Modern Data Mining for Software Engineer, A Machine Learning PaaS Review

Master of Science Thesis
University of Turku
Department of Future Technologies
Software Engineering
2020
Marko Loponen

UNIVERSITY OF TURKU
Department of Future Technologies

MARKO LOPONEN: Modern Data Mining for Software Engineer, A Machine Learning PaaS Review

Master of Science Thesis, 77 p.
Software Engineering
November 2020

Using data mining methods to produce information from the data has been proven to be valuable for individuals and society. Evolution of technology has made it possible to use complicated data mining methods in different applications and systems to achieve these valuable results. However, there are challenges in data-driven projects which can affect people either directly or indirectly. The vast amount of data is collected and processed frequently to enable the functionality of many modern applications. Cloud-based platforms have been developed to aid in the development and maintenance of data-driven projects. The field of Information Technology (IT) and data-driven projects have become complex, and they require additional attention compared to standard software development.

On this thesis, a literature review is conducted to study the existing industry methods and practices, to define the used terms, and describe the relevant data mining process models. We analyze the industry to find out the factors impacting the evolution of tools and platforms, and the roles of project members. Furthermore, a hands-on review is done on typical machine learning Platforms-as-a-Service (PaaS) with an example case, and heuristics are created to aid in choosing a machine learning platform. The results of this thesis provide knowledge and understanding for the software developers and project managers who are part of these data-driven projects without the in-depth knowledge of data science.

In this study, we found out that it is necessary to have a valid process model or methodology, precise roles, and versatile tools or platforms when developing data-driven applications. Each of these elements affects other elements in some way. We noticed that traditional data mining process models are insufficient in the modern agile software development. Nevertheless, they can provide valuable insights and understanding about how to handle the data in the correct way. The cloud-based platforms aid in these data-driven projects to enable the development of complicated machine learning projects without the expertise of either a data scientist or a software developer. The platforms are versatile and easy to use. However, developing functionalities and predictive models which the developer does not understand can be seen as bad practice, and cause harm in the future.

Keywords: data mining, machine learning, software development, data scientist, software developer

Contents

List of Figures

List of Tables

1 Introduction

The field of Information Technology (IT) has received its fair share of bad karma from the unsuccessful software development projects in its history. These have caused people working in this field to search for the most optimal solutions and methods to develop software products and manage these projects as well as possible. All of these solutions and methods aim not to waste the time of the developers and provide the stakeholders with the product they need. Finding an optimal solution to develop or maintain projects can provide better quality end-products, making them cost-efficient and making them more maintainable. Furthermore, as for the projects which consist of gathering and processing data, finding a solution for these data-driven project is a challenging task as it consists of the burden from previously used data mining process models, modern cloud-based environments and roles of a development team with different expertise in agile development.

The IT field has been advanced in great length; it has caused our society to thrive in many ways by providing ways to help people in their everyday tasks. The solutions implemented in a wide variety of industries have significantly improved the quality of life. These solutions transfer and create an enormous amount of data frequently. The methods for data processing have evolved since the dawn of computers, and in 2020 we have capability and knowledge to solve complex problems containing vast amounts of data.

The data have always been collected in some form in the computer era, and traditional data mining has given information about the nature of the collected data. The traditional

data mining consists of process models which give easy to follow steps and procedures to ensure the quality of the data handling during the whole project life cycle. These data-driven projects are not limited to a specific field or industry. From self-learning cars to smart devices, they try to find solutions to different problems by gathering data. Processing the data in different ways gives methods to predict and receive new knowledge.

Data Mining (DM), Artificial Intelligence (AI) and Machine learning (ML) are becoming part of a common software development project. For succeeding in these projects, cloud-based platforms have been developed to ease the manual labour of the developers by automating tasks as much as possible.

A cloud-based platform which has necessary functionalities such as version control, data storage, and automated processes, can be used by a data scientist and a software engineer to make it possible to develop these data-driven projects. In these projects, data scientist and software engineer are the two roles which are responsible for the project's outcome and results. As they have their methods and tools to practice their expertise, the platform combines outcomes to be used in the project. The platform can be sophisticated enough to enable either role to use it without the other role present. This can be a pitfall in a case the platform is not understandable enough for the part the user is not accustomed to.

The process models and roles of project members for handling the data have also evolved over the years. Everything can not be automated, and the human factor is still in the centre of developing new methods and tools to take advantage of the data which would go to waste without proper processes. For succeeding in bringing value from the gathered data, right tools, people, and expertise are required. When computers began to be common in the business field, we experienced a boom in data mining. Process models used in data mining became insufficient for modern agile software development. Nevertheless, the data mining process models provide valuable insights and mindset for handling the data in a consistent way to avoid pitfalls and risks.

The motivation for this research is to find important elements when developing data-driven projects. The data mining process models are introduced for the new data scientists and software developers, yet they are not taking into account modern agile software development. The pitfalls and risks are crucial to know beforehand for the software developer to avoid them. The data-driven projects will keep increasing in the future; hence it is necessary for current software developers and project managers to prepare for the projects which revolve around data.

1.1 Research Questions

In this thesis, we are conducting research about these presented data mining process models insufficiency in the modern era where the data is frequently collected, and software development is happening in a fast tempo. We try to find the connection and overlapping between the two main roles involving data-driven projects. When Platform-as-a-Services (PaaS) has been created to ease the development of this kind of projects. The combination of data mining process models, roles, and platforms create an interesting environment, and we conduct research for finding the choices to make during these projects.

A data scientist requires appropriate knowledge for handling the data, and easy to follow procedures to do their task successfully. The data mining process models have provided support on these matters.

A software developer has a comprehensive job description, and they are required to know many different tools and frameworks. They are required to have the ability to maintain all the software and systems they develop. To achieve this, the standard solution for development is to enable automation with PaaS.

PaaS provides concrete pipelines and tools to deploy faster and closely monitor the behaviour of the development software. Furthermore, from both of these viewpoints, we examine what the overlapping tasks and obligations are for these two roles to find out are

the requirements for the modern data scientist and software developer.

The research questions on this thesis are

1. How data mining has been implemented in modern software development?
2. How well data scientist and software developer work together in same projects?
   (a) What are the factors enabling trouble-free work for the software developer and data scientist?
3. How well machine learning platforms support software development on integrating machine learning into a software project?
4. What are the choices to make during the data-driven IT project?

This thesis provides valuable information for a software developer delve into the realm of data science, or data scientist getting their feet wet in the pool of software development.

Research question 1: We provide information about the factors which have caused the change of data processing in modern IT systems. Moreover, we study the state of the art platforms used in modern machine learning - examining their abilities to evaluate how well they take into account the previously used Data Mining process models and methods. The data is in the central role in several IT systems and the decisions which guide the business is based on the data and its derivatives. As the software developer is developing the systems and capabilities of them to ensure proper data handling and understanding, the software developer is moving into the realm of a data scientist.

Research question 2: Also, research is conducted about the similarities of these two different important roles, data scientist and software developer. These two are the key players in any data-focused software development, as each of them provides their knowledge and expertise to the project. However, the environment to develop, and maintain software products and systems have become a complicated job. Data scientist and software developer is demanding certain functionalities to enable the work to be done. Hence,

it has been a natural evolution of the industry to come up with tools and platforms to ease, and automate the development of these data-focused systems.

Research question 3: By analyzing the common platforms for practising Machine Learning in the cloud environments, we can find out how understandable and transparent they are for the users. Transparency is becoming more relevant in modern society as users and people are demanding justification for the decisions made by the computer. These kind of applications are affecting millions of people around the globe continuously. The systems vary in their nature, and each one can use one or more of the several learning methods[1] to learn itself to fulfil the wanted task as accurately as possible.

Some of the developed machine learning models might require justification for their predictions, and depending on the methods used the justification could be nearly impossible to give. This compels data scientists and software developers to understand the inner methods used in a machine learning model, and there is no room for not understanding the used platforms or tools. We create limited heuristic for choosing an optimal platform from the viewpoint of a software developer to minimize problems and technical debt in the future. These created heuristics helps the developers to learn the factors needed for producing quality systems and software.

Research question 4: As the majority of IT projects revolve around data; we evaluate what is the feasible approach for building the modern data-focused IT projects to ensure the quality and robustness of the end product. To achieve a good end product, the project needs to have; methods to understand the data, competent people, and tools to build, test, and monitor the behaviour of data before and after deployment. This falls back into the previous three research questions for summarizing the best insights of each one of them.

[1] Unsupervised learning, Supervised learning, Reinforced learning, Self-learning, Deep learning

1.2 Research Methods & Structure of the thesis

This thesis consists of a literature review with an analysis of the essential roles and about the changes happening in the industry. A hands-on review is conducted on a machine learning platform as a service. From this review, a heuristics for choosing the platform is created. After the analysis and review, the results are discussed. Furthermore, a conclusion is completed about the current challenges with the possible future solutions for these challenges.

The literature review consists of a study about the existing industry methods and practices, to define the terms used, and describe the relevant data mining process models (chapters 2 and 4). The literature material consists of academic publications from many different countries and instances to find out an overview of the subject at hand.

An analysis of the changes in the industry is conducted to find out what are the factors fueling the evolution of tools and platforms, and the roles of project members creating this data-focused software (chapter 5). Additionally, to this analysis, a review of typical machine learning platforms as a service is executed by an example case which generalizes the functionalities to create these kinds of projects (chapter 6). From this analysis, an heuristics for choosing a machine learning platform is created (chapter 7.2).

The data-driven project consists of three main elements, and they are illustrated in figure 1.1. Data mining process models (chapter 4), roles (chapter 3), and platforms (chapter 6) are essential to understand to successfully comprehend the data-driven project environment. The elements are affecting each other in different ways, and these connections are summarized in the conclusion chapter (chapter 8).

Furthermore, a discussion provides ways to understand the impacts and effects of data scientist's and software developer's role development (chapter 8). This drafts a possible future of the industry from the viewpoint of data scientist and software developer roles by understanding their requirements and strengths. Our whole society also plays a significant role in the future for these roles and platforms, and discussion outlines the future from that

viewpoint as well.

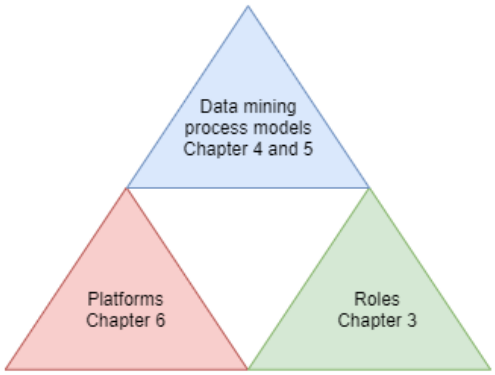

Figure 1.1: The three elements in data-driven project environment.

The conclusion at chapter 8, we summarize our findings to provide suggestions for individuals practising data related software development and as well for the companies to change their methods to understand the data and its nature better.

2 Background

On this chapter, we go through the basics of data mining and reasoning why data mining term has faded away. The machine learning has increased in the spoken language with its utilisation increased. Implementing machine learning capabilities to a software product have proven to yield great results. However, the wide variety of methods and practices can create pitfalls for inexperienced data scientists or software developers when these products with machine learning are deployed into production.

The Ancient Greeks excavated materials from the earth to further process them to valuable goods [1]. Data mining can be thought of as extracting new knowledge from the data. Same way as a product fabricated from iron ore is valuable, the new knowledge from data is valuable for its owner. The new knowledge gives means to increase the profits in business and further understand the field the data was collected.

When the digital age dawned, and computers began to be more powerful each year, the available storage space also increased, and the people in IT sector started to store data in vast amounts in the computers and the servers. The digital age changed the way people run their businesses. When IT sector entered the Big Data[1]-era, we noticed that traditional static database methods are not feasible anymore to produce results from the data in a fast and a reliable way. The newly developed methods, tools and processes models were to automate, to ease and to organise projects relating to this enormous amount of data.

Data mining process models give guidance for the data-driven projects. Following

[1]Large amounts of data, currently more than a terabyte in size

them enable solid results, and these results are providing insight or understanding of the data. However, data mining has been shifting to different services for the customers or the business people, as the world is becoming more dynamic and fast-paced. The desired result in data mining is to learn from the past and predict the unforeseen data. This falls under machine learning which is become common in the modern world. Data mining has been transformed into machine learning, enabling the learning and prediction without any interpreting of the user.

As the earlier nature of the data was used to be static, it was collected into a database and used when needed. Now, data is evolved into a more dynamic and stream-like structure. Many process models and methodologies were developed during the heavily data-driven years, such as KDD (Knowledge Discovery in Databases), SEMMA (Sample, Explore, Modify, Model, Assess) and CRISP-DM (CRoss-Industry Standard Process for Data Mining). They are *waterfall-like* models and fit poorly into modern agile software development.

Developing an application from the result of the data mining or machine learning project where these process models were used is challenging. Data scientists and software developers are not speaking the same language[2] as the data scientist sees the data as a more static object than the developer. By studying these roles, and their most commonly used habits, we can notice a gap between these two parties.

2.1 Basic Terms

The field of data science is evolving fast, and the terminology is still forming. Specific terms are used differently in scientific papers and practical jargon. Hence we determine the most used terms which we use in this thesis.

- Data analysis is a process to obtain information from data to guide business be-

[2] figuratively speaking

haviour or give insights to decision-makers

- Data mining, as a part of Data Analysis, is a process or processes to extract new knowledge from static or dynamic data with a variety of methods
- Process model is a set of phases or steps to abstractly describe a process to achieve the desired goal
- Methodology is a theoretical point of view description of methods used in a specific area
- Paradigm is an abstract theoretical framework which can include the above-mentioned
- Machine learning is a process to let the algorithm access the data and improve the performance of the algorithm over time
- Artificial intelligence is a system which can interpret given information and act according to it correctly
- Software development Life Cycle (SDLC) is a process which covers software's development during production, maintenance, and deployment
- Database, a system to hold persistent data in various forms and types
- Data scientist is a person who develops new algorithms and methods to find useful information from the data
- Data analyst is a person who uses his or her knowledge of data analysis to provide a conclusion for the given data
- Software developer is a person who is part of a team developing the software
- DevOps contains practices to join software development (Dev) and IT operations (Ops)

- Black box is an object which inner functions can not be viewed, and only its inputs and outputs can be perceived.
- Data dimension is a term used to describe the number of features for the data points

These terms are defined in academic publications and other reputable publications. It is easier to describe technologies or concrete methods than roles in a company or a project as there is diversity between companies, persons and projects what are the tasks behind each role.

2.2 Data Mining Projects

To further define data mining projects and solidify its usage in businesses, we go through the advancement of it over the past thirty years. The data mining projects can be seen as part of business processes and methods (i.e. Business Intelligence) where results from collected data and generated reports steer the business to stay profitable and viable. The end products of these projects can be decisions or automated task; they enable the user to decide with deeper awareness and knowledge. The information received is valuable for the company regardless of its field, and hidden information discovered from the data can be even more valuable. Different methods, tools and expertise are required to access this hidden information. As well, the required expertise from various fields is needed to understand the data used or the meaning of the outcome of the project.

Data mining projects can be different in sizes and complexity. Nevertheless, they have in common that they have aspects from different fields; databases, machine learning, statistics and visualisations. Sometimes the data mining projects start from the optimal start point, which is that the data has been collected with the data mining procedures in mind. This improves the possibility of having high data quality, and thus, the results are robust in quality. If the data is not collected data mining in mind, problems might

arise regarding the overall quality and applications of the project; as well the integrations between different systems can cause additional burden to the project. [2]

The term data mining is often referred to as the whole process where *knowledge is discovered from data*. As such, we can conclude that the data mining project is a project where interesting patterns and knowledge are discovered from a large amount of data. This data is stored in databases or streamed from one or multiple sources; where database represents more static storage of the data than streaming from the sources. These discovered patterns are some consistent structure inside the vast amount of data which present the nature of the data. With a great amount of data comes to a great number of patterns, and finding the most interesting patterns can be challenging, finding them represents an optimisation problem in data mining. Understanding the nature of the data and its meaning can lead to insights and discoveries which further guide the company's direction in their businesses or simply put give valuable information on the individual level. Sometimes the information received is not the interesting part, but the information that is missing or the data points which greatly differ from the rest of the data can be the most valuable information for the business. [3]

For adding data mining project's processes into existing business processes, adept persons are needed for bringing their knowledge and perspective of the data mining to the business processes. It could be intriguing to use data processing as a *black box* to restrict its influence and side-effects from the whole business process. However, it is too valuable to be handled as a black box. Furthermore, the wrong conclusions of the results can lead to wrong business decisions.

Integrating data mining into business processes is achievable, although it can be tedious, and additional rules must be followed as it will cause changes to ongoing business processes. Using data mining phases as black boxes in the business processes or the software development also reflects into the concrete technical development and usage of the developed system. As in the later chapters, enterprise tools and platforms are examined

which almost all use the *black box* approach. New flexible systems and frameworks can ease the adaption of data mining processes into the business processes. [4]

To sum it up, data mining projects are projects which aim to provide stakeholders knowledge from the data, whether it is from large data storage or streamed from different sources. This knowledge can either describe the nature of the data, predict where new unknown data belongs, associate the data with other data points, or detect anomalies, outliers or errors. These data mining technologies and methods used are close to the machine learning field. [5]

2.3 Data Mining Technologies

Data mining embraces techniques from different fields to achieve its goals. Each one of the techniques has its advantages and disadvantages. The techniques include statistics, machine learning, database systems and information retrieval.

Statistical models use statistical assumptions to approximate the real nature of the data and make predictions based on these received assumptions. They can be used on data to handle noisy and low-quality aspects of it for other methods to perform better. Statistics can also be used to validate the results of data mining processes. However, these models can be computationally heavy as they use all the data with its every dimension.

With machine learning, user can create a model which essentially makes the created model to learn from the given data and make predictions for the new unseen data. This learning can either be; supervised, unsupervised, semi-supervised or active. The supervised machine learning model is trained with training data. The model can decide a label or value for the new unknown data input, which is missing the said label or value field. Unsupervised model clusters the given data into similar clusters without labelling any new value for the given data. Semi-supervised uses methods from both of these previously mentioned ones to achieve decision boundaries for the new upcoming data. Inac-

tive learning, the user is asked to label or validate the prediction the model has created for improving the model in the future.

These machine learning models can require a large amount of data to learn the model to perform well enough. However, the learning (i.e. training) might need to be stopped for not to overfit the model for the given training data. It is not wanted to entirely create a complex model of the training data, yet to find the generalised model for the new unknown data points. With a large enough of a dataset, a machine learning model could be created which can quickly make a prediction, for example, detection of a fraud or a recommendation of new products to the customer in a website. Some of these tasks human could never do quickly enough as the dimensions of the data are over the human apprehension.

Database systems have been used for storing the data from the early 1970s [6] and they have been part of the data mining field as a method to store data and to use different methods on it efficiently and reliably [7]. These database systems hold all the data in different structures, such as in NoSQL, network, hierarchical and as a relation. The popular ones are NoSQL which has no strong relationships between data points in the database and the relation (SQL) database, which has a strong relationship between data points.

Different methods can also be used to process data in an efficient way. Active database systems can be used automatically in data mining tasks as the user made rules and triggers can be created to react to the changes in the data. Information retrieval (IR) can be described as something that fetches the information from a large set of data which keeps updating frequently. Hence, IR is strongly related to the mindset of data mining as new information is achieved from previously gathered data. The most common use case of IR is searching the internet with search engines[3]. These IR systems use different types of indexing methods to access the data quickly. These are just a few examples of possible methods and technologies to be used in data mining projects to achieve the wanted results,

[3] such as Google and DuckDuckGo

yet these technologies vary with their end goals.

To summarise, with statistics user can achieve a broad view of the data at hand, and with machine learning user can make a future prediction from the previously learned data. The database systems and applications using these are in everyday use in different businesses to achieve the knowledge to perform well in their businesses. With Information retrieval tools, the user can find knowledge from a vast amount of data to investigate different matters, such as the frequency of a word used in literature. IR can also be used to further enrich information in data mining applications or quickly achieve precise knowledge. Other examples would be different visualisations, and domain-specific solutions to provide the knowledge or insights of the gathered data. [3]

2.4 Risks and Benefits

The benefit of using any machine learning methods comes from the fact that when there are a lot of data available, it manifests itself in high dimensions. Humans can not see the patterns in the data fast enough to achieve the wanted task. A system to act on changes in fast as possible manner can provide valuable benefits for the stakeholders, i.e. profit, or guide further in the business strategy.

Risks of using machine learning techniques emerge when these techniques are used without good enough expertise relating the tasks at hand. Inexperienced users could misinterpret the results or cause the results to be biased. Causing the results to be biased could even be fatal if the machine (i.e. predictive model) acts wrongly. These kinds of systems commonly require additional support from human to act correctly [8]. Some of the drawbacks come from the methods used. The model acts like it has learned, and might act wrongly on the new data. In the study presented by Poland, McKay, Bruce, *et al.* [8], Tesla Autopilot caused a fatal accident when it did not recognise turning truck properly, and the driver did not provide any additional input or actions to the self-driving

car. Furthermore, studies have shown that different machine learning applications can be biased or even racist to some extent. This can not be generalised as the applications are working in their expertise fields and they vary about the methods used, and as well reasoning for machine learning to act certain way comes from the training data, and previously experienced cases.

Noor [9] presented a case study where an AI application could not detect melanoma in dark-coloured skin as melanoma is rarer on dark skin. This was because the training data has fewer images of melanoma on the dark-skinned person compared to lighter-skinned persons. Similarly, USA Today reported that Google's Photos application labelled dark-skinned persons as gorillas, and The Wall Street Journal reported that Staples was changing its prices for each customer based on their location. As the application development and society is becoming more and more data-driven this kind of behaviour could be seen as bugs in the system, which would require significant effort to be fixed.

When a machine learning model has given the training data, and it does predictions based on it, and if the model is created without malicious intent, the predictions are justified and not biased. For example, suppose a bank does not approve a credit card for an individual based on their machine learning model, which has been taught with relevant data. In that case, the decision is justified and not biased. Still, when the machine learning model takes unrelated information into accounts, such as postal code or ethnicity, then the model is biased and practising discrimination. A study by Ruggieri, Pedreschi, and Turini [10] lines that data mining techniques can be seen against fair treatment. These models try to find and add weights to variables for making the distinguished decision boundaries to classify or predict faster and accurately. It comes for the data scientist to decide what kind of data can be given for the model not to cause discrimination.

A study by Bose and Mahapatra [5] covered journals and conference proceedings to find the most used methods, problem types and domains used in data mining business applications. These were finance, telecommunications, web analysis and marketing, yet

data mining applications were developed for other domains, such as legal, medical, insurance and software development. All of these, they are heavily focused on machine learning techniques and methods. Their study shows the most used techniques; Rule induction, Neural Networks, Genetic Algorithms, Case-based Reasoning and Inductive logic programming. These can be used in businesses by using them diversely to achieve their goals to classify, predict, associate and detect.

One of the breakthroughs caused by the increase in computation power is the ability to use Deep Learning[4]. It uses supervised, semi-supervised and unsupervised learning as a multiple-layer network to distinguish low-level features to higher-level ones until the end prediction can be recognised from the input. Deep Learning networks are called Neural Networks and using representation learning; they can learn complex data and predict correctly in high probability, providing that if enough learning data is provided. Deep learning models can achieve the tasks which humans can not achieve fast enough. The wide variety of different methods in data mining enables its usage in many different applications. As the tools and platforms evolve further, the implementation of these is on the rise.

2.5 Summary

The past has taught developers to use methods which help their daily work in the data-driven IT field. However, the people working on these roles have experienced evolving themselves into broader roles. Concurrently to this, the data have evolved to be more dynamic, and it is gathered continuously from different sources. This has caused the data-driven IT field to be transformed into an entity which requires reliable and concrete methods to ensure the quality of the developed products. As the field is becoming more complex and broader, the people taking part in the development needs to understand sub-

[4] which is a subset of machine learning

jects they were not taught in their studies, and are not inside their expertise zone.

Software developers who are working with the data-driven projects are part of an environment which has three parts, see figure 1.1. The data mining process models provide good to know knowledge about data science. The definition of roles and their responsibilities guide the division of the work in a project. And, the platforms enable the comprehensive environment for the development of large and broad data-driven projects.

3 Roles in a Data-driven Project

Even though *"A jack of all trades is a master of none, but oftentimes better than a master of one"* is an excellent utopian point of view. Yet it is not feasible for one person to be involved in every part of the software development project in the real world. People have an interest in different subjects and receive an education based on their interest. As Beranek, Zuser, and Grechenig [11] point in their study, the software development team should consist of different types of persons to ensure the best possible outcome of the project. Teams are commonly consisting of people who are more focused on the technical tasks and others who are focused on documenting, planning, and also in group management tasks. We can quickly notice the data scientist and software developer has a similar scope of the roles in the project.

A data scientist needs to plan the data collection, ensure data consistency during the project, and develop the required data processing models to achieve the wanted outcome. A software developer has the responsibility to bring the project into life in the form of usable software and maintain it during the software's lifetime and provide the necessary documentation. Analyzing these roles give us an initial understanding of these roles.

The data scientist in the project naturally needs to know the field of data science, which consist of machine learning & statistics, computer science and understanding of businesses. The data scientist also needs good skills on interacting with different people, as his or her expertise can be hard to understand by inexperienced people and a data scientist is required to explain complicated models and methods to different stakeholders in

the projects. It is a difficult task to summarize this role as there is no standardization what skills a data scientist should have or what tasks he or she should do [12]. Furthermore, Miller [13] points out businesses need more than a theoretical view of the subjects; they need strong technical skills and business understanding to benefit from the developed methods and machine learning models. Also, other skills are valuable, as the enforcement of GDPR (General Data Protection Regulation) in Europe has shown that the data is valuable for individuals and ethical questions are on the rise.

For example, the internet provides an excellent platform for collecting data for research. However, sometimes it is hard to differentiate has the user give consent to share the data in research or in what kind of purposes[14]. Even though the data is anonymized, with the power of data science, it could be de-anonymized for malicious purposes. In 2018 Farr [15] reported Cambridge Analytical used data from Facebook in unethical ways and further investigations showed that Facebook was trying to receive medical information from hospitals to link anonymized data of persons to medical records [16].

> *"Data scientists' most basic, universal skill is the ability to write code. This may be less true in five years when many more people will have the title "data scientist" on their business cards. More enduring will be the need for data scientists to communicate in a language that all their stakeholders understand—and to demonstrate the special skills involved in storytelling with data, whether verbally, visually, or—ideally—both."* By Patil and Davenport [17] in 2012.

The software engineer is required to have the skills to develop the wanted outcome in the environment provided and knowledge to design the software in a way it satisfies the requirements of the stakeholders. These skills enable constructing of the software in clean and understandable code. Debugging, testing, maintenance, and documenting the software are equally important parts when developing the software. [18]

Finding an exact definition for a software developer is also a challenging task because the nature of software engineering lies in the business sector and not in academic peer-reviewed reviews. In a study of Meade, O'Keeffe, Lyons, *et al.* [19], they have gone through interviews and peer-reviewed materials to define the software engineer, and with its tasks and responsibilities. The role of a software engineer has shaped by the frequent changes in the used tools and frameworks. These new tools, frameworks, programming languages, and platforms are developed continuously for improving the previously used methods. Software developer's role also consists of more detailed tasks or roles as these require additional expertise to become good at it, such as architecture design or supervising team members. The entry-level skills and knowledge required for the role of software developer are programming, testing, documentation, and co-operation. These are seen as mandatory skills for being the core part of the software developer's daily work.

The development in SDLC has also shaped the role, and the Agile methodology is enforcing further the software engineer to continuously build and deliver the software with the help of automation. Continuously delivering also enables continuous feedback, and using Agile methodology also gives new tasks for software engineer making his or her job wider than just focusing fully on programming. The automation can easily be achieved by using cloud-based platforms for faster deployment and reducing the maintenance overhead, also to have easily scalable environment if necessary. Using a certain platform, specific knowledge about the platform is required to use the platform in its full potential. The role has also been greatly affected by the increase of artificial intelligence and machine learning in modern software projects, and it is a valuable asset for a software developer to know these. [18][19]

The data scientist and software developer roles overlap as both of these has to understand the environment (i.e. the platform) they are working and the software projects want to have machine learning and AI on the software to benefit from it. This causes the software developer to delve into the realm of a data scientist.

When the DevOps paradigm was created, software development and operations have become more intertwined, and it had seen as beneficial for both of these. The data scientist and software engineer currently represent the same separation where DevOps was a few years ago. Future projects make these two parties co-operate together with overlapping areas. It is becoming necessary to understand from both of these worlds regardless of individuals previous expertise either in software engineering or in data science.

It is an open question should these roles be separated either by previously gone through process models or with the platforms. Miller [13] suggests increasing the analytical thinking and data science education on every part of the teaching in business and technical universities. One of the reasons to suggest this is due to the vast amount of data gathered and processed continuously in the generic IT projects.

4 Data Mining Process Models

On this chapter, we describe what data mining process models are and their history; mainly the three most famous models and what disadvantages they have. We also cover reasons behind why these process models need to have elements from traditional and agile project management.

As companies and other organisations are shifting continuously for using data as a valuable asset, they are collecting more data than a human can comprehend in a single glance. Due to this, it is necessary to develop models, methods, processes and frameworks to enable understanding of the data and its ever-growing data streams.

These data mining process models provide tools to develop applications, which are using data consisting of millions of fields within millions of records [20]. This vast amount of data is valuable for the businesses to either guide or support decisions in their business, improve sales or increase customer satisfaction and automatically make decisions which are too challenging for a human to make [4].

One of the earliest process models for discovering knowledge was the *knowledge discovery in databases*, commonly referred to as KDD. Fayyad, Piatetsky-Shapiro, and Smyth [20] and Brachman and Anand [21] presented KDD to tackle the problem caused by data overload and to find useful information within the data in an understandable form. This form could be a short report, making the data more abstract or making a predictive model for future unforeseen data points. Other popular models are SEMMA and CRISP-DM.

SEMMA [22] was developed by SAS Institute, and CRISP-DM [23] was developed in co-operation by DaimlerChrysler[1], SPSS[2], NCR[3] and OHRA[4].

SEMMA was evolved by improving the KDD process, and CRISP-DM was derived by combining SEMMA and two different process models, as seen in the figure 4.1. This makes the CRISP-DM process model the most versatile and sophisticated one, and also the most popular one. These three cover the most used processes for data mining projects. However, many companies use their own established and domain-specific processes which are not publicly available [24]. In the following sections, we will discuss KDD, SEMMA and CRISP-DM in more detail, what are their strengths and weaknesses, and to provide understanding about their usability in today's data mining and machine learning projects.

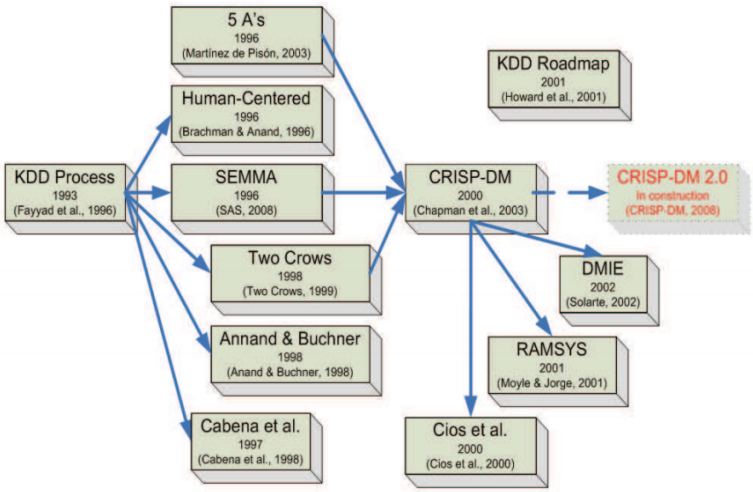

Figure 4.1: Evolution of Data mining and knowledge discovery process models and methodologies. [25]

[1]German automobile manufacturer

[2]IBM owned software package for statistical analyses

[3]American technology company

[4]An insurance company

4.1 Analysis of Process Models

KDD, SEMMA and CRISP-DM provide users steps to ensure the data mining project is taking into account the data and parts of the project the data affects. SEMMA and CRISP-DM are defined as abstract level methodologies by their developers. However, due to their general nature of providing steps to follow and ability to provide concrete results, they are treated as process models. All of these models have numerous steps, and a comparative study by Shafique and Qaiser [22] has shown, that KDD and SEMMA have corresponding steps. However, KDD has more steps than SEMMA, and yet CRISP-DM, and KDD have similar steps, table 4.1. However, CRISP-DM covers more extensively the step at hand to provide a more detailed understanding of the whole process.

Table 4.1: Summary of the correspondences between KDD, SEMMA and CRISP-DM. [22]

Category	KDD	SEMMA	CRISP-DM
Analysis	Cleaning	-	Business understanding
	Selection	Sample	Data understanding
	Preprocessing	Explore	
Active	Transformation	Modify	Data preparation
	Data mining	Model	Modeling
	Evaluation	Assessment	Evaluation
Deployment	Knowledge	-	Deployment

The steps can be divided in three following categories by their nature, *analysis*, *active* and *deployment*. The *analysis* category contains steps which are related to analysing the data and the project. The steps in all of these three models which can be characterised into this category are about understanding the wanted outcome of the project by different stakeholders, the data and its quality and processing of the data for its final form.

The *active* category includes steps having actions and methods to be implemented on the data used in the project. These actions modify, transform and enhance the data to provide better results. Some of the possible actions are; attribute selection, feature extraction, feature selection, dimension reduction, and finding outliers. After these actions, the data is in its best quality condition, and different modelling techniques can be used on it to create the models of the data. When a model or models are created, they are reviewed, and the results are evaluated; are they achieving the objectives which were defined in the earlier phases in *analysis* category and are they legitimate and not biased. Phases in this category can be run multiple times to improve the possible outcome.

The *deployment* category consist of the final phase; where solutions and results are given to the stakeholders of the project, which is also the most interesting one from a software development point of view. The outcome of this phase is the actual usage of the data, even if it is just a simple report about the obtained knowledge or analysis of the data. In a typical case, the result is a predictive model and stakeholder wants to use it in the real environment. The predictive model could be given to a different team to do the development of an application. This development team could then use their own software development life cycle models, where the starting phase is the end of the data mining process model. However, for the overall process to be accurate; the persons being part of it must have an understanding of how to do data mining properly.

KDD

Finding and understanding the knowledge from data can be a complicated process involving people from different background and knowledge of data mining in different steps. The KDD process model outlines steps for these people to follow consistently, depicted in figure 4.2.

Steps for KDD process model are the following [22]:

1. Data is retrieved from a source

2. Selection-step narrows the data set to become more relevant for the whole data mining process

3. Preprocessing-step further cleans the data with different methods (e.g. *feature extraction*) to provide uniform data

4. Transformation-step reduces the complexity of the data by reducing dimensions of the data

5. Data mining-step is the most labour intensive step; the data goes through various data analysis methods to obtain wanted knowledge

6. Interpretation/evaluation-step consists of interpreting and evaluating the knowledge received from the previous step

7. Knowledge is extracted from the data by analysing the results

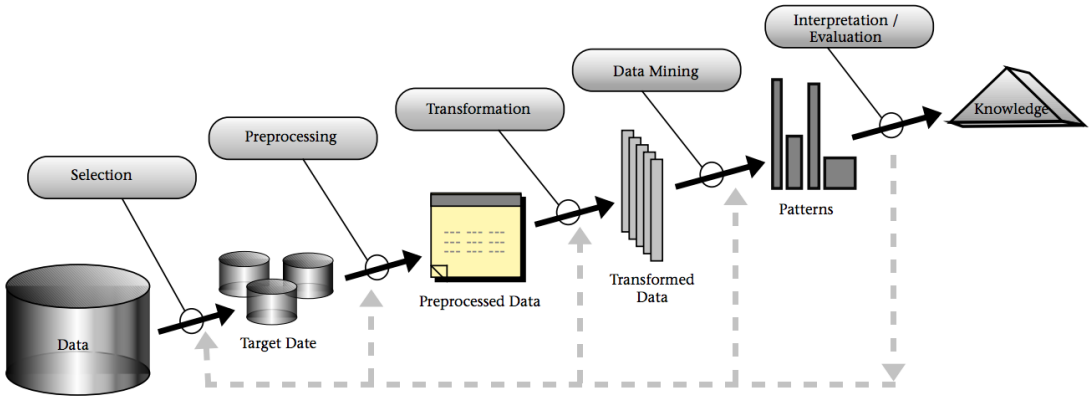

Figure 4.2: An overview of the steps that form the KDD process. [20]

Fayyad, Piatetsky-Shapiro, and Smyth [20] have drafted challenges regarding KDD which professionals struggle with, yet they are not solely unique to KDD. These important ones are listed below.

- New algorithms need to be used as the size of the databases require efficient algorithms.
- High dimensionality, which makes the data complex. The size of the dimensionality must be reduced by different methods, such as Principal Component Analysis (PCA)
- Overfitting: The pattern discovered includes noise and errors from an optimistic algorithm.
- Changing data and knowledge; When the data is frequently changed, the previously discovered pattern does not represent the new data causing wrong interpretation of data.

Missing and noisy data affects the overall process of knowledge discovery and originates from the initial development of the database or system and methods used to receive the data. The result (i.e. patterns and extracted knowledge) must be presented in a human-understandable format. Integrating KDD into existing systems is challenging as many companies, and organisations use different kind of database management systems and software which have their own interfaces and procedures to work with.

SEMMA

SEMMA is an acronym for a waterfall-like process guide for extracting new knowledge from the data [22].

Sample step is for the user to lower the complexity of the data for it to become usable and calculatable for the processing unit.

Explore step focuses on finding useful patterns and recursions to provide a better understanding of the data.

Modify step further modifies the data set for the next step by transforming the data.

Model step is for generating the actual model of the data by different modelling techniques depending on the problem and data.

Access step is to evaluate the produced model and its usefulness.

This process model is simple and meant to be used with SAS Institute's Enterprise Miner. The functionality of this application is based on the SEMMA process model and therefore using it without SEMMA can be challenging [26].

CRISP-DM

CRISP-DM was developed to have a process model with a robust framework to make data mining projects affordable, easy to implement and to monitor. Aim of its development was to be independent and not chained into a particular industry or a specific technology.

CRISP-DM defines four levels of abstraction, starting from general tasks and going into more detailed ones: *phases*, *generic tasks*, *specialized tasks* and *process instances*. These are meant for the user to understand and divide the steps to a smaller pieces for their team to complete [23], as seen in figure 4.3.

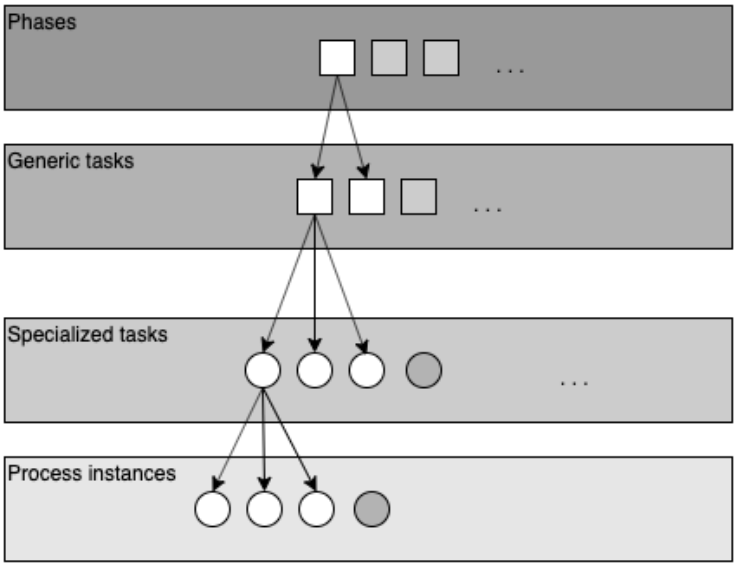

Figure 4.3: Hierarchical breakdown of CRISP-DM methodology. [23]

Each phase is the highest level of abstraction and contains generic tasks that are meant to cover possible data mining situations and the development of data mining techniques in a general way. The specialised tasks are part of the parent's generic task to define how generic task can be achieved. Process instances describe the concrete actions, reasoning with specific decisions and results of particular action made by the user. These instances are documented precisely, and concretely and they are completed without any generalisation of the instance.

The CRISP-DM methodology is divided into two parts; the Reference model and the User Guide. The Reference model (see figures 4.4 & 4.5) provides overview of phases which further has more detailed tasks inside of the phases. The reference models start by understanding the needs from the business side of the view after they have been understood and locked in, the projects start experiencing with the data by gathering it and making an initial abstract level understanding of it. From that point on the data, scientist focuses fully on preparing the data by excluding outliers and reducing data dimensions for the next phase to generate the wanted model. When the data scientist is building the model, he or she might have to enhance the data quality by different methods further, i.e. going back to the data preparation phase. After the model is created, it can be used for evaluating against the business needs or starting the whole process again if new understanding is achieved, or model can be transferred to deployment.

Furthermore, the User Guide further clarifies the overall tasks of the data mining project. The User Guide enables people working on different steps or even outside of the data mining project to understand the methods used on it, and also the challenges and solutions for these challenges. [27]

CRISP-DM 2.0 was supposed to be a new adaptation of 1.0 version by upgrading, adding and removing certain phases. It was under development in 2008 by Special Interest Group (SIG) [25], but there have not been any new publications regarding it. The first version has given the solid background for developing the second version or even some

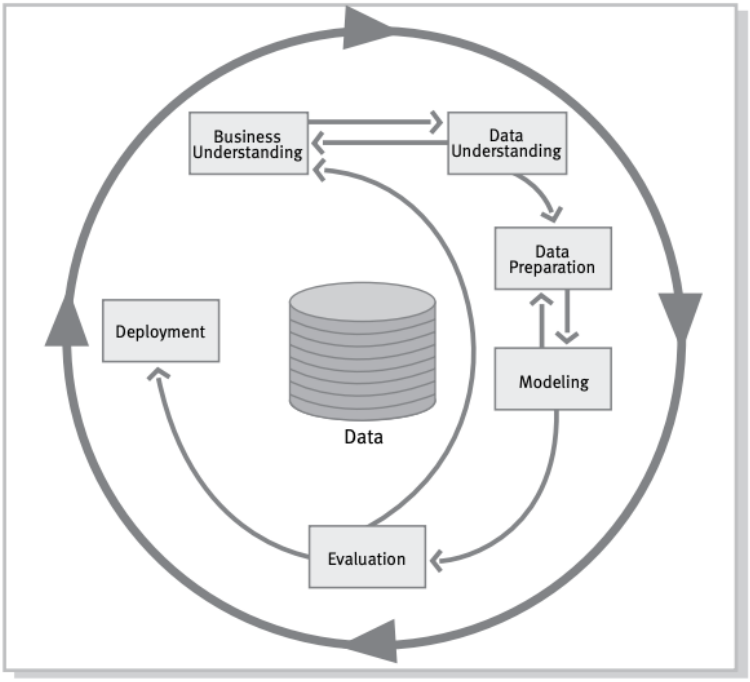

Figure 4.4: Phases of the CRISP-DM reference model. [23]

other process models which take into account in a more detailed view the aspects of software development and business sector [28].

4.2 Challenges with Models

These process models used in data mining projects have steps for understanding, enhancing, transforming, modelling and evaluating the data. However, the ongoing and dynamic nature of the data collecting might reveal some changes in the data, and the applications developed for that data must adjust to reflect the newly changed data. There is not enough support for the users to acknowledge and understand the nature of the data when it is continuously changing. For example, if a model has been fitted to some specific dataset, and the data changes considerably, the model does not fit on that dataset anymore and would provide inaccurate predictions or interpretations of the results.

None of these models covers the implementing or deployment phase sufficiently, and

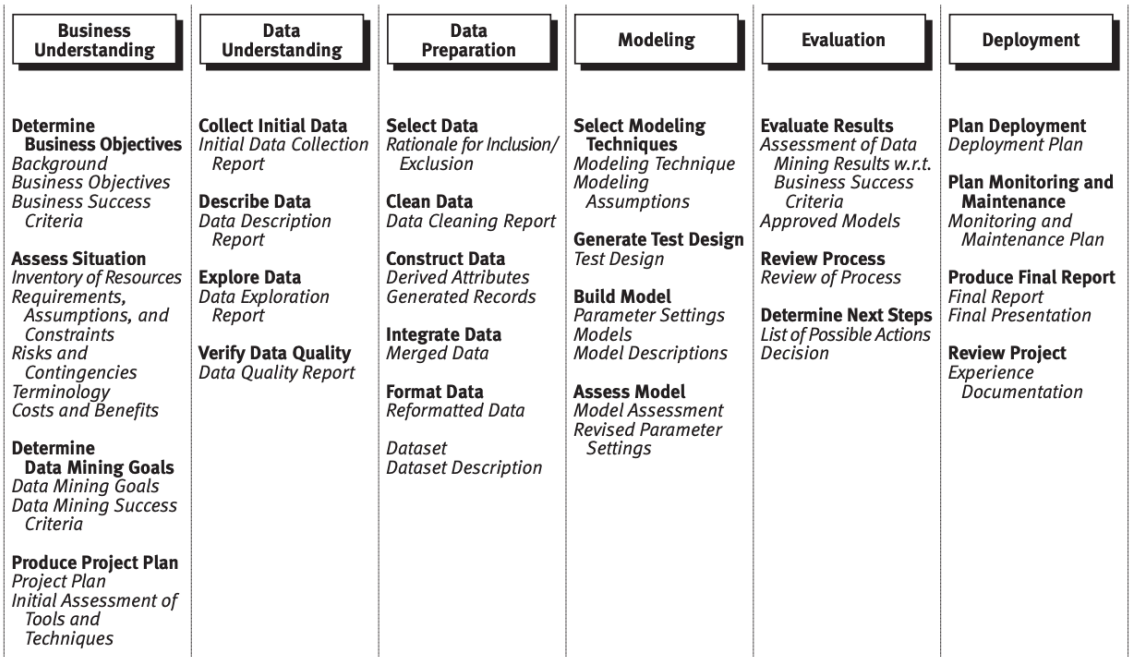

Figure 4.5: Phases, generic tasks (bold) and their outputs (italic) in the CRISP-DM reference model. [23]

the one who is making the deployment or maintaining the previously deployed application is usually not the data analyst who used these process models to produce the result. In KDD, the deployment phase is treated as "discovered knowledge"-phase where the knowledge is added into the other system or reported to interested parties and also evaluating the achieved results [20]. In SEMMA the final step, *Assess* is only for the user to evaluate the practicality and reliability of the extracted knowledge to a new received unseen data [29]. In CRISP-DM the Deployment-phase is briefly described as something that provides a report or a more complicated process to be implemented by other parties which did not take part in the previous phases of CRIPS-DM [27].

To recap previously mentioned problems:

- Previous phases in process models do not take the final deployment phase into consideration.
- The user doing the final deployment phase does not know about previous phases.

- Changes in data after deployment are not taken into account by maintainers.

Furthermore, none of these models covers tasks regarding project management, quality assurance and continuous integration which are found in software engineering models. This is no surprise as they are have been developed for different kind of projects. However, due to the overall complexity of data mining projects and large amounts of people taking part in them, the data mining projects can be seen to resemble modern software development projects closely [25].

4.3 Summary

After reviewing the process models, we have noticed that these models provide ways to take into account the nature of the data and provide steps for the users to understand how the data should be treated. However, these lack the connection to modern software development, and companies have not implemented the new data mining process models into their development processes. This proposes a great problem for the data scientist, as he or she might have problems for managing time and frequently delivering deliverables in case the requirements often change in the project. As data science is more like *science* than engineering; it requires additional effort to join the expertise of the data scientist into modern Agile development.

5 Modern Data Mining for Software Developer

In this chapter, we go through the changes in the nature of software development from the viewpoint of data-driven projects. We study what are the factors causing these changes and possible solutions for the challenges which are still present when developing data-driven projects.

The end goal of the data mining is to provide new knowledge from data, and the software engineering projects end goal is to provide a new application to be used for the needs of the stakeholders. Combining these two creates a data-driven project. In a data-driven project, an application is developed by the needs of the stakeholders, and the data used in the application is in an important role and processed in certain ways to achieve the functionality of the application.

The nature of the data has become more streamlined and dynamic, causing data mining projects to shift considerably towards software engineering. Combining these two domains is a challenging task, and a few researchers have proposed solutions to join these two together for a complete model which cover aspects of both of the domains. Many of these solutions have been developed and presented in academic publications.

Computing power is required for data mining and machine learning applications, and it has been increasing steadily from the 1950s, and after 1970s, new processors have been developed which have been using silicon-based solutions. Following Moore's law,

processor manufactures have doubling transistors in the processors to deliver a significant boost in performance. This enables using computers in a wide variety of ways, especially in data mining and machine learning tasks which were not possible in everyday use, as the processors were not fast and efficient enough. Even though the advancement of a single-core performance has been stalled, the multi-core processors have advanced greatly over the past decade. [30]

This performance increase has made it possible to use certain algorithms, such as K-nearest neighbours (k-NN) classification algorithm. The k-NN is an excessive computation algorithm, as it needs to calculate distance from each data point between all of the data points. For this reason, this algorithm has demanded plenty of processing power. It has not been worked well time-wise in machine learning, and data mining applications before the computing power and memory amount had been increased enough.

As the computer power of the processors increased and the cost decreased, many companies were established, to offer their customers software development to automate and to increase customer's efficiency in their business field. This created a new industry; Information Technology industry, currently expected to reach over 1,900 billion U.S. Dollars in the year 2021 [31].

5.1 Software development life cycle processes

Different SDLC process models have been developed in the past 50 years since the *software crisis* in the 1960s. The crisis started from the fact that the developers of the era were not making efficient and modular software solutions, causing an increased cost of the development and cost of maintaining the software projects [32]. This led customers to be unhappy about the products they received, which were expensive and poorly performing. Most used SDLC was Waterfall model and its derivative V-model. Both of these models have subsequent steps which are rigid and not capable of sudden changes in the

environment or the requirements changes by customer [33].

Since the appearance of Agile manifesto in 2001, many software companies have started using some form of an Agile methodology in their software development. Using an Agile software development makes the development happen incrementally, straightforward, adaptive to sudden changes and in co-operation with the customer. The Agile methodology defines certain phases and roles for developers to follow, yet it is more what you would call 'guidelines' than actual rules [34].

Big companies such as Amazon and Google has taken the lead to shift development and operations of software and products to be more agile and enabling continuous delivery of the products. This is achieved by enforcing a paradigm called DevOps; pushing the development and operations close together by merging them. [35]

Software life cycle process standardization exists, and ISO 12207 has become the most popular one. It is an international standard to define processes used to develop and maintain software during its life cycle. It is not meant to be used as a life cycle process model; instead, the process model needs to be chosen when using ISO 12207 standard.

These software process models, life cycle models and standards provide a wide range of methods and tools for project managers to carry out software production efficiently with a quality outcome. Choosing the right methods for the project is vital for their size, requirements and changeability. [36]

5.2 Present State, from Data Mining to Machine Learning

The information technology industry is providing solutions to a wide variety of problems, ranging from low-level hardware implementations into a universal mobile application. Many of these projects have common elements, such as the usage of the internet, and sharing data through servers. The development of these applications require adept persons

in different fields and involve people from different stakeholders.

Alnoukari, Alzoabi, and Hanna [37] has proposed a new methodology for more agile development of predictive data mining projects. The proposed methodology uses Adaptive Software Development (ASD) characteristics with the CRISP-DM processes; hence, it is called ASD-DM. It aims to replace more static development cycle with an agile life cycle. Their case study is from automotive manufacturing domain where the case company had a data warehouse holding various kinds of data, such as inventory and sales data. Different parties in the company wanted to use the data to enhance their productivity and support in their decision making. For this task, they developed a system using ASD-DM, which could predict relevant information of the next four weeks every week with a Neural Networks and ANFIS (Adaptive Neuro-Fuzzy Inference System).

Rohanizadeh and Bameni [26] approach implements SEMMA and CRISP-DM like methodology more into an industrial procedure by adding detailed steps. In its core, it is similar to other proposed and already existing methodologies. However, these new detailed steps provide ways to evaluate and duplicate the results and to notice where the errors have appeared.

Marbán, Mariscal, Menasalvas, *et al.* [38] have proposed process model called *Refined Data Mining Process* which combines IEEE and ISO software engineering standards to CRISP-DM, giving it a better chance to succeed in both fields. This model delves into problems when developing a data mining project aiming to be a usable application. However, this model was not ready in 2007 and needed to further developed. Their study was continued three years later, which further developed this model by adding more detailed sub-processes [25].

There are no publications or statistics about how many companies are using these proposed models or methodologies in their software development. As the use of data mining and data is increased in modern software development, the information technology industry is in great need for implementing new phases and processes on to software

development. These new phases and processes could bring aspects from the data mining to ensure proper usage of the data.

The term data mining and static usage of results (i.e. yearly reports) has faded to the background when dynamic data is used in applications with dynamic results. Machine learning methods allow the processing of this type of data. This dynamic data flow further enhances the benefits of machine learning as it can provide accurate and straightforward use of predictions from stakeholders to end-users (i.e. the customers). According to Google Trends, we can see search words 'machine learning' took over 'data mining' in June 2015 in search counts, figure 5.1.

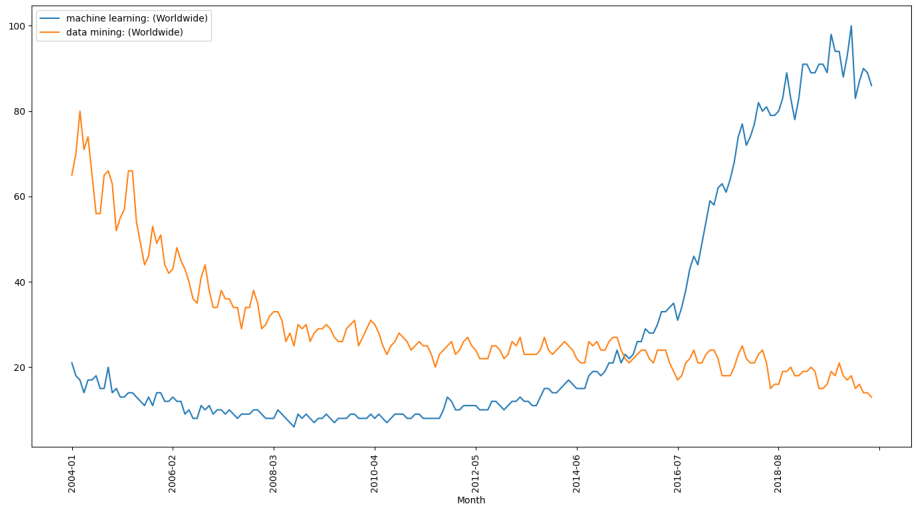

Figure 5.1: Google Trends: data mining vs. machine learning.

Using machine learning in a different type of projects has grown steadily as the computational power increase in processors have allowed machine learning to provide accurate results in complex tasks.

5.3 Recognizing the Gap Between Data Scientist and Software Developer

The vast amount of different methods and algorithms to teach the machine learning model can make starting the usage of machine learning in businesses cumbersome. As the usage of these machine learning models and algorithms have skyrocketed in many fields, the data science field has been researching the most suitable methods and algorithms for different types of use cases and types of input data.

The usage of data mining and machine learning in a different kind of projects has been snowballing in the software development and business sector. It has caused many different platforms, tools and methods to appear helping developers to implement these kinds of projects into production efficiently and reliability, and these tools help the data scientist to have a more comprehensive view of the life cycle. However, these platforms can hide the complexity of a machine learning behind interfaces, which could be argued to be bad for the transparency of the end product [39]. Going through the most popular platforms, we can find out what features data scientist, and software developers have available when they are working on projects which are heavily focused on data.

The data scientist and the software developer are experts in their own domain, and all of these platforms and tools aim to make the gap smaller between them by allowing both of them to use the same platform. Yet, our hypothesis is that there still exists a large gap. One of these gaps is how machine learning model made by the data scientist can be easily transferred for the software developer to be used in the application. There are four main ways to transfer models, e.g. predictive machine learning model from the data scientist to the software developer and into a production environment. These are

- Developer rewrites the model from the documentation provided by the data scientist in the programming language needed in the production environment
- Data scientist provides the software developer with a PMML (Predictive Model

Markup Language) file, ONNX (Open Neural Network Exchange), or similar which he or she can quickly transfer into a production environment

- Data scientist capsules the model into a serialized object which software developer can use 'as is' in the same programming language
- Data scientist and software developer use the same tool or platform to create and use the trained model

All of these methods have their own benefits and drawbacks. Software developer rewriting the algorithm of a machine learning model into a different programming language, where it was originally written, is the most cumbersome and significant waste of human resources as the model would need to be trained again. Other formats can keep the already achieved information. PMML is a markup language and hence easy to use and share between platforms, yet it has its drawbacks on type of models it can support. ONNX is an open-source format to represent machine learning models and easily share them between platforms. Capsuling the trained model into a serialized object enforces the *black box* mindset.

However, all of these could cause the software developer to use the provided model in a way the data scientist was not meant it to be used, or software developer is using it without any understanding about the underlying structure. The use of tools, platforms, methods, and services are essential for optimizing the time of the employees. This also helps to respond to errors or changes, as fast and as reliable as possible.

5.4 Machine learning Platforms

As the data is dynamic and plenty in the modern era, many platforms ranging from free-to-use to enterprise-level have been developed to compete with the dynamic nature of the data.

All major cloud platform providers have developed their machine learning services such as Google Cloud AI Platform, AWS SageMaker, Microsoft Azure Machine Learning, Oracle Advanced Analytics and IBM Watson. They are providing customers with a complete service to provide a common platform for different users, such as data scientist and software developer. Many other platforms, tools, and frameworks exist. Yet, they are restricted in some way, either on purpose, or they are not yet developed enough, such as Peltarion[1], H20.ai[2], and Valohai[3].

Machine learning specific platforms also assist companies to develop and deploy machine learning models, yet they lack the complete architecture for implementing the deployed models quickly to other tools. One of these specific platforms is RapidMiner[4], which offers full life cycle for enterprise applications. Also, many open-source platforms can provide a similar outcome as the previously mentioned enterprise platforms, such as TensorFlow[5] and Apache Spark[6]. Nevertheless, these platforms, tools, and frameworks require additional expertise to run them successfully compared to enterprise platforms. On these enterprise platforms, an inexperienced user could use the platform to developed and deploy his or her machine learning models.

5.5 Enterprise-grade Machine learning Platforms

In this thesis, we focus on these enterprise-grade platforms:

- Google Cloud AI Platform
- AWS SageMaker

[1] https://peltarion.com/

[2] https://www.h2o.ai/

[3] https://valohai.com/

[4] https://rapidminer.com/

[5] https://www.tensorflow.org/

[6] https://spark.apache.org/

- Microsoft Azure Machine Learning
- IBM Watson

These cloud-based platforms take the extra burden from the company by allowing their employees to focus on their expertise and not on to maintaining servers and other infrastructure. These platforms aim to give users a system which is easy to scale, constantly available, and contains a huge amount of computing power.

Exploring these previously mentioned platforms with a simple task in mind we can experience the benefits and drawbacks of a platform and examine it from the viewpoint of an inexperienced student who knows data science yet his or her essential skills are in software engineering and software development.

All of these services are complex in a positive way; they offer additional tools which can further enhance the overall usage and benefits of data mining and machine learning projects. They all focus on providing repeatable pipelines for the tasks, which automatically accomplish the tasks previously done by a human.

5.6 Summary

The modern data mining for a software developer is, in essence, machine learning on the cloud environments, as the platforms enable the comprehensive connectivity across systems. The HTTP endpoints, PMML, ONNX, and other ways provide this wide variety of connectivity. The cloud environments with their enormous data centres across the globe provide a huge amount of computing power for those who need it.

The cloud-based environments, commonly used SDLCs, and agile methods are overriding these previously used data mining process models. Nevertheless, the tools and methods used do not mitigate the usefulness and importance of the previously used data mining process models. The data mining process models provide valuable information for the software developer in data-driven projects for understanding the viewpoint of the

data scientist. We have seen different parties trying to develop process models for the current data-driven projects, yet having an industry-standard process model is still to be found.

6 Machine learning PaaS Review

On this chapter we use action research methodology to study the enterprise-grade machine learning platforms to find out the capabilities of them, what are the ways to ease the learning phase of a platform, and how well they answer the following questions:

- How cumbersome the end result is to implement into software developer's application?
- How well the user understood the platform or inner functionality of the developed model?
- How the prediction was justified or can be explained?
- What were the benefits and drawbacks using the model the way it was deployed?

For analyzing machine learning platforms, a simple problem is presented (figure 6.1) to test the platform at hand by using it in a way most of the users would use. The problem consists of four main parts and covers the basic functionality needed from the platform, (1) importing data to the platform, (2) training and evaluating the performance of the trained model, (3) deploying the model, and (4) using the model for predictions. In this problem, the dataset is static, yet a similar approach would work for dynamic data in a self-learning model.

After the problem has been successfully executed on the platform, an overview of the whole platform is examined. The aim is to gain insights into the platform from the viewpoint of the inexperienced student who is new to the platform. One of the critical

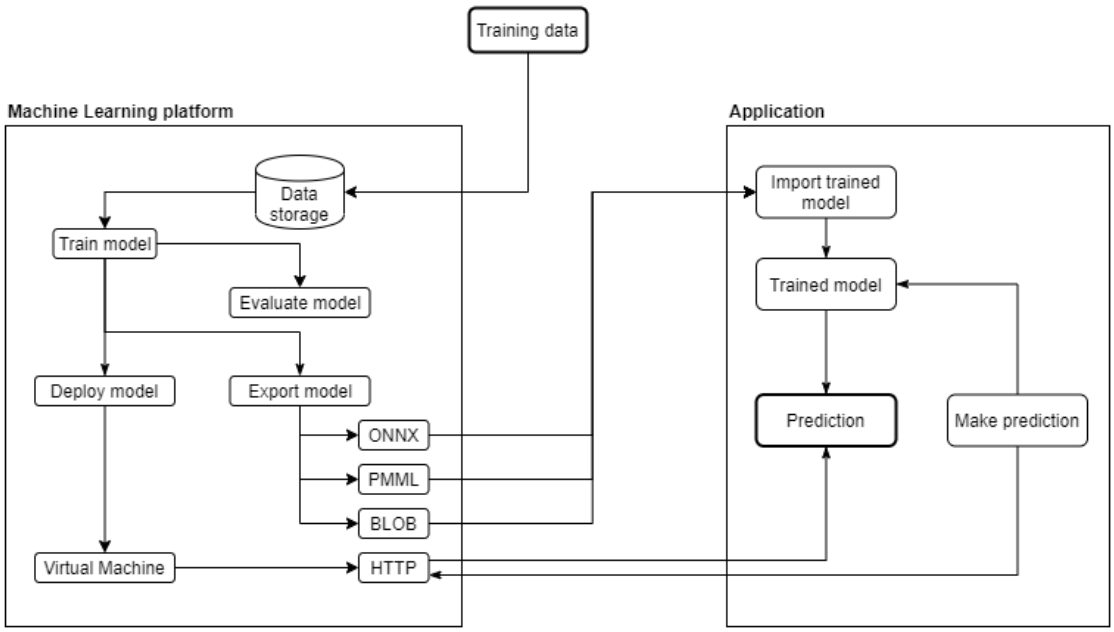

Figure 6.1: Overview of the solution for our machine learning problem.

values is understandability which yields from transparency. How things work and why they work as they do? In case the user does not understand or can not understand the inner workings of a system, it is called a black box. Sometimes having a black box is desired as this approach is common in testing to hide the testing functions from the developer to validate the correct functionality of the software and not a developer to target the testings functions to make the test pass [40].

Transparency has many meanings depending on the context. In the context of software development, transparency means having a good view and understanding of the software or system that is used, and validation of the methods used that they have not been biased or on purpose altered to provide a specific type of results [41]. In a more abstract view, machine learning models transparency is vital as hiding functionality from the user can cause undesired usage of the said model or even privacy issues. Transparency and trust are valuable aspects of the system or machine learning model as they will be used in applications which involve human lives, such as self-driving cars [42].

The dataset we use in the experiment is famous in the field of machine learning; the

Iris dataset presented by Fisher [43] in 1936. The Iris dataset contains information about Iris flowering plants; their sepal length and width, petal length and width, and the strain (class) of the flower. The lengths and widths provide distinct patterns for machine learning algorithms to recognize the most probable strain of the flower from the given input data (lengths and widths). The data is in a comma-separated values file (CSV), and it consists of 150 measurements from three different Iris strains; Setosa, Virginica and Versicolour.

One of the first simplest algorithms students learn when starting their studies in machine learning is *k*-NN supervised classification algorithm; the input data belongs to the majority of *k* nearest neighbours' class measured in Euclidean distance, see figure 6.2.

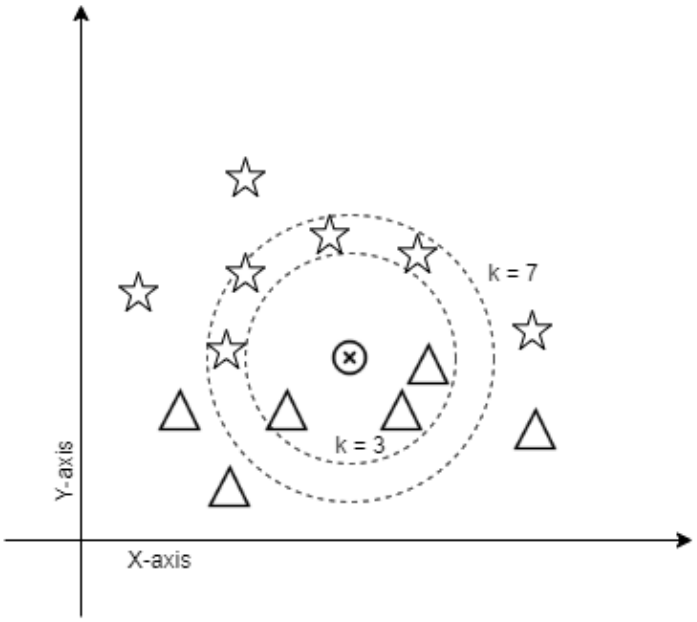

Figure 6.2: Simplified presentation of the logic behind *k*-NN classification. New unknown measurement belongs to the majority of the *k* nearest neighbours, △-class if hyperparameter *k* is three, yet it belongs to the ☆-class if *k* is seven. Hence optimizing the hyperparameter *k* is important.

It is common for the data to be high dimensional; hence, the data dimensions can be reduced by Principal Component Analysis (PCA) or other methods. This ensures notable

differences in the calculated Euclidean distances or for visualizations. Yet, in our dataset where we have only four features; reducing data dimensions are not necessary.

In our machine learning task, we aim to train predictive machine learning model in the platform to predict the most probable prediction for the given input data. The machine learning model can either be accessed from the platform through the HTTP endpoint, or deploying it to the application in ONNX, PMML, or blob -format.

After creating necessary accounts for these platforms, we can start reviewing them.

6.1 PaaS Review

In our defined problem; the k-NN algorithm is used to review the machine learning platform. The model's hyperparameter (the k) can be optimized with Cross-validation. This is done by splitting the training data into folds of new training and test sets and finding the best hyperparameter to achieve the model with the highest accuracy. Also, the data could be standardized and the dimension reduced before the training phase causing the input data needs to go through the same procedure. As our dataset is small and the feature dimension are low, we simplify our model by only tuning the hyperparameter k for the training set to find the most optimal value for it to produce the highest accuracy. This highest accuracy is an estimate of how well the model could predict the right class for the new unknown data as the machine learning field has many algorithms and methods to choose from, ranging from unsupervised to supervised. The k-NN classification algorithm represents the most basic and easy to understand method to produce predictions for a new unknown data, and due to that, we are testing these machine learning platform with it.

Many of these platforms have semi-automated or automated build tools to develop the classification model for our problem. These tools are called AutoML (Automated Machine Learning), and they are becoming popular for their easy to use. However, it

is an open question should people who do not understand anything regarding machine learning develop and use these tools. Problem data scientists have when they face a machine learning problem is; what kind of algorithm to use for the problem and with what hyper-parameters? Some approach might work best for the previously known data, yet if the future data shifts considerably, some other approach or hyper-parameters might perform better. As many approaches exist and each one of them can have multiple hyper-parameters to optimize, it requires great effort from the data scientist to manually find the optimal model and hyper-parameters. These AutoML tools help their customers to use machine learning models in their projects more efficiently than more traditional ways. Simply put; user inputs data, tells the system what kind of data is in each row and what column is the one to be predicted, and in what way the model can be accessed (deployed). The AutoML identifies the problem, normalizes and standardizes the data, and tests most suitable machine learning models for the training data with different hyper-parameters. User is given these models as a complete list ranked by their accuracy score from which user can choose what model to use in his or her project. However, some tools do not provide any freedom to choose the model, and it will naturally use the one which performed best.

These automated tools allow faster and easier development of machine learning models by using automated procedures to decide the best algorithms and data preparations for the provided learning data. However, we are interested in reviewing the platform for this specific problem with the previously mentioned way where data scientist desire to develop a simple yet effective model in his or her terms. Furthermore, adding to our original problem in a case where these automated tools exist in the platform, we examine their usability, transparency, and understandability.

Deploying the model is having an interface or ability to use the model as a resource to predict with the input data. The environment where the software application is developed can make requirements for accessing and using the model (e.g. no internet connection).

However, the machine learning platform can also make requirements for the software application (e.g. internet connection is required). In chapter 5.3, we discussed how data scientist could share his or her developed machine learning model for the software developer. When we review these machine learning platforms, we examine what ways are possible in the platform and what are the least effortless for both of these persons and also what different drawbacks these ways posses.

When the deployed model is available for the software application, we examine the journey to achieve the said model from a technical point of view, but also a more abstract point of view. This examination gives us an understanding of the life cycle and insights about the pitfalls different methods and platforms possess.

In our review process, the software developer has developed a simple Python application to test the model presented by data scientist, figure 6.3. Software developer can implement the provided model by either as an HTTP endpoint with a schema, PMML, ONNX or blob. As the Python programming language is diverse, these methods in the presented test application could be used in a wide variety of applications; e.g. in server-side, FPGAs [1] or handheld devices. The application consists of a versatile user interface with different options to add the model; such as previously mentioned, importing the model as PMML or ONNX, importing the model as a serialized object in Python, or using the machine learning platform's HTTP endpoints (REST API). After implementing the machine learning model to the application, a simple test is run to validate the correct functionality of the model.

Microsoft Azure Machine Learning

Microsoft Azure cloud platform provides a wide variety of tools such as virtual servers, databases, blockchain, DevOps and naturally machine learning. A machine learning platform is an enterprise-grade tool for controlling the whole life cycle of a machine learning

[1] Field-programmable gate array, a reprogrammable microchip

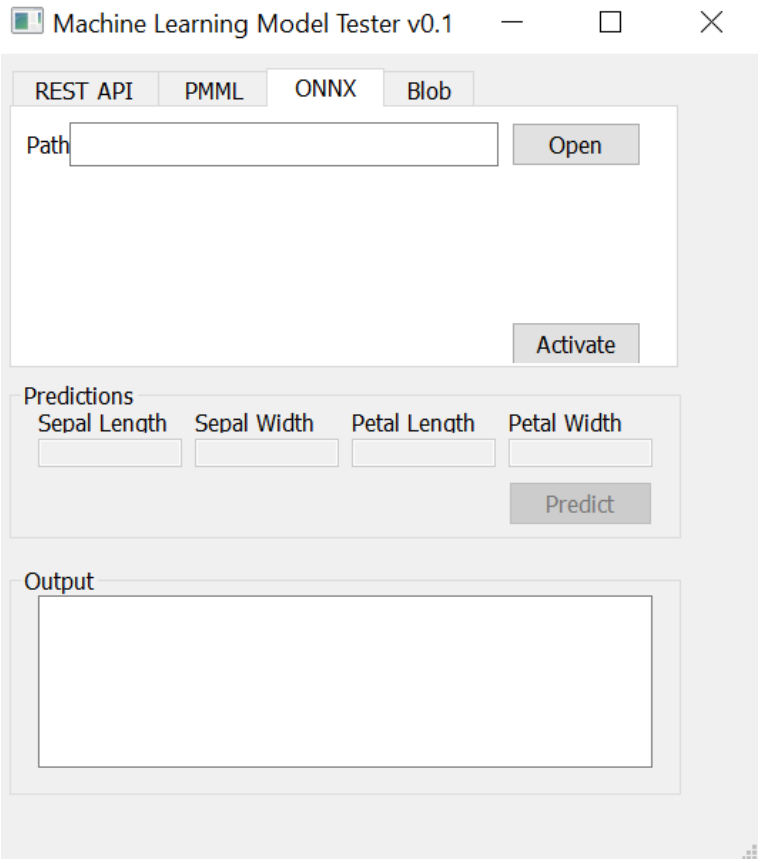

Figure 6.3: Application developed in Python programming language to test the models provided by different platforms.

project over the web interface, and interfaces for using whole Azure cloud platform exist in many different programming languages and environments. Microsoft is using a new term in the machine learning field, MLOps (DevOps for Machine Learning). It combines the data mining knowledge from the data scientist, continuous integration and continuous delivery know-how from the development team and the demand of the constant changes in the end product to enable full life cycle of the machine learning project [44]. Microsoft is trying to lower the bar for skills needed to be part of this kind of projects by making them as easy as possible.

Using the platform Microsoft Account and either a free trial period or pay-as-you-go plan are required to do the actual calculations; in our case training, evaluating and deploy-

ing the model, and as well using the model to make the predictions. This platform has two options to develop the machine learning model; Designer and AutoML, and both of these are accessed over the web interface, yet they could be used over Azure SDK in different programming languages. In the Designer, it is also possible to insert custom Python scripts to achieve more complex functions or procedures than the pre-made modules offer.

Using this platform starts by creating necessary resource group and workspace, which holds all the information and resources regarding this machine learning project. For both of these ways, a compute target (i.e. virtual machine) needs to be created in the platform. Importing the data set to the platform can be either uploading the file as CSV-file to the file storage or reading it directly from an URL.

The Designer is a Drag and Drop editor with pre-made modules which cover the most used procedures when creating machine learning model, such as data normalizations and transforms, feature extraction and selection, machine learning algorithms and evaluation of the built models. This designer does not have the *k*-NN algorithm as an available module, although it is possible to implement it with a *Create Python model* -module. This will restrict using some of the pre-made modules, such as *Evaluate Model*-module which can be compensated by using *Execute Python script*-module where the model evaluation is done manually with Python. After the model has been successfully trained, the platform has a tool to deploy it as an HTTP endpoint quickly. The endpoint can be secured by Token-based or with a Key-based authentication and with using TLS (Transport Layer Security). This Designer tool does not have modules for easily export the model in different formats and in case these models are needed they must be created with custom Python script either in the *Create Python model*-module or in the *Create Python model*-module. This is mildly inconvenient as the libraries in Python are easy to use for this task. However, it makes things complicated to transfer the generated model files to software developer as they are stored in persistent file storage in the cloud environment.

For more a controlled development environment, this platform offers an ability to

develop machine learning models with the Jupyter Notebook, which is an open-source environment and used by many in the data science field. In its simplest form, it shows the code in code-blocks and prints outs or graphs out information after the code-block has been run. This makes reading the code and understanding it much more straightforward than reading only code. Nevertheless, running the Jupyter Notebook does precisely the same as running the same code in plain Python or R.

The documentation data scientist can provide for the software developer when deploying the model as an HTTP endpoint, are the endpoint URL and as well the information for Token or Key authentication. The platform also offers example codes in C#, Python, R, and to further expand the usage of the endpoint. Furthermore, Swagger documentation of the endpoint is available. This documentation expresses OpenApi Specification for the RESTful API (i.e. the endpoint) which can be used in a wide variety of applications to ease the development.

The AutoML is a tool to let all the tasks of model development and evaluation for the platform. User can similarly import the data as in the Designer and only set the feature for prediction (in this case, the 'species' column in the dataset). The tool runs dozens of different machine learning algorithms to find the most suitable one for the given data. For each model, a simple accuracy score is given, yet the user can further examine the model for more detailed metrics about the model performance. The chosen model can be deployed as an HTTP endpoint or downloaded as a Python blob -object. In the case, the AutoML tool was used the models could be downloaded and saved in PMML or ONNX format through the SDK as using the web-interface this is not possible.

To recap the Azure Machine Learning platform; it provides the user with a simple interface to develop and deploy the models. Simplicity is lost when something more challenging task is required, yet Microsoft has provided users with proper documentation regarding their platform and SDKs. The data scientist can provide the trained machine learning model for the software developer in many ways; HTTP endpoint, PMML, ONNX

or as a blob. All of these, the software developer can implement in Python environment and as well in other environments with a little amount of effort.

Amazon Web Service SageMaker

Similar to the Microsoft Azure cloud platform, Amazon Web Services offers a wide variety of tools to develop applications in the cloud environment. Their machine learning focused tools help in a specific kind of tasks, natural language processing, forecasting, real-time personalization, visual and speech recognition. A tool called SageMaker enables the user to develop machine learning models as he or she wishes. The SageMaker works mainly over the web user interface with the Jupyter Notebooks and addition to that; different SDKs exist for popular programming languages, such as Python, R and Java. Easy to use interface with the drag and drop ability is not available on this service, causing it to be more challenging for the inexperienced user. Due to this reason, we can see that the target end-user group is different from Microsoft Azure Machine Learning platform.

When using the web IDE (Integrated Developed Environment) to create the machine learning model is done in the Jupyter Notebook with a Python programming language. SageMaker uses Docker containers for building and for deploying the machine learning models. These containers are controlled, and isolated environments where only necessary modules and services are loaded. The container approach gives consistency for developing and deploying the machine learning models by making the environment always the same and robust in security as they do not have access to interfaces or modules which are not allowed.

In the Marketplace, user can use previously created Docker files containing specific machine learning algorithm to train them with the user-provided data or even use pre-trained models. These pre-trained models are created to solve specific tasks in the real environment, such as computer vision, natural language processing, speech recognition,

and image recognition. Using these models can add value quickly for the user, yet the user has not a clear insight into the model or what was the dataset used to train the model.

In a case, the user desire to create their machine learning model from scratch, user can use pre-made Docker file (i.e. the container). The container includes all the necessary libraries and environment to write the model or user can create his or her own Docker file to precisely define what the required libraries for the model are. Either way, the container is necessary to have for training the model and using it for prediction by SageMaker. After the user have created the model and added it to the container with the specific functions and parameters added then the SageMaker could use it on training and prediction tasks. Creating the own model gives the most flexible implementation. However, it is more time consuming, and knowledge about creating the model in Python or R is required and as well as knowledge about Docker containers.

The SageMaker is working with the containers, which is a method the software developers are familiar with, yet they do not possess the knowledge to develop the machine learning models. In a case the data scientist is using the pre-made models and containers, he or she can provide an HTTP endpoint quickly for the software developer to consume over the internet connection. However, in our case where we would like to use our k-NN algorithm and data to make predictions, we would need either software developer to use the SageMaker for creating the containers for the data scientist or data scientist would need to learn creating the containers by themself. Either way is cumbersome in some extend yet not too much. Creating the blob, PMML or ONNX file from the model is effortless as we can run the Python code in the container and save these files in the file storage.

The AWS cloud platform is versatile to achieve a wide range of usage, yet this causes the platform to be sophisticated. However, the excellent documentation of the platform helps to develop applications on it. In essence, the SageMaker tool is a tool to handle a specific type of containers to train machine learning models and use them for predictions.

The result to our initial problem is similar to the Microsoft Azure Machine Learning, we managed to deliver HTTP endpoint, PMML, ONNX and a blob to the software developer, yet it was achieved in a way which required additional knowledge about Docker containers. SageMaker's strong point is to provide the users with a wide range of methods and tools for managing the operations of the machine learning project [45].

Google Cloud AI Platform

Google, as one of the biggest companies in the cloud business, has many tools and platforms to take away the cumbersome manual task of running its servers. Their AI and machine learning platform are called Google Cloud AI Platform. It is similar to Microsoft Azure Machine Learning platform for enabling users to develop his or her machine learning models in the Jupyter Notebook environment or use pre-made models to achieve the desired result. As we have previously mentioned, this notebook approach gives us the same freedom as running Python scripts in our computers or servers. Importing and exporting existing machine learning models in Python, PMML, ONNX or in blob format is not a challenging task with the Jupyter Notebook environment. The cloud platform has easy to use file storage system for using files or data inside the platform.

The build-in machine learning algorithms; XGBoost, Distributed XGBoost, Linear Learner, Wide and Deep learner, Image Classification and Image Object Classification gives user all-around and necessary algorithms to develop solid estimators for the required tasks. As for the initial problem, we need to implement the k-NN algorithm manually in the Jupyter Notebook instance with Python. After saving the un-trained model to the file storage as a model artefact, we can create a job in the platform to train the model for the data we have also added to the file storage. When using the Jupyter Notebook, we can easily export the trained model in PMML-, ONNX- or in blob-format to the said file storage to be transferred for the software developer. As for the models trained with the build-in algorithms; it is not easily possible to save the model in the previously mentioned

format, and due to this reason, the software developer would need to use Google's Python SDK or HTTP requests to receive predictions for the given input data. Additional data pre- and post-processing can be implemented to the data flow if necessary. Evaluating the model performance can either be used by creating Evaluation job in the platform or manually in the Jupyter Notebook using the metrics necessary for the task.

Similar to Amazon's and Microsoft's cloud platform, Google's cloud platform has the tools to quickly and efficiently develop machine learning applications, yet developing more demanding applications would require additional knowledge about the platform and technologies used.

IBM Watson Studio

Watson is an artificial intelligence developed by IBM from 2007 up to this day. Its main functionality understands human inserted questions in the natural language, and to answer them precisely. Around Watson, IBM has developed infrastructure to offer various tools for companies and individuals to benefit from. These tools vary from understanding the natural language to monitoring complex data streams from IT projects.

As for machine learning, the tool IBM has created is called IBM Watson Studio. After creating the project and inserting the data for the project as an asset; we have three options to chose from. Make a Jupyter Notebook instance, import already trained machine learning model in PMML format or use the AutoAI tool. After creating and selecting Machine Learning Service Instance, we can use these methods to develop the machine learning model. Using the Jupyter Notebook instance does not differ from previously mentioned platforms as the user has complete freedom to develop and evaluate his or her machine learning model as they see best. The Notebook environments can either be in Python, R or Scala programming languages; this enables versatile platform compared to the previously mentioned platforms as they only had Python- and R-notebooks available. Watson Studio is the only one which can import already trained machine learning model

in PMML format. This will make faster deployment in the platform in the case only PMML file is given for the software developer who is using the platform without any data science knowledge. Using the AutoAI tool is to automatically train many different models for the given dataset and choosing the best performing one for the deployment.

The primary method to deploy and use the trained model for predictions is over web service, i.e. the HTTP endpoint. The platform provides example codes in cURL, Java, JavaScript, Python and in Scala to use the HTTP endpoint. When using the Jupyter Notebook we can easily export the model in PMML, ONNX or blob-format, yet when using the AutoAI tool, we can create the HTTP endpoint or save it as a Notebook to further increase the possibilities to export the model in different formats.

6.2 Overview of Review

Using all of these platforms for our simple prediction task has given us valuable information to answer the previously mentioned questions in chapter 5.4. Using different cloud platforms has become a norm in the IT industry as it takes away the burden and required knowledge (i.e. additional employees) to maintain the server infrastructure from the company. Creating the machine learning model in all of these platforms were easy and not challenging.

How cumbersome it was to implement the model into the software developer's application? All of these platforms performed well regarding this question, and we were able to deploy the machine learning model as an HTTP endpoint or export it in different formats (Python/R Notebook, PMML, ONNX, and as a blob) for us to use in the software developer's application. The main reason to achieve this was the fact that all of these platforms were able to run the Jupyter Notebook environment in the same language as the software developer's application (i.e. Python). Additionally, suppose the software developer would not desire to enter the platform through the web interface. In that case,

he or she could use the platform's provided SDK to develop and use the trained models. Using SDKs can be challenging as they are unique to their specific platforms compared to the commonly used formats which could be seen as standardization (HTTP endpoint, PMML, ONNX, or blob).

How well the user understood the platform and inner functionality of the model? As the Jupyter Notebook environment was the common factor in these platforms, the user should be fully aware and understand of the functions and models used. All of these platforms use different methods to track down the trained models to validate the model's operations in the production environment. Nevertheless, when using the HTTP endpoint deployment, if the used model information is not documented well enough, the software developer could not be sure what model version the endpoint would use. Also, as the model behave how it has been trained, the training data is essential to be tracked down if needed.

These platforms provide storage systems in the cloud to save the files and models for easy access inside the platform. Opposite to Jupyter Notebook environment, the AutoML tool provided the user complicated models which at first glance could not be understood clearly. For example, IBM Watson Studio AutoAI tested over twenty different models with different parameters, and it suggested to choose the one with the highest accuracy. Although, metrics of specific properties were displayed for the user, yet it would require additional expertise in statistics to clearly understand how the model would behave in real-world applications. A model created by AutoML tools can be seen as a black box for the inexperienced user. This creates a dilemma, these tools are easy to use without a good knowledge of data science, yet it requires good knowledge to pick the best black box when juggling multiple black boxes.

Explanation of how machine learning model reasons to the given input data can not be explained by "it is AI". There was some minor difference between platforms regarding how easy it was to receive different metrics about the developed machine learning model.

The platforms were more informative for the user when the user-developed the machine learning model with the pre-made modules or systems (AutoML).

How the prediction was justified or can be explained? These platforms failed to provide substantial justification in the predictions of how and why they predicted it like that. In our case, the k-NN classifier machine learning model is easy to justify as the prediction belongs to the majority of the k nearest neighbours. However, the Jupyter Notebook environment enables the user to have full control of the evaluation metrics of the developed model or justification of the choices. For example, it is easy to visually represent the decisions of a Decision Trees by drawing it as a graph. When using the AutoML tools or evaluation modules, these platforms could provide metrics on how well the machine learning model could perform for the future unknown data. This would not provide anything for explaining the model's predictions yet to validate that the model is working as intended and could be used in the production. A recent study from Obermeyer, Powers, Vogeli, *et al.* [46] shows that US health system was racially biased and they had to validate that they did not discriminate by the colour of the skin or heritage, but it was biased because they used certain choices in the labels.

What were the benefits and drawbacks using the model the way it was deployed? Deploying the model for prediction as an HTTP endpoint into the platform causes additional cost due to the computing compared to the PMML, ONNX or blob ones as the prediction computing is done on the device where the application is run or part of the application. This separates the used deployment models into two categories; online (HTTP endpoint) and local (PMML, ONNX, blob). Although, a model deployed on to the server as an HTTP endpoint could still use 'local' models to serve the predictions through the HTTP endpoint interface.

The prediction is not using many resources on the trained model, yet if millions of customers are using the said machine learning model to predict daily or even hourly, the cost of using a cloud platform can increase quickly. The ability to use internet connection

make the most significant distinct separation of possible methods to use on maintaining the machine learning model. New data is all the time gathered and updating the machine learning model could be beneficial to produce more accurate results. Receiving a new updated version of the model in these formats can be challenging in the case, there is no internet connection or if the models are complicated and enormous in their file size.

The models implemented into the software application can have lower latency when making the prediction inside the application environment compared to using HTTP endpoint over an internet connection, yet when using HTTP endpoint additional monitoring services can be created to track down the usage of the machine learning model and its predictions. When using the AutoML tool or pre-made models of the platform to develop and train the model, it restricts exporting options of the model in PMML, ONNX or blob formats.

As we evaluated these platforms by the example project and answered the four questions for each one of them, a question arose about what are the factors to keep in mind when choosing between these and other platforms. Henceforth, we created heuristics for software developers to choose a platform for their next data-driven project.

6.3 Summary

To sum it up, benefits and drawbacks depending on the implementation of the model and capabilities of the software application.

We used these platforms in a simple way to understand the architecture of the platforms and their capabilities. In essence, they are similar to each other, as each one has their ways to produce similar results. We learned that each one of them provides ways to deploy the machine learning model as an HTTP endpoint and with a virtual machine environment, and to develop the models in Jupyter Notebooks or via SDK. The tools using AutoML could not develop them with the Jupyter Notebook as per se they could use SDK

inside the Jupyter Notebook environment to achieve this.

IBM Watson Studio was the only one which could import natively previously created machine learning model in PMML format to the platform, and their AutoAI could export the developed model as a complete Jupyter Notebook file in Python language. These two features enable more versatile usage of the platform compared to others. Moreover, Azure ML was the only one which could export AutoML model in PMML, ONNX or as a blob with the SDK. We can conclude that preferable usage of these AutoML tools is to deploy them as an HTTP endpoint and further make the predictions in the platform they were developed.

These platforms enforce the gap between a data scientist and software developer in case they are two different persons. However, in essence, these platforms enforce software developer to delve into the machine learning and data science either as to develop necessary models by him- or herself, or to support the data scientist to use these platforms. For the software developer to decide what platform to use in a data-driven project, we go through simple heuristics to assist in deciding the platform. Although, often deciding the platform comes from inside the company or from the customer requirements.

7 Results

On this chapter, we go through the results obtained in this thesis. By analyzing the elements in the data-driven project environment, we have gained knowledge to understand its nature and challenges. The first section answers research questions, and in the second section, a heuristics for choosing a machine learning platform is presented. Finally, the limitations and future research is presented in the third section.

7.1 Development of Data-driven Projects

Answering the research questions gives us an understanding of the current and future development of data-driven projects. This helps us to develop new methods and ways to answer the challenges in the future.

Q1 asked how traditional data mining process models are being implemented in modern software development. Our study found out that these process models fit badly into modern agile software development. Yet the data mining process models provide valuable insights for the software developers as a form of educational material. These insights help the developer to avoid possible problems later in the development or in the maintenance phase. New data mining process models, which try to combine them to modern software development, have been created by companies and academic researchers. However, there are no studies about their benefits or statistics about their usage in these data-driven projects.

Q2 asked how these two important roles work together in these data-driven projects,

and what are the factors enabling their trouble-free working. We described the roles and their skills, and we found out the differences between them. Naturally, the data scientist has more detailed knowledge about the data, and software engineer has more expertise in the technical development of software. There has been a gap between these two roles in their collaboration. This gap can probably never be closed, yet these the cloud-based platforms, and modern agile development methods help in the communication and collaboration between these roles. Also, further educating the software engineers about data science, and about these data mining process models can mitigate future problems in this communication and collaboration.

Q3 asked about how well machine learning can be implemented on software development projects with cloud-based machine learning platforms. The question was also about how easily software engineer can learn these platforms, and what are the factors to look for when choosing the machine learning platform.

The platforms give a wide variety of ways to develop the predictive machine learning model, and export it in different formats or deploy it as an HTTP endpoint. Most effortless data-driven software development is done when the development of the software is done on the same platform as the machine learning development. However, it is not cumbersome to export a trained machine learning model in a certain format and implementing it in the software by its native programming language.

As we found out in Q2, a software engineer should have the necessary skills to understand quickly new platforms, tools, and frameworks. Understanding these cloud-based machine learning platforms are not a hard job for a proficient software engineer. However, understanding the data science aspect of the platform can be troublesome. The enterprise-grade platforms have automatic tools and pre-made solutions to implement predictive functionalities, which software engineer can use without proper understanding for the software in development. Implementing any system into the software without proper knowledge of the underlying functionalities, or capabilities can be seen as bad practise.

Often the final decision to choose the used cloud-based platform comes from somewhere else than the software development team. However, in a case software engineers have part of making this decision, or they are solely in charge of deciding the used platform, we have created heuristics to help on making the decision. These heuristics cover the important elements which should be taken into account when making this decision. Yet, they should not be followed precisely and adjusted if necessary.

Q4 summarizes previous questions by asking what are the choices to make during data-driven software development. We found out that the optimal approach for data-driven IT project is to have a valid process model or methodology, precise roles, and versatile tools or platforms for the people working in the project. Each one of these provides valuable interactions to other elements on this data-driven project platform, see figure 7.1

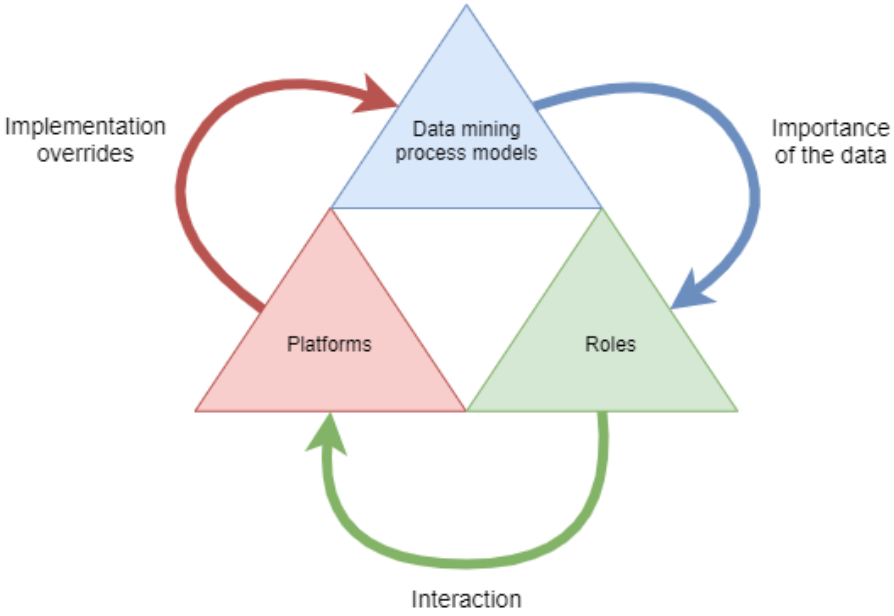

Figure 7.1: The effect of the elements in data-driven project environment.

The software development industry is commonly using agile development methods to iteratively deliver software until the requirements from the stakeholders are met. As this is the environment data-driven projects exists, the agile methods need to take into account

the data-focused tasks and requirements. It is not feasible for the data mining process models to start implementing agile software development tasks and process models into them. Hence, the valid methodology is to use an agile methodology which has been extended with tasks to ensure the requirements of the data.

The whole life cycle of the data-driven project must be transparent and repeatable as different laws require to validate methods used to ensure no discrimination have not happened. Furthermore, society and stakeholders require an explanation of the methods used and the security of the data. In this kind of situation, the company could be in great trouble if they can not reproduce and validate their model used in data mining, machine learning or artificial intelligence applications. Scientists are developing models for the purpose to explain the reasoning behind the decision of the predictive models.

The requirements caused by the nature of the data causes challenges for these two professions, data scientist and software developer, working on these data-driven projects. Good collaboration allows proper data processing and development at different stages of the project. Providing knowledge and added value to the stakeholders can be achieved with the help of appropriate tools and expertise to create, evaluate, monitor, and deploy models.

7.2 Heuristics for Choosing Machine learning Platform

Heuristics can provide valuable guidance when choosing a different system over others. These are simple points to consider, and these are not to be followed precisely - similar to the Agile manifesto - these heuristics are guidelines and *rules of thumb* to aid the reader by writer's experience related to the subject at hand. Making a decision from experience or logic is always the best one to tackle the problem at hand, and in a case where a user lacks these, he or she can use heuristics for aid. [47]

Heuristics are used in software, website or game development for evaluating user

interface and user experience to provide a robust outcome of the development, and to satisfy a majority of the end-users who are using it [48][49][50][51]. Their value comes from the fact they can be used when a person has limitations to the matter at hand, in our case the user is clueless about the machine learning platforms.

As we went through the most used machine learning platforms, we can provide following heuristics for software engineering students who are inexperienced in data science and machine learning platforms, and for project managers who are leading these data-driven projects. Or other persons who are involved with enterprise-grade machine learning. Even though there might be one optimal solution for choosing the right platform for individual and his or her tasks for current and in future projects, we are interested about finding a platform which is sufficient and suitable enough for individual's use.

Past experience guides you to the future. Aim for the platform which you have used before or are familiar with. Go through the marketing material or documentation about the platform and find out what kind of technology is required or preferred to enable the functionality of the platform entirely. For example, if you are familiar with Docker images and AWS's platform is heavily focused on Docker images, their platform would be an obvious choice for you. Choosing a platform which you are even a little bit familiar can provide faster results.

Documentation brings knowledge about the platform's capabilities. Create a checklist of the tasks you know you need to achieve at the moment (and in the future if possible) and check if the platform you are evaluating can achieve them. Different platforms vary significantly in their capabilities as they might be specialized in certain types of tasks, for example; Platform created by Peltarion is restricted only on models using image recognition or natural language processing.

All of these platforms we went through provide a way to develop in Jupyter Notebook environment. As it is common in the data science field, a prerequisite for using any of these platforms would be the knowledge to develop machine learning models in the

Jupyter Notebook environment. Many other platforms exist which do not use Jupyter Notebook, and these should be ignored when deciding machine learning platform as they are too far away from the methods people are commonly using.

Justification for the masses. Use a platform where you can create a system or procedure to answer the "why and how" your machine learning model is predicting and making a decision. As society begins evolving more and more around data, and Artificial Intelligence is being implemented in everyday objects or tasks. This has caused people becoming interested in how these systems are using the data they are gathering or provided with and how they are working to be sure they are treating people fairly.

Compatibility of the environment. Find out what kind of systems your company or you would use now or in the future regarding machine learning projects as some of the platforms could be easier to implement to the machine learning project. For example; Microsoft has created a complete toolset for small to international size companies. These tools include tools for sharing data and monitor business-relevant information, such as Microsoft Power BI. It would be beneficial to use a platform which is in the same environment as your company's other tools and platforms are if possible.

Let transparency be with you. Similar to the justification heuristic, the transparency of the system for the user is essential. After using the system for a while, ponder how many times you have achieved some tasks. Nevertheless, you have not clearly understood what you were doing and still received the desired outcome[1]. This guides you to realize how transparent the system is for you.

Following these heuristics might require additional usage of the system to fully get the knowledge about how well are you following them, as the documentation varies significantly among these platforms and can not be used entirely to validate the performance of the platform.

The created heuristics can guide you to decide the machine learning platform for you

[1] similar to copying functions from StackOverflow without understanding their inner functionality

to get experience. However, in the company level, a more wide aspect of the machine learning platform review would be required, such as future costs, and compatibility to with the company's other systems. Different heuristics exist for different tasks, and as this is one of the possible heuristics for a particular task, it is highly advised to evaluate different heuristics and strategies for the tasks at hand [52].

7.3 Limitations and Future Research

It is a limitation of this thesis for not having hands-on experience in developing machine learning models in a large company with these presented platforms. In the future, it would be beneficial to research modern software development in large companies to compare how much CRISP-DM or similar process models or methodologies, are used in projects having vast amounts of data. And, a complete heuristic evaluation of these machine learning platforms could be performed by heuristics created by Nielsen [50] to evaluate their usability. Further research is needed to validate the findings of this thesis by performing cross-industry inquiry for the data scientists and software developers working in this field in different size companies.

8 Discussion & Conclusion

After reviewing these enterprise-grade machine learning platforms, we have found exciting insights. We used these platforms in an example project and studied several scientific articles regarding data mining, machine learning, software engineering, process models and business intelligence. These insights are the factors on the changes from static data-driven projects to more dynamic modern machine learning.

Firstly, as we learned from the mid-1970s software crisis, it is vital to have reliable procedures to guide the development of a software application. When doing straightforward data mining projects, the data mining process models have worked well. The process models provide easy to follow step-by-step instructions for the users by offering a robust process with reliable outcomes. However, they are sequential waterfall-like process, which can fit poorly for the modern agile software development. Due to this, it is easy to underestimate the benefits of these data mining process models. Nevertheless, they have provided solid frameworks for these data-driven projects.

The Agile methodology is becoming the mainstream method for software development as it has been proven to work well when developing applications in a wide variety of projects. The iterative process with continuous delivery allows the development team to receive feedback from the stakeholders. This has a clear advantage to the previously used waterfall approaches, as agile enables changing the end product when the requirements change.

Artificial intelligence and machine learning have become buzz-words and hyped a lot

in the business sector for their ability to offer new solutions for companies in different fields. They are being implemented into projects which could benefit greatly from these technologies. However, developers current project management framework fall short on the requirements that data-intensive application type brings along.

Nevertheless, as we have seen, the Agile software development has given software developers ways to develop the software in an *agile* way, and as the software development is using more and more AI and ML in their software. It is logical to conclude that there will be a need for people who are developing software and still understand the field of data science. Enterprise solutions are being developed to implement AI and ML into different projects quickly. In the future, these would be developed further with the more easily understandable ways to increase the justification of it. However, compared to the AI or ML as a science, their algorithms are driven to become even more accurate and better performing when the data is limited - which has been the main problem of AI and ML for their need for large quantities of data.

In essence, the most sophisticated data mining process models can work well if the result of it is treated correctly in software development. There is a great need in adopting any of these already presented process models into modern software development as according to KDnuggets [24] they are still being used.

Secondly, artificial intelligence and machine learning are being implemented into projects to ease or bring added value to the stakeholders. Fitting them to a project can not be treated as a simple module which developers could use as a black-box or without proper knowledge about the requirements. The data has an impact on the whole project, and due to this reason, data with its requirements can not be ignored. The data-driven project's success ensued from the handling of the data, and due to this reason, an adept person is required to be in the project's development team. The society is becoming aware of the fact that the data is valuable and a great asset. Personal information is collected with and without consent. Even if the data is collected with consent, it can be used in-

tentionally wrongfully. There have been cases where IT systems have been noticed to be inconsistent with their decisions, and people have started to question these decisions. In a case, people have detected any discrimination; they are demanding companies to justify the decisions their products are making, i.e. the decisions the software, including artificial intelligence or machine learning, are making. Generally, problems like this are caused by improper usage of the data, and not by wrongfully making artificial intelligence or machine learning models.

As artificial intelligence and machine learning are behaving based on experience. As humans can be affected either subconsciously or by force [53][54] and artificial intelligence or machine learning models can be affected as well. By altering the training algorithm or selecting specific training data set, the algorithm can be altered to perform as the developer desires. Nevertheless, sometimes a machine learning algorithm can learn 'wrongly' even if there were no malicious intent when it was developed [46][9]. In the future justification and validation of models used in customer, applications are growing to satisfy the requirements made by the customers, or from governments. They demand to be treated equally, and they want their data to be safe.

Thirdly, as we have noticed, the data-driven software development project requires some of the members to have expertise from data science and software development. For now, the leading developers in projects will be the software developers with his or her training. To morph software developers into data scientists is not feasible without great effort. On the data-driven projects, the data scientist and software developer are becoming closer together by their work tasks. Both of them are giving their input on the tasks at hand. Yet, we are starting to see the rise of a role combining both of these roles, a data engineer. As a role, data engineer manages both sides of the coin. The person working on this role has a more substantial background either on data science or from software engineering. With this background, data engineer can bump into problems if strong skills are needed of either one.

There are not many education instances to teach data engineering specifically. However, we have noticed universities starting to take data engineer in their curriculum to provide employees with a broader skillset for the companies. Motivation to have this kind of person in a project is to mitigate the support data scientist or software developer would need during the project about the field he or she is not expert of.

The software developer receives some knowledge about data science through the learning of these data mining process models. It is crucial to have a basic understanding of them when working with the data-driven projects in order not to make the whole project crumble down by stepping on some simple pitfalls.

Fourthly, even if we have data engineer as a role in our data-driven project, it still does not solve all problems. The models are becoming complex and enormous, making them harder to explain and justify their decisions. There is not much room to provide models which can not be explained easily, as it would further increase the gap data engineer is trying to fill.

The IT field is moving towards using machine learning models which can easily explain their decisions [55]. These models need to be developed from the ground up to provide users with the justification as it is cumbersome to create these explanations from pre-trained models [56]. The more complex the model is, such as Neural Networks, the more challenging it is to create these explanations [57]. An approach called eXplainable AI (XAI) is bringing new ideas to create models which can explain themselves. Rudin [58] suggests developing models in a way they can be interpreted by humans, especially when they are making high stake decisions.

Lastly, these cloud-based platforms provide an environment where data scientist, software developer, or data engineer can comfortably develop, and maintain data-driven software projects. Even if these explaining models are work in the progress, and these platforms are enabling a wide variety of methods to be used in the project, they still enforce people to use AutoML tools. Using these tools allows developing predictive models to be

used as fast as possible in the final product. This kind of AutoML tools can use any given algorithm to create the most performing model for the giving task, by optimizing either the algorithm itself or the hyperparameters of it. This increase the chance the developer and end-users do not know how it is working or what training data was used. As for this reason, machine learning platforms can be seen to be behind for offering methods to develop models which also explain themselves firmly.

These platforms can use Jupyter Notebook to develop the models manually, which can then be developed so that they can explain their predictions. The Jupyter Notebook environment has become the norm in the data science field, and as we saw in the platform review, all of the platforms were using it. This kind of approach of standard technology is always greatly apprised in the fast-growing and ever-changing field of Information Technology.

Combining this insight with the third insight, we can understand that the primary user of these platforms is the software developers when developing data-driven projects. The software developers are required to step out of their comfort zone and understand the world of data science. However, the platforms are capable of providing ways for the software developer to understand data science, predictive models, and data in a transparent way.

These insights also describe the environment in which data-driven projects are developed. Understanding these insights during the development can help to organize the project, and improve the quality of the result, i.e. the software. The environment in which the data-driven project is located is broad and with challenging elements. The companies are still finding best practices to develop this kind of projects, and it looks like the Agile methodology will be strongly involved in this development.

These insights also assisted in creating heuristics for choosing a machine learning platform. Following these heuristics can help individual software developer to choose the most appropriate platform for their use.

To conclude, we studied the complex nature of modern data-driven projects and found the insights that are causing the evolution of these projects. The data and how to store it has significantly evolved over the 50 years. The evolution of methods to store data from simple punch cards into modern HDDs has given developers the possibility to store a tremendous amount of data in a cost-efficient way. As the data is gathered in enormous amount, from multiple sources and over a long period, the systems have had to change the way they can manage the complexity and connectivity of the data across different systems. The old data systems and ways to *discover knowledge* have been poorly connected to other systems, yet modern systems allow easier ways to connect between them. Projects and products have evolved to be dynamic as possible and transfer data either locally or over the internet. The traditional data mining process models have problems to be not agile enough to fit into modern software development, and new ways to develop the data-driven software products are in development by researchers.

As the data can shift and evolve, monitoring the developed system is required. The development needs to continue even after the deployment, in the same essence as DevOps mindset yet for the machine learning predictive models. As the standard DevOps software is evolving during its operation, the previous versions are not interesting anymore if the new version is successfully added into production.

Nevertheless, when working with these predictive models, previous versions with their training data are valuable to store in the case someone is demanding justification of a prediction which happened on the previous versions. Version control is equally important as with solid pipelines allow reproduction for validation purposes.

We have noticed that society is becoming interested in how a developed software is using machine learning are making predictions. Making any predictions of the given input data requires to provide some justification or reasoning for the prediction if someone demands it. There are ways to create methods to provide this justification, either the data scientist needs to build the justification into the used machine learning model, or the

software developer can use pre-made tools or models which can justify their decisions.

Our research has pointed out the traditional data-mining process models are being overrun by the software development process models and platforms. The data-mining process models are too rigid to fit into agile software development. They still provide a valuable understanding of the nature of the data and methods to handle it properly. This understanding gives software developers ways to avoid possible pitfalls and mitigate risks. The roles in developing these data-driven projects are a data scientist, software developer, and data engineer. The data engineer role can cover both expertise of these other two. However, the persons working on this role has more substantial expertise in either one of these roles. The software developer is still in the primary role for executing the development in the project. Because of that, the platforms used are necessary to be transparent for their users and especially for the software developers. All of the enterprise-grade machine learning platforms have extensive functionalities to satisfy even the most demanding needs of the stakeholders and the users. Nevertheless, there are differences between these platforms, and we have provided heuristics for choosing a platform for a data-driven project.

It can be argued that using any data mining related methodology or process model is not bringing anything to the table when developing modern software application revolving around data and its complex usage. The data mining or machine learning phase could be a module or a phase in a big architecture plan or workflow, and it would be sufficient to explain it as "it is Artificial Intelligence". As long it is providing correct results, everyone is happy. However, as we have seen, the machine learning field can be quite complex, not to even mention the several algorithms a user could use on any given task. There is no one fit for all solution, and using knowledge from data mining process models will provide valuable insights to keep in mind when developing software.

Data scientist and software developer could work on the project either simultaneously or alternately, yet they need ways to combine their expertise and results. One of these

ways is to use a machine learning platform. These platforms are providing them with comprehensive tools and necessary functions to achieve the tasks these two roles need to accomplish. Additionally, these platforms can give ways for the data scientist to make part of the software developer's tasks, and vice versa.

The platforms provide a great way to develop and deploy in an easily scalable environment, yet using the tools or pre-made models can hinder the sight of the inner workings of the whole product and the justification of the predictions. Contradictory of providing ways to closer the data scientists and developer team; these services provide the result as an API service which the end application can use over the standard HTTP connection or with a specific integration. This type of way leads to a black-box approach, shrouding the model and behaviour from the developers of the end application.

Acknowledgements

I want to show my greatest appreciation to Antero Järvi, whose comments and suggestions were innumerably valuable throughout my study. I am also in debt to Tapani Joelsson, whose opinions and information have helped me very much throughout the production of this thesis.

References

[1] Z. Stos-Gale and N. Gale, "The sources of mycenaean silver and lead", *Journal of Field Archaeology*, vol. 9, no. 4, pp. 467–485, 1982.

[2] A. Feelders, H. Daniels, and M. Holsheimer, "Methodological and practical aspects of data mining", *Information & Management*, vol. 37, no. 5, pp. 271–281, 2000.

[3] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011, ch. 1.

[4] D. Wegener and S. Rüping, "On integrating data mining into business processes", in *International Conference on Business Information Systems*, Springer, 2010, pp. 183–194.

[5] I. Bose and R. K. Mahapatra, "Business data mining—a machine learning perspective", *Information & management*, vol. 39, no. 3, pp. 211–225, 2001.

[6] M. L. Neufeld and M. Cornog, "Database history: From dinosaurs to compact discs", *Journal of the American society for information science*, vol. 37, no. 4, pp. 183–190, 1986.

[7] M.-S. Chen, J. Han, and P. S. Yu, "Data mining: An overview from a database perspective", *IEEE Transactions on Knowledge and data Engineering*, vol. 8, no. 6, pp. 866–883, 1996.

[8] K. Poland, M. P. McKay, D. Bruce, and E. Becic, "Fatal crash between a car operating with automated control systems and a tractor-semitrailer truck", *Traffic injury prevention*, vol. 19, no. sup2, S153–S156, 2018.

[9] P. Noor, "Can we trust ai not to further embed racial bias and prejudice?", *BMJ*, vol. 368, 2020.

[10] S. Ruggieri, D. Pedreschi, and F. Turini, "Data mining for discrimination discovery", *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, no. 2, pp. 1–40, 2010.

[11] G. Beranek, W. Zuser, and T. Grechenig, "Functional group roles in software engineering teams", in *Proceedings of the 2005 workshop on Human and social factors of software engineering*, 2005, pp. 1–7.

[12] D. Harris, *Kaggle now has 100k data scientists, but what's a data scientist?*, 2014. [Online]. Available: https://gigaom.com/2013/07/11/kaggle-now-has-100k-data-scientists-but-whats-a-data-scientist/.

[13] S. Miller, "Collaborative approaches needed to close the big data skills gap", *Journal of Organization design*, vol. 3, no. 1, pp. 26–30, 2014.

[14] K. Rodham and J. Gavin, "The ethics of using the internet to collect qualitative research data", *Research Ethics*, vol. 2, no. 3, pp. 92–97, 2006.

[15] C. Farr, "Here's everything you need to know about the cambridge analytica scandal", *CNBC*, 2020 (accessed September 20, 2020). [Online]. Available: https://www.cnbc.com/2018/04/05/facebook-building-8-explored-data-sharing-agreement-with-hospitals.html.

[16] S. Meredith, "Facebook sent a doctor on a secret mission to ask hospitals to share patient data", *CNBC*, 2020 (accessed September 20, 2020). [Online]. Available: https://www.cnbc.com/2018/03/21/facebook-cambridge-analytica-scandal-everything-you-need-to-know.html.

[17] T. Patil and T. Davenport, “Data scientist: The sexiest job of the 21st century”, *Harvard business review*, vol. 90, no. 10, pp. 70–76, 2012.

[18] I. Ozkaya, “What should a software engineer know?”, *IEEE Software*, vol. 37, no. 1, pp. 3–6, 2019.

[19] E. Meade, E. O’Keeffe, N. Lyons, D. Lynch, M. Yilmaz, U. Gulec, R. V. O’Connor, and P. M. Clarke, “The changing role of the software engineer”, in *European Conference on Software Process Improvement*, Springer, 2019, pp. 682–694.

[20] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases”, *AI magazine*, vol. 17, no. 3, pp. 37–37, 1996.

[21] R. J. Brachman and T. Anand, “The process of knowledge discovery in databases”, in *Advances in knowledge discovery and data mining*, American Association for Artificial Intelligence, 1996, pp. 37–57.

[22] U. Shafique and H. Qaiser, “A comparative study of data mining process models (kdd, crisp-dm and semma)”, *International Journal of Innovation and Scientific Research*, vol. 12, no. 1, pp. 217–222, 2014.

[23] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, R. Wirth, *et al.*, “Crisp-dm 1.0: Step-by-step data mining guide”, *SPSS inc*, vol. 16, 2000.

[24] KDnuggets. (2014). “What main methodology are you using for your analytics, data mining, or data science projects? poll”, [Online]. Available: `https://www.kdnuggets.com/polls/2014/analytics-data-mining-data-science-methodology.html` (visited on 08/09/2019).

[25] Ó. Marbán, G. Mariscal, and J. Segovia, “A data mining & knowledge discovery process model”, in *Data mining and knowledge discovery in real life applications*, IntechOpen, 2009.

[26] S. S. Rohanizadeh and M. M. Bameni, “A proposed data mining methodology and its application to industrial procedures”, 2009.

[27] R. Wirth and J. Hipp, "Crisp-dm: Towards a standard process model for data mining", in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, Citeseer, 2000, pp. 29–39.

[28] L. A. Kurgan and P. Musilek, "A survey of knowledge discovery and data mining process models", *The Knowledge Engineering Review*, vol. 21, no. 1, pp. 1–24, 2006.

[29] D. Olson and D. Delen, *Advanced Data Mining Techniques*. Jan. 2008, ISBN: 978-3-540-76916-3. DOI: 10.1007/978-3-540-76917-0.

[30] Z. Abbasi, G. Varsamopoulos, and S. K. Gupta, "Tacoma: Server and workload management in internet data centers considering cooling-computing power tradeoff and energy proportionality", *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 9, no. 2, pp. 1–37, 2012.

[31] Statista. (Aug. 2020). "It market revenue in the world from 2016 to 2021 (in billion u.s. dollars)", [Online]. Available: https://www.statista.com/forecasts/959558/it-revenue-in-the-world (visited on 05/08/2020).

[32] B. Fitzgerald, "Software crisis 2.0", *Computer*, vol. 45, no. 4, pp. 89–91, 2012.

[33] N. B. Ruparelia, "Software development lifecycle models", *ACM SIGSOFT Software Engineering Notes*, vol. 35, no. 3, pp. 8–13, 2010.

[34] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods: Review and analysis", *arXiv preprint arXiv:1709.08439*, 2017.

[35] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "Devops", *Ieee Software*, vol. 33, no. 3, pp. 94–100, 2016.

[36] S. Balaji and M. S. Murugaiyan, "Waterfall vs. v-model vs. agile: A comparative study on sdlc", *International Journal of Information Technology and Business Management*, vol. 2, no. 1, pp. 26–30, 2012.

[37] M. Alnoukari, Z. Alzoabi, and S. Hanna, "Applying adaptive software development (asd) agile modeling on predictive data mining applications: Asd-dm methodology", in *2008 International Symposium on Information Technology*, IEEE, vol. 2, 2008, pp. 1–6.

[38] Ó. Marbán, G. Mariscal, E. Menasalvas, and J. Segovia, "An engineering approach to data mining projects", in *International Conference on Intelligent Data Engineering and Automated Learning*, Springer, 2007, pp. 578–588.

[39] R. Parloff. (2016). "Why deep learning is suddenly changing your life", [Online]. Available: `https://fortune.com/longform/ai-artificial-intelligence-deep-machine-learning` (visited on 04/06/2020).

[40] S. Nidhra and J. Dondeti, "Black box and white box testing techniques-a literature review", *International Journal of Embedded Systems and Applications (IJESA)*, vol. 2, no. 2, pp. 29–50, 2012.

[41] P. Meunier, "Software transparency and purity", *Communications of the ACM*, vol. 51, no. 2, pp. 104–104, 2008.

[42] L. M. Cysneiros, M. Raffi, and J. C. S. do Prado Leite, "Software transparency as a key requirement for self-driving cars", in *2018 IEEE 26th International Requirements Engineering Conference (RE)*, IEEE, 2018, pp. 382–387.

[43] R. Fisher, *UCI machine learning repository*, 1936. [Online]. Available: `https://archive.ics.uci.edu/ml/datasets/Iris`.

[44] N. T. C.-F. CTO and V. E. ParallelM. (2018). "Mlops (and not just ml)", [Online]. Available: `https://www.aitrends.com/machine-learning/mlops-not-just-ml-business-new-competitive-frontier/` (visited on 02/01/2020).

[45] Amazon. (2020). "Amazon aws sagemaker", [Online]. Available: `https://aws.amazon.com/sagemaker/build/` (visited on 02/01/2020).

[46] Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan, "Dissecting racial bias in an algorithm used to manage the health of populations", *Science*, vol. 366, no. 6464, pp. 447–453, 2019.

[47] G. Gigerenzer, "Why heuristics work", *Perspectives on psychological science*, vol. 3, no. 1, pp. 20–29, 2008.

[48] R. Marinescu, "Measurement and quality in object-oriented design", in *21st IEEE International Conference on Software Maintenance (ICSM'05)*, IEEE, 2005, pp. 701–704.

[49] A. Sutcliffe, "Heuristic evaluation of website attractiveness and usability", in *International workshop on design, specification, and verification of interactive systems*, Springer, 2001, pp. 183–198.

[50] J. Nielsen, *Ten usability heuristics*, 2005.

[51] H. Desurvire, M. Caplan, and J. A. Toth, "Using heuristics to evaluate the playability of games", in *CHI'04 extended abstracts on Human factors in computing systems*, 2004, pp. 1509–1512.

[52] S. Talman, R. Toister, and S. Kraus, "Choosing between heuristics and strategies: An enhanced model for decision-making", in *International Joint Conference on Artificial Intelligence*, LAWRENCE ERLBAUM ASSOCIATES LTD, vol. 19, 2005, p. 324.

[53] F. Liang, V. Das, N. Kostyuk, and M. M. Hussain, "Constructing a data-driven society: China's social credit system as a state surveillance infrastructure", *Policy & Internet*, vol. 10, no. 4, pp. 415–453, 2018.

[54] Z. Raza, "China's 'political re-education'camps of xinjiang's uyghur muslims", *Asian Affairs*, vol. 50, no. 4, pp. 488–501, 2019.

[55] N. Narodytska, A. Shrotri, K. S. Meel, A. Ignatiev, and J. Marques-Silva, "Assessing heuristic machine learning explanations with model counting", in *International Conference on Theory and Applications of Satisfiability Testing*, Springer, 2019, pp. 267–278.

[56] D. A. Melis and T. Jaakkola, "Towards robust interpretability with self-explaining neural networks", in *Advances in Neural Information Processing Systems*, 2018, pp. 7775–7784.

[57] O. Li, H. Liu, C. Chen, and C. Rudin, "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions", *arXiv preprint arXiv:1710.04806*, 2017.

[58] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead", *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.