for training and testing
- Twitter Analyses is implemented on tweets related to US elections 2016 as the text from this tweets contain in most cases opinion statements
- Classification is limited to Naïve Bayes classifier
- Tweets are divided into two sentiment polarities – positive and negative, thus there can be strong inclination towards negative sentiment as the classifier classifies by default all neutral and unprocessed tweets into negative category
- Tweets may contain bias, which may be difficult to recognize by classifier
- Credentials used for creating Twitter application in order to obtain access token are authors. Impacts are discussed in the conclusion.
- Solution works as concept and is limited to offline application. Web-based approach is discussed in the future work
- Privacy and security issues with Twitter processing are not on the projects scope
- Business aspects are also not on the projects scope

2 Methodology

Methodology chapter is dedicated for describing methodological approach in order to answer projects research question. Firstly I started with brainstorming process, which allows defining initial keywords for this project. My keywords list contained words as: big data, processing, data mining, machine learning and social media, Twitter.

Secondly, I started to formulate problem definition and narrow down the scope to be more focused on specific research field. Then, I continued with desktop research by studying scientific papers and publications within defined scope. I researched state of the art technologies in relation to Big Data applications, machine learning algorithms and relevant work from academy and industry.

Methodology continues with collecting data from Twitter with API and performing data analyses with chosen state of the art technologies. Moreover, methodological process deals with the pre-processing tasks and data normalization. Furthermore, data mining on Twitter is developed with sentiment analyses approach and visualization with graph. Lastly, conclusion and future work are discussed.

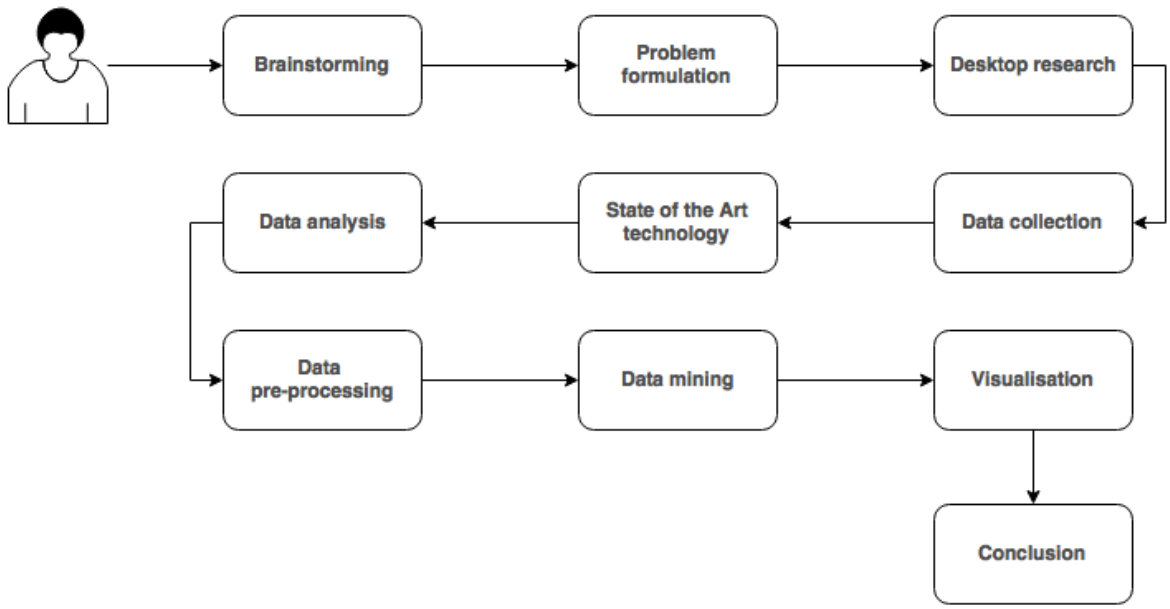


Figure 2: Twitter data mining methodology

Broader discussion on methodology is presented in the next sub-chapters, which includes: project planning, desktop research, twitter data analyses, experimental approach and concept development.

2.1 Project Planning

In the beginning phases, project has been divided into timescale that served as tool for keeping track with thesis deadline. For this purpose, Gantt charts have been planned and can be found in Appendices.

2.2 Desktop Research

Methodological framework started with brainstorming sessions when I initiated generating possible research questions for this project. One of the ICT segments that took my attention is data mining. Techniques they implement can be deployed over Big Data, which presents another arising trend within ICT. Therefore, I have started defining problem formulation, which can deal with challenges within data mining, processing and Big Data, and micro-blogging social network, Twitter. Furthermore, it helped me to narrow down the scope of this project to data mining with sentiment analyses.

In order to discuss and reflect the challenges from problem formulation, I started with desktop research, which helped me to understand theory behind this ICT field of study. My research started from studying relevant academic papers accessed on Internet, latest articles from state of the art technologies, and data mining and Big Data publications. Theory studied from the mentioned sources helped me to gain insights about position of Twitter in the architecture design, which could handle real-time streaming data, pre-processing, data mining and visualization. Broader knowledge from theoretical framework enabled me to later implement sentiment analyses as data mining technique over Twitter. Literature study helped me to identify state of the art technologies, and various algorithms that can be applied within data mining.

Theoretical approach of sentiment analyses helped me to understand challenges behind text processing and methodology how it can be implemented with

Twitter. Purpose of desktop research was to get insights and understating on the technologies, algorithms, and theories that focuses on Twitter data mining.

2.3 Twitter Data Analyses

Next step in the methodology was dedicated to Twitter data collection and its analyses. As Twitter offers publically accessible Application Programming Interface (API) my goal was to analyses different approaches to API, tokens, their implementation from theoretical side and finally collect tweets for sentiment analysis in order to develop classification method for training and testing. After collecting sample data set from Twitter, I started with data analyses in order to understand data structure of Twitter stream. Data analyses on tweet attributes are important in order to understand its format and prepare tweets for sentiment analyses, pre-processing phase and classification.

2.4 Experimental Approach

Experimental approach is part of methodological framework, which discusses process of tweets manipulation after they were collected from Twitter Streaming API. Problem question of this project requires to experiment with collected tweets in a way suitable for text processing and sentiment analyses. Therefore, I decided to divide experimental approach into separated tasks according their order. Each tweet contains various attributes, such as: timestamp, text, created_at, favourite_count, id, hashtag, URL and emoticons. Hence, the first experiments will lead towards tweet pre-processing.

Since sentiment analyses are based on getting sentiments insights (positive or negative) from the tweet text, experimental approach is divided into data cleaning, data selection and data transformation. Data cleaning is an important task for preparing dataset for further analyses. It means that each tweet text is tokenized and divided into sequence of single words separated by comma. In addition, noise from dataset such as stop words are filtered out to remove redundancy. Data selection phase is important for retrieving relevant data for further analysis. Moreover, the last phase of pre-processing is data transformation, which will prepare tweets text format for implementing algorithm that will extract sentiment.

Purpose of pre-processing is to reduce unnecessary attributes from tweets and filter out only relevant features for data mining sentiment algorithm. Important part of the experimental approach is also investigation of different classification algorithms from supervised machine learning. Purpose of these algorithms is to build a model that will make predictions based on pre-labelled data, on which they are trained. According to this input data, algorithms can train their accuracy based from model observations and further use this model for making new predictions on completely unknown datasets.

2.5 Concept Development

Concept development is part of the methodological framework, which is dedicated to solution development for Twitter sentiment analyses. Experimental approach gave the broader perspective of working with Twitter data and streaming processes.

Data mining offers different approaches to sentiment analyses with various algorithmic techniques. At the beginning, the architecture for Big Data concept solution is discussed and elaborated. Technology is capable to create real-time concepts with emphasis on Big Data characteristics and Twitter streaming API. Real-time processing capabilities are discussed and supported by concept architectural design for sentiment analyses. In addition, text classification algorithm with Naïve Bayes is included within this framework, which discusses its application on Twitter sample dataset.

Understanding the processing with Naïve Bayes is important for sentiment analyses on Twitter. Moreover, natural processing language approach takes part, which is considered as the building concept for solution development after previous data analyses and study. Classification algorithm is applied on normalized tweets from experimental approach. Moreover, it performs as the algorithm that evaluates sentiment outcome from streamed tweets.

Visualization is achieved by showing the sentiment curve on live graph after user specifies particular keyword. Furthermore, graphical interface concept is developed in order to enhance user experience with Twitter sentiment tool.

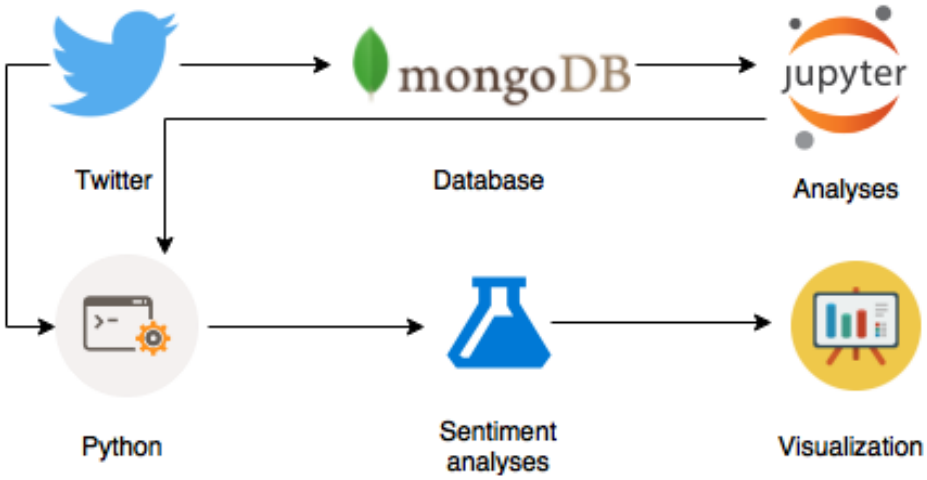


Figure 3: Twitter sentiment analyses methodology

Methodological approach for Twitter sentiment analyses is illustrated on Figure 3, where tweets from Twitter Streaming API are stored in MongoDB database in order to perform data analyses. Hence, tweets data format and attributes are analysed with Jupyter, an interactive Python interface, in order to understand their structure and prepare for data normalization method. This approach helps to use Python development for creating API connection to Twitter and develop classification algorithm for sentiment analyses with natural language processing techniques. Moreover, the output with sentiment polarity is visualized on graph.

3 State of the Art

Data mining implementation on top of Twitter requires broad knowledge of different technologies and algorithms that are capable of handling Big Data generated by Twitter users. Therefore, this chapter introduces the most influential tools in the field of data mining, both open source and commercial solutions. Moreover, overview on the related work from academic and industrial perspective is part of this chapter.

State of the art chapter research helps to get broader overview about Big Data technologies, third party platforms, databases, machine learning algorithms, related work from academics and data mining solutions from industry.

3.1 Technologies

Technologies chapter includes research on the Big Data technologies available in the industry, which can be implemented for Twitter processing solutions.

3.1.1 Apache Hadoop ecosystem

Apache Hadoop is an open source framework developed for parallel, distributed Big Data processing across many clusters. Its design architecture enables to scale effectively from couple of servers to hundreds by adding new commodity hardware. Hence, the processing power can be divided into several nodes in cluster and the processing time is reduced for computing given dataset. Platform offers distributed storage, but also has computational capabilities that are enabled by other technologies within Apache ecosystem. Main technologies that are the core of Apache Hadoop are Hadoop Distributed File System (HDFS), Hadoop YARN and Hadoop MapReduce. [12]

HDFS

HDFS is a Java based distributed file system, designed with capabilities to detect and handle errors during data processing and prevent hardware failures, which may relate also to electricity blackout. Big Data that comes from social media, such as Twitter usually of terabytes volumes or more, have to scale to architecture that consists of nodes and clusters. Hence, traditional storage systems lack the features of distributed ones, such as HDFS.

It provides reliable data storage and data replication that is needed in order to avoid data loss. Moreover, HDFS enables to store datasets in any format, whether structured, semi-structured or unstructured data. Furthermore, data integrity is secured by checksum algorithm that is stored within HDFS namespace. In addition, data are stored in distributed file system as block files of fixed size. Usually, it is 64 MB and each block is stored on different data node within cluster. Finally, in HDFS architecture, Name Node keeps track of location where all data blocks are stored on clusters. [13]

Apache MapReduce

Apache MapReduce is a framework, developed to achieve parallel batch processing of Big Data over clusters. Algorithms are usually written in Java, but the development within Apache Hadoop ecosystem enables to use also other tools, such as Apache Hive or Apache Pig to achieve the same goal. Advantage of MapReduce is the computation on dataset can go closer to data, because the initial input dataset is divided into blocks as discussed before.

Development with MapReduce requires implementing two functions. Map function read the input dataset and processes it into key-value pairs. Next phase continue with sorting pairs by key and creating output file for Reduce function. Architecture of MapReduce is based on single master node named Job Tracker, which schedule and monitor jobs, MapReduce tasks. Moreover, it has also role of keep track of failed tasks and can re-execute them. Furthermore, architecture consists also of Task Tracker that executes all master tasks. [14]

Output of one MapReduce task can be input for other task and the final results are stored on distributed file system.

Apache YARN

Apache YARN is a cluster resource management system that extends MapReduce capabilities by performance and linear-scale storage. Aim of YARN is to spit up two major functionalities of Job Tracker and Task Tracker into separate entities in order to support other data processing methods apart batch processing

that is used in Apache MapReduce. This method includes SQL and real-time data streaming that can be used for Twitter data mining. [15] [16]

Apache Hive

Apache Hive is a data warehouse infrastructure developed to enable data processing into structured tables, which can be queried from distributed file system the same way as from relational databases. Language to achieve this task is called HiveQL. Purpose of using Hive is to simplify MapReduce development. [17]

Apache Pig

Apache Pig is a procedural programming language developed for processing Big Data. It was build on top of Apache Hadoop, thus it can do the job similarly as Apache Hive. However, while Hive uses queries to get data from file system, with Pig there must be specified series of steps in order to perform actions on dataset. It has form of scripting language called Pig Latin, which enables to create MapReduce jobs in simplified manner. [19]

Apache HBase

Apache HBase is a database management system, designed as open source project similar to Google's BigTable, so it relies on the clone of the Google File System called HDFS that is also part of Apache Hadoop project. HBase has structure of NoSQL database, which can handle unstructured data that are stored across several servers in cluster. Since it is a distributed system, MapReduce jobs, which request data from HBase can result to latency between reads and writes and thus reduced processing performance of MapReduce algorithms. [19]

Apache Sqoop

Apache Sqoop is a tool developed for the purpose of transferring structured data from relational databases into distributed file system inside Apache Hadoop. It is compatible with relational database systems such as Oracle or MySQL. [19]

Apache Flume

Apache Flume, similar as Apache Sqoop is designed for scalable data gathering from different sources. It then aggregates data into bigger blocks that can be effectively stored on HDFS. Only difference is that Apache Flume works better with unstructured or semi-structured data, while Apache Sqoop with only structured data. [19]

Apache Oozie

Apache Oozie is a tool within Apache Hadoop ecosystem that works as job scheduler for planning and managing MapReduce jobs. [19]

Apache Ambari

Apache Ambari is software for provisioning; monitoring and managing Apache Hadoop clusters through web user interface. Cluster provisioning enable to install and configure Hadoop services within any cluster. Ambari can be well integrated with different enterprise solutions based on Apache Hadoop through RESTful API. [18]

Apache Mahout

Apache Mahout is an open source framework created for running machine learning applications on large datasets. It provides a library with various algorithms, such as clustering, object classification into groups or item recommendations based on user's preferences and ratings. Moreover, Mahout can implement collaborative filtering algorithms. Machine learning systems are used in order to learn from the datasets and make implicit decisions. [19]

3.1.2 Apache Spark

Apache Spark is a framework for distributed and highly scalable in-memory computing, which supports development of applications in several programming languages, such as Java, Python, Scala or R. Purpose is to run these applications in parallel mode among several clusters. It is compatible with other Apache projects, such as Apache Mahout, which uses its processing engine instead MapReduce. Hence, it provides more computational performance and time efficiency during Big Data processing. [20]

Apache Spark supports four sub-applications that are modular types, SQL, MLlib, GraphX and Streaming. They are interoperable and each of these modules can be extended by projects open-source from Apache Hadoop ecosystem. Therefore, there is close connection between Spark and Hadoop, however the architecture is little different. Initial idea of Apache Spark was created by research project at the University of California - Berkeley in order to achieve distributed in-memory data processing with parallelism capability and implements it for machine learning analytics. [20] Machine learning with Spark can be achieved by MLlib module, which offers various choices for algorithm development. It includes functionality for statistics, regression, classification, clustering, dimensionality reduction, or collaborative filtering. [21]

Survey from the Big Data company Syncsort [22] researched that Big Data processing trend in 2016 goes towards Spark with 70% rather than its main competitor MapReduce at 55%. Moreover, survey discusses that Apache Spark is the most active project within Big Data industry nowadays. MapReduce, which is the default processing engine for Apache Hadoop, should be replaced by its successor Apache Spark. Furthermore, companies such as Cloudera, Hortonworks and MapR that are top three vendors when it comes to Big Data support Spark as the default processing framework over Hadoop. [22]

Comparison between Spark and Hadoop processing capabilities clearly shows that Spark, according to [23] can run programs up to 100 times faster than Apache Hadoop MapReduce in memory settings, or 10 times faster on local storage.

3.1.3 Apache Storm

Apache Storm belongs to the Apache projects dedicated to process Big Data. It is an open source distributed system for real-time computations, similar to Apache Spark. [24] On the other hand, comparison between Storm and Hadoop can be based on the processing capabilities, because Hadoop can be better implemented in projects that need batch processing and Storm real-time processing. Storm is well integrated with many use cases within machine learning or real-time analytics, because it is scalable, easy to use and fault-tolerant.

According to official project overview [24], Storm integrates processing power in well-known technological services, such as Yelp, Spotify or Twitter. Apache Storm

joined Twitter in July of 2011, which helped with Storm development in its initial stages to make it interesting between software engineers by setting up Github project to promote it. In September of 2013 was Storm proposed for incubation in Apache. [25]

3.1.4 Heron

In 2015, Twitter engineers developed new real-time processing system, should replace their Apache Storm. Heron has several advantages compared to its predecessor in terms of more efficient data resource management on the data infrastructure at Twitter. According to Twitter engineer, Karthik Ramasamy, Heron implementation has enabled to reduce hardware from the data infrastructure and so make it more efficient. [26]

Apache Storm replacement for Heron provided Twitter up to 14 times more throughput and up to 10 times less latency on a word count topology. [30] Twitter requirements for data processing system are based upon real-time analytics of billions of events per minute. Moreover, other requirements go towards scalability and high data accuracy optimized in case of data processing failures. [27]

Topologies at Twitter are deployed on Apache Aurora, which is a framework for long-running services and cron jobs. Hence, services can run in parallel on shared machines and in case of failures, Aurora can implicitly reschedule those jobs on new machines. [28] Furthermore, Heron is fully compatible with Storm API. [29]

3.1.5 Apache Flink

Apache Flink is an open source platform for distributed stream and batch processing over datasets. Core part is made of streaming engine, written in Java and Scala [31], which provides distribution of data, communication, and fault tolerance within distributed data streams. [32] Similarly as Apache Storm, Flink supports additional libraries to target specific scenarios within data processing and data mining. FlinkML is a machine-learning library for Flink, that provides API for building scalable machine learning algorithms. [33]

Moreover, in order to visualize data, it is connected with Gelly, the Graph API for Flink. [34]

Despite the platform does not include any data storage system, Flink can be integrated with other open-source projects and store input data on distributed systems, such as HDFS or Hbase. Streamed data feeds can flow into Flink through high-throughput and low-latency platform Apache Kafka. [35]

Flink and Spark can be compared to each other, since both can run on top of Hadoop, both support extra libraries for machine learning, graph visualisations and they can perform in memory computations. However, the main difference took place in their processing capabilities.

According to [36], Spark is not pure streaming engine, because it performs fast-batch processing over small chunks of data, known as micro batching and Resilient Distributed Dataset (RDD). While with Flink, data can be accessed in real-time without pausing the stream and divide it into smaller pieces. This means that Spark streaming can process data only in near-real time, Flink can process whole stream, row after row in real-time. [37]

Required low-latency responses that Apache Spark lacks can be added with Apache Storm through API. Comparison showed that Apache Flink could overcome this drawback, as it supports running the same algorithm for batch processing and streaming. [36]

3.1.6 Apache technologies overview

Overview on Apache state of the art technologies, which are used nowadays within Big Data industry, depicts Figure 4. Heron technology does not belong under Apache; however, it is technology developed by Twitter with fully compatible API to Apache Storm.

Apache - State of the Art Technologies	
HDFS	Hadoop distributed file system with fault-tolerance capabilities
MapReduce	Parallel batch processing framework
YARN	Cluster resource management system
Hive	Data querying data warehouse
Pig	Procedural language that simplifies MapReduce development
HBase	NoSQL database system
Sqoop	Structured data transfer from relational DB to NoSQL
Flume	Unstructured data transfer
Oozie	Job scheduler for MapReduce
Ambari	Cluster management system
Mahout	Machine learning system

Hadoop	Distributed and scalable cluster processing framework
Spark	In-memory, scalable, distributed and fast-batch cluster processing, faster than MapReduce, near-real time streaming
Storm	Fault-tolerant, scalable, distributed and real-time processing over Big Data
Heron*	Higher throughput than Storm
Flink	Similar to Storm with extended libraries for scalable data mining; real-time streaming

Figure 4: Apache state of the art technologies for Big Data

3.2 Third Party Big Data Platforms

Technologies for Big Data processing can be divided into open-source projects under Apache license, as discussed before, but also into solutions developed for enterprises that are built on top of Apache or have capabilities to support Big Data analytics.

3.2.1 Cloudera, Hortonworks, MapR

Apache Hadoop with its broad capabilities and wide supporting projects from Apache ecosystem has created new opportunities for innovations within data processing. Increased demand for technologies like Hadoop contributed to Big Data market and thus several vendors introduced enterprise solutions to tackle Big Data challenges.

Cloudera has the biggest market share that offers platform for data management and analytics based on Apache Hadoop. It provides in cloud solutions for exploring enterprise data and enables new customers with live demo, which is dedicated to explore the features of the product before its adoption within enterprises data infrastructure. [38] Moreover, Cloudera provides solution that is easily extended with other Apache technologies accessible through web-based GUI, called HUE. [38]

Intel is also key market player since they launched their platform for data processing, Trusted Analytics Platform, which includes Hadoop and Spark in its data layer. Moreover, Intel also supports Cloudera development with investments, because company believes it will drive the Big Data market and thus will increase demand for Intel's processors. [39]

Another great innovators in the industry and also the main competitors on the market with Cloudera are companies Hortonworks and MapR.

3.2.2 Splunk and Hunk

Splunk is a technological platform that provides engine for real-time data analytics. It was developed in order to process Big Data streamed data generated by various data sources, such as social networks, sensors or enterprise data infrastructures. Splunk can turn machine-generated data into valuable insights, called Operational Intelligence. [40]

Based on the real-time data processing, Splunk can improve understanding data more effectively and so optimize business processes in proper manner and based on data-driven decisions.

Hunk is a platform that enables to explore, analyse and visualize Big Data in Apache Hadoop and NoSQL databases. Hunk provides enhanced simplicity and speed capabilities Big Data processing of unstructured data. Therefore, Hunk is platform that can find anomalies within terabytes or petabytes of raw data. It is fast to deploy in the data infrastructure, support interactive search patterns that are easily identified by keywords, reporting and drag-and-drop features. [41]

3.3 Big Data Databases

Big data addresses many challenges in regards to data storage. Every project implementation needs to have specified requirements for suitable database that will enable to effectively retrieve data. These requirements should analyse the purpose of the project, data structure in the system architecture, what role-plays Big Data in concept solution and how they are accessed.

3.3.1 NoSQL

NoSQL databases refer to Not Only SQL, which are efficient when solution needs more than one storage mechanism. [42] They are not based on relational schema, defined joins, referential integrity such as SQL, however these NoSQL supports SQL-like query capabilities. [43] NoSQL are competent within Big Data volume issues as they can run effectively on parallel clusters.

Advantages rise towards data structures of the solutions, because developers can overcome limitations between relational and in-memory application data structures without transforming in-memory structures into relational ones.

Moreover, with the capabilities of web as a service, databases can be encapsulated within applications that communicate through web service interface. Furthermore, NoSQL is viable to implement when there is demand for improved performance over large data volumes processing, latency reduction or throughput optimization.

NoSQL can be divided into four categories, which differs by the perspective of its usage. These categories are: column-family stores, document stores, key-value stores and graph databases. [42] Figure 5 depicts the comparison between popular NoSQL databases based on their characteristics.

Storage Type	Usage	Implementations
Column store	Read/write-intensive application	HBase, Cassandra
Document store	Consistent/occasionally changing data	MongoDB, CouchDB
Key-value store	Application with often changing data where high data availability is needed	Riak, Redis
Graph databases	Spatial data for Graphical Information Systems (GIS)	Neo4j

Figure 5: NoSQL comparison [43]

Big Data databases in comparison with traditional data warehouses must operate over large unstructured data and clusters. Choosing the right technology for project solution can positively influence its performance and enhance capabilities. State of the Art Big Data technologies can address decision making analytical techniques based on machine learning or predictive analysis. Moreover, NoSQL databases can scale proportionally the data architecture and process large datasets on cluster nodes. [44]

3.3.2 Data Warehouse

On the contrary, traditional databases (data warehouses) lack these capabilities of storing unstructured data volumes. Rather data warehouses are viable for operating per day transactions, OLTP (Online Transaction Processing). Moreover, they are viable within data analytics techniques, OLAP (Online Analytical Processing), but limited to architecture scalability. [44] Figure 6 shows the comparison between Big Data oriented databases and data warehouse.

Data Warehouse vs. Big Data			
Data Warehouse	Transaction oriented	Database handling	Small to medium infrastructure
	Query languages OLTP, OLAP, Decision support tools	Structured relational data File system on single node or cluster of nodes	Transaction systems Metadata distributed over storage nodes
Big Data	Decision support	Large scale data handling	Large scale infrastructure
	Machine learning Natural Language Processing, Predictive Analyses	Rapid velocity and data volumes Un/semi structured data Scalable data	Massively distributed system Scalable architecture Commodity hardware

Figure 6: Big Data and Data Warehouse comparison [44]

3.4 Development Technologies

Research on the development technologies used for Twitter sentiment analyses implementation is discussed within this section.

3.4.1 Python

Python is a programming language developed under open source license. It is an interpreted language applied within various development tasks, such as web applications, data analyses and data visualizations. Many contributors support it and so it offers extensive library capabilities to integrate other programming languages code.

Python is excellent in developing scientific applications and so it will perfectly fit as tool for this project. Some essential Python libraries are discussed, as they are viable for solution development. NumPy is abbreviation for Numerical Python, which is Python library for scientific computations and processing. Pandas is open source data analysis tool built on top of NumPy with high performance features and rich data structures capabilities. Matplotlib is popular Python library for creating 2D data visualisations. It gives developers to easily integrate plots, histograms, and scatterplots with data solutions. [79]

IPython offers interactive computing within shell environment for Python oriented solutions. Its extensive features also provide interactive data visualizations and can serve for running, testing and debugging Python code. [80] Moreover, IPython offers integration with Jupyter notebook, which is an open source interactive web application for scientific programming. Besides Python, Jupyter supports more than 40 programming languages in particular with data analyses for example: Scala or R. Moreover, it supports data cleaning and transformations, statistical modelling or machine learning. Furthermore, Jupyter notebooks can be shared and opened in the web browser. [81] IPython serves as Python kernel for Jupyter.

Moreover, Tweepy is Python open source library build in order to enable access from Python code to Twitter's API. [82]

Python libraries	
NumPy	Scientific computing
Pandas	High performance data manipulation
Matplotlib	Data visualisation
IPython	Interactive computing for Python
Juypter	Manipulate and share Python code
Tweepy	Provides access to Twitter API

Figure 7: Summary of Python libraries for data science

3.4.2 Natural language processing

Twitter Data mining approach to sentiment analyses can be implemented with Natural Language Processing, (NLP). It is part of computer science, which deals with text analysis, spell corrections, spam classifiers or machine translations. Natural language can be perceived as language used by humans for communication, such as English. Thus, implemented solution with NLP must recognize and understand at least basic language structures for creating meaningful insights with sentiment analyses. [84]

Creating solution with NLP requires understanding of processing tools. Natural Language Tool Kit (NLTK) is open source library developed in Python. This technology contains necessary tools for the purpose of text processing and sentiment analyses on Twitter.

Text processing belongs to pre-processing data analysis phase in methodology, because text as attribute in tweet has to be processed into suitable

form before actual data mining. On the other hand, sentiment analyses, as data mining technique will follow in development part of the project.

NLTK as a tool for natural language processing provides various features applicable for sentiment analyses. Discussion on the methods will follow in this section. Tokenization is a method for splitting up text into sentences and words. Text as a string is thus divided into a list of tokens. In addition, token represent word within sentence and also sentence among paragraph. [85]

Part of NLTK is dictionary, such as WordNet that includes collection of words and their meanings, thus it provides options to search for synonyms, word relations and semantics or find disambiguated words with same meaning. [85] Moreover, it can be used for semantic similarity measures between words.

Regular expressions allow tokenizing text with advanced approach for dividing text by space or period.

Another challenge with NLTK implementation is to reduce words without any contribution to sentiment analyses. These are called stop words and generally include words, such as: the, he, was, a, etc.

Stemming is a method of data normalization, where the suffix of the word is removed from its base root structure, called stem. For example word *play* has many forms from the linguistic view: *playing*, *played*, *plays*, etc. Texts analyses do not require differentiate between various tenses of the word.

Lemmatization is similar technique to stemming for text normalization. Main difference is that while stemming may convert root of word to non-existing word, lemmatization bases on part of speech rules and convert root word, called lemma into actual existing word searched in the NLTK dictionary.

Part of Speech, (POS) is the method of identifying part of speech for words within sentence. It means tagging words, whether they are nouns, adverbs, verbs, and so on. [85] With NLTK tagged words are presented as tuple, where each tag has dedicated description as showed on Figure 8.

Part of Speech tags	
NNP	Proper noun, singular
RB	Adverb
VB	Verb, base form
VBD	Verb, past tense

Figure 8: Part of Speech tag example [85]

Name Entity Recognition (NER) is a method, which is capable to implicitly recognize entities from analyses text. It is natural language processing approach to chunking, which is method of grouping words into chunks based on the entity type. Usual entity tags are: Person, Location and Organization as depicted on Figure 9.

Organization	Person	Location
Apple CEO	Tim Cook	Cupertino, California, U.S.

Figure 9: Named Entity Recognition tag example [85]

Summary of NLTK tools is showed on Figure 10.

NLTK	
Tokenization	Split text into list of tokens
Stop words	Remove redundant words
Stemming	Text normalization
Lemmatization	Text normalization, more complex
Part of Speech	Tag words based on POS
Named Entity Recognition	Implicit entity recognition

Figure 10: NLTK summary

3.5 Algorithms

Volumes of data generated from social media networks, referred also as Big Data, increased dramatically over the past years. Data engineers have responsibility to find information from this data that is useful for further analysis. Goal is to seek for patterns between data with state of the art algorithms and technologies.

Data mining are can be also perceived as “knowledge mining” and usually are known under synonym knowledge discovery from data, KDD. “Data mining is process of discovering interesting patterns and knowledge from large amounts of data”. [45] Process starts with data cleaning, which is necessary to remove redundant data from the dataset. The last step after pattern discovery, visualization took place. Data mining is closely related to machine learning.

Machine learning turns data into information based on statistics. It is a science field in the intersection of engineering, computer science and statistics. According to [46], *“Things can learn when they change their behaviour in a way that makes them perform better in the future”.*

Deterministic solutions are the ones, which always solve the given problem formulation. Figure 11 depicts the connections between data mining domains.

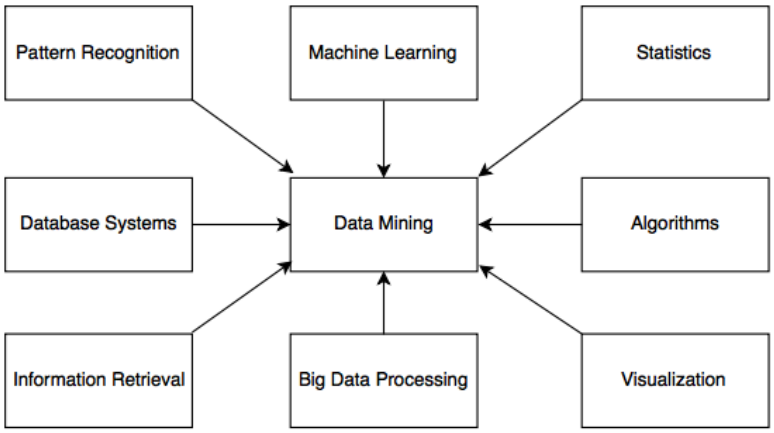


Figure 11: Data mining application domains [45]

However, there are solutions that are not deterministic, when there is not much knowledge about given problems or there is a lack for computing capabilities. These issues can be overcome by using statistics. [47] Hence, data mining as a way of finding data patterns are influenced by other techniques from the data-driven application domain, such as above mentioned machine learning, statistics, visualisations, databases and data warehouses. [45]

Machine learning deals with assumptions that computers can implicitly recognize patterns within data and then make data-driven decisions. As mentioned above, machine learning is a data mining application domain and therefore, discussion about its techniques will took place in this sub-chapter. Machine learning algorithms can be divided as follows: supervised learning and unsupervised learning. Latent semantic analyses find similarities between concepts.

3.5.1 Supervised machine learning

Supervised learning consists of predicting dependant variable from the independent. Thus, supervised can be interpreted as classification, where the goal is to predict what class an instance of data belongs to. [47] For example, prediction of future stock prices on the stock market based on historical data.

According to [48], researchers investigated whether it is possible to correlate public mood from Twitter and changes on the stock market. Results have showed that changes of public mood settings can be tracked from Twitter large-scaled data and applied on the economical factors, such as stock market.

Supervised machine learning looks for patters in labelled datasets are relevant at various scenarios in which after successful patterns discovery, algorithm can use them for future predictions. [49]

Classification predicts discreet valued output, whether 0 or 1, called label, such as e-mail is spam or not. Regression on the other hand is used for prediction of numeric value, such as stock prices example. [50] Classification can be further divided into: linear models, decision trees and naïve Bayes models. Apache Spark supports all these models in its MLlib library for machine learning. [51]

Logistic regression is probabilistic type of classification models, implemented by logistic functions. It can be divided into binary type with only two possible outcomes for dependant variable to occur. On the other hand, multinomial logistic regression can lead into more outcomes. [52]

3.5.2 Unsupervised machine learning

Unsupervised learning is on the other hand synonym for clustering. In unsupervised learning is not dependant variable (target value) for prediction. Moreover, these algorithms deal with unlabelled datasets. [47]

Purpose of the clustering technique is to analyse collection of data points (objects) and grouping them into clusters based on the distance measures between each data points. Moreover, the algorithm should group all data points into the same cluster and achieve that different clusters will have far distance from each other.

Datasets for clustering purposes consists of collection of points as objects that are belonging to some space. In Euclidian space, points are considered as vectors of real numbers, where Euclidian distance is measured. On the other hand, clustering also involves non-Euclidian measures, like Hamming, cosine or Jaccard distance. [3]

K-Means is unsupervised point-assignment clustering techniques, which uses Euclidian space and defines number of clusters (k) with centroids for each cluster. Centroids are central points within cluster to which data points are assigned considering the closest distance between centroid and data point. Moreover, clustering can be processed in parallel by using MapReduce based clustering. [54]

Clustering can be divided into two categories: single machine clustering techniques and multiple machine clustering techniques. [53] With the Big Data

complexity it is challenging to implement clustering data mining algorithms under effective computational time in order to get relevant results.

Single machine clustering can be divided into data mining algorithms clustering or dimension reduction, which are categories for further division. Data mining algorithms includes partitioning methods, hierarchical methods, density-based, grid-based and model-based methods. [55]

Partitioning based clustering classifies data points into clusters based on their similarities. As mentioned before, K-Means belongs to this category, as it is non-deterministic technique because of predefined K parameters, which is main drawback.

Hierarchical based clustering divides data into different levels of the hierarchy. Purpose of this method is to group objects into classes increasingly wide, based on the similarity or distance measures. [55] Advantage is that data are clearly visualised as hierarchical tree. However, main disadvantage of this method is fact, that when a stage in finished, it cannot be changed. Figure 12 shows overview of different Big Data clustering approaches.

Big Data Clustering Overview		
Technique	Advantages	Limitations
Data mining algorithms	Easy to implement	Massive data volumes
Dimension reduction	Reduce dataset size, efficient and scalable	Implemented before classification
Parallel clustering	Time efficiency, scalable	Implementation
MapReduce clustering	Scalability, parallel	Requires more processing power

Figure 12: Big Data clustering overview [55]

Overview of the clustering techniques is shown on Figure 13.

With density-based algorithms, clusters are defined as dense regions, separated by low-density areas. Hence, they are not recommended for using with big datasets. [55]

Dimension reduction data mining algorithms deal with problems related to existence of several dimensions in which data size can be measured, the number of variables and the number of examples. [55]

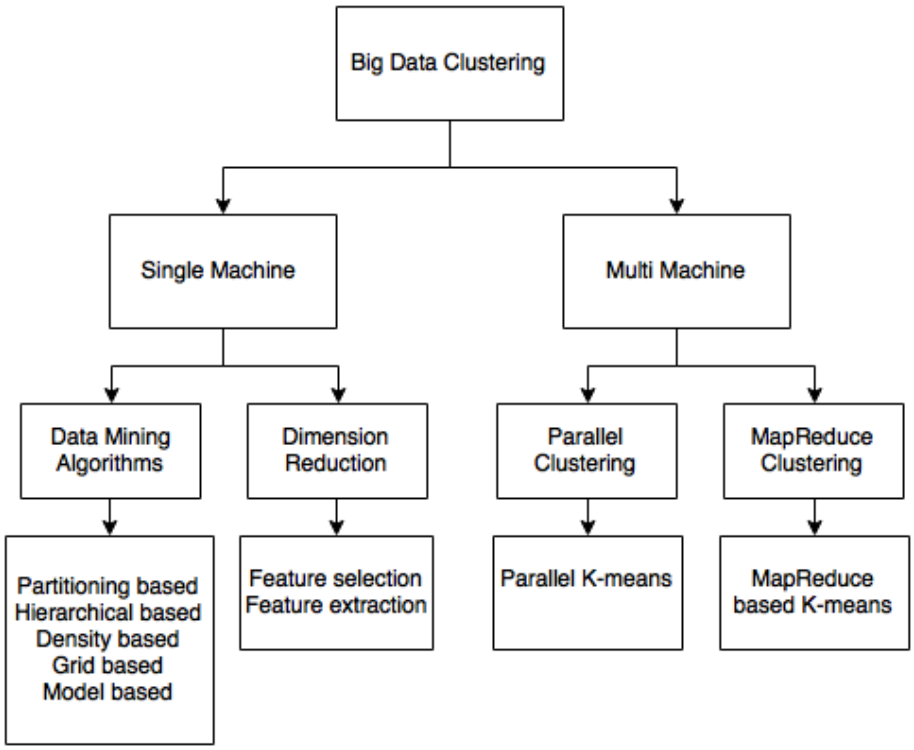


Figure 13: Big Data clustering techniques [55]

Data pre-processing is required to reduce dimensionality before applying clustering algorithms. Aim is to achieve data relevancy by eliminating redundancy from the given dataset.

Multi machine clustering categorise parallel clustering and MapReduce based clustering. Parallel clustering tries to achieve distributed computing over several machines with high time efficiency. It focuses on the parallel algorithms that divide data into smaller chunks with the focus on scalability and enhanced performance computations. [55]

Moreover, MapReduce besides clustering (K-Means) supports also other algorithms, such as classification (Naïve Bayes) or dimensionality reduction (Stochastic value decomposition). [56]

3.5.3 Latent semantic analyses

Latent semantic analysis (LSA) is a method of analysing text documents to find similarities between words that are represented as concepts. Moreover, LSA has close relationship with neural models, but it is based on matrix decomposition by using Singular Value Decomposition (SVD) technique. LSA is a vector-based implicit

technique for unveiling relations of contextual usage of words from text. However, it is not part of artificial intelligence, because it is not based on explicitly defined dictionaries of words or knowledge data pools. In fact, input data are presented as raw text that is parsed into words defined with its uniqueness and distinguished into meaningful sentences. [57]

LSA, starts with creating matrix from the given text document that will contain rows and columns. Each row will represent unique word, while each column will represent some text passage from the document. Moreover, each matrix cell contains number of times of each unique word appeared in the text passages. Furthermore, each cell frequency from matrix is transformed by function, which is developed to reveal the importance of the given unique word from the text.

Core of LSA presents SVD, which is about to be applied on the matrix. With singular value decomposition, the rectangular matrix is decomposed into three different matrices. Purpose of it, is to reduce dimension of matrix in order to find the close patterns between words, while reducing the noise. Matrix must not be centred as it would increase computational resources and turn sparse matrix into dense matrix. [58] [59]

According to [57] any matrix can be decomposed by using no more factors than contains the smallest dimension of the matrix. Removing coefficients in the diagonal matrix can reduce dimensionality.

3.6 Related Work

Related work chapter is a part of State of the Art, in which research on various data mining approaches towards Twitter will be analysed from two perspectives. Academic is dealing with papers from the academia environment, while industry research will analyse developed solutions that are available on the market.

3.6.1 Academy

Research [60] is presenting case study on machine learning integration with Twitter, based on Apache Hadoop and Apache Pig processing framework, which incorporates predictive analytics. It is used in the research paper for mainly supervised classification. Furthermore, with Apache Pig was integrated in order to deploy machine learning pre-processing steps, such as data sampling, feature

generation, training and testing. Their solution is presented by developed Pig scripts, which allowed large-scale integration of machine learning with their existing Hadoop data infrastructure. [60]

Moreover, implemented Hadoop Distributed File System (HDFS) supports batch and real-time processes that can run as application jobs. Furthermore, in order to specify data schema, serialization framework took place to avoid processing different data formats. Serialization compiler can implicitly generate code for manipulating and data managing for Protocol Buffer and Thrift messages.

Apache Thrift is framework for scalable service development that enables to work with software stack and code generation engine. Purpose is to add effectiveness into working environment with various programming languages like Java, Python, C++ and more. [61]

Twitter and its workload analytics are divided into two groups: aggregation and ad hoc queries. First mentioned are queries for analytics and relevant to business intelligence jobs as scanning through massive datasets.

On the other side, ad hoc queries, according to [60] are submitted explicitly by users, however these queries have not any computational pattern. As job scheduler is implemented workflow manager, Oink that aims to take care of dependencies between submitted application jobs.

Twitter in 2008 has acquired start-up company that was developing machine-learning solutions, concretely sentiment analysis. Moreover, this acquisition enabled Twitter detect spam messages in the platform. [60]

Concept for Twitter data mining must include scalable architecture to handle large datasets. Furthermore, it is important to consider processing capabilities, which are available.

Thus, researchers at [60] have used Hadoop cluster with single node, HDFS, Pig and Java library with included classifiers for regression, decision trees and others. Sentiment analysis is part of machine learning, which can be applied with Twitter to analyse customer's opinions about products, brands and their behaviour on the market. One approach is to predict positive or negative sentiment, polarity classification based on the Twitter's emoticons. [60]

Another approach to Twitter sentiment analysis is researched at [62]. Researchers deal with challenges towards effective identifying of Twitter data based on hashtag (#). However, for the data training they used also emoticon data set and also iSieve data with sentiment (positive, negative and neutral) for evaluation only. [62] Pre-processing stage was divided into: tokenisation, normalization and part-of-speech (POS) tagging. It is part of linguistic pipeline and basic form for syntactic analysis. Tweet sentence can be build from noun, verb, adjective and other parts of the speech. [63] Tokenization is a process of identifying of abbreviations and emoticons, such as BRB, stop words or other micro-blogging features. [62] On the other hand normalization is process of transforming of abbreviations into their actual meaning, such "be right back" (BRB). Moreover, capital letters within tweets were modified into small letters. Results [62] showed that part of the speech features that count number of appearances of nouns, verbs, etc. are not viable for Twitter sentiment analysis. However, hashtag and emoticons approaches for sentiment analysis has proved their relevancy.

Other approach to Twitter sentiment analysis is researched in [64] study. Classification of tweets is build upon model with two tasks. Binary classifies tweets according to sentiment into positive and negative groups only, while three-way task consider also neutral sentiment. Pre-processing stage was in [64] proposed with emoticon dictionary used in order to label each emoticon with sentiment statement and acronym dictionary, which translated acronyms, such as "gr8t" into "great".
Moreover, URLs include in tweets were replaced with llUll, targets such as (@John) with llTll, all negations (e.g. no, not, cannot) with "NOT" and repetitive word characters (e.g. coooool) were shortened. Research has showed that feature analysis gives the best accuracy results by combining prior polarity of words and POS tags. [64]

Proposal on implicit pattern extractions from semantically similar words in tweets is discussed in research [65]. Semantic patterns are defined as words group into clusters with similar sentiment.

SentiCircle model [66] presents the process of extracting semantically patterns from tweets, Figure 14. Collected tweets are syntactically pre-processed in order to filter out the noise. Later, sentiment lexicon is applied to capture contextual

semantics among tweet words. Then, similar words within tweets are clustered in order to form semantic patterns. [65]

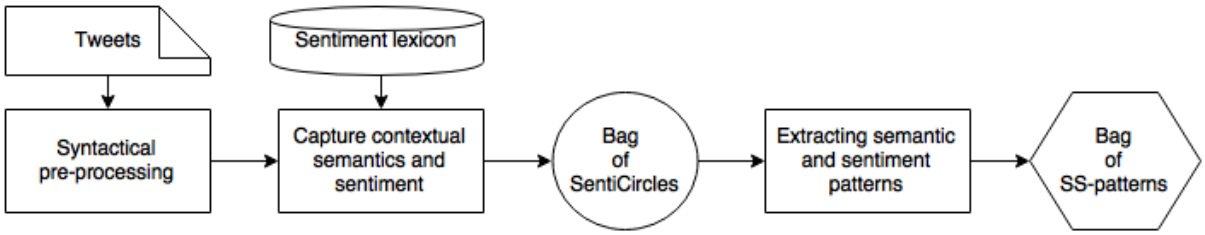


Figure 14: Twitter pattern extraction [65]

Research [65] resulted into findings of entity sentiment classification that positive sentiment was easier to detect from the tweets, rather than negative or neutral. Reason to this can be influenced by chosen classifier or by number of tweets with positive sentiment in dataset.

Social media and social networking were also input into study, which is dealing with future predictions. Research at [67] study scenario if rates of tweets created by Twitter users about specific topics can surpass market based revenue predictions about latest movies. Sentiment analysis was used to discover how would people behave on different movies. Reviewers can be powerful influencers about ones meaning about new movie releases, thus such sentiment shared between "followers" can dramatically reveal on the revenue figures. Dataset contained tweets requested by Twitter Search API and searched by movie title. Key attributes of each tweet were author, timestamp and text. Moreover, researchers analysed period one week before the new movie was released as during which all promotional and marketing related actions are set, such as photos, trailers and promos.

Solution was developed with linear regression model that aimed to predict revenues of movies before their release dates. Furthermore, it showed the correlation between popularity in Twitter topics related to certain movie and its success in the future. [67]

Stock market prediction analysis based on Twitter is researched in paper [68]. Aim is to predict statement, if collective mood extracted from tweets is correlated with Dow Jones Industrial Average (DJIA) index over period of time. Behavioural

economics define that human decision-making can be influenced by our emotions or mood settings. [68]

Opinion Finder tools was deployed in solution to analyse tweets by positive or negative values and Google-Profile of Mood States (GPOMS) to analyse user mood from tweets that measure six distinct mood dimensions. [68]

Moreover, Self-Organizing Fuzzy Neural Network was used to make testing whether accuracy of DIJA prediction can be enhanced by mood measured of Twitter users. Results showed that not all mood dimensions could correlate to DIJA, which means changes of public mood within GPOMS dimensions can match changes in DIJA figures that occur later up to four days. [68]

Twitter as social network offers capabilities for predicting various scenarios, such as mentioned before: correlations between Twitter and movie revenues or stock market.

Research at [69] analysis Twitter opportunities with prediction capabilities of 2010 U.S. political elections. Dataset contained thousand of Twitter users that were actively engaged in political discussions during election period. Results has showed that a support vector machine (SVM) based on hashtag user metadata gained 91% accuracy with predicting whether user in tweets expresses right or left political inclination.

SVM is content-based classification technique used for tasks that deal with high dimensional data. [69] LSA were used to reveal hidden patterns within tweet metadata generated by users. Network clustering algorithm helped to identify communications between users in dataset and visualise topology in order to enhance accuracy classification. Moreover, results also revealed fact that re-tweets with political content can lead to predictions of user political inclinations with 95% accuracy.

3.6.2 Industry

Discussions in previous research studies from academic related work to Twitter and data mining have showed different approaches to solution concepts. Social media expansion over the years enabled entrepreneurs build businesses and gave new opportunities for innovative service development. Therefore, this sub-

chapter will discuss some of the solutions that combine data mining with Twitter social media.

Stock Twits is a service developed on top of social media streams. Human thinking can influence stock market as discussed before, which can lead to sentiment differentiations. Service offers real time insights from stock traders and companies involved. It gives customers opportunities to reveal what other people think about current situation on stock market and investing trends.

Moreover, it supports features with sentiment analysis, which can automatically evaluate viability of stocks. Platform created $TICKER tag that can group stock market information from different financial media streams and social networks, Twitter, Facebook and LinkedIn. Furthermore, Stock Twits supports smart devices with applications for Android and iOS. [70] [71]

Crowd Flower is platform, which combines human contributors and machine intelligence to solve data challenges. They can handle with different types of use cases, such as sentiment analysis, improved search results relevancy, issues with brand protection and content moderation, data categorization, and more. Purpose of this business is to help their customers to reveal value propositions from their datasets by implementing Crowd Flower algorithms. [72]

Dataminr is company oriented with real-time information discovery from Twitter and other publically accessed media. It provides solutions within finance, news, corporate security, crisis management and public sector. Their solution is based on identifying relevant information about specific topic online, transforming it into real-time signals and providing it to customers through application as well as API.

According to [73] Dataminr for Finance is used by hedge funds and investment banks across the world to keep updated on market trends from other perspective. Data mining technique for event detection based on Twitter depicts Figure 15. Nowadays, market with social media and data mining also various solutions based on content aggregation.

Right Relevance [74] offers solution for searching through articles, influences and conversations that are aggregated from Twitter. Various filters enable users

search by keywords and sort by relevancy or timescale. Another platform for hashtag searching through social media is called #tagboard. [75]

Keyhole is social media based solution for Twitter analytics and real-time insights exploration. It offers features, such as geodata, heatmaps, influencer data, activity timeline, topic tracking, gender analysis, share of posts, top posts or device recognition. [76]

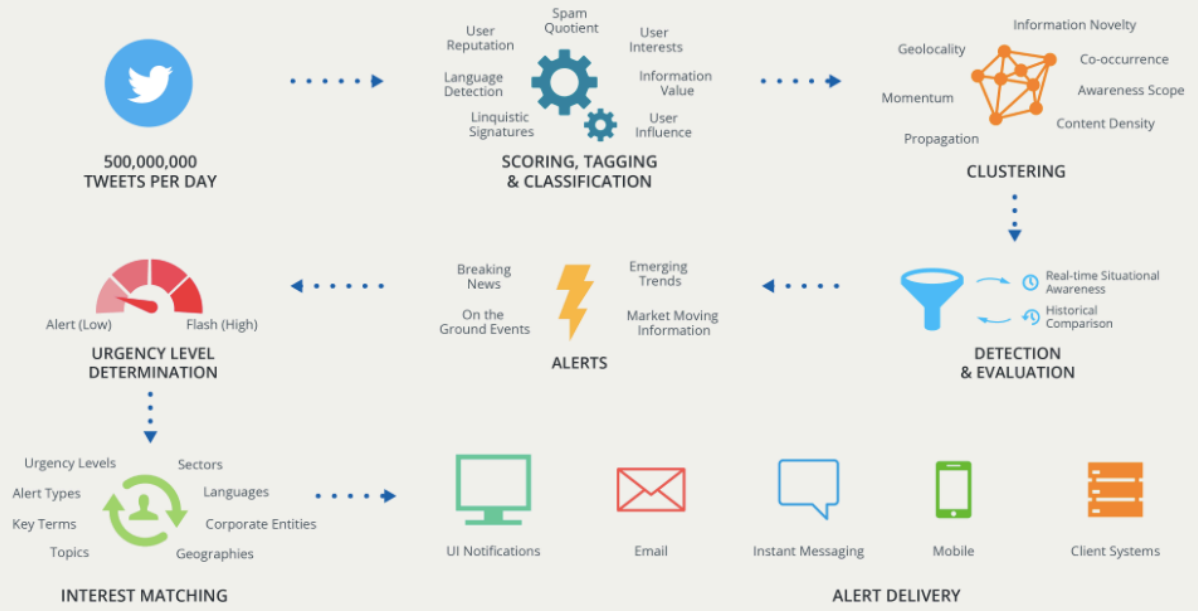


Figure 15: Dataminr event detection technological process [77]

4 Twitter and Apache Spark

Solution for Twitter sentiment analyses can be implemented with Apache Spark. This chapter discusses the Spark streaming and architecture design. As researched in state of the art chapter on page 23, it is system for cluster computing for Big Data.

4.1 Apache Spark Streaming

Advantages of Spark enable to stream Twitter feed with Spark Streaming module that supports fault-tolerant and scalable streaming processing. API provides interaction with Spark with several programming languages, such as Java or Python, however the most proffered is Scala. However, demand for computing resources for Spark development limits this approach in this project and therefore lightweight implementation is preferred with natural language processing.

Discussion on the concept of Spark streaming architecture and cluster computing on Twitter data mining took place within this pages. In the applications that produce big amounts of data, such as Twitter, processing tasks of such data belongs to the main challenges.

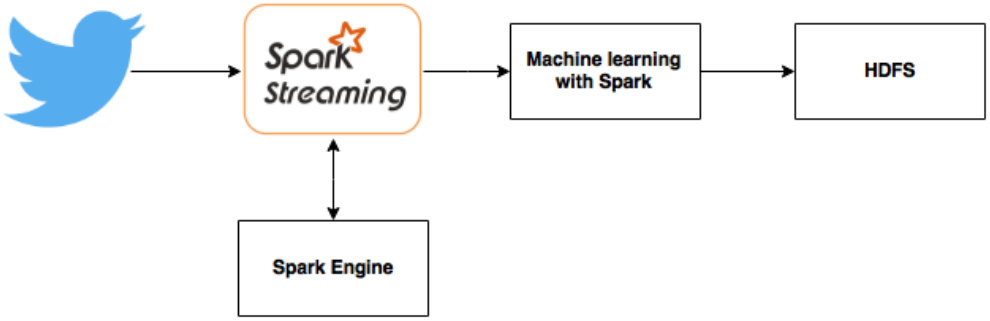


Figure 16: Apache Spark Twitter streaming [21]

Above Figure 16 depicts Twitter streaming process with Apache Spark. HDFS is distributed file system that is used with Apache Hadoop implementations and serves as storage system for clustered processing. From the perspective of Big Data, Spark Streaming takes the input Twitter stream and transforms it into batch files with Spark Engine where are being processed in clusters. Moreover, machine-learning techniques can be implemented with MLlib to process streamed Twitter data.

4.2 Discrete Spark Stream

Continues stream of data that flows into Spark Streaming module is called discrete stream, which is implemented as a sequence of abstraction in Spark called Resilient Distributed Dataset (RDD). RDDs represent fault tolerant collection of objects that are distributed across cluster nodes and can work in parallel computing. Input data source for RDDs can be fetched also from local file stored on HDFS. [21] Discrete stream for Twitter case is presented on Figure 17.

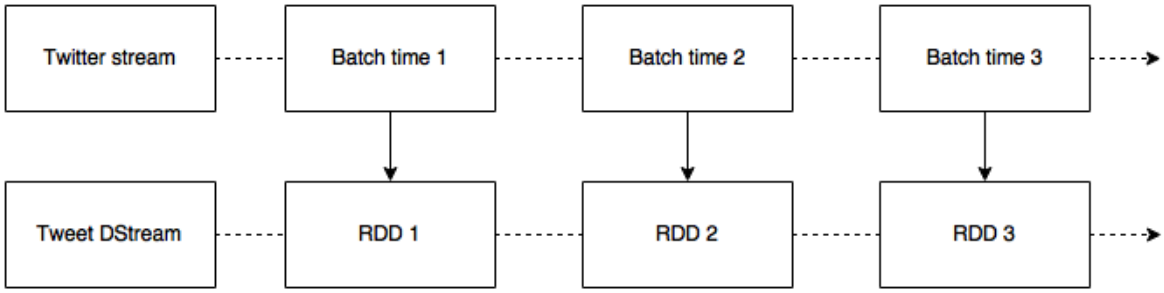


Figure 17: Twitter discrete stream [20]

Data from Twitter stream source are divided into RDD data files, with certain batch time interval, which can be specified by developer according to specific needs. RDDs from DStream are stored in memory and with every new operation; such as function initialization on current RDDs will create new resilient distributed dataset for every batch time.

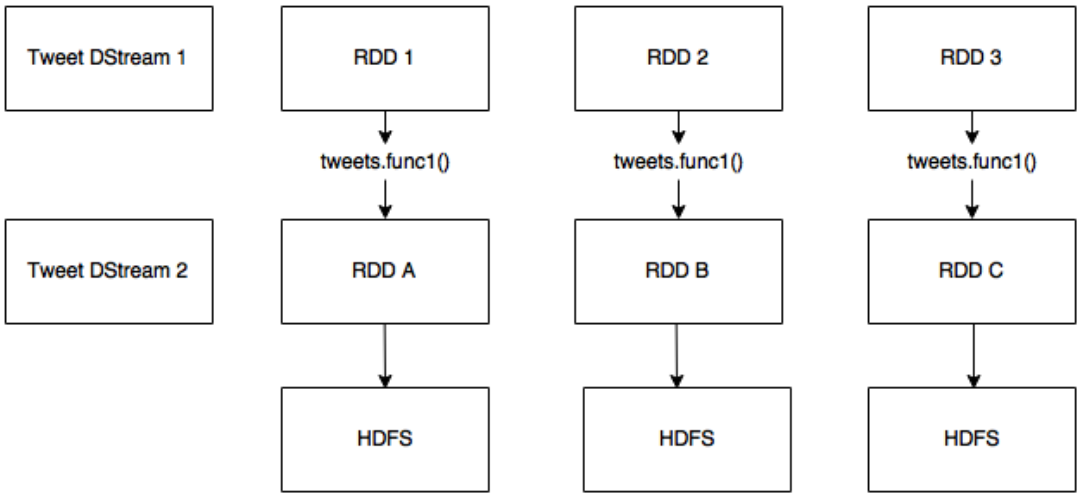


Figure 18: Process of RDDs creation [20]

Above Figure 18, shows how RDDs are transformed. Function *tweets.func1()* is applied on each RDD from Tweet DStream 1 in order to generate new RDDs A, B

and C. These operations are processed by Spark Engine system. At the end every batch file is stored as single file on distributed file system from Hadoop, or other database systems such as HBase. [21]

Advantage of parallel cluster computing with Apache Spark is the time efficiency and computational capabilities for Big Data processing. Moreover, This technology provides options to process data locally in various formats, but also supports streaming in-memory processing for massive data volumes. In combination with machine learning and visualization techniques, Apache Spark provides powerful tools for Big Data analyses on Twitter.

Other input sources from the state of the art technologies that can implement solutions for streaming services are Kafka or Flume. Fault tolerance capabilities enables recreate computations due to Spark failure on working nodes. This feature is available because batched data are replicated in-memory during processing. [93]

Spark Streaming functionality is defined by object *StreamingContext*, which needs to be created to initiate streaming process. It has provided the Spark cluster URL as spark context *sc* and a batch time interval set to 5 seconds. Tweet stream is then created from *StreamingContext* and attached to object *tweetStream*, which is Spark DStream of RDDs. Figure 19 below shows the process of creating Twitter stream and initializing *map()* operation that maps status objects in order to create DStream.

```
val ssc = new StreamingContext(sc, Seconds(5))
val tweetStream = TwitterUtils.createStream(ssc, None)

val user_status = tweetStream.map(status => status.getText())
    user_status.print()

ssc.start()
```

Figure 19: Twitter stream in Spark [94]

Streaming process starts by initializing start() method on the streaming context. Spark uses check pointing for errors handling and recovery purposes. It enables to setup checkpoint within streaming application, which will provide options to start the processing from certain point and not start all over when failure occur. Hence, the

batch data that are processed at failure time are reprocessed with data from the state before failure.

4.3 Architecture for Spark Sentiment Analyses

Apache Spark can implement solution for Twitter data mining with text classification and sentiment analyse on the scalable level. Architecture will require nodes in clusters, which will work in parallel for Twitter processing and other state of the art technologies.

Apache Spark can run in several modes [95]:

- Standalone local mode - Spark processes run as part of Java Virtual Machine process
- Standalone cluster mode - Spark runs in one node cluster when job-scheduling framework is applicable
- YARN – a resource management and central architecture design for Hadoop cluster computing systems
- Mesos, - an open source resource abstraction kernel for distributed cluster computing

Moreover, Spark clusters can be deployed by third party platform such as Elastic Cloud Compute by Amazon, IBM Bluemix or Microsoft Azure HDInsight. In addition, other Big Data platforms for data management, analytics and distributed computing are available, such as Cloudera, Hortonworks and MapR.

Architecture design for Twitter data mining and sentiment analyses solution implemented with Big Data technology, Apache Spark is showed on Figure 20.

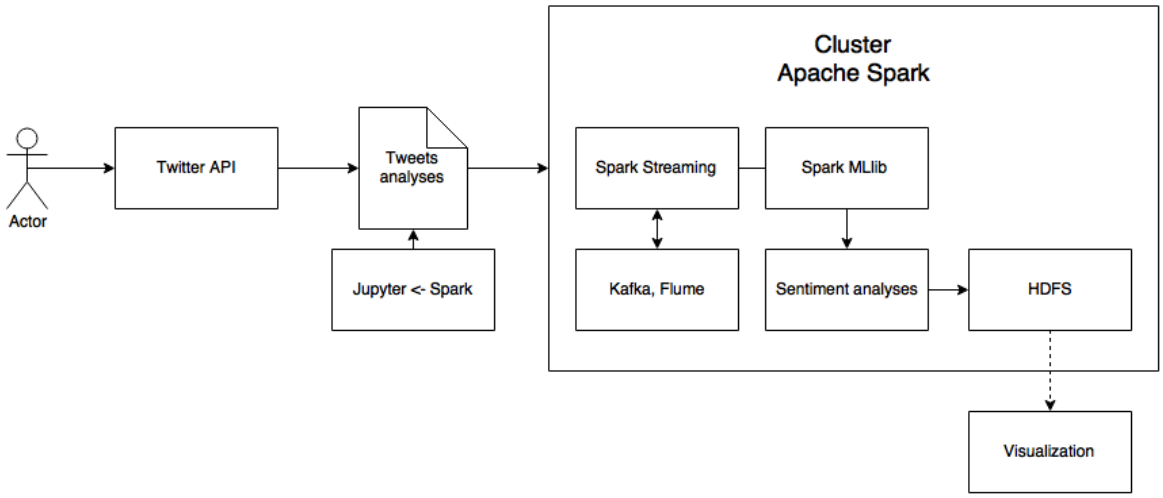


Figure 20: Twitter data mining architecture on Apache Spark cluster

Connecting to Twitter API by actor - data engineer leads to tweets analyses, which is done in order to understand data structures and format. As a tool for analyses can be used Jupyter that provides implementation within Spark. Cluster setup made with several available options as mentioned earlier depends on the processing needs of the solution. Inside cluster runs Spark with 3 modules.

Spark Streaming supports as input sources open-source technologies from Apache – Kafka and Flume.

Sentiment analyses on tweets can be implemented with Spark scalable machine learning library. Distributed file system then saves the output file into database. Visualization of the results from sentiment analyses is the final process within data flow.

5 Twitter Classification

Text classification became an important study field with expansion of social media, such as social networks. Challenges towards classification are connected with classifiers and their accuracy capabilities that depend on the specific model.

Classification is nowadays needed within various use cases in IT, such as spam detection, sentiment analyses, recommenders or predictions in near real-time.

5.1 Naïve Bayes Theorem

Naïve Bayes an algorithm based on Bayes theorem of probability, which purpose is to predict outcomes of unlabelled data. Bayes theorem is classification method that assumes the independence between predictors (features) and class.

Naïve Bayes models are divided within several types depending how they handle features. Bernoulli model uses features vector as Boolean, which means that each feature must have binary variable. Moreover, Bernoulli model does not care about frequency information of each word in document. Multinomial model cares about counts as features, thus word frequencies in document matters. [89]

Naïve Bayes classification algorithm is useful to classify even datasets with high volumes as it performs efficiently and is easy to implement. [88] [90]

$$P(c \mid x) = [P(x \mid c) * P(c)] / P(x)$$

- P (c | x) = Posterior Probability
- P (x | c) = Likelihood
- P (c) = Class Prior Probability
- P (x) = Predictor Prior Probability

$$P(c \mid X) = P(x1 \mid c) \times P(x2 \mid c) \times \ldots \times P(xn \mid c) \times P(c)$$

Posterior probability calculates the probability of outcome to occur given new information. In P(c | x), c represents class and x is predictor. Likelihood is the probability of predictor within class and other two probabilistic values belongs to

class and predictor. In order to understand Naïve Bayes theorem, an example will show probability prediction of an event to occur on Monday, Wednesday or Friday.

5.2 Naïve Bayes Twitter Classification

Training set will consist of days and variable, which will indicate whether an event occurs. In case an event will occur, variable indicate Y (Yes), otherwise N (No).

Day	M	W	W	M	F	M	F	W	W	M	F	W	M
Event	Y	Y	Y	Y	N	N	Y	N	Y	Y	Y	N	Y

Figure 21: Training set

Frequency distribution figure shows number of event occurrences by particular days. Moreover, the total sum for each possible event outcome is calculated.

Day	Yes	No
Monday	4	1
Wednesday	3	2
Friday	2	1
Sum	9	4

Figure 22: Frequency distribution

Probability for each row and column is calculated. Total sum of day occurrences is 13, thus probability for Monday is P (Monday) = [Y (4) + N (1)] / 13 = ~ 0.38%. Hence, the probability for Monday is 38%.

Day	Yes	No	Probability
Monday	4	1	~ 0.38
Wednesday	3	2	~ 0.38
Friday	2	1	~ 0.23
Sum	9	4	Sum = 13
Probability	~ 0.69	~ 0.31	

Figure 23: Probabilities calculations

With Naïve Bayes theorem, we can now calculate posterior probability that event will occur on Monday:

P (Monday | Yes) = 4 / 9 = ~ 0.44

P (Yes | Monday) = P (Monday | Yes) * P (Yes) / P (Monday)

P (Yes | Monday) = 0.44 * 0.69 / 0.38 = ~ 0.80

Calculation results show that probability is 80% for event to occur on Monday. This example should give an overview about Naïve Bayes theorem in order to understand its process. Moreover, example was created from random numbers, class and predictor. Hence, it should serve only for information purposes.

As Naïve Bayes is machine-learning technique, it can be applied for text classification for Twitter and sentiment analyses.

5.2.1 Bernoulli model

As mentioned before, for Bernoulli model matters only Boolean features, hence it does not matter if the same word occurred within tweet once or several times. This model of Naïve Bayes for Twitter sentiment analyses looks as example suggests.

Variable	Description
C	Tweet class for sentiment polarity (positive/negative)
F1	Word “amazing” occurs at least once in tweet (feature 1)
F2	Word “terrible” occurs at least once in tweet (feature 2)

Figure 24: Bernoulli model on tweet [91]

Figure 24 shows the probability of class given features P (C | F1, F2). According to Bayes, [88] [90], the probability for this class cannot be estimated, hence the theorem suggests:

$$P(A) * P(B \mid A) = P(B) * P(A \mid B)$$

As we replace A with the probability of features and take B as class C, we get an equation that will calculate probabilities for each class.

$$P(F1, F2) * P(C \mid F1, F2) = P(C) * P(F1, F2) \mid C)$$

Calculating posterior will look as follows:

$$P(C \mid F1, F2) = [P(C) * P(F1, F2) \mid C)] / P(F1, F2)$$

Likelihood is expressed by P (F1, F2) I C), which determines the probability of features occurrences within class C. In order to estimate the likelihood, probability theory comes with this equation:

$$P (F1, F2 \mid C) = P (F1 \mid C) * P (F2 \mid C, F1)$$

Since Naïve Bayes assumes the independence between features, above equation can be simplified into likelihood:

$$P (F1, F2 \mid C) = P (F1 \mid C) * P (F2 \mid C)$$

The final structure for posterior probability can now be expressed such as:

$$P (C \mid F1, F2) = [P (C) * P (F1 \mid C) * P (F2 \mid C)] / P (F1, F2)$$

Posterior can now calculate probabilities for C (class) taking the Boolean values for sentiment analyses – positive or negative. However, calculated probabilities give only the estimated prediction to which class given tweet will classify. Hence, class with the higher probability classifies the tweet.

Tweet word	Class
Amazing	Positive
Amazing	Positive
Amazing	Positive
Terrible	Positive
Terrible	Negative
Terrible	Negative

Figure 25: Classified tweet words

If we assume manually classified tweets by class as on Figure 25, we can discuss Naïve Bayes technique. Aim of this example is to use above probability formulas in practice. From the above figure we can see that tweet word “Terrible” is classified also as positive word, which is meant to simulate human statements that

may be ambiguous. For example: “We won the match, even tough the first period was terrible”. The probabilities for classes are:

$$P (C = positive) = 4 / 6 = \sim 0.67 \text{ and } P (C = negative) = 2 / 6 = \sim 0.33$$

Probabilities for features, F1 and F2 of class C are calculated according to feature occurrences within tweets. For example, the probability of occurrence for the word “Amazing” within tweets, knowing it is classified as positive is:

$$P (F1 = 1 \mid C = positive) = 3 / 4 = 0.75$$

Probability is calculated as number of positive tweets that contain feature F1 divided by number of all positive tweets. Lastly, the evidence of posterior probability defined for features as P (F1, F1) has to be known in order to classify tweets based on sentiment.

$$P (F1, F2) = P (F1, F2 \mid C = \text{“positive”}) * P (C = \text{“positive”}) + P (F1, F2 \mid C = \text{“negative”}) * P (C = \text{“negative”})$$

Taken probability that tweet contains feature 1 (word “Amazing”) only, the evidence is defined on this formula:

$$P (F1 = 1, F2 = 0) = (3 / 4) * (1 / 4) * (4 / 6) + 0 * 0 * (2 / 6) = 1 / 8$$

Tweet classification with Naïve Bayes enables to classify tweets text as sentiment value as showed on example. Knowledge about its algorithm has important meaning while implementing concept solution about Twitter sentiment analyses.

Knowing the all calculations necessary for Naïve Bayes classification from the example, we can classify tweet word “Amazing”.

$$P (C \mid F1, F2) = [P (C) * P (F1 \mid C) * P (F2 \mid C)] / P (F1, F2)$$

Figure 26, shows the probabilities for positive and negative sentiment classification of the tweet word “Amazing” and feature F1, as the result of Naïve Bayes.

Tweet word	F1	F2	Probabilities	Sentiment classification
Amazing	1	0	P (C = “positive” I F1 = 1, F2 = 0) = 1 P (C = “negative” I F1 = 1, F2 = 0) = 0	Positive

Figure 26: Tweet sentiment classification

6 Twitter Data Analysis

Application Programming Interface (API) is an interface that enables interaction with web services. It gives developers and public to develop service products on top of API and thus implement it within own service solutions. Access to Twitter service is possible by two types of different methods: Streaming API and REST API.

6.1 Twitter Streaming API

Streaming API offers access to service, when tweets are retrieved as continuous stream of information. It is up to developer's algorithm what kind of tweets are retrieved from API, such as keyword-based tweets, however in case of singular searches for user profile information, REST API will fit better. [83]

REST API is stands for Representational State Transfer. Its architecture is based on network principles, which defines access methods to resource data. RESTful services communicate over request-response HTTP protocol, however in comparison with Streaming API, it does not require to keep persistent HTTP connection opened. [86] Such service applications make requests to API only when user explicitly requests data retrieval, for example information about followers or re-tweeted post on Twitter.

Twitter service offers several types of streaming API endpoints, which has different capabilities, as showed on Figure 27. Twitter data are retrieved in JSON format.

Twitter Streaming endpoints	
Public streams	Contain samples of public tweets, recommended for data mining
User streams	Contain all information about particular Twitter user
Site streams	For service applications that requires handle streams from many users

Figure 27: Twitter Streaming API comparison [86]

From the above discussion, REST API will not serve sufficiently for purposes of this project, because of the HTTP server connection with user. On the other hand, Streaming API provides capabilities for real-time data mining with public stream

endpoint, which is enabled by separation between streaming connection process and HTTP request process.

Data mining technique, sentiment analyses will be implemented based on realtime stream from Twitter.

6.1.1 Streaming API architecture

Hence, Streaming API design architecture enable to collect necessary data for pre-processing stage in methodology. Architecture of Streaming API is showed on Figure 28.

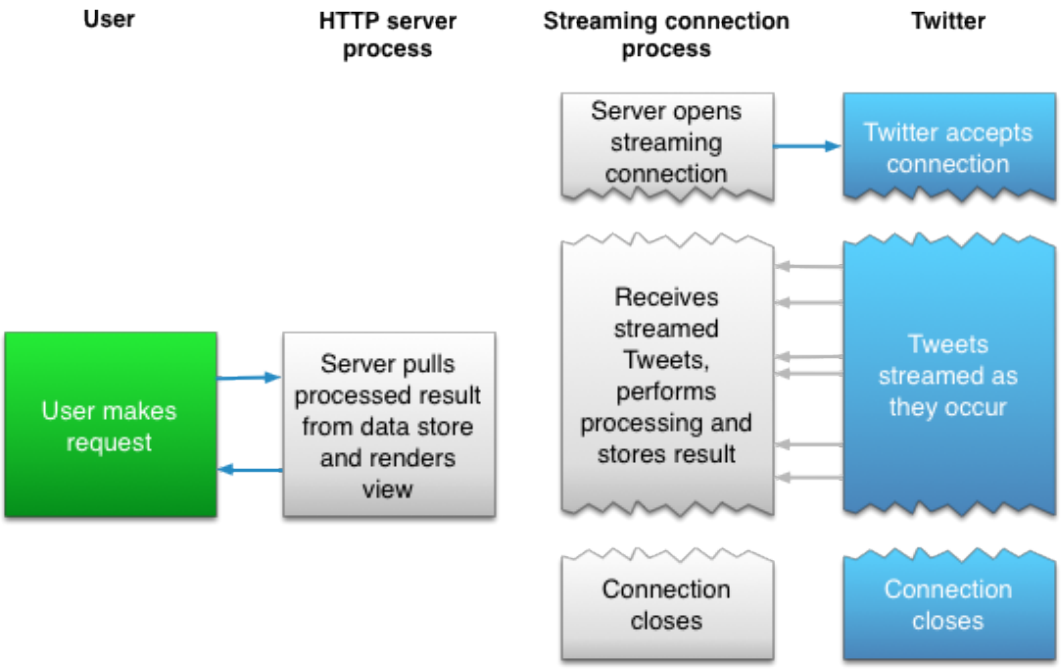


Figure 28: Twitter Streaming API dataflow [86]

Above architecture has advantage because after streaming process retrieve tweets from stream, it can manipulate data before store the results. User requests to access data are then handled with HTTP process, which query requested data from data storage.

6.2 Twitter Mining Application Setup

In order to initiate authorized calls to Streaming API and collect data for pre-processing phase, we need to create Twitter application that will obtain access token. Open Authentication (OAuth) is standard for authentication that provides

capabilities for applications to access data from other service without revealing credentials. In our solution for data mining we want to establish connection to Streaming API and thus Twitter control panel for developers, apps.twitter.com offers generating our access token.

6.2.1 Creating Twitter application

Firstly, the process of getting access token starts with creating new application from Twitter Application Management panel by filling up required attributes for application, such as: Name, Description and Website.

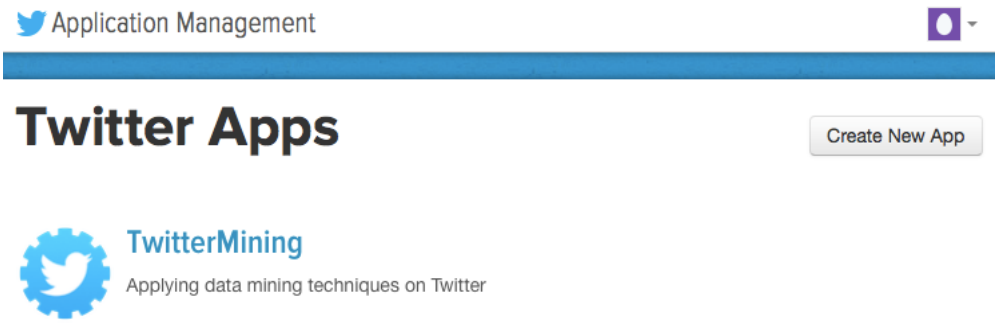


Figure 29: Twitter application setup

6.2.2 Obtaining Twitter credentials

For the development and Streaming API access are important credentials located under Keys and Access Tokens tab in newly created TwitterMining application settings. There are four credentials to note: Consumer Key (API Key) and Consumer Secret (API Secret), Access Token and Access Token Secret.

These credentials provide everything for TwitterMining application to authorize itself and make API requests on its Owner (mholubb) behalf. Owner username – mholubb represents researchers own Twitter account. Figure 30 shows configuration control panel with Twitter application settings. Credentials are blacked, as they present sensitive information.

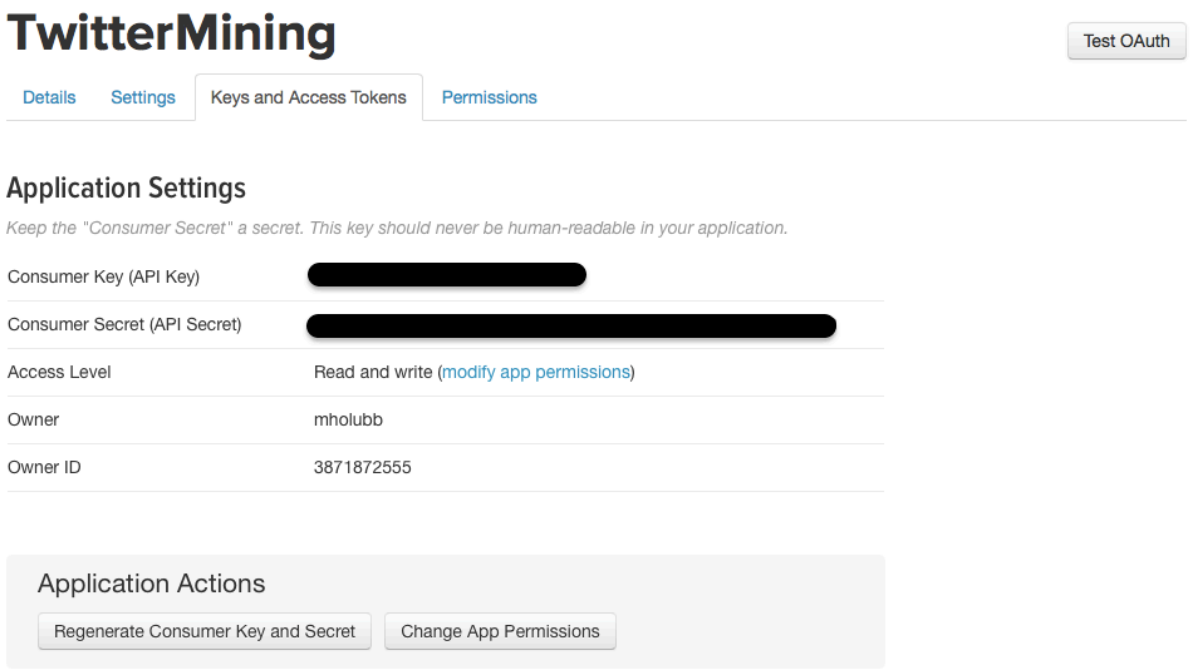


Figure 30: Twitter Mining Application Settings

Access tokens are used to make API requests to Twitter service from owners account. Moreover, access level is set to Read and write for purpose of this project, however if necessary it can be changed according to permission settings of application. Furthermore, tokens can be regenerated or revoked as show Figure 31.

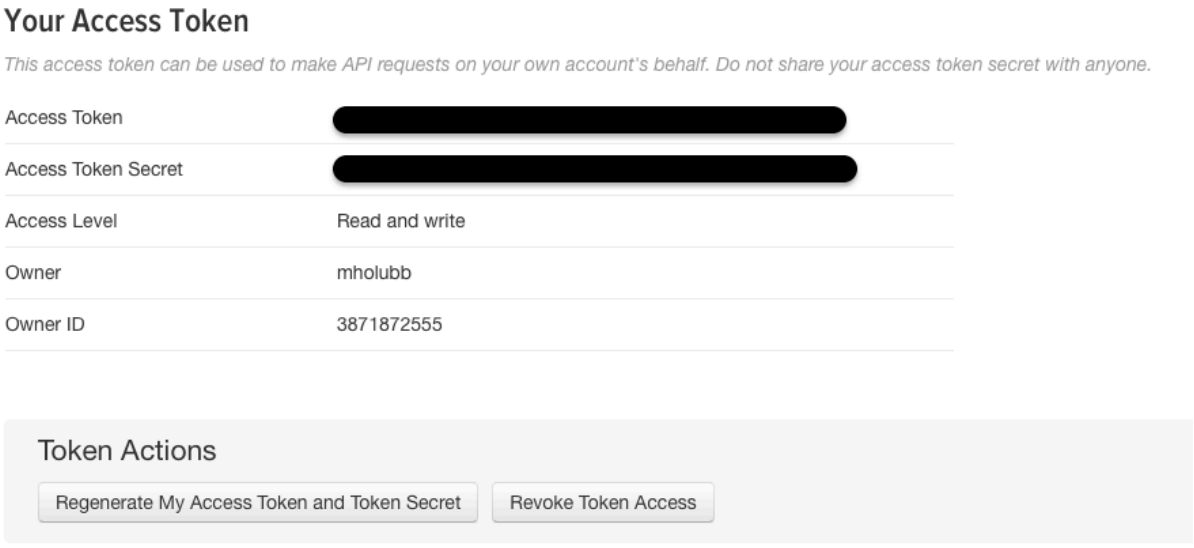


Figure 31: Twitter access token credentials settings

6.3 Creating Streaming Connection

After successfully obtained credentials for Streaming API, we need to initialize connection and collect sample tweets in order to analyse structure of each tweet and

attributes they contain. API key and API secret have to be passed to OAuthHandler that will create object *auth* in order to setup authentication while function set_access_token will setup Access Token and Access Token Secret. Code is included in the attachment file that comes with this project. Figure 32 shows the approach.

```
access_token=""
access_secret=""
consumer_key=""
consumer_secret=""
auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)
```

Figure 32: Twitter Streaming API authentication [86]

Data analyses require some sample partition of tweets to get familiar with its content, therefore the following code will stream near real-time tweets based on the user input. All tweets are printed in console, but for better overview they are piped also into text document. As example for user input: *Donald Trump,* tweet stream looks as on Figure 33.

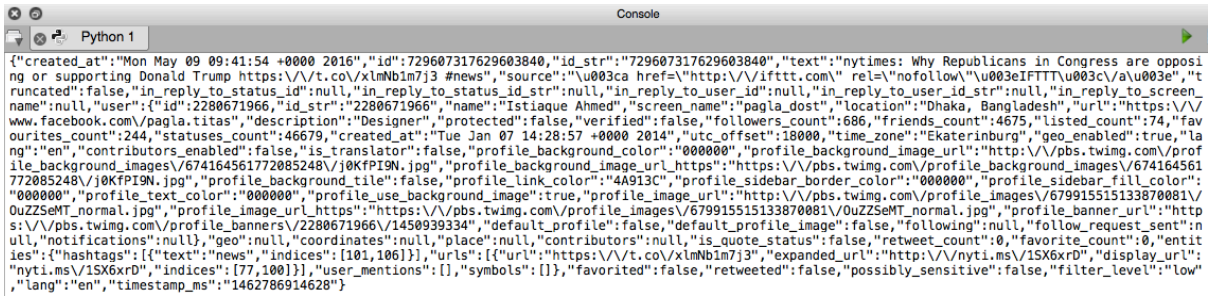


Figure 33: Twitter stream in JSON

Idea of analysing tweets as part of pre-processing before sentiment analyses is to understand attributes within JSON format for each tweet and normalize them according to natural language processing paradigms. Following *class* is listener instance for Twitter streaming that will print tweets and save them into document – *TwitterMining.txt* and inform user with message if any error occur with streaming.

```
class listener(StreamListener):

    def on_data(self, data):
        try:
            print(data)
            return (True)
        except BaseException as e:
            print("Error" ,str(e))

    def on_error(self, status):
        print (status)
        return (True)
```

Figure 34: Listener class [86]

Twitter streaming is ready after *auth* and *listener()* instances are passed to *Stream* object, that will now contain credentials for authentication and information about tweets, which have to be streamed. Moreover, the *filter()* method called on *twitterStream* will take input stream word from user and collect only tweets in English language.

```
twitterStream = Stream(auth, listener())
twitterStream.filter(track=[wstream], languages=['en'])
```

Figure 35: Streaming filter [86]

Process of collecting and analysing tweets is an important part of data mining, because it prepares data in format needed for processing. Collected Twitter stream of sample tweet dataset is now capable to store data in JSON format. JavaScript Object Notation (JSON) is format for data interchange based on comma-separated key-value pairs. It is human readable data format that is easy to parse.

6.4 Storing tweets in MongoDB

In order to store dataset from API, not only in text document but also into database, I have implemented MongoDB. It is NoSQL open-source database that is capable of storing massive volumes of data and supports effective data integration in dynamic formats such as JSON. Developing in Python enables to implement library for MongoDB called *pymongo*. From the methodology chapter, page 19 on data

mining and technologies, the middleware between Twitter Streaming API and MongoDB is *Jupyter*. It is a tool for interactive Python development, which serves in this project for Twitter data analyses. Following code will save *tweets* into MongoDB database, *"USA elections"*.

```
import pymongo
tweets = pymongo.MongoClient()["tweets"]["USA elections"]
tweets.find_one()
```

Figure 36: MongoDB connection

6.5 Tweets Attributes

Anatomy of the tweets stored in the database can be analysed from the Jupyter. Following are some interesting attributes that contain each tweet:

Tweet Attributes	
id	Unique tweet ID, that serves as identifier
created_at	Date when tweet was created
favourite_count	Number of favourites this tweet have
lang	Language
retweet_count	Number of retweets
text	Text of the tweet
entities	Url, hashtags #, user mentions
user: followers_count, friends_count	Information about user

Figure 37: Tweet attributes overview

From the attributes overview we can implement different data mining algorithms to perform processing, such as the most favourited tweets in comparison to users profile or analyses towards most used hashtags. However, for the sentiment analyses and classification with machine learning, text attribute matter the most. Based on this tweet text will perform algorithm implicitly decisions if it contains positive or negative statement. Classification on text is performed in the next chapter as part of machine learning techniques.

Collecting tweets from Streaming API are stored in database, however there is capability to view near real-time streaming output also in text file, which is used for purpose in pre-processing methodology.

6.5.1 Creating data frame

In data analyses, an important role is lead towards data selection based on data structures. Effective organized data into tables gives better overview for data scientists about its structures. In our tweet data analyses model, we use *Data Frames* enabled with Python Pandas. Data frame is capable to store data into tables and label its rows and columns. Hence, data mining can process data-driven solutions and deliver meaningful insights for companies, marketers, researchers, or brand management. Code is included in the attachment that comes with this project. Tweets that are stored in database are organized into Data Frames, Figure 38.

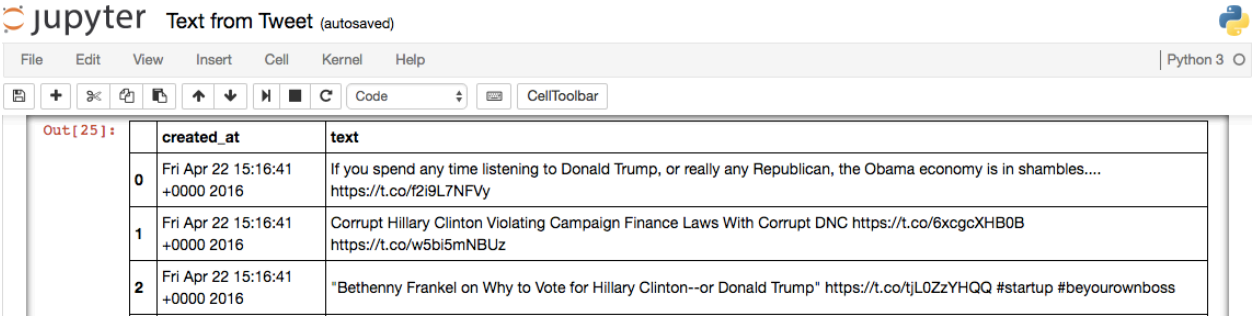
```
tweet_data = [{"created_at": item["created_at"],
               "text": item["text"],}

                   for item in tweets.find()]

tweet_data = pd.DataFrame(tweet_data)
pd.options.display.max_colwidth = 200
```

Figure 38: Organizing tweets into Data Frame

Following code will create Data Frame from collected tweets and displays in tabular view attributes *created_at* and *text* for each tweet in MongoDB database. Moreover, width of columns is configured for proper text displaying within its full range. Figure 39: shows the output as Data Frame, where tweets are organized into data structure from Jupyter environment.



	created_at	text
0	Fri Apr 22 15:16:41 +0000 2016	If you spend any time listening to Donald Trump, or really any Republican, the Obama economy is in shambles.... https://t.co/f2i9L7NFVy
1	Fri Apr 22 15:16:41 +0000 2016	Corrupt Hillary Clinton Violating Campaign Finance Laws With Corrupt DNC https://t.co/6xcgcXHB0B https://t.co/w5bi5mNBUz
2	Fri Apr 22 15:16:41 +0000 2016	"Bethenny Frankel on Why to Vote for Hillary Clinton--or Donald Trump" https://t.co/tjL0ZzYHQQ #startup #beyourownboss

Figure 39: Tweets Data Frame

6.5.2 Twitter data domain

Text attribute contains statements from Twitter Streaming API about upcoming USA presidential elections 2016. Reason for this particular study domain is because of its attention from worldwide media and social networks. It is highly discussed event, which is necessary for obtaining near real-time streaming data.

Capability of implementing algorithm on dataset has to be within domain, which outputs opinions or statements about particular event, product, brand, etc. Attention from users on Twitter on the event such as presidential elections provides necessary background for data mining and sentiment analyses, which distinguishes tweets based on their sentiment into positive, neutral and negative category.
All tweets are streamed from public API and therefore the opinions about elections come from random Twitter users.

6.5.3 Tweets mining and visualization

As social media can have huge impact on perception, aim is to deliver solution that classifies streaming tweets based on sentiment and results into public opinion, and visualized with graph.

Twitter data are now stored under assigned variable *tweet_data*. Hence, different analyses methods can be applied directly on this data frame. [92] In order to explore time zones and get geographical insights about Twitter users, which posted tweet with keyword “Donald Trump”, we can apply count method and visualize the output.

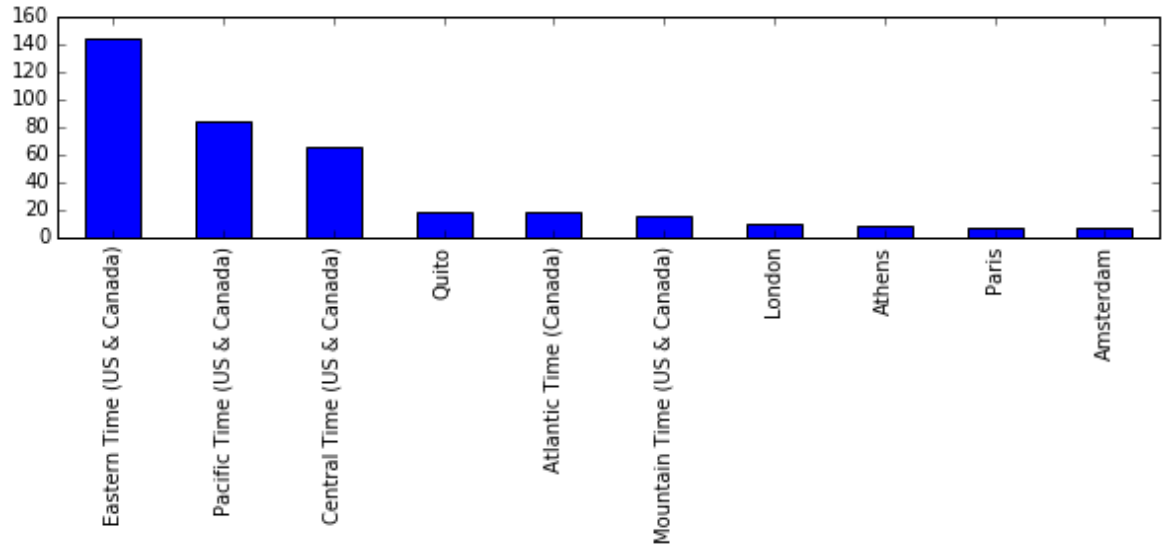


Figure 40: Tweets time zone counts

As the Figure 40 shows, the most tweets regarding search term “Donald Trump” comes from Twitter accounts set to US & Canada time zone, which makes sense since the search term relates to United States presidential elections. The x-axis depicts the time zones and y-axis number of tweets.

Moreover, analyses over Twitter data enable to get information about the most popular Twitter sources. Such insight brings overview on how users interact with the service. Following Figure 41 shows the 3 most used Twitter source applications.

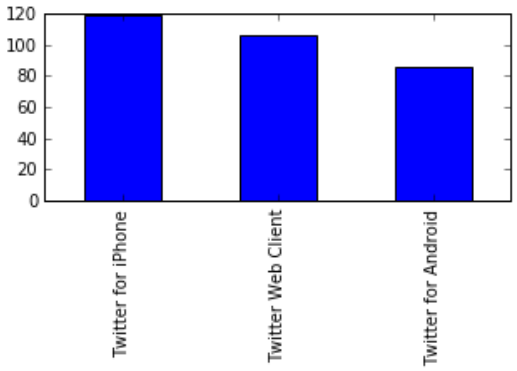


Figure 41: Twitter source overview

The x-axis depicts the most used Twitter client, which users use. This example shows capabilities of Twitter data mining and should serve for information purpose. Hence, the results of this insight were streamed from Twitter API and requested from data frame for analyses. The y-axis shows the number of tweets from the dataset that contained specific Twitter client attribute. Stream sample shows the most used client - Twitter for iPhone with 119 tweets, Twitter Web Client with 106 and third Twitter for Android with 86 tweets.

6.6 Tweet Sentiment Classification

Sentiment classification is part of machine learning techniques, which are further analysed on top of Twitter dataset. Figure 42 is an example on tweet sentiment classification. Text was manually taken from tweets that are stored in database to show sentiment classification capabilities.

Tweet sentiment classification example	
Positive	Bernie Sanders Wins The Climate Primary
Positive	I just want to say, "Donald Trump will be the GREATEST PRESIDENT EVER!!!
Neutral	NYC Fire District Releases Statement About Donald Trump
Neutral	Bethenny Frankel on Why to Vote for Hillary Clinton or Donald Trump
Negative	If you spend any time listening to Donald Trump, or really any Republican, the

	Obama economy is in shambles...
Negative	Bernie has turned into version of Trump. He is sneaky and low handed.

Figure 42: Sentiment categories

Further Twitter data analyses continue with natural language processing phase for sentiment analyses. Purpose is to clean tweets from redundant attributes and get normalized dataset for training and testing.

6.7 Natural Language Processing Analyses

Tokenization is a process of splitting text strings into tokens, which are represented by words in sentences. As part of Twitter data analyses, this method will help with Named Entity Recognition. From the tweet dataset we can request sample data and tokenize them to see the results. [84] Purpose of tokenization is to analyse ways of efficient tweet splitting into tokens and comma-separated mentions. Moreover, analyses lead towards data normalization, thus clean tweets from so-called mentions – retweets (RT), hashtags (#) or URLs.

Stop-words are words that have not any value for sentiment analyses, because they do not contain any meaning in context. Natural Language Toolkit provides library with stop-words, which can be implemented in solution. They are categorized by language and so our solution will attach only English stop-words. Natural language processing provides also extensive techniques for implementation into data mining solutions and enables machines to understand semantic structures between words in sentences. Functionality based on above-mentioned pre-processing techniques depicts Figure 43.

Twitter Data Analyses	
"RT @BernieSanders: There is nothing I would like more than to take on and defeat Donald Trump, someone who must never become president of 2016"	
NLTK technique	Output
Tokenization	['RT', '@', 'BernieSanders', ':', 'There', 'is', 'nothing', 'I', 'would', 'like', 'more', 'than', 'to', 'take', 'on', 'and', 'defeat', 'Donald', 'Trump', ',', 'someone', 'who', 'must', 'never', 'become', 'president', 'of', '2016']
Stop-words	['RT', '@', 'BernieSanders', ':', 'There', 'nothing', 'I', 'would', 'like', 'take', 'defeat', 'Donald', 'Trump', ',', 'someone', 'must',

	'never', 'become', 'president', '2016']

Figure 43: NLTK techniques on sample tweet

Outputs of code implementation show functionalities of natural language processing. From stop-words column we can analyse that redundant words from sample tweet, such as: “is”, “more”, “than”, “to”, etc. are removed.

Chunking is a method of grouping words into categories based on their natural expressions. It means to group different words from sentences into part-of-the speech (POS) chunks. Hence, sentence can be divided into several chunks, which depends on number of POS occurrences, such as nouns or verbs in sentence.

Natural language processing provides chunking method known as Named Entity Recognition (NER), which belongs to advanced forms of entity recognition from text. Purpose of use is to implicitly train algorithm to distinguish between names, locations or organizations and POS entity tagging. NER based on machine learning model requires tokenizing tweets in order to process each word of tweet as token, as well as part-of-the speech tagging. Chunking method will perform tagging on tokenized tweet and organize it by entities.

Twitter Data Analyses
“RT @TheDailyShow: Donald Trump is now the presumptive Republican Party nominee for President of the United States”
Named Entity Recognition (NER)
(S RT/NNP @/NNP TheDailyShow/NNP :/: (PERSON Donald/NNP Trump/NNP) is/VBZ now/RB the/DT presumptive/JJ (ORGANIZATION Republican/NNP Party/NNP) nominee/NN for/IN President/NNP of/IN the/DT

(GPE United/NNP States/NNPS))

Figure 44: Named Entity Recognition on sample tweet

Chunking and NER natural language technique recognized from the sample tweet various entities. Algorithm has tagged different POS and performed chunking under PERSON, ORGANIZATION and LOCATION (GPE) entities as showed on Figure 44.

With Twitter data analyses other important techniques for data normalization are stemming and lemmatizing.

Stem is represented as root for word, where different forms of the same word exist in natural languages. Stemming technique operates as algorithm that removes redundant suffix from words. Algorithm for sentiment analyses does not require knowing different variations and tenses of tweets.

On the other hand, lemmatization is techniques that results into valid words as lemma perform as root word. As oppose to stemming, lemmas are words looked up from wordnet NLTK. Lemmatization as advanced normalization technique has challenges in sense of explicitly specifying POS for words, because default is set to noun. This may cause issues with resulting in different lemmas.

Twitter Data Analyses	
"RT @YahooFinance: Donald Trump believes he is a financial genius, which is a problem"	
Stemming	Lemmatization
RT @ YahooFin : Donald Trump believ he is a financi geniu , which	RT @ YahooFinance : Donald Trump belief he is a financial genius , which

is a problem	is a problem

Figure 45: Comparison between stemming and lemmatization on sample tweet

Differences between these two techniques are showed on Figure 45. Lemmas will always keep form of real words, meaning they will compare with wordnet dictionary for normalization. Hence, if there is not explicitly specified POS they will remain in the default form.

From the above comparison figure we can discuss that word – "*believes"* from the sample tweet has two form with stemming and lemmatization, which shows their main difference. While stemming shorthand *"believes"* into *"believ"*, lemmatizor outputs valid word – *"belief"*. Natural language processing analyses on Twitter data has evaluated capabilities for pre-processing phase in Twitter data mining.

6.8 Tweets Normalization

Twitter Streaming API stream its data in JSON format as discussed earlier. Tweets contain various key-value attributes about its context and information about user profile. For sentiment analyses is most important the text of actual tweets on which base is deployed algorithm that will classify them into sentiment categories by their polarity.

In order to perform data cleaning algorithm on dataset with streamed tweets, initial code has to specify which segments of JSON attributes to include. Attribute under which is tweet sentence located is in JSON called – *text*. Hence, splitting up tweet data by *text* gives ability to clean tweets from other unnecessary attributes, such as: "created_at", "id_str", "source", etc.

Twitter Data Analyses
Stream keyword: Donald Trump
Twitter Streaming API data format
{"created_at":"Fri May 13 16:54:25 +0000 2016","id":728266610721574913,"id_str":"728266610721574913","text":"RT @Always_Trump: Donald #Trump supporters are actually SMARTER than average Americans, study finds - https:\/\/t.co\/V0NBMC2CtH","source":"https:\/\/t.co\/\u2026 href=\"http:\/\/twitter.com\/download\/android\" rel=\"nofollow\"\u003eTwitter for Android\u003c\/a\u003e" …

Splitting by *text* tweets cleaning
RT @Always_Trump: Donald #Trump supporters are actually SMARTER than average Americans, study finds - https:\/\/t.co\/V0NBMC2CtH
Pre-processing tweet format
donald trump supporters are actually smarter than average americans, study finds

Figure 46: Twitter data cleaning

Pre-processing phase of Twitter data mining shows Figure 46. Tweets that are streamed from Twitter Streaming API in JSON format are now cleaned from redundant information and normalized for purpose of sentiment analyses. Above figure depicts the difference between tweets format before and after data cleaning, which is requirement for further classification.

Natural language processing in combination with regular expressions enable to implement algorithm that will pre-process dataset with stored tweets. Regular expressions are used in order to specify strings that follow certain rules in their structure. Since text attribute in JSON format contain Twitter mentions, such as hashtags, URL's or re-tweet sign (RT), by implementing regular expressions, these information can be removed. Twitter mentions are replaced by whitespace and each tweet is transformed into lower case text format. Code is included in the attachment file that comes with this project. Function for data cleaning is showed on Figure 47.

```
def normalize(tw):
    tw = tw.lower()
    tw = re.sub(r"rt", "", tw)
    tw = re.sub(r"http\S+", "", tw)
    tw = re.sub('@[^\s]+','',tw)
    tw = re.sub('[\s]+', ' ', tw)
    tw = re.sub(r'#([^\s]+)', r'\1', tw)
    tw = tw.strip('\'"')

    return tw
```

Figure 47: Function for tweets data cleaning

Dataset in this analyses phase contains tweets regarding upcoming United States presidential elections 2016. Streaming is based on keywords, which represent two politicians – Donald Trump and Hillary Clinton. They are leaders of the

Democratic and Republican parties, thus the most attention from public goes towards them.

Pooling takes part in election period as research tool for finding public opinions on current event. Sentiment solution for presidential elections as expected outcome of this project could help with pooling, because social networks, such as Twitter are media that produces massive volumes of data, which can refer to Big Data. Hence, the effective processed streaming tweets can output valuable insights about public opinion about elections.

Expected outcome after training and testing phase and solution implementation is to classify tweets on their polarity and polarity confidence. Polarity value will represent “pos” for positive and “neg” for negative tweet sentiment.

7 Concept Development

Solution for sentiment analyses on Twitter is discussed within concept development chapter that includes implementation with natural language processing, machine learning algorithm and graphical visualization.

7.1 Supervised Machine Learning

Classifiers can be divided into supervised and unsupervised, which indicates machine-learning technique that is related to classification. Solution for sentiment analyses of this project requires training and testing classifiers in order to classify unlabelled tweet stream. Hence, with supervised machine learning algorithm it is possible to acquire feature set, which has a form of tuple, build classification model and use it for sentiment label classifier on live Twitter stream.

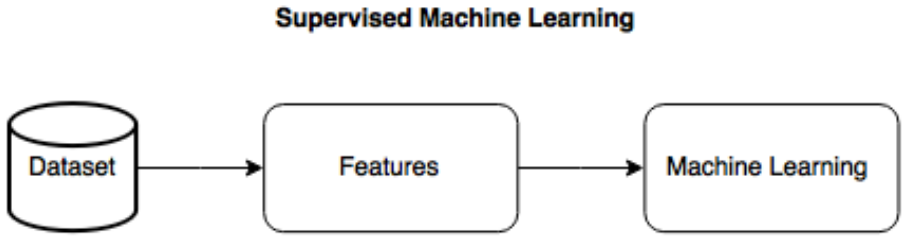


Figure 48: Training data flow

Process of supervised machine learning shows Figure 48. This data flow consists of input dataset from NLTK, which is necessary for sentiment classification. Dataset has classified sentiment into two categories, also called “labels”.

Training process takes the input dataset and accordingly algorithm analyses features from it. Hence, each text within training dataset is categorized as unique word and part of a feature set. In addition, feature sets with particular labels are fetched to machine learning algorithm that perform training and build classification model based on extracted features.

7.2 Natural Language Processing Implementation

NLTK corpus is build of different corporas that are capable to implement algorithms for defining classifier. In order to access already labelled data to train and test classifier, natural language processing provides dataset with movie reviews that are divided into positive and negative text files. [87] They are stored under “pos” and

"neg" folders within NLTK data directories, as for example: "pos/cv000_29590.txt" or "neg/cv000_29416.txt". Reason of choosing this approach with movie reviews is because of its structure of already labelled files.

Due to scope of this project, manually labelled tweets with sentiment would be time inefficient, because of the number of tweets in dataset. Moreover, best percentage accuracy for sentiment classifier may be assessed, when it is trained on more data. Code is included in the attachment file that comes with this project.

7.2.1 Tokenization

As mentioned before, movie reviews are categorized into positive and negative categories, while each review has assigned file ID number. Algorithm has to tokenize all words and create tokenized version of review. Moreover, at the end of the tokenized word list is defined sentiment polarity, as shows Figure 49.

```
'on', 'television', 'shows', 'can', 'be', 'counted', 'on', 'one', 'hand', '(', 'even', 'one',
'that', "'", 's', 'missing', 'a', 'finger', 'or', 'two', ')', '.', 'the', 'number', 'of', 'ti
mes', 'that', 'i', 'checked', 'my', 'watch', '(', 'six', ')', 'is', 'a', 'clear', 'indicatio
n', 'that', 'this', 'film', 'is', 'not', 'one', 'of', 'them', '.', 'it', 'is', 'clear', 'tha
t', 'the', 'film', 'is', 'nothing', 'more', 'than', 'an', 'attempt', 'to', 'cash', 'in', 'o
n', 'the', 'teenage', 'spending', 'dollar', ',', 'judging', 'from', 'the', 'rash', 'of', 'rea
lly', 'awful', 'teen', '-', 'flicks', 'that', 'we', "'", 've', 'been', 'seeing', 'as', 'of',
'late', '.', 'avoid', 'this', 'film', 'at', 'all', 'costs', '.'], 'neg')
```

Figure 49: Tokenized dataset with negative sentiment label – "neg"

7.2.2 Frequency distribution

Frequency distribution of words from datasets is function, which can filter out stop-words, convert all characters to lower case and avoid whitespaces. It shows the most used words within processed text. In order to build classifier from the movie reviews dataset, it must define distribution of the most frequent words. Thus, variable *movie_feat* will list 2000 most frequent words, which are set to lower case characters. Moreover, the stop words from NLTK are removed as showed on Figure 50.

```
stop_words = stopwords.words('english')
movie_words = FreqDist(j.lower() for j in movie_reviews.words()
if j.lower() not in stop_words and j.lower() not in string.punctuation)
movie_feat = list(movie_words)[:2000]
```

Figure 50: 2000 most frequent words

Method **def list_feat**(movie) will process whole dataset with both positive and negative reviews and learn about their sentiment based on as the most frequent words occur. Method for classifier is an important part of the algorithm as it iterates through all movie reviews and learns about the most frequent words. It gives an overview about each word whether it appears more within positive or negative reviews and thus can be deployed for testing the classifier's accuracy.

When this method is initialized on the specific tweet, it will output "True" or "False" statements depending if the words from text tweet appear in the feature set dictionary.

7.3 Training Classifier

In order to train and test classifier and predict labels from new reviews based on their text attribute, dataset has to be divided into testing and training set. Supervised machine learning is technique used within this section, as the algorithm tries to predict unlabelled data, after it run on the labelled positive and negative movie reviews. In order to get the higher accuracy of classifier, training and testing is directed on the random shuffled movie dataset.

Feature set contain set of words that are features for classifier. Therefore, training phase of algorithm will have dedicated ¾ of the dataset volume with both positive and negative sentiments. On the other hand, classifier testing will be performed on the remaining ¼ to follow classification accuracy.

Naïve Bayes is machine-learning algorithm that can perform efficiently for data classification. It is probabilistic classifier that assigns labels to feature values, while this label is taken from to known dataset. Bayes classifiers have assumptions about features, where one feature is independent from other, if they belong to particular class, meaning positive or negative. Moreover, Naïve Bayes chooses the label for the input data based on the probability of the labels as they occur in the training set. Likelihood estimation is determined for each label and the highest label with estimation reference the input data. [88]

In addition, features can output various types models that they specify. Hence, different models can be analysed and run as classifiers for sentiment analyses. Bernoulli classifier consider Boolean features, thus it does not make difference in accuracy if same words appeared several times within tweet/movie text

attribute. Multinomial classifier classifies text and outputs accuracy, while it takes word counts as features.

7.3.1 Classifiers accuracy

Training different classifiers on the dataset will result in comparison for their accuracies. Classifier with the highest accuracy will be saved for the Twitter sentiment analyses. For text classification, three Naïve Bayes classifier models will be trained and tested on the movie reviews features.

Classifiers comparison	
Classifier	Accuracy
Naïve Bayes	65%
Bernoulli	63,2%
Multinomial	64,4%

Figure 51: Classifier accuracy of different Bayes models

After training and testing phases, we received accuracy percentages for each classifier as showed on Figure 51. However, because these documents are random picked by algorithm, new iteration of training and testing may output skewed percentages in accuracy as algorithm showed. In addition, range for the accuracy balances from 60% to 67% in accuracy. However, results of Naïve Bayes performed the best within training and testing iterations. Scikit-learn tools for Python enabled implementation of machine learning with classification.

7.4 Supervised Sentiment Classification

Classification data flow of supervised machine learning that is built on the training model shows Figure 52.

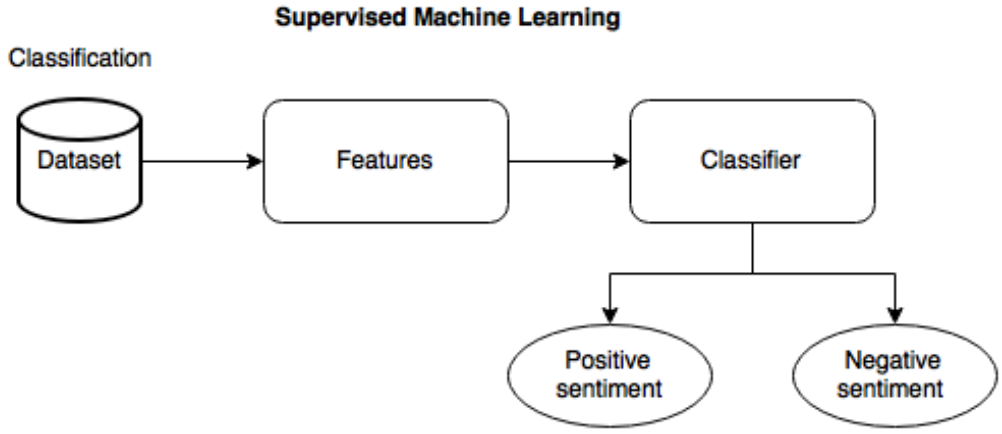


Figure 52: Supervised classification

In this model, the dataset represents unlabelled data, which are going to be classified. Hence, no sentiment value is tied up with the input data. Process continues with feature extraction from the input data. In addition, the trained classification model is implemented on the features from which classifier classifies the input data as positive or negative.

7.4.1 Naïve Bayes classifier

Since, Naïve Bayes classifier showed the highest accuracy result, it would be used as classifier for Twitter sentiment analyses. Advantage, is that this training classifier can be stored with as object and initialized after connection to Twitter Streaming API. JobLib is a pipelining tool that enables to store Naïve Bayes classifier as persistent object in a file, and load it to the memory when necessary.

```
joblib.dump(naiveBayes, 'naiveBayes.pkl')
naiveBayes = joblib.load('naiveBayes.pkl')
```

Figure 53: Saving and loading trained model

Naïve Bayes is classifier that will load *"joblib"* object – naiveBayes.pkl, which is then called from method "*tweet_sent*" in the return.

```
def tweet_sent(text):
    feats = list_feat(text)
    return naiveBayes.classify(feats)
```

Figure 54: Twitter sentiment method for Naïve Bayes classification

Calling method *"tweet_sent"* showed on Figure 54 from the **class listener(StreamListener):** that defines the streaming format of tweets will implement Naïve Bayes classifier on the streamed tweets. Sentiment is then defined and stored in variable "tweet_sentiment" that references to "sent" for sentiment file - **tweet_sentiment = sent.tweet_sent(tweet)**. Code is included in the attachment file that comes with this project.

Solution for Twitter sentiment analyses now includes sentiment classifier and streaming API algorithm that will take input from user for desired streaming keyword.

7.4.2 Sentiment visualization

As mentioned before in the Twitter data analyses chapter, solution was tested on the candidates in United States presidential elections 2016. When user writes the keyword, algorithm will initiate the streaming and sentiment classifier, which decides about the tweets sentiment in near real-time. Moreover, the polarity of each tweet is stored in file, which is given as input for live graph, which visualizes sentiment about the streamed keyword.

When the tweet polarity is evaluated with positive sentiment, graph will record increase point in the line. On the other hand, if negative sentiment, the graph line will decrease. X-axis represents the number of streamed tweets, which have been evaluated by algorithm about their sentiment. Y-axis represents the sentiment score about current keyword.

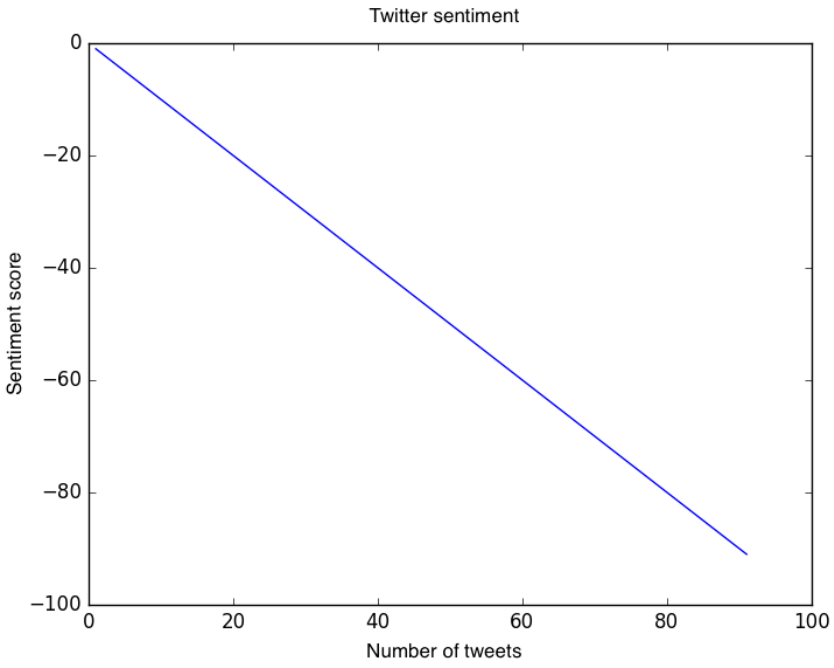


Figure 55: Twitter sentiment graph

Figure 55 depicts the live graph that shows result from the Twitter sentiment analyses. Testing keyword was – Donald Trump. From the figure, we can see strongly negative sentiment towards this keyword. We can observe trend based on 95 tweets that declined from the origin by 95 points. This means the statements within tweets were negative oriented, without any positive variation. This outcome might be caused by implicit sentiment conversions of neutral tweets to negative.

Thus, implementation is trained only on two outcomes, and therefore all neutral tweets will output negative sentiment. Outcome for keyword tested showed negative sentiment, as we assume is caused by tweets text attributes, which contains more opinion statements compared to different keywords. Implementation for such keywords resulted into negatively declining sentiment graph curve without any positive values.

Aim was to train classifiers on features that are extracted from positive and negative movie reviews and after use the classifier with the highest accuracy for Twitter sentiment classification. Testing showed that accuracy of classifiers changes in the range with variation up to 7%. This finding indicates the disadvantage of classification on movie reviews.

7.5 Explicit Labelled Tweets

Correctness of sentiment polarity as solution is also implemented as concept with explicitly labelled tweets. Tweets are manually labelled with sentiment after they were streamed and saved into text file.

7.5.1 Training classifier

Features are extracted from training Twitter set and passed to classifier for testing. Model is trained from explicitly labelled tweet dataset and its extracted features as Figure 56 shows. This example is meant to discuss the sentiment analyses that are built upon tweets rather than movie reviews.

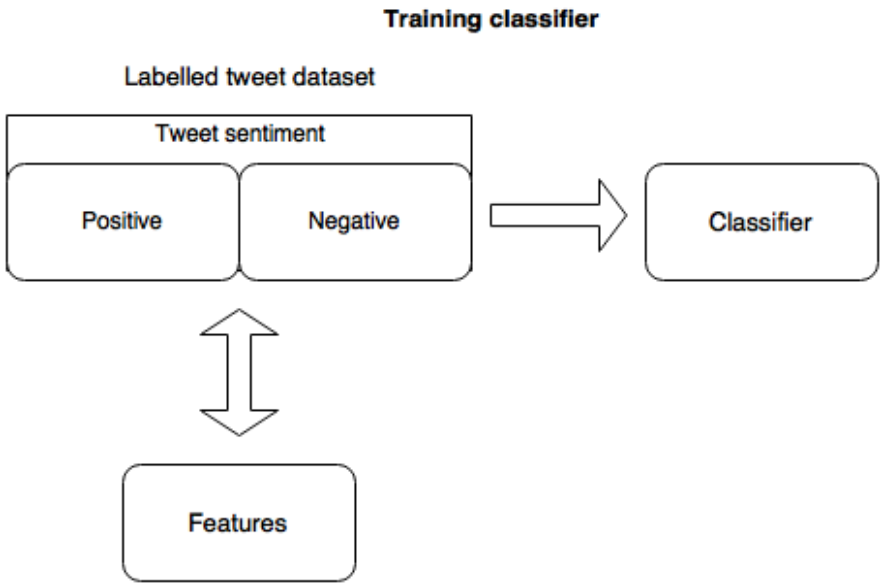


Figure 56: Training classifier with tweets

However, this solution should be looked at only as concept because the requirements for supervised machine-learning approach are not met as with the previous implementation with movie reviews.

Positive and negative related tweets are organized into sample dataset as shows Figure 57 and Figure 58.

Positive tweets
RT @WeNeedTrump: Donald Trump has won 11.1 million votes so far this primary season
Our troops deserve a strong commander in chief. We need Donald Trump. #MakeAmericaGreatAgain
Donald Trump will rebuild our infrastructure, end our corruption, and make the United States a winning country.

Figure 57: Positive tweets

As showed on the above figure with positive tweets, particular words indicate positive attitude towards tweet text. In the third tweet, we can observe words as: “corruption” or “winning”. In human language the tweet is considered as positive, which implies from its context. However, I assume that machine algorithms can classify it differently because of the word “corruption”. Hence, it depends on the feature training and classifier model accuracy. Such situation could lead to biased sentiment results.

Negative tweets
RT @BradThor: Donald Trump - the worst Republican nominee ever
Hillary Clinton is the single biggest liar in American politics. She would be a disaster in office.
I look so forward to debating Hillary Clinton! Democrat Primaries are rigged, e-mail investigation is rigged

Figure 58: Negative tweets

Negative tweets contain words such as: “worst”, “disaster”, “liar” and “rigged”. Assumptions are made that algorithm will find such words more in negative oriented tweets than positive. Frequency distribution after features are extracted from words can show better overview about how the classifier treats various words. Moreover, the probability score applied on the classifier can show the most valuable features that were extracted from training tweets.

7.5.2 Tweets sentiment classification

Classification continues with pre-processing phase, when all tweets are normalized with method for normalization. Tweets have now removed redundant information and labels according their sentiment. In addition, tweets are stored in array as tuples Figure 59. Figure depicts example of tuples from positive tweets. Each tuple consists of sentiment polarity and tweet text.

Tweets tuple	
Sentiment	Text
"pos"	"donald", "trump", "11", "million", "votes", "far", "primary", "season"
"neg"	"donald", "trump", "worst", "republican", "nominee", "ever"

Figure 59: Example of tweets tuple

Tweet text is tokenized and reduced from stop-words after method initialize tweet cleaning algorithm. Format of tweets after normalization depicts Figure 60.

```
>>> runfile('/Users/marekholub/Desktop/ThesisRes/Thesis/tweetSentimentThesis.py', wdir='/Us
ers/marekholub/Desktop/ThesisRes/Thesis')
[('pos', ['donald', 'trump', 'won', '11', 'million', 'votes', 'far', 'primary', 'season']),
('pos', ['troops', 'deserve', 'strong', 'commander', 'chief', 'need', 'donald', 'trump']),
('pos', ['donald', 'trump', 'rebuild', 'infrastructure', 'end', 'corruption', 'make', 'unit
ed', 'states', 'winning', 'country']), ('neg', ['donald', 'trump', 'worst', 'republican', '
nominee', 'ever']), ('neg', ['hillary', 'clinton', 'single', 'biggest', 'liar', 'american',
'politics', 'would', 'disaster', 'office']), ('neg', ['look', 'forward', 'debating', 'hilla
ry', 'clinton', 'democrat', 'primaries', 'rigged', 'e-mail', 'investigation', 'rigged'])]
```

Figure 60: Twitter data normalization

Figure 61 depicts classification process on unlabelled tweet data, which are going to be classified with Naïve Bayes algorithm.

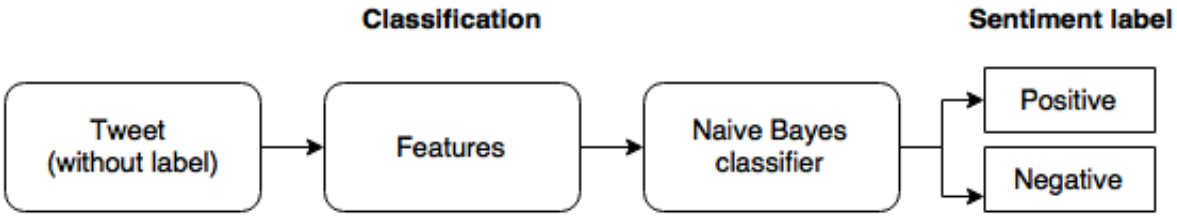


Figure 61: Classification

Moreover, classification process includes Naïve Bayes classifier that is implemented from the joblib file and tested on sample tweet without any label.

```
print(tweet_sent("hillary clinton with huge success in primaries"))

neg
```

Figure 62: Negative sentiment outcome

Outcome of the Naïve Bayes sentiment classification on sample tweet text resulted into negative sentiment label. However, the expected outcome was positive label as the tweet suggests positive statement. This implementation showed that the reliability of sentiment classification with this method is limited by the accuracy. Hence, the solution is not able to produce relative results, because of the inefficient initial dataset used for algorithm training, which should contain only relevant tweets from the same domain, in this case politics.

Sentiment analyses can be applied also with other use cases, such as movies or brands. Valuable insights that sentiment analyses offering can be applied within marketing applications. Moreover, the value proposition from Twitter data mining can lead to increased volume sales and competitiveness on the market.

7.6 Graphical User Interface

Insights that sentiment analyses offers can be used within various services and serve also as tool for users to track worldwide perception from specific keywords. Concept application for this purpose is designed as on Figure 63.

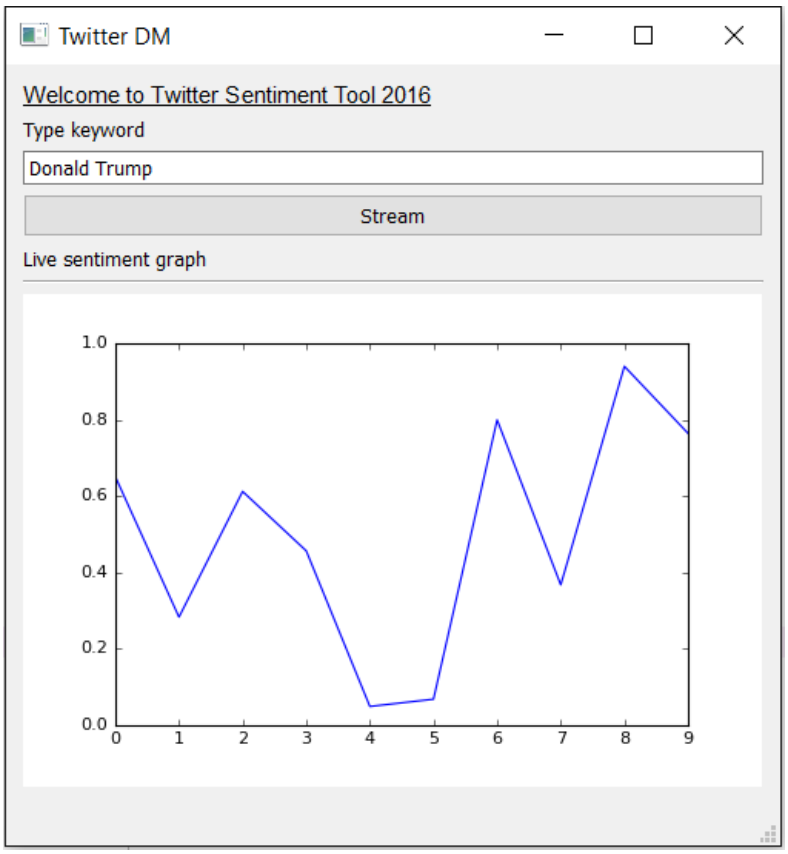


Figure 63: Twitter Sentiment Tool GUI

Graphical user interface for Twitter Sentiment Tool is implemented with Qt designer, which is a platform to create innovative and modern UIs and applications. GUI design is accordingly converted into Python file application. Purpose of this design is to enable users to easily track the sentiment from Twitter. User interface has responsive design and the key aspects for simplicity. Application serves as concept design and suggests the possibilities on displaying live graph to users. Graph on Figure 63 is random and used only for graphical interface design, how it could look in the future.

When user types keyword in the field box, the input value is given to Twitter stream filter method. From the figure, the chosen keyword is: *Donald Trump*. Then, if user clicks the “Stream” button, it initiates the near real-time streaming process, which includes the data normalization, classification and graph visualization.

Since, the Twitter application uses author’s personal credentials for authentication, this concept is developed as prototype, however, in the future development, users should directly authenticate with their own credentials to the Twitter sentiment tool. Therefore, the application has no functionality and the graphical interface serves only as potential design for the future implementation, which is discussed in the final chapter.

8 Conclusion

Purpose of this project was to develop a concept for sentiment analyses, which is able to process real-time streaming feed from Twitter API and classify its polarity in order to help industry and users achieve valuable insights. In order to accomplish the problem formulation, academic research framework was specified in the methodology chapter, which divides project into several sections.

Firstly, the introduction chapter presents the initial research in the Twitter field of study, analysing Big Data characteristics, which have to be noticed when developing such solutions. Sentiment analyses overview research gives the input source of information about the challenges it contains and to be dealt with. Desktop research supports the problem formulation and literature study discusses different fields of given problem domain. State of the art chapter includes discussions about Big Data technologies that relates to Twitter processing, but also elaborate on third party applications and databases. Machine learning algorithms are important input in this project; therefore discussion on supervised and unsupervised techniques is researched. In addition, study on the natural language processing capabilities and threats take part, because it defines the core implementation of sentiment classification.

Secondly, Twitter data analyses are investigated with creating application and connecting to Twitter Streaming API in order to get application credentials. In addition, tweets are analysed on their format and structure, which is needed for normalization and pre-processing task. Tweets are stored in database in order to perform data mining and visualize outcome.

Thirdly, project discusses Twitter in relation to Apache Spark and the architecture design is presented for Big Data solution. In addition Naïve Bayes classification algorithm was discussed based on Twitter example.

After research and analyses, solution as developed concept for Twitter sentiment analyses took place. Solution in this project consists of several data mining techniques that contain natural language processing in order to normalize tweets, text classification with supervised machine learning algorithm, which was trained and tested on labelled movie reviews in order to use it for tweet classification, which was in this project tested on tweets from United States presidential elections 2016

candidates. Moreover, the sentiment outcome was visualized on the live graph. In addition, graphical interface for concept application was developed for enhanced user experience.

However, the challenges with natural language processing and classification can to be overcome with specifically classified tweets within domain of sentiment analyses. Hence, the training and testing datasets must be explicitly classified in order to achieve right results.

Project answered the problem formulation and showed different data mining techniques how to find valuable insights on Twitter. Twitter analyses discussed the techniques how to convert streamed tweets into data frame and find insights about the Twitter users, geographical time zones and their interactive with service by number of devices, such as iPhone, Web or Android. Moreover, the classification and natural language processing resulted into sentiment analyses, on which is build concept solution.

However, I conclude that the concept development for Twitter sentiment analyses dealt with inaccurate classification because of the possible bias within tweets. This can be caused by tweets strong inclination into negative polarity, which could be implicitly chosen also for tweets with neural sentiment.

Solution with graphical interpretation is biased with negative tendency to classify all the live streaming tweets. Hence, the solution of this project presents only concept for developing sentiment applications, because in the ideal implementation it would require initial labelled dataset with normalized tweets for training and testing the classifier in order to extract relevant features.

The relevancy of the outcome for Twitter sentiment analyses is also affected by quality of tweets text, which is limited to number of characters. Implementation with the different data sources; machine learning algorithms and web-based application can be researched in the later development.

8.1 Future Work

Future work with sentiment analyses can be improved by developing solution that will handle Big Data in-memory processing as presented within Apache Spark

chapter. Cluster and parallel computations over nodes could enhance the operational algorithm effectively and produce fast real-time streaming services.

Approach in this project is limited to hardware capabilities. However, in the future development can take input streams from different social media or machine sensors that can track outcomes from smart cities applications.

Moreover, other machine learning algorithms for text classification can be implemented in future work. Comparing their accuracy can lead to enhanced results.

Pre-processing approach with data normalization can be extended with natural language processing on the higher level, which could optimize the sentiment classification.

Since, Twitter text classification has many challenges to optimize the streaming, different supervised and unsupervised machine learning algorithms can be implemented in order to recognize patterns between tweets and make data driven predictions.

As the main challenge with Twitter classification is the potential bias within tweets can influence the whole streaming. Hence, in the future work, more detailed research on how to deal with it is in place.

Concept in this project uses the same credentials for Twitter API application, but in the future development solution could run as web-based application where Twitter users authenticate with their personal Twitter accounts in order to use Twitter sentiment tool.

Implementing Twitter sentiment application online will require socket layers for persistent connection to the stream. Web Sockets implemented by Socket.io can serve for sessions and Node.js for handling stream. CherryPy or Django can deploy the server and application development. In addition, the graphical interface can be developed with Bootstrap for HTML, CSS and JavaScript.

Due to the time scope and hardware limitations of this project, these suggestions are not implemented in this project. However, the concept for sentiment analyses can be improved and developed in the future with broader research in this study field.

9 References

[1]"New user FAQs," *Twitter Help Center*. [Online]. Available: https://support.twitter.com/articles/13920?lang=en. [Accessed: 14-Feb-2016].

[2]"How Much Data Is Generated Every Minute On Social Media?," WeRSM | We Are Social Media. [Online]. Available: http://wersm.com/how-much-data-is-generated-every-minute-on-social-media/. [Accessed: 14-Feb-2016].

[3]P. Baker, "84% of enterprises say big data will completely reshape industries within a year," *FierceBigData*. [Online]. Available: http://www.fiercebigdata.com/story/84-enterprises-say-big-data-will-completely-reshape-industries-within-year/2014-11-03. [Accessed: 15-Feb-2016].

[4]B. Pang and Lillian Lee, "Opinion mining and sentiment analysis," [Online]. Available: http://www.cs.cornell.edu/home/llee/omsa/omsa.pdf. [Accessed: 21-Apr-2016].
[Accessed: 15-Feb-2016].

[5]"Company | About," *Twitter About*. [Online]. Available: https://about.twitter.com/company. [Accessed: 17-Feb-2016].

[6]J. Garside, "Twitter puts trillions of tweets up for sale to data miners," *The Guardian*. [Online]. Available: http://www.theguardian.com/technology/2015/mar/18/twitter-puts-trillions-tweets-for-sale-data-miners. [Accessed: 18-Feb-2016].

[7]"Using machine learning to predict gender." [Online]. Available: https://www.crowdflower.com/using-machine-learning-to-predict-gender. [Accessed: 20-Feb-2016].

[8]"The Twitter glossary," *Twitter Help Center*. [Online]. Available: https://support.twitter.com/articles/166337. [Accessed: 20-Feb-2016].

[9]M. A. Russell, "Mining the Social Web, Second Edition," Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472 in 2014, ISBN: 978-1-449-36761-9.

[10]"Using hashtags on Twitter," *Twitter Help Center*. [Online]. Available: https://support.twitter.com/articles/49309. [Accessed: 23-Feb-2016].

[11]"Twitter", Andy Murray. [Online]. Available: https://twitter.com/andy_murray/with_replies. [Accessed: 25-Feb-2016].

[12]"What Is Apache Hadoop?," Welcome to Apache Hadoop!. [Online]. Available: https://hadoop.apache.org. [Accessed: 27-Feb-2016].

[13]M. Adnan, M. Afzal, M. Aslam, R. Jan, M.-Enriquez A.M, “Minimizing Big Data Problems using Cloud Computing Based on Hadoop Architecture,” [Online]. Available: http://ieeexplore.ieee.org.zorac.aub.aau.dk/stamp/stamp.jsp?tp=&arnumber=7029370 [Accessed: 1-Mar-2016].

[14] J. Leskovec, A. Rajaraman, J. D. Ullman, “Mining of Massive Datasets,” [Online]. Available: http://infolab.stanford.edu/~ullman/mmds/book.pdf. [Accessed: 04-Mar- 2016].

[15]”YARN,”, The Architectural Center of Enterprise Hadoop, Hortonworks. [Online]. Available: http://hortonworks.com/hadoop/yarn/#section_2. [Accessed: 04-Mar-2016].

[16]”Apache Hadoop YARN”, Apache Hadoop 2.7.2, [Online]. Available: https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html. [Accessed: 04-Mar-2016].

[17]E. Dumbill, “What is Apache Hadoop? - O’Reilly Radar.” [Online]. Available: http://radar.oreilly.com/2012/02/what-is-apache-hadoop.html. [Accessed: 04-Mar-2016].

[18]”Apache Ambari,” Introduction,. [Online]. Available: https://ambari.apache.org. [Accessed: 05-Mar-2016].

[19]P. Warden, “Big Data Glossary,” O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, in 2011, ISBN: 978-1-449-31459-0.

[20]N. Pentreath, “Machine Learning with Spark,” Packt Publishing, Livery Place, 35 Livery Street, Birmingham B3 2PB, UK, in 2015, ISBN: 978-1-78328-851-9.

[21]M. Frampton, “Mastering Apache Spark,” Packt Publishing, Livery Place, 35 Livery Street, Birmingham B3 2PB, UK, in 2015, ISBN: 978-1-78398-714-6.

[22]”New Hadoop Survey Identifies Big Data Trends to Watch in 2016,” [Online]. Available: http://www.syncsort.com/en/About/News-Center/Press-Release/New-Hadoop-Survey-Identifies-Big-Data-Trends. [Accessed: 07-Mar-2016].

[23]”Apache Spark,” Lightning-fast cluster computing, [Online]. Available: http://spark.apache.org. [Accessed: 07-Mar-2016].

[24]”Apache Storm,” Why use Storm?, [Online]. Available: http://storm.apache.org. [Accessed: 08-Mar-2016].

[25]A. Toshniwal, S. Taneja, A. Shukla, K. Ramasamy, J. M. Patel, S. Kulkarni, J. Jasckson, K. Gade, M. Fu, J. Donham, N. Bhagat, S. Mittal, D. Ryaboy, ”Storm@twitter,” [Online]. Available: http://dl.acm.org/citation.cfm?id=2595641. [Accessed: 08-Mar-2016].

[26]J. Novet, ”Twitter details Heron, a real-time stream-processing system that outperforms Apache Storm,” [Online]. Available: http://venturebeat.com/2015/06/02/twitter-details-heron-a-real-time-stream-processing-system-that-outperforms-storm/. [Accessed: 08-Mar-2016].

[27]A. Avram, “Twitter Has Replaced Storm with Heron,” [Online]. Available: http://www.infoq.com/news/2015/06/twitter-storm-heron. [Accessed: 09-Mar-2016].

[28]”What does Aurora do?,” Apache Aurora, [Online]. Available: http://aurora.apache.org. [Accessed: 10-Mar-2016].

[29]K. Ramasamy, “Flying faster with Twitter Heron,” [Online]. Available: http://conferences.oreilly.com/strata/big-data-conference-ny-2015/public/schedule/detail/44632. [Accessed: 11-Mar-2016].

[30]S. Kulkarni, N. Bhagat, M. Fu, V. Kedigehalli, Ch. Kellogg, S. Mittal, J. M. Patel, K. Ramasamy, and S. Taneja. 2015. “Twitter Heron: Stream Processing at Scale,” [Online]. Available: http://dl.acm.org/citation.cfm?id=2742788. [Accessed: 11-Mar-2016].

[31]S. Haridi, ”Research Challenges in Data-Intensive Computing,” The stratosphere Project, Apache Flink, Ericsson [Online]. Available: https://www.sics.se/sites/default/files/pub/sh-stratosphere-format_16-10.v1_0.pdf. [Accessed: 12-Mar-2016].

[32]F. Beligianni, ”Streaming Predictive Analytics on Apache Flink,” [Online]. Available: http://www.diva-portal.se/smash/get/diva2:843219/FULLTEXT01.pdf;jsessionid=amPOFJ_4yWQESez47vK7vW9GOHMmr1bs8HgtZzWS.diva2-search7-vm. [Accessed: 11-Mar-2016].

[33]”FlinkML – Machine Learning for Flink,” Apache Flink. [Online]. Available: https://ci.apache.org/projects/flink/flink-docs-master/apis/batch/libs/ml/index.html. [Accessed: 11-Mar-2016].

[34]”Gelly: Flink Graph API,” Apache Flink. [Online]. Available: https://ci.apache.org/projects/flink/flink-docs-master/apis/batch/libs/gelly.html. [Accessed: 12-Mar-2016].

[35]J. Kreps, N. Narkhede, J. Rao, ”Kafka: a Distributed Messaging System for Log Processing,” [Online]. Available: http://research.microsoft.com/en-us/um/people/srikanth/netdb11/netdb11papers/netdb11-final12.pdf. [Accessed: 12-Mar-2016].

[36]I. Pointer, “Apache Flink: New Hadoop contender squares off against Spark,” *InfoWorld*, 07-May-2015. [Online]. Available: http://www.infoworld.com/article/2919602/hadoop/flink-hadoops-new-contender-for-mapreduce-spark.html. [Accessed: 13-Mar-2016].

[37]R. Metzger, ”Architecture of Flink’s Streaming Runtime,” dataArtisans [Online]. Available: http://events.linuxfoundation.org/sites/events/files/slides/ACEU15-FlinkArchv3.pdf. [Accessed: 13-Mar-2016].

[38]A. Hadoop, associated open source project names are trademarks of the A. S. F. F. a complete list of trademarks, and C. Here, “Components,” *Cloudera*. [Online]. Available: http://www.cloudera.com/. [Accessed: 14-Mar-2016].

[39]“Intel’s TAP Big Data Platform Gains Healthcare, Cloud Partners,” InformationWeek. [Online]. Available: http://www.informationweek.com/healthcare/analytics/intels-tap-big-data-platform-gains-healthcare-cloud-partners/d/d-id/1322456. [Accessed: 14-Mar-2016].

[40]P. Zadrozny, R. Kodali, “Big Data Analytics Using Splunk,” Expert’s Voice in Big Data, Published by Apress in 2013, Distributed by Springer, ISBN: 978-1-4302-5762-2

[41]”Hunk: Splunk Analytics for Hadoop and NoSQL Data Stores,” Splunk Product Data Sheet. [Online].Available: https://www.splunk.com/web_assets/pdfs/secure/Hunk_Product_Data_Sheet.pdf. [Accessed: 20-Mar-2016].

[42]J. Pokorný, “NoSQL Databases: a step to database scalability in Web environment,” Conference Paper in international Journal of Web Information Systems 9(1): 278-283. January 2011, DOI: 10.1145/2095536.2095583 [Online]. Available: https://www.researchgate.net/publication/221237715_NoSQL_Databases_a_step_to_database_scalability_in_Web_environment. [Accessed: 22-Mar-2016].

[43]T. O’Brien, “Big Data Projects: How to Choose NoSQL Databases,” *DATA SCIENCE REPORT - TODAY!*, 22-Jan-2015[Online]. Available: http://datasciencereport.com/2015/01/22/big-data-projects-how-to-choose-nosql-databases/#.Vv4fCGMppR2. [Accessed: 23-Mar-2016].

[44]R. Kune, P. K. Konugurthi, A. Agarwal, R. R. Chillarige and R. Buyya, “The Anatomy of Big Data Computing,” [Online]. Available: http://arxiv.org/pdf/1509.01331.pdf. [Accessed:23-Mar-2016].

[45]J. Han, M. Kamber, J. Pei, “Data Mining, Concepts and Techniques,” Third Edition, Morgan Kaufmann Publishers is an imprint of Elsevier. 225 Wyman Street, Waltham, MA 02451, USA, in 2012, ISBN: 978-0-12-381479-1.

[46]I. H. Witten, E. Frank, M. A. Hall, “Data Mining, Practical Machine Learning Tools and Techniques,” Third Edition, Morgan Kaufmann Publishers is an imprint of Elsevier. 30 Corporate Drive, Suite 400, Burlington, MA 01803, USA, in 2011, ISBN: 978-0-12-374856-0.

[47]P. Harrington, “Machine Learning in Action,” Special Sales Department, Manning Publications Co., 20 Baldwin Road, PO Box 261, Shelter Island, NY, 11964, in 2012,

ISBN: 9781617290183.

[48] J. Bollen, H. Mao, and X. Zeng, “Twitter mood predicts the stock market,” Journal of Computational Science, vol. 2, no. 1, pp. 1–8, Mar. 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S187775031100007X. [Accessed: 25-Mar-2016]

[49]K. L. Wagstaff, “Machine Learning that Matters.” Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109 USA, [Online]. Available: http://teamcore.usc.edu/WeeklySeminar/Aug31.pdf/. [Accessed: 28-Mar-2016].

[50]R. Yadav, “Spark Cookbook, Quick answers to common problems,” Published by Packt Publishing Ltd., Livery Place, 35 Livery Street, Birmingham B3 2PB, UK, in 2015, ISBN: 978-1-78398-706-1.

[51]N. Pentreath, “Machine Learning with Spark, Community Experience Distilled,” Published by Packt Publishing Ltd., Livery Place, 35 Livery Street, Birmingham B3 2PB, UK, in 2015, ISBN: 978-1-78328-851-9.

[52]V. Prajapati, “Big Data Analytics with R and Hadoop,” Packt Publishing, Livery Place, 35 Livery Street, Birmingham B3 2PB, UK in 2013, ISBN: 978-1-78216-328-2.

[53]A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah, and T. Herawan, “ Big Data Clustering: A Review,”In Computational Science and Its Applications – ICCSA 2014. Springer International Publishing, p. 707-720. 2014.

[54]N. Shi, X. Liu, Y. Guan,”Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm,” Conference Paper, April 2010. DOI: 10.1109/IITSI.2010.74, [Online]. Available: https://www.researchgate.net/publication/221600313_Research_on_k-means_Clustering_Algorithm_An_Improved_k-means_Clustering_Algorithm. [Accessed: 1-Apr-2016].

[55]A. A. L. Btissam Zerhari, “Big Data Clustering: Algorithms and Challenges,” 2015. [Online]. Available: https://www.researchgate.net/publication/276934256_Big_Data_Clustering_Algorithms_and_Challenges. [Accessed: 1-Apr-2016].

[56]M. Holub, “Big Data processing for Clickstream analysis,” CMI, ICTE, Aalborg University Copenhagen, Fall 2015

[57]T. K. Landauer, P. W. Foltz, D. Laham, “An Introduction to Latent Semantic Analysis,” [Online]. Available: http://lsa.colorado.edu/papers/dp1.LSAintro.pdf. [Accessed: 2-Apr-2016].

[58]S. Deerwester, S. T. Dumais, R. Harsham, “Indexing by Latent Semantic Analysis,” [Online]. Available: http://lsa.colorado.edu/papers/JASIS.lsi.90.pdf. [Accessed: 2-Apr-2016].

[59]B. Rosario,"Latent Semantics Indexing: An overview," INFOSYS 240 Spring 2000 Final Paper, [Online]. Available: http://www.cse.msu.edu/~cse960/Papers/LSI/LSI.pdf [Accessed: 3-Apr-2016].

[60]J. Lin and A. Kolcz, "Large-Scale Machine Learning at Twitter," [Online]. Available: http://www.umiacs.umd.edu/~jimmylin/publications/Lin_Kolcz_SIGMOD2012.pdf. [Accessed: 3-Apr-2016].

[61]Apache Thrift, "Getting Started Overview," [Online]. Available: https://thrift.apache.org. [Accessed: 3-Apr-2016].

[62]E. Kouloumpis, T. Wilson, J. Moore, "Twitter Sentiment Analysis: The Good the Bad and the OMG!," [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2857/3251. [Accessed: 4-Apr-2016].

[63]K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan and N. A. Smith, "Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments," [Online]. Available: http://www.cs.cmu.edu/~ark/TweetNLP/gimpel+etal.acl11.pdf. [Accessed: 5-Apr-2016].

[64]A. Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau, "Sentiment Analysis of Twitter Data," [Online]. Available: http://www.cs.columbia.edu/~julia/papers/Agarwaletal11.pdf. [Accessed: 6-Apr-2016].

[65]H. Saif, Y. He, M. Fernandez, and H. Alani, "Semantic Patterns for Sentiment Analysis of Twitter," [Online]. Available: http://iswc2014.semanticweb.org/raw.githubusercontent.com/lidingpku/iswc2014/master/paper/87970321-semantic-patterns-for-sentiment-analysis-of-twitter.pdf?raw=true. [Accessed: 6-Apr-2016].

[66]Saif,H.,Fernandez,M.,He,Y.,Alani,H.:Senticirclesforcontextualandconceptualsemantic sentiment analysis of twitter. In: Proc. 11th Extended Semantic Web Conf. (ESWC). Crete, Greece (2014)

[67]S. Asur, B. A. Huberman, "Predicting the Future With Social Media," [Online]. Available: http://www.hpl.hp.com/research/scl/papers/socialmedia/socialmedia.pdf. [Accessed: 7-Apr-2016].

[68]J. Bollen, H. Mao, Xiao-Jun Zeng, "Twitter mood predicts the stock market," [Online]. Available: http://arxiv.org/PS_cache/arxiv/pdf/1010/1010.3003v1.pdf. [Accessed: 7-Apr-2016].

[69]M. D. Conover, B. Goncalves, J. Ratkiewicz, A. Flammini, and F. Menczer, "Predicting the Political Alignment of Twitter Users," [Online]. Available:

http://cnets.indiana.edu/wp-content/uploads/conover_prediction_socialcom_pdfexpress_ok_version.pdf. [Accessed: 9-Apr-2016].

[70]"StockTwits message," StockTwits. [Online]. Available: http://stocktwits.com/. [Accessed: 9-Apr-2016].

[71]"StockTwits Mobile Applications," StockTwits. [Online]. Available: http://stocktwits.com/mobile. [Accessed: 9-Apr-2016].

[72]"Use Cases," CrowdFlower. [Online]. Available: http://www.crowdflower.com/use-cases/. [Accessed: 8-Apr-2016].

[73]"About Dataminr," Dataminr, [Online]. Available: http://www.dataminr.com/about/. [Accessed: 7-Apr-2016].

[74]Right Relevance, "Search for topical articles, influencers and conversations," [Online]. Available: http://www.rightrelevance.com. [Accessed: 8-Apr-2016].

[75]"#Tagboard on Tagboard," Tagboard. [Online]. Available: https://tagboard.com/Tagboard/231885/embed/grid. [Accessed: 9-Apr-2016].

[76]"Hashtag Tracking for Twitter, Instagram and Facebook - Keyhole." [Online]. Available: http://keyhole.co/social-analytics-features. [Accessed: 10-Apr-2016].

[77]"Real-time Information Discovery," Dataminr, [Online]. Available: http://www.dataminr.com/technology/. [Accessed: 10-Apr-2016].

[78]"Hashtag Tracking for Twitter, Instagram and Facebook - Keyhole." [Online]. Available: http://keyhole.co/preview. [Accessed: 10-Apr-2016].

[79]"Matplotlib," Python 2D plotting library, [Online]. Available: http://matplotlib.org. [Accessed: 15-Apr-2016].

[80]W. McKinney, "Python for Data Analyses," Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, in 2013, ISBN: 978-1-449-31979-3.

[81]"Project Jupyter." [Online]. Available: http://www.jupyter.org. [Accessed: 16-Apr-2016].

[82]"Tweepy, An easy-to-use Python library for accessing the Twitter API," [Online]. Available: http://www.tweepy.org. [Accessed: 20-Apr-2016].

[83]H. Kwak, Ch. Lee, H. Park, S. Moon, "What is Twitter, a Social Network or a News Media?. Department of Computer Science, KAIST. [Online]. Available: http://www.eecs.wsu.edu/~assefaw/CptS580-06/papers/2010-www-twitter.pdf. [Accessed: 21-Apr-2016]

[84]J. Perkins, ”Python 3 Text Processing with NLTK 3 Cookbook,” Published by Packt Publishing Ltd., Livery Place, 35 Livery Street, Birmingham B3 2PB, UK, in 2014, ISBN: 978-1-78216-785-3.

[85]N. Hardeniya, “NLTK Essentials,” Published by Packt Publishing Ltd., Livery Place, 35 Livery Street, Birmingham B3 2PB, UK, in 2015, ISBN: 978-1-78439-690-9.

[86]“The Streaming APIs,” *Twitter Developers*. [Online]. Available: https://dev.twitter.com/streaming/overview. [Accessed: 25-Apr-2016].

[87]”Movie Review Data,” Sentiment polarity datasets. [Online]. Available: http://www.cs.cornell.edu/people/pabo/movie-review-data/. [Accessed: 15-May-2016].

[88]T.M. Mitchell, M. Hill, ”Generative and Discriminative classifiers: Naïve Bayes and Logistic Regression,” [Online]. Available: https://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf. [Accessed: 14-May-2016]

[89]S. Wang, L. Jiang, Ch. Li, “Adapting naïve Bayes tree for text classification,” Knowledge and Information Systems, July 2015, Volume 44, Issue 1, pp 77-89

[90]R. F. Murray, K. Patel, A. Yee, “Posterior Probability Matching and Human Perceptual Decision Making,” Published: June 16, 2015. [Online]. Available: http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004342 [Accessed: 15-May-2016]

[91]L. P. Coelho, W. Richert, “Building Machine Learning Systems with Python,” Published by Packt Publishing Ltd., Livery Place, 35 Livery Street, Birmingham B3 2PB, UK, in 2015. ISBN: 978-1-78439-277-2.

[92] W. McKinney, “Python for Data Analyses,” Published by O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, in 2013, ISBN: 978-1-449-31979-3.

[93]”Spark Streaming,” Databricks, Stanford EDU, [Online]. Available: http://stanford.edu/~rezab/sparkclass/slides/td_streaming.pdf. [Accessed: 29-May-2016]

[94]”Realtime processing with Spark Streaming,” AmpCamp Berkeley EDU, [Online]. Available: http://ampcamp.berkeley.edu/big-data-mini-course/realtime-processing-with-spark-streaming.html. [Accessed: 29-May-2016].

[95]N. Pentreath, “Machine learning with Spark,” Published by Packt Publishing Ltd., Livery Place, 35 Livery Street, Birmingham B3 2PB, UK, in 2015. ISBN: 978-1-78328-851-9.

10 Appendix

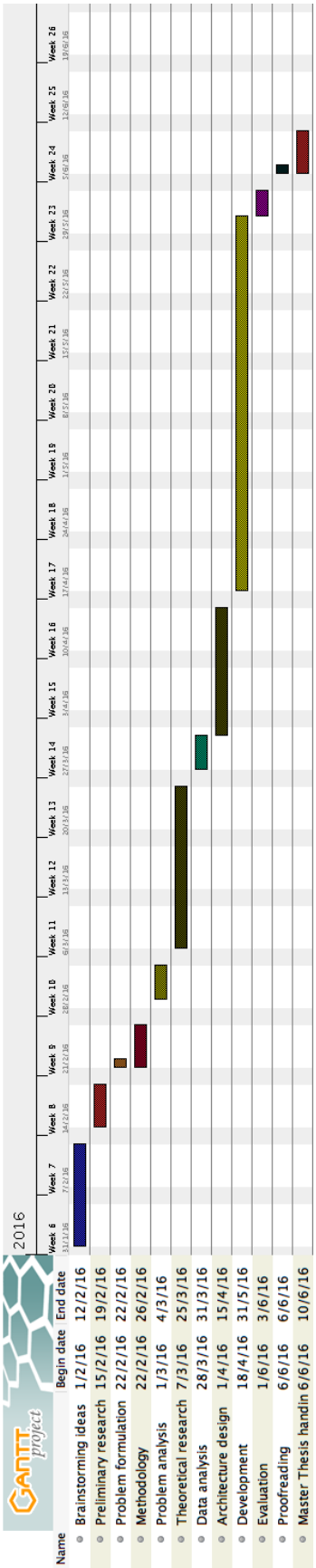


Figure 64: Gantt chart

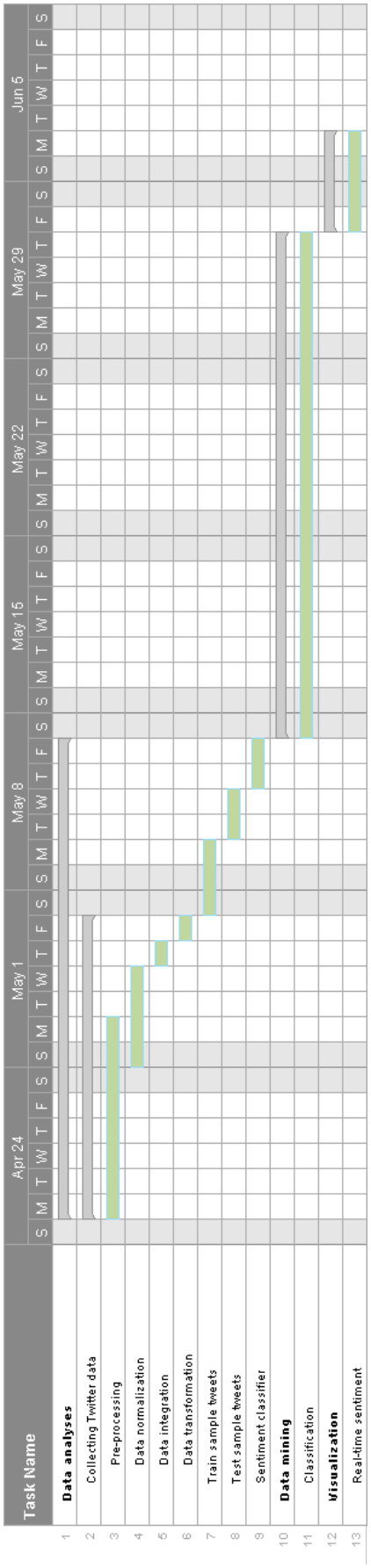


Figure 65: Development milestones