Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

Magnus Olai Aarvold
Wilhelm Jan Hartvig

Machine Learning on Complex Projects:

Multivariate time series data analysis through utilization of the sequential algorithm LSTM

Master's thesis in Mechanical Engineering
Supervisor: Nils Olsson
June 2021

NTNU
Norwegian University of
Science and Technology

Magnus Olai Aarvold
Wilhelm Jan Hartvig

Machine Learning on Complex Projects:

Multivariate time series data analysis through utilization of the sequential algorithm LSTM

Master's thesis in Mechanical Engineering
Supervisor: Nils Olsson
June 2021

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Department of Mechanical and
Industrial Engineering

Master's Thesis

TPK4920 - Project and Quality Management

Machine Learning on Complex Projects:

Multivariate time series data analysis through utilization of the sequential algorithm LSTM

Magnus Olai Aarvold

Wilhelm Jan Hartvig

Spring 2021: June 10, 2021

Supervised by:
Nils Olsson

NTNU | Norwegian University of Science and Technology

Preface

This thesis is written as a finalization of the five-year master program in *Mechanical Engineering* at the Norwegian University of Science and Technology (NTNU). The thesis is written as part of a specialization towards *Project- and Quality Management* at the Department of Mechanical and Industrial Engineering and Faculty of Engineering. It is written in the spring of 2021 and counts toward 30 credits.

We were first introduced to the topic by our supervisor, Nils Olsson, and through the Department of Mechanical and Industrial Engineering in the spring of 2020. In the fall of 2020, we wrote our specialization report on artificial intelligence, especially as machine learning, in the construction industry. The lessons learned from the specialization project were many. After applying machine learning to classify static data regarding success in construction projects, we wanted to inspect another aspect of machine learning, namely time series data. Opposed to stationary data point problems, times series problems must account for the sequence of the data points. Thus, it poses a new set of challenges to overcome. Nevertheless, these kinds of data points are of more relevance to the industry today. With ever-increasing magnitudes of data captured by companies, a way to utilize the data points to uncover the elusive future is alluring.

From the program at Mechanical Engineering, the writers have acquired knowledge in project management theory. Project management is a necessary profession but lags at the adoption of new technologies. We are grateful to contribute to the ever-evolving development of project management and how it can be further digitized. When the opportunity opened up to investigate the opportunities and implications of machine learning applied to project-related data, it had to be taken.

Acknowledgement

This thesis has relied on several vital supportive persons.

First, we would like to thank our supervisor, Nils Olsson. He introduced us to the pioneering theme of combining project management and machine learning. The supervising have been most satisfying by balancing the degree of independence and guidelines for progress throughout the semester. Nils has been a driver to both reach out to possible partners and maintain relations. Without that drive, this thesis could have resulted in no dataset to analyze. In times of indecisions, Nils has guided us and always replied quickly.

We want to thank the collaborating companies. As companies tend to hold rewarding information in-house, we are incredibly thankful that the companies provide both time and data to this thesis. Without them sharing the data, this thesis would be a literature research study. The companies showed much interest in the progress and results, making us more enthusiastic about doing a thorough analysis. Additionally, they set aside time for meetings to provide valuable domain knowledge of the information inherent in the data. A special thank goes to the expert representative from the project management software to schedule weekly meetings to guide us in the new and overwhelming software.

Next, we direct gratitude to our families and are grateful for their endless support and the encouragement they have shown during our studies. Both writers have ties to NTNU through earlier generations in the family, awakening our interest in new technology. A special greeting to Ingvild and Katrine for the guidance and support both on and off our studies.

Lastly, we would like to thank our friends at NTNU for making the studies an incredible and unforgettable time.

Abstract

The adoption of machine learning has exploded over the last decade, and the many success stories have shed light on its value. Many of these success stories stem from companies already in the digital economy. The energy industry has lagged in the adoption of new technology. For instance, oil and gas in the energy industry have had low break-even price and globally high demand over several decades. The effect has led to less attention to optimization and low adoption of new technology because the future cash flow will be high regardless of new implementations. An increasingly higher market share of renewable energy and more emphasis on development has evolved the industry over the last decades. The evolving industry has increased its digitization with new technologies such as the Internet of Things and smart systems. Consequently, data is now gathered from multiple sources, and companies investigate their readiness to support machine learning.

New technologies, rapid development, and a high degree of projectification have presented the project management team with a new set of challenges. Project management is often considered a traditional profession using old technology. However, the digitization of project management is increasing, and companies acknowledge the value of more advanced technology. Nevertheless, machine learning is still a term for the future, and there has not been conducted much research combining machine learning and project management. From the increased digitization, both in project management and in the energy industry, there could be considerable potential in using the gathered information. Forecasts and classification analysis powered by machine learning could prove to aid the project management team to handle the novel challenges of tomorrow. Therefore, project management in the industry is ripe to implement machine learning to enhance the efficiency and the success rate of projects.

This thesis will serve as a pilot for how machine learning can be of use for the project management team in the industry. It is written in collaboration with two companies, each providing datasets. Machine learning is implemented on two types of datasets, the first is from project follow-up with progress data, and the second is from a project management tool, in which both types are from physical projects. The thesis presents how to format the datasets to support machine learning for time series and classification analysis, and the accuracy models today can achieve. The progress data is aggregated weekly based on

the discipline of the work done, while the project management tool's data are structured as individual activities. This is done to maintain key information of each activity in the project. In addition to implications for project management, the thesis will present challenges for implementing machine learning and how the data could be structured by the companies to be more readily implemented.

The findings of this thesis prove that machine learning might extract more value from project-related data in the energy industry, and it gives directions for what can be done to increase the accuracy. The most promising results are the machine learning model's ability to learn by training on more projects. Training on two projects instead of one decreased the Root Mean Squared Error (RMSE) by 37%. Regarding the data from the project management tool, the model achieved low accuracy with an RMSE approximately equal to the average earned value for the project. However, the analysis of the progress data from follow-up achieved higher accuracy and is thus more readily for practice. First, these findings suggest which areas are most applicable for implementing machine learning at the time of writing. The data generated from the project management software is of higher resolution than the data logged by the operator, and could prove better in the long term. Secondly, the findings support the importance of structuring the incoming data to be complete and consistent.

Sammendrag

Bruken av maskinlæring har vokst kraftig i løpet av de siste tiårene, og de mange suksesshistoriene har kastet lys over dens iboende verdi. Flere av disse suksesshistoriene stammer fra selskaper som allerede er i den digitale økonomien. Energiindustrien har falt etter i bruken av ny teknologi. For eksempel har olje og gass i energibransjen gjennom flere tiår hatt en lav break-even og en høy etterspørsel globalt. Som følge av dette har mindre oppmerksomhet blitt rettet mot optimalisering og det har vært lite bruk av ny teknologi. Dette er fordi den fremtidige kontantstrømmen vil være høy uavhengig av eventuelt nye implementeringer. De siste årene har industrien blitt utviklet som følge av en stadig høyere markedsandel av fornybar energi og mer vektlegging på teknologi. Den stadig voksende industrien har ført til økt digitalisering som følge av nye teknologier som Internet of Things og andre smarte systemer. Følgelig blir data nå samlet inn fra flere kilder og selskaper undersøker deres beredskap til å støtte fremtidig maskinlæring. Ny teknologi, rask utvikling og høy grad av prosjektifisering har gitt prosjektledelsen et nytt sett med utfordringer som de nå står overfor. Prosjektledelse blir ofte sett på som et tradisjonelt yrke som bruker den samme teknologien som de alltid har gjort. Digitaliseringen av prosjektledelse øker, og selskapene anerkjenner verdien av mer avansert teknologi. Likevel er maskinlæring fremdeles et begrep fra fremtiden, og det har ikke vært mye forskning som kombinerer maskinlæring og prosjektledelse. Fra den økte digitaliseringen, både i prosjektledelse og i energibransjen, kan det være et stort potensial i det å bruke den innsamlede dataen til noe nytt. Prognoser og klassifiseringsanalyse drevet av maskinlæring kan vise seg nyttig for prosjektledelsen for å takle morgendagens utfordringer. Derfor er bransjen til prosjektledere moden for å implementere maskinlæring for å øke effektiviteten og suksessgraden for deres prosjekter.

Denne oppgaven vil fungere som en indikasjon for hvordan maskinlæring kan være til nytte for prosjektledelsesteamet i industrien. Maskinlæring i oppgaven er implementert på to typer datasett, det første er fra prosjektoppfølging med fremdriftsdata. Det andre er fra et prosjektledelsesverktøy. Begge typene er hentet fra faktiske prosjekter. Fremdriftsdataene aggregeres ukentlig basert på disiplinen til arbeidet, mens prosjektledelsesverktøyet strukturerer hver aktivitet for seg. Sistnevnte er for å opprettholde nøkkelinformasjon om hver aktivitet i prosjektet. I tillegg til prosjektledelsesaspektet, vil oppgaven presentere

hvordan man kan formatere dataene for å støtte maskinlæring, hvilke utfordringer det er i renseprosessen, og hvordan dataene kan struktureres av selskapene slik at det blir lettere for dem å implementere maskinlæring senere.

Funnene i denne studien viser at maskinlæring kan tilføre verdi til prosjektrelaterte data i energibransjen. Videre gis det beskrivelser for hva som kan gjøres for å øke nøyaktigheten til modellene. De mest lovende resultatene er maskinlæringsmodellens evne til å lære ved å trene på flere prosjekter etter hverandre. Trening på to prosjekter i stedet for kun ett reduserte erroren med 37%. Når det kommer til dataene fra prosjektledelsesverktøyet, oppnådde modellen lav nøyaktighet med en error som var omtrent lik den gjennomsnittlige opptjente verdien for prosjektet den uken. Imidlertid oppnådde analysen av fremdriftsdataene høyere nøyaktighet, og dermed regnes disse dataene som lettere å ta i bruk i praksis. For det første antyder disse funnene hvilke områder som er mest anvendelige for implementering av maskinlæring i skrivende stund. Dataene som frembringes fra prosjektledelsesprogramvaren har høyere oppløsning enn dataene som er logget av operatøren. For det andre støtter funnene viktigheten av det å strukturere innkommende data slik at de er komplette og konsistente.

Contents

Page

1 Introduction **1**

2 Theoretical Framework **4**

2.1 Project Management 4

2.1.1 Challenges for the project manager in the energy industry 5

2.1.2 Estimations 7

2.1.3 Project management methodologies 8

2.2 Planning and Executing the Project 10

2.2.1 Baselines 10

2.2.2 Work breakdown structure 12

2.2.3 Scheduling 13

2.2.4 Performance measures 16

2.3 Applying Machine Learning to Businesses 18

2.3.1 History of artificial intelligence 18

2.3.2 What is machine learning? 19

2.3.3 State-of-the-art in machine learning 20

2.3.4 Challenges with machine learning 21

2.3.5 Applications of machine learning in the energy industry 21

2.3.6 Datapoints 22

2.3.7 Classifications theory 23

2.3.8 Classification metrics 26

2.3.9 Scalers 27

2.3.10 Predictive theory 28

2.4 Machine Learning Models for Time Series Data 29

2.4.1 Error functions to optimize the model 30

2.4.2 Recurrent Neural Network 31

2.4.3 Long Short-Term Memory 32

2.4.4 ARIMA 33

2.4.5 Multivariate and univariate prediction models 34

3 Methodology **35**

3.1 Literature Search . . . 35
3.2 Applying Machine Learning on the Datasets . . . 36
3.3 Company A . . . 39
3.3.1 Exploratory data analysis . . . 39
3.3.2 Cleaning . . . 42
3.3.3 Preprocessing . . . 44
3.3.4 Train and fit the model . . . 46
3.3.5 Summary of the complete model . . . 49
3.4 Company B . . . 50
3.4.1 Exploratory data analysis . . . 50
3.4.2 Cleaning . . . 52
3.4.3 Preprocessing . . . 53
3.5 B1 - Time Series Analysis . . . 56
3.5.1 Cleaning . . . 56
3.5.2 Preprocessing . . . 57
3.5.3 Train and fit the model . . . 57
3.5.4 Summary of the complete model . . . 62
3.6 B2 - Classification Analysis . . . 63
3.6.1 Exploratory data analysis . . . 63
3.6.2 Preprocessing . . . 63
3.6.3 Train and fit the model . . . 64
3.6.4 Summary of the complete model . . . 66

4 Results **67**

4.1 Results from the Time Series Analysis on Company A . . . 67
4.1.1 Initial results . . . 67
4.1.2 The configuration of the hyperparameters . . . 68
4.1.3 Results with final configuration . . . 71
4.2 Results from the Time Series Analysis on Company B . . . 74
4.2.1 Initial results . . . 74
4.2.2 The configuration of the hyperparameters . . . 75
4.2.3 Time series with MCS . . . 79

4.3 Results from the Classification Analysis on Company B 84
4.3.1 Preliminary results 84
4.3.2 The configuration of the hyperparameters 86
4.3.3 The classification analysis 88

5 Discussion **91**
5.1 Company A - Time Series 91
5.1.1 Performance of the model 91
5.1.2 Specific handling of the data 93
5.1.3 Implications for the project management team 94
5.2 Company B - Time Series 97
5.2.1 Performance of the model 98
5.2.2 Specific handling of the data 99
5.2.3 More thorough EDA and preprocessing 101
5.2.4 Implications to the project management team 102
5.3 Company B - Classification 104
5.4 Further Utilization 110
5.5 Handling Different Datasets 111
5.6 Is the Data in the Companies Ready to Support Machine Learning? . . . 114
5.7 Correlation 115
5.8 Limitations 116
5.8.1 General limitations 116
5.8.2 Company A limitations 118
5.8.3 Company B limitations 119

6 Conclusion **120**

7 Further Research **124**

Appendices **141**

Appendix A Company A **141**
A.1 Augmentations 141
A.2 Figures 144

Appendix B Company B **150**

B.1 Tables 150

B.2 Figures 151

B.3 Code Listing 167

List of Figures

2.1 Simple project management development methodologies 9
2.2 WBS example with 6 levels of detail. 14
2.3 A project using activity on arc (top) and activity on node (bottom). . . . 15
2.4 A project using a Gantt-chart to visualize the duration of certain activities. 16
2.5 K-nearest-neighbor classifier. 25
2.6 The confusion matrix with notes on precision and recall. 27
2.7 Example of different scalers. 28
2.8 RNN on the left opposed to feedforward Neural Networks on the right. . . 31
2.9 Example of the ReLU activation function. 32
2.10 The core of hidden layers in the LSTM model. 33
3.1 Overview of a generic machine learning flowchart. 37
3.2 Expended value plotted against earned value for the same project. 41
3.3 The figure shows the work done in the same discipline between projects. . 42
3.4 Sliding window and forward chaining illustration. 47
3.5 The custom chaining algorithm for two chains and the number of augments set to 5. 49
3.6 Plot of the starting activities per week. 56
3.7 How the number of features decrease as a summation limit is set for the dataset. 57
3.8 The figure present how the windows slides through the data. 59
3.9 Train-test split with n_past of 3. 60
3.10 Illustration of the Monte Carlo simulation, mean, and binarized method. 66
4.1 The augmentations of the projects from Company A. 68
4.2 Plots from the first chain to compare the best normalization scaler to the standardization scaler. 69
4.3 Plot of the results from the first chain with StandardScaler. 69
4.4 The two plots show the different performance from training on one and two projects with the same configuration. 72
4.5 The two plots show the different performance from training on one and two projects with the same configuration. 73
4.6 Plot of the results from the second chain with StandardScaler. 74

4.7 When the activities starts. Before drop of empty weeks and the reset of week number. 75
4.8 Training and validation loss results. 76
4.9 Results from determining the scaler to include for further analysis. 79
4.10 Plot of the earned value with the forecast mean for the configuration with lowest RMSE score when the split is 0.73 and n_past of 3. 80
4.11 Plot of the earned value with the forecast mean for the configuration with lowest RMSE score when the split is 0.73 and n_past of 6. 81
4.12 Plot of the earned value with the forecast mean for the configuration with lowest RMSE score. 82
4.13 Plot of the earned value with the forecast mean for the configuration with lowest RMSE score when the split is 0.83 and n_past of 3. 84
4.14 Mean Confusion Matrices for three classifiers. 85
4.15 Confusion Matrices for three classifiers. 85
4.16 Plot of the SHAP values of Model 4. 90
5.1 A plot of the simulations with 100 MCS. 92
5.2 Results from determining the scaler to include for further analysis. 99
5.3 Plot of the train-test means. 101
5.4 Mean binarizer BEI and CM plot 106
5.5 Plot of the SHAP values of Model 1, features set "Description". 108
5.6 Plot of the SHAP values of Model 1, feature set "All except scope". . . . 109
5.7 Plot of the SHAP values of Model 1 and 3, feature set "Codes". 110
5.8 How a user later can use the trained model this thesis proposes. 111

List of Tables

3.1 The initial shapes of the datasets. 40
3.2 Examples of the Pandas functions front and back fill. 43
3.3 The values for the hyperparameters to be tested in the grid search. . . . 47
3.4 The final shapes of the datasets. 50
3.5 A summary of how complete certain dates are filled in the software. 51
3.6 The number of features in each feature set. 55
3.7 The values for the hyperparameters to be tested in the grid search. . . . 62
4.1 Table of the ten best performing configurations of hyperparameters on the second chain. 70
4.2 The table shows the best configuration of the hyperparameters. 71
4.3 Table of the ten best performing configurations of hyperparameters with a split of 0.73 and n_past of 3. 77
4.4 Table of the ten best performing configurations of hyperparameters with a split of 0.73 and n_past of 6. 78
4.5 Table of the ten best performing configurations of hyperparameters with a split of 0.83. 78
4.6 How many of each type of activity metric found by model. 84
4.7 The number of features in each feature set, and the splits dependent on the RF model. 87
4.8 Comparison between 100 MCS and two 5000 MCS tests. 88
4.9 The F1 scores for all feature sets for all models. 89
5.1 Number of times the Tsh feature was over or under the Tsh mean, compared to the times the activity was a hit or a miss. 107
5.2 Description keyword "cost" versus start hit classification. 108

List of Abbreviations

AC Actual Value

ADAM ADaptive Moment Estimation

AOA Activity On Arc

AON Activity On Node

ARIMA Auto Regressive, Intergrated and Moving Average

BEI Baseline Execution Index

CM Confusion Matrix

CPI Cost Performance Index

CPLI Critical Path Length Index

CPM Critical Path Method

DF DataFrame

DT Desicion Tree

EAC Expected Cost At Completion

EDA Exploratory Data Analysis

ED Expected Duration At Completion

EF Early Finish

ESA Early Start Analyzed

ES Early Start

EV Earned Value

FLDA Fisher's Linear Discriminant Analysis

GP Gaussian Process

GRU Gated Recurrent Unit

IoT Internet of Things

LF Late Finish

LSTM Long Short-Term Memory

LS Late Start

MAE Mean Absolute Error

MCS Monte Carlo Simulation

MLP Multi Layered Perceptron

ML Machine Learning

MSE Mean Square Error

NaN Not a Number

PERT Program Evaluation and Review Technique

PMI Project Management Institute

PV Planned Value

QTY Quantity

ReLU Rectifying Linear activation Unit

RF RandomForest

RMSE Root Mean Square Error

RNN Recurrent Neural Networks

SHAP SHapley Additive exPlanations

SPI Schedule Performance Index

SQL Structured Query Language

SSP Successive Schedule Planning

SVM Support Vector Machines

VOR Variation Order Request

VO Variation Order

XGB eXtreme Gradient Boost

1 Introduction

The purpose of this thesis is to investigate how machine learning can be applied to project-based datasets. The pioneering idea is that there is currently an immense potential in this data not yet taken advantage of. Three research questions have been formulated to break down the thesis into solvable tasks.

1. How can project plans and follow up data be formatted to support machine learning, and what is the achievable model accuracy?
2. What machine learning techniques are relevant for this type of data?
3. What are the project management implications of the results?

The first question is vital to investigate the companies' readiness for machine learning and which steps could be taken to make them more mature for the digital transformation and to apply machine learning in the future. The two following are closely related as the results are controlled by which machine learning techniques can be implemented.

The adoption of machine learning has exploded in several industries while the energy industry is lagging on the implementation of modern technology [1]. In offshore, the trend is towards larger investments and more complex projects at deeper water level [2; 3]. The energy industry is evolving with an increasing market share of renewable energy with modern technology, and new business strategies [4; 5]. Additionally, the industry is becoming more digitized with the implementation of the Internet of Things (IoT), advanced analytic techniques, and multi-agent systems [6; 7; 8]. Despite the implementation of new technology, the industry is experiencing cost overruns and longer project duration [9; 10]. There are also defined several challenges for project management in the industry [11; 3]. The need for more efficient processes and quality in project management is evident.

Similar to the energy industry, project management is a traditional profession often associated with infrequent use of new technology and causing overruns to appear. Over the recent years, applying artificial intelligence to the gathered data have become more frequent in several industries [12]. With increased digitization in the industry, the project managers got their hands on large quantities of digital information. However, the industry has not yet made it common to implement machine learning on project-related data. As a pilot, this thesis will apply machine learning on project management related data

regarding complex projects to discover its current applicability and the type of results that can be derived.

The presence of a professional project manager can be argued to be essential and necessary to be able to meet the project goals [13]. In short, the task of the project manager is to control the progress and activities of the project to meet the desired goals set by the project owner [14]. The knowledge area of the project manager is often associated with soft skills, meaning that the manager must reach out to several key stakeholders and be able to communicate well and efficiently. It is argued that the knowledge pool of the project manager is progressing to include non-engineering knowledge in addition to traditional areas such as responsibility for technical content and to stay within cost and time estimates [15; 16]. Nevertheless, the profession of project management is subject to traditional methods and best-practice methodologies. Arguably, it might be due to the inherent characteristics of projects, namely their need to be treated uniquely. Also, project management methodologies are argued to be dependent on national environments, in which certain countries have more developed methodologies [17]. Compared to other professions, project management is lagging behind others on the adoption of new technology. The digitization of project management has been focused on the documents and project management tools to track the scheduled and current performance. Recently, advanced programs that include various artificial intelligence principles have been developed to aid the project management team [18; 19].

There is agreement that the energy industry is becoming more complex because of globalization, and there are challenges to deliver projects in time and within the planned budget [2; 20]. The oil and gas sector has historically had low break-even prices and globally high demand. However, the recent years have proved to be challenging for the industry as the oil price fluctuates and there is more emphasis on renewable energy. Additionally, the renewable energy industry has had exponential growth in the last century, taking a more significant part of the total energy market [4]. There is a need for successful project management to handle the increasing number of projects and execute them efficiently. To maintain a competitive position in the market, actors are increasing outsourcing and globalization [5]. Instead of maintaining relations in a small region, the project manager must now plan and handle multiple bilateral incumbents and agreements. The industry

must adjust its business strategy to cut costs and optimize efficiency [2]. Implementation of modern technology could be a method to overcome these challenges. Some of the most popular applications have been the development of multi-agent systems to handle the complexity, deployment of digital twins and to create smart programs that monitor and control the assets [6; 7].

The energy industry has discovered, like several other industries, the potential there is in introducing smart technology and expert systems [21]. Computers empowered by machine learning are more fit to do calculations and recognize patterns and trends in the data than what humans are [22; 23]. The algorithms will perform increasingly better when there are large amounts of correct data to train with. Correct data implies that the most interesting variables are logged and that the data is consistent such that there is a low amount of missing data. Large parts of the energy industry are in a luxurious position as they have gathered data from multiple sources and have gathered over time summing up large amounts of data [7]. However, a downside of machine learning is that it is sometimes associated with low return of investment and difficulty in developing customized models fit to tackle specific tasks [24; 25].

This thesis is structured as a report including theoretical framework, methodology, results, discussion, and a conclusion. The theoretical framework is split into four parts. The first presents characteristics and challenges in project management. The second one presents tools for the project manager during the project, like scheduling and performance measures. The third is about machine learning, what it is, its current use in the industry, and why it is relevant now, and the last part presents machine learning models appropriate for time series data. The methodology elaborates on how machine learning is applied to the datasets from the collaborating companies and will constitute several parts. The results are presented and discussed before a conclusion is drawn. Ultimately, further work will be presented.

2 Theoretical Framework

The theory consists of four main parts: project management, planning methodology, machine learning in businesses, and machine learning models for time series data. The section on project management will present specific challenges and tasks faced by the project management team during projects. It will also present tools provided to the project manager and how digitization might help. Planning methodology, which is an essential part of good project management practice, shows how the project manager might utilize tools to systematize and make estimations of cost and time. Later on, the part on applying machine learning to business contains machine learning's history, what it is, challenges, applications, and essential characteristics of types of input data telling which type of model is appropriate. Lastly, the final part presents various models for multivariate time series data and how these models will be compared using error functions.

2.1 Project Management

The project management team faces several decision points during a project, and several of these are at an early phase. At an early phase, some of the decisions the team must decide upon are how the project will be managed, which models to use, which digital tools to help communication and visualization, how communication will flow, and how present the manager should be [26; 11; 27]. However, there is a need for good project management in almost all businesses, making it a well-known profession with several best practices. In addition, increasing projectification of economies makes project management an increasingly important profession [28]. To be of aid to project managers, the Project Management Institute (PMI) has written the PMBOK, which includes definitions of knowledge areas and steps for how the project managers should proceed [26]. As the world is constantly changing, so are the expectations of the project manager [15]. The PMBOK explains ten traditional knowledge areas for the project manager [26]. Arguably, the knowledge pool is expanded to include several other aspects [15; 16]. The new knowledge requirements for the project manager include non-engineering knowledge to meet the desired level of professionality. It might be incorporating the voices of the stakeholders, resolving trade-offs, meeting sustainability goals, or working concurrently on all aspects

of the multi-functional teams [16; 3]. The new knowledge is added to the traditional knowledge areas concerning responsibility for technical content and to stay within cost and time estimates [27]. Findings from Mir and Pinnington state that project management has a positive influence on project success, supporting the idea that good project management yields higher success rates [13]. Although the presence of a project manager is essential, PMI conducted an in-depth study involving 65 case study organizations confirming the value of a project manager but emphasize the importance of culture and implementation that fits with the organization's needs [11]. Other studies support this [16; 13]. Several authors suggest that the current project management education is not adequate to prepare managers to deal with the complex relations of the real world, and thus, they are not ready to manage projects [29]. Also, that there still are high failure rates despite considerable efforts put in the education [29].

2.1.1 Challenges for the project manager in the energy industry

The project management in the energy industry has expanded from traditional energy sources like petroleum, coal and hydro to include more renewable sources like solar, thermal, and wind [4]. Within these, there is a drift towards implementation of industrial engineering, which is the design and development of integrated systems of people, machines, and information resources for producing products and services, to meet the increasing globalization [30]. Because projects in oil and gas are among the ones that generate an environmental sentiment, Badiru and Osisanya argue that formal project management should be a part of planning, organizing, scheduling, and controlling projects in the industry [31]. The negative characteristics for project management in the energy industry include lack of sense of shared responsibility, inadequate advanced planning, insufficient attention to standards during procurement processes, limited public engagement and reporting, and weak collaboration between firms and subcontractors [3]. Research has shown that some of these characteristics might be overcome by collaborative technology, but it also stated that substituting one actor with technology is not frictionless [8].

The PMI argue that the project manager should influence the company's operations and processes towards more sustainable solutions because they are managing how the changes and new solutions are implemented [3]. To reduce the emissions, it is proposed a method to recycle and reuse the oilfield wastewater in irrigation, livestock or wildlife, and various

industrial uses [32]. However, China's dependence on oil imports will increase and is expected to reach 64-66 percent of the oil used by 2030 [33]. Consequently, actors in the industry are still expanding their exploration and engaging production in oil fields beyond their national borders. The increased globalization poses additional challenges to manage the projects as it increases the complexity and the need to develop new management styles to balance multiple objectives [2]. Globalization is not only for locations of oil wells and wind farms but also for the production of the rigs and windmills and their associated equipment [31]. The renewable energy industry has increased its outsourcing and globalization to be competitive in the energy market [5]. For the project management team, this means additional decisions points for which supplier to choose and how to manage the communication.

The project manager must now manage, organize, and coordinate production and delivery across multiple actors in several countries. Carvalho et al. argue that national environments play a crucial role for project performance where some countries have more developed project management methodologies [17]. Consequently, the complexity in project management looks to be increasing. Typically, the assemblies are outsourced to yards in lower-labour cost countries [31]. Additionally, the products are becoming more prominent, and the average project duration is three to six years [9]. However, there has been an increase in average water depth for offshore projects from 200 meters to 400 meters, resulting in larger investments. It is argued that these projects have not performed well, where more than 60 percent have experienced cost overruns of more than 33 percent of the original target [10]. Ahiaga-Dagbui et al. state that the industry struggles to find a cost-effective procedure to decommission offshore assets as there is a tendency for larger companies to undertake decommissioning projects which can be constrained by the administrative department and corporate policies [34].

The transfer of tacit knowledge is argued to be necessary for innovation. However, the industry finds it difficult to exchange this knowledge because of barriers related to personnel, team, organizational, and external [35]. The transfer of information is vital in many aspects. As many as over 90 percent of project managers say that transferring of information is a crucial ingredient to implement sustainability in project management [3]. To better transfer knowledge, Chen and Pang use fuzzy networks to guide the project

management team on how knowledge should be distributed in the project group [5].

The digital transformation has begun to take shape in the energy industry. As top management visualize the value in digitization, the project manager should implement new technologies in the projects [36; 3]. This might create an initial barrier for the project manager as they must acquire new competencies. However, the digital transformation is proposed to drive productivity and achieve higher quality [36]. A study by Kolloch and Dellermann shows that the project manager still faces traditional challenges when introducing new technologies [8]. The study presents cases in which the supplier struggles to deliver modern technology, so the project manager must quickly find new suppliers. Actors in solar energy have developed new strategies, acted upon by the project manager, to exchange and create new knowledge to increase their competitiveness against other energy sources [5]. An emerging technology in the industry is the deployment of digital twins, and the two most popular application areas are asset monitoring and maintenance, and project planning and life cycle management [37]. The high reward in improvements for maintenance and monitoring will be to reduce downtime, which is crucial in an industry producing high-value streams over a short time.

2.1.2 Estimations

Reasonable estimates and accurate progress control is vital for the project manager. Because projects by definition are unique, there will be inherent uncertainty that must be managed [38]. Leading businesses today spend much money on tools to better their estimations, from having a monthly prediction to one that may differ from day to day [39]. The world is a complex and interwoven timeline with contingent and confusing factors, making time-series predictions hard [40]. Extensive research by UC Berkeley professor Philip E. Tetlock was done regarding the expert predictions [41]. He asked almost 300 experts to make predictions of the future. This accumulated to over 82 000 predictions. The conclusion was that they performed only a little better than a dart-throwing chimpanzee. However, one interesting metric came to light during the experiment. That experts of a broad topic, or as it is called - a jack of all trades, yielded the highest scores. In comparison, the scientists that had excellent knowledge of mostly one specific topic performed the worst. A computer is more fit to handle large volumes of data than a human is and could provide more accurate predictions [42]. As Weinberger

puts forward in his book, machine learning will broaden the knowledge gap to take us to the next level of wisdom [43]. Artificial intelligence may prove to understand correlated events on a broader scale better than humans can today. Also, as Kahneman writes in his book, humans are not evolved to think in lengthy and complex terms [44]. Thus, computers may be the remedy to perform accurate estimations regarding the future.

In the process of estimating in a project, there are three usually utilized methods [45]. The first one is called analog estimations and is quick to implement, and thus, much utilized. This method often uses approximations or comparisons to make estimations. Therefore, the level of accuracy will directly depend on the guesses or knowledge pool of the person making the comparisons. New technology can aid in this endeavour. The second is called the order of magnitude. This method takes key outlines of the projects as input and base an estimation upon this. It is not regarded as a robust estimation and is commonly used in the early phases of a project. The third method, which is the most accurate, is called the definitive method. It takes a high number of inputs to give a detailed estimation of the project. Because estimations are intricate in nature, project managers often compensate with reserves to act as a buffer [46]. This buffer is often 10% - 15% of the estimate. One known statistical method to calculate an estimation and buffer is by utilizing the 3-point estimate [47].

2.1.3 Project management methodologies

At the initial part of the project, it must be decided which project management methodology to utilize for the rest of the project. This may be subject to the project management team or corporate guidelines. The most appropriate methodology depends on several factors, and choosing the correct one might simplify much during execution. The model is meant to standardize how execution is done, ensure that experiences are collected, and achieve predictability during the project [48]. However, there are many methodologies to choose among and can be sorted as overview models, models based on phases and stages, and process models. Overview models seek to communicate the complete or parts of the discipline of project management. Popular methodologies are IPMA, Prince2, and PMI [48]. The methodology can also be based on sequential separation of the phases. Figure 2.1 shows the bare bone of a simple development cycle in project management, starting at the project definition process. This is then broken into the technical-, functional and

management-, and financial baseline. The number and definition of phases are often customized to fit the company or type of projects [38]. Process models are developed to highlight specific processes in the project, such as estimation, uncertainty, or work breakdown structure [48]. These type of models seeks to define how a process or activity is connected to other activities. Other metrics such as key performance indicators (KPIs) are put forward. Then, the monitor and control phase begins. As shown in Figure 2.1, it is clear that this is an iterative process, and very seldom, only one iteration is needed.

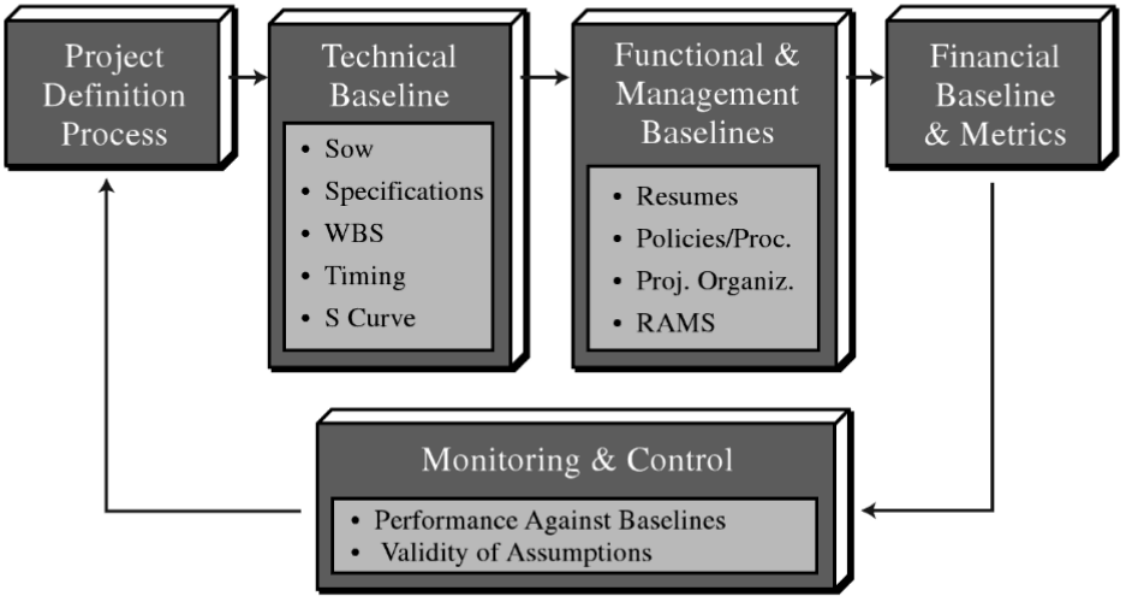


Figure 2.1: Simple project management development methodologies [38].

Actors in the energy industry will continue to engage in major capital projects that will push performance to higher levels and evolve their project management. The projects are often referred to as mega as they often exceed 2 billion dollars for a five-year project [49]. To evolve project management, more actors have adopted the stage-gate management style. After studying performance between projects and companies, Walkup and Ligon argue that the introduction of stage-gate has not yet helped the industry and that project failure still occur after its implementation [49]. However, evidence provided by Project Management Solutions suggests that implementing a project management office decreases failed projects, delivering projects ahead of schedule, under budget, and improve productivity [20]. A project management office, as defined in the PMBOK, is a management structure that standardizes the project-related governance processes and facilitates the sharing of resources, methodologies, tools, and techniques [50].

2.2 Planning and Executing the Project

This subsection includes how the team uses several tools to break down the project into understandable tasks and track progression and performance. Project plans are broken down to understand better which tasks that need to be done and allocate resources. Additionally, it might be easier for the project management team to control the project well by tracking current performance.

2.2.1 Baselines

A usual way to manage projects today is through the usage of baselines [51]. It is an established and well-known practice in the field of project management. Tereso et al. identified which project management practices are used in most private organizations. The results show that baseline plans are among the most used practises, and it is argued that activity list and baseline plan are essential for project success [52]. As the name gives away, a baseline is a line with which the future measurements are being compared. One could look at it as the preliminary base truth. The baseline is set at the beginning of a project. It must be accepted and approved as a temporal project plan. The project manager utilizes the baseline to evaluate current progress to the planned baseline, so a prerequisite for this is to have a good initial plan [53]. If this monitorization shows that the project is lagging, actions are required. Thus, a solid baseline is paramount in project management. One methodology to make a solid baseline is through the Project Management Maturity Model described by Kerzner [54]. Also, a baseline must be stored and be of easy access throughout the project. If it is not stored correctly, it will be useless and a waste of time. It is usual to construct and store multiple baselines. This comes from the inherent nature of a project. A project is synonymous with change, and since changes or other unforeseen events will appear in the project lifeline, changes must be done with the baseline. Usually, the number of baseline versions will be a function of the project size, complexity and time horizon [55]. Larger projects are more complex [56]. Further, if the project plan changes, the baseline also can change.

Baselines are most often created using a project management software. Thus, the project management team will sketch out a plan based on knowledge, and prior experiences in terms of the expected duration of different tasks or when it must be completed if the

deadline is most the dominant criteria [57]. Afterwards, the plan must be approved. Additionally, the order of tasks and required resources should be adjusted and optimized. Thus, the management has a baseline to compare and compute variances against during the execution. A project management tool will often be able to do this for you. Even though there is a tool to do the calculations, it is crucial to keep in mind that the quantification of the tasks originates in a human mind. Therefore, the plans are subject to uncertainty stemming from the project manager's knowledge [52]. The search for the most stable baseline and how to achieve it can be found in the literature [58]. Various approaches to find the best pre-baseline are defined to minimize the risk of disruptions and baseline changes.

The act of creating multiple baselines can be valuable because of the information it stores. It will be easier to track and pinpoint necessary changes during the execution. Also, this may be used to remember poor project management, team member accolades, or subcontractors deliveries. Further, bad decisions can be avoided through the active use of baselines. However, for continuity and predictability, the number of changes should be kept at a minimum. In an ideal world, there should not be necessary with more than one version of a baseline. So as a guideline, the baseline should only be revisited if necessary. Examples of this may be significant changes to the cost or scope of new requirements. It also may be a result of inadequate prior planning that resulted in poor performance and ripple effects. In the literature, the ripple effect is described as a phenomenon where even minor discrepancies, also from external forces, leads to delays and affect the progress negatively [55].

Baselines come in many forms, depending on the type of input data or the wanted comparisons by the management. One baseline may contain different kinds of baselines to more easily illustrate the different aspects of the project [38]. These kinds of baselines may be:

- Cost or budget baselines where the cost performance is the main driver
- Scope baseline covers the deliverable of the project. This may be both functional, technical and physical requirements
- The schedule baseline monitors the time aspects of the project. Here will also the

supporting elements be accounted for

- The quality or risk baseline will show the uncertainties and changes that may alter the performance of the project

Tracing of baselines is the main driver for effectiveness. To compare the actual progress with the planned estimates of the original baseline will give valuable insights into the project's overall performance. Using baseline as a reference yields two main advantages to the users. Firstly, the managers can easily see the actual progress versus the estimated one. This will show the overall performance and trend of the project. Deviations from the baseline should spark warnings and actions. Secondly, the usage of baselines makes it easier to assess the project manager in terms of experience, knowledge and quality [57]. If there is little need to revisit the baseline, the project manager is more likely to have been well prepared and experienced. Thus, baselines can be viewed as a learning tool for project managers [57]. Disruptions affect the project's productivity, and it follows that projects with several disruptions have the lowest productivity. As productivity stagnates, the baseline will also be affected. It is only in extreme cases in which the baseline is unaffected by disruptions. It is also argued that baseline productivity is a function of design complexity [59].

2.2.2 Work breakdown structure

The PMBOK defines the work breakdown structure (WBS) as a way to group the elements of a project by the deliverables so that the organizing and definitions of the scope of the project are more easily understood [26]. Another way to look at it is a decomposition of the total scope of work to be carried out to meet the targets of the project [48]. WBS is a one-dimensional breakdown which means it only breaks down the project's scope, and there is no information on the integration between each task. The detail for requirements in finishing the task increases for each subsequent level in the breakdown structure. This can also be seen in Figure 2.2. Therefore, the lowest level is essential because it will be the most detailed and the one used for reporting progress and costs [48]. The WBS can be based on different principles for how the breakdown is done. Some of the main principles are to break down on physical components, functional components, geographical, concerning business processes, or with respect to departments at the company [60]. The

project itself is the top chain in this structure. The project is broken down into work packages or elements evaluated in terms of difficulty and resource needs. Thus, the WBS will uncover bottlenecks and packages of high risk. The advantage of the WBS is that it is clear and thus yields a precise definition of the tasks and the responsible person to perform the task. By utilizing a WBS, the process is more lucid, and it makes it easier to understand how the project is set up and which activities are necessary to reach the end goal [48]. Another positive trait of the WBS is that it aids as a way to achieve a common understanding by all the persons of interest to the project. This remedy is not to be underestimated [38].

However, there are some pitfalls when using WBS. One is that it requires a solid plan and excellent knowledge of the project before the start. This may become easier by utilizing domain experts and allocating more time to the planning phase. Another downside is tunnel vision by the workers. If the workers become too invested in their small work package, they may lose the executive view of the project. If tunnel vision is the case, valuable and clever solutions may be lost [60]. Also, it is often less motivational to work in a small sub-package of a significant project if one does not know or remember that it is a valuable piece to complete the overall project. Further, if the WBS is too detailed or too small, it may yield unwanted results. If the WBS is too fine, it may become challenging to supervise. If it is too coarse, the workers may become uncertain and start to guess how the work should be performed. Thus, it may be beneficial to include the workers in the planning [61]. At the same time, this will give them a feeling of ownership of the project, which can yield high-quality results. However, the inclusion of many workers requires much coordination since there are now many opinions and suggestions to consider and compare. Hence, final prioritization of the suggestions may be difficult.

Hence, WBS is used to decompose a complex task into a more manageable work package by component, i.e. software, hardware, labor, and delivery.

2.2.3 Scheduling

Scheduling often begins by analyzing the work packages in the WBS. This results in a schedule baseline as mentioned in Section 2.2.1. Through network analysis, dependencies are highlighted, and tangent decisions are structured [38]. Many factors are considered,

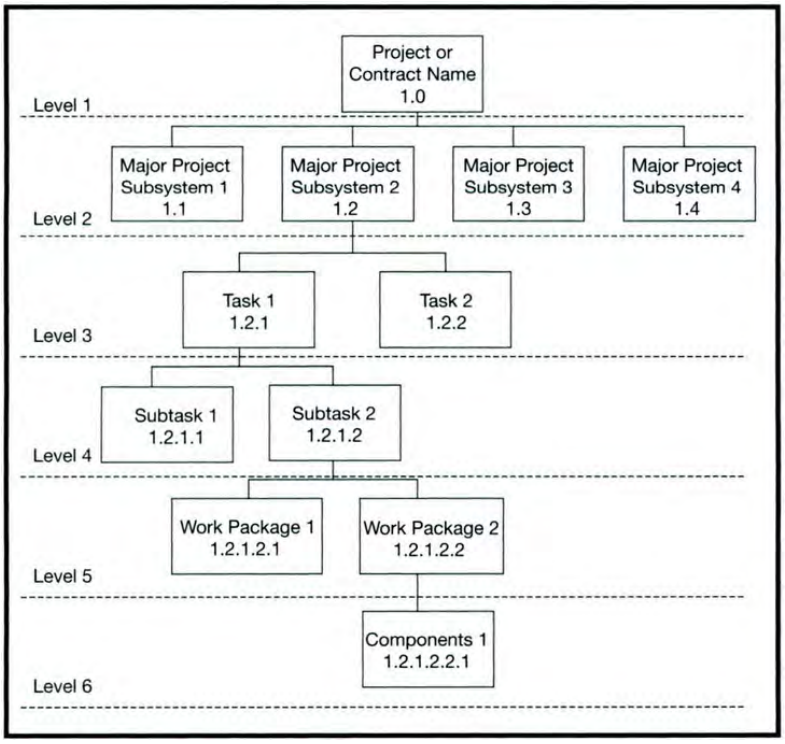


Figure 2.2: WBS example with 6 levels of detail [38].

and an estimate of the time necessary to complete the task is put forward. The human factor here is substantial. The available skill level and prior knowledge of the work packages are factors that could make or break the project. However, these skills may also come at a premium.

It is said that a good schedule requires both an analytic and artistic touch [38]. It has to be logical and fulfill all the necessary dependencies, yet have a certain finesse to overcome political and other interpersonal obstacles. Further, the importance of a good schedule is not to be underestimated. Experts state that a good schedule is one of the tasks with the highest impact on the whole project [62]. A skilled project manager will start with making a sound schedule, then estimate cost and other unique resources [38]. Other obstacles are constraints and assumptions in scheduling. This could be from deadlines or specifications and if the needed information is not available or correct. These hinders have to be thoroughly documented and validated. Flow network models are often used as an aid to the project manager to analyze the duration and how the activities should be linked [63]. Activity on node (AON) networks uses nodes to represent the project activities, and the arcs are used to tell the interdependence between nodes. In contrast, activity on arc (AOA) have the activity description on the arc, and the head and tail represent the start

and finish [64; 65]. This is also visualized in Figure 2.3.

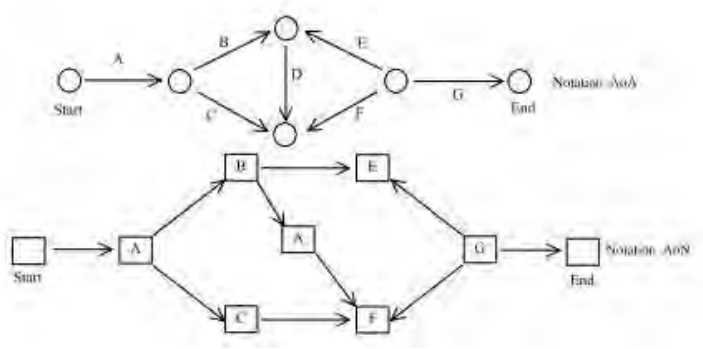


Figure 2.3: A project using activity on arc (top) and activity on node (bottom) [63].

Risk in network analysis such as in the WBS is often challenging to do correctly. Risk should be included in the early stages of all projects. WBS is a great tool to start this task with. Here, all tasks are described, and necessary resources are emerging [38]. During the network analysis, contingent dependencies are located and analyzed. In Figure 2.3 both task "E" and "F" are dependant on the finalization of other tasks from two separate paths. When one of these converging paths are behind schedule and thus delays the contingent dependencies, it is called schedule slippage [38]. The immediate downside to this is that all subsequent tasks now also will be delayed. Further, in Figure 2.3, the critical path is the path through the system, from the start node to the end node, which yields the lowest overall flow or time necessary.

Apart from networks like AON and AOA, Gantt-diagrams are often used in planning and scheduling. As shown in Figure 2.4, the activities in Gantt-diagrams are sorted along the vertical axis, and the horizontal represents time [66]. This layout makes Gantt charts very clear and easy to understand. On the other side, the charts do not tell how resources are distributed over time nor dependencies between activities [48].

The network consists of multiple activities, in which each activity represents a task with an output. The number and complexity of the activities are normally aligned with project duration. The activities can be linked in four ways: start-to-start, start-to-finish, finish-to-start, and finish-to-finish [65]. In addition, the connection between activities might have an inherent delay, meaning that activity "B" should start two months after "A" is finished. When the duration of each activity is deterministic, the Critical Path Method (CPM) can be applied to discover the most critical path for the project to finish on time [48].

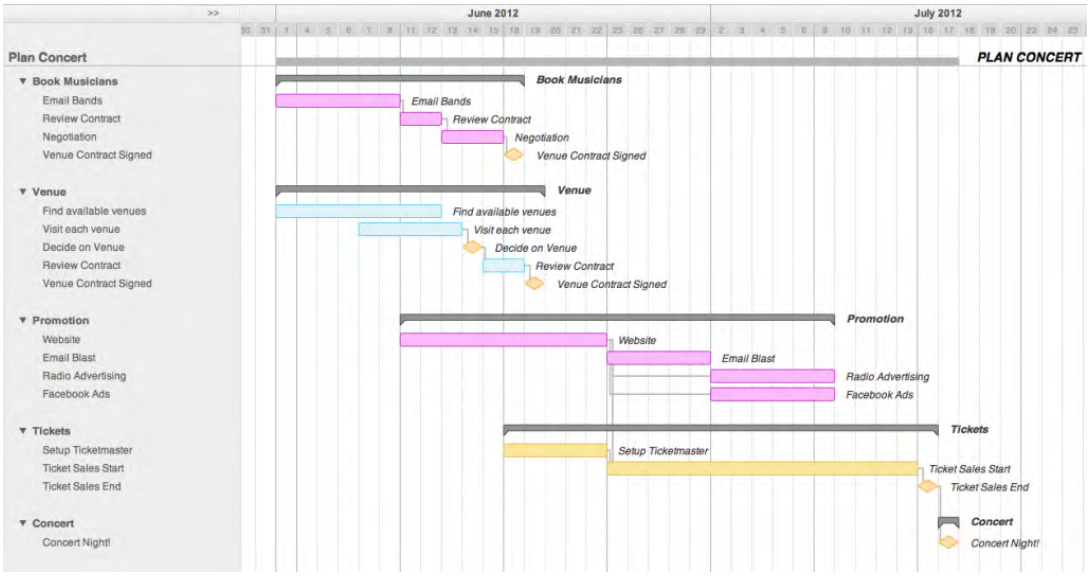


Figure 2.4: A project using a Gantt-chart to visualize the duration of certain activities [66].

If the activities are stochastic, probability theory is applied to the duration of each event to calculate the total project duration. A common approach is to assign the activities with L-, M-, and H-values representing lowest possible value, average value, and highest possible value, respectively [64]. Program Evaluation and Review Technique (PERT) and Successive Schedule Planning (SSP) may be used to estimate the total project duration. PERT most commonly utilizes β-distribution. In both PERT and SSP, the mean and variance will be calculated for each activity duration which later will be used to estimate total duration [48]. However, it is argued that the mentioned static methods are inadequate to account for today's dynamic projects properly [67].

2.2.4 Performance measures

Planned value (PV), earned value (EV), and actual cost (AC) are values commonly used to measure the performance of a project. Firstly, PV is based on values from the work packages in the WBS. Secondly, AC represents the actual costs that have occurred during the project. Lastly, EV is a measure for what is actually earned along the project [48]. It requires additional knowledge about the progression so that the value is equal to PV times the percentage of completion [45]. The three can be used to track the current performance of the project simply by using the cumulative values during execution [48]. The cumulative values form S-curves of varying shape, in which the shape of the curve is important to tell how the progression is [68]. For instance, cumulative earned value or actual cost is often

compared with the cost estimation baseline or planned value. In addition, they are the basis for two indexes used to indicate performance concerning time and cost. One is called cost performance index (CPI) and states how much work is completed to the actual cost. The equation to calculate it is $CPI = EV/AC$ [69]. Another, which is quite similar, is the schedule performance index (SPI), which measures actual progress to planned progress. It is derived as $SPI = EV/PV$ [69]. These indexes are without denomination, and a value equal to 1 means the project is on target. A value below 1 indicates a worse-than-planned performance, whereas a value above 1 indicates a better-than-planned performance.

From these indexes, one can further derive measures that are interesting to the project. These are the expected cost at completion (EAC) and the expected duration at completion (ED), see Equation 2.1 and 2.2 respectively. By inspection, it is clear that these measures deal with forecasting. They are calculated as follows.

$$EAC = \frac{PV}{CPI} = \frac{PV \cdot AC}{EV} \tag{2.1}$$

$$ED = \frac{D}{SPI} = \frac{D \cdot PV}{EV} \tag{2.2}$$

D in Equation 2.2 is an abbreviation of the planned duration.

The Baseline Execution Index (BEI) is another performance measure frequently used as an early warning metric. The index will suggest to the project manager if they are ahead or behind schedule. As equation 2.3 shows, the cumulative BEI shows the relation between the number of activities that should have started against the activities that have started [70]. The index will be calculated throughout a project to give a continuous measurement, and thus, it is cumulative. Therefore, the index is called the cumulative BEI or BEI for short.

$$BEI_{start} = \frac{Total\ \#\ of\ Tasks\ Started}{Total\ \#\ of\ Tasks\ that\ Should\ have\ Started} \tag{2.3}$$

Equation 2.4 shows that the BEI can also be used as an index to show the number of activities that missed their finish date.

$$BEI_{finish} = \frac{Total\ \#\ of\ Tasks\ Finished}{Total\ \#\ of\ Tasks\ that\ Should\ have\ Finished} \tag{2.4}$$

2.3 Applying Machine Learning to Businesses

Artificial intelligence, often as machine learning, has found its way into several businesses and received much attention during the last years. The increase in success stories and their alluring high rate of return makes it an interesting field to implement, but there are still numerous unsuccessful projects trying to implement artificial intelligence. This subsection describes what machine learning is with its most essential characteristics and which applications have been introduced to the industry.

2.3.1 History of artificial intelligence

The term artificial intelligence was first coined in 1956 by John McCarthy [71]. However, already in 1950, Alan Turing created the "Turing Test" to check whether a computer can fool a human by believing it is also human. Thus, the computer has gained real intelligence [72]. Moreover, in 1952, Arthur Samuel wrote a program in which the IBM computer learned by playing the game and improved its performance [73]. Up until the late nineties, there is a slow pace in innovations and new algorithms. The reason for the slow pace in development might be from Minsky and Papert's argument that current technology, as of 1969, did not have adequate processing power [1]. In 1997, IBM's Deep Blue beat Kasparov in a chess match. In 2016, Google's AlphaGo won a game of the Chinese board game Go against the world champion. An essential difference between Deep Blue and AlphaGo is that AlphaGo is substantially more effective as it analyzes more scenarios but only generates the most likely ones. Thus, it requires less memory at the same time as it searches more space [74]. The methods implemented in Deep Blue and AlphaGo is machine learning because they analyze large amounts of data to learn how to win. Machine learning is the branch of artificial intelligence that focuses on learning from a database [75]. By viewing the announcements of reaching new milestones in computer science, there is strong evidence that computer's ability to see, understand, and interact with the world is growing at a remarkable rate [73].

2.3.2 What is machine learning?

Machine learning is the science of algorithms that make machines act without being explicitly programmed and to autonomously improve themselves [76]. In the past decade, machine learning has been utilized to provide self-driving cars, speech and facial recognition, targeted advertising, and automation of repetitive tasks. The machine learning algorithms are written, so the model is trained to find patterns and trends in large amounts of data. The features found by the algorithms can afterwards be used as an aid in decision making. As more products are digitized and connected to the Internet, there are enormous amounts of data the programs can learn from. With the introduction of the Internet of Things, there will be generated more data readily usable by machine learning [77]. However, the learning phase requires that the right set of data be applied. The programmer must have adequate knowledge of which machine learning models to apply to the used set of data and that the set of data have inherent information to solve the problem.

Machine learning can be categorized into four sub-groups: supervised learning, unsupervised learning, reinforcement learning, and deep learning [78]. Supervised learning is preferred when the programmer has labeled data and a specific understanding of how the data is classified. It is intended to find patterns in the data, like distinguishing between types of fish. Unsupervised learning is preferred when the problem requires a massive amount of unlabeled data. The program will analyze the data without human intervention and seeks to understand the meaning behind the data. A popular field for unsupervised learning is for media applications, or the detection of spam mail [79]. Reinforcement learning is a behavioral learning model. The user will be guided to the best outcome by receiving feedback from the data analysis. Reinforcement differs from the two above as it does not train on the sample data set. Deep learning incorporates neural networks and is especially useful when trying to learn patterns from unstructured data. Deep learning is often used in image recognition, speech, and computer vision applications [78]. Common for all groups is that the algorithms perform ranking and keeps the best, often with some mutations, which is related to the survival of the fittest theory [80; 81].

2.3.3 State-of-the-art in machine learning

During the last decade, there has been a lot of research on machine learning, and the technology has been implemented through several businesses. Even though machine learning has come a long way since the beginning, there is still much unused potential. Machine learning technologies are made to solve specific tasks. Therefore, there are recent upgrades and newly created algorithms in several fields. Among the fields proposed to flourish in the future are computer vision, natural language processing, recommendation systems, and speech recognition [82]. The mentioned fields are somewhat more general and can be applied to several businesses. However, others argue that automated machine learning (AutoML) and real-time decision making will contribute most to drive business value [19]. The goal for AutoML is to automate the steps the data scientist will undertake to adjust the model and preprocess the dataset to achieve meaningful output [83]. AutoML is a general algorithm that does much of the work for the data scientist [84]. However, it comes with a hefty computation cost. Nevertheless, this method, depending on how much time and energy one gives it, may turn up with good results [85]. AutoML will perform the prepossessing, scaling, weighting, model selection and hyperparameter tuning by itself. Traditionally, the data scientist must decide which model to use and how to tune the model. Elshawi et al. have proposed several implementation for AutoML [83]. Mjolsnes and DeCoste argue that models learning from nonvector data and feature selection will be important areas in future research [86]. It is argued that the future will be more about intelligence and less about artificial. This means that the models will be more general and have more efficient reasoning, requiring less data to derive statements. Development in this directions will enable artificial intelligence to be more broadly applied [18].

Another field of research that has gained more attention is explainable techniques. For a long time, machine learning and artificial intelligence have been treated as a black box where one gives specific input and receive output without knowing how the model weight the different features [18]. Gill et al. researched responsible machine learning techniques. They concluded that there are new methods that take a solid step toward interpretability, explanation, and minimal discrimination for machine learning decisions which can lead to more fairness [87]. One of the proposed methods to open the black box in machine learning is the Model Explainability, which is approaches to enhance and increase the use

of visual results. Another one is the Model Performance Diagnosis, which is a method to link performance metrics to input-output instances [88].

2.3.4 Challenges with machine learning

Even though machine learning is increasingly adopted in several industries, several challenges must be overcome to obtain a successful model. One of the first challenges when starting a project is to transform the input data into features expected by the model [24]. Another problem argued in empirical research is the split in the model for training, validation, and testing, which is most often partitioned as 80 percent, 10 percent, and 10 percent. This is, in many cases, a good partition, but as machine learning models must be made unique to each project, the partition must also suit the quality of the input data [24]. Wuest et al. recognized specific challenges in a manufacturing perspective when applying machine learning techniques [89]. One problem is that the data may have a high degree of irrelevant and redundant information leading to low performance. The other is that the programmer needs good knowledge about the dataset to preprocess it properly. Another research, which gathered information from programming events with over 600 participants, confirm the fact that the best-performing machine learning programs only are appropriate to the problems they seek to solve [25]. Hajizadeh argues that the energy industry still uses waterfall methods for implementing new developments in machine learning. Thus, they suffer from slow adoption and struggle to meet expectations [7]. Further, new algorithms relating to time series data are in the works [90]. Time series data is data where the sequencing is of importance. The critical challenge is to both deliver good predictions on the user's short and long term goals. Other main challenges are typical entries to the syllabus of machine learning courses [12]. These focus on the challenges related to structuring the data, cleaning the data, and the necessary domain knowledge to implement the model. Some of these challenges will be addressed in this thesis.

2.3.5 Applications of machine learning in the energy industry

Machine learning has been slower to establish itself in the energy industry [7]. However, there is much potential to realize. Machine learning methods have the potential to transform discovery processes in oil and gas as they can quickly and accurately detect

signals and noise in seismic data [91]. Numerous industry publications state that machine learning is used in exploration, drilling, reservoir engineering, production operations, and virtual twins, and there is high potential to apply in hardware acceleration, continuous integration, and to the Internet of Things [7; 92]. One application in oil and gas is to cluster the wells based on characteristic data generated through machine learning, and then treat the wells in each cluster as similar and apply an identified model for the group of wells [93]. It is also proposed to use multi-agent systems to tackle the energy industry's complex, dynamic, and uncertain relations. The agents can be assigned several independent tasks, which are suitable methods when there are conflicting goals. The system can be introduced in several departments, such as production, safety and maintenance, or supply chain management. Then, implement machine learning to introduce learning from past mistakes and provide reputable strategies [6]. Other applications have introduced virtual power plants through using IoT and analyze how the human and non-human actors cooperate [8]. A study comparing machine learning models in the energy industry recognized ten often used technologies and argue that novel hybrid models are superior to the more conventional models. The study gathered 70 originals papers and tested the effectiveness of each model [94]. Machine learning needs data to analyze and learn from, and most luxuriously, a lot of the correct data. Oil and gas in the energy industry are lucky to have a lot of raw data, and from multiple sources [7]. The attached sensors record measurements every few seconds, and a seismic survey can routinely acquire six terabytes of data [95]. As the energy industry is extensive and worldwide, many experts attain critical knowledge to be infused in machine learning methods [7].

2.3.6 Datapoints

To end up with a solid machine learning model, the data points have to meet certain criteria. The accuracy of the model is limited to the accuracy of the input data. In other words, if the input is of low quality, the output will also be of low quality. Thus, there are some demands to the quality of the data [96]. Models often require vast amounts of data to learn, thus the name "Big Data". Courtney and Russom coined the original definition of Big Data [97; 98]. They called it "the V's of Big Data". These are volume, variety, and velocity. Volume refers to the "Big" in Big Data. It is not uncommon to have datasets of magnitude tera- or petabytes. A DVD contains around 5 GB of data. There are 1024

GBs in one terabyte and as many terabytes in one petabyte. To do machine learning regarding precision medicine, around 4000 petabytes are necessary [99]. The V of variety states something about the kind of input data. For instance, if it is structured, complete, balanced, or distributed according to a particular distribution. The velocity aspect will give insights into the way the data is coming into the model. More precisely, it is updated with different frequencies, whether it is 1/100 of a second, daily, or monthly. Further, two more Vs have been put forward [100]. These are veracity and value. Veracity is regarding the quality of the data. If the data quality is poor, the uncertainty may be too high and yield the data useless. These four Vs, the volume, variety, velocity and veracity, combined state something about the value of the data. All four have to be in place for the data to have value to the model, and thus, to the company performing the analysis.

The path from raw data to a working model is long and often contain many obstacles [101]. For a company to get their hands on a solid artificial intelligence model, many man-hours must be put down. Two of the biggest pitfalls are lack of domain knowledge and allocating too little time to the data exploration phase. The first factor is necessary to understand the data, what it means, the types of data, and the value in the data itself. This is paramount to advance in the process and end up with valuable results. This insight is what could make or break a model. The other factor is closely related to domain knowledge. If the data scientist is allocated enough time to learn the domain or work closely with someone who got that perk, the task can prove fruitful. If the domain expert suggests practical A/B tests or hypothesis testing, it will be quick for the data expert to implement them. What many companies misunderstand during the implementation of, for example, a predictive model is that the model only will be able to work on one task. The model will be an expert on only one type of problem. Further, it will require time and resources to implement, monitor, and maintain the model [96].

2.3.7 Classifications theory

Classification is a method that has two distinct meanings, one under unsupervised learning and one under supervised learning. The first one is to establish classes or clusters in the given data and in which the certain classes is not known beforehand. The second meaning is when there are certain known classes in the data, and the aim is to establish a rule for how to classify new observations into the existing classes [102]. Terms related to

classification are accuracy, speed, comprehensibility, and the time to learn [103]. Accuracy is the reliability of the model and tells how often it places the instances in the correct classes. Speed is how fast the code runs. Comprehensibility is the degree of how easily understood the method is if it were to be done by someone else than its creator. The time to learn is a measure of how quickly the classification rules are learnt. There is a trade-off between these four characteristics where every data scientist tunes the parameters for their specific task [102].

Decision trees (DT) are a type of classifiers that classify instances by sorting them based on their values. The instances are classified by representing their feature values as nodes, and each branch represents a value the node can assume [104]. The optimal tree is constructed recursively with dynamic programming and look-ahead capabilities to achieve global optimality [105]. The classification starts at the root node, and subsequently, the instances are sorted based on the feature values [106]. Decision trees can be used to multistage decision making, meaning that they can deconstruct complex decisions into several more straightforward decisions on which to conclude with [105]. The most crucial of decision trees are the choices of the tree structure, and the choice of appropriate feature subset [106]. One of the oldest classification methods is Fisher's Linear Discriminant Analysis (FLDA) which is an unsupervised technique [102]. FDLA reduces the dimensionality and searches in the data that have the largest variance. By searching for the most significant variance, the data will be classified into distinct classes [107]. K-nearest-neighbor is another frequently used classifier that is done through supervised learning [108]. The method is based on the idea that the new observation, i.e. test data, is most likely to be the same as the most popular class within a pre-specified radius. The letter "k" of the method is a number and determines how many neighbors are to be included to classify the new observation [102]. To find the appropriate value of "k" is difficult, and the problem is illustrated in Figure 2.5. The new observations is classified as red if $k = 3$ and green if $k = 5$.

Support vector machines (SVM) is a computer algorithm that is used for classification problems and has been successfully applied to multiple businesses. SVM draws hyperplanes to separate the data into classes and chooses the hyperplane that maximizes the distance from the separating hyperplane to the nearest expression vector [109]. In addition, SVM

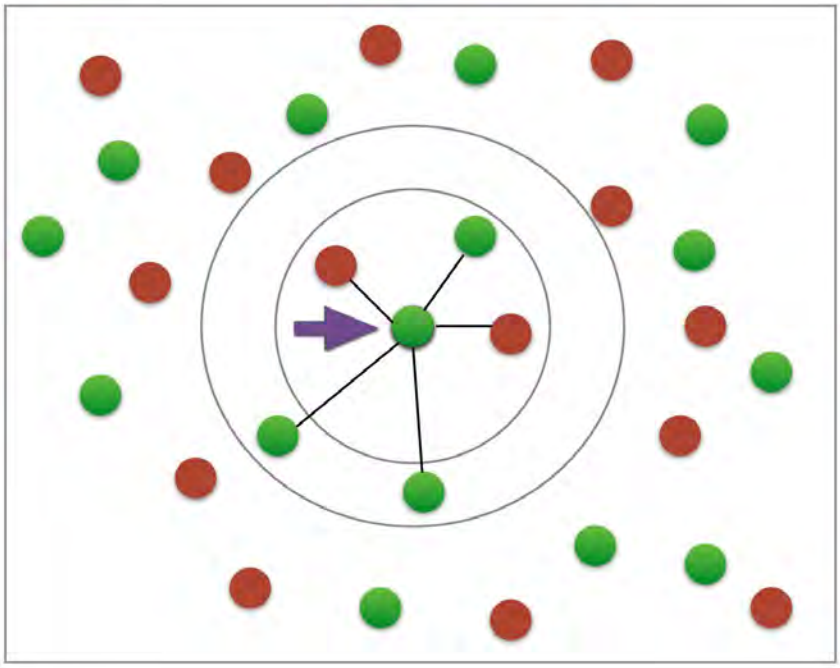


Figure 2.5: The arrow points at the new observation to be classified in the k-nearest-neighbor classifier [108].

will also use a kernel function to add a dimension to achieve higher accuracy than in lower dimensions. Thus, choosing the appropriate kernel function is not simple, but it will yield better results [110].

Classification can be either on a class or a label and is used on categorical data [111]. If it is a class that will be predicted, there is only one class that is the correct one, i.e. the classes are mutually exclusive. For example, if the task is to classify if a picture is of a dog or a horse, the output must be either-or. On the flip side of this, for multi-label data, the labels are not mutually exclusive, i.e. a classification can share labels with another class. For example, if the task is to label how many legs or the color of the fur the dog and horse have in the picture, both classes can get predictions with the same labels. Multi-label is when the labeled data can be overlapping. In multi-class, the possible labels are disunion per definition.

Two main classes of classifiers are the algorithms based on decision trees and those based on neural networks. A decision tree is a way to logically split the dataset into smaller subsets to segregate classes in an orderly fashion. By using many DTs, then aggregate and find the average, the variance will become low [112]. Alone, the variance of a single DT is high. The algorithms based on decision trees further utilize ensemble techniques to

lower the chances of a overfit model. If a model overfits on the training dataset, it will not predict well on other unseen datasets, which is not wanted. One of the DT-based algorithms is the XGBoostClassifier (XGB, eXtreme Gradient Boost) [113]. This is a powerful model that utilizes decision trees paired with gradient descent and leaf weighting. This is efficient on both small and big datasets and decreases the variance while improving the bias. On small sets, it utilizes a novel trick to perform well. While XGB uses boosting as its ensemble method, another DT based algorithm, the RandomForestClassifier (RF), uses bagging. It bags random trees together and then taking the average to compare with the other bags. The strength of the RF is in the number of trees in a bag and the number of bags. Like XGB, the RF has low variance due to its ensemble method. The other main class of classifiers is the MultiLayer Perceptron (MLP) which utilize neural networks (NNs). It consists of many layers with many nodes or neurons, i.e. a neural network. All nodes in one layer are connected to all the nodes in the subsequent layer. The model is among the most popular models used for both classifiers and regression [114]. The model is highly parametrized, but it is possible to control the complexity with the number of hidden nodes. See Figure 2.8.

2.3.8 Classification metrics

To be able to discuss and compare different models, one got to have a metric that enables that. The F1-score metric is a well-known metric in the community [115]. It allows for different weighting and is often viewed as one of the most robust metrics in the classification domain [116]. Its range is from 0 to 1, in which higher is better. To find the correct weighting, experimental approaches are recommended [117]. The F1-score aim to represent the harmonic equilibrium between precision and recall [118]. Confusion matrices (CM) are a great way to visualize the balance between precision and recall, as illustrated in Figure 2.6. In a CM, one wants a higher number on the main diagonal, illustrated by the green squares in the figure, and low numbers in the off-diagonal, i.e. the red squares. Further, the SHAP values are a more novel classification metric from 2017 [119]. SHAP, which stands for "SHapley Additive exPlanations", is a metric that gives each feature an importance score for every classification. In supervised learning, i.e. the data is labeled, and thus the importance score can be found by introducing Shapley values [120]. These game-theoretical values are said to ensure that the reasoning is consistent throughout

the dataset [121]. This is partly done by mixing prior proved explanatory methods [122]. SHAP values range from 0 to 1, where a higher score means higher importance to the model.

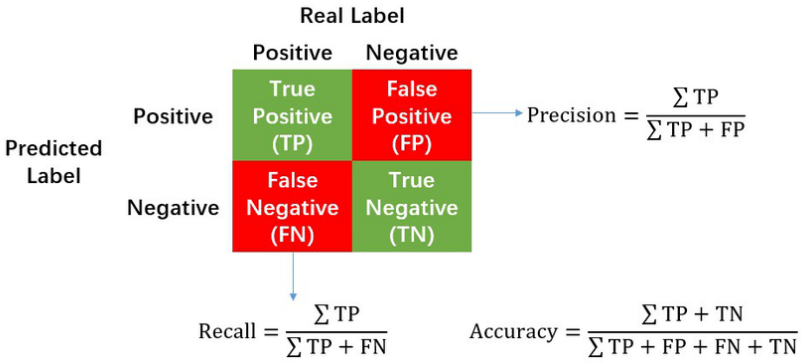


Figure 2.6: The confusion matrix with notes on precision and recall [123].

2.3.9 Scalers

When modelling, scaling of the data could prove helpful. When scaling, it is essential to know which scaler to use [124]. There is a difference between normalization and standardization. Data normalization is when implementing a scaler that scales the data into a fixed range, often between 0 and 1 [125]. The MinMaxScaler and the RobustScaler are examples of normalization scalers [126]. Data standardization is when each feature is scaled by subtracting the mean of the column and dividing it by the standard deviation. This action will move the distribution so that the mean will be 0 and the standard deviation will be 1. An example of this is the StandardScaler. Figure 2.7 shows the effect of the different scalers on the same data. Notice the axis changes and the ranges. There is no conclusion as to which is the best, so multiple scalers should be tried during the preprocessing [127].

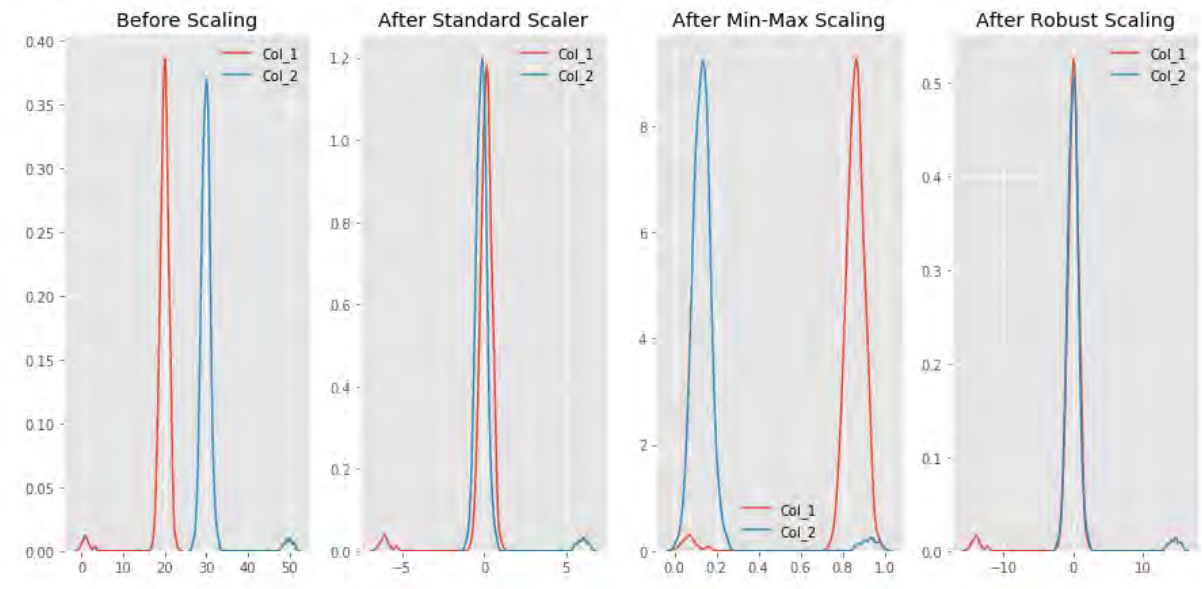


Figure 2.7: Example of different scalers [128].

2.3.10 Predictive theory

To be able to predict future events is a welcoming ability in several situations. Predictive models have long been used in several businesses, such as the stock market, the energy market, retailers, and grocery stores. The fundamental principle is to calculate future values based on statistical approaches with current, and past data [129]. However, predictive models have previously tended to have unsatisfying performance and accuracy due to several reasons. Firstly, there is inherent uncertainty about whether past events will happen again. Secondly, there may be internal bias from the operator in choosing the parameter values and model [130]. Lastly, forecasting is often associated with a time-consuming process [131]. Traditionally, predictive methods have used regression methods, possibly with exponential smoothing to account for seasonality [132]. The most common prediction models in the last decade that can provide meaningful predictions are decision trees, neural networks, and support vector machines [133]. By comparing several pieces of research, it is evident that the standard methodology is to use existing neural network models, which are enhanced with new emerging technology to create hybrid models [134]. Decision trees have shown to be model-independent, efficient, and flexible to scale between application [133].

Fuzzy logic is a method used for solving uncertain and vague problems, thus the word fuzzy. Put against boolean logic, fuzzy logic has no absolute true or false; it assigns values

based on inaccuracies and uncertainties [135]. Fuzzification is the process to convert inputs into fuzzy sets. There on, the fuzzified data will be processed in an inference engine before being defuzzified to provide the output [136]. Fuzzification of other models like decision trees and SVM gives them increased flexibility [135].

Support Vector Machines are covered in Section 2.3.7. The fundamental principles are transferred from classification to regression and prediction. Like decision trees, SVM is also model-independent, efficient, and have a small number of free parameters. In addition, SVM can also guarantee convergence to an optimal solution and handles nonlinearity [129].

The Gaussian Process is a model that utilizes the Gaussian kernel as its covariance function [137]. The GP is non-parametric and based on observed responses from the different training data [114]. The GP model is easy and flexible in use since the hyperparameters can be adjusted by maximizing the marginal likelihood [137].

Sapankevych and Sankar propose an SVM application for time-series prediction that can handle highly nonlinear data. The mentioned application uses support vector regression in which a function is estimated using observed data to train the SVM [129]. Xing Yan et al. concluded that SVM outperforms other applications like artificial neural networks and Bayesian networks because of its simple and easy learning method [138]. Mahalakshmi et al. concluded that hybrid forecasting models based on fuzzy c-means and neural networks yield good results compared to other models [132]. A comparative study by Ahmed et al. concludes that MLP and GP outperform other models like Support Vector Regression (SVR) and Bayesian neural networks for time series prediction [114]. On the other hand, Liu et al. show that SVM outperforms MLP to predict seven days ahead in time about load forecasts in natural gas [139]. In forecasting stock exchange prices, the dominant machine learning technique is identified as neural networks [134].

2.4 Machine Learning Models for Time Series Data

The machine learning models need to have a function to measure how good it is performing to learn how to adjust itself to become better. The error functions serve this purpose. This subsection will present the main error functions in use today and specific machine learning models for time series data. Only the most appropriate time series models for

the data in this thesis will be presented.

2.4.1 Error functions to optimize the model

Machine learning utilizes statistical principles to learn from the input data efficiently. To achieve good scores on the model, it is essential to have meaningful measurements of the performance, including multiple aspects of the model. Therefore, the error is a measurement one wishes to minimize [140]. Which method for error is a frequently discussed area in machine learning. Especially which measurement is the most appropriate according to the learning. The methods to calculate different aspects of errors are obtained from statistical theory. Mean Square Error (MSE) is a standard method to measure the performance of machine learning models. MSE have the ability that its value decreases when the sample size increases, which can be seen in Equation 2.5. In the context of this study, this means that the machine learning error decreases when the size of the dataset increases.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y} - Y)^2 \tag{2.5}$$

An issue with MSE is that the order of loss is more than the order of the data [140]. The solution to this is to take the root of the MSE, which results in Root Mean Square Error (RMSE) and shown in Equation 2.6. RMSE is also frequently used.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{Y} - Y)^2} \tag{2.6}$$

Based on the mathematical characteristics of RMSE, which can be derivated from Equation 2.6, the error term seeks to align with the average of the input data. This is good as it makes it unbiased, but on the other side it will be very affected by outliers [141].

Another measure for performance is the Mean Absolute Error (MAE). MAE takes the absolute value of the difference between the predicted and actual value and does not consider directions of the error [142]. On the other hand, MSE and RMSE square the difference, which makes directions insignificant. The equation for MAE can be seen in Equation 2.7.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2.7}$$

As can be derivated from Equation 2.7, the error term seeks to align the the median of the data. This has the potential to make it biased and treats outliers and inliers predictions the same way [141].

2.4.2 Recurrent Neural Network

Recurrent Neural Networks (RNN) are neural networks that are distinguished from traditional feedforward neural networks. On one side, traditional feedforward neural networks assume input and output are independent of each other, while RNN, on the other side, depends on prior elements from the same sequence. RNN are well suited for tasks in which the system must store and update information from past input to update current input and produce the output. [143]. Some of the popular application areas for RNN is language translation, natural language processing, image captioning, and speech recognition. It is also incorporated in known trademarks such as Siri and Google Translate [144]. Bengio et al. list three requirements of a recurrent neural network: 1) That the system can store information for an arbitrary duration, 2) That the system is resistant to noise, 3) That the system parameters are trainable [145]. In addition, they concluded that RNN is very powerful to represent context, often outperforming static networks, but may be inadequate for tasks involving long-term dependencies [145]. Figure 2.8 visualizes how RNN differs from feedforward by having a loop to include prior input. In addition, RNN uses backpropagation through time, which is an enhancement of traditional backpropagation to find a better fit to the sequential data [144].

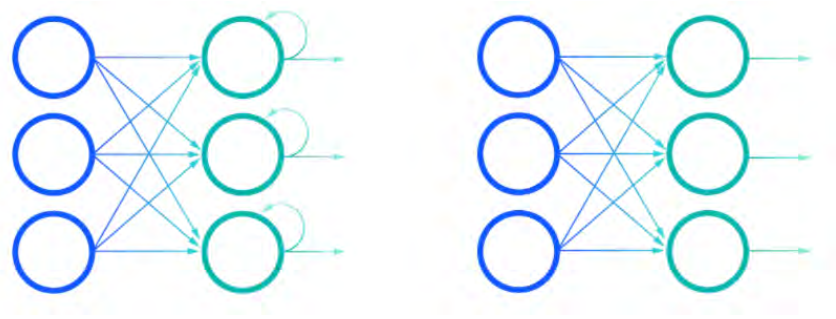


Figure 2.8: RNN on the left opposed to feedforward Neural Networks on the right [144].

The activation function, as the name suggests, determines if a neural network should be activated. The nonlinear activation functions usually convert a node's output to a value between zero and one or minus one and one. The most common activation functions are

sigmoid, tangus hyperbolicus, and ReLU [144]. The ReLU function, or Rectifying Linear activation Unit, transforms all negative input into 0, whilst the positive input remains at their value. This is shown in Figure 2.9.

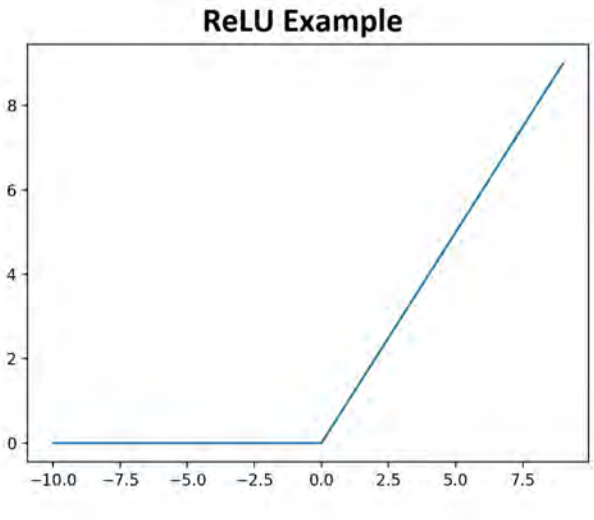


Figure 2.9: Example of the ReLU activation function.

2.4.3 Long Short-Term Memory

Long Short-Term Memory (LSTM) is a machine learning model that takes advantage of gates to decide between long, and short-term memory [146]. Hochreiter and Schmidhuber developed LSTM in 1997 to overcome problems related to long-term dependencies [147]. It is an enhancement of RNN. Therefore, it is constructed in the same way as RNN regarding its feedback loops and its activation functions. The enhancement is such that it can learn from both short and long-term memory and includes dropout-layers to prevent it from overfitting [148]. The main distinction from RNN is that LSTM has cells in the hidden layers. These cells have three gates: an input gate, an output gate, and a forget gate, as shown in Figure 2.10 [144]. The forget gate is the one where data is either passed on or dropped. The gate utilizes a sigmoid function to scale the output from zero to one, in which zero represents to forget long-term memory and a value equal to one represents to remember all from that long-term memory [146]. An interesting aspect of LSTM is that it is Turing complete [149]. This means that it can learn any algorithm that a computer can perform. In addition, LSTM is outperforming several state-of-the-art algorithms in time series forecasting and anomaly detection [150].

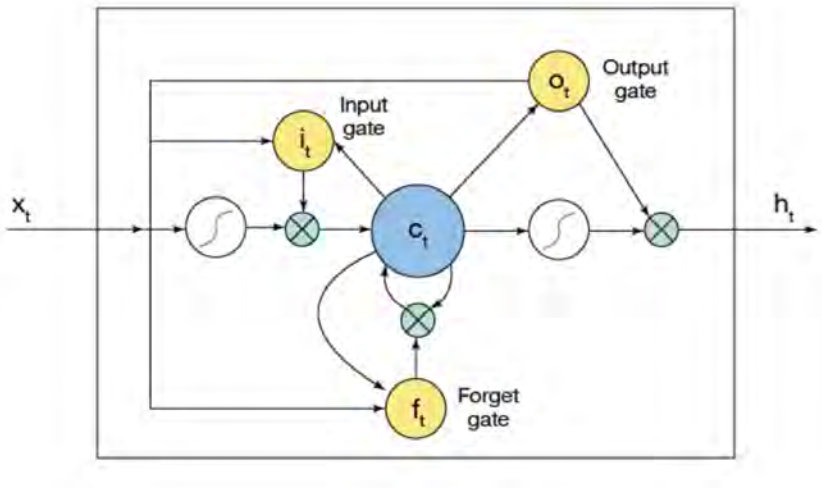


Figure 2.10: The core of hidden layers in the LSTM model [149].

2.4.4 ARIMA

ARIMA is a machine learning method, and the name comes from its three pillars: Auto-Regressive, Integrated and Moving Average. Linear regression attempts to use the past observation of the target variable to forecast its future values. An essential characteristic of ARIMA is that the model is stationary. For a series to be stationary, three conditions must be occurring. Firstly, the mean must be constant. Secondly, the standard deviation must be constant. Lastly, the series must not have seasonality. Most time series generated from reality will not be able to satisfy all three conditions. However, the I in ARIMA, which represents Integrated, will transform the series to make the mean and standard deviation constant. Multiple transformations may be necessary to make it successful, where the number of transformation is known as the order of differencing [151]. The auto-regressive part of the models builds on the statistical concepts of auto-correlation (AC) and partial auto-correlation (PAC). In short, AC is when one looks at how correlated the previous data points are to the current ones. AC will take both direct and indirect effects or correlations into account. This is where the PAC differs from the AC. The "partial" in PAC means that the formula only will use the direct effects from the previous data points. Both AC and PAC uses the lag factor, a term that determines how far into past data points to look for correlation [152]. Lastly, MA, representing the moving average, analyses the error between the predicted next value and the actual next value from lagged observations. The errors are included to perform better estimations in the next iteration. Subsequently, the model will make a forecast based on its past errors [151].

Because of the limitations inherent in the model, it will not provide perfect forecasting but will indicate whether forecasting is possible. Nevertheless, because ARIMA is simple and able to generalize, it is a powerful tool for early forecasting.

2.4.5 Multivariate and univariate prediction models

The input data to be forecasted can be categorized into two groups: univariate data and multivariate data. With univariate input data, past observations are being used to predict future values of the same type. With multivariate, several parameters in the input data is used to produce multiple forecast values and that capture the relations between the input parameters [153]. Because of multiple input variables, multivariate predictions need a different architecture than univariate.

A known problem with RNN is that it cannot learn from early layers when the sequence is long enough. This is because the gap between relevant information and the point where it is needed grows [148]. In other words, RNN will forget what it has seen earlier in the process. LSTM has been designed to overcome the challenge of long-term dependency. The LSTM is implemented with gates that learn which data in a sequence is essential and passes it down the chain [154]. The LSTM also contains a dropout layer to prevent the neural networks from overfitting [148]. Another model for handling long-term dependency is the the Gated Recurrent Unit (GRU) which also enhances the RNN. GRU's method is similar to LSTM in the way that it passes important information to later layers; there are gates to distinguish important information, but it distinguishes from LSTM because it has no cell state on do not use hidden states to transfer information [154]. In addition, Mishra has done a comparative study on which preprocessing methods yields best results and concluded that Attention mechanisms and Deep Convolutional Network perform better with respect to computation time and overall performance [148].

3 Methodology

This section presents how relevant theory was researched and selected based on specific characteristics. It will also show how the datasets were analyzed, cleaned, and preprocessed before initiating the machine learning model. The building of the model is included, and it will be presented reflections during the process.

This thesis is done in collaboration with two independent companies. Each company provided datasets of their projects, most of them already completed. Because the datasets are generated from different companies, the structure and inherent information are logged differently. As each type of dataset requires unique preprocessing, applying machine learning must also be unique. Henceforth, the datasets from the different companies are mentioned as Company A and Company B to make the subsequent sections more intuitive.

Throughout the methodology, different names for the same thing will be used. When speaking of a project, it will be called a dataframe or a dataset. When speaking of a dataframe, two numbers will be utilized to describe them, for example (200, 50). These numbers represent the shape of the dataframe and are very important for its usability. The first number is the number of rows in the dataframe. These rows describe each *instance*, or sample, of the dataframe, often individual activities or weekly aggregates. This number is wanted to be as high as possible for the model to have a large amount of data to train and then test the algorithms. However, working with projects, this number is often limited in size due to the nature of a project. The second number in the shape is the number of columns. The columns describe different *features*, or attributes, of the instance. A low number is often wanted, yet be describing and diverse enough to provide robustness to the model [155].

3.1 Literature Search

A substantial part of this master thesis is searching relevant literature and selecting the articles most fitting to the task. Google Scholar was mostly used to find relevant research in theory regarding both project management and machine learning. Google Scholar has then redirected to sites like ResearchGate, Asce Library, IEEE, and Science Direct. The

snowball method has been used from the most well-written and interesting articles. Articles were also selected based on the number of authors, where it is published, date of publishing, and how easily lines can be drawn to the theme in this thesis. In addition, comparative researches on machine learning models have been studied thoroughly, also how to measure performance on the models. However, a selection of keywords for searching on Google Scholar have been: "project management", "baseline", "energy industry", "state of the art", "comparative study", "machine learning", "planning", "scheduling", and "digitization".

Reports by consultancy firms have been used to find information regarding trends in society and numbers from the industry and other businesses. Consultancy firms are closely related to several businesses and have frequent communication with the leading companies. Major consultancy firms make annual reports regarding trends in society, businesses' adoption of new technology, and interview chiefs to predict how the business will likely be in the future. Therefore, the numbers presented in these reports are thought of as reliable and close to reality.

The best machine learning methods are often found in forums and blogs made by data scientists. Popular sites are towardsdatascience.com, stackoverflow.com, machinelearning-mastery.com, and specific academic channels on YouTube.com.

3.2 Applying Machine Learning on the Datasets

The method for applying machine learning to datasets is most often similar [156]. See Figure 3.1 for a generic overview. The differences stem from differences in the type of data, its size, where it originated, the information in the features, and how complete it is. The subsequent steps are also dependent on the type of model. The *label*, or target, in a dataset is the feature the model will try to classify or predict. The label is always unknown to the model during training to not give away the problem's solution.

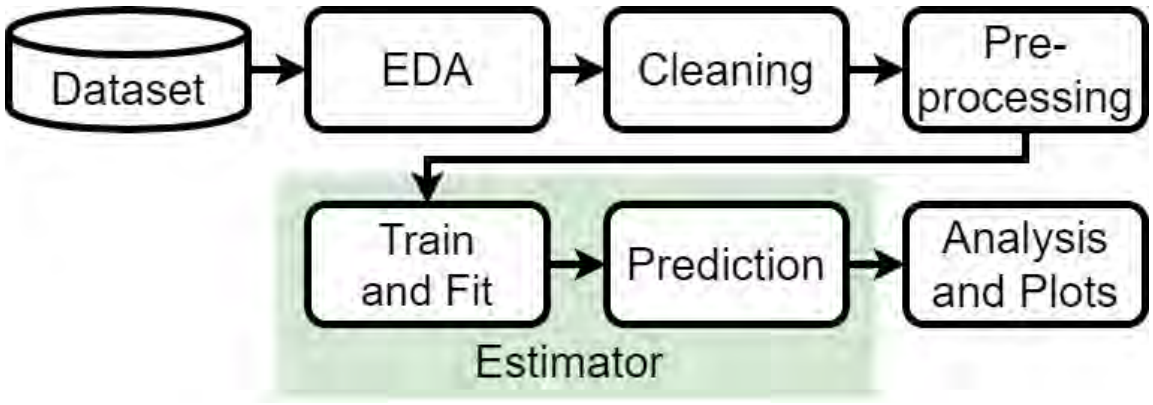


Figure 3.1: Overview of a generic machine learning flowchart, adapted from [157].

The first step the data scientist must do is to gather domain knowledge regarding the data. The exploratory data analysis (EDA) is to research, read documentation, and conduct interviews with companies and experts. However, the EDA is an iterative process. After digging into the data, new questions should have appeared to the data scientist. Thus, the communication with the domain experts should be kept during the whole analysis timeline to seek answers to the questions that could come. In the EDA, which is the first primary analysis, it is vital to understand how complete the dataset is, what kind of information is in the dataset, and inconsistencies.

After gathering dataset-specific knowledge, the data scientist can start cleaning the dataset and making it ready as input to the model. The cleaning is to drop columns or rows with a high percentage of missing values, referred to as "NaN" or "NaT". If the feature contains valuable information and cannot be dropped, it is possible to fill in the missing values. One way to fill the gaps is to impute a value based on, for example, the mean of the feature. However, this is most often not wanted as the inserted values can be inaccurate and might inherent bias by the data scientist. After the cleaning is done, the data scientist should have kept the essential features used to solve the problem.

Preprocessing is the step in which the dataset is made ready for implementation to the model. The processing normally includes scaling the columns, one-hot encoding, and other methods to prepare the data for training. However, certain datasets need extra preprocessing. For instance, it might be needed to augment the data to generate more datasets for training and pad the dataset to control the values in special areas. Main activities in this step is variable classification, transformation, and creation [158]. Other

preprocessing tasks include splitting, labelling, and scaling. The choice of the scaler is a decision that may have a major impact on the results and should therefore be thoroughly analyzed, see Section 2.3.9.

After preprocessing, the implementation of the model can begin. First, the data scientist must have decided on what the model will try to accomplish, i.e. choose a target label. Then, the most appropriate model for the problem can be implemented. When setting it up, the data scientist must decide on the model's parameters and start with an initial configuration of the hyperparameters. The hyperparameters are the parameters specific to the model, such as number of layers, number of epochs, number of batch size, number of nodes, how far into the past and future to look in time series analysis which optimizer to use. Most often, the hyperparameters must be adjusted to fit the dataset. To find these functions, a grid search and the "RandomSearch" can be utilized.

A grid search is a time-consuming but valuable method to find the best combination of the hyperparameters. The grid search will set an interval for the values of the hyperparameters and run the model with all combinations [159]. Because the search tries all combinations, it is expensive both in time and computing power. After the grid search, the model can be run with a unique set of hyperparameters. RandomSearch is similar to a grid search, but with only a subset of the possible combinations, chosen by random [160]. This attribute makes it faster than the grid search. Due to the randomness inherent in the training used in machine learning, Monte Carlo Simulations (MCS) is preferred to apply in the end to obtain the average values of the results, if applicable.

To perform machine learning in Python, several python libraries have to be utilized. A library, or package, is a set of functions that aid the data scientist by offering optimized and complete functions. The libraries make the process faster and minimize the number of bugs and human errors. The three main used libraries in this model are "Pandas", "Tensorflow", and "SKLearn" [161] [162] [163]. Pandas is an open-source, high-performance, and easy-to-use tool to aid in the structuring, cleaning, and interpreting the data. Pandas can be thought of as an Excel spreadsheet where one can perform row or column-based operations such as filtering, sorting, and calculations. When a dataset is imported into Pandas, it is called a DataFrame (DF). Tensorflow is Google's machine learning platform, where the sub-library "Keras" was utilized to perform machine learning on multivariate

time series data by utilizing the sequential algorithm LSTM. SKLearn, or SciKit-Learn, is an add-on that has many functions related to data cleaning, data prepossessing, machine learning algorithms, and prediction measuring. So instead of the data scientist writing and implementing a classification algorithm from the bottom up, she can use the complete and optimized algorithms that experts have written. Other necessary libraries are "Numpy" for data processing and operations, "Matplotlib" and "Seaborn" for visualizations, and "Tpot" for automatic machine learning pipelines.

In terms of hardware, the models have been run on a laptop's GPU and CPU. The GPU was utilized when Tensorflow's Keras. The rest of the analysis in this thesis is run on the multi-core CPU. Big MCS often took around 12 hours to complete. Further, an even more powerful setup or cloud service might have yielded other or better results, but these were not implemented due to the scope of this pilot project. Furthermore, when possible, the simulations and algorithms ran in parallel to maximize the number of computations, only limited by the hardware, i.e. GPU and CPU.

3.3 Company A

Company A is an operator in the energy industry. The data from Company A is from a project owner perspective and consists of four projects of a similar sort. The fourth project is not yet finished and is meant to test and evaluate the model. The goal for Company A is to predict the next week's value throughout the projects. All projects are logged in the same way. The follow-up on the projects was on a weekly cut-off. Therefore, the rows are set as the number of weeks from the start. The columns contain progress data for the different disciplines in the project. For example, the columns representing a discipline may be planned or expended man-hours per week or tonnes per week. There is a total of ten disciplines, in which each can represent an input to the model. After preprocessing and cleaning, the final dataset for each project has a shape from 97 to 171 in length with 13 features.

3.3.1 Exploratory data analysis

The exploratory data analysis will discover the characteristics of the data and how complete it is. Table 3.1 present the initial shapes of the datasets. As can be seen, the number of

features is more than twice the number of weeks in all projects. The percentage of unique values in each feature were plotted to give an understanding of the variety in the data [97]. Additionally, the percentage of the value 0 was plotted to see which features will provide valuable information. If there are features with a high degree of 0, the rows with different values may be necessary. On the other end of the spectrum, the features with few zeros may score higher in veracity [100].

Table 3.1: The initial shapes of the datasets.

Project	**# Weeks**	**# Features**
1	172	484
2	198	484
3	184	484
4	147	484

From the mentioned technique, it was discovered that many features had no data. On the other side, the features that contain data are completely filled in. These features were most often the expended quantities during the projects. The analysis showed that the projects were logged with the same template. Thus, the features between the datasets are identical, but the number of weeks and the values in the cells differ. Because the projects were logged with the same template, the features contain data of the same sort and the same magnitude. Even though the features are identical between the projects, they were not always filled in all datasets. For example, if a feature in project A is not filled, but the same feature is filled in project B and C, the feature cannot be used in the model because the model will not be able to compare them. Each feature is associated with a particular discipline. Therefore, the disciplines' features must be completely filled in all projects to be included in the model. Thus, the feature that contains data in all projects were discovered and stored. Unfortunately, the fourth project had to be dropped since it did not contain the necessary features found in the others. This action is regrettable but necessary since the model must have similarity between the datasets.

According to the theory in Section 2.2.4, the earned value (EV) of the activities should be used to compute the overall earned value from the project. According to theory, the relation between actual value, often referred to as expended value, and earned value, will show productivity in relation to the plan. As Figure 3.2 shows, the earned value has a substantial linear increase over a few weeks. These data were considered unreliable as

they could have been logged inaccurate, such as values for several weeks being combined for one or a few weeks. Figure 3.2 shows that the actual progress follows a smooth curve with no sudden increases or decreases. The figure shows the cumulative values. The goal put forward by Company A is to predict next week's value, and thus, non-cumulative data would be more appropriate. The significant increase in EV in Figure 3.2 will also yield large instances for the non-cumulative values. Additionally, the most filled features were for expended quantities. Therefore, the actual progress will be used as a label of the overall progression of the project. Because the actual progress is used, the features containing actual values are the ones to be included in the model.

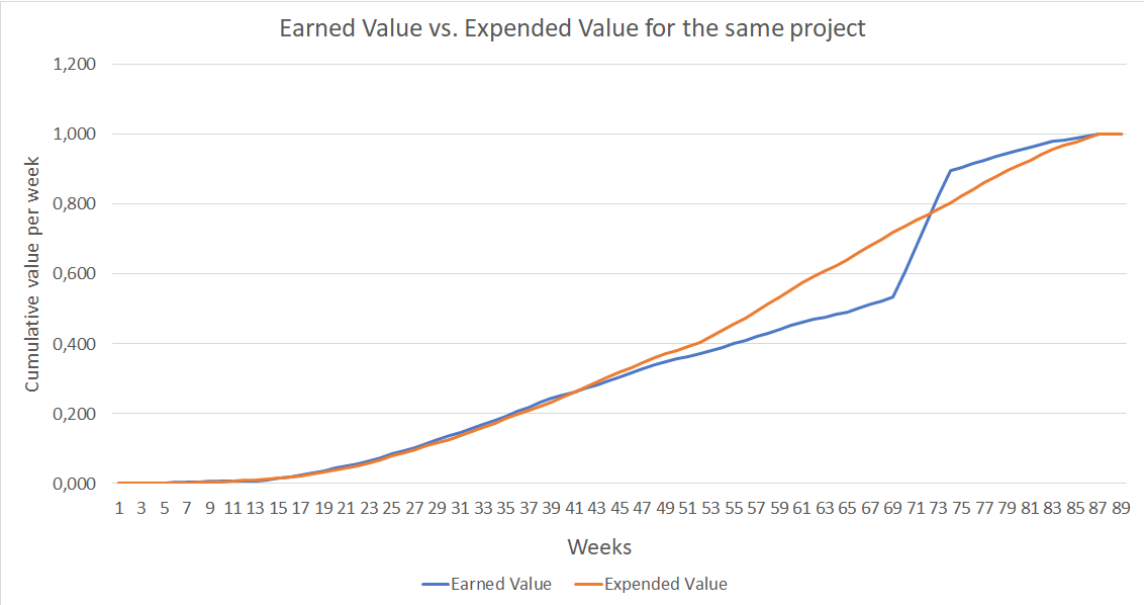


Figure 3.2: Expended value plotted against earned value for the same project.

As mentioned, the projects received from Company A were of varying length, resulting in a number of weeks from 90 to 170. These lengths might be adequate to obtain high accuracy if the characteristics of the project's production were very similar. As shown in Figure 3.3, this is not the case for the received projects as the production curves are quite different. Even though the projects are received from the same company, are of the same type, and manufactured in the same region, their internal processes have differed. The differences for when the resources are expended might stem from several reasons, and were not further investigated. However, interviews with Company A deducted that it could come from the different scopes of the projects, the different levels of complexity, the different sub-contractors, and the sub-contractors unique production methodology. Figure 3.3 shows the amount of work done for the same discipline between the projects.

The figure shows that for Project 1, the first spike is around week 129. For Project 2, it is around week 73, and for Project 3, it is around week 41. There is no clear pattern when the spikes appear that the algorithms can learn from to predict future values. This evidence is also true for the other disciplines, see Figures A.8 to A.10.

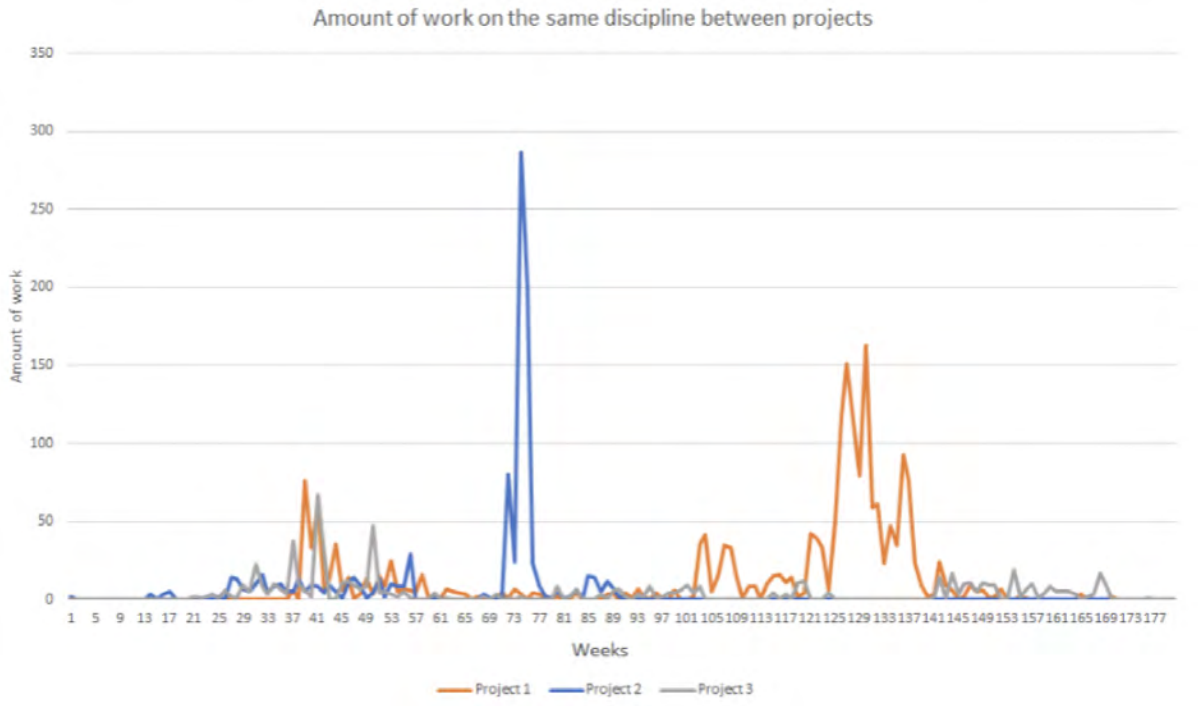


Figure 3.3: The figure shows the work done in the same discipline between projects.

3.3.2 Cleaning

Often, the data is far from ready to run through a machine learning algorithm when receiving it. For instance, features with a high percentage of missing values must be dropped or filled and features considered only to add noise to the data must be dropped. The imported datasets had the number of weeks according to project length and 484 features.

Only the first dataset will be discussed in-depth, but similar cleaning had to be performed on the other datasets. By inspection on the first dataset, some features were based on the monthly cut-off, and some were based on the weekly cut-off. To keep the resolution as high as possible, the features based on the weekly cut-off are included in the model. Consequently, the ones based on monthly cut-off were dropped. The first weeks in the features representing a type of production were empty because the type of production had not yet begun. These were filled with 0 to represent no production. Further, there were

weeks with missing values after production had started, which were classified to represent no production for that week. Also, it was natural to insert 0. Further, NaN values that appeared suddenly was filled with the value 0. After interviews with the representative from Company A, this was a valid method for the respective features.

It was discovered that the last 16 weeks of certain features had missing values. By deeper inspection and interviews with the company, domain knowledge was gathered to understand how to deal with these weeks with missing values. After the planned finish, some activities were not yet done. In other words, some parts of the dataset carried on while others seized. The representative said that the activities in the dataset performed after the final week were not of interest to them, based on multiple reasons. Thus, these last 16 weeks were dropped in the dataset. For a column to affect the model, the values in the column must be different [164]. If they are all of the same value, the column will not affect the model. Thus, the columns that only have one unique value were located and duly dropped.

Front fill and back fill were used to fill the rest of the missing values. The characteristics of these were that they appear not to follow a pattern of missing values. The missing value might be rooted in the fact that there was no production that week. Front fill keeps the last known value and inserts it when it finds a NaN value. See Table 3.2 for examples. The opposite is applied for back fill, where it keeps the next known value and fills backwards from there. An example of this is in Project 1's week 31. None of the man-hours features contained values while the other features were filled, so the front fill method was utilized since it seemed like there was supposed to be production.

Table 3.2: Examples of the Pandas functions front and back fill.

Original data	**Front fill applied**	**Back fill applied**
10	10	10
NaN	10	12
12	12	12

As mentioned in the EDA in Section 3.3.1, the features were not always filled across the datasets. For the machine learning alorithm to learn between projects and make predictions, the set of features, i.e. available columns, must be equal and filled for each project. The model needs this consistency to be able to compare them. The EDA

discovered which features were equally filled and gathered them. Consequently, some disciplines had to be excluded from the model. After discussions with the company, it was concluded that dropping these disciplines did not weaken the remaining resolution of the dataset. Then, it had to be decided which of these features were interesting for the model to predict the expended value. After this step, the model had five different disciplines, summing up to 15 features to learn from. Three of these features that were available in all the datasets were the signing of different check records. These often occurred at the end of the projects. The scale of these data points was varied, and it seemed varied how detailed these were tracked. Thus, after interviews with the representative, these features were decided to be dropped.

3.3.3 Preprocessing

Due to the limited dataset size, it was decided to augment each dataset into multiple datasets to have more data to train the model. Then, the model will be tested on the original dataset. The reason to augment is to get a more robust model that will be able to receive a more varied dataset in the future. A robust model is stated to be more generalizable, i.e. the model does not overfit as quickly since it has seen many projects that differ from one and another [165]. The augmentation function randomly selected one of seven ways to augment the dataframe. Multiple augmentation functions were tried in the first implementation. The augmentation could either be one or multiple augmenting processes performed on the dataframe. Even if the function calls to augment two times by the same method, the augmentation itself is different each time it is called. The changes will be equal between all features when the method is chosen. Thus, if some amount scales the label, all the input features included in the model are altered by the same scale. Therefore, proportionality is kept during this augmentation. However, some of these augmentation methods proved to be unsatisfactory. This could either be that the augmentation was too drastic. Alternatively, the length of the augmented dataset got changed too much. Further, changes were made to provide more diversity in the augmentations. For instance, all probabilities for occurrence were set equal to 1. For the augmentation to occur, the method must be called, and then there is a probability function to adjust how often the method occurs after it is called on. Therefore, if a method was called, but the method's probability of happening did not occur, there would be no

changes to the feature.

Another change was to add noise and alter the values over time. However, this leads to shifts of zero values. Because values of zero are important as an off-switch, drifting of the values had to be reduced. To do this, the scale of the drift would be reduced to minimize the effect. The Pandas function "clip" was used to limit the possible values. Thus, negative values would be replaced by 0. The lower clip of 0 was then implemented to all the features except the label. By implementing this clip, the augmentation function could more freely utilize functions as drift and the AddNoise. The AddNoise function will alter the values within a defined standard deviation of the input value [166]. An important attribute of the AddNoise method is what kind of operator the noise is added with. The first is additive, and the second is multiplicative. The first one is chosen by default and the one utilized in the first iterations. However, during the inspection, it was discovered that the multiplicative was more applicable to this dataset. The source code of [166] shows the way these methods are being multiplied. That is, noise near 0 remains close to 0, and higher values yield more noise. The formula from the source code states: $X_{aug} = X \cdot (1.0 + noise)$. The last step for the augmentations was the Convolve method. This method will perform a convolution on the time series data and augment it by sliding a certain window past the values, which provides the altered output.

Padding of the dataset is a method to add generic values around the dataset to allow the model to capture the whole actual set [167]. Regarding time series data, mainly two types of padding exist; pre and post padding. According to Reddy and Reddy, pre padding is the one to utilize when using an LSTM (sequential) model [167]. More specifically, pre padding will teach the model to always start on zero before the progression starts. The pre padding was implemented by taking the existing dataframe and add a certain number of weeks at the top of the original dataframe, in which all values are zero. Since the original data is date indexed on the weekly cut-off dates, the padded dates had to match this format. The necessary dates were found, then these two dataframes were then concatenated into one continuous dataframe. In this dataframe, the leading data point in all features was of value 0. The date index was changed into weeks to ensure the anonymity of the projects and the company.

By interviews with the company, it was concluded that a holiday feature should be

implemented. The specific dates were given partly by the company and partly by looking at the Internet. Due to anonymity, the holiday citation will not be cited. This feature was created in Excel by giving the weeks with holidays the value of 1. This is called one-hot encoding. One-hot encoding will make new features and set the value dependent on if the activity contains a specific value. Another example is that strings, i.e. text, has to be one-hot encoded. This is since the algorithms of the models is not capable of handling strings as input. That is why one must vectorize these attributes so that the model can understand them as input [168]. Thus, each dataframe had 13 features that included discipline-specific operations, man-hours in the specific discipline, dates, holidays, and the expended value.

3.3.4 Train and fit the model

Based on the theory in Section 2.4, it was decided that LSTM were the most appropriate model with the given datasets. Recurrent Neural Networks have difficulties in long-term memory, and thus, outperformed by LSTM [145]. On the other hand, ARIMA is a linear model best suited for stationary data with a trend [169]. Also, ARIMA is most fit for univariate time series, and thus, it was considered as outperformed by LSTM [170]. The model needs a label to compare its training against and will also be the type of value the model will predict. After discussions with the representative, it was concluded that the model would try to predict next week's expended value and the shape of how the production progresses. As mentioned in Section 3.3.1, the expended values are better logged than the earned value which supported the choice of the label. Additionally, the chosen label increases to a maximum and then decreases to zero again with small weekly fluctuations. These types of data are not suited for ARIMA [169]. Therefore, the decision on LSTM remains the best model to be implemented. During the training, the way the model learns is by looking at training and validation loss. Training loss is the error each epoch experience. The validation loss is how well the current weighting of the nodes are performing on an unseen part of the dataset.

Different methods were tested in the search for the best hyperparameters. This was an extensive search that utilized a grid search. The tests calculated the error terms from all possible combinations of hyperparameters. The calculated error terms were the ones mentioned in Section 2.4.1, namely RMSE and MAE. The grid search will provide

valuable insights into which values of the hyperparameters the model prove the best. The hyperparameters to test was how many weeks from the past the model could see in each time window (n_past), number of nodes, number of epochs, number of batches, and number of augments. Because the model is sequential, it will run through subsequent time intervals commonly called the sliding window. Figure 3.4 shows the method of the sliding window. The values included in the search are shown in Table 3.3 and results in 32 combinations of the values. The number of epochs and batches determine the amount of training for the model. Batches are how many samples, in this case weeks, the model processes at one time and the number of epochs, are how many times all the training samples are trained. Therefore, the number of epochs is closely related to the overfitting of the model. Too many epochs and the model will overfit, too few, and the model will not have time to dial the nodes in an optimum way. A model that overfits will only be able to perform well on the one dataset it trains on [171]. Thus, it will lose its ability to capture essential elements in new datasets.

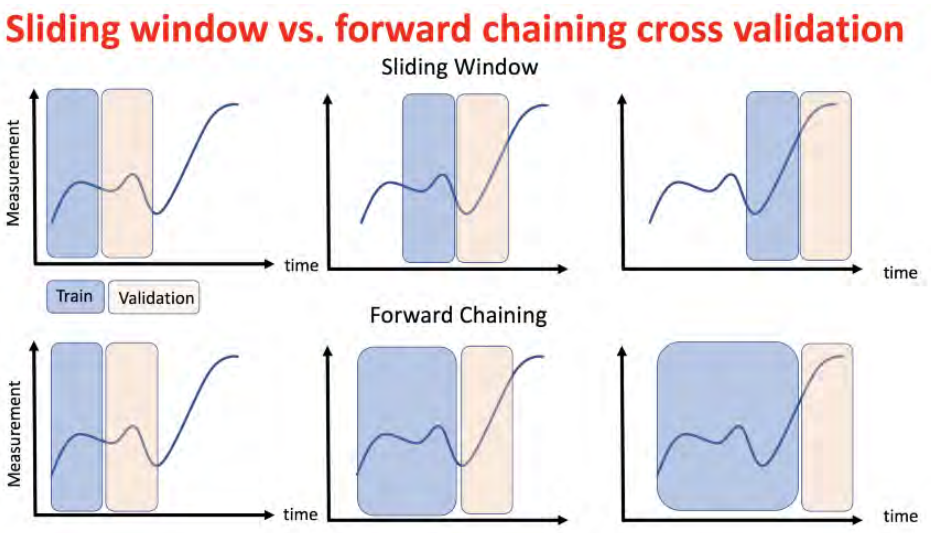


Figure 3.4: Sliding window and forward chaining illustration [172].

Table 3.3: The values for the hyperparameters to be tested in the grid search.

Hyperparameter	First value	Second value
n_past	3	12
# of nodes	62	102
# of epochs	2	8
# of batches	8	30
# of augmentations	8	12

Besides the grid search, the model will also be tested with different scalers to understand the importance of an appropriate scaler. The scalers to be tested are StandardScaler, RobustScaler, and MinMaxScaler. See Section 2.3.9. However, the test on scalers will include fewer configurations than the grid search. Because this model will predict future expended value, the model should punish large deviations rather than small ones. Therefore, the results of the search will lay most emphasis on the error term of RMSE. The model was implemented with two hidden layers. It is a constant discussion in academia regarding deep versus shallow neural networks [173]. Due to the nature of LSTM as time-dependent, the decision was to implement with two layers and a high number of nodes. This decision is based on characteristics discovered in the EDA. It is the nodes that learn how to weigh the different features according to their perceived importance to minimize the loss function [113].

The data from Company A contain multiple complete datasets, which are set up with equal layouts. Therefore, the method of nested, or chained, cross-validation can be utilized [115]. Typically when dealing with validation, the most common method is k-fold cross-validation [174]. In this model, this would not be feasible due to the time dimension. Further, since events in the dataset may only occur in a specific time frame of the process, traditional folding is a sub-par approach [175]. Since the LSTM model cannot be scrambled and traditionally folded, the chained method proves as a remedy [176]. The chained validation method uses the fact that the model can learn and validate across datasets in a sequential manner. The model can train on the first dataset and test on the second, which is called the first chain. Then, for comparison reasons, the model can train on the first and second dataset, and test on the third called the second chain. After this, the three first datasets can be used to train for testing on the fourth, and so on. This canonical method is illustrated in Figure A.4. How this method differs from the sliding window method utilized within each chain is illustrated in Figure 3.4. From this idea, a custom chaining algorithm was written. It is described in Figure 3.5. In the figure, the general chain is illustrated to the left. It will train on the first N dataframes, then test in the last available dataframe. Ideally, the first chain would test on the second dataset. However, the order was adjusted to have more comparable results. Therefore, both the first and second chain will test on the third dataset as illustrated in Figure 3.5. Thus, the main difference between this method and the one shown in Figure 3.4 is that

whole projects are utilized at the time. Further, augmented datasets from the original projects are used to make the model less likely to overfit. By chaining, the results might also tell if the model has higher accuracy by training on two datasets instead of one. The grid search was tested on the second chain. Therefore, it trained on Project 1 and 2, and then tested on Project 3.

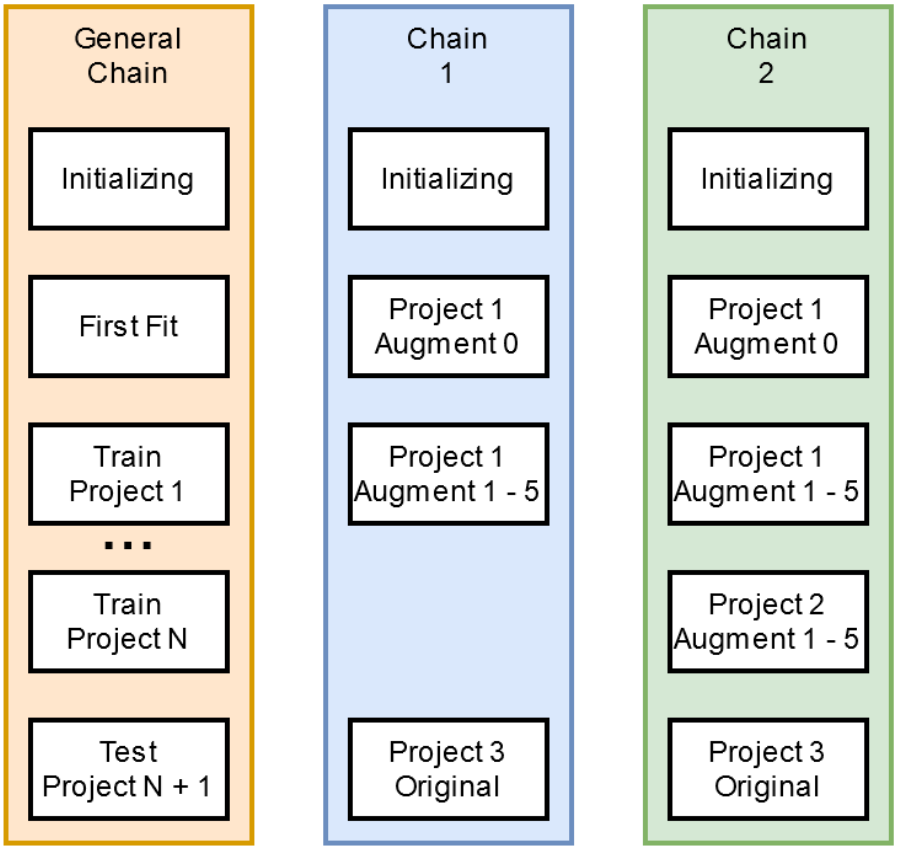


Figure 3.5: The custom chaining algorithm for two chains and the number of augments set to 5.

3.3.5 Summary of the complete model

After cleaning, the dataset contains three completed projects, each with the number of weeks according to project length and 13 features as shown in Table 3.4. The features represent production done in the different disciplines, both in the number of man-hours and in material units. The LSTM-model is set up with two layers and looks 3 or 12 weeks into the past to predict the next week's expended value. The model will make a forecast for the next week for all weeks throughout the project.

Table 3.4: The final shapes of the datasets.

Project	# Weeks	# Features
1	171	13
2	97	13
3	112	13

3.4 Company B

Company B is a service provider in the industry. The dataset from Company B consists of one project. The company wants to predict the next week's value and classify activities that start on time. The data is gathered from a supportive software for the project management team. The gathered data contain all activities that constitute the project. After preprocessing, the dataset from Company B is of the shape of 13047 rows and 110 features.

3.4.1 Exploratory data analysis

For the project from Company B, the exploratory data analysis (EDA) consisted of multiple parts. Initially, it is required to understand key characteristics regarding the physical project. For the model to provide valuable insight, the characteristics of the label should be reflected in the information it receives. The information gathered in the EDA is vital for understanding the magnitude and the number of resources in the project. Also, it is vital to extract in which region of the world each sub-project of the production is done [17].

Analysis of the project management software showed how data is stored and logged throughout the project. The received file contained over 7281 activities, six baselines, and a total duration of five years. The software is custom made to fit the contractor and type of project. Because of this, there are user fields determined by the contractor in addition to the standard fields for project management. The software gathers information on each activity, and the amount of information depends on the activity's discipline. It contains up to thirty data points if it is an activity done in-house, whereas if it is an outsourced activity, it contains under ten data points. One of the fundamentals of the software is to track progression in the project. To do this, there are logged several start and finish dates. There are two primary plans used in the project of Company B, the baseline plan and the current plan. The baseline plan is set when the project starts, and then

the dates are updated to the latest baseline. Each activity has its own set of dates, for example, early start (ES), early finish (EF), late start (LS), and late finish (LF). Analysis showed that the ES is not logged if the activity is reported to be finished. Instead of using the ES, the program has created an early start analyzed (ESA), which equals the EF minus the duration. Because the EF and duration are logged for every activity, the ESA have no missing values. The current plan is used to show the progression of the project. The progression is reported every week, commonly mentioned as a weekly cut-off. The activities will update the percent of completion each week, and the data is used to calculate the remaining quantity until completion. The current actual start is set as the Monday in the first week of reported progression for that activity. The current actual finish is set as Friday in the same week as the activity is reported to be complete. In the software, there is reported that 5311 activities have an actual start, 4990 have an actual finish, and 4852 are reported to be complete. How complete the dataset is filled concerning the total count of activities is shown in Table 3.5. In addition to the mentioned dates, the software also logs current early start and finish and current late start and finish. However, the analysis showed that these features were missing much data.

Table 3.5: A summary of how complete certain dates are filled in the software.

Type	# of activities	Degree of filled
Actual Start	5311	73%
Actual Finish	4990	69%
Complete	4852	67%

The data in the software is stored in its databases and must be extracted through SQL codes. The data is categorized into several tables whom each has its purpose. For instance, one table is for the logged progression while another is for the baseline updates. Therefore, when first extracting the data through SQL from the project management software, it must be extracted from multiple tables with varying resolution. Further, the data had to be merged to gather everything in one dataframe. To minimize overlap, corrupted values, and to have the most up-to-date version, the software writers had to be contacted. After the professional programmers had written the correct SQL, the shape of the dataset was 1 386 496 instances by 58 features in total.

The software stores other important information about the project and characteristic for each activity. These include duration, float, scope, discipline, phase, description, main area,

activity role, and category. Duration, float, and scope are mostly filled for every activity as an integer. The rest of the characteristics are dependent on the activity, summing up to millions of combinations. However, the analysis showed that the characteristics phase, discipline, main area, activity role, and category follows a pattern. First, several categories are unused by the project team. Secondly, the characteristics seem to be linked together. For instance, if the main area is general and the phase is design engineering, then the main category is engineering. This made the number of combinations substantially lower. The description is a feature containing words to describe necessary information about the activity. Humans fill it out in the project, and the feature has a high count of words.

Humans and the software partially fill out the dataset. Due to the manual registration, the information is prone to mistakes. The original information might be wrong, in the wrong place, or the information might be changed later in the project by mistake. Analysis showed that several dates that should be constant changed during the project. For instance, the date for the current actual start changed multiple times after the initial start. Additionally, the dates for actual finish and EF changed after the activity was finished. It was first suspected due to the merging and exportation performed by the SQL. After interviews with the representatives, it seemed like something wrong had happened in the back end of the program during the execution of the project. The wrong dates were caused by later fixing to adjust reporting where the progression was set to zero at a certain point. Thus, the actual finish was again set to the Friday in the current period week. For some activities, the ESA was set to a date before the start of the project. Interviews with the representatives for the software concluded that the change sprung from a change in the duration. The duration may have changed because the initial analysis of the activity's length was wrong or could have been registered wrong.

3.4.2 Cleaning

Interviews with Company B suggested which features could be cleaned first. All activities flagged in the "on_target"-feature were dropped because this represents an activity that is set to always be on schedule. Since this model aims to predict next week's EV and how it progresses, these automated activities decreased the cleanness of the data. Then the "on_target" feature itself was dropped since it now contained the same value for all the activities. Further, activities with a duration equal to zero were dropped. By

investigating the affected activities where this was the case, it was found that these were mostly procurement and milestones. After discussions with Company B, the conclusion was that activities regarding procurement and milestones could safely be dropped. After the preliminary cleaning, the shape of the dataset was 199 433 instances by 38 features.

Then, the dataframe was sorted on activities, descending from the earliest start. The sorting showed that activities were logged for a large part of the project, even though they only lasted over a few weeks. Consequently, there were a high amount of duplicate rows above and below the planned interval of the activity. Duplicate values do not add new information. Thus, they were determined to be dropped where the "current_progress" either was 0 or 100. When performing this action, it is crucial for the dataset which duplicate should be kept to maintain consistency. To keep the valuable information regarding when an activity logged its first progression, meaning it went from 0 in progress to a non-zero value, the last duplicate of 0 were kept. Thus, the continuity for the activities and the resolution in the dataset were kept. The opposite was the case for the duplicates when the activities were finished, and progression was set to be 100. The first duplicate was kept to maintain information on when the process was completed. Activities that only have one entry in the dataset means either they have a start or a finish, but not both. Therefore, the activity does not have sufficient information and must be dropped. This was an edge case that affected around 200 activities. Lastly, features with only one unique value were dropped. At the end of these drops, the shape of the dataset was 13 047 instances by 38 features.

3.4.3 Preprocessing

Machine learning algorithms should have input as floats or integers, not as strings [177]. The original data contain several features with strings, such as discipline, phase, description, main area, activity role, and category. Therefore, one-hot encoding was applied, which increased the number of features from 38 to 215. However, the analysis showed that several new features have more than 98 percent of zeros. Also, interviews with Company B provided input on the degree of importance between the newly created features. After these interviews, several of the new features could safely be dropped. As previously mentioned, dates in important features were changing where they should not have. These dates are the current_as, EF, and ESA. The remedy came by front filling them as illustrated in

Table 3.2.

The features the model had access to could be sorted into three main categories. The three categories were features regarding the scope and duration of the activity, features created by keywords in the description, and the activity codes used by the company. Examples of the first one are duration and resources, see Figure B.8. The first feature set, henceforth addressed as "Numbers", consists of six features. The descriptions contained a lot of similar words that were considered to be interesting to the model. See Figure B.2. A custom function was created to separate the keywords in the description. First, all the descriptions had to be extracted and separated based on the space character. The descriptions had a minimum of 1 word, a maximum of 13, and a mean of 6 words. Further, these split descriptions were concatenated and transformed into lower characters. This was performed to make the comparisons easier to perform. If this was not performed, the words "test" and "Test" would have been counted as two different words. Special characters such as "%", "&", "'", "(", and ")" were removed. Then all words were counted and listed with their count number. The total count of words were over 80 000, and the count of unique words were 528. Many of the words were of no apparent value to the model and was duly dropped, such as "nan", "to", "and", and "of". Then, for each row in the dataframe, if the description contained one of the 28 most occurring description keywords, the respective feature would get the value 1 for each keyword mentioned. In short, the top 28 keywords got one-hot encoded.

Regarding the last category of features, the codes used by the company, a mapping tool had to be used to one-hot encode it. The way, for example, an activity that is under the "General" class of the "main area"-code is structured in the database is by vectorization. Thus, to read what this vectorization value means, it was required to map it back into a string. The result of this mapping is shown in Figure B.4. Further, some of the features logged the same information but at different depths. This created duplicates in which the feature containing the lowest sum of values was dropped. For example, if 40 one-hot encoded activities had the feature "Discipline_Planning", and only three one-hot encoded activities had the feature "Main_Discipline_Planning", the feature with the lowest number of one-hot encoded activities would be dropped. This was to ensure that the feature with the highest value to the model survived and that the other feature which stated the same

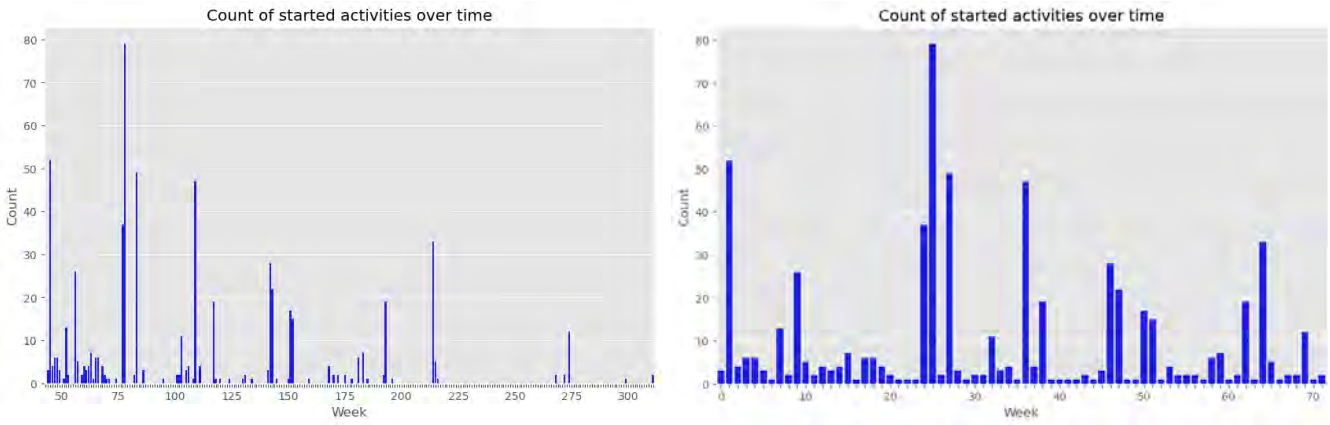
attribute was dropped. Structuring features into sets reduces the computational cost and reduce the possibility of confusion in the model.

In Table 3.6, the different feature sets are presented, and the number of features in each of them is shown. This is implemented to test the robustness of the model by using feature selection [178].

Table 3.6: The number of features in each feature set.

Feature Set	# Features
Description	28
Codes	78
Number	6

In Figure 3.6, the number of starting activities per week throughout the project lifetime is plotted in a histogram. An activity is defined as started the week that the progress goes from zero to another positive value. The figure shows that the activity starts are spanned out over approximately five years. However, there are time slots where it does not start any activities. The sporadic characteristics would confuse the sequential model. The obstacle was solved by dropping the weeks where there are no activities that start. In practice, the bars in Figure 4.7 is squeezed together, and thus, removing the gaps between them. Then, the weeks are reset, so the first week, which is week 44 before the drop, becomes week 0, and so on. However, if the model were to look at the overall progress of the project and not the subsection of start activities, these intervals of no started activity would contain valuable data. However, for the analysis in question, these points in time are not applicable, and the need to reset the time axis is necessary. After this step, the shape of the dataset is 72 weeks by 110 features.



(a) Before drop of empty weeks.

(b) After drop of empty weeks.

Figure 3.6: Plot of the starting activities per week.

3.5 B1 - Time Series Analysis

Company B emphasized the value of having a model powered by machine learning to provide accurate predictions. First, it was interesting to check whether a forecasting model based on machine learning could provide higher accuracy in predictions than those in use today. Secondly, it was interesting to analyze if the collected data had acceptable resolution and size to be used in machine learning. It was chosen to predict the earned value to answer these two questions.

3.5.1 Cleaning

When cleaning a dataset, one wants to keep as much data as possible yet remove data that either is dirty or of no value. Another unwanted form of feature is a feature that is too correlated with the label. By calculating the correlations between the label and the rest of the features, it was concluded that no feature had a correlation coefficient higher than 0.37 [179], which translates into a medium correlation. This is illustrated in Figure B.1. The highest correlation was the feature "main area general". Still, this correlation was only at the medium part of the spectrum, positively correlated with the label [180]. This is illustrated in Figure B.1.

Then, after some initial tests, the number of necessary columns was to be tested. The idea was that if the sum of the values in a column was below a specific number, it appeared too seldom for the model to learn its impact [177]. This idea is based on one of the fundamental lessons learned in machine learning; if the model can not get value out of

the feature, drop it. The number of features that remains is illustrated in Figure 3.7. As the sum limit increases, the remaining features decrease. Thus, the slope is negative since more and more features are excluded because the sum of the affected activities is small.

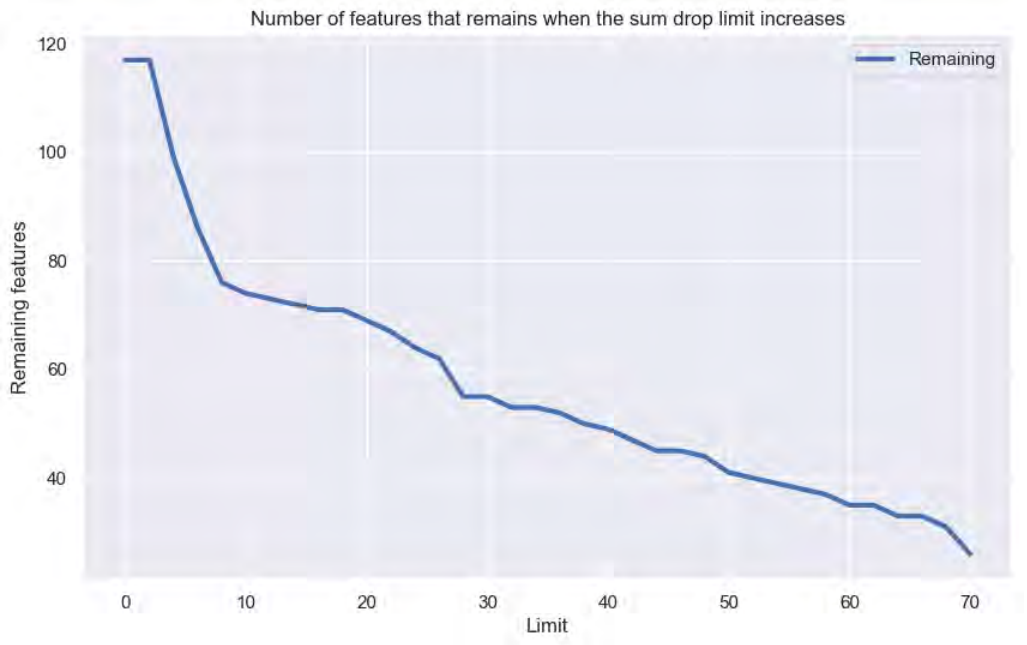


Figure 3.7: How the number of features decrease as a summation limit is set for the dataset.

3.5.2 Preprocessing

The label, which is the next week's EV, resulted from what was wanted by the company and what was accessible. Through the project management software, it was possible to extract the EV on a weekly basis. By going this route, the data collected through the SQL queries had to be aggregated weekly as well. Thus, the following logic was used. All activities starting in the same week were aggregated, and the values of their features were summed. The shape of the dataset was now (72, 110). The number of rows decreases since the data is aggregated. The number of features decreases since dates, strings, and other non-numeric data types will be dropped in this aggregation step.

3.5.3 Train and fit the model

The hyperparameter of "validation_split" sets the split of how much of the training data goes to validation and actual training, as mentioned in Section 3.3.4. It was set to 0.1. This model will utilize the ReLU as activation function which the same used for the model in Company A, see Figure 2.9. Another paramount setting when dealing with time-series

data is to disable all shuffle functions. If one were to forget this, the whole aspect of time would be lost.

The optimizer is an important hyperparameter because it is closely related to the type of data, like if there are large differences in extremities or high volatility. For optimizer, the preliminary analysis will test "ADAM", "NADAM", and "RMSprop". ADaptive Moment Estimation (ADAM) is an optimizer that is built on RMSprop, but with momentum in the form of an exponentially decaying average of gradients from the past [181]. NADAM is the same concept as ADAM except that the momentum is found before the the gradient computation. This yields the Nesterov accelerated gradient version of the ordinary momentum. In the search for the best optimizer, optimizers such as ADAM, NADAM, and RMSprop were tested. The optimizers were used with RMSE as the loss function because the model will later be optimized by using Equation 2.6. The function for RMSE had to be a custom-written Tensorflow loss function as this function is not yet implemented in the Keras library. Preliminary analysis showed that it was the NADAM that performed the best. Thus, it was the one that was utilized henceforth [182].

A method to choose the number of epochs appropriate for the model is through training and validation loss. The plot of these losses has the number of epochs on the horizontal axis and the amount of loss on the vertical axis. Finding the number of epochs that fit the data and model will enhance its performance. The number of epochs to choose is the point on the curve where the training and validation loss has the lowest difference, and the values are close to constant [183]. If the validation loss is of greater magnitude than the training loss, the model overfits. The other way around, and the model underfits.

Next, tuning of the "n_past" parameter had to be done. This parameter states how many days back into the past the model can look, the aforementioned time window. The parameter is essential to the model since it directly dictates the model's available information flow. From early analysis, something became apparent. With a high value, the model's ability to predict the trend is great, while the spikes are neglected. With a low value, it will only follow the spikes and disregard the trend. It may be because, with a too high value, the model will have difficulties predicting accurately since it will become harder to know which features are important at every time step. With a too low value, the model will not follow the trends and carry on the learned scaling of the features. Thus,

a value in the middle could give the model the necessary information without being too generic or too specific. A grid search was utilized with 3, 6, and 9 as values for n_past. The search showed that the values of n_past affect the smoothness of the forecast curves.

The method of sliding window is illustrated in Figure 3.8 with n_past of 12. Here, one can see that the time series data is split up into 47 patches, which equals the number of sliding windows. The last time step the model sees is 58, which is the penultimate. The model is to train on the final value as well, i.e. the EV on time step 59. The model will thus make 47 predictions over this timeline. Then, the training is over, and the testing begins. Illustrated by the blue line, the test section is from week 59 to 72. There is a parameter that is called "n_future". It may be altered from the value of one to predict multiple weeks into the future, for example, three weeks. The user can adjust n_future to predict as far into the future as there is interest.

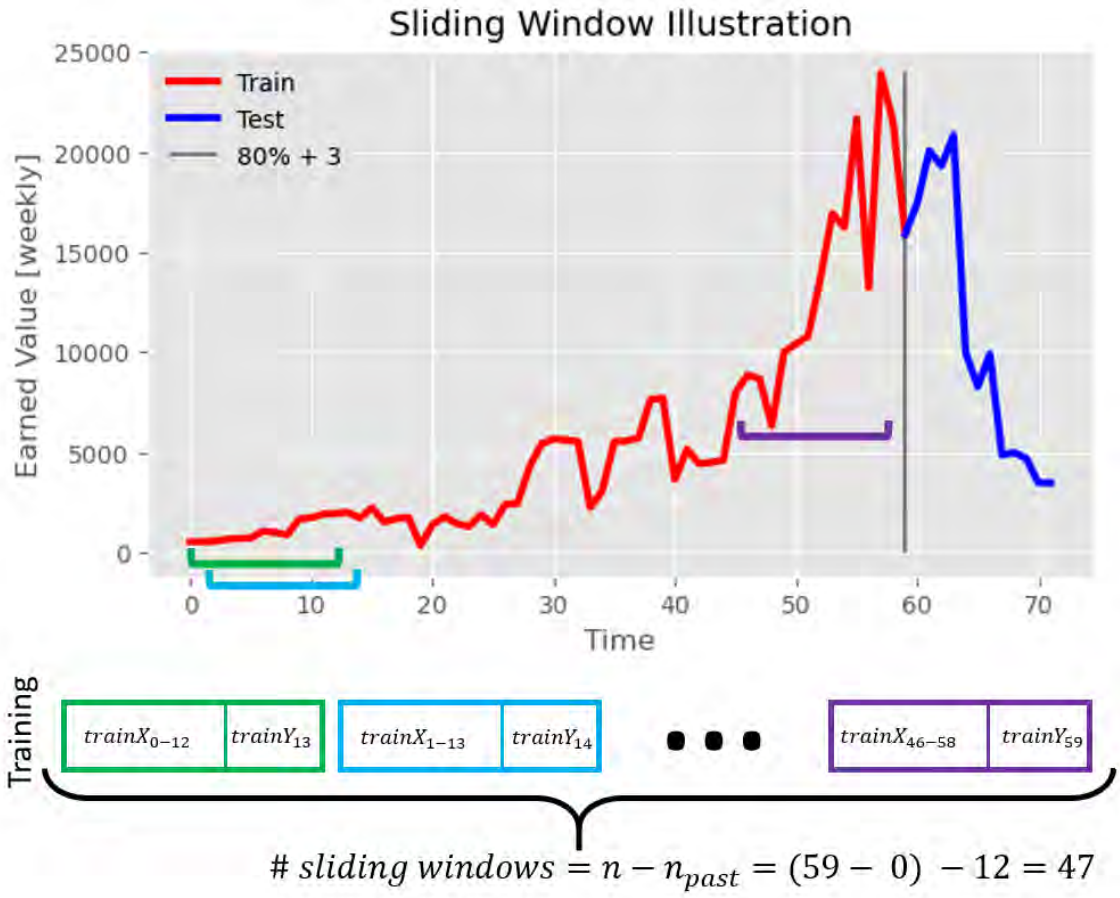


Figure 3.8: The figure present how the windows slides through the data.

To perform a grid search on this dataset, the first action was to reset the index of the dataframe. This was to ensure the LSTM model was set up correctly. Having learned from

the time series analysis of Company A, similar techniques were implemented. NADAM were utilized as optimizer. Opposed to Company A, a split had to be implemented to differentiate the training set from the testing set. As can be seen in Figure 3.9, there is a high peak in the weekly EV at the end of the project. For the model to learn how to forecast this peak, it must be included in the training. The split is commonly set as 80 percent for training and 20 for testing, as presented in Section 2.3.4. Figure 3.9 also shows the importance of the split and is implemented with an n_past of 3. Similar figures with n_past of 6 and 9 is shown in Figures B.11a and B.11b, respectively. If the split is like in Figure B.11b and n_past is 9, the model will not have data left to test on. However, if the split is like in Figure 3.9 and n_past is 3, the model might not have learned how to adjust for the peak. Therefore, splits of 0.73 and 0.83 were included in the grid search to have an early and a late split. For the split of 0.83, only n_past of 3 were considered to be interesting.

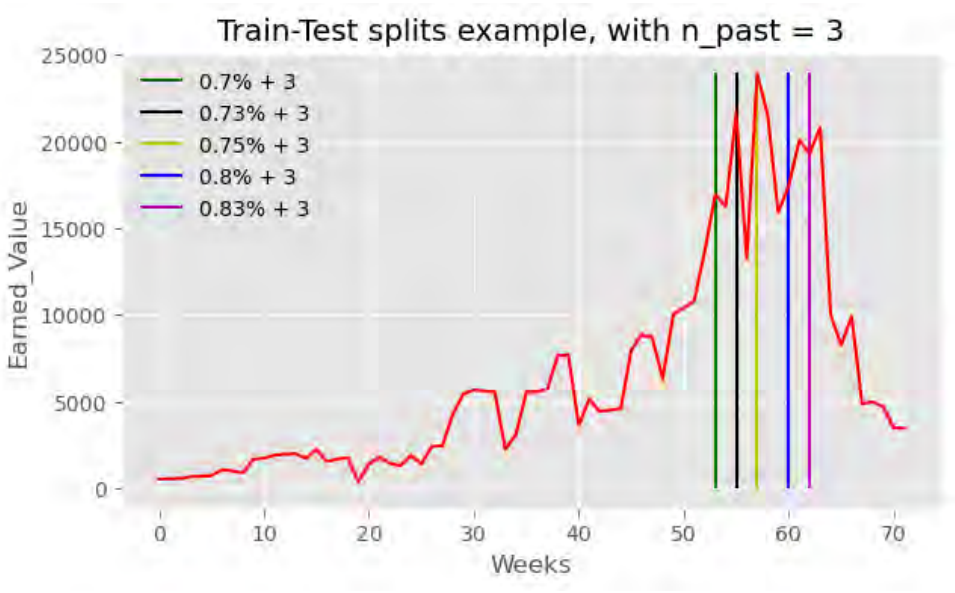


Figure 3.9: Train-test split with n_past of 3.

Next, the data can to be scaled. Scaling increases the model's ability to compare and thus predict values. Different scalers were tested in a grid search. The three most commonly used scalers in the SKLearn library, i.e. the StandardScaler, the RobustScaler, and the MinMaxScaler, were tested, as per Section 2.3.9. An important distinction between these scalers is that Robust and MinMax perform normalization where Standard performs standardization. Thus, Standard scales the features into a standard distribution, one feature at a time.

The number of batches is a hyperparameter of the model that determines the amount of data included in each training interval [184]. The number of batches was included in the grid search to find the values that fit the model best. It included 8, 16, and 32 as values for the batch size. The search also included different methods to drop features. As shown in Figure 3.7, the number of features decreases drastically with a summation limit of around 10. Additionally, Table B.1 shows that there is a high number of features with a summation limit of 0, 1, or 2. These were considered to be too infrequent to be included. Therefore, it was decided that a summation limit of 5 would drop these features and still maintain valuable information. Thus, this is the value that was utilized henceforth.

After the cleaning, the shape of the dataset was (72, 110). Because the number of columns is higher than the rows, a method to further reduce the number of features should be implemented. After interviews with a machine learning specialist at the institute, as a rule of thumb, the number of input features should not exceed 40. Therefore, the grid search included a hyperparameter that chooses which feature sets to use as input. Based on the EDA and preprocessing, there are three main categories for the input features. The first, "Numbers", includes characteristic numbers of each activity, such as scope, duration, expended quantity, and sums up to six features. The following, "Description", includes information manually put in as words by employees of Company B. For instance, it can be "design", "cost", "engineering", or "management". After preprocessing, the number of features from descriptions is 28. The third, "Codes", is categorical information for each activity. It includes discipline, scope type, phase, activity role, area, site code, category, and sums to 78 features. The grid search includes these three classes for the number of input features, and the fourth is to include all features. A summary of the values for the hyperparameters included in the grid search is shown in Table 3.7. The search will run through all values of n_past for the split of 0.83, but only n_past of 3 is considered as interesting as mentioned above. The values in the table sum to 216 different combinations of the hyperparameters.

Table 3.7: The values for the hyperparameters to be tested in the grid search.

Hyperparameter	First value	Second value	Third value	Fourth value
n_past	3	6	9	-
Train-test split	0.73	0.83	-	-
# of batches	8	16	32	-
Feature set	Codes	Number	Description	All
Scaler	Standard	MinMax	Robust	-

Another important hyperparameter is the number of layers in the LSTM model, as it will determine the depth of the model. It was wanted an efficient model. Therefore, it was determined that the model implement two layers. Another hyperparameter is the number of nodes in each layer. One may think of these nodes as dials the model twist as it learns throughout all the batches and epochs. In this model, the second layer will have half the nodes of the first one. After the LSTM layers, a dropout function is put in. This is to combat overfitting and make the model more general. The dropout value was set to 0.2.

Monte Carlo Simulations (MCS) was utilized to determine which hyperparameters were the best. Simulations had to be performed because of the inherent nature of sequential machine learning models such as LSTM. For preliminary analysis, a small number of simulations were used. This was to get quick answers on which parameters provided the best results. Then the mean from all these predictions was found and plotted against the original. One mistake made in the beginning was that the model was initialized outside of the MCS loop. By doing this, the model remained the same but was called to repeat on the same dataset. It resulted in the same overfitting that was witnessed with a very high epoch number. Thus, the model's initialization was moved outside, and so was the other necessary parameters. In the end, the aforementioned hyperparameters were found and set. However, there was still a debate on which scaler that was the best; see Section 2.3.9. Lastly, the MinMaxSacler were implemented with 500 simulations in the MCS with the top-performing configurations.

3.5.4 Summary of the complete model

After cleaning and preprocessing, the shape of the dataset was (72, 110). Using the LSTM-algorithms, the model will predict the next week's EV from week 58 or 62 to 71 by using from three to nine previous weeks as input, which is the values for n_past.

It will train on 73% and 83% of the data and test on 27% and 17%, respectively. The number of batches to be tested were 8, 16, and 32 and a summation limit of 5 was used. Additionally, the grid search included different combinations of input features and scalers. Ultimately, the model ran through 10 MCS in the preliminary stage to obtain weak means of the values to include in the final configuration. Then, with 500 MCS with the final configuration to achieve higher accuracy of the mean performance of the model.

3.6 B2 - Classification Analysis

The target label in this classification will be whether each activity is a start hit or miss. Identifying start hits is necessary information to find the BEI score as presented in Equation 2.3. Based on the fact that the dataset from Company B was more detailed than the ones from Company A, it was wanted to perform a secondary analysis. Classification analysis differs from regression analysis as it aims to predict a class rather than a numeric value, as mentioned in Section 2.3.7.

3.6.1 Exploratory data analysis

It was of interest to utilize the same features as in the time series analysis on Company B (B1) to investigate if classification could indicate which features were the most important to determine a hit or miss. Thus, the dataframe with shape of 13 047 instances by 38 features from the B1 analysis was used. However, some of the features needed to calculate the label were already dropped from this dataframe. Therefore, two dataframes were made, one from the B1 analysis and another one that contained additional features to calculate the label. Afterwards, these dataframes could be merged on activity level, and a one-hot encoded feature named "start_hit" could be created.

3.6.2 Preprocessing

Because the dataframe from B1 was cleaned, it proved as a fitting stepping stone in this analysis and only minor preprocessing was needed. As shown in Section 2.2.4, a BEI score can be calculated for both the start and finish of an activity. This analysis will show what is possible with the type of data, and there was limited time to investigate multiple aspects. Therefore, the analysis focuses on the start of activities. Interviews with representatives from Company B stated this to be important for these types of projects.

To calculate the BEI, new features to indicate start hit must be established. As presented in Equation 2.3, if the actual start date was before or during the planned week, it was counted as a hit. The start hit was calculated for each activity to maintain the highest possible resolution of the model.

$$start_{hit} \coloneqq current\ \ actual\ \ start \leq Early\ \ start\ \ analyzed \tag{3.1}$$

To obtain accurate hits for the starts, the two variables from Equation 3.1 and have to be correct and static. One of them was, but changes had to be made to the other. As mentioned in Section 3.4.1, the date of "current_as" could have changes in some subsequent rows within the same activity. The new date was set as the value of "per_start", which is the weekly period the change was done. For ESA, the new date could be set to a date before the project started. A cumulative week counter was made based on "per_start" to track the global cumulative week number. Another feature for weekly counters was generated and based on ESA to relate the date to the corresponding week number. These were made to make the plotting and comparisons between the actual and predictions easier. After interviews with Company B, it was adequate to have a resolution on a weekly basis instead of the actual date because it was considered good enough if the activity started in the same week as planned. Then, the features for start hit could be created. Afterwards, other auxiliary features could be created to be used to calculate the BEI score. These include the feature to keeping track of the cumulative sum of the hits and a feature keeping track of the sequence of the activities. From the BEI Equation 2.3 in Section 2.2.4, one can now input these new features like illustrated in Equation 3.2.

$$BEI_{Start\ i} = \frac{Start_{hit}\ \ sum_i}{Start\ \ sequence\ \ counter_i} \tag{3.2}$$

3.6.3 Train and fit the model

The label to compare against and strive for is the BEI start score. In classification, the label is the measurement for how the different classes are classified. The label fits Company B's data because there is has a high number of activities that must be done to meet the planned progress. Ultimately, this label will partly tell the performance of the project.

In the first search for hyperparameters, the RandomSearch method was utilized. This is a non-exhaustive algorithm that is quicker than a grid search. By using this method, one can more quickly zoom in on the best range for each hyperparameter. The search was performed in a nested fashion where the RandomSearch received different dataframes for each loop. In this outer loop, another grid search determined the scaler and the summation limit of the dataframe. See Figure B.10 for a visual interpretation.

A grid search was utilized to find the best hyperparameters. For this analysis, a nested grid search was used in two rounds. Additionally, it was done a preliminary analysis on which machine learning classifier to utilize. Parallel to this, an alternative method to find the best machine learning algorithm to conduct this classification analysis are to take advantage of the novel method of AutoML. As presented in Section 2.3.3, AutoML has it strengths and weaknesses [83]. Nevertheless, the method was tried out in the dataset. The two main classes of classifiers tested was the MLPClassifer, which is a neural network classifier, and different ensemble classifiers, such as RandomForest (RF) and Extreme Gradient Boosting (XGB). The most extensive grid search was the one regarding the RF. For the RandomForest classifier, hyperparameters illustrated in Section B.3 were tested. Similar grid searches were performed for the other classifiers. Then, a more precise grid search could be performed. The highest scoring hyperparameters that were found were extracted for each feature set. Then, a more thorough search could begin around the vicinity of the best hyperparameters. After this search, even higher scoring hyperparameters could be found. Several tests were run, and the results were prominent and saved for deeper analysis.

Preliminary analysis showed that if the model shuffles the activities to include in the train set, the output plot can have a more diversified set for training. The advantage of the shuffle is that the model will randomly select which activities to include in the training set. Additionally, through running several simulations, each activity could be included in the training set multiple times. Thus, the idea was to run a high number of MCS to take advantage of the law of large numbers [185]. Then, the randomly selected test activities could be saved in an outer dataframe. These would be saved at their respective position. From the outer dataframe, the mean of all these iterations could be found, disregarding the Na values. Lastly, the mean would be binarized, i.e. rounded to 0 or 1, yielding the

rounded mean prediction per activity based on a high number of MCS. This method is illustrated in Figure 3.10.

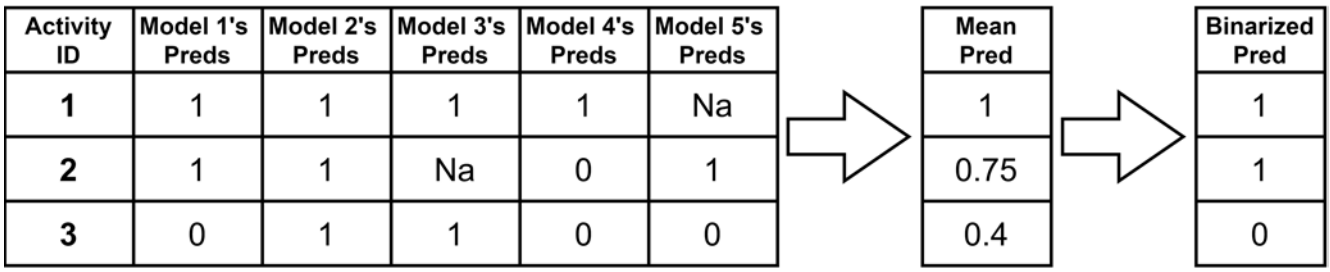


Figure 3.10: Illustration of the Monte Carlo simulation, mean, and binarized method.

In the end, eight models persisted. Two RF and two MLP persisted from the grid search, and three XGB and one RF persisted from AutoML. After a quick MCS with 100 simulations, the best two from each classifier class advanced to the following MCS tests. In the first test, Test 1, the number of MCS were set to 100. The number of 100 was considered low enough to obtain preliminary results while detailed enough to showcase the differences between the models. In Test 2, the number of MCS was increased to 5000. This step yielded six models. To be sure that the models performed consistently, Test 3 was conducted to gauge the quality of the models. As presented in Section 2.3.8, SHAP was introduced on the trained models to capture the internal weighting of the model. This score for feature importance will prove as a visualization of the analysis to understand what goes on inside the machine learning algorithm. Traditionally, machine learning models are viewed as black-box methods [120].

3.6.4 Summary of the complete model

To predict whether or not an upcoming activity will be a start hit or miss could be of great importance to meet the plan and get a metric on the project's progress. After cleaning and preprocessing, the shape of the dataset was 627 activities by 110 features. The model will predict whether an activity will hit or miss its start date by utilizing the multiple classification algorithms. It will train on 71% of the data and test on 29%. The summation limit was set to be 3. Ultimately, the model ran through 100 MCS in the preliminary stage. This was to obtain preliminary results. Then, tests with 5000 MCS were conducted. This final configuration gave high accuracy of the mean performance of the models.

4 Results

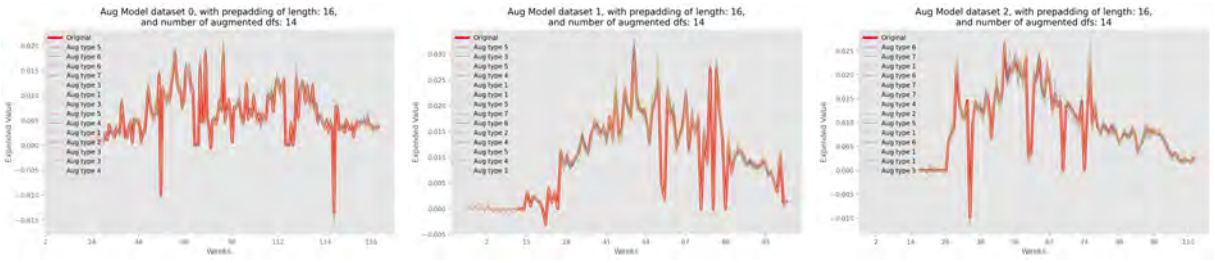
Because the analysis is done on different datasets, the results are split into three parts. The first and second part are the results from the time series analysis on the data from Company A and B. The last part presents the results from the classification analysis on the project from Company B.

4.1 Results from the Time Series Analysis on Company A

The deployed method for time series analysis on Company A provided several results in predicting the expended value of the projects. The process produced results to describe the underlying data, how the hyperparameters were chosen, and the final forecasting after the model had trained.

4.1.1 Initial results

As presented in Section 3.3.3, the data was augmented to provide the model with more datasets. The final method for augmentation was to pre-pad the dataset, use the AddNoise function, and the probability for how often it is activated. The model's label, which is the incremental step in expended value, has numbers of small magnitude. Because of this, the augmented data will not deviate of great magnitude from the original. This can be seen in Figure 4.1 where the stacked augmentations lie within an acceptable range from the original. The figure also shows that the augmentations have small deviations from the original sets. Therefore, it can be argued that the augmentations are well suited to add robustness to the model, so the model will be less likely to overfit. The augmentation plots for the rest of the features in each project can be found in the Section A.1. They present similar characteristics as in Figure 4.1. Another small finding was that if the batch size was small, the robust scaler performed well. If, however, it was large, the robust scaler often predicted the opposite of the actual. Lastly, in the figure, some strange properties of the expended value can be seen. It is negative at week 20. During the interview with Company A, it was uncovered that these areas must be a result of rework. They state that this value should never be negative in theory, but sometimes something may have gone wrong and has to be fixed.



(a) The augmentation on expended value on dataset 0.

(b) The augmentation on expended value on dataset 1.

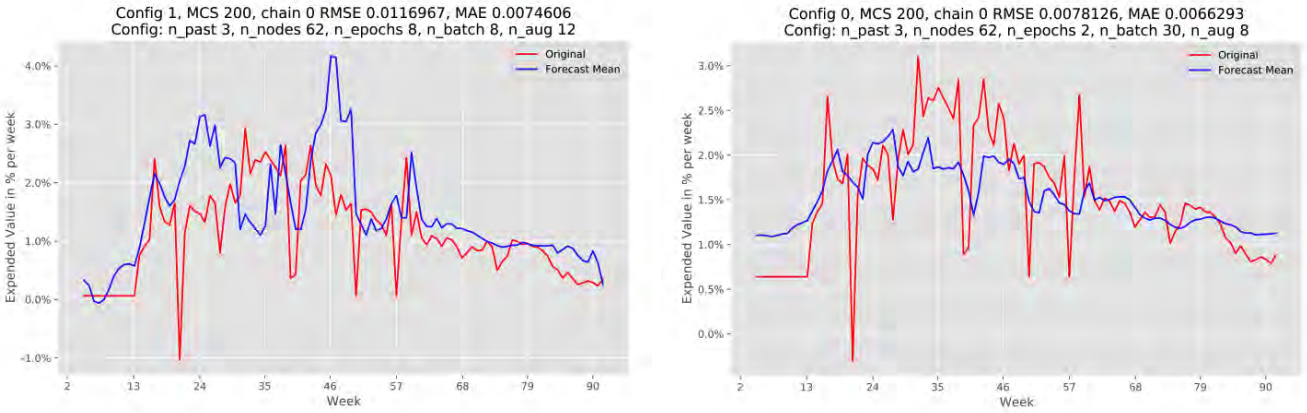
(c) The augmentation on expended value on dataset 2.

Figure 4.1: The augmentations of the projects from Company A.

4.1.2 The configuration of the hyperparameters

As Section 3.3.4 presents, it is vital to find the most appropriate scaler to fit the nature of the data. Therefore, three scalers, Standard, Robust, and MinMax were tested. As presented in Section 2.3.9, the RobustScaler and MinMaxScaler perform normalization, whereas StandardScaler performs standardization. Figure 4.2 shows a comparison between RobustScaler and MinMaxScaler on the first chain in which the vertical axis shows expended value in percent per week and the horizontal axis shows the week number. The most important aspects of the figure are the difference in forecasting the spikes. In Figure 4.2a, the RobustScaler looks to weigh the spikes more than it should. This can be seen in week 24 and 46, in which there are two large spikes in the forecast that are slightly delayed from the smaller spikes in the original. However, the shape of the forecast looks to fit the original as its values fluctuate as frequently.

At the worst performance in Figure 4.2a, at week 46, the forecast is too large, resulting in a deviation of two percentage points. From week 50 and onward, the forecast looks to perform better. The MinMaxScaler, in Figure 4.2b, does not forecast large enough values, something the RobustScaler looks to be more fit to do. When compared to the RobustScaler, the forecast in Figure 4.2b looks to follow the trend in the original more tightly. However, the MinMaxScaler performs the worst in the interval from week 24 to 46. In this interval, the shape of the forecast curve is similar to the shape of the original, but the values are off by approximately one percentage point. Additionally, in Figure 4.2b, the initial value of the forecast is too high, approximately a half percentage point higher compared to the original. However, the RMSE score of MinMaxScaler is lower than the one of RobustScaler.



(a) A plot of the results with RobustScaler.

(b) A plot of the results with MinMaxScaler.

Figure 4.2: Plots from the first chain to compare the best normalization scaler to the standardization scaler.

Figure 4.3 shows the plot of the forecast with the first chain when the model is implemented with StandardScaler. The performance is not satisfactory. From week 24 to 30, the forecast is from a half to one percentage point above the original curve. From week 30 to 50, the forecast has large deviations from the original curve. However, the two valleys at week 40 and 50 look to be predicted by the model. By looking at Figures 4.2 and 4.3, it is evident that the model implemented with StandardScaler has the lowest RMSE score. Based on results from the first chain, the StandardScaler looks to be performing the best.

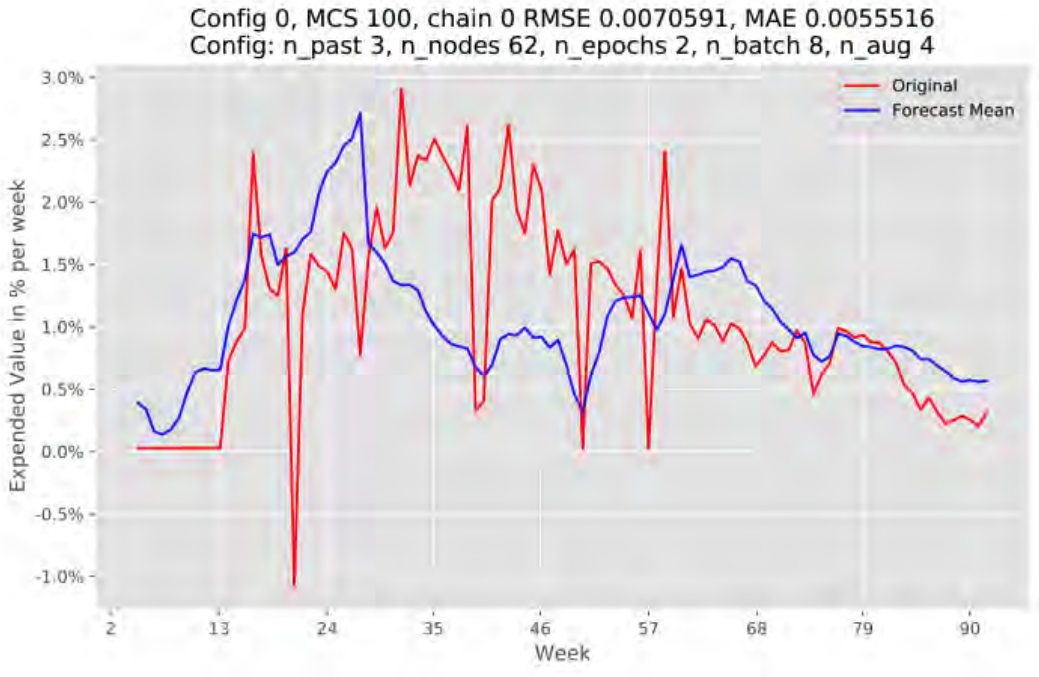


Figure 4.3: Plot of the results from the first chain with StandardScaler.

As mentioned in Section 3.3.4, the configuration of the hyperparameters was tested through a grid search with a total of 32 configurations. Moreover, as Section 3.3.4 presented, each configuration is only run on the second chain. The results from the top-performing configurations of the grid search are shown in Table 4.1. The results are sorted on the error term of RMSE, in which a lower score is better. From the table, it is evident that the error terms of MAE are not sorted. Thus, if the outliers are not the most important deviations, the ranks of configuration would be different. It is interesting to see that Table 4.1 only shows StandardScaler. This supports the results from the first chain, in which StandardScaler also provided the lowest RMSE score. As shown in Table 4.1, some values of the hyperparameters vary between the configurations while others are constant. The number of epochs should be 2, that n_past should be 3, and the most fitting scaler is StandardScaler. Regarding the number of nodes, there is not enough data to tell if 62 is better than 102. The same yields for the number of batches, which is either 8 or 30. The number of augments is varying among the top ten list. However, since a number of 4 augments are most apparent in the top, it is considered to be the best configuration. Because the top-performing configurations have variations in the number of nodes and batches, the top is considered as the best performing with values of 62 and 8, respectively.

Table 4.1: Table of the ten best performing configurations of hyperparameters on the second chain.

Rank	Nodes	Epochs	Batches	n_past	Scaler	n_aug	RMSE	MAE
1	62	2	8	3	Standard	4	0,00444	0.00303
2	62	2	30	3	Standard	8	0.00448	0.00308
3	102	2	8	3	Standard	4	0.00448	0.00306
4	102	2	30	3	Standard	4	0.00453	0.00321
5	102	2	30	3	Standard	8	0.00456	0.00315
6	62	2	30	3	Standard	4	0,00466	0.00334
7	62	2	30	3	Standard	12	0.00477	0.00330
8	62	2	8	3	Standard	8	0.00490	0.00339
9	102	2	8	3	Standard	8	0.00505	0.00353
10	102	2	30	3	Standard	12	0.00513	0.00354

As shown in Table 4.1, there is absence of RobustScaler and MinMaxScaler. The best performing configuration with RobustScaler had an RMSE score of 0.00626, which is large compared to the difference between the configurations of rank one and ten. The best performing configuration with MinMaxScaler had an RMSE score of 0.00696. Therefore, it

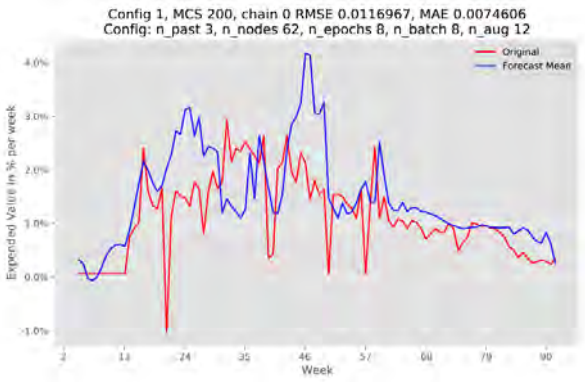
is assumed that StandardScaler outperforms the two others. Regarding n_past of values larger than 3, the best performing configuration had an RMSE score of 0.00528, which is at rank 12. The top-performing configuration with a number of epochs larger than 2 is at rank 19. Thus, n_past of 3 and the number of epochs to be 2 looks to be the best values. Table 4.2 summarizes the values for the hyperparameters that are proved to provide the most accurate model.

Table 4.2: The table shows the best configuration of the hyperparameters.

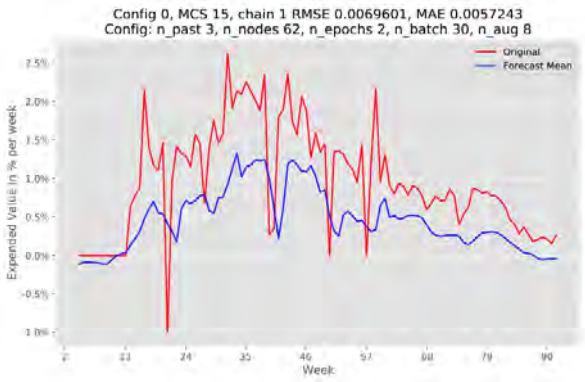
Nodes	62
Epochs	2
Batches	8
n_past	3
Scaler	Standard
n_aug	4

4.1.3 Results with final configuration

As mentioned in Section 3.3.4, the model was run in different chains. The first chain used Project 1 to train on and tested on the third, and the second chain trained on Project 1 and 2 and then tested on the third, see Figure 3.5. The training is independent in each chain. Therefore, the results from the two chains will make it able to show the importance of more data to train on, i.e. that the model is generalizing more than overfitting, and thus, is more robust to handle unseen data. The results also proved that the type of scaler affected the results more than initially assumed. Figure 4.4 shows the plots from the first and second chain when run with RobustScaler. The model performs better when training on two projects and testing on a third than only training on one. The RMSE score of Figure 4.4b is substantially lower than of Figure 4.4a, approximately 0.4 percentage points. More importantly, the shape of the forecast curve in Figure 4.4b is more similar to the original. The forecast curve has much of the same fluctuations, such as peaks and valleys. Even if the original and forecast shape are similar, the latter looks to be a half percentage point wrong throughout the project.



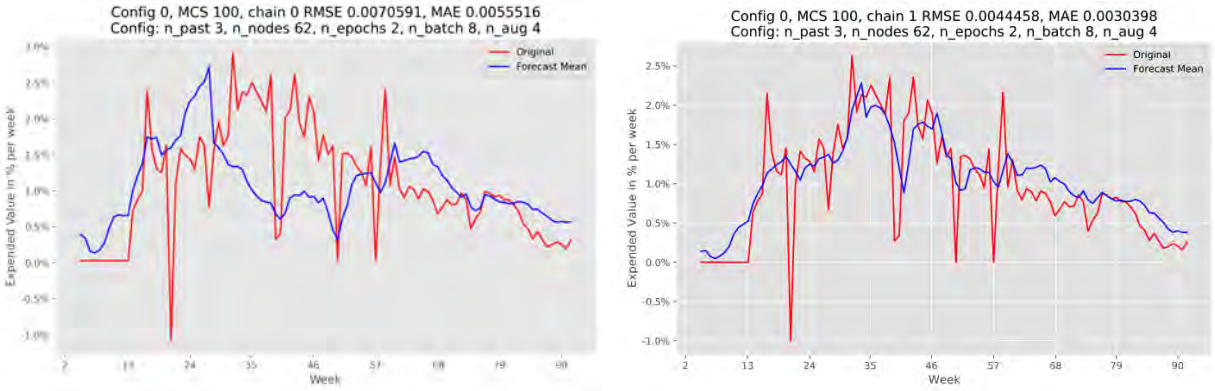
(a) The plot from training on one project in the first chain with RobustScaler.



(b) The plot from training on two projects in the second chain with RobustScaler.

Figure 4.4: The two plots show the different performance from training on one and two projects with the same configuration.

Figure 4.5 shows the results from the model with identical configuration run on the first and second chain. Likewise, for the RobustScaler, the implemented model with StandardScaler performs better on the second chain than the first. Figure 4.5a has low performance in the interval from week 20 to week 57, except in the two negative spikes at week 40 and 50. In this interval, it looks like the model predicts a too high value at the start and then compensates by forecasting low values. Except for the negative spikes, the forecast often deviates by over one percent in the interval. However, after week 57, the forecasts are acceptable, with the largest deviation of approximately a half percent. The forecast values in Figure 4.5b tracks the original data more closely as it manages to follow the trend and the small local fluctuations. As can be seen, the forecast curve follows the shape of the original curve tightly. However, it does not forecast the magnitude of the change in certain spikes, specifically those at weeks 15, 20, 50, and 57. The forecast looks to predict a certain change in value in these spikes, but not how high or low. Still, the RMSE of the second chain is about 37% lower than the RMSE of the first chain's prediction. Figures 4.4 and 4.5 supports that the choice of scaler is important. From Figures 4.4a and 4.5a, it is evident that StandardScaler achieves higher accuracy than RobustScaler on the first chain in terms of RMSE score. Based on Table 4.1 and Figures 4.4b and 4.5b, StandardScaler seems to be the better choice on the second chain as well.



(a) The plot from training on one project in the first chain with StandardScaler.

(b) The plot from training on two projects in the second chain with StandardScaler.

Figure 4.5: The two plots show the different performance from training on one and two projects with the same configuration.

As shown in Figure 4.6, the performance of StandardScaler on the second chain is good. The forecast curve is following the original curve but with a delay at certain intervals. For instance, for the highest peak in the data, approximately at week 30, the corresponding peak in the forecast is a couple of weeks later. This behaviour is repeating itself throughout the project, especially around the spikes. It indicates that the model must see past values of the label to predict values of similar magnitude, and thus, it is not an adequate prediction. The indications of good performance is by the increase in value at week 13. At this point, the forecast is predicting an increase before the original have increased. The same behaviour applies to the decreasing values from week 35. Throughout the plot, the forecast curve is smoother than the original. Therefore, the original data tend to fluctuate above and below the forecast curve. The RMSE score of 0.0044 means that the model is on average 0.44 percentage points wrong every week. From Figure 4.6, it looks like average production is fluctuating between one percent and one and a half percent. Thus, an error of 0.44 percentage points could turn an average to low production into an average to high.

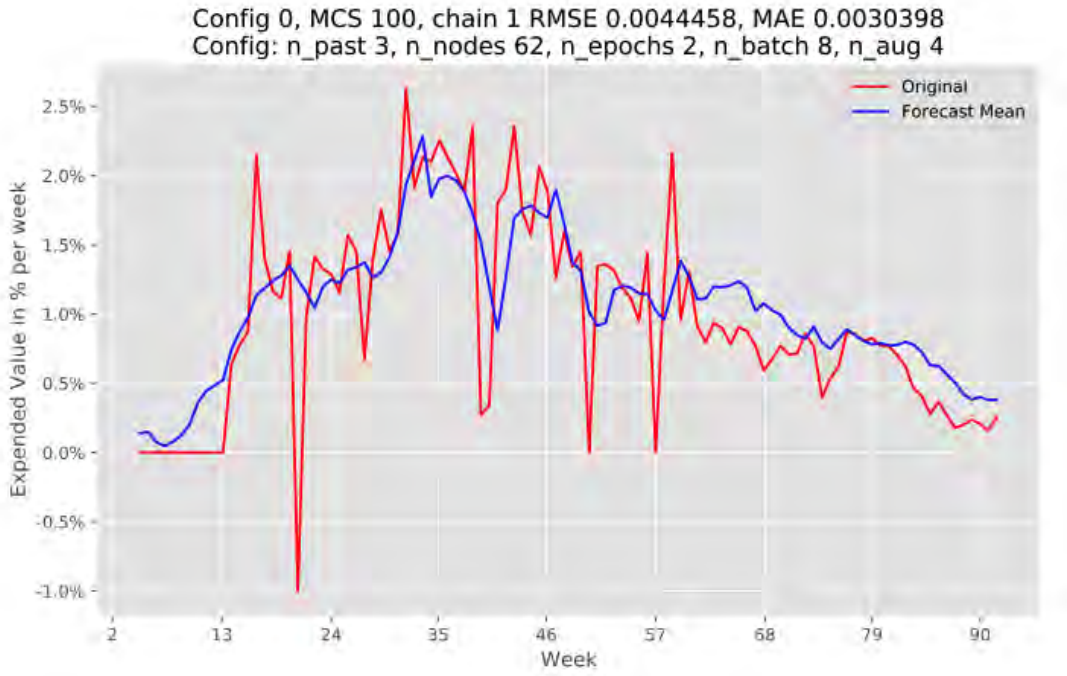


Figure 4.6: Plot of the results from the second chain with StandardScaler.

4.2 Results from the Time Series Analysis on Company B

The deployed method for time series analysis on Company B provided several results in predicting EV. The process produced results to describe the underlying data, how the hyperparameters were chosen, and the final forecasting after the model had trained. As shown in Table 3.7, the grid search included n_past, how the train-test split was set, scaler, which input parameters to include, and the batch size, which sums to 216 different combinations. The values of the hyperparameters, n_past and the split, provides results of different characteristics. Therefore, the results will be categorized based on n_past and the split.

4.2.1 Initial results

An early result from the model is how the activities are distributed throughout the project. Figure 4.7 shows the distribution through a bar plot. This figure provides information on whether the model has a chance to perform well. For instance, the model might not have enough instances to learn from if only a few weeks have starting activities. As expected, there are more starts early in the project than late. However, the distribution shows a high number of weeks that have starting activities. This indicates that the model has a

good variety in the data to train on. The histogram also shows that after 200 weeks, there are only two short time intervals that have starting activities. More specifically, there are only starting activities around week 215 and 275. Thus, there are long time intervals with no starting activities. As shown in Figure 3.8, there is a peak in the EV at the end of the project. The model might find it difficult to forecast accurately because of the few activities at the end of the project.

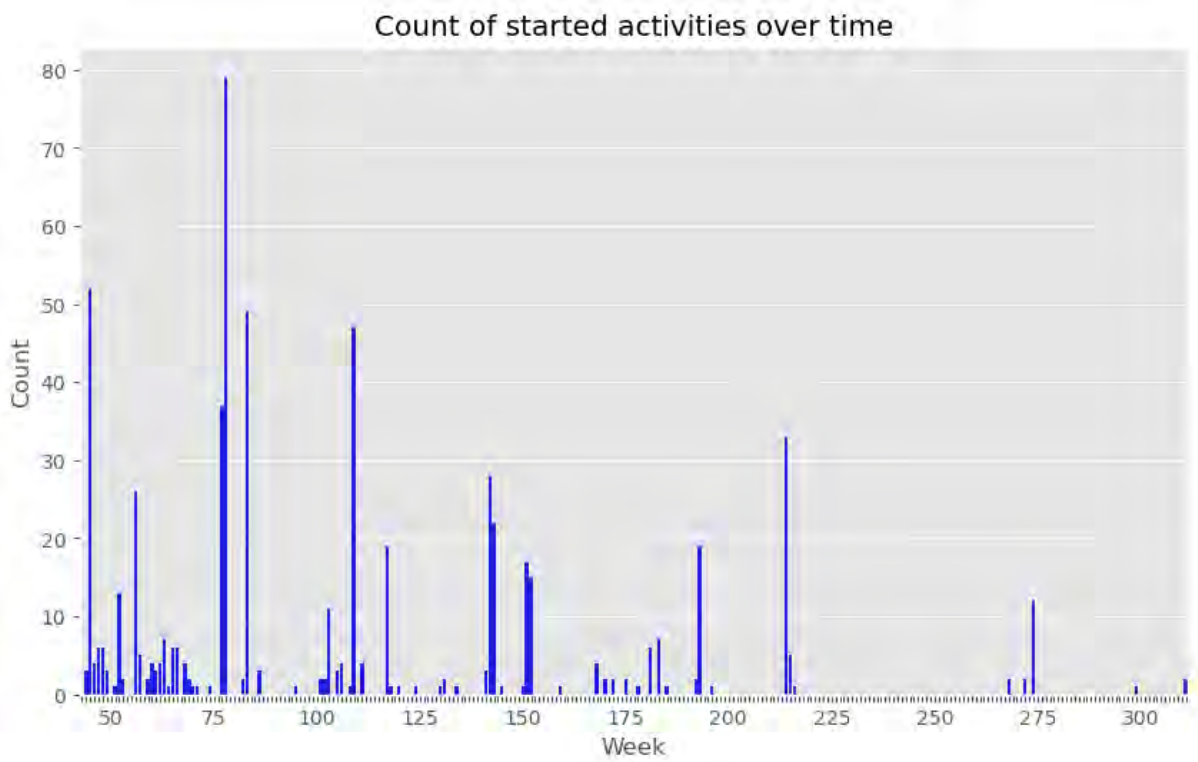
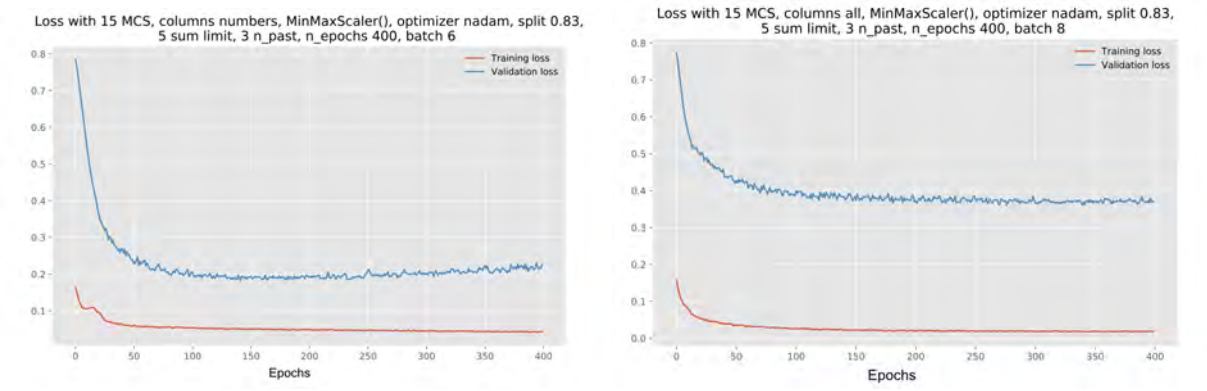


Figure 4.7: When the activities starts. Before drop of empty weeks and the reset of week number.

4.2.2 The configuration of the hyperparameters

As mentioned in Section 3.5.3, multiple tries were made to obtain the best configuration of the hyperparameters. The method to choose the number of epochs was done by analyzing the training and validation loss. The plots in Figure 4.8 shows the results from the test, in which the number of epochs is on the horizontal axis, and the amount of loss is on the vertical axis. As mentioned in Section 3.5.3, the test included a varying number of input features because, in machine learning models, the number should generally not exceed 40. The most important aspect of the plots is that the curves and magnitude of the vertical axis are alike. Figure 4.8a shows that the validation loss is slightly above the training loss

with a difference of approximately 0.15. Figure 4.8b shows the same, but with a difference of approximately 0.4. The higher difference in Figure 4.8b might indicate that the model with all of the input features performs worse than the model with fewer input features. Thus, the number of input features might affect the performance of the model. Based on the plots, the number of epochs was set to be 100. Usually, validation loss higher than training loss represents some overfitting. Like in this case, the validation loss is twice as much as the training loss. However, since there is only one dataset, it is not easy to not have the model overfitting. Further, there is the degree of overfitting. If the validation loss is tenfold higher than the training loss, other options must be investigated. In this pilot, the results were thought to be satisfactory for a proof of concept model.



(a) Number feature set included. **(b)** All feature set included.

Figure 4.8: Training and validation loss results.

To provide more insight into which parameters performs the best, a grid search was deployed. See Section 3.5.3. The search provided valuable insights into which values these hyperparameters proved the best. Table 4.3 shows the results from the grid search with a train and test split of 0.73 and n_past as 3. The table includes the top ten performing configurations and is sorted on the error term RMSE. Interestingly, all possible feature sets for the input are in the table. This indicates that all the information in the dataset can be used as input features in the model. However, the inclusion of all input features results in the best model, as is the case for the top three performing models. Regarding the scaler, the MinMaxScaler provides the lowest error score in terms of RMSE and MAE. StandardScaler only appears twice in the top ten list, and thus, is considered as outperformed by MinMaxScaler. Regarding the number of batches, there looks to be no pattern of which perform the best. Therefore, the top-performing is kept for further

analysis. Based on Table 4.3, the configuration to run through a higher number of MCS is to include all input features, MinMaxScaler, and batch size of 16. By comparing Table 4.3 to Figures 4.8a and 4.8b, it is possible to see the lower training and validation loss do not yield the lowest RMSE score. The configuration with all input features is the one to perform the best for MinMaxScaler and is the one with the highest difference between training and validation loss.

Table 4.3: Table of the ten best performing configurations of hyperparameters with a split of 0.73 and n_past of 3.

Rank	Input	Scaler	n_past	Batch	RMSE	MAE
1	All	MinMax	3	16	7983	6908
2	All	MinMax	3	8	8101	7036
3	All	MinMax	3	32	8123	6982
4	Codes	Standard	3	16	8146	6512
5	Numbers	MinMax	3	16	8280	6198
6	Codes	Standard	3	32	8331	6650
7	Description	MinMax	3	16	8347	6236
8	Numbers	MinMax	3	32	8484	6385
9	Numbers	MinMax	3	8	8590	6321
10	Description	MinMax	3	32	8645	6473

Table 4.4 shows the results from the grid search with a train and test split of 0.73 and n_past as 6. The table includes the top ten performing configurations and is sorted on the error term of RMSE. Opposed to Table 4.3, the results in Table 4.4 shows that the input features regarding numbers are not on the top ten list. However, there are close similarities between the tables. Both have all input features as the top three and varying sets of input features for the rest. The feature set of all input features should be kept for further analysis. For the scaler, the MinMaxScaler provides the lowest error score of both RMSE and MAE. Further, the results show that it is the same configuration that has the lowest error score of both RMSE and MAE. StandardScaler only appears once among the top ten, and thus, is considered as outperformed by MinMaxScaler. Regarding the number of batches, there looks to be no pattern of which performs the best. Therefore, the top-performing is kept for further analysis. Based on Table 4.4, the configurations to run through a higher number of MCS is to include all input features, MinMaxScaler, and a batch size of 8. Interestingly, the values of the error scores are lower in Table 4.4 than in Table 4.3. Table 4.3 includes more of the peak in the data, which might give distortions

for the forecasting.

Table 4.4: Table of the ten best performing configurations of hyperparameters with a split of 0.73 and n_past of 6.

Rank	Input	Scaler	n_past	Batch	RMSE	MAE
1	All	MinMax	6	8	6131	5401
2	All	MinMax	6	32	6455	5455
3	All	MinMax	6	16	6582	5984
4	Codes	MinMax	6	16	6872	5821
5	Description	MinMax	6	16	6920	5719
6	Description	MinMax	6	32	7050	5850
7	Codes	MinMax	6	32	7177	6105
8	Description	MinMax	6	8	7255	5937
9	Codes	Standard	6	16	7988	6163
10	Codes	MinMax	6	8	8106	6903

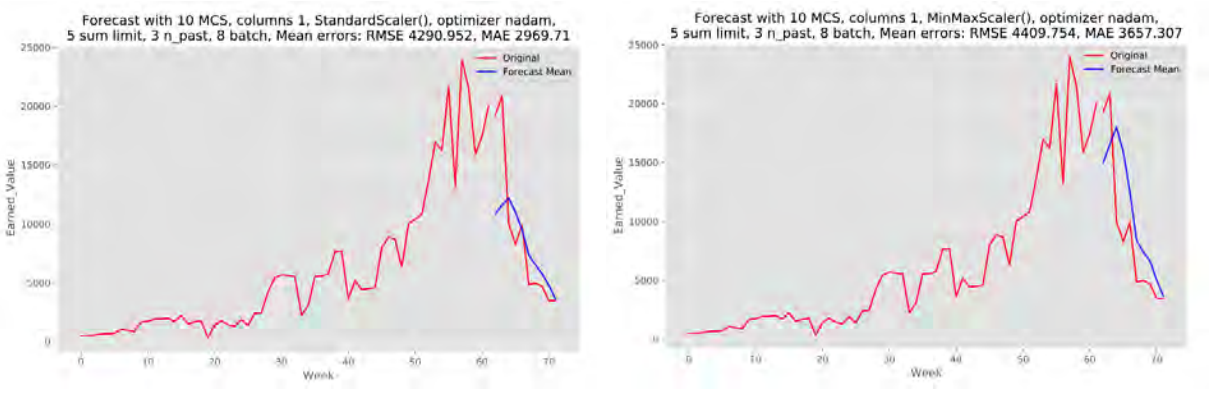
Table 4.5 shows the results from the grid search with a train and test split of 0.83 and n_past as 3. Higher values for n_past are not included because they would only forecast the last three or six weeks. The table includes the top ten performing configurations and is sorted on the error term of RMSE. Table 4.5 shows different characteristics than Tables 4.3 and 4.4 regarding input features and scalers. Input features regarding description look to be the best when the split is 0.83. Table 4.5 also shows that there are equal amounts of StandardScaler and MinMaxScaler among the top-performing, in which the first is at the top. Regarding the number of batches, there looks to be no pattern of which performs the best. Therefore, the top-performing is kept for further analysis.

Table 4.5: Table of the ten best performing configurations of hyperparameters with a split of 0.83.

Rank	Input	Scaler	n_past	Batch	RMSE	MAE
1	Description	Standard	3	8	4290	2969
2	Description	MinMax	3	8	4409	3657
3	Description	MinMax	3	16	4442	3686
4	Description	Standard	3	32	4446	2952
5	Description	Standard	3	16	4485	3007
6	Description	MinMax	3	32	4529	3749
7	Codes	MinMax	3	8	5181	4396
8	All	MinMax	3	16	5405	3964
9	Codes	Standard	3	16	5457	3908
10	Codes	Standard	3	8	5497	3985

Figure 4.9 shows the plot of the two best performing configurations when the split is 0.83

and a n_past of 3. For the StandardScaler in Figure 4.9a, the initial forecast is close to half of the original. After week 66, the performance is better, and there are low deviations against the original. In Figure 4.9b, the initial forecast is closer to the original. There looks to be a delay when the forecast decreases compared to the original, which results in a larger error score. However, the shape of the curve with MinMaxScaler is considered to be closer to the original than with StandardScaler. Based on Table 4.3 and Figure 4.9, the best configurations is input features from descriptions, MinMaxScaler, and batch size of 8.



(a) Testing of the model with StandardScaler. **(b)** Testing of the model with MinMaxScaler.

Figure 4.9: Results from determining the scaler to include for further analysis.

4.2.3 Time series with MCS

After the configurations were set, the model could be run with a higher number of MCS. For the results on the split of 0.73 with both n_past of three and six, the model ran with 500 MCS. For the split of 0.83, the results showed an adequate density of the simulations. This can be seen in Figure B.12. Therefore, the results from 10 MCS are kept as the final results.

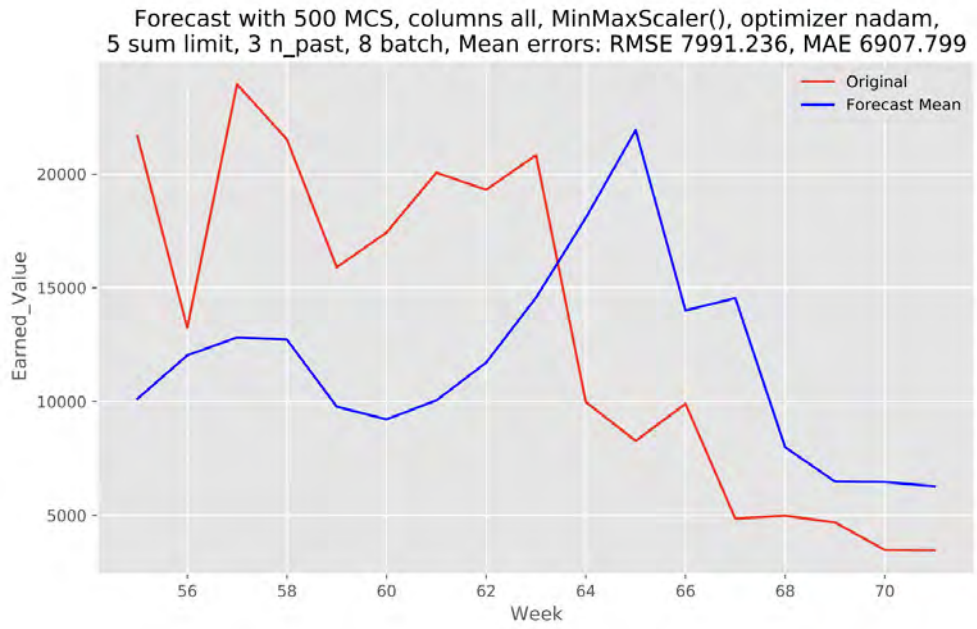


Figure 4.10: Plot of the earned value with the forecast mean for the configuration with lowest RMSE score when the split is 0.73 and n_past of 3.

Figure 4.10 shows the performance of the model implemented with MinMaxScaler. This configuration of the hyperparameters is the one with the lowest error when the split is 0.73 and an n_past of 3. As shown, the weeks in the plot are the forecast weeks with an RMSE score of 7983. The forecast has large deviations, with a maximum deviation of over 10 000 in EV at week 65. The forecast values are lower than the original until the intersection at week 63, after which the forecast values are higher. However, if the forecast curve would be shifted two weeks to the left, the curves would be tight. The shape of the forecast curve resembles the original curve but with some delay. Therefore, the forecast values are close in magnitude but with a delay. For instance, the peak at week 63 for the original appears in week 65 for the forecast. At week 65, the original has decreased while the forecast increases, resulting in a deviation over 10 000 in EV. The delay might be that the model does not have enough or correct information to forecast accurately. Additionally, the valley in the original between week 57 and 61 is also apparent in the forecast, but with an offset of approximately 5000 in EV. As can be seen in Figure 4.10, the shape of the forecast curve is smoother than the original. This is a sign of good performance by the model. It indicates that the model has learned from its input features to adjust the forecast value. Would it be an identical curve with a week delay, it would mean that the model only uses the inherent week's EV to forecast the next week's value.

Figure 4.11 shows the performance of the model implemented with MinMaxScaler. This configuration of the hyperparameters is the one with the lowest error when the split is 0.73 and a n_past of 6. As shown, the included weeks are the ones with the forecast with an RMSE score of 6196. The forecast has large deviations, with a maximum deviation of over 8000 in EV at week 58. The shape of the forecast curve looks to be close to the original regarding increasing and decreasing areas. However, the forecast curve in Figure 4.10 is able to achieve higher values to fit the peaks than the forecast curve in Figure 4.11. The forecast in Figure 4.11 have large deviations until week 64. The model looks to not predict high enough values, resulting in a deviation of over 5000 in EV over several weeks. Evidently, by visual inspection of the figures, the forecast curve will be smoother with a higher n_past. Compared to Figure 4.10, the forecast curve in Figure 4.11 is smoother and has lower slopes. Additionally, in the latter, the peak in the forecast is lower than the original. The forecast values in Figure 4.11 have a span in values of approximately 6000 in EV. For the forecast in Figure 4.10, the span is approximately 15 000 in EV. In Figure 4.11, the peak at week 63 for the original appears in week 65 for the forecast. At week 65, the original has decreased so that the deviation is approximately 5000 in EV. Additionally, the valley in the original between week 57 and 62 in the original is also apparent in the forecast, but with an offset of 5000 in EV.

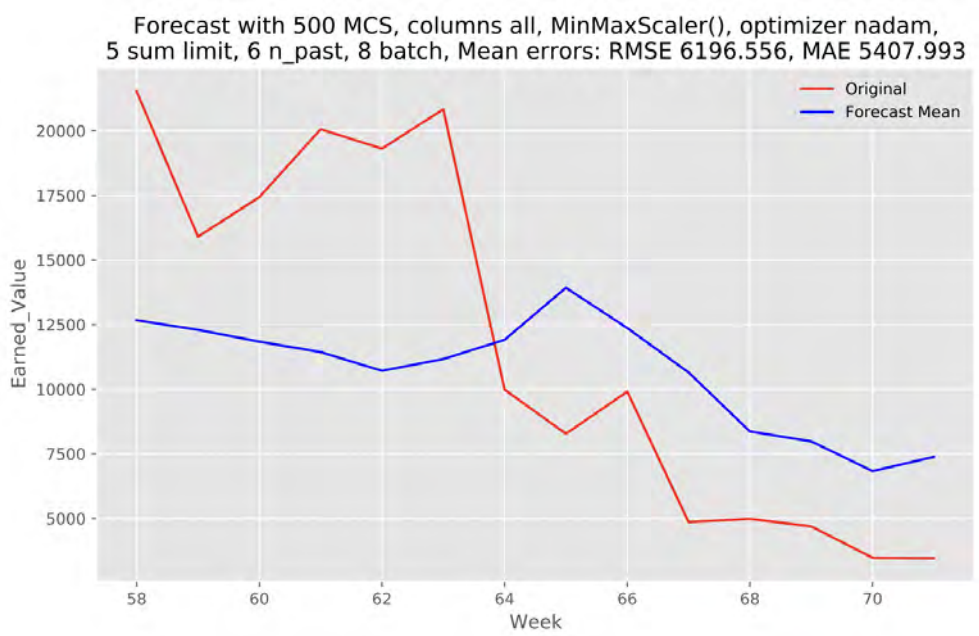


Figure 4.11: Plot of the earned value with the forecast mean for the configuration with lowest RMSE score when the split is 0.73 and n_past of 6.

Interestingly, the model with the lowest error may not be the one to perform best on the most critical steps of the project. Figure 4.12 shows the performance of the model implemented with Standard Scaler. As can be seen in Figure 4.5, this configuration is the best performing with an error of 4290 in terms of RMSE. According to Figure 4.12, the performance is acceptable as the forecasted value tracks in which areas the original increases or decreases. Just after week 60, where the forecasting begins, there is a large deviation from the original. The error at this point is approximately 8000 in EV. This deviation is larger than the earned value per week for most weeks of the project. However, from week 63 and onward, the forecast looks to perform better. The forecast intersects with the original at two points, and the shape of the curves are similar. The signs of good performance lie in the area where the forecast decreases by the slope approximately to the original. On the other side, the forecast looks to be unable to predict the rapid changes. For instance, the original is decreasing until week 65 and then increase to week 66. In the same area, the forecast is continually decreasing until the end of the project but with a lower slope. As shown in Figure 3.6, there are only three dates at the end of the project in which activities occur. Therefore, the model only has three chances to adjust its predictions at an interval of 10 weeks. This might make the errors higher in the end.

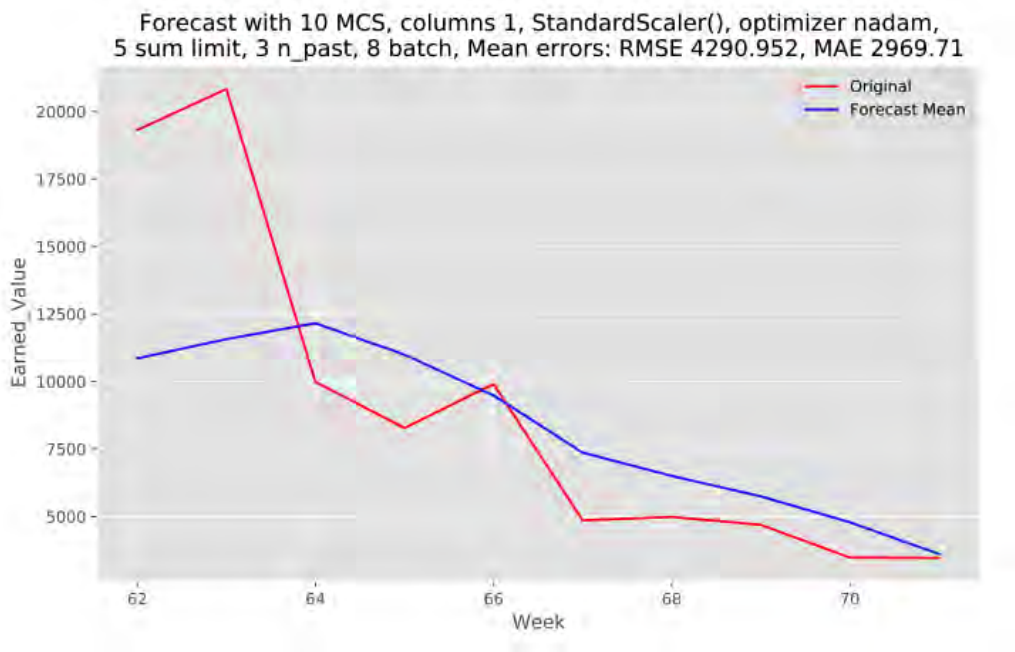


Figure 4.12: Plot of the earned value with the forecast mean for the configuration with lowest RMSE score.

Figure 4.13 shows the performance of the model implemented with MinMaxScaler. This configuration of the hyperparameters is the one with the lowest error when the split is 0.83 and an n_past of 3. The plot only shows the weeks from 62 to 71 because these are the forecast weeks. As shown, the RMSE score is 4409. Even though these errors are higher than of Figure 4.12, the performance of the MinMaxScaler is considered to be better. The forecast curve resembles the original more than in Figure 4.12. This makes sense since the MinMaxScaler is better at keeping outliers in the datasets away from the mean than the StandardScaler, see Section 2.3.9 [186]. In many cases, it might be the most valuable for a company to predict changes in the trend, making the MinMaxScaler more advantageous. The forecast in Figure 4.13 starts too low like the StandardScaler, but with a deviation of approximately 4500 in EV. Because the original data, and the nature of projects, have several spikes in the data, it is important to make a model to catch these spikes. By visual inspection, the forecast in Figure 4.13 is considered as the closest to catch these spikes. However, after the maximum at week 64, the forecast is constantly decreasing until the end of the project. The original has a maximum at week 63 and then decreases until week 65. By trying to hit the correct magnitude of the spikes at a delayed time, the model will provide forecasts that have large deviations. For instance, the peak in the forecast is a week delayed from the original, which results in a deviation of 7500 in EV at week 64. If the forecast managed to predict the peak to be in week 63, the deviation would only be 2500. This looks to be the cause of the large deviation from week 64 to week 66 in Figure 4.13. The forecast look to be ending with the same EV as the original. By comparing the change in slopes between the original and forecast curves, the latter looks to be one week delayed. The model looks at the current week and two weeks back to predict the next week. Therefore, the model looks to emphasize the inherent week to predict the next week's value.

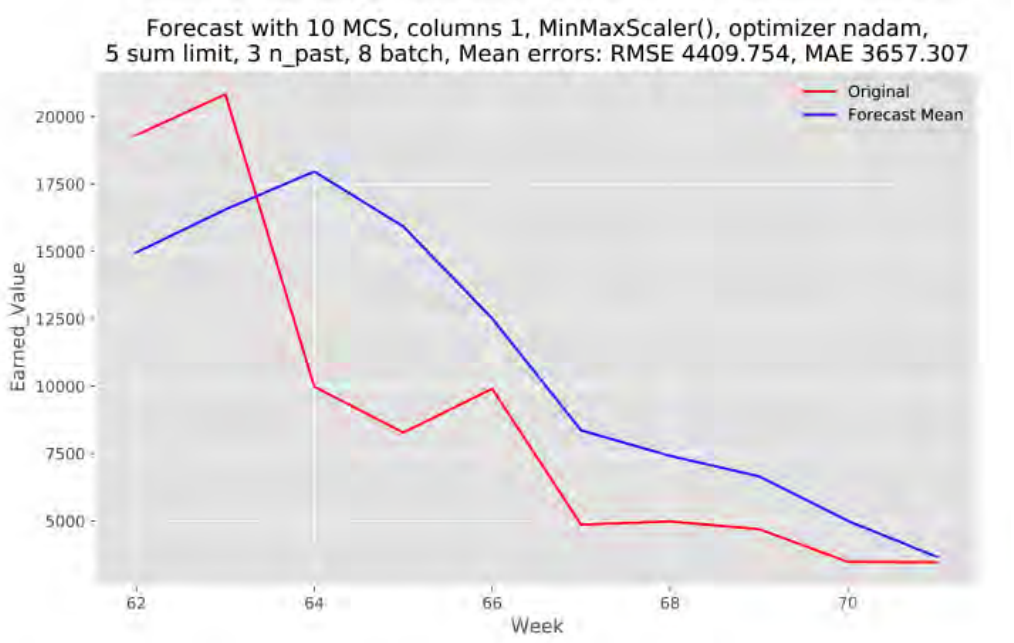


Figure 4.13: Plot of the earned value with the forecast mean for the configuration with lowest RMSE score when the split is 0.83 and n_past of 3.

4.3 Results from the Classification Analysis on Company B

Multiple intermediary results were obtained in the process to achieve final results in the classification analysis. The intermediary results include best hyperparameters, the number of hits and misses, and confusion matrices. Further, the results from the classification with its accuracy are presented.

4.3.1 Preliminary results

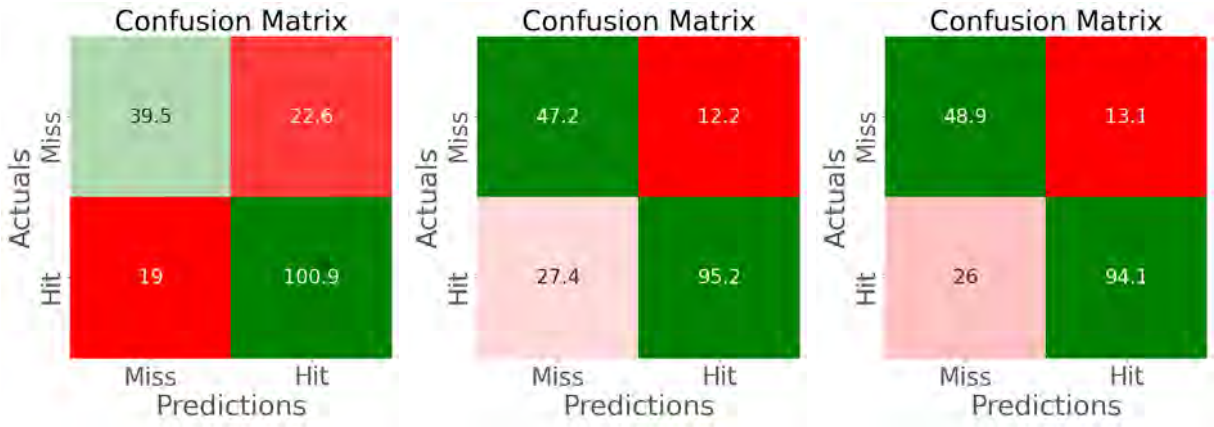
By using the Equations 3.2, the number of started and finished activities and the percentage of hits were calculated. This is illustrated in Table 4.6. Most of the project's activities were a hit. In fact, 65% of the 643 activities was labeled as a hit.

Table 4.6: How many of each type of activity metric found by model.

Type	Hits	Misses	Total	Hit%
Start	415	228	643	65%
Finish	456	187	643	71%
Start and Finish	364	279	643	57%

The confusion matrices (CM) from the preliminary analysis on the shuffle parameter are illustrated in Figures 4.14 and 4.15. In Figure 4.14, the shuffle setting is set on "True".

This implies that the train-test selection will be mixed throughout the dataset. All models did it quite well in this task. In Figure 4.15 the shuffle parameter is set to "False". The train-test split was set to 0.71, and thus, the first 71% of the dataset be used for testing. The F1 scores of the individual CMs are written in their captions. The F1 scoring is set to "binary", i.e. two classes. As shown, the mean F1 is lower in Figure 4.15 than in Figure 4.14. For Figure 4.15c the model wrongly classifies the actual hits into misses to a high degree, yielding an F1 score of only 0.393. From Table 4.6, it is known that the hits account for 65% of the dataset. Thus, a no skill classifier would reach this accuracy by pure guesswork.

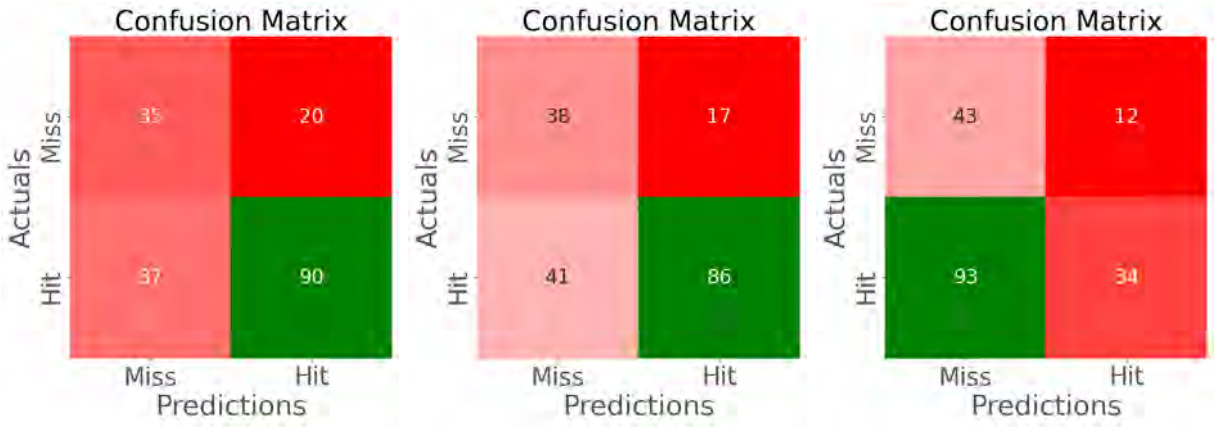


(a) XGBClassifier. F1 = 0.829

(b) RandomForestClassifier 1. F1 = 0.852

(c) RandomForestClassifier 2. F1 = 0.827

Figure 4.14: Mean (N=100) Confusion Matrices for three classifiers, **shuffle = True**.



(a) XGBClassifier. F1 = 0.759

(b) RandomForestClassifier 1. F1 = 0.748

(c) RandomForestClassifier 2. F1 = 0.393

Figure 4.15: Confusion Matrices for three classifiers, **shuffle = False**.

4.3.2 The configuration of the hyperparameters

On closer inspection of the six models, one can find subtle differences. In Section B.3, the model's hyperparameters are listed as they are sent as input to the model. The default hyperparameter value will be written as a number in parenthesis after the parameter name. Between the two XGB classifiers, one can see five main differences [113]. The first one uses only 100 decision trees, or n_estimators (100), in its forest, while Model 2 has 1000. Table 4.8 shows that Model 2 achieves higher F1 scores in all tests. However, the number of DTs is increased tenfold, making each iteration ten times more computationally expensive. Further, the second model has a higher minimum child weight (1), which means it is more conservative in the importance weight the model gives each child nodes. A child node is the next node in the tree, i.e. one level deeper than its parent node. This may reduce the chances of overfitting since it introduced regularization on the Hessian, i.e. the second derivative [187]. On the other hand, Model 2 has a lower maximum depth (6) at 4 instead of 6. This means that each decision tree only will be four levels deep. The learning rate (0.3) is lower for Model 1 than for Model 2, at 0.001 versus 0.5. A lower learning rate often prevents overfitting [113]. Lastly, the first model has a subsample (1) of 0.4 while the second has 0.75. The subsample hyperparameter states how the train split gets split up internally in XGB to train before decision trees are grown. Other minor differences between the two XGB will not be discussed further.

The two RF models also have five main differences [188]. First, the number of estimators (100) in the forest differ. The first model, Model 3, used the default value, while the second uses 118 decision trees. As a rule of thumb, more trees will be more computational expensive since it results in a more complex and robust model, which may give better results. However, the difference in the number of trees is only 18 and should thus not yield a much more complex model. The maximum features ("auto") setting is different between the two models. The first model has the value set to 0.25, which dictates the maximum number of features. It does this by computing the following equation to the closest integer: max_features $\cdot$ n_features. This will equal a quarter of the features that come into the model since the setting is 25%. This is shown in Table 4.7. The second model uses "auto", the default value, which takes the square root of the number of features in the feature set, see the before mentioned table. The minimum samples per leaf (1) is

also different. The last child node in a tree is called a leaf. Model 3 uses 17, while Model 4 uses 5. This hyperparameter sets a constraint on how few samples must be in a leaf to keep it [188]. A higher value will decrease the possible depth of the model, which prevents overfitting and makes it quicker. Further, the minimum sample split (2) is set to the value 11 for Model 3. Model 4 has the value 4. Two events may stop the tree from growing. Either if the minimum samples per leaf constraint mentioned above happen. Alternatively, if the minimum sample split becomes lower than this threshold, of 11 or 4, respectively. A higher number leads to less overfitting. Lastly, the ccp_alpha (0.0) hyperparameter is a complexity parameter that relates to Minimal Cost-Complexity Pruning [189]. By default, no pruning is performed, which is the method of Model 3. Model 4 has a ccp_alpha of 0.001. Proper prior pruning prevents poor performance, overfitting, and makes the model more adaptable. By selectively removing parts of the tree, the overall health of the tree increases [190].

Table 4.7: The number of features in each feature set, and the splits dependent on the RF model.

Feature Set	# Features	25% of Features RF Model 1	Square root of Features RF Model 2
Description	28	7	5
Codes	78	20	9
Number	6	2	2
All except scope	107	27	10
All	110	28	10

Lastly, the fifth and sixth model are not based on decision trees but on neural network [115]. The two MLP models have three main differences. First, the depth and width of the hidden layers (Depth = 1, Width = 100) are different. Model 5 has two layers with 20 nodes in each, while Model 6 has 20 nodes in its first layer and 10 in the second layer. The number of nodes and layers were discussed in Section 3.3.4. Secondly, the maximum number of epochs (200) is 300 for the first MLP model and 1000 for the second. More epochs may result in both better accuracy and overfitting. Thus, it is a double-edged sword. Two events may stop the model from running. Either that the model converges on a score, i.e. finds a local optimum. Alternatively, that the maximum number of epochs is reached. See Section 3.5.3 for the discussion on number of epochs. Lastly, the alpha (0.0001) hyperparameter is different. For Model 5, it is ten times the default, and for

Model 6, it is five times the default with values of 0.001 and 0.0005, respectively. This parameter denotes how the L2 penalty is implemented [115]. The L2, or penalizing term, constrains the size of the weights used in the nodes, making overfitting less likely. This is done by changing the decision boundary. If one increases the alpha, the weights will be smaller, which decreases the variance. Contrary, if one decreases the alpha, it may aid in keeping the bias low and may give a more accurate decision boundary.

4.3.3 The classification analysis

The analysis started by looking at the MCS of the models on a feature set of all the features that were available to the models. Multiple tests were carried out. Further, a deeper investigation looked into how the models behaved as the feature sets were changed. This will indicate the robustness of the model and is a way to perform sensitivity analysis [178].

In Table 4.8, the eight original models with name and respective F1 scores are shown. The shuffle setting is set to "True" since MCS were to be implemented. Of the eight classifiers in Test 1, there was three XGB, three RF, and two MLP. The XGB were all quite close in terms of F1 score. Among the RF, the one with the lowest performance was duly dropped. MLP achieved the lowest performance with all values under 0.8. From Figure 3.10 the way the MCS are averaged and binarized is illustrated, i.e. allowing the shuffle parameter to be set to "True". For Test 2, all the six models achieved quite similar results as in Test 1 in terms of F1 score. As can be seen, the numbers still vary from Test 2 to Test 3. The biggest change was that of the sixth model, where the F1 score dropped from 0.793 to

Table 4.8: Comparison between 100 MCS and two 5000 MCS tests. Also which models that got used in the end. The **bold** font is the most accurate model in each test. Higher F1 score is better. The shuffle setting is set to "True".

	Test 1: 100 MCS		**Test 2: 5000 MCS**		**Test 3: 5000 MCS**	
#	**Model**	**F1**	**Model**	**F1**	**Model**	**F1**
1	XGBClassifier	0.835	XGBClassifier	0.830	XGBClassifier	0.837
2	XGBClassifier	0.839	XGBClassifier	0.838	XGBClassifier	0.838
–	XGBClassifier	0.833	—	–	—	–
3	RFClassifier	0.853	RFClassifier	0.854	RFClassifier	0.847
–	RFClassifier	0.825	—	–	—	–
4	RFClassifier	**0.884**	RFClassifier	**0.883**	RFClassifier	**0.898**
5	MLPClassifier	0.780	MLPClassifier	0.779	MLPClassifier	0.822
6	MLPClassifier	0.794	MLPClassifier	0.793	MLPClassifier	0.692

0.692. A score of 0.692 is only four percentage points above the no skill classifier as the dataset contains 65% hits. However, the other MLP model increased its accuracy.

Table 4.9 show the F1 scores for all the feature sets for all models. In this table, certain visual aids such as bold, italic, and underline are used. The meaning of these can be found in the table's caption. It is interesting to see that the feature set of all features achieved the highest score in all but the XGB classifiers, i.e. four out of six. It seems like this model is a robust enough model to perform well even though the number of features decreases. In Table 4.7 the number of features per feature set is shown. Model 1 has the highest mean F1 score of 0.839, and the two RF models come as second and third. Visual inspection shows that the MLP models performed consistently with the lowest F1 scores for all feature sets. This is reflected in the mean of the models. Another interesting aspect is that of Model 1 in the Description feature set. The F1 score is the lowest in the Model 1 portfolio, yet the best performing within the feature set.

Table 4.9: The F1 scores for all feature sets for all models. **Bold** text show the best score and the *italic* text show the worst score per model. Underlined text show the best score per feature set.

MCS 1000	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	–
Model	**XGB**	**XGB**	**RF**	**RF**	**MLP**	**MLP**	**Mean**
F1 score feature set Codes	0.841	0.808	0.845	*0.777*	0.758	0.779	0.801
F1 score feature set Description	*0.814*	*0.783*	*0.787*	0.803	0.789	0.786	0.794
F1 score feature set Numbers / Scope	0.851	0.798	0.838	0.848	0.787	0.742	0.811
F1 score feature set All except scope	**0.861**	**0.851**	0.821	0.857	*0.756*	*0.685*	0.805
F1 score feature set All	0.826	0.834	**0.850**	**0.882**	**0.795**	**0.826**	0.836
Mean	0.839	0.815	0.828	0.833	0.777	0.764	–

As mentioned in Section 3.6.4, SHAP values of the classification were found [119]. Figure 4.16 shows the SHAP plot for Model 4. The F1 score in the top right corner is not the average F1 score of the model on the given feature set but the static F1 score of that iteration. Thus, it is only one of the many MCS. In other words, the F1 score listed in Table 4.9, is more representative. The figure shows the three main features used by the model to predict the start hit class are scope related features. Interviews with the

representatives and documentation from Company B revealed the information behind these three features. All these scope features are estimated by the software that summates on activity level after the activities listed in the project scope in the software. The feature of "Rsh" relates to the current scope. It consists of the current scope, which is the original scope, along with variation orders (VOs), variation order requests (VORs), internal changes and transactions, and sub-contracted work. This feature will be updated at each baseline update and at the weekly status runs. Tsh is defined to be identical to Rsh, while "csh" is set up to only contain the baseline scope, the original scope, and VOs. "csh" will be only updated at the initiation of new baselines.

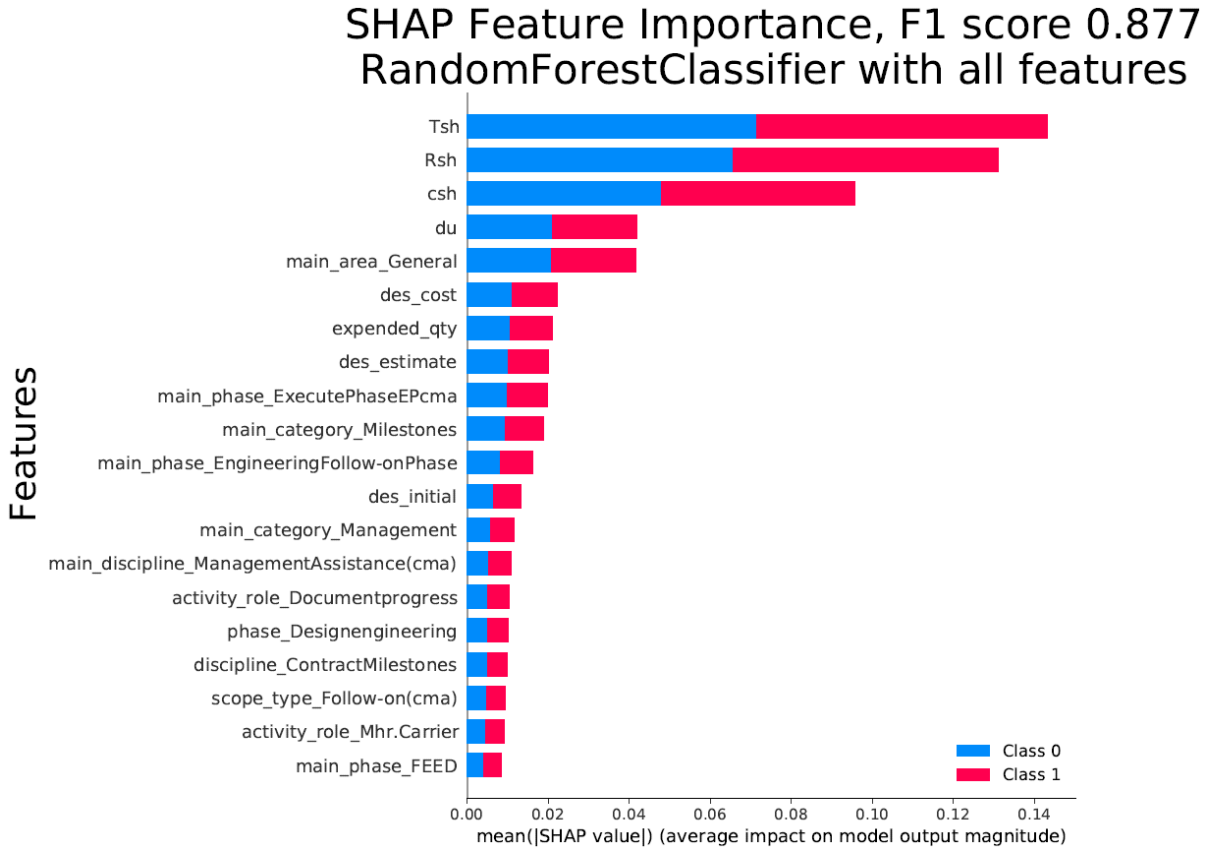


Figure 4.16: Plot of the SHAP values of Model 4.

5 Discussion

The results showcased how the different models in the respective datasets performed. Due to the nature of the two company's datasets, it is possible to see what is possible to accomplish given the different resolutions and count. The inherent attributes, possibilities, results, and limitations will be further discussed here.

5.1 Company A - Time Series

This subsection will discuss the model's performance from the analysis on Company A and which implications there might be for project management. The implications for project management do not only apply to Company A and can be extended to yield for similar companies with multiple projects on a template to support machine learning. Also, certain aspects of how the data is handled are discussed.

5.1.1 Performance of the model

As presented in Section 3.3.4, the model ran with 100 MCS. As there are stochastic parameters included in machine learning, several simulations should be run to obtain an average performance value. Figure 5.1 shows the performance of the different simulations in which each curve is from one MCS. The vertical axis shows the label, expended value in percent, and the horizontal axis is weeks from start. The figure shows the weeks from 6 to 92, which is the weeks the forecast is done. The interesting aspect of this figure is how dense the blue curves are. If they make up a thick curve, the simulations are close in value, whereas identifying individual curves represent larger deviations. For instance, around week 35, it is possible to spot single curves. At the peak of the forecast, the difference between the lowest and highest individual curve is over one percentage point. Thus, the difference is similar to a week with normal production. From Figure 4.12, it is evident that the forecast mean deviates from the original with an RMSE score of 0.0044. Comparing the mean to the individual simulations in Figure 5.1, it is clear that the simulations might deviate from the mean by more than 0.4 percentage points. Therefore, to the project management team, the model might provide values not meeting the desired accuracy level at the parts with the largest fluctuations. Inaccurate cost estimations are discussed to be

one of the challenges for project management in the energy industry [10]. The project management team should identify the areas with large differences between the forecasts as areas with high uncertainty and do more thorough calculations. For many time intervals of the project, there is a high density for the blue curves. This suggests good performance so that the next value acquired by the model is close to the average obtained from MCS. The high density of the blue curves at several time intervals indicate high precision of the model, which is why it is not run through a higher number of MCS after the first of 100.

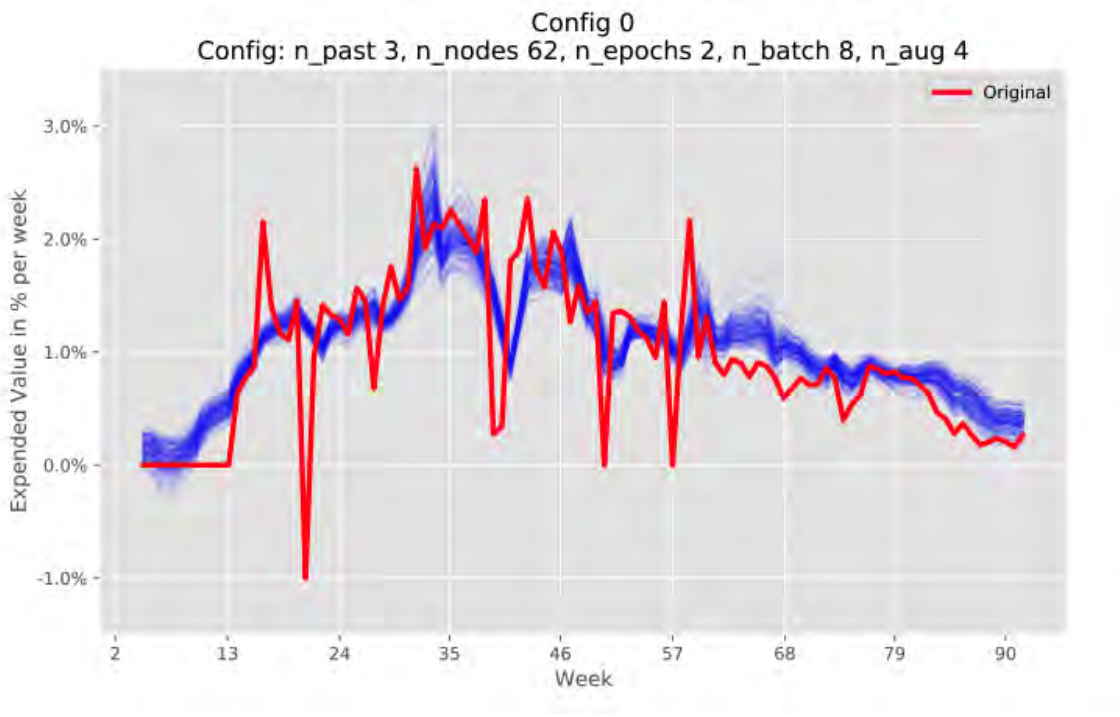


Figure 5.1: A plot of the simulations with 100 MCS.

As can be seen in Figure 5.1, there is some delay in the forecast. The original curve has a maximum point approximately at week 30, whereas the corresponding maximum for the forecast is approximately at week 32. The delay of two weeks is lower than n_past and may indicate that the model must see the spike in the input to be able to forecast it. By this reasoning, the model lay most emphasis on the label closest to the inherent week. However, this performance is not the case throughout the project. In certain time intervals, the model accurately predicts the EV for the same week as the original. More precisely, this occurs approximately at weeks 15, 20, 30, 50, 60, and 75. Further, in the same figure, a negative expended value can be seen. This was stated by the company during interviews to be a result of necessary rework that had to be performed but is not the norm of the

company's methodology. Further, in Figure 4.5, an interesting observation regarding pre-padding is visible. The wanted results of the pre-padding are not yet achieved in Figure 4.5a. However, it seems like the pre-padding is somewhat better understood in the second chain, in Figure 4.5b. In LSTM models, one wants to add pre-padding the dataframe so that the model starts at a constant baseline of zero [167]. In hindsight, the pre-padding could have been even longer.

In Table 4.3 it can be seen that mostly the MinMaxScaler is in the top ten. However, at rank 4 and 6, the feature set "Codes" and the StandardScaler appears. These are the only times they appear in this table. This may indicate that this combination is a good combination of these two parameters. Further, it is the combination with the lowest batch number that ranks the highest of these two. It is interesting to see that just these codes for internal management purposes can achieve almost the same RMSE score as the model with access to all the available data.

5.1.2 Specific handling of the data

The model's label is a value for how much the expended value has increased in percent in the inherent week. Thus, it is not cumulative. The intuition was that a non-cumulative label could provide more information than a cumulative label. Additionally, interviews with the representative lead to the decision that a non-cumulative label is more interesting from a planning and project management perspective. With this method, the project management team could more easily identify which weeks there is a steep slope in the expended value. It will also be possible to identify if there will be frequent fluctuations throughout the project or settle on a value to be close to constant production each week. Unlike a cumulative expended value, the non-cumulative values have frequent fluctuations and might have spikes in the data. As shown, the model is not able to accurately forecast the spikes. However, the model can forecast the shape of the original curve and its frequent fluctuations. To forecast the shape of the cumulative expended value might have been easier for the model. For cumulative values, the most common shape is an S-curve. The time interval of the inflection point in the S-curve would be one of the critical features to forecast. This interval is considered the most difficult during the project, and thus, any aid an ML model can give would prove helpful and increase the knowledge pool of the project management team. It was decided that the model will forecast the non-cumulative

expended values.

One way that was thought to make the projects more uniform was to reshape all the projects into a certain length. For example, to make the length of all projects to 100 weeks and see how the model would perform. This gives the algorithms in the models a standardized input shape, but it may also decrease the robustness of the models. Further, to alter the datasets in this manner is not usual nor that logical. It was chosen not to pursue this idea any further in this model. When it comes to augmenting methods, many more methods are available. However, the methods used in this analysis were thought to be enough for this pilot model. Still, more advanced augments may have yielded different results and could be further researched.

In one of the datasets, a feature called "Milestone" was logged. This was considered as a feature that could indicate important attributes to the model and was wanted to include. However, since only one dataset had it, it was not possible to keep it. This lack of completeness between the datasets limited the possibilities, and thus, may have held back the possible accuracy of the models. As shown in Table 3.4, there are only 13 available features from the 484 possible ones that are fully logged between the datasets. That is just above 3% of the possible available data the model could utilize. It is important to remember that it is not an issue specific to Company A but rather a global challenge many companies face today. The challenges of how to structure the data, capture the valuable data points, and save them are something that constantly is being addressed in this new industry of data mining, management, and analysis [12]. However, these 484 features are not all unique, nor are they all relevant to the models. However, some features were considered valuable but were not tracked in the project summary template. This spurred the thought of which features would be practical to have available, such as the VOs and VORs, an indication on baseline dates, and quarter-end dates. The plan data was available for some of the datasets but not for all. Thus, it was not possible to utilize these features.

5.1.3 Implications for the project management team

According to the plot in Figure 4.6, the model will make a conservative forecast. Therefore, the actual values might be higher in the peaks and lower in the valleys. As mentioned

in Section 2.1.1, project management in the energy industry finds it difficult to have adequate advanced planning. The non-cumulative expended value might aid the advanced planning to identify the most critical time intervals of the project. To exemplify, the project management team of Company A could help the providers quantify the expected amount of production each week so the providers can plan for it. By visualizing the trend of the original curve, the forecast value will most often be within a deviation lower than the RMSE score. The small fluctuations have mostly 0.2 to 0.3 percentage points deviations from the forecast curve. Therefore, the performance of the model looks to be adequate for practical use. Thus, the model can aid in the step of advanced planning by providing a forecast of the expended value with low deviations from the actual values.

A 37% drop in RMSE was seen between the first and second chain in Figure 4.5. This may indicate that a chained model with more projects could perform at a satisfactory level to the project management forecast team. By satisfactory, what is meant is that the overall trend and the spikes of the project are predicted before the events occur so that actions may be taken. Additionally, the chained model will make it possible to forecast throughout the project. Machine learning trains typically on 80% of the data and tests on the rest. The model will train on complete projects and test on the complete subsequent project in a chained model. Testing on all phases of the project will provide the project management team with more information. For instance, the team can analyze the forecasting and identify the most challenging time intervals. The tasks at the project's peak could be rescheduled or make adjustments in the WBS to make it more manageable. Consequently, this could make advanced planning easier to achieve.

After interviews with the representative, the spikes in the data would be interesting to forecast. However, they look to be too difficult to predict as none of the configurations accurately predicts their magnitude. The top-performing configurations can predict a significant change in value but not of the same magnitude as the actual value. For the original values of the expended values per week, only seven instances are below 0.5%. As shown in Figure 4.6, the weeks with such production is either before week 15, in the negative spikes, or after week 75. Thus, the RMSE score of 0.4 percentage points represents a week below low production. In the interval between these weeks, the expended value per week fluctuates by more than the RMSE and most of the expended values are in the

interval from 1% to 2%. As the span of 1% is more than twice the error score, the model can provide realistic and precise values. From Figure 4.6, it is also evident that weeks with approximately 2% in expended value represents weeks with high production, whereas weeks with approximately 1% represents weeks with average production. Therefore, a forecast error of 0.4 percentage points will not make a week with average production into a week with high production. This precision is essential for the project management team as it is stated that a large part of the projects in the energy industry exceeds their cost estimations [10]. A forecast value at the same magnitude as the original could help the team keep track of the costs and recognize when the project exceeds planned costs. As can be seen in Figure 4.6, the forecast curve is smoother than the original. Because the average error per week is smaller than the difference between the weeks with medium and high production, the model can help the project management team to forecast which weeks will have low, normal, or high production. After the project management team has produced plans for expected production within each discipline for each week, the model can provide the shape of the actual production.

In Figure 4.6, there is a delay at certain time intervals. The largest is at the spike at approximately week 18, where the model forecast the same spike at week 22. Otherwise, the delay looks to be from one to two weeks. This might pose a challenge for the project management team. If they were to schedule deliveries based on the forecast in Figure 4.6, the materials would arrive after the intended date. As projects often are associated with delays, an additional delay in the forecast model would not be satisfactory. Project management in the energy industry is stated to have faced challenges with increased globalization and to balance multiple objectives [2]. Delayed deliveries and activities in a global perspective make it more crucial to have a model that predicts on time. Having an inaccurate model could increase the uncertainty because it provides value that will not be realized. Additionally, planning with a delay of four weeks may be too high for the project management team and might make scheduling activities more complex and challenging. Conversely, retrieving forecasts on expected expended value could make it easier for the project management team to balance multiple objectives. For instance, to delegate more effort to the goals with the highest priority defined by the project owner if the forecasts show a negative trend in meeting the planned values. As discussed in Section 5.1.1, there are several time intervals in which the models predict accurately. These findings may

suggest that the model could be used to schedule deliveries and aid in a global overview of the model would retrieve more datasets and enhance the accuracy.

Because Company A is an operator in the energy industry, they must collaborate with actors worldwide. As stated by Tanaka, the increased globalization poses additional challenges for the project management team [2]. An ideal machine learning model could include more input features to reflect characteristics in the industry to overcome this. For instance, by having more projects in the model, it could learn to distinguish between regions where the production is done and which features are most important for the respective regions. The holiday feature was implemented to include information that is specific to a region. If Company A were to implement forecasts powered by machine learning, they could include more features specific to regions. Mullaly and Thomas argue that limited reporting and weak collaboration between firms and subcontractors is a negative characteristic in the energy industry [11]. A forecasting model powered by machine learning could increase the information sharing between the actors. As production often is outsourced to countries with lower labour cost, the operators in the industry might have more resources to invest in new technology to enhance forecasting [13]. An accurate forecast from the operators perspective could aid the subcontractors to identify areas with low efficiency. Additionally, increased information sharing could aid to even out the differences found by Carvalho et al. [17]. They found that national environments play a crucial role in project performance as some countries have more developed project management methodologies.

5.2 Company B - Time Series

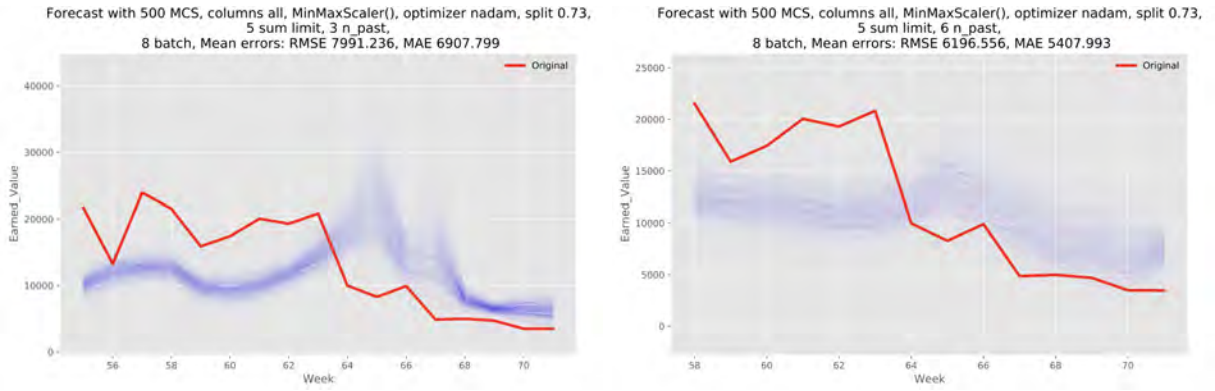
The time series analysis of Company B was based on data from the project management software. The following paragraphs will discuss the model's performance, the information it provides, areas to further investigate, and if it results in implications for the project management team in the energy industry. Likewise, for the implications discussed for Company A, the implications from the analysis on Company B serves as a reference for which implications machine learning can result in for project management.

5.2.1 Performance of the model

As presented in Section 3.5.3, the model first ran with 10 MCS and then with 500 MCS. Figure 5.2a and 5.2b shows the performance of the different simulations when running 500 MCS. The configurations for the figure are the ones with lowest RMSE score in Table 4.5 and 4.4, respectively. The figure only shows from the start of the forecast until week 71, which is the ultimate week of forecasting. The density of the blue curves will indicate the accuracy and precision, and thereby, the performance of the models. If they are tight enough to make up a thick curve, like at week 69 in Figure 5.2a, the precision is high. On the other side, at week 65, separate curves can be identified. This week, the difference between the lowest and highest value in the forecast is 30 000 in EV. This difference is approximately 125% larger than the maximum value in the original data. As shown in Figure 4.10, the mean forecast value in week 65 is approximately 22 500 in EV, and it is approximately 7500 in the original. Therefore, the average value is near 300% higher than the value the model tries to predict. This shows that the model has low precision and low accuracy for the week of 65. The reasoning can be extended to include the time interval from week 63 to 67, the time interval with the largest change in the original curve. From week 63 to 65, the original curve decreases with approximately 12 000 in EV. This looks to pose problems for the model. Besides the interval from week 63 to 67, the density of the curves is higher. The average difference between the lowest and highest incremental forecast looks to be approximately 2500 in EV. This indicates higher precision of the model.

Figure 5.2b shows different characteristics. As can be seen, there are no points with a high density of blue curves. However, at week 65, the difference between the lowest and highest forecast value is approximately 15 000 in EV, which is half the difference for the same week in Figure 5.2a. Thus, the forecast curves of Figure 5.2b indicate a model that is more precise in the most difficult time interval of the data. However, the average difference between the lowest and highest forecast value in each week looks to be 5000 in EV. The weekly EV is under 10 000 for most parts of the project. Therefore, a deviation from the mean in both directions by 2500 in EV is inadequate and could provide the project management team with uncertain values. Consequently, the performance in Figure 5.2a might be considered to outperform the performance of Figure 5.2b. An n_past of 3 will

yield higher precision than an n_past of 6 based on the density of the forecast curves. On the other side, an n_past of 6 will yield higher accuracy than an n_past of 3 based on the RMSE scores, as shown in the figures. Because the model is imprecise at certain time intervals, the project management team should not run only one simulation as it might provide inaccurate values. Inaccurate cost estimations are discussed to be a challenge for the project manager in the energy industry [10]. To overcome this problem, the team should run many MCS to obtain an average value for the forecast.



(a) Simulations of the model with n_past of 3. **(b)** Simulations of the model with n_past of 6.

Figure 5.2: Results from determining the scaler to include for further analysis.

As can be seen in Figure 5.2, there is some delay in the forecast. The original curve in Figure 5.2a has a peak at week 63, whereas the corresponding maximum for the forecast is at week 65. The delay of two weeks is lower than the value of n_past, and thus, the model can see the peak in the original curve. Therefore, a delay of two weeks indicates that the model gives high weight to the past week to forecast the next. Thus, the model must receive feedback before it can forecast. The delay of two weeks might be much for the project management team when planning the following week's production. Consequently, the performance of the model might be inadequate.

5.2.2 Specific handling of the data

To maintain progression with the time series analysis on the dataset from Company B proved to be a challenge. It might spur because even if the dataset is detailed, the model never sees a complete project. Thus, it will never have seen a project at the end of its life cycle. The results from the analysis on Company A showed the model's ability to achieve higher accuracy by training on two projects instead of one, which is the chained structure

mentioned in Section 3.3.4. This is one of the most promising results from the analysis on Company A. From Figure 5.3 it is shown that in the first 59 weeks, the EV is mostly below 5000, with a mean of 3347. Thus, for 70% of the total project duration, the model will regard this as the standard EV of the project. The overall mean of the project is 6375 in EV. As shown in the figure, the peak of the project starts its increasing from week 50. The EV fluctuates a bit in the peak area before it drops down at week 63 at the 90% mark. The mean in the peak of the project, from week 50 to 63, is 17 285 in EV. From week 63 until 72, the mean is 6206 in EV. Dependant on the train-test split, as shown in Figure 3.9, the model has not seen many weeks of high EV. Thus, it will not predict high values. Furthermore, it is not certain that the input features have a large increase in EV in the time interval between weeks 50 and 63 to indicate the peak. In Figures B.4 to B.7, the features in the dataset are plotted. Of these, 29 features are active in the weeks from 50 to 60. Further, of the 29, only 15 are active during the beginning of the project. Thus, the model will have limited time to learn what these features may indicate in terms of the EV. Some activities have different characteristics than the rest in the data set. These are the activities that have the scope features set to 0.001. During the interview, it was discovered that these have no scope for the time aspect. However, these activities do have a cost associated with them. The activities in question represent an important part of the model because they support the overall process and allow for progress. Examples are management tasks and follow-up. These activities constitute about 30% of the dataset, representing a substantial part of the overall model at 189 activities. As the model's label is related to value, these activities were considered too important to drop.

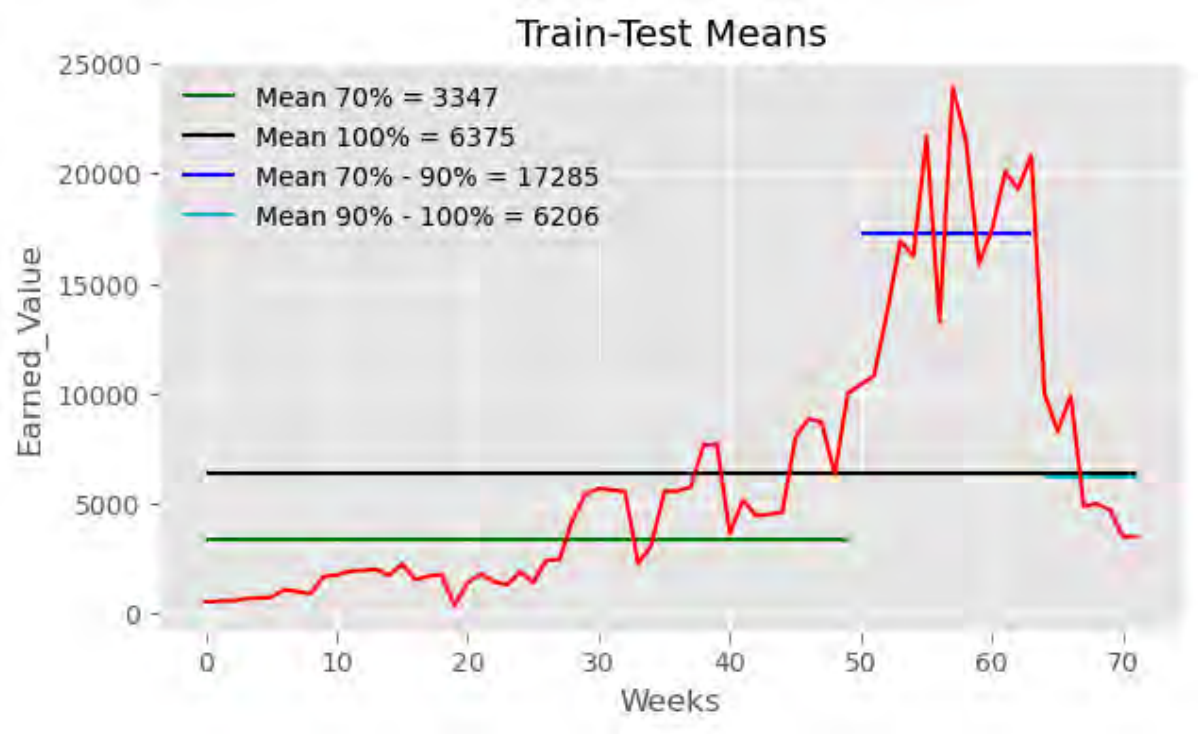


Figure 5.3: Plot of the train-test means.

5.2.3 More thorough EDA and preprocessing

The search for information in the project management software was time-consuming. The software was new to the writers. It took time to get familiar with it and to learn how to navigate. The extracted information is just a minor fraction of all the information in the software. Among the additional information are a change register, risk analysis, number of predecessors and successors, and internal comments on each activity. For instance, the model could include the number of changes and dates there was a change in the activity. This could be a more fit feature related to EV than the start of an activity. However, there were difficulties related to the extraction of the most interesting information from the software. The changes were kept in another format than the characteristic information of the activities. Additionally, the changes were related to the activities by another key than the activity number. Therefore, the information of changes must have been manually inserted into the dataframes. From the extraction, there were initially over one million rows in the dataframe. The manual insertion of information would have been too time-consuming. Thus, the choice was to move on from extracting as much information as possible from the software to investigate what type of information could be extracted through SQL. The databases to extract from containing characteristic information of each

activity, as presented in Section 3.4.1. Keeping in mind that machine learning for time series should not have many input features, it seemed like a rational choice.

For this dataset is was decided not to augment the data. It may have proved fruitful if this aspect was implemented in the preprocessing of the data. It may have made the training more varied and thus, have produced a more robust model. Further, it could have made the model more accustomed to why there will be significant increases in EV. Next, since the EV was calculated weekly, the activities themselves had to be aggregated every week, presented in Section 3.5.2. By doing this, a higher possible resolution was lost. However, weekly was how the company wanted it, and thus what was done. Further, it can be argued if the resolution of EV per activity is at all wanted. Other options could be EV per day. It will depend on what the company want and what the available data will allow.

5.2.4 Implications to the project management team

As presented in the theory, Section 2.1.1, more than 60 percent of the projects in the energy industry have experienced cost overruns of more than 33 percent of the original target [10]. Applying machine learning to the project plan might be of help to the project management team. The results show that the model performs well when including all input features when the split is early, according to Tables 4.3 and 4.4. When the split is later, meaning that the model has more data to train on. Table 4.5 shows the most essential input features are the descriptions. With the model as currently presented in Section 3.5.3, the implications for the project management team is that they can more accurately predict how the progression will be at the end of the project. An ideal model could forecast the EV throughout the project by describing key information of the activities. To have a prediction of the EV beforehand could help get an idea of when the project might lag behind the planned schedule. For instance, if the plots show that the EV is behind in a specific time interval, the team could delegate extra resources to get back on schedule. The plots also show the peaks and valleys of the EV. The project management team can use this to flatten the curve by scheduling some of the work from the peaks to the time intervals of the valleys.

As Sections 4.2.3 presents, the choice of n_past is highly influencing the accuracy of the

model. Based on the results from Figures 4.10 and 4.11, the forecast curve is more volatile when n_past is 3 instead of 6. Also, the latter has a lower RMSE score. Therefore, there is a decision on whether to have a model with the lowest error score or to have one to forecast the frequent fluctuations in EV per week. Based on Figure 4.11, the project management team could forecast the shape of the EV with n_past of 6 more accurately than with n_past of 3. However, the smoother curve that comes with n_past of 6 will not tell if there can be sudden increases or decreases for the weekly EV. Because scheduling is stated to be one of the tasks with the highest impact on the whole project, forecasting the shape of the EV early in the project could be helpful [62]. This could help the team analyze the scheduling and look for possibilities to reschedule in the most difficult time intervals.

With an n_past of 3, the project management team will more accurately know at which magnitudes the EV will be throughout the project. This might be of value to recognize the more difficult time intervals of the project. For instance, if particular time intervals have a weekly EV of magnitudes higher than other weeks. The forecast curve in Figure 4.10 has the characteristic to have frequent fluctuations in the weekly EV like the original data. This curve could help the project team categorize the project intervals with large changes in EV. To discover the difficult time intervals in the project is an important step to know which activities might need rescheduling. Machine learning models can provide more advanced methods to discover these problematic steps, as opposed to the more traditional methods described in Section 2.2.3 [23].

It is argued that projects in the energy industry have increasingly longer average duration, now up to six years [9]. The forecasts will become more uncertain when the duration increases. The model set up for the dataset from Company B implements a train-test split of 73% and 83%, which could prove to be a valuable set-up for projects with long duration. With this implementation, the project management team could use the actual values from the start until the current time as input to the model. Thus, there will be a possibility to compare planned and actual values until the current time to have more information into the forecasts for the rest of the project. The team will obtain measures of the project's deviation from the planned progression and the expected finalization. Gathering information from early phases for input to the model could decrease the error

score, yielding an accurate and precise model to be trusted. A prediction powered by machine learning could prove to outperform the predictions made by experts found in the study by Tetlock [23; 41].

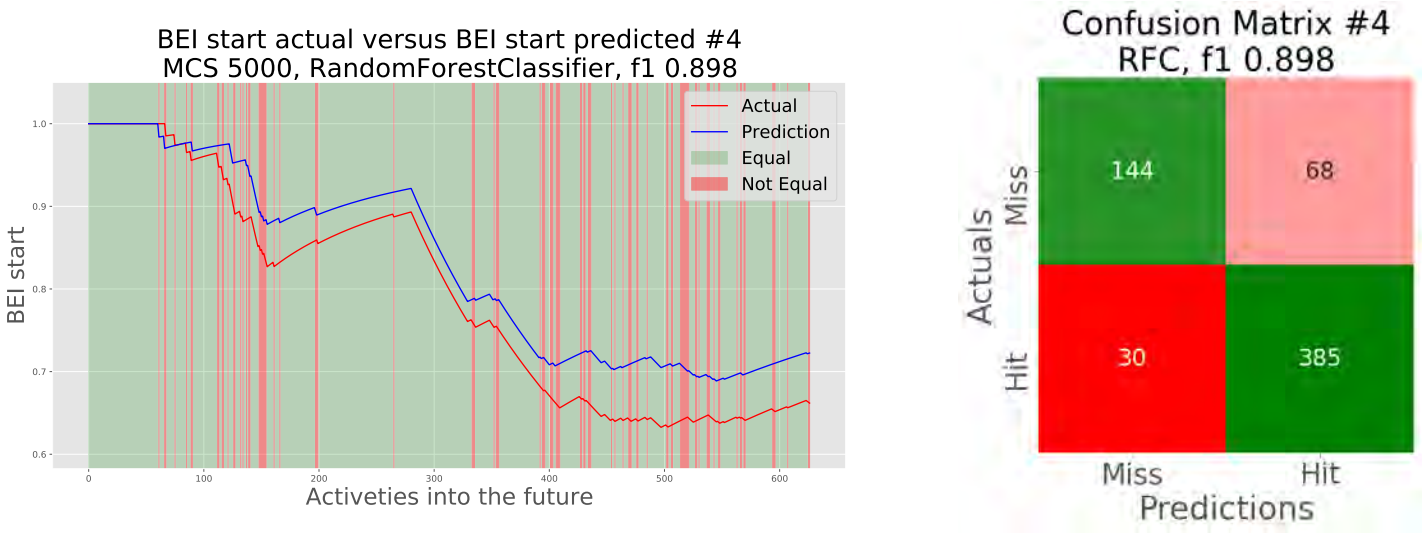
As shown in Table 4.5 which is for a split of 0.83, the feature set providing the lowest RMSE score looks to be the descriptions. The feature of each description is shown in Table B.1 by the code "des_keyword". As mentioned in Section 3.4.1, the description field is manually inserted by employees of the company. Interestingly, the descriptions are the essential features for the model when the split is late. Therefore, the project management team can describe each activity early in the project to obtain a forecast of the project finalization. This could be of help early in the project to indicate the shape of the EV curve. When the shape is obtained, it can be used by the project management team to share resources and techniques, which is argued to help in delivering projects ahead of schedule and under budget [20]. The RMSE score of the results in this thesis might be too high for the model to be trusted in practice. However, the implementation shows which features are the most important in a late split and how they could be of use if the RMSE score decreases.

5.3 Company B - Classification

The main idea behind the secondary analysis of Company B was to see what was possible with a single detailed dataset and shed some light on which features of the dataset that was the most important to the models. It is not given that the regressive analysis using the LSTM model takes advantage of the same features during its training as the classification model. However, it may uncover which features that an ML algorithm will want to use when making predictions. The chosen features are often descriptive and diverse, so the model can learn that, for example, a high value in the scope features gives a higher chance of a start hit or miss. Then, it is up to the analyst to discuss if the high scope value indicates an essential and extensive activity, and thus, that the label is a hit. Alternatively, if the high scope value makes the model learn that they often are delayed, i.e. miss since it is not that important to start on time since the scope is so considerable. As mentioned in Section 5.2.2, there are some activities with the scope features of 0.001. Of these 189 activities, there are about 31% that have been classified as a start hit. Thus, thee split

between hits and misses in this subset is quite different from the dataset as a whole. From Table 4.6, this split is 65%, which may also have been a reason why the SHAP values of these scope features were so high. For example, if the model is to predict an activity that has a Tsh value of 0.001, the model has learned that 69% of these are a miss. Since there are fewer misses than hits in this dataset, a consistent way to classify the misses is excellent learning of the model, which is paramount.

Further, in Figure 5.4a, one can see the actual versus the predicted BEI start score over the activities of the project. The rest of these plots for each of the six main models are plotted in Figures B.14 to B.16. The areas with the green backdrop are where the predicted class is equal to the actual class, and thus, the model has correctly classified the activity. The red backdrop highlights the activities where the model predicted wrong. The plotted figure is from the best performing model, Model 4 from Table 4.8. In the end, the actual BEI score is 0.65, while the final predicted BEI score is a bit higher, at 0.72. This states that the model predicts more hits than misses, and thus, is more optimistic than the real world. This comes as a result of a someway wrongfully weighted model. From Table 4.6, the percentage of actual start hits is 65%. Thus, the model gains by predicting hit more than if it had predicted miss since the chances are greater for a correct prediction. This can also be seen in Figure 5.4b. This is the corresponding CM to the aforementioned Model 4. The high F1 score is because of the high numbers in the main diagonal compared to the low number in the off-diagonal, as presented in Section 2.3.8. By comparing the number in the columns for hit and miss in the predictions and summate them, one finds that the model predicts hits about 73% of the time. The CMs for the other models can be found in Figures B.17 to B.19.



(a) BEI. **(b)** CM.

Figure 5.4: Mean binarizer BEI and CM plot from 5000 MCS of RandomForestClassifier 2, **shuffle = True**. F1 = 0.898

From Table 4.8, it seems that the RF models perform the best when the feature set is whole or as informative as possible. While the XGB models perform better when the main hitters are removed. This is the case in the "Codes", the "Description", and the "All except scope" feature sets. Further, the "All" feature set got the highest overall F1 score of 0.882. This is also the set that yields the highest mean score of 0.836. Further, it is worth noting that Model 1, the first XGB, has three out of five possible best scores per feature set. It is also close behind on second place on the Codes features set, only 0.004 behind Model 3.

Tsh and the label have only a Pearson's correlation r of 0.147, which is a low correlation. Thus, there must be an inherent attribute of the Tsh that makes it so important to predict the start hit class. Therefore, a more thorough analysis was performed. Since Tsh is a number describing the scope estimate of the activity, the mean of the feature was subtracted from each instance to give a reference point for comparison reasons. Then, this number was compared with zero to find if the activity's scope was bigger or smaller than the mean. This was then again compared with whether or not the start hit class was classified as a hit or miss. The results are illustrated in Table 5.1. From the table, it can be seen that if the Tsh is over the mean, there is about an 83% chance that it is a start hit, and if it is under the mean, it is about a 63% chance that it is a start hit. From Table 4.6 the hit percent in the dataset is 65%. Thus, if the Tsh of an activity is over the

mean, there are 18 percentage points to gain for the model when predicting a start hit. If the Tsh is under the mean for an activity, only a two percentage points are lost when the classification actually is a miss. This reasoning shows one of the reasons why the model would choose this feature in its learning. At fourth and fifth place, the duration and the feature regarding main area equal to general were assigned the same SHAP score. The description keywords with the highest SHAP score were cost and estimate. To get a sense of the accuracy, Table 4.8 may be conferred with.

Table 5.1: Number of times the Tsh feature was over or under the Tsh mean, compared to the times the activity was a hit or a miss.

Tsh	Start Hit	Start Miss
>Mean	87	20
<Mean	328	192

In Figure 5.5, the weighting of the feature importance according to the SHAP values is plotted of the first XGB model on the feature set of descriptions. Table 4.9 shows the mean F1 score of this model with this feature set to be 0.814. Interestingly, this model performs well on a feature set the other models have difficulties with. The one-hot encoded keywords from the activity's description show mainly one feature determining the start hit class. That is if the description contains the word "Cost". To find out if this cost feature points to the fact that if it is activated, i.e. equal to 1, yields a hit, requires further analysis. As with the Tsh feature, a more in-depth analysis will be performed. In Table 5.2 the number of times the one-hot encoded "des_cost" feature was activated versus the start hit classification is shown. If the keyword "cost" is mentioned, it is over ten times as likely that the activity is a hit. While if it is not mentioned, there is still about a 63% chance that it is a hit. Then, one might compare these number with the percentage of hits in the project to see the possible gain of weighting this feature heavily, as shown in Section **??**. In addition, the number of affected activities is relatively high compared with another feature that not is weighted heavily. For example, the feature "des_phase" has only six instances where it occurs, which can be found in Table B.1. Thus, it will not be a good feature to split the dataset on. This may help in edge cases, but the summation limit proposed could have been set higher in hindsight. Further, from the plot, three other features have an impact on the classification. That is if the description contains the words "estimate", "initial", or "project". All these features affect 52, 67, and 20 activities,

respectively.

Table 5.2: Description keyword "cost" versus start hit classification.

Cost	Hit	Miss
Mentioned	62	6
Not mentioned	353	206

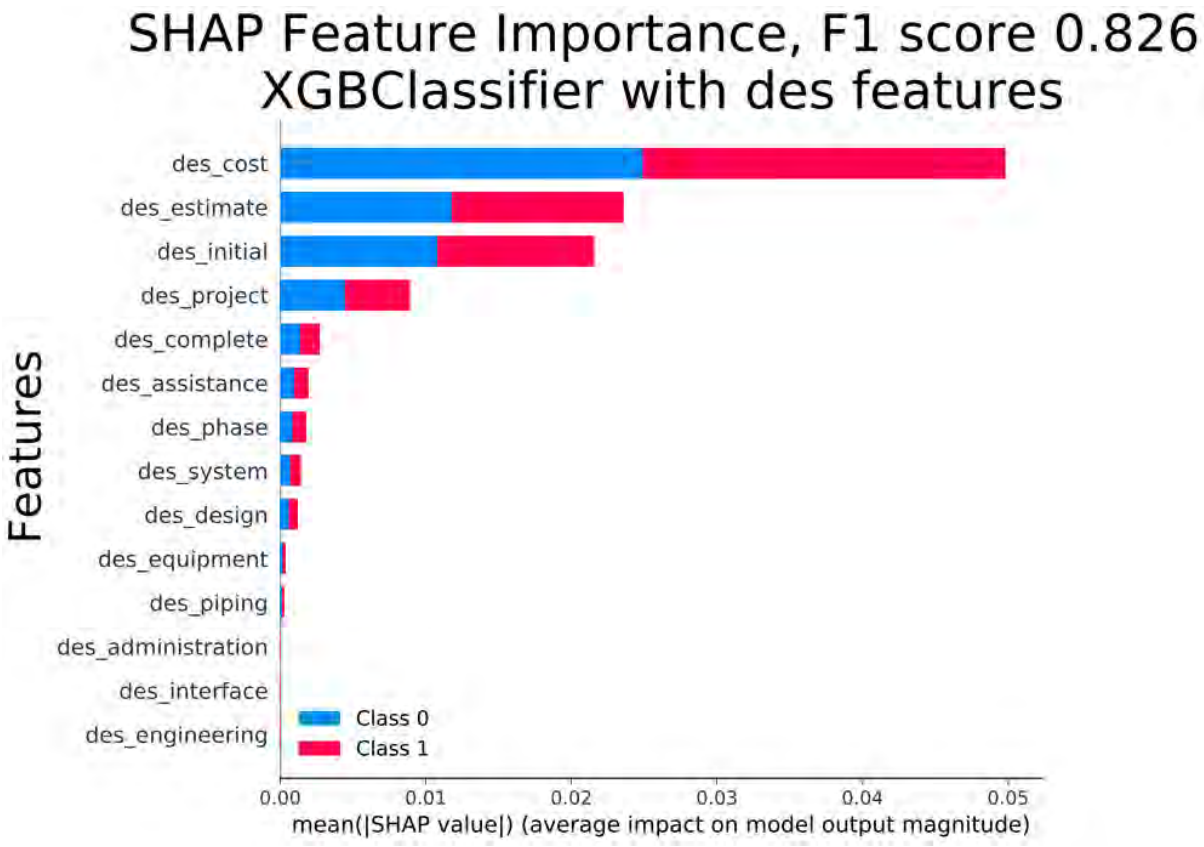


Figure 5.5: Plot of the SHAP values of Model 1, features set "Description".

In Figure 5.6 the first XGB model is plotted for its SHAP values regarding the feature set "All except scope". There is yet again one feature that the model weights heavily, the "main_area_General" feature. The meaning of this feature is that the work done is related to the project in a general perspective, not to a specific component of the physical product. It has an average impact on the model with a magnitude of almost 0.15 in the SHAP value. This is about three times as much as the runner up. Thus, this feature is three times as important to the model as the duration feature. The duration feature states how many weeks the activity is scheduled to take. This feature is correlated with the scope columns with a r score of around 0.5. The duration feature and the third most important feature are both from the the feature set of Numbers. However, as the feature

set "All except scope" entails, only the three features directly connected to the scope, i.e. the Tsh, Rsh, and csh was expelled. Then, two activity role features regarding document processes are essential to predict the start hit class.

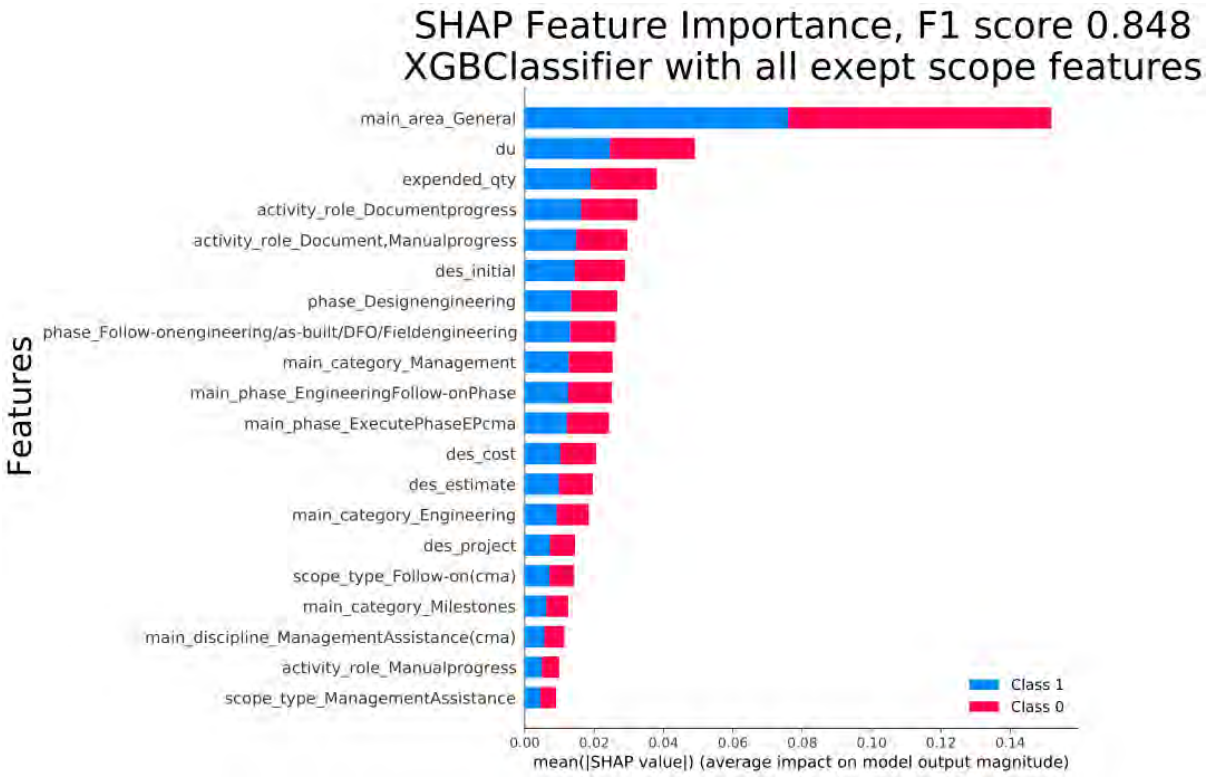
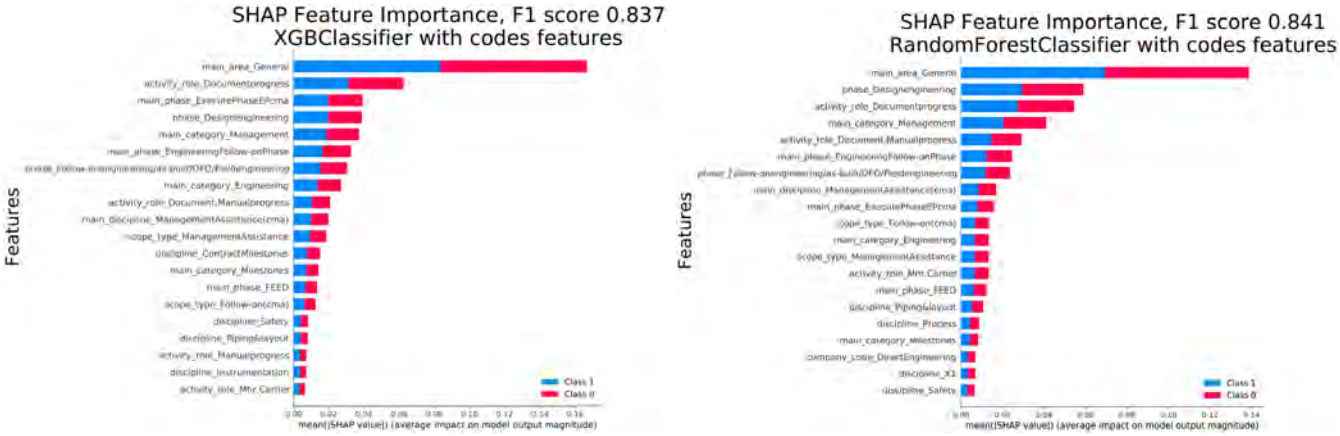


Figure 5.6: Plot of the SHAP values of Model 1, feature set "All except scope".

In Figure 5.7, Model 1 and 3 are compared regarding the feature set Codes. In Table 4.9, Model 3, which is an RF model, achieved slightly higher accuracy than the XGB model. Visual analysis uncovers that they both weight the "main_area_General" feature the most. Model 1 puts the weight at 0.165, while Model 3 puts it in at around 0.14. Further, the top ten features of Model 1 is among the top eleven of Model 3. The biggest rank change is the feature that Model 1 deems the third most important, the "main phase Execute Phase Engineering Procurement Company Management Assistance". Model 3 puts this in at rank nine. Other important features to both the models are the "activity role document progress" feature, the phase of design engineering, and the main category management. These minor differences are what may cause the RF model to outperform the XGB model slightly. However, as we saw in Table 4.9, these differences are minor and may also disappear if the MCS were further increased. Further, it is the ensemble classifiers that perform the best on this dataset's feature sets. In Table 4.9, none of the

best scores per feature set, i.e. underlined values, are in the MLP columns. As presented in Section 2.3.7, these are based on neural network, while the two others are based on decision trees ensembles.



(a) Model 1. **(b)** Model 3.

Figure 5.7: Plot of the SHAP values of Model 1 and 3, feature set "Codes".

5.4 Further Utilization

For a model to have value, it must be able to aid the user to solve a problem. The models shown in this thesis are built so they can be used later on. The end goal is to predict or classify the new data the companies will acquire in the future correctly. Figure 5.8 shows an illustration of a proposed use case for the models. The upper flow represents the work done in this thesis by creating the models and feeding data from previous projects, which has resulted in a trained machine learning model. Then, a company, a project management team, or a user can take advantage of the trained model to predict or classify their new data. One example is to use it on data from a project they are working on, and the goal is to predict the following week's EV. Another can be to classify which activities will be start hits or misses based on data containing the plans for the coming months. If the model classifies an activity to become a miss, the user can investigate the reason and take actions. In this way, the model can be looked at as an expert system powered by machine learning [191; 192]. Expert systems are often described as models that can provide suggestions or tips on a variety of tasks [21]. So, instead to confer with an expert, an expert system can be "asked" for advice.

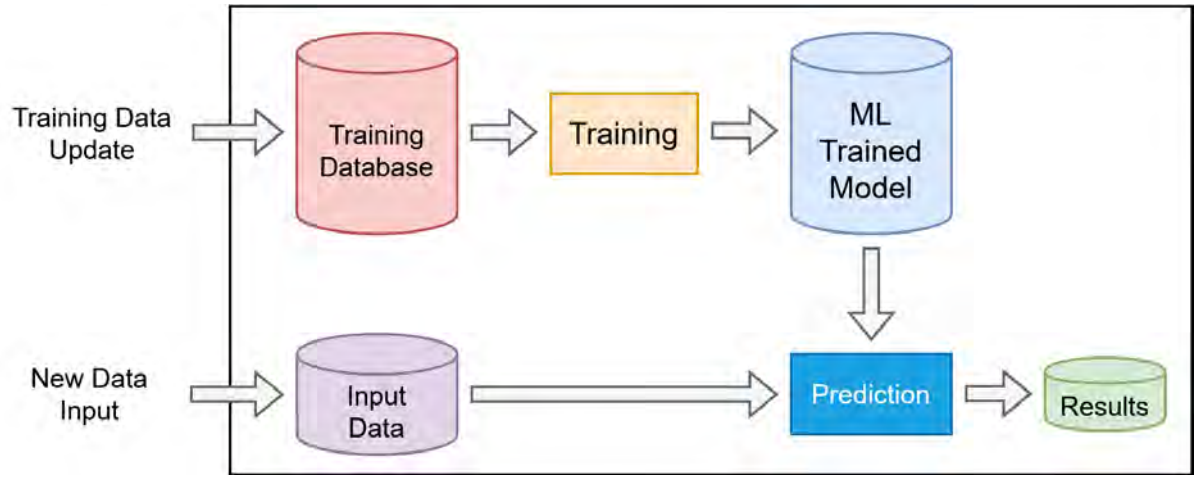


Figure 5.8: How a user later can use the trained model this thesis proposes. Adapted from [193].

5.5 Handling Different Datasets

This thesis has analyzed two different datasets from two different companies. The nature of the projects are very similar, but as Sections 3.3.1 and 3.4.1 presents, the datasets are far from similar. Several interviews with the representatives were held to get a grasp of the embedded information in the datasets. This was the most time-consuming part of the thesis. From the beginning, the energy industry was a new industry to both of the authors. Therefore, understanding the information and the most important aspects of it required time to comprehend. Meetings with the representative were held once every two weeks to fit their busy calendar and to gather new questions to be answered. Secondly, having to focus on three independent analysis proved to be more difficult than first considered. To acquire domain knowledge on one dataset is time-consuming, and doing this for three processes simultaneously, makes it more complicated. For instance, the process to delegate equal amounts of time between the datasets resulted in frequent changeovers. According to LEAN, frequent changeovers is not desired when a changeover requires time to adjust to the new sets [194]. This applies to soft projects like machine learning as well. In hindsight, a process to delegate resources to one company and then the next could provide a more structured process.

The interviews with the representatives revealed that a fast processed approach to machine learning was satisfactory. If it would be possible to have a quick deployment, machine learning could be applied during a project to provide predictions. However, because the

model was quickly applied, the predictions might turn out to have low accuracy. The low accuracy was acceptable for the representatives as it could explain the magnitude of the value in the future. Nevertheless, the interest to deploy a fast process aligns well with the lack of time associated with a master's thesis. The latter should have a fast and narrowed process to obtain results in a short time horizon. A fast approach to machine learning means that there is limited time to clean, preprocess, and adjust the hyperparameters to achieve the optimal configuration. Instead, the desired approach is to try a low number of values based on an educated guess or based on similar research. Early stages revealed limited research for applying machine learning on follow-up data or data acquired from a project management software. To determine the number of epochs, a more thorough approach could be to start at a specific value and increase it until the training and validation loss stops to decrease or until the error score of the predictions starts to increase.

Further, as the results regarding the best hyperparameters came in, a paper regarding the tuning of these in LSTM modelling was uncovered [182]. This paper by Reimers and Gurevych states that the following hyperparameters are a good starting point when searching for hyperparameters regarding datasets of the magnitude in this thesis. As optimizers go, ADAM and NADAM are recommended. This aligns powerfully with this thesis' findings as well, as seen in Section 3.5.3. Regarding the number of nodes, the cited paper states that good numbers are 60 to 120. With the apex around 100 nodes, or recursive neurons as it is called in the paper. Table 4.1 shows values that align well with the findings of Reimers and Gurevych. Next, Reimers and Gurevych state that a mini-batch size of between 8 to 32 could be an optimal place to start a grid search. From Table 4.1 this also aligns up with the best configurations of this thesis. Lastly, the cited paper weighs in on the discussion of Section 3.3.4 regarding the deep versus wide machine learning debate [173]. If there are two or more layers in a model, it will classify as a deep model. The paper concludes that a two-layered LSTM got the best results four times out of five compared to LSTM models of other depths. This was equal to the findings in the exploratory data analysis of this thesis as well. However, it also stated that a variational dropout is the best. In this thesis, a naive dropout has been utilized with a value of 0.2. Thus, 20% of the nodes in each epoch will be dropped. Further research could implement this variational dropout function on a model. Overall, the best practices of the cited

paper align quite closely with the findings of this thesis.

The data from Company A have logged the number of hours and materials used. In contrast, the data from Company B have logged information used by the project manager of each activity during the project. Even though Section 3.2 presents a unified method to apply machine learning to projects, the actions to clean and preprocess the dataset were different. For the data from Company A, the training and fitting of the model proved to be the most important step in the analysis. The data were already aggregated weekly and cleaned only to include information of interest. Therefore, the following steps were to check for completeness between the projects and sort out the most important ones as input to the model. However, as Section 3.3.2 shows, a high number of features were dropped as they contain a high number of missing values. To implement a fast approach to a complete model, it was decided only to use the features completely filled in all projects. However, continuing interviews with the representative revealed that some of the gathered features only reflect the project from 60% complete until it is finished. This might be some of the cause of the low performance of the model. To overcome the problem of the low performance, the planned and actual hours in each discipline could be obtained so they could be included.

For Company B, much of the difficulties lay in which data to extract and how to aggregate it to a dataframe as input in the model. As mentioned in Section 3.4.1, several interviews with the representatives were held to acquire domain knowledge on which information to extract. However, machine learning on time series data needs sequential data in its input. To have sequential data, it was determined to use the starts of each activity. This decision faced several challenges. First, the number of rows was reduced from 13047 to 627, giving the model fewer data points to train with. Secondly, later analysis showed that a high number of activities starts on the same date and that there can be several weeks between the start of an activity and the subsequent activity. Thirdly, using the information at the starts of the activities might be insufficient when using the EV as a label.

5.6 Is the Data in the Companies Ready to Support Machine Learning?

One of the research questions in this thesis was to investigate the readiness for datasets from Company A and B to support machine learning. Therefore, large parts of the analysis in this thesis investigated how to extract data, the inherent information, and how it can be formatted to support machine learning. In most machine learning projects, the EDA and preprocessing is often the most time-consuming parts [89]. The datasets from Company A is of a template that is more ready to support machine learning than that of Company B. For Company A, interesting features are logged, and the template is identical between the projects. As presented in Section 3.3.1, the features are not completely logged between the projects. Therefore, the challenges for Company A would be to systematize the incoming flow of information so that all projects contain the information the template is intended to store. At the time of writing this thesis, there are only a few features that are logged complete and correct between the projects from Company A. First of all, Company A should collect these features into one uniform database. Further, a few correct features are much better than many incorrect or partly filled in features. Therefore, they should log the few correct features from their other projects to increase the proposed project management database. This enables comparisons to be made between projects. This will provide the model with more training and testing and will make it more robust.

The project management software used by Company B contains the wanted information. Opposed to the challenge of Company A, the challenge in Company B were to acquire knowledge about the software and which features could be significant. Because the software is used during the execution of the project, there have been certain workarounds implemented for practical reasons. For instance, the duration of an activity could be changed because it proved to be longer than initially assumed. This resulted in an ESA to be before the project start. Another workaround is the values of 0.001 in scopes. In the worst case, the workarounds may make the information irrelevant. Datasets with a high degree of irrelevant information are stated to be one of the challenges in implementing machine learning [89]. The challenge to make the inherent information ready to support machine learning stems from the intended use of the software. It is set up to be practical for the project management team, such as scheduling activities, calculating essential

performance measures, and performing risk analysis. To make the data more ready to support machine learning, changes could be done to enhance the structure for stored data. This thesis revealed inconsistencies and the possibilities that are embedded in the software. Consequently, the information is integrated into the software but not in the wanted format and across several tables through SQL. Therefore, the work required to make the datasets from Company B ready to support machine learning is more prominent than for Company A.

For machine learning to provide accurate and precise values, the goal should be reflected in the input data. For Company A, the data most completely logged features were the expended values, not the planned. The values for planned progression, like planned value, should be logged to tell how the project is performing compared to the plans. The results from the analysis on Company A tell at which values the expended value will be, but it does not tell how it performs compared to the planned value. For instance, the project used for testing has a peak in expended value over 3%. However, there is no information if this were according to the plan or is caused by, for example, extra resources delegated to the project to meet a deadline.

5.7 Correlation

The correlations in the Appendix are based on Pearson's r [179]. If a feature has a high positive correlation to the label, it will mean that when the feature increases, the label also increases. If the correlation is negative and the feature increases, the label will tend to decrease. The correlation plots are interesting to analyze since they may give some insights into which features the models use to predict the label. For Company A, there are three Figures A.5, A.6, and A.7 are for Project 1, Procent 2, and Project 3, respectively. For the first project, the top three correlations are positive. Only the most correlated have a r over the threshold of a moderate correlation set at 0.3. This is the man-hours of the mechanic discipline. What is interesting is that some features, like the module erection, are negatively correlated with the label. Thus, when the expended value increase, the module erection decrease. Project 2's figure shows that the top six most correlated features are positive. Furthermore, all these are over the moderate correlation threshold, with three of them over the strong correlation threshold of 0.5. Due to the limited dataset

and that the correlated features often are dropped when there is no need to, the features were kept [195]. Thus, it would seem like these features are highly important to the model. Further, this project's correlation analysis states the opposite of what the first project's analysis stated. For Project 2, the daily manpower in the mechanic discipline is negatively correlated to the expended value label. This is also the case for the third project. This raises the question, why? From Figures A.8 to A.10, the values of each feature is plotted over the length of the project. Further, as presented in Section 3.3.3, the feature "Holiday" is a custom feature that was put in during the prepossessing phase of the analysis. It is interesting to see that it is correlated to the expended value. This was thought to be the case from the interviews with Company A, but this plot confirms it. In all three projects, this feature is among the top correlated features. However, none of the projects has the correlation r of the Holiday feature higher than a moderate positive correlation compared with the label. This strengthens the foundation of the idea that more custom features may yield better results, as mentioned in Section 5.1.2.

5.8 Limitations

There will always be limitations to a model trying to simulate and predict the real world. This pilot project has met many of them. A more extensive and more robust model and project will often perform better, i.e. be more accurate, but comes at a premium regarding scope, time, and cost. In terms of limitations, some are global for all analysis. These will be addressed first. Then, analysis specific limitations will be addressed.

5.8.1 General limitations

Perhaps the main limitation of this pilot is the available data in the industry. The scope is limited to what data is possible to receive from the industry, what the industry wants to share, and how structured the gathered data is. Further, a master's thesis is often associated with limited time to explore other solutions or repeat several steps to fine-tune the method. For this thesis, the small size of the datasets was no problem with regards to computational costs. The longest time for a model to run was 18 hours. With these sizes of the datasets, the project management can apply machine learning and fine-tune the hyperparameters without considering the time to run the models. However, companies can gather data orders of magnitude larger, especially with the introduction of IoT. Pecholcs

et al. state that sensors used in seismic surveys routinely acquire up to six terabytes of data [95]. At this point, the companies should consider investigating more efficient algorithms not to make the costs severe [196].

In terms of the dataset themselves, the small size of them posed a problem. Additionally, the length was limited compared to the number of features. This stems from the fact that these datasets are collected from real-world projects no longer than five years. However, with increased granularity, i.e. daily numbers instead of weekly, the LSTM model could have been more accurate and may have learned more of the more detailed aspects of the task at hand. However, from the interviews with the representatives to have a resolution on a daily basis is not of high interest from a project management perspective. An activity is considered to be on time if it starts or finishes in the same week as planned. Further, the EV is also generally used with a weekly time horizon.

Since the industry yet have fully systematized how to store project data to support machine learning, the datasets came to have a large percentage of missing values. The data structure is one of the main challenges companies faces today [12]. This leads to the necessity for increased emphasis on cleaning, filling, and compiling the data. The lack of completeness can make the model less accurate and unable to reflect the real world.

As the search for the optimal hyperparameters is an exhaustive search, the time allocated to this task becomes a limitation due to the scope of the thesis. Given more time for this step, other potentially better parameters could have been found. However, this pilot wanted to investigate what was possible to uncover in a more limited time frame. This aspect was also pointed out by the companies the datasets came from. They wanted to know how demanding the implementation of this new way to analyze project data would be. This time constraint was also a factor when using the AutoML algorithm. As presented in Section 2.3.3, a grid search is utilized in this method to search through different classifiers, preprocessing steps, and the hyperparameters themselves. Similarly, the number of increments in the preprocessing step was tried to hold to an effective level. Preprocessing activities such as advanced binarizers, feature selection and grouping, and kernels were not included in this thesis. In a more in-depth implementation, these increments could be utilized to increase the accuracy of the models. However, some of these preprocessing tools may increase the difficulty of getting the logic behind the

decisions of the machine learning algorithm out to the analyst. One example of this was when the AutoML found a model with a binarizer and a Nyström kernel. Here, the feature importance was hidden behind grouping and aggregations. Advanced backwards engineering had to be utilized to extract the feature importance. Thus, this method was outside of the scope of this thesis. In addition to that, given the companies wishes for a fast processed approach that is easily explainable, this decision had the necessary top-level support.

5.8.2 Company A limitations

The sizes of the datasets in Company A vary from 97 rows to 171 rows, with the number of features being 13. The length of the datasets might be too small for the LSTM to learn the dependencies and relations in the data. However, there are only 13 features the model can learn from. For the size of the data, an ARIMA model could prove to be a better fit. On the other side, ARIMA is argued not to be a good fit for data with large fluctuations from row to row [169]. Thus, the decision was on an LSTM as it outperforms RNN [145]. The practical implications of this decision are to utilize a model intended for large quantities of data to be used on datasets with row lengths of under 200. The limitations are for this thesis and are considered to be overcome if Company A structures their data and include more projects.

As mentioned in Section 3.3.5, the number of features is reduced to 13 from 484. Multiple interesting features were dropped in the cleaning and preprocessing because they were not filled, either in all or one of the projects. The resulting features represent five disciplines, which are argued to be the most important for the project, based on the interviews. However, there might be bias in how the company weighs the importance of the disciplines. As a rule of thumb, the model should have input features representing multiple aspects so the model can identify the most critical features on its own. Thus, the model is limited as it only includes five of the ten disciplines. Additionally, each discipline should have enough features to represents its characteristics in the project. For instance, there should be features for consumed man-hours and features for material units and the number of tasks done to reflect multiple aspects of the discipline for structure discipline. Ultimately, this could sum up to 40 features, of which there are four for each discipline.

5.8.3 Company B limitations

A custom function that counted the number of activities that started per week was written. The same was done for the finishes and the number of ongoing activities each week. However, these functions proved unstable because of the before mentioned date features updates, noted in Section 3.4.1. Thus, these custom functions were not included in the final model. If these numbers had been found and showed the correct data, it could prove fruitful for the model. These features would then state something about the weekly workload in another way than what is captured by the scope features.

For the dataset from company B, one might dig deeper into the databases to manually export interesting data points. This thesis mostly used SQL, but there is much more information in the project management software that can be easily exported. One example of this is the calendar feature in the software. This may include team size, efficiency scores of teams and tasks, degree of importance to the project. Other features that may be gathered to a more full data mining are actual float and if the activity is in the critical path.

As mentioned in Section 5.5, the dataset contained the progress of each activity. This could have been utilized in another analysis where the target was different. Further, the dropping of the weeks with no started activities contains much information regarding the overall progress. See Section 3.4.3. For example, if the model were to look at the project's overall progress and not the subsection of start activities, these periods of no started activity would contain valuable data. However, for the analysis in question, these points in time are not applicable, and the need to reset the time axis was necessary.

6 Conclusion

The first research question addresses how project plans and follow-up data can be formatted to support machine learning and which accuracy is reachable today. The progress data from Company A were structured according to a specific template defined by the company. The features in the template are ready to support machine learning and have the potential to contain important information as it consists of 484 features. The number of input features to the model was reduced to 13 and contained valuable information. However, Company A is recommended to enhance the process to receive information on all features in their template. Alternatively, create more uniform features that are easier to use across a wide range of project types. Then, these projects should be saved into a project management database for storing. Creating a database will further enable easy extraction of the data to be used as input to ML models. In this thesis, the difficulties were identifying which features were of the highest importance, the physical implication of them in the project, and which configuration is the best fit for the data. Many potential features with interesting information had to be dropped due to the lack of consistency in the data. Among the 13 remaining features were man-hours expended in the most critical disciplines and material units expended in the disciplines. Based on these input features, the model tightly predicts the data's overall trend but suffers in predicting the graph's spikes. Based on the Root Mean Squared Error (RMSE) of the best performing configuration, the model successfully learns from previous projects to forecast the following week's expended value. The model could achieve higher accuracy and precision if it acquired more projects to train on.

For Company B, the project management software utilized during the project contains information on each activity and relations between the activities. The amount of information depended on the activity; if it was done in-house, it had up to 30 data points. If not, it had under ten data points. Additionally, specific values that should be constant changed sporadically later in the project. Thus, the information in the software was not consistent. Consequently, the most time-consuming part of implementing machine learning was the Exploratory Data Analysis (EDA). One interest for Company B was to use the model to predict future values for the project, for instance, the next week's earned value. To do this, the activities had to be aggregated on a weekly basis. The tables gathered by SQL

had information of when the activity started, the logged progression in the subsequent weeks, and when it was finished. It proved difficult to include all this information when aggregating. The decision was to only use the information of the start of the activities, and thus, information regarding progression and changes were lost. However, it could be included in the model if there were more time to investigate other aggregations and more possibilities for extraction. Thus, the information is embedded in the software, but the best extraction and aggregation are still to be uncovered.

To use machine learning on the project from Company B proved to be more difficult than for Company A. Much of the reasons lie in the task to get familiar with the new software, comprehend the wide-ranging information, and how to extract it to a spreadsheet format. Thus, the work done in this thesis is just a minor part of what could have been done with more time. If the project management software made minor tweaks to the way of storing and structuring the data, this hurdle could become much smaller. Ultimately, the accuracy and precision of the model were not as satisfactory as the results of the analysis on Company A. For Company A, the best model achieved an RMSE score of 0.0044, which is less than a week with low production. The RMSE score was reduced by 37% by training on two projects instead of one project. For Company B, the best model achieved an RMSE score of 4290, representing a week with small production. The mean EV throughout the project is 6357, and thus, an RMSE score of 4290 yields too high uncertainty for practical use. Based on the analysis on Company A, the model on Company B could also achieve higher accuracy if it trained on more projects. The best accuracy from the classification analysis of Company B was achieved by a RandomForest model. The F1 score was 0.898, which is a higher accuracy than a no skill classifier.

The second research question addresses which ML techniques are related to these types of data. There are many methods to utilize for time series analysis in machine learning. This thesis implemented the sequential model Long Short-Term Memory (LSTM). The model was implemented with two layers and 62 nodes. StandardScaler proved to be the best fit for the data from Company A, and MinMaxScaler proved to be the best fit for Company B. The number of epochs should be 2 for the datasets from Company A. For Company B, the number of epochs were set to be 100. There is not enough information to determine a most fitting value for the number of batches for either company. For the

value of n_past, the value of 3 fits the frequent fluctuations while the value of 6 fits the trend more accurately. For practical use, the project management team should do the grid search analysis early in the project. Because the projects constitute of distinct phases, the model should train on complete projects to learn the characteristics of each phase. Ultimately, the model delivered adequate accuracy and precision.

The classification analysis of Company B yielded insights and exemplified how a more detailed dataset can be utilized. The example in this thesis revolved around the classification of start hits used to calculate the Baseline Execution Index (BEI). Thus, start hits became the label of the model, but the label is easily changed to another wanted metric. Results regarding hyperparameters, classifiers, and methodologies were shown. Further, SHAP values for the feature subsets showed that the scope features were the most important of all the features for the model. Among the code features was the "main_area_general" feature the most important. As for the description's keywords, "cost", "estimate", and "initial" were the most important. The BEI score was predicted too optimistically since the model weighted the start hit class more heavily.

Finally, the third research question addresses the implication for the project management. For the project management team, forecasts powered by machine learning can complement traditional methods. The intended use of the time series analysis is for the companies to insert information of plans and schedules to measure what the actual values can be throughout the project based on knowledge gathered from earlier projects. For both time series analysis, the model has achieved what may be considered as adequate performance considering the small datasets. However, the time series model performed better on the dataset from Company A than from Company B. First, higher certainty in the forecasts could be of use in scheduling the activities and adjust for the peak of the project. With increased digitization and IoT implementation, there will be more information to make the model more accurate. Secondly, the information in the forecasts can be shared to strengthen the collaboration between operators and suppliers and to better transfer knowledge. Increased sharing of information and techniques is argued to reduce the amount of failed projects. Thirdly, the forecast will aid the project management team to more accurately know at which levels the EV will be throughout the project. Lastly, for a project with a long duration in which early forecasts will be inaccurate, the model

could receive live information to indicate the status at the last phases of the project. On-time delivery is one of the three sides of the project management's iron triangle. The B2 classification stated which features are best to analyze to decide whether an activity will be a miss or a hit, i.e. a delayed start or not. The time series analysis, A and B1, relates to another of the three sides: cost. It was concluded that these analyses powered by machine learning could be used to reduce the forecast error.

7 Further Research

This thesis proved significant in its scope, containing data of different sizes, sectors, resolutions, and targets. Thus, this thesis may be considered a stepping stone to further analysis. The implementations and adjustments of such models can continuously be enhanced and deeper analyzed. In this chapter, some suggestions for further work will be put forward. These might include model-specific tweaks, general changes, or new areas of interest to increase the possible knowledge pool of machine learning within the field of project management.

A more in-depth sensitivity analysis could be performed by being selective of which features to include. Based on the correlation plots, there is a correlation between the features and the label and among the features themselves [195]. Another sensitivity analysis could be done similar to the one in this thesis' regarding feature sets by taking away one or more features from the total feature pool and analyze the model's performance [178].

For Company A's datasets, a sensitivity analysis could test if the project's order has something to say regarding the model's accuracy. In this thesis, the training was performed on Project 1 in the first chain and on Project 1 and then Project 2 in the second chain, and both tested on Project 3. One example is thus reversing this sequence so that, for example, Project 3 and 2 are the ones in the dataset training pool, and Project 1 is the one being tested on. Another interesting sensitivity analysis that could be conducted is to remove the augmentation of the datasets. Instead of augmenting, the same total number of epochs has to run to be able to compare. Each augment will run through the number of epochs, and thus, the count for training on the same dataset is to multiply the number of augments, the number of epochs, and the number of chains. Additionally, the model will train on the original dataset with the number of epochs at the initiation. When it comes to augmenting methods themselves, more methods are available. However, the methods used in this analysis were considered to be enough for this pilot model. Still, more advanced augments may have yielded different results and could be further researched.

Another aspect that could be analyzed is how the models would have been different based on other depths and widths in the layers than what was used in this pilot [173]. Some node-width hyperparameters were tested, but the effect of more layers was only implemented during preliminary testing. However, Reimers and Gurevych state that two

layers may be optimal in these kinds of LSTM analysis [182].

For the analysis of Company B, the effects of the baselines were not studied. As mentioned in Section 2.2.1, a new baseline may yield significant changes in terms of the scope of the project. In Figure B.9, the six baselines of the project from Company B are illustrated. One method could be to add a feature for the baseline, like the "Holiday" feature in the dataset for Company A. Another method, which could prove to yield a higher resolution, could be to have each baseline interval as input in a sequential manner. If a model had been written to analyze each baseline, a different dimension would have been retrieved from the data. Then, the model could learn the differences between each baseline and compared them. Additionally, a split of the data on each baseline could further utilize the chained method shown in the analysis of Company A's datasets. Lastly, the inclusion of quarter end dates could also be implemented. Near quarter ends, companies may be willing to put in extra time and energy to boost their quarterly numbers to appear stronger [197].

The expended value, the earned value, and start hits and misses for the BEI have been implemented in this thesis. However, there are several other interesting metrics to be used as the label of models, such as the ones presented in Section 2.2.4. In addition to the standard indexes, metrics like the Critical Path Length Index (CPLI) could be rewarding to analyze [198]. The equation is shown in Equation 7.1.

$$CPLI = \frac{Critical\ \ Path\ \ Length\ \ +\ \ Critical\ \ Path\ \ Total\ \ Float}{Critical\ \ Path\ \ Length} \tag{7.1}$$

This index is favourable if it is over 1.0 and is a metric for how effective it is to complete its critical activities on time. As BEI start was used in this thesis, BEI finish could also be found and compared with. The two BEIs are plotted in Figure B.13. It was chosen not to pursue this idea any further in this model.

As mentioned in Section 5.5, a variational dropout function could have been implemented. There are indications that this could make the model able to perform better on the given datasets [182].

Further, the MCS-based mean binarizing method can give valuable insights into each activity. If one were to stop before the binarizing step in Figure 3.10, one could see the

average prediction of each activity. If this average either is close to 1 or 0, it will mean that there is consensus among the models in the MCS that the activity was a hit or a miss, respectively. However, if the mean is about 0.5, it will mean that this activity was difficult to predict since the different models in the MCS predicted differently. Thus, further analysis on these scores could provide valuable insights into which activities are easy to predict, and thus, which features these activities have in common. This could give information regarding how to improve the model or why some activities are difficult to predict. In addition to this, the scores could be used to compare the different classifiers to analyze if there is consensus among them or if one of the classifiers is better at predicting a certain type of activity.

References

[1] M. Haenlein and A. Kaplan, "A brief history of artificial intelligence: On the past, present, and future of artificial intelligence." https://journals.sagepub.com/doi/full/10.1177/0008125619864925. [Downloaded 2th of February 2021].

[2] H. Tanaka, "Toward project and program management paradigm in the space of complexity: a case study of mega and complex oil and gas development and infrastructure projects," *Procedia - Social and Behavioral Sciences*, vol. 119, pp. 65–74, 2014.

[3] R. Michaelides, D. Bryde, and U. Ohaeri, "Sustainability from a project management perspective: are oil and gas supply chains ready to embed sustainability in their projects?," Project Management Institute, 2014.

[4] IEA, "Renewables information: Overview." https://www.iea.org/fuels-and-technologies/renewables. [Downloaded 8th of April 2021].

[5] H. H. Chen and C. Pang, "Organizational forms for knowledge management in photovoltaic solar energy industry," *Knowledge-Based Systems*, vol. 23, no. 8, pp. 924–933, 2010.

[6] K. M. Hanga and Y. Kovalchuk, "Machine learning and multi-agent systems in oil and gas industry applications: A survey," *Computer Science Review*, vol. 34, 2019.

[7] Y. Hajizadeh, "Machine learning in oil and gas; a swot analysis approach," *Journal of Petroleum Science and Engineering*, vol. 176, pp. 661–663, 2019.

[8] M. Kolloch and D. Dellermann, "Digital innovation in the energy industry: The impact of controversies on the evolution of innovation ecosystems," *Technological Forecasting and Social Change*, vol. 136, pp. 254–264, 2018.

[9] IEA, "Average time to market for conventional oil and gas projects, 2010-2018." https://www.iea.org/data-and-statistics/charts/average-time-to-market-for-conventional-oil-and-gas-projects-2010-2018. [Downloaded 8th of April 2021].

[10] Z. Rui, C. Li, F. Peng, K. Ling, G. Chen, X. Zhou, and H. Chang, "Development of industry performance metrics for offshore oil and gasproject," *Journal of Natural Gas Science and Engineering*, vol. 39, pp. 44–53, 2017.

[11] M. Mullaly and J. L. Thomas, "Exploring the dynamics of value and fit: Insights from project management," *Project Management Journal*, vol. 40, no. 1, pp. 124–135, 2009.

[12] D. Aker ASA and Cognite, "Industrielle data." https://digitalnorway.com/kurs/industrielle-data/. [Downloaded 31th of Mai 2021].

[13] F. A. Mir and A. H. Pinnington, "Exploring the value of project management: linking project management performance and project success," *International journal of project management*, vol. 32, no. 2, pp. 202–217, 2014.

[14] M. Sołtysik, M. Zakrzewska, A. Sagan, and S. Jarosz, "Assessment of project manager's competence in the context of individual competence baseline," *Education Sciences*, vol. 10, no. 5, 2020.

[15] F. Edum-Fotwe and R. McCaffer, "Developing project management competency: perspectives from the construction industry," *International Journal of Project Management*, 2000.

[16] A. B. Alotaibi and O. P. Mafimisebi, "Project management practice: redefining theoretical challenges in the 21st century," *Project Management*, vol. 7, no. 1, pp. 93–99, 2016.

[17] M. M. de Carvalho, L. A. Patah, and D. de Souza Bido, "Project management and its effects on project success: Cross-country and cross-industry comparisons," *International journal of project management*, vol. 33, no. 7, pp. 1509–1522, 2015.

[18] H. J. Wilson, P. R. Daugherty, and C. Davenport, "The future of ai will be about less data, not more." https://hbr.org/2019/01/the-future-of-ai-will-be-about-less-data-not-more. [Downloaded 22nd of February 2021].

[19] R. Bean, "The state of machine learning in business today." https://www.forbes.com/sites/ciocentral/2018/09/17/the-state-of-machine-learning-in-business-today/?sh=67f585ec3b1d. [Downloaded 22nd of February 2021].

[20] P. Solutions, "The state of the pmo 2010," *Project Management Solutions*, vol. 4, 2010.

[21] F. Hayes-Roth, D. Waterman, and D. Lenat, "Building expert systems,"

[22] J. Pavlus, "Computers now recognize patterns better than humans can." https://www.scientificamerican.com/article/computers-now-recognize-patterns-better-than-humans-can/. [Downloaded 14th of May 2021].

[23] M. Roser and H. Ritchie, "Technological progress." https://ourworldindata.org/technological-progress. [Downloaded 14th of May 2021].

[24] S. Schelter, F. Biessmann, T. Januschowski, D. Salinas, S. Seufert, and G. Szarvas, "On Challenges in Machine Learning Model Management," 2018.

[25] I. Guyon, L. Sun-Hosoya, M. Boullé, H. J. Escalante, S. Escalera, Z. Liu, D. Jajetic, B. Ray, M. Saeed, M. Sebag, A. Statnikov, W. Tu, and E. Viegas, "Analysis of the automl challenge series2015–2018," *Automated Machine Learning*, pp. 177–219, 2019.

[26] J. Westlands, "The 10 project management knowledge areas (pmbok)." https://www.projectmanager.com/blog/10-project-management-knowledge-areas. [Downloaded 2nd of February 2021].

[27] J. Russell, E. Jaselskis, and S. Lawrence, "Continuous Assessment of Project Performance," *Journal of Construction Engineering and Management*, 1997.

[28] Y.-G. Schoper, A. Wald, H. T. Ingason, and T. V. Fridgeirsson, "Projectification in western economies: A comparative study of germany, norway and iceland," *International Journal of Project Management*, vol. 36, no. 1, pp. 71–82, 2018.

[29] J. Ramazani and G. Jergeas, "Project managers and the journey from good to

great: The benefitsof investment in project management training and education," *International Journal of Project Management*, vol. 33, pp. 41–52, 2015.

[30] B. S. Blanchard, W. J. Fabrycky, and W. J. Fabrycky, *Systems engineering and analysis*, vol. 4. Prentice hall Englewood Cliffs, NJ, 1990.

[31] A. B. Badiru and S. O. Osisanya, *Project management for the oil and gas industry: a world system approach*. CRC Press, 2016.

[32] F.-R. Ahmaduna, A. Pendashteha, L. C. Abdullaha, D. R. A. Biaka, S. S. Madaenic, and Z. Z. Abidin, "Review of technologies for oil and gas produced water treatment," *Journal of Hazardous Materials*, vol. 170, pp. 530–551, 2009.

[33] O. Babordina, M. Garanina, P. Garanin, and E. Chirkunova, "Digitalization and project management method in improving efficiency of drilling wells construction," in *International Scientific and Practical Conference*, pp. 348–353, Springer, 2020.

[34] D. D. Ahiaga-Dagbui, P. E. Love, A. Whyte, and P. Boateng, "Costing and technological challenges of offshore oil and gas decommissioning in the uk north sea," *Journal of construction engineering and management*, vol. 143, no. 7, p. 05017008, 2017.

[35] O. J. Olaniran, "Barriers to tacit knowledge sharing in geographically dispersed project teams in oil and gas projects," *Project Management Journal*, vol. 48, no. 3, pp. 41–57, 2017.

[36] M. Angelopoulos, C. Kontakou, and Y. Pollalis, "Digital transformation and lean management. challenges in the energy industry of utilities. a review," 2019.

[37] T. R. Wanasinghe, L. Wroblewski, B. K. Petersen, R. G. Gosine, L. A. James, O. De Silva, G. K. Mann, and P. J. Warrian, "Digital twin for the oil and gas industry: overview, research trends, opportunities, and challenges," *IEEE Access*, vol. 8, pp. 104175–104197, 2020.

[38] J. Taylor, *Project Scheduling and Cost Control: Planning, Monitoring and Controlling the Baseline*. J. Ross Publishing, 2008.

[39] S. Banker, "Demand planning solutions improve forecasting by consuming more and more data." https://www.forbes.com/sites/stevebanker/2019/04/01/demand-planning-solutions-improve-forecasting-by-consuming-more-and-more-data/?sh=7fd4e4cd3ee7. [Downloaded 8th of February 2021].

[40] M. Shermer, "Why economic experts' predictions fail." https://www.scientificamerican.com/article/financial-flimflam/. [Downloaded 8th of February 2021].

[41] P. E. Tetlock, *Expert Political Judgment: How Good Is It? How Can We Know?* Princeton University Press, stu - student edition ed., 2005.

[42] L. Bernardi, "Rebirth of artificial intelligence." https://www.linkedin.com/pulse/rebirth-artificial-intelligence-linda-bernardi/. [Downloaded 19th of October 2020].

[43] D. Weinberger, *Everyday Chaos: Technology, Complexity, and How We're Thriving in a New World of Possibility*. Harvard Business Review Press, 2019.

[44] D. Kahneman, *Thinking, fast and slow*. New York: Farrar, Straus and Giroux, 2011.

[45] H. Kerzner, *Project management: a systems approach to planning, scheduling, and controlling*. John Wiley & Sons, 2017.

[46] J. Pinto, "Achieving competitive advantage," *New Jercy*, 2010.

[47] H. Hong, "An efficient point estimate method for probabilistic analysis," *Reliability Engineering & System Safety*, vol. 59, no. 3, pp. 261–267, 1998.

[48] A. Rolstadås, A. Johansen, N. Olsson, and J. A. Langlo, *Praktisk prosjektledelse: fra idé til gevinst*. Fagbokforlaget, 2020.

[49] G. W. Walkup, B. J. Ligon, *et al.*, "The good, bad, and ugly of stage-gate project management process as applied in the oil and gas industry," in *SPE Annual Technical Conference and Exhibition*, Society of Petroleum Engineers, 2006.

[50] M. P. Jalal and S. M. Koosha, "Identifying organizational variables affecting project management office characteristics and analyzing their correlations in the iranian project-oriented organizations of the construction industry," *International Journal of Project Management*, vol. 33, pp. 458–466, 2015.

[51] T. Ghioca, "Managing projects using project baselines." https://www.rationalplan.com/projectmanagementblog/managing-projects-using-project-baselines/. [Downloaded 31th of Jan 2021].

[52] A. Tereso, P. Ribeoro, G. Fernandes, I. Loureiro, and M. Ferreira, "Project Management Practices in Private Organizations," *Project Management Journal*, vol. 50, pp. 6–22, 2019.

[53] C. Besner and B. Hobbs, "Contextualized Project Management Practice: A Cluster Analysis of Practices and Best Practices," *Project Management Journal*, vol. 44, no. 1, pp. 17–34, 2012.

[54] H. Kerzner, *Strategic Planning for Project Management Using a Project Management Maturity Model*. John Wiley Sons, 2001.

[55] H. R. Thomas and I. Završki, "Construction baseline productivity: Theory and practice," *Journal of Construction Engineering and Management*, vol. 125, no. 5, pp. 295–303, 1999.

[56] T. M. Williams, *Managing and modelling complex projects*, vol. 17. Springer, 2013.

[57] M. A. Seely and Q. P. Duong, "The dynamic baseline model for project management," *Project Management Journal*, vol. 32, no. 2, pp. 25–36, 2001.

[58] W. Herroelen and R. Leus, "The construction of stable project baseline schedules," *European Journal of Operational Research*, vol. 156, no. 3, pp. 550 – 565, 2004.

[59] H. R. Thomas and I. Zavrski, "Construction Baseline Productivity: Theory and Practice," *Journal of Construction Engineering and Management*, vol. 125, pp. 295–303, 1999.

[60] F. M. Webster, "The wbs." https://www.pmi.org/learning/library/work-breakdown-structure-basic-principles-4883. [Downloaded 22nd of March 2021].

[61] E. Kristoffersen, *Jegerånden: Å lede i fred, krise og krig*. Gyldendal, 2020.

[62] W. S. H. Erik Leuven Demeulemeester, *Project Scheduling: A Research Handbook*. Springer Science Business Media, 2006.

[63] L. V. Tavares, "A review of the contribution of operational research to project management," *European Journal of Operational Research*, vol. 136, no. 1, pp. 1–18, 2002.

[64] J. Vatn, "TPK5115 - Risk Management in Projects," 2017.

[65] H. N. Ahuja, S. Dozzi, and S. Abourizk, *Project management: techniques in planning and controlling construction projects*. John Wiley & Sons, 1994.

[66] Project-Management.com, "What is a gantt chart?." https://project-management.com/what-is-a-gantt-chart/. [Downloaded 11th of February 2021].

[67] F. P.-M. S. H. Lee and M. Park, "Dynamic planning and control methodology for strategic and operational construction project management," *Automation in Construction*, vol. 15, no. 1, pp. 84–97, 2006.

[68] B. Hussein, *Veien til suksess*. Fagbokforlaget, 2016.

[69] R. D. H. Warburton and V. Kanabar, "The practical calculation of schedule variance in terms of schedule," 2008.

[70] T. Six, "The dcma 14-point assessment baseline execution index (bei)." https://tensix.com/2017/11/understanding-the-dcma-14-point-assessment-baseline-execution-index-bei/. [Downloaded 4th of May 2021].

[71] Wikipedia, "Artificial intelligence." https://en.wikipedia.org/wiki/Artificial_intelligence. [Downloaded 2th of February 2021].

[72] S. E. of Philosophy, "The turing test." https://plato.stanford.edu/entries/turing-test/. [Downloaded 2th of February 2021].

[73] B. Marr, "A short history of machine learning – every manager should read." https://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read/?sh=ebf186915e78. [Downloaded 2th of February 2021].

[74] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. L. S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskevera, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, 2016.

[75] I. C. Education, "What is machine learning?." https://www.ibm.com/cloud/learn/machine-learning. [Downloaded 2th of February 2021].

[76] H. M, "Overview: State-of-the-art machine learning algorithms per discipline per task." https://towardsdatascience.com/overview-state-of-the-art-machine-learning-algorithms-per-discipline-per-task-c1a16a66b8bb. [Downloaded 2nd of February 2021].

[77] B. Merr, "How much data do we create every day? the mind-blowing stats everyone should read." https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/?sh=30466c5c60ba. [Downloaded 2th of February 2021].

[78] IBM, "Machine learning solutions." https://www.ibm.com/dk-en/analytics/machine-learning?p1=Search&p4=43700052280872134&p5=e&cm_mmc=Search_Google-_-1S_1S-_-EP_NO-_-machine%20learning_e&cm_mmca7=71700000064495232&cm_mmca8=aud-382859943522:kwd-59020306&cm_mmca9=CjwKCAiAjeSABhAPEiwAqfxURXTMtNL_UYVe5iS7ZTzFtLrIYStYU3pV83_vLObW58R5EssOCG7jbRoC_IYQAvD_BwE&cm_mmca10=449224482873&cm_mmca11=e&gclid=CjwKCAiAjeSABhAPEiwAqfxURXTMtNL_UYVe5iS7ZTzFtLrIYStYU3pV83_vLObW58R5EssOCG7jbRoC_IYQAvD_BwE&gclsrc=aw.ds. [Downloaded 2th of February 2021].

[79] SAS, "Machine learning - what it is and why it matters." https://www.sas.com/en_us/insights/analytics/machine-learning.html. [Downloaded 2th of February 2021].

[80] C. Darwin, *On the Origin of Species by Means of Natural Selection*. London: Murray, 1859.

[81] R. Dilao, *From Charles Darwin to evolutionary genetic algorithms*, pp. 1–11. 01 2009.

[82] A. Ng, "Machine learning." https://www.coursera.org/learn/machine-learning. [Downloaded 22nd of February 2021].

[83] R. Elshawi, M. Maher, and S. Sakr, "Automated machine learning: State-of-the-art and open challenges," 2019.

[84] R. S. Olson, N. Bartley, R. J. Urbanowicz, and J. H. Moore, "Evaluation of a tree-based pipeline optimization tool for automating data science," in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16, (New York, NY, USA), pp. 485–492, ACM, 2016.

[85] I. Y. Javeri, M. Toutiaee, I. B. Arpinar, T. W. Miller, and J. A. Miller, "Improving neural networks for time series forecasting using data augmentation and automl," *arXiv preprint arXiv:2103.01992*, 2021.

[86] E. Mjolsness and D. DeCoste, "Machine learning for science: State of the art and future prospects," *Science*, vol. 293, pp. 2051–2055, 2001.

[87] N.Gill, P. Hall, K. Montgomery, and N. Schmidt, "A responsible machine learning workflow with focus on interpretable models, post-hoc explanation, and discrimination testing," *Information*, vol. 11, no. 3, 2020.

[88] Y. Ma, T. Xie, J. Li, and R. Maciejewsk, "Explaining vulnerabilities to adversarial machine learning through visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 1075–1085, 2020.

[89] "Machine learning in manufacturing: advantages, challenges, and applications,"

[90] Y. Zhu, H. Li, Y. Liao, B. Wang, Z. Guan, H. Liu, and D. Cai, "What to do next: Modeling user behaviors by time-lstm.," in *IJCAI*, vol. 17, pp. 3602–3608, 2017.

[91] K. Cheung, "10 applications of machine learning in oil gas." https://algorithmxlab.com/blog/10-applications-machine-learning-oil-gas-industry/. [Downloaded 9th of February 2021].

[92] H. Maniar, S. Ryali, M. S. Kulkarni, and A. Abubakar, *Machine-learning methods in geoscience*. 2018.

[93] R. S. Patwardhan, "Machine learning in oil gas industry: A novel application of clustering for oilfield advanced process control," 2019.

[94] A. Mosavi, M. Salimi, S. F. Ardabili, T. Rabczuk, S. Shamshirband, and A. R. Varkonyi-Koczy, "State of the art of machine learning models in energy systems, a systematic review," *Energies*, vol. 12, no. 7, 2019.

[95] P. I. Pecholcs, R. Al-Saad, M. Al-Sannaa, J. Quigley, C. Bagaini, A. Zarkhidze, R. May, M. Guellili, S. Sinanaj, and M. Membrouk, "A broadband full azimuth land seismic case study from saudi arabia using a 100,000 channel recording system at 6 terabytes per day: acquisition and processing lessons learned," *SEG Technical Program Expanded Abstracts*, vol. 1, no. 5, 2012.

[96] B. Midtlid, "Slik unngår du at dårlige data kommer i veien for innovasjon." https://www.pwc.no/no/pwc-aktuelt/slik-unngar-du-at-darlige-data-kommer-i-veien-for-innovasjon.html. [Downloaded 17th of February 2021].

[97] M. Courtney, "Puzzling out big data," *Engineering & Technology*, vol. 7, no. 12, pp. 56–60, 2012.

[98] P. Russom *et al.*, "Big data analytics," *TDWI best practices report, fourth quarter*, vol. 19, no. 4, pp. 1–34, 2011.

[99] V. Gligorijević, N. Malod-Dognin, and N. Pržulj, "Integrative methods for analyzing big data in precision medicine," *Proteomics*, vol. 16, no. 5, pp. 741–758, 2016.

[100] "The four v's of big data." https://www.ibmbigdatahub.com/infographic/four-vs-big-data. [Downloaded 17th of February 2021].

[101] A. Voje, "Hvordan lykkes med ai: Realisering." https://www.pwc.no/no/pwc-aktuelt/hvordan-lykkes-med-ai-realisering.html. [Downloaded 17th of February 2021].

[102] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, "Machine learning, neural and statistical classification," 1994.

[103] N. Dogan and Z. Tanrikulu, "A comparative analysis of classification algorithms in data mining for accuracy, speed and robustness," *Information Technology and Management*, vol. 14, no. 2, pp. 105–124, 2013.

[104] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[105] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[106] P. H. Swain and H. Hauska, "The decision tree classifier: Design and potential," *IEEE Transactions on Geoscience Electronics*, vol. 15, no. 3, pp. 142–147, 1977.

[107] M. Welling, "Fisher linear discriminant analysis," 2005.

[108] E. Allibhai, "Building a k-nearest-neighbors (k-nn) model with scikit-learn." https://towardsdatascience.com/building-a-k-nearest-neighbors-k-nn-model-with-scikit-learn-51209555453a@misc. [Downloaded 24th of February 2021].

[109] W. S. Noble, "What is a support vector machine?," *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[110] S. Suthaharan, "Support vector machine," in *Machine learning models and algorithms for big data classification*, pp. 207–235, Springer, 2016.

[111] Dr. Michael J. Garbade, "Regression versus classification machine learning: What's the difference?." https://medium.com/quick-code/regression-versus-classification-machine-learning-whats-the-difference-345c56dd15f7. [Downloaded 31th of March 2021].

[112] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, "An introduction to decision tree modeling," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 18, no. 6, pp. 275–285, 2004.

[113] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 785–794, ACM, 2016.

[114] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny, "An empirical comparison of machine learning models for time series forecasting," *Econometric Reviews*, vol. 29, no. 5-6, pp. 594–621, 2010.

[115] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[116] J. Opitz and S. Burst, "Macro f1 and macro f1," *arXiv preprint arXiv:1911.03347*, 2019.

[117] Z. Chase Lipton, C. Elkan, and B. Narayanaswamy, "Thresholding classifiers to maximize f1 score," *arXiv e-prints*, pp. arXiv–1402, 2014.

[118] M. Buckland and F. Gey, "The relationship between recall and precision," *Journal of the American society for information science*, vol. 45, no. 1, pp. 12–19, 1994.

[119] S. Lundberg and S. Lee, "A unified approach to interpreting model predictions," *CoRR*, vol. abs/1705.07874, 2017.

[120] L. S. Shapley, "Stochastic games," *Proceedings of the national academy of sciences*, vol. 39, no. 10, pp. 1095–1100, 1953.

[121] L. Antwarg, R. M. Miller, B. Shapira, and L. Rokach, "Explaining anomalies detected by autoencoders using shap," *arXiv preprint arXiv:1903.02407*, 2019.

[122] E. Kalai and D. Samet, "On weighted shapley values," *International journal of game theory*, vol. 16, no. 3, pp. 205–222, 1987.

[123] J. MA, Y. DING, J. Cheng, Y. Tan, V. Gan, and J. ZHANG, "Analyzing the leading causes of traffic fatalities using xgboost and grid-based analysis: A city management perspective," *IEEE Access*, vol. PP, pp. 1–1, 10 2019.

[124] J. Brownlee, "How to use standardscaler and minmaxscaler transforms in python," 2020.

[125] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems, Amsterdam: Morgan Kaufmann, 3 ed., 2011.

[126] "sklearn.preprocessing.minmaxscaler." https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html?highlight=minmaxscale#sklearn.preprocessing.MinMaxScaler. [Downloaded 6th of April 2021].

[127] M. Kuhn and K. Johnson, "Applied predictive modeling," 2013.

[128] M. Fuchs, "Feature scaling with scikit-learn." https://michael-fuchs-python.netlify.app/2019/08/31/feature-scaling-with-scikit-learn/, 2019. [Downloaded 6th of April 2021].

[129] N. I. Sapankevych and R. Sankar, "Time series prediction using support vector machines: a survey," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 24–38, 2009.

[130] L. Šečkute and A. Pabedinskaite, "Application of forecasting methods in business," *Journal of Business Economics and Management*, vol. 4, no. 2, pp. 144–157, 2003.

[131] Planettogether, "The disadvantages of sales forecasting." https://www.planettogether.com/blog/the-disadvantages-of-sales-forecasting. [Downloaded 25th of February 2021].

[132] G. Mahalakshmi, S. Sridevi, and S. Rajaram, "A survey on forecasting of time series data," in *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, pp. 1–8, IEEE, 2016.

[133] S. Mehrmolaei and M. R. Keyvanpourr, "A brief survey on event prediction methods in time series," in *Artificial Intelligence Perspectives and Applications*, pp. 235–246, Springer, 2015.

[134] B. Krollner, B. J. Vanstone, and G. R. Finnie, "Financial time series forecasting with machine learning techniques: a survey.," in *ESANN*, 2010.

[135] E. Hüllermeier, "Does machine learning need fuzzy logic?," *Fuzzy Sets and Systems*, vol. 281, pp. 292–299, 2015.

[136] GeeksforGeeks, "Fuzzy logic - introduction." https://www.geeksforgeeks.org/fuzzy-logic-introduction/. [Downloaded 25th of February 2021].

[137] D. Nguyen-Tuong, J. Peters, and M. Seeger, "Local gaussian process regression for real time online model learning and control," in *Proceedings of the 21st International Conference on Neural Information Processing Systems*, pp. 1193–1200, 2008.

[138] X. Yan and N. A. Chowdhury, "A comparison between svm and lssvm in mid-term electricity market clearing price forecasting," in *2013 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1–4, IEEE, 2013.

[139] H. Liu, D. Liu, G. Zheng, Y. Liang, and Y. Ni, "Research on natural gas load forecasting based on support vector regression," in *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788)*, vol. 4, pp. 3591A–3595, IEEE, 2004.

[140] C. kathuria, "Regression — why mean square error?." https://towardsdatascience.com/https-medium-com-chayankathuria-regression-why-mean-square-error-a8cad2a1c96f. [Downloaded 10th of March 2021].

[141] N. Vandeput, "Forecast kpis: Rmse, mae, mape bias." https://towardsdatascience.com/forecast-kpi-rmse-mae-mape-bias-cdc5703d242d. [Downloaded 17th of March 2021].

[142] JJ, "Mae and rmse — which metric is better?." https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d. [Downloaded 16th of March 2021].

[143] J. Brownlee, "A gentle introduction to long short-term memory networks by the experts." https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/. [Downloaded 15th of March 2021].

[144] IBM Claud Education, "What are recurrent neural networks?." https://www.ibm.com/cloud/learn/recurrent-neural-networks. [Downloaded 17th of March 2021].

[145] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[146] P. Esposito, "Building a lstm by hand on pytorch." https://towardsdatascience.com/building-a-lstm-by-hand-on-pytorch-59c02a4ec091. [Downloaded 15th of March 2021].

[147] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[148] S. Mishra, C. Bordin, K. Taharaguchi, and I. Palu, "Comparison of deep learning models for multivariate prediction of time series wind power generation and temperature," *Energy Reports*, vol. 6, pp. 273–286, 2020.

[149] R. Keinzler, "Introducing deep learning and long-short term memory networks." https://developer.ibm.com/technologies/iot/tutorials/iot-deep-learning-anomaly-detection-1/. [Downloaded 18th of March 2021].

[150] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proceedings*, vol. 89, pp. 89–94, Presses universitaires de Louvain, 2015.

[151] T. Yiu, "Understanding arima (time series modeling)." https://towardsdatascience.com/understanding-arima-time-series-modeling-d99cd11be3f8. [Downloaded 8th of April 2021].

[152] A. Rajbhoj, "Arima simplified.." https://towardsdatascience.com/arima-simplified-b63315f27cbc. [Downloaded 8th of April 2021].

[153] J. Brownlee, "How to develop multivariate multi-step time series forecasting models for air pollution." https://machinelearningmastery.com/how-to-develop-machine-learning-models-for-multivariate-multi-step-air-pollution-time-series-forec [Downloaded 2nd of March 2021].

[154] M. Phi, "Illustrated guide to lstm's and gru's: A step by step explanation." https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21. [Downloaded 2nd of March 2021].

[155] J. Hua, Z. Xiong, J. Lowey, E. Suh, and E. Dougherty, "Optimal number of features as a function of sample size for various classification rules," *Bioinformatics (Oxford, England)*, vol. 21, pp. 1509–15, 05 2005.

[156] J. Kang, T. Rancati, S. Lee, J. H. Oh, S. Kerns, J. Scott, R. Schwartz, S. Kim, and B. Rosenstein, "Machine learning and radiogenomics: Lessons learned and future directions," *Frontiers in Oncology*, vol. 8, 06 2018.

[157] A. Quemy, "Two-stage optimization for machine learning workflow," *Information Systems*, vol. 92, p. 101483, 2020.

[158] D. Deutch, A. Gilad, T. Milo, and A. Somech, "Explained: Explanations for eda notebooks," *Proc. VLDB Endow.*, vol. 13, p. 2917–2920, Aug. 2020.

[159] P. Lerman, "Fitting segmented regression models by grid search," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 29, no. 1, pp. 77–84, 1980.

[160] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization.," *Journal of machine learning research*, vol. 13, no. 2, 2012.

[161] The pandas development team, "pandas-dev/pandas: Pandas," Feb. 2020.

[162] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[163] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning

in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[164] B. Ward, "Messy data cleaning for data set with many unique values." https://medium.com/@BW.benward/messy-data-cleaning-for-data-set-with-many-unique-values-interesting-eda-tutorial-with-pandas-ea [Downloaded 31th of Mai 2021].

[165] R. Matheson, "How to tell whether machine-learning systems are robust enough for the real world." https://news.mit.edu/2019/how-tell-whether-machine-learning-systems-are-robust-enough-real-worl-0510. [Downloaded 31th of Mai 2021].

[166] Arundo Analytics, "tsaug." https://tsaug.readthedocs.io/en/stable/index.html. [Downloaded 19th of March 2021].

[167] M. Dwarampudi and N. V. S. Reddy, "Effects of padding on lstms and cnns," *CoRR*, vol. abs/1903.07288, 2019.

[168] A. Mackenzie, "The production of prediction: What does machine learning want?," *European Journal of Cultural Studies*, vol. 18, no. 4-5, pp. 429–445, 2015.

[169] J. Brownlee, "How to develop a skillful machine learning time series forecasting model." https://machinelearningmastery.com/how-to-develop-a-skilful-time-series-forecasting-model/. [Downloaded 5th of June 2021].

[170] A. Singh, "Build high performance time series models using auto arima in python and r." https://www.analyticsvidhya.com/blog/2018/08/auto-arima-time-series-modeling-python-r/. [Downloaded 5th of June 2021].

[171] Y. Grushka-Cockayne, V. R. R. Jose, and K. C. Lichtendahl Jr, "Ensembles of overfit and overconfident forecasts," *Management Science*, vol. 63, no. 4, pp. 1110–1130, 2017.

[172] Pourya, "Time series machine learning regression framework." https://towardsdatascience.com/time-series-machine-learning-regression-framework-9ea33929009a. [Downloaded 31th of Mai 2021].

[173] S. Zagoruyko and N. Komodakis, "Wide residual networks," *CoRR*, vol. abs/1605.07146, 2016.

[174] J. D. Rodriguez, A. Perez, and J. A. Lozano, "Sensitivity analysis of k-fold cross validation in prediction error estimation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 3, pp. 569–575, 2009.

[175] C. Cochrane, "Time series nested cross-validation." https://towardsdatascience.com/time-series-nested-cross-validation-76adba623eb9. [Downloaded 30th of March 2021].

[176] K. Zaamout and J. Z. Zhang, "Improving neural networks classification through chaining," in *Artificial Neural Networks and Machine Learning – ICANN 2012*

(A. E. P. Villa, W. Duch, P. Érdi, F. Masulli, and G. Palm, eds.), (Berlin, Heidelberg), pp. 288–295, Springer Berlin Heidelberg, 2012.

[177] G. Seif, "The art of cleaning your data." https://towardsdatascience.com/the-art-of-cleaning-your-data-b713dbd49726. [Downloaded 31th of Mai 2021].

[178] N. Sánchez-Maroño and A. Alonso-Betanzos, "Feature selection based on sensitivity analysis," in *Current Topics in Artificial Intelligence* (D. Borrajo, L. Castillo, and J. M. Corchado, eds.), (Berlin, Heidelberg), pp. 239–248, Springer Berlin Heidelberg, 2007.

[179] "Spss tutorials: Pearson correlation." https://libguides.library.kent.edu/SPSS/PearsonCorr. [Downloaded 17th of Mai 2021].

[180] D. E. Hinkle, W. Wiersma, and S. G. Jurs, *Applied statistics for the behavioral sciences*, vol. 663. Houghton Mifflin College Division, 2003.

[181] A. Tato and R. Nkambou, "Improving adam optimizer," 2018.

[182] N. Reimers and I. Gurevych, "Optimal hyperparameters for deep lstm-networks for sequence labeling tasks," *CoRR*, vol. abs/1707.06799, 2017.

[183] J. Brownlee, "How to use learning curves to diagnose machine learning model performance." https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/. [Downloaded 28th of May 2021].

[184] J. Brownlee, "Difference between a batch and an epoch in a neural network." https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/. [Downloaded 5th of June 2021].

[185] P.-L. Hsu and H. Robbins, "Complete convergence and the law of large numbers," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 33, no. 2, p. 25, 1947.

[186] Sci-Kit Learn, "Compare the effect of different scalers on data with outliers." https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html. [Downloaded 31th of Mai 2021].

[187] F. Leoni, "From zero to hero in xgboost tuning." https://towardsdatascience.com/from-zero-to-hero-in-xgboost-tuning-e48b59bfaf58. [Downloaded 31th of Mai 2021].

[188] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[189] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[190] A. Ahmad, A. Mustapha, E. D. Zahadi, N. Masah, and N. Y. Yahaya, "Comparison between neural networks against decision tree in improving prediction accuracy for diabetes mellitus," in *Digital Information Processing and Communications* (V. Snasel, J. Platos, and E. El-Qawasmeh, eds.), (Berlin, Heidelberg), pp. 537–545, Springer Berlin Heidelberg, 2011.

[191] D. Waterman, "A guide to expert systems,"

[192] D. Chua, P. Loh, Y. Kog, and E. Jaselskis, "Neural networks for construction project success," *Expert Systems with Applications*, vol. 13, no. 4, pp. 317–328, 1997. Selected Papers from the PACES/SPICIS'97 Conference.

[193] A. Osman, *Radiation Oncology in the Era of Big Data and Machine Learning for Precision Medicine*, pp. 1–30. 03 2019.

[194] G. Kim, P. Debois, J. Willis, and J. Humble, *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press, 2016.

[195] B. Pietracatella, "Are you dropping too many correlated features?." https://towardsdatascience.com/are-you-dropping-too-many-correlated-features-d1c96654abe6. [Downloaded 31th of Mai 2021].

[196] P. Dhar, "The carbon impact of artificial intelligence," *Nature Machine Intelligence*, vol. 2, pp. 423–25, 08 2020.

[197] D. R. Gallagher, P. Gardner, and P. L. Swan, "Portfolio pumping: An examination of investment manager quarter-end trading and impact on performance," *Pacific-Basin Finance Journal*, vol. 17, no. 1, pp. 1–27, 2009.

[198] K. E. Small, "Earned value management system (evms) program analysis pamphlet (pap)." https://www.dcma.mil/Portals/31/Documents/Policy/DCMA-PAM-200-1.pdf?ver=2016-12-28-125801-627. [Downloaded 4th of June 2021].

Appendices

Appendix A Company A

A.1 Augmentations

Figure A.1: Features 1-3 augmented data.

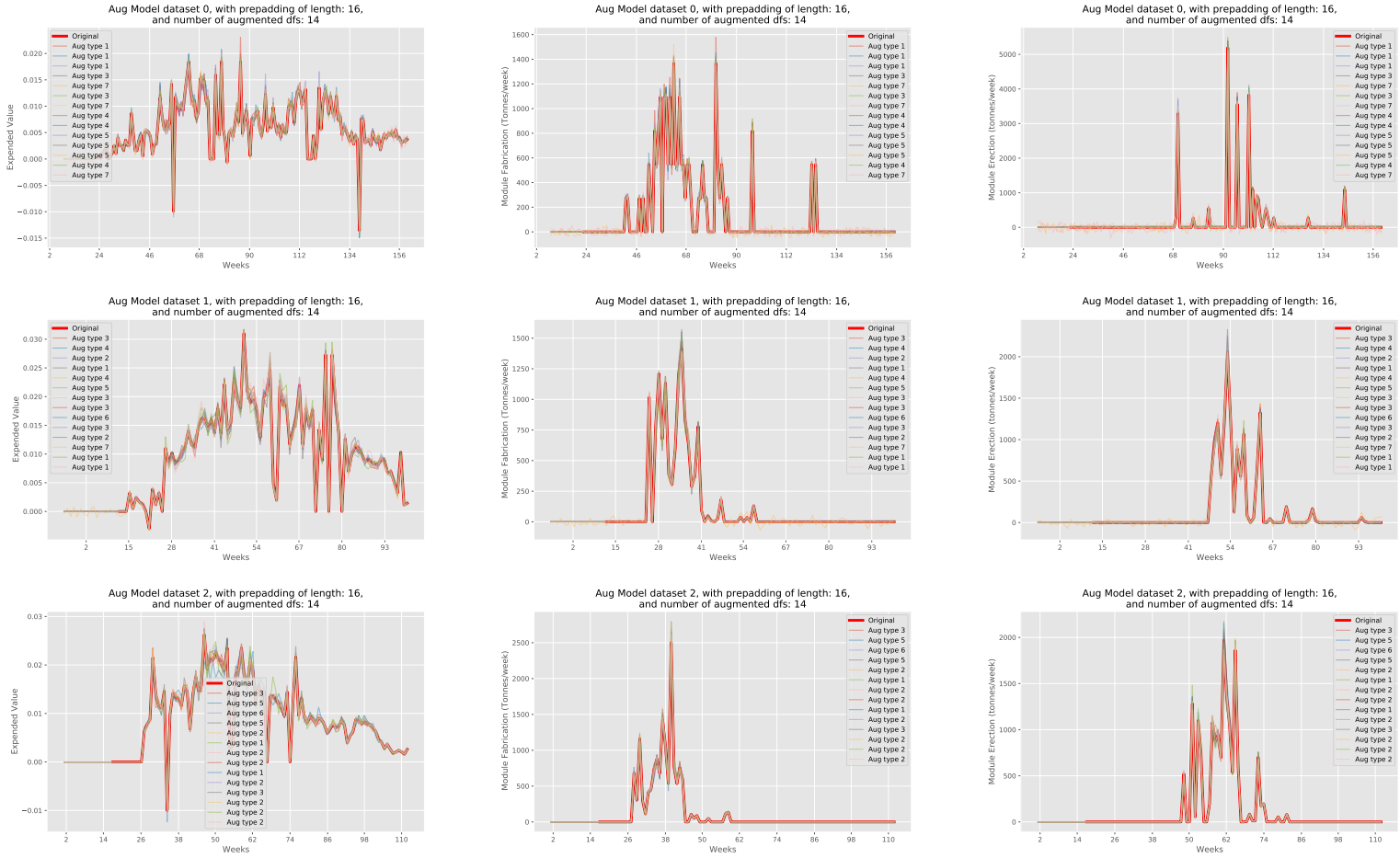


Figure A.2: Features 4-7 augmented data.

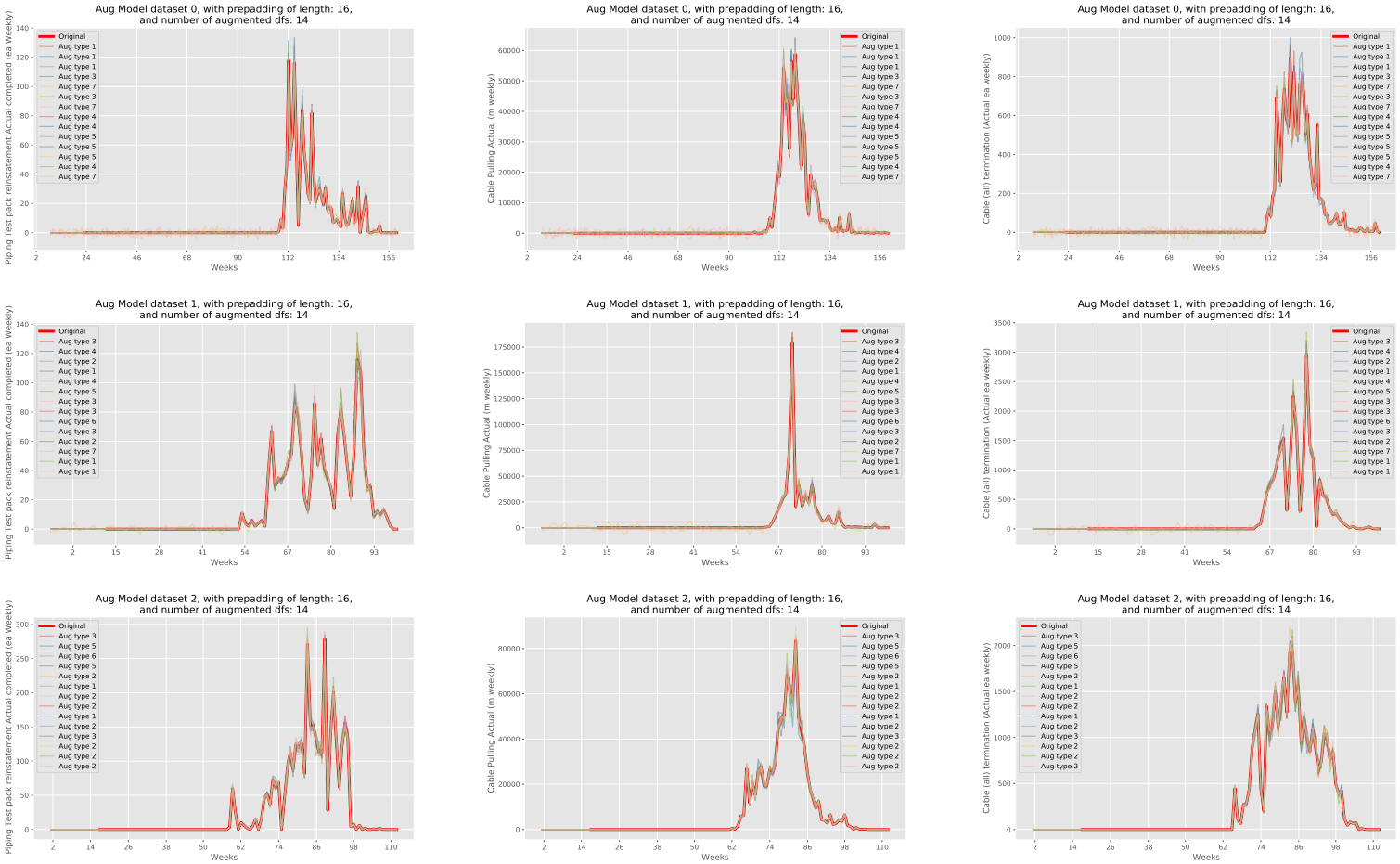
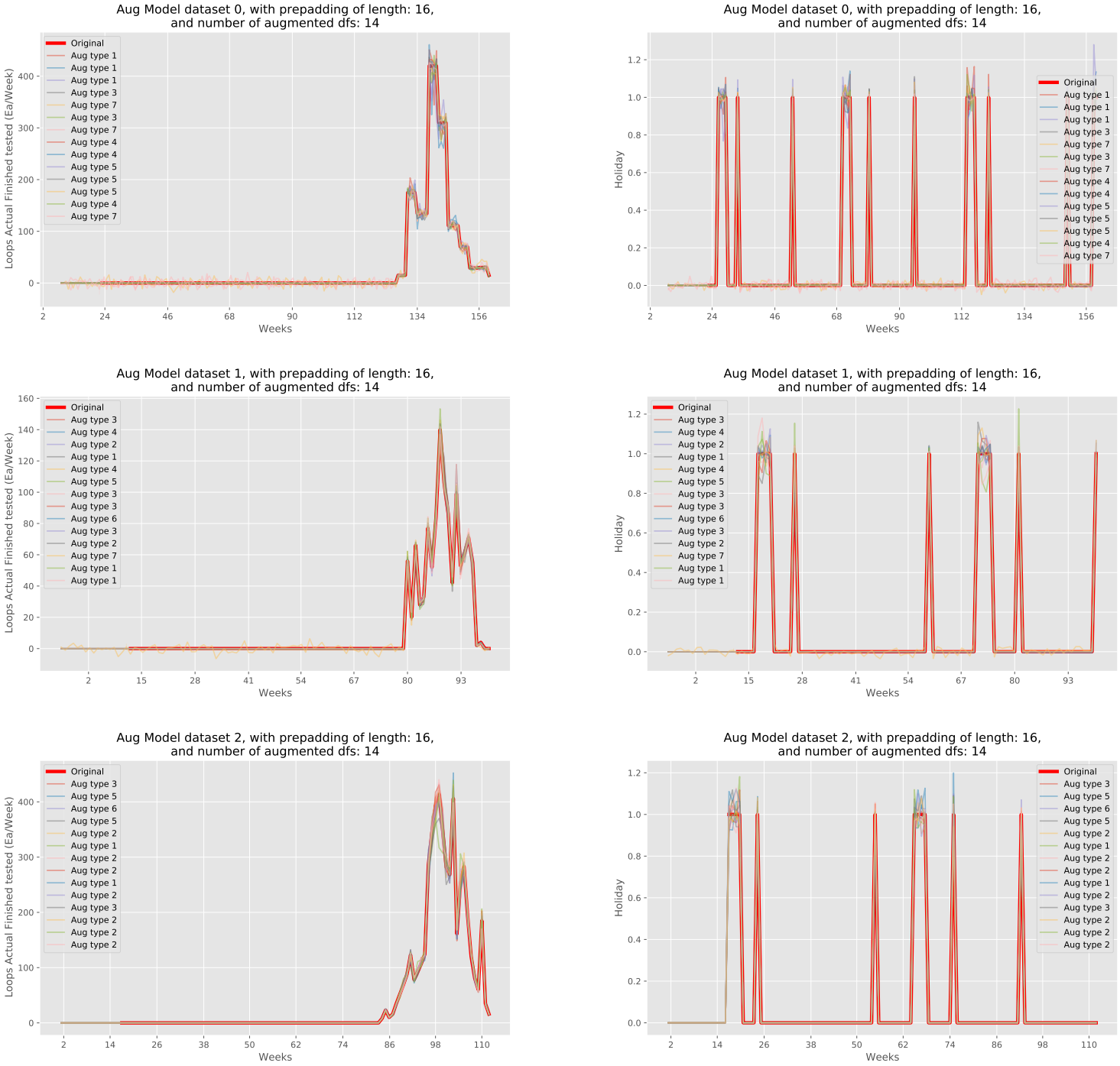


Figure A.3: Features 8-9 augmented data.



A.2 Figures

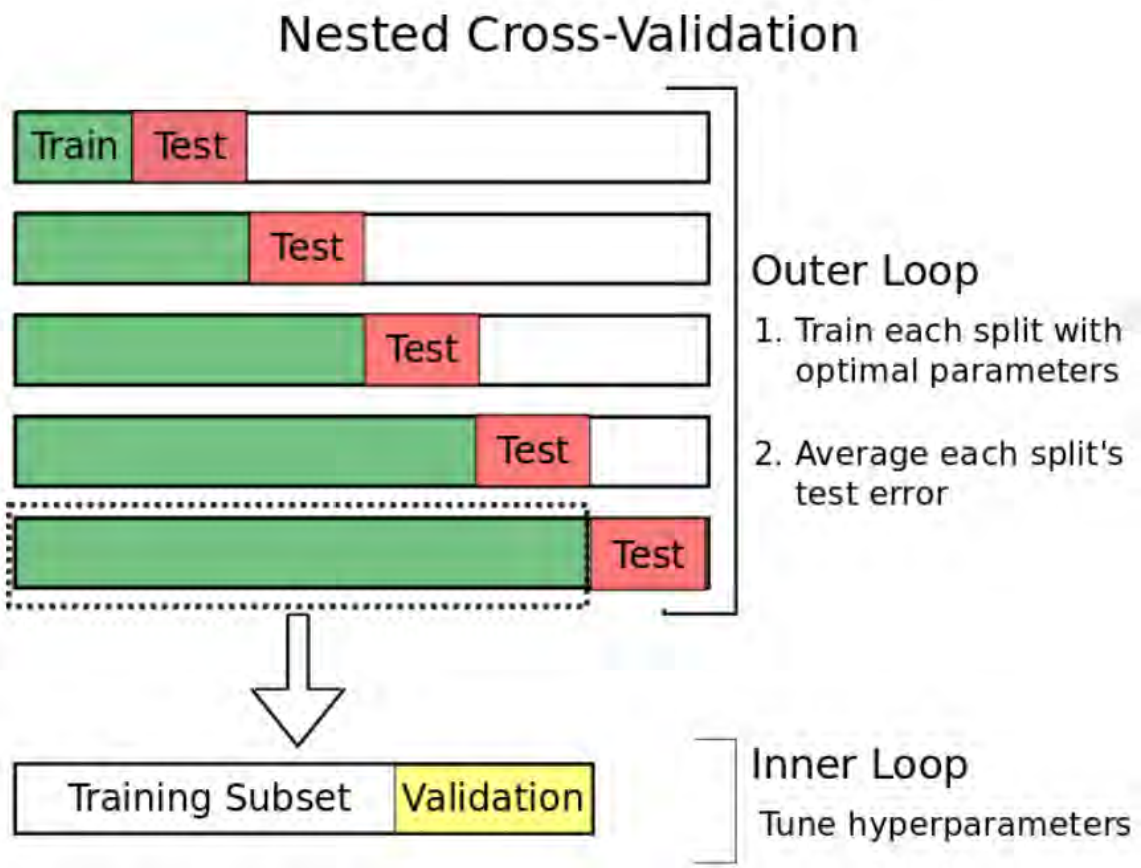


Figure A.4: Example of nested cross-validation.

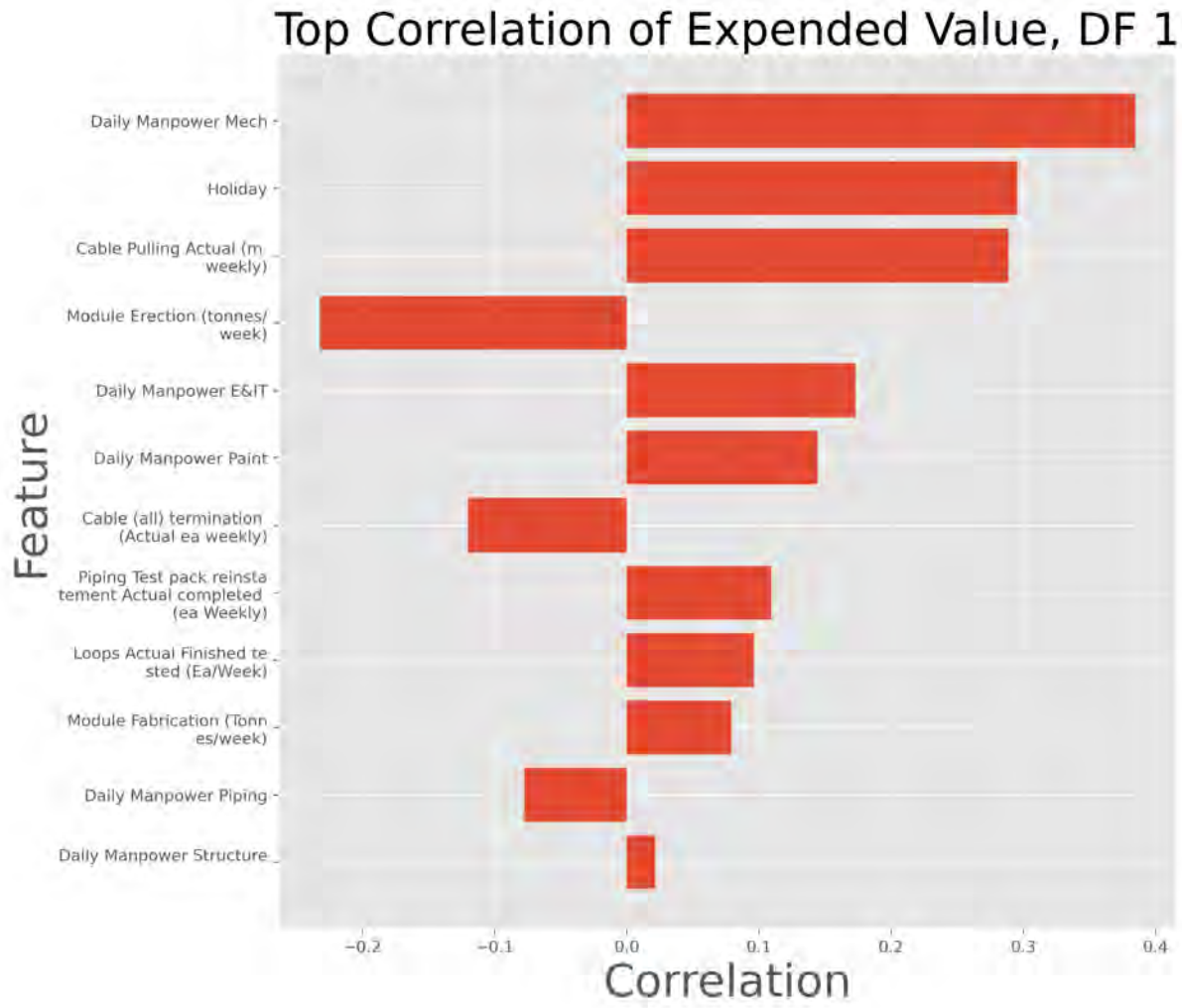


Figure A.5: Top absolute correlation between label Earned Value and other features, Company A, dataframe 1.

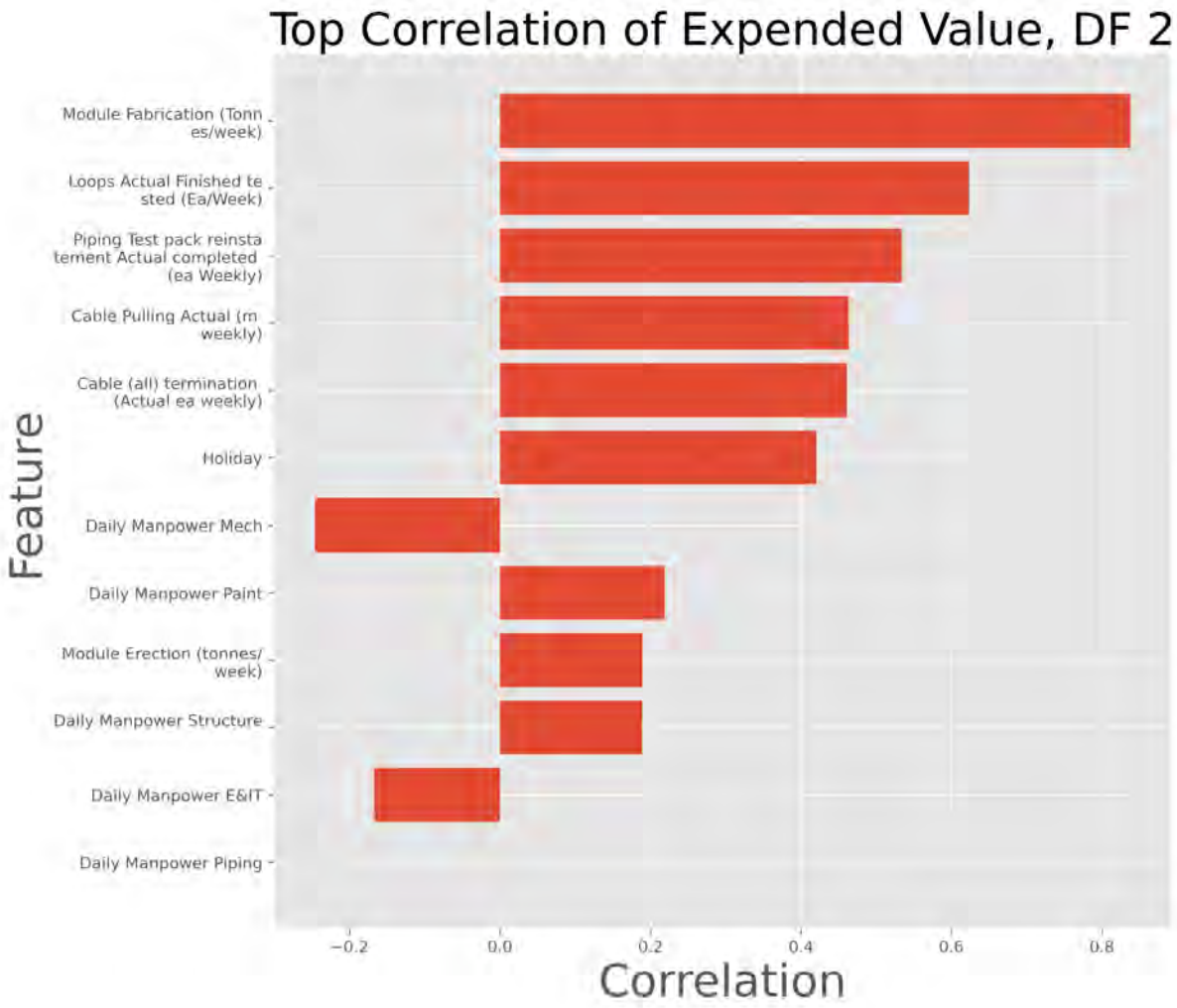


Figure A.6: Top absolute correlation between label Earned Value and other features, Company A, dataframe 2.

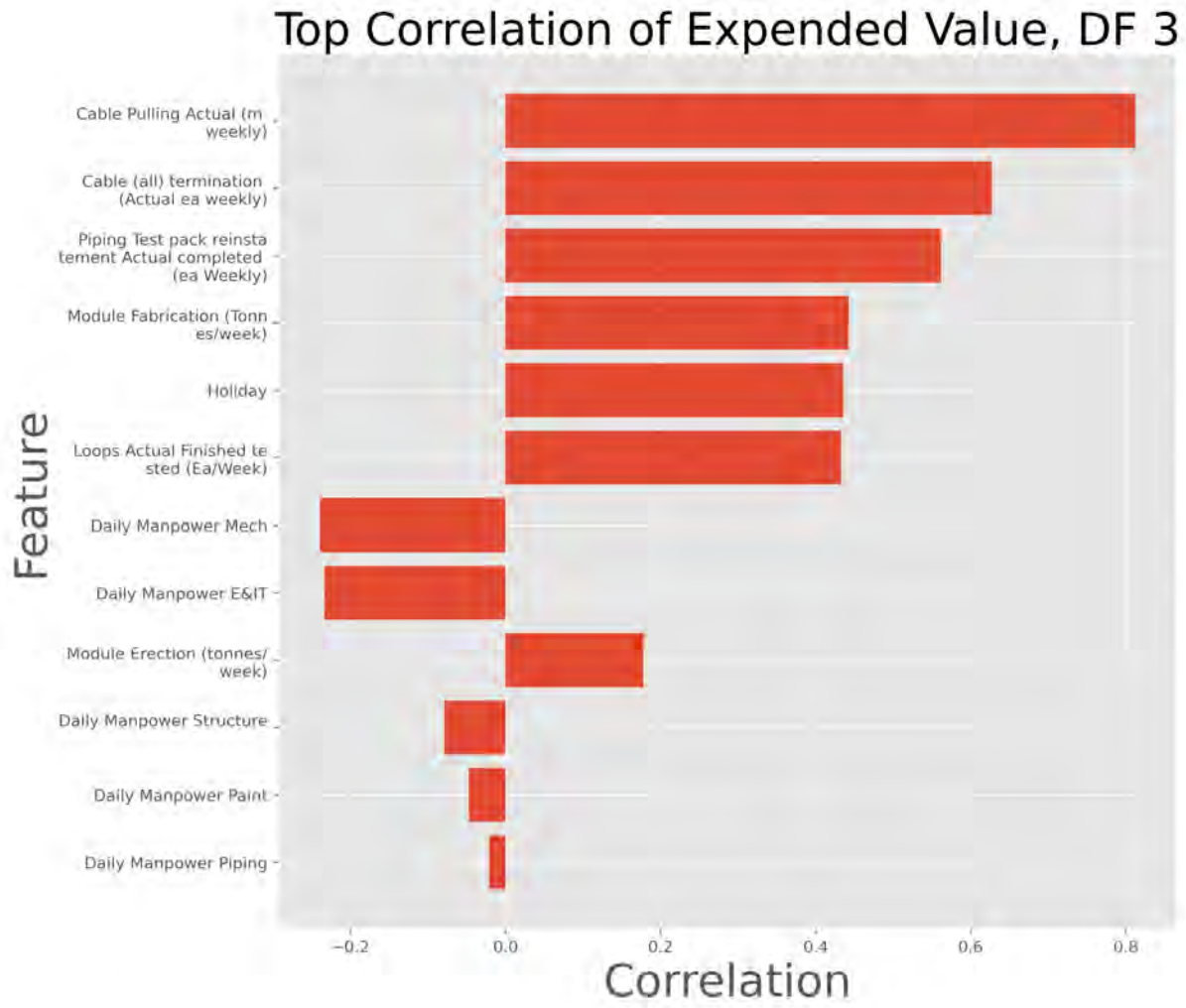


Figure A.7: Top absolute correlation between label Earned Value and other features, Company A, dataframe 3.

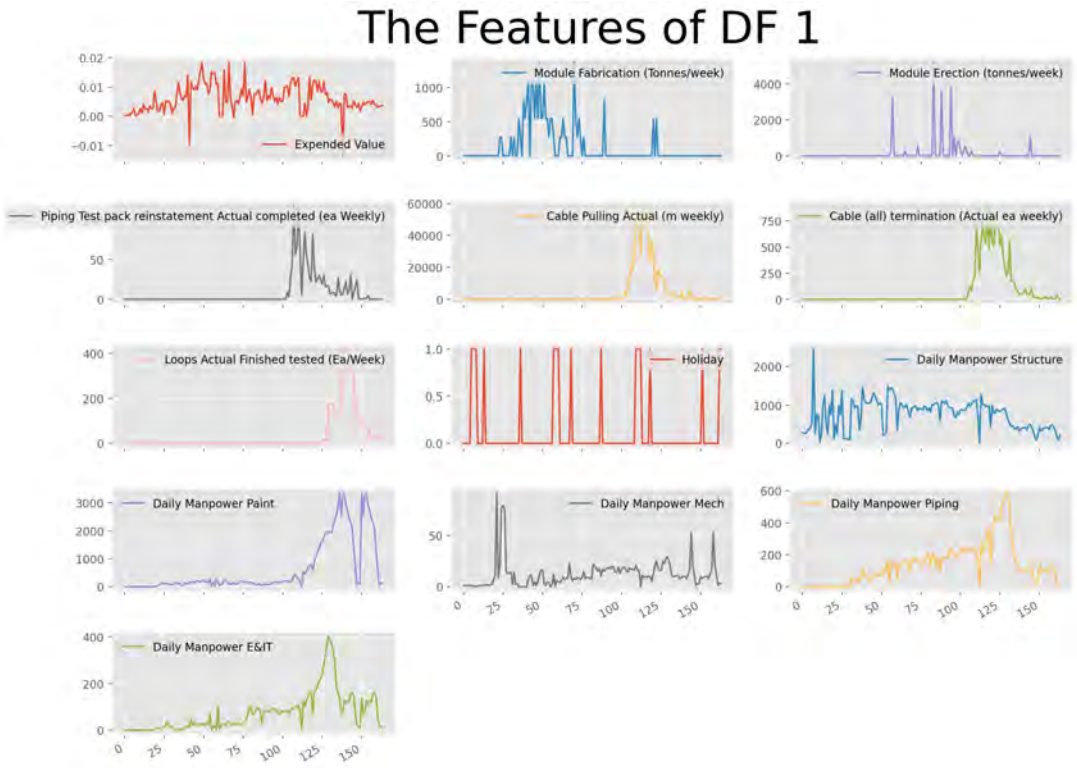


Figure A.8: Features in dataframe 1.

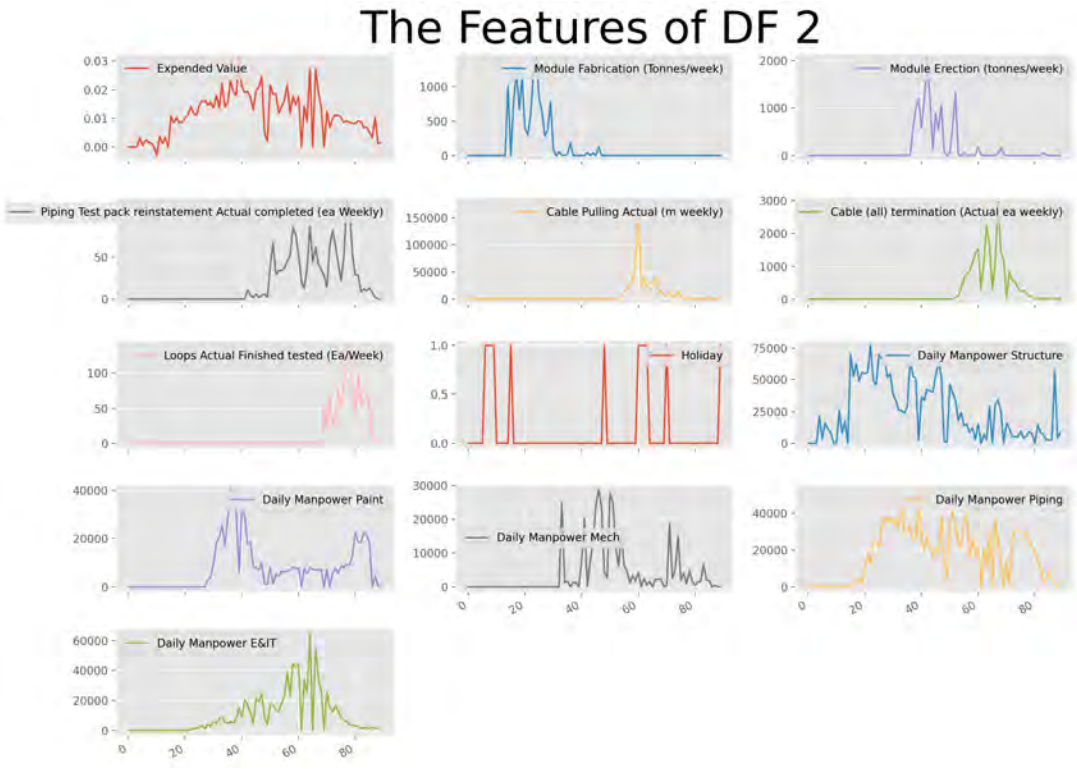


Figure A.9: Features in dataframe 2.

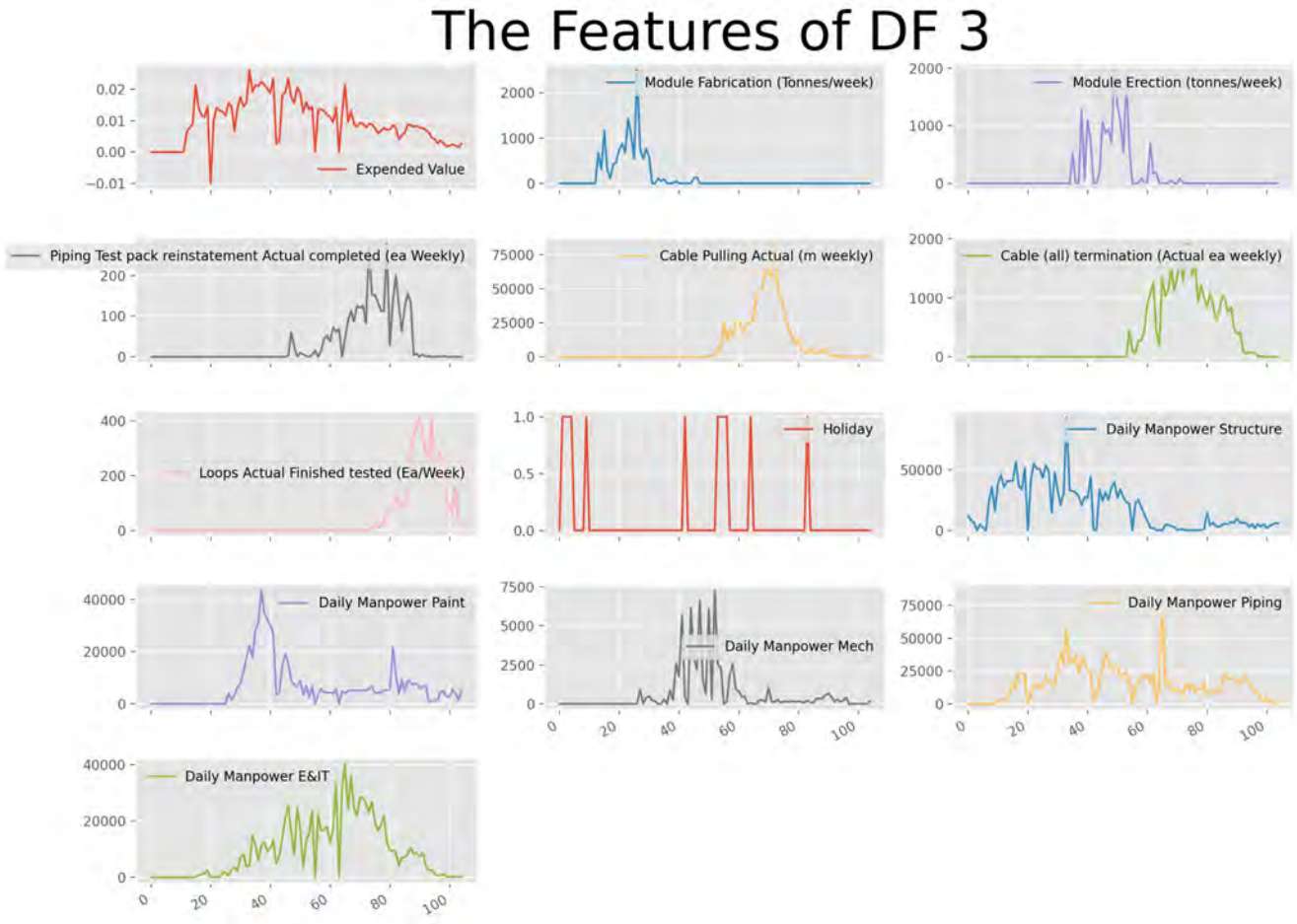


Figure A.10: Features in dataframe 3.

Appendix B Company B

B.1 Tables

Table B.1: Sum of each column in the dataset. Some of the features are anonymous.

Feature	Sum	Feature	Sum	Feature	Sum
EV	459047	discipline_Electrical	49	activity_role_Manualprogress	62
du	94334	discipline_EngineeringSupervision	9	activity_role_Mhr.Carrier	62
cumweekly_mean_du	94334	discipline_XA1	2	activity_role_ProcurementMilestone	2
expended_qty	2237302	discipline_Fab. Management/Supervision	5	index_phase_Execution	1
csh	660643	discipline_HSSEQ	5	index_phase_FeedPhase	385
Rsh	776523	discipline_HVAC	1	index_phase_Follow-on	70
Tsh	777357	discipline_InformationManagement	6	index_phase_ManagementAssistance	155
des_feed	0	discipline_Instrumentation	67	company_code_DirectEngineering	154
des_design	6	discipline_InternalMilestones	1	company_code_IndirectandMngmt.	54
des_initial	67	discipline_X1-1	47	scope_type_ EngineeringandProcurement	1
des_project	20	discipline_Mechanical	24	scope_type_Follow-on(cma)	69
des_cost	68	discipline_X2	51	scope_type_ManagementAssistance	152
des_estimate	52	discipline_PersonnelAdministration	2	site_code_X7	170
des_engineering	5	discipline_Piping&layout	30	site_code_X8	43
des_complete	49	discipline_Planning&Methods	6	site_code_X9	1
des_management	2	discipline_Process	41	main_discipline_Architect	1
des_interface	3	discipline_Procurement	4	main_discipline_CLN	42
des_X10	1	discipline_Procurement/ PackageAdm.Mngt.	1	main_discipline_Completion	12
des_cln	0	discipline_ ProjectManagementGeneral	2	main_discipline_ContractMilestones	22
des_contract	0	discipline_Risk	5	main_discipline_CostControl	1
des_review	2	discipline_Safety	71	main_discipline_Electrical	41
des_assistance	15	discipline_Secretaries	6	main_discipline_X1-2	2
des_administration	4	discipline_Structural/steeloutfitting	26	main_discipline_HSSEQ	2
des_X11	0	discipline_Surfacetreatment/ Materialtechnology	30	main_discipline_HVAC	1
des_material	1	discipline_Telecommunication	18	main_discipline_IT	6
des_mhrs	0	discipline_Weight	6	main_discipline_Instrumentation	59
des_cma	1	discipline_Workpreparation/Method	4	main_discipline_Management	5
des_system	6	main_area_General	460	main_discipline_ ManagementAssistance(cma)	155
des_X12	1	main_area_X3	6	main_discipline_X1	26
des_phase	6	main_area_X4	1	main_discipline_Mechanical	19
des_cancelled	0	scope_type_ EngineeringandProcurement	1	main_discipline_Multidiscipline	1
des_piping	4	scope_type_Follow-on(cma)	69	main_discipline_Multidiscipline/ General	39
des_startup	0	scope_type_ManagementAssistance	152	main_discipline_Personnel Administration	2
des_equipment	6	sub_area_GlobalEngineering	57	main_discipline_Piping&layout	26
main_category_ Completion	11	sub_area_X5	37	main_discipline_Planning	4
main_category_ Engineering	481	procurement_X6	2	main_discipline_Process	35
main_category_ Equipmentsourcing	2	phase_Designengineering	280	main_discipline_Procurement	4
main_category_ Management	87	phase_Follow-onengineering/ as-built/DFO/Fieldengineering	78	main_discipline_ProjectManagment	2
main_category_ Milestones	27	phase_General	9	main_discipline_ProjectSecretary	3
main_category_ Sourcing	5	main_phase_Engineering Follow-onPhase	80	main_discipline_Risk	2
discipline_Architect	4	main_phase_ExecutePhaseEPcma	123	main_discipline_Safety	49
discipline_ CLNEngineering	21	main_phase_FEED	385	main_discipline_ Structural/steeloutfitting	26
discipline_ CLNManagement	22	main_phase_InitialInterim EngineeringPhase	36	main_discipline_Surfacetreatment/ Materialtechnology	26
discipline_Completion	24	activity_role_Doc.prg,Mhr.Carrier	23	main_discipline_Weight	3
discipline_ ContractMilestones	26	activity_role_Document, Manualprogress	82	main_discipline_Workpreparation/ Method	2
discipline_Costest.& Accounting	3	activity_role_Documentprogress	59	—	—

B.2 Figures

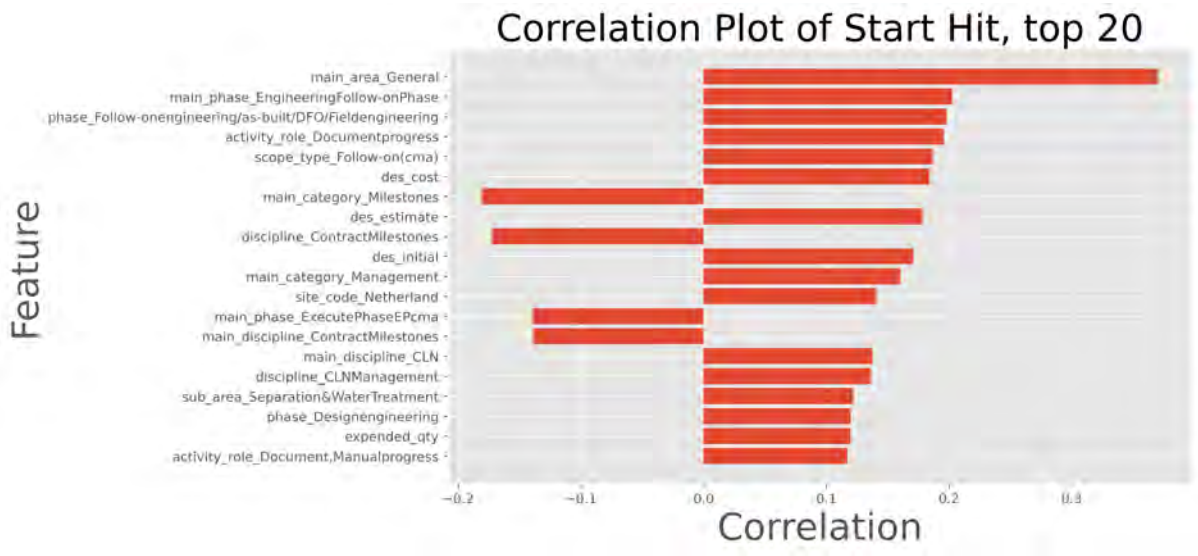


Figure B.1: Top absolute correlation between label Start Hit and other features, Company B.

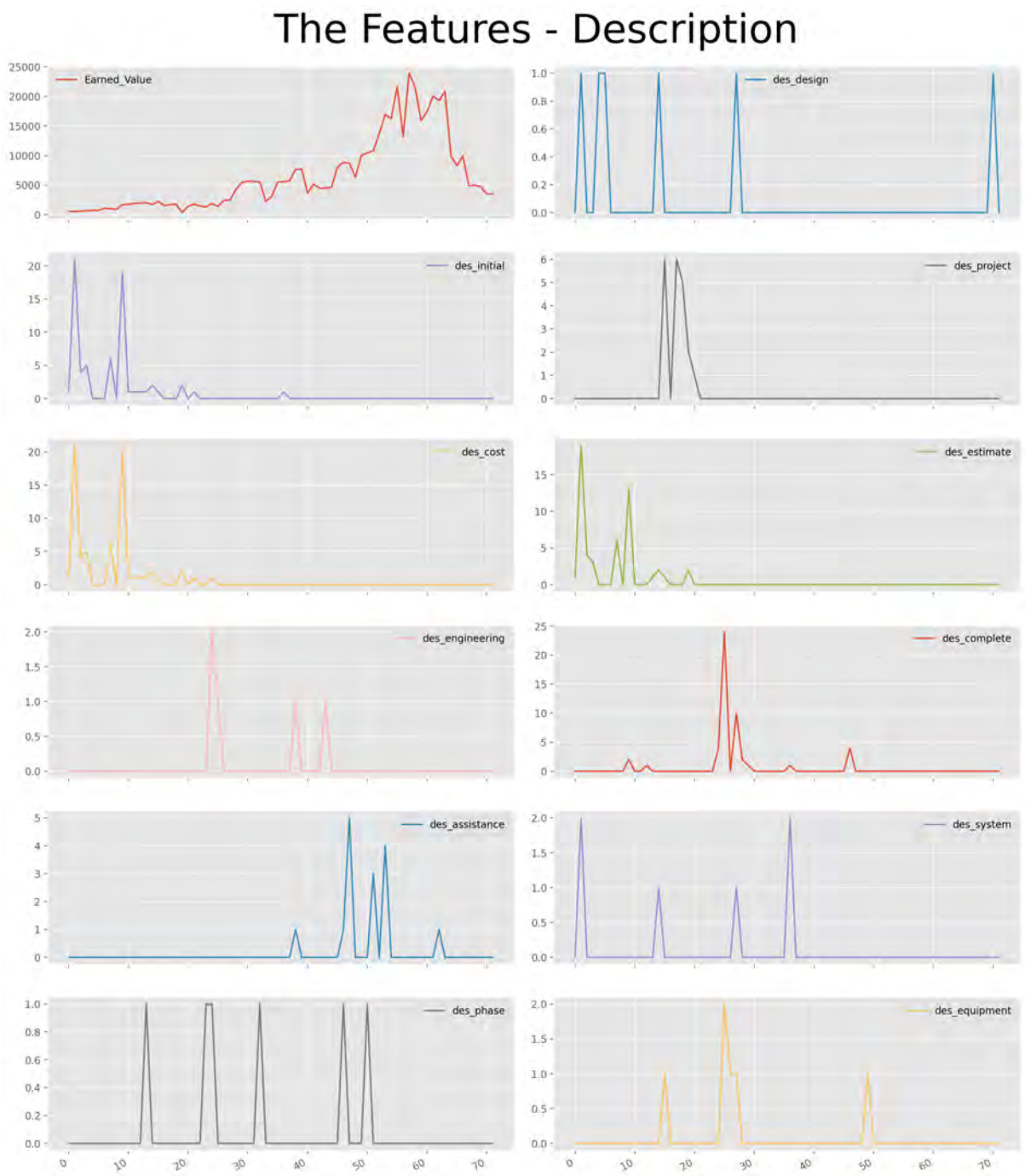


Figure B.2: Features of the description columns type.

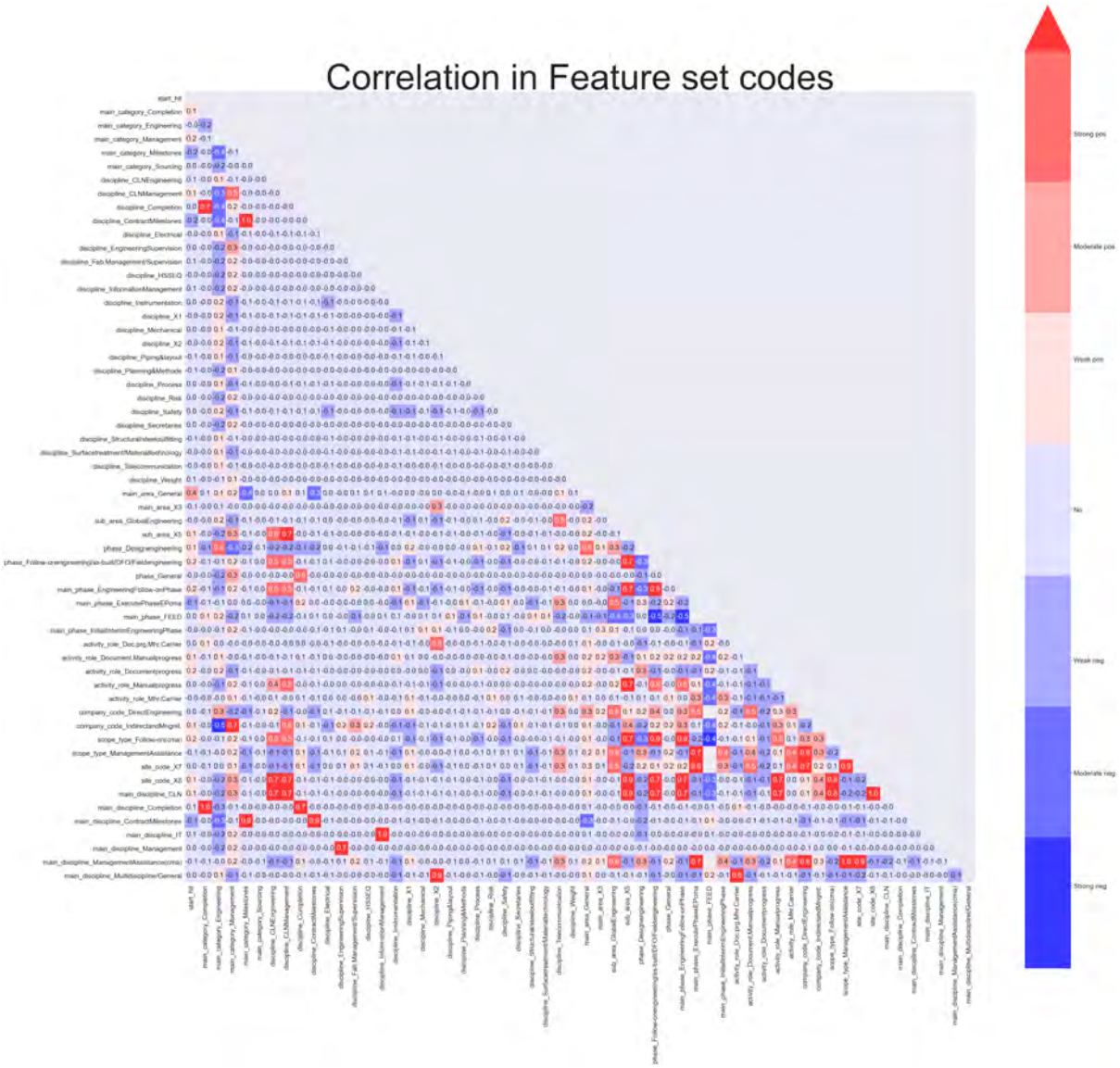


Figure B.3: Features of the codes columns type.

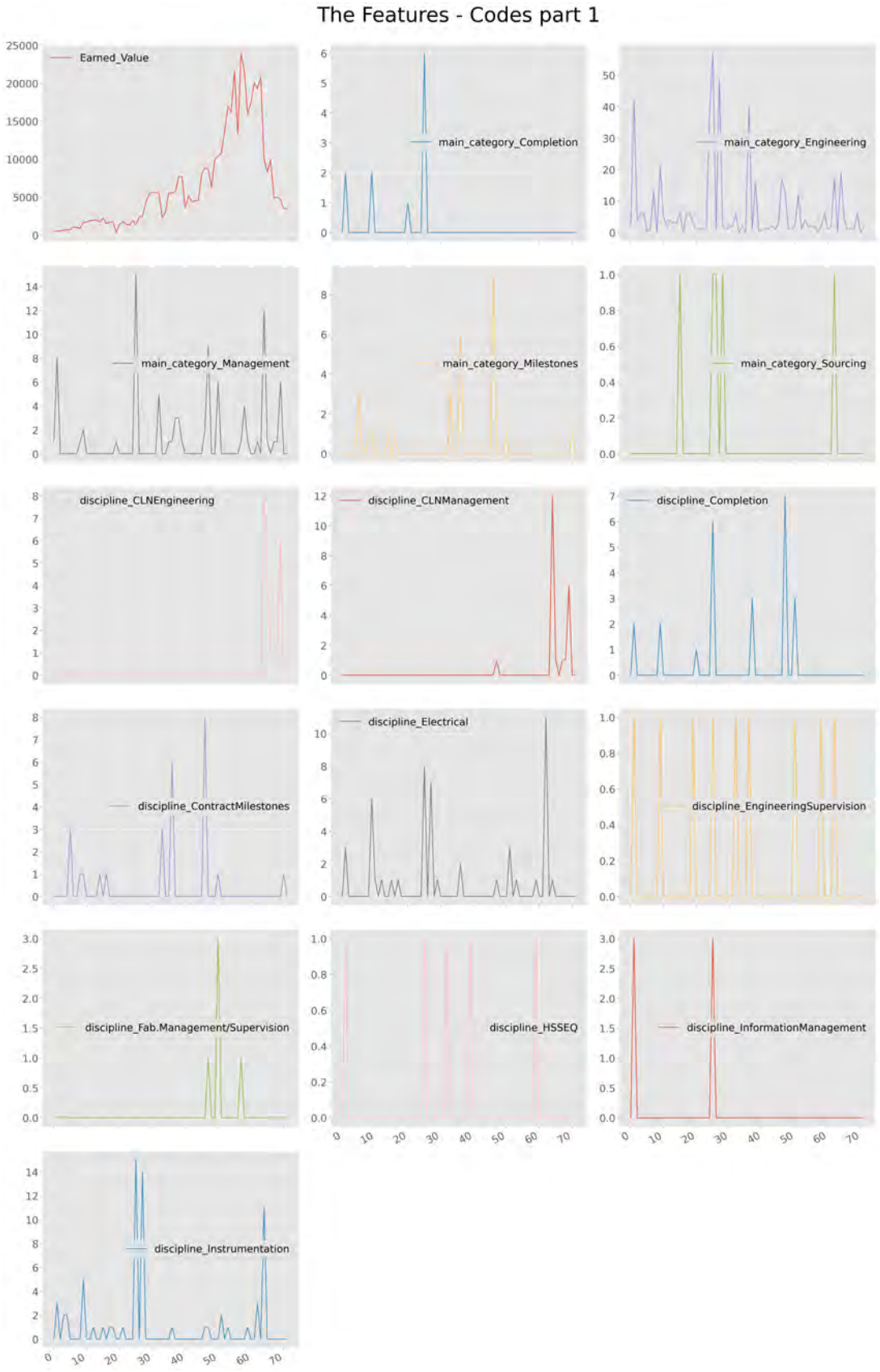


Figure B.4: Features of the codes columns type, part 1.

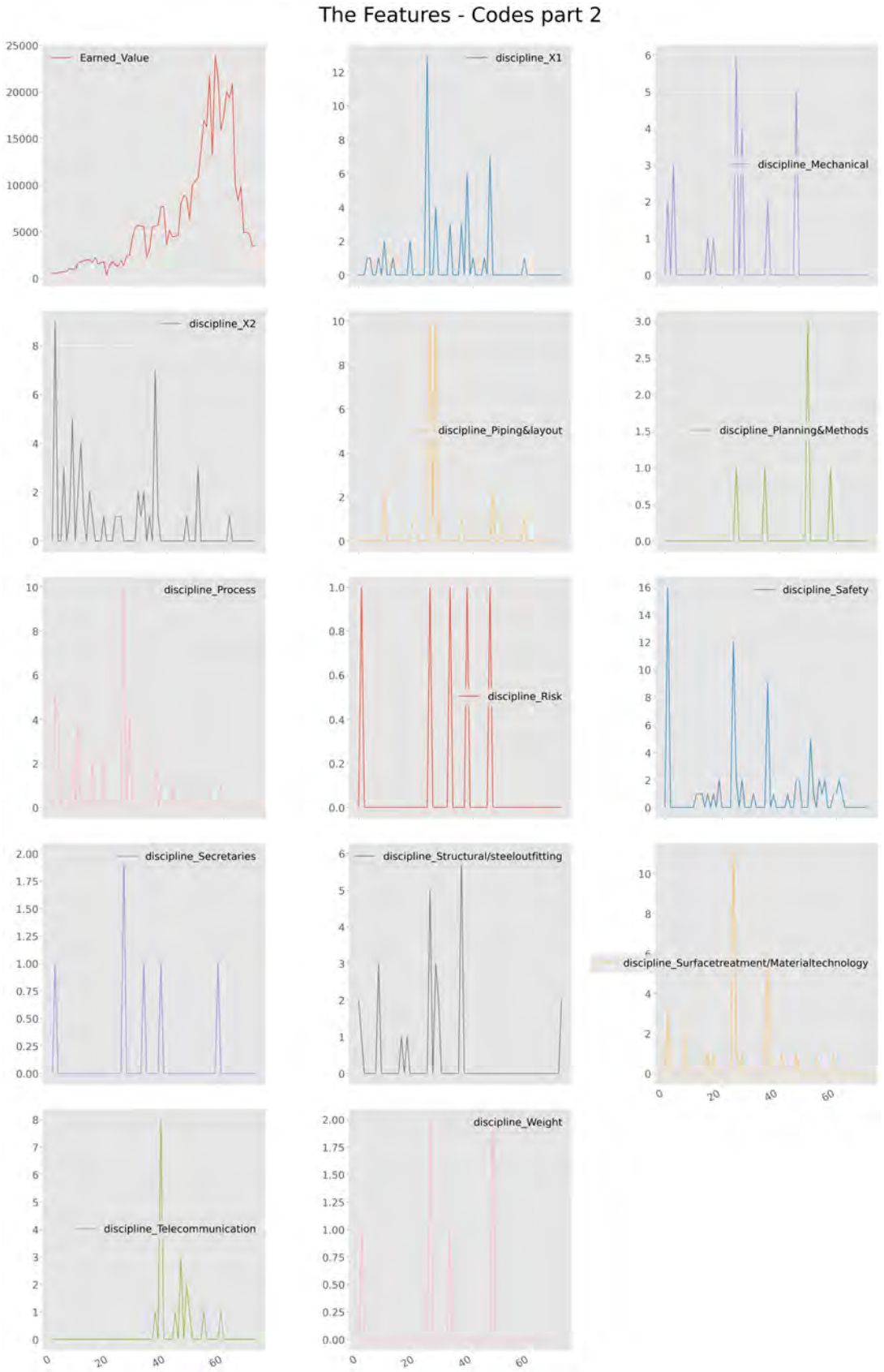


Figure B.5: Continuation of Figure B.4. Features of the codes columns type, part 2.

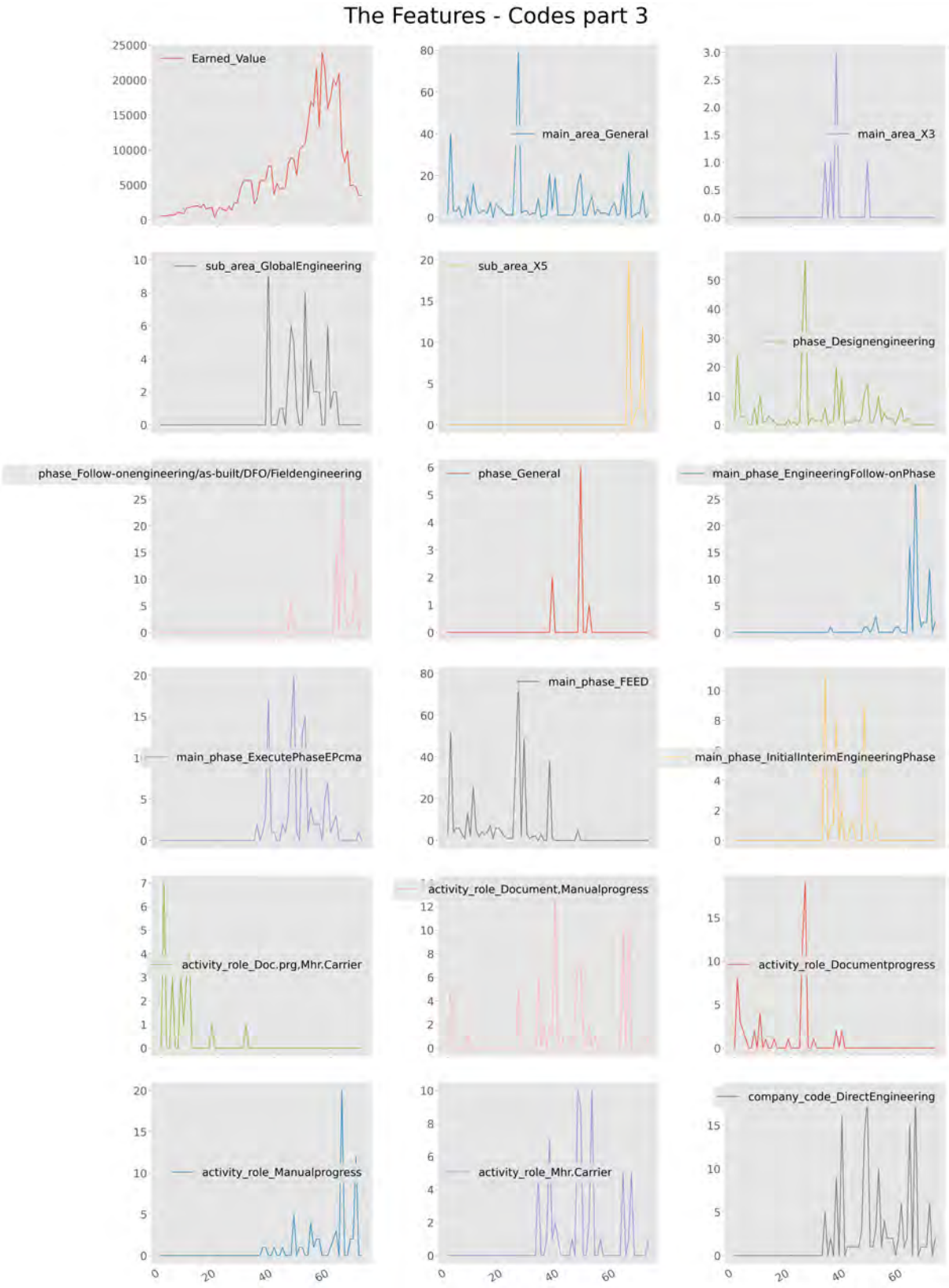


Figure B.6: Continuation of Figure B.4. Features of the codes columns type, part 3.

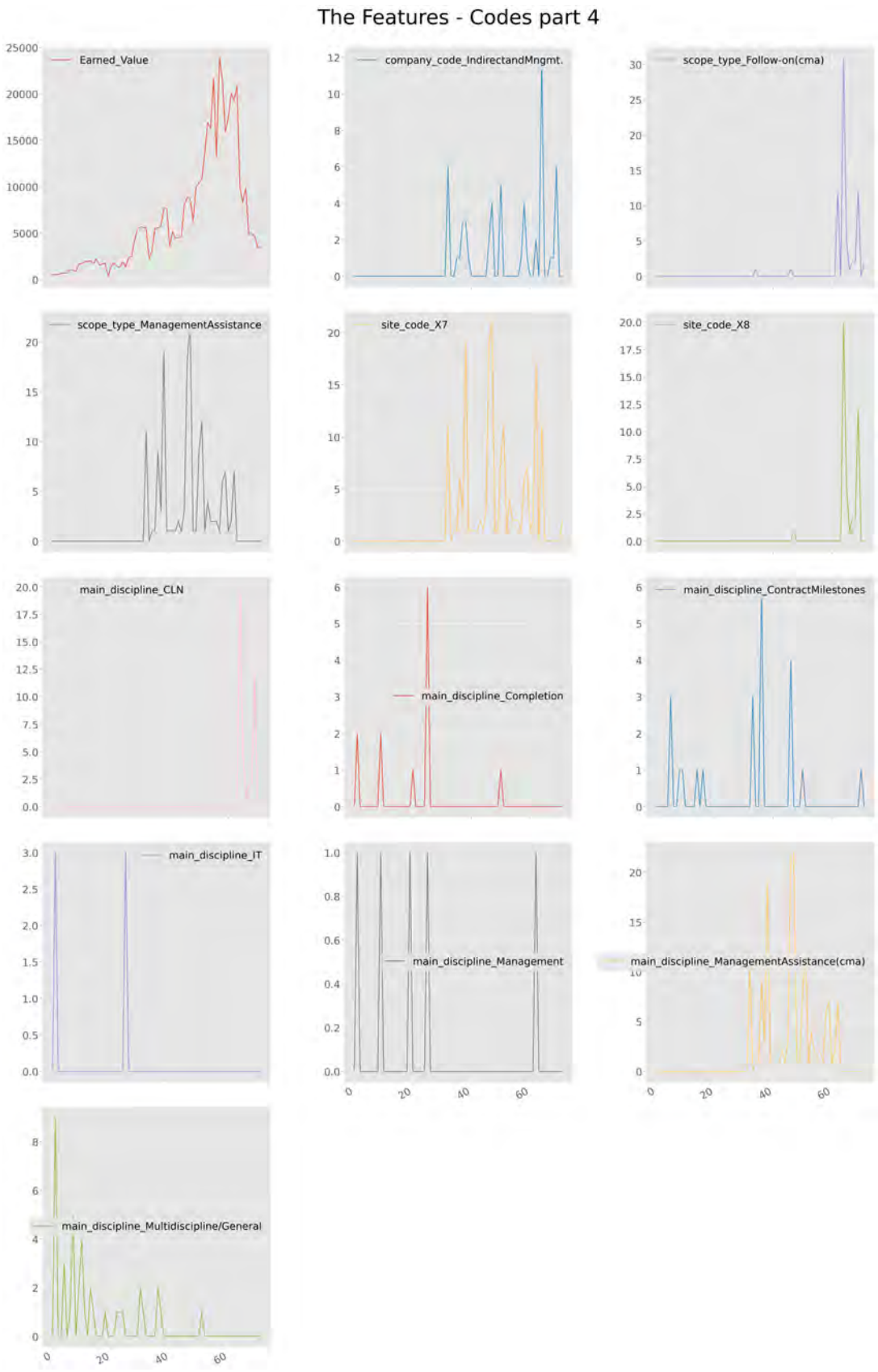


Figure B.7: Continuation of Figure B.4. Features of the codes columns type, part 4.

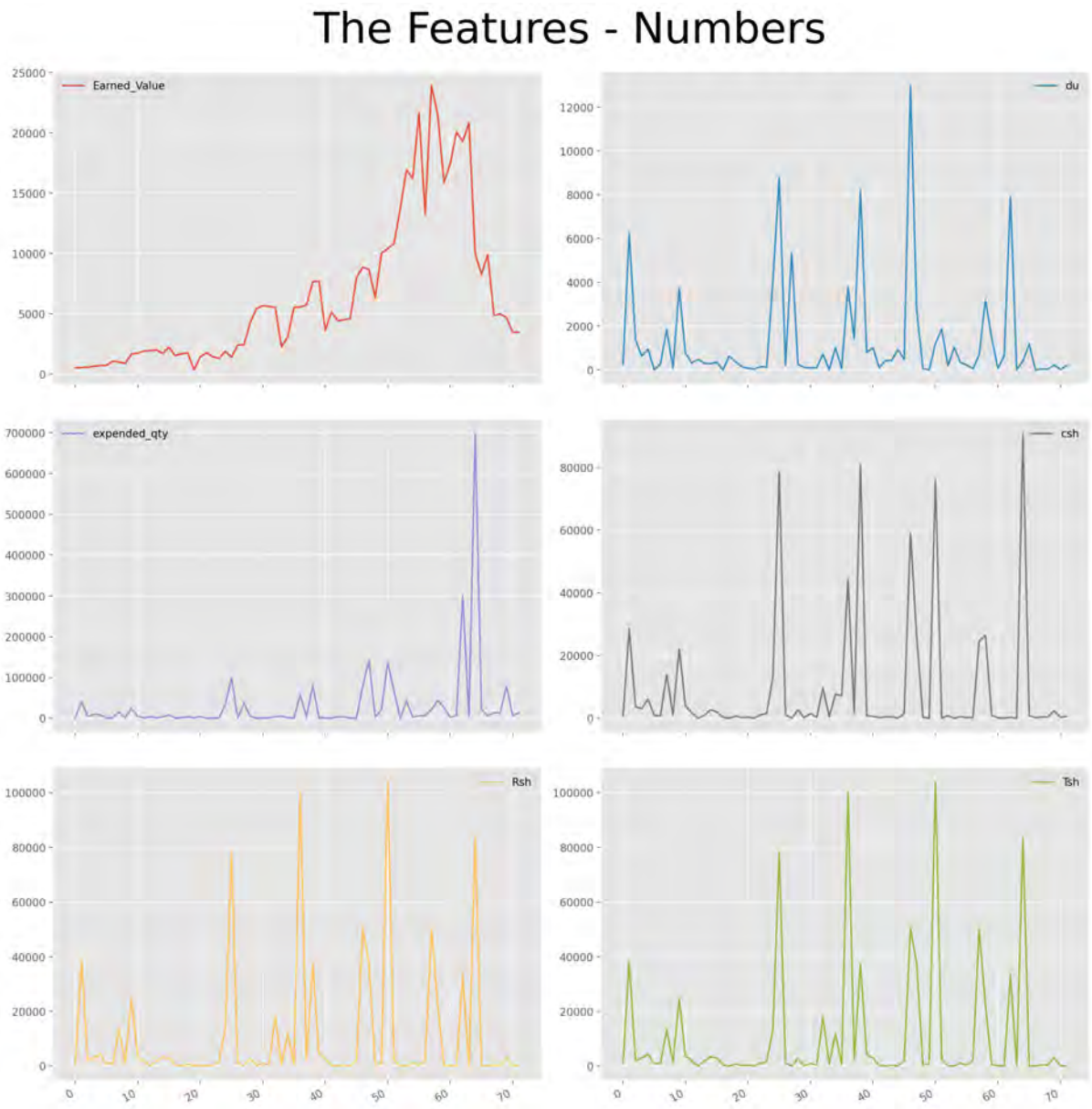


Figure B.8: Features of the number columns type.

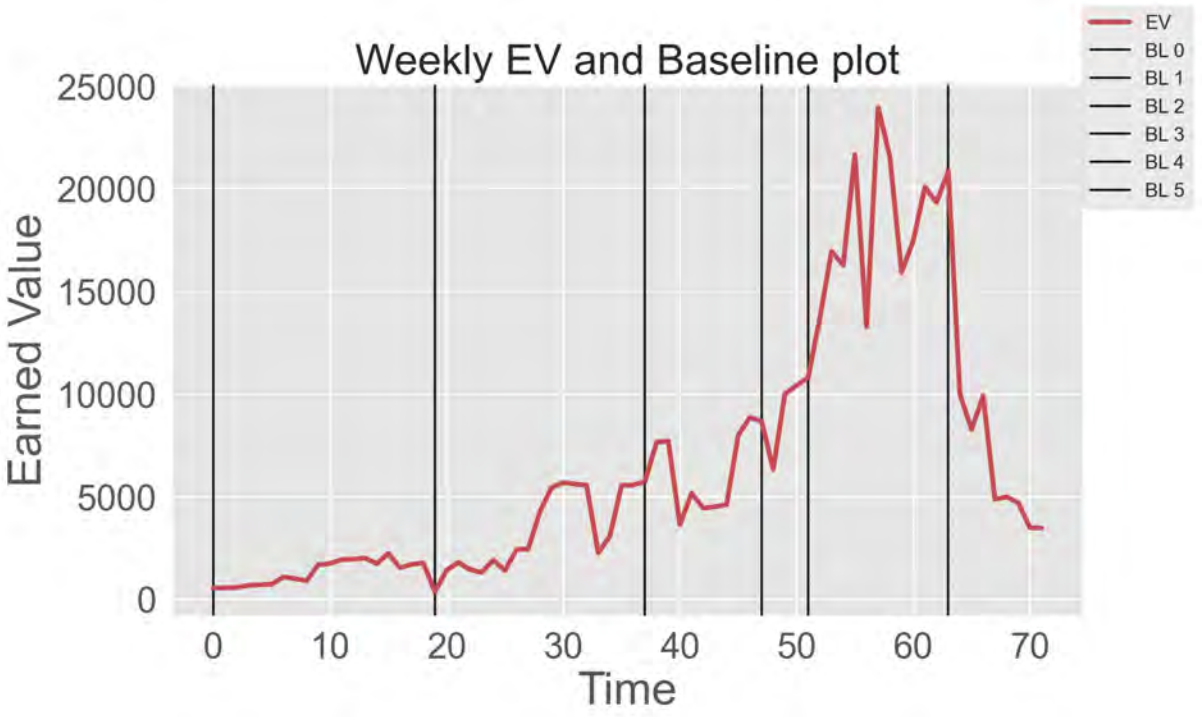


Figure B.9: The baselines of the project.

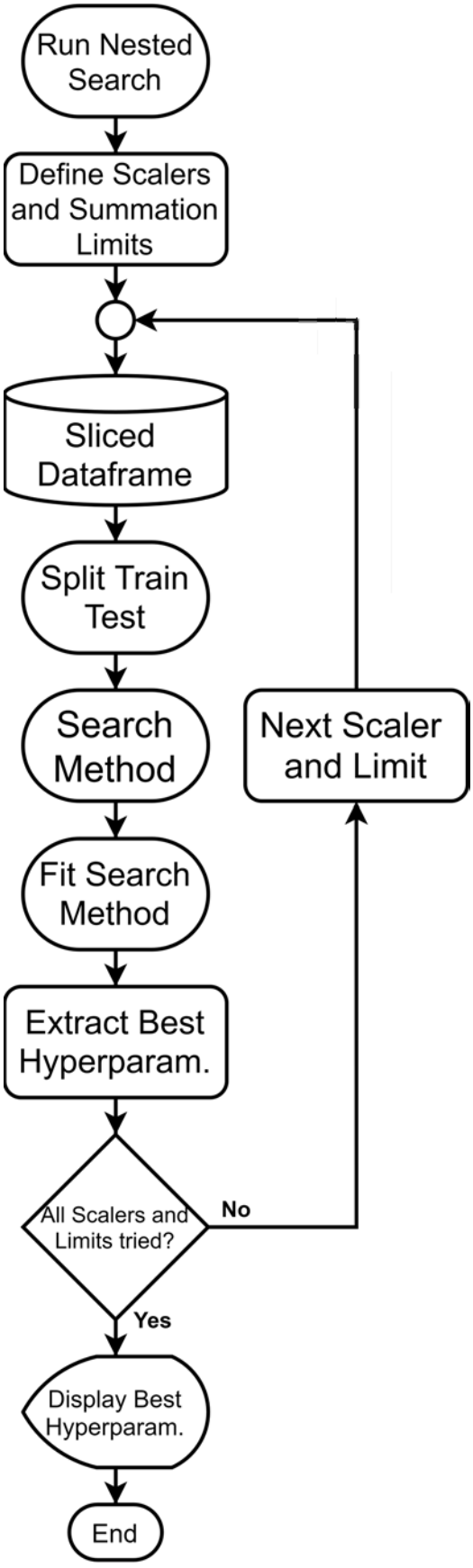
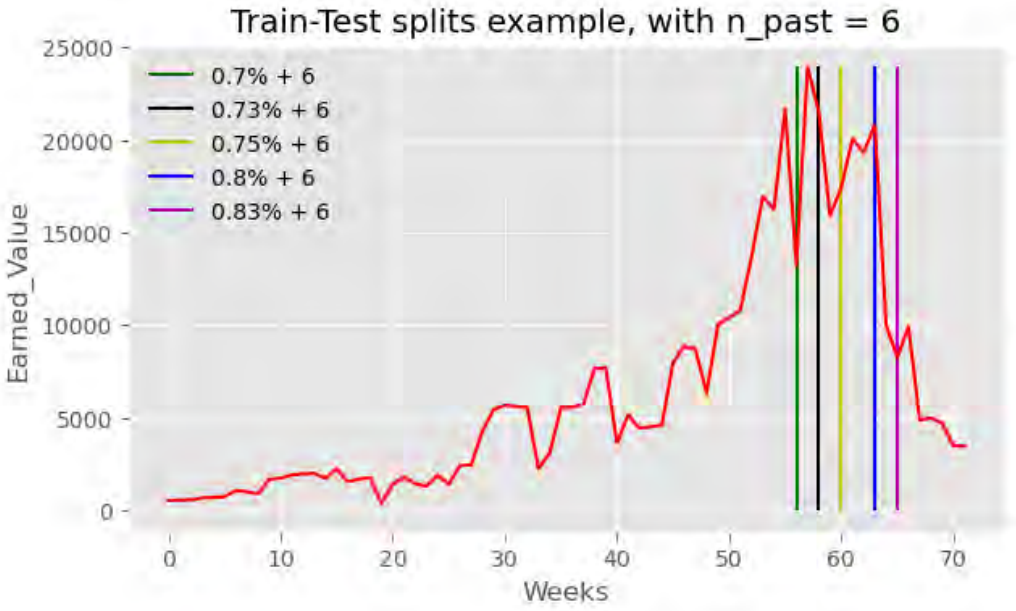
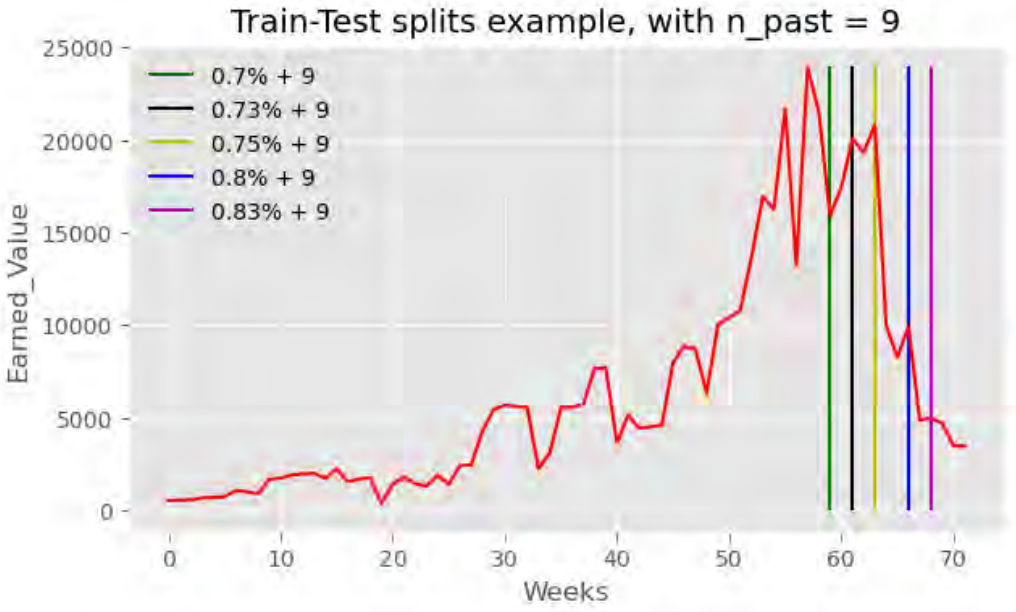


Figure B.10: Nested search method illustration.



(a) n_past = 6.



(b) n_past = 9.

Figure B.11: Where the dataset splitting tests in the grid search were set.

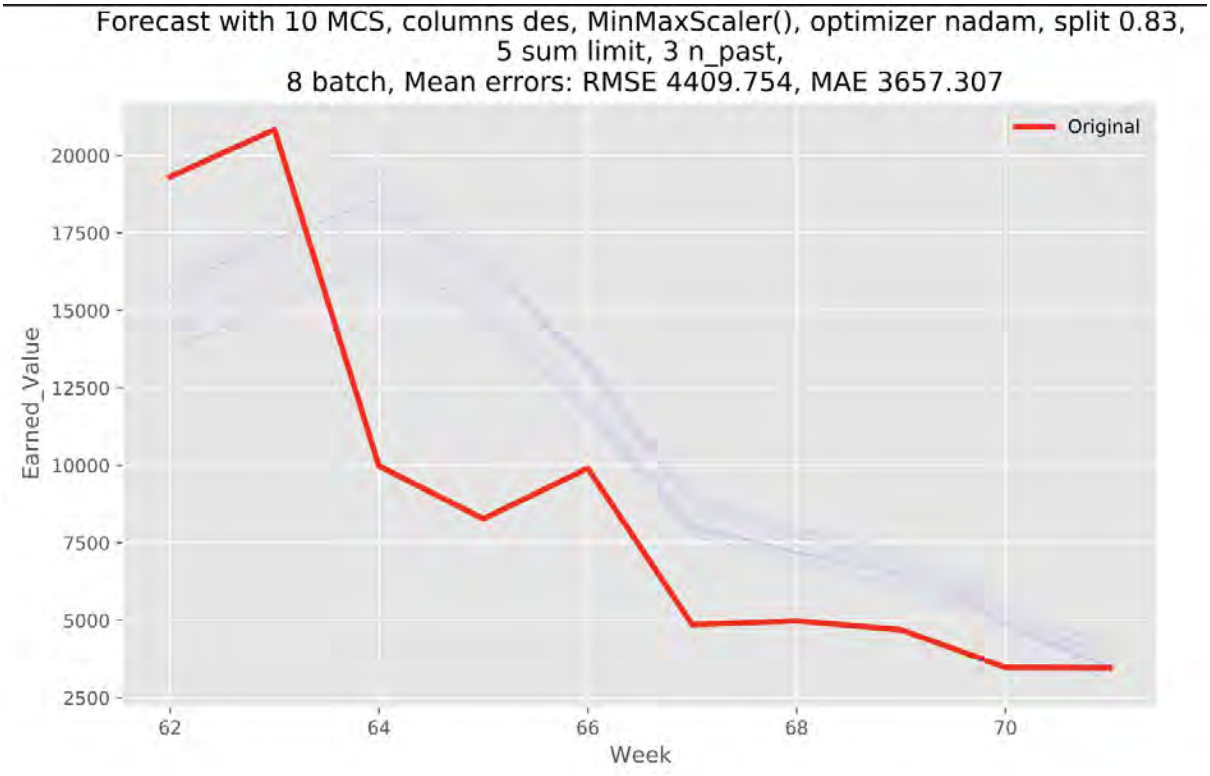


Figure B.12: Plot of the forecast in every MCS with MinMaxScaler and best configuration of hyperparameters

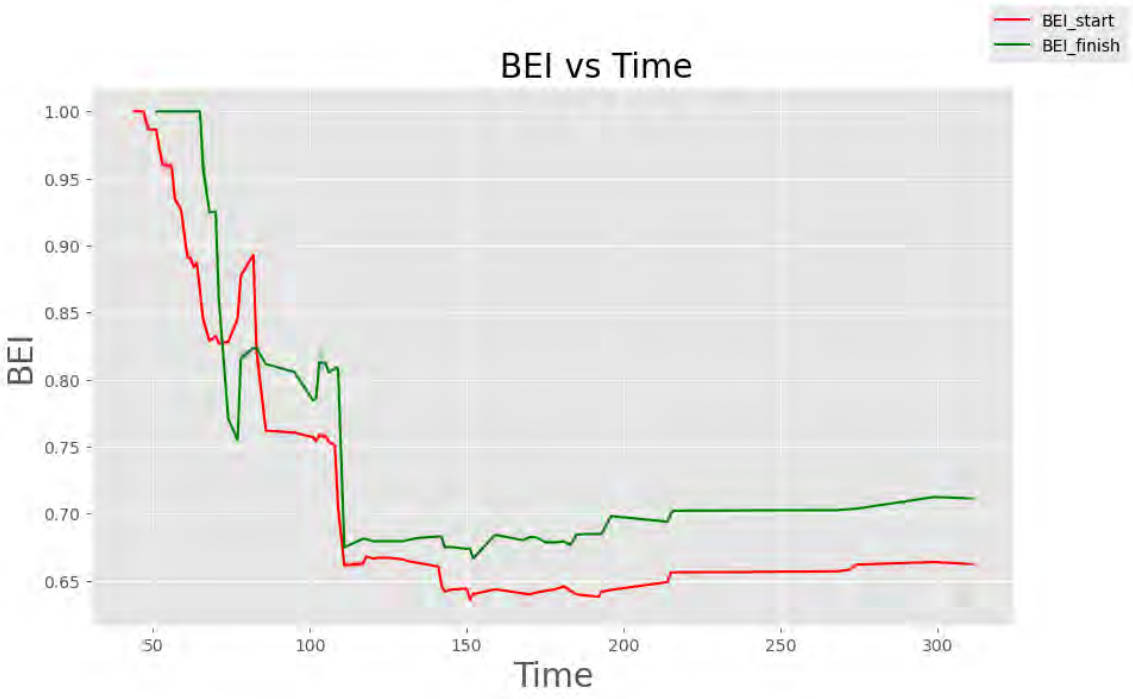
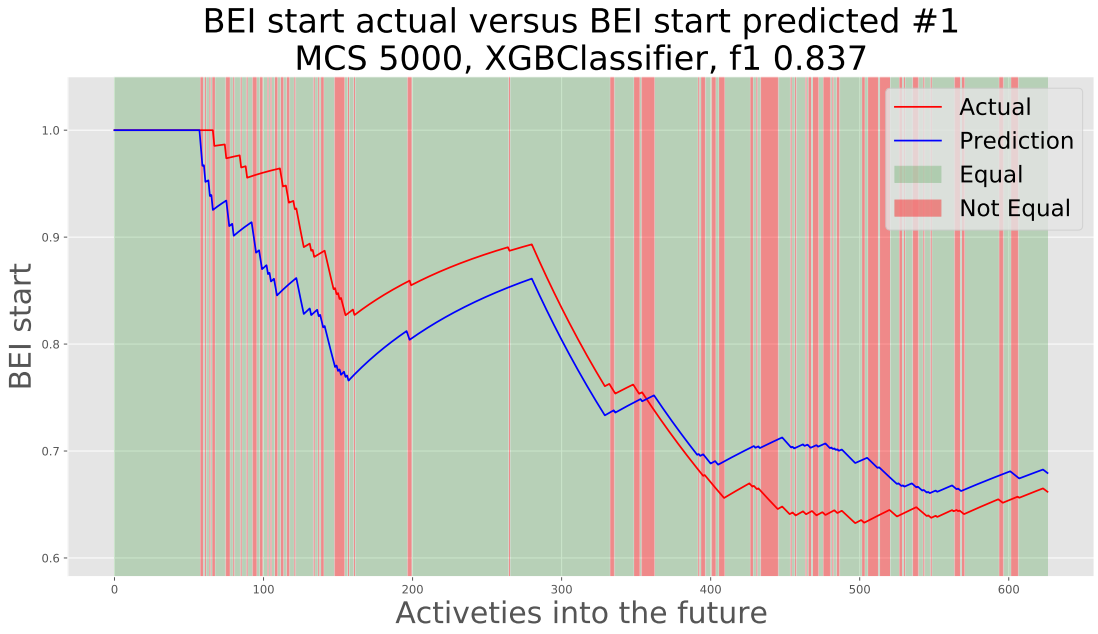
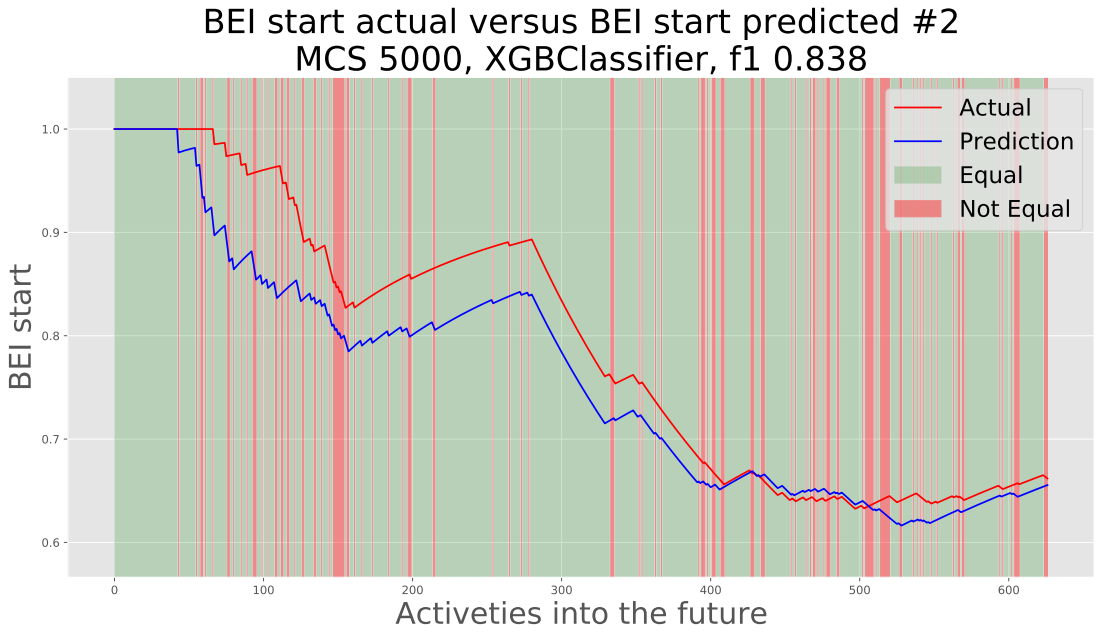


Figure B.13: How the BEI start and finish interacts over time. The index is not reset in this figure.

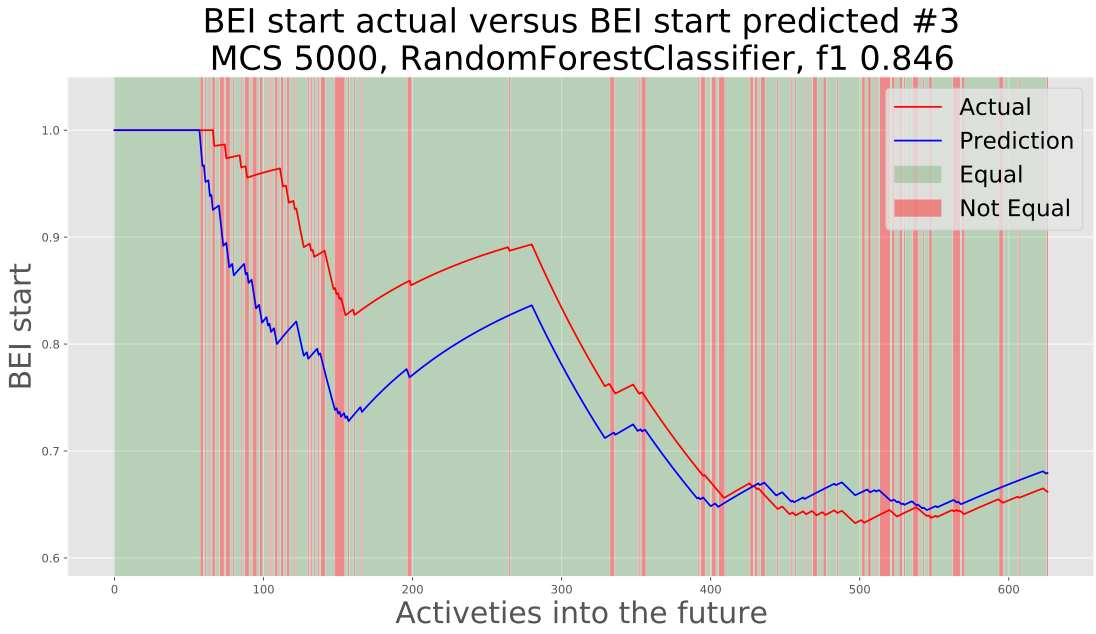


(a) XGBClassifier 1. Test 3.

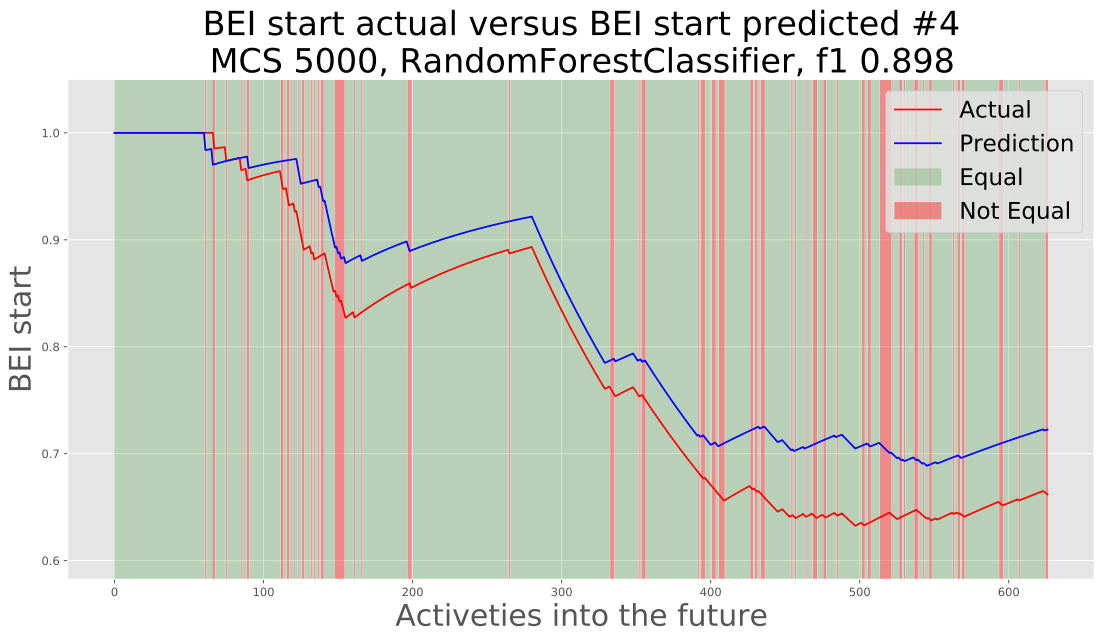


(b) XGBClassifier 2. Test 3.

Figure B.14: Mean binarized predictions from 5000 MC simulations, **shuffle = True**.

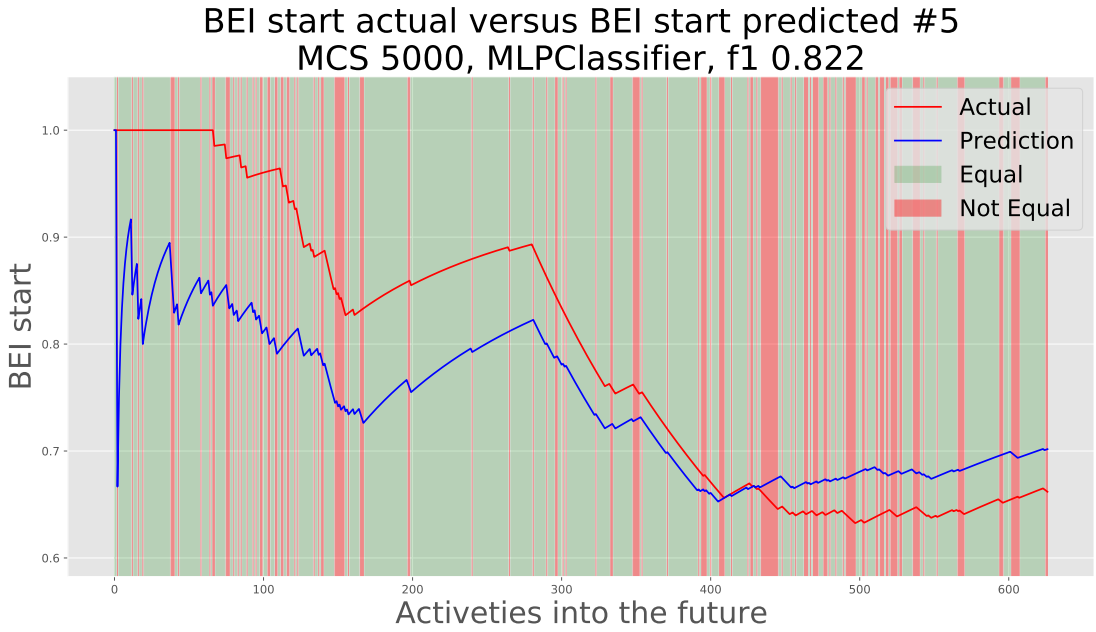


(a) RandomForestClassifier 1. Test 3.

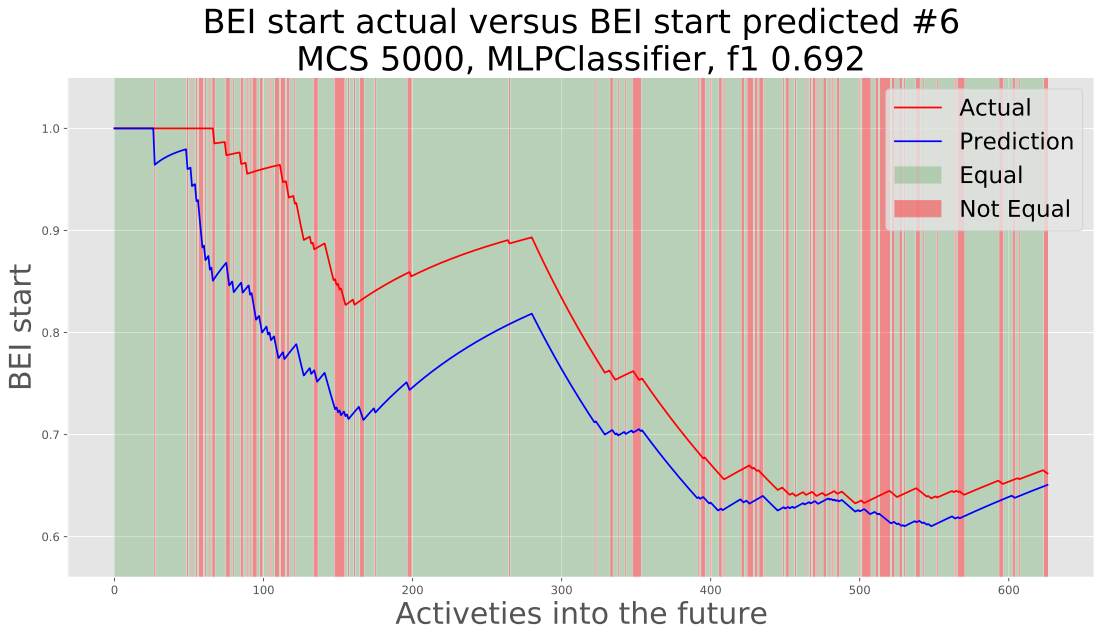


(b) RandomForestClassifier 2. Test 3.

Figure B.15: Mean binarized predictions from 5000 MC simulations, **shuffle = True**.

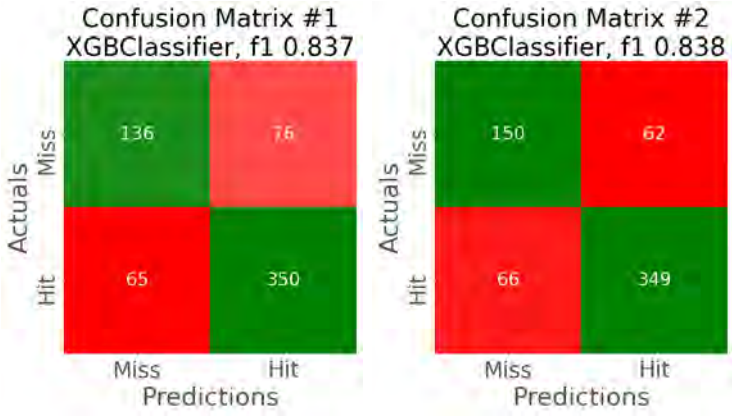


(a) MLPClassifier 1. Test 3.



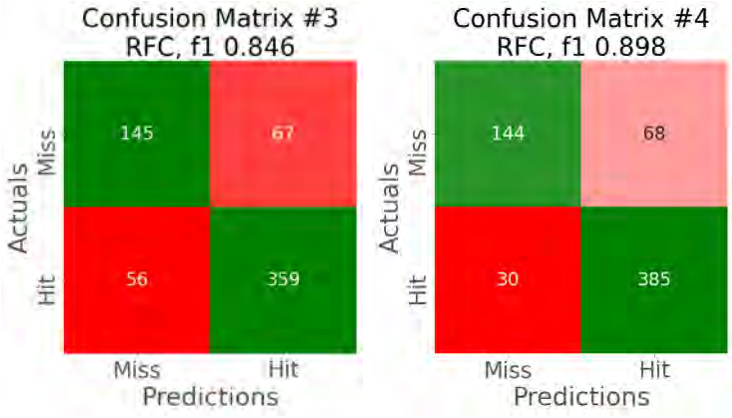
(b) MLPClassifier 2. Test 3.

Figure B.16: Mean binarized predictions from 5000 MC simulations, **shuffle = True**.



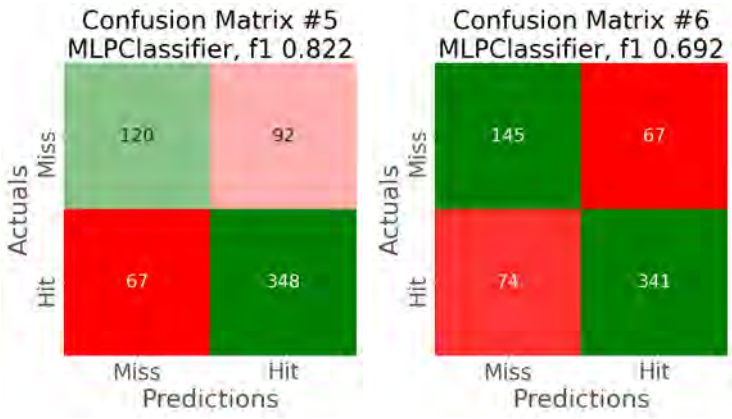
(a) **(b)**

Figure B.17: Mean binarized CM from 5000 MC simulations of XGBClassifier, **shuffle = True**. Test 3.



(a) **(b)**

Figure B.18: Mean binarized CM from 5000 MC simulations of RandomForestClassifier, **shuffle = True**. Test 3.



(a) **(b)**

Figure B.19: Mean binarized CM from 5000 MC simulations of MLPClassifier, **shuffle = True**. Test 3.

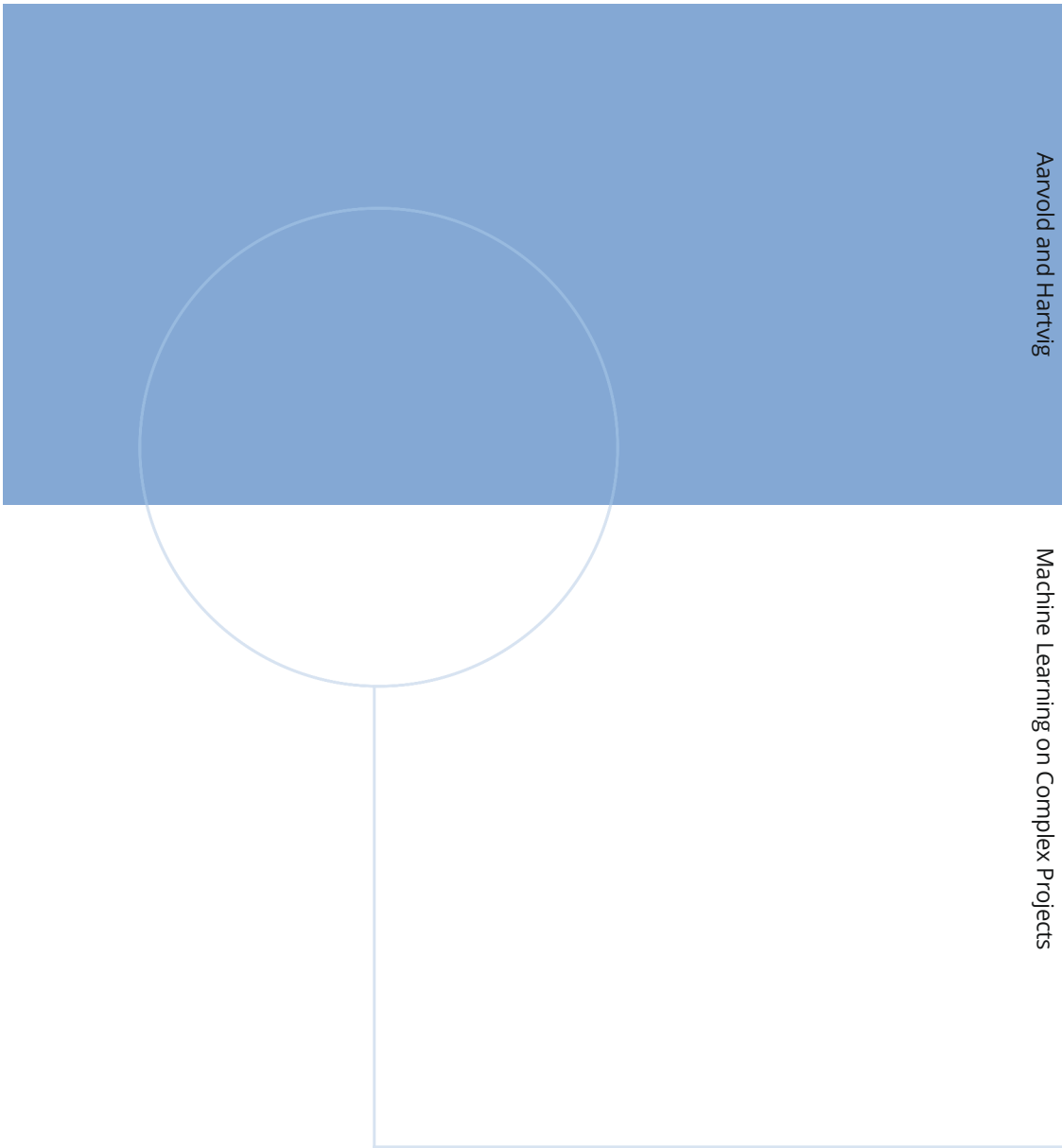
B.3 Code Listing

```
classifier_list = [

        XGBClassifier(learning_rate=0.001, max_depth=6,
                        min_child_weight=8,
                        n_estimators=100, n_jobs=-1,
                        subsample=0.4,verbosity=0,
                        use_label_encoder=False),

        XGBClassifier(base_score=0.5, booster='gbtree',
                        colsample_bylevel=1, colsample_bynode=1,
                        colsample_bytree=1, gamma=0, gpu_id=-1,
                        importance_type='gain',
                        interaction_constraints='',
                        learning_rate=0.5,
                        max_delta_step=0, max_depth=4,
                        min_child_weight=19,
                        monotone_constraints='()', n_estimators=1000,
                        n_jobs=-1, num_parallel_tree=1,
                        random_state=42, reg_alpha=0,
                        reg_lambda=1, scale_pos_weight=1,
                        subsample=0.7500000000000001,
                        tree_method='exact',
                        validate_parameters=1, verbosity=0),

        RandomForestClassifier(bootstrap=False, criterion="gini",
                        max_features=0.25, min_samples_leaf=17,
                        min_samples_split=11, n_estimators=100),

        RandomForestClassifier(ccp_alpha=0.0001, max_depth=50,
                        min_samples_leaf=4, min_samples_split=5,
                        n_estimators=118),

        MLPClassifier(alpha=0.001, hidden_layer_sizes=(20, 20),
                        max_iter=300, random_state=42),

        MLPClassifier(random_state=42, max_iter=1000,
                        hidden_layer_sizes=(20,10), alpha=0.0005),
    ]
```

Listing 1: Classifier Models

```
random_grid =
            {'n_estimators': [10,20,30,40,50,60,80,100,300],
                   'max_features': ['sqrt', 0.2,0.3],
                   'max_depth': [6,10,14],
                   'min_samples_split': [4,5,6,8,11],
                   'min_samples_leaf': [2,4,6,17],
                   'bootstrap': [True, False],
                   'ccp_alpha': [0.00005,0.00008, 0.0001, 0.0015],
                   'criterion': ['entropy', 'gini'],
                   'class_weight':['balanced', 'balanced_subsample']
            }
```

Listing 2: Random Grid parameters



Aarvold and Hartvig

Machine Learning on Complex Projects

NTNU

Norwegian University of
Science and Technology