Data-driven corrective maintenance

MR root cause analysis from machine logs

Shahrestani, A.

Award date:
2020

Department of Industrial Engineering and Innovation Sciences
Information Systems Research Group

Data-Driven Corrective Maintenance: MR Root Cause Analysis from Machine Logs

Master Thesis

A. (Arash) Shahrestani BSc.

In partial fulfillment of the requirements for the degree of
Master of Science, in Operation Management & Logistics

Supervisors:

dr. Ir. H. (Rik) Eshuis,	TU/e, Information Systems
dr. Y. (Yingqian) Zhang,	TU/e, Information Systems
dr. W.L (Willem) van Jaarsveld,	TU/e, Operations Planning Accounting & Control
dr. Ir. G. (Godfried) Webers,	Philips, MR Customer Services

Eindhoven, January 24th, 2020

TUE. School of Industrial Engineering.
Series Master Thesis Operations Management and Logistics

Keywords: Root Cause Analysis, Corrective Maintenance, Machine Learning, Multi Class Classification, Market Basket Analysis, Data-Driven, Failure Mode Analysis, First Visit Fix

v1.1, Classification: Public

ABSTRACT
This master thesis describes a graduation project conducted at Philips Healthcare with the objective to explore possibilities that root cause analysis based on historical cases can provide, to support maintenance decision making for on-site corrective maintenance cases with part replacement. A description of the current maintenance policy is presented and how the developed and evaluated solution can contribute to solve customer complaints in as few visits as possible; by predicting required spare parts – root causes – completely data-driven, along with additional complementary tools based on association mining, given specific part usage.

MANAGEMENT SUMMARY

Background

This Master's thesis focusses on improving on-site corrective maintenance activities with part replacement of Philips Healthcare, specifically for magnetic resonance systems – a project in collaboration with the customer services department.

Problem Orientation and Definition

Field services are heavily focused on customer satisfaction. When something goes wrong, the client would like the issue to be solved straight way; putting processes on hold as little as possible and losing both time and money as few as possible, while an engineer attempts to rectify the issue. Although historically productivity of the workforce and its utilization, mean time to repair and on-time performance have received attention measuring field service performance, organizations are now slowly beginning to track 'First Visit Fix' as a vital metric of field service efficiency. Generally, this term refers to *'solving a service work order during the first customer on-site visit'*.
Although FVF rates are improving over the past couple of years, difference in average cost between single and multiple visits CM cases with the use of at least one spare part replacement, is increasing. Potentially, field service improvements can be made to increase the metric rate, decrease the total maintenance cost difference, and improve customer satisfaction and retention on the long term. Additionally, knowledge of why and how CM cases are structured is often implicit, and provided documentation field engineers might not always provide concrete service actions for occurring issues in practice, decisions about replaceable parts and ordering thereof are hence made ad-hoc.

The goal of this research project is to improve the cost-effectiveness of on-site corrective maintenance operations for certain MR devices by exploring ways to transition the ad-hoc decision process for part replacement during CM activities to a predictive-based fully data-driven corrective maintenance approach. Based on the problem context, and gap analysis resulting from discussions with the project team and aforementioned department, the following main research question is formulated:

"How can we accurately predict the required spare parts for system' component failure based on available data?"

Method of gathering data and analysis

To bridge the gap between the current and desired situation, the objective is to choose a solution design for fully data-driven RCA and provide complementary data and decision rule visualizations to the predicted required part type for a given customer call. A solution direction is chosen based on the comparison of several classifiers. The tool is created on a computational modeling environment called 'R' – statistical programming language. This language is used, as it also is a standard programming and analysis language of the Philips Healthcare department of this project's context. In order to thoroughly answer the aforementioned research question, the widely used Cross Industry Standard Process for Data Mining (CRISP-DM) methodology is used for a structured report. This methodology splits the data mining process into six major phases. Business understanding is the initial phase and focuses on establishing a deeper understanding of the problem and business context, understanding underlying causes, and the current MR troubleshooting process. The subsequent data understanding phase relates to identifying which relevant data exists in the first visit fix and corrective maintenance cases with part replacement context, where this is located and how the quality of the data can be identified. Data preparation focuses on creating useful and suitable data sets for the prediction models. Specifically for this study, all collected errors related to customer calls, during a two week period prior to a call, have also been analyzed for independence and causality by performing error sequence analysis. Additionally, the vast amount of spare parts have been grouped in understandable and clear clusters. During the modelling phase different machine learning

techniques are discovered, selected and applied for intended predictions, which are tested and validated in an evaluation phase. Although CRISP-DM seems sequential, it can be iterative.

The analysis and model creation is performed by writing and comparing three machine learning techniques in 'R': XGBoost, Random Forest, and Structured Vector Machines. Model variations are created per technique based on the four different studied MR system chains (labeled C1 - C4) given the scope of the study, and variations based on different data re-sampling, if required. The selection for these methods is based on ranking different supervised learning algorithms on criteria such as prediction accuracy, result interpretability, sparse data handling, and others.
Based on the output – part predictions – of the models, additional analysis such as association rule mining is provided related to the prediction, and decision rules are visualized with a single decision tree algorithm. This results in a three-phase solution design in which, 1) a general part cluster is determined of which spare parts are required to solve a customer CM call, based on case, error and system data, followed by 2) a specific part (type) prediction given the – in phase 1 – predicted part cluster, and 3) additional (listed) specific spare parts required if solely replacing the – in phase 2 – predicted part does not solve the CM issue, in terms of association rule visualizations. Additionally, decision rule visualization is provided, if required for service engineers. These phases form the full RCA provided solution.

Overview of main finding

The Random Forest and Structured Vector Machine base-models have an average balanced accuracy performance (.52-.62), while XGBoost clearly scores significantly higher than other estimators do with very good balanced accuracy of .72 - .84. Moreover:

- Performance of the XGBoost models per MR chain are overall much better than those obtained for Random Forest and SVM, while the last types show similar results with a few differences for all considered performance metrics.
- XGBoost models are able to reach very good specificity values (ranging from .94-.97) (slightly better than other classifiers, which are still great specificity scores) and decent precision outcomes (.60-.70).
- Precision values are relatively low for all Random Forest models (ranging .29-.47) and SVM's (ranging .30-.40), while recall values are slightly better, but moderate at best (ranging .43-.51 and .39-.62, respectively); with an exception of high recall for RFC C2 (Smote) of .74 and SVM C4 (Linear) of .67.
- XGBoost models are the only classifiers with ideally high precision and recall values, as this results in models returning many correctly labeled results. However, XGBoost C3 and C4 models have lower than preferred recall values compared to C1 and C2, which can results in models returning fewer results but mostly correctly predicted compared to training labels.
- Lastly, the same behavior in performance can be observed for all classifiers, where C1 and C2 models tend to perform better overall, compared to C3 and C4. While for C4 this can be explained due to the much more complex system design and errors that are more dependent and seem to behave differently.

The final results for the best classifier predicting the required spare part cluster given service work order data, logged errors and system registry data, after parameter optimization (*Table i*):

Table i - Classifier performance after parameter optimization

Classifier		Evaluation Metric				
		μ Recall	μ Specificity	μ F1	μ AUC	Kappa
XGBoost	Chain 1 Cases	0.71	0.95	0.76	0.80	0.69
	Chain 2 Cases	0.70	0.98	0.68	0.80	0.66
	Chain 3 Cases	0.55	0.95	0.57	0.73	0.55
	Chain 4 Cases	0.52	0.96	0.55	0.71	0.54

After a spare part cluster is predicted based on the highest probability over all different classes, a Random Forest classifier is able to predict a more specific spare part based on the data of that cluster. As an example, the following results are obtained for a certain chain, and cluster:

Table ii - RFC performance Example, after Part Cluster prediction

	Anterior Coil	Base Coil	Body Coil	Breast Coil	Circu-lator	Coil Assembly	External Coil	Flex Coil	Foot Ankle Coil	Head Coil
F1 Score	0.29	0.69	1.00	0.95	0.97	0.53	0.87	0.41	0.89	0.74
Balanced Accuracy	0.59	0.82	1.00	0.99	0.99	0.95	0.94	0.63	0.92	0.87

	Head Neck Coil	Head Neck Spine Coil	Knee Coil	NVC Coil	PHC	QBC	RF Amplifier	Shoulder Coil	Wrist Coil
F1 Score	0.93	0.57	0.66	0.85	0.80	1.00	0.88	0.59	0.98
Balanced Accuracy	0.98	0.62	0.70	0.85	0.98	1.00	0.89	0.70	0.98

Along with the performance evaluation of proposed solution, the business impact of the model is also determined in terms of ideal potential savings if a root cause of customer call is determined correctly given the part predictions, soft savings and a more realistic estimation of savings falling in between the first two mentioned saving types.
In addition to the predicted part type from *Table ii*, association rules are discovered and presented – such as the examples of *Fig. i*. This functions as a supportive tool to find additional parts sometimes used when the predict part has been replaced; useful for when a problem is not solved with only the predicted part, or if a part replacement per definition involves other (low-level) parts as well.

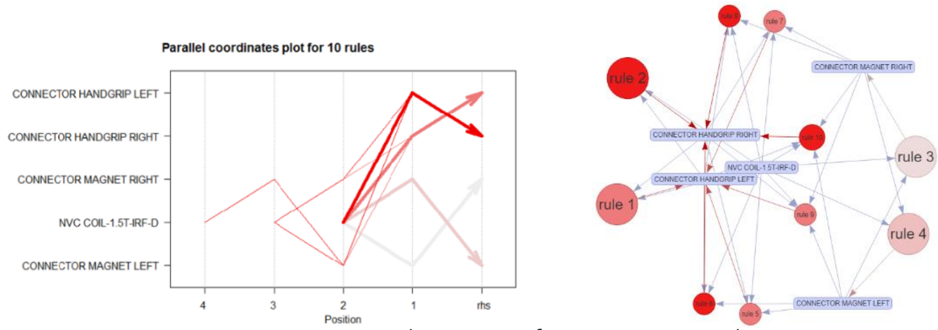


Fig. i – Interactive visualization, significant Association Rule Mining

Future research suggestions

For future research there are several areas were additional research can lead to insights to improve results in terms of data-driven corrective maintenance, prediction accuracy and cost savings. These areas are as follows:

- An important assumption is made with this decision of focusing on four specific MRI chains. Only looking at error data from these chains, means that one assumes a customer call' root cause lies within one of these chains. Not a single data source elaborate on what exactly was the problem of a customer complaint. Thereby, focusing solely on machine behavior and the array and variety of logged errors regarding collecting machine failure information. Hence, for additional research, other chains should be included or a methodology is needed to identify cases with spare part consumptions that under no circumstance can be related to one of the chains and therefore excluded cases from the data set(s). This challenge was also apparent given the model assessment phase where part clusters were included and predicted in test sets that likely cannot be the root cause of certain chains, based on various SME discussions.
- Optional fault descriptions have not been considered during this study, as it requires intensive text mining techniques and dealing with multiple language-related issues as there is also no standardized way of input. But, such techniques can be used to detect patterns and tendencies, structuring free text input to generate an additional data source for RCA.

PREFACE AND ACKNOWLEDGEMENTS

Eindhoven, January 2020

At the moment of writing this chapter, I almost finished my studies at the Eindhoven University of Technology (TU/e). This report is the result of a Master Thesis project, to receive the Master of Science degree in Operations Management and Logistics at the Eindhoven University of Technology; conducted for the Information Systems Group of the faculty 'Industrial Engineering & Innovation Sciences'.

The project has been performed at Philips Healthcare during the past eight months, for which this work is the final deliverable of studying at the TU/e. It therefore marks the end of my student life. During the period of my studies, I learned a lot by the large variety of different courses and projects, both at the TU/e and my exchange period, but also the many educational moments during the graduation internship. Before my masters is officially completed, I would like to take the opportunity to express my gratitude to several people for their support, and motivation during this project and the previous years.

Firstly, I would like to thank my company supervisor Godfried Webers for the excellent supervision during the challenging and interactive project. I enjoyed the weekly meetings – more often twice a week – for the entire period of the internship to discuss various topics of this study in a very structured manner. These sessions and your availability to approach you outside the meetings, for input, opinions, and brainstorming, whenever I faced some challenges were indispensable. Your cooperation, and encouragement to continuously perform better, improving the study on such a timely and head-scratching topic, both for practical application for Philips Healthcare as well as university' academic standards are greatly appreciated.
Thank you for also including me in the team, which made me feel welcome from the beginning, along with all other colleagues from the department that were very helpful and willing to aid when needed or providing the occasional distractions.

Besides the great supervision from the company, there was strong support from the university as well. Special thanks goes out to Rik Eshuis as my first supervisor for introducing me to the company, giving me the opportunity conducting my graduation thesis at Philips Healthcare. During our regular meetings, Rik provided me with many useful tips and feedback. It was a great pleasure having you as supervisor. Thank you for the always-quick responses to questions, and being available for meetings.
I would also like to thank my second supervisor, Yingqian Zhang. Although we met each other occasionally, I always appreciated your critical thoughts and feedback on data analysis aspects of the study. Additionally, thanks to Willem van Jaarsveld for completing the Assessment Committee.

For introducing me to various healthcare projects, such as societal, lifestyle, and environmental related projects, many thanks to Pieter Van Gorp. My appreciation for your always-enthusiastic cooperation, input, support and interesting discussions throughout a large part of my bachelor and master program. It did not only improve my academic work, but also personal development. I greatly enjoyed our work in preventive health and health promotion studies.
Along with other healthcare related project during my studies, and this work regarding MRI maintenance, I had the opportunity to work on and learn from various aspects within the health domain; improving and aiding in different ways to such an important and challenging topic.

Lastly, friends and family, thank you for all the fun times over the years and your unconditional confidence in me and your support.

"Coming together is a beginning,
Keeping together is progress,
Working together is success"
- Henry Ford

Arash Shahrestani

TABLE OF CONTENTS

TABLE OF TABLES

TABLE OF FIGURES

Table of Abbreviations

Table 1 - List of Abbreviations

Abbreviation	Explanation	Abbreviation	Explanation
12NC	12 digit numerical code, Product ID according to standard article coding system	MRI	Magnetic Resonance Imaging
Aop	Annual Operating Plan	NEMO	No Engineer Material Only
AUC	Area Under Curve	OEM	Original Equipment Manufacturer
BCa	Bias-corrected and accelerated bootstrap	OOB	Out-Of-Bag Error
BN	Bayesian Networks	PFEI	Front End Interface, Electronics rack close to the MRI magnet
CAGR	Compound Annual Growth Rate	PM	Planned Maintenance
CBM	Condition-based Maintenance	r	Pearson correlation coefficient, Pearson's r
Ci	Confidence Interval	R&D	Research and Development
CM	Corrective Maintenance	RCA	Root Cause Analysis
cp	Complexity Parameter	RF	Radio Frequency
CRISP-DM	Cross-industry standard process for data mining	RFC	Random Forest Classifier
CT	Computed Tomography	RHS	Right Hand Side
FAQ	Frequently Asked Questions	ROC	Receiver Operating Characteristics
FMEA	Failure Mode Effect Analysis	ROSE	Random Over-Sampling Examples
FMECA	Failure Mode Effects and Criticality Analysis	RPN	Risk Priority Number
FN	False Negatives	RSE	Remote Service Engineer
FP	False Positives	SD	Standard Deviation
FPG	Frequent Pattern - Growth	SI	Service Innovation
FSE	Field Service Engineer	SME	Subject Matter Expert
FTA	Fault Tree Analysis	SMOTE	Synthetic Minority Over-sampling Technique
FVF	First Visit Fix	SPC	Spare Part Replacement
ixR	interventional X-ray	SPD	Service Procedure Document
JDBC	Java Database Connectivity	SQL	Structured Query Language
KDD	Knowledge Discovery in Databases	SVM	Support Vector Machines
KPI	Key Performance Indicator	SWO	Service Work Order
LHS	Left Hand Side	TN	True Negatives
MAFTA	Multi-Attribute Failure Model Analysis	TP	True Positives
MBA	Market Basket Analysis	TPR	Technical Parts Review
MD	Mahalanobis Distance	TSD	Troubleshooting Document
MR	Magnetic Resonance		

1 INTRODUCTION AND PROJECT CONTEXT

This chapter serves as an introduction to the topic at hand in this document. It starts with elaborating on the increasing need of imaging technology in the global healthcare sector. Subsequently, the chapter continues describing the importance of field service, also regarding these challenges, the introduction of a vital service metric, its impact on business and customer, and potential strategies to improve this vital metric.

1.1 CURRENT CHALLENGES IN IMAGING TECHNOLOGY

In any hospital setting, medical device systems such as Magnetic Resonance Imaging (MRI), Computed Tomography (CT) and interventional X-ray (ixR) are vital equipment, which play an important role in diagnosing, monitoring, screening and testing of medical conditions for medical intervention; in short the diagnosis and treatment of patients. A market study has shown that demand of these devices is increasing daily, with a projected growth rate of 5.2% Compound Annual Growth Rate (CAGR) over a period from 2014 to 2020; $26.5 bn of global diagnostic imaging devices market-valuation in 2014, which is expected to reach $35.8 bn by 2020 (Persistence Market Research, 2016). Additionally, Research and Markets (2016) expects the MRI systems market to register a 6% CAGR between 2015 and 2020. Although technological advancements in diagnostic imaging devices and corresponding imaging technology have propelled the global diagnostic imaging devices market, other factors are expected to contribute further in usage of such devices. It is predicted that the growing aging population, increasing occurrence of injuries, chronic diseases (i.e cancers), neurological and cardiac disorders and the increasing number of applications of diagnostic imaging devices increase this usage even more in the years to come (Persistence Market Research, 2016; Research and Markets, 2016).

The demand trend and current challenges for the industry show the important of these devices in the current healthcare setup. Therefore, it is also important that these type of devices are available to clinicians and potentially other personal, all the time. Any downtime of diagnostic imaging devices can have major implications or complications due to delayed diagnosis of patient' rescheduling in rescheduling and loss of productivity. Especially, in the cost sensitive healthcare industry, any unplanned downtime of such devices can be a burden of both the hospitals as well as the original equipment manufacturers (OEM's) financially. Downtime of these devices is determined as roughly 10% of their total cost over a period of ten years (Hockel & Hamilton, 2011).

1.2 FIELD SERVICE – FIRST VISIT FIX

Nowadays in the era of connectivity, it is easy to connect these devices to an OEM monitoring station. Once connected, OEM's can monitor the health and status of these devices remotely, and take corrective actions based on preventive maintenance strategies, if needed. Although such methods aim to mitigate the risk of unplanned downtime and thereby increasing device usage, it is also important to have an efficient corrective strategy and on-site service once customers do experience, and raise repair requests for sub-optimally working systems, or unexpected device or component failure(s).

Such field services are heavily focused on customer satisfaction. When something goes wrong, the client would like the issue to be solved straight way; putting processes on hold as little as possible and losing both time and money as few as possible, while an engineer attempts to rectify the issue. Although historically productivity of the workforce and its utilization, mean time to repair and on-time performance have received attention measuring field service performance, organizations are now slowly beginning to track 'First Visit Fix' as a vital metric of field service efficiency (AberdeenGroup, 2013). Organizations could have slightly different variations of this Key Performance Indicator (KPI) in terminology and definition but generally, 'First Visit Fix' refers to *'solving a service work order during the first customer on-site visit'*. Each subsequent on-site customer contact for the issue at hand, is no longer considered a first visit fix (Reichheld & Schefter, 2000; Zwetsloot, Buitenhuis, Lameijer, & Does, 2015).

This change in KPI focus is due to its potential wide scale implications; it does not only affect field service performance, but also key customer-oriented and financial measures that reflect business health. A less than satisfactory 'First Visit Fix' measure could also mean an early indicator of unhappy customers, likely decrease in customer attrition, retention and turnover, and reduced service profitability, as service quality has proved to be a determinant of satisfaction and loyalty of customers (Kursunluoglu, 2014; Mosahab, Ramayah, & Mahamad, 2010; Politis, Giovanis, & Binioris, 2014). Which subsequently also provide an indication regarding an organizational perspective with a potential impact on operating cost, margins and turnover (AberdeenGroup, 2013).

Field service research has shown that additional dispatches due to multiple visits required add a cost burden to the service organization, in terms of ordering new parts, but also necessary traveling cost as every additional visit adds $200-300 on average (AberdeenGroup, 2013). Moreover, all these extra visits take field resources away from servicing new repair requests. Less field visits for new work impact revenue as well. Organization with an 80% First Visit Fix rate in this study have experiences a 6.2% increase in service revenues over a period of twelve months compared to 1.6% increase with a sub-80% First Visit Fix rate. For those organizations below 50%, the revenue decreased to 2.8% over the previous year.

Organizations that have taken steps in First Visit Fix performance have experienced significant benefits in the form of reported improved customer satisfaction and retention (*Table 2*).

Table 2 - First Visit Fix and Customer Satisfaction Impact (edited from AberdeenGroup (2013))

Metric	**Average Results**		
	First Visit Fix <50%	**First Visit Fix <80%**	**First Visit Fix >80%**
Satisfaction	46%	64%	87%
Retention	60%	68%	88%
Service Margin	23%	28%	29%

1.3 First Visit Fix – Root Cause

The main reason for multiple visits is determined to be part unavailability; meaning that field engineers did not bring any parts on-site, either none have been ordered or are still in transit, or brought incorrect parts. Hence, First Visit Fix performance is heavily dependent on service parts management. Additionally, engineer' experience has been identified as the second reason; not having the knowledge of which parts to repair for a given system failure, potentially contributing to the final aspect: insufficient time to complete the task. Several remedies can aid in metric improvement, such as a better or more extensive diagnosis of the failure or triage at the call level, scheduling of repairs based on part availability, and improved training of field engineers (AberdeenGroup, 2013). However, none tackle the root cause of the First Visit Fix issue: improved field-based access to (correct) parts.

Using aforementioned system connectivity with OEM's for system monitoring, organizations can use logged data regarding historic repairs and system failure reporting to identify correct parts to bring for on-site repairs. In general: use service history and knowledge base of resolution steps to provide visibility into required parts.

This project investigates the possibilities to support maintenance decisions for Philips, specifically with regard to corrective maintenance cases with consumed parts during on-site Magnetic Resonance repairs, to improve diagnostics and decrease required number of visits; aiming to achieve a higher first time fix rate. These concepts are discussed separately below.

1.4 PHILIPS HEALTHCARE

The project has been conducted at Philips Healthcare in Best – the Netherlands, as one of the leading players in the global diagnostic imaging devices market. Philips is one of the most influential companies in the Netherlands and the Dutch history, being the most valuable Dutch brand with a brand value of $11.7 billion in 2019 (Hinde, 2019). Founded in 1891 in the city of Eindhoven – Noord-Brabant by Gerard Philips and his father Frederik, its mission is to improve people's life through meaningful innovation, and therefore promise that they deliver innovation that matters to you**.** After an initial expanse in vertical integration, followed by a horizontal integration, the company achieved multiple technological landmarks. However, it had to sell several departments and thereby focusing on a few core competencies in order to achieve higher flexibility in a competitive environment (Hinde, 2019).

Philips is known for both its strong presence in the lighting and healthcare industry. Since 2016, the company also split off the former business, which has adopted the name Signify N.V. Nowadays, Philips is a leading provider of Health technology. The company leverages advanced technology and deep clinical and consumer insights, delivering integrated solutions to improve people's health over the entire health continuum. It therefore sees two main business opportunities in this holistic continuum (*Fig. 1*). The business is split into 1) Professional Healthcare, and 2) Consumer Health & Wellbeing; aiming for the:

- Industrialization of care: Standardizing and optimizing the building blocks of healthcare to enable health systems to deliver better outcomes at lower cost.

- Personalization of care: Convergence of professional and consumer healthcare, enabled by digitalization, increasing self-management and individualized treatment.

The graduation project has taken place in the professional healthcare continuum; a subset of healthcare where the organization also fulfills the role of the original equipment manufacturer (OEM) and delivers final products (e.g. MRI-scanner), with corresponding responsibilities and collaboration with suppliers to aid producing the product. The next section elaborated on the specific business unit of Philips where this study is performed.

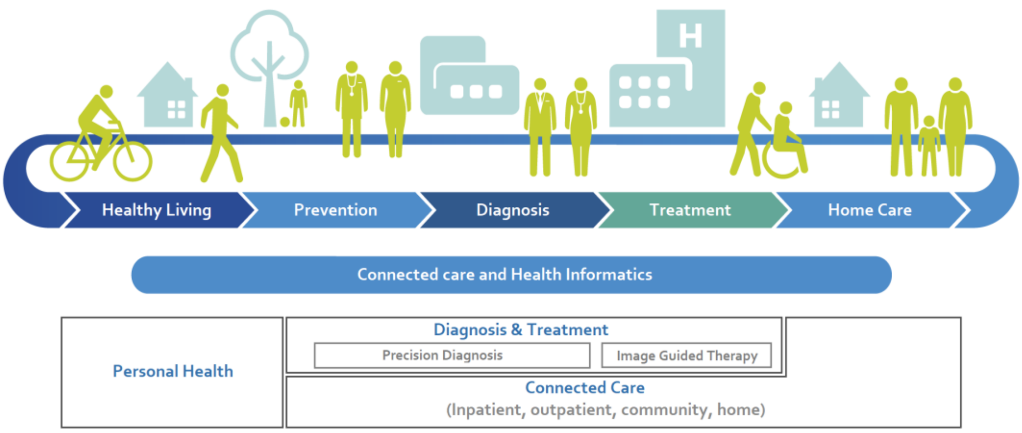


Fig. 1 - Holistic Health Continuum, edited from (Hinde, 2019)

1.4.1 MR Customer Service

Philips' campus at Best focuses on professional Healthcare; among others, one of the product groups focused on, on this site is Magnetic Resonance Imaging (MRI) – scanners which are also the focus regarding this 'First Visit Fix'-related graduation thesis.
Customer service and maintenance of MRI-scanners is the responsibility of the corresponding department: Magnetic Resonance (MR) Customer Service, along with monitory the state of the systems continuously, providing repair services, upgrading the system (software) and providing relevant documentation for their products. The department' goal is to support the Philips' markets in minimizing unplanned downtime by keeping the state of the machines in the field as high as possible. Given the job tasks and goal of the department, it can be categorized in the Image Guided Therapy support within the Diagnosis and Treatment phase of the aforementioned Philips Healthcare Continuum (*Fig. 1*).

1.5 Problem Statement

The organization currently uses corrective maintenance (CM) manuals that describe the required CM activities, replacement procedures, tests and repairs for each product model and part. Subject matter experts (SMEs) in Research and Development (R&D) and Service Innovation (SI) have defined these activities for some of the failures that can occur for a system chain and part, specifically for Field Service Engineers to use during their on-site visits. However, there is no or incomplete documentation explaining the correct parts or combination thereof to use for specific failures modes, ideally based on success stories of similar previous repairs and why certain parts may be ordered (simultaneously) based on FSE's experiences. Knowledge of why and how CM cases are structured is often implicit knowledge of these engineers.
Because this knowledge is implicit, and provided documentation to FSE's does not always provide service actions for occurring issues in practice, decisions about replaceable parts and ordering thereof are made ad-hoc. With regard to RSE's, they can provide recommendations regarding service actions if they have not been able to remotely solve the issue, but do not provide a list of parts to be replace for the situation at hand. It requires significant time and effort in some cases to solve the client's problems as new or additional parts have to be order or returned to successfully close the case in as few visits as possible.

Initial review of the distribution of distinct CM cases over the years 2012 - 2019 is performed, specifically plotting the amount of cases where the case was solved during the first visit (single visit cases) and cases that needed multiple visits to solve (multiple visit cases)); for one specific type of MR system (given the study' scope, see *Section 1.9*). Note that the 2019 data is still partial based on approximately six months of data [1]. However, it is clear that a significant percentage of cases, varying per year, could not be solved in a single visit; yielding an acceptable FVF rate for the years 2012 – 2019 with opportunities to improve. This is specifically for on-site repairs where CM has taken place with at least one spare part replacement, over all available distinct cases within the scope. These percentages are slowly increasing over the years for the specific in-scope MR system, which could be explained due to engineer' experience on repairing the five different *Ingenia* machines. However, net repair cost, differ for CM cases with single or multiple visits significantly as explained later on. *0* and *Section 7.3* provide detailed analysis and explanation of the FVF rate(s) analysis, based on corresponding research objectives. Overall, we can observe that multiple visit repairs cost significantly more, even solely based on spare parts (excluding labor and travel costs). Hence, there is a great opportunity to reduce service cost; in terms of hard but also soft savings. *Fig. 2* shows a general overview of the yearly average net costs for *Ingenia* CM cases with part replacement for one or more on-site visits required.

[1] Data retrieval from the VERTICA database via squirrel SQL Client. 2019 data is incomplete; based on records until June 22th, 2019, due to data availability upon start of the project

Therefore, Philips Healthcare believes there is potential to enhance customer satisfaction and reduce the number of CM visits & unnecessary costs as improvement to CM activities potentially can be made via the historical repair data, such as Spare Part Replacement records, and FMEA provided guidelines for potential part failures of specific machine chains. Until now, the CM processes are too dependent on expert knowledge that can cause activities to be redundant or to be performed too frequently.

In short, although FVF rates are slowly improving over the past couple of years, difference in average cost between single and multiple visits for a CM case with the use of at least one spare part replacement, is increasing. Potentially, field service improvements can be made to increase the metric rate and decrease the total maintenance cost difference. Especially, since considerable effort is required to identify the issue at hand based on implicit knowledge and ad-hoc decision making which spare parts are likely needed for a solution, by the FSE.

For the last couple of years, Philips Healthcare has used a big data resource called Vertica. This database contains, among others, historical data on all the maintenance activities, in theory resulting in easier access to the raw data. Although some log file analysis might have been done previously, historical data analysis of the CM data has not happened yet, combined with the FVF data or including any other potential data sources for improving CM maintenance. Therefore, there is an opportunity to analyze and review the existing historic on-site CM activities where parts have been consumed aiming to improve the FVF-rate on the long term.

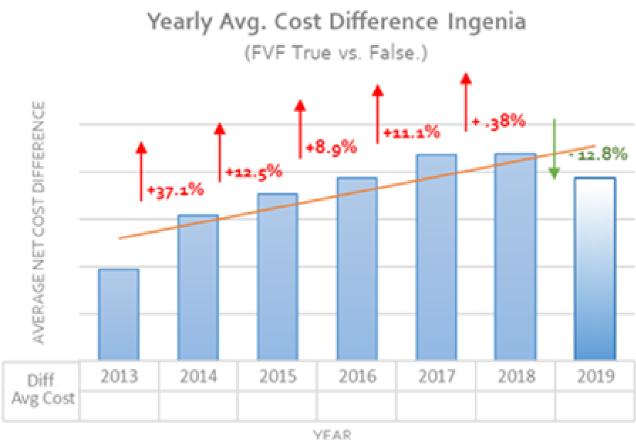


Fig. 2 - Yearly Average Cost Difference FVF per Case, for distinct Ingenia CM with parts cases

1.6 Research Objectives

The goal of this research project is to improve the cost-effectiveness of on-site corrective maintenance operations for certain MR devices by exploring ways to transition the ad-hoc decision process for part replacement during CM activities to a predictive-based corrective maintenance approach (*Table 3*). This objective has been determined based on the developed cause-and-effect tree resulting from interviews and discussions with employees of the Customer Service department (*Fig. 3*). The desired field service metric (First Visit Fix) to improve is depicted in green, with potential aspects to improve in orange.

First Visit Fix KPI information is logged into the Vertica database for each spare part replacement. Although Philips employees know FVF needs to be improved and calculations and analysis of aspects of the metric are occasionally done, this tends to happen manually. A quantitative and systemic analysis of this KPI over all major MR systems is missing and is required to motivate FVF potential improvements and validate KPI information received from other departments.

Table 3 - GAP Analysis

As-Is	To-Be
First Visit Fix KPI not systemically and quantitatively determined	First Visit Fix KPI systemically and quantitatively determined and validated
Motivations for performing CM service actions implicit	Motivations for performing CM service actions explicit
Ad-hoc FSE spare part ordering	Spare part ordering analytically supported for FSE's
Logged CM knowledge base not used by RSE or FSE's	Logged CM knowledge base used for on-site service actions by RSE and FSE's

Moreover, as FVF related teams within the organization are nowadays aiming to analyze historic data for RSE support, no systemic or developed solutions exist per the knowledge or the researcher. Developing a model that aims to motivate CM service actions explicitly and thereby using the logged (raw data) knowledge base can be highly helpful in the organization. Not only can it aid in better RSE diagnosis and advise for a FSE' SWO, and thereby increasing RSE Service Quality, it also makes sure that the logged knowledge base is used. Additionally using a data-driven method the risks and potential other reasons for multiple visits (e.g. FSE Problem Solving Skills, incorrect on-site tools and more repair time available due to upfront help with part issue identification) could be mitigated.

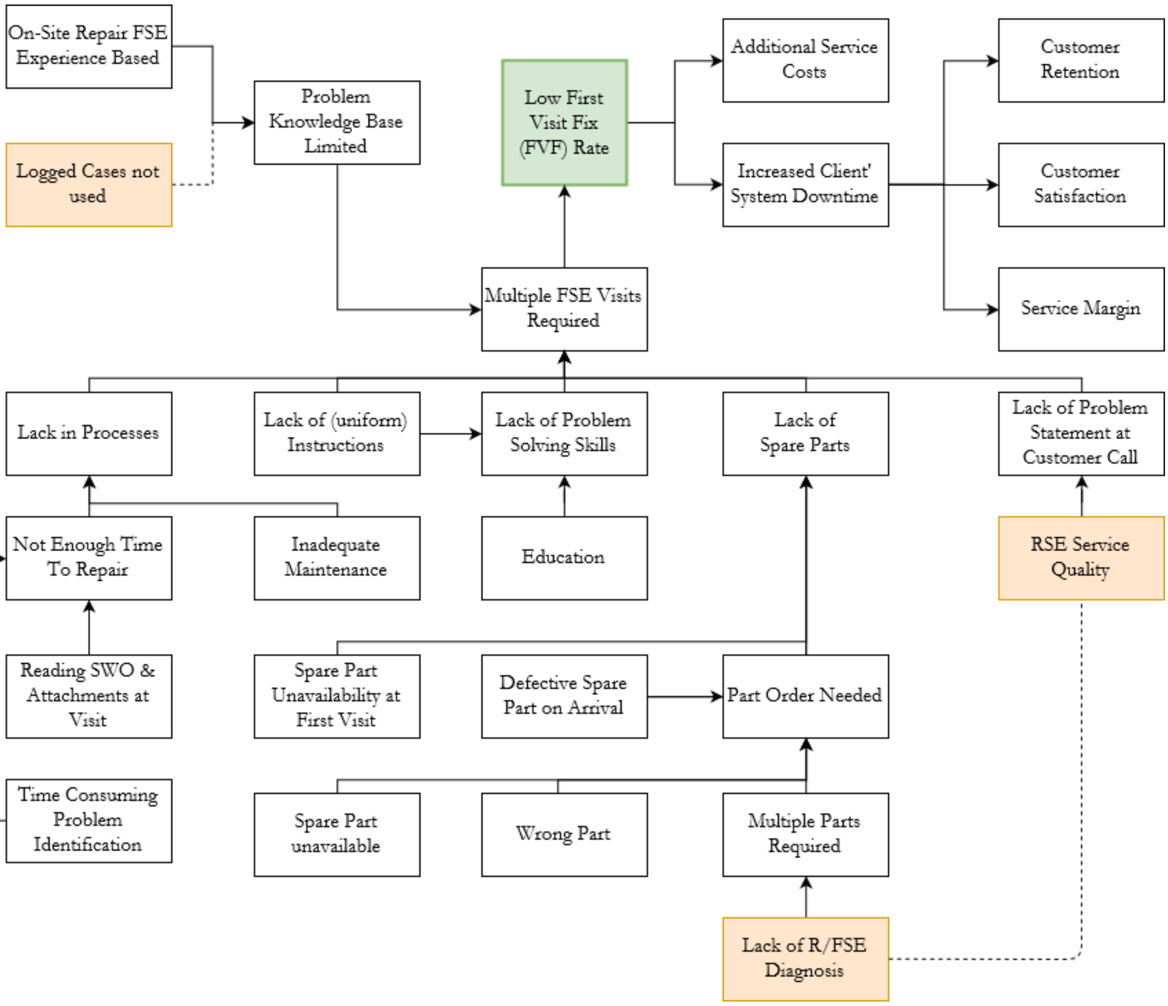


Fig. 3 – Cause- and-effect Tree

1.7 Research Questions

Based on the problem context, the following main research question is formulated, which captures data analysis of failure and spare part specific data,

"How can we accurately predict the required spare parts for system' component failure based on available data?"

This section further derives respective sub-questions (SQ) for the current and desired situation:

RQ1. How is the maintenance process currently designed?

- SQ1.1 What are possibilities for support in the AS-is situation of the Magnetic Resonance troubleshooting process?

RQ2. What are the available sources and types of data that are suitable to predict the required parts for maintenance?

- SQ2.1 What CM activity data is available to analyze maintenance activities?
- SQ2.2 Which data sources and pre-existing documents are available for failure condition and service action identification?

RQ3. How should the relevant data be used for a prediction model development to propose improved decision making for CM activities?

- SQ3.1 Which model type is appropriate for the prediction analysis?
- SQ3.2 How can failures conditions be identified corresponding to the part replacement data?
- SQ3.3 What is the co-dependency, if any, between spare parts used in historic cases?
- SQ3.4 What is the co-dependency, if any, between historic cases' failures/system errors?

RQ4. How can the model outcome support maintenance decisions?

- SQ4.1 What are the validated model' performances, in terms of suitable performance metrics?
- SQ4.2 What is the business potential of the First Visit Fix metric in general?
- SQ4.3 What is the business potential of the deliverable?

1.8 DELIVERABLES

To bridge the gap between the current and desired situation, two deliverables are required and therefore developed. Understand and quantify the current state of CM activities in terms of the FVF performance indicator, and what the potential benefit can be for improvements of this metric. Moreover, the study aims for a systemic approach to develop a model for Philips; as support for RSE and FSE to provide advice or decide, respectively, which spare parts to use for a certain CM activity. The model ideally leverages several sources of data to tailor the output to different device' part failures. Since CM activities for MR devices are rather diverse in terms of system type, system chain, and CM purpose, a primary focus on a set of CM activities is required where at least one part has been used for on-site repairs, specifically for *Ingenia* systems. This project aims to function as a proof of concept for a systemic data-driven approach to come to concreate diagnostic improvements and reduce CM costs at Philips; with the potential of increasing FVF rates. Therefore, the following deliverables are pursued:

1. An overview and systemic determination of First Visit Fix KPI values and quantification for MR-systems.
2. Development of a model, scoped as elaborated in the next section, which determines which spare parts most likely are needed for a system repair, during a customer visit based on historic data.
3. Evaluation of the model' performance and explore the potential business impact if implemented.

The above mentioned model is not is not further specified at this stage, as appropriate prediction model selection, depending on the amount of data and complexity of the problem, takes place in *Section 6.4*.

1.9 SCOPE

This study' goals is to develop a systemic approach in order to improve the First Visit Fix rate for corrective maintenance operations for Philips' medical devices, specifically: Improve First Visit Fix by improved Remote Diagnostics and bringing the right part(s) to fix the problem. On long-term also meaning an improved cost-effectiveness for maintenance, along with potentially improved customer satisfaction, retention and improved service margins.

However, given the specific Philips department and the fact that First Visit Fix is a general term for the healthcare organization the scope of this research is defined in terms of 1) medical equipment and 2) maintenance type.

1.9.1 Medical Equipment

The scope in terms of medical equipment is defined as the magnetic resonance (MRI) devices produces by Philips. Over the years, the healthcare organization has developed several generations of these devices. For this study, the scope is two-fold, and limited to the four install based systems (Achieva, Ingenia, Multiva and Intera) for the first deliverable. For the second deliverable, the scope is limited to all systems part of the '*Ingenia'* product model; decided based on install base, system design and priority. All '*Ingenia'* models are connected to Philips' database and in some cases a system can be repaired by a 3rd party, but this repair and used parts information is also available, providing a relatively complete set of considered data.

1.9.2 Maintenance Type

Aiming to improve First Visit Fix by bringing the correct parts to client' site for the first visit, historic data regarding aforementioned MRI scanners is be considered for 1) corrective maintenance cases where First Visit Fix rates and information is known, along with 2) cases with the use of spare parts.
Based on the urgency, cases are categorized in five categories: P1) Critical Need, P2) System Down, P3) System Restricted, P4) Intermittent Problem, and P5) Scheduled Activity; where the first two priorities can be labeled as hard failures, and the others as soft failures. The latter lead to systems that will not stop working completely, but will continue to operate potentially at lower performance (Taghipour & Banjevic, 2012). Note that in terms of technology scheduled activities are not the same as planned maintenance activities, and therefore still fall under corrective maintenance. These categories are all included in the data set.

1.10 STAKEHOLDER RELEVANCE

This study contributes to both scientific and practical knowledge. The scientific relevance and the practical relevance for both Philips Healthcare, MR SI and the end user is described in this section:

1.10.1 Business relevance

The findings of this research could be used to the benefit of Philips Healthcare, specifically in the MR Customer Service department, and might be used as basis of extension to other MR systems. Current MR system repair maintenance is based on planned maintenance strategy and an ad-hoc corrective maintenance strategy. Changing the ad-hoc process to a data-driven one, or at least use historic data analysis as a complementary source for field service decisions can aid the business in several ways:

Firstly, the findings can prove significant in terms of practical relevance for RSE's, as a complementary method to formulate their advice and corresponding service actions towards an FSE when setting up a SWO, as previously logged cases are not used currently for this. It could also be relevant for FSE's to aid in the decision of required spare parts to order once on-site or before even visiting the customer.
Additionally, findings can be important for Philips MR as a basis to increase the First Visit Fix rates for MR customer calls, resulting in business potential in terms of decreased corrective maintenance and

labor costs for single, and multiple visit repairs; especially decreasing the latter as average cost difference between them significantly differs.

Moreover, a societal relevance for the and user could be achieved on the long term, as improvements in field service and thereby First Visit Fix help customers in terms of shorter machine downtime, decreasing time before clients can continue with their own health practices. Which can result in increased customer satisfaction, beneficial for Philips.

If Philips wants to move towards data-driven corrective maintenance based on historic data, there also must be scientific evidence for implementing this strategy, in the form of an accurate model that shows the applicability or predictive modeling on corrective maintenance policies.

1.10.2 Scientific relevance

Multiple studies have shown that data-driven maintenance policies have a better result in terms of costs, quality and performance (Lawrence, Anuj, & Gerald, 1995; Morant & Larsson-Kråik, Kumar, 2016; Rosmaini & Shahrul, 2012). This justifies exploring the utility of gathered MR system data by Philips for their current corrective maintenance process.

This research extends knowledge on the domain of data-driven corrective maintenance, in terms of combining decision and multi-class classification trees based on historic maintenance data, with failure mode trees, for providing maintenance service actions and/or required parts advice to remote and field engineers. As further elaborated in the motivation of *Section 2.3*, this study contributes to the domain of maintenance and root cause analysis by not relying on customer or service engineer system failures' descriptions and solutions, but rather solely on machine logs and system reported errors prior to a customer complaint. Aiming to predict required types of spare parts or even specific parts when certain (sets) of errors are observable in logs files for a system. Additionally, the analysis is extended by a method to provide advice regarding extra (often used) parts along with the predicted part (root cause), if required.

For further elaboration on the scientific contribution, and an explanation of the theoretical background, see *Section 2*.

2 THEORETICAL BACKGROUND

This chapter presents an in-depth theoretical background about (data-driven) maintenance and root cause analysis, different corresponding methods for analysis; aiming to summarize and elaborate on relevant work while identifying the gap of knowledge in existing literature.

Fig. 4 illustrates the value of the theoretical background, using a conceptual project design model based on (van Aken, Berends, & van der Bij, 2012). In which, the top left contains a set of theoretical perspectives required to study the subjects of analysis depicted on the top right. Both form the basis of data-driven corrective maintenance as the deliverable of the project.

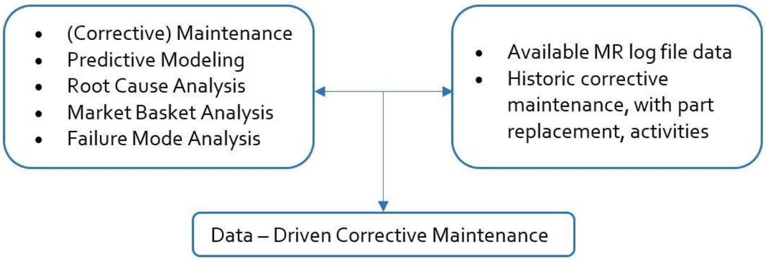


Fig. 4 - Conceptual Project Design

2.1 MAINTENANCE

Different maintenance management policies along with associated information systems are used in order to reduce or repair unexpected failures, eliminate unscheduled downtimes, and minimize maintenance-related costs (Bousdekis, Magoutas, Apostolou, & Mentzas, 2015; Mosaddar & Shojaie, 2013). The availability of historic data potentially allows organizations to make improvements to maintenance processes or to monitor and react proactively to mitigate problems using real-time data (Buchmann, 2014). Maintenance is traditionally categorized as corrective maintenance (CM) or preventive maintenance, including planned maintenance (PM) and condition-based maintenance (CBM) (Cipollini, Oneto, Coraddu, Murphy, & Anguita, 2018; Huiguo, Rui, & Pecht, 2009; Niu & Pecht, 2009).

Maintenance operations within Philips are organized by means of cases, which can be categorized into two categories: preventive- and corrective maintenance. Philips defines the former in the following way:

> *"Preventative Maintenance is used for the service request to deliver the scheduled maintenance activities, which are contractually agreed upon in advance by a warranty or service contract, or purchased by the customer."*

However, this study focusses on CM analysis aiming to improve First Visit Fix rates. Corrective Maintenance is equivalent to Breakdown Maintenance, with other common terms including Reactive Maintenance and Run-to-Failure Maintenance. It is the simplest form of maintenance policy that consists of repairing equipment or assets once it has already failed. A failure of a system can be defined as 'the inability of a system to meet a specified performance standard': called a functional failure (Corrosionpedia, 2018). A CM strategy has the advantage that the useful lifetime of a part is always fully utilized. This means that there is no 'waste' of resources caused by a preventive replacement of the part. However, the disadvantages of a CM strategy are that it 1) relies on a quick reaction time to avoid significant losses during downtime and 2) can potentially cause indirect costs, by potentially secondary damages caused by the part failure. Additionally CM requires the least planning; but the resources that it saves in planning and day-to-day operations often are considerably strained due to unpredictable, frequent, and potential severe breakdowns. According to this policy, system' user only acts after the system or component breaks down (Jardine, Lin, & Banjevic, 2006). Philips Healthcare defines CM as:

> *"Corrective Maintenance is used for the service request to fix a problem of broken equipment; the fix may be delivered remotely with or without parts, onsite with or without parts, or delivered in a bench repair center."*

2.2 ROOT CAUSE ANALYSIS

In order to improve future CM service by understanding what kind of service actions might be required for problems or symptoms that occur, Root Cause Analysis (RCA) can be used to this extent. RCA has an important role in maintenance decision-making and problem solving, as its purpose is to find the underling source of observed symptoms of a problem. In other words, it aims to identify factors causing system and/or equipment failures - sometimes caused by a cascade of events – finding the correct and sustainable solution(s) for failures (Chemweno et al., 2016; Schoenfisch, Stülpnagel, Ortmann, Meilicke, & Stuckenschmidt, 2016). Once identified, maintenance strategies can be implemented more effectively, thereby maximizing equipment uptime.

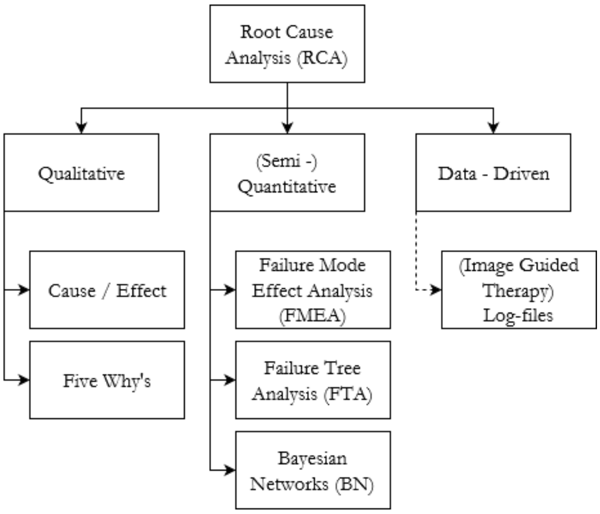


Fig. 5 - Root Cause Analysis Categories

Techniques for root cause analysis for maintenance, or just in general, can be classified into three broad categories given the state of literature; qualitative, semi-quantitative and quantitative (data-driven) methods (*Fig. 5*) (Reid & Smyth-Renshaw, 2012). However, the paradigm currently is mainly characterized by primarily qualitative or semi-quantitative methods (Chemweno et al., 2016). These categories and scientific contribution to them are described below, along with a tabular summary at the end of this section. For specifics regarding this theoretical background, see *Appendix EE* for the literature review.

2.2.1 Qualitative techniques

Qualitative techniques mostly consist of applying the Ishikawa cause-and-effect diagram, and the '5-whys' analysis in the context of maintenance RCA studies. The Ishikawa diagram is discussed in Sharma & Sharma (2010) where the technique is applied for diagnosing root causes of equipment unreliability in a paper pulping machine. In the study, the root causes are enumerated through brain storming session, and subsequently visualized on the diagram. Halawani & Ahmad (2011) similarly analyzed the root causes of repair-related delays in the failure-based maintenance strategy. In which, root causes were derived through expert elicitations from which, the causes are classified into four broad categories; machine-related, material-related, manpower-related and method-related. Nebl & Schroeder (2011) also analyzed the root causes of quality losses in manufacturing systems using the Ishikawa diagram, while more recently, Papic, Kovacevic, Galar, & Thaduri (2016) use the Ishikawa for analyzing the root causes of productivity losses in mining equipment. Many different studies use this technique, also across domains such as the service delivery system domain as an example (Dorsch, Yasin, & Czuchry, 1997). However, all apply the same methodology and are based on SME and brainstorming sessions, surveys or alternative method, to identify failure modes and corresponding causes.

Additionally, the '5-whys' analysis is reported, where decision makers follow a deductive query process from which a series of five or more questions are asked as to the potential causes of a given undesirable event. The query stops when no further causes can be ascribed to the failure event, meaning the root causes are deduced and service actions and solution can be drafted for a given failure. Studies discussing the '5-whys' analysis include Viveros, Zio, Nikulin, Stegmaier, & Bravo (2014) where the approach is used for analyzing the root causes of failure for equipment used for calibrating truck engines. More recently, Benjamin, Marathamuthu, & Murugaiah (2015) apply the '5-whys' for analyzing root causes of speed losses of equipment at a steel barrel manufacturing facility.

Conclusions can be drawn from studies using aforementioned qualitative techniques. Firstly, both techniques are predominantly driven on SME and corresponding (tacit) knowledge. Hence, derived causal associations are almost never empirically linked to (observed) failure events. Moreover, Chemweno et al. (2016) mentioned additional disadvantages: 1) significant time requirements, 2) root cause identification is solely expert-reliant, and 3) high amount of required manpower throughout the process. Similar deficiencies are also noted by Medina-Oliva, Lung, Barberá, Viveros, & Ruin (2012) while stressing that such methods inadequately map the potential causal dependencies.

2.2.2 (Semi-) Quantitative techniques

In order to address deficiencies of qualitative methods, (semi-) quantitative techniques have been introduced, such as Failure Mode and Effect Analysis (FMEA); which is a method to identify - in advance - the failure modes, effects and if possible potential causes of failures and/or general service actions to prevent a failure mode. FMEA is mainly used to prioritize potential failure events based on a calculate risk priority number (RPN). This technique or variations of it, is discussed and used in many studies, e.g. in the domains of manufacturing, automotive, sugar- and coal-fired thermal energy industry.

For example, the quality of automotive leaf spring production has been greatly increased, by an improved FMEA based on SME, explaining the cause of different failure modes and their effect on (the quality) of the product, along with recommended service actions to take if the problem rises (Vinodh & Santhosh, 2012). The same holds for the Zadry, Saputra, Tabri, Meilani, & Rahmayanti (2018) improving the reliability and ergonomic design of their sugarcane machine using this technique by finding root causes to often occurring problems. Additionally the study used cause-and-effect diagrams to further identify failure modes and corresponding actions, leading to reduction of RPN values of multiple failures.

Other papers have expanded the FMEA method by including a criticality analysis of failure modes (FMECA) or including economic considerations (Adhikary, Bose, Bose, & Mitra, 2014; Braglia, 2000; Braglia, Frosolini, & Montanari, 2003). Although all use a similar technique, Braglia et al.(2003) combines FMECA with fuzzification and SME knowledge for a potential cause for failure modes; Braglia (2000) also use multi-attribute failure model analysis (MAFMA) for the selection process of most critical cause of a failure event. De Sanctis, Paciarotti, & Di Giovine (2016) and Sharma, Kumar, & Kumar (2005) take another unique process (in the offshore industry domain) by determining the best maintenance strategy (e.g. preventive, corrective or condition-based) for an item failure after determining failure modes for specific items and their impact on the complete system; with the latter study combining fuzzy linguistic modeling with FMEA.

However, FMEA has been criticized as a basis for RCA and decision support, specifically regarding the computation of RPN; as this is determined as a product of three ordinal risk indices elicited from SME, their knowledge and experience. As a result, the failures prioritized using such methods are not linked to empirical failure events (Chemweno et al., 2016). More importantly, FMEA does not take into account the inter-dependencies or associations between failure events (Chemweno et al., 2016; Liu, Liu, & Liu, 2013); even though this is an important step necessary for RCA and equipment or part failure.

Therefore, other quantitative techniques are suggested and applied in literature, such as Fault Tree Analysis (FTA), and Bayesian Networks (BN); which are graphical illustrations or diagrams depicting cause-and-effect relations or probable associations between failure events and causes (Chemweno, Pintelon, Van Horenbeek, & Muchiri, 2015).

2.2.3 Data-Driven techniques

Despite vast, and growing, data availability of machine maintenance data and log files, literature that leverages this data for decision support and RCA is scarce, specifically in the medical imaging devices and maintenance domain(s). Few data-driven methodologies are reported using maintenance data for RCA decision support (Chemweno et al., 2015; Reid & Smyth-Renshaw, 2012).

The work of Mathur (2002) bears some relevance to the problem at hand, although based in the aviation domain. The paper presents a vision regarding CM diagnostics support to assist in fault-isolation, troubleshooting, and prognostics support in condition-based preventive maintenance by anticipating failures, recommending service actions prior to a catastrophic system failure. Their work actually includes the use of system log files to determine system health and status, and management of diagnostic models, but historical data is not yet (successfully) used for faultfinding and repair troubleshooting.

Chemweno et al. (2016) propose a novel data exploration methodology for RCA at a thermal power plant using (hierarchical, fuzzy, and k-means) clustering for general failure modes. Each cluster with different failure modes is checked for criticality, and if a failure mode potentially can be solved by modifying a component. The methodology' main outcome is finding potential maintenance strategies for identified general root causes of a cluster of failure modes (FBM, CBM, or DOM). Specific service actions or required parts for occurred failure events are not provided.

Another approach is used in the study of Zhu, Liyanage, & Jeeves (2019), developing a data-driven approach in understanding and detecting failures in Emergency Shutdown systems in the Oil and Gas Industry. Along with identified failure modes, the studies outcome is a list of potentially relevant system chains, determining how likely it is if certain failure modes can be detected in the different chains or if specific chains are irrelevant for a corresponding failure mode. Moreover, it was concluded that the understanding of failure mechanisms and the complex dependencies between different parts are helpful and even critical in the diagnosis of root causes.

Other works have used log-data and employed a methodology based on Markov Logic Networks to provide a ranked diagnoses in the form of a list of possible RCF based on probability, or to predict whether a failure is permanent or transient, hence raising an alarm in case of proper severity (Majumder, Sengupta, Jain, & Bhaduri, 2016; Zawawy, Kontogiannis, Mylopoulos, & Mankovskii, 2012). The latter study also aims to identify possible root cause devices in the airline flight check-in infrastructure management, in other terms system chains that might be causing a failure.

Unfortunately, machine log files and/or maintenance data is more often used for other purposes than RCA: predicting part of equipment failure. Historic data can be used to 1) identify and classify faulty components, and determine the probability of fault occurrence using fuzzy logic and artificial neural networks (Wu, Liu, & Ding, 2003), 2) estimate machine breakdown probability during a future time interval using random forest machine learning based on log messages, event logs and operational information (Gutschi, Furian, Suschnigg, Neubacher, & Voessner, 2019) , or 3) determine the high and low risk time intervals of failure for each individual asset for a next year given earlier performance (Rezvanizaniani, Dempsey, & Lee, 2014).

Specifically in the healthcare domain and imaging devices, similar studies have been performed. Sipos, Wang, & Moerchen (2014) used equipment event logs and multiple-instance learning to predict medical scanner failures. While MRI log data and corrective maintenance data served as an input to researchers for predicting MRI component failure 14 days in advance of the actual failure; as a first study within the healthcare and imaging devices domain, aiming to reduce machine downtime and cost savings for OEM's (Patil, Patil, Ravi, & Naik, 2017).

2.3 Motivation Study

As aforementioned, root cause analysis (in the maintenance domain) is mainly focused on qualitative and semi-quantitative approaches. Such approaches rely on subject matter expert(s) and unfortunately tend to introduces bias in the root cause analysis process and risk of incompleteness. Moreover, such analyses usually take place knowing upfront what different failures are for which a root cause is identified. The mentioned quantitative techniques for root cause analysis, e.g. fault trees and Bayesian networks, are sometimes limited to analyzing root causes in simple systems, or are based on the same data source as qualitative methods; other than generated log files.
Performing such an analysis is also challenging due to the complex failure associations existing between inter-connected system components. However, as the collection of maintenance and system data has been enhanced in recent years, this could assist in deriving meaningful failure associations in the data.

In this study, we do not know upfront why certain maintenance cases have been created, only that system repairs have taken place upon customer complaints; contrary to aforementioned (semi-) quantitative or data-driven studies, where a concrete failure mode is known or identified for which a root cause is aimed to be determined. This study identifies failure modes occurred before case calls, and assumes (simultaneous) multiple failure occurrences that can lead to a part replacement. Hereby, the study focusses on machine behavior and logged errors within a specific timeframe - prior to customer calls - across different system chains, combined with maintenance records, to predict required spare parts for when (a set of) errors occurs. Improving on discusses work by not only providing a general service action and direction, but based on this predicting a specific required part. Moreover, in order to take into account the conclusion of Zhu, Liyanage, & Jeeves (2019), 'understanding failure mechanisms and the complex dependencies between different parts' being critical in root cause diagnosis, we aim to investigate – and include into the RCA process – any failure dependencies or potential causality, and providing additional case-based part consumption dependency.
This is a novel approach to data-driven RCA to identify the most likely part cluster and of which corresponding most likely part required, to solve cases within a first customer visit, based on machine, error, and case data rather than actual customer reported issues.
Hence, based on the deficiencies discussed in this theoretical background and the fact that the application of data exploration approaches is really under-reported in the maintenance and imaging health devices literature, the need for a data-oriented approach for root cause analysis is imperative, which motivates to address these challenges as proposed in this study.

3 METHODOLOGY

Structural research methodologies help to analyze the problem and design towards a solution. This section provides the overall methodology that has been used in this study. The nature of this project and the presented research questions form a good foundation for data analysis approaches. To answer these questions, the widely used cross-industry standard process for data analytics and -mining is used: *Cross Industry Standard Process for Data Mining* (CRISP-DM) (Bošnjak, Grljević, & Bošnjak, 2009; Provost & Fawcett, 2013; Shearer, 2000). This process breaks data mining down into six major phases, see *Fig. 6*, and shows the possible iteration points in this framework. For a more detailed overview and a step-by-step user guide the reader is referred to Pete et al. (2004); written by members of the 1999 consortium responsible for the methodology.

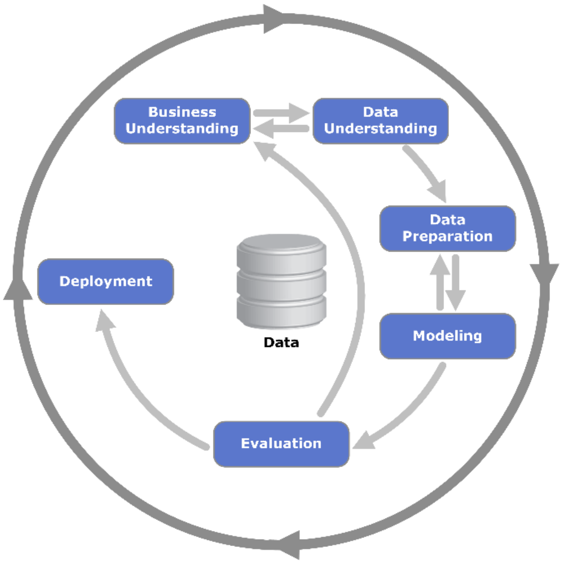


Fig. 6 - CRISP Data Mining Cycle

3.1 PROJECT OUTLINE

An overview of the relation between the below discussed CRISP-DM phases and the research questions introduced in *Section 1.7*, is presented in *Table 4*, along with listed required deliverables per phase. Note that, the sequence of the phases is not strict and moving back and forth, iterating between different phases might sometimes be required for the duration of the project, per definition of the methodology and the nature of the study.

Table 4 - Project Phase, Research Questions Relation

Relation Project Phases to Research Questions		
	Research Question	Deliverables
Business Understanding	1	Business objective, Service performance
Data Understanding	2	Data summary and initial insights
Data Preparation	2, 3	Formatted (cleaned) data
Modeling	3, 4	Model descriptions, test design, parameter settings
Evaluation	4	Main findings, model performance insights
Deployment		Final report, presented findings

3.2 CRISP-DM CYCLE

The remainder of this section introduces each phase of the cycle, also projected on the study as hand. This methodology provides a generic list of tasks for each phase as well, which is used as a guideline to define the approach for the project (Wirth & Hipp, 2000). Authors of the articles encourage selecting and leaving out tasks based on their relevance and including additional ones if providing value to the project. The basic approach is presented in *Table 5*, before elaborating per phase.

3.2.1 Business Understanding

Understanding the project objectives from a business perspective and converting it to data problem, reviewing existing corrective maintenance, spare part replacement, data. First, the business objectives and state regarding on-site corrective maintenance with parts usage are determined. This is achieved by acquiring input from the respective Business Innovation Unit (BIU), responsible for remote service and forwarding work orders to the field for corrective maintenance.

Table 5 - CRISP general tasks

Business Understanding	Data understanding	Data Preparation	Modeling	Evaluation
Determine Business Objectives and Project Goals	Explore available Database	Data Selection	Select and perform modeling technique	Evaluate results with performance tests
Assess Maintenance Operations	Explore Additional Data Sources	Data Aggregation	Consider performance metrics	Consider importance of data cleaning
Assess Field Service Performance	Interpret the data	Feature Selection	Implement potential operational conditions	Evaluate results with regard to business potential improvement
		Data Cleaning	Transform prediction output in suggested course of action for R/FSE	
		Data Output		

Secondly, an understanding of the current maintenance operations within Philips for MR systems is required, specifically what the process is from the moment a customer call takes place until the issue is solved or is escalated to form a thorough AS-is situation. Moreover, understanding corresponding field service performance metrics was needed for obtaining the reasoning of why the topic at hand is important to focus on and why future improvements in such a metric are needed. Identifying potential causes for a less than desirable FVF rate and how the researches can contribute to aspects that can be improved or supported with this study are part of this phase as well.

3.2.2 Data Understanding

The goal of this phase is to get a good understanding of the available data. This process started by exploring a large database that is primarily constructed for research purposes and contains raw and processed data from a variety of databases. Exploring this 'Vertica' database provides an understanding of the availability, type and linking of the data. Interaction with experts from Philips Healthcare is sometimes required for finding and understanding the terminology, since each topic and corresponding knowledge is distributed among different people and departments; such as Research & Development, Customer Service, Imaging Remote Services, and data analysts of the Service Innovation team, and FVF team in Bangalore, India, for Vertica specific SQL query issues.
For getting a basic insight into the data by collecting, familiarizing, and performing quality analysis, and especially preparing the data in the next phase, the general process takes place (*Fig. 7*), where the researcher uses a SQL Environment (in this case SQuirreL Client) to access the data. Initial data merging and cleaning already took place due to limitations of the Vertica and personal system to extract data an mass, and further steps take place in '*R*' and '*MS Excel*'.

3.2.3 Data Preparation

Extracting, and converting raw data into a structured data set in a workable tabular format, containing variables useful for the prediction model as shown in *Fig. 7*. Preparation of the data is an important step in order for it to be usable for modeling, and can be one of the most time consuming aspects of a project. Five tasks similar to those presented in the CRISP-DM methodology are required to complete for this study: 1) Identified potential data during the previous phases are extracted from the database, 2) which are subsequently aggregated. Quantitative data can originate from different sources, and consist of different formats. Additionally, aggregation does not necessarily only need to happen due to cross database data availability, but also records-wise. Philips Healthcare registers spare part replacement and labor information per activity for example, resulting in one record per activity while a complete case can and most certainly consists of multiple records.

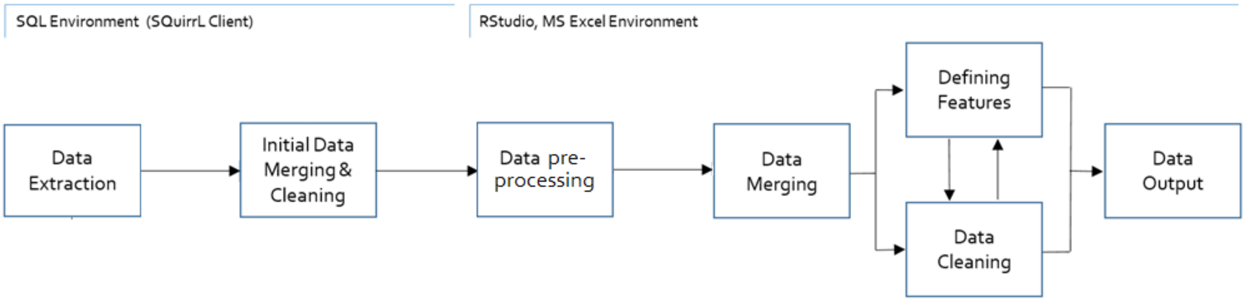


Fig. 7 - Data Understanding, and Preparation Process

Moreover, 3) data is pre-processed and cleaned; potential missing values and unreliable data have been identified, along with outliers; after which data distribution is checked and transformed accordingly. Missing values are excluded or imputed based on the amount or type of missing variable value, discussed in *Section 5*. Reasons for ex- or including data is documented, especially important during the evaluation phase preventing incorrect conclusions or relevant for reproducing them. Although data can contain all MR System repairs, a subset of data had to be extracted since Philips Healthcare FSE's do not always perform these repairs. In some cases, these tasks are outsourced. FVF KPI data is not available for such cases. The same holds for cases where customers call with the request of sending specific parts, exactly knowing what has to be replaced. This is logged as a NEMO-case: No Engineer Material Only. Additionally, to understand why SWO's were created, what the reason was for a customer call, and therefore MR system error behavior; different raw MRI system error log files were useful to identify failure information. 4) Features are derived from the extracted data to characterize *Ingenia* systems by means of input variables, useful for the modelling phase to predict the outcome. Lastly, 5) data required formatting, to be used as input file for modeling. An overview of the final dataset along with corresponding descriptive statistics is reported in the preparation section of the report.

Due to its size and various different identified Vertica data tables, the complete entity-relation diagram, with examples of relevant attributes and primary and foreign keys, is depicted in *Appendix D*.

3.2.4 Modeling

This phase is aimed at creating a model to generate a suggested course of action with respect to certain service actions and spare parts potentially required during a corrective maintenance visit given a device failure, using corresponding modeling techniques. The concept of the model is defined and test design set up. The data is split into a training- and test set, and checked for any potential class imbalance. The training set is used to train the model, however, this shapes the model entirely on the training set. Resulting in less data samples to train on, and taking into account potential overfitting, cross-validation is used. Cross-validation allows the researcher to validate the model that has been trained before running it on the test data without losing training samples to the validation set. Avoiding any bias where models are trained solely on the training set, k-fold cross-validations can be used (Kohavi, 1995).

Fig. 8 - Cross-Validation Schematic Overview

K-fold cross-validation works by dividing the training set into k consecutive subsets of equal size. One of the created subsets acts as the validation set, while the others used as training set. The model trains on the k-1 sets, subsequently evaluated on the validation set. This evaluation results in a prediction error, which is the average of the values, computed in the loop. The value of k should be around 10, or in some cases even higher, to get rid of potential bias (Kohavi, 1995).

3.2.5 Evaluation

Model evaluation supersedes the previous step, in which several performance tests are used to analyze the model output. Subsequently, the performance of the model is translated to potential business' impact, per corresponding research question. Evaluation metrics allow this assessment. A confusion matrix provides a graphical review of the model' behavior and is usable for the calculation of the evaluation metrics, also discussed in this section.

This confusion matrix classifies made predictions into four different categories: True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TP), where the x-axis of the matrix is set to the actual label, and the y-axis as the predicted label. All cases should fall within of the cells of the matrix. Moreover, the Area Under the Curve is a suitable visualization for (multi class) classification problems, where the curve(s) represent(s) the Receiver Operating Characteristics (ROC) curve; a trade-off between the true positive rate and the false positive rate of the confusion matrix. This area represents a different way of calculating the performance of the model. AUC retrieves a number between zero and one, where 0.5 would mean that the model makes an uninformed decision. No threshold value is needed for the classification, being an advantage of the AUC. However, when data is highly imbalanced it is not the best metric, only to be used, due to FN influencing the true positive rate.

Additionally, conclusions regarding the model, considering the feature importance, considering alternative models, along with business impact, are all included in the evaluation as well.

For all metrics considered and used in this study, *Table 22* in *Section 6* provides a brief explanation and corresponding formula.

3.2.6 Deployment

With regard to the developed model, it is necessary to make sure that it also works or can easily work correctly with new data. With regard to finalizing the project, this final report with relevant appendices and presentation are provided as components of this phase. As mentioned, the report contains detailed description of all CRISP-DM phases, the rationale of decisions made, and visualizing the different data mining results. Finally, a reflection step is included to discuss which aspects during the project went right and what could be improved during future research.

4 BUSINESS AND DATA UNDERSTANDING

The analyses for the first research question are presented in the first part of this chapter. It reflects the first phase of the CRISP-DM cycle, described in the previous section. The goal is to analyze the As-Is situation of the Philips Troubleshooting process and provide other relevant information for the business understanding whilst contributing as input for other data-mining phases, moreover get a better insight in the As-Is situation of the First Visit Fix metric. The research questions and sub questions are be tackled in chronological order, starting with the As-Is of the maintenance process to answer **RQ-1.1**. The second part of the chapter focusses on the second phase of the CRISP-DM cycle, data understanding, and aims to answer **RQ2**.

4.1 AS-IS SITUATION - PHILIPS MR MAINTENANCE

Before diving into specifics, first displayed is a consolidated overview of the current troubleshooting process for MR devices in the As-Is situation, is presented in *Fig. 9*. This process is constituted based on five, not all mandatory, different subtasks for troubleshooting, which take place either off-site, remotely, or on-site. These subtasks are: Customer Call, Remote Quick Fixes, Remote In-Depth Analysis and Fixes, On-Site Repair, Additional (remote) Assistance, and Problem Escalation. The general BPMN model is described below, while the whole process in more detail can be found in *Appendix A*.

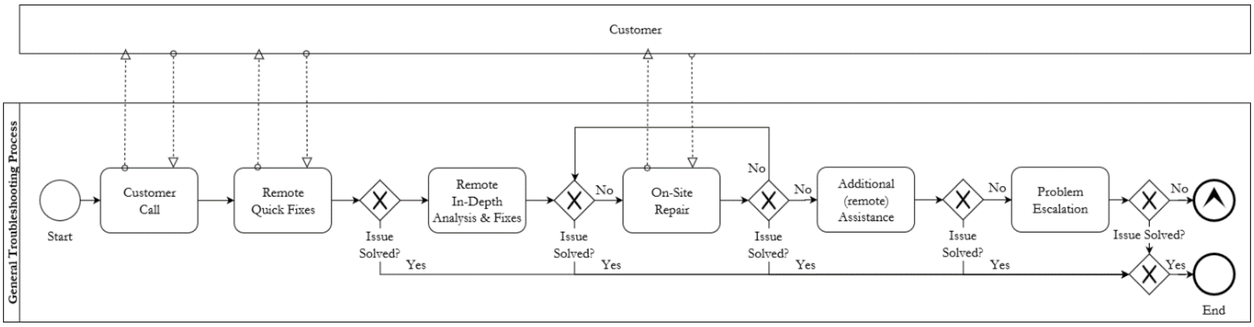


Fig. 9 - Consolidated Overview MR Troubleshoot Process

<u><Customer Call>:</u> CM activities are usually the result of a (impending) system failure. However, there are several ways CM cases could be initiated at Philips (*Fig. 10*), although it is assumed that all CM historic cases are initiated due to the customer. In this case, the customer has a complaint or notices an issue with the MRI scanner. Once Philips receives this call, the call center deals with the initial phase of the MRI Troubleshooting Process, after which Remote Service Engineers (RSE) aim to solve the issue at hand remotely, before potentially involving Field Service Engineers (FSE) or other Philips employees.

Alternatively, Philips can initiate a CM case due to a FSE whom notices another problem during a CM on-site visit. If the issue is related to the actual scheduled CM activity, the FSE could try to fix the newly noticed issue during the same visit. However, if it is unrelated a separate CM case with corresponding SWO will be created.

RSE's continuously monitor the state of a system based on the system logs that are uploaded daily and along with their predictive models. Once a RSE notices a potential issue, or finds that certain threshold values have been exceeded (or will be soon), the RSE will alert the local market and advise them with potential actions to take based on the observation(s). This does not yet lead to a maintenance case. If the market decides that actions should be taken to solve the issue(s) at hand, and if and only if a visit to the corresponding on-site location has already been scheduled for another case, this will count as a CM case and will be added to the already scheduled visit and corresponding SWO.

In the most likely event that no visit has been scheduled this, a new SWO will be created for a FSE, categorized as PM; and therefore not counting as a CM case.
Every time a CM case is initiated, a corresponding Troubleshooting Document (TSD) is also created, which – at this stage – can contain information regarding the customer, usually inquired based on five standardized questions. After the call center wraps up the incoming call, the RSE is provided with the TSD. An RSE contact the customer to find answers regarding twelve standardized questions to obtain problem related information.

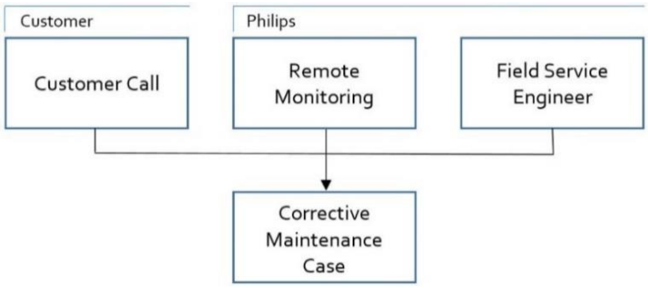


Fig. 10 - CM Case Sources Philips

<Remote Quick Fixes>: Depending on the answers to the questions, an RSE can start with the quick fixes phase. Several fixes such as rebooting the system, power cycle of a suspected chain or the entire system can be performed, and results are noted in the TSD. Additional fixes by the RSE are possible also, if deemed necessary or helpful. If the quick fixes do not bring a solution within a limited time, the RSE should continue with the next phase. In case the issue at hand is solved, the process ends successfully and the updated TSD and case information is written to the OneEMS database.

<Remote In-Depth Analysis & Fixes>: Based on the previous actions, the RSE can use one or more – in any order possible – flow charts based on the potential issue. Specific flow charts are set up for different use cases to 1) find the root cause remotely and solve the problem remotely, 2) ask for cooperation on site to find the root cause and solve it, and 3) make a SWO for an FSE to go on site. Flow charts exist in case: 1) error messages are displayed by the Application Software, 2) system does not start (completely), or 4) the image on the monitor has insufficient image quality or is missing. For 4) potential mechanical malfunction or leakage of cooling liquid or helium such chart does not exist. In any case, employees are referred to relevant SPD's and FAQ's when needed.

If the problem is still not solved, other troubleshooting actions or tests can be doen by the RSE, using (technical) drawings of the suspected system chain if needed. If there are useful actions, they are recorded in the TSD in chronological order, otherwise a SWO is immediately created for an FSE to go on site. For this, the RSE attaches relevant log files and images to the TSD before it is send to a FSE.

<On-Site Repair>: Repair of technical installations, in most cases, requires additional pieces of equipment to replace defective parts, spare or service parts. Three main types of spare parts are distinguished in maintenance operations: retables, repairables, and consumables (Fortuin & Martin, 1999; Gu, 2013):

- Repairables
  - Non-Interchangeable:
    Such parts need to be repaired and when fixed used to solve the issue at hand. These are components, which can technically and economically be repaired, but have to risk of a temporarily unavoidable system shutdown.
  - Rotables:
    A module that can quickly be swapped with a service one, and returned to the factory or workshop for repair, since it is generally less expensive to repair it rather than discard it for a new one. After repair, the part becomes 'as good as new' and is stored as a spare.

- Expendables: These parts cannot be repaired at all or are not economical to do so, and will be scrapped after removal and inspection results as unserviceable.
- Consumables: When such a part becomes defective, it is removed from the system, replaced and discarded by a new item bought from a supplier.

With regard to Philips' operations, parts will always be removed from the system during an on-site repair and always replaced by a newly ordered part. Even though the initial defective part might be repairable, and therefore considered as a rotable, most parts are very expensive to actually repair. Philips has a rule of thumb that only parts with repair costs lower than €1000,- can be considered to be repaired afterwards (hence, refurbished), avoiding potentially exceeding the cost price for a new item. For on-site repairs, either spare have been ordered and have arrived (on location) up front, or parts need to be ordered after exact problem identification. In case the problem at hand has not been solved during the first visit, either due to time constraints regarding problem identification or problem solving, or because additional parts are required (e.g. multi part solution, dealing with a defective spare part on arrival or required part has not been ordered yet), the FSE has to return for another visit.

<Additional (remote) Assistance>: For unsuccessful repairs, assistance of Tier 2 (helpdesk) can be acquired; whom aim to provide further ad-hoc solutions. FSE performs actions based on available immediate advice; otherwise, Tier 2 studies course of action and hands over a TSD copy to Tier 3.

<Problem Escalation>: The problem is escalated and Tier 3 (e.g. specialists, engineers) are contacted. Support of this tear attaches the TSD to OneEMS and assists accordingly.

Based on above troubleshooting understanding, previous problem statement along with the cause-and-effect tree (X), and gap analysis one can find the opportunities for the RCA deliverables of this study within the current business. The data-driven RCA can aid in problem solving at the RSE level, as additional tool for remote in-depth analysis of the customer complaint. After quick-fixes and initial service directions, the part-prediction(s) can be useful in formulating a (general) service action when creating an SWO for FSE's on the field. However, modeling results can also be visualized and transformed to be used for FSE's as well for on-site complementary input.

4.2 Field Service Metrics

The following FVF related Field Service metrics are used at the organization, but only for cases and repairs explicitly solved by Philips (RSE and FSE) employees. Cases where material is sent but no employees are needed do not contribute to the metrics. (External) third parties can also perform on-site repairs, but no FVF information is available for these cases; only parts used or returned information.

1) *First Time Right represents the ability of Philips Healthcare Customer Service organization to complete Corrective Maintenance calls **remotely** or during first customer visit (with 0 or 1 visit maximum).*

2) *First Time Fix represents the ability of the Philips Healthcare Customer Service organization to complete **on-site** repairs during the first customer visit. Hence, only focusing on on-site repairs, while FTR includes remote service as well.*

3) *Number of Visits represents the number of distinct days with Service Work Order time booking with an on-site activity type (excluding Travel).*

Part Replacement Data of both single and multiple visits with regard to the FVF Field Service Metric, has been analyzed; resulting in FVF performance. This is done, based on cleaned data from *Section 5*, and presented in *Appendix K* and *Appendix L* due to confidentiality. Part of the FVF analysis is used in *Section 7* (*Appendix CC*) to discuss potential business impact of created model(s) of *Section 6*.

Next up is the data understanding phase, containing the following steps: collect initial data, describe data (volume, attributes), explore data (basic statistics, sub-populations), and verify data quality (Wirth & Hipp, 2000). As announced previously, corrective maintenance data, specifically where parts replacement has taken place, is collected using the Vertica database. Using this data source directly contributes to **RQ2**, specifically **SQ2.1**. Other considered data sources are also mentioned in this section, referring to **SQ2.2**. Vertica Systems is a column-oriented based database management system used at Philips Healthcare, and it contains most of the 'big data', where data can be extracted using SQL queries in the Java Database Connectivity (JDBC) format using SQL clients like Squirrel, or, with additional libraries. Since Vertica is a shared resource Philips Healthcare wide, queries with low computational requirements are preferred.

4.3 General Data Collection Procedure

The specific Vertica tables that are referenced for this study are documented in *Appendix D*. The queries have been written in a way to only include data columns that are relevant; also, to minimize the strain on querying servers and taking into account the maximum amount of memory allocated to the SQL client on the local system. Hence, only data for MR devices is included, and rows containing null values for important fields are excluded. In some cases, queries had to be extended and join several tables to retrieve required data from these tables, via identification fields such as *SapSWO* or *CaseNumber* or *ProductGroup*, referring to the unique identifier or type of an MR device. All extracted data is stored in dedicated files for later use.

Alternatively, an active connection between the database and a modeling environment could have been set up. This would make sure, that the up-to-date data can be queried in real-time while building the model or performing additional analyses. However, the decision to store the extracted data in dedicated local files was a deliberate decision by the researcher and is preferred because of several reasons. First, an active connection seemed unnecessary, as the relevant data is based on historic cases and will not change; only additional cases could be added over time.

Moreover, ample CM cases were available over a time-period of multiple years along with six months' worth of 2019 data. Furthermore, implementing an active connection could result in querying data more frequently than required, putting more strain on the querying servers of Philips and local devices than necessary. This also guarantees that during the remainder of the CRISP-DM phases the same dataset is used.

Additional non-Vertica database sources are identified as well, such as 1) low-level design documents for MR models, 2) Service Procedure documents, and 3) Failure Mode Effect Analysis Sheets. These sources either describe the specification of MR models, potential failure modes along with potential prevention controls or general service actions required. However, these documents are not always complete in terms of fault finding and follow up instructions. As we are focusing on a fully data-driven method for RCA, these documents have been considered but not further used for the remainder of the study, apart from their potential usefulness in identifying specific failures for which it is known upfront that no part replacements are required.

Based on these documents a few errors – collected from MR log files corresponding to specific SWO's – have been excluded from analysis as we know certainly these do not require any part consumption but rather some form of (part) adjustment.

4.4 System Type Identification

Before we can extract relevant data for different MRI devices, it is important to know how these devices are referenced to, or identified in Vertica. The Spare Part Replacement (SPC) section of the database contains a table called "*SPC_Product*". Focusing on MR systems in this table one can observe the following relevant columns: '*MaterialCode*', '*Description*', '*ProductGroup*', '*ProductGroupDescription*', '*Modality*', '*MainArticleGroupDescription*', '*ArticleGroupDescription*'. Using the following query one can identify the following Product Groups as identifier for the following MR systems (*Table 6*):

```
SELECT MaterialCode, Description, MainArticleGroup, ProductGroup, ProductGroupDescription, Modality,
MainArticleGroupDescription, ArticleGroupDescription
FROM "Development"."SPC_product"
WHERE "Development"."SPC_product"."Modality" ='MR'
AND "Development"."SPC_product"."Description" LIKE '%*enter System Type name*%';
```

Table 6 - Identified System Types based on Description attribute

Product Group	Identified System Type	Product Group	Identified System Type	Product Group	Identified System Type
I_MRI001	Achieva	I_MRI005	Intera	I_MRI009	MRI Workstation
I_MRI002	7.0T Achieva	I_MRI006	Marconi	I_MRI010	Panorama
I_MRI003	Gyroscan	I_MRI007	Ingenia	I_MRI011	Panorama LFO
I_MRI004	Sonalleve	I_MRI008	Magnets	I_MRI012	Multiva

Please note that the '*Description*' attribute, specifically, has to be used identifying the product groups, as '*ProductGroupDescription*' does not always contain the names of the MR systems, and '*MainArticleGroupDescription*' contains the specific type of MR device for the main system identified (e.g. the *Ingenia* system type exits of several different *Ingenia* Systems). Moreover, software versions of the MR system as shipped or maintained, is mentioned in '*ArticleGroupDescription*'.
Additionally, so-called '(digital) dStream' version exists for certain MR devices. This is possible for Achieva and *Intera* systems, which can be upgraded during their lifetime. *Ingenia* dStream systems exist as well, but manufacture as is, and not upgraded to mid-lifecycle. Afterwards, these systems are called *Achieva* dStream, *Intera* Achieva, and *Ingenia* dStream, respectively. However, as the *Achieva* and *Ingenia* upgraded systems fall within the same category and product group, *Intera* Achieva systems are not. These are officially labeled as *Achieva* dStream systems and change their product group to I_MRI001.

Alternatively, one can use system codes (called 6Nc numbers or Product Codes), which are available externally and refer to a specific type within one of the system types. These numbers can be matched within this same Vertica table with the '*MaterialCode*' attribute. However, the researcher deliberately decided not to use this identification method, as the list of 6Nc numbers is continuously expanding and other parties aiming to achieve the same goals need to be aware of the this and the up-to-date 6Nc list.

4.5 Data Description

This section contains a descriptive analysis of the final data set. This analysis is split into a summary of the data, followed by a brief introduction to the insights found in the data. This analysis has been conducted using various tools and software; 'R', 'SQuirreL SQL Client', and 'MS Excel'.
From the extracted raw data, it is clear that records representing different cases require to be combined; as cases can consist of multiple SWO records (*Fig. 11*). The figure serves as an illustration from part of the raw data and features. Duplicate Case Number's therefore should not be mistaken for duplicate data, knowing that the Vertica database consists – as aforementioned – from different data sources and generational databases. *Section 5* provides the description of this process.

CaseNumber	Priority	CaseOpenDate	Part12Nc	Quantity Consumed	Quantity Returned	Market	First Visit Fix	Number Visits
0102244653	3-System Restricted	11/23/2013 2:15	453530258682	1	1	NAM	0	2
0102723968	3-System Restricted	2/13/2014 2:21	924051217100	3	0	NAM	0	2
0102723968	3-System Restricted	2/13/2014 2:21	459800108101	1	0	NAM	0	2
0105368040	5-Scheduled Activity	4/24/2015 2:21	459800108101	1	0	NAM	0	2
0105368040	5-Scheduled Activity	4/24/2015 2:21	452213300932	2	0	NAM	0	2
0105368040	5-Scheduled Activity	4/24/2015 2:21	989603024151	2	0	NAM	0	2
0104426581	3-System Restricted	10/28/2014 2:13	453530258682	1	1	NAM	0	2
0106105981	3-System Restricted	10/7/2015 2:19	453530360285	1	0	NAM	0	2

Fig. 11 - Illustration of SWO data requiring data aggregation

4.5.1 Data Quality

Data quality refers to the condition of a dataset containing a set of values of quantitative or qualitative variables. Although there are several definitions of data quality it is generally considered of high quality if it is "*fit for (its) intended uses in operations, decision making and planning*" (Redman, 2008). Alternatively, data is perceived to be of high quality if it correctly represents the real-world construct to which it refers. As data volume increases, the importance of internal data consistency becomes significant. It this section the main data quality components are be discussed (DAMA, 2013):

Accuracy refers to the degree at which the data actually represents the real world status that it is measuring. As the study involved working with raw data recorded from CM cases and machine file uploads, the accuracy of the data can only partly be determined. Accuracy of part consumption is difficult to verify as it is unknown how ordered parts are logged into Vertica specifically or if any manual input errors are made, but extreme or unrealistic cases with regard to labor time durations, costs, visits, and other SWO related aspects are addressed in *Section 5.4*.

Completeness is defined as the level at which desired data attributes are supplied. Assessing data completeness for this project requires looking at two different aspects: 1) overall data set completeness, and 2) data completeness regarding field service metrics. For the former a missing value analysis was made, presented in *Section 5.3*. *Table 7* shows an overview of available SWO's, regarding the latter; specifically the amount of extracted records. Note that in the raw data multiple records can represent a specific case. Due to Vertica and local system restrictions, while extracting the researcher has already included a filter on the amount of used spare parts (Quantity Used > 0) and amount of returned parts (Quantity Returned >= 0). This results in raw data based on on-site repair CM cases with parts used and excluding remote solved cases.
A clear distinction can be made between the majority of the extraction containing field service (FVF) metric data and a small section that does not contain this information (Nemo data). For some cases, the customer exactly knows which parts are needed to solve the MR issue and is able to perform the repair themselves; for whom these parts are sent and no engineer is required. FVF metrics are only registered for FSE solved cases. Meaning, that this subset of Nemo data (24%) is not relevant for this study.
In terms of error occurrence and log file availability, completeness can be determined after preprocessing e.g. completeness of data regarding daily machine uploads. *0* provides the complete analysis. In short, a sudden factor two increase for average number of distinct errors per case is observed for the years 2016 and 2017. Two potential factors contributing to more observed (distinct) errors have been identified together with SME's: 1) Log File Availability, and 2) Software Release related causes. An increase in logged (distinct) errors can potentially be explained by the overall increased log file availability (of 5-10%) in these years, and (significant) increases of these numbers for certain markets.
Although we can conclude that software release can be a contributing factor to increased average error in 2016-2017, nothing explicit can be said about the average amount of distinct errors per case, per year, for each release.

Timeliness is the last data quality component considered in this section. This represents the age of data, in other words the extent to which data is updated for and thereby usable. Timeliness is an important prerequisite for data science when it comes to data analysis. Extracted data for this study contains records from as early as January 2012 and is updated since on a daily basis. Hence, relevant data until June 18th, 2019 has been queried.

Table 7 - Initial Raw Dataset descriptives before data cleaning[2]

CM Cases	Total	On-site repair with parts ($CM_{on-site}$)	No Engineer Material Only (Nemo)
# Records (case lines)	100%	80.4%	19.6%
# CaseNumbers	100%	80.4%	19.6%
# Number of Visits records	100%	100%	0%
# FVF & FTR records (not distinct)	100%	100%	0%

4.5.2 Initial Insights

All records of each MR system have been plotted per month over the whole period of 2012 – 2019, for an initial insight of the raw data (*Fig. 12*). Although mostly increasing trends can be observed for the MR systems in terms of total records per month, the most striking and important observations is the timeliness of the MR systems individually; instead as a whole. Three out of four systems have available data for on-site CM cases from January 2012, while only one has SWO information available from 2013 onwards. This is important for further analysis, as only overlapping years need to be taken into account to prevent any unnecessary bias. Therefore, during data cleaning the 2012-related data points have been excluded.

The same holds for, in general, unrealistic amount of visits per CM case. Number of Visits have been plotted in strip charts which show the number of visits per records for each system type or summarized per year (*Fig. 13*). Mean values have been indicated in red. Occasional records with high customer visit values might be considered outliers; as discussed in the next chapter.

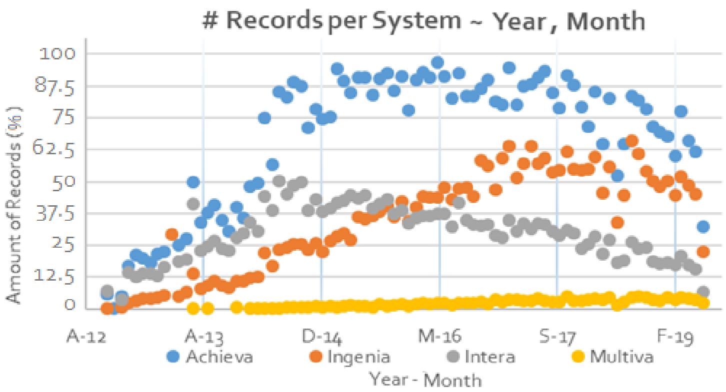


Fig. 12 - Scatterplot Records per System per Month

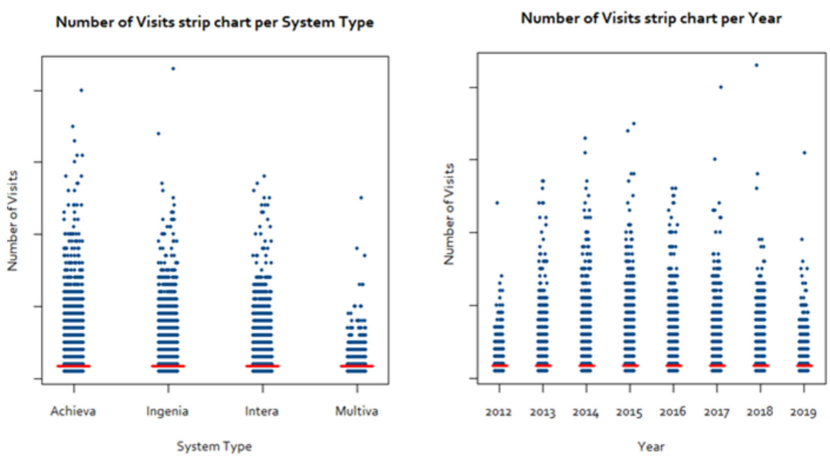


Fig. 13 - Jitter Strip Charts Number of Visits per System Type & Year

[2] Exact values regarding the initial, raw, data extraction based on the project's scope are not available due to confidentiality. Values in the $CM_{on-site}$ column are identical per criteria. The same holds for the number of nemo related records and case numbers. No number of visit records, FVF or FTR records are available for nemo cases.

5 DATA PREPARATION

This chapter explains the steps taken during the data cleaning steps performed on the acquired data set. The output of this phase are the train and test sets that are used for the modeling phase or FVF metric analysis. This first main section of the chapter is divided in three main steps, based on the Knowledge Discovery in Databases (KDD) - process framework. These steps include feature selection, data cleaning along with outlier identification and system error identification, and data integration. Of which, the error identification – towards the end of this first section – contributes in providing an answer to **SQ2.2**.

After the KDD- process, the second section of this chapter focusses on the occurred errors corresponding to cases along with potential independence and clustering of said logged errors. However, before any results and corresponding conclusions can be drawn regarding SWO's and between multiple and single visit cases in this or later chapters, a crucial preceding step is required; statistically significant difference between these two subsets.

5.1 FEATURE SELECTION

Feature selection plays a vital role in machine learning or specifically classification problems, and essentially function as a set of input variable that represent a property of the entity observed. This section aims to define features with potentially good predictive power that may or may not be used for predictive purposes. Selecting the relevant data for features is difficult as it is a challenge to know upfront which data could lead to interesting features. In this project, features have primarily been selected based on available data discovered during the data-understanding phase. Along with SME's potentially valuable features are defined, which were extracted and merged appropriately as elaborated on in *Section 4* and *Section 5.2* respectively. *Table 8* provides an overview of the main feature categories along with a couple of example features. Some of these features are static as they are similar for multiple cases such as MR system type or location, while others are more important such as consumed parts. Additionally, from the data tables used (*Appendix D*), some columns could immediately be disregarded as potential features, since they either contained duplicate data (in a different format), non-explanatory variables, or simply empty columns.

Table 8 - Feature Types

Feature Category	Features	Data Type	Description
System Characteristics	○ Product Group ○ Priority ○ SRN ○ Market	Categorical Ordinal Numeric Categorical	Specific MR System model Case priority referring to a hard or soft problem Numerical system identifier, unique per machine Location / region from where the customer originates and location of the system.
Case Characteristics	○ CaseNumber ○ CaseOpenDate ○ CaseCloseDate ○ Parts Consumed (Part12Nc) ○ MaterialDescription ○ Quantity Consumed ○ Quantity Returned ○ LaborDuration ○ LaborActivity ○ Number of Visits ○ Entitlement Type	Numeric Timestamp Timestamp String String Numeric Numeric Timestamp Categorical Numeric Categorical	Numerical case identifier Timestamp at which the customer reported the problem Timestamp at which the case was reported to be resolved 12 numerical Identifier of the part involved in the case Consumed Part12Nc's descriptions Number of parts consumed during the case Number of parts returned during the case Total labor duration calculated during the case Specific labor activities taken place during the case Number of customer visits required for the case Financial classification of a case
Error Characteristics	○ MR Chain ○ Chain Unit ○ Error / FaultCode ○ LogDateTime	Ordinal String String Timestamp	Specific chain in which an error occurred Specific subunit currently installed in MR System Specific error or error description occurred Logged timestamp of registered error

5.2 Transformation to case-based data

In order to transform the raw dataset into a pre-processed set where distinct cases are presented per row with corresponding information, several steps need to be taken depending on the variable information that is being transformed to a case-based format.

5.2.1 General Process

For most variables the transformation to a case-based format can be done using standard MS Excel functions; in order to transform multiple records per case to a one record per case dataset, to be applied for system or call related information, monetary or used and returned spare part variables. Due to the vast amount of raw data, this process has been performed for each MR system individually, after which the separate case-based (pre-processed) data have been merged. The result of distinct cases of the pre-processing phase is shown in *Table 9*. Percentages are presented instead of absolute values due to confidentiality.

Table 9 - Pre-processed Dataset descriptives before data cleaning

	Total ($CM_{on-site}$)	Achieva	Ingenia	Intera	Multiva
# Records (case lines)	100%	52%	24%	23%	1%
# CaseNumbers	100%	52%	24%	23%	1%
# Distinct Cases	61%	51%	25%	23%	1%

5.2.2 Parts 12NC & Material Description

Transforming spare part variables to a case-based data set takes a few more steps than the data columns discussed so far. Two spare part variables are relevant: '*Part12Nc'* and '*MaterialDescription'*; both need to be considered as String variables, where the former acts as an identification number and the latter describes what the part specifically is. Even though '*Part12Nc'* technically is a discrete numerical variable, consisting of length 12, it is considered as String since in rare occasions it might have a different format.

For each distinct case, a VBA based vertical lookup function is used to search and sum all '*Part12Nc'* numbers and descriptions among the extracted pre-processed row-based dataset, see *Appendix E*

5.2.3 Data Construction

Some data construction by creating additional variables is required for further analysis, based on extracted cost data. This is required for further FVF analysis, and more importantly having a suitable continuous variable for subsequent outlier identification and normality checks. The latter is not possible with aforementioned discrete variables (*Table 8).* Total on-site corrective maintenance costs are divided from component (cost) price and labor cost at Philips. Determining CM cost for extracted distinct SWO cases is discussed below.

Total Corrective Maintenance Costs – Net Part Cost

Cost prices of components are stored and available for retrieval in corresponding data tables, and can range from a few euros to tens of thousands euros, depending if one is dealing with a replacement fuse or whole coils or magnets of a MR system.

At first I intended to use average part cost prices, by determining the average prices for all distinct parts over the period of 2012-2019, which can be applied for specific system types. Additionally, it was aimed to obtain a list with distinct cases, all corresponding parts and exact cost price instead of the average part cost based over the years. Using the attribute '*postingdate'* we could potentially identify the cost price for a part, for a specific case with the '*postingdate'* and case date closest to each other.

However, we made the explicit decision afterwards to use the sum of the 'Annual Operating Plan (Aop) Currency' ('*AopCurrency')* instead, as this takes into account the net cost for a spare part, e.g.

including costs for returning the replaced component, or returning unused components. This decision has been made, as all service costs regarding parts are included in further calculations, instead of solely average cost price. This information is accessible for SWO's in the local market currency, as well as conversions to Euros. To determine the total (net) part cost for each case, these values must be summed over all records belonging to a distinct case. This additional dataset and the larger primary set of distinct CM cases are merged using '*CaseNumber'* as identifier.

Total Corrective Maintenance Costs – Labor Cost

Additionally, labor information for each repair is recorded along with information regarding technicians and performed activities, which can be used to determine the overall or specific labor cost per work order. Field Service Engineers (FSE) whom perform the on-site repairs and Remote Service Engineers (RSE) approximately cost €100,- per hour; which is used as a rule of thumb at Philips. Labor costs are not directly available in Vertica and must be derived from other data in the "*SPC_labor*" table, such as duration of a repair per SWO, number of engineers and activity types.

Labor cost can be broken down into travel, corrective maintenance, remote maintenance, installation, and application costs based on 27 available activities types. *Table 22* provided an overview with set up labor cost categories and corresponding activity codes. This distribution of codes and the specific categories have been verified with the R&D department. Further elaboration on the four length-strings is available in *Appendix C*.

Table 10 - Labor Categories and Activity Codes (derived from SPC_labor)

Labor Cost Category	Labor Activity Codes
Total Travel Cost	TRAV, TRVL
Total Corrective Maintenance Cost	CMAI, DIAG, FILL, MONI, RPCL, RTST, SWSU
Total Remote Cost	APAS, RMSE, TESU
Total Installation Cost	BCKO, DEIN, IN01, IN02, IN03, IN04, IN05, LOCA, PRCO, REIN, SIRE, UPGR, UTRA
Total Application Cost	APSE, BTRA

The query outputs the cost categories per record, after which a total labor cost variable can be calculation based on a summation of the cost categories. Note that for the purpose of this study, remote cost are not relevant for total labor cost and need be excluded, as the project only focuses on on-site CM repairs. To determine the cost for each distinct case, all total cost values for all records belonging to a distinct cased are summed. See *Appendix E* for a snippet of the script calculating the labor cost.

5.3 DATA CLEANING

It is not always clear what data should be in- or excluded of the dataset due to decisions to be made, and therefore it can be – at times - subjective. This section describes these decisions, based on previous and new observations, categorized in fixed decisions and ones that are debatable or modifiable depending on the focus of future studies using the same data sources.

Excluding data happens by filtering the processed dataset and removing distinct cases that meet certain conditions. This filtering can be done at different moment in the process, while meeting the only requirement that it happens before the modeling phase. Therefore, some of the filtering and checks have been performed in the SQL queries while extracting data, and others in 'R'.

5.3.1 Fixed Data Cleaning Decisions

Given the scope of the project, the following filters have been applied to every Vertica query:

- Modality: Magnetic Resonance (MR)
- Number of visits: > 1, in order to exclude remote cases.
- Quantity consumed: > 1, due to the focus of CM with used parts.
- Nemo: False, excluding cases with no field service metric information, leaving only cases where on-site repairs by FSE's were required.

Additionally,

- N/A values: Exclude cases where data is missing and is not obtainable otherwise, such as CaseNumber, CaseOpenDate, CaseClosedDate, QuantityConsumed, Spare parts used (Part12Nc).
- (Net) part cost: Exclude cases where (net) part cost information is missing corresponding to the service work orders, as this is an important variable; which also cannot be imputed due to highly varied part prices.

5.3.2 Modifiable Data Cleaning Decisions

Distinct CM cases subject to the following have also been removed from the dataset. However, this are debatable decisions and may vary given the scope and aim of the study:

- 2012 data: As observed previously, only 2013-2019 data is used for all analyses as not all MR systems have available data for the year 2012. Hence, these records are excluded for prevent unnecessary skewness.
- Negative (total) costs: Exclude cases with calculated negative (part and labor) cost.
- SRN range: Some SRN values are known to be test systems, therefore SRN < 1000 and SRN > 97 000 can be eliminated from the dataset.
- Labor information Marginalize CaseNumbers without any or incomplete labor information to determine the total labor cost are removed.

Regarding the last decision, there are several ways to deal with missing or incomplete data, especially if it is cost related (Das, Datta, & Chaudhuri, 2018; Garcia-Lacencina, Sanch-Gomez, & Figueiras-Vidal, 2010; Hair, Black, Babin, & Anderson, 2014). Both *imputation* and *marginalization* techniques could be applied. Marginalization excludes data points with unobserved features from the dataset. Alternatively, imputation attempts to fill in the missing features by making reasonable estimates based on the values observed for the corresponding features over the rest of the dataset. There are arguments to be made for each method, but in this research, marginalization is used. Hence, cases corresponding to missing labor information values are ignored and excluded.

Marginalization can be a good method if data is characterized by *missing at random* (MAR), meaning that missing data is random and depends neither on any observed and unobserved data. Consequently, in the case of MAR, removing data from the dataset will not affect the shape of the dataset and if the dataset remains large enough, proper models can be constructed. Even if the dataset is not characterized by MAR if the rate of missing values is relatively low, approximately 1-5%, incomplete data instances can be removed safely. Lastly, imputation was not chosen as a method as imputed values are treated just as reliable as the truly observed data, but these values are only as good as the assumptions made to impute the missing values. In case of imputation, these calculated labor costs will not be reliable as 'required number of FSE's' and 'Repair Duration' values are highly varied, and not yet considering that total labor cost calculation will be even more difficult if corresponding activity types are also missing.

After all data cleaning checks have been performed, the following set (*Table 11*) remains. *Table 12* shows the number of distinct cases (SWO's) that have been excluded during the data cleaning process.

Table 11 - Case Based Dataset descriptives after Data Cleaning

	Total	Multiple Visit	Single Visit
# Distinct Cases	180 600	76 338	104 262

Table 12 - Missing and Filtered Values

Missing Data Attribute / Filtered Negative Values	Total	Achieva	Ingenia	Intera	Multiva
(not available) Casenumber	0	0	0	0	0
(not available) CaseCloseDate	0	0	0	0	0
(not available) QuantityConsumed	0	0	0	0	0
(not available) Part12Nc	0	0	0	0	0
(not available) Part Cost Information	4	3	1	0	0
(not available) Labor Cost Information	2	0	2	0	0
Negative sum part (Aop) cost	9	4	2	1	2
Negative total cost	25	10	8	7	0
2012 data	2 099	1 099	171	829	0
Total	2 139	1 116	184	837	2

5.4 OUTLIER IDENTIFICATION

Before the data preparation is completed, one should check the data set on potential outliers. Outliers do not follow the pattern of the bulk of the data. Therefore, 1) outliers might indicate an error in the data collection, or might be unrepresentative of the population (substantive concern), or 2) outliers have a disproportionate influence on statistical analysis (practical concern) (Hair et al., 2014). This section describes this process and starts with determining the distribution of the data set based on the relevant continuous variable. Parametric distribution tests usually include the use of a histogram, boxplots and Q-Q plot; descriptive graphical and descriptive numerical methods are used (Baghban et al., 2013; Ghasemi & Zahediasl, 2012).

5.4.1 Normality Check

First, a normality check is performed; although this is always important, it especially is required when a monetary variable is used for outlier detection. The aforementioned total (net part + labor) cost calculations per distinct case number are used for the process, as it is the only usable case-related continuous variable. Number of Visits would seem logical at first as detection and normality check variable but the underlying assumption for a normality test is that the data is continuous. Number of visits is a discrete variable. When viewing discrete data, one lacks information between any two integer values and this loss of information makes assessing normality difficult. Although a discrete variable-based histogram could seem plausible, it is disastrous for a normality check (Ghasemi & Zahediasl, 2012; Minitab, 2011).

Fig. 14 shows the histogram based on the available data for total cost based on all systems. A heavy right-tail (right skewed) distribution is very common in business data, usually observed when histograms are made of revenues, prices, wages, or company sizes, for example (Hair et al., 2014)b. This histogram is an effective graphical technique for showing both the *skewness* and *kurtosis* of the data set. These values have been determined as *90.02* and *15849.95* respectively. The probability distribution of the variable about its mean is highly asymmetrical and the data is highly skewed, as the skewness is must greater than 1 (considered as the upper limit of this numeric indicator) (Bulmer, 1979). The heavy tail and non-normality is also confirmed with the high kurtosis value.

Hence, the data set needs to be transformed before any outliers are detected. The log transformed histogram of the total cost variable and the corresponding Q,Q-plot, also based on transformed data are depicted in *Fig. 15*. Log transformation proves to result in a normal distribution based on this graphical method and confirmed with *skewness = -0.21* and *kurtosis = 2.94*.

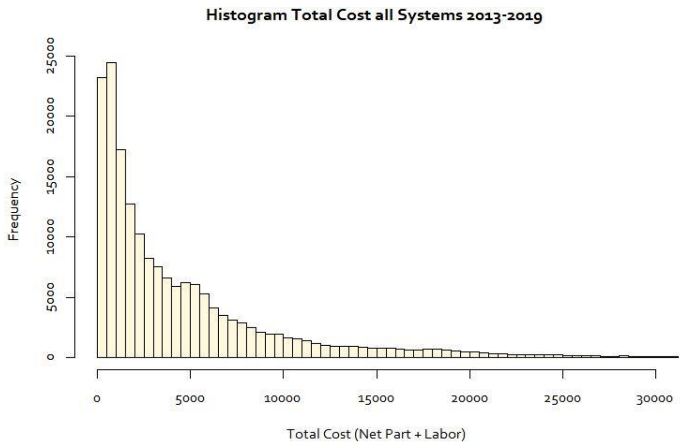


Fig. 14 - Histogram of All System's Total CM Cost

According to Bulmer (1979) data are considered approximately symmetrical if the skewness is between -0.5 and +0.5. Some are more lenient and consider an absolute skew values below 2 as reference for normality, given sample sizes are greater than 300 (Kim, 2013). Normal distributions have a kurtosis of 3. A distribution with kurtosis < 3 are platykurtic: shorter and thinner tails compared to a normal distribution, however our post transformation kurtosis is extremely close to 3 and can be considered mesokurtic, hence a normal distribution (Sheskin, 2011; Westfall, 2014). This is also verified by the Q,Q-plot which plots the observed quantiles against the theoretical quantiles of the normal distribution: sample quantiles are very close or even equal to the theoretical quantiles (Hair et al., 2014).

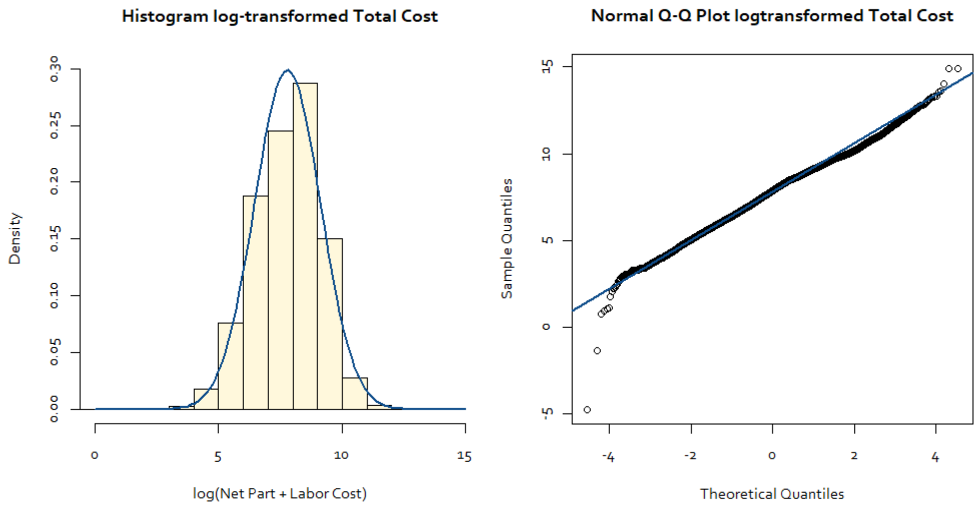


Fig. 15 - Histogram & Q,Q-Plot of Log Transformed Total Cost Values

5.4.2 Boxplot – Outlier Identification

Boxplots for single and multiple visit CM cases are displayed in *Fig. 16*. Note that the y-axis is log-transformed and no negative data points are shown. Plots for either the single or multiple visits look similar, but do have medians and more clearly, different confidence intervals. Comparing pairs of boxplots per year, show a clear difference between total cost for single and multiple visit cases.

Plotted scores greater than 1.5 times the interquartile range are out of the boxplot and could be considered as outliers, and those greater than 3 times the interquartile range are potentially extreme outliers. Symmetric plots with the median line at approximately the center of the box and with symmetric whiskers that are longer than the subsections of the center box also suggest that the data may have come from a normal distribution (Ghasemi & Zahediasl, 2012). Although some plots do seem to have an extreme outliers as a few data points are depicted far from the whiskers, most of the potential outliers reside just out of the whiskers.

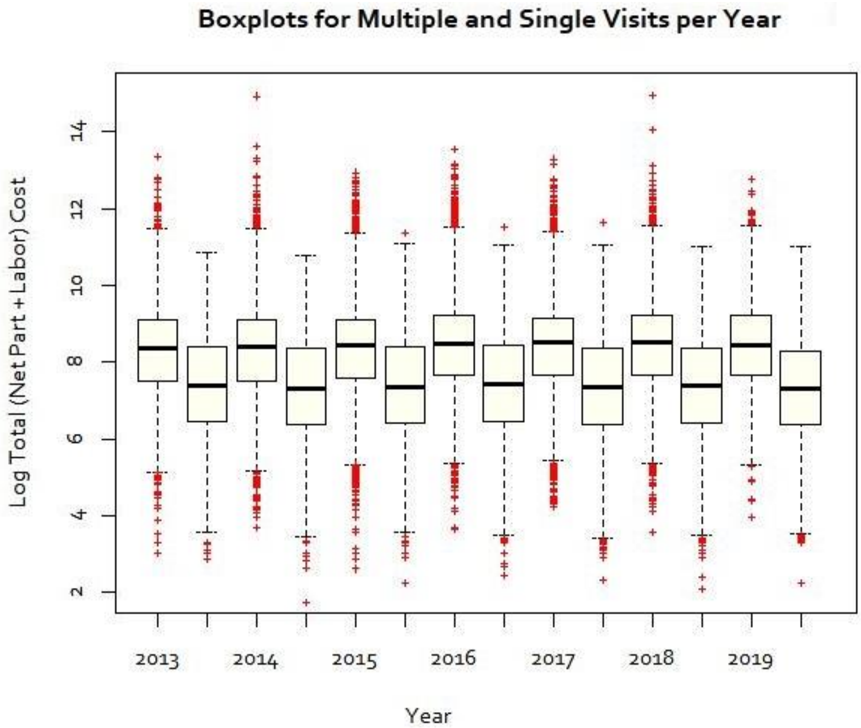


Fig. 16 - Boxplots for Multiple and Single Visit Total Cost per year, respectively

5.4.3 Mahalanobis Distance – Outlier Identification

As outliers cannot clearly be detected with boxplot, a multivariate statistical measure of effect and distance, the Mahalanobis Distance (MD), is used to determine potential outliers; a method introduced by Mahalanobis (1936). MD is defined as the distance between a (multidimensional) point and a distribution. It is the multivariate form of the distance measured in units of standard deviation; hence, it measures how many SD's a data point is from the mean, where MD values greater than 3 can be considered as outliers (Hair et al., 2014). Given a normal distribution, as is the case with our log transformed data, with covariance matrix S and the mean μ, squared MD of a data point is given by: $d^2 (x) = (x - \mu)' S^{-1} (x - \mu)$ (Hair et al., 2014; Sapp, Obiakor, Gregas, & Scholze, 2007); given observation, x, mean μ and S^{-1} as inverse covariance matrix.

A preliminary check regarding correlation between log total cost and log number of visits variables and potential multivariate distributions has been performed to make sure we are not dealing with univariate distributions while using the MD method. A plot of MD values suggest correlation. After all MD values were determined for the data set, ranging from 0.35 to 30 and two extreme values of 72 and 103, 18.08% of values were considered as potential outliers conform the MD > 3 rule of thumb. Although all these distinct cases could be discarded, the researcher decided to further look into what exactly would be excluded from the dataset.

5.4.4 Mahalanobis Distance > 3 subset analysis

Further analysis of the 18.08% potential outliers yields the data presented in *Table 13*. The subset of MD > 3 values consists of (and to be excluded). Based on these results, the decision was made to focus on cases with one, two, or three required visits, as almost all cases with more than three visits would already be excluded due to the MD criteria. Meaning, that excluding all cases with four or more visits corresponds to excluding **6.39%** of the data set.

Table 13 - Analysis of MD > 3 cases

%	Description
99.93%	Of all cases with four or more visits required to be excluded.
20.25%	Of all cases with three visits required to be excluded.
5.33%	Of all cases with two visits required to be excluded.
14.77%	Of all cases with one visit required to be excluded.
12.49%	Of all cases with one, two, or three visits required to be excluded.

However, the MD calculation also suggests to exclude some of the cases with fewer number of visits. To determine if these should also be taken into account, an overview was made of total cost for these cases (*Fig. 17*). Reasoning that if this smaller subset mostly consists of high total cost (expensive parts and/or high labor costs) to exclude it from the data set; as we know that there is a correlation between log total cost and log number of visit. As the majority of this subset consists of relatively low total cost (and cheap parts) the explicit decision with Philips Healthcare is made that this subset (11.69% of the total dataset) is not considered as outliers. No more outliers are detected at this point.

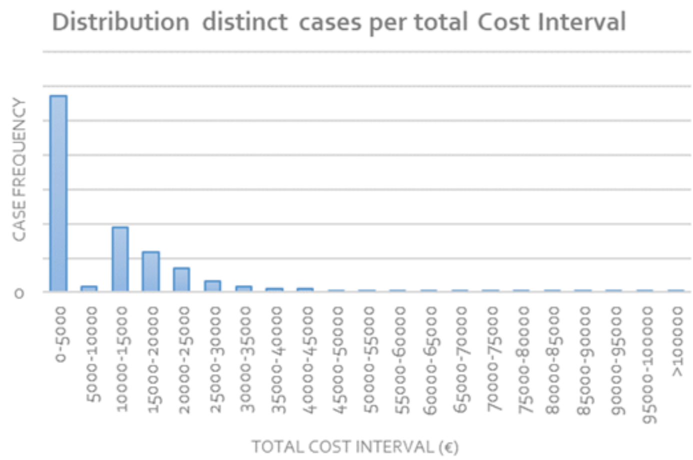


Fig. 17 - Distribution Distinct Cases (MD > 3) per Total Cost Interval

5.5 Error Identification

Data of the extracted CM dataset is two-fold; on the one hand is should consist of case-related data and corresponding variables, while on the other hand it also needs to identify why a certain case was created. In other words, what are the errors occurred, resulting in a customer call for repair? Just as the SWO data, to identify the error(s) raw error data is extracted via Vertica as well from tables and views corresponding to those presented in the Entity Relation Diagram (*Appendix D*).

Note that, as the Root Cause/model-based Analysis of this study is clearly scoped on *Ingenia* MR Systems, Error Identification only takes place for these system types. This in comparison with the data collection, creation, preparation and cleaning, along with outlier identification in the previous sections, which were performed for multiple MR Systems as these *are* included in the FVF Field Metric Analysis.

Raw error data for different chains needs to be preprocessed before error occurrence is analyzed for specific CM cases. The general process is presented in *Fig. 18*. Occurred errors generally consist of a fault code or string description, but these are not unique to a chain or part type, and do not necessarily have the same error description. Hence, it is important to know what specific part type or sub-unit is installed in a system, at the time of the error and for which chain an error is given. All potential part types and distinct errors for the different chains are available in *Appendix G*. Our prepared subset of *Ingenia* systems may only consist of a few distinct ones.

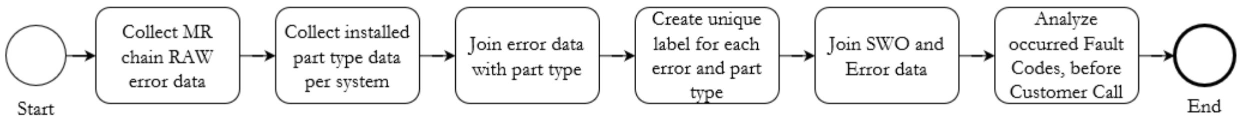


Fig. 18 - General Error Collection Process

For each of the **four** analyzed MR chains the SRN corresponding to occurred error is matched with the last modified/installed part type before the error' timestamp, corresponding to the same chain; identifiable with an SRN.
A new variable is created in an attempt to mitigate the issue of identical error names but being substantially different, and simultaneously anonymizing the data; consisting of chain number, C_{xx}, part type, T_{yy}, and error code, E_{zz}: C_{xx} T_{yy} E_{zz}; xx, yy, zz of any possible (distinct) chain, part type and error observed in the error data (*Table 14*).

Table 14 - Part Types and Errors per Chain

	Chain Label (C)	# Distinct Part Types (T)	# Distinct Errors (E)
Chain 1	1	14	70
Chain 2	2	18	117
Chain 3	3	7	63
Chain 4	4	1	45

5.5.1 Equipment Number – SNR

Now that the error data of different MR chains is processed, this information has to be joined with the SWO data. The processed error dataset at this point is structured in such a way that for each occurred error: 1) an error code is available, 2) along with which type of spare part (of the specific MR chain) was installed in the system at the time of error occurrence, 3) a unique serial number (SRN) of the system, and 4) – if applicable – the subunit of the chain.
While the SRN functions acts as key identifier in this case, SWO data has a case number with an '*AssetID*' as corresponding variable. However, since SWO and Error data have a different data source from which a copy of the data is sent to the Vertica Database, the identifiers for the MR systems do not match. The '*ISDA_medicalsystem_vw*' table is useful for this problem, linking '*AssetID*' to an identical '*EquipmentNumber*' and corresponding 'SRN_INT'.

SWO's with corresponding SRN's are matched with the unique re-labeled error codes. For each SWO we have analyzed the historic error data preceding the SWO *CaseOpenDate*. This is because of two important assumptions:

- The *CaseOpenDate* for a SWO is considered to be the timestamp when a customer call has taken place. Therefore, all occurred distinct errors within a fixed time period before the *CaseOpenDate* – be it a single distinct error or a set of distinct errors – are seen as the reason for an on-site customer repair.
- The pre-defined time period in which relevant errors are taken into account is two weeks.

Initially, the researcher aimed to make a distinction between an on-site CM request (SWO)-timestamp and a second timestamp of when customer call have taken place, respectively: *CaseOpenDate* and *CallOpenDate*. Although both variables seemingly can be found in Vertica, it seems that these timestamps are identical and – just like other instances – are the same variable and data but with a different variable name. Hence, due to the unavailability of this data, this assumption has been made. Therefore, instead of analyzed only error occurrences during t_2, and potentially disregarding t_1, t_3 -duration has been the focus (*Fig. 19*).
For this purpose, three different Visual Basic for Applications (VBA) codes in MS Excel have been developed. The first one acts as a multiple string values lookup given multiple criteria. Each SRN corresponding to a SWO is searched for in the processed error data, and outputs these string values

in an array/concatenation; given these error values have occurred between *TimeCondition1* and *TimeCondition2* (respectively the date exactly two weeks prior to the *CaseOpenDate*, and the *CaseOpenDate* itself). The second VBA function is required to remove duplicate fault codes; or in other words, selecting the distinct values from the previous function' output, within a cell.

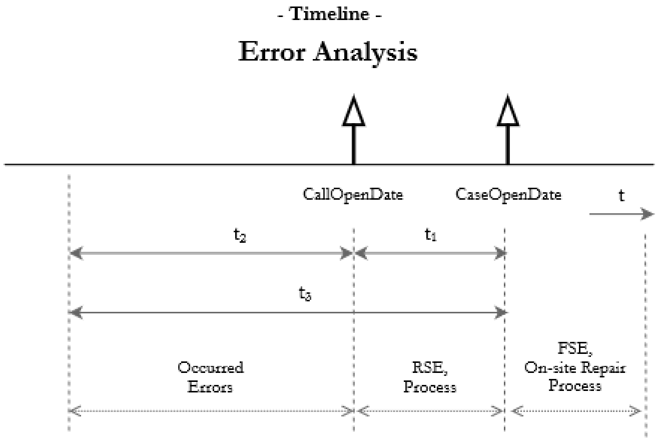


Fig. 19 - Timeline Analyzed Occurred Errors

The last function deals with the alphabetical (ascending) order of the errors. It might be the case when the same set of errors have occurred for two or more different SWO's, but these errors have occurred in a different order. For example, one SWO might have the set {1, 2, 3} as error codes, while another SWO may have a corresponding set of {3, 2, 1,}. Although this does not make any different due to equality in set theory, given '$for\ any\ sets\ A\ and\ B,\ A = B\ if\ and\ only\ if\ \forall x\ [x \in A \leftrightarrow x \in B\ ]$' , all sets of error codes have been ordered alphabetically to mitigate the risk of any potential related issues (*o*).

5.5.2 Additional Case Exclusions

It is observed that not all '*AssetID*'s' have a corresponding SRN, using the only view available in Vertica with this required information. These missing values occur for **7.13%** of the remaining *Ingenia* cases, which are excluded for further analysis after some discussions. It is not possible to obtain the remaining SRN at all or without considerable time consumption.
This surprising observation was further studied, and the following conclusions were made for potential future improvement of Vertica:

1) Values might be missing due to the introduction of a new database containing SWO data; as a replacement of the current OneEMS. The Vertica database consists of raw data collected from different sources; it could be the case that MR Systems, for specific countries, are already connected to this new database, of which raw data is not visible in Vertica yet. This has been confirmed not to be the cause as data regarding these countries (based on a random sample check) is available in Vertica, hence not the cause of missing SRN's.

2) Cases with an available AssetID, but a missing EquipmentNumber and SRN; it seems that this subset of the missing values corresponds to MR Systems that are not remotely (online) connected. The only view available and used joining AssetID's and SRN's apparently only consists of remote systems. Potentially fault(code) data is registered and available for these systems and corresponding cases, but cannot be accessed due to an unknown (required) SRN number.

3) Cases with no available AssetID in the SWO related tables, and therefore also no available EquipmentNumber and SRN; a random sample of known case numbers where manually searched for using a tool used by RSE's. This shows that some of these cases, in fact, do have a known AssetID. It is unrealistic to manually look up the corresponding AssetID's for **1.37%** of such cases, not knowing if all will result in known AssetID's and SRN's.

Table 15 - Final Data Set Ingenia based on scope, after Preparation and Cleaning

	Total	Multiple Visit	Single Visit
# Cases including all SRN's	42 971	15 792	27 179
# Cases excluding 1000 < SRN > 97 000 range	40 056	14 546	25 510
# Cases with Chain 1 Errors	1 926	773	1 153
# Cases with Chain 2 Errors	4 048	1 544	2 504
# Cases with Chain 3 Errors	4 400	2 945	1 455
# Cases with Chain 4 Errors	11 153	6 608	4 545
# Usable Cases: Chain 1–4 errors or combination	17 136	6 618	10 518

5.6 DATA INTEGRATION

The final step was to merge the different data-frames after preprocessing into the final data set, as well as dividing the data into a train and test set. Merging all the input data together results in a cleaned data set with 169k rows and 17 columns or features, with no missing values. This dataset, containing four different MR systems, is used for FVF and parts analysis and consists of all system and case characteristics-related selected features (*Section 5.1*, *Table* 8), except for '*CaseCloseDate*', '*Labor Duration*' and '*Labor Activity*'. These last two are replaced with the data construction variables net part cost and labor cost (*Section 5.2.3*).
However, for RCA, the focus is on the subset with *Ingenia* (42k cases, 17k usable) systems, as aforementioned in *Section 1.9*; consisting of the following features: '*Parts Consumed (Material Description)*', '*Product Group*', '*Priority*', '*Market*', '*Entitlement Type*'', '*CaseNumber*', '*MR Chain*', '*Chain Unit*', and '*Errors*'.
Finally, splitting the data results in a train and test, a distribution of 70% and 30% respectively is applied as is often used in machine learning settings (Adi Bronchstein, 2017). During modelling, k-fold cross validation method is used for validation, whereby the data set is divided into ten equal sets. For a more detailed data overview and modeling of RCA, see *Section 6*.

5.7 STATISTICAL SIGNIFICANCE

Now that the data preparation phase is complete, it is important to check whether the two sub-sets of FVF, namely cases with single required visits and those with multiple required visits, differ significantly, before further FVF KPI analysis is done as part of related research questions, and conclusions drawn based on further calculations. As the '*Total CM Cost*' has been used as variable to determine normality in the log transformation regarding outlier identification, we use the same variable to determine which conclusions can be drawn based on above FVF calculations. All relevant 'R' code for the following section can be found in *Appendix G*.

Table 16 presents the calculated average cost for single and multiple visit data per year, along with the corresponding sample sizes and standard deviations, based on the log-transformed data. For all subsets per year and overall, Welch's Two Sample t-test has been performed and corresponding p-value is determined. The researcher has made the decision to use this specific t-test instead of the more common Student's t-test as Welch performs better than Student's t-test whenever sample sizes and variances are unequal between groups, and gives the same result when sample sizes and variances are equal.
For all cases, p-values are very small and can be considered equal to zero. Hence, there is a significant difference in total (net part & labor) cost for single and multiple visit-cases. Meaning that FSE's who spend multiple visits per case, on average, spend significantly more in terms of total costs than those who do not.
To confirm that sample differences are indeed statistically different, we additionally use the concept of 95%- confidence intervals and notched boxplots. Confidence Intervals have been determined and presented in *Table 17*. These values, more easily observable via notched box plotted scales with 95% CI, further strengthen this finding, since intervals are not overlapping. Due to this result, there is

Table 16 - Statistical Significance 'Total CM Cost' - 1

Year	Log Avg. Multiple Visit Cost	Log Avg. Single Visit Cost	N (Multiple Visits)	N (Single Visits)	Log SD (Multiple Visits)	Log SD (Single Visits)	p-value
2013	8.125362	7.373383	6 964	8 363	1.053871	1.267979	2.2e-16
2014	8.133249	7.312643	9 189	13 177	1.064133	1.296779	2.2e-16
2015	8.163601	7.352897	11 611	17 192	1.061210	1.279057	2.2e-16
2016	8.255559	7.409511	11 957	18 952	1.044438	1.290363	2.2e-16
2017	8.254501	7.321626	11 855	20 417	1.042176	1.310895	2.2e-16
2018	8.278300	7.341018	9 586	18 260	1.053283	1.321552	2.2e-16
2019	8.290751	7.323257	3 633	7 900	1.055865	1.322577	2.2e-16
Overall	8.212886	7.349294	64 795	104 262	1.054563	1.299942	2.2e-16

strong evidence that the difference between the two medians – per value pair – is statistically significant at the 0.05 level (Mcgill, Tukey, & Larsen, 1978).

However, while p-values can inform whether an effect exists, it does not reveal the size of the effect, even though p-values depend on two aspects: the size of the effect and the size of the sample. One could theoretically get a 'significant' result either if the effect were very large (despite having only a small sample size) or if the sample were very large (even if the actual effect size were tiny). Therefore, it is important to also know the size of the effect by reporting effect sizes; reporting these with their own CI along with the p-values (Coe, 2002; Sullivan & Feinn, 2012). In other words, it might be the case that a statistically significant results was obtained only because of a very large sample size. Therefore, effect sizes should be reported as they are independent of sample size, unlike significance tests (Lakens, 2013).

Table 17 additionally presents the effect size in terms of Cohen's, *d* (based on pooled SD), which suggests that d should be 1) at least equal 0.2 to be considered a small effect, 2) at least 0.5 to be considered a medium effect, and 3) 0.8 or greater to be considered a large effect. Even though *d* is widely used, it does not take into account the sample sizes of the paired comparisons. This underlying assumption is important to this analysis as single and multiple visit subsets are not (always) the same size, as shown previously. Another, but very similar, effect size metric that considers this is Hedges' *g*. Hedges & Olkin, (1985) mention: "*Although using the pooled standard deviation to calculate the effect size generally gives a better estimate than the control group SD, it is still unfortunately slightly biased and in general gives a value slightly larger than the true population value*". Hedges' *g* formula provides an approximate correction, and might result in a different and more accurate outcome. However, the difference with Cohen's *d* disappears with larger sample sizes (Sullivan & Feinn, 2012).

Overall, effect sizes and corresponding CI might slightly differ between *d* and *g* calculation, but are overall very similar. All effect sizes are considered good: classified as medium, or large. Based on this and previous statistics, we can accurately develop conclusions between single and multiple visit data.

Back-transformation is sometimes used as a method for readability purposes of (above) log-transformed data. However, it often will not output the values expected, as, for example, mean values might not match the untransformed mean unless the data are perfectly Gaussian. In addition, confidence intervals might not be symmetrical anymore as (McDonald, 2014). To avoid these issues, but still get a good estimate of the CI's, the bias-corrected and accelerated (BCa) bootstrap interval method is applied on the non-transformed data (Almonte & Kibria, 2004; Burbrink & Pyron, 2008; DiCiccio & Efron, 1996). This resampling CI calculation creates multiple resamples, and computes the effect size of interest on each of these resamples. The bootstrap resamples of the effect size are used to determine the 95% CI. The resampling distribution of the difference in means approaches a normal distribution.

Table 17 - Statistical Significance 'Total CM Cost' - 2

Year	Log 95% - CI, Multiple Visits	Log 95% - CI, Single Visits	Effect Size – Cohen's, d	Log Cohen's d, 95% - CI	Effect Size – Hedges', g	Log Hedges' g - 95% - CI
2013	8.1006 – 8.1501	7.3462 – 7.4006	0.64	0.6071 – 0.6723	0.64	0.6071 – 0.6722
2014	8.1115 – 8.1550	7.2905 – 7.3348	0.69	0.6527 – 0.7075	0.68	0.6527 – 0.6801
2015	8.1443– 8.1829	7.3338 – 7.3720	0.68	0.6536 – 0.7020	0.68	0.6536 – 0.7020
2016	8.2368 - 8.2743	7.3911 – 7.4279	0.70	0.6808 – 0.7279	0.73	0.6808 – 0.7279
2017	8.2357 – 8.2733	7.3036 – 7.3396	0.77	0.7418 – 0.7886	0.77	0.7418 – 0.7886
2018	8.2572 – 8.2994	7.3219 – 7.3602	0.76	0.7329 – 0.7840	0.76	0.7329 – 0.7839
2019	8.2564 – 8.3251	7.2941 – 7.3524	0.78	0.7367 – 0.8178	0.78	0.7367 – 0.8178
Overall	8.2048 – 8.2210	7.3414 – 7.3572	0.71	0.7026 – 0.7228	0.71	0.7026 – 0.7228

This is due to the Central Limit Theorem where a large number of independent random samples will approach a normal distribution even if the underlying population is not normally distributed. This gives us the benefits of: no need to assume and test for normality and it accounts for any skew (Ho, 2019). The 'R'-script and corresponding output of this method is presented in *Appendix J*.

5.8 Error Code Correlation

Correlation values have been determined for all occurred errors in the past two weeks prior to a customer call for the four separate chains, based on the dataset as a whole. The underlying hypothesis is that *Error Codes represent independent failure states*. To test this hypothesis, the correlation between error X frequency and error Y frequency is calculated. If the correlation is weak, the error codes X and Y represent independent failure states, if not; error codes represent dependent failure states. As input to this calculation the output (error code sets) for all CM cases of *Section 5.5* has been used. CM cases where no Chain 1–4 errors occurred were not included as input; hence, potential bias of these values was mitigated.

The 'R'-script is included in *Appendix M*, which includes the creation of contingency tables, error frequency plots, and correlation calculation. For this, Pearson correlation coefficients, Pearson's r, have been determined, associations are considered very high when .90 to 1.00 (-.90 to -1.00), high 0.70 to 0.90 (-.70 to -.90), moderate .50 to .70 (-.50 to -.70), low if .30 to .50 (-.30 to -.50), and negligible in case of .00 to .30 (.00 to -.30) (Chen & Popovich, 2002; Hinkle, Wiersma, & Jurs, 2003; Mukaka, 2012). The correlation matrix per chain and part type is presented in *Appendix N* due to size, conclusions based on this output and relevant correlations are discussed in the paragraphs below. Additionally, corresponding significance are provided in terms of p-value (categorized as $p < .0001$, $p < .001$, $p < .01$, and $p < .05$).

5.8.1 Ingenia – Chain 1

For the cleaned subset of *Ingenia* system CM cases, eight out of 70 distinct Chain 1 (C1) errors have occurred (in any combination) during the analyzed time-period; specifically for part type T01. Additionally, two different failure conditions for T05, 21 for T07 and 28 distinct errors for T08. Correlation coefficients << .70 hold true for most C1 errors, except for: (T01E17, T01E29: $r(3) = .71$, $p < .001$), (T01E20, T01E25: $r(1) = .86$, $p < .001$), (T05E54, T05E60: $r(1) = .1.00$, $p < .001$), (T07E15, T07E16: $r(2) = .72$, $p < .001$), (T07E59, T07E60: $r(2) = .94$, $p < .001$), (T07E37, T07E59: $r(1) = .75$, $p < .001$), and (T08E12, T08E49: $r(10) = .70$, $p < .001$). However, the hypothesis "*errors represent independent failure states*", is proven for most errors, as well as most aforementioned error combinations, due to unsatisfactory degrees of freedom. Only (T08E12, T08E49: $r(10) = .70$, $p < .001$) can be considered dependent. Although not highly correlated, (T08E01, T08E12: $r(54) = .40$, $p < .001$) can be interesting for further investigation based on a larger dataset or system design check, due to its significantly higher degrees of freedom for C1.

5.8.2 Ingenia – Chain 2

Error analysis revealed five different installed part types (T04, T05, T15, T16, and T18) in the *Ingenia* systems, regarding Chain 2 (C2). Errors for T15 were all independent, as can be seen in *Appendix N***.** In spite of interesting and surprising results, recommended for further research, none of the correlated errors, although significant, can be found dependent; due to low degrees of freedom. This holds for: (T04E07, T04E24: $r(2) =.85$, $p < .0001$), (T05E13, T05E38: $r(5) =.77$, $p < .0001$), (T05E28, T05E41: $r(4) =.71$, $p < .0001$), (T16E53, T16E63: $r(4) =.80$, $p < .0001$), and (T18E01, T18E02: $r(1) =1.00$, $p < .0001$). In terms of frequency of occurrence, it is recommend finding out why T05E28 occurs significantly more often (199 times) than other C2 highly correlated errors.

5.8.3 Ingenia – Chain 3

Chain 3 (C3) errors have taken place for seven different part types (T01-07); all errors, albeit a small amount, for T02, T03, and T04 have proven independent due to very low coefficients and non-significant p-values. Further results are two-fold; a few errors are highly correlated and occur very frequently in our data set, and have high degrees of freedom. These are potential errors to cluster as they might also be causal; this holds for: (T06E61, T06E62: $r(857) = .72$, $p < .0001$), (T06E61, T06E63: $r(1659) = .92$, $p < .0001$), and (T06E62, T06E63: $r(857) = .82$, $p < .0001$).
Others, might be useful to look into during future research, but do not provide sufficient power for this study: (T01E13, T01E14: $r(6) = .86$, $p < .0001$), (T01E21, T01E23: $r(5) = .72$, $p < .0001$), (T01E17, T01E18: $r(3) = .73$, $p < .0001$), (T05E53, T05E54: $r(15) = .73$, $p < .0001$), and lastly for another part type (T07E56, T07E58: $r(6) = 1.00$, $p < .0001$).

A large cluster of sixteen errors with extremely high Pearson values can be observed for T05 related errors for C3, ranging from .73 to 1.00 and statistically significant ($p < .0001$) as shown and color marked in *Appendix N* due to its size. These, mostly very high, values unfortunately have low sample sizes and df's ranging from two to six and therefore insufficient data to make any conclusion. However, with SME's discussions took place regarding required future study of these errors and potential clustering as they seem to also be related system design-wise upon further inspection.

5.8.4 Ingenia – Chain 4

With regard to C4; this last MR chain and corresponding errors seems to behave very differently and more random compared to other chains; as can be observed from frequency of and correlations between errors of the only part type (C4T1). Contrary to C1-C3, which either have (very) high r-values or no or very low correlations, overall C4 correlations are relatively high across the board; given that C4 also as a chain operating completely different than the earlier three. Why this is the case cannot be explained, at the time of writing, by the researcher or SME's. For a complete overview of (relevant) correlations, see *Appendix N*, but strong results are as follows: (T01E13, T01E44: $r(4516) = .73$, $p < .0001$), (T01E43, T01E44: $r(2750) = .73$, $p < .0001$), (T01E13, T01E43: $r(2750) = .76$, $p < .0001$), (T01E03, T01E13: $r(845) = .71$, $p < .0001$), (T01E13, T01E29: $r(797) = .71$, $p < .0001$), and (T01E20, T01E21: $r(871) = .85$, $p < .0001$).

5.8.5 Ingenia – Inter Chain & Conclusion

Thus far, results have been presented on different part types concerning a specific MR chain and corresponding errors. It is not necessary to determine cross-part type Pearson's r-values as MR systems only have one installed part type given a specific chain and system; meaning, for example, that a system only contains T01 regarding C1, and cannot have another part type (i.e. T02) positioned at the same time.
Additionally, cross chain values are not required also, as MR chains operate completely independent from each other, i.e. Chain 1 errors cannot results in Chain 2 errors. However, this has been verified and found to be the case.

Concluding, it was assumed that all errors would be independent due to system design, but the section provides insight into potential dependence of some errors across different MR chains, and are candidates to cluster these similar and dependent error codes.
Other output resulted in additional surprising results but even though some very (highly) correlated, and significant (p-values) based on the sample size, the relatively low sample size resulted in low degrees of freedom. These concepts are directly related to each other, making sure that not one is directly the cause of being unable to conclude that errors are correlated to each other; but rather, the low frequency of occurrence (and thus sample size containing such an error) (Hair et al., 2014; Jawlik, 2016; Walker, 1940).

The potential candidate errors for clustering, presented above based on correlation analysis, are not clustered at this point. Since correlation does not necessarily mean causation should exist between errors, further analysis is required. Although causation is difficult to prove in general, the next section focuses on error sequence and frequent pattern analysis, after elaborating on different techniques considered and their suitability, before any decisions are made.

5.9 Error Sequence Analysis

Although above Pearson analysis provides a good idea of how error codes correlate and can potentially be clustered based on this, additional analysis is required which implies or determines causality for and between error codes, to this extend. Due to the nature of the data and system design, it is not preferable to use other methods such as Confirmatory Factor Analysis for clustering or causality on the final data set, as this method is correlation based. Results, after performing said analysis on all error combinations, might not be accurate or realistic as MR system's chains (C_x) operate independently and each system only has one part type (T_y) installed at the time.

However, other techniques are available to analyze the occurred errors within a two-week period before case open dates and order of occurrence. Sequence Analysis methods were considered, such as 1) SPADE (Sequential PAttern Discovery using Equivalence classes), and 2) Granger Causality test. The former algorithm outputs sequence rules, using a vertical id-list database format and assumes that no sequence has more than one event with the same time-stamp, so that the timestamp or interval can be used as event identifier (Zaki, 2001). However, this assumption cannot be made with the dataset of this study, as multiple errors and corresponding cases to different MR systems can happen at the same time or same two-week time interval, which impedes the successful application of this method. The latter method is a time series method for causality between two variables in a time series, finding patterns of correlation and forecasting which error(s) might occur when a certain (set) of error(s) has been observed (Wei, 2013).

5.9.1 Frequent Pattern – Growth (FPG)

One suitable method found is Frequent Pattern – Growth (FPG), which is a mining method for extracting frequent item sets with application in association rule learning (Han, Jian, Yiwen, & Runying, 2004). Although causality requires knowledge about the causal and effect attributes in ones data and involves relationships occurring over time, FPG can imply causal structures along with strong co-occurrence relationship between events (Tan, Steinbach, Karpatne, & Kumar, 2019). FPG aims to build a causality graph of events in which graph nodes make up the events and edges depict discovered rules representing causality between events (Benslimane, Damiani, Grosky, & Hameurlain, 2017; Nguyen & Ha, 2014; Silverstein, Brin, Motwani, & Ullman, 2000).
The technique uses an extended prefix-tree structure on a complete set of frequent patterns by pattern fragment growth (Shidhu, Meena, Nawani, Gupta, & Thakur, 2014); making it very efficient and scalable for larger data sets (Wang & Cheng, 2018). Another advantage of the method is its ability of run without candidate generation, as a results no candidate test is required and no repeated scans of the complete database (Han et al., 2004; Shidhu et al., 2014).

More important, specifically for this study, is its ability to deal with transactional data as input, in which each transaction can contain a set of items. The algorithm output provides information regarding associations between item sets, and thereby providing an indication of causality between errors; which error(s) are likely to follow once a specific error (set) has occurred. Users can provide a minimum support value for a corresponding parameter to which item sets need to comply before they are included in a discovered rules list. The collection of frequent items is ordered automatically by decreasing sequence of support count (Zheng, Yin, Liu, & Zhang, 2015).

For the final dataset consisting of single visit cases of this study, the result set, L, is presented in *Table 18*, providing 16 discovered rules out of 10k instances and 200 error attributes. The list is ordered by decreasing support and provides additional metrics, such as confidence, lift, leverage, and conviction.

Table 18 - FP-Growth Rules (n=16 rules, 10k instances, 200 attributes)

#	Item set A (frequency A) > Item set B (frequency B)	Confidence	Lift	Leverage	Conviction
1	C3T06E63 (1106) > C3T06E61 (1048)	0.95	4.08	0.08	14.39
2	C4T01E13, C4T01E43 (926) > C4T01E44 (661)	0.71	2.24	0.03	2.37
3	C4T01E44, C4T01E43 (934) > C4T01E13 (661)	0.71	2.63	0.04	2.49
4	C4T01E43 (1690) > C4T01E44 (934)	0.55	1.74	0.04	1.52
5	C4T01E43 (1690) > C4T01E13 (926)	0.55	2.04	0.04	1.61
6	C4T01E30 (1046) > C4T01E44 (567)	.54	1.70	0.02	1.49
7	C4T01E13 (2831) > C4T01E44 (1381)	0.49	1.53	0.05	1.33
8	C4T01E44, C4T01E13 (1381) > C4T01E43 (661)	0.48	2.98	0.04	1.61
9	C3T06E61 (2444) > C3T06E63 (1048)	0.43	4.08	0.08	1.57
10	C4T01E44 (3348) > C4T01E13 (1381)	0.41	1.53	0.05	1.24
11	C4T01E43 (1690) > C4T01E44, C4T01E13 (661)	0.39	2.98	0.04	1.43
12	C4T01E13 (2831) > C4T01E43 (926)	0.33	2.04	0.04	1.25
13	C4T01E44 (3348) > C4T01E43 (934)	0.28	1.74	0.04	1.16
14	C4T01E13 (2831) > C4T01E44, C4T01E43 (661)	0.23	2.63	0.04	1.19
15	C4T01E44 (3348) > C4T01E13, C4T01E43 (661)	0.20	2.24	0.03	1.14
16	C4T01E44 (3348) > C4T01E30 (567)	0.17	1.70	0.02	1.08

Fig. 35 and *Fig. 36* in *Appendix O* provide corresponding association pictures (Frequent Pattern Graphs) to L for an alternative overview of the same 16 rules, along with confidence values for each relation between (a set of) errors. These overviews are created for Chain 3 and Chain 4 errors, as no sequence patterns were found for Chain 1 and Chain 2. These results seem to validate the Pearson correlation calculations between distinct errors within chains as rules between similar errors are displayed. Graphs including specific errors description, instead of the re-labeled ones, are also available in the Appendix.

Based on the results of the Pearson correlation and Frequent Sequence analyses, and the similarity of the output, we can conclude that most of the errors and chains are independent from each other, and the following error codes for Chain 3 and Chain 4 can be clustered:

- C3T06E61 & E62 & 63 > C3T06E616263
- C4T01E20 & E21 > C4T01E2021
- C4T01E13 & E43 & E44 > C4T01E134344

5.10 SPARE PART CLUSTERING – APPROXIMATE STRING MATCHING

Given the approximately 1k distinct spare parts in the final dataset, in which many parts are very similar as they represent different variations of the same spare part, or intended for different systems, clustering of these parts is required. This results in more general and understandable part clusters. For this, some basic text mining is needed. Approximate string matching is an important subtask of many data processing applications applied in the context of i.e. statistical matching, text search, text classification, spell checking, and genomics. At its core lies the ability to quantify the similarity between two strings in terms of string metrics; which can be divided in 1) edit-based distances, 2) q-gram based distances and 3) heuristic distances (van der Loo, 2014). To determine edit-based distances one counts - possibly weighted - the number of fundamental operations necessary to turn one string into another. These operations include substitution, deletion, character insertion, and/or character transposition. Distances based on q-grams are obtained by comparing the occurrence of q-character sequences between strings. Lastly, heuristic measures have a less strong mathematical underpinning, although not less effective, but have been developed as a practical tool for certain applications (van der Loo, 2014; Vogler, 2013).

Although various metrics could be used in 'R', depending on the application and input, some general considerations are useful. The choice between an edit-based or heuristic metric on one hand or a q-gram based distance on the other, is to an extend prescribed by string length. Contrary to edit-based or heuristic metrics, q-gram metrics can be computed between very long strings since the number of q-grams encountered in natural language (i.e. q≥3) is usually less than the q-grams allowed by the alphabet. The choice of edit-based distance mostly depends on required accuracy. For example, in a dictionary lookup where differences between matched and dictionary items are small, an edit distance that allows for more types of edit operations (i.e. optimal string alignment or Damerau-Levenshtein) may give better results. The heuristic Jaro- and Jaro-Winkler distances were designed with human-typed, relatively short strings in mind, hence area of application is clear and useful for this project (van der Loo, 2014).

5.10.1 Heuristic Distance Measure: Jaro-Winkler

Since the goal of this string clustering is to group together similar spare parts, which are all relatively short string-based descriptions, a heuristic measure would be preferable. Which, quickly results in the use of the Jaro-Winkler method.

The Jaro distance has been successfully applied to statistical matching problems concerning fairly short strings (Jaro, 1989). It therefore measures the number of matching characters between two strings that are not too many positions apart and adds a penalty for matching characters that are transposed. The distance measure is given by (Dreßler & Ngonga Ngomo, 2017; Jaro, 1989; van der Loo, 2014):

$$d_{jaro}(s,t) = \begin{cases} 0 \; when \; s = t = \varepsilon \\ 1 \; when \; m = 0 \; and \; |s| + |t| > 0 \\ 1 - \frac{1}{3}\left(w_1 \frac{m}{|s|} + w_2 \frac{m}{|t|} + w_3 \frac{m-T}{m}\right) otherwise \end{cases}$$

Here, the w_i are adjustable weights but in most publications they are chosen equal to 1. Furthermore, m is the number of characters that can be matched between s and t, and t representing the number of transpositions. Also, assuming that $s_i = t_j$ are considered a match only when:

$$|i - j| < \left\lfloor \frac{\max\{|s|, |t|\}}{2} \right\rfloor,$$

and every character in s can be matched only once with a character in t. Finally, if s' and t' are substrings of s and t respectively, obtained by removing the nonmatching characters, then T is the number of transpositions necessary to turn s' into t'. Here, nonadjacent transpositions are allowed.

Winkler extended the Jaro distance as he believed that similarity score between two strings that have a longer set of symbols in common at their beginning should have a higher similarity score than those which contain a mistake or difference in first few symbols (Winkler, 1999). This was achieved by incorporating an extra penalty for character mismatches in the first four characters. The Jaro–Winkler distance uses a prefix scale p, which gives more favorable ratings to strings that match from the beginning for a set prefix length, l. The lower the Jaro–Winkler distance for two strings is, the more similar the strings are. The score is normalized such that 1 equates to no similarity and 0 is an exact match. The Jaro-Winkler distance is given by (Winkler, 1990):

$$d_{jw}(s,t) = d_{jaro}(s,t)[1 - p\ell(s,t)] \, ,$$

where $\ell(s,t)$ is the length of the longest common prefix, up to a maximum of four characters and p is a user-defined weight. Restrictions set to $p \in \left[0, \frac{1}{4}\right]$ making sure that $0 \leq d_{jw}(s,t) \ \leq 1$. The factor p determines how strongly differences between the first four characters of both strings determine the total distance. If $p = 0$, the Jaro-Winkler distance reduces to the Jaro distance; therefore, all characters contribute equally to the distance function. If $p = \frac{1}{4}$, the Jaro-Winkler distance is equal to zero, even if only the first four characters differ. Both Winkler (1990) and (Cohen, Ravikumar, & Fienberg (2003) use a value of $p = 0.1$ and report better results in a statistical matching benchmark than with $p = 0$.

With over a thousand distinct spare parts in our final data set of single visit cases, a fair balance needs to be obtained between number of clusters and the number of distinct parts allocated per cluster. Distinct spare parts are clustered based on the determined Jaro-Winkler distances. After trial and error the number of clusters has been set to 105; a lower number would results in multiple clusters with large amount of parts, while a higher number results in much more clusters consisting of single digit number of parts.

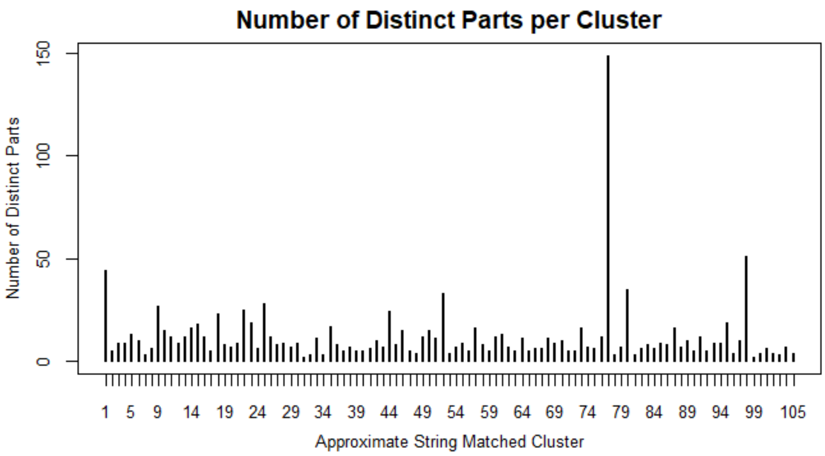


Fig. 20 - Number of Distinct Parts per Cluster

In order to give a slight important to similar parts (starting with the same characters), the p value has been set to 0.01. Increasing this value does provide a technically better clustering in terms of approximate string matching, but results in spare part matching that are functionally not similar. *Fig. 20* provides a visual overview of number of spare parts over all *105* clusters, while the "average number of parts per cluster" – overall – equals *11.73*. *Table 19* provides an example of cluster output, with part number, cluster number, and part description.

Table 19 - Approximate String Matching - Example Cluster Output

Part Number	Cluster	Part Description	Part Number	Cluster	Part Description
658	77	FRU, dS Wrist 8ch 3.0T	1029	90	PHANTOM BOTTLE NiCL
657	77	FRU, dS Wrist 8ch 1.5T	1028	90	PHANTOM HOLDER SH8
656	77	FRU, DS HEADNECK COIL 3.0T	1027	90	PHANTOM BOTTLE 2000CC L13
655	77	FRU, DS HEAD COIL 1.5T	1026	90	PHANTOM BOTTLE 1000CC 3T SPECTRASYN
654	77	FRU, dS Head 32 ch 3.0T	1025	90	PHANTOM BOTTLE 3000CC 3T SPECTRASYN
653	77	FRU, dS ANTERIOR 1.5T	1024	90	PHANTOM BOTTLE 1000CC L11
…	77	…	1023	90	PHANTOM BOTTLE 3000 CC L13
650	77	FRU, dS FOOTANKLE 8CH 1.5T	1022	90	PHANTOM HOLDER NVC-ACR
646	77	FRU, DS BASE COIL 1.5T			

Although Jaro-Winkler provides a proper output clustering similar part descriptions, this has to be checked manually as well. Part descriptions may look the same and have a small similarity distance; practically and effectively, they may be completely different. Alternatively, part descriptions can vary while the same part is being described, such as the use of abbreviations. Hence, to make sure that not only clusters are created based on string similarity, but also on part functionality, a manual check has been done by the researcher along with 'System Specialist' and 'Architect Serviceability' SME's. Due to this, the number of clusters could have been decreased; spare parts belonging to the same category have been grouped further, small mistakes have been fixed, and checked for alternative ways of writing for similar or identical part descriptions. As a result, now the number of clusters has been reduced to a total of 15 (labeled as: PartCluster1 – PartCluster 15). The defined part clusters used for the remainder of document, along with original labels and class imbalance (further elaborated in *Section 6.2*), is presented in *Appendix S*.

6 MODELING

This chapter is the result of the fourth phase of the CRISP-DM methodology (*Section 3.2*). In the following sections, first an overview of the modeling assumptions is presented, followed by model selection and corresponding theoretical descriptions of these different (selected) data mining techniques. Most importantly, the chapter concludes with an assessment and validation of the generated model(s).

6.1 MODELING ASSUMPTIONS

The algorithms, models and underlying data rely on assumptions made during the modeling process, hence the importance of validating the unsupervised learning methods via several model quality tests. The assumptions made underlying the machine learning techniques are listed below:

- Available data instances are independent and identically distributed (IID). Models are trained on a subset and tested on another; assuming that the two are correlated when taking care of overfitting.
- The final dataset discussed in *Chapter 5* is representative of the original, raw, data.
  This implies that after all the original data has been cleaned, the remaining dataset, based on the scope, cleaning and preparation, is still representative of the original data. This is important, since only this data is used to construct the model. However, ideally, the model is applied to all operational systems and MR chains, including those excluded, for a complete model. Said chapter discusses the data preparation and cleaning including justification for made decisions.
- The input order of the data is not important to the models/algorithms; all data is treated the same.

6.2 DATA OVERVIEW

This subsection provides a brief overview of the final dataset used for modeling and the RCA. A couple of characteristics, that are the result of processed data and decisions made in *Section 5*, are mentioned that should be considered in this phase of the study.
Binary classification applications and studies tend to deal with so-called imbalanced datasets, which are datasets with high difference between proportions of class labels: class distribution skew (He & Garcia, 2009; Sun, Wong, & Kamel, 2009). Awareness of such imbalance is important so that appropriate considerations are made during model development.

A first imbalance is observed the classification of single or multiple visit cases, as previously shown i.e in *Section 5.5*. However, due to the aim of the models it is wise to only use the subset of single visit cases, to get a trained model based on problems solved with parts in a single try. Hence, this imbalance is not an issue. Within the single visit subset, another imbalance is found for the amount of cases of the different chains of a MR system. Compared to other chains, chain 4 cases are present in the data set by a factor of 2. Hence, it might be an option to build separate models for the different chains to avoid overfitting to the chain 4 related cases, along with one large, all including model. *Table 20* shows this skew on four levels.

Thirdly, looking at the frequency of all different errors, also a large skew is seen, with a long right tail, with almost half of the distinct errors occurring a few amount of times. This issue is less apparent zooming in on distinct errors per chain (*Fig. 21*), however, it is still worthwhile to mention and take into account during modeling (*Section 6.6*); potentially by creating models per chain or using one of the available cross validation methods.

Table 20 - (Class) Skew of final dataset on multiple levels

	Total	Multiple Visit Cases	Single Visit Cases
# Usable Cases: Chain 1–4 errors or combination	17 136 (100%)	6 618 (38.62%)	10 518 (61.38%)

Entity	% of Cases Containing Chain X Errors	Number of Distinct Occurred Errors
Chain 1	8.73%	53
Chain 2	18.96%	63
Chain 3	22.29%	43
Chain 4	50.02%	36

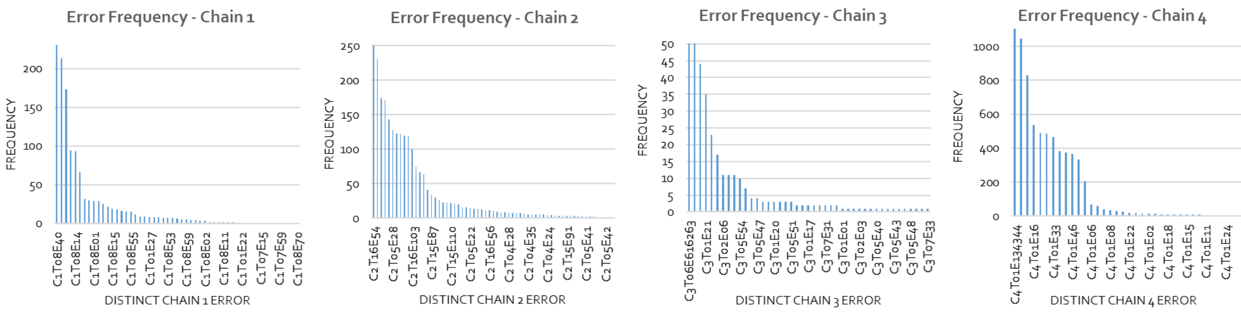


Fig. 21 - Distinct Error Frequency Skew per Chain [3]

Lastly, as the complete dataset has an imbalance in terms of the target variable (part cluster), created train and test sets most likely will also be skewed with regard to this feature as well. Several resampling techniques exist which aim to modify imbalanced data into balanced distribution using a specific method. The modification takes place by altering the size of original or training data set and provide the same proportion of balance. Note that such re-sampling methods are to be applied if a machine learning algorithms is not inherently capable of handling unbalanced datasets. Methods for resampling include (Tantithamthavorn, Hassan, & Matsumoto, 2018):

- *Undersampling* – is one of the commonly used strategies to deal with unbalance in empirical data, which uses majority classes to reduce the number of observations (randomly eliminate) from such classes to balance the data set. The is recommended to use the strategy when the data set is large, additionally improving model run time, but may result in training data losing important information pertaining to majority classes.

- *Oversampling* – Contrary to the previous strategy, it replicates observations from minority classes to balance the data; also called up sampling. Random oversampling has the advantage of not having any information loss, but it does add replicated observations in the original data set; hence, potentially risk of overfitting. The strategy results in higher training accuracy, but lower accuracy for unseen data.

- *Synthetic Data Generation (Hybrid)* – Combining the above techniques by generating an augmented sample of data is possible via a smoothed-bootstrapping approach to draw artificial samples from the feature space neighbourhood from the minority classes: called SMOTE (Synthetic Minority Over-sampling Technique) (Blagus & Lusa, 2013; Chawla, Bowyer, Hall, & Kegelmeyer, 2002). Note that Random Over-Sampling Examples (ROSE) is also very well known, but this – at the time of writing – only is suitable for binary classification.

The study' test design explains the selected method(s) for resampling for our purpose, in *Section 6.6*.

[3] Note, these graphs are just for simple visualization purposes. Axis have been altered for a general idea of error frequency and skewness. The first Chain 2 error occurred 749x, while the first two errors of Chain 2 occurred 2675 and 177x respectively, and the first Chain 4 error took place 5289 times. These large numbers are, in some cases, due to error clustering.

6.3 MODEL FORMULATION

Classification aims to identify to which of a set of categories a specific data instance belongs to; based on a set of training instances for which the category is already known. Every data instance can be defined as a vector x consisting of n features and a corresponding class label y.
Once the classification algorithm is implemented, it aims to predict the class label – noted as $\hat{y}$ given a vector $x = \{x_1, x_2, \dots , x_n\}$ based on previous training with upfront known x and y values. As mentioned before this study refers to conducted *part replacements as the target variable (outcome)*, defined class label y. And, x as a feature vector representing characteristics of the conducted CM case and corresponding MR system. Although the classification problem in this study's context is not binary in nature, with only two possible class labels such as *true* and *false*, or *successful* and *unsuccessful*, due to the multiple part clusters (as presented in *Section 5.10*), it does represent a multi-class classification problem, which does not have a constraint on number of classes. Based on input values, the model determines the probability for parts from a cluster required to a CM case. The part cluster with the highest probability is labeled as the predicted outcome; either the part cluster is predicted correctly or not. A simple visualization of the multi-class classification output and logic is depicted in *Fig. 22*.

	$\hat{y}_1$	$\hat{y}_2$	…	$\hat{y}_n$	$\hat{y}_{max}$	$\hat{y}$	y
Data Instance 1, *Probability Part Cluster n*	0.72	0.08	…	0.20	$\hat{y}_1$	1	1
Data Instance 2, *Probability Part Cluster n*	0.35	0.54	…	0.11	$\hat{y}_2$	2	1

Fig. 22 - Classification Output Example

Given common terminology, class labels are defined as positives and negatives. Therefore, the set of class labels outcomes for this classification problem is defined as $\{+, -\}$, where $'+'$ represents a positive class, and $'-'$ represents an negative class. For data instances in the validation (test) set, each one has a real class y and can be assigned a predicted class label $\hat{y}$. The differences between y and $\hat{y}$ function as input to assess the performance of a classifier the following is defined for this purpose:

- TP: Correctly Predicted Real Positives, where $(y = + \,\&\, \hat{y} = +)$
- TN: Correctly Predicted Real Negatives, where $(y = - \,\&\, \hat{y} = -)$
- FP: Wrongly Predicted Positives (Real Negatives), where $(y = - \,\&\, \hat{y} = +)$
- FN: Wrongly Predicted Negatives (Real Positives), where $(y = + \,\&\, \hat{y} = -)$

6.4 MODEL SELECTION

General (candidate) models have been presented in *Section 2*, as part of the theoretical background. Finding suitable estimators for the problem at hand is often a hard part of solving machine learning projects; requiring careful consideration and not necessarily one best method. However, different estimators are better suited for different data and contexts. *Table 21* provides a list of candidate models applicable for our classification problem. Due to the project duration, a choice and focus was made on a subset of modeling techniques, based on the tabular criteria comparison (the darker the grey marking, the more important the criteria); derived from Provost & Fawcett (2013) and Lim, Loh, & Shih (2000).

Based on the theoretical background and discussions with the project group it was decided to initially start with decision tree-based algorithms as they are not considered to be completely black box contrary to other, potentially also suitable, techniques. Specifically, random forest and extreme boosting (XGBoost) are good candidates given their predictive accuracy and speed and ability for parameter tuning and handling of sparse data. Such techniques do not hide all algorithm details from the user and can be visualized, which is especially important in the context of this study. Assuming, such methods are suitable to predict spare part (types) given occurred errors, it is also important for FSE's and RSE's to understand why or how a certain decision or output is provided.

Table 21 - Learning Algorithm Comparison

	Decision Trees	SVM	Naïve Bayes	Neural Networks	Random Forests	XGBoost
Type of Problem, Classification?	Yes	Yes	Yes	Yes	Yes	Yes
Result Interpretability	Moderate	Moderate	Moderate	Bad	Moderate	Moderate
Predictive Accuracy	Moderate	Moderate	Moderate	Good	Good	Good
Training Speed	Fast	Fast	Fast	Slow	Slow	Slow
Prediction Speed	Fast	Fast	Fast	Fast	Moderate	Fast
Parameter Tuning Needed	Moderate	Minimal	Moderate	A Lot	Moderate	Moderate
Irrelevant Feature Contribution Handling	No	Yes	No	Yes	Yes	Yes
Feature Scaling Required	No	No	No	Yes	No	No
Sparse Data Handling	Fine	Fine	Fine	Bad	Fine	Fine
Large amount of observations required	Yes	No	Yes	Yes	Yes	Yes

This is also key for RSE's that provide advice to FSE's in terms of parts relevant to certain customer complaints as these methods are understandable and output reasoning is visible.
Although, other methods such as multi class multi label neural networks are also very interesting for our cause, and might in theory provide interesting results, this algorithm is not included. This is due to the inability of handling very sparse data making it unsuitable for our current final data set – as proven by extremely high error rates in initial testing of various algorithms.

6.5 Model Performance Metrics

Evaluating the classification model' performances, is expressed using confusion matrix-based metrics, as preluded in *Section 3.2*. Due to different output labels in the multi class classification models, solely using the matrix is not as meaningful, hence the use of other common used metrics (Powers, 2011). *Table 22* shows the selection of different measures, along with a brief explanation and formula used determining its value.

Intuitively, one might use accuracy as a metric to determine how often a model predicts a (new) case correctly, as this metric exactly quantifies this proportion. However, for this study this might not be the best metric as this might provide an optimistic accuracy estimate in case of imbalanced data by classifying cases to the dominant class(es) and does not consider relative importance between all correctly classifying positive and negatives. This is not an issue if equal importance is given to each of the classes, but as aforementioned, not all classes are equally represented in the data set. Hence, accuracy itself is not the best evaluation method for this multi class classification. This problem can be overcome by introducing another estimate of a balanced accuracy (Brodersen, Ong, Stephan, & Buhmann, 2010). The balanced accuracy is defined as: $\frac{1}{2}(\frac{TP}{P}+\frac{TN}{N})$. If the classifier performs relatively well on the different classes, then this term reduces to the conventional accuracy which is the number of correct predictions divided by the number of predictions. However, if the value of the conventional accuracy is solely high due to the imbalanced test set, then the balanced accuracy decreases to chance (Brodersen et al., 2010).

The balanced accuracy metric is derived via TPR and TNR values, which represent the sensitivity and specificity respectively. Both are other common used metrics and express the fraction of correctly classified positives or negatives from their corresponding real set. These are useful as can be argues that both are not directly related to class skew as they are defined over a real set. The same study mentions that class skew of a training set does not need to be the same as the skew during operation (Korst, Pronk, Barbieri, & Consoli, 2019; Sokolova & Lapalme, 2009). However, comparing model performances it might not always be clear which one is superior due to the comparison of two separate metrics. For this reason, the Area Under the Curve (AUC) is used which summaries the

combination of TPR and TNR by averaging all the different AUC's from each class' Receiver Operating Characteristic (ROC) space; which is the area under the classifiers curve as a fraction of the unit square (Provost & Fawcett, 2013).
All metrics are determined for the different models, and are used to evaluate their performances.

Table 22 - Model Evaluation Metrics

Metric	Alternative Term	Explanation	Formula
Accuracy	Conventional Accuracy	Depicts the amount of rightfully classified labels. Note, using this metric on imbalanced data, the model will have the tendency to favor a majority class. The AUC can be used to deal with this risk, as it indicates overall model performance.	(TP + TN) / (TP + FN + TN + FP)
Balanced Accuracy	-	Adjusted accuracy value in case of highly imbalanced data with classifications of cases to dominant classes.	(TPR + TNR) / 2
Error Rate	-	Number of all incorrect predictions divided by the total number of the dataset	(FP + FN) / (TP + FN + TN + FP)
Precision	Predictive Positive Value	Ratio of correctly predicted positive observations to the total predicted positive observations	TP / (TP+FP)
Sensitivity	True Positive Rate (TPR), Recall	Number of correct positive predictions divided by the total number of positives	TP / (TP+FN)
Specificity	True Negative Rate (TNR), Selectivity	Proportion of negatives correctly classified as negatives.	TN / (TN+FP)
F1	Weighted Average	Weighted average of Precision and Recall. Takes both the FP and FN into account. More useful than accuracy, given an uneven class distribution	2*(Recall * Precision) / (Recall + Precision)
Cohen's Kappa	-	How well a classifier performs as compared to how well it would have performed simply by chance.	(1-observed agreement) / (1- hypothetical probability of chance agreement

6.6 Test Design

Given the data overview the decision was made with the project team to create multiple multi-class classifiers per machine learning algorithm. The complete single visit final data set is used for an overall model consisting of all potential errors, along with other system and case variables such as: *System Model* (categorical, ranging from 1–8, *Priority* (categorical, ranging from 1-5), *Market* (categorical, 1-14), *EntitlementType* (categorical, either 1 or 2), and of course different *Chain 1-4 errors*. And the set is also divided in cases with errors only from a specific chain, such that we have four separate subsets with SWO's for each chain.

6.6.1 k-fold Cross Validation

The data sets are divided randomly in a training set and a test set based on a 70:30 ratio, and in order to avoid over- and or under fitting, the k-fold cross validation method is used. This has a single parameter 'k' which refers to the number of groups that a given data sample is split into; hence the naming of the procedure to 'k-fold' cross validation. As explained in *Section 3.2*, this is primarily used in applied machine learning when using a limited sample to estimate how the model performs when required to make predictions based on non-training data. Moreover, this approach has the advantage of not wasting much data. It is important to select the value of k such, that any risk of model accuracy misrepresentation (e.g. bias or high variance) is mitigated. Through experimentation and studies, it is found that generally a model evaluation with k = 10 results in low bias and modest variance, hence the choice for said k-value (James, Witten, Hastie, & Tibshirani, 2013; Kuhn & Johnson, 2013).

6.6.2 Data resampling

Before training the models, training sets are each resampled to create models based on over-sampled and SMOTE-resampled train data as well. It is important not to resample the whole data set or test sets in order to avoid overfitting. Note that each model uses the same randomly created training and test sets (resampled or not) to have similar input and being able to compare model output and performance fairly. Out of the three presented resampling methods, under sampling is not selected. Although under sampling can be more helpful than oversampling, as it does not change the classification rule, or that it can outperform the third mentioned (hybrid) SMOTE technique, it is not wise to use this resampling method for our dataset; some classes occur in relatively low amount of cases, such that the (train)dataset(s) shrinks significantly (Hulse, Khoshgoftaar, & Napolitano, 2007). Hence, to fulfill the first data mining goals of this project of predicting the most likely spare part cluster based on the input data, and subsequently a specific spare part based on the aforementioned prediction, the following prediction models are created as potential viable solutions: Extreme Gradient Boosting (XGBoost), Random Forest, and Support Vector Machines.

6.7 MODELING TECHNIQUES

Selected modeling techniques from *Section 6.4* are briefly introduced here, in which the same definition for output and input is used, followed by initial modeling results in the next section based on aforementioned test design and baseline models elaborated on here.
Decision trees use a technique called ensemble – further classified into Boosting and Bagging - that helps reducing factors such as variance, bias and noise. Such an ensemble is a collection of predictors, or classification and regression trees (CART's), which are combined in one to give a final prediction; based on e.g. a mean of all tree predictions. A CART provides a real score for each of the model leaves, compared to a traditional decision tree, which contains a single decision value.

6.7.1 XGBoost

Boosting is an ensemble technique in which predictors are not made independently, but rather sequentially. It is an umbrella term for techniques that combine multiple weak learners iteratively into strong learners for improved prediction accuracy. Subsequent predictors learn from the mistakes of previous predictors and therefore observations have an unequal probability of appearing in subsequent models. Moreover, observations with the highest error appear most; hence predictors are not selected based on a bootstrap process but on error. Because of this, it takes less time and iterations for such learners to reach actual predictions.

eXtreme Gradient Boosting (XGBoost) is applied as a boosting technique which creates models that predict the residuals or errors of prior models, added together for a final prediction. The gradient part of the name refers to the used gradient descent algorithm to minimize the loss (Ruder, 2016). This algorithms along with the Random Forest introduced in the next section, use the same ensemble algorithms, but differ in the way models are created. Because of the boosting technique, XGBoost is inherently capable of dealing with unbalanced data, and improving performance of cases that likely are predicted badly; while still being praised by its overall performance and speed (Chen & Guestrin, 2016). However, as a downside, such methods might not handle noise that well.

Before introducing the actual initial training process, XGBoost parameters are highlighted. The booster and task parameters are key as they define the environment of the learning process and specify the learning task, along with the corresponding learning objective. The parameters that are relevant, are:

- **eta** (default = 0.3, range: (0,1)); Step size shrinkage used in update to prevent overfitting. After each boosting step, we can get the weights of new features, and eta shrinks the feature weights to make the boosting process more conservative.
- **objective** (default = reg:linear); most important parameter of the model, specifying which objective function is used for optimization. Although there are 10+ different objective functions, the default value is linear regression, which is not suitable for our goal. Here, specifically a choice is made for 'multi:softmax'. This activation function transforms numeric outputs into class probabilities that sum to one. This is the correct options as we have defined 16 distinct part cluster outputs for our model earlier.

- **max_depth** (default = 6, range: (0, ∞)); Maximum depth of a tree, of which an increased value makes the model more complex and more likely to overfit.
- **eval_metric** (default based on objective); evaluation metric for validation data. A default metric is be assigned according to the objective (e.g root mean square error for regression, and error for classification). For our model, we select multiclass logloss (mlogloss).

Along with other parameters, the snippet below shows the *R* script for the initial set up:

```
numberOfClassesCx <- length(unique(casesCx$partcluster))
paramsCx <- list(booster = "gbtree", objective = "multi:softprob",
            num_class = numberOfClassesCx, eval_metric = "mlogloss",
            eta = 0.03, silent = 1)
xgbcvCx <- xgb.cv(params = paramsCx, data = xgb_trainCx, nrounds = 500,
            nfold = 10, showsd = TRUE, stratified = TRUE, print_every_n = 10,
            early_stop_round = 10, maximize = FALSE, prediction = TRUE)
```

The initial model is trained based on nrounds = 500, which should be enough. Especially including an early stopping, meaning that training should stop if no more learning is done in (in this case) 10 consecutive iterations. However, as apparent from the mlogloss plot (*Fig. 41*), the logloss value has not been stabilized after 500 training iterations; although it has drastically reduced from its initial value starting the training process. Therefore, this is something do take into account during parameter tuning.

6.7.2 Random Forest

Alternatively, bagging is a technique in which many independent predictor models (learners) are built and combined using an averaging technique (e.g. majority vote, normal average). Each observation is selected with replacement to be used as input for each of the models. Thus, each model has different observations based on the bootstrap process. As this method takes many uncorrelated learners to make a final model, it reduces error by reducing variance. One of the selected algorithms within this technique is the Random Forest Classifier (RFC). This essentially works by constructing multiple decision trees and performs aforementioned averaging techniques to perform classification. Its random part comes from the random selection of observations along with features to create each tree. Summarized in a few steps: 1) select random samples from the data (training) set, 2) construct a decision tree for each sample and get a predictions results of each tree, 3) perform a vote for each predicted result, and 4) select the prediction result with the most votes as the final prediction.

Although cross-validation is introduced in *Section 6.6*, this technique does not necessarily need this as there is another way to estimate the test error of random forest models using out-of-bag (OOB) error estimation. For this estimation, roughly 2/3 of the samples are used to construct the tree, and therefore not using the remaining 1/3 of the observations. However, these are rather used as test data. Although OOB estimation is useful for setting parameters, in this study we look at these results as well but (conclude to) primarily rely on k-fold cross validation. Additionally, RFC are known for suffering in performance predicting underrepresented classes in unbalanced datasets: *Section 6.2* (Chen, Liaw, & Breiman, 2004). Without any aforementioned resampling techniques, models focus on improvement by predicting the most represented classes, hence the inclusion of over- and SMOTE sampling methods; being the only classifier type to require resampling.

The baseline model is initiated as follows:

- **mTry** (default =√(number of variables in data frame); amount of variables randomly sampled as candidates at each split.
- **ntree;** number of trees to grow.

```
rf_Cx <- randomForest(partcluster ~., data = trainRFCx,
            ntree = 1000,  na.action=na.roughfix, obb.prox = FALSE )
```

6.7.3 Support Vector Machines

Another widely used classifier technique, but less exposed in the theoretical background, are Support Vector Machines (SVM's). SVM's are a set of related supervised learning methods for classification and regression analysis based on the concept of decision planes that define decision boundaries. Such a decision plane ideally separates objects having different class memberships (Grove & Faytong, 2012; Nisbet, Miner, & Yale, 2018). It creates boundaries for which the margin between the classes is maximized, creating what is termed "optimal separation" (Simske, 2019). While the SVM approach can provide excellent results for used training data, it can be sensitive to noise with small- and medium sized data sets, resulting in lower than expected accuracy. To construct an optimal hyperplane SVM employs an iterative training algorithm, for which k-fold cross validation is applicable, and minimizing an error function. For this study, SVM Type 1 is used, known as C-SVM classification, as one of four distinct type SVM model (two classification based, and two regression based).

Furthermore, SVM's use kernels that can either be linear, polynomial, radial basis function, or sigmoid. While the latter choice closely relates to neural networks, equivalent to a two-layer perceptron network (Kaufmann, 2011), in this study the most commonly known linear classifier is used along with the polynomial kernel.

The baseline model is initiated as follows, based on parameters discussed above:

```
modelCxSVM <- svm(training_setCxSVM$parts ~ ., data = training_setCxSVM,
            type = 'C-classification', kernel = 'linear', scale = FALSE,
            probability = TRUE, cross = 10, gamma = 0.1 )
```

6.7.4 Decision Trees – Visualizing Decision Rules

After the best performing classifier(s) are selected based on one of the three techniques discussed above, decision rules between error data, system data, and case data can be visualized complementary to the predicted part cluster. Training sets for the example trees of C1–4 errors are either up-sampled or not at all, based on part distribution within the subset of the predicted part cluster. In case of equal or slight imbalance, the 70% subset is used for training, otherwise it is upsampled. SMOTE has not been applied as we are – in this stage – dealing with even smaller subsets, and it is decided to use all cases available, hence not down sampling the already relative small majority classes.

The complexity parameter (cp) (default = 0.01, range: (0, 1)) is used to control the size of the decision tree, reduce the likelihood of overfitting, hence pruning the default tree and selecting the optimal tree size. If the cost of adding another variable to the decision tree from a specific node is above the value of cp, then tree building does not continue. In order words, tree construction does not continue unless it would decrease the overall lack of fit by a factor of cp. Using this function in the '*rpart*' package, a penalty is imposed to a tree for having too many splits. A too small cp-value results in overfitting, while a too large value outputs a small tree. As with all other models, 10-fold cross validation is used to train the model, via **trControl** and in this case we also specified the number of cp-value to evaluate via **tuneLength**: 10 instead of the default 3. The optimal cp-value is selected based on the lowest cross-validation error (xerror), also taking into account xstd and the relative error.

```
ControlC2 = trainControl(method = "repeatedcv", number = 10, repeats = 10, classProbs = TRUE,
summaryFunction = multiClassSummary)
cartC2 = caret::train(SpecificPartType ~., data = *traindataset*, method = "rpart", trControl = ControlC2,
tuneLength = 10)
```

6.8 MODEL ASSESSMENT

For a proper comparison, the algorithms are evaluated on the same data using the same metrics. For each model a 10-fold cross validation was used, as aforementioned, and retrieved the results numerically. Configured such, with the same random seed, to ensure same data splits and performance. *Table 24* shows the result of running the code, with the model average accuracy presented first. As mentioned, this metric is not at all suitable for our goal, but this serves as a comparison to the actual (proper) weighted balanced accuracy. In addition, the min and max columns display the minimum and maximum scores corresponding to the worst and best predicted classes. Lastly, the mean AUC value, determined by averaging the AUC under the ROC curve of each class performance of the specific classifier - using the *multiclass.roc* function of the *pROC* package – ranging from .6 to .8.

Random Forest model performance, specifically, was only determined based on resampled trained data, based on the resampling methods comparison on *Table 23*. It is observed that the SMOTE and oversampling methods outperformed non-resampling significantly in OOB error estimation and training error rate. Interestingly, accuracy per models does not differ that much compared to the gain in oob.error rat, but each model does perform better compared to non-resampled data, although no significant differences between SMOTE and oversampling are observed. The models per chain do perform better than the overall model, which tends to focus on acquired their perceived performance by focusing on majority classes. The 'best fitting' model depends on the preferences and requirements of the user, when it comes to these resampling methods.

We can see that the classifiers have an average performance (.52 - .62), while XGBoost clearly scores significantly higher than other estimators do with very good balanced accuracy of .72 - .84 (*Table 24*). For further insight, average test scores for other metrics such as specificity or F1 are shown in *Table 25*, for which a 0.5 F1 score indicates a poor model and values above 0.7 are an indication of a strong model. Comparing results from the classifiers above after parameter tuning, the following conclusions can be drawn:

- Performance of the XGBoost models per MR chain are overall much better than those obtained for Random Forest and SVM, while the last types show similar results with a few differences for all performance metrics. Very good results for the different metrics and models are marked green in the results tables below.
- XGBoost models are able to reach very good specificity values (ranging from .94-.97) (slightly better than other classifiers, which are still great specificity scores) and decent precision outcomes (.60-.70).
- Precision values are relatively low for all Random Forest models (ranging .29-.47) and SVM's (ranging .30-.40), while recall values are slightly better, but moderate at best (ranging .43-.51 and .39-.62, respectively); with an exception of high recall for RFC C2 (Smote) of .74 and SVM C4 (Linear) of .67.
- XGBoost models are the only classifiers with ideally high precision and recall values, as this results in models returning many correctly labeled results. However, XGBoost C3 and C4 models have lower than preferred recall values compared to C1 and C2, which can results in models returning fewer results but mostly correctly predicted labels compared to training labels. Systems with high recall and low precision are not preferred at all, as such models (as the linear and poly SVM models) return many results but most labels incorrectly predicted.
- Given the very low recall value (.48) and moderate precision (.57) the overall (Chain 1-4) boosting model is not preferred to separate models per MR chain.
- Moreover, the same behavior in performance can be observed for all classifiers, where C1 and C2 models tend to perform better overall, compared to C3 and C4. While for C4 this can be explained due to the much more complex system design and errors that are more dependent and seem to behave differently as concluded in *Section 5.8*, an explanation cannot be given for C3 at this point.

- No definite conclusion can be drawn regarding which resampling methods are best. Both oversampling and SMOTE seem to perform relatively the same. Best fitting models might depend on the requirements of the user. Although for C2 subset SMOTE outperforms oversampling. In any case, no resampling is not advised given overfitting and poor OOB and error rates. Given the overall slightly higher scoring performance metrics for SMOTE, we continue with RFC based on SMOTE sampling.

Table 23 - Comparison Resampling Methods, OBB, Train & Test Accuracy (%) [4]

Sampling Method / Subset	None		Oversampling		Hybrid Sampling (SMOTE)	
	OOB [5]	Train Set Accuracy [6]	OOB	Train Set Accuracy	OOB	Train Set Accuracy
Chain 1	68.10	69.89	30.33	77.69	24.68	77.59
Chain 2	62.71	62.40	32.58	74.35	33.84	70.51
Chain 3	68.39	33.52	63.25	41.27	58.00	43.47
Chain 4	63.53	63.53	38.66	66.02	35.29	66.75
All Chains	62.26	52.55	44.74	59.85	46.48	60.02

Table 24 - Classifier evaluation based on model test set - 1

#	Multi Class Classification Model	Accuracy (μ)	Accuracy Balanced (μ)	Accuracy Balanced (min)	Accuracy Balanced (max)	μ AUC
1	XGBoost					
	All Chains Model	0.550	0.716	0.593	0.874	0.693
	Chain 1 Model	0.720	0.841	0.662	1.000	0.803
	Chain 2 Model	0.706	0.822	0.664	1.000	0.802
	Chain 3 Model	0.599	0.742	0.619	0.847	0.726
	Chain 4 Model	0.602	0.722	0.648	0.865	0.697
2	Random Forest (Oversampling)					
	All Chains Model	0.168	0.550	0.498	0.748	0.595
	Chain 1 Model	0.168	0.564	0.469	0.745	0.680
	Chain 2 Model	0.195	0.516	0.500	0.676	0.611
	Chain 3 Model	0.124	0.523	0.483	0.745	0.611
	Chain 4 Model	0.122	0.544	0.500	0.748	0.700
3	Random Forest (SMOTE)					
	All Chains Model	0.128	0.544	0.500	0.745	0.616
	Chain 1 Model	0.323	0.594	0.500	0.797	0.713
	Chain 2 Model	0.258	0.520	0.500	0.747	0.533
	Chain 3 Model	0.324	0.523	0.500	0.678	0.576
	Chain 4 Model	0.289	0.538	0.500	0.824	0.692
4	Support Vector Machines (Linear)					
	Chain 1 Model	0.283	0.575	0.449	0.720	0.631
	Chain 2 Model	0.351	0.604	0.480	0.856	0.608
	Chain 3 Model	0.376	0.620	0.562	0.719	0.741
	Chain 4 Model	0.342	0.540	0.448	0.809	0.638
5	Support Vector Machines (Poly)					
	Chain 1 Model	0.292	0.534	0.450	0.746	0.646
	Chain 2 Model	0.330	0.596	0.462	0.796	0.639
	Chain 3 Model	0.337	0.576	0.475	0.718	0.717
	Chain 4 Model	0.332	0.543	0.454	0.735	0.640

[4] Accuracy = (1 – Training or Test Set Classification Error Rate) * 100. Percentage, ranging 0-100, higher value is preferred
[5] Percentage, ranging 0-100, Lower value is preferred
[6] Percentage, ranging 0-100, Higher value is preferred

Table 25 – Classifier evaluation based on model test set - 2

#	Classifier		Evaluation Metric			
			Precision (avg) (min, max)	**Recall** (avg) (min, max)	**Specificity** (avg) (min, max)	**F1** (avg) (min, max)
1	XGBoost					
		All Cases Chain 1 – 4	0.570 (0.411, 0.935)	0.487 (0.189, 0.870)	0.944 (0.812, 0.998)	0.501 (0.292, 0.788)
		Chain 1 Cases	0.659 (0.333, 1.000)	0.708 (0.333, 1.000)	0.974 (0.944, 0.990)	0.764 (0.400, 1.000)
		Chain 2 Cases	0.696 (0.417, 1.000)	0.676 (0.333, 1.000)	0.969 (0.898, 1.000)	0.674 (0.400, 1.000)
		Chain 3 Cases	0.650 (0.441, 1.000)	0.532 (0.250, 0.877)	0.953 (0.776, 1.000)	0.565 (0.364, 0.800)
		Chain 4 Cases	0.601 (0.390, 0.770)	0.490 (0.304, 0.888)	0.953 (0.841, 0.998)	0.529 (0.378, 0.825)
2	Random Forest (Oversampling)					
		Chain 1 Cases	0.472 (0.067, 0.769)	0.433 (0.333, 0.526)	0.938 (0.857, 0.963)	0.482 (0.333, 0.667)
		Chain 2 Cases	0.355 (0.183, 0.395)	0.451 (0.351, 0.484)	0.938 (0.619, 1.000)	0.317 (0.293, 0.467)
		Chain 3 Cases	0.463 (0.308, 0.833)	0.357 (0.121, 0.667)	0.940 (0.620, 1.000)	0.335 (0.142, 0.521)
		Chain 4 Cases	0.401 (0.137, 0.833)	0.397 (0.200, 0.650)	0.939 (0.763, 0.996)	0.207 (0.150, 0.370)
3	Random Forest (SMOTE)					
		Chain 1 Cases	0.392 (0.250, 0.484)	0.359 (0.167, 0.667)	0.943 (0.805, 1.000)	0.323 (0.200, 0.600)
		Chain 2 Cases	0.285 (0.249, 0.384)	0.743 (0.700, 0.828)	0.938 (0.2825, 1.000)	0.393 (0.383, 0.405)
		Chain 3 Cases	0.459 (0.333, 0.522)	0.379 (0.143, 1.000)	0.942 (0.352, 1.000)	0.324 (0.205, 0.543)
		Chain 4 Cases	0.336 (0.157, 0.753)	0.305 (0.158, 0.727)	0.947 (0.804, 0.993)	0.262 (0.145, 0.740)
4	Support Vector Machines (Linear)					
		Chain 1 Cases	0.359 (0.189, 0.333)	0.511 (0.250, 0.684)	0.892 (0.253, 1.000)	0.386 (0.295, 0.577)
		Chain 2 Cases	0.375 (0.200, 0.778)	0.536 (0.167, 0.824)	0.904 (0.439, 1.000)	0.426 (0.194, 0.626)
		Chain 3 Cases	0.370 (0.275, 0.464)	0.397 (0.407, 0.947)	0.899 (0.781, 1.000)	0.339 (0.211, 0.623)
		Chain 4 Cases	0.358 (0.217, 0.682)	0.670 (0.550, 0.826)	0.904 (0.569, 0.990)	0.444 (0.275, 0.747)
5	Support Vector Machines (Poly)					
		Chain 1 Cases	0.337 (0.235, 0.586)	0.521 (0.250, 0.680)	0.889 (0.571, 1.000)	0.388 (0.242, 0.630)
		Chain 2 Cases	0.363 (0.200, 0.667)	0.624 (0.333, 0.882)	0.903 (0.295, 1.000)	0.353 (0.290, 0.580)
		Chain 3 Cases	0.300 (0.200, 0.456)	0.397 (0.156, 0.960)	0.888 (0.707, 1.000)	0.343 (0.170, 0.620)
		Chain 4 Cases	0.401 (0.200, 0.510)	0.567 (0.120, 0.940)	0.898 (0.431, 1.000)	0.363 (0.280, 0.662)

6.9 REVISED PARAMETER SETTINGS

To improve the created models, usually two options are available in such supervised machine learning studies: 1) gather more data, and perform feature engineering, or 2) revising parameters using the same data. Although the former options tends to provide the best payoff in terms of model performance and – depending on required (pre-) processing of newly extracted data – invested time. However, as SWO's from as early as 2013 until mid- 2019 has been used already, not a lot of additional work orders can be added to the data set. Other data sources have been exhausted as well regarding cases, MR product and chains; hence the decision for parameter tuning on existing models.

6.9.1 Grid Search

To determine the best parameters for the models, each model allowed for a GridSearch CV optimizing strategy. It generates candidates from a defined grid of parameter values that are set up with the *param* function or equivalent. Parameter search uses the score function. Parameter settings are searched and evaluated based on the F1 metric. Subsequently the grid search was fitted for each model and displayed the best hyper-parameters, and corresponding performance.

Hyper-parameters are tuned on the existing models, unlike general parameters that are trained during the learning process, these are not directly learned within estimators. In the case of Random Forest models, the hyper-parameters to be tuned include the 'number of trees' and 'number of features considered when splitting a node (mTry)'. As all RF models have been trained with nTrees = 1000, error rates for each class were observed to be constant. Error rates already started to reach a constant value from nTrees = 650 onwards. Therefore, solely decreasing or increasing this value does not have a positive impact on model performance:

```
controlRFCx  <- trainControl(method = "repeatedcv", number = 10, repeats = 3, search = "grid")
tunegridRFCx <- expand.grid(.mtry = c(1:X))
                # X depends on MR Chain , and total number of distinct errors as features
rf_gridsearchCx <- train(partcluster ~., data = trainCXRFParts_SMOTE , method = "rf",
                metric = "Accuracy", tuneGrid = tunegridRFCx, trControl = controlRFCx)
                print(rf_gridsearchCx )
```

XGBoost models are tuned based on the eta, colsample_bytree, depth, and gamma parameters, but most importantly also the number of rounds. From the baseline models, we observed that after 500 rounds log loss still did not reach a constant error rate, hence this number has also been doubled to see if it leads to better models or if early stopping is achieved:

```
ControlParamteresCx <- trainControl(method = "cv", number = 10,
                                    savePredictions = TRUE,
                                    classProbs = TRUE, verbose = 2 )
parametersGridCx <-  expand.grid(eta = c(0.02, 0.03, 0.05), colsample_bytree=c(0.0, 0.5,0.7),
                                 max_depth=c(3,6,9), nrounds=c(500, 1000), early_stop_round=10
                                 gamma=c(0, 0.5, 1), min_child_weight=2, nrounds=1000,
                                 num_class=numberOfClassesCx, silent=1, nfold=10,
                                 eval_metric="mlogloss", print_every_n=10 )
modelxgboostCxTuned <- xgb.cv(data = xgb_trainCx, booster="gbtree",
                              trControl = ControlParamteresCx,
                              tuneGrid = parametersGridCx )
```

SVM's are optimized, tuning the hyper-parameters 'C' and 'Gamma'. The former represents the cost of misclassification, where a larger C-value give a low model bias and relatively high variance and vice versa. The latter is the parameter of a Gaussian Kernel (to handle non-linear classification, which results in low bias and higher variance with smaller values (Hsu, Chang, & Lin, 2016). Again, parameter values are specified, after which the grid search was instantiated and fitted for each SVM model. Lastly, the best hyper-parameters are displayed and corresponding performance of the best tuned model are determined, via:

```
linear.tuneCXPoly    <-  tune.svm(partcluster ~., data = training_setCXSVM, type = 'C-classification',
                          kernel = "polynomial", scale = FALSE, probability = TRUE, cross = 10,
                          cost = c(0.001, 0.01, 0.1, 0.5, 1), gamma = c(0, 0.1, 0.2, 0.3, 0.4, 0.5)
summary(linear.tuneCXPoly)
best.linearCXPoly    <- linear.tuneCXPoly$best.model
tune.testCXPoly      <- predict(best.linearCXPoly, newdata = test_setCXSVM)
CMCXSVM_tunedPoly    <- table(tune.testCXPoly, test_setCXSVM$parts)
```

Table 26 shows the model performance using tuned hyper-parameters and corresponding improvement. Additionally, Cohen's Kappa metric is shown, which is a score for inter-rater agreement and is often used to compare the observed accuracy with the expected accuracy (random chance). The calculation is based on the difference between how much agreement is actually present (observed) compared to how much agreement would be expected to be present by chance (expected). The difference is standardized to on a -1 to 1 scale, where 1 is perfect agreement, 0 is exactly what would be expected by chance, and negative values indicate agreement less than chance. The original rule of thumb is defined as: < 0 (less than chance agreement), .01-.20 (slight agreement), .21-.40 (fair agreement), .41-.60 (moderate agreement), .61-.80 (substantial agreement), and .81-.99 (almost perfect agreement) (Landis & Koch, 1977; Viera & Garrett, 2005).

Table 26 - Classifier Performance after Parameter Optimization, model.test.set

#	Classifier		Evaluation Metric				
			µ Recall	µ Specificity	µ F1	µ AUC	Kappa
1	XGBoost [7]						
		Chain 1 Cases	0.71	0.95	0.76	0.80	0.69
		Chain 2 Cases	0.70 (+ 0.03)	0.98 (+0.01)	0.68 (+ 0.01)	0.80	0.66
		Chain 3 Cases	0.55 (+ 0.02)	0.95	0.57	0.73	0.55
		Chain 4 Cases	0.52 (+ 0.03)	0.96 (+ 0.01)	0.55 (+ 0.02)	0.71 (+ 0.01)	0.54
2	Random Forest (SMOTE) [8]						
		Chain 1 Cases	0.56 (+0.2)	0.94	0.44 (+0.12)	0.62	0.42
		Chain 2 Cases	0.74	0.94	0.42 +(0.03)	0.71	0.45
		Chain 3 Cases	0.38	0.94	0.32	0.53	0.38
		Chain 4 Cases	0.42 (+0.11)	0.95	0.31 (+0.05)	0.68 (+ 0.1)	0.39
3	Support Vector Machines (Linear) [9]						
		Chain 1 Cases	0. 51	0.94 (+ 0.04)	0. 39	0.73 (+ 0.10)	0.33
		Chain 2 Cases	0.62	0.95 (+ 0.05)	0.43	0.66 (+ 0.05)	0.32
		Chain 3 Cases	0.40 (+ 0.01)	0.95 (+ 0.06)	0.44 (+ 0.1)	0.75 (+ 0.05)	0.24
		Chain 4 Cases	0.67	0.95 (+ 0.04)	0.44	0.64 (+ 0.02)	0.32

After parameter tuning, it can be observed that both the Random Forest and SVM models perform slightly better. The former mostly have an improved F1 score or for C1 and C4 models an improved recall as well. The latter surprisingly performed much better based on the specificity and AUC metrics; now having similar specificity values as the Random Forest models, just underperforming compared to the XGBoost. Smaller improvements are observed for the XGBoost classifiers as well, but these still perform much better across the board compared to the other classifiers. Especially when comparing Kappa values, for which all four classifiers being considered substantial agreement. Whereas RFC's are score lower to this regard, and SVM's very poorly. Concluding, that one should opt for boosting models to predict part clusters or specific part types.

6.10 Association Mining

Based on de outcome of the preferred model, one knows which part or type is most likely required given an occurred error. However, the CM data used to this extend can be used for other insights as well. As part of **RSQ3.3**, this data can also function as input to find out any co-dependency between spare parts. Once a (type of) spare part is identified via above methods, association mining is helpful for additional historic insights regarding additional spare parts likely used with the predicted part.

6.10.1 Association Rules - Apriori

As the final data base – specifically consumed spare parts per case attribute - of this study resembles transactional data, methods as Market Basket Analysis (MBA) (also called association discovery, association rules, or affinity analysis) are very suitable based on the Apriori algorithm. Data preparation of *Section 5* has taken place with this technique in mind as well, such that the final data is suitable as input for MBA and other analyses. MBA is usually used in other domains and context rather than Healthcare, RCA, and MR log file data; as it is part of analytics in retail organizations to determine the placement of goods, designing sales promotions for different segments of customers to improve customer satisfaction and therefore also supermarket' profit (Loraine & Ashok, 2012). Researchers are aiming to use such techniques in other fields as well, such as marketing,

---

[7] Rounds = 1000, max-depth = 6, colsample_bytree = 0, eta = 0.01, gamma = 0

[8] ntree = 1000 (classifiers), mTry & Kappa values-training set based, for C1: (43; 0.92), C2: (49; 0.87), C3: (32; 0.59), and C4: (42; 0.56)

[9] optimal cost and gamma values respectively, for C1: (0.5; 0), C2: (1; 0), C3: (1; 0), and C4: (0,5; 0).

bioinformatics, and education (Kaur & Kang, 2016). Examples of studied MBA aspects in academic literature are, using customer interest profile and interests on particular products for one to-one marketing (Weng & Liu, 2004), and purchasing patterns in a multi-store environment to improve the sales (Chen, Tang, Shen, & Hu, 2004). As a data mining method it focuses on discovering purchasing or consumption patterns by extracting associations on organization' transactional data; and determines which items are brought or used together (Berry & Linoff, 2004).

As mentioned, this technique attempts to structure knowledge by finding associations between items based on transactions involving them. This is a search through the data for combinations of items, resulting in a rule: "*If A (antecedent) occurs then B (consequent) is likely to occur as well*" (Grabot, 2018). It is likely that many co-occurrences occur of which a section is simply due to chance, instead of a generalizable pattern. For this reason, and risk mitigation for too many produced associations and being able to analyze them all (Valle, Ruz, & Morras, 2018), 'support' is used; requiring rules to apply to at least a– user defined – percentage of transactions (Provost & Fawcett, 2013).

6.10.2 Criterion Values

Hence, association rules were determined using criterion values of support, confidence and/or lift. An association rule is thus expressed in the form $X \Rightarrow Y$, where $X \cap Y = \emptyset$. Let X be a set of variables in I, $I = \{i_1, i_2, \ldots, i_m\}$ be a set of all possible variables (in this context all possible spare parts), likewise Y be a set of other variables in I. The support criterion can be defined as the probability of X and Y co-occurring in the transaction data set. A support value, z, would mean that z% of the transactions in the data involve item consumptions of the item focus in the rule. Hence, the support indicates goodness of the choice of rule (Azevedo & Jorge, 2007; Kikuchi, 2016): $support(X \Rightarrow Y) = P(X \cap Y)$. The confidence of a rule $X \Rightarrow Y$ is the conditional probability of observing Y given that X is present in a transaction. Hence, the conditional probability that X is consumed and also Y, and thus indicating the correctness of the rule (Azevedo & Jorge, 2007; Kikuchi, 2016): $confidence(X \Rightarrow Y) = \frac{P(X \cap Y)}{P(X)}$. Lastly, lift of a rule is the ratio of the support if X and Y are independent. If lift is greater than 1 this would indicate that the presence of X in the transaction has increased the probability that Y will occur in the same transaction. Similarly, if smaller than 1 it would indicate that the presence of X has decreased the probability that Y will occur (Kikuchi, 2016; Provost & Fawcett, 2013): $lift(X \Rightarrow Y) = \frac{P(X \cap Y)}{P(X)P(Y)}$.

6.10.3 Association Significance

Although lift is also used for (in)dependence of X and Y, this ratio – in practice – can be >> 1, implying that the relationship between items is more significant than would be expected if the two sets were independent. The dependence and significance of an association can be further quantified.

The dependence between antecedent and consequence of an association rule can be expressed via the Chi-square test for independence, along with the significance of these items. This method could also be used in case there is any suspicion for spurious correlation between antecedent and consequence (Alvarez, 2003; Brijs, Vanhoof, & Wets, 2003). Hence, these values are also determined for each found associations as an indicator.

However, if one only focuses on association mining based for a specific item (spare part), it might be the case that low frequencies or low expected values provide an incorrect view of dependence. In addition, or even as replacement, Fisher's Exact Test is a more appropriate (alternative) method than chi-squared in such situations; this is a statistical significance test used in the analysis of contingency tables and is valid for all sample sizes (Bower, 2003; Chen, 2011); testing significance of deviation from a null hypothesis (e.g. p-value) (VanPool & Leonard, 2011). Using this test, found associations rules are tested for significance for stronger and more specific results. Although, some studies argue that Fisher's test can be conservative, i.e. its actual rejection rate can be below the significance level, it is still a very suitable method for association significance (Andres & Tejedor, 1995).

7 RESULTS

In this section, the results and main findings of the modeling techniques of the previous chapter are discussed. The structure of this chapter is as follows: first, I report on the more complex relations that underlie the ensemble tree models in terms of feature importance, followed by the main findings of predicting part cluster of different models, followed by predicting even more concrete spare part as a next step in the RCA, visualizing and validating decision rules, potential business impact calculations, and concluding with the results of the MBA analysis.

7.1 FEATURE IMPORTANCE - XGBOOST

The section reports on the relation between features and output variable represented by the model' feature importance for additional insights. Feature importance of the XGBoost C1 – C4 models are depicted in *Fig. 23*, as these overall outperformed the Random Forest models. The overall model, containing all chain cases is not included here due to its lower performance (recall = 0.49, precision = 0.57, F1 = 0.50) and extreme imbalance in terms of cases per MR chain (*Section 6.2*). Due to high number of features the figure shows an extracted selection based on features' Gain. The Gain implies the relative contribution of a feature to the model calculated by taking each feature's contribution for each tree in the model. A higher Gain value compared to another feature implies its importance for generating a prediction: simply the improvement in accuracy brought by a feature. Although this is one of the most important factors to look at, also graphs based on Cover importance are in included in *Appendix X*; which measures the relative quantity of observations concerned by a feature. To plot the importance I use the *R* function *xgb.plot.importance* of the *DiagrammeR* package. To plot the output tree the functions *xgb.plot.tree* for one tree, specifying the ordinal number of the target tree or *xgb.plot.multi.trees* for combined multiple trees.

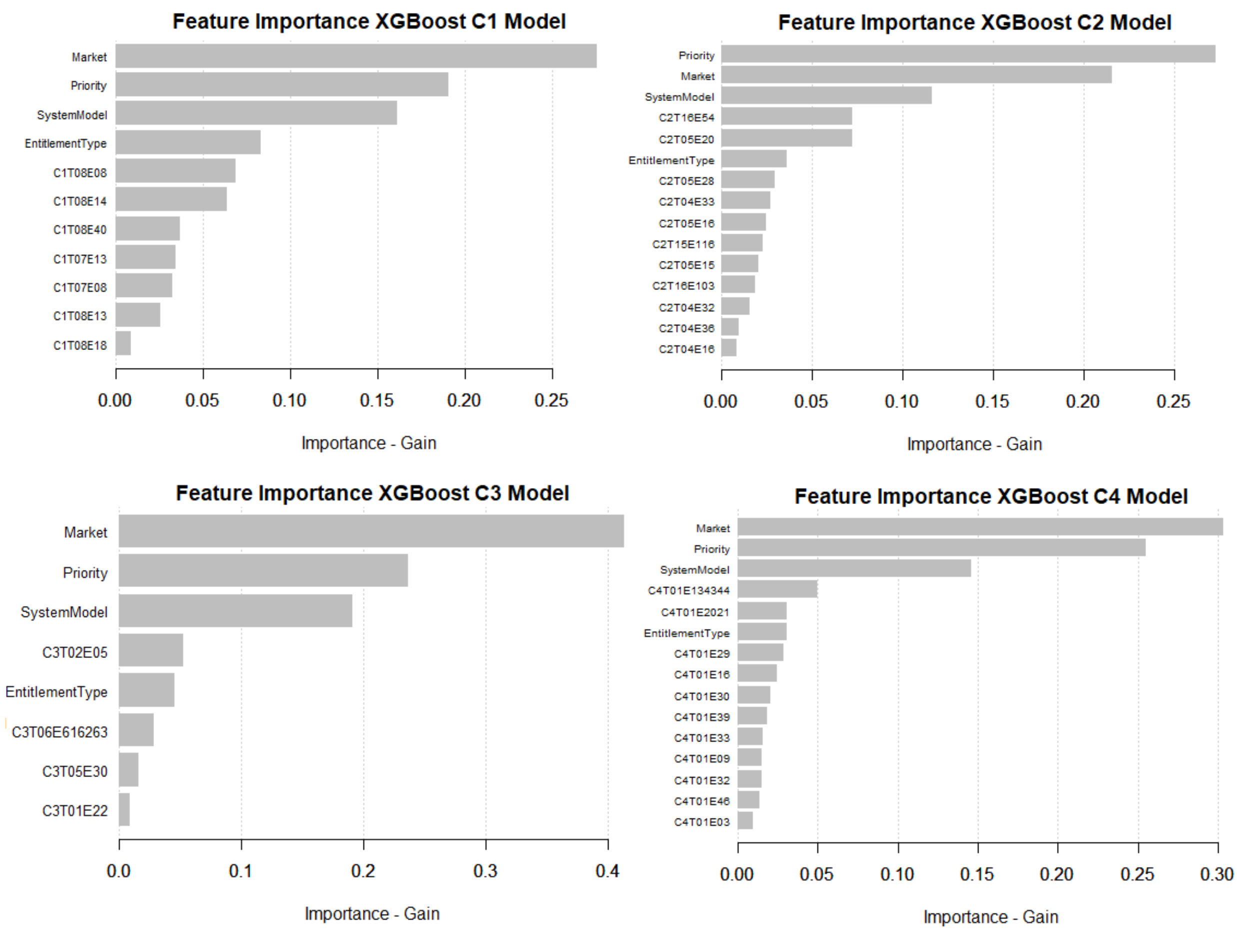


Fig. 23 - Feature Importance XGBoost Models - Gain

What emerges from these plots is that a number of features score high in all four models. Apart from the different errors with high predictive power over the various models, we observe that system model scores high across all plots. This makes sense as different *Ingenia* models not only differ in part composition, and thereby different potential error codes, but also in functionality and price category consisting of more expensive parts. Moreover, an SWO's priority attribute (total of five categories) determines whether the customer complaint is based on a hard or soft failure. In other terms, is the system still working but with caveats or completely nonfunctional; which can determine the severity of the problem. Hence, other required part usage, such as less expensive low level spares or more complex and expensive core system components.
Market, however, is a very surprising feature. More in-depth analysis shows that not all part clusters have been consumed in all markets, in our final single visit data set (*Appendix Z*). Now, this can simply be due to no systems defects have taken place yet requiring parts from those clusters in said markets. Different system models are delivered to all those markets, and therefore we cannot draw any conclusions regarding certain parts needed for different countries. However, specific system usage and occupancy regarding different MRI scans or frequency thereof, might be an external factor that determines more frequent (specific) breakdowns. Unfortunately, this cannot be validated with our dataset, these remain educated guesses.

7.2 Model Findings

From *Table 27* it is clear that some part clusters are predicted much better or worse than others for each of the different XGBoost models. Looking at the results more in depth, we observe that the four different classification error values per part cluster for the different MR chain subset models are also not always consistent. This behavior can be explained by showing that overall the models tend to predict the correct part clusters required, based on evaluation with SME.
Based on domain knowledge and experience, it is to be expected that Chain 1 cases often require Cluster 2 parts to solve maintenance issues, along with Cluster 15 items for various lower level items or Cluster 14 items to support or being able to perform the repair, such as the use of service tools. All these parts are mostly correctly classified. Cases with other consumed cluster items with high classification errors such as cluster 1, 4, 5, and 11 for chain 1 cases are not surprising as they are not expected to be used.
Chain 2 items have multiple cluster potentially required; such as cluster 7, 9, 11 or 13. Unfortunately, cluster 13 items are not included in the test set. Moreover, cases with consumed items of cluster 8, 10 and 12, score well also. These are lower level spare parts to the aforementioned four clusters.
Chain 4, as discussed earlier, is a much more complex MR chain compared to others, and this is also clear from this section along with the XGBoost results from the modelling section. Customer complaints and errors regarding this chain could either require spare parts from cluster 4 or 5 if severe hard or soft failures occur, but also cluster 15 items (complementary to the replacement part) such as cables, accessories or positioning aids. An example of the output provided by the XGBoost model is shown in *Appendix X*, along with corresponding confusion matrices of the four (C1-4) models in *Appendix Z*.

Table 27 - Test Confusion Matrix-based Error rate – XGBoost Models

Classification Error	Part Cluster 1	Part Cluster 2	Part Cluster 3	Part Cluster 4	Part Cluster 5	Part Cluster 6	Part Cluster 7	Part Cluster 8
Chain 1	0.50	0.18	0.00	0.67	0.60	-	0.40	0.06
Chain 2	0.33	0.27	0.16	0.27	0.33	-	0.25	0.20
Chain 3	0.60	0.24	0.33	0.56	0.63	-	0.32	0.56
Chain 4	0.65	0.36	0.69	0.49	0.36	-	0.51	0.52

Classification Error	Part Cluster 9	Part Cluster 10	Part Cluster 11	Part Cluster 12	Part Cluster 13	Part Cluster 14	Part Cluster 15
Chain 1	0.29	0.07	0.46	0.25	-	0.19	0.00
Chain 2	0.30	0.25	0.33	0.20	-	0.00	0.19
Chain 3	0.33	0.50	0.41	0.74	-	0.30	0.12
Chain 4	0.47	0.40	0.49	0.44	-	0.40	0.11

The third model based on chain 3 cases, seems not to perform as expected. The difference in performance between the first two and last two models were apparent from *Table 25* and *Table 26*, but the chain 3 model tends to improve its performance attempting to predict the most likely part cluster (based on SME validation) and surprisingly underrepresented cluster (#1) in this subset. Apart from the final part clustering *0* also presents the part cluster distribution per chain subset. From this, we can see that cluster 1 is really underrepresented in this model.
However, maintenance cases based on this MR chain do not always require replacing complete parts within cluster 1. Discussions with RSE's revealed that chain 3 is essentially the connection between the MR backend and frontend. Due to system design, parts within cluster 1, contrary to others, can also be repaired by replacing separate components. These (low level) items are classified in part cluster 6 and 15.

Surprisingly, after above modelling results (*Table 27)*, cases with part consumption are observable in de test sets per model (see also confusion matrices - *Appendix Z)*, from part clusters that are not expected by SME's. This brings up an interesting point of discussion regarding the different chains selected in the project' scope and underlying assumption to find root causes for SWO's within these chains. *Section 8.5* elaborates on this.

Although XGBoost models have proven to be the better option in predicting required spare part clusters, Random Forest can be used successfully, taking it a step further, in predicting the specific spare parts types given we focus on the predicted part cluster(s) and let a subset of cases having consumed parts from that part cluster be selected. As example, I focus on the C2 subset, where part clusters 7, 9, and 11 have the highest probability of being required for cases from experience; as we know from FVF analysis that RF Coils are the parts for which most financial savings can be achieved as they dominate as consumables (*Section 7.3*). These clusters represent these parts (apart from the low-level RF related parts) also. In essence, one is building a RFC model on a subset of cases, automatically selected based on the previous model output, with these consumed parts and the part cluster as additional feature, for which the results of *Table 28* are achieved; predicting a specific part based on the SWO and Error data. Overall the classifier performs very well with a Kappa of .88, and mid to high F1 scores. Lower balanced accuracy and F1 are marked in orange, but are still acceptable apart from Anterior Coils and Flex Coils having very low F1 scores.

Table 28 - RF Coil Output – RFC.test – PartCluster 7, 9, 11 (mTry = 35, Kappa = 0.88)

	Anterior Coil	Base Coil	Body Coil	Breast Coil	Circu-lator	Coil Assembly	External Coil	Flex Coil	Foot Ankle Coil	Head Coil
F1 Score	0.29	0.69	1.00	0.95	0.97	0.53	0.87	0.41	0.89	0.74
Balanced Accuracy	0.59	0.82	1.00	0.99	0.99	0.95	0.94	0.63	0.92	0.87

	Head Neck Coil	Head Neck Spine Coil	Knee Coil	NVC Coil	PHC	QBC	RF Amplifier	Shoulder Coil	Wrist Coil
F1 Score	0.93	0.57	0.66	0.85	0.80	1.00	0.88	0.59	0.98
Balanced Accuracy	0.98	0.62	0.70	0.85	0.98	1.00	0.89	0.70	0.98

7.2.1 Visualizing Decision Rules

As discussed in *Section 6*, focusing on the subset of '*PredictedPartCluster'* data, insights that are more detailed can be found. Random Forest models, as evaluated for *Table 28*, are referred to as black box models, but it is possible to visualize complete individual trees in the ensemble based from the modelled classifier. One could plot a Conditional Inference Trees (CIT) for i.e. the classifier with highest accuracy; however, this tree would not capture the whole decision rule for the classifier. It would just depict one possible tree; while many others still exist that influence the predicted class. For this reason, if one is interested in the relation between system errors and parts, a single decision tree should be used, which can function as supportive outcome ideally for FSE's on the field. Hence, based on the same data a single decision tree algorithm was used as elaborated on in *Section 6.7*. *Table 29* shows an overview of the results, while *Appendix Z* provides corresponding confusion matrices.

Table 29 – Pruning results, test set prediction performance

Decision Tree	Train set re-sampled?	CP-value	xError	xStd	Kappa	F1
Chain 1 – Part Cluster 2	No	0.010	0.75	0.07	0.63	0.84
Chain 2 – Part Cluster 7, 9, 11	Yes	0.011	0.36	0.01	0.67	0.73
Chain 3 – Part Cluster 15	No	0.020	0.59	0.04	0.42	0.56
Chain 4 – Part Cluster 5	Yes	0.014	0.56	0.01	0.58	0.65

Fig. 24 and Fig. 25, show the classification to the, distinct and potential part types based on the highest predictors '*Priority*', '*SystemType*', and '*EntitlementType'* and distinct system errors for Chain 1 and Chain 2 cases, respectively, as example: PartCluster 2 and (PartCluster 7, 9, 11). 'Market' has been left out at this stage, as it is not fully explainable based on current data why this exactly is a high ranking feature. The outcome has been reshaped from a traditional decision tree – for improved formatting and more detail - via the '*partykit*' package in *R*, to only present the part type with the highest determined probability instead of the usual probability distribution shown for each class per node, along with the error rate. The package has a downside, in which p-values are not shown at the nodes. Significance has been determined to be p < 0.001 at each node. Or, a more traditional tree can be obtained as *Fig. 25*; due to size restrictions for this document. Other decision trees (visualizations) for the other Chains are available in *Appendix X*.

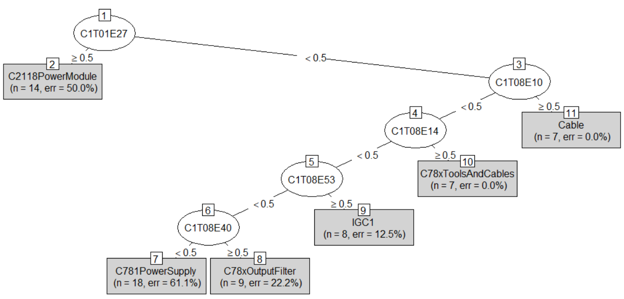


Fig. 25 - Single Decision Tree, PartCluster 2, C1 Cases

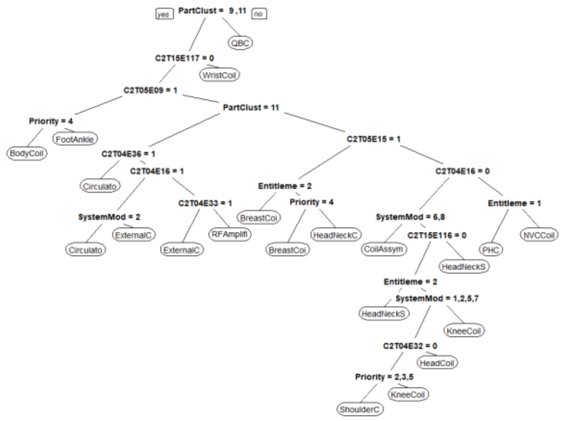


Fig. 24 - Single Decision Tree, PartCluster 7, 9, 11, C2 Cases

Field Service Engineer Validation

Although the decision trees are validated in terms of a test set, an attempt is made to additionally validate it with a SME. The visualization are discussed with a FSE, who was able to provide insights in the depicted C1 and C2 trees. Unfortunately the SME was not able to help with the C4 visualization, as these error codes were not familiar. In this study's dataset, errors regarding chain 4 are labeled with a specific fault code, where the SME is used to a brief failure description provided by the on-site system. Regarding C3; *Table 29* suggests that this model performs relatively poorly, this is confirmed with the decision tree, clearly biased towards a specific error code; and therefore not useful to further validate.

The two other pruned decision trees were deemed to be very insightful to the SME. Relations between error codes and predicted spare parts seemed very plausible based on SME's experience. Four out of five presented error codes in the C1 model have been confirmed predicting a very likely required spare part, given the error code and after a check with the corresponding SPD by the SME. However, error C1T01E27 can have additional root cause aside from the predicted C2118PowerModule. Furthermore, the SME stated that although it is very interesting that system models – among other non-error attributes – is included in these decision trees, for C1 it generally does not matter which specific C1 type is installed in a system and therefore is not expected to be present in related decision trees.

The same general conclusion holds true for the C2 decision tree as well, for 67% of the error codes, the related spare parts seems very possible to be causing the error. The SME was not able to recall the other error codes are corresponding service actions. However, the SME warned that the current C2 view for part cluster 7,9,11 might not be complete as for a specific error (C2T05E09) it was recalled that other root causes do exist apart from those presented in relation to this error. Moreover, for this chain, system model and priority are very useful, as different failures (soft or hard, represented by priority) does require other parts, and not all parts presented in these clusters are used in all Ingenia system models.
Using the corresponding SPD, I further observe that this document – as mentioned during the problem context – does not always mention root causes for failures in terms of parts, but rather provides general service actions. Comparing the errors of this decision tree with the documentation, I also observe that the model provides information regarding three errors that are not included at all in the faultfinding section of the SPD: C2T05E116, C2T15E116, and C2T15E117.

With regard to priority, the discussion with the SME provides a valuable insight. The priority of a customer call can affect the part-purchase behavior of an FSE considerably. Lower priority values indicate hard failures. In order to help the customer and solve the issue as quickly as possible, FSE (unintentionally) can order multiple (different) spare parts to solve the problem during a single visit, when the multiple parts are not necessarily needed. If these parts are not consumed and also not returned by the FSE, or not correctly stated to be returned in Vertica, such behavior can influence the performance of such models negatively.

Lastly, it is observed that the amount of error codes in the C1 ,C2 and C4 decision trees are roughly half of the corresponding chain error codes that are present in the respective part cluster subsets. It is not clear why this is the case; it could be that not all errors are represented in both the train and test sets, given the error distribution per chain dataset (*Fig. 21*), or due to model performance, but also that some errors do not require part replacements. Unfortunately, the SME is not aware of any documentation explaining if there are any chain specific errors that should not result in part replacement, but rather on-site system adjustments.

Feature importance
Feature importance for all four decision trees has been determined and is presented in *Appendix Y, Y.3.* These figures differ a bit from the previously discusses feature importance graphs, which is not necessarily surprising; as the decision trees are not based on the same data set as the XGBoost models, but rather a subset of a potentially predicted part cluster. Such s part cluster subset can contain a lower amount of distinct errors. This is apparent is overall, the four different feature importance plots have many similar errors being with high or low importance as the XGBoost feature importance, but some errors are disappeared from the top x features list, as they are not present in these data subsets.
However, even in these plots, System Model, Priority and Entitlement Type are still considered very important attributes, apart from the C1 – Part Cluster 2 decision tree. This finding corresponds with the insights gathered from the SME validation, that given system C1 parts can be present in all different *Ingenia* system types; hence System Model should not be an important attribute in such a model. In *Appendix Y* we can see that this attribute is not present in the related feature importance graph.

7.3 BUSINESS IMPACT

Based on the created model that predicts from which cluster spare parts or part type are needed, potential business impact and savings can be determined by analyzing the (financial) differences of the single multiple visit cases subsets, as part of the remaining questions of **RQ4**. The section starts with how the financial potentials for the FVF metric have been determined, as part of **SQ4.2**, followed by **SQ4.3** using the FVF metric findings to elaborate on the business impact of the modelling phase.

7.3.1 FVF Business Impact

The potential business impact by improving FVF rates can be determined by analyzing the differences between the subset of single and multiple visit cases consisting of the four different system types in our FVF analysis scope (*Section 1.9*). Conclusions can be drawn as statistical significance between groups has been proven (*Section 5.7*); aiming to 1) determine the average total cost (net part & labor cost) difference between single and multiple visits, 2) how different parts and part cost intervals contribute to this average, and 3) which type of parts dominate in the CM cases looking at most consumed parts.

Regarding the first aspect, average total case costs per year have been calculated for both single and multiple visits, defined as $C_{single,\ year}$ and $C_{multiple,\ year}$; where year ranges from 2013 to 2019. Per year the average difference between determined ($\Delta C_{multiple-\ single,year}$), after which the overall average difference in total cast per case is found, to be: $\Delta C_{multiple-\ single\ cases}$. To iterate, this finding is based on all SWO's from Achieva, Ingenia, Intera and Multiva systems within our project scope, with 1 to 3 visits, limited to 1 to 20 part consumptions, and all different financial classifications types. For further insights, this cost difference per year has been split up between labor ($\Delta C_{labor,\ year}$) and net part cost ($\Delta C_{net\ part\ cost,\ year}$) contribution and into the different products, along with case distribution per year and FVF rates. Exact values seen in *Appendix CC*. From these, we observe:

- $\Delta C_{multiple-\ single,year}$ increases approximately by 10% per year
- This average additional cost per case increase is also consistent over the product models, and $\Delta C_{labor,\ year}$ and $\Delta C_{net\ part\ cost,\ year}$.

Subsequently, we are also interested in what the contribution of different type of cases is, to $\Delta C_{multiple-\ single\ cases}$. Therefore, eight different case cost intervals $C_{SWO\ Interval}$ ($k€$) have been introduced: <=1k€, 1-5k€, 5-10k€, 10-15€, 15-20k€, 20-25k€, 25-30k€, and >30k€. *Appendix CC* shown a tabular overview for each cost interval with corresponding case count per category, further specified into average cost per interval [10] and sum of case cost and number of cases for multiple and single cases separately. Lastly, the differences between multiple and single costs are determined ($\Delta C_{multiple-\ single\ cases,\ interval}$) [11]; where interval ranges from 1 to 8. These values each represent the contribution of the cost interval to the overall determined $\Delta C_{multiple-\ single\ cases}$. *Fig. 26* generally depicts the output (see appendix for the non-confidential figure). From this we observe:

- Cases in cost-category 0-5 k€ for single visit cases contribute 80% to the case count and 30% to the average case cost.
- Cases in cost-category 0-10 k€ for single visit cases contribute 90% to the case count and 55% to the average case cost.
- Cases in cost-category 0-10 k€ for multiple visit cases contribute 80% to the case count and 45% to the average case cost.
- Cases in cost-category 0-15 k€ for multiple visit cases contribute 90% to the case count and 65% to the average case cost

[10] (sum-of-cost-per-category/total-count-of-single-or-multiple-visit-cases)

[11] (sum-of-cost-per-category/total-count-of-cases)$_{multiple\_visits}$ - (sum-of-cost-per-category/total-count-of-cases)$_{single\_visits}$

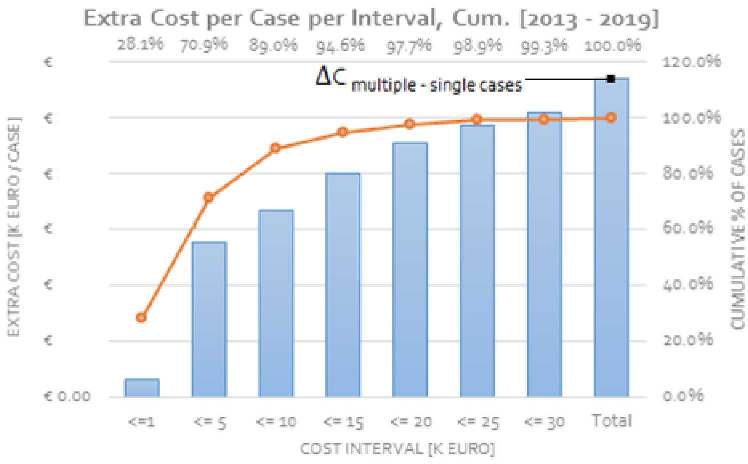


Fig. 26 - Cumulative Contribution to extra cost per interval

Moreover, from the figure it is clear that lower cost intervals contribute far more to the overall cost difference between the two subsets than larger cost intervals. Above calculations are based on the SWO's within scope where quantity consumed parts > 0 and quantity returned >= 0. The same average difference in cost calculation is performed for different combinations of quantity consumed > 0 or =1 and quantity returned >=0, >0, and =0. Further differentiation is made between different financial classifications: all SWO's in scope, SWO's based on warranty entitlement or time & material entitlement. This breakdown is shown in *Table 75 (Appendix CC)*. Main finding here is that entitlement type does not have an impact on difference cost for each combination of quantity consumed and returned, as these values stay relatively constant. Although $\Delta C_{multiple-\ single\ cases}$ is the ideal financial saving per case, if we can predict the required part type for each case, based on logged data, another – 30% lower – value might be more realistic, based on the cost calculation for cases with 1 consumed part and 0 returned parts: $\Delta C_{multiple-\ single\ cases,1con,1ret}$.

Section K (K.3) shows detailed report on the top 25 parts per aforementioned cost interval, along with information regarding its absolute frequency consumed, and % of total parts in top 25, and % of all parts in the cost category. An arbitrary amount of 25 top consumed parts is selected, resulting in coverage values of approximately between 30 and 70%, depending on the service cost interval and single or multiple visit subsets. Moreover, these parts are manually clustered which is subsequently used to create Pareto's to visualize how each cluster of part types contributes to the total amount of consumed parts within a certain interval. From these pareto's it is observed that:

- Lowest cost-category contains low-cost parts and parts that are customer facing
- Cost categories 5-10 and 10-15 k€ are dominated by RF coils (70%), and RF amplifiers (6%)
- Higher cost categories also often contain RF coils.
- Low cost parts such as DCI chassis and connector (magnet, handgrip) are consumed together with RF coils.

Summary graphs for the lower cost categories are shown in the same appendix.

Since, RF Coils are very dominant, a further analysis is done for cases with consumed RF Coils. Section K4 and K5 in *0* present the average new buy prices based on 2013-2019 data, and visualize how different RF coils contribute to different new buy price intervals and how often they are consumed. This is done as RF coils are often more expensive spare parts and are expected to be consumed in cases with higher service cost intervals. The hypothesis for this analysis therefore was that less expensive RF coils (new buy price) would be consumed in lower cost intervals, and more expensive ones in higher intervals.

Lastly, a Technical Part Review (TPR) – analysis is performed for all RF coils observed in the top 25 analysis. This because it is generally assumed that such parts, among others, take multiple visits to replace. To understand why these parts are also included in the single visit population, we look at the TPR process. It is assumed that parts included in TPR, per definition, will require multiple visits to replace. A detailed explanation is provided in *Appendix DD*, but a brief conclusion is that one can understand why RF coils are consumed in single visit cases as roughly half of our observed distinct

12Nc's are not included in TPR. It therefore seems that RF coils do not always require multiple visits to be replaced. These RF Coils are also consumed in multiple visit cases. Knowing upfront that these specific RF Coils are defective, one can at least have some soft savings in terms of labor costs.
Other RF Coils have been included in TPR just towards the end of 2017 or 2018, while being consumed in many cases in prior years. Once a part is added to TPR it seems to be consumed significantly less or not at all for single visit cases. For specific part descriptions and 12Nc's observed in single and multiple visit populations, and relevant TPR information, see *Appendix L*.

7.3.2 Model Business Impact

Above general FVF and corresponding cost analysis can be combined with the model output, into a more specified business impact based on this studies deliverable. We continue with the previous example of the RFC model based on RF Coils – the dominating part type in the scope' SWO data. For each of the 19 classes of the multi-class prediction model, a balanced accuracy and F1 value was determined (*Table 28*). Although balanced accuracy values can be used for calculating the model business impact, I prefer the F1 scores per part type instead, in order to avoid a too optimistic output. Given balanced accuracy and F1- scores per part type (class) predicted, and the calculated savings based on 2013-2019 data and case count per class (*Section 7.3.1*), potential model-based savings can be determined via the following - where i equals the different RF part types (classes) ranging 1 to 19:

$$Potential\ (ideal) savings = \sum_{i=1}^{n} (F1_i * \ CaseCount_i) \cdot \Delta C_{multiple-\ single\ cases}$$

$$Potential\ (realistic) savings = \sum_{i=1}^{n} (F1_i * \ CaseCount_i) \cdot \Delta C_{multiple-\ single\ cases, 1con, 1ret}$$

$$Potential\ (soft) savings = \sum_{i=1}^{n} (F1_i * \ CaseCount_i) \cdot \Delta C_{labor}$$

Potential soft savings represent savings for which a customer visit is reduced due to correct part type prediction and only labor cost is decreased. Potential ideal savings can be calculated based on $\Delta C_{multiple-\ single\ cases}$ (total amount that in theory could be saved in total net part cost and labor cost), but a more realistic saving can be obtained if correct part type is predicted and a customer case is solved with one part replacement and no parts to be returned (based on $\Delta C_{multiple-\ single\ cases, 1con, 1ret}$).

7.4 Market Basket Analysis

The MBA script is usable for results based on different aggregates. Although this was originally indented for part co-dependency analysis, this is set up in such a way that it can be used for mining 1) on the whole dataset, 2) subset for specific parts, and 3) on part cluster level, as a complementary tool to the predicted part type. This section first presents the results and relevant output for all single visit cases, followed by a part-specific example, and closing with part cluster associations.

7.4.1 Item Rule Mining – All Single Visit Cases

The '*R*' – package *arulesViz* is used as a core package for this analysis; which automatically loads other required packages like *arules* to handle and mine the associations (Hahsler et al., 2010). For readability purposes and the potential of extracting too detailed and extensive rules, the number of items per rule where limited upfront; with a minimum number of items per rule of one, and a maximum of five. Additionally, given the large number of spare parts, low support and confidence values were expected, hence the upfront cutoff value of .001. The spare part transaction data for single visits, indeed contains 17 136 transactions, with 2 895 distinct items. Summary statistics of the data set show a rather sparse set with a density just above 0.05% and an average transactions containing less than three items.
Association rules are mined next; for which a more detailed general script is included in *Appendix V*:

```
association.rules <- apriori (trallcases, parameter =
list(sup = 0.001, conf = 0.001, minlen = 1, maxlen = 5,  target='rules'))
summary(association.rules)
```

Testing for Chi-squared and Fisher's Exact Test rules significance, the number of discovered rules reduces from 5460 to 1310. Such large amount of rules can best be visualized in a scatter plot with two interest measures on the axes, instead of manually checking individual rule as this is not a viable option. *Fig. 27* depicts the scatter plot using support and confidence value on the axes with a third measure 'lift' as the color or gray level of the data points. We see that rules with high lift values tend to have relatively low support. The most interesting results reside on the confidence/support border. However, for further details and an interactive two-key plot is created. The two-key plot uses determined support and confidence values on the axes, while data point' color indicates the "order"; corresponding to the number of items contained in a rule. Note that the maximum is "order 5", as the maximum rule length was capped at five items previously. Order and support seem to have a strong inverse relationship, as this observation is also confirmed in the study of Seno & Karypis (2005). The interactive aspect added, provides the user with features to: 1) Inspect individual rules upon selecting, 2) Inspect rule sets by selected plot region, 3) Zooming on a selected plot region, and 4) Filtering rules based on cutoff points.

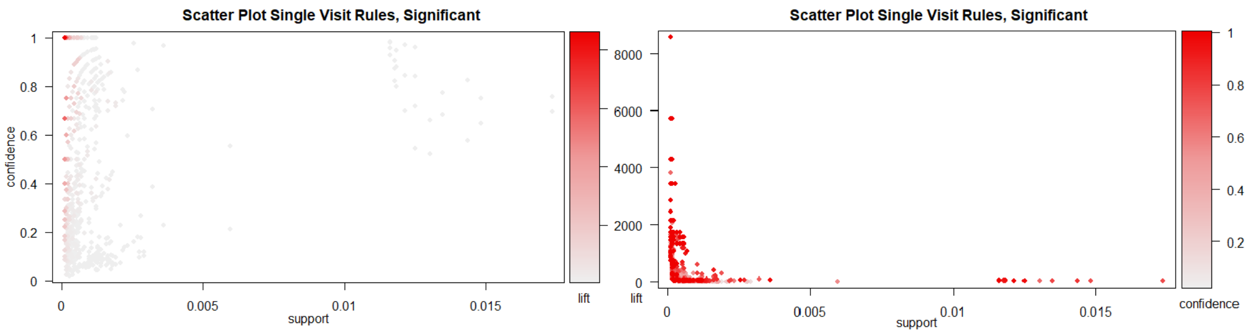


Fig. 27 - Significant Rules, All Single Visit Cases, Scatter Plots

Fig. 29 shows the two-key plot for all observed statistically significant rules based on all single visit cases, along with a random user-selected example of specific rule inspection. The output presents one statistically significant 'order 2' rule where part consumption of a '*C78x Power Module*' is followed by a '*FUSE 400A*' part consumption in single visit CM cases with a confidence of 39% and lift of 85, based on a frequency of 55.

7.4.2 Item Rule Mining – Specific Spare Part

Above plots are not suitable when one wants to mine for a specific spare part (type) or cluster; although they can provide great insight in large number of rules, relatively low amount of associations are best inspected otherwise. Although on this aggregate individual rules (or top x) can easily be checked manually, parallel coordinates plots are also a great addition. Such visualizations plot multidimensional data separately on an x-axis with a shared y-axis. A line connecting values, in this case spare parts, for each dimension, represents each data point.

Since, *Section 7.3* revealed that RF coils should be the focus and be included in the funnel we randomly selected a related coil to show the MBA analysis for this item. The script allows for a specific part description or partial match based on user input to get relevant significant rules. The subset value therefore is "*NVC COIL-1.5*". Specifically, we are interested to know which other spare parts tend to be consumed if one replaces a *NVC COIL-1.5*. Hence, in rule format this part functions as an antecedent: $NVC\ COIL\ 1.5 \Rightarrow Y$, representing the left hand side (LHS). The script allows the part to be identified as right hand side (RHS) as well, if interested.

Fig. 28 shows the parallel coordinates plot for the top 10 rules out of 42 – based on confidence – mined for the general associations mentioned above, taking into account Chi-squared and Fisher's Exact Test rules significance. The width of the arrows represent support while the intensity of the arrow'

color represents the confidence value. As the number of line crossovers can increase with a larger rule set, the items on the y-axis are re-ordered to minimize the number of crossovers.
The figure shows that either the part in question can be the only item in the rule (and therefore the only consumed item for repairs) or other subsequent parts might be required to solve the problem at hand. Given the fixed lhs value, all arrows start from the same spare part.

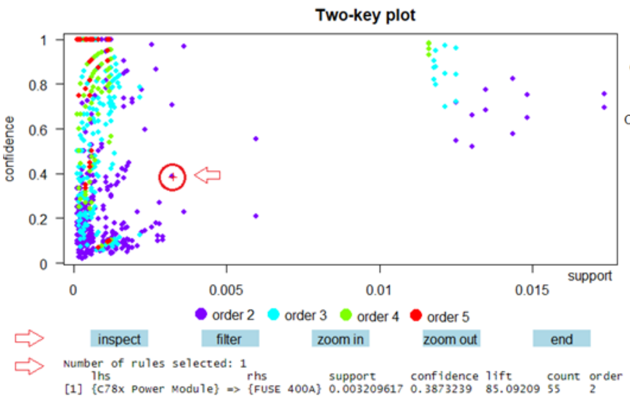


Fig. 29 - Two-Key Plot sign. SingleVisit Rules, part selection

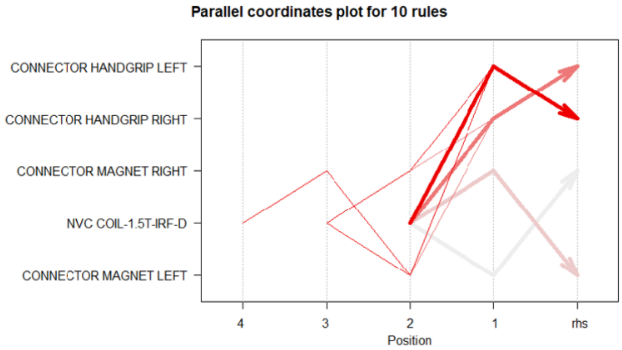


Fig. 28 - Parallel Coordinates Plots

For further assistance, the '*igraph*' package is used rendering an interactive graph-based visualization, in which vertices are annotated with item labels representing the different items, while edges are arrows pointing from items to rule vertices indicating LHS items and arrows from a rule to an item indicating RHS items. Moreover, aforementioned interest items are added, as depicted in *Fig. 30*.

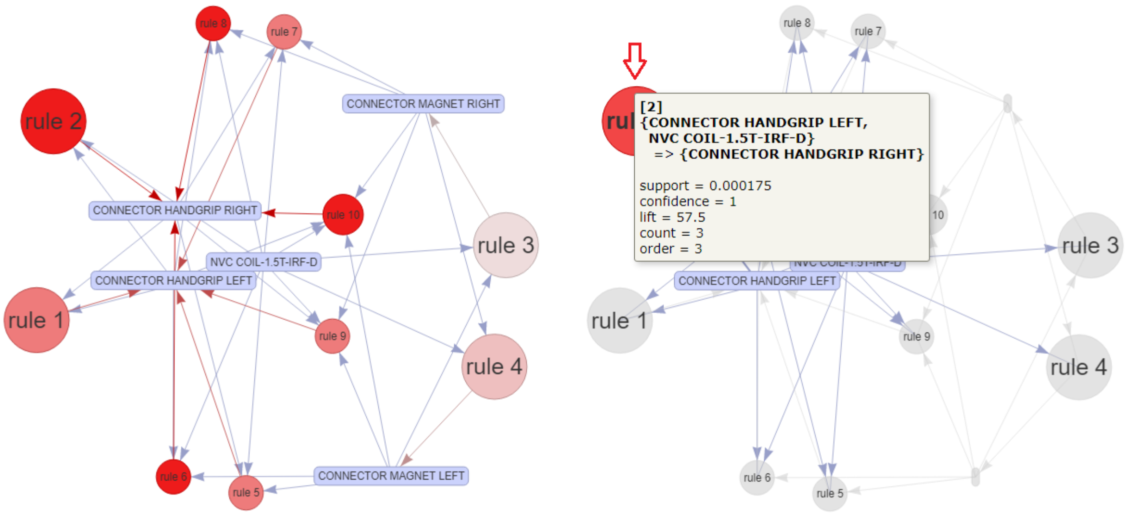


Fig. 30 - Interactive Part-specific Association Graph

From above MBA analysis, we can conclude that cases do not always have exact one part replacement for each SWO, since 1310 significant rules are discovered that range from an order of two up to an order of five. Hence, there are co-dependencies between some of the 2.8k distinct (spare) parts in the final data set of single visit cases, and multiple parts could be required for solving customer and MR system problems. The discussed data visualizations can therefore be created for specific spare parts as yet another complementary data source for RSE's and FSE's as an extend of the RCA models, to find out what other parts might be required if the predicted part types from *Section 6*, do not solve the problem. The predicted part might be correct, but not necessarily, enough for a specific issue; as the example of the NVC Coil in this section shows that Connector Handgrip parts are likely to be used as well during an NVC Coil replacement. Such item rule mining results, based on RCA prediction model' output, therefore can aid in avoiding trial and error of part usage, and saving multiple visits and potentially soft and hard savings.

8 CONCLUSION

This final section of the report addresses the conclusion of this study in which I have presented a data mining tool for the implementation of data-driven corrective maintenance for four different MR chains; as a solution to the defined problem statement and gap analysis. Based on answers to the research question – discussed throughout *sections 4* to *7*, I present whether the business problem is solved, along with managerial implications, and discussion.

8.1 RESEARCH CONCLUSIONS

The research questions formed the backbone of the first visit fix project, derived from the problem statement, project goal and gap analysis (*Section 1*). Firstly, an explorative research has been performed regarding the current process of MR maintenance, identifying possibilities for support within the AS-is situation with this work. The described troubleshooting process shows potential for a tool to be used by RSE's when escalating a customer call to a FSE, creating a SWO and initiating an on-site visit for maintenance (*Section 4*). If potential service actions can be provided to the FSE based on a data-driven RCA tool, this can beneficial for handling the issue and solving the complaint in as few visit as possible. Aside from used multi-class classification models to this extend, visualized decision trees can be helpful for FSE on-site with the usual ad-hoc decision making. This has been supported with the identified cause and effect tree in the problem statement, needing a solution to deal with lack of spare parts at the moment of required on-site repair (due to lack of R/FSE diagnosis) and lack of problem statement at customer call (*Section 1.5*). Potential impact of proposed solution on the troubleshooting process is further elaborated in *Section 8.2*.

Requiring input for a data-driven tool was fulfilled with the enhanced accessibility of machine data with the Vertica database; with the majority of MR systems in scope uploading raw machine log files centrally (*Section 4.3*). Leading towards the completion of preparation of relevant data to be used for the classifiers, it is discovered that various failure modes are not independent due to system design, as previously assumed by SME's. Surprising insights are provided into potential dependence of error codes across different MR chains, with highly correlated and significant p-values (*Section 5.8*). Potential dependence was further studied with error sequence analysis, using frequent Pattern – Grown analysis, also as part of error clustering (*Section 5.9*).

Based on the cleaned single visit *Ingenia* data set, XGBoost, Random Forest and Support Vector Machines modeling techniques were studied, starting a three-step RCA solution design, in which first the most likely part cluster is determined via multi-classification classifiers, along with variations based on the analysis of data re-sampling methods where required. Part clusters corresponding to the model' output classes have been defined using the Jaro Wrinkler approximate string matching technique, based on hundreds of available parts, resulting in 15 part clusters. After a grid search method to determine the optimal parameters, the XGBoost models per MR Chain have been selected as the most effective and promising, thanks to its higher recall values (ranging from .52 – .71), acceptable to good F1 values (.55 – .76), very high AUC (.71 – .80), and substantial agreement Kappa results (.54-.69), outperforming the other considerations (*Section 6 – 7*).

Given the predicted part cluster based on the cluster with highest probability, as a next step, the subset of cases is taken with part consumptions of the part cluster, resulting in a more specific part prediction. For this RFC perform just as well as XGBoost, providing a similar multi-class classification output as the previous model, but for a concrete part type. The example in *Section 7.2* shows that RFC is able to predict a specific part, when focusing on the subset of data based on cases from the aforementioned predicted part cluster. F1 values range from .29 to 1.00; where 14 out of 19 specific parts (classes) score a F1-value of > .60.

To provide supportive visualizations of decision rules, instead of visualizing a RFC tree out of ntree=1000, a single decision tree algorithm is used for this purpose to show the relation between chain' error codes, system and SWO attributes such as priority and system model, along with specific spare parts. Pruned decision trees performance varies as the C1 and C2 example (*Section 7.2.1*) score well on the Kappa metric (.63 and.67 respectively – substantial agreement), with even higher very good F1 values (.84 and .73 respectively). The C4 example performs satisfactory as well (Kappa = .58, F1 = .65), but the C3 example scores even lower and upon further inspection this model is very biased towards a particular error code and seems to over fit. C1 and C2 trees have been validated with a SME, although not all errors could be discussed, these trees deemed to be a very good start in visualizing the decision rules; not only regarding a part and error relation, but also very useful to include other SWO and system attributes. Unfortunately, the C4 tree could not be validated with the specific SME, therefore it was aimed to validate with a corresponding SPD (*Section 7.2.1*).

Lastly, MBA can assist in generating interactive visualizations or separate statistical significant association rules to show part co-dependency (based on Chi-Squared and Fisher's Exact Test, $p < .001$) for other spare parts that might be useful in addition to the predicted part – given historic repairs, in case other parts might still be required for the repair (*Section 7.4*).

Based on the above findings, we can reflect on the gap analysis (*Section 1, Table 3*). The proposed three-phase solution uses the complete logged CM knowledge for *Ingenia* systems, based on the scope, to provide RCA results for on-site service actions. Although, other MR systems have been mentioned in this study, these where not included in the RCA but rather in the FVF analysis; however, the underlying methodology is applicable to other MR systems as well.

The solution provides an additional source for determining and motivating CM part replacements. CM service actions were either implicit, based on engineer experience, or SPD's were utilized to determine general service actions and tests to find the problem's root cause. This is made explicit, but given potential future research, models can still be improved (*Section 8.3* and *8.5*). However, the third aspect of the gap analysis regarding analytically supporting spare part ordering for FSE's, has not been fully achieved. Although the resulted trees have been validated with a corresponding test set of the data, the visualized decision rules could not all be validated with FSE's; both due to required and available subject matter expertise along with time constraints of the project, and improvements required specifically for the C3 tree.

Lastly, the FVF field service metric is systemically analyzed, not only with regard to FVF rates per system or market, but also focus on business funnel, and potential soft and hard savings in general, additionally applied to proposed solution (*Section 7.3*, along with *Appendix K, Appendix L*, and *Appendix CC, Appendix DD*). This is discussed in more detail in the next section.

8.2 MANAGERIAL IMPLICATIONS

With this research, Philips Healthcare can start analyzing all historic and incoming CM activities and calls, with required part replacement, with the designed methodology. The analyzed FVF metrics shows that some of the parts usage and therefore labor cost also, might be redundant. The FVF analysis concludes that there is the potential to have an ideal financial saving, on average, per case equal to $\Delta C_{multiple-\ single\ cases}$, and a 30% lower value equal to $\Delta C_{multiple-\ single\ cases,1con,1ret}$ might be more realistic to achieve, taking into account cases with one consumed part and none returned. Looking at average total service (net part + labor) cost made across the years, we observe a difference of approximately 15% between single and multiple visit cases for the four (Achieva, Ingenia, Intera & Multiva) CM cases from the scope of this project, while this number drops to 12% only looking at the Ingenia subset.

To bridge the gap, the solution design contributes to these savings and improving CM case performance and can fit the company's future strategy. In terms of presented part cluster and part type prediction models, along with association rule mining interactive visualizations, these can ideally be part of Remote Service Workspot or complementary as additional data sources to be used by RSE's to provide concrete service actions in terms of required spare parts, hence repair actions to undertake.
In terms of presented decision rules visualized, these contribute by aiding FSE's on the field with additional decision-making information of which spare parts are required given the MR failure mode and context. The corresponding methodology can also be used to create different decision trees to complete missing or incomplete faultfinding sections in SPD's.
Implementing proposed solution, should not have an impact on the current troubleshooting process, identified in *Section 4*, and presented in detail in *Appendix A*, in terms of changes; but rather help or extend current tasks. Specifically, with regard to 'Remote In-Depth Analysis & Fixes' of an RSE, additions to the BPMN model occur at the 'Create SWO'-task, and regarding "On-Site Repair" to the FSE task 'Execute relevant test & repair actions', as visualized in *Fig. 31* and *Fig. 32*.

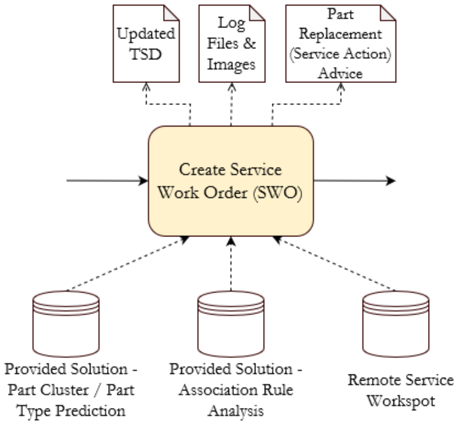


Fig. 31 - Changes Process - SWO

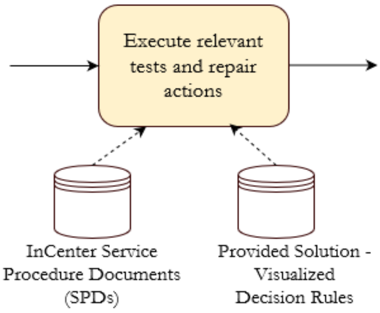


Fig. 32 - Changes Process - On-site Repair

Assuming the average $\Delta C_{multiple-\ single\ cases,1con,1ret}$ per case is achieved, by predicting the correct spare part and replacing it in the first visit, each successful CM case would contribute by .11% or .03% to the FVF total cost difference for the *Ingenia* or four MR system scope, respectively.

With regard to future FVF projects, the following additional conclusions were drawn during this study' FVF analysis:

- $\Delta C_{multiple-\ single,year}$ increases approximately by 10% per year. This average additional cost per case increase is also consistent over the product models, and $\Delta C_{labor,\ year}$ and $\Delta C_{net\ part\ cost,\ \ year}$ (*Appendix AA*).
- Although FVF rates are improving in general over all studied MR products, these rates still vary significantly across different markets (*Appendix K* – K.1 and K.2 respectively). Performing the same FVF analysis for different markets, may provide insights regarding market specific (part replacement) behavior. As a side note, market specific analysis is also interesting regarding logged errors, to find area specific frequent issues. Such studies can also contribute to understanding why market is found to be a very important feature in data-driven CM prediction models (*Section 7.1*).
- Lowest cost-category contains low-cost parts and parts that are customer facing. Cost categories 5-10 and 10-15 k€ are dominated by RF coils (70%), and RF amplifiers (6%). Higher cost categories also often contain RF coils. Low cost parts such as DCI chassis and connector (magnet, handgrip) are consumed together with RF coils.
- Given the previous, and the service cost contribution analysis to $\Delta C_{multiple-\ single\ cases}$: do not only focus on the most expensive parts, when creating models or analyzing how to further improve FVF.
- Lastly, the discussed RCA solution has shown that RF Coils related cases are predicted very well. As these part types are also found to be dominating in the FVF analysis, such models can aid in identifying when these parts are required for customer calls and to be solved in one visit. Especially, as TPR analysis suggests that these parts do not necessarily require multiple visits per definition, as previously assumed.

8.3 Limitations

Although we have aimed to predict a part cluster or even specific spare parts based on machine logs and error occurrences based on consumed parts of prior corrective maintenance cases, it should be noted that only parts and errors are analyzed that have been recorded. Many different errors can theoretically be logged by faults that occur in different subsystems (*Appendix G*) but this analysis is based on those that have actually been registered within a predefined timeframe (*Section 5.5*). The same holds for consumed spare parts, the researcher is aware of the many different distinct 12Nc's, but only those present in the subset of single visit cases are taken into account. If part consumption or error logging has not been properly recorded (as deemed possible from *Appendix P*), this will have gone unnoticed in previous sections. Assumed that the different MR chains and log data thereof are identifiable causes of the SWO's, the researcher is unaware of other (external) factors or sporadic part failures having contributed to SWO's.

Given project time constraints and the very computationally intensive task of analyzing all logged errors within a two week period of a case open date, it was decided to focus on whether a certain error has been logged or not. Hence, an array of errors occurred in a MR system, corresponding to the one up for maintenance in a SWO; consist of errors that have occurred at least once (*Section 5.5*). As a result, error importance and weight could not be determined at this stage, by looking at how often certain failures have been registered and therefore might me more important than others.

Moreover, all occurred errors have been taken into account, as domain knowledge was missing in identifying all failure modes that actually require part replacements. Some failures might just need on-site system or part adjustment instead of resulting in a part replacement. This information also was not available for all errors in scope in different data sources, such as service procedure documents.

Lastly, the data-driven aspect of the research is currently limited to historical maintenance and machine, with the potential in using newly available (remote) data from June 22th, 2019 onwards. Ideally, the tool would be based on up-to-date information, including regularly newly introduced 12Nc's, system models, and potential for contribute to error and part class imbalance issues of the current data set.

8.4 Scientific Relevance

Along with the exploratory nature of the project for Philips, it was also exploratory with regard to scientific knowledge. The RCA domain is dominated by knowledge-driven methods such as 5 why's, fishbone diagram or even semi-quantitative approached with use of machine learning methods (*Section 2*), again based on FSE's expertise. Which does raise questions regarding RCA completeness, accuracy and bias, given the various spare- and interacting parts, and failure modes of a MRI system. This study has successfully shown that root causes can be identified to MR failure modes, although many future research possibilities lie ahead to achieve better performing prediction models and decision trees as performance for different optimized models does vary.

Based on generated error log files, and service work orders data – both regarding logged consumed parts and customer' system information - the study provides a methodology to assist in deriving meaningful failure associations in this domain. A completely data-driven RCA to assist both remote service engineers as well as field service engineers for on-site maintenance. Based on the deficiencies discussed in this theoretical background and the fact that the application of data exploration approaches is under-reported in the maintenance and imaging health devices literature, the need for a data-oriented approach for root cause analysis was imperative; for which this study serves as a proof of concept.

However, to further improve scientific contribution, additional upcoming classification techniques such as 'LightGBM' and 'CatBoost' can be applied to study their performance for RCA and improvement potential compared to more known methods used in this project. Additionally, per the researcher's knowledge, market basket analysis based on the Apriori algorithm has been used to analyze spare part consumption and use as complementary tool for statistically significant associations for specific part usage. A technique often used in marketing, bioinformatics, education or sales, not implemented for (imaging devices) CM maintenance (Chen et al., 2004; Kaur & Kang, 2016; Weng & Liu, 2004).

8.5 FUTURE RESEARCH

To further strengthen the study' results, various directions for future research are provided; of which some reflect directly on the study limitations.

First, although all SWO data has been used given the project scope and data cleaning restrictions over an almost seven year period, this was not the case actual error data for potential root cause. Mainly because time constraints, and already focusing on the four MR chains at hand; which would result in enough complexity, an important assumption is made indirectly with this decision. Only looking at error data from four chains, means that one assumes that a SWO root cause lies within one of these chains. SWO data nor other data sources elaborate on what exactly was the problem of a customer complaint. Hence, for additional research other chains should be included or a methodology is needed to identify cases with spare part consumptions that under no circumstance can be related to one of the chains and therefore excluded cases from the data set(s). This challenge was also apparent given the model assessment phase where part clusters were included and predicted in test sets that likely cannot be the root cause of certain chains, based on various SME discussions.

Regarding the errors themselves, only errors that have occurred in a two week period before each case open date have been taken into account. This was decided with the project team, but still remains an arbitrary number. A longer period, hence more potential errors, might provide other model results. Another way of making sure more of the potential errors are included in the data set(s), is to expand the presented methodology to other MR systems, besides the current scope of *Ingenia* systems. Or, include error / fault descriptions – provided by customers or engineers – as this study focused on (predefined) logged numerical or string based error codes. Fault descriptions require intensive text mining techniques and dealing with multiple language-related issues as there is no standardized way of input. The latter has explicitly been decided on not to include in this study at all by the project team.

9 REFERENCES

AberdeenGroup. (2013). *Fixing First-Time Fix: Repairing Field Service Efficiency to Enhance Customer Returns*.

Adhikary, D. Das, Bose, G. K., Bose, D., & Mitra, S. (2014). Multi criteria FMECA for coal-fired thermal power plants using COPRAS-G. *International Journal of Quality and Reliability Management*, *31*(5), 601–614. https://doi.org/10.1108/IJQRM-04-2013-0068

Almonte, C., & Kibria, B. M. G. (2004). On some classical, bootstrap and transformation confidence intervals for estimating the mean of an asymmetrical population. *Model Assisted Statistics and Applications 4*, *4*, 91–104. https://doi.org/10.3233/MAS-2009-0110

Alvarez, S. A. (2003). *Chi-squared computation for association rules: preliminary results*.

Andres, A. M., & Tejedor, I. H. (1995). Is Fisher's exact test very conservative? *Computational Statistics & Data Analysis*, *19*(5), 579–591. https://doi.org/https://doi.org/10.1016/0167-9473(94)00013-9

Azevedo, P. J., & Jorge, A. M. (2007). Comparing Rule Measures for Predictive Association Rules. *Proceedings of ECML 2007 - 18th European Conference on Machine Learning*, 17–21. Retrieved from 10.1007/978-3-540-74958-5_47

Baghban, A. A., Younespour, S., Jambarsang, S., Yousefi, M., Zayeri, F., & Jalilian, F. A. (2013). How to test normality distribution for a variable: a real example and a simulation study. *Journal of Paramedical Sciences (JPS)*, *4*(1), 73–77. Retrieved from https://pdfs.semanticscholar.org/0a73/a2b260a477bf49641ee89597f3a37098a37d.pdf

Benjamin, S. J., Marathamuthu, M. S., & Murugaiah, U. (2015). The use of 5-WHYs technique to eliminate OEE's speed loss in a manufacturing firm. *Journal of Qualitative Maintenance Engineering*, *21*, 419–435.

Benslimane, D., Damiani, E., Grosky, W. I., & Hameurlain, A. (2017). Database and Expert Systems Applications. In A. Sheth & R. R. Wagner (Eds.), *28th International Conference, DEXA*. Springer.

Berry, M. J. A., & Linoff, G. S. (2004). *Data Mining Techniques: for Marketing, Sales and Customer Relationship Management* (2nd Editio). Hungry Minds Inc.

Blagus, R., & Lusa, L. (2013). SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, *14*(106). https://doi.org/https://doi.org/10.1186/1471-2105-14-106

Bošnjak, Z., Grljević, O., & Bošnjak, S. (2009). CRISP-DM as a framework for discoverin knowledge in small and medium sized enterprises' data. *Proceedings - 2009 5th International Symposium on Applied Computational Intelligence and Informatics, SACI 2009*, 509–514. Retrieved from http://doi.org/10.1109/SACI.2009.5136302

Bousdekis, A., Magoutas, B., Apostolou, D., & Mentzas, G. (2015). A proactive decision making framework for condition-based maintenance. *Industrial Management & Data Systems*, *115*(7), 1225–1250. https://doi.org/10.1108/IMDS-03-2015-0071

Bower, K. M. (2003). When To Use Fisher's Exact Test? *ASQ Six Sigma Forum Magazine*, *2*(4), 1–5.

Braglia, M. (2000). MAFMA: Multi-attribute failure mode analysis. *International Journal of Quality and Reliability Management*, *17*(9), 1017–1033. https://doi.org/10.1108/02656710010353885

Braglia, M., Frosolini, M., & Montanari, R. (2003). Fuzzy criticality assessment model for failure modes and effects analysis. *International Journal of Quality and Reliability Management*, *20*(4), 503–524. https://doi.org/10.1108/02656710310468687

Brijs, T., Vanhoof, K., & Wets, G. (2003). Defining interestingness for association rule. *International Journal Information Theories & Applications*, *10*, 370–375.

Brodersen, K. H., Ong, C. S., Stephan, K. E., & Buhmann, J. M. (2010). The balanced accuracy and its posterior distribution. *2010 International Conference on Pattern Recognition*, 3121–3124. https://doi.org/10.1109/ICPR.2010.764

Buchmann, A. K. F. F. (2014). Modeling and execution of event stream processing in business processes. *Information Systems*, *46*, 140–156.

Bulmer, M. G. (1979). *Principles of Statistics (Dover Books on Mathematics)* (New Editio). Dover Publications Inc.

Burbrink, F. T., & Pyron, R. A. (2008). The Taming of the Skew: Estimating Proper Confidence Intervals for Divergence Dates. *Systematic Biology*, *57*(2), 317–328. https://doi.org/10.1080/10635150802040605

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chemweno, P., Morag, I., Sheikhalishahi, M., Pintelon, L., Muchiri, P., & Wakiru, J. (2016). Development of a novel methodology for root cause analysis and selection of maintenance strategy for a thermal power plant: A data exploration approach. *Engineering Failure Analysis*, *66*, 19–34. https://doi.org/https://doi.org/10.1016/j.engfailanal.2016.04.001

Chemweno, P., Pintelon, L., Van Horenbeek, A., & Muchiri, P. (2015). Development of a risk assessment selection methodology for asset maintenance decision making: An analytic network process (ANP) approach. *International Journal of Production Economics*, *170*, 663–676. https://doi.org/10.1016/j.ijpe.2015.03.017

Chen, C., Liaw, A., & Breiman, L. (2004). *Using random forest to learn imbalanced data. 110*(24), 1–12.

Chen, P. Y., & Popovich, P. M. (2002). *Correlation - Parametric and Nonparametric Measures* (C. D. Laughton, V. Novak, & D. E. Exelsen, Eds.). Thousand Oaks, CA: SAGE Publications.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *In Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 785–794.

Chen, Y. L., Tang, K., Shen, R. J., & Hu, Y. H. (2004). Market basket analysis in a multiple store environment. *Decision Support Systems*.

Chen, Y. P. (2011). Do the Chi-Square Test and Fisher's Exact Test Agree in Determining Extreme for 2 × 2 Tables? *The American Statistician*, *65*(4), 239–245. https://doi.org/https://doi.org/10.1198/tas.2011.10115

Cipollini, F., Oneto, L., Coraddu, A., Murphy, A. J., & Anguita, D. (2018). Condition-Based Maintenance of Naval Propulsion Systems with supervised Data Analysis. *Ocean Engineering*, *149*, 268–278.

Coe, R. (2002). It's the Effect Size, Stupid - What effect size is and why it is important. *Annual Conference of the British Educational Research Association*. Retrieved from https://www.leeds.ac.uk/educol/documents/00002182.htm

Cohen, W., Ravikumar, P., & Fienberg, F. (2003). A comparison of string distance metrics for name-matching tasks. *In Proceedings of the IJCAJ-03 Workshop on Information Integration Onthe Web*, 73–78. Retrieved from http://www.isi.edu/integration/workshops/ijcai03/proceedings.htm.

DAMA. (2013). *THE SIX PRIMARY DIMENSIONS FOR DATA QUALITY ASSESSMENT - Defining Data Quality Dimensions*.

Das, S., Datta, S., & Chaudhuri, B. B. (2018). Handling data irregularities in classification: Foundations, trends, and future challenges. *Pattern Recognition*, *81*, 674–693. https://doi.org/https://doi.org/10.1016/j.patcog.2018.03.008

De Sanctis, I., Paciarotti, C., & Di Giovine, O. (2016). Integration between RCM and RAM: a case study. *International Journal of Quality and Reliability Management*, *33*(6), 852–880. https://doi.org/10.1108/IJQRM-02-2015-0026

DiCiccio, T. J., & Efron, B. (1996). Bootstrap Confidence Intervals. *Statistical Science*, *11*(9), 189–228. Retrieved from https://projecteuclid.org/download/pdf_1/euclid.ss/1032280214%0D

Dorsch, J. J., Yasin, M. M., & Czuchry, A. J. (1997). Application of root cause analysis in a service delivery operational environment: A framework for implementation. *International Journal of Service Industry Management*, *8*(4), 268–289. https://doi.org/10.1108/09564239710174372

Dreßler, K., & Ngonga Ngomo, A. C. (2017). On the efficient execution of bounded Jaro-Winkler distances. *Semantic Web 8*, 185–196. https://doi.org/10.3233/SW-150209

Excel Forum. (2012). Can you sort data alphabetically in one particular cell? Retrieved August 21, 2019, from https://www.excelforum.com/excel-general/833503-can-you-sort-data-alphabetically-in-one-particular-cell.html

Fortuin, L., & Martin, H. (1999). Control of service parts. *International Journal of Operations and Production Management*, *19*(9), 950–971. https://doi.org/10.1108/01443579910280287

Garcia-Lacencina, P. J., Sanch-Gomez, J. L., & Figueiras-Vidal, A. R. (2010). Pattern classification with missing data: a review. *Neural Computing and Applications*, *19*(2), 263–282. https://doi.org/https://doi.org/10.1007/s00521-009-0295-6

Ghasemi, A., & Zahediasl, S. (2012). Normality tests for statistical analysis: a guide for non-statisticians. *International Journal of Endocrinology and Metabolism*, *10*(2), 486–480. https://doi.org/10.5812/ijem.3505

Grabot, B. (2018). Rule Mining in Maintenance: Analysing large knowledge bases. *Computers & Industrial Engineering*. https://doi.org/https://doi.org/10.1016/j.cie.2018.11.011

Grove, R., & Faytong, J. (2012). Chapter 4 - Machine Learning and Event-Based Software Testing: Classifiers for Identifying Infeasible GUI Event Sequences. *Advances in Computers*, *86*, 109–135. https://doi.org/https://doi.org/10.1016/B978-0-12-396535-6.00004-1

Gu, J. (2013). *Proactive and Efficient Spare Parts Inventory Management Policies Considering Reliability Issues* (University of Windsor). Retrieved from http://scholar.uwindsor.ca/etd/4888/%5Cnpapers3://publication/uuid/E5CE418A-49E3-41B8-8561-69FB6E51439A

Gutschi, C., Furian, N., Suschnigg, J., Neubacher, D., & Voessner, S. (2019). Log-based predictive maintenance in discrete parts manufacturing. *12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018*, *79*(March), 528–533. https://doi.org/10.1016/j.procir.2019.02.098

Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2014). *Multivariate Data Analysis* (7th Editio). Essex: Pearson Education Limited.

Halawani, S. M., & Ahmad, A. (2011). A costing analysis for decision making grid model in failure-based maintenance. *Adv. Decis. Sci.*

Han, J., Jian, P., Yiwen, Y., & Runying, M. (2004). Mining Frequent Patterns without Candidate Generation - A Frequent Pattern Tree Approach. *Data Mining and Knowledge Discovery*, *8*(1), 53–87.

He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

Hedges, L. V., & Olkin, I. (1985). Statistical Methods in Meta-Analysis. *Journal of Educational Statistics*, *20*(1). https://doi.org/10.2307/1164953

Hinde, G. (2019). Philips Company Profile. Retrieved May 24, 2019, from Intranet - Company Presentation website: https://intranet.philips.com/Pages/Company-profile.aspx

Hinkle, D. E., Wiersma, W., & Jurs, S. G. (2003). *Applied Statistics for the Behavioral Sciences* (5th Editio). Boston, MA: Houghton Mifflin.

Ho, J. (2019). Bootstrap Confidence Intervals. Retrieved July 24, 2019, from https://cran.r-project.org/web/packages/dabestr/vignettes/bootstrap-confidence-intervals.html

Hockel, D., & Hamilton, T. (2011). Understanding total cost of ownership. *Healthcare Purchasing News*, *35*(9), 1–14.

Hsu, C., Chang, C., & Lin, C. (2016). *Practical Guide to Support Vector Classi cation*. Taiwan.

Huiguo, Z., Rui, K., & Pecht, M. (2009). A hybrid prognostics and health management approach for condition-based maintenance. *IEEE International Conference On Industrial Engineering and Engineering Management*, 1165–1169. Retrieved from https://doi.org/10.1109/IEEM.2009.5372976

Hulse, J. V., Khoshgoftaar, T. M., & Napolitano, A. (2007). Experimental perspectives on learning from imbalanced data. *Proceedings of the 24th International Conference on Machine Learning*, 935–942. Corvallis, Oregon: Oregon State University.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *Statistics An Introduction to Statistical Learning* (1st Editio). Springer-Verlag New York.

Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). aA review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 148301510.

Jaro, M. A. (1989). Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, *84*(406), 414–420. https://doi.org/10.1080/01621459.1989.10478785

Jawlik, A. A. (2016). *Statistics from A to Z: Confusing Concepts Clarified*. https://doi.org/10.1002/9781119272021

Kaufmann, M. (2011). *CUDA Application Design and Development*. https://doi.org/https://doi.org/10.1016/B978-0-12-388426-8.00011-2

Kaur, M., & Kang, S. (2016). Market Basket Analysis: Identify the changing trends of market data using association rule mining. *Procedia Computer Science - International Conference on Computational Modeling and Security (CMS 2016)*, 78–85.

Kikuchi, J. (2016). Application of Market Basket Analysis for the Visualization of Transaction Data Based on Human Lifestyle and Spectroscopig Measuremetns. *Analytica Chemistry*. https://doi.org/10.1021/acs.analchem.5b04182

Kim, H. Y. (2013). Statistical notes for clinical researchers: assessing normal distribution (2) using skewness and kurtosis. *Restorative Dentistry & Endodontics*, *38*(1), 52–54. https://doi.org/10.5395/rde.2013.38.1.52

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, 1137–1143.

Korst, J., Pronk, V., Barbieri, M., & Consoli, S. (2019). Introduction to Classification Algorithms and their Performance Analysis using Medical Examples. In *Data Science for Healthcare: Methodologies and Applications*. https://doi.org/10.1007/978-3-030-05249-2_2

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling.* (5th Editio). https://doi.org/10.1007/978-1-4614-6849-3

Kursunluoglu, E. (2014). Customer Service Effects on Customer Satisfaction and Customer Loyalty : A Field Research in Shopping Centers in Izmir City - Turkey. *International Journal of Business and Social Science*, *2*(17), 52–59.

Lakens, D. (2013). Calculating and reporting effect sizes to facilitate cumulative science: a practical primer for t-tests and ANOVAs. *Frontiers in Psychology*, *4*, 863. https://doi.org/10.3389/fpsyg.2013.00863
Landis, J. R., & Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, *33*(1), 159–174. https://doi.org/10.2307/2529310
Lawrence, M., Anuj, S., & Gerald, M. K. (1995). Statistical-based or condition-based preventive maintenance? *Journal of Quality in Maintenance Engineering*, *1*(1), 46–59.
Lim, T., Loh, W., & Shih, Y. A. (2000). A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. *Machine Learning*, *40*, 203–228. https://doi.org/10.1023/A:1007608224229
Liu, H. C., Liu, L., & Liu, N. (2013). Risk evaluation approaches in failure mode and effects analysis: a literature review. *Expert Syst. Appl.*, *40*, 828–838.
Loraine, C. A., & Ashok, K. D. (2012). Market Basket Analysis for a Supermarket based on Frequent Itemset Mining. *IJCSI International Journal of Computer Science Issues*, *9*(5).
Mahalanobis, P. C. (1936). On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 45–55. Retrieved from https://insa.nic.in/writereaddata/UpLoadedFiles/PINSA/Vol02_1936_1_Art05.pdf
Majumder, B. P., Sengupta, A., Jain, S., & Bhaduri, P. (2016). Fault Detection Engine in Intelligent Predictive Analytics Platform for DCIM. *4th International Conference on Business Analytics and Intelligence (ICBAI 2016)*, (Dcim).
Mathur, A. (2002). Data Mining of Aviation Data for Advancing Health Management. *Proceedings Volume 4733, Component and Systems Diagnostics, Prognostics, and Health Management II*, *4733*, 61–71. https://doi.org/10.1117/12.475495
McDonald, J. H. (2014). *Handbook of Biological Statistics* (3rd Editio). Retrieved from http://www.biostathandbook.com/transformation.html
Mcgill, R., Tukey, J. W., & Larsen, W. A. (1978). Variations of Box Plots. *The American Statistician*, *32*(1), 12–16. https://doi.org/10.2307/2683468
Medina-Oliva, G., Lung, B., Barberá, L., Viveros, T., & Ruin, T. (2012). Root cause analysis to identify physical causes. *11th International Probabilistic Safety Assessment and Management Conference and the Annual European Safety and Reliability Conference*.
Minitab. (2011). Assumptions and Normality. Retrieved July 25, 2019, from https://blog.minitab.com/blog/quality-data-analysis-and-statistics/assumptions-and-normality
Morant, A., & Larsson-Kråik, P. O. Kumar, U. (2016). Data-driven model for maintenance decision support: A case study of railway signalling systems. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 220–234. Retrieved from https://doi.org/10.1177/0954409714533680
Mosaddar, D., & Shojaie, A. A. (2013). A data mining model to identify inefficient maintenance activities. *International Journal of Systems Assurance Engineering and Management*, *4*(2), 1890192. Retrieved from https://doi.org/10.1007/s13198-013-0148-7
Mosahab, R., Ramayah, T., & Mahamad, O. (2010). Service Quality , Customer Satisfaction and Loyalty : A Test of Mediation. *International Business Research*, *3*(4), 72–80.
Mukaka, M. M. (2012). A guide to appropriate use of Correlation coefficient in medical research. *Malawi Medical Journal*, *24*(3), 69–71. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3576830/
Nebl, T., & Schroeder, A. K. (2011). Understanding the interdependencies of quality problems and productivity. *TQM Journal*, 480–495.
Nguyen, T. T., & Ha, Q. T. (2014). Novel Operations for FP-Tree Data Structure and Their Applications. In *Advanced Computational Methods for Knowledge Engineering*.
Nisbet, R., Miner, G., & Yale, K. (2018). *Handbook of Statistical Analysis and Data Mining Applications - Chapter 8 - Advanced Algorithms for Data Mining* (2nd Editio). https://doi.org/https://doi.org/10.1016/B978-0-12-416632-5.00008-6
Niu, G., & Pecht, M. (2009). A framework for cost-effective and accurate maintnance combining CBM RCM and data fusion. *Proceedings of 2009 8th International Conference on Reliability, Maintainability and Safety.*, 605–611.
Papic, L., Kovacevic, S., Galar, D., & Thaduri, A. (2016). *Investigation of Causes of Mining Machines Maintenance Problems, Current Trends in Reliability, Availability, Maintainability and Safety*.
Patil, R. B., Patil, M. A., Ravi, V., & Naik, S. (2017). Predictive modeling for corrective maintenance of imaging devices from machine logs. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 1676–1679. https://doi.org/10.1109/EMBC.2017.8037163
Persistence Market Research. (2016). Global Market Study on Diagnostic Imaging Devices: Asia-Pacific to

Witness Highest Growth by 2020. Retrieved June 27, 2019, from http://www.persistencemarketresearch.com/market-research/diagnosticimaging-devices-market.asp

Pete, C., Julian, C., Randy, K., Thomas, K., Thomas, R., Colin, S., & Wirth, R. (2004). CRISP-DM 1.0. *CRISP-DM Consortium*. Retrieved from https://doi.org/10.1109/ICETET.2009.239

Politis, Y., Giovanis, A., & Binioris, S. (2014). Logistics service quality and its effects on customer satisfaction in the manufacturing companies' supply chain: Empirical evidence from Greece. *Journal of Modelling in Management*, *9*(2), 2150237. https://doi.org/10.1108/JM2-05-2012-0016

Powers, D. M. W. (2011). EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION. *Journal of Machine Learning Technologies*, *2*(1), 37–63. Retrieved from https://www.academia.edu/3090500/Evaluation_From_Precision_Recall_and_F-Measure_to_ROC_Informedness_Markedness_and_Correlation

Provost, F., & Fawcett, T. (2013). *Data Science for Business - What You Need to Know About Data Mining and Data-Analytic Thinking* (2nd Editio; M. Loukides & M. Blanchette, Eds.). Sebastopol: O'Reilly Media, Inc.

Redman, T. C. (2008). *Data Driven: Profiting from Your Most Important Business Asset*. Boston, MA: Harvard Business Review Press.

Reichheld, F. F., & Schefter, P. (2000). E-loyalty: your secret weapon on the web. In *Harvard Business Review*. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:E-loyalty:+your+secret+weapon+on+the+web#1

Reid, I., & Smyth-Renshaw, J. (2012). Exploring the Fundamentals of Root Cause Analysis: Are We Asking the Right Questions in Defining the Problem? *Quality and Reliability Engineering International - Special Issue: ENBIS 11*, *28*(5), 535–545. Retrieved from https://doi.org/10.1002/qre.1435

Research and Markets. (2016). Global MRI Market Worth USD 7 Billion by 2020 - Analysis, Technologies & Forecasts Report 2015-2020. Retrieved June 28, 2019, from https://www.businesswire.com/news/home/20160725005674/en/Global-MRI-Market-Worth-USD-7-Billion

Rezvanizaniani, S. M., Dempsey, J., & Lee, J. (2014). An effective predictive maintenance approach based on historical maintenance data using a probabilistic risk assessment: PHM14 data challenge. *International Journal of Prognostics and Health Management*, *5*(2).

Rosmaini, A., & Shahrul, K. (2012). *An overview of time-based and condition-based maintenance in industrial application*. Universiti Sains Malaysia.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *ArXiv Preprint*.

Sapp, M., Obiakor, F. E., Gregas, A. J., & Scholze, S. (2007). Mahalanobis Distance: A Multivariate Measure of Effect in Hypnosis Research. *Sleep and Hypnosis*, *9*(2), 67–70. Retrieved from https://www.researchgate.net/publication/232537967

Schoenfisch, J. ., Stülpnagel, J. ., Ortmann, J. ., Meilicke, C. ., & Stuckenschmidt, H. (2016). Root Cause Analysis through Abduction in Markov Logic Networks. *2016 IEEE 20th International Enterprise Distributed Object Computing Conference (EDOC)*, 1–8. https://doi.org/10.1109/EDOC.2016.7579386

Seno, M., & Karypis, G. (2005). Finding Frequent Itemsets Using Length-Decreasing Support Constraint. *Data Mining and Knowledge Discovery*, *10*, 197–228.

Sharma, R. K., & Sharma, P. (2010). System failure behavior and maintenance decision making using, RCA, FMEA and FM. *Journal of Qualitative Maintenance Engineering*, *16*, 64–88.

Sharma, Rajiv Kumar, Kumar, D., & Kumar, P. (2005). Systematic failure mode effect analysis (FMEA) using fuzzy linguistic modelling. *International Journal of Quality and Reliability Management*, *22*(9), 986–1004. https://doi.org/10.1108/02656710510625248

Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing*, *5*, 13–22.

Sheskin, D. J. (2011). *Handbook of Parametric and Non-Parametric Statistical Procedures* (5th Editio). London, United Kingdom: Chapman & Hall/CRC.

Shidhu, S., Meena, U. K., Nawani, A., Gupta, H., & Thakur, N. (2014). FP Growth Algorithm Implementation. *International Journal of Computer Applications*, *93*(8). https://doi.org/10.5120/16233-5613

Silverstein, C., Brin, S., Motwani, R., & Ullman, J. (2000). Scalable Techniques for Mining Causal Structures. *Data Mining and Knowledge Discovery*, *4*(2–3), 163–192.

Simske, S. (2019). *Meta-Analytics: Consensus Approaches and System Patterns for Data Analysis - Chapter 1 - Introduction, overview, and applications*. https://doi.org/https://doi.org/10.1016/B978-0-12-814623-1.00001-0

Sipos, R., Wang, Z., & Moerchen, F. (2014). Log-based Predictive Maintenance. *KDD '14 Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1867–1876.

https://doi.org/10.1145/2623330.2623340
Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, *45*(4), 427–437. https://doi.org/https://doi.org/10.1016/j.ipm.2009.03.002
Sullivan, G. M., & Feinn, R. (2012). Using Effect Size—or Why the P Value Is Not Enough. *Journal of Graduate Medical Education*, *4*(3), 279–282. https://doi.org/10.4300/JGME-D-12-00156.1
Sun, Y., Wong, A. K. W., & Kamel, M. S. (2009). CLASSIFICATION OF IMBALANCED DATA: A REVIEW. *International Journal of Pattern Recognition and Artificial Intelligence|*, *23*(4), 687–719. https://doi.org/https://doi.org/10.1142/S0218001409007326
Taghipour, S., & Banjevic, D. (2012). Optimal Inspection of a complex system subject to periodic and opportunistic inspections and preventive replacements. *European Journal of Operational Research*, *220*(3), 649–660. https://doi.org/https://doi.org/10.1016/j.ejor.2012.02.002
Tan, P. N., Steinbach, M., Karpatne, A., & Kumar, V. (2019). *Introduction to Data Mining* (2nd Editio). Pearson.
Tantithamthavorn, C., Hassan, A., & Matsumoto, K. (2018). The Impact of Class Rebalancing Techniques on the Performance and Interpretation of Defect Prediction Models. *Software Engineering*, 1–20.
Valle, M. A., Ruz, G. A., & Morras, R. (2018). Market Basket Analysis: Complementing association rules with minimum spanning trees. *Expert Systems with Applications2*, *97*, 146–162. https://doi.org/https://doi.org/10.1016/j.eswa.2017.12.028
van Aken, J., Berends, H., & van der Bij, H. (2012). *Problem Solving In Organizations - A Methodological Handbook for Business and Management Students (2nd Edition)* (2nd Editio). Cambridge: Cambridge University Press.
van der Loo, M. P. J. (2014). The stringdist Package for Approximate String Matching. *The R Journal*, *6*(1). https://doi.org/10.32614/rj-2014-011
Viera, A., & Garrett, J. M. (2005). Understanding Interobserver Agreement: The Kappa Statistic. *Family Medicine*, *37*(5), 360–363.
Vinodh, S., & Santhosh, D. (2012). Application of FMEA to an automotive leaf spring manufacturing organization. *TQM Journal*, *24*(3), 260–274. https://doi.org/10.1108/17542731211226772
Viveros, P., Zio, E., Nikulin, C., Stegmaier, R., & Bravo, G. (2014). Resolving equipment failure causes by root cause analysis and theory of inventive problem solving. *Proc. Inst. Mech. Eng. O J Risk Reliab.*, 93–111.
Vogler, R. (2013). Comparison of String Distance Algorithms. Retrieved October 15, 2019, from Joy of Data website: https://www.joyofdata.de/blog/comparison-of-string-distance-algorithms/
Walker, H. M. (1940). Degrees of Freedom. *Journal of Educational Psychology*, *31*(4), 253–269.
Wang, J., & Cheng, Z. (2018). FP-Growth based Regular Behaviors Auditing in Electgric Management Information System. *Procedia Computer Science*, *139*, 275–279. https://doi.org/10.1016/j.procs.2018.10.268
Wei, W. (2013). *Vertical Specialization and Trade Surplus in China*. Woodhead Publishing Limited.
Weng, S. S., & Liu, J. L. (2004). Feature-based recommendations for one-to-one marketing. *Expert Systems with Applications*, *26*, 493–508.
Westfall, P. H. (2014). Kurtosis as Peakednes. *The American Statistician*, *68*(3), 191–195. https://doi.org/10.1080/00031305.2014.917055
Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. *). Proceedings of the Section on Survey Research Methods*, 354–359. American Statistical Association.
Winkler, W. E. (1999). *The State of Record Linkage and Current Research Problems*. Retrieved from https://www.census.gov/srd/papers/pdf/rr99-04.pdf
Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a Standard Process Model for Data Mining. *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining.*
Wu, H., Liu, Y., & Ding, Y. (2003). A method of aircraft unit fault diagnosis. *Aircraft Engineering and Aerospace Technology*, *75*(1), 27–32. https://doi.org/10.1108/00022660310457266
Zadry, H. R., Saputra, D. A., Tabri, A. B., Meilani, Di., & Rahmayanti, Di. (2018). Failure Modes and Effects Analysis (FMEA) for evaluation of a sugarcane machine failure. *International Mechanical and Industrial Engineering Conference 2018 (IMIEC 2018)*, *204*, 1–5. https://doi.org/10.1051/matecconf/201820401012
Zaki, M. J. (2001). SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning*, *42*, 31–60.
Zawawy, H. ., Kontogiannis, K. ., Mylopoulos, J. ., & Mankovskii, S. (2012). Requirements-Driven Root Cause Analysis Using Markov Logic Networks. *International Conference on Advanced Information Systems Engineering*, 350–365. https://doi.org/10.1007/978-3-642-31095-9_23

Zheng, Yin, Liu, & Zhang. (2015). Research on Improved FP-Growth Algorithm in Association Rules Mining. *Scientific Programming*, *1*, 1–6. https://doi.org/http://dx.doi.org/10.1155/2015/910281

Zhu, P., Liyanage, J., & Jeeves, S. (2019). Data-driven failure analysis of emergency shutdown systems in oil and gas industry: Evaluation of detectability from failure progression perspective. *Journal of Quality in Maintenance Engineering*. https://doi.org/10.1108/JQME-03-2018-0022

Zwetsloot, I. M., Buitenhuis, M., Lameijer, B. A., & Does, R. J. M. M. (2015). Quality Quandaries: Increasing the First Time Fix Rate in a Customer Contact Center. *Quality Engineering*, *27*(3), 393–400. https://doi.org/10.1080/08982112.2015.1036297

Appendices

Appendix A MR Troubleshooting Process

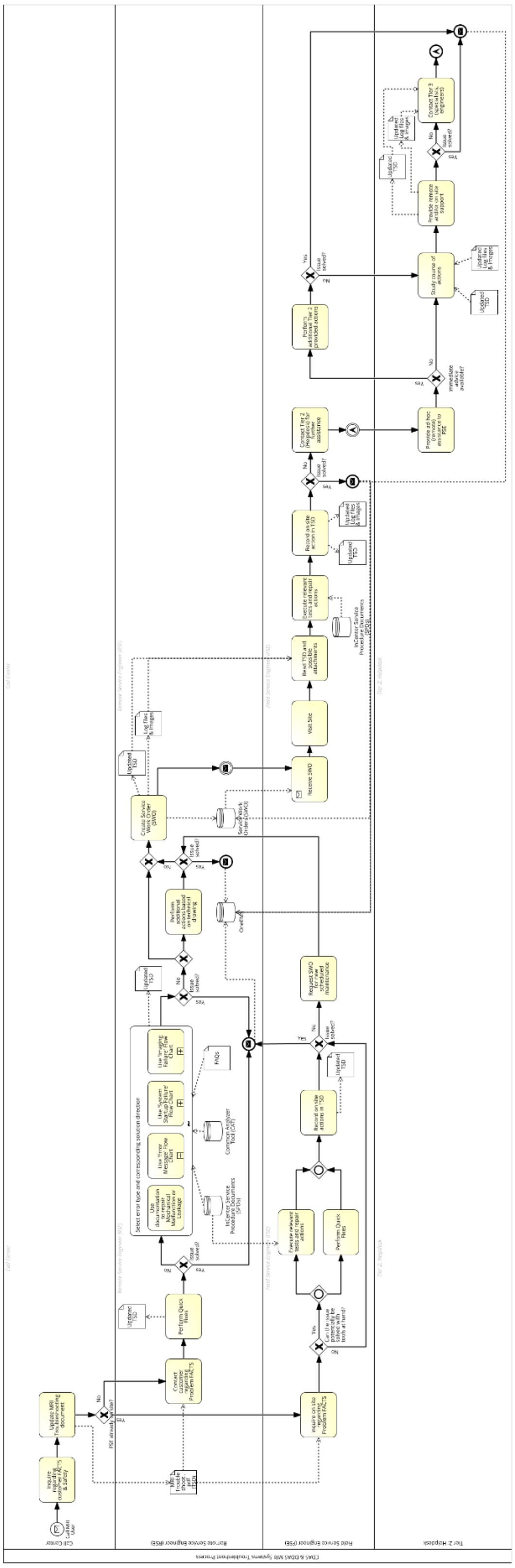
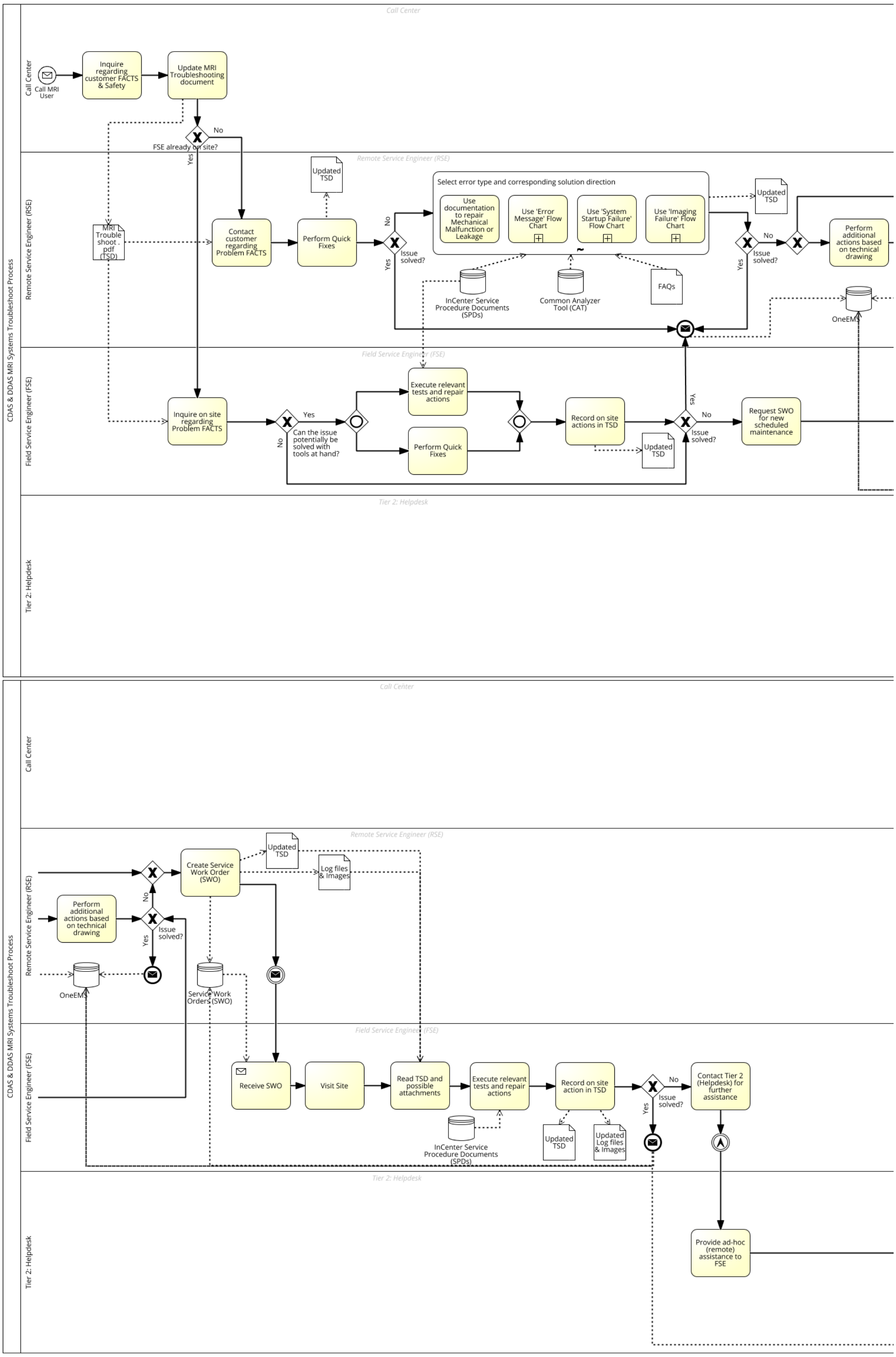
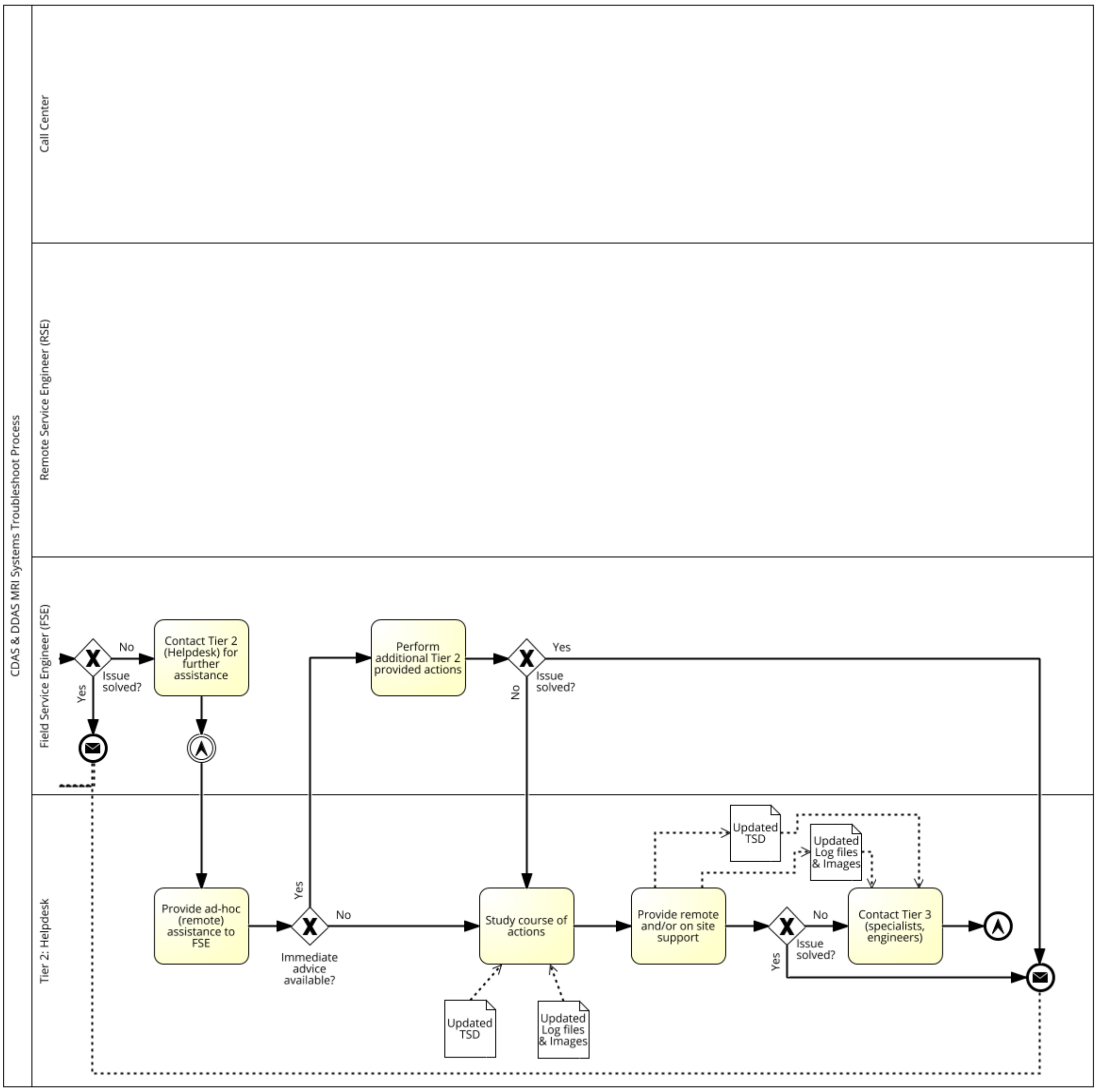


Fig. 33 - MR Troubleshooting Process (complete)



MR Troubleshooting Process (1/2)



MR Troubleshooting Process (2/2)

Appendix B R SCRIPT - DATA UNDERSTANDING

```
# DATA UNDERSTANDING, SPC, LABOR, FVF EXTRACTED DATA
# A. (Arash) Shahrestani

# LOAD REQUIRED PACKAGES
library(MASS)
library(ggplot2)
library(readxl)
library(dplyr)
library(moments)
library(writexl)
library(robustbase)
library(tidyverse)
library(extrafont)
font_import()           # Importing fonts, writing font table & fontmap might take a few minutes.
# Solely for aesthetics, not necessary to take into account, change script accordingly.
loadfonts(device="win") # Register fonts for Windows bitmap output
fonts()                 # Overview of available fonts.
library("ggpubr")

#Read Extracted CaseBased Data
New_FilteredData_v2_1_sheet <- read_excel("C:/xxx.xlsx")
View(New_FilteredData_v2_1_sheet)
df2 <- New_FilteredData_v2_1_sheet

sapply(df2, mean, na.rm=TRUE)
df2$NumberVisits <- as.integer(df2$NumberVisits)

# STRIP CHARTS
windowsFonts(corbel = windowsFont("Corbel"))
stripchart(df$NumberVisits ~ df$`System Type`, family = "corbel", font = 1, font.lab = 1, font.axis = 3,
           main="Number of Visits strip chart per System Type",
           xlab="System Type",
           ylab="Number of Visits",
           col="dodgerblue4",
           cex= 0.5,
           group.names=c("Achieva","Ingenia","Intera","Multiva"),
           vertical=TRUE,
           method = "jitter",
           las = 1,
           pch=16)
meanVisits = tapply(df$NumberVisits,df$`System Type`,mean)
segments(x0 = c(1,2,3,4) - 0.17,
         y0 = meanVisits,
         x1 = c(1,2,3,4) + 0.17,
         y1 = meanVisits, lwd = 3,
         col= "red")

windowsFonts(corbel = windowsFont("Corbel"))
stripchart(df$NumberVisits ~ df$Year,
           main="Number of Visits strip chart per Year", family = "corbel", font = 1, font.lab = 1,
font.axis = 3,
           xlab="Year",
           ylab="Number of Visits",
           col="dodgerblue4",
           group.names=c("2012","2013","2014","2015","2016","2017","2018","2019"),
           vertical=TRUE,
           cex= 0.5,
           method = "jitter",
           las = 1,
           pch=16)
meanVisits = tapply(df$NumberVisits,df$`System Type`,mean)
segments(x0 = c(1,2,3,4,5,6,7,8) - 0.17,
         y0 = meanVisits,
         x1 = c(1,2,3,4,5,6,7,8) + 0.17,
         y1 = meanVisits, lwd = 3,
         col= "red")
```

Appendix C LABOR ACTIVITIES

Table 30 – Labor Activity Explanation, per Labor Category

Labor Cost Category & Codes	Explanation
Total Travel Cost	
TRAV	Travel Time
TRVL	Travel Time International
Total Corrective Maintenance Cost	
CMAI	Corrective Maintenance
DIAG	Diagnostics
FILL	Filling
MONI	System Observation
RPCL	Repair Center Labor
RTST	Regulatory Testing
SWSU	Software Support
Total Remote Cost	
APAS	Remote Application Support
RMSE	Remote Service
TESU	Technical Support during Corrective Maintenance
Total Installation Cost	
BCKO	Backorder Installation
DEIN	Equipment De-Installation
IN01	Transport / Unpacking / Mounting
IN02	Installation
IN03	Setting to Work
IN04	Performance Check
IN05	Installation Hand-Over
LOCA	Installation Local Addition/n-PMS; additional, non-default, option for client' system
PRCO	Project Coordination
REIN	Equipment Reinstallation
SIRE	Site Readiness Activity
UPGR	Installing Upgrade
UTRA	User Training
Total Application Cost	
APSE	On-site Application Support
BTRA	Product Service Biomed

Appendix D ENTITY RELATION DIAGRAM

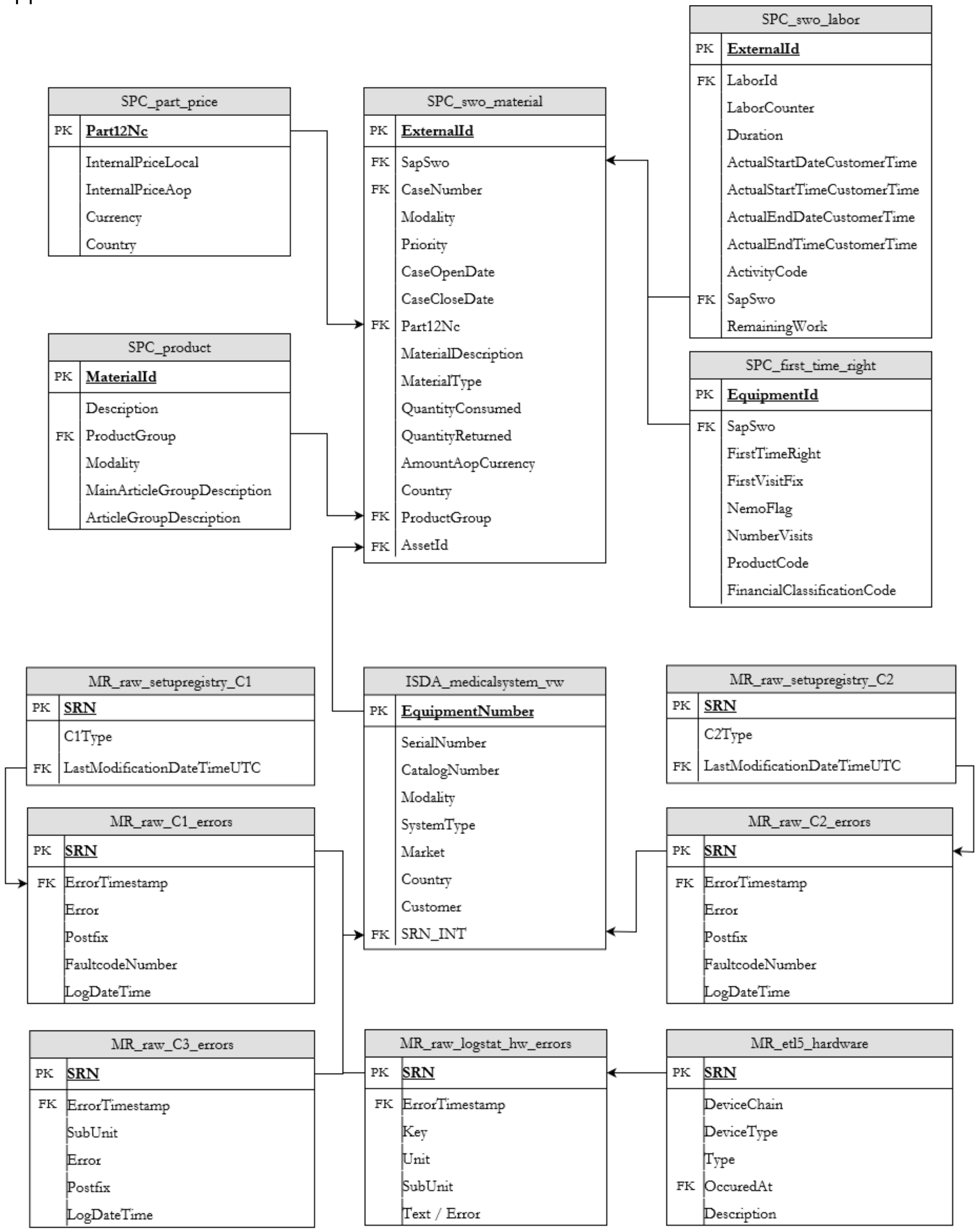


Fig. 34 - Entity Relation Diagram

Appendix E DATA TRANSFORMATION VBA'S & SQL

Transforming Part12Nc's and Material Descriptions from record based to casebased:

```
Function MYVLOOKUP(pValue As String, pWorkRng As Range, pIndex As Long)
Dim rng As Range
Dim xResult As String
xResult = ""
For Each rng In pWorkRng
    If rng = pValue Then
        xResult = xResult & ";" & " " & rng.Offset(0, pIndex - 1)
    End If
Next
MYVLOOKUP = xResult
End Function
```

Additionally, an extra column has been introduced, *'#12Nc per Case'*, counting the number of parts used for a distinct case based on the output per case of the VBA code above:

```
=(LEN(C2)) - LEN(SUBSTITUTE(C2," ",""))
```

Average Part Cost Prices from Vertica

```
SELECT DISTINCT "Development"."SPC_part_price"."Part12Nc",
AVG("Development"."SPC_part_price"."InternalPriceAop"), COUNT ("Development"."SPC_part_price"."Part12Nc")
AS FrequencyInPartPriceTable, "Development"."SPC_swo_material"."CaseNumber"
```

Sum AopCurrency per Casenumber in Vertica

```
SUM("Development"."SPC_swo_material"."AmountAopCurrency"), COUNT
("Development"."SPC_swo_material"."CaseNumber") AS FrequencyInSWOMaterialTable
```

Snippet Labor Cost Calculation in Vertica

```
,sum(CASE WHEN "Development"."SPC_swo_labor"."ActivityCode" IN ( 'TRAV','TRVL') THEN
"Development"."SPC_swo_labor"."Duration" * 100 ELSE 0 END) AS totaltravelcosts
,sum(CASE WHEN "Development"."SPC_swo_labor"."ActivityCode" IN
('CMAI','DIAG','FILL','MONI','RPCL','RTST','SWSU') THEN "Development"."SPC_swo_labor"."Duration" * 100
ELSE 0 END) AS totalcmcosts
,sum(CASE WHEN "Development"."SPC_swo_labor"."ActivityCode" IN ('APAS','RMSE','TESU') THEN
"Development"."SPC_swo_labor"."Duration" * 100 ELSE 0 END) AS totalremotecosts
,sum(CASE WHEN "Development"."SPC_swo_labor"."ActivityCode" IN
('BCKO','DEIN','IN01','IN02','IN03','IN04','IN05','LOCA','PRCO','REIN','SIRE','UPGR','UTRA') THEN
"Development"."SPC_swo_labor"."Duration" * 100 ELSE 0 END) AS totalinstallationcosts
,sum(CASE WHEN "Development"."SPC_swo_labor"."ActivityCode" IN ('APSE','BTRA') THEN
"Development"."SPC_swo_labor"."Duration" * 100 ELSE 0 END) AS totalapplicationcosts
```

Appendix F R SCRIPT - OUTLIER IDENTIFICATION

```
# Required Packages
library(MASS)
library(ggplot2)
library(readxl)
library(dplyr)
library(moments)
library(writexl)
library(robustbase)
library(tidyverse)
library(extrafont)
library("ggpubr")
font_import() # Importing fonts, writing font table & fontmap might take a few minutes.
# Solely for aesthetics, not necessary to take into account, change script accordingly.
loadfonts(device = "win") # Register fonts for Windows bitmap output
fonts() # Overview of available fonts.

# import data
New_FilteredData_v2_1_sheet <- read_excel("C:/xxx/New_FilteredData_v2_1_sheet.xlsx")
View(New_FilteredData_v2_1_sheet)
df2 <- New_FilteredData_v2_1_sheet

# Filter negative total costs, if applicable
df <- df2 %>% filter(df2$`Total sumaop + labor` >= 0)

# Skewness & Kurtosis Calculation
skewness(df$`Total sumaop + labor`)
kurtosis(df$`Total sumaop + labor`)

# Histrogram Pre Transformation
windowsFonts(corbel = windowsFont("Corbel"))
hist(df$`Total sumaop + labor`, breaks = 10000, xlim = range(0:30000), main = "Histogram Total Cost all
Systems 2012-2019 ", xlab = "Total Cost (Net Part + Labor)", col = "cornsilk1", freq = TRUE, family =
"corbel", font = 1, font.lab = 1, font.axis = 3)

# Histrogram Post Transformation
hist(log(df$`Total sumaop + labor`), main = "Histogram log-transformed Total Cost", xlab = "log(Net Part +
Labor Cost)", xlim = c(0, 15), ylim = c(0, 0.30), col = "cornsilk1", freq = FALSE, family = "corbel", font
= 1, font.lab = 1, font.axis = 3)
curve(dnorm(x, mean = mean(log(df$`Total sumaop + labor`)), sd = sd(log(df$`Total sumaop + labor`))), add
= TRUE, col = "dodgerblue4", lwd = 2)

# Histogram - Compare distributions Total Cost (Labor + Part)
# Righ Skewed Data, Hence Log Transform
# Three similar histograms, different amount of breaks
windowsFonts(corbel = windowsFont("Corbel"))
hist(log(df$`Total sumaop + labor`), main = "Histogram log-transformed Total Cost", xlab = "log(Net Part +
Labor Cost)", xlim = c(0, 15), ylim = c(0, 0.30), breaks = 25, col = "cornsilk1", freq = FALSE, family =
"corbel", font = 1, font.lab = 1, font.axis = 3)
curve(dnorm(x, mean = mean(log(df$`Total sumaop + labor`)), sd = sd(log(df$`Total sumaop + labor`))), add
= TRUE, col = "dodgerblue4", lwd = 2)

windowsFonts(corbel = windowsFont("Corbel"))
hist(log(df$`Total sumaop + labor`[df$FirstVisitFix == 0]), main = "Histogram log-transformed Total Cost,
FVF=False", xlab = "log(Net Part + Labor Cost)", xlim = c(2.5, 15), ylim = c(0, 0.40), breaks = 25, col =
"cornsilk1", freq = FALSE, family = "corbel", font = 1, font.lab = 1, font.axis = 3)
curve(dnorm(x, mean = mean(log(df$`Total sumaop + labor`[df$FirstVisitFix == 0])), sd = sd(log(df$`Total
sumaop + labor`[df$FirstVisitFix == 0]))), add = TRUE, col = "dodgerblue4", lwd = 2)

windowsFonts(corbel = windowsFont("Corbel"))
hist(log(df$`Total sumaop + labor`[df$FirstVisitFix == 1]), main = "Histogram log-transformed Total Cost,
FVF=True", xlab = "log(Net Part + Labor Cost)", xlim = c(0, 15), ylim = c(0, 0.30), breaks = 25, col =
"cornsilk1", freq = FALSE, family = "corbel", font = 1, font.lab = 1, font.axis = 3)
curve(dnorm(x, mean = mean(log(df$`Total sumaop + labor`[df$FirstVisitFix == 1])), sd = sd(log(df$`Total
sumaop + labor`[df$FirstVisitFix == 1]))), add = TRUE, col = "dodgerblue4", lwd = 2)
```

```
# Boxplots
# Boxplot First Visit Fix Single/Multiple visit, non-transformed data
par(family = "Corbel")
boxplot((df$`Total sumaop + labor`) ~ df$FirstVisitFix,
  xlab = "First Visit Fix", ylab = "Total Cost",
  main = "Overall FVF ~ Total Cost per Year",
  font = 1,
  font.lab = 3,
  font.axis = 6,
  names = c("FVF Multiple Visit", "FVF Single Visit"),
  horizontal = FALSE,
  col = "ivory",
  whisklty = 2,
  outcol = "darkred",
  staplelty = 6,
  pch = 3,
  cex = 0.4
)

# Boxplot First Visit Fix Single/Multiple visit, log-transformed data
par(family = "Corbel")
boxplot(log(df$`Total sumaop + labor`) ~ df$FirstVisitFix,
  xlab = "First Visit Fix", ylab = "Total Cost,  log-transformed",
  ylim = c(0, 15),
  main = "Overall FVF ~ Total Cost per Year",
  font = 1,
  font.lab = 3,
  font.axis = 6,
  names = c("FVF Multiple Visit", "FVF Single Visit"),
  horizontal = FALSE,
  col = "ivory",
  whisklty = 2,
  outcol = "darkred",
  staplelty = 6,
  pch = 3,
  cex = 0.4
)

# Boxplot Collection - Total Cost per Year for Single and Multiple Visits, all data
par(family = "Corbel")
boxplot(log(df$`Total sumaop + labor`) ~ df$FirstVisitFix * df$Year,
  xlab = "Year", ylab = "Log Total (Net Part + Labor) Cost",
  ylim = c(2, 15),
  main = "Boxplots for Multiple and Single Visits per Year",
  font = 1,
  font.lab = 3,
  font.axis = 6,
  names = c("2012", "2012", "2013", "2013", "2014", "2014", "2015", "2015", "2016", "2016", "2017",
"2017", "2018", "2018", "2019", "2019"),
  horizontal = FALSE,
  col = "ivory",
  whisklty = 2,
  outcol = "red",
  staplelty = 6,
  pch = 3,
  cex = 0.4
)

# Additional Boxplot for Single and Multiple visit separately.
par(family = "Corbel")
boxplot(log(df$`Total sumaop + labor`[df$FirstVisitFix == "0"]) ~ df$Year[df$FirstVisitFix == "0"], xlab =
"Year (Multiple Visits)", ylab = "Total (Net Part + Labor) Cost", ylim = c(0, 15), main = "Multiple Visits
per year ~ Total Cost", pch = 3, cex = 0.4)
par(family = "Corbel")
boxplot(log(df$`Total sumaop + labor`[df$FirstVisitFix == "1"]) ~ df$Year[df$FirstVisitFix == "1"], xlab =
"Year (Single Visits)", ylab = "Total (Net Part + Labor) Cost", ylim = c(0, 12), main = "Single Visits per
year ~ Total Cost", pch = 3, cex = 0.4)
```

```
# Welch's t-tests
t.test((df$TotalLaborCost[df$FirstVisitFix == "0"]), (df$TotalLaborCost[df$FirstVisitFix == "1"]))
t.test(log(df$`Total sumaop + labor`[df$FirstVisitFix == "0"]), (log(df$`Total sumaop +
labor`[df$FirstVisitFix == "1"])))

## CONCLUSION TWO-SAMPLE T-TEST OF EQUAL MEANS:
# FOR FVF FALSE (0) MEAN: 2.72209
# FOR FVF TRUE (1) MEAN: 1.00000
# t = 303.54, df= 77632, p-value <2.2e-16
# 95% CI: 1.710970, 1.733209
# Field service engineers who spend multiple visits per case, on average, spend significantly more on cost
than those that do not.

# Compute the Mahalanobis distances. Present graphically.
LognTotalCost <- log(df$`Total sumaop + labor`) # New variable logtransformed
LognNrVis <- log(df$NumberVisits) # New variable logtransformed Nr of employees
Combined2 <- cbind(LognNrVis, LognTotalCost) # Combine both new variables to matrix

meanData2 <- colMeans(Combined2) # Mean of logtransformed
S2 <- cov(Combined2) # covariancematrix of logtransformed
MD2 <- mahalanobis(Combined2, meanData2, S2) # Calculating mahalanobis distance
boxplot(MD2, main = "Mahalanobis distance") # Boxplot MD distance
plot(Combined2, main = "Multivariate outliers") # Plot

max(MD2)
which.max(MD2)
Combined2[which.max(MD2), ]
df <- cbind(df, Combined2, MD2)

## Q,Q Plot Normality Check
qqnorm((LognTotalCost), main = "Normal Q-Q Plot logtransformed Total Cost", family = "corbel", font = 1,
font.lab = 1, font.axis = 3) # Check for normality
qqline(LognTotalCost, col = "dodgerblue4", lwd = 2) # draw line

## Post Transformation Skewness & Kurtosis (Normality) Check
skewness(log(df$`Total sumaop + labor`))
kurtosis(log(df$`Total sumaop + labor`))

# Export Outlier MD Calculation as .XLSX File
write_xlsx(x = df, path = "C:/xxx/RStudioOutput.xlsx", col_names = TRUE)
```

Appendix G ERROR AND PART TYPE RE-LABELING PER CHAIN

[Appendix unavailable due to confidential content]

Appendix H MS EXCEL, VBA SCRIPT FOR ERROR IDENTIFICATION

H1. Multiple string values concatenation, given multiple conditions

```
Function ConcatenateIf(CriteriaRange As Range, Condition As Variant, TimeRange As Range, TimeCondition1 As Variant, TimeCondition2 As Variant, ConcatenateRange As Range, Optional Separator As String = ",") As Variant

Dim xResult As String
On Error Resume Next
If CriteriaRange.Count <> ConcatenateRange.Count Then
    ConcatenateIf = CVErr(xlErrRef)
    Exit Function
End If
For i = 1 To CriteriaRange.Count
    If CriteriaRange.Cells(i).Value = Condition And TimeRange.Cells(i).Value >= TimeCondition1 And TimeRange.Cells(i).Value <= TimeCondition2 Then
        xResult = xResult & Separator & " " & ConcatenateRange.Cells(i).Value
    End If
Next i
If xResult <> "" Then
    xResult = VBA.Mid(xResult, VBA.Len(Separator) + 1)
End If
ConcatenateIf = xResult
End Function
```

H2. Identify unique fault (string) codes; remove duplicates from concatenation

```
Function RemoveDuplicateFaultCodes(txt As String, Optional delim As String = " ") As String
    Dim x
    With CreateObject("Scripting.Dictionary")
        .CompareMode = vbTextCompare
        For Each x In Split(txt, delim)
            If Trim(x) <> "" And Not .exists(Trim(x)) Then .Add Trim(x), Nothing
        Next
        If .Count > 0 Then RemoveDuplicateFaultCodes = Join(.keys, delim)
    End With
End Function
```

H3. Alphabetical fault code sorter within a cell (*Adapted from Excel Forum (2012)*)

```
Function SortWithinCell(CelltoSort As Range, DelimitingCharacter As String, IncludeSpaces As Boolean) As String
CelltoSortString = WorksheetFunction.Substitute(CelltoSort.Value, " ", "")
MyArray = Split(CelltoSortString, DelimitingCharacter)
    For N = 0 To UBound(MyArray)
        For M = 1 To UBound(MyArray)

                If MyArray(M) < MyArray(M - 1) Then
                    TempValue = MyArray(M)
                    MyArray(M) = MyArray(M - 1)
                    MyArray(M - 1) = TempValue
                End If
        Next M
    Next N
For N = 0 To UBound(MyArray)
    SortWithinCell = SortWithinCell & MyArray(N) & DelimitingCharacter
Next N
SortWithinCell = Left(SortWithinCell, Len(SortWithinCell) - 1)
If IncludeSpaces = True Then SortWithinCell = WorksheetFunction.Substitute(SortWithinCell, ",", ", ")
End Function
```

Appendix I R SCRIPT – STATISTICAL SIGNIFICANCE

```
library(styler)
library(effsize)

# STATISTICAL SIGNIFICANCE  w/ SOME MD/q>3 VALUES EXLUDED, ONLY 123 VISITS

# SIGNIFICANTIE CHECK OUTLIER DATASET ONLY 123 VISITS
dfOnly123Visit <- read_excel("C:/xxx.xlsx",
  sheet = "FilteredData+Copy+Interval.Tab5"
)
View(dfOnly123Visit)
dfOnly123Visitv <- df%>% filter(df$NumberVisits<4)

hist(log(dfOnly123Visit$`Total sumaop + labor`), main = "Log transformed TotalCost", xlab =
"Logtransformed TotalCost", xlim = c(0,15)) # log transformed histogram
boxplot(log(dfOnly123Visit$`Total sumaop + labor`) ~ dfOnly123Visit$FirstVisitFix, xlab = "First Visit
Fix", ylab = "Total Cost")


# SPLIT DATA SET IN SUBSETS PER YEAR
dfOnly123Visit2013 <- dfOnly123Visit %>% filter(dfOnly123Visit$Year == 2013)
dfOnly123Visit2014 <- dfOnly123Visit %>% filter(dfOnly123Visit$Year == 2014)
dfOnly123Visit2015 <- dfOnly123Visit %>% filter(dfOnly123Visit$Year == 2015)
dfOnly123Visit2016 <- dfOnly123Visit %>% filter(dfOnly123Visit$Year == 2016)
dfOnly123Visit2017 <- dfOnly123Visit %>% filter(dfOnly123Visit$Year == 2017)
dfOnly123Visit2018 <- dfOnly123Visit %>% filter(dfOnly123Visit$Year == 2018)
dfOnly123Visit2019 <- dfOnly123Visit %>% filter(dfOnly123Visit$Year == 2019)


# WELCH' T_TEST FOR EACH YEAR ON LOG-TRANSFORMED DATA
t.test(log(dfOnly123Visit2013$`Total sumaop + labor`[dfOnly123Visit2013$FirstVisitFix == "0"]),
(log(dfOnly123Visit2013$`Total sumaop + labor`[dfOnly123Visit2013$FirstVisitFix == "1"])))
t.test(log(dfOnly123Visit2014$`Total sumaop + labor`[dfOnly123Visit2014$FirstVisitFix == "0"]),
(log(dfOnly123Visit2014$`Total sumaop + labor`[dfOnly123Visit2014$FirstVisitFix == "1"])))
t.test(log(dfOnly123Visit2015$`Total sumaop + labor`[dfOnly123Visit2015$FirstVisitFix == "0"]),
(log(dfOnly123Visit2015$`Total sumaop + labor`[dfOnly123Visit2015$FirstVisitFix == "1"])))
t.test(log(dfOnly123Visit2016$`Total sumaop + labor`[dfOnly123Visit2016$FirstVisitFix == "0"]),
(log(dfOnly123Visit2016$`Total sumaop + labor`[dfOnly123Visit2016$FirstVisitFix == "1"])))
t.test(log(dfOnly123Visit2017$`Total sumaop + labor`[dfOnly123Visit2017$FirstVisitFix == "0"]),
(log(dfOnly123Visit2017$`Total sumaop + labor`[dfOnly123Visit2017$FirstVisitFix == "1"])))
t.test(log(dfOnly123Visit2018$`Total sumaop + labor`[dfOnly123Visit2018$FirstVisitFix == "0"]),
(log(dfOnly123Visit2018$`Total sumaop + labor`[dfOnly123Visit2018$FirstVisitFix == "1"])))
t.test(log(dfOnly123Visit2019$`Total sumaop + labor`[dfOnly123Visit2019$FirstVisitFix == "0"]),
(log(dfOnly123Visit2019$`Total sumaop + labor`[dfOnly123Visit2019$FirstVisitFix == "1"])))

t.test(log(dfOnly123Visit$`Total sumaop + labor`[dfOnly123Visit$FirstVisitFix == "0"]),
(log(dfOnly123Visit$`Total sumaop + labor`[dfOnly123Visit$FirstVisitFix == "1"])))


# ST DEV CALCULATIONS OF TOTAL COST FOR SINGLE/MULTIPLE VISITS PER YEAR, TRANSFORMED DATASET
# FOR EACH SINGLE AND MULITPLE VISIT VALUE PER YEAR, USE THE FOLLOWING CODE
# WHERE sdlog201XY: X: 3-9 (YEAR), Y: 0 OR 1 (FVF SINGLE (1) OR MULITIPLE (0) VISITS)

sdlog201X0 <- sd(log(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix == "0"]))
sdlog201X1 <- sd(log(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix == "1"]))

# CONFIDENCE INTERVALS 95-% BASED ON NORM. DISTR LOG TRANSF.
# FOR EACH SINGLE AND MULITPLE VISIT VALUE PER YEAR, USE THE FOLLOWING CODE
# WHERE X: 3-9 (YEAR)
# FORUMLA: MEAN +- z*(stDev/sqrt(N))

z <- 1.96 # GIVEN Z-VALUE FOR CI FORMULA, BASED ON 95% CI.

# 201X: ORDER OUTPUT: 0 (Multiple) Upperbound, 0 Lowerbound, 1 (Single) Upperbound, 1 Lowerbound
```

```
mean(log(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix == "0"])) + z *
(sdlog201X0 / (sqrt(NROW(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix ==
"0"]))))        #UPPERBOUND MULTIPLE VISIT
mean(log(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix == "0"])) - z *
(sdlog201X0 / (sqrt(NROW(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix ==
"0"]))))        #LOWERBOUND MULTIPLE VISIT
mean(log(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix == "1"])) + z *
(sdlog201X1 / (sqrt(NROW(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix ==
"1"]))))        #UPPERBOUND SINGLE VISIT
mean(log(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix == "1"])) - z *
(sdlog201X1 / (sqrt(NROW(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix ==
"1"]))))        #LOWERBOUND SINGLE VISIT


# CALCULATE EFFECT SIZE COHENS D, ADDITIONAL STAT SIGN.
# PERFORM LINE OF CODE BELOW FOR ALL YEARS SEPARATELY.
# WHERE X: 3-9 (YEARS)

cohen.d(log(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix == "0"]),
(log(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix == "1"])), pooled = TRUE,
paired = FALSE, na.rm = FALSE, hedges.correction = FALSE, conf.level = 0.95, noncentral = FALSE)


# CALCULATE EFFECT SIZE HEDGES G, ADDITIONAL STAT SIGN.
# PERFORM LINE OF CODE BELOW FOR ALL YEARS SEPARATELY.
# WHERE X: 3-9 (YEARS)

cohen.d(log(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix == "0"]),
(log(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix == "1"])), pooled = TRUE,
paired = FALSE, na.rm = FALSE, hedges.correction = TRUE, conf.level = 0.95, noncentral = FALSE)
```

Appendix J R Script & Output – BCA Bootstrapped CI

```
# Estimate the BCa 95% - Confidence Interval based on Total Cost for Single and Multiple Visits
# Directly from the non – transformed data. Log transformation or normality check not needed due
# to the nature of the method used. Any skew is also taken into account and bias corrected.
# Additionally plot histogram of the data subset.
# Bootstrap CI based on 10000 bootstrap replicates; X: 3-9 based on the year ranging from 2013 to 2019.
library(boot)

# 201X Multiple Visit
Mboot201X0 = boot(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix=="0"],
             function(x,i) mean(x[i]),
             R=10000)
mean(Mboot201X0$t[,1])
boot.ci(Mboot201X0,
        conf = 0.95,
        type = c("norm","bca") )
Mboot201X0
hist(Mboot201X0$t[,1],
     col = "darkgray")

# 201X Single Visit
Mboot201X1 = boot(dfOnly123Visit201X$`Total sumaop + labor`[dfOnly123Visit201X$FirstVisitFix=="1"],
                  function(x,i) mean(x[i]),
                  R=10000)
mean(Mboot201X1$t[,1])
boot.ci(Mboot201X1,
        conf = 0.95,
        type = c("norm","bca") )
Mboot201X1
hist(Mboot201X1$t[,1],
     col = "darkgray")
# Output:
```

[Tables below unavailable due to confidential content]

Table 31 – Statistical Significance 'Total CM Cost' - 3

Year	BCa - 95% - CI, Multiple Visits	Mean	Bias	Standard Error
2013				
2014				
2015				
2016				
2017				
2018				
2019				

Table 32 – Statistical Significance 'Total CM Cost' - 4

Year	BCa - 95% - CI, Single Visits	Mean	Bias	Standard Error
2013				
2014				
2015				
2016				
2017				
2018				
2019				

Appendix K FVF Field Metric Analysis

[Appendix unavailable due to confidential content]

K.1 - First Visit Fix Rates per MR Product Group
K.2 - Market Based CM Cases with Parts – FVF Performance
K.3 - Top 25 – Item Frequency per Cost Interval per Single & Multiple Visit Cases, and Clustered Pareto's
K.4 - Top 25 – Chain 2 Subset – Average New Buy (Part) Price Calculation
K.5 - Top 25 – Chain 2 New Buy Price Analysis

Appendix L TECHNICAL PARTS REVIEW – FVF FIELD METRIC ANALYSIS ~ CHAIN 2 COIL

[Appendix unavailable due to confidential content]

Appendix M R SCRIPT – ERROR CODE CORRELATION

```
library(readxl)
library(ggplot2)
library(lubridate)
library(RColorBrewer)
library(GGally)
library(corrplot)
library(graphics)
library(questionr)
library(htmlTable)
library(xtable)
library(Hmisc)

# import
ErrChain12wk <- read.transactions('C:/XXX.csv', format = 'basket', sep=';')
ErrChain22wk <- read.transactions('C:/XXX.csv', format = 'basket', sep=';')
ErrChain32wk <- read.transactions('C:/XXX.csv', format = 'basket', sep=';')

summary(ErrChain12wk)
summary(ErrChain22wk)
summary(ErrChain32wk)
ErrChain12wk
ErrChain22wk
ErrChain32wk
#sep how items are separated, using ';'

# Chain 1 Errors ItemFrequencyPlot
windowsFonts(corbel = windowsFont("Corbel"))
itemFrequencyPlot(ErrChain12wk,topN=10,type="absolute",col=brewer.pal(8,'Pastel2'), main="Absolute Error Frequency Plot", ylim=c(0,30), xlim=c(0,20))
# Contingency Table
tblChain1 <- crossTable(ErrChain12wk, sort =TRUE)
tblChain1 [1:9,1:9]

# Chain 2 Errors ItemFrequencyPlot
windowsFonts(corbel = windowsFont("Corbel"))
itemFrequencyPlot(ErrChaing22wk,topN=69,type="relative",col=brewer.pal(8,'Pastel2'), main="Relative Error Frequency Plot", ylim=c(0,0.3), xlim=c(0,70))
# Contingency Table
tblChain2 <- crossTable(ErrRF2wk, sort =TRUE)
tblChain2 [1:69,1:69]

# Chain 3 Errors ItemFrequencyPlot
windowsFonts(corbel = windowsFont("Corbel"))
itemFrequencyPlot(ErrChain32wk,topN=52,type="relative",col=brewer.pal(8,'Pastel2'), main="Relative Error Frequency Plot", ylim=c(0,0.85), xlim=c(0,60))
# Contingency Table
tblChain3 <- crossTable(ErrChain32wk, sort =TRUE)
tblChain3 [1:52,1:52]


# Chain 1 Correlation
chisq.residuals(tblChain1, digits = 2, std = TRUE, raw = FALSE)
mosaicplot(tblChain1, shade = TRUE, las=2,
           main = "Chain 1 Errors")
cor(tblChain1)
ggcorr(tblChain1, label=TRUE, label_alpha=TRUE)
ggcorr(tblChain1)


# Chain 2 Correlation
chisq.residuals(tblChain2, digits = 2, std = TRUE, raw = FALSE)
mosaicplot(tblChain2, shade = TRUE, las=2,
           main = "Chain 2 Errors")
cor(tblChain2)
ggcorr(tblChain2, label=TRUE, label_alpha=TRUE)
ggcorr(tblChain2)
```

```
# Chain 3 Correlation
chisq.residuals(tblChain3, digits = 2, std = TRUE, raw = FALSE)
mosaicplot(tblChain3, shade = TRUE, las=2,
           main = "Chain 3 Errors")
cor(tblChain3)
ggcorr(tblChain3, label=TRUE, label_alpha=TRUE)
ggcorr(tblChain3)

## Elegant correlation table using xtable R package, Improve output of above cor(x) tables
Chain1CorrRound <-round(cor(tblChain1),2)
Chain2CorrRound <-round(cor(tblChain2),2)
Chain3CorrRound <-round(cor(tblChain3),2)

# - x : is the correlation matrix, - diag : if TRUE the diagonal are not included in the result
lower.tri(Chain1CorrRound, diag = FALSE)
upper.tri(Chain1CorrRound, diag = FALSE)
lower.tri(Chain2CorrRound, diag = FALSE)
upper.tri(Chain2CorrRound, diag = FALSE)
lower.tri(Chain3CorrRound, diag = FALSE)
upper.tri(Chain3CorrRound, diag = FALSE)

#Hide lower triangle – Chain 1
lowerChain1<- Chain1CorrRound
lowerChain1 [lower.tri(Chain1CorrRound, diag=FALSE)]<-""
lowerChain1<-as.data.frame(lowerChain1)
lowerChain1
#Hide lower triangle – Chain 2
lowerChain2<- Chain2CorrRound
lowerChain2 [lower.tri(Chain2CorrRound, diag=FALSE)]<-""
lowerChain2<-as.data.frame(lowerChain2)
lowerChain2
#Hide lower triangle – Chain 3
lowerChain3<- Chain3CorrRound
lowerChain3 [lower.tri(Chain3CorrRound, diag=FALSE)]<-""
lowerChain3<-as.data.frame(lowerChain3)
lowerChain3

tableChain1 <- xtable(lowerChain1)
htmlTable(tableChain1)
tableChain12<- xtable(lowerChain2)
htmlTable(tableChain2)
tableChain13<- xtable(lowerChain3)
htmlTable(tableChain1)
```

Appendix N ERROR CODE CORRELATION MATRIX

Table 33 - C1T01 Pearson Correlation Matrix

C1T01	E27	E29	E17	E20	E22	E23	E24	E25
E27		-0.02	-0.03	0.37**	-0.02	-0.02	-0.02	0.65****
E29			0.71****	-0.03	-0.02	-0.02	-0.02	-0.04
E17				-0.04	-0.02	-0.02	-0.02	-0.04
E20					-0.03	-0.03	-0.03	0.86****
E22						-0.02	-0.02	-0.03
E23							-0.02	-0.03
E24								-0.03
E25								

$p < .0001$ (****); $p < .001$ (***), $p < .01$ (**), $p < .05$ (*)

Table 34 - C1T05 Pearson Correlation Matrix

C1T05	E54	E60
E54		1.00****
E60		

$p < .0001$ (****); $p < .001$ (***), $p < .01$ (**), $p < .05$ (*)

Table 35 - C1T07 Pearson Correlation Matrix

C1T07	**E08**	**E40**	**E14**	**E54**	**E16**	**E12**	**E45**	**E38**	**E01**	**E10**	**E55**	**E59**	**E15**	**E35**	**E36**	**E60**	**E03**	**E37**	**E50**	**E53**
E13	0.03	0.13	-0.02	-0.02	-0.02	0.19	0.11	-0.02	-0.02	-0.02	-0.02	-0.03	-0.01	-0.03	-0.02	-0.03	-0.02	-0.03	-0.02	-0.02
E08		-0.02	0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.03	-0.03	-0.02	-0.02	-0.03	-0.02	-0.03	-0.02	-0.02
E40			-0.02	-0.02	0.16	0.11	-0.01	-0.02	-0.02	-0.02	-0.02	-0.03	0.60 ****	-0.03	-0.02	-0.03	-0.02	-0.03	-0.02	-0.02
E14				-0.02	-0.02	-0.02	-0.02	0.17	-0.02	-0.02	-0.02	-0.03	-0.03	0.46 ***	-0.02	-0.03	-0.02	-0.03	-0.02	-0.02
E54					-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.03	-0.03	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02
E16						-0.01	-0.02	-0.03	-0.02	-0.02	-0.02	-0.04	0.72 ****	-0.03	-0.02	-0.03	-0.02	-0.03	-0.02	-0.02
E12							0.00	-0.03	-0.02	-0.02	-0.02	-0.04	0.02	-0.03	-0.02	-0.03	-0.02	-0.03	-0.02	-0.02
E45								-0.02	-0.02	-0.02	-0.02	-0.03	-0.03	-0.03	-0.02	-0.03	-0.02	-0.03	-0.02	-0.02
E38									-0.02	-0.02	-0.02	-0.03	-0.04	0.05	-0.02	-0.03	-0.02	-0.03	-0.02	-0.02
E01										-0.02	-0.02	-0.03	-0.03	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02
E10											-0.02	-0.03	-0.03	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02
E55												-0.03	-0.03	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02
E59													-0.05	-0.04	-0.03	0.94 ****	-0.03	0.75 ****	-0.03	-0.03
E15														-0.04	-0.03	-0.04	-0.03	-0.04	-0.03	-0.03
E35															-0.02	-0.03	-0.02	-0.03	-0.02	-0.02
E36																-0.02	-0.02	-0.02	-0.02	-0.02
E60																	-0.02	0.48 ***	-0.02	-0.02
E03																		-0.02	-0.02	-0.02
E37																			-0.02	-0.02
E50																				-0.02
E53																				

p < .0001 (****); p < .001 (***), p < .01 (**), p < .05 (*)

Table 36 - C1T08 Pearson Correlation Matrix

C1T08	E13	E08	E14	E12	E01	E54	E18	E16	E55	E15	E45	E10	E69	E38	E59	E68	E35	E49	E53
E40	0.12	0.01	0.01	0.23	0.23	0.00	-0.02	0.06	0.13	0.02	0.04	-0.02	-0.02	0.15	0.07	-0.02	-0.02	0.15	-0.02
E13		0.03	0.02	0.21	0.03	0.02	0.00	0.04	0.06	-0.02	0.09	-0.02	-0.02	0.07	0.15	-0.02	0.08	0.15	0.09
E08			0.00	-0.03	0.04	0.00	0.01	0.03	-0.02	-0.02	-0.02	-0.02	0.06	0.06	-0.02	-0.02	0.08	-0.03	-0.02
E14				-0.03	-0.02	0.03	-0.02	-0.02	-0.02	-0.02	0.04	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.03	-0.02
E12					0.40**	0.08	-0.03	0.26*	0.10	0.08	-0.01	-0.03	0.06	0.02	0.01	0.07	-0.02	0.70****	-0.02
E01						-0.01	0.01	0.10	0.01	-0.01	-0.02	0.04	-0.01	0.10	-0.01	-0.01	-0.02	0.12	-0.03
E54							-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	0.28*	-0.02
E18								-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	0.08	-0.02	-0.02	0.10	-0.03	-0.02
E16									-0.01	-0.01	-0.02	-0.03	0.18	-0.02	-0.02	0.09	-0.02	0.07	-0.02
E55										0.13	-0.01	-0.03	-0.03	0.11	-0.01	-0.02	-0.02	0.03	-0.02
E15											-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.03	0.01	-0.02
E45												-0.02	-0.03	-0.02	-0.01	-0.03	-0.02	-0.02	-0.01
E10													-0.03	-0.03	-0.03	-0.02	-0.02	-0.04	-0.02
E69														-0.03	-0.03	0.15	-0.02	0.00	-0.03
E38															-0.01	-0.03	-0.01	-0.01	-0.02
E59																-0.03	-0.01	0.00	-0.01
E68																	-0.03	0.01	-0.02
E35																		-0.03	-0.01
E49																			-0.02

(C1T08 related table continued on the next page.)

C1T08	E02	E11	E03	E19	E42	E50	E65	E70
E40	-0.02	-0.03	0.35**	0.00	-0.02	-0.02	-0.02	-0.02
E13	-0.02	-0.03	0.00	0.39**	-0.02	-0.02	-0.02	-0.02
E08	-0.02	-0.02	-0.03	-0.02	0.40**	-0.02	-0.02	-0.02
E14	-0.02	-0.02	-0.03	-0.02	-0.03	-0.02	-0.02	-0.02
E12	-0.03	-0.04	0.48***	0.03	0.03	-0.03	-0.03	-0.03
E01	-0.03	-0.03	0.48***	-0.03	0.40**	-0.03	-0.03	-0.03
E54	-0.02	-0.03	-0.02	-0.02	-0.03	-0.02	-0.02	-0.02
E18	-0.02	-0.02	-0.03	-0.02	-0.01	-0.02	-0.02	-0.02
E16	-0.03	-0.03	0.44***	-0.02	0.00	-0.03	-0.03	-0.03
E55	-0.02	-0.03	0.03	-0.01	-0.04	-0.02	-0.02	-0.02
E15	-0.02	-0.03	0.00	-0.03	-0.03	-0.02	-0.02	-0.02
E45	-0.02	-0.03	-0.03	0.01	-0.03	-0.02	-0.02	-0.02
E10	-0.02	0.35**	-0.02	-0.03	-0.01	-0.02	-0.02	-0.02
E69	-0.02	-0.03	0.03	-0.04	-0.01	-0.02	-0.02	-0.02
E38	-0.03	-0.03	0.03	-0.01	0.02	-0.03	-0.03	-0.03
E59	-0.02	-0.03	-0.02	0.52****	-0.04	-0.02	-0.02	-0.02
E68	-0.02	-0.03	0.01	-0.04	-0.04	-0.02	-0.02	-0.02
E35	-0.02	-0.03	-0.05	0.00	0.00	-0.02	-0.02	-0.02
E49	-0.03	-0.04	0.21	0.01	-0.05	-0.03	-0.03	-0.03
E53	-0.02	-0.02	-0.04	0.02	-0.03	-0.02	-0.02	-0.02
E02		-0.02	-0.04	-0.03	-0.03	-0.02	-0.02	-0.02
E11			-0.05	-0.04	-0.04	-0.02	-0.02	-0.02
E03				-0.06	0.09	-0.04	-0.04	-0.04
E19					-0.05	-0.03	-0.03	-0.03
E42						-0.03	-0.03	-0.03
E50							-0.02	-0.02
E65								-0.02
E70								

$p < .0001$ (****); $p < .001$ (***), $p < .01$ (**), $p < .05$ (*)

Table 37 - C2T16 Pearson Correlation Matrix

C2T16	E54	E103	E100	E47	E101	E97	E56	E99	E48	E50	E71	E96	E53	E63	E57
E54		0.09	0.10	0.19	0.19	0.09	0.04	0.39***	0.19	0.07	0.15	0.34**	0.53****	0.40***	-0.02
E103			-0.01	0.10	0.00	-0.01	-0.01	0.14	0.00	0.08	0.27*	0.13	0.02	0.01	-0.02
E100				0.00	0.00	-0.01	-0.01	0.02	-0.01	-0.01	-0.01	0.02	0.03	0.02	-0.02
E47					0.08	0.00	0.07	0.08	0.66****	0.00	0.27*	0.05	0.31*	0.06	-0.02
E101						0.00	-0.01	0.06	0.03	0.00	0.01	0.05	0.09	0.06	-0.02
E97							0.20	0.02	-0.01	-0.01	-0.01	0.01	0.02	0.01	-0.02
E56								-0.01	0.01	-0.02	-0.01	-0.01	0.01	-0.01	-0.02
E99									0.27*	0.02	0.08	0.12	0.20	0.13	-0.02
E48										-0.01	0.35**	0.04	0.36**	0.07	-0.03
E50											0.01	0.02	0.01	0.01	-0.02
E71												0.06	0.11	0.03	-0.03
E96													0.15	0.11	-0.02
E53														0.80****	-0.03
E63															-0.02

Table 38 - C2T15 Pearson Correlation Matrix

C2T15	E116	E101	E87	E110	E85	E83	E102	E115	E91	E88	E117	E99	E89	E92	E90	E82
E116		0.03	0.05	-0.02	-0.01	0.02	-0.01	-0.02	-0.02	-0.01	-0.02	-0.02	0.24	-0.02	-0.02	-0.02
E101			0.09	0.19	0.64****	0.33**	0.30*	0.05	0.02	0.15	-0.02	0.50****	-0.02	0.01	0.00	-0.02
E87				0.16	0.01	0.27*	0.47****	-0.02	0.15	-0.01	-0.03	0.02	-0.01	0.63****	0.33**	-0.02
E110					0.05	0.24*	0.06	-0.02	0.60****	0.00	-0.03	0.04	-0.03	0.27*	0.00	-0.02
E85						0.23	0.10	0.01	-0.01	0.06	-0.03	0.22	-0.03	-0.03	-0.03	-0.02
E83							0.34**	-0.02	0.49****	0.01	-0.03	0.10	-0.03	0.31**	0.04	-0.03
E102								-0.01	0.06	0.01	-0.03	0.28*	-0.03	0.38**	0.38**	-0.03
E115									-0.03	-0.01	0.24*	0.23	-0.02	-0.04	-0.03	-0.02
E91										-0.03	-0.03	-0.05	-0.03	0.40***	-0.01	-0.03
E88											-0.02	0.06	-0.02	-0.04	-0.03	-0.02
E117												0.00	-0.02	-0.04	-0.03	-0.02
E99													-0.03	-0.02	0.01	-0.02
E89														-0.04	-0.03	-0.02
E92															0.19	-0.03
E90																-0.02

$p < .0001$ (****); $p < .001$ (***), $p < .01$ (**), $p < .05$ (*)

Table 39 - C2T05 Pearson Correlation Matrix

C2T05	E20	E15	E28	E16	E06	E09	E11	E19	E08	E22	E30	E12	E40	E31	E44	E39	E21	E13	E38	E41	E17	E14	E42
E20		0.17	0.09	0.10	0.10	0.11	0.05	0.01	0.16	0.07	0.09	0.04	-0.04	0.08	0.06	-0.03	0.07	0.12	-0.03	0.00	-0.02	-0.02	-0.02
E15			0.04	0.41***	0.04	0.12	0.06	0.06	0.00	0.23	-0.01	0.04	-0.04	0.25*	0.20	-0.03	-0.02	-0.02	-0.04	-0.02	-0.02	-0.02	-0.02
E28				0.06	0.03	0.12	-0.02	0.07	0.12	-0.02	0.51****	0.04	0.12	-0.02	0.14	0.08	-0.03	0.29*	0.28*	0.71****	-0.02	-0.02	-0.02
E16					0.01	0.17	0.09	-0.01	0.02	0.02	0.04	-0.01	-0.05	0.03	0.01	-0.03	-0.04	-0.03	-0.04	0.00	-0.02	-0.03	-0.02
E06						0.04	-0.02	0.13	-0.01	0.09	0.00	0.05	0.06	0.19	0.17	-0.02	-0.01	0.00	-0.01	0.00	-0.02	0.00	-0.02
E09							0.02	0.09	0.04	-0.01	0.20	-0.02	-0.02	0.00	0.01	-0.03	-0.03	0.00	0.00	0.04	-0.02	0.00	-0.02
E11								-0.03	0.23	0.04	-0.02	0.29*	0.04	-0.02	-0.03	-0.03	-0.01	-0.04	-0.05	-0.03	-0.02	-0.03	-0.02
E19									-0.03	0.27*	0.09	-0.02	0.64****	0.36**	0.62****	0.04	0.16	0.43***	0.55****	0.01	-0.03	0.67****	-0.03
E08										-0.03	0.11	0.18	0.06	-0.03	-0.03	-0.02	-0.02	0.01	-0.01	0.05	-0.02	-0.03	-0.02
E22											-0.04	0.17	0.29*	0.45***	0.51****	0.28*	0.64****	0.22	0.07	-0.05	-0.03	0.07	-0.03
E30												-0.01	0.14	-0.03	0.15	0.02	-0.03	0.32**	0.30*	0.31**	-0.03	-0.01	-0.03
E12													0.00	0.01	0.00	0.00	0.15	-0.02	-0.05	0.01	-0.03	-0.04	-0.03
E40														0.13	0.48****	0.47****	0.50****	0.56****	0.60****	0.05	-0.04	0.31*	-0.04
E31															0.60****	0.00	0.10	0.13	0.11	-0.05	-0.03	0.14	-0.03
E44																0.05	0.17	0.58****	0.58****	0.05	-0.04	0.27*	-0.04
E39																	0.49****	0.09	0.06	0.04	-0.03	-0.04	-0.03
E21																		0.13	0.06	-0.04	-0.03	0.03	-0.03
E13																			0.77****	0.16	-0.04	0.16	-0.04
E38																				0.16	-0.04	0.26*	-0.04
E41																					-0.02	-0.03	-0.02
E17																						-0.02	-0.01
E14																							-0.02
E42																							

p < .0001 (****); p < .001 (***), p < .01 (**), p < .05 (*)

Table 40 - C2T18 Pearon Correlation Matrix

C2T18	E01	E02	E03
E01		1.00****	-0.02
E02			-0.02
E03			

p < .0001 (****); p < .001 (***), p < .01 (**), p < .05 (*)

Table 41 - C3T1 Pearon Correlation Matrix

C3T1	E22	E21	E09	E14	E12	E19	E13	E17	E07	E23	E20	E05	E15	E18	E01	E02	E28	E29
E22		0.02	-0.02	-0.05	-0.03	-0.04	-0.04	-0.04	0.11	0.06	-0.04	-0.03	-0.05	-0.04	-0.07	-0.09	-0.04	-0.04
E21			-0.01	-0.05	-0.01	-0.04	-0.05	-0.05	0.15	0.72****	-0.04	-0.03	-0.05	-0.04	-0.08	-0.08	-0.04	-0.04
E09				-0.05	0.00	-0.05	-0.05	-0.06	0.08	0.03	-0.05	-0.03	-0.06	-0.05	-0.09	-0.11	-0.05	-0.05
E14					0.44**	-0.05	0.86****	-0.06	-0.08	-0.08	-0.05	-0.04	0.39**	-0.05	0.46***	-0.19	-0.05	-0.05
E12						-0.08	0.21	-0.09	0.13	0.05	-0.08	-0.06	0.65****	-0.08	0.59****	-0.16	-0.08	-0.08
E19							-0.04	0.53****	-0.06	-0.06	0.39**	-0.03	-0.05	0.12	-0.08	-0.15	-0.04	-0.04
E13								-0.05	-0.07	-0.07	-0.04	-0.03	0.42**	-0.04	0.21	-0.16	-0.04	-0.04
E17									-0.07	-0.08	0.39**	-0.03	-0.06	0.73****	-0.09	-0.18	-0.05	-0.05
E07										0.46***	-0.06	-0.04	-0.08	-0.06	-0.12	0.07	-0.06	-0.06
E23											-0.06	-0.04	-0.08	-0.06	-0.12	-0.03	-0.06	-0.06
E20												-0.03	-0.05	0.10	-0.08	-0.16	-0.04	-0.04
E05													-0.04	-0.03	-0.06	-0.11	-0.03	-0.03
E15														-0.05	0.22	-0.20	-0.05	-0.05
E18															-0.08	-0.15	-0.04	-0.04
E01																-0.31*	-0.08	-0.08
E02																	-0.16	-0.16
E28																		1.00****
E29																		

p < .0001 (****); p < .001 (***), p < .01 (**), p < .05 (*)

Table 42 - C3T2, T3, T4, T6 - Pearson Correlation Matrices

C3T2	E05	E06	E03
E05		0.45***	0.01
E06			0.45***
E03			

p < .0001 (****); p < .001 (***), p < .01 (**), p < .05 (*)

C3T6	E61	E63	E62
E61		0.92****	0.71****
E63			0.82****
E62			

p < .0001 (****); p < .001 (***), p < .01 (**), p < .05 (*)

C3T3	E60
E60	-

p < .0001 (****); p < .001 (***), p < .01 (**), p < .05 (*)

C3T4	E59
E59	-

p < .0001 (****); p < .001 (***), p < .01 (**), p < .05 (*)

Table 43 - C3T5 Pearson Correlation Matrix

C3T5	E30	E53	E54	E51	E36	E46	E47	E37	E40	E48	E35	E39	E41	E42	E43	E44	E45	E49	E52	E38
E30		0.46 ***	0.59 ****	0.21	0.07	0.14	0.07	0.07	0.09	0.09	0.09	0.09	0.09	0.09	0. 09	0.09	0.09	0.09	0.09	-0.04
E53			0.73 ****	0.18	0.13	0.19	0.13	0.14	0.16	0.16	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	-0.05
E54				0.23	0.17	0.24	0.17	0.18	0.20	0.20	0.21	0.21	0.21	0. 21	0. 21	0. 21	0. 21	0. 21	0. 21	-0.06
E51					0.79 ****	0.77 ****	0.79 ****	0.82 ****	0.86 ****	0.86 ****	0.87 ****	0.87 ****	0.87 ****	0.87 ****	0.87 ****	0.87 ****	0.87 ****	0.87 ****	0.87 ****	0.02
E36						0.80 ****	0.84 ****	0.87 ****	0.91 ****	0.91 ****	0.92 ****	0.92 ****	0.92 ****	0.92 ****	0.92 ****	0.92 ****	0.92 ****	0.92 ****	0.92 ****	0.04
E46							0.80 ****	0.82 ****	0.86 ****	0.86 ****	0.88 ****	0.88 ****	0.88 ****	0.88 ****	0.88 ****	0.88 ****	0.88 ****	0.88 ****	0.88 ****	0.02
E47								0.87 ****	0.91 ****	0.91 ****	0.92 ****	0.92 ****	0.92 ****	0.92 ****	0.92 ****	0.92 ****	0.92 ****	0.92 ****	0.92 ****	0.04
E37									0.94 ****	0.94 ****	0.95 ****	0.95 ****	0.95 ****	0.95 ****	0.95 ****	0.95 ****	0.95 ****	0.95 ****	0.95 ****	0.35*
E40										0.99 ****	0.99 ****	0.99 ****	0.99 ****	0.99 ****	0.99 ****	0.99 ****	0.99 ****	0.99 ****	0.99 ****	0.05
E48											0.99 ****	0.99 ****	0.99 ****	0.99 ****	0.99 ****	0.99 ****	0.99 ****	0.99 ****	0.99 ****	0.05
E35												1.00 ****	1.00 ****	1.00 ****	1.00 ****	1.00 ****	1.00 ****	1.00 ****	1.00 ****	0.06
E39													1.00 ****	1.00 ****	1.00 ****	1.00 ****	1.00 ****	1.00 ****	1.00 ****	0.06
E41														1.00 ****	1.00 ****	1.00 ****	1.00 ****	1.00 ****	1.00 ****	0.06
E42															1.00 ****	1.00 ****	1.00 ****	1.00 ****	1.00 ****	0.06
E43																1.00 ****	1.00 ****	1.00 ****	1.00 ****	0.06
E44																	1.00 ****	1.00 ****	1.00 ****	0.06
E45																		1.00 ****	1.00 ****	0.06
E49																			1.00 ****	0.06
E52																				0.06
E38																				

p < .0001 (****); p < .001 (***), p < .01 (**), p < .05 (*)

Table 44 - C4T01 Pearson Correlation Matrix

C4T01	E34	E06	E28	E26	E08	E01	E04	E22	E25	E17	E41	E10	E02
E44	0.62****	0.62****	0.57***	0.55***	0.58***	0.59****	0.43**	0.60****	0.46**	0.52***	0.67****	0.44**	0.53***
E13	0.63****	0.64****	0.59****	0.58***	0.63****	0.61****	0.48**	0.68****	0.54***	0.59****	0.67****	0.51***	0.58***
E43	0.66****	0.65****	0.57***	0.57***	0.65****	0.59****	0.47**	0.63****	0.48**	0.53***	0.64****	0.40*	0.50**
E30	0.57***	0.66****	0.59****	0.50**	0.60****	0.57***	0.47**	0.58****	0.55***	0.55***	0.59****	0.48**	0.55***
E20	0.59****	0.57***	0.51***	0.43**	0.55***	0.63****	0.41**	0.55***	0.50**	0.58****	0.50**	0.41**	0.56***
E16	0.62****	0.64****	0.60****	0.54***	0.61****	0.57***	0.60****	0.62****	0.55***	0.55***	0.62****	0.55***	0.56***
E21	0.59****	0.56***	0.50**	0.42**	0.54***	0.62****	0.36*	0.53***	0.52***	0.59****	0.50**	0.39*	0.54***
E03	0.60****	0.59****	0.57***	0.55***	0.58***	0.60****	0.46**	0.61****	0.54***	0.56***	0.57***	0.47**	0.50**
E29	0.62****	0.61****	0.59****	0.60****	0.63****	0.60****	0.49**	0.65****	0.53***	0.57***	0.63****	0.58***	0.55***
E33	0.60****	0.61****	0.56***	0.53***	0.62****	0.53***	0.44**	0.55***	0.42**	0.48**	0.63****	0.43**	0.45**
E46	0.49**	0.47**	0.43**	0.42**	0.48**	0.44**	0.32*	0.46**	0.31	0.33*	0.52***	0.35*	0.27
E09	0.60****	0.61****	0.61****	0.61****	0.62****	0.62****	0.45**	0.66****	0.59****	0.62****	0.62****	0.50**	0.54***
E32	0.60****	0.59****	0.57***	0.62****	0.60****	0.63****	0.45**	0.62****	0.55***	0.58***	0.55***	0.53***	0.55***
E39	0.59****	0.59****	0.57***	0.55***	0.59****	0.55***	0.45**	0.58***	0.47**	0.48**	0.64****	0.44**	0.44**
E34		0.57***	0.57***	0.49**	0.58****	0.50**	0.42**	0.55***	0.44**	0.48**	0.55***	0.41*	0.49**
E06			0.54***	0.54***	0.59****	0.53***	0.45**	0.57***	0.47**	0.53***	0.56***	0.47**	0.49**
E28				0.54***	0.55***	0.57***	0.50**	0.57***	0.60****	0.63****	0.49**	0.64****	0.62****
E26					0.53***	0.63****	0.45**	0.61****	0.59****	0.65****	0.49**	0.64****	0.56***
E08						0.55***	0.46**	0.56***	0.49**	0.64****	0.50**	0.46**	0.50**
E01							0.44**	0.61****	0.70****	0.75****	0.45**	0.71****	0.70****
E04								0.38*	0.45**	0.42**	0.39*	0.45**	0.43**
E22									0.57***	0.71****	0.53***	0.54***	0.60****
E25										0.80****	0.36*	0.79****	0.85****
E17											0.39*	0.75****	0.79****
E41												0.33*	0.34*
E10													0.81****

p < .0001 (****); p < .001 (***), p < .01 (**), p < .05 (*)

(C4T01 related table continued on the next page.)

C4T01	E14	E19	E15	E47	E12	E18	E11	E07	E23	E24	E40	E36
E44	0.44**	0.49**	0.42**	0.57***	0.29	0.27	0.23	0.33*	0.18	0.61****	0.33*	0.50**
E13	0.48**	0.61****	0.50**	0.66****	0.39*	0.38*	0.34*	0.48**	0.29	0.53***	0.56***	0.53***
E43	0.43**	0.50**	0.37*	0.63****	0.30	0.28	0.19	0.35*	0.18	0.64****	0.26	0.56***
E30	0.50**	0.55***	0.50**	0.58****	0.44**	0.37*	0.31	0.41**	0.35*	0.57***	0.33*	0.56***
E20	0.42**	0.48**	0.35*	0.49**	0.30	0.30	0.15	0.28	0.25	0.69****	0.36*	0.34*
E16	0.57***	0.53***	0.46**	0.54***	0.42**	0.39*	0.33*	0.48**	0.30	0.51**	0.24	0.40*
E21	0.42**	0.43**	0.35*	0.44**	0.30	0.28	0.18	0.33*	0.19	0.61****	0.26	0.34*
E03	0.44**	0.45**	0.44**	0.60****	0.41**	0.34*	0.27	0.40*	0.16	0.58****	0.26	0.63****
E29	0.53***	0.57***	0.50**	0.59****	0.42**	0.40*	0.37*	0.49**	0.30	0.58***	0.26	0.64****
E33	0.46**	0.46**	0.32*	0.60****	0.26	0.22	0.13	0.36*	0.16	0.51***	0.23	0.42**
E46	0.24	0.35*	0.23	0.41*	0.13	0.09	0.01	0.29	0.11	0.60****	0.34*	0.32*
E09	0.50**	0.62****	0.50**	0.58***	0.43**	0.46**	0.38*	0.45**	0.31	0.59****	0.25	0.64****
E32	0.56***	0.55***	0.43**	0.54***	0.44**	0.42**	0.30	0.47**	0.39*	0.72****	0.19	0.41**
E39	0.49**	0.48**	0.38*	0.56***	0.29	0.28	0.24	0.36*	0.26	0.58****	0.20	0.64****
E34	0.40*	0.46**	0.38*	0.54***	0.28	0.22	0.15	0.25	0.27	0.63****	0.21	0.42**
E06	0.47**	0.55***	0.44**	0.52***	0.32*	0.34*	0.25	0.43**	0.30	0.50**	0.38*	0.45**
E28	0.58****	0.53***	0.56***	0.50**	0.54***	0.51***	0.47**	0.50**	0.44**	0.54***	0.17	0.42**
E26	0.52***	0.59****	0.52***	0.45**	0.51***	0.53***	0.48**	0.56***	0.43**	0.45**	0.14	0.43**
E08	0.55***	0.61****	0.47**	0.51***	0.41**	0.37*	0.27	0.32*	0.39*	0.51***	0.19	0.45**

E01	0.60****	0.56***	0.57***	0.45**	0.59****	0.62****	0.49**	0.63****	0.51***	0.59****	0.17	0.40*
E04	0.45**	0.53***	0.40*	0.37*	0.41**	0.45**	0.33*	0.27	0.45**	0.33*	0.08	0.33*
E22	0.51***	0.60****	0.49**	0.55***	0.47**	0.44**	0.39*	0.65****	0.29	0.50**	0.38*	0.48**
E25	0.74****	0.58***	0.81****	0.33*	0.88****	0.81****	0.77****	0.69****	0.68****	0.39*	0.06	0.36*
E17	0.66****	0.71****	0.68****	0.43**	0.72****	0.70****	0.63****	0.67****	0.60****	0.43**	0.12	0.38*
E41	0.35*	0.46**	0.29	0.51***	0.19	0.14	0.13	0.25	0.11	0.49**	0.26	0.50**
E10	0.75****	0.57***	0.78****	0.32*	0.82****	0.81****	0.78****	0.76****	0.71****	0.34*	0.07	0.28
E02	0.78****	0.57***	0.77****	0.37*	0.81****	0.80****	0.73****	0.65****	0.66****	0.40*	0.12	0.29
E14		0.54***	0.77****	0.29	0.78****	0.72****	0.75****	0.63****	0.68****	0.32*	0.01	0.29
E19			0.55***	0.42**	0.55***	0.57***	0.48**	0.41*	0.62****	0.37*	0.17	0.60****
E15				0.24	0.87****	0.83****	0.84****	0.66****	0.72****	0.24	0.06	0.31
E47					0.17	0.15	0.09	0.24	0.06	0.47**	0.26	0.49**
E12						0.92****	0.89****	0.68****	0.80****	0.19	-0.05	0.24
E18							0.87****	0.69****	0.79****	0.17	-0.04	0.23
E11								0.68****	0.75****	0.04	-0.07	0.22
E07									0.43**	0.23	0.26	0.22
E23										0.14	-0.09	0.12
E24											0.19	0.41**

$p < .0001$ (****); $p < .001$ (***), $p < .01$ (**), $p < .05$ (*)

Appendix O FREQUENT PATTERN – GROWTH OUTPUT

Frequent Pattern (FP) Graphs of Chain 3 and Chain 4 are shown in the figures below, which show the frequent error patterns of respective chains based on the final data set of single visit cases. These overviews validate the Pearson correlation calculations between distinct errors within chains. More detailed output, in terms of metrics, are provided in the corresponding section in the report, while the figures below only show the confidence values per found association. No error – sequence patterns were found for Chain 1 and Chain 2.

=== Run information === Relation: *error matrix .csv file* Instances: 10518 *number of cases* Attributes: 200 *number of distinct errors* === Associator model (full set) === FPGrowth found 16 rules (displaying top 16)

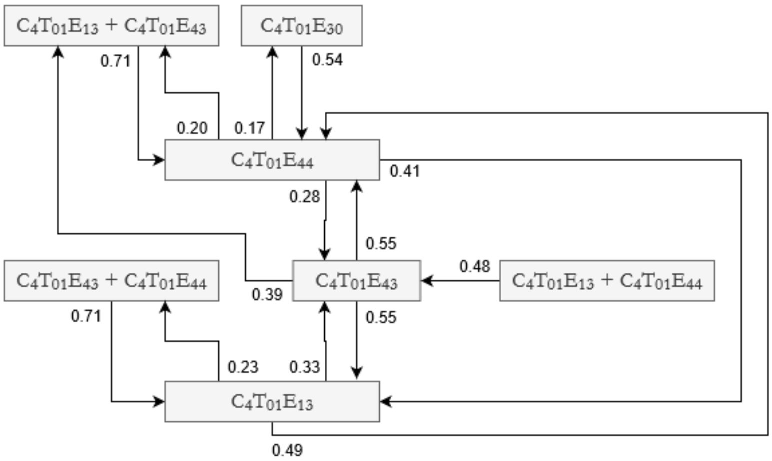


Fig. 35 - FP Growth Graph, Chain 4 - Single Visit Observed Errors (n=16 rules) (re-labeled)

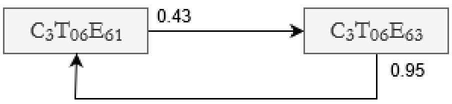


Fig. 36 - FP Growth Graph, Chain 3 - Single Visit Observed Errors (n=16 rules) (re-labeled)

Appendix P Error Distribution & Logfile Availability

From the *Section 5.9* we know about error' behavior; if different errors are correlated, or independent from each other, and how to cluster certain errors. However, error distribution is also important; if any out of the ordinary happens with the amount of distinct errors over time. This section focusses on the failures per chain over time, and what potential causes can be for striking observations.

Error Distribution – Failures over time

Table 45 includes the number of cases within the scope of the project per chain along with the amount of distinct chain related errors occurred per year. Again, note that the distinct errors are observed over a two-week period before any case open date. In terms of absolute numbers nothing strange can be observed, only that the total number of distinct errors increases slightly for chain 1, almost doubles for chain 2, and increases drastically for the remaining two chains. However, this seems to be in line with (similar factor) increase in number of cases for corresponding chain. Normalizing this data and determining the average number of distinct errors per case, provides a more interesting insight. The first three chains have a (relative) steady average over time (1.1, 1.2, and 1.6 respectively), but the last chain has a sudden factor two increase for the years 2016 and 2017. This is a surprising observation and needs further investigation. Two potential factors contributing to more observed (distinct) errors have been identified together with SME's, based on our data sources: 1) Log File Availability, and 2) Software Release related causes; which will be the focus of the next section.

Table 45 - Case, Distinct Error Count, and Avg. Errors per Chain ~ Year

Count of Cases with Chain X Errors (Number of Occurred Chain X Errors) [Avg. #Distinct Chain X Errors per Case]												
	Chain 1			Chain 2			Chain 3			Chain 4		
2013	2.1%	(2.2%)	[1.1]	1.1%	(1.2%)	[1.3]	1.5%	(1.5%)	[1.5]	3.6%	(2.3%)	[1.4]
2014	12%	(12%)	[1.1]	5.6%	(6.8%)	[1.2]	7.5%	(7.8%)	[1.6]	8.7%	(5.9%)	[1.4]
2015	0.8%	(0.8%)	[1.1]	16%	(15%)	[1.1]	12%	(11%)	[1.5]	12%	(7.2%)	[1.3]
2016	24%	(24%)	[1.1]	30%	(29%)	[1.1]	19%	(19%)	[1.5]	30%	(36%)	[2.5]
2017	29%	(27%)	[1.1]	25%	(26%)	[1.2]	29%	(30%)	[1.6]	39%	(44%)	[2.4]
2018	24%	(28%)	[1.1]	16%	(17%)	[1.2]	23%	(22%)	[1.6]	5.6%	(4%)	[1.5]
2019	7.7%	(7%)	[1.1]	5.7%	(4.9%)	[1.2]	8%	(8.2%)	[1.6]	1.2%	(0.9%)	[1.6]
Total	100% (100%)			100% (100%)			100% (100%)			100% (100%)		

Logfile Availability

In order to determine the log file availability per case (and corresponding) system, using Vertica table *"MR_etl5_dailylog"*, the number of daily logs are counted during the same two-week period in which error data was analyzed. As we are interested how complete the error data is, we only want to know the amount of days with uploaded log files. This is important, as MR systems can upload multiple log files per day. Hence, for each case and system log file availability rate is determined based on a maximum of 14 days (*Fig. 37*).

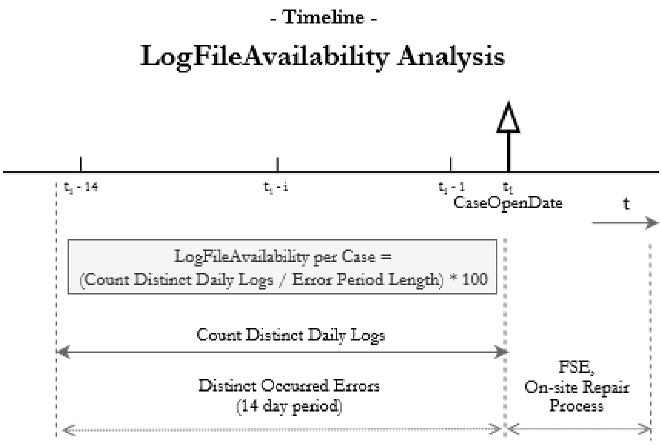


Fig. 37 - Timeline Analyzed LogFileAvailability

Unfortunately, daily log and error data in Vertica have different data sources. For an unknown reason, daily log data is not always pushed to or pulled from the corresponding table, appearing that no log files have been registered for a certain day, while in fact error data has been retrieved for those days (originated from another data source). Resulting in only 16k of 17k cases in the final dataset having daily log related data available, hence further conclusions are based on a more than acceptable 94.00% of total log file availability. Most of the cases have 100% daily log count, as data has been logged for all fourteen days (*Fig. 38*). Overall, average daily log count equals 89.22%. Detailed numbers, per country and market, are available in *o*.

Based on these numbers, an increase in logged (distinct) errors can potentially be explained by the overall increased log file availability (of 5-10%) in the years 2016 and 2017, and (significant) increases of these numbers for certain markets. A clear negative trend is observed in average missing daily log files overall, with only (on average) 1 day with missing data for these years. To understand which markets specifically contribute to the (factor 2) increase of average distinct errors per case, see *o* for specifics.

As a second contributing factor, it is hypothesized that the sudden increase in average number of chain 4 errors is due to software releases; introduced or installed at the time of errors observed. This analysis is based on 12k out of 17k cases as historic release numbers per SRN over time are only available in "MR_*etl5_dailylog*", which only contains release information from 2016-2019. "*ISDA_medical_view*" is explicitly not used, as it only provides the latest/current release version for each system. This data is preprocessed and aggregated such that 'service packs' are not considered, along with a few cases with two or more observed

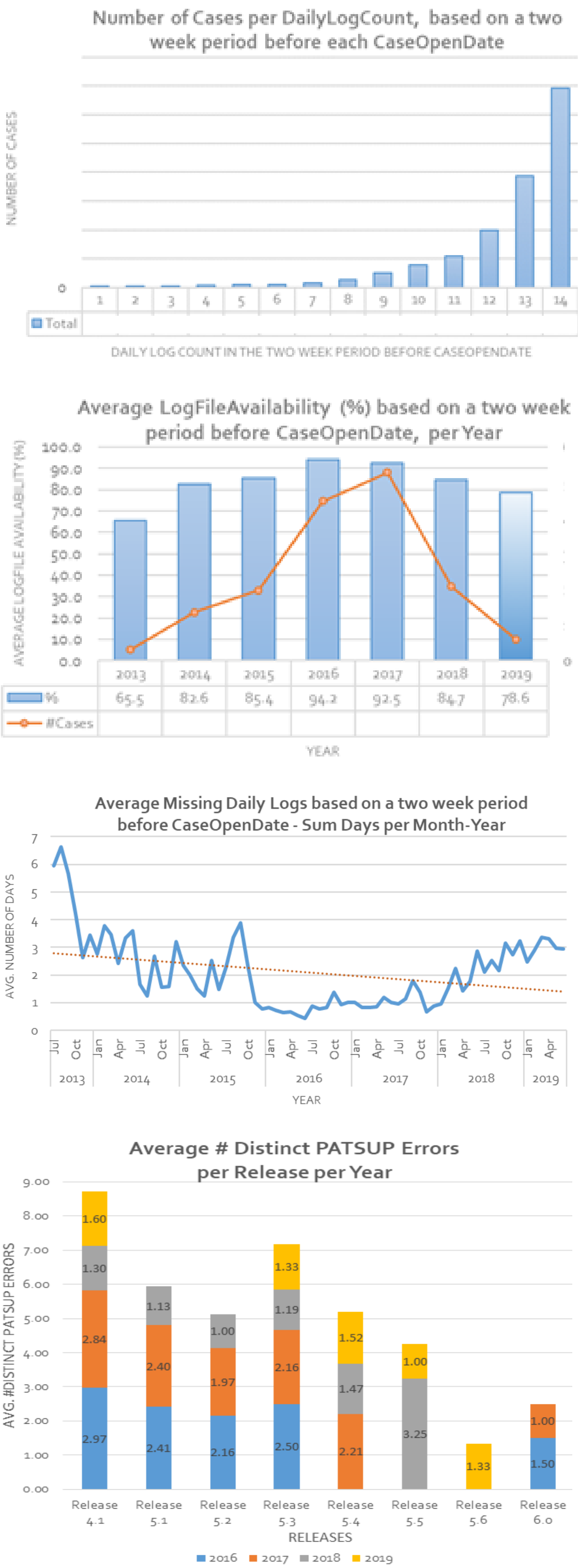


Fig. 38 - Log file Availability - Error Codes

releases within the analyzed time period (6%), and consolidated region-specific releases which are labeled with a different release, but effectively the same as another release number. *Fig. 39* shows a decrease in average distinct chain 4 errors per case, as new software releases are introduced; a decline from 2.89 to 1.25. Also within a release, a decline is observed as new versions or service packs are released. Since these releases are mostly *Ingenia* related we could conclude, that new software results in fewer chain 4 errors per case.

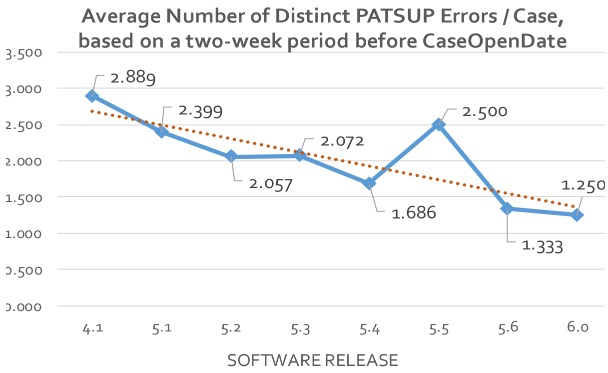


Fig. 39 - Errors per Software Release

For the years of 2016 and 2017, v4.1, v5.1, and to some extend v5.2 were dominant in the software distribution over MR systems in our dataset (while having the highest average number of errors on case-basis), while higher releases and amount of corresponding cases were (almost) nonexistent is this time-period. More recently (2018-2019), the amount of chain 4 errors and cases drastically decreases, while newer releases are introduced and corresponding errors occur, as depicted in *Fig. 38*. Although we can conclude that software release can be a contributing factor to increased average error in 2016-2017, nothing explicit can be said about the average amount of distinct errors per case, per year, for each release. On aggregate, this stays relatively constant over time.

Appendix Q Log File Availability per Market & Country

[Appendix unavailable due to confidential content]

Appendix R AVERAGE DISTINCT C4 ERROR PER CASE, PER MARKET

[Appendix unavailable due to confidential content]

Appendix S FINAL PART CLUSTERS & CLASS IMBALANCE

Table 46 - PartCluster (Class) Distribution per Data(sub)set

Defined Part Cluster	New Label	# Single Cases in Final Dataset				
		All Cases	Chain 1 Cases	Chain 2 Cases	Chain 3 Cases	Chain 4 Cases
	PartCluster1	144	8	17	23	54
	PartCluster10	686	38	81	107	302
	PartCluster11	283	4	101	26	60
	PartCluster12	348	17	53	49	146
	PartCluster13	295	19	35	44	132
	PartCluster14	101	3	11	14	47
	PartCluster15	585	27	71	91	280
	PartCluster2	446	64	59	52	165
	PartCluster3	483	23	48	82	223
	PartCluster4	702	52	106	95	316
	PartCluster5	671	22	66	92	357
	PartCluster6	363	18	56	56	164
	PartCluster7	27	-	3	3	13
	PartCluster8	94	1	8	30	19
	PartCluster9	1724	84	228	251	786

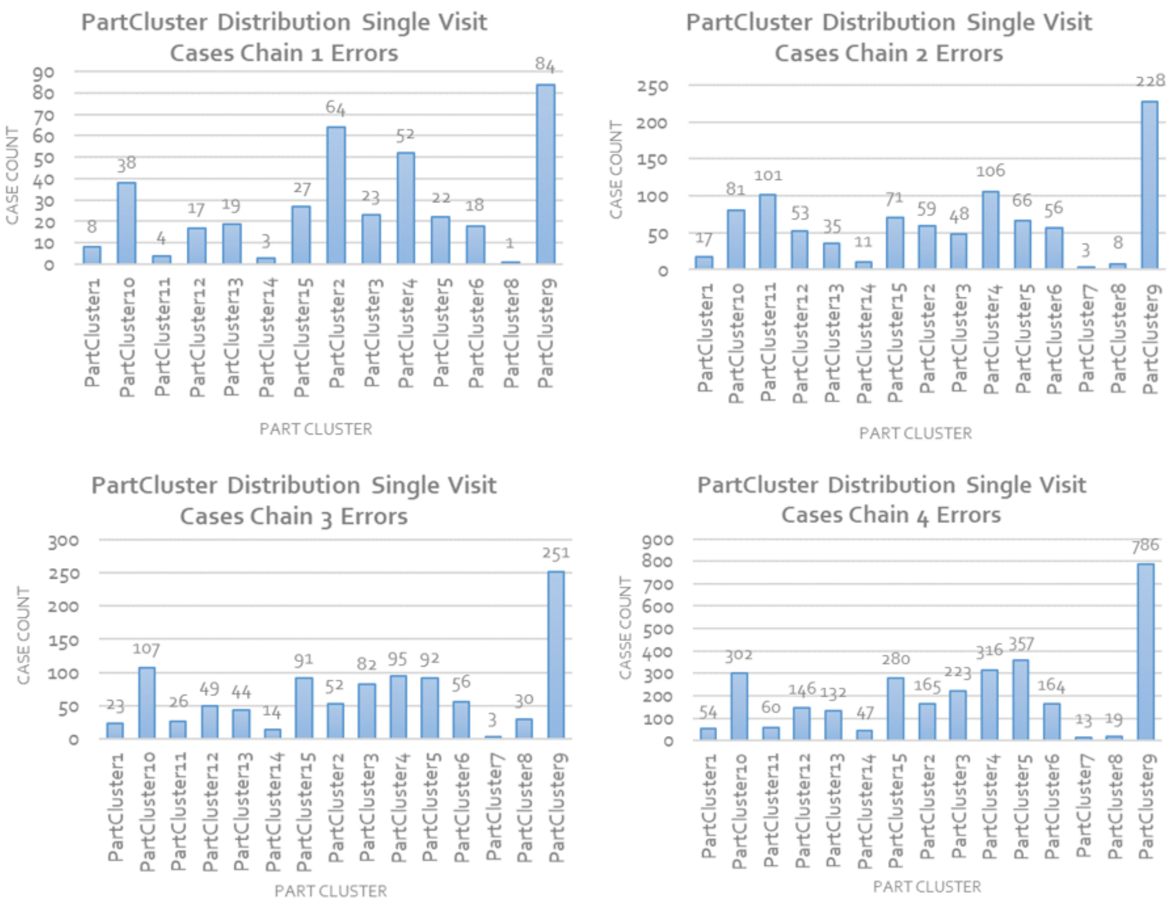


Fig. 40 - Class Imbalance per Chain subset

Appendix T R SCRIPT – RANDOM FOREST

```
library(UBL)
library(DMwR)
library(caret)
library(randomForest)
library(forcats)
library(pacman)
library(party)

# General script for one of the data subsets for cases with errors of a specific chain
dataRFC1ErrorsNewClust <- read.csv("C:/…/XXX.csv", header = TRUE)
str(dataRFC1ErrorsNewClust)
dataRFC1ErrorsNewClust$parts <- as.factor(dataRFC1ErrorsNewClust$parts)
dataRFC1ErrorsNewClust$SystemModel <- as.factor(dataRFC1ErrorsNewClust$SystemModel)
dataRFC1ErrorsNewClust$Priority <- as.factor(dataRFC1ErrorsNewClust$Priority)
dataRFC1ErrorsNewClust$IntervalTotalCost <-
as.factor(dataRFC1ErrorsNewClust$IntervalTotalCost)
dataRFC1ErrorsNewClust$Market <- as.factor(dataRFC1ErrorsNewClust$Market)
dataRFC1ErrorsNewClust$EntitlementType <- as.factor(dataRFC1ErrorsNewClust$EntitlementType)

# Data Partition
set.seed(33)
ind3 <- sample(2, nrow(dataRFC1ErrorsNewClust), replace=TRUE, prob = c(0.7, 0.3))
trainRFC1ErrorNew <- dataRFC1ErrorsNewClust[ind3==1,]
testRFC1ErrorNew <- dataRFC1ErrorsNewClust[ind3==2,]
table(trainRFC1ErrorNew$parts)
trainC1_smote <- SMOTE(parts~., trainRFC1ErrorNew, k=5, perc.over = 6500, perc.under = 7000)
summary(trainC1_smote$parts)
summary(trainRFC1ErrorNew$parts)
trainC1_upSAM <- upSample(trainRFC1ErrorNew, trainRFC1ErrorNew$parts)
summary(trainC1_upSAM$parts)
summary(trainRFC1ErrorNew$parts)
trainC1_underSAM <- downSample(trainRFC1ErrorNew, trainRFC1ErrorNew$parts)
summary(trainC1_underSAM)

trainRFC1New <- droplevels(trainRFC1ErrorNew)
trainRFC1New$parts <- factor(trainRFC1ErrorNew$parts)
str(trainRFC1New)
trainRFC1NewSMOTE <- droplevels(trainC1_smote)
trainRFC1NewSMOTE$parts <- factor(trainRFC1NewSMOTE$parts)
str(trainRFC1NewSMOTE)
trainRFC1NewUPSAM <- droplevels(trainC1_upSAM)
trainRFC1NewUPSAM$parts <- factor(trainRFC1NewUPSAM$parts)
str(trainRFC1NewUPSAM)
trainC1_upSAM
trainRFC1NewUPSAM2 <- trainRFC1NewUPSAM[,-60]
trainRFC1NewDOWNSAM <- droplevels(trainC1_underSAM)
trainRFC1NewDOWNSAM$parts <- factor(trainRFC1NewDOWNSAM$parts)
str(trainRFC1NewDOWNSAM)
trainRFC1NewDOWNSAM2 <- trainRFC1NewDOWNSAM[,-60]
str(trainRFC1NewDOWNSAM2)

# Train Models
rf_C1chainnew <- randomForest(parts~., data=trainRFC1New, ntree = 1000)
print(rf_C1chainnew)
attributes(rf_C1chainnew)
rf_C1chainnewSMOTE <- randomForest(parts~., data=trainRFC1NewSMOTE, ntree = 1000,
na.action=na.roughfix)
print(rf_C1chainnewSMOTE)
attributes(rf_C1chainnewSMOTE)
```

```
rf_C1chainnewUPSAM2 <- randomForest(parts~., data=trainRFC1NewUPSAM2, ntree = 1000)
print(rf_C1chainnewUPSAM2)
attributes(rf_C1chainnewUPSAM2)

# Prediction & Confusion Matrix - train data
p3 <- predict(rf_C1chainnew, trainRFC1New)
p7 <- predict(rf_C1chainnewSMOTE, trainRFC1NewSMOTE)
p9 <- predict(rf_C1chainnewUPSAM2, trainRFC1NewUPSAM2)
confusionMatrix(p3, trainRFC1New$parts)
confusionMatrix(p7, trainRFC1NewSMOTE$parts)
confusionMatrix(p9, trainRFC1NewUPSAM2$parts)

# Prediction & Confusion Matrix - test data
testRFC1ErrorNew2 <-droplevels(testRFC1ErrorNew)
testRFC1ErrorNew2 <- factor(testRFC1ErrorNew2$parts)
testRFC1ErrorNew2Smote <- droplevels(testRFC1ErrorNew)
testRFC1ErrorNew2Smote <- factor(testRFC1ErrorNew2$parts)
testRFC1ErrorNew2Smote <-factor(testRFC1ErrorNew2Smote$parts,
levels=levels(trainRFC1NewSMOTE$parts))
# […] CHECK OCCURING VALUES OR CLASSES PER VARIABLE VIA
table(testRFC1ErrorNew2Smote$parts)
# […]
table(trainRFC1NewSMOTE$Market)

# CHECK LEVEL DIFFERENCE OR BETWEEN TRAIN & TEST SET AND FIX ACCORDINGLY, BELOW EXAMPLES
levels(testRFC1ErrorNew2Smote$parts) <- c(levels(testRFC1ErrorNew2Smote$parts),
"PartCluster1", "PartCluster8")
levels(testRFC1ErrorNew2Smote$SystemModel) <- factor(testRFC1ErrorNew2Smote$SystemModel,
levels=levels(trainRFC1NewSMOTE$SystemModel))
levels(testRFC1ErrorNew2Smote$IntervalTotalCost) <-
c(levels(testRFC1ErrorNew2Smote$IntervalTotalCost), "7")
levels(testRFC1ErrorNew2Smote$IntervalTotalCost) <-
factor(testRFC1ErrorNew2Smote$IntervalTotalCost,
levels=levels(trainRFC1NewSMOTE$IntervalTotalCost))
levels(testRFC1ErrorNew2Smote$Market) <- c(levels(testRFC1ErrorNew2Smote$Market), "6", "8")
# […] SAME LEVEL CHECK OF FACTOR VARIABLES AS ABOVE
testRFC1ErrorNew2UPSAM <- droplevels(testRFC1ErrorNew)
testRFC1ErrorNew2UPSAM <- factor(testRFC1ErrorNew2UPSAM$parts)

p4 <- predict(rf_C1chainnew, testRFC1ErrorNew)
CMC1 <- confusionMatrix(p4, testRFC1ErrorNew$parts)
p8 <- predict(rf_C1chainnewSMOTE, testRFC1ErrorNew2Smote)
CMC1SMOTE <- confusionMatrix(p8, testRFC1ErrorNew2Smote$parts)
p9 <- predict(rf_C1chainnewUPSAM2, testRFC1ErrorNew2UPSAM)
CMC1UP <- confusionMatrix(p9, testRFC1ErrorNew2UPSAM$parts)
as.matrix(CMC1SMOTE, what = "overall")
as.matrix(CMC1SMOTE, what = "classes")

# Error rate of Random Forest
plot(rf_C1chainnew)  # IDEM: / plot(rf_C1chainnewSMOTE)  / plot(rf_C1chainnewUPSAM2)

# Feature Importaince Plot - Vairable Gain
varImpPlot(rf_C1chainnewUPSAM2, sort = T, n.var=20,main="Variable Importance Random Forest C1
Upsampling")
varImpPlot(rf_C1chainnewSMOTE, sort = T, n.var=20,main="Variable Importance Random Forest C1
SMOTE")
varImpPlot(rf_C1chainnew, sort = T, n.var=20,main="Variable Importance Random Forest C1")

ctree_model <-  ctree(parts ~ ., data = trainRFC1NewSMOTECTREE5)
plot(ctree_model, main="Conditional Inference Tree C1 data")
```

Appendix U R Script – XGBoost

```
library(keras)
library(tensorflow)
library(xgboost)
library(ade4)
library(readr)
library(stringr)
library(caret)
library(car)
library(data.table)
library(onehot)
library(dplyr)
library(MLmetrics)
library(mltest)
library(DiagrammeR)

# General script for one of the data subsets for cases with errors of a specific chain
casesC1 <- read.csv("C:/…/XXX.csv", header = TRUE)
features = colnames(casesC1)
set.seed(456)
casesC1$parts<-as.factor(casesC1$parts)
summary(casesC1)

trainC1 <- casesC1
trainObsC1 <- sample(nrow(trainC1), .7*nrow(trainC1), replace = FALSE)
testObsC1 <- sample(nrow(trainC1), .3*nrow(trainC1), replace = FALSE)
train_datC1 <- trainC1[trainObsC1,]
test_datC1 <- trainC1[testObsC1,]
train_labsC1 <- as.numeric(train_datC1$parts)-1
test_labsC1 <- as.numeric(test_datC1$parts)-1

new_trainC1 <- model.matrix(~. +0, data=train_datC1[,-1])
new_testC1 <- model.matrix(~. +0, data=trainC1[testObsC1, -1])
xgb_trainC1 <- xgb.DMatrix(data=new_trainC1, label=train_labsC1)
xgb_testC1 <- xgb.DMatrix(data=new_testC1, label=test_labsC1)

numberOfClassesC1 <- length(unique(casesC1$parts))
numberOfClassesC1
paramsC1 <- list(booster="gbtree", objective ="multi:softprob", num_class=numberOfClassesC1,
eval_metric="mlogloss", eta=0.03, silent = 1)
xgbcvC1 <- xgb.cv(params=paramsC1, data=xgb_trainC1, nrounds=500, nfold=10, showsd=TRUE,
stratified=TRUE, print_every_n=10, early_stop_round=10, maximize=FALSE, prediction=TRUE)

plot(xgbcvC1$evaluation_log$train_mlogloss_mean, main = "Train Log Loss - Mean per XGBoost
Model", xlab="Number of Rounds", ylab="Log Loss Value", type="l", col="blue", ylim=c(0.2,3))
lines(#mlogloss evaluation plots for other chains/datasubsets
legend("topright", legend = c("C1 (Cases w/ C1 Errors)","C2 (Cases w/ C2 Errors)","C3 (Cases
w/ C3 Errors)","C4 (Cases with C4 Errors)","AllChains"),
col=c("blue","red","green","orange","purple"), lty=1:2, cex=0.9)

classification_errorC1 <- function(conf_matC1) {
  conf_matC1=as.matrix(conf_matC1)
  errorC1 = 1-sum(diag(conf_matC1))/sum(conf_matC1)
  return(errorC1)
}
xgb_train_predsC1 <- data.frame(xgbcvC1$pred) %>% mutate(max = max.col(., ties.method="last"),
label = train_labsC1 + 1)
head(xgb_train_predsC1)
```

```
xgb_conf_matC1 <- table(true = (train_labsC1 + 1), pred=xgb_train_predsC1$max)
cat("XGB  Training C1 Classification Error Rate", classification_errorC1(xgb_conf_matC1),
"\n")
xgb_conf_mat_C1_2 <-
confusionMatrix(factor(xgb_train_predsC1$label,levels=1:14),factor(xgb_train_predsC1$max,
levels=1:14), mode = "everything")
print(xgb_conf_mat_C1_2)

xgb_modelC1 <- xgb.train(params = paramsC1, data = xgb_trainC1, nrounds = 1000)
xgb_test_predsC1 <- predict(xgb_modelC1, newdata = xgb_testC1)

xgb_test_outC1 <- matrix(xgb_test_predsC1, nrow = 15, ncol = length(xgb_test_predsC1) / 15)
%>%
  t() %>%
  data.frame() %>%
  mutate(max = max.col(., ties.method = "last"), label = test_labsC1 + 1)

xgb_test_confC1 <- table(true = test_labsC1 + 1, pred = xgb_test_outC1$max)
xgb_test_confC1
cat("XGB Validation C1 Classification Error Rate:", classification_errorC1(xgb_test_confC1),
"\n")

xgb_test_confC1_2 <- confusionMatrix(factor(xgb_test_outC1$label,levels=1:15),
factor(xgb_test_outC1$max,levels=1:15),mode = "everything")
print(xgb_test_confC1_2)
Accuracy(xgb_test_outC1$label, xgb_test_outC1$max)
mlTestC1 <- ml_test(xgb_test_outC1$max, xgb_test_outC1$label, output.as.table = TRUE)
min(mlTestC1$#SELECT REQUIRED METRIC)
max(mlTestC1$#SELECT REQUIRED METRIC)
mean(mlTestC1$#SELECT REQUIRED METRIC)

bstC1 <-xgb_modelC1
xgb.plot.tree(model = bstC1, trees = 0, show_node_id = TRUE)
importance_matrixC1 <- xgb.importance(NULL, model=bstC1,trees = seq(from=1, length.out=100))
xgb.plot.importance(importance_matrixC1, top_n = 20)
```

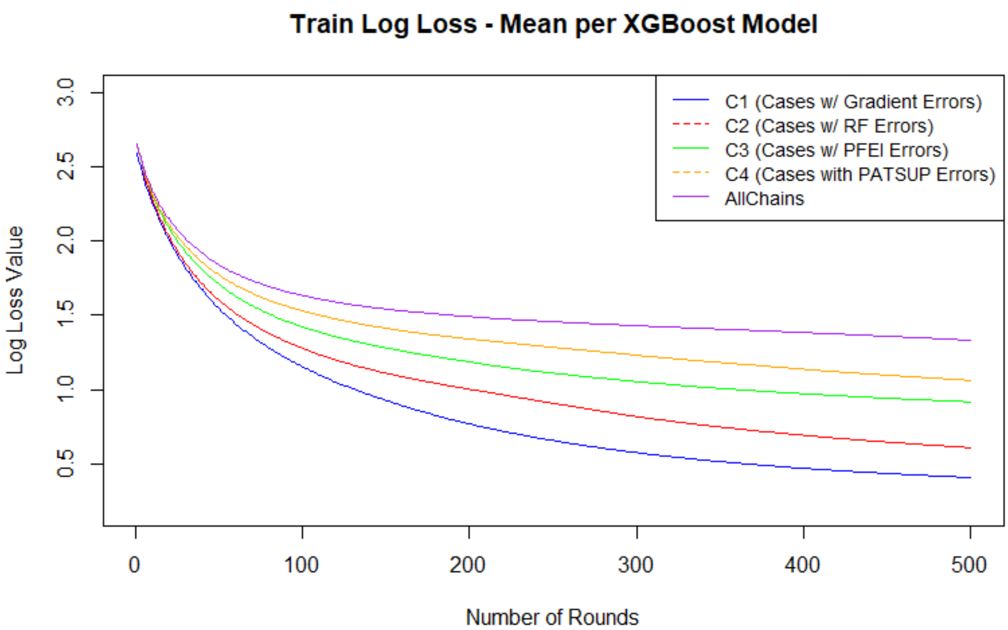


Fig. 41 - Log Loss Training Set XGBoost Models

Appendix V R Script – Support Vector Machines

```
library(caTools)
library(e1071)
library(xlsx)
library(gdata)
library(recipes)
library(caret)

datasetC1SVM = read.csv('C:/xxx.csv')
# [...]

# General script below based on one model, for a specific chain.
# Applicable for other subsets as well, to create the other SVM models, either lineary or poly.

## LINEAR SVM ##
## C1 SVM & PARAMETER TUNED
## Split dataset into two parts: 70% for training and 30% for testing
set.seed(234)
splitC1SVM = sample.split(datasetC1SVM$parts, SplitRatio = 0.70)
training_setC1SVM = subset(datasetC1SVM, splitC1SVM == TRUE)
test_setC1SVM = subset(datasetC1SVM, splitC1SVM == FALSE)
yC1SVM <- test_setC1SVM$parts
test_setC1SVM

## Create SVM Model with training dataset
modelC1SVM <- svm(training_setC1SVM$parts ~ ., data = training_setC1SVM, type = 'C-
classification',kernel = 'linear',scale=FALSE, probability=TRUE, cross=10, gamma=0.1)
print(modelC1SVM)
summary (modelC1SVM)

## Use model to predict the test dataset for accuracy of model
predC1SVM <- predict(modelC1SVM,test_setC1SVM)
table(predC1SVM, yC1SVM)

## Compute decision values and probabilities
predC1SVM <- predict(modelC1SVM, test_setC1SVM, decision.values = TRUE, probability = TRUE)
attr(predC1SVM, "probabilities")

CMC1SVM = table(test_setC1SVM[,1],predC1SVM)
CMC1SVM
confusionMatrix(CMC1SVM)

## Parameter Tuning for SVM, Grid, based on Cost and Gamma variables
## New model performance is determined based on the best model and input variables
linear.tuneC1 <- tune.svm(parts~., data=training_setC1SVM, type = 'C-classification',
kernel="linear", scale=FALSE, probability=TRUE, cross=10, cost=c(0.001, 0.01, 0.1, 0.5, 1),
gamma=c(0, 0.1, 0.2, 0.3, 0.4, 0.5))
summary(linear.tuneC1)
best.linearC1 <- linear.tuneC1$best.model
tune.testC1 <- predict(best.linearC1, newdata=test_setC1SVM)
CMC1SVM_tuned <- table(tune.testC1, test_setC1SVM$parts)
confusionMatrix(CMC1SVM_tuned)

## POLY SVM ##
## Similar script as above, change kernel to ‘polynomial’.
```

Appendix W R SCRIPT – MARKET BASKET ANALYSIS, ASSOCIATION MINING

```
library(arules)
library(arulesViz)
library(tidyverse)
library(readxl)
library(knitr)
library(ggplot2)
library(lubridate)
library(plyr)
library(dplyr)
library(extrafont)
font_import()
loadfonts(device="win")

#read excel into R dataframe
UsedParts12Nc <- read_excel('C:/XXX.xlsx')

# Pre-processing
# Complete cases(data) will return a logical vector indicating which rows have no missing values.
UsedParts12Nc2 <- UsedParts12Nc[complete.cases(UsedParts12Nc), ]
# Mutate function from dplyr package. It is used to edit or add new columns to dataframe.
# Here Description column is being converted to factor column.
# as.factor converts column to factor column. %>% is an operator with which you may pipe values
# to another function or expression
UsedParts12Nc %>% mutate(MaterialDescription = as.factor(MaterialDescription))
UsedParts12Nc %>% mutate(Part12Nc = as.factor(Part12Nc))
UsedParts12Nc %>% mutate(Country = as.factor(Country))

# Converts character data to date. Store CMDate as date in new variable
UsedParts12Nc$Date <- as.Date(UsedParts12Nc$Date)
# IF NEEDED, Extract time from CMDate and store in another variable
CMDate<- format(UsedParts12Nc$Date,"%mm:%dd:%yyy")
# Convert and edit CaseNumber into numeric
CaseNumber <- as.numeric(as.character(UsedParts12Nc$CaseNumber))

# Bind new columns CMDate and CaseNumber into dataframe retail
cbind(UsedParts12Nc,CMDate)
cbind(UsedParts12Nc,CaseNumber)
# get a glimpse of your data
glimpse(UsedParts12Nc)

# Transform data from single format to basket data
library(plyr)
transactionData <- ddply(UsedParts12Nc,c("CaseNumber"),
                         function(df1)paste(df1$MaterialDescription,
                                            collapse = ";"))
transactionData2 <- ddply(UsedParts12Nc,c("CaseNumber"),
                          function(df1)paste(df1$Part12Nc,
                                             collapse = ";"))
# The R function paste() concatenates vectors to character and separated results using
# collapse=[ optional character string ]. Here ';' is used

# As CaseNumber will not be of any use in the rule mining, set to NULL.
# Set column CaseNumber of dataframe transactionData
transactionData2$CaseNumber <- NULL
#Rename column to Part Description
colnames(transactionData2) <- c("Part Description")
#Show Dataframe transactionData
transactionData2  #Basket Data
write.csv(transactionData2,"C:/XXX/MaterialDescripton_Basket_Data.csv", quote = FALSE, row.names = FALSE)

# [...]
# IF Data needs to be imported, via:
transactionsImport <- read.transactions('C:/XXX.csv', format = 'basket', sep=';')
# Inspect
transactionsImport
summary(transactionsImport)
```

```r
# Density tells the percentage of non-zero cells in a sparse matrix.
# The total number of items that are purchased divided by a possible number of items in that matrix.
# Calculate how many items were purchased by using density: transactions x items x density

# Element describes:
# How many transactions there have been for 1-itemset, for 2-itemset etc.
# First row: number of items, second row: the number of transactions.
# e.g. 120197 transactions for 1 item, 36710 transactions with two items, etc.

# [...]

# Absolute/Relative Frequency Plot
# Create an item frequency plot for the top X items, for a specific dataset
if (!require("RColorBrewer")) {
  # install color package of R
  install.packages("RColorBrewer")
  #include library RColorBrewer
  library(RColorBrewer)
}
windowsFonts(corbel = windowsFont("Corbel"))
a <-
itemFrequencyPlot(transactionsImportSubset,topN=X,xlim=c(0,17),ylim=c(0,1.0),type="relative",col=brewer.pa
l(8,'Pastel2'),main="Relative Frequency Plot X, ...", cex.lab=1.2, cex.axis=1.1, cex.main=1.1,
cex.sub=1.3)
tblX <- crossTable(transactionsImportSubset, sort=TRUE)
tblX[1:11,1:11]
tableX <- xtable(tblX[1:11,1:11])
htmlTable(tableX)
diag(as.matrix(tableX[,]))
FrequencyX <- diag(as.matrix(tableX[,]))
barlabels(a,FrequencyX, pos=3, prop=1.1, cex=1.0, border=FALSE, xpad=0, ypad=0,
bg=ifelse(match(par("bg"),"transparent",0),"transparent",par("bg")))

# [...]

# Contingency Table
tbl <- crossTable(transactionsImport, sort =TRUE)
tbl[1:11,1:11]
crosstableX <- crossTable(transactionsImport, sort =TRUE)
diagcrosstableX <- diag(crosstableX)
diagcrosstableX
write.xlsx((diagcrosstableX), file = "XXX.xlsx")

# Shows number of occasions when these items were purchased together.
# Items sorted by frequency of purchase (note decreasing counts diagonal)
crossTable(transactionsImport, measure='lift',sort=T)[1:11,1:11]
crossTable(transactionsImport, measure='chi')
# low p-value would exclude possibility that lift less than 1 is due to chance.
# If absolute it will plot numeric frequencies of each item independently.
# If relative it will plot how many times these items have appeared as compared to others.

# [...]

# APRIORI ALGORITHM
association.rulesfinal <- apriori(transactionsImport, parameter = list(supp=0.0001,
conf=0.0001,minlen=1,maxlen=5, target='rules'))
summary(association.rulesfinal)
inspect(association.rulesfinal[1:200])
subsetvalue <- "NVC COIL-1.5"
association.rulesfinal_subsetlhs <- subset(association.rulesfinal,(lhs %pin% c(subsetvalue)))
association.rulesfinal_subsetrhs <- subset(association.rulesfinal,(rhs %pin% c(subsetvalue)))
summary(association.rulesfinal_subsetlhs)
summary(association.rulesfinal_subsetrhs)
inspect(association.rulesfinal_subsetlhs, by='lift', decrease=T)
inspect(association.rulesfinal_subsetrhs, by='lift', decrease=T)

windowsFonts(corbel = windowsFont("Corbel"))
itemFrequencyPlot(transactionsImport,(subset=items %in% (subsetvalue)),topN=30,
type="relative",col=brewer.pal(8,'Pastel2'),main="Relative Item Frequency Plot, Subset")
```

```
# In case there is a suspicion for spurious correlation: ChiSquared test
quality(association.rulesfinal)$chi <- interestMeasure(association.rulesfinal, measure='chi',
significance=T, transactionsImport)
inspect(sort(association.rulesfinal, by='lift', decreasing = T)[1:30])
inspect(subset(association.rulesfinal, subset=items %in% (subsetvalue) & confidence >.01), by='lift',
decreasing=T)
inspect(subset(association.rulesfinal, subset=items %pin% ("subsetvalue") & confidence >.01), by='lift',
decreasing=T)
subsetfinal <-inspect(subset(association.rulesfinal, subset=items %pin% (subsetvalue) & confidence >.001),
by='lift', decreasing=T)

shorter.association.rulesfinal <- apriori(transactionsImport, parameter = list(supp=0.001,
conf=0.1,maxlen=10))
summary(shorter.association.rulesfinal)
inspect(shorter.association.rulesfinal[1:30])

shorter.association.rulesfinallhs <- apriori(transactionsImport, parameter = list(supp=0.001,
conf=0.1,maxlen=10))
summary(shorter.association.rulesfinallhs)
inspect(shorter.association.rulesfinallhs[1:30])

shorter.association.rulesfinalrhs <- apriori(transactionsImport, parameter = list(supp=0.001,
conf=0.1,maxlen=10))
summary(shorter.association.rulesfinalrhs)
inspect(shorter.association.rulesfinalrhs[1:30])

# [...]

# Filter rules with confidence greater than X%
subRules<-association.rulesfinal[quality(association.rulesfinal)$confidence>0.001]
subRules3<-association.rulesfinal_subsetlhs[quality(association.rulesfinal_subsetlhs)$confidence>0.01]
subRules5<-association.rulesfinal_subsetrhs[quality(association.rulesfinal_subsetrhs)$confidence>0.01]
summary(subRules)
summary(subRules3)
# NOTE: subRules = complete data set, subRules 3 & 5 = LHS & RHS (resp.) given specific part subset

# Plot SubRules
plot(subRules, jitter=0, main="Scatter Plot Rules")
plot(subRules, jitter=0, interactive = TRUE)
plot(subRules, measure = c("support", "lift"), shading = "confidence", jitter=0, main="Scatter Plot X")

plot(subRules,method="two-key plot",jitter=0)
plot(subRules,method="two-key plot",jitter=0, interactive=TRUE)

# Check for Rule significant, as discussed in report based on Fisher Exact Test.
is.significant(subRules,transactionsImport, method='fisher', adjust='bonferroni')
inspect(subRules[is.significant(subRules,transactionsImport, method='fisher', adjust='bonferroni')])
subRules9 <- (subRules[is.significant(subRules,transactionsImport, method='fisher', adjust='bonferroni')])

plot(subRules9, jitter=0, main="Scatter Plot X, Significant")
plot(subRules9, jitter=0)
plot(subRules9, measure = c("support", "lift"), shading = "confidence", jitter=0, main="Scatter Plot X,
Significant")
plot(subRules9,method="two-key plot",jitter=0, main="Two-Key Plot X, Significant")
plot(subRules9,method="two-key plot",jitter=0, interactive=TRUE)

# Also check for significance for lhs and rhs subsetvalue subsets and assign new subRules name
quality(subRules6)$chi <- interestMeasure(subRules6, measure='chi', significance=T, transactionsImport)
inspect(sort(subRules6, by='lift', decreasing = T)[1:10])

is.significant(subRules4,transactionsImport, method='fisher', adjust='bonferroni')
inspect(subRules4[is.significant(subRules4,transactionsImport, method='fisher', adjust='bonferroni')])
subRules42 <- (subRules4[is.significant(subRules4,transactionsImport, method='fisher',
adjust='bonferroni')])

subRules44 <- (subRules43[is.significant(subRules43,transactionsImport, method='fisher',
adjust='bonferroni')])
summary(subRules44)
inspect(subRules44)
```

```
is.significant(subRules6,transactionsImport, method='fisher', adjust='bonferroni')
inspect(subRules6[is.significant(subRules4,transactionsImport, method='fisher', adjust='bonferroni')])
subRules62 <- (subRules6[is.significant(subRules4,transactionsImport, method='fisher',
adjust='bonferroni')])

# Plot lhs and rhs Rules
plot(subRules3, jitter=0)
plot(subRules3,method="two-key plot",jitter=0)

plot(subRules5, jitter=0)
plot(subRules5,method="two-key plot",jitter=0)

# Filter top 20 rules with highest lift
subRules2<-head(subRules, n=20, by="confidence")
plot(subRules2, method="paracoord")
plot(subRules2, method = "graph",  engine = "htmlwidget")
library(arulesViz)
plot(subRules2,method="graph",engine='interactive',shading=NA)

subRules4<-head(subRules3, n=20, by="confidence")
plot(subRules42, method="paracoord", reorder=TRUE)
plot(subRules42, method = "graph",  engine = "htmlwidget")
library(arulesViz)
plot(subRules4,method="graph",engine='interactive',shading=NA)
inspect(subRules3)

subRules6<-head(subRules5, n=20, by="confidence")
plot(subRules6, method="paracoord", reorder=TRUE)
plot(subRules6, method = "graph",  engine = "htmlwidget")
library(arulesViz)
plot(subRules6,method="graph",engine='interactive',shading=NA)

# Grouped matrix plot
plot(subRules4, method="grouped")
sel <- plot(subRules6, method="grouped", interactive=TRUE)
```

Appendix X EXAMPLE OUTPUT OPTIMIZED XGBOOST MODEL – CHAIN 2

The following figure shows part of the multi-classification model output, obtained from the XGBoost prediction model for Chain 2 with the 15 different part clusters, based on the test set data. The labels in the figure represent the different clusters, ranging from X1 to X15. The model outputs a predicted part cluster based on the class with the highest probability (max) and compares this to the actual class (label). The revised parameters of the XGBoost model are: rounds = 1000, max-depth = 6, colsample_bytree = 0, eta = 0.01, gamma = 0.

For example, given the following input:

#	ConsumedParts	SystemModel	Priority	Market	EntitlementType	$C_xT_yE_z$	...	C2T05E09	...	C2T05E15	...	C2T05E20
10	PartCluster9	2	5	7	1	0	...	1	...	1	...	1

The following prediction is made:

#	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	max	label
10	0.0001	0.1131	0.0000	0.0001	0.0002	0.0001	0.0276	0.0033	0.8134	0.0321	0.0009	0.0087	0.0002	0.0000	0.0002	9	9

Snippet model output:

Table 47 - XGBoost Output Example

#	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	max	label
1	0.0105	0.3386	0.0004	0.3088	0.0141	0.0015	0.0031	0.0004	0.0003	0.0082	0.0063	0.0006	0.0001	0.0008	0.3063	2	15
2	0.0002	0.0019	0.7229	0.0001	0.0287	0.0014	0.0009	0.0002	0.0008	0.0003	0.0016	0.0002	0.0006	0.0004	0.2398	3	3
3	0.0003	0.0029	0.0010	0.0014	0.0004	0.0010	0.0001	0.0001	0.0076	0.0003	0.0031	0.0021	0.0002	0.0002	0.9790	15	15
4	0.0005	0.0069	0.0003	0.0008	0.0003	0.0042	0.4356	0.0063	0.4261	0.0268	0.0273	0.0638	0.0002	0.0003	0.0006	7	9
5	0.0002	0.8648	0.0000	0.0999	0.0004	0.0001	0.0014	0.0003	0.0007	0.0259	0.0025	0.0016	0.0000	0.0003	0.0019	2	2
6	0.0005	0.0057	0.0026	0.0007	0.0012	0.0167	0.0937	0.0031	0.0803	0.2622	0.4842	0.0469	0.0001	0.0001	0.0020	11	10
7	0.0002	0.1213	0.0008	0.2347	0.0075	0.0004	0.0002	0.0024	0.0009	0.0179	0.0008	0.0200	0.0002	0.0003	0.5925	15	15
8	0.0003	0.0001	0.2561	0.0010	0.0761	0.0016	0.0003	0.0023	0.0002	0.0009	0.0002	0.0015	0.0001	0.0001	0.6593	15	15
9	0.0018	0.0034	0.0002	0.0034	0.0035	0.0056	0.3484	0.1433	0.0026	0.1088	0.0215	0.3421	0.0002	0.0037	0.0115	7	1
10	0.0001	0.1131	0.0000	0.0001	0.0002	0.0001	0.0276	0.0033	0.8134	0.0321	0.0009	0.0087	0.0002	0.0000	0.0002	9	9
11	0.0001	0.0003	0.8851	0.0001	0.0007	0.0001	0.0002	0.0002	0.0013	0.0014	0.0059	0.0004	0.0002	0.0000	0.1041	3	3
12	0.0003	0.0002	0.0008	0.0000	0.0115	0.0001	0.0000	0.0001	0.0011	0.0000	0.0001	0.0002	0.0001	0.0038	0.9816	15	3
13	0.0005	0.1664	0.0001	0.0006	0.0001	0.0063	0.5484	0.0352	0.0767	0.1433	0.0151	0.0069	0.0001	0.0001	0.0002	7	2
14	0.0001	0.0017	0.0000	0.0003	0.0001	0.0004	0.0385	0.0089	0.8601	0.0015	0.0004	0.0873	0.0000	0.0001	0.0007	9	9
15	0.0013	0.0133	0.0015	0.1672	0.1824	0.0003	0.0115	0.0123	0.0001	0.1867	0.0177	0.2062	0.0031	0.0039	0.1925	12	12
16	0.0002	0.0054	0.0000	0.0008	0.0002	0.0234	0.2959	0.0224	0.3411	0.2731	0.0081	0.0278	0.0013	0.0001	0.0001	9	9
17	0.0309	0.7535	0.0011	0.1173	0.0023	0.0014	0.0032	0.0039	0.0200	0.0308	0.0275	0.0007	0.0004	0.0013	0.0056	2	2
18	0.0753	0.0011	0.0006	0.0003	0.0001	0.0013	0.0019	0.8534	0.0048	0.0426	0.0016	0.0126	0.0014	0.0029	0.0002	8	8
19	0.7624	0.0002	0.0000	0.0002	0.0001	0.0023	0.0030	0.0199	0.0042	0.1809	0.0121	0.0136	0.0006	0.0001	0.0004	1	1

Appendix Y FINAL MODEL VISUALIZATION & FEATURE IMPORTANCE

Y.1 XGBoost – Feature Importance - Cover

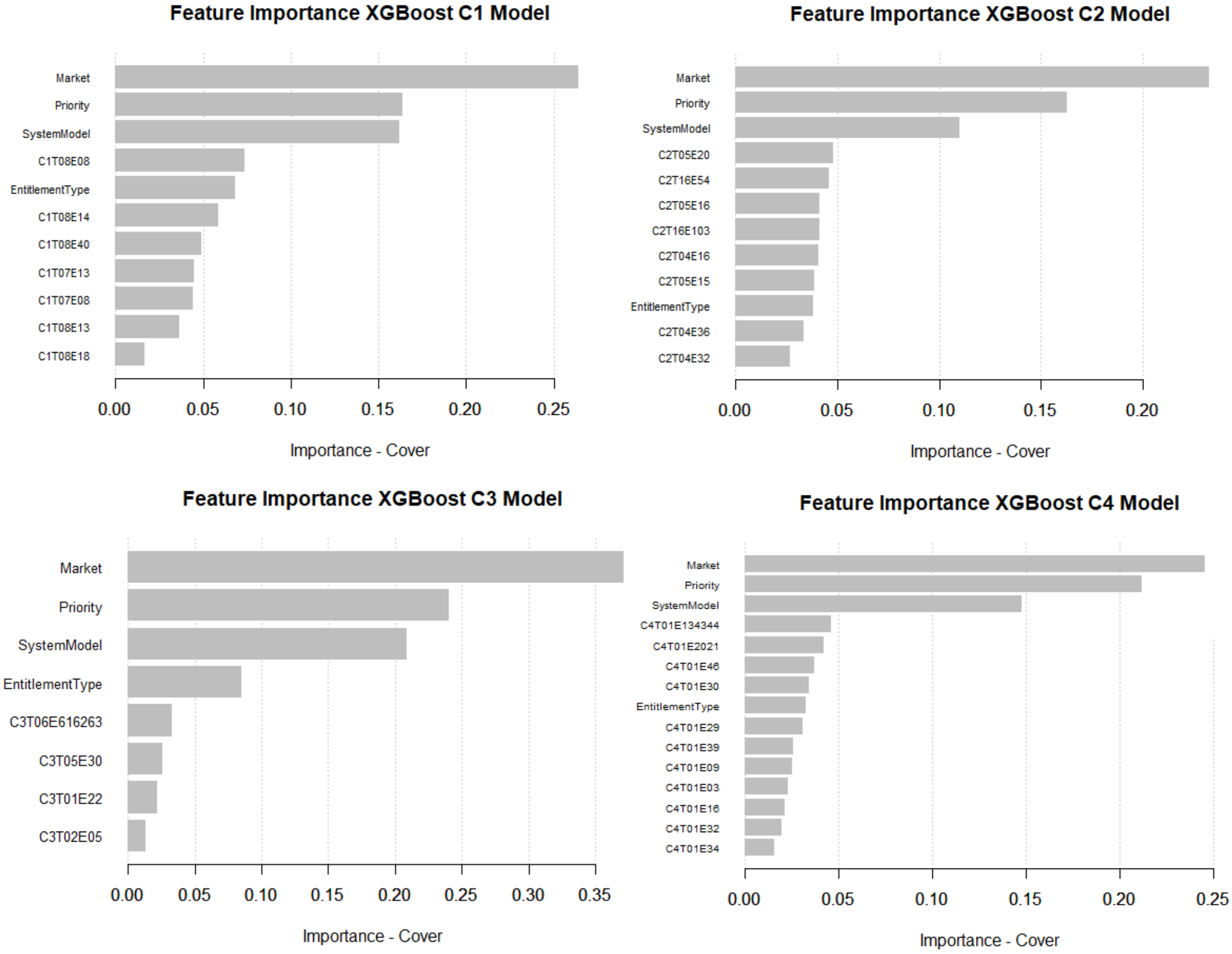


Fig. 42 - Feature Importance XGBoost Models - Cover

Y.2 Decision Rule Visualization – Pruned Decision Trees:

Chain 1 – Part Cluster 2

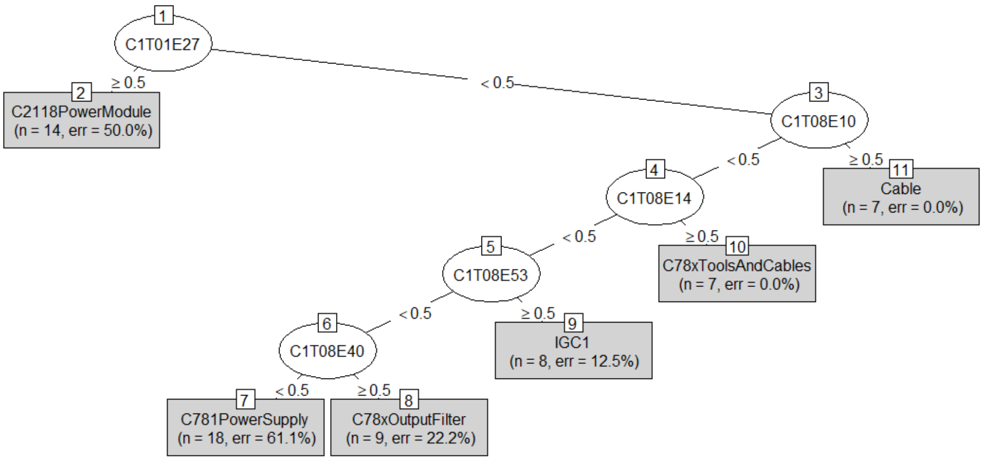
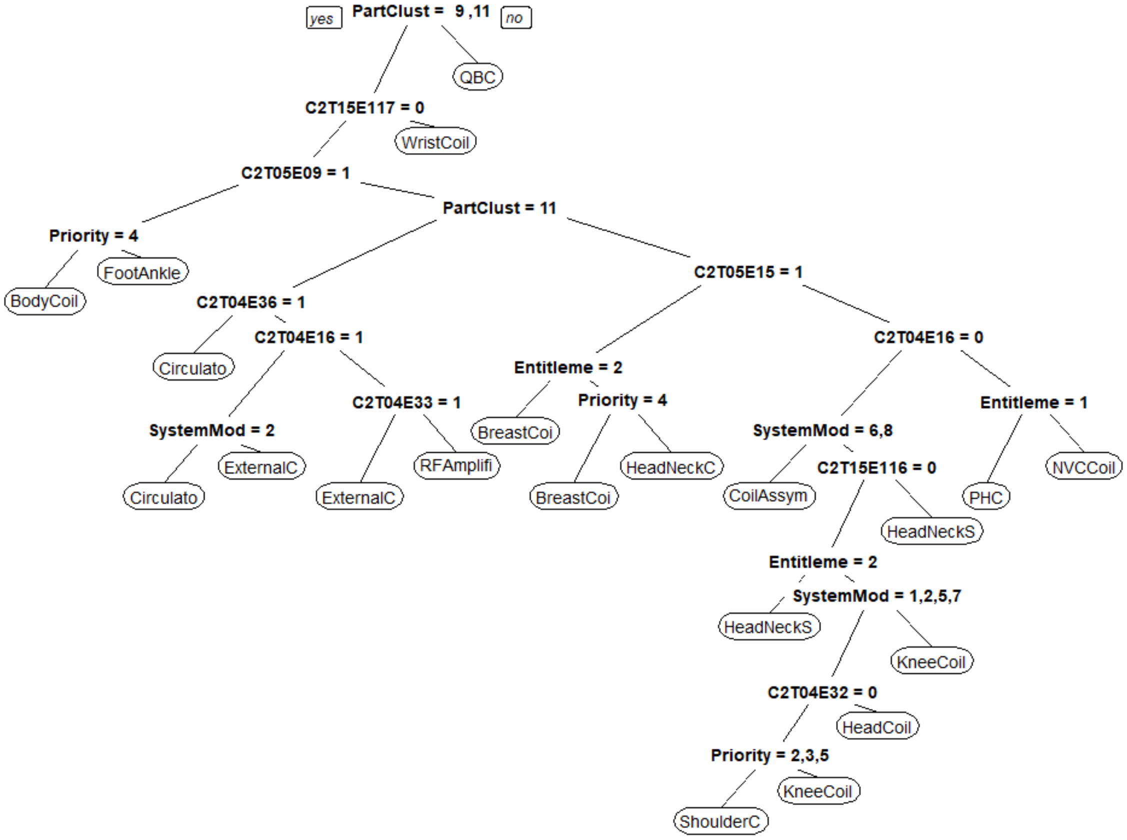


Fig. 43 - Decision Tree C1, PC2

Chain 2 – Part Cluster 7, 9, 11 – Alternative View



Decision Tree C2, PC7,9,11 Alternative

Chain 3 – Part Cluster 15

As discussed in corresponding section in the results chapter, while validating the model results, a SME elaborated on Part Cluster 1 being the cluster with spare parts that would be expected for chain 3 maintenance cases. From model results it was clear that this part cluster was very underrepresented in the model' dataset and seems to perform relatively poorly contrary to acceptable performance metrics. Validation with RSE's showed that parts often used for this chain come from other clusters, as maintenance for such cases often does not require to replace the whole unit, but separate components suffice as well. Hence, the decision tree for part cluster 15 is displayed here based on chain 3 data set, instead of part cluster 1.

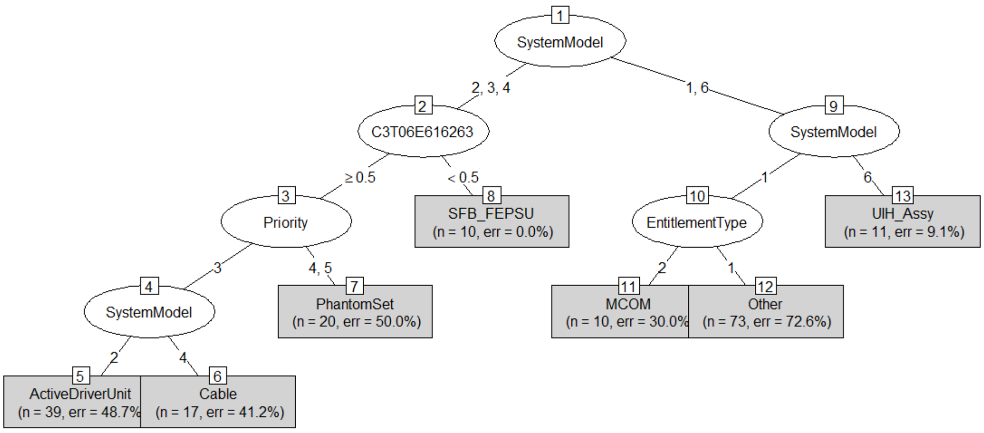


Fig. 44 - Decision Tree C3, PC15

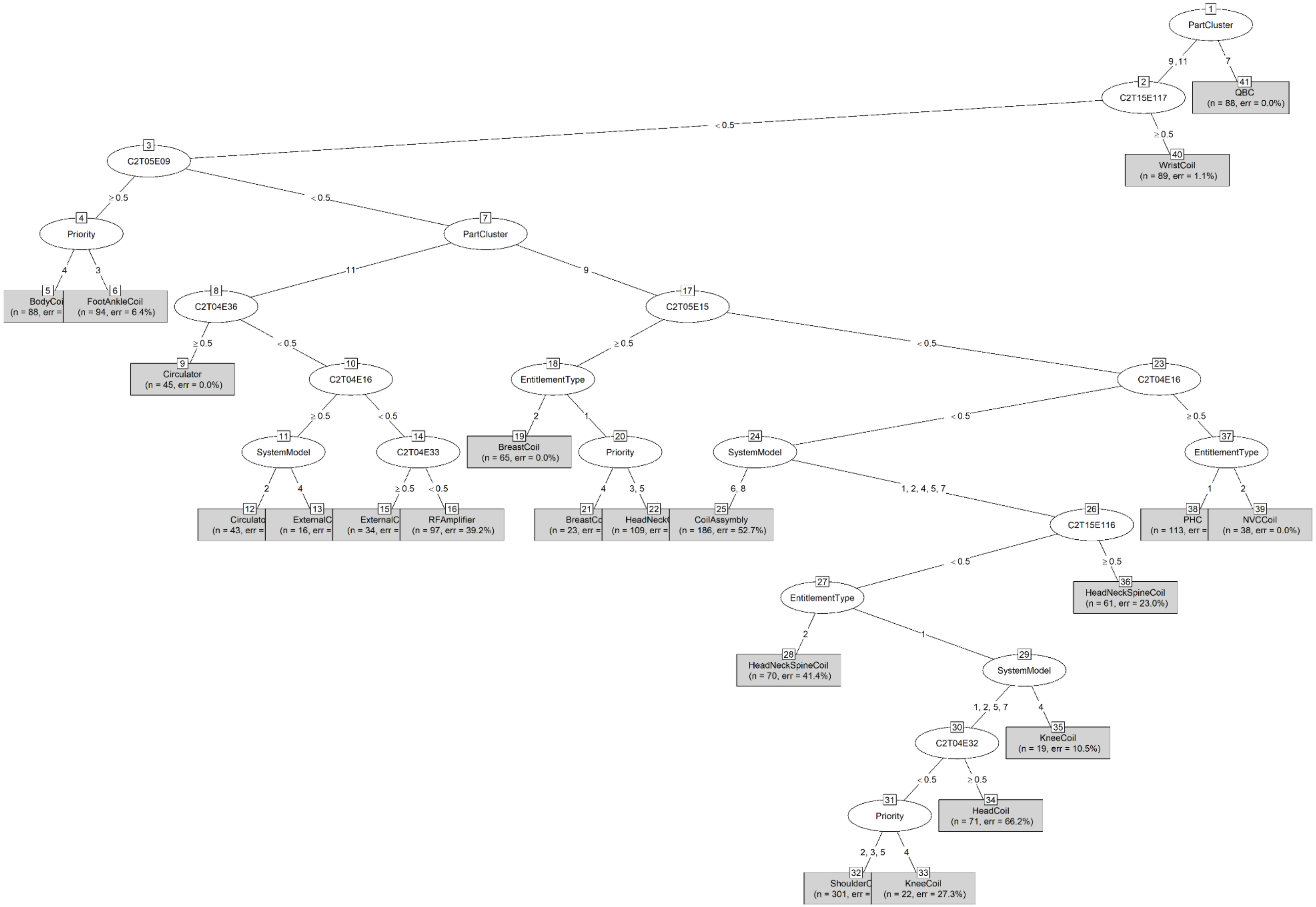


Fig. 45 - Decision Tree C2, PC7,9,11

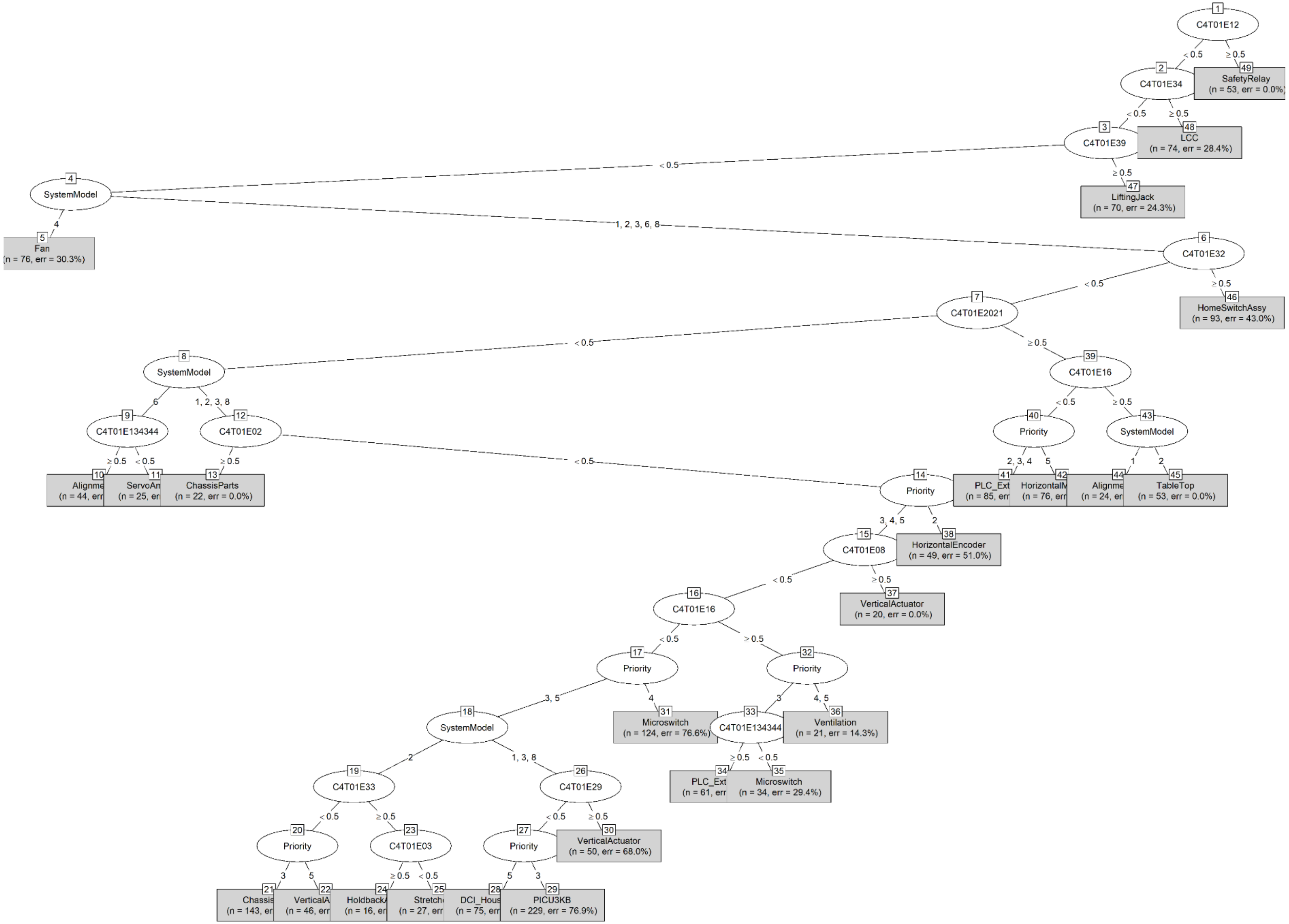
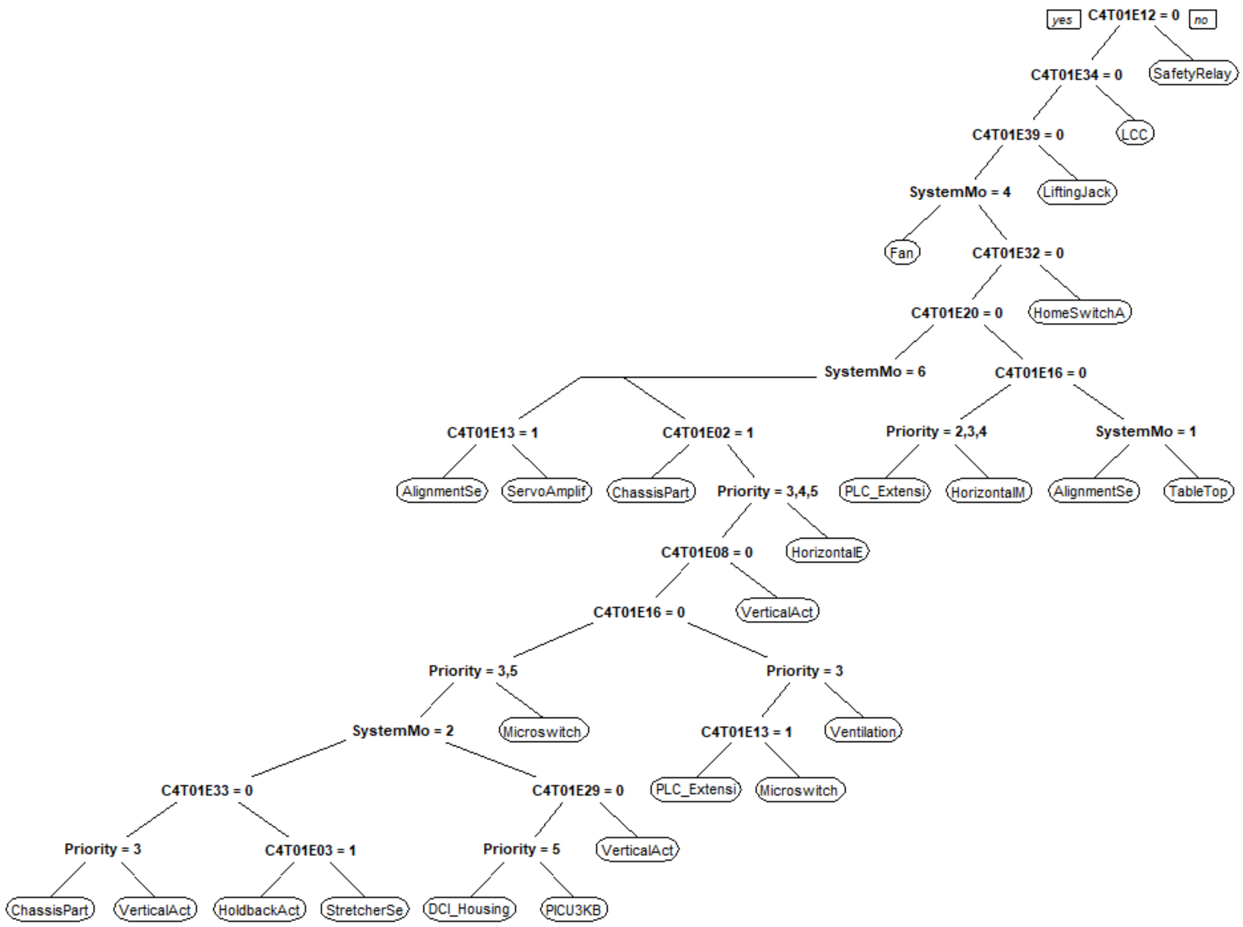


Fig. 46 - Decision Tree C4, PC5

Chain 4 – Part Cluster 5 – Alternative View



Decision Tree C4, PC5 Alternative

Y.3 Decision Rule Visualization – Decision Tree, Feature Importance

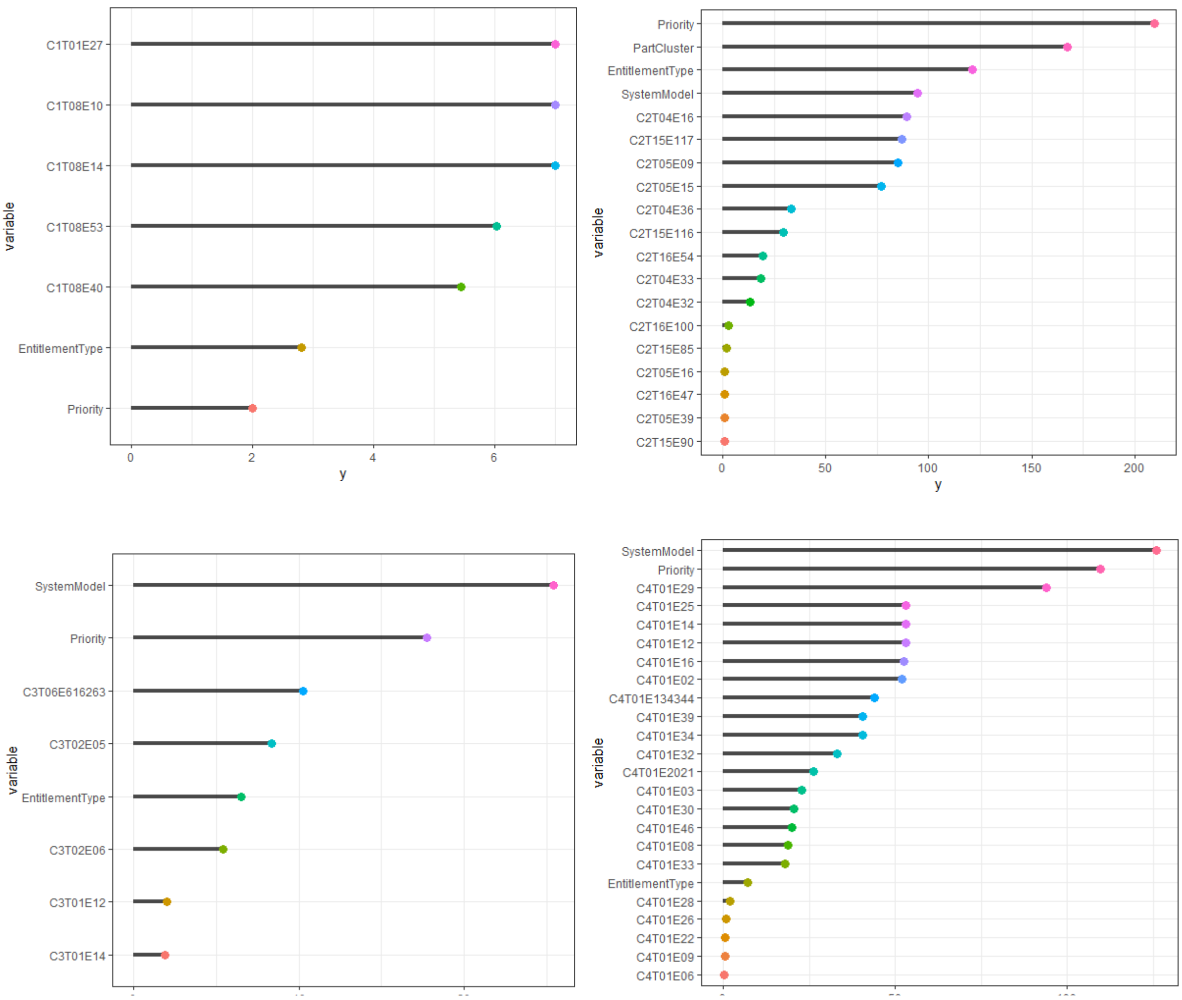


Fig. 47 - Decision Trees, Feature Importance

Above feature importance graphs are generated for the decision trees presented in X.2. Hence the feature importance for C1, C2, C3, and C4 (subset) cases are based Part Cluster 2, (7, 9, 11), 15, and 5 respectively.

Appendix Z PART CLUSTER, MARKET ANALYSIS

For the purpose of illustrating that not all part clusters (PC's) are consumed in all markets, as part of the XGBoost – Feature Importance, below an overview is given where '*x*' indicates that (spare) parts from PC's have been consumed during cases from a specific market. Some PC's are consumed significantly more often than other PC's in markets; an interesting observation for future analysis regarding replacement behavior and machine usage. For this overview, confidential consumptions numbers suffice.

Table 48 - Part Cluster usage per Market

Data subset	**Part Cluster (PC)**	**Market (M)**													
		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
Chain 1 Cases	PC 1	x		x						x	x	x			
	PC 2	x	x	x	x	x	x	x		x	x	x	x		x
	PC 3	x	x	x	x	x	x	x	x	x		x	x		x
	PC 4	x	x	x	x	x		x		x	x	x			x
	PC 5	x	x	x	x	x		x	x	x	x	x	x		x
	PC 6			x	x	x	x	x		x		x			x
	PC 7														
	PC 8			x											
	PC 9	x	x	x	x	x	x	x	x	x	x	x	x		x
	PC 10	x	x	x	x		x	x		x	x	x	x		x
	PC 11	x		x				x							x
	PC 12		x	x		x		x	x	x	x	x			x
	PC 13	x	x	x		x	x	x		x	x	x			
	PC 14			x			x								
	PC 15	x	x		x	x		x		x	x	x	x		x
Chain 2 Cases	PC 1	x	x	x		x				x		x			
	PC 2	x	x	x	x	x		x		x		x	x		x
	PC 3	x	x	x		x	x	x		x	x	x	x	x	x
	PC 4	x	x	x	x	x	x	x	x	x	x	x	x		x
	PC 5	x	x	x	x	x	x		x	x	x	x	x		x
	PC 6	x	x	x	x		x	x	x	x		x	x		
	PC 7			x						x			x		
	PC 8			x	x						x		x	x	
	PC 9	x	x	x	x	x	x	x	x	x	x	x	x		x
	PC 10	x	x	x	x	x	x	x		x	x	x	x	x	x
	PC 11	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	PC 12	x	x	x	x	x	x	x	x	x	x	x	x		x
	PC 13	x	x	x	x			x	x	x		x			
	PC 14			x	x	x		x		x		x			
	PC 15	x	x	x	x	x	x	x		x	x	x	x		x

Data subset	Part Cluster (PC)	Market (M)													
		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
Chain 3 Cases	PC 1	x	x	x	x					x	x	x	x		
	PC 2	x	x	x		x		x		x		x	x		x
	PC 3	x	x	x	x	x		x	x	x	x	x	x	x	x
	PC 4	x		x	x	x	x	x	x	x	x	x	x		x
	PC 5	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	PC 6	x	x	x	x	x	x	x	x	x	x	x		x	x
	PC 7			x						x					
	PC 8	x		x		x		x	x	x	x	x	x		
	PC 9	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	PC 10	x	x	x	x	x	x	x	x	x	x	x	x		x
	PC 11	x	x	x	x			x		x			x		
	PC 12	x	x	x	x	x	x	x		x		x	x	x	
	PC 13	x	x	x	x	x		x		x	x	x	x		
	PC 14			x	x	x	x	x		x		x			
	PC 15	x	x	x	x	x	x	x	x	x		x	x		x
Chain 4 Cases	PC 1	x	x	x	x	x		x	x	x	x	x	x	x	x
	PC 2	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	PC 3	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	PC 4	x	x	x	x	x	x	x	x	x	x	x	x		x
	PC 5	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	PC 6	x	x	x	x	x	x	x		x	x	x	x	x	x
	PC 7		x	x	x					x		x			x
	PC 8	x	x	x					x	x		x			
	PC 9	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	PC 10	x	x	x	x	x	x	x	x	x	x	x	x		x
	PC 11	x	x	x	x	x		x		x		x			x
	PC 12	x	x	x	x	x	x	x	x	x	x	x	x		x
	PC 13	x	x	x	x	x	x	x	x	x		x	x		x
	PC 14	x	x	x	x		x	x	x	x		x			x
	PC 15	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Appendix AA Confusion Matrices XGBoost

This appendix includes the confusion matrices for the created XGBoost models for part clusters (PC) for the four different chains.

** represent specific classes that are not included in the test set for the corresponding XGBoost model, due to random splitting the dataset.*

Table 49 - Chain 1, test.confusion.matrix XGBoost

Reference: / Prediction:	PC1	PC2	PC3	PC4	PC5	PC6 *	PC7	PC8	PC9	PC10	PC11	PC12	PC13 *	PC14	PC15
Part Cluster 1	3	0	0	0	3	0	0	0	0	0	0	0	0	0	0
Part Cluster 2	0	13	0	1	0	0	0	1	0	1	0	0	0	0	0
Part Cluster 3	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
Part Cluster 4	0	0	0	2	0	0	0	1	0	0	0	0	0	3	0
Part Cluster 5	0	0	0	0	4	0	0	2	0	0	0	2	0	2	0
Part Cluster 6 *	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Part Cluster 7	0	1	0	1	0	0	3	0	0	0	0	0	0	0	0
Part Cluster 8	0	0	0	0	0	0	0	15	0	1	0	0	0	0	0
Part Cluster 9	0	1	0	0	0	0	0	0	5	0	0	0	0	1	0
Part Cluster 10	1	0	0	0	0	0	0	0	0	13	0	0	0	0	0
Part Cluster 11	0	0	0	0	0	0	2	0	0	4	7	0	0	0	0
Part Cluster 12	0	0	0	0	0	0	0	0	1	0	0	3	0	0	0
Part Cluster 13 *	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Part Cluster 14	1	1	0	0	0	0	0	1	0	0	0	0	0	17	0
Part Cluster 15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4

Table 50 - Chain 2, test.confusion.matrix XGBoost

Reference: / Prediction:	PC1	PC2	PC3	PC4	PC5	PC6 *	PC7	PC8	PC9	PC10	PC11	PC12	PC13 *	PC14	PC15
Part Cluster 1	2	0	0	0	0	0	1	1	0	0	0	0	0	0	0
Part Cluster 2	1	22	0	0	0	0	3	1	2	0	0	0	0	0	1
Part Cluster 3	0	0	27	0	0	0	0	0	0	0	0	1	0	0	4
Part Cluster 4	0	1	0	8	0	0	0	0	0	0	0	1	0	0	1
Part Cluster 5	0	1	0	0	4	0	0	0	0	1	0	0	0	0	0
Part Cluster 6 *	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Part Cluster 7	0	0	0	0	0	0	15	0	0	4	0	0	0	0	1
Part Cluster 8	0	0	0	0	0	0	1	12	0	1	0	0	0	0	1
Part Cluster 9	0	0	1	0	0	0	2	2	12	0	0	0	0	0	0
Part Cluster 10	0	1	0	0	1	0	3	0	0	15	0	0	0	0	0
Part Cluster 11	0	0	0	0	0	0	3	1	0	2	10	0	0	0	0
Part Cluster 12	0	0	0	0	0	0	0	0	0	2	1	12	0	0	0
Part Cluster 13 *	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Part Cluster 14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Part Cluster 15	0	4	4	1	2	0	0	0	0	2	0	2	0	0	61

Table 51 - Chain 3, test.confusion.matrix XGBoost

Reference: Prediction:	PC1	PC2	PC3	PC4	PC5	PC6 *	PC7	PC8	PC9	PC10	PC11	PC12	PC13 *	PC14	PC15
Part Cluster 1	2	0	0	0	0	0	0	0	0	0	0	1	0	0	2
Part Cluster 2	0	22	0	2	1	0	0	1	0	0	1	0	0	0	2
Part Cluster 3	0	0	0	4	0	0	0	0	0	0	0	0	0	0	2
Part Cluster 4	0	2	0	7	0	0	1	0	0	0	0	1	0	1	5
Part Cluster 5	1	1	0	0	3	0	0	0	0	0	0	0	0	0	3
Part Cluster 6 *	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Part Cluster 7	0	1	0	0	0	0	15	1	2	1	1	0	0	0	0
Part Cluster 8	0	1	0	0	0	0	4	8	3	1	0	0	0	0	1
Part Cluster 9	1	1	0	0	0	0	2	0	16	0	1	1	0	0	2
Part Cluster 10	0	2	0	0	0	0	5	0	3	15	4	0	0	0	2
Part Cluster 11	0	0	0	0	0	0	3	0	2	0	13	0	0	0	4
Part Cluster 12	0	4	0	1	0	0	2	0	4	3	1	6	0	0	2
Part Cluster 13 *	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Part Cluster 14	0	0	0	0	0	0	1	0	0	1	0	1	0	7	0
Part Cluster 15	0	3	2	3	0	0	0	0	0	1	0	0	0	0	64

Table 52 - Chain 4, test.confusion.matrix XGBoost

Reference: Prediction:	PC1	PC2	PC3	PC4	PC5	PC6 *	PC7	PC8	PC9	PC10	PC11	PC12	PC13 *	PC14	PC15
Part Cluster 1	7	1	0	1	0	0	2	0	3	3	0	0	0	0	4
Part Cluster 2	0	55	0	2	0	0	8	0	4	5	2	2	0	1	7
Part Cluster 3	0	0	7	0	0	0	0	0	0	0	0	0	0	0	16
Part Cluster 4	0	7	0	23	0	0	0	1	0	0	1	3	0	0	10
Part Cluster 5	1	3	0	1	21	0	1	1	0	0	0	0	0	0	5
Part Cluster 6 *	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Part Cluster 7	2	2	0	2	0	0	37	3	10	12	9	1	0	0	3
Part Cluster 8	0	0	0	0	0	0	3	25	7	9	9	1	0	0	3
Part Cluster 9	1	1	0	0	0	0	11	2	39	13	4	0	0	0	2
Part Cluster 10	4	12	0	0	0	0	9	2	7	58	9	3	0	0	4
Part Cluster 11	2	2	0	0	1	0	7	1	10	16	46	4	0	0	3
Part Cluster 12	0	3	0	1	1	0	3	0	3	2	3	23	0	0	2
Part Cluster 13 *	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Part Cluster 14	0	0	0	0	0	0	1	0	0	2	0	1	0	6	0
Part Cluster 15	1	4	5	5	5	0	2	0	0	0	4	1	0	0	214

Appendix BB CONFUSION MATRICES DECISION TREES

This appendix includes the confusion matrices for the created decision trees to visualize the decision rules for part clusters (best predicted or expected) for the four different chains. Training sets were either re-sampled based on up sampling or not based on the (too imbalanced) class (part) distribution in the data subset. Below the tables represent the confusion matrices for the test sets, for which it is indicated whether the created model was based on a re-sampled data set or not. Although SMOTE re-sampling was also discussed in the modeling and results section of the report, SMOTE was not used here, due to – in some cases – low amount of cases per class were even lowering the amount of observation per class due to under sampling of dominant classes would not be beneficial.

To re-iterate, based on optimal CP-value: for pruned Chain 1 to 4 models respectively, the *CP*, *Kappa*, *xError* and *xStd* values are:

Chain 1: 0.01; 0.63; 0.75; 0.07 | **Chain 2**: 0.011; 0.67; 0.36; 0.01 | **Chain 3**: 0.02; 0.42; 0.59; 0.04 | **Chain 4**: 0.014; 0.58; 0.56; 0.01

** represent specific part classes that are not included in the test set for the corresponding decision tree, due to random splitting the dataset.*

Table 53 - Chain 1, Part Cluster 2, test.confusion.matrix

	C2118 PowerModule	**C21x AxisController**	**C781 PowerSupply**	**C787 Switch**	**C78x OutputFilter**	**C78x PowerModule**	**C78x ToolsAndCables**	**Cable**	**IGC1**
C2118 PowerModule	7	7	0	0	0	0	0	0	0
C21x AxisController *	0	0	0	0	0	0	0	0	0
C781 PowerSupply	0	0	7	7	0	4	0	0	0
C787 Switch *	0	0	0	0	0	0	0	0	0
C78x OutputFilter	0	0	0	0	7	2	0	0	0
C78x PowerModule *	0	0	0	0	0	0	0	0	0
C78x ToolsAndCables	0	0	0	0	0	0	7	0	0
Cable	0	0	0	0	0	0	0	7	0
IGC1	0	0	0	0	0	1	0	0	7

Table 54 - Chain 3, Part Cluster 15, test.confusion.matrix

	ActiveDriverUnit	**BoreLight**	**Cable**	**MCOM**	**Other**	**PhantomSet**	**SFB_FEPSU**	**UIH_Assy**	**UIM**
ActiveDriverUnit	20	4	10	0	0	1	0	0	4
BoreLight *	0	0	0	0	0	0	0	0	0
Cable	0	2	10	0	0	0	5	0	0
MCOM	0	3	0	7	0	0	0	0	0
Other	0	5	0	13	20	9	4	10	12
PhantomSet	0	6	0	0	0	10	0	0	4
SFB_FEPSU	0	0	0	0	0	0	10	0	0
UIH_Assy	0	0	0	0	0	0	1	10	0
UIM *	0	0	0	0	0	0	0	0	0

Table 55 - Chain 2, Part Cluster 7,9,11, test.confusion,matrix.upsampling

	Anterio r Coil	**Base Coil**	**Body Coil**	**Breast Coil**	**Circu-lator**	**Coil Assem-bly**	**Extern al Coil**	**Flex Coil**	**Foot Ankle Coil**	**Head Coil**	**Head Neck Coil**	**Head Neck Spine Coil**	**Knee Coil**	**NVC Coil**	**PHC**	**QBC**	**RF Ampl-ifier**	**Shoul der Coil**	**Wrist Coil**
Anterior Coil *	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BaseCoil *	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Body Coil	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Breast Coil	0	0	0	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Circulator	0	0	0	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coil Assembly	22	25	0	0	0	88	0	39	0	4	0	0	0	0	0	0	0	8	0
External Coil	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
Flex Coil *	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Foot AnkleCoil	0	0	0	0	0	0	0	0	88	0	0	0	5	0	0	0	1	0	0
Head Coil	1	0	0	0	0	0	0	0	0	24	0	0	18	23	0	0	5	0	0
Head NeckCoil	7	10	0	0	0	0	0	8	0	8	76	0	0	0	0	0	0	0	0
HeadNeck SpineCoil	8	0	0	0	0	0	0	9	4	21	5	69	13	0	0	0	2	0	0
Knee Coil	3	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	5	0	0
NVC Coil	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0	0	0	0	0
PHC	11	0	0	0	0	0	0	0	0	6	0	0	4	0	79	0	4	9	0
QBC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88	0	0	0
RF Amplifier	0	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0	59	0	0
Shoulder Coil	40	56	0	0	0	0	0	34	0	43	0	0	15	27	0	0	13	73	0
Wrist Coil	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	92

Table 56 - Chain 4, Part Cluster 5, test.confusion,matrix.upsampling (1/3)

	Alignment Set	Chassis Parts	Connector Block	Connector Handgrip	Connector Magnet	Cover	DCI_Chassis Handgrip	DCI_Chassis Magnet	DCI_ HousingPins	Fan	Holdback Actuator
AlignmentSet	47	0	0	0	0	2	4	2	0	0	0
ChassisParts	0	43	13	3	6	7	0	14	0	0	10
ConnectorBlock *	0	0	0	0	0	0	0	0	0	0	0
ConnectorHandgrip *	0	0	0	0	0	0	0	0	0	0	0
ConnectorMagnet *	0	0	0	0	0	0	0	0	0	0	0
Cover *	0	0	0	0	0	0	0	0	0	0	0
DCI_ChassisHandgrip *	0	0	0	0	0	0	0	0	0	0	0
DCI_ChassisMagnet *	0	0	0	0	0	0	0	0	0	0	0
DCI_HousingPins	0	0	0	11	4	11	0	1	18	0	16
Fan	0	0	8	8	0	0	0	1	0	53	0
HoldbackActuator	0	0	0	0	0	0	0	0	0	0	16
HomeSwitchAssy	0	0	0	0	0	0	0	2	0	0	0
HorizontalEncoder	0	0	0	0	0	2	0	0	0	0	0
HorizontalMotorAssy	0	0	0	0	1	1	0	1	0	0	0
LCC	0	0	0	0	3	0	0	2	0	0	0
LiftingJack	0	0	0	3	0	0	5	0	0	0	0
Microswitch	0	0	0	2	8	5	8	11	11	0	11
PICU3KB	0	0	11	20	11	10	3	7	24	0	0
PLC_Extension	0	0	0	6	6	7	28	10	0	0	0
PSControlUnit *	0	0	0	0	0	0	0	0	0	0	0
PSU *	0	0	0	0	0	0	0	0	0	0	0
SafetyRelay	0	0	0	0	0	0	0	0	0	0	0
SBM *	0	0	0	0	0	0	0	0	0	0	0
ServoAmplifier	0	0	0	0	0	0	0	0	0	0	0
Stop&Go *	0	0	0	0	0	0	0	0	0	0	0
StretcherSet	0	0	0	0	0	0	0	0	0	0	0
TableTop	0	0	0	0	0	0	0	0	0	0	0
Trolley *	0	0	0	0	0	0	0	0	0	0	0
Ventilation	0	0	0	0	2	1	0	0	0	0	0
VerticalActuator	0	0	21	0	9	3	5	2	0	0	0

Table 57 - Chain 4, Part Cluster 5, test.confusion.matrix.upsampling (2/3)

	Home SwitchAssy	Horizontal Encoder	Horizontal MotorAssy	LCC	Lifting Jack	Microswitch	PICU3KB	PLC_Extension	PSControl Unit	PSU	Safety Relay
AlignmentSet	0	0	0	0	0	0	0	0	0	0	0
ChassisParts	0	9	0	0	0	0	0	0	0	26	0
ConnectorBlock *	0	0	0	0	0	0	0	0	0	0	0
ConnectorHandgrip *	0	0	0	0	0	0	0	0	0	0	0
ConnectorMagnet *	0	0	0	0	0	0	0	0	0	0	0
Cover *	0	0	0	0	0	0	0	0	0	0	0
DCI_ChassisHandgrip *	0	0	0	0	0	0	0	0	0	0	0
DCI_ChassisMagnet *	0	0	0	0	0	0	0	0	0	0	0
DCI_HousingPins	0	0	0	0	0	0	0	0	0	0	0
Fan	0	0	0	0	0	0	0	0	0	0	0
HoldbackActuator	0	0	0	0	0	0	0	0	0	0	0
HomeSwitchAssy	53	0	0	0	0	0	0	0	12	0	0
HorizontalEncoder	0	24	0	0	0	0	0	0	0	3	0
HorizontalMotorAssy	0	0	41	0	0	0	0	0	0	0	0
LCC	0	0	0	53	0	0	0	0	15	1	0
LiftingJack	0	0	0	0	53	0	0	0	0	3	0
Microswitch	0	0	0	0	0	53	0	0	0	9	0
PICU3KB	0	0	0	0	0	0	46	12	15	4	0
PLC_Extension	0	20	0	0	0	0	0	41	0	3	0
PSControlUnit *	0	0	0	0	0	0	0	0	0	0	0
PSU *	0	0	0	0	0	0	0	0	0	0	0
SafetyRelay	0	0	0	0	0	0	0	0	0	0	37
SBM *	0	0	0	0	0	0	0	0	0	0	0
ServoAmplifier	0	0	0	0	0	0	0	0	0	0	0
Stop&Go *	0	0	0	0	0	0	0	0	0	0	0
StretcherSet	0	0	0	0	0	0	0	0	0	4	0
TableTop	0	0	0	0	0	0	0	0	0	0	0
Trolley *	0	0	0	0	0	0	0	0	0	0	0
Ventilation	0	0	0	0	0	0	0	0	0	0	0
VerticalActuator	0	0	0	0	0	0	0	0	11	0	0

Table 58 - Chain 4, Part Cluster 5, test.confusion.matrix.upsampling (3/3)

	SBM	Servo Amplifier	Stop & Go	Stretcher Set	Table Top	Trolley	Ventilation	Vertical Actuator
AlignmentSet	7	0	0	0	0	0	0	0
ChassisParts	10	0	4	0	0	6	4	0
ConnectorBlock *	0	0	0	0	0	0	0	0
ConnectorHandgrip *	0	0	0	0	0	0	0	0
ConnectorMagnet *	0	0	0	0	0	0	0	0
Cover *	0	0	0	0	0	0	0	0
DCI_ChassisHandgrip *	0	0	0	0	0	0	0	0
DCI_ChassisMagnet *	0	0	0	0	0	0	0	0
DCI_HousingPins	9	0	2	0	0	3	0	0
Fan	4	0	0	0	0	0	2	0
HoldbackActuator	0	0	0	0	0	0	0	0
HomeSwitchAssy	0	0	8	0	0	7	11	0
HorizontalEncoder	0	0	20	0	0	0	0	0
HorizontalMotorAssy	9	0	0	0	0	10	13	0
LCC	0	0	0	0	0	0	0	0
LiftingJack	0	0	6	0	0	0	0	0
Microswitch	4	28	1	0	0	7	0	0
PICU3KB	3	0	7	30	0	6	6	0
PLC_Extension	10	0	5	0	0	4	6	0
PSControlUnit *	0	0	0	0	0	0	0	0
PSU *	0	0	0	0	0	0	0	0
SafetyRelay	0	0	0	0	0	0	0	0
SBM *	0	0	0	0	0	0	0	0
ServoAmplifier	0	25	0	0	0	0	0	0
Stop&Go *	0	0	0	0	0	0	0	0
StretcherSet	0	0	0	23	0	0	0	0
TableTop	0	0	0	0	53	0	0	0
Trolley *	0	0	0	0	0	0	0	0
Ventilation	0	0	0	0	0	0	18	0
VerticalActuator	0	0	0	0	0	12	0	50

Appendix CC Business Impact – FVF Cost Analysis

[Appendix unavailable due to confidential content]

Appendix DD BUSINESS IMPACT – TPR ANALYSIS

From the figures based on top 25 most frequent consumed (clustered) items it is striking that RF coils are dominating the different service cost intervals. Especially when it is generally assumed that such parts, among others, take multiple visits to replace. To understand why these parts are also included in the single visit population, we will look at the Technical Parts Review (TPR) process. It is assumed that parts included in TPR, per definition, will require multiple visits to replace.

Focusing on the dominating category of RF coils, we observe that x (100%) distinct coils have been consumed for single and multiple visit cases over the years, of which all appear at least once in the former subset, while only 72.1% appear in multiple visit cases. Comparing this list with parts that are included in the TPR process, it is observed that only 51% registered in the TPR list. Again, 100% of these have been consumed in single visit cases, and 63.6% required multiple field visits. Unfortunately, there is no distinction in 12Nc's used for both subsets, and therefore we cannot conclude that certain coils are replaced with one visit while others require more time and effort. Although, initially being surprised by a lot of RF coils present in single visit cases, it would seem plausible, as approximately half are not found in the TPR process.

Since the FVF analysis spans across multiple years (2013 – 2019), it is also useful to find out when the other parts have been added to TPR and if this inclusion has had any impact on the number of visits for these 12Nc's; which can further explain our striking observation. Unfortunately, data sources regarding TPR are not complete and accurate, even though – at the time of writing – the latest version has been used. Of the 51% 12Nc's, 27.3% are listed under TPR but have no available date-related information and another 31.8% only have a registered month and day of the month, but no corresponding year of addition to TPR; meaning that further conclusions are based on the remaining 40.9% of 12Nc's.

All these parts have been added to TPR either at the end of 2017 or 2018, whilst already being consumed during cases since 2013. We can clearly see that no parts have been consumed for three of those in single visit cases over the next few years, after their exact "added to TPR"-date. However, they are used in multiple visit cases. Unfortunately this is not the case for the remaining six 12Nc's, as they still are required for single visit cases before and after the TPR date, but a (significant) decrease can be observed in part frequency and amount of single visit cases (per year) for these, after the TPR date.

Aiming to understand why RF coils are also included in single visit cases, a careful conclusion can be drawn, based on a TPR analysis. *Assuming that spare parts included in the TPR process, per definition, require multiple visits*, we can understand why RF coils are consumed in single visit cases as roughly half of observed distinct 12Nc's are not included in TPR. Other that are have been includes just towards the end of 2017 or 2018, while being consumed in many cases in prior years. Once a part is added to TPR it seems to be consumed significantly less or not at all for single visit cases.
For specific part descriptions and 12Nc's observed in single and multiple visit populations, and relevant TPR information, see *Appendix L*. Additionally, this appendix also includes the distribution of RF Coils – as dominating part type in the above analysis – over the different service cost intervals.

Appendix EE LITERATURE REVIEW

Table 59 - Literature Review - Summary (1/2)

ID	Author(s)	Title	Year	Data Driven	Root Cause - Qualitative	Root Cause - Quantitative	Error (codes)	Failure Mode	ServiceActions / Parts	Association	Maintenance Field
1	Wu, Liu & Ding	A method of aircraft unit fault diagnosis	2003	X					X		~
2	Vinodh & Santhosh	Application of FMEA to an automotive leaf spring manufacturing organization	2011			X		X			~
3	Dorsch, Yasin & Czuchry	Application of root cause analysis in a service delivery operational environment	1997		X			X			
4	Chemweno, Pintelon, Van Horenbeek & Muchiri	Development of a risk assessment selection methodology for asset maintenance decision making: Ananalytic network process (ANP) approach	2015		X	X					X
5	Chemweno, Morag, Sheikhalishahi, Pintelon, Muchiri & Wakiru	Development of a novel methodology for root cause analysis and selection of maintenance strategy for a thermal power plant: A data exploration approach	2016	X	X	X		X	~		X
6	Zhu, Liyanage & Jeeves	Data-driven failure analysis of emergency shutdown systems in oil and gas industry	2018	X		X		X			Oil and gas
7	Shaker, Shahin & Jahanyan	Developing a two-phase QFD for improving FMEA: an integrative approach	2019			X		X			Steel industry
8	Lorenzi & Ferreira	Failure mapping using FMEA and A3 in engineering to order product development	2017			X		X			Automation comp.
9	Shahin, Labib, Emami & Karbasian	Improving Decision-Making Grid based on interdependence among failures with a case study in the steel industry	2018			X		X			X (Steel industry)
10	Braglia, Frosolini & Montanari	Fuzzy criticality assessment model for failure modes and effects analysis	2002			X		X			Manufacturing systems
11	Braglia	MAFMA: multi-attribute failure mode analysis	2000			X		X	~		Refrigerator manufacturing.
12	Adhikary, Bose, Bose & Mitra	Multi criteria FMECA for coal-fired thermal power plants using COPRAS-G	2013			X		X	for a few pars, general failure mode and cause listed		Coal-fired thermal power plant
13	Sharma, Kumar & Kumar	Modeling and analysing system failure behaviour using RCA, FMEA and NHPPP models	2007		X	X		X			X
14	Sanctis, Paciarotti & Di Giovine	Integration between RCM and RAM: a case study	2015			X		X			Offshore industry
15	Mohideen & Ramachandran	Strategic approach to breakdown matinenance on construction plant - UAE perspective	2012		X	X	X	X	~		Construction plant

16	Sharma, Kumar & Kumar	Systematic failure mode effect analysis (FMEA) using fuzzy linguistic modelling	2005	X		X		X			
17	Zawawy, Kontogianmis, Mylopoulos & Mankovskii	Requirements-Driven Root Cause Analysis Using Markov Logic Networks	2012	X		X	~				Loan Application
18	Mathur	Data Mining of Aviation Data for Advancing Health Management	2002	X		X		~	~		Aviation
19	Gusmão, Silva, Poleto, Silva & Costa	Cybersecurity risk analysis model using fault tree analysis and fuzzy decision theory	2018			X		~			Cybersecurity
20	Kang, Sun & Soares	Fault Tree Analysis of floating offshore wind turbines	2019	~		X	X	X			Offshore Wind Power
21	Sipos, Fradkin, Moerchen, Wang	Log-based Predictive Maintenance	2014	X			X		~		Siemens Medical Scanner
22	Okoh & Mehnen	Predictive Maintenance Modelling for Through-Life Engineering Services	2017	X					~		X
23	Lokrantz, Gustavsson & Jirstrand	Root Cause Analysis of failuers and quality deviations in manufacturing using machine learning	2018	~		X				X	X
24	Boutora & Bentarzi	Ferroresonance Study Using False Trip Root Cause Analysis	2019			X		X			Emerging and Renewable Energy
25	Soewardi & Wulandari	Analysis of Machine Maintenance Processes by using FMEA Method in the Sugar Industry	2019	-		X		X	~ (gen pred main service actions)		Sugar Industry, Machine Maintenance
26	Rezvanizaniani, Dempsey & Lee	An Effective Predictive Maintenance Approach based on Historical Maintenance Data using a Probabilistic Risk Assessment: PHM14 Data Challenge	2014	X							X
27	Zadry, Saputra, Tabri, Meilani & Rahmayanti	Failure Modes and Effects Analysis (FMEA) for evaluation of a sugarcane machine failure	2018		X			X	~ (gen pred main service actions)		Sugar Industry, Machine Maintenance
28	Zhao, Liu, Hu & Yan	Anomaly detection and fault analysis of wind turbine components based on deep learning network	2018	X							Renewable Energy (Wind Turbines)
29	Majumder, Sengupta, Jain & Bhaduri	Fault Detection Engine in Intelligent Predictive Analytics Platform for DCIM	2016	X		X		X	X	X	Airline flight check-in - Data Center Infrastructure Management
30	Gutschi, Furian, Suschnigg & Neubacher	Log-based predictive maintenance in discrete parts manufacturing	2019	X							Milling machines
31	Patil, Patil, Ravi & Naik	Predictive Modeling for Corrective Maintenance of Imaging devices from Machine Logs	2017	X							Medical Imaging Device Maintenance

Table 60 - Literature Review - Summary (2/2)

ID	Aim	Result
1	Aircraft unit fault diagnosis. Historic data, identify and classify faulty components. And determine fault frequency and probability of fault occurrence.	Supervised learning, Artificial neural network. Propose new method for fault probability of components, of which an aircraft unit is comprised, is generated by the adjusted SOM and fuzzy logic.
2	Report the application of failure mode and effect analysis (FMEA) to an automotive leaf spring manufacturing organization.	Quality of leaf springs produced also has been improved by an improved FMEA. Failure mode > potential effect > possible cause (description) > generic recommended action
3	Integrates a framework for the implementation of root cause analysis method to obtain a conceptual framework for an effective service delivery system	Cause-and-effect diagram for the voicemail investigation based on failure modes
4	Propose a selection methodology for risk assessment techniques in the maintenance decision making domain. Generic selection criteria for the FMEA, FTA and BN are derived, prioritized based on AnalyticNetworkProcess	General guideline for selecting appropriate techniques, as output selection of failure qual/semi quan root cause analysis method.
5	To propose novel data exploration methodology for root cause analysis is proposed which consists of four steps: 1) data collection and standardization step; 2) data exploration framework incorporating multivariate and cluster analysis; 3) causal mapping; and 4) maintenance strategy selection.	Cluster analysis (hierarchical, kmeans, fuzzy) for failrue associations. Per group/clustur of failure modes, determined if failure is critical, fixed by modifying component or failure is measurable. Additionally optional selection for strategies of fixing (FBM, CBM, or DOM) Methodology is comparedwith two conventional qualitative root cause analysis techniques – Ishikawa cause-and-effect diagram, and the '5-whys' analysis.
6	Develop a logical data-driven approach to enhance the understanding and detectability of ESD system failures	Proposed method to identify potential causes for failure modes. promotes a data-driven practice to implement failure analysis of ESD systems on different taxonomy levels. The study defines critical data sources and explains how they are used in the data-driven approach with an industry case. It is concluded that the understanding of failure mechanisms and the complex dependencies between different components and parts are helpful and even critical in the diagnosis of root failure causes in addition to data analysis.
7	To propose an integrative approach for improving failure modes and effects analysis: First phase of QFD: prioritizing failure modes based on failure effects, Second phase of QFD: prioritizing failure causes based onfailure modes	
8	To improve the failure analysis and troubleshooting process in engineering to order (ETO) product development, and reduce the amount of parts with failures.	Feasibility of the proposed method for both failure analysis and knowledge generation.
9	Decision-Making Grid (DMG) is used for determining maintenance tactics and is associated with the reliability and risk management of assets. PURPOSE: improve DMG by recognizing interdependence among failures. USING: Fault Tree Analysis and Reliability Block Diagram have been applied for improving DMG	Creation of FTA model and corresponding RBD diagram, compute new mean time to repair and frequency. Plot each failure mode on DMG canvas. Result suggestion of FTM, DOM or CBM strategy.
10	A tool for reliability and failure mode analysis based on an advanced version of the popular failure mode effects and criticality analysis (FMECA) procedure.	Probability distribution of the judgement given by maintenance experts is the output based on fuzzification. Paper provides these figures for the FMECA parameters (chance to failure, severity of its failure effect and chance of being undetected) for a potential cause.
11	Develop a new tool for reliability and failure mode analysis by integrating the conventional aspects of the popular failure mode and criticality analysis (FMECA) procedure with economic considerations	OUTPUT: raking list of potential causes for a failure for predefined FMEA criteria (chance failure, severity, expected cost and chance not detected) | Multi-attribute failure model analysis (MAFMA) appears to be a powerful tool for performing a complete criticality analysis on prioritising failures identified in a reliability study for corrective actions. MAFMA makes it possible to obtain a ranking of failure causes which includes several type of information (failure rate, non-detection, severity, expected cost for each fault). In particular, the use of an AHP-based approach for the multi-attribute analysis provides a framework with interesting characteristics for the selection process of the most critical cause of failure.
12	Present a multi criterion failure mode effect and criticality analysis for coal-fired thermal power plants using uncertain data as well as substituting the traditional risk priority number estimation method	

13	To analyze system failure behavior more consistently and plan suitable maintenance actions accordingly.	In-depth analysis of a system using RCA and FMEA helps to create a knowledge base to deal with problems related to process/product unreliability. From the results, it is observed that NHPPP models adequately analyse time-dependent rate of occurrence of failures. Thus, assisting the maintenance analyst in development of suitable aintenance strategy by properly understanding the mechanism of failure (through modeling of failure data); adopting adequate aging management actions (such as predictive or periodic testing) to predict or detect the degradation of components; and performing cost analysis
14		Establishing the failure mode belonging to each item and its effect on the system. All that allows us to calculate the items criticality that has been useful for the best maintenance strategy selection.
15	To develop a systematic strategic approach to handle corrective maintenance onto the failures/breakdowns of construction equipment.	Based on discussions, maintenance crew knowledge, breakdown codes were created using cause-effect methods. With FMEA methods, tables are created with fault description and potential sub codes and general solutions for predefined codes. Final: Fault tree with failures and separate tree for break down codes
16	To permit the system safety and reliability analysts to evaluate the criticality or risk associated with item failure modes.	Paper integrates the use of fuzzy logic and expert database with FMEA and may prove helpful to system safety and reliability analysts while conducting failure mode and effect analysis to prioritize failures for taking corrective or remedial actions.
17	Aimed to adopt a hybrid approach based on modeling the diagnostic knowledge as goal trees and on a probabilistic reasoning methodology based on Markov Logic Networks (MLNs).	Data-driven RCA method using markov-logic networks, which provides ranked diagnoses, is described. SQL is used to extract data that is used and the output is a list of possible RCF ranked on their probability calculated by the underlying logic.
18	Regarding diagnostics support, corrective (on-condition) maintenance, paper presents an idea to among others (such as system health and status analysis) assisting in fault-isolation and troubleshooting, and prognostics supports condition-based preventive maintenance by anticipating failures and recommending preemptive/preventive maintenance prior to catastrophic system failure. Paper actually uses system data (log files) for system health and status, and mentions it is usefull for fault isolation as well but this is still a vision.	Their database application currently includes the management of diagnostic models and the service of diagnostic and health status information to the onboard and ground-based fault isolation and troubleshooting tools. However, updating of such parameters using mined output of historical data collected from various fleets would greatly contribute to efficient maintenance by reducing time to troubleshoot and repair and improving availability - but not implemented or presented, just as a vision.
19	Proposes a model that integrates fault tree analysis, decision theory and fuzzy theory to (i) ascertain the current causes of cyberattack prevention failures and (ii) determine the vulnerability of a given cybersecurity system	Research shows that fault tree analysis and fuzzy decision theory complement each other, and were relevant to providing an effective definition of the causes of possible accident scenarios and a fuzzy assessment of potential accidents regarding cybersecurity risks
20	Fault Tree Analysis method is adopted for both qualitative and quantitative evaluation of semi-submersible floating offshore wind turbine failure characteristics	Output: not root cause in terms of maintenance, but failure probabilities for different failure modes/system failures. Analyzed based on a set of (incomplete in terms of system completeness) generic failure information. Calculated results are generally in conformity with statistical data, indicating that most of the failures are caused by several basic factors.
21	Data-driven approach based on multiple-instance learning for predicting equipment failures by mining equipment event logs which, while usually not designed for predicting failures, contain rich operational information.	
22	The analysis of the modelling uses synthetic data validated by industry domain experts. Develop a predictive maintenance strategy applicable to system reliability in the manufacturing, aerospace gas turbine, and other domains relative to concurrent system operations. The goal of this paper is to model and simulate an engine assembly in order to predict the number of parts expected to fail at a given inspection time.	
23	Propose a machine learning framework using Bayesian networks to model the causal relationships between manufacturing stages using expert knowledge, and demonstrate the usefulness of the framework on two simulated manufacturing processes.	The main findings are how knowledge from equipment experts can be used to pose a manufacturing process as a Bayesian network and in particular, how knowledge about links in this network as well as measurement nodes may improve the model. Next steps in the development of the framework will be to try it on a problem where we have continuous data and to use it for real world applications.

24	Root Cause Analysis (RCA) based on fault tree analysis has been used to identify root cause of the ferroresonance.	The quantitative analysis of the improved model shows An increase of about 30% of the security, which implies an appreciable enhancement of the reliability of the considered protection system. The use of the RCA in the ferroresonance study is a revolution in the predictive maintenance. This new fault diagnostic model opens the door for other researches based on the use of new methods like RCA, applied in many other fields.
25	Study aimed to investigate the potential failure of manufacturing process to provide some recommendations for improvement. Failure Mode and Effect Analysis (FMEA) method was applied to analyze the situation, Logic Tree Analysis (LTA) was implemented to classify the types of improvement	
26	Main objective is to develop a model based on first two years data set (training) and determine the high risk and low risk times of failure for each individual asset for the third year.	The paper presents the method in three main steps: the first step is to recognize the PM pattern based on time and type of maintenance activity via the training data. The second step is to determine the high-risk time intervals based on PM times by checking the frequency of the failures at specific times between each PM. The third step is to predict the high risk time intervals in the testing data using the information acquired from the training data
27	Evaluate the causes of failure in the use of sugarcane machine that have been designed in the previous studies. FMEA approach anticipated the failures at the design stage, so that a more reliable and ergonomic design can be produced for future sugarcane machine.	Study found that capacity issues are the priority problems that cause the machine failure. Then, this study proposed some actions to reduce the risk priority number (RPN) on 12 failures, using cause/effect diagram.
28	To achieve anomaly detection and fault analysis of wind turbine components, this paper proposes a deep learning method based on a deep auto-encoder (DAE) network using operational supervisory control and data acquisition (SCADA) data of wind turbines.	Method can be implemented for early warning of fault components and the effectiveness of the proposed method was verified by some reported failure cases of wind turbine components.
29	Novel, complete architecture of an intelligent predictive analytics platform, Fault Engine, for huge device network connected with electrical/information flow.	Fault Engine leverages log-data from concerned devices available in the device chain and employs a Markov Process based Failure Model to predict whether the failure is permanent or transient hence raising alarm with proper severity. It also indicates the recovery probability at any given time stamp after the failure has occurred. And, identifies possible the root cause devices in a situation of failure based on probability.
30	Data-driven approach for estimating the probability of machine breakdown during specified time interval in the future.	Machines failures can be reliably predicted up to 168 hours in advance using random forest machine learning.
31	Predict failures in turn resulting in reduced machine downtime, improved customer satisfaction and cost savings for the OEMs. Novelty also lies in the Health Domain.	Solution predicts component failure up to 14 days in advance of the actual failure