UNIVERSITY OF VAASA

FACULTY OF TECHNOLOGY

COMMUNICATIONS AND SYSTEMS ENGINEERING

Anders Hermans

MACHINE LEARNING – ANALYSIS AND UTILISATION OF PROJECT DATA

Master's thesis for the degree of Master of Science in Technology submitted for assessment, Vaasa, 20 December, 2019.

Supervisor Professor Mohammed Elmusrati

Instructor Rami Aihinen

FOREWORD

I would like to take this opportunity to thank my employer VEO Oy and my foreman Mats Warg for giving me the possibility to do my Master's thesis at VEO. I would also want to thank my instructor Rami Aihinen for his guidance and help throughout this thesis work.

Furthermore, I would like to thank my supervisor, Professor Mohammed Elmusrati, for his patience, guidance and feedback, not only during the progress of this thesis but also throughout the Master's program.

Last, but not least, I would also like to thank my wife and family for their support.

Kvevlax, Finland, December 2019
Anders Hermans

TABLE OF CONTENTS

ABBREVIATIONS

AI Artifical Intelligence

ALS Alternating least squares

API Application Programming Interfaces

BI Business Intelligence

BOM Bill of Material

CFO Chief Marketing Officer

CSV Comma-Separated Values

DBSCAN Density-Based Spatial Clustering of Applications with Noise

ECLAT Equivalence Class Clustering and Bottom-up Lattice Traversal

EDA Exploratory Data Analysis

EDW Enterprise Data Warehouses

GIGO Garbage In Garbage Out

IM Information Management

IP Internet Protocol

IT Information Technologies

KDD Knowledge Discovery from Data

k-NN K-Nearest Neighbor

LV Low Voltage

MV Medium Voltage

ODS Operational Data Store

OLAP Online Analytical Processing

OLTP Online Transaction Processing

OS Operating System

PC Personal Computer

PDF Probability Density Function

SDK Software Development Kits

SVD Singular Value Decomposition

LIST OF FIGURES

LIST OF TABLES

UNIVERSITY OF VAASA
Faculty of technology
Author: Anders Hermans
Thesis title: Machine Learning - Analysis and Utilisation of Project Data
Degree: Master of Science in Technology
Supervisor: Professor Mohammed Elmusrati
Evaluator: Professor Timo Mantere
Instructor: Rami Aihinen
Major of Subject: Communications and Systems Engineering
Year of graduation: 2020 **Number of pages:** 147

ABSTRACT:
Data mining, big data and machine learning are topics that have grown increasingly popular during the last decade. Declining hard disk drives prices and advancements in technology lets us store more data to a lower cost. The amount of data that is constantly generated is also increasing due to factors like the Internet, generation of metadata from data, and social media. This thesis has been done at VEO Oy as a case study on how data mining techniques or machine learning algorithms could provide insights that could help to improve manufacturing processes of the company. VEO Oy manufactures control panels, medium voltage and low voltage switchgear that are used in various industry and power generation applications. The main source of data used in this thesis has been generated during the design stages of projects and the data is stored in several databases, such as Access and SQL. The outcome of this thesis was a list of mined association rules between the parts that are listed in the company's bill of material. The mined association rules could be used for rearranging the warehouse in order to optimize the placement of the parts. Another suggestion is a recommendation system add-in for the design software used in the company that would suggest the next part to the design engineer based on previously added parts.

Keywords: Data Mining, Machine Learning, Association Rules, Supervised Learning, Unsupervised Learning, Project Data

1 INTRODUCTION

VEO Oy, founded in 1989 as Vaasa Engineering Oy, is specialized in developing drives, electrification and automation systems for industry customers and the main implementation areas of these products are diesel and gas power plants, process industries and ships. VEO Oy has business units in Finland, Sweden, Norway and the United-Kingdom. VEO Oy's headquarters and factory is located in Vaasa and the majority of the 400 employees works here.(VEO, 2019)

Products like medium voltage (MV), low voltage (LV) switchgear and control panels are assembled in the factory in Vaasa. In 2017, VEO Oy acquired I.C. Electrical Ltd. in the United-Kingdom, a company specialized in industrial electrical installations. Other services offered to customers by VEO Oy are testing of projects, user education and consulting. The business segments of VEO Oy are hydropower, wind power, industry, engine and hybrid power, thermal power and marine. (VEO, 2019)

1.1 Engine and hybrid power

The biggest customer of the engine and hybrid power business segment of VEO Oy is Wärtsilä. This business segment designs LV switchgear and control systems that are used in Wärtsilä diesel and gas power plants. In addition to the previously mentioned services the engine and hybrid power business segment offers commissioning services of power plants as well as spare parts for marine projects and engine and hybrid power plants. (VEO, 2019)

1.2 Other business segments

Hydropower builds turnkey projects as well as performs upgrades the existing equipment in older power plants. Hydropower designs and delivers electrification and automation

solutions and offer excitation and protection systems for generators. The hydropower business segment also offers project management which includes project supervision, consulting services and delivery. The consulting services comprises development and presentation of automation and system models, feasibility studies and cost estimates to the customer. (VEO, 2019)

The thermal power business segment designs and delivers automation and electrification solutions for combined heat and power plants. Turnkey projects include design and programming of automation system, delivery of control system automation panels, delivery of switchgear, project management and implementation, installation and testing, and training of the end user. (VEO, 2019)

1.3 Scope of this thesis

Data is generated during the whole life cycle of a project. Emails and specification documents are just some examples of data that is acquired from the customer and stored in VEO's storage system. However, the majority of the data is generated during the design stage of a project and is derived from different design software used in the company when designing products like control panels or LV switchgear.

The advancements in technology has entailed a decrease of storage prices. This trend has enabled companies, as well as as individuals, to increase their storage capabilities without investing a big amount of money. From a company's point of view, it means that it is possible to store more project related data and to store the data for a longer period of time rather than deleting it.

The scope of this thesis is to study the possibilities of implementing data mining techniques and machine learning algorithms utilising the data that has been generated during project design and testing. The fact that data is fragmented and stored in several databases can eventually bring some difficulties when gathering the data that is going to be used in

the data mining algorithms. The internal databases will act as the main source of data in this thesis. The employer is very interested to know what could be achieved with data mining and machine learning, but the employer does not have a specific problem that must be solved with data mining and machine learning. It is, however, the author's responsibility to research these topics and to find a suitable use case. The goal is to present a use case where machine learning algorithms or data mining techniques have provided insights, based on the project related data, that could lead to improvements within the company.

2 BIG DATA

2.1 Why now?

Big Data is a term and phenomena that has been trending during recent years and its priority is increasing in business organizations. The development and improvements of technology has enabled Big Data to get a foot hold in business organizations to improve productivity, innovation and competition. The Big Data revolution is a slowly growing trend that have been evolving during the last decade. Factors that have the most vital to why Big Data have become so popular the recent years are listed below. (Simon, 2013)

- The connected consumer
- The declining costs of hardware
- The increase of data science
- Internet, search engines and infonomics
- The platform economy
- Social media and other factors (Simon, 2013)

Figure 1 illustrates how the data volume generated by organizations have grown over time. Back in the 1990s the amount of information managed by organizations could be measured in terabytes. At this point, organizations used relational databases and data warehouses to handle, at that time, the big amount of enterprise information. It was common for organizations in 1990s to perform data analytics on structured data i.e., data that is suited to be organized into rows and columns. In the years that followed, there was a rapid increase in the number of different data sources, mostly productivity and publishing tools like content management repositories and networked attached storage system. The amount of information that organizations had to manage in the 2000s is estimated to be

in the petabyte scale. In the 2010s, the number of data sources that generated data in organizations had continued to grow and by this time, the amount of information could be measured in exabytes, see **Figure 1**. The applications that generate a high data volume will certainly provide the opportunity to implement new analytics and to discover insights among the huge data volume. Today, almost everyone and everything is leaving a digital footprint and data now comes from sources like smart devices, video surveillance, medical information, non-traditional IT devices and photos and videos uploaded to the internet. (Dietrich, 2015)

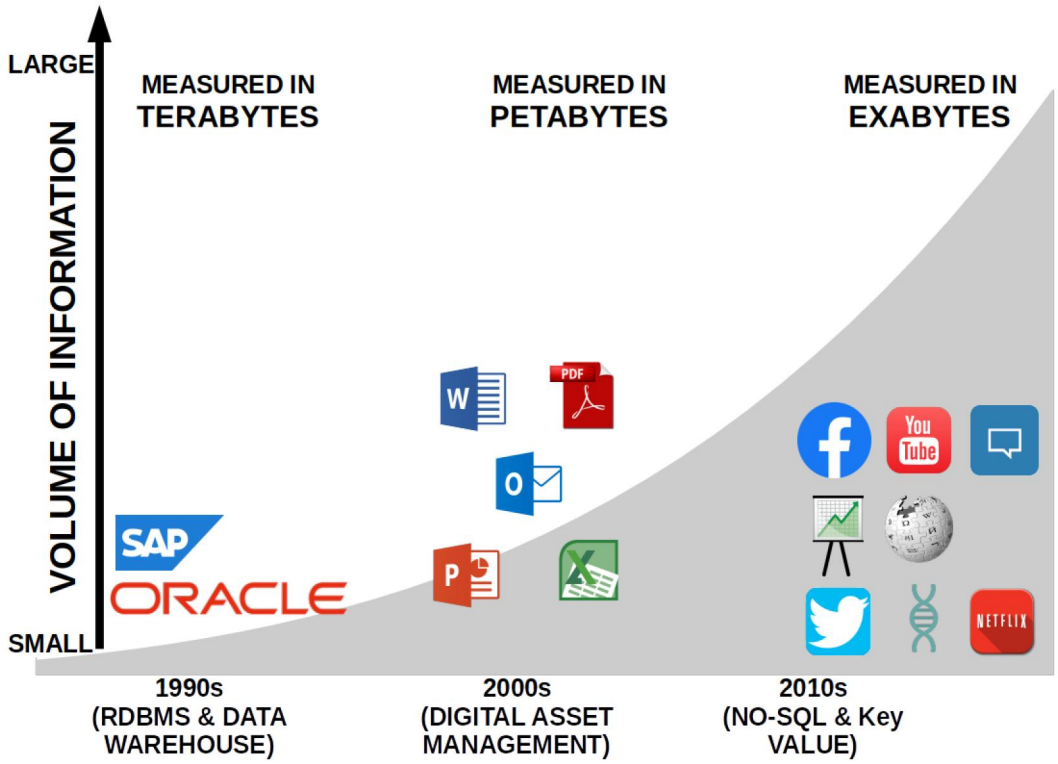

Figure 1: Big Data sources

2.1.1 The connected consumer

Big Data is considered to be consumer driven, it is utilising the data that the people are consuming and generating when using different online services. The Big Data trend got a foot hold when the technological improvements, such as, lower data storage costs, cell phones and later smartphones, cloud computing and broadband connections were avail-

able for the general public. (Simon, 2013)

2.1.2 The declining costs of hardware

Storage costs have been decreasing steadily thanks to technological advancements. As shown in **Figure 2**, the cost per gigabyte for storage were remarkably higher in 1990 compared to what it is today. Without the declining storage costs, one can argue that Big Data would not have been possible today. Consumers face no problems with staying connected to the web and other services, and by doing so, we are consuming and generating a large amount of data. Organizations and service providers are collecting and storing this data for further processing. (Simon, 2013)

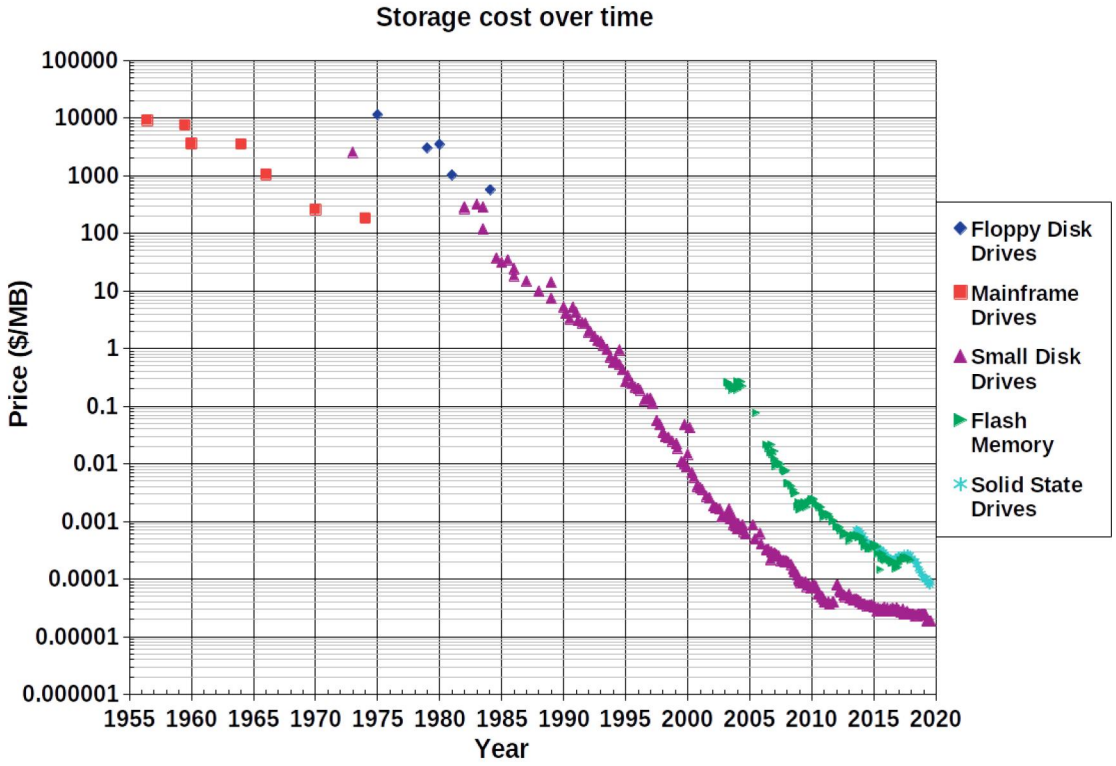

Figure 2: The price development of storage over time

2.1.3 Data science

Since Big Data has been on the rise during the latest years, it has also created a demand of practitioners that have an analytical skill to make effective decisions based on Big Data. This type of field of work is termed *Data science*. The field data science comprises a large set of fields, including data engineering, statistics, pattern recognition and learning, uncertainty modeling, data warehousing, statistics, math, and high-performance computing. (Simon, 2013)

From an industry perspective, a chief data scientist is responsible for engineering the infrastructure for collecting data, logging, and privacy concerns. The chief data scientist must decide how much of the collected and processed data will be user-facing and to what extent the organization can use data in decision making, and also how to implement it back into a product. It is also upon the chief data scientist to maintain the communication with the leadership within a company. (O'Neil & Schutt, 2014)

In general terms, a data scientist is a person who is able to find and extract useful information from data. She is also able to, with the help of tools and methods from statistics and machine learning, interpret the data. Utilising statistics and software engineering skills,the data scientist can preprocess the data and also understand the biases in the data. As mentioned before, the data scientist is trying to find meaningful insights and patterns in the data sets with the help of visualization and data sense. This particular way of working can also be referred to as exploratory data analysis. (O'Neil & Schutt, 2014)

2.1.4 Internet, search engines and infonomics

Infonomics is a term that have been coined by Douglas Laney. Laney made extensive research during the late 1990s on how valuable information is and how it is managed by companies. According to Laney, information certainly met the definition of a formal business asset and that it also should be treated as a formal business asset. After the 1990s,

companies gradually accepted that information as a business asset and Google has been one of the early adopters. Utilising software that runs on widespread and contextual data, Google has made great deal of money by displaying the right ads at the right time to its users. Google made it very obvious how important it can be for a company's success to understand its users and customers. (Simon, 2013)

2.1.5 The platform economy

Platforms have played a central role in the Big Data revolution. To mention a few, companies like Facebook, Amazon, Apple and Google have both contributed and benefited from Big Data. Apple made the software development kits (SDKs) and application programming interfaces (APIs) available to the public so that applications could be developed and published in their application store. Companies like Google, Microsoft and Facebook noticed how Apple's application store grew and they eventually made their own APIs and SDKs available to the public. These events enabled further growth of Big Data.(Simon, 2013)

2.1.6 Social media and other factors

Social media have had noticeable impact on Big Data. It cannot be unseen that Twitter, Facebook and LinkedIn, just to mention a few, is the driving force of the Big Data revolution. Additionally, advancements in RFID and sensor technologies also plays a part in the Big Data revolution. This will be discussed in more detailed in section 2.8. (Simon, 2013)

2.2 What is Big Data?

Reflecting back on **Section 2.1.2**, it is stated that storage cost is decreasing and the biggest changes have already happened. This makes it cheaper to store large quantities of data.

The volume of data and the number of data formats keeps on growing. Processing the data volume and the different formats makes up the fundamental problem of Big Data. The Big Data phenomenon has already existed for many years but it has escalated with the digitalization of our world, as **Figure 3** suggests. Many organizations are already utilising Business Intelligence (BI) tools but it is important to keep in mind that Big Data is not a tool that should replace BI. Organizations should strive to implement Big Data in parallel with the existing processing tools in order improve decision-making. Having the ability to store a lot of raw data comes with both opportunities and obstacles. One obstacle with the unprocessed data is that it can contain noise. A noisy dataset used as a training set can have negative impact on data model being built. But the opportunity with a high volume of raw data is the possibility to find new insights. Finding new insights would be harder if the data have been filtered and aggregated before storage. (Iafrate, 2015)

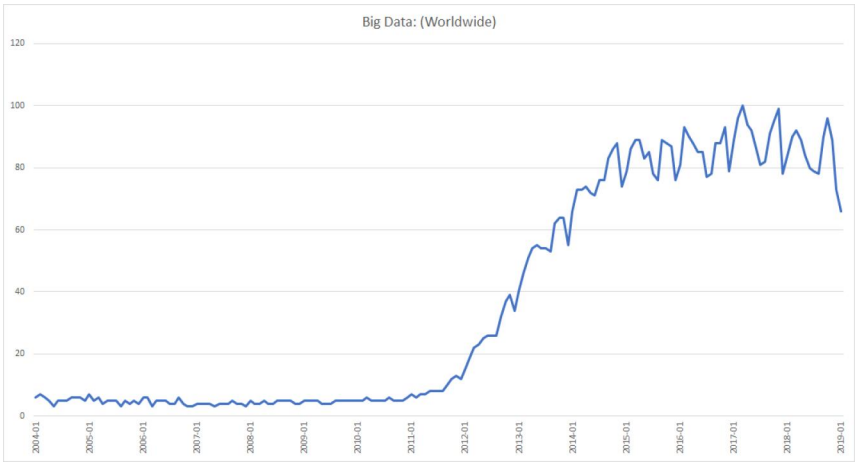

Figure 3: World wide Google search trend for the term Big Data (Google, 2019)

2.2.1 Characteristics of Big Data

It is hard to find a perfect definition of Big Data. Tech vendors are promoting their own definition of Big Data but there are three primary characteristics that were defined in 2001 by Douglas Laney. The primary characteristics are volume, variety and velocity. These three ”V’s”are still considered to be very accurate. In recent years, value has been

accepted as a new characteristic and added to the list and the author will discuss the four characteristics of Big Data in the following sections. (Simon, 2013)

2.2.2 Volume

In 2017, the number of internet users was 3.4 Billion and the number of connected devices about 18 Billion. Devices that are connected to the internet can be servers, personal computers (PCs), smartphones and tablets. Furthermore, the average traffic per user was 29 GB per month. By 2022, it is estimated that 4.8 Billion users and 24.8 Billion devices will be connected to the Internet. The average traffic per user is estimated to be 85 gigabyte per month by 2022. (Cisco, 2019)

The Internet of things (IoT) is indeed a key instrument in the growing number of connected peers. To mention a few devices, domestic appliances, security cameras and televisions are everyday devices that we already can connect to the Internet. The estimated amount of data that will be generated is more than 40000 exabytes, see **Figure 4**. To find value among the billions of events that occurs every minute, the events must be read and sorted. Data can be "reduced" by being sent through a storage, filtering, organization and analysis zone. (Iafrate, 2015)

2.2.3 Variety

Data have been selected for processing from transaction system because of its good structure for a long time. Decision-support and transaction databases have a different data model compared to other databases. A data model describes two things about a database, the way data is stored and the relationship between the data. Transaction data model prioritizes execution speed of writing, reading and modification actions in the database. It seeks to get the lowest possible response time. Having a database with low response time means that each transaction has a low execution duration, thus maximizing the number of parallel actions that can performed. (Iafrate, 2015)

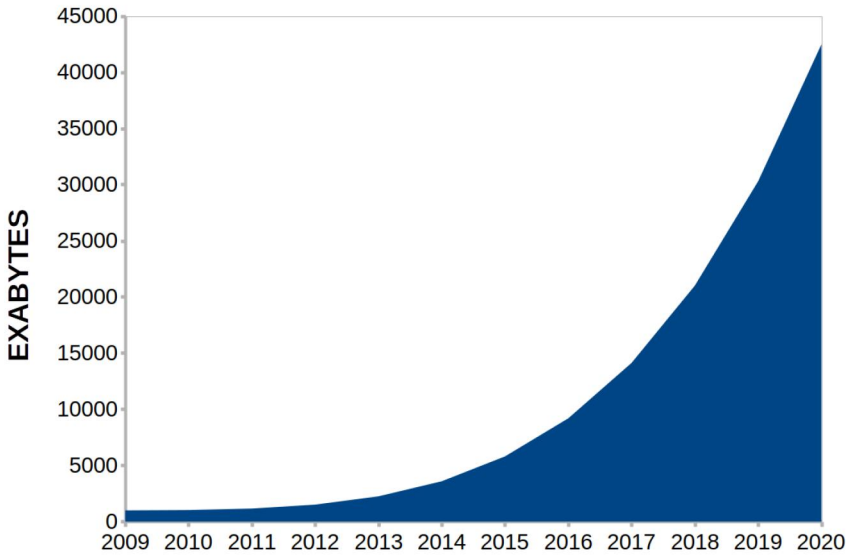

Figure 4: Research by the IDC on the evolution of digital data between 2010 and 2020

E-commerce sites must be able to handle several thousands of simultaneous transactions from customers that are browsing the site's product catalogue and prices. These kind of transactions uses very selective criteria, thus lowering the access requirement to historical data. This kind of data model is defined as a "normalized" data model, see **Figure 5**. A normalized data model stores e.g. client data in a separate structure and it stores data structures into entities. The advantage is that there is no data redundancy in a normalized data model, but the disadvantage is the complex relations and joints between the entities that requires a very good knowledge of the existing data model. The joint actions between the entities can easily become too complex to be implemented directly in BI solutions and analysis. To overcome this problem, a part of the data tables from the transactional database can be transferred to an operational reporting database. Operational reporting databases have a simpler data model compared to the normalized data model. Analysts are able to, without prior knowledge of the data model, analyze and create reports with the help of the simpler data model and BI-tools with metadata support. (Iafrate, 2015)

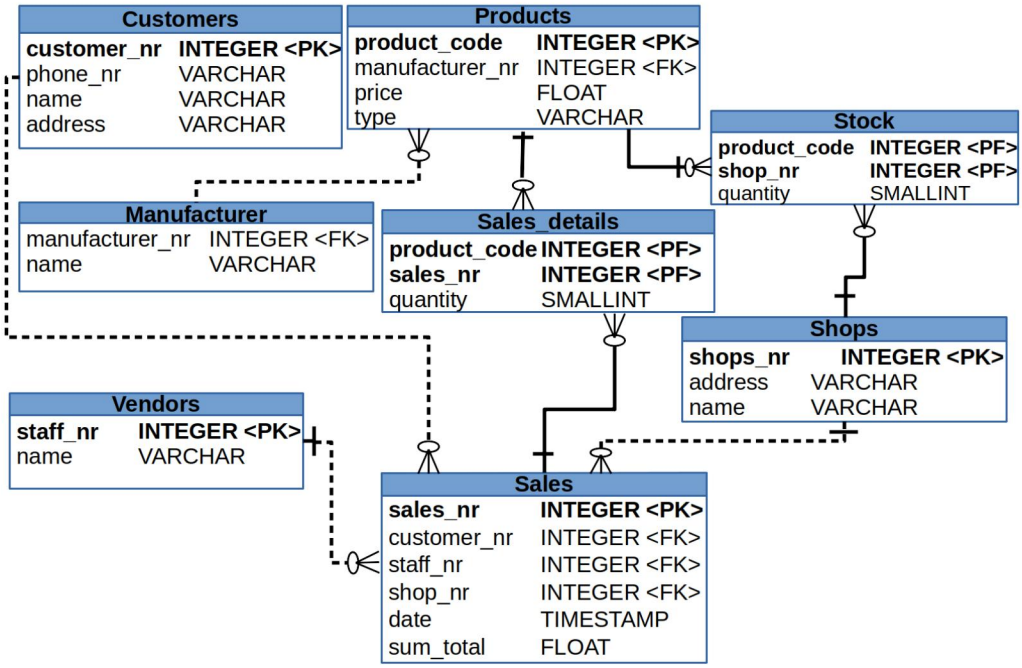

Figure 5: (Normalized) transaction data model

Decision data models are better suited for databases with a large volume of historical information, that is in the range of several years with a broad data access criterion. Joints and relations between entities associated with volume caused a noticeable performance impact on the execution time of queries in relation databases. Decision data model focuses on analysis, modeling and data mining. By implementing a denormalized data model one can mitigate the performance problem. Denormalized data models are also referred to as ”star” models due to the structure, consisting of sets of stars connected by their dimensions, see **Figure 6**. When the source data is stored in one structure containing all entities, it can be accessed from different analytical dimensions (time, product category, location etc.). Consider **Figure 6**, the fact table comprises all entities, such as, the product, the price, the invoice and the client. This storage method of information will cause data redundancy and therefore increasing the processing volume. (Iafrate, 2015)

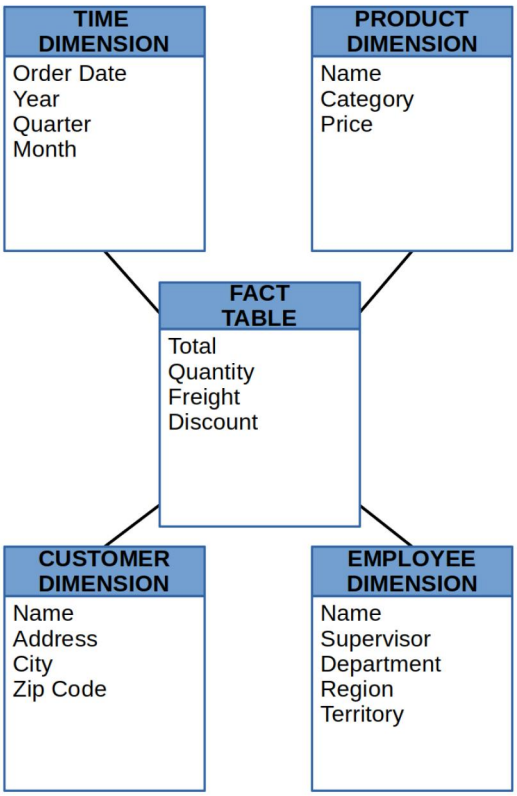

Figure 6: ”Star” data model

2.2.4 Velocity

Software agents, social networks, e-commerce sites, blogs etc., are generating a continuous flow of data on daily basis. The loyalty of a client is said to have changed and that a business only maintains a relationship with a client for as long as a client wants one. Considering that a client does not solely belongs to one brand, it is very important for businesses to act and react in order to meet the expectations of a client. A business must be able to offer content, products and prices in real time to the client. ”Real time” is best defined as, in this scenario, the amount of time that is associated with the duration of the client’s session. (Iafrate, 2015)

2.2.5 Value

Businesses are still focusing on using well-structured data, thus not making use of all the data available. The toughening competition, due to globalization and digitalization, makes it crucial to find the value in every piece of data. The insights gained will increase a business's situation awareness, this can be seen as an advantage over competitors. Big data must be seen as an additional source of information that will give assistance in a business's decision making process. As an example, in the year 2012, UN Global Pulse made a study in Indonesia on the correlation between the sale price of rice and the number of tweets about the high price of rice. From **Figure 7** it can be seen that the price of the rice itself correlates with the number of tweets about the price, hence it can be assumed that the tweets are linked to purchase. This kind of real-time information can be used to buy rice at lower price, giving some buyers an advantage over buyers without this real-time information. (Iafrate, 2015)

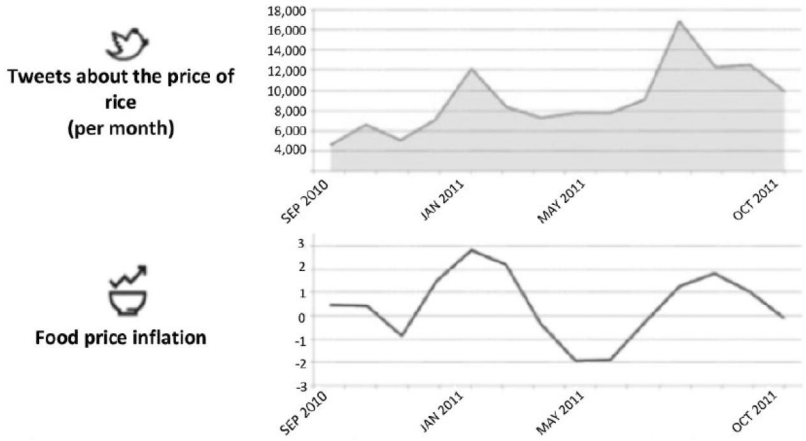

Figure 7: Comparison of the sales price of rice and tweets about the price of the rice (Iafrate, 2015)

2.3 Data collection, sampling, and preprocessing

The key component in analytical projects is data. Data can come in various forms. The first step when starting the analysis is to identify the types of data sources that are available and of interest. A general rule of thumb is the more data, the better. One problem

that arises along with this statement is that real life data can often be considered to be chaotic. If data is not preprocessed at all, and creating an analytical model based on the raw data will yield chaotic analytical models. This principal is called garbage in garbage out (GIGO). The inconsistencies, incompleteness, duplication, and merging problems are the main factors behind chaotic data. Analytical modeling steps comprises data filtering methods that will clean up and reorganize the available data to a more manageable and relevant size. To maintain the usability of data for further analysis, it is very important that each data preprocessing step is justified, carried out, validated and documented. This kind of verification will ensure that the data is usable throughout the analytical process. (Baesens, 2014)

2.4 Types of data sources

Transactions are the first important data source. The core purpose of transactional data is to represent the key information of a customer transaction, made up of structured, low-level and detailed information. A transaction can be e.g., online purchases, payments, cash transfers. Transactional data is often stored in online transaction processing (OLTP) relational databases. It is common that transactional data is aggregated into minimum/maximum, averages, summation, absolute/relative trends over a longer time period. (Baesens, 2014)

According to estimations, unstructured or semi-unstructured data holds 80% business-relevant information. Unstructured or semi-unstructured data comprises data that is not possible to group and store into columns, rows and fields. It is often text-based data, such as text files and emails, but can also be videos and photos. (Marr, 2015)

Unstructured data is not excluded from being used in Big Data analytics, on the contrary, unstructured data that is embedded in e.g., emails, web pages or multimedia can contain vital insights. However, before the unstructured data can be included in analytical tasks, it must undergo extensive preprocessing. (Baesens, 2014)

Expert-based data is seen as an important data source, it is also called qualitative data. What defines an expert? It is a person that has acquired a great deal of knowledge or skill within a particular area during his career. The knowledge originates from common sense and business experience. Before running any analytics on the gathered data, it is recommended to use the gathered knowledge of the expert as much as possible in order to guide the data modelling in the right direction directly from the start. Additionally, an expert helps with the interpretation of the analytical results. There are several good examples of applying expert-based validation, but a simple and popular example is checking the univariate signs of a regression model. It can be inferred that higher debt will have a negative impact on the credit risk in such a manner that it will show negative signs on the final scoreboard. If an analytical model would not indicate the same due to bad data quality, it would be very likely that the expert would disregard the analytical model. The choice not to use an analytical model is based on that it contradicts prior expectations. (Baesens, 2014)

The unstructured data have been increasing in amount and it is a very common data source within organizations. Even though organizations have access to a very big amount of unstructured or semi-structured data, the information is not processed and used for analytical tasks to gain insights, it is completely ignored. The main reason for ignoring the unstructured or semi-structured data is because most organizations are struggling with managing their transactional and structured data. (Simon, 2013)

The main problems comprise the lack of master data, poor data quality and integrity, and no semblance of data governance. Information management (IM) tend have a problem with employees not considering how their actions will affect others involved. A big issue with organizational data management is when the big picture is not taken into account, and instead, data is managed to only serve a specific application. This kind of management will, of course, benefit the functionality of a single application but it will certainly not benefit and support the needs of the whole enterprise. The growing number of data sources, data volume and variety of data, will certainly not make the data management process

easier unless it is agreed upon to implement a organizational data management strategy instead of application strategy. From a single department's perspective, it may only seem logical to prioritize its own data management and application needs. When each department follow this approach, it will result in organizational dysfunction. Poor data management will affect the employee's efficiency to answer common business-related questions. **Table 1** represents a generalized overview of questions, grouped by departments, that employees might spend more time on than necessary trying to answer. How fast the answer can be found is dependent on how the company have handled its data management strategy and how it has implemented information-based decision making. (Simon, 2013)

Table 1: Simplified overview of department specific questions (Simon, 2013)

Department	**Common business questions**
HR	Number of employees? What skills do each employee possess? Is there a need for courses to learn new skills?
Payroll	Are the wages correctly adjusted?
Finance and Accounting	Which departments are exceeding their budgets?
Sales	What is the total number of products the organization sell? Which specific product is more popular than others? Who are our customers? How many customers from a specific area have bought something within a specific time period?
Supply Chain	What is the status of the current inventory levels of key products? What is the estimated time before they can be replenished? Will the inventory level meet the current and future demand?
Marketing	What is the company's market share and has it changed compared to last year or last quarter?

According to (Simon, 2013), the case for organization that have very good data management systems will also have employees that is more efficient at their work. Employees does not need to waste time on manipulating the organizational data but can instead focus on answering more broader organizational questions. For example, the head of HR can focus on developing an effective succession plan since a well-planned data management system will easily provide the answer to what skills each and every employee of the company possesses. Another example is that marketing spend can be optimized by the

chief marketing officer (CMO) when there is good data management system handling the customer and sales data.

Social media and other online media platforms are also vital data sources that contains useful information that can be utilized to answer organizational questions of high value, for example:

- What opinion does our customer have towards the organization's customer services?
- When is the best time of the year to launch a new product?
- What is the general opinion about the latest commercial?
- What is the general opinion of the brand? (Simon, 2013)

In the year 2012, it was merely half of the companies that were subjects to a survey performed by IBM that collected and analysed data from social media platforms. Some speculations to why there were so few companies that used this information was that data that originates from social media was potentially less valuable data and a waste of time to analyse. The early adopters, on the other hand, that understood the great potential of Big Data faced another issue, they lacked the necessary tools to effectively handle Big Data and to make use of it. Standard reports, ad hoc queries, data warehouses and BI applications may not be adequate tools to handle the unstructured data since they were not designed to store, process and retrieve this type of data. (Simon, 2013)

The opportunities that are associated with a Big Data strategy is that it cannot only provide an organization with insights into known existing problems but also address unforeseen problems. Big Data can, based on data and information, help to identify trends, issues, and opportunities that human beings easily fail to identify. Organizations with a Big Data strategy will have a better understanding of the past, i.e. what has happened and why it

happened. Organizations will also have a better understanding of the present, i.e. what is happening and why it is happening. Additionally, organizations will also have a better understanding of the future. i.e. what will happen and why. (Simon, 2013)

2.5 Analyst perspective on data repositories

When organizations started to use spreadsheet programs, it was easy for users to create their own analyses of business problems. Spreadsheets does not require a database administration training and are configurable to perform tasks quickly and independently of IT groups. Spreadsheets are also easily distributed to other users and also, in most of the cases, the logic behind the analytics is editable by the end users. The ease of distribution creates a problem by itself, and that is how it can be assured that every user involved in solving a problem have the latest reversion of a spreadsheet, containing the most recent data and logic. However, with these problems emerges the need of a centralized data system. (Dietrich, 2015)

As the number of data sources and data needs increased, also increased the number of centralized data warehousing solutions. Organizations that implemented a centralized data warehousing solution could manage data centrally, gained the benefits of security, hardware redundancy, and users could rely on getting the latest official source of data for critical tasks. The centralized data warehousing solutions gave rise to the creation of Online Analytical Processing (OLAP) cubes and BI analytical tools. Enterprise Data Warehouses (EDWs) are essential for reporting and BI. EDWs, in combination with a solid BI strategy, enables access to data feeds that originates from data sources that are backed up and secured. (Dietrich, 2015)

An international company that consists of several organizational branches in different regions, or parts of the world, may have organized its data in such a way that each region have its own database. Providing analysis that is dependent on a fragmented data structure would be complicated but having a data warehouse that transfers the data from all regions,

and at the same time transforms the data, will facilitate the analysis. Before importing the data to the warehouse, it must undergo cleaning, integration, transformation and data loading, and refreshing. The data warehouse is a centralized storage place with unified data schema for data and information from different sources. **Figure 8** illustrates how the data warehouse is constructed and the usage of its data. (Han Jiawei, 2012)

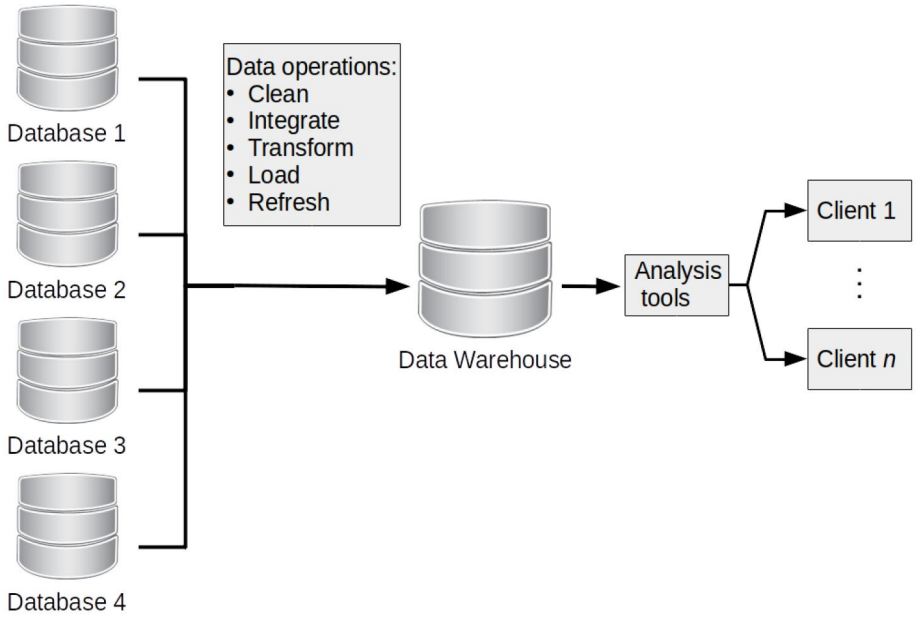

Figure 8: Data warehouse framework

Data that resides in data warehouses is commonly structured around key subjects, such as, item, activity, and supplier and this type of structure will help the information-based decision making. A data cube, a multidimensional data structure, is used to model a data warehouse and each dimension of the data cube is related to an attribute or a set of attributes. Furthermore, each cell in the data cube contains an aggregated value, the pre-computation provides fast access to summarized data. These two features, multidimensional data views and pre-computation of summarized data, enables support for OLAP. Drill-down and roll-up are two examples of OLAP operations that will help users to view the data at different levels of abstraction since OLAP operations benefit from background

knowledge regarding the domain of the data. As seen in **Figure 9**, it is possible for a user to examine the data available in the data cube by drilling down on e.g. sales data that have been summarized by quarter to see figures summarized by month. The roll-up operation will summarize sales data for cities to create a data view that presents sales data summarized by country. (Han Jiawei, 2012)

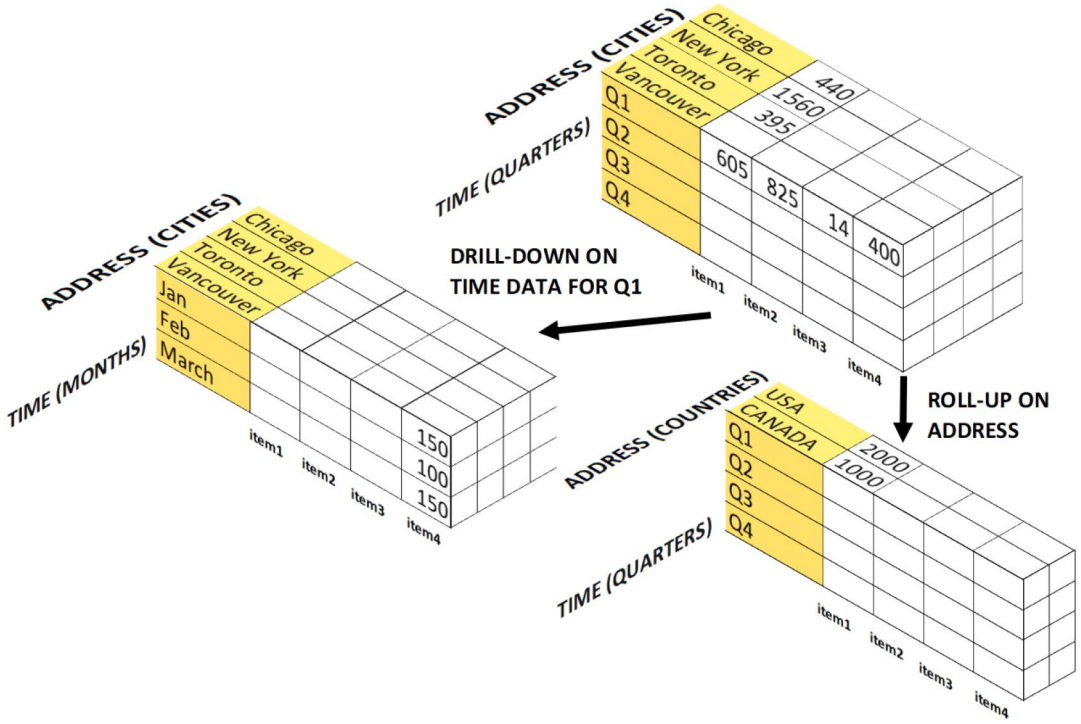

Figure 9: Multidimensional datacube used in data warehouses

In the EDW model, it is normally IT groups and database administrators (DBAs) that manage the data thus making the data analyst dependent on the IT groups in order to access data and data schema. This will, obviously, affect the lead time negatively since the data analyst must wait for appropriate approvals. Additional datasets, critical to the analytical process, are created and they are often managed by local power users. EDW systems have rules that restricts analysts from creating and managing datasets since IT groups generally dislike the idea of not having full control over the additional data sources. Furthermore, another negative aspect is that the additional datasets are not managed, secured or backed up. The positive aspect of EDW and BI, from an analyst's point of view, is that problems

related to data accuracy is solved, but the negative aspect is that it also induces problems related to flexibility and agility. (Dietrich, 2015)

Implementing analytical sandboxes, also referred to as *workspaces*, is a solution to the problems that emerges with EDWs and managed corporate data for analysts and data scientists. The analytical sandbox model gives IT groups the opportunity to manage and secure the analytical sandboxes, but they are at the same time designed to support robust analytics. The datasets in the analytical sandboxes are not commonly used as a data source for enterprise BI, thus giving the analytical teams the freedom of exploring the datasets. The analytical sandboxes commonly deliver high-performance computing using in-database processing. This is possible when analytics are done in the database rather than exporting the data and performing the analytical tasks in tools installed on another system. A significant benefit with in-database processing used for deep analytics is faster developing and testing of new analytical models while at the same decreasing the cost related to storing data on a local file system. Another great benefit with analytical sandboxes is the that it supports both structured and unstructured data without interfering with databases that are critical in the production process. **Table 2** contains an overview data repositories. (Dietrich, 2015)

Table 2: Types of Data Repositories from Analyst Perspective (Dietrich, 2015)

Data Repository	**Characteristics**
Spreadsheets	Spreadsheets and low-volume databases for recordkeeping Analyst depends on data extracts
Data Warehouses	Centralized data containers in a purpose-built space Supports BI and reporting but restricts robust analysis Analyst dependent on IT and DBAs for data access and schema changes Analysts will have to reserve a lot of time to collect aggregated and disaggregated data extracts from several different sources
Analytic Sandbox	Data assets, used for analysis, that originates from several different sources and technologies Makes flexible and high-performance analysis in non-production environment possible Can leverage in-database processing Reduces costs and risks associated with data replication into "shadow" file systems "Analyst owned" instead of "DBA owned"

2.6 Emerging Big Data ecosystem

Organizations have realized that there is an economical value in collected data that have been generated by users accessing their services. The four main groups that forms the backbone of the ecosystem are shown in **Figure 10**. The main groups are interacting with each other by communicating over the Internet. (Dietrich, 2015)

Label (1) in **Figure 10** is the data devices. To data devices and IoT group belongs e.g. smartphones, computers, card readers, etc. These devices are collecting data and are also generating new data about the collected data, i.e. metadata. An estimation is that for every gigabyte of collected data, an additional petabyte of metadata will be generated. To clarify this statement, consider a smartphone. Smartphones can be considered as a rich data source because, besides the data generated from basic phone usage, the phone itself collects and transmit information about Internet usage, SMS usage and real time location. Collecting the real time location and analyzing traffic patterns helps to estimate the traffic

congestion. By sharing this information to other GPS devices, it is possible to offer other routes to its users. (Dietrich, 2015)

Label (2) in **Figure 10** are the organizations that collects data from service users and devices. To this group belongs retail stores, websites and government agencies. Cable TV providers can e.g. create a profile of a user based on the content and channels the users tends to watch. From the collected data, the cable TV provider can tell which channels an user is willing to pay for in order to watch on demand content, furthermore, the cable TV provider is also capable of estimating a price for premium TV content that an user is willing to pay. (Dietrich, 2015)

Label (3) in **Figure 10** illustrates the data aggregators. Data aggregators compiles the data that originates from IoT data group and the usage patterns that are collected by organizations. The transformed data is sold to list brokers that will use this information target offers and campaigns to the right group of people. (Dietrich, 2015)

Label (4) in **Figure 10** illustrates the data users and buyers. To this group belongs those who benefits from the data that is sold by the data aggregators. For example, banks can buy data from data aggregators to estimate which customers that have the highest probability score to apply for additional mortgage or home equity line of credit. Independent variables, e.g., demographic area where customer lives, level of debt, credit score and customers searching the internet for topics on paying off debts, can be used as input for building a machine learning model that will calculate the probability score for a customer submitting a debt application. Thanks to the data available through data aggregators and high-performing techniques, the marketing campaigns are more likely to be better targeted compared to the time before age of Big Data. (Dietrich, 2015)

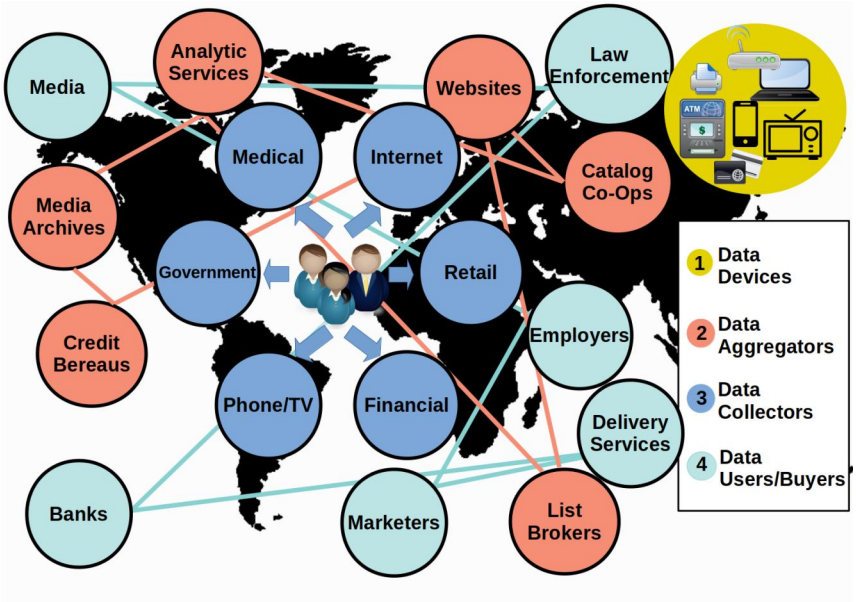

Figure 10: Big Data ecosystem

2.7 Organizational implementation

An organization seeking to incorporate Big Data into its business strategy will gain driving competitive differentiation if it succeeds. Organizations want to invest in such technology that will enhance its competitive differentiation. An organization must be able to identify the key business initiatives, identify as many data sources as possible and think about how to implement the data sources in their Big Data strategy. The biggest challenge for companies striving to implement a Big Data strategy is often the identification of how and where to start. (Schmarzo, 2016)

2.8 Big Data solutions

NEST

Nest is company that is developing IoT devices for our homes. Their main product line consists of smart thermostats, security cameras, smoke and carbon monoxide detectors and was acquired by Google in 2013. The problem that Nest has solved by implementing

Big Data is energy wasted in houses with inefficient heating systems. An ordinary thermostat is switching on or off based on time settings or temperature set-points. Since we have some sort of activity patterns in our daily lives this type of control can be regarded as inefficient. Optimizing heating systems will also benefit energy companies by distributing the load, ensuring that sufficient supply is available during peak use. Nest is using Big Data in thermostats so it can "learn" the most efficient strategy in order to maintain the temperature in your home. Nest thermostat is fitted with a motion sensor, temperature and a humidity sensor that collects data and information to learn when a person is home or away in order to create the best heating profile for the that particular home. The user's temperature set-points are also included in the heating profile along with the time it takes for an individual heating system to heat the home to a required temperature, enabling a more efficient system. (Marr, 2015)

Nest has developed their own proprietary operating system (OS) derived from the open-source OS Linux and other open-source technologies. Nest has also developed a communication protocol that is used between Nest devices independently of the wireless network infrastructure that exists in the intended user environment. Furthermore, Nest have developed a protocol called *works with Nest* that is intended as an interface for third-party IoT devices, such as, fitness trackers, washing machines and smart wall plugs. (Marr, 2015)

3 DATA SCIENCE

The core process of data science can be generalized to what is seen in **Figure 11**. With the help of mathematics, statistics, machine learning and artificial intelligence (AI) it is possible discover hidden information from data. (Williams, 2017)

Whatever the goal is, the first step is always to collect raw data. The data can be both structured and unstructured data, e.g. logs, employee records, emails, or fault reports. In the processing part, the collected data is processed using pipelines to join and scrape the data. The tools used in the processing part can be Python, R, SQL or shell scripting to get the collected data into a usable format. After the processing part, it is necessary to perform exploratory data analysis in order to find exceptions in the dataset, such as, missing values, outliers or incorrectly logged data. Depending on the results of exploratory data analysis, it might be necessary to collect more input data or put more effort in processing the available data. (O'Neil & Schutt, 2014)

Models can be constructed by using algorithms e.g. k-nearest neighbor, linear regression, Naive Bayes, decision trees or some other algorithm that is suitable for the initial problem. The outputs from the model can be interpreted and summarized into a report in which the insights are presented. Later on, the report can be distributed to key persons within the organization to help with decision making or it can also be published. The final step of the data science process is implementing the data product and creating a feedback loop. This means that the data product, based on the model, is made available for other users to interact with. (O'Neil & Schutt, 2014)

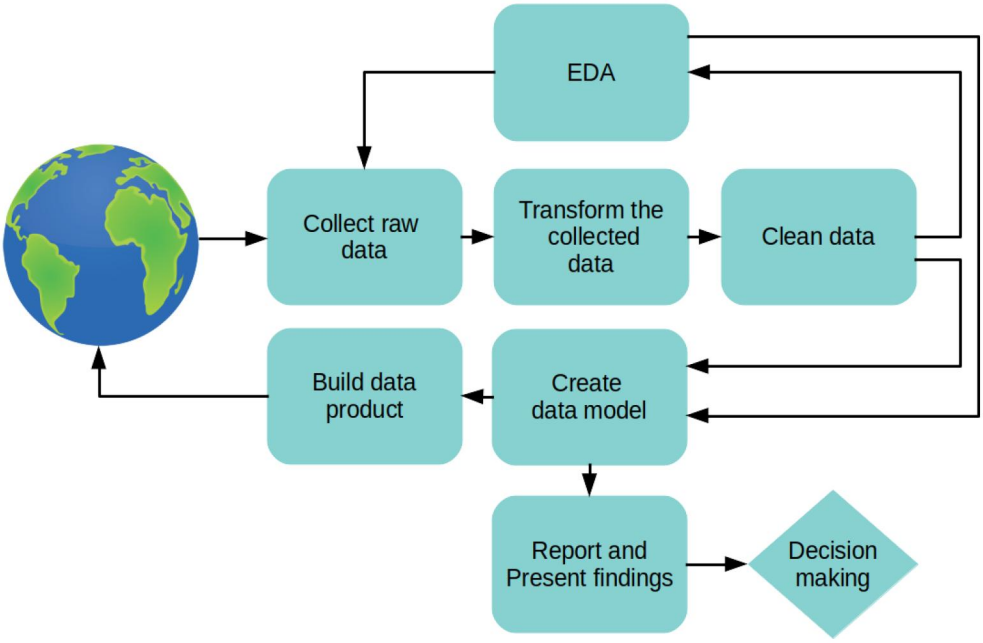

Figure 11: Data science process

Programming languages like R, Python, and Julia are considered to be the most commonly used languages in the data science field. Python is a general-purpose, high-level programming language. Python has very large user base where many individuals have scientific background. R language and programming environment is a free software that has its roots in the statistical community. (Williams, 2017)

3.1 Business intelligence versus data science

Table 3 contains four categories of common business problems that organizations are constantly struggling to surmount. The three first items presented in **Table 3** are not problems that have emerged suddenly the past few years, on the opposite, businesses have been trying to increase sales, reduce customer churn and cross-sale customers for a long time. In other words, it is not the business problem itself that is interesting, instead, it is the opportunity to solve the traditional business problems and gain additional information

from the analysis with the use of Big Data and advanced analytical methods. The last item in **Table 3**, on the other hand, requires advanced analytical techniques to be managed properly and complied with. Laws concerning anti-money laundering, and laws in general, are constantly being updated with additional requirements, thus increasing the complexity and data requirements for organizations. All of the four business drivers that are listed in **Table 3** are subjects to advanced analytical methods in order to be properly addressed. (Dietrich, 2015)

Table 3: Business drivers for advanced analytics (Dietrich, 2015)

Business Driver	**Examples**
Optimize business operations	Sales, pricing, profitability, efficiency
Identify business risk	Customer churn, fraud, default
Predict new business opportunities	Upsell, cross-sell, best new customer prospects
Comply with laws or regulatory requirements	Anti-Money laundering, fair lending

It is important to discuss the difference between BI and data science and **Figure 12** illustrates a few ways to compare these analytical techniques. The x-axis in **Figure 12** represents the time while y-axis represents the type of analysis being performed. From **Figure 12** we can see that BI provides standard and ad hoc reports, dashboards and queries for current time or from the past. By aggregating and grouping the historical data, BI can explain current or past behavior and answer questions related to "when" and "where" events occurred but is dependent on highly structured data. BI greatly facilitates answering questions related to quarterly reports, such as, quarter-to-date revenue, progress towards quarterly targets and to give information on the number of sold units for a specific product in a specific quarter or year. Generally, BI provides mostly hindsights but also some insights to organizational questions. Data science relies more on disaggregated data and prioritizes to analyse the events happening real-time and summarize their characteristics in order to enable information based decision making about the future, as seen in **Figure 12**. Exploratory data analysis can e.g., provide more accurate data forecasting than extending simple trend-lines. Data science tends to focus more on questions like "why" and "how" events occur. As discussed in **Section 2.4**, there are lot of different data sources available and data science tend to use a great number of those. BI relies heavily

on highly structured data that is organized into columns and rows. (Dietrich, 2015)

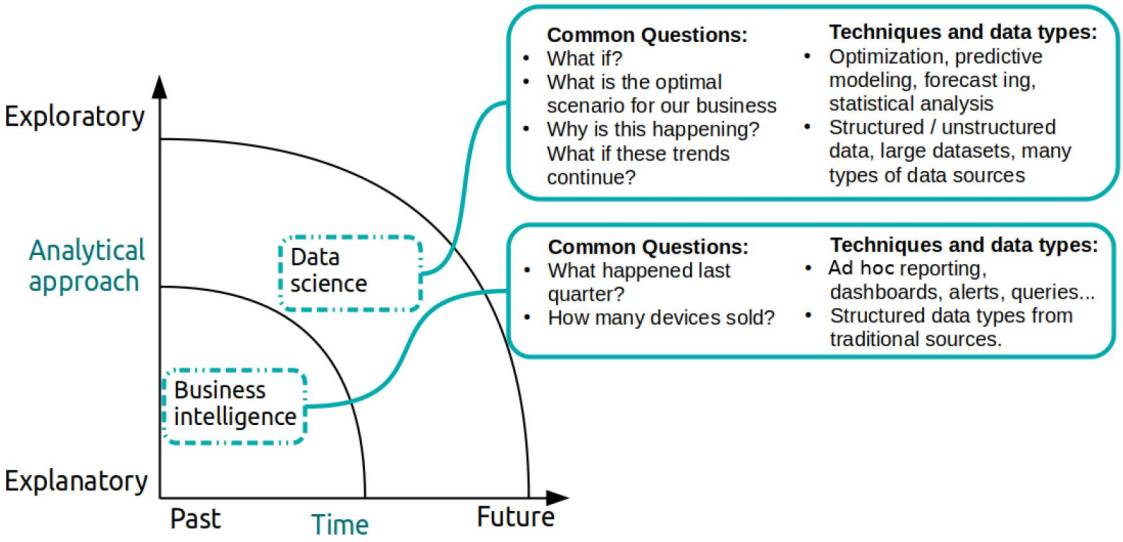

Figure 12: Comparison between BI and Data Science

3.2 Exploratory data analysis

A necessary step before applying more advanced analytical analysis is to use visual data exploration. Even though it is considered to be an informal way to start the analysis, visual data exploration will provide initial insights that can be useful throughout the modeling process. Common tools that are used in this process are plots, graphs and summary statistics. Pie charts visualizes a variable's distribution as a pie, where a section of the pie represents the total percentage taken by each value of the variable. Making separate pie charts from different perspectives can reveal insights that can be very useful at the beginning of the analysis. Bar charts visualizes the frequency of values, either absolute or relative, as bars. Initial analysis with histograms provides the general tendency of the data and can determine the variability and spread of the data. Using a histogram, the analyst can compare the observed data with known standard distributions, e.g., normal distribution. (Baesens, 2014)

The main idea behind the exploratory data analysis (EDA) is to help the analyst create an understanding of the connection between the source generating the data and the actual

data. In order to do so, it is indeed necessary to systematically examine the data from different point of views with the help of graphs, plotting time series, examining the relationship between variables, and producing summary statistics. When EDA is performed on the available data, it can be possible to outline at which points the dataset contains null values that have to be dealt with accordingly. Additionally, it works as a sanity check that shows if the data is of expected scale and format. (O'Neil & Schutt, 2014)

With the help of the programming language R, we can create an example that will show the intuition gained from EDA. The code snippet shown in **Figure 13** will create a dataset consisting of two variables, *x* and *y*. Both variables have 50 observations with normal distribution. (Dietrich, 2015)

```r
#Generation of random observations with normal distribution
x <- rnorm(50)
#Generation of random observations with normal distribution
y <- x + rnorm(50, mean=0, sd=0.5)
data <- as.data.frame(cbind(x, y))
#Show descriptive statistics for dataset
summary(data) -> descrp_obj
#Output object to csv file
write.csv(descrp_obj, file="descriptive_summary.csv")
#Visualize the data with scatter plot
library(ggplot2)
ggplot(data, aes(x=x, y=y)) +
  geom_point(size=2) +
  ggtitle("Scatterplot of X and Y") +
  theme(axis.text=element_text(size=12),
        axis.title = element_text(size=14),
        plot.title = element_text(size=20, face="bold"))
```

Figure 13: EDA example in R

The *summary()* function in R will provide a description of the data that will give the magnitude and the range of the data, shown in **Table 4**. However, what we cannot conclude from the descriptive statistics information is the distribution and the actual relationship between the two variables. (Dietrich, 2015)

Table 4: Descriptive summary of data observations

x	y
Min. :-1.89913	Min. :-2.04321
1st Qu.:-0.72347	1st Qu.:-0.77131
Median :-0.05805	Median : 0.06287
Mean : 0.03696	Mean : 0.03072
3rd Qu.: 0.69383	3rd Qu.: 0.75596
Max. : 2.25636	Max. : 2.19926

By visualizing the data, it is easier to assume the relationship between the two variables, as shown in **Figure 14**. The relationship between the two variables is important to determine in an early stage, preferably before model planning and building juncture. (Dietrich, 2015)

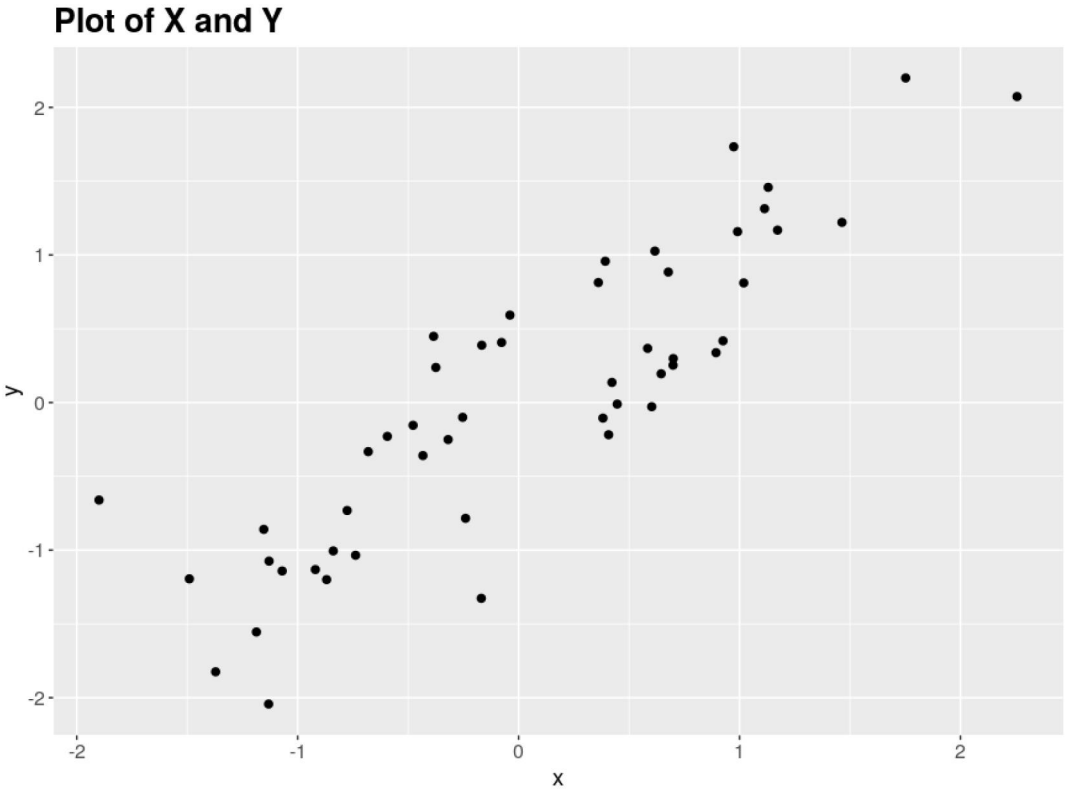

Figure 14: EDA data visualization

Statistician Francis Anscombe constructed the four datasets as an example to prove the importance of graphing data before starting to analyse. As mentioned, Anscombe's quarter is built of four datasets and each dataset have a x and an y variable. **Table 5** shows the four datasets with 11 rows in each dataset. (Dietrich, 2015)

Table 5: Anscombe's quarter represented as datasets

Anscombe's quarter							
Dataset 1		Dataset 2		Dataset 3		Dataset 4	
x	y	x	y	x	y	x	y
4	4.26	4	3.1	4	5.39	8	6.58
5	5.68	5	4.74	5	5.73	8	5.76
6	7.24	6	6.13	6	6.08	8	7.71
7	4.82	7	7.26	7	6.42	8	8.84
8	6.95	8	8.14	8	6.77	8	8.47
9	8.81	9	8.77	9	7.11	8	7.04
10	8.04	10	9.14	10	7.46	8	5.25
11	8.33	11	9.26	11	7.81	8	5.56
12	10.84	12	9.13	12	8.15	8	7.91
13	7.58	13	8.74	13	12.74	8	6.89
14	9.96	14	8.1	14	8.84	19	12.5

Anscombe's quarter is an example that indeed shows the usefulness of plotting data before starting the analysis and **Figure 15** shows the sample code used in this example. The programming language used is R.

```r
#Load Anscombe data set
data(anscombe)
anscombe
#Output object to csv file
write.csv(anscombe, file="anscombe_data.csv")
#Create factor levels
levels <- gl(4, nrow(anscombe))
# Group anscombe into a data frame
mydata <- with(anscombe, data.frame(x=c(x1,x2,x3,x4), y=c(y1,y2,y3,y4),
                                    QuarterDataSetNum=levels))
#Retrieve statistics for x and y for in each dataset
ds <-  plyr::ddply(mydata, "QuarterDataSetNum", plyr::summarise,
                   mean_x = mean(x),
                   variance_x = var(x),
                   mean_y = format(round(mean(y), 2), nsmall = 2),
                   variance_y = format(round(var(y), 2), nsmall = 2),
                   #correlation_xy = format(round(cor(x,y),3), nsmall = 3),
                   correlation_xy = round(cor(x,y)/.002)*.002,
                   estimate_line = paste(format(round(summary(lm(y~x))$coefficients[1,1],2), nsmall = 2),
                                         "+", paste(format(round(summary(lm(y~x))$coefficients[2,1],2), nsmall = 2),
                                         "x", sep = "")))
#Output object to csv file
write.csv(ds, file="anscombe_df.csv")
library(ggplot2)
theme_set(theme_bw())
# create the four plots of Figure 3-7
ggplot(mydata, aes(x,y)) +
  geom_point(size=2, colour = 'black') +
  geom_smooth(method="lm", fill=NA, fullrange=TRUE, colour = 'green') +
  facet_wrap(~QuarterDataSetNum)
```

Figure 15: Anscombe's quarter example source code

Contemplating the retrieved fundamental statistical properties for each dataset, shown in **Table 6**, it can be easily concluded that the four datasets are very similar. But by visualizing the datasets, we can conclude the contrary, as seen in **Figure 16**. The number above the scatterplots are related to the dataset number of Anscombe's quarter. The green line drawn in each scatterplot is the fitted line generated from linear regression models. **Figure 16** shows that the trends of datasets 1 and 3 are linear while the trend of dataset 2 is nonlinear. Dataset 4 on the other hand, have only two x-values and therefore it is not possible to decide if the linear assumption is correct. Looking at the estimated regression line in datasets 1 and 4, it can be concluded that they have a good fit, except for the outlier in dataset 4. (Dietrich, 2015)

Outliers are in many data mining cases treated as exceptions or noise. In unsupervised clustering, outliers are instances that are not grouped with the other observations in a dataset. Credit card fraud detecting application is a good example of a case where finding outliers in a dataset, containing information of credit card purchases, can be very impor-

tant. An outlier, in that case, is likely a positive identification of credit card fraud. (Roiger, 2017)

Table 6: Statistical properties for all datasets in the Anscombe's quarter

Dataset	*Mean of x*	*Variance of x*	*Mean of y*	*Variance of y*	*Correlation between x and y*	*Estimated regression line*
1	9	11	7.50	4.13	0.816	3.00 + 0.50x
2	9	11	7.50	4.13	0.816	3.00 + 0.50x
3	9	11	7.50	4.12	0.816	3.00 + 0.50x
4	9	11	7.50	4.12	0.816	3.00 + 0.50x

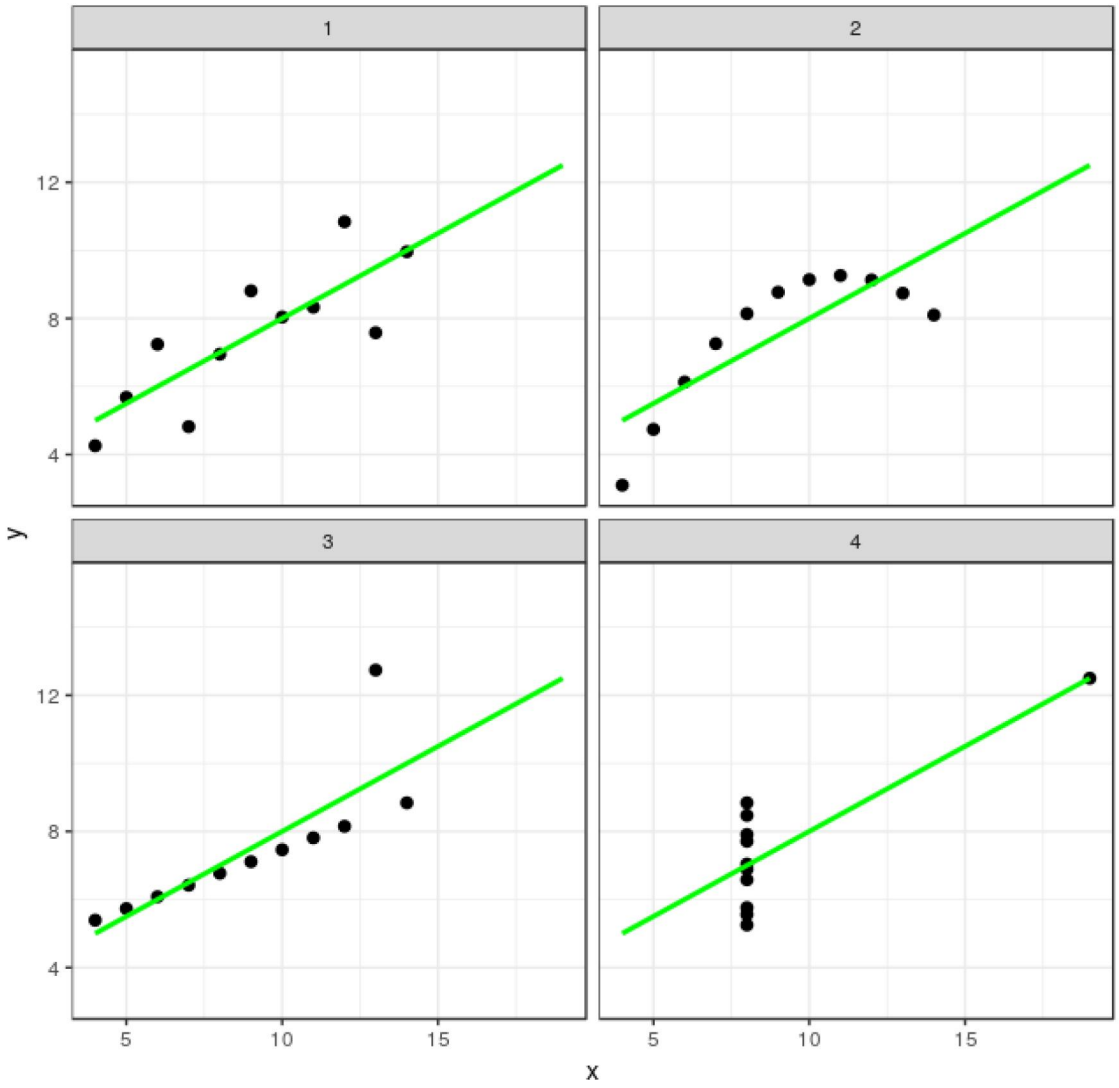

Figure 16: Scatterplots of datasets in Anscombe's quarter

It is necessary to point out the difference between EDA and data visualization. With data visualization one is presenting one's findings and this is generally done in the end of the analysis process as EDA is done in the beginning of the analysis process. (O'Neil & Schutt, 2014)

3.3 Data mining

It has been discussed earlier, in **Section 2.2.2**, that the amount of data that we have at our disposal is very large. The low storage costs also tend to make us buy new storage units rather than deleting data. Additionally, all our activities and actions on the World Wide Web are recorded. We have a huge increase in the data volume, leading to gap between the generation of data and our understanding of it. The data volume is constantly increasing, but at same time, the proportion of what we understand about it decreases. The electronically stored data is subject to finding patterns in data augmented by computers. Data mining has mostly been utilized by economics, forecasters, statisticians, and communication engineers. In these fields of work, it has been of interest to find patterns in data using an automated process. After the automated process, the data needs to be identified and validated before it can be used for predictions. The increase in the amount of available data has made data mining a rising business technology. (Witten et al., 2016)

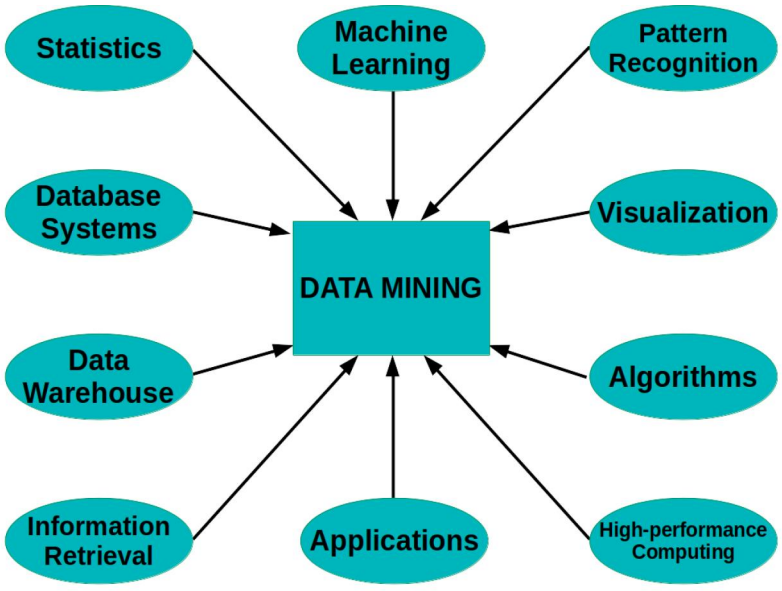

Figure 17: Domains of Data mining

The Knowledge Discovery from Data (KDD) process of data mining comprises the following iterative steps.

1. Data preprocessing. In this step, noise and inconsistent data is removed from the data set.

2. Data integration. Gathering data from different sources.

3. Data selection. Data that is needed to perform the analytical tasks is retrieved from the database.

4. Data transformation. Summary and aggregated operations are performed on the dataset in order to transform the data into more suitable form for data mining processes.

5. Pattern evaluation. The goal in this step is to identify the patterns of value.

6. Knowledge presentation. Present the findings to the users using knowledge-based representation and visualization. (Han Jiawei, 2012)

Data mining can be defined as a semi-automatic process that searches for and discovers patterns in data. The source data should, preferably, be available in considerable quantities and the discovered patterns from the source data must have some value that can be used to gain an organizational advantage. Data mining is the process of solving problems by using data from several sources. The discovered patterns enable the analyst to make non-trivial predictions on newly collected data. One way to discuss how the patterns are expressed is by looking at the two extremes, black boxes and transparent boxes. The properties of the two extremes, assuming that both extremes make very good predictions, are that the internal construction of the black box is very complex and unintelligible while the transparent box reveals the construction of the patterns. The difference between these two extremes is whether or not they are able to explain something about the data that have been mined. The patterns should also be presented in such a way that its structure can be examined and used in decision making. Such patterns are called *structural patterns*. (Witten et al., 2016)

3.3.1 Structural patterns

To be able to describe what structural patterns means, it is necessary to take **Table 7** into consideration. **Table 7** contains conditions from where an optician can determine what type of lenses that is to be recommended to a patient. One way to express the information given the rows in **Table 7** is to summarize them by rules. The rules creates a structural description, generalized from the available data, as shown in **Figure 18**. (Witten et al., 2016)

Table 7: Contact lens data (Witten et al., 2016)

Age	**Spectacle Prescription**	**Astigmatism**	**Tear Production Rate**	**Recommended Lenses**
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	no	reduced	none
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbyopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	myope	yes	normal	hard
presbyopic	hypermetrope	no	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none

```
IF Tear Production Rate == reduced
THEN recommendation = none
ELSE IF Age == young AND Astigmatism == no
THEN recommendation = hard
```

Figure 18: Structural description

In **Table 7**, age attribute can have three different values while attributes like astigmatism, tear production rate and spectacle prescription can each have two different values. The rules obtained from the input data presents a summary of the data. In real world learning processes, the input data may be incomplete and a part of the task is to come up with new examples by summarizing the input data. Ignoring some of the rows in **Table 7** where tear production rate takes the value of *reduced*, it would still be possible to make the assumption that lenses is not recommended if the tear rate production is reduced. This generalization would still be valid for the missing rows from the dataset. (Witten et al., 2016)

It is possible to process the data in **Table 7** and try to create a complete set of rules that tries to classify all possible cases in the dataset. But in some cases, more than one rule may be valid for the same instance which will lead to conflicting recommendations. To avoid conflicting recommendations, probabilities or weights can be mapped to the rules themselves. The mapping of probabilities or weights will induce the notion that some rule have a higher priority than other rules. The same structural pattern could be presented as a decision tree. The decision tree algorithm will be discussed in more detail in **Section 3.5.4**. (Witten et al., 2016)

3.3.2 Data preprocessing

The source data may not always be of good shape and quality when it is collected in real-world applications. The data may contain null-values, outliers due to sensor errors, instrument-induced bias and other faults that will have a negative effect on the model fitting. It is an absolute necessity to preprocess the data in order to eliminate these errors. (Bonnin, 2017)

If a dataset is missing some features or values, the best way to approach this problem would be to use an automatic strategy. The automated strategy will use the existing values to calculate either the mean, median, or frequency to fill in the missing features. **Figure**

19 shows a Python example on how to use different strategies to process an incomplete dataset. **Table 8** shows the individual results of the example. (Bonaccorso, 2017)

```
1 from sklearn.preprocessing import Imputer
2 import numpy as np
3 data = np.array([[1, np.nan, 2, 3], [2, 3, np.nan, 3],
4                           [-1, 4, 2, 4],[1, 4, 2, np.nan]])
5 imp = Imputer(strategy='mean')
6 array_mean = imp.fit_transform(data)
7 imp = Imputer(strategy='median')
8 array_median = imp.fit_transform(data)
9 imp = Imputer(strategy='most_frequent')
10 array_mostFrequent = imp.fit_transform(data)
```

Figure 19: Python example of replacing missing values

Table 8: Result of different strategies of imputation transformer

Sample data			
1	nan	2	3
2	3	nan	3
-1	4	2	4
1	4	2	nan

Mean			
1	3.66	2	3
2	3	2	3
-1	4	2	4
1	4	2	3.33

Median			
1	4	2	3
2	3	2	3
-1	4	2	4
1	4	2	3

Most Frequent			
1	4	2	3
2	3	2	3
-1	4	2	4
1	4	2	3

To calculate the mean of a dataset containing numerical values, one takes the sum of all elements and divide it by number of elements in the data set. **Equation 1** shows the formula for the mean value. The mean value is the weighted center of the samples in the dataset. (Bonaccorso, 2017)

With N elements in a dataset, median is the value that is separating the lesser half of a dataset from the upper half, if the dataset contains an odd number of elements. If the dataset has an even number of elements, then median is calculated as the mean of the two middle values. The dataset should be ordered so the elements are in descending or ascending order. (Dangeti, 2017)

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1}$$

where:

μ: is the mean value

N: is the number of data points in the dataset

x_i: is a data point

Using the frequency strategy to fill in the missing values, the most frequent value is replacing all the missing values in the dataset. (Bonaccorso, 2017)

Other preprocessing methods are also available, such as, creating a sub-model to predict features or removing the whole line. The first method can become very complex to implement since a supervised strategy is needed to train a model for each feature and to predict a value. The second method can only be considered as an alternative if the dataset is very large, or the dataset contains a large number of missing values, or if prediction is risking the integrity of the data. (Bonaccorso, 2017)

As mentioned before, mean is the weighted center of all the samples in a dataset, but it will not give much information about dispersed samples. The variance, on the other hand, will describe how dispersed the samples are in a dataset. **Equation 2** shows the mathematical definition of the variance. A high value indicates that the spread of the samples in the dataset is high. A low value indicates that the spread of the samples in the dataset is low. (Bonnin, 2017)

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu)^2 \tag{2}$$

where:

σ^2: is the variance

N: is the number of data points in the dataset

x_i: is a data point

μ: is the mean value of the data

From **Equation 2** it is possible to derive the equation for standard deviation shown in **Equation 3**. (Bonnin, 2017) Standard deviation is a measurement on how much the samples differs from the mean value of the same dataset. The standard deviation is, for instance, used when calculating similarity scores. (Joshi, 2017)

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{3}$$

where:

σ: is the standard deviation

N: is the number of data points in the dataset

x_i: is a data point

μ: is the mean value of the data

Normalization and feature scaling are two common techniques that are used when preprocessing data. Transforming inconsistent data will facilitate further processing of the data while, at the same time, maintaining the integrity of the data. The transformed data will provide better stochastic properties and eliminate the risks of a negative impact on the data model. Especially optimization techniques will converge better after data has been normalized. **Figure 20** shows a comparison between raw data and its normalized version. (Bonnin, 2017)

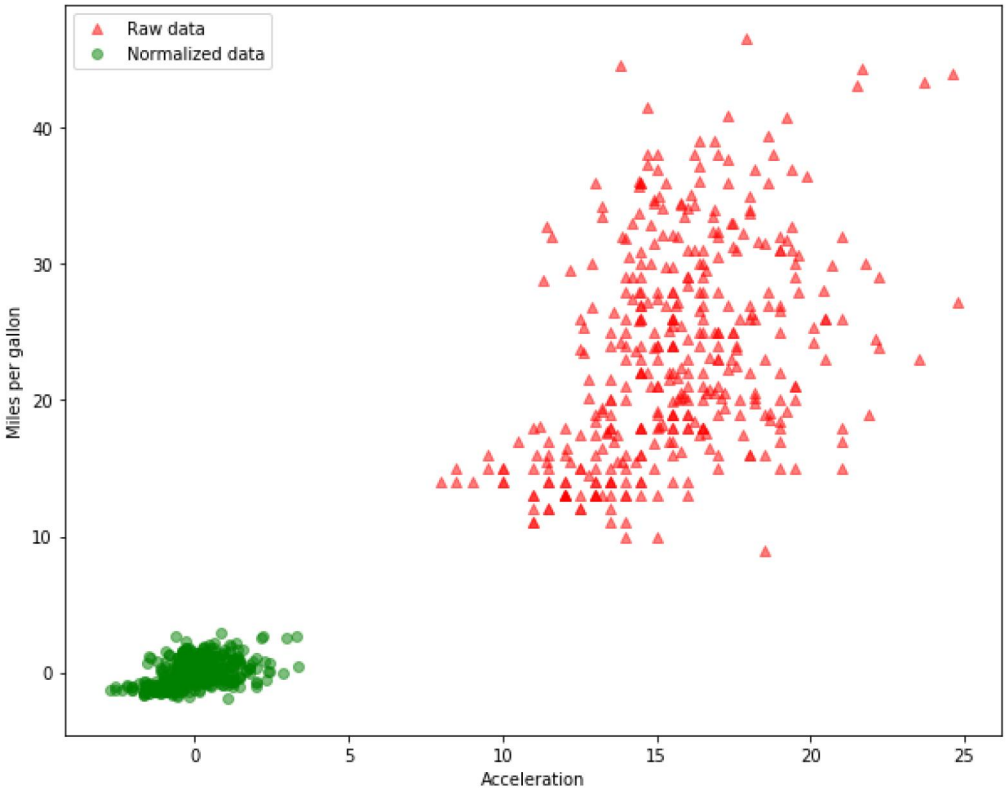

Figure 20: Comparison of raw and normalized data

Standardization techniques aims to transform the data source so data points get properties similar to the normal distribution, that is a mean value of 0 and standard deviation value of 1. **Equation 4** is the probability density function (PDF) of the normal distribution. (Bonnin, 2017) According central limit theorem, all samples of size N from a population with μ and σ^2 approaches normal distribution. Other useful probability distributions are, Bernoulli distribution, uniform distribution, and logistic distribution. (Dangeti, 2017)

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{4}$$

where:

σ^2: is the variance

μ: is the mean value of the data

However, to make the dataset obtain the properties of normal distribution, **Equation 5** can be used where the mean value of all data points is subtracted from the a feature and divided by the variance. The result is a scaled feature with mean of 0 and variance of 1. (Zheng & Casari, 2018)

$$z = \frac{x - \mu}{\sigma} \tag{5}$$

where:

z: is the z-score

σ: is the standard deviation

μ: is the mean value of the data

In classification problems, datasets will probably contain human readable text, or categorical labels, which cannot be directly fed to an algorithm. The categorical data needs to be encoded, i.e. transformed into numerical format, so it can be processed by algorithms. There are several approaches to encode the labels in a dataset and each approach has its advantage and disadvantage. Classes, available in Python, that will handle the encoding process are e.g. *LabelEncoder*, *LableBinarizer* and *DictVectorizer*. (Bonaccorso, 2017)

3.3.3 Principal component analysis

Principal Component Analysis (PCA) is used in the data pre-processing step as feature selection. A dataset containing features that are highly correlated with each other might induce extra error in our machine learning models if no feature selection is made before the model training. (Burger, 2018)

The key concept of PCA is to reduce dimensionality by replacing redundant features with only a few features that are capable of providing the same information which the original feature space provides. To explain the concept a little more detailed we need to look at the data points in the feature space, shown in **Figure 21**. The column space in **Figure 21 (a)** has full rank and the data points distributed evenly in the two feature dimensions. Here, the whole set of data points is referred to as a blob. However, in **Figure 21 (b)**, some of the features are scalar multiple of another feature, thus making them duplicates. In this particular case, the intrinsic dimensionality of the dataset, or blob, is one. The third scenario is where features are almost identical, this also called an emaciated blob. When two features are relatively close to each other, it is possible to replace the two original features with a feature that is situated on a diagonal line between the two original points. (Zheng & Casari, 2018)

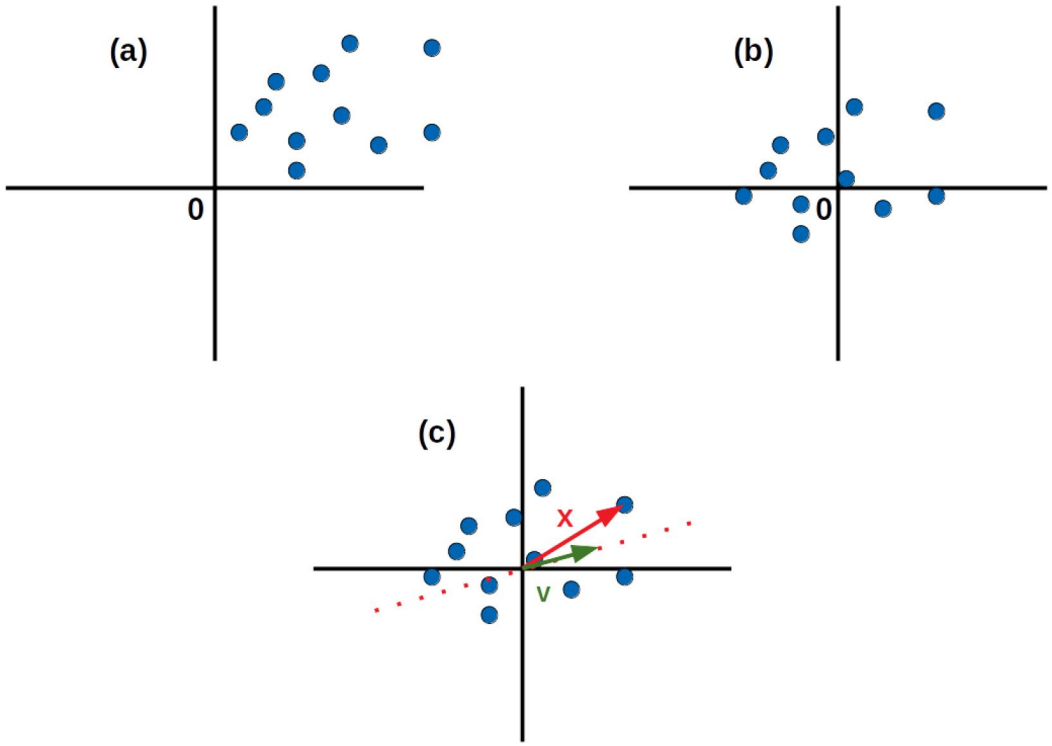

Figure 21: Initial steps of PCA

(Müller & Guido, 2017)

One of the first steps of PCA is to center the data matrix by subtracting the mean from every data point in the data matrix using **Equation 6**. After the data matrix have been centered, we want to calculate singular value decomposition (SVD) of the centered data matrix, see **Equation 7**. (Zheng & Casari, 2018)

$$C = X - 1\mu^T \tag{6}$$

where

1: column vector filled with 1s.

μ: column vector containing the average value of the rows of X.

$$C = U\Sigma V^T \tag{7}$$

where

C: is the centered data matrix of dimension n x d, n is the number of data points and d is the number of features.

V and U: are orthogonal matrices.

Σ: is a diagonal matrix.

Next, the data is transformed into a new feature space and we want to find the projection coordinates, see **Equation 8** and **Equation 9**, of the data vectors, x, when projected onto the vector v, see **Figure 21 (c)**. Vector v is constrained to have unit norm. The projection coordinates will be used when calculating the variance. (Zheng & Casari, 2018)

$$z = x^T v \tag{8}$$

where

z: is a scalar

x and v: are column vectors.

$$\bar{z} = Xv \tag{9}$$

where

$\bar{z}$: column vector containing projection coordinates.

X: is the data matrix.

v: is the new feature vector.

It is necessary to mention that the empirical variance is used, see **Equation 10**, since the variance of a random variable Z is based on observed data points $z_i, \ldots, z_n$. (Zheng & Casari, 2018)

PCA algorithm will then proceed to find the direction in which the features have the highest correlation in respect to each other. This can be achieved by finding the direction of maximum variance, denoted as *Component 1* in **Figure 22** top left plot. When *Component 1* has been found, PCA will continue to seek for *Component 2*, which will be the direction that contains the most information and that is also orthogonal with respect to the first component. (Müller & Guido, 2017)

$$Var_{emp}(Z) = \frac{1}{n-1}\sum_{i=1}^{n} z_i^2 \tag{10}$$

where

z_i: is the scalar of the projection points

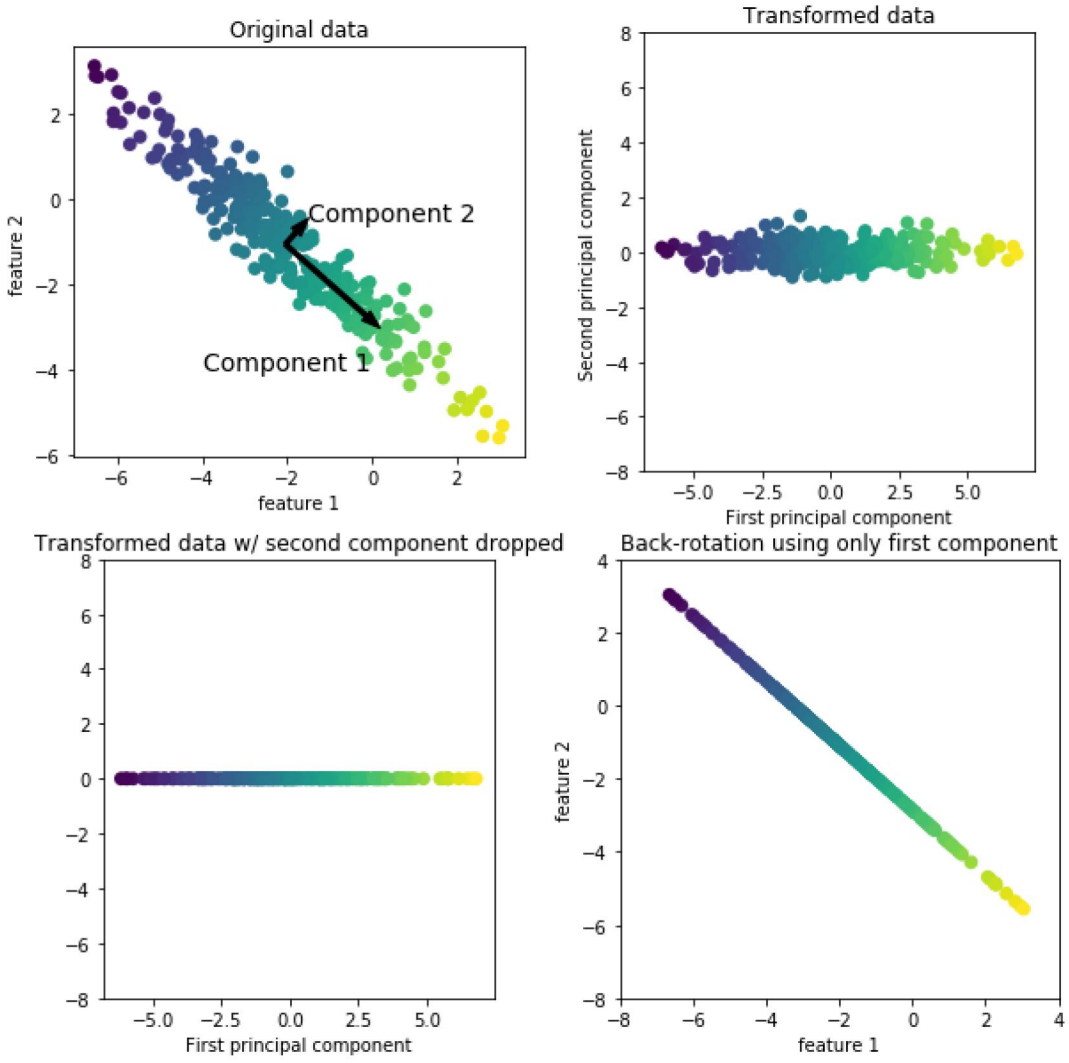

Figure 22: Data transformation using PCA

(Müller & Guido, 2017)

From the projected data we can find the maximized variance. By combining the mathematical formula of **Equation 8** and **Equation 10** we will get the objective function of principal components, seen in **Equation 11**. The denominator, $n-1$, from **Equation 10** can be left out from the formula since it will not have an impact on where the maximizing value is. The equation can be rewritten in matrix-vector notation, due to the sum of squares identity. (Zheng & Casari, 2018)

$$max_w \sum_{i=1}^{n} \left(x_i^T w\right)^2, \ \ where \ \ w^T w = 1 \tag{11}$$

$$max_w w^T, \ \ where \ \ w^T w = 1 \tag{12}$$

The plot in the top right corner in **Figure 22** shows a transformed version of the original data. Here, the data matrix has been centered around zero and two principal components have been found. The principal components have been aligned according to the x- and y-axis respectively. (Zheng & Casari, 2018) The features can be transformed by using linear projection. From **Equation 13** we get the SVD of data matrix X. V_k is a matrix that holds the first k left singular vectors in its columns. The projection matrix, seen in **Equation 14** can replace the projection vector presented in **Equation 9**. (Zheng & Casari, 2018)

$$X = U\Sigma V^T \tag{13}$$

where

X: is the data matrix of dimension $n \, x \, d$, n is the number of data points and d is the number of features.

V and U: are orthogonal matrices.

Σ: is a diagonal matrix.

$$W = V_k \tag{14}$$

We can use **Equation 15** to calculate the matrix containing the coordinates of the projections. The equation can be reduced to since singular vectors are orthogonal.

$$\begin{aligned} Z &= XW \\ &= XV_k \\ &= U\Sigma V^T V_k \\ &= U_k \Sigma_k \end{aligned} \tag{15}$$

The dimensionality reduction in PCA is done by choosing one or a few of the principal components, as shown in **Figure 22** lower left plot where only the first principal component has been kept. This action will reduce the dataset from two dimensions to one dimension. The final step of PCA is to undo the data transformation and add the mean back to the data. The last step is performed to remove noise from the data and to get a visual representation of what information that have been kept after the PCA process has been completed, see **Figure 22** bottom right plot. (Müller & Guido, 2017)

3.3.4 Data integration

Organizations that maintains a data warehouse faces the problem with data integration, that is, how to combine data from different data sources into a consistent data store. Object matching and schema integration are time consuming tasks that also requires special

attention. For example, in one database we have an attribute named *customer_number* while in another database we have an attribute named *cust_id*. Both attributes contain the same information, thus requiring special attention in the integration process. This is also known as *entity identification problem*. (Han Jiawei, 2012)

To facilitate the data integration process, it is recommended to maintain metadata for attributes that include name, data type, meaning, range of permitted values, and rules for handling blank, zeros, or null values. Furthermore, the structure of the data must be maintained so that all of the attribute's dependencies and references in the source database is maintained when integrated in the destination database. Consider a source system that applies a discount on the whole order whereas in another system, the discount is applied on each item. In this case, it is important to examine the data structure in detail before data integration to maintain the functionality. (Roiger, 2017)

Some data mining algorithms, e.g. neural networks and nearest neighbor classifiers, will perform poorly if datasets are populated with attributes with a low degree of predictive value. It has been shown that the required number of training instances needed to develop an accurate supervised model is affected by the amount of unrelated attributes in the dataset. Attribute selection can be done with the help of filtering methods, wrapper techniques, correlation analysis, probability scores, principal component analysis, or self-organizing maps. Most of the techniques can be used in both supervised and unsupervised clustering. (Roiger, 2017)

By performing correlation analysis in the data integration stage, it can be possible to reduce data redundancy in a data warehouse. An attribute is considered to be redundant if the same attribute can be derived from a set of other attributes, or if an attribute obtains a high correlation coefficient. (Han Jiawei, 2012)

The correlation coefficient for numerical data can be calculated using **Equation 16**, note that $-1 \leq r_{A,B} \geq 1$. A correlation value that is near plus one $(+1)$ indicates a strong

positive correlation, i.e. the variables move in the same direction. A correlation value that is near minus one (-1) indicates a strong negative correlation, i.e. the variables move in opposite direction. A correlation value of zero (0) indicates that there is no correlation between the variables. (Bonnin, 2017)

$$r_{A,B} = \frac{\sum_{i=1}^{n} (a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A \sigma_B} \tag{16}$$

where:

n: is the number of tuples

a_i and b_i: are the respective values of A and B in tuple i

$\bar{A}$ and $\bar{B}$: are the respective mean values of A and B

σ_A and σ_B: are the respective standard deviation values of A and B

3.3.5 Data transformation

Data transformation overlaps with the preprocessing tasks described earlier in **Section 3.3.2**, but the purpose of data transformation is to increase the efficiency of the mining process. Besides normalization, discussed in **Section 3.3.2**, strategies like data type conversion and attribute selection belongs to the data transformation process. (Roiger, 2017)

When building data mining models, attribute selection strategies are needed to mitigate the performance impact attributes with a low predictive value have on data mining algorithms. Algorithms like neural networks and nearest neighbor classifiers fails to distinguish relevant attributes from the irrelevant attributes, since some attributes are not predictive of class membership. (Roiger, 2017)

A dataset containing a large number of attributes of varying predictive value will have a negative impact on data mining algorithms like neural network and nearest neighbor classifiers. When building a supervised learner model, the number of training instances needed depends on the number of irrelevant attributes in the dataset. This is the reason why *wrapper*, *filter* and *embedded* techniques are implemented. (Zheng & Casari, 2018)

Filtering methods, e.g. decision tree, uses a measure of quality to filter out attributes from the dataset. One way to measure the quality is to calculate the correlation or mutual information between every attribute and the response value. Additionally, some threshold value must be carefully chosen in order to select the right attributes that will be used in the algorithm for building the final model. (Roiger, 2017)

Wrapper methods views the model as an black box that measures the goodness of an attribute set used as input while another method will refine the subset. An example of a wrapper technique is to use forward selection in combination with nearest neighbor operator (k-NN) to get the best set of input attributes that is going to be used as input in a machine learning algorithm. (Zheng & Casari, 2018)

Embedded methods are not considered to be as powerful as wrapper methods but they are less computationally expensive. Embedded methods selects the attribute as a part of the training step. Consider a decision tree, this method will split the decision tree based on one attribute. L_1 regularizer uses a few attributes and can be implemented to the training object of any linear model. (Zheng & Casari, 2018)

3.3.6 Pattern evaluation

The evaluation part of the KDD is to determine if the model can provide results that are understandable by users, if so, it can be used outside the test environment. The interpretation and evaluation can be performed with the help of the following analysis methods.

- *Statistical analysis.* A method of determining if there is a notable performance difference between data mining models created using distinct sets of attributes.
- *Heuristic analysis.* Data mining tools can provide calculated heuristics.
- *Experimental analysis.* The scientist can experiment with attribute and instance choices to generate a better model. K-means and Neural network machine learning techniques will build models with some minor variations using the same data and parameter settings. Other techniques can generate recognizably different models with some minor changes in data and parameter settings.
- *Human analysis.* The data scientist must decide if discovered knowledge can be applied on future problems. (Roiger, 2017)

3.4 Machine learning algorithms

In machine learning, computer science is used for extracting useful information from noisy data. Machine learning is a building block to artificial intelligence and it consists of three different categories; *supervised learning*, *unsupervised learning*, and *reinforced learning*. The typical problem each category can solve is shown in **Table 9**. (Kirk, 2017)

Table 9: Typical problems that are solved with the help of machine learning (Kirk, 2017)

Problem	**Machine learning category**
Function approximation or fitting data to a function	Supervised learning
Understand the data with absence of feedback	Unsupervised learning
Maximize rewards over time	Reinforcement learning

As mentioned in previous section, machine learning is the essential part of AI systems and it helps us build AI systems. However, deep learning is a specific part of machine learning that uses multilayered models that are built of simpler statistical components, generic learning techniques and iterative experiences. Problems that are commonly solved with the help of deep learning are e.g. image classification, real-time visual tracking and speech recognition. (Bonaccorso, 2017) Statistical analysis is a fundamental building

block of machine learning. The relation between AI, machine learning and deep learning is shown in **Figure 23**. (Chio & Freeman, 2018)

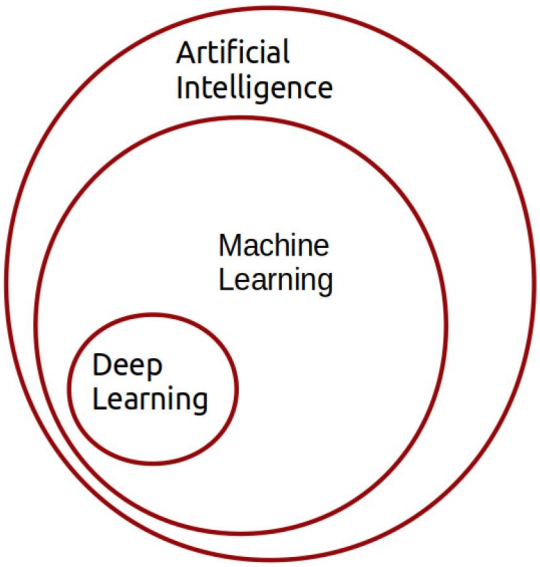

Figure 23: An overview of how AI relates to machine learning

3.5 Supervised learning

The majority of machine learning algorithms fall under the category *supervised learner*. In short terms, in supervised learning, the machine learning models are graded and later tuned against some known quantity. (Burger, 2018)

When developing a supervised learning model, it is desirable to have a model that can generalized new, unseen data as accurately as possible. If the built model is rather good at making accurate predictions, it can be said that the model is able to *generalize* from the training set to the test set. It can be expected of the model to make good predictions on new data if the data used in the training set and test set have enough in common. (Müller & Guido, 2017)

When building the machine learning model, which purpose is to estimate the possible underlying function, it is very important to remember to leave some room for the model

to make generalizations for unknown inputs during the training process. It is during the training stage one can face problems like *underfitting* and *overfitting*. Overfitting can sometimes be more challenging to detect because it can easily be concluded that the model has obtained a perfect fit. Underfitting can, on the other hand, be detected by the prediction error of the model. (Bonaccorso, 2017)

- **Underfitting** means that the machine learning model is not capable of capturing the dynamics of the training set to its full potential.
- **Overfitting** means that the machine learning model is able to generalize the original dynamics of the training set. It can thus associate the majority of the known samples to the correct output value. However, when the model processes a new unknown sample it can result in a very high prediction error. (Bonaccorso, 2017)

3.5.1 Regression

Regression modeling is commonly used when we want to predict future values. From a dataset containing numerical values, we want to find a function that is able to describe the relationship between a dependent variable and an independent variable. Such function is called a *regression function*. The most common function types that are used when modeling data are linear, polynomial, and exponential functions. (Bonnin, 2017)

In linear modeling, the assumption is that the all the data points, both known and unknown, lies on a hyperplane and that the maximum error is proportional to both the training quality and the adaptability of the original dataset. **Equation 17** represents a dataset consisting of real-values vectors and each input vector can be associated with a real value, shown in **Equation 18**. The mathematical formula for the regression process is shown in **Equation 19**. (Bonaccorso, 2017)

$$X = \{\bar{x_1}, \bar{x_2}, ..., \bar{x_n}\} \; where \; \bar{x_n} \in \mathbb{R}^n \tag{17}$$

$$Y = \{y_1, y_2, ..., x_n\} \; where \; y_n \in \mathbb{R} \tag{18}$$

$$\tilde{y} = \sum_{i=1}^{m} \alpha_i x_i \quad A = \{\alpha_0, \alpha_1, ..., \alpha_m\} \tag{19}$$

In R, there are several inbuilt datasets available. To demonstrate linear regression, the inbuilt *mtcars* dataset has been used. This dataset contains several features for 32 different cars. The following example is a very simplified one, where we do not split the dataset into a training set and a test set. A linear regression model which model the fuel efficiency as a function of engine displacement will be created and the sample code for this linear regression example can be seen in **Figure 24**. (Burger, 2018)

```
#Create a linear model using the inuilt
# mtcars:
model <- lm(mtcars$mpg ~ mtcars$disp)
#Create plot
par(mfrow=c(2,1))
plot(y = mtcars$mpg, x = mtcars$disp, xlab = "Engine size (cubic inches)",
     ylab = "Fuel efficiency (Miles per Gallon)",
     main = "Scatterplot of engine displacement vs. mpg")
plot(y = mtcars$mpg,
     x = mtcars$disp, xlab = "Engine size (cubic inches)",
     ylab = "Fuel efficiency (Miles per Gallon)",
     main = "Dataset with regression function")
abline(a = coef(model[1]), b = coef(model)[2], lty = 2)
#Call the summary function to retreive info. about the model
summary(model)
```

Figure 24: Linear regression model source code

But let us contemplate on the output, shown in **Figure 25**, of the *summary()* function. *Residuals* is the vertical distance for each data point to the regression line and this describes the fit of the line, its mathematical representation is shown in **Equation 20**. A small value means that it has a good fit. We can also find the estimated coefficients to the linear equation. In this particular example, the equation takes the form of $y = (0.04 \pm 0.005)x + (29.59 \pm 1.23)$ with the error estimates included. (Burger, 2018)

$$\forall i \in (0,n)\ r_i = x_i - \tilde{x}_i \tag{20}$$

The *summary()* output contains a lot of information about our model, one of the most important piece of information from the output is the multiple R^2 value. This value will provide an accuracy assessment of the model. The closer this value is to 1, the better the model. The R^2 value is, however, not so relevant in this particular example since only one predictor is used. The mathematical representation of R^2 is shown in **Equation 21** (Burger, 2018)

$$R^2 = 1 - \frac{\sum_{i=1}^{n} r_i^2}{\sum_{i=1}^{n} (x_i - E[X])^2} \tag{21}$$

```
Call:
lm(formula = mtcars$mpg ~ mtcars$disp)

Residuals:
    Min      1Q  Median      3Q     Max
-4.8922 -2.2022 -0.9631  1.6272  7.2305

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 29.599855   1.229720  24.070  < 2e-16 ***
mtcars$disp -0.041215   0.004712  -8.747 9.38e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.251 on 30 degrees of freedom
Multiple R-squared:  0.7183,    Adjusted R-squared:  0.709
F-statistic: 76.51 on 1 and 30 DF,  p-value: 9.38e-10
```

Figure 25: Summary output

The end result of the simple regression example can be seen in **Figure 26**. A regression function that describes the relation between size of the engine and the fuel consumption has been created and its estimations visualized. (Burger, 2018)

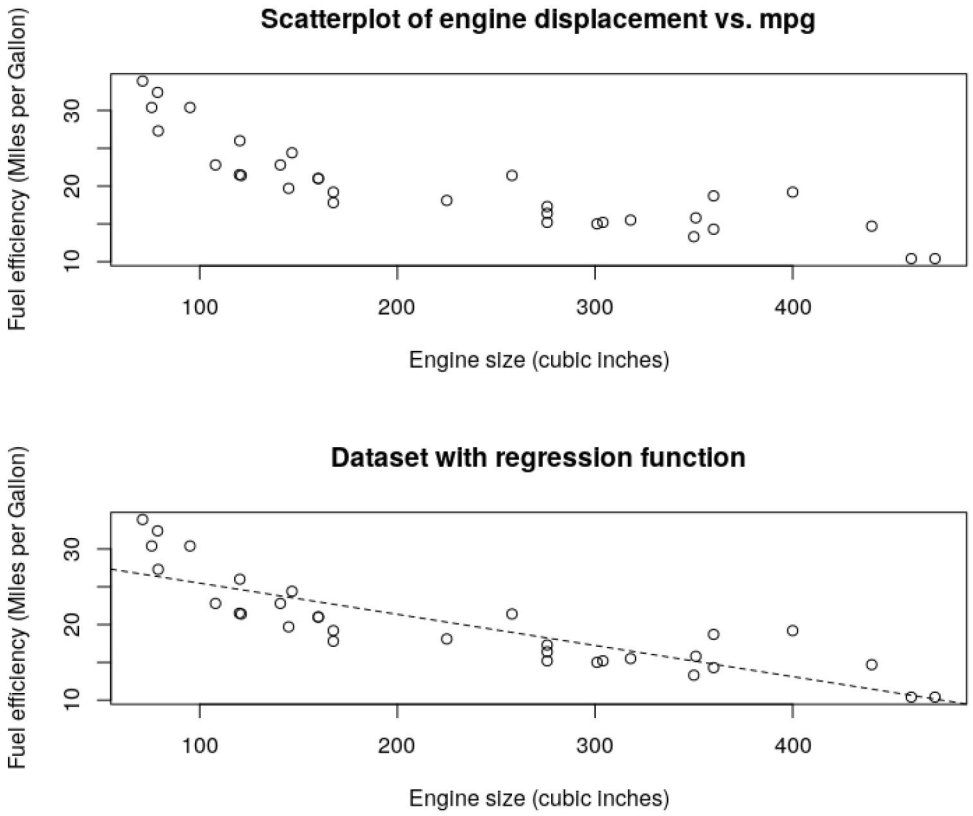

Figure 26: Linear regression model

In the previous example, a simple linear regression model was made with only one variable and the whole dataset was used as training set. By including more variables in the model training, the accuracy of the model might increase. More variables will not necessary guarantee an increase in model accuracy since some of the coefficients might have very little statistical value. (Burger, 2018)

Figure 27 shows an example of multivariate regression model fitting. The dataset is a list of 50 fictional startups containing the information on how much money the companies spend on marketing, research and development (R&D) and administration. The dataset also contains the geographical location of the companies and their profit. The dataset used in this example is available in **Appendix A**. In this example, the dataset will also be split into a train data and test data by 80 percent - 20 percent. A seed value is set in order to

repeat random sampling each time we have the same observations in training set and test set. (Bonnin, 2017)

```
#Python - Multiple Linear Regression
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
# Importing the dataset
dataset = pd.read_csv('50_Startups.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 4].values
# Encoding categorical data
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder = LabelEncoder()
X[:, 3] = labelencoder.fit_transform(X[:, 3])
onehotencoder = OneHotEncoder(categorical_features = [3])
X = onehotencoder.fit_transform(X).toarray()
#Split dataset into training set and test set
#Split ratio        80%              20%
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size = 0.2,
                                                    random_state = 0)
# Fit the linear multople linear regression model to
# training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
# Predicting the Test set results
y_pred = regressor.predict(X_test)
```

Figure 27: Multivariate regression model

In the dataset, categorical data is also present. Since regression can only be used on numerical data, the categorical data must be encoded so it can be represented numerically. Utilising *One-Hot encoding* method transforms the categorical data into groups of bits where each bit represents a category. If a variable can only belong to one category, then only one bit in the group of bits can obtain the value 1. Since each bit is a feature, a categorical variable with k categories will be transformed into a feature vector of length k. If $k-1$ bits are 0, then the last bit must be 1. The mathematical representation of the constraint is shown in **Equation 22** while the end result of the One-hot encoding method of three cities is shown in **Table 10**. (Zheng & Casari, 2018)

$$e_1 + e_2 + \ldots + e_n = 1 \tag{22}$$

Table 10: One-hot encoding

	e_1	e_2	e_3
New York	1	0	0
California	0	1	0
Florida	0	0	1

The trained multivariate linear model will use the independent variables from the test set and output predicted values. When comparing these values with the ones available from the test set, shown in **Figure 28**, we can see that the model is able to generalize the data quite well. It could also be possible to minimize the number of coefficients used in the model by utilising regularization technique to only include the coefficients with higher statistical value. The *lasso* regression method will scale down the coefficients according to their impact on the model and some coefficients can be scaled down to zero. After refining the model by eliminating the coefficients with no impact, it is possible to rerun the lasso regression with the new set of features. This is an iterative process that is done until you have a set of features with the highest impact on your model. It is necessary to point out that features which obtain a value of zero after the lasso regression still have some sort of relationship to the dependent variable, but its impact is unimportant compared to other features. (Burger, 2018)

Real values	Predicted values
103282.38	103015.2
144259.4	132582.28
146121.95	132447.74
77798.83	71976.1
191050.39	178537.48
105008.31	116161.24
81229.06	67851.69
97483.56	98791.73
110352.25	113969.44
166187.94	167921.07

Figure 28: Comparison between real values and values predicted by the model

3.5.2 K-Nearest neighbors

The k-nearest neighbor (k-NN) machine learning algorithm belongs to the group of algorithms that gathers the training data points, that are passed in to the algorithm, only to learn the generalizations around a test sample during the classification. This type of algorithm is called *lazy learner*. The training phase of the k-nearest neighbor algorithm is made up of storing the available feature vectors and matching sample labels in the model. The target is predicted by measuring the distance to the test sample's k nearest neighbors. The *Euclidean distance*, denoted in **Equation 23**, is calculated for continuous variables. For discrete variables, the *Hamming distance* is calculated. (Chio & Freeman, 2018)

$$d(x,y) = \sqrt{\sum_{i=0}^{n} (x_i - y_i)^2} \tag{23}$$

The fact that k-NN model stores all the available feature variables will result in a large model size. Furthermore, the classification speed will also suffer from the generalization work being postponed until the actual classification time. It is also important to have balanced training sets, otherwise the model will have higher probability to shift classifications of the test samples towards the training sets with more test samples. Even if k-NN is a simple and a popular machine learning example it is seldom used in practice due to the drawbacks mentioned. (Chio & Freeman, 2018)

3.5.3 Naive Bayes classifiers

Naive Bayes classifier, classifies the available data based on probabilities using Bayesian statistics. This type of classification will, based on the available features, estimate the probability that the output belongs to a certain class with the assumption that the predictors are independent of each other. (Burger, 2018)

Using conditional probability, we can for example calculate the probability that label Y for a given data point belongs to class C. The given data pint has a feature set of $X = x_1, ..., x_n$. See **Equation 24**. **Equation 24** can be rewritten by applying Baye's theorem, see **Equation 25**. (Chio & Freeman, 2018)

$$Pr\,[Y = C \,|\, X = (x_1, ..., x_n)] \tag{24}$$

$$\frac{Pr[X = (x_1, ..., x_n)\,|Y = C] \bullet Pr[Y = C]}{Pr[X = (x_1, ..., x_n)]} \tag{25}$$

Equation 26 is obtained by making the strong assumption that features for samples from each class are selected independently of each other. When dealing with two-class classification problems, it is only necessary to calculate the ratio of the probability estimates,

that is, for class C_1 and C_2. It is possible to estimate the numerator of **Equation 24** from labeled data. It can be concluded that, from all samples in class C, $Pr[X_i = x_i|Y = C]$ is the fraction of samples with the ith feature equal to x_i. $Pr[Y = C]$ is the fraction of samples from the same class C of all labeled samples. Using **Equation 27**, we can calculate the ratio of the estimates which is an indication of which class is more likely for an output. If $\theta > 1$, then it is more likely to belong to $C1$. If $\theta < 1$, then it is more likely to belong to $C2$. (Chio & Freeman, 2018)

$$Pr[X = (x_1, ..., x_n)|Y = C] = \prod_{i=1}^{n} Pr[X_i = x_i|Y = C] \tag{26}$$

$$\theta = \frac{Pr[Y = C_1|X = (x1, ... ,x_n)]}{Pr[Y = C_2|X = (x1, ... ,x_n)]} \tag{27}$$

If we are calculating the probability for dependent events, we need to approach the problem in a different way. In this type of problems, the probability will change after that the first event has occurred, in this case, the probability can be expressed as in **Equation 28**. (Müller & Guido, 2017)

$$P(B \mid A) = \frac{P(B) * P(A \mid B)}{P(A)} \tag{28}$$

Naive Bayes classifier can be used for identifying fraudulent orders. Spam filters can also be built with the help of Naive Bayes classifier, where the words are used as features. Naive Bayes classifiers has also been used in studies for classifying breast cancer. (Chio & Freeman, 2018)(Kirk, 2017)

3.5.4 Decision trees

Decision trees belongs to the supervised category of machine learning algorithms and it can be a good alternative to complex methods. Decision trees are often used for solving decision making problems by building a binary tree data structure using the datapoints in the dataset. Classification trees uses categorical values as input while regression trees use real values. Decision trees can use raw numerical and categorical data, it does not require any normalization or dummy variable creation. An example of how a decision tree might look like can be seen in **Figure 29**. (Bonaccorso, 2017)

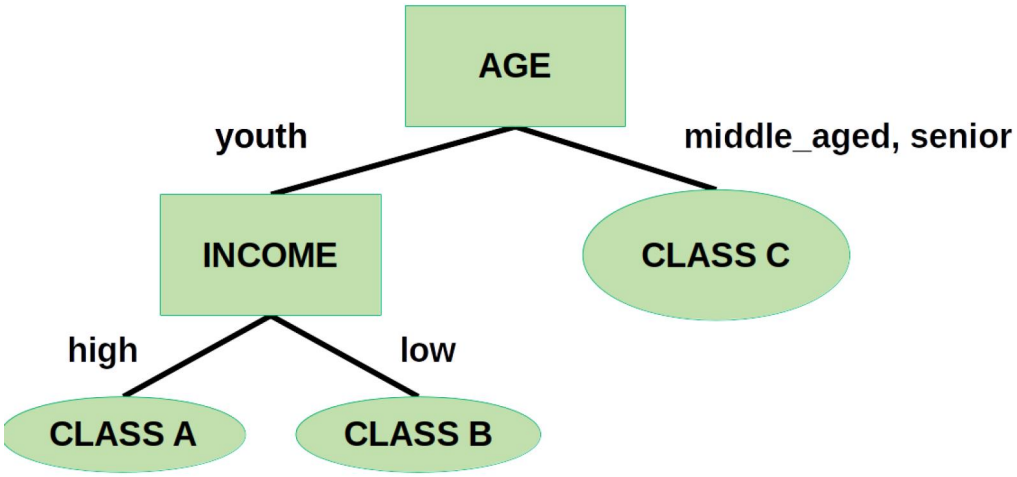

Figure 29: What a decision tree might look like

Decision trees, starting at the top of the tree, will split a dataset into subsets based some binary condition. Each subset is also divided into smaller subsets based on some other binary condition. Each split is based on some quality measurement, e.g. *Gini impurity*, *variance reduction* or *information gain*. The formula for calculating the Gini impurity is shown in **Equation 29**. (Bonaccorso, 2017)

$$I_{Gini}(f) = \sum_{i=1}^{m} p(f_i)(1 - p(f_i)) \tag{29}$$

The dessication of the dataset continues:

- Until a predefined maximum branch depth has been reached.
- Until either of child nodes does not contain a defined minimum number of samples.
- Until all nodes in the decision tree are populated with samples from the same class. If this is the case, then the *impurity* is null. (Chio & Freeman, 2018)

The optimal output of the algorithm is a tree structure that shows the binary decision made for each leaf node and the two possible outcomes. Overfitting is a well-known problem for the decision tree algorithm, the overfitting in this case, will lead to complex decision trees and poor generalization. (Chio & Freeman, 2018)

3.6 Unsupervised learning

Unsupervised learning algorithms seeks information and patterns from data while, at the same time, sets the tuning parameter itself. The most common technique in the unsupervised learning category is clustering. Clustering algorithms tries to find similarities in the dataset and group these into meaningful clusters. (Müller & Guido, 2017)

3.6.1 K-Means clustering

The K-Means clustering has a predefined value, K, for number of clusters the data should be grouped into. K-Means is applied to real-valued vectors. K-Means algorithm will start with randomly defining K points in the dataset and using these as centroids. When

the initial K centroids has been defined it will start to assign all available data points to clusters. (Kirk, 2017)

The second step of the algorithm is to recalculate the distance of each data point and reassign the data point to the nearest cluster center. The algorithm wants to minimize the sum of distances for each point to the center of a cluster in the vector space. The distance to the cluster center for a data point can be calculated with e.g. Manhattan distance, Euclidean distance or Minkowski distance. Euclidean and Minkowski distances have been discussed in earlier sections. The mathematical representation of the Manhattan distance is shown in **Equation 30**. The loss function that is to be minimized is show in **Equation 31**. (Chio & Freeman, 2018)

$$d_{manhattan}(x,y) = \sum_{i=1}^{n} |x_i - y_i| \tag{30}$$

$$L(X) = \sum_{i} d\left(x_i, c_{f(x_i)}\right) \tag{31}$$

where

X: is the dataset $X = \{x_1, \ldots, x_n\}$.

c_j: is the jth cluster center.

d: is the distance between two points.

The k-Means algorithm is done when the assignment of data points to clusters is unchanged. A flowchart of the algorithm is shown in **Figure 30** (Burger, 2018)

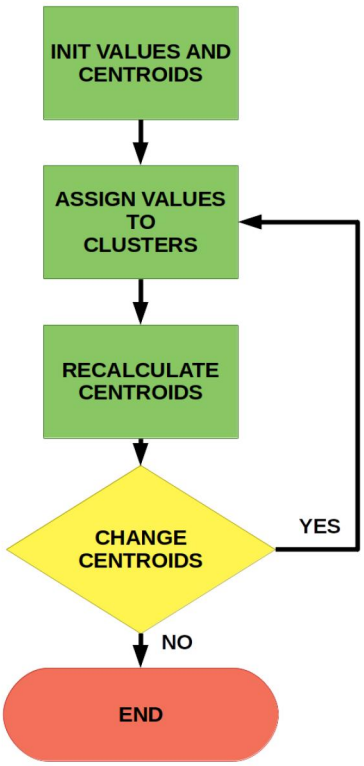

Figure 30: Flowchart of k-Means algorithm

3.6.2 DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is another popular clustering algorithm. DBSCAN does not rely on a predefined number of clusters, as in k-Means clustering. DBSCAN is searching grouping datasets according to high-density regions. A region in the feature space is referred to as a *dense* region. The DBSCAN algorithm assumes that in between dense regions there are regions that contains very little data. (Müller & Guido, 2017)

The user of the algorithm can define the search radius, ε, within which the neighbors will be searched for. Furthermore, the user can also set the minimum number of points that are needed to form a cluster, *minPoint*. With these parameters, it is possible to classify a data point in the feature space as either a *core point*, *border point* or *noise point* (Chio &

Freeman, 2018)

DBSCAN algorithm will start the classification with an arbitrary point in the feature space. It will count and the number neighbors based on the ε radius. A point is classified as *noise* if there is less than the defined *minPoint* in its neighborhood. On the contrary, if more points are found within the ε than the defined *minPoint*, then the data point will be labeled as a *core point*, tagged as *visited* and assigned a cluster label. The algorithm proceeds to evaluate the data points within ε and assigns them to the newly created cluster label, if the data point is unassigned. If the neighboring point is also a core point, its neighbors will be visited in turn. This process is ends when all available points have been visited. (Müller & Guido, 2017)

Running DBSCAN on the exact same dataset several times can produce different cluster membership for the border points while the core points and noise points will always obtain the same cluster membership. The reason why border points will be assigned different cluster membership is due to the fact that they can be neighbors to several core points of different clusters and in which order the border points are visited. (Müller & Guido, 2017)

One of the drawbacks with DBSCAN is that the user must have a very good understanding of the data's distribution and densities so that the ε and *minPoint* are properly chosen in order to maintain good performance. Another drawback is that the algorithm does not perform well on high-dimensional data since the Euclidean distance is used in the algorithm for calculating the distance between data points. (Chio & Freeman, 2018)

3.7 Other advanced methods

Affinity analysis is a general technique that will determine the relationship between objects. A very common application is market basket analysis, in which the goal is to find the products that are most frequently bought together. For example, those who have bought

item A have also bought item B etc. The resulting list can be used in marketing purpose or product placement planning. (Roiger, 2017)

The data needed for building recommendation systems is transaction data. In other words, the basic concept are users, items and feedback about the products. The feedback can be for example, ratings or the information if a user have bought a particular product. The training of the models is done with known data. Next, the author going to discuss three different approaches to market basket analysis. The first strategy is *content-based filtering*, the second one is *collaborative filtering* while the third one is the *apriori algorithm*. (Bonaccorso, 2017)

3.7.1 Content-based filtering

Content-based filtering generates subsets of similar items with respect to an item that have been bought or rated. The attributes of the bought item will be matched with other items that have some degree of similarity based on the attributes. To calculate the similarity we can use *cosine similarity* or use a clustering method like *k-nearest neighbors*. *Cosine similarity* calculates the cosine of the angle between two nonzero vectors of an inner product space, mathematical representation shown in **Equation 32**. The cosine distance is within $[0, 1]$ since it is used in a positive space. Unit vectors will obtain the highest similarity score while vectors that are perpendicular will obtain the lowest similarity score. **Figure 31** shows vector A and B. (Dangeti, 2017)

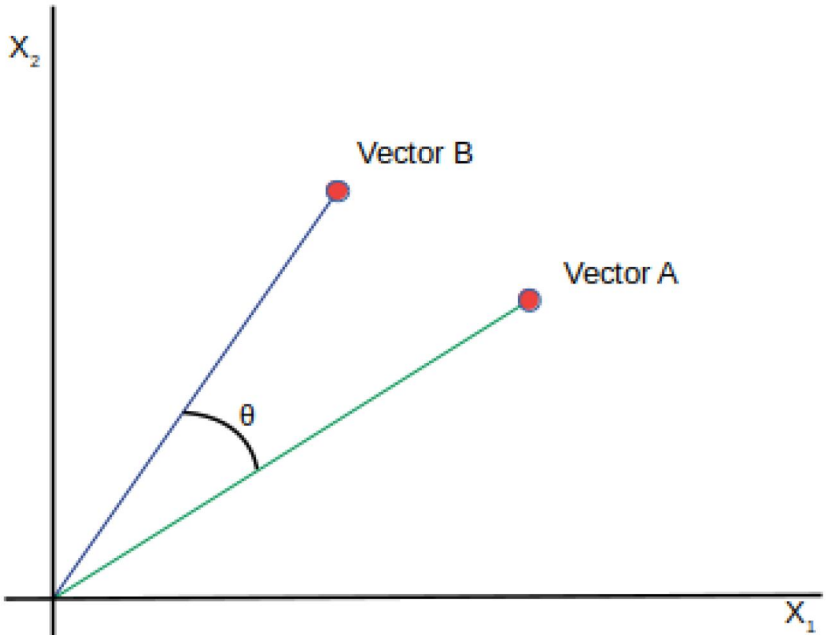

Figure 31: Cosine distance

$$cos(\theta) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{32}$$

where

A_i: are the components of vector A

B_i: is the components of vector B

Let us assume that the components of $A = [2,1,1,2,0,1]$ and $B = [2,1,1,1,1,0]$. The calculated cosine distance for these two vectors is $cos(\theta) \approx 0.86$. (Dangeti, 2017)

Another technique is to use k-nearest neighbors to build a content based recommendation system. It is possible to have the products as feature vectors, as shown in **Equation 33**. It is possible that some of the features are categorical and after encoding these values,

as described earlier in **Section 3.5.1**, they can be used in combination with numerical values. The clustering method k-nearest neighbors allows us to adjust the size of every neighborhood, thus allowing us to determine the quality and number of suggestions. In this case, the Minkowski distance, shown in **Equation 34**, is used for calculating the metric distance between the vectors. (Bonaccorso, 2017)

$$I = \{\bar{i_1}, \bar{i_2}, ..., \bar{i_n}\} \; where \; \bar{i_n} \in \mathbb{R}^n \tag{33}$$

$$d_{Minkowski} = \left(\sum_{i=1}^{n} |a_i - b_i|^p \right)^{\frac{1}{p}} \tag{34}$$

where

a_i: are the components of vector b

b_i: is the components of vector a

p: parameter for controlling the distance type

If the feature vectors on the other hand are binary, a disagree proportion can be calculated between the two vectors. Calculating the Hamming distance, it is possible to get the normalized number of different bits in two binary vectors. This information tells us the absence of particular features. Another method to use when we want to examine the dissimilarity of binary vectors is the Jaccard distance, shown in **Equation 35**. The Jaccard distance is bounded between 0 and 1, where 0 means that the two vectors are equal and 1 means that we have a case of total dissimilarity. **Figure 32** shows an example where the author calculates both the Hamming distance and Jaccard distance between two binary vectors. (Bonaccorso, 2017)

$$d_{Jaccard} = 1 - J(A,B) = 1 - \frac{|A \cap B|}{|A \cup B|} \tag{35}$$

where

A: is a binary vector

B: is a binary vector

```
In [14]: import numpy as np
    ...: from scipy.spatial.distance import hamming
    ...: from scipy.spatial.distance import jaccard
    ...:
    ...: #Create two binary vectors
    ...: A = np.array([0,1,0,1,1,1,0,0,1,0])
    ...: B = np.array([1,1,0,0,1,0,0,0,1,1])
    ...: #Calc. the Hamming distance
    ...: d_Ham = hamming(A,B)
    ...: #Calc. the Jaccard distance
    ...: d_Jacc = jaccard(A,B)
    ...: print('Hamming distance: %f' % (d_Ham))
    ...: print('Jaccard distance: %f' % (d_Jacc))
Hamming distance: 0.400000
Jaccard distance: 0.571429
```

Figure 32: Hamming and Jaccard distance of two binary vectors

3.7.2 Collaborative filtering

Collaborative filtering can be used when building recommendation engines. Collaborative filtering builds a model that estimates the preferences of a single user based on the shared or similar preferences of many users in a dataset. The collaborative filtering has several advantages over the content based technique that were discussed earlier. The main advantage is that collaborative filtering is dynamic, so it is capable of capturing changes in preferences of users. Furthermore, there is no so called *cold-start* problem since the ratings can still be predicted without the user buying the product. (Dangeti, 2017)

One optimization method, suited for recommendation engines, that have been proven to have good performance is the alternating least squares (ALS) method. ALS is a method for solving matrix factorization problems and it belongs to a class of latent-factor models. ALS aims to give an explanation of interactions between users and items using a small number of unobserved, underlying factors. (Dangeti, 2017)

The user-item matrix can be defined as in **Equation 36**. We assume that latent factors are present for both users and items, thus a generic user can be defined as in **Equation 37** and a generic item can be defined as in **Equation 38**. The ranking can be calculated by using **Equation 39**. From the latent space of size k, which is the number of latent variables to be included in the model, we can obtain the ranking. The underlying reason for implementing the ALS method in this type of problem is because the number of variables we need to find becomes very high, very easily. If we have 10000 users and 50 items then the user-item matrix, M, will have 500000 elements. Furthermore, having a rank of 10 will result in finding 5000000 variables constrained by known ratings. (Bonaccorso, 2017)

$$M_{UxI} = \begin{pmatrix} r_{11} & \dots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \dots & r_{mn} \end{pmatrix} \tag{36}$$

$$\bar{p_i} = \{p_{i1}, p_{i2}, \dots, p_{ij}\} \; where \; p_{ij} \in \mathbb{R} \tag{37}$$

$$\bar{q_j} = \{q_{j1}, q_{j2}, \dots, q_{jt}\} \; where \; q_{jt} \in \mathbb{R} \tag{38}$$

$$r_{ij} = \bar{q_i} \cdot \bar{q_j}^T \tag{39}$$

By defining the loss function shown in **Equation 40**, we can express the problem of finding latent factors as a least square optimization problem. L is bounded only to the known samples. The two main iterating steps of ALS algorithm can be expressed as:

- p_i is fixed while q_j is optimized
- q_j is fixed while p_i is optimized

The iteration will stop when a predefined threshold has been reached. If we have matrices with high dimensions, the algorithm can be implemented with parallel strategies for faster execution time. (Bonaccorso, 2017)

$$L = \sum_{i,j} \left(r_{ij} - \bar{p}_i \cdot \bar{q}_j{}^T\right)^2 + \alpha \left(\| \bar{p}_i \|^2 + \| \bar{q}_j \|^2\right) \tag{40}$$

3.7.3 Apriori algorithm

This approach on the market basket analysis is an iterative process that creates association rules between items in a dataset. *Support* is a key term in the Apriori algorithm, it gives the percentage of how many of the transactions that contains a certain itemset M. The first step in the iterative process determines all the possible 1-itemsets, e.g. {{milk}, {butter}, {yogurt}, ...}. The algorithm is working from the bottom of the dataset up to the beginning while at the same time trying to identify which items that frequently occurs in the registered transactions. The algorithm will discard itemsets that does not meet the minimum support threshold value. A minimum support threshold of 0.5 will only leave itemsets that appears in more than 50% of the transactions. The second iteration will pair the frequent 1-itemsets into 2-itemsets, e.g. {{milk, butter}, {milk, yogurt}, {butter, yogurt}, ...} while trying to find the frequent 2-itemsets based on the support criterion.

This will continue until a predefined length is reached or until the algorithm runs out of support, see **Algorithm 1**. (Dietrich, 2015)

Algorithm 1 Apriori algorithm

```
APRIORI(D, δ, N)
k ← 1
M_k ← {1-itemsets greater than min. support δ}
while M_k ≠ ∅ do
  if ∄ N ∨ (∃ ∧ k < N) then
    C_{k+1} ← {itemsets derived from M_k}
    for each transaction t in database D do
      increment the counts of C_{k+1} contained in t
    end for
    M_{k+1} ← candidates in C_{k+1} that satisfy δ
    k ← k + 1
  end if
end while
return ⋃_k M_k
```

Confidence is the second key measurement in the Apriori algorithm and it is used in the evaluation process of the association rules. The confidence is indication of certainty for every association rule that the Apriori algorithm has discovered. In short, it is the percent of transactions in the dataset that contain item A and item B given that a transaction contain item A, see **Equation 41**. (Baesens, 2014)

$$Confidence(A \rightarrow B) = \frac{Support(A \wedge B)}{Support(A)} \tag{41}$$

Lift is another measurement used in the evaluation process. Lift value for the association rules is a measure on the relationship between itemset A and itemset B. If the lift value

is close to 1, then A and B are considered to be statistically independent. However, a large lift value indicates that there is a strong association between two items and that the association rule has some useful information. The mathematical formula for lift can be seen in **Equation 42**. (Dietrich, 2015)

$$Lift(A \rightarrow B) = \frac{Support(A \wedge B)}{Support(A) * Support(B)} \tag{42}$$

4 CASES STUDY

4.1 Warehouse arrangement based on association rules

Available from VEO's internal databases we have the bill of material (BOM) table that is automatically updated from the different design tools used in the company. The BOM interface is a transaction database that contains the BOM of the products for every project. Every product that is to be built in the factory, e.g. different types of switchgear or control panels, have the required parts aggregated together in the BOM database table. Requested spare parts are also found in the same table. It is worth mentioning that all the parts listed in the BOM interface may not necessarily be stored in VEO's warehouse, but the majority of the parts are. Data entries are available from the year 2017 and forward.

With the data available from the BOM interface, it could be possible to use association rule learning to optimize placement of parts in the warehouse. A warehouse layout suggestion could be generated using the Apriori method discussed in **Section 3.7.3**. In **Section 3.7.3** the author used transactions from a store to described the functionality and the mining process of Apriori algorithm in order to find the associations between bought products. However, it could also be implemented to learn the association between the parts requested for the assembly of products in the factory.

In other words, we want the algorithm to learn association between the parts. With this information, it is possible to form an general picture of parts that often are requested together and what kind of association they have to each other. The discovered insights can then be used for optimizing the placement of the parts in the warehouse.

In the following sections, the author will implement data mining techniques that will produce a list of association rules between the requested parts.

4.1.1 APRIORI

The author has decided to solve this data mining problem using the programming language R. To start with, the exported data from the BOM table needs to be transformed. The data exported from the database is organized so that each part is printed on its own row, see **Figure 33**. With this structure, we will have several rows that belongs to the same project. The desired output is to have one project per row and the parts in the columns. The source data has been exported to an Excel file and later been transformed and saved as a comma-separated delimited (CSV)file.

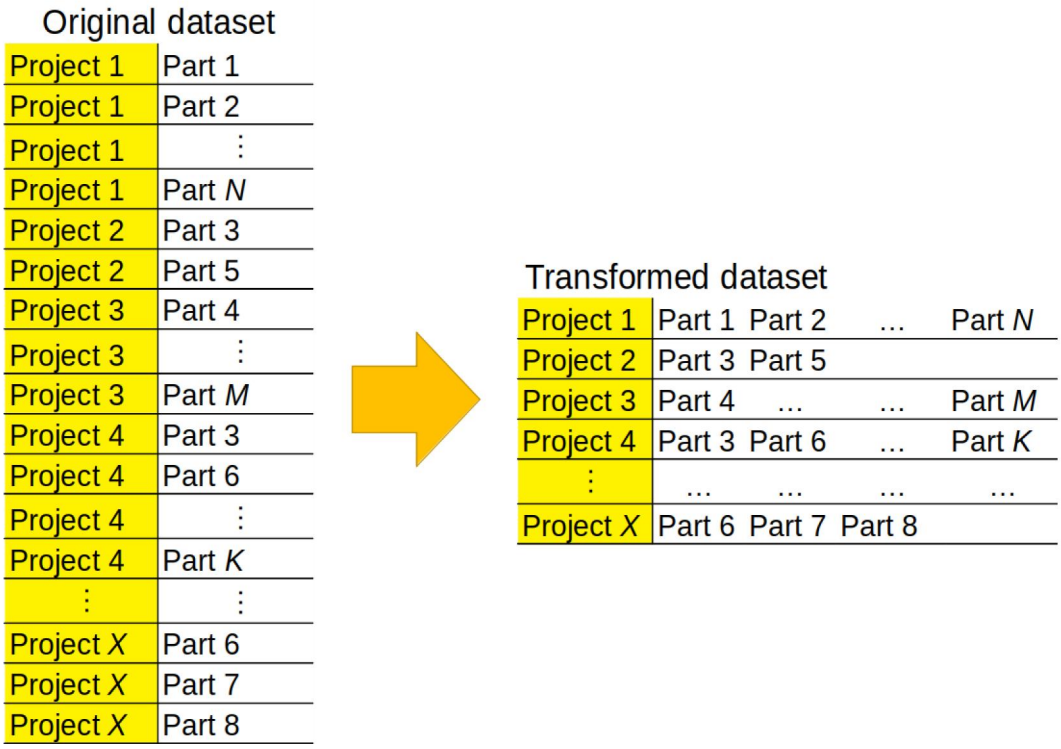

Figure 33: Transforming the data before implementing data mining techniques

After completing the data transformation, we can proceed with the analysis of the dataset. The R-script that has been used to solve this problem can be seen in **Appendix B**.

The source code is dependent on two libraries, *arules* and *arulesViz*, which must be installed and imported to the R programming environment. These libraries contain functions

that will facilitate the mining of the frequent itemsets and visualize the results. The data will be loaded as a sparse matrix into R environment.

Removing duplicates from the dataset is an important step and during the data import stage, and a Boolean value can be passed with the *read.transactions()* function to eliminate duplicate entries. This is the only data preprocessing that have be performed on the dataset and it is not necessary to any other technique discussed in **Section 3.2**. After that the data has been loaded into R we can obtain detailed information regarding the dataset.

The *summary()* function will return information about the sparse matrix. The density value of the sparse matrix is the proportion of non-zero values in the dataset. Other useful information from the output of the *summary()* is the *element length distribution.* This aggregates the data by counting how many parts are requested for each transaction, in this case projects. The author will refer to the **Figure 34**, where it can be seen that 1 project have only 1 part requested, 2 projects with 13 parts requested etc. Furthermore, from the same output we can obtain minimum number of parts (*Min.*), maximum number of parts (*Max.*) and the average number of parts (*Mean.*) requested.

```
## transactions as itemMatrix in sparse format with
##  128 rows (elements/itemsets/transactions) and
##  2322 columns (items) and a density of 0.07996205
##
## most frequent items:
##     S2C-H6R  NI40500001 TS_35/F6_Sz   5.134.131    WEW 35/2     (Other)
##         121         118         104          99          96       23228
##
## element (itemset/transaction) length distribution:
## sizes
##   1  13  16  17  21  24  25  26  28  29  30  33  35  36  39  40  43  46
##   1   2   1   3   3   3   4   6   2   2   2   1   2   2   1   1   1   1
##  51  53  54  62  73  88 115 116 126 142 156 165 167 171 179 192 193 197
##   1   2   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 204 205 206 208 209 210 228 236 237 241 244 245 246 256 258 260 261 265
##   1   1   1   1   3   1   1   1   1   1   1   1   1   2   1   1   1   1
## 270 273 277 280 281 285 287 288 289 291 294 295 296 299 300 301 302 303
##   1   4   1   1   2   1   1   2   1   1   1   1   3   1   1   1   1   1
## 306 308 309 310 311 313 314 315 316 319 324 329 332 334 341 351 352 363
##   1   1   1   1   2   1   1   1   1   1   2   3   1   1   1   1   1   1
## 375 377 433
##   1   1   1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   35.75  209.50  185.67  296.00  433.00
##
## includes extended item information - examples:
##         labels
## 1     010010TD
## 2 075-8519-000
## 3 075-8520-000
```

Figure 34: Summary of BOM data collected from internal database

The importance of EDA was discussed in **Section 3.2** and therefore it is necessary to create a frequency plot of the available parts to form an initial understanding of the dataset. The code snippet shown in **Figure 35** will generate a plot that lists the top parts in descending order according to their relative frequency, the output can be seen in **Figure 36**. From the frequency plot we can get some hints about what the minimum support value could be set to. It is recommended not to set the minimum support too close to the maximum support value since this will affect the number of rules mined.

```
itemFrequencyPlot(dataset, topN = 10, xlab = "Parts")
```

Figure 35: Code snippet to print most frequent items

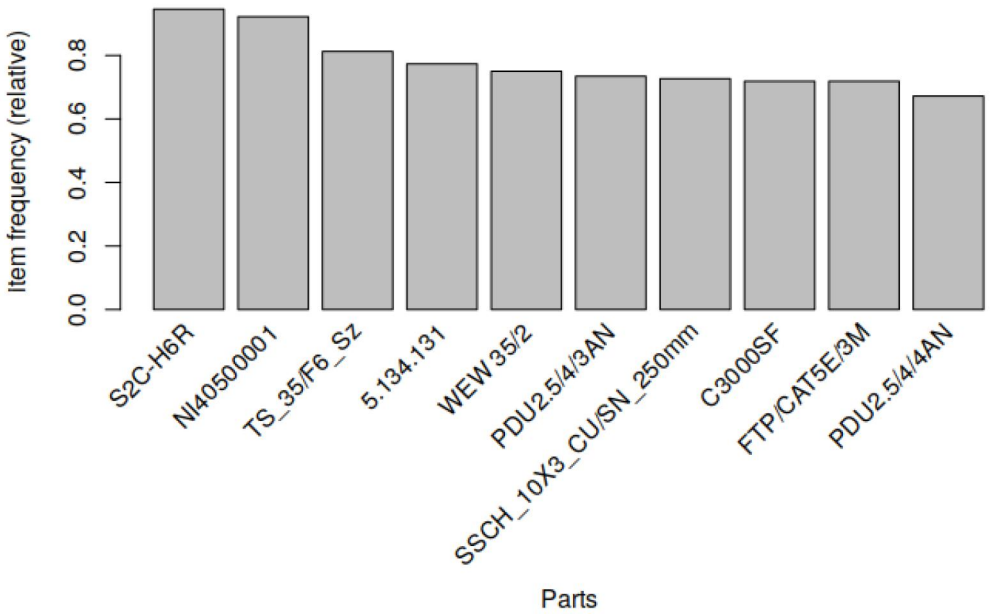

Figure 36: The most frequent items in the BOM

The actual association rules will be generated using the *apriori()* function. Different values of the minimum support and the confidence will have an impact on how many association rules that will be generated. In this particular example, the author has set the minimum support value to *0.65* and the confidence threshold to *0.7*, as seen in **Figure 37** and the corresponding output of the mining process can be seen in **Figure 37** which also prints the information of how many rules that has been generated with the given thresholds parameters.

```
rulesets = apriori(data = dataset, parameter = list(support = 0.65, confidence = 0.7,
minlen = 1))
```

Figure 37: Code snippet to print most frequent items

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.7    0.1    1 none FALSE            TRUE       5    0.65      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 83
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[2322 item(s), 128 transaction(s)] done [0.01s].
## sorting and recoding items ... [12 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.00s].
## writing ... [338 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

Figure 38: Output from the model learning

As discussed in **Section 3.7.3**, lift is a measurement on how useful an association rule is, where a higher value indicates that two itemsets have strong association to each other. In **Figure 39** the association rules are plotted as an scatter plot with the confidence as a function of the support. In this case, from the visualization of the rules, we can draw the conclusion that the highest lift occurs where we have a high confidence and relatively low support.

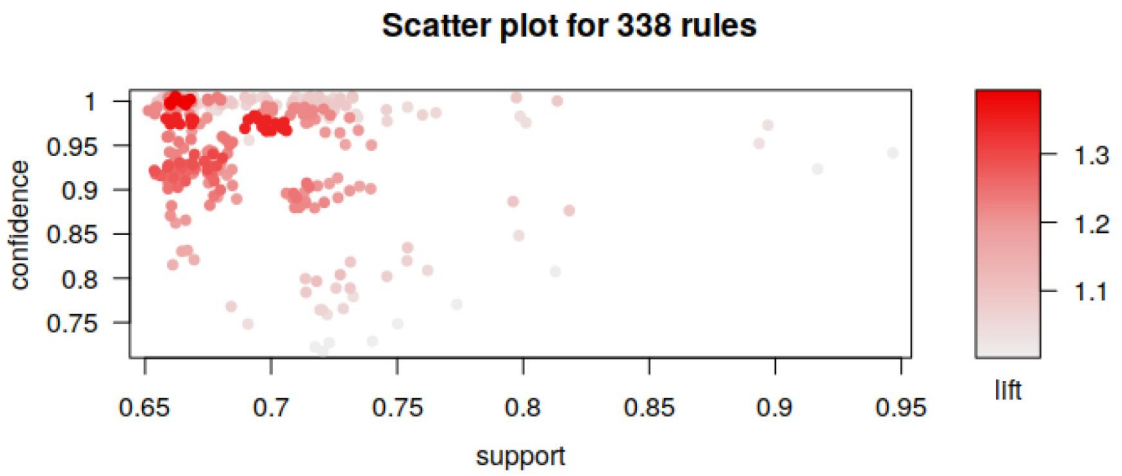

Figure 39: Scatterplot of the association rules

Appendix C lists the mined association rules generated by applying Apriori algorithm on the dataset extracted from from the BOM. **Appendix C** is the raw, unsorted output of all the mined association rules. **Appendix D** lists the association rules that has a higher confidence value than 0.97. By looking at output in **Appendix D** we can see the LHS and RHS parts that have a high confidence value. Furthermore, in **Appendix E** the author has listed the parts with the highest lift.

To be able to compare the support, confidence and the lift of the association rules we need to plot the scatterplot matrix of these features. **Figure 40** illustrates these properties of the generated association rules and we can clearly see linear groupings in the plot. From **Equation 41** and **Equation 42** we get **Equation 43**. This indicates that when the support value of B is unchanged, the lift will be proportional to confidence. Furthermore, this implies that the slope of the linear trend is a reciprocal of the *Support(B)*. The slope values and their occurrences can be calculated using the code snippet seen in **Figure 41**. It can be concluded that we only have 8 different slope values and we can see that the slope values matches the subplot at the last row and second column in **Figure 40**

$$\begin{aligned} Lift &= \frac{Confidence}{Support(B)} \\ &= \frac{1}{Support(B)} \end{aligned} \tag{43}$$

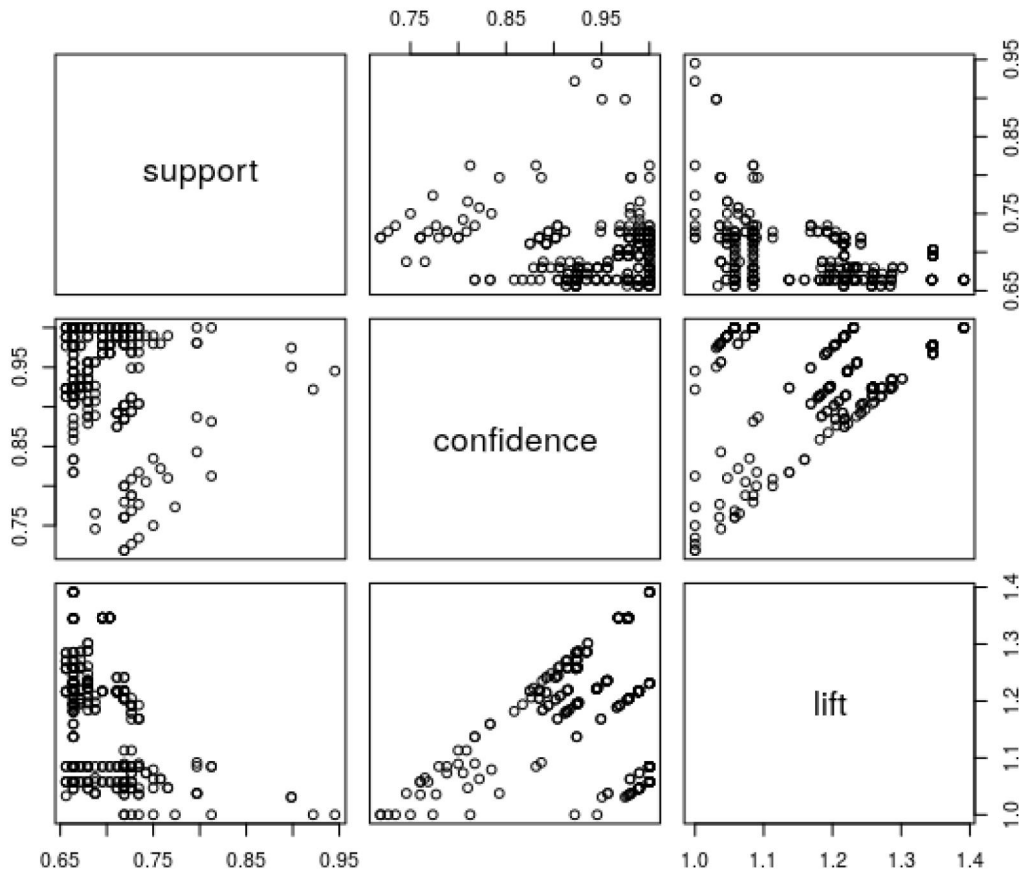

Figure 40: Comparison of support, confidence and the lift values

```
slope = sort(round(rulesets@quality$lift / rulesets@quality$confidence, 2))
unlist(lapply(split(slope,f=slope),length))
```

```
## 1.06 1.08 1.23 1.29 1.33 1.36 1.38 1.39
##   58   56   56   40   16   24   40   48
```

Figure 41: Computing the slope values of the linear groupings

To have a look at the rules with a very high confidence, a threshold value must be set for this interest measure. These association rules can be obtained by using the code snippet seen in **Figure 42**. This will provide a list of the rules with higher confidence than the required threshold. The resulting rulesets can be examined with the help of a matrix-based visualization, seen in **Figure 43**. The matrix-based visualization plots the antecedent and consequent itemsets on the x and y-axis respectively. At the intersection of the antecedent and consequent of an association rule is an interest measure. A blank area is an indication of non-existing antecedent-consequent combinations.

```
confidentRules1 = rulesets[quality(rulesets)$confidence > 0.97]
```

Figure 42: Obtaining rules with high confidence

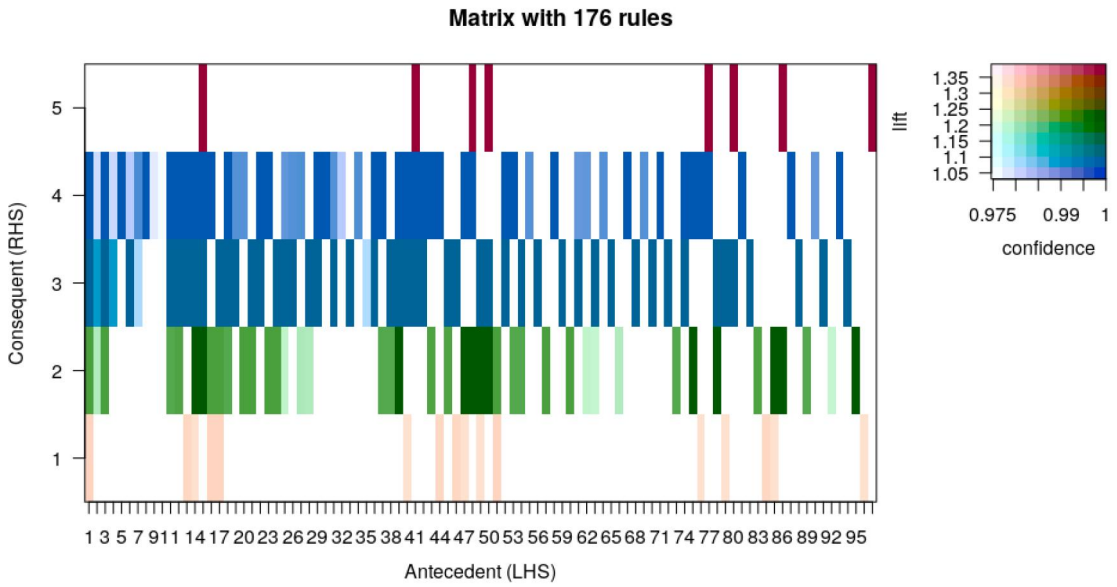

Figure 43: LHS and RHS with mapped to color coded lift and confidence values

Furthermore, we can filter out the rules with the highest lift. As discussed in **Section 3.7.3**, lift values that is greater than the value one have a stronger association. With the help of a graph plot, not only can we visualize the individual items that a rule is built from, but also the shared items between different rules. The graph consists of vertices and

edges. Items are connected to rules or itemsets with directed arrows, which makes up the LHS of the rule. Arrows that goes from a rule to an item represents the RHS. To map the interest measures to the rules in the graph plot, lift will be presented by color and support is represented by size.

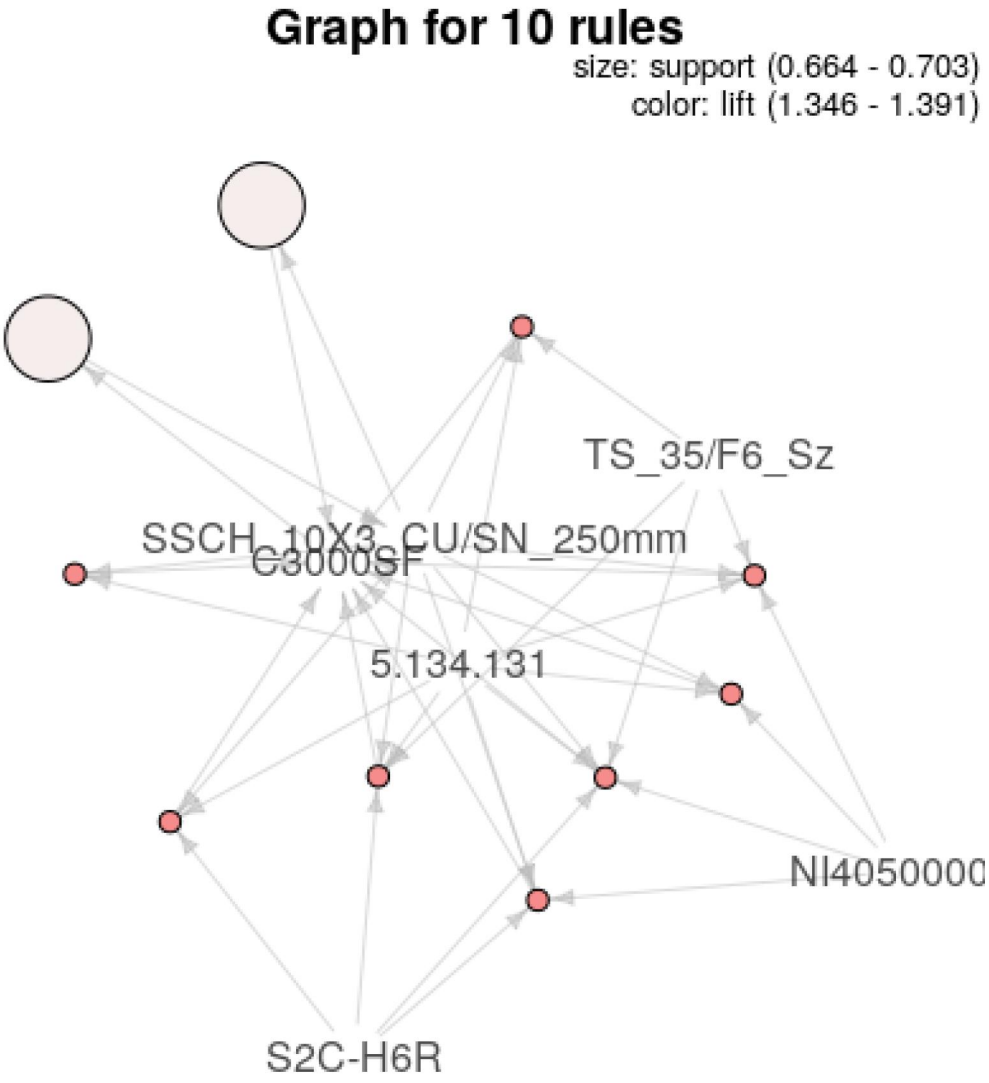

Figure 44: Graph-based visualization of the 10 rules with highest lift

4.1.2 ECLAT

Equivalence class clustering and bottom-up lattice traversal (ECLAT) algorithm is a more trivial way to approach this task. ECLAT has only the support value as interest value, i.e. the other values like lift and confidence is not available. ECLAT focuses more on the frequency of itemsets, that is how often a set of items occurs in the dataset. It is necessary to remind the reader that one itemset can contain several items.

The ECLAT algorithm relies only on the minimum support value and it will generate the most frequent itemsets based on the support value and minimum length value. It is, in some sense, a more trivial algorithm to work with since there are only the minimum support and the minimum length parameters that can be adjusted. **Appendix F** contains the source code for solving this problem with the ECLAT algorithm. The generated itemsets are available in **Appendix G**.

The itemsets produced by the ECLAT algorithm is not as interesting as the association rules that are created with Apriori algorithm since we are lacking the interest values. But the output from the ECLAT algorithm is still useful and some knowledge can be obtained from the most frequent requested itemsets.

5 RESULTS

The case study that the author conducted was to research how data from the design stage of a project could be used for warehouse rearrangement. The outcome of the case study, discussed in **Section 4.1**, was a list of itemsets that are frequently requested together and that have some degree of association to each other.

The association rules that have been mined using the Apriori algorithm, could be used to rearrange the warehouse in such a way that parts with high confidence and lift values are placed closer to each other. This would minimize the time it takes to gather the needed parts that for the assembly line.

Furthermore, the itemsets in the list of the association rules could also be used as a part of an add-in for the design software that are used when designing control panels, MV or LV switchgear. The add-in could suggest related parts depending on what parts that have recently been added.

The author also decided to try a more trivial algorithm that would only list the most frequent itemsets in the available data, discussed in **Section 4.1.2**. The list of itemsets produced by the Eclat algorithm did not calculate interest values, such as lift and confidence, and the output can therefore not replace results of the Apriori algorithm.

One suggestion of improvement to the R-script would be to implement an iterative method that would mine new association rules based on a time window while neglecting previously mined time windows. As an example, the association rules list could be generated once per month or even once per quarter, however, the size of the time window is directly related to the application of the mined information. An add-in for design software could benefit from a more frequent mining process compared to rearranging the warehouse. A flowchart of the process can be seen in **Figure 45**.

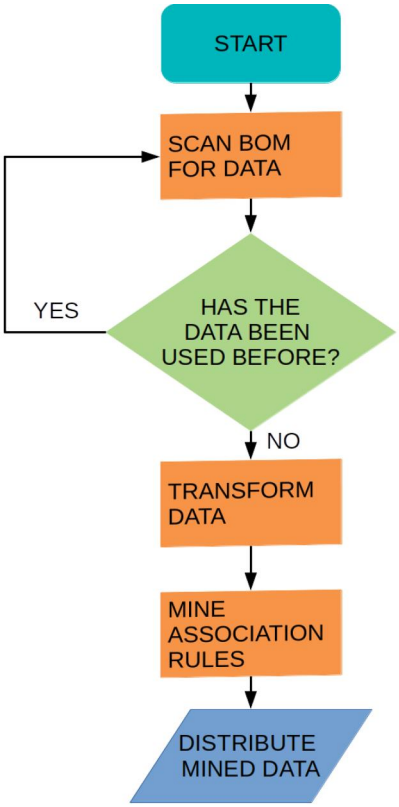

Figure 45: Flowchart of an improved data mining process

6 CONCLUSION

The employer gave the author the task to research the topics of data mining and machine learning. The main goal of this thesis was to research how it could be possible to implement data mining techniques and machine learning algorithms utilising the project related data that is generated during projects at VEO Oy and to present a use case where the insights gained from the case study could lead to improvements within the company.

A fair amount of theory has been discussed around various topics related to data mining and machine learning that has eventually enabled the author to find a suitable use case of the theory. Even though some of the techniques and methods discussed in the theory part of the thesis were not implemented in the case study, the author has gained knowledge about the topics to further develop the task for the employer. The outcome of the thesis was a case study on how data generated from projects, in this case the content of the BOM, could help with rearranging the warehouse using data mining techniques. The results provided some new insights about sets of parts that are frequently requested together from the warehouse and which parts that should be placed near each other to optimize the arrangement of the warehouse.

According to the author's opinion, a topic that would be very interesting for further research would be how to implement a data warehouse structure or, preferably, analytical sandboxes specifically design for data mining. The fragmented structure of the databases makes the gathering of data in some sense very complicated and to fully harness the possibilities of data mining an analytical sandbox would be needed. The research could address how one could implement a data warehouse or an analytical sandbox and examine what kind of solutions that are available. The data warehouse structure and analytical sandboxes was briefly discussed in **Section 2.4** but it would require a thorough research to understand how to properly implement such solutions.

REFERENCES

Baesens, B. (2014). *Analytics in a big data world: The essential guide to data science and its applications*. John Wiley & Sons.

Bonaccorso, G. (2017). *Machine learning algorithms*. Packt Publishing.

Bonnin, R. (2017). *Machine learning for developers*. Packt Publishing.

Burger, S. V. (2018). *Introduction to machine learning with r: Rigorous mathematical analysis*. O'Reilly Media, Inc.

Chio, C., & Freeman, D. (2018). *Machine learning and security: Protecting systems with data and algorithms*. O'Reilly Media, Inc.

Cisco. (2019). *Cisco visual networking index: Forecast and trends, 2017-2022*. `https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html`. (Accessed: 2019-09-11)

Dangeti, P. (2017). *Statistics for machine learning*. Packt Publishing.

Dietrich, D. (2015). *Data science & big data analytics: discovering, analyzing, visualizing and presenting data*. Wiley.

Google. (2019). *Big data search trend world wide*. `https://trends.google.com/trends/explore?date=all&q=Big%20Data`. (Accessed: 2019-01-29)

Han Jiawei, P. J., Kamber Micheline. (2012). *Data mining: Concepts and techniques*. Morgan Kaufmann.

Iafrate, F. (2015). *From big data to smart data* (Vol. 1). John Wiley & Sons.

Joshi, P. (2017). *Artificial intelligence with Python*. Packt Publishing.

Kirk, M. (2017). *Thoughtful machine learning with Python: A test-driven approach*. O'Reilly Media, Inc.

Marr, B. (2015). *Big data: Using SMART big data, analytics and metrics to make better decisions and improve performance*. John Wiley & Sons.

Müller, A. C., & Guido, S. (2017). *Introduction to machine learning with Python: a guide for data scientists*. O'Reilly Media, Inc.

O'Neil, C., & Schutt, R. (2014). *Doing data science: Straight talk from the frontline*.

O'Reilly Media, Inc.

Roiger, R. J. (2017). *Data mining: a tutorial-based primer*. Chapman and Hall/CRC.

Schmarzo, B. (2016). *Big data mba: Driving business strategies with data science*. John Wiley & Sons.

Simon, P. (2013). *Too big to ignore: the business case for big data* (Vol. 72). John Wiley & Sons.

VEO. (2019). *Company*. `https://www.veo.fi/company/`. (Accessed: 2019-11-11)

Williams, G. J. (2017). *The essentials of data science: knowledge discovery using R*. Chapman and Hall/CRC.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Zheng, A., & Casari, A. (2018). *Feature engineering for machine learning: principles and techniques for data scientists*. O'Reilly Media, Inc.

APPENDICES

Appendix A 50 Start-ups

Table 11: 50 fictional startup companies

R&D Spend	**Administration**	**Marketing Spend**	**State**	**Profit**
165349.2	136897.8	471784.1	New York	192261.83
162597.7	151377.59	443898.53	California	191792.06
153441.51	101145.55	407934.54	Florida	191050.39
144372.41	118671.85	383199.62	New York	182901.99
142107.34	91391.77	366168.42	Florida	166187.94
131876.9	99814.71	362861.36	New York	156991.12
134615.46	147198.87	127716.82	California	156122.51
130298.13	145530.06	323876.68	Florida	155752.6
120542.52	148718.95	311613.29	New York	152211.77
123334.88	108679.17	304981.62	California	149759.96
101913.08	110594.11	229160.95	Florida	146121.95
100671.96	91790.61	249744.55	California	144259.4
93863.75	127320.38	249839.44	Florida	141585.52
91992.39	135495.07	252664.93	California	134307.35
119943.24	156547.42	256512.92	Florida	132602.65
114523.61	122616.84	261776.23	New York	129917.04
78013.11	121597.55	264346.06	California	126992.93
94657.16	145077.58	282574.31	New York	125370.37
91749.16	114175.79	294919.57	Florida	124266.9
86419.7	153514.11	0	New York	122776.86
76253.86	113867.3	298664.47	California	118474.03
78389.47	153773.43	299737.29	New York	111313.02
73994.56	122782.75	303319.26	Florida	110352.25
67532.53	105751.03	304768.73	Florida	108733.99

Continued on next page

Table 11 50 fictional startup companies (Continued)

R&D Spend	Administration	Marketing Spend	State	Profit
77044.01	99281.34	140574.81	New York	108552.04
64664.71	139553.16	137962.62	California	107404.34
75328.87	144135.98	134050.07	Florida	105733.54
72107.6	127864.55	353183.81	New York	105008.31
66051.52	182645.56	118148.2	Florida	103282.38
65605.48	153032.06	107138.38	New York	101004.64
61994.48	115641.28	91131.24	Florida	99937.59
61136.38	152701.92	88218.23	New York	97483.56
63408.86	129219.61	46085.25	California	97427.84
55493.95	103057.49	214634.81	Florida	96778.92
46426.07	157693.92	210797.67	California	96712.8
46014.02	85047.44	205517.64	New York	96479.51
28663.76	127056.21	201126.82	Florida	90708.19
44069.95	51283.14	197029.42	California	89949.14
20229.59	65947.93	185265.1	New York	81229.06
38558.51	82982.09	174999.3	California	81005.76
28754.33	118546.05	172795.67	California	78239.91
27892.92	84710.77	164470.71	Florida	77798.83
23640.93	96189.63	148001.11	California	71498.49
15505.73	127382.3	35534.17	New York	69758.98
22177.74	154806.14	28334.72	California	65200.33
1000.23	124153.04	1903.93	New York	64926.08
1315.46	115816.21	297114.46	Florida	49490.75
0	135426.92	0	California	42559.73
542.05	51743.15	0	New York	35673.41
0	116983.8	45173.06	California	14681.4

Appendix B Apriori - R code optimizing warehouse

This page has been intentionally left blank.

Appendix C Apriori - Mined association rules

Table 12: Mined association rules

Rules	**Confidence**	**Lift**

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

Appendix D Apriori - Association rules with high confidence

Table 13: Association rules with high confidence

Rules	**Confidence**	**Lift**

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

Appendix E Apriori - Association rules with high lift

Table 14: Association rules with high lift

Rules	**Confidence**	**Lift**

This page has been intentionally left blank.

This page has been intentionally left blank.

Appendix F Eclat - R code

This page has been intentionally left blank.

Appendix G Eclat - Mined itemsets

Table 15: itemsets generated with ECLAT

Items	**Support**

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.

This page has been intentionally left blank.