CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Exploring Automated Early Problem Identification Based on Diagnostic Trouble Codes

A Data-Driven Approach in the Automotive Industry

Master's thesis in Computer science and engineering

Mathias Gil Forsman
Yihan Yang

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2024

Master's thesis 2024

Exploring Automated Early Problem Identification Based on Diagnostic Trouble Codes

A Data-Driven Approach in the Automotive Industry

Mathias Forsman
Yihan Yang

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2024

Exploring Automated Early Problem Identification Based on Diagnostic Trouble Codes
A Data-Driven Approach in the Automotive Industry
Mathias Forsman
Yihan Yang

Supervisor: Hans-Martin Heyn, Department of Computer Science and Engineering
Advisor: David Issa Mattos,Volvo Cars
Examiner: Daniel Strüber, Department of Computer Science and Engineering

Master's Thesis 2024
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Description of the picture on the cover page (if applicable)

Typeset in LaTeX
Printed by [Name of printing company]
Gothenburg, Sweden 2024

Exploring Automated Early Problem Identification Based on Diagnostic Trouble Codes
A Data-Driven Approach in the Automotive Industry
Mathias Forsman, Yihan Yang
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

In the current automotive industry, problem identification is a reactive process. It starts when the customer experiences a vehicle problem and goes to the workshop. Subsequently, all the problem-related data will be collected from the workshop and forwarded to the vehicle manufacturer. After that, the engineers will start looking into the problem and figuring out the root cause with the cooperation from internal and external departments. It is a case-sensitive procedure and each unforeseen factor may further prolong the process and affect customer satisfaction.

This study cooperates with Volvo Cars to explore the possibility of providing a proactive data-driven insight into the problem identification process in the automotive system using the Diagnostic Troubleshooting Code (DTC). The purpose is to identify the most affected group before the problem scales and affects most of the customers.

This study involves two case studies and one laboratory experiment. The first-round case study helps to gain a better understanding of the current problem identification process. Also, some challenges and limitations encountered in this process have been identified. Other than these, five cases, including three different car parts: the car part A unit, the climatization system, and an add-on system, have been collected to conduct the following laboratory experiment. In total, four models are constructed and refined using several basic and machine learning techniques, including Group-by, Linear Regression, and K-means Clustering. This process evaluates different models' capabilities to provide early warnings and the corresponding correctness. It further assesses each technique's strengths and limitations in predicting the most affected group. The last case study serves as an evaluation action to receive feedback from the industrial experts about model performance and discuss the potential solution to integrate the model construction into the current workflow. In the end, a data-driven approach has been proposed and comprehensively described.

The influencing factors, advantages, and limitations of the research have also been discussed, leading to various interesting directions for future research.

Keywords: Automotive Industry, Early Problem Identification, Diagnostic Trouble Code, Case Study, Laboratory Experiment , Machine Learning, Linear Regression, K-means Clustering

Acknowledgements

We would like to give our special gratitude to our supervisor from Volvo Cars, David Issa Mattos whom assisted us through the entire research from the proposal to the final submitted version by providing consistent support from both knowledge and mental perspectives. His expertise and selfless dedication played a significant role in directing and refining our thesis project.

We express our appreciation to our academic supervisor, Hans-Martin Heyn. We proposed this thesis topic at an unusual time, making it challenging to find supervision. However, we are thankful that he agreed to supervise us, expressed a positive opinion about our thesis topic and aided us with feedback.

We are also grateful to our examiner, Daniel Strüber. He carefully reviewed our proposal, actively participated in our mid-term presentation, and offered various invaluable suggestions that enriched our research.

Last but not least, we would like to thank Peter Gunnarsson, Volvo Cars, and Chalmers for providing us with this precious opportunity.

Thank you all for being part of this rewarding journey!

Mathias Forsman, Yihan Yang, Gothenburg, 2024-01-30

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

A/C	Air Conditioner
AI	Artificial Intelligence
BEV	Battery Electric Vehicle
CNN	Convolution Neural Network
DAG	Directed Acyclic Graph
DIM	Driver Information Module
DTC	Diagnostic Trouble Code
ECU	Electronic Control Unit
FDD	Fault Detection Diagnosis
ML	Machine Learning
QE	Quality Engineer
RCA	Root Cause Analysis
SOP	Start of Production
SWPN	Software Part Number
TTP	Trigger-Total-Percentage
VIN	Vehicle Identification Number

Contents

List of Acronyms **ix**

List of Figures **xiii**

List of Tables **xv**

1 Introduction **1**

2 Related Work **3**

3 Background **5**
3.1 Troubleshooting 5
3.2 Machine Learning 6
3.2.1 Supervised learning 7
3.2.2 Unsupervised Learning 7
3.2.3 Normalization 8
3.2.4 Standardization 8
3.2.5 Correctness 8
3.2.6 Thresholds 8
3.2.7 Linear Regression 9
3.2.8 K-means clustering 9
3.2.8.1 Elbow Method 10

4 Research Methods **13**
4.1 First-round Case Study 14
4.1.1 Data Collection 14
4.1.2 Data Analysis 14
4.2 Laboratory Experiment 15
4.2.1 Data Preparation 16
4.2.2 Baseline Model 16
4.2.3 Process 17
4.3 Second-round Case Study 18
4.4 Threats to Validity 19

5 Results **21**
5.1 First-round Case Study 21

5.1.1 Introduction . . . 21
5.1.2 Thematic Analysis . . . 22
5.1.3 Problem Identification Process . . . 23
5.1.4 Limitations & Challenges . . . 24
5.2 Laboratory Experiment . . . 27
5.2.1 Reported Cases . . . 27
5.2.2 Correctness Calculation . . . 29
5.2.3 Model Development - Baseline Model . . . 30
5.2.4 Model Development - Refined Baseline Model . . . 31
5.2.5 Model Development - Multiple Linear Regression . . . 34
5.2.6 Model Development - K-means . . . 37
5.2.7 Model Development - K-means & Linear regression . . . 39
5.2.8 Model Comparison . . . 41
5.2.8.1 Development Cases . . . 41
5.2.8.2 Evaluation Cases . . . 48
5.2.9 Techniques Comparison . . . 50
5.3 Second-round Case Study . . . 52
5.3.1 Integration Solution . . . 54

6 Discussion **57**
6.1 Research Questions Summary . . . 57
6.1.1 RQ1 . . . 57
6.1.2 RQ2 . . . 57
6.1.3 RQ3 . . . 58
6.2 Influencing Factors . . . 58
6.2.1 Vehicle Dependencies . . . 59
6.2.2 Car Settings . . . 59
6.2.3 DTC Selection . . . 60
6.2.4 Expert Decision . . . 61
6.2.5 Correctness Calculation . . . 62
6.3 Advantages & Limitations . . . 63
6.3.1 Continuous & Dynamic Output . . . 63
6.3.2 Providing Data-driven insight . . . 63
6.3.3 Static Thresholds . . . 64
6.3.4 Feature Selection . . . 64

7 Conclusion **67**
7.1 Future Work . . . 67

Bibliography **69**

List of Figures

3.1 Supervised ML model 6
3.2 Unsupervised learning 7
3.3 K-means clusters 9
3.4 Elbow 10

5.1 Thematic Analysis Phase 2 23
5.2 Problem Identification Process 24
5.3 Data Collection in Problem Identification Process 24
5.4 Data Transformation in Problem Identification Process 25
5.5 Problem Pre-investigation in Problem Identification Process 25
5.6 Root Cause Analysis in Problem Identification Process 26
5.7 An example of correctness calculation 29
5.8 Equation 4.1: Trigger-Total-Percentage (TTP) Calculation 30
5.9 First-round Baseline Model Construction 30
5.10 Second-round Baseline Model Construction 31
5.11 Refined Baseline Model Construction 31
5.12 Baseline Refined Model Correctness (1) 32
5.13 Baseline Refined Model Correctness (2) 33
5.14 Equation 4.2: Multiple Linear Regression 35
5.15 Multiple Linear Regression Model Construction 36
5.16 K-means Model construction with Elbow methods 38
5.17 K-means & Linear Regression Model 40
5.18 Case 1 DTC 1 - Correctness Comparison 42
5.19 Case 1 DTC 3 - Correctness Comparison 42
5.20 Case 1 (DTC 2 & DTC 4) - Correctness Comparison 43
5.21 Case 1 (DTC 2 & DTC 4) - DTC reported date 43
5.22 Case 3 DTC 1 - Correctness Comparison 44
5.23 Case 3 DTC 2 - Correctness Comparison 45
5.24 Case 3 DTC 3 - Correctness Comparison 46
5.25 Case 3 DTC 4 - Correctness Comparison 46
5.26 Case 4 - Correctness Comparison (left) & DTC trend (right) 47
5.27 Case 2 - Correctness Comparison 48
5.28 Case 5 - Correctness Comparison 49
5.29 Case 5 - DTC trend 50
5.30 The Power BI plugin integration solution 54

6.1 Case 1 Correctness Comparison - all DTCs & individual DTC 61

List of Tables

4.1 Overview of the first-round interviewees 15
4.2 DTC Data Collection Table 16
4.3 Overview of the second-round interviewees 18

5.1 Laboratory Experiment Data Summary 28
5.2 Car Settings Types in Refined Baseline Model vs. Experts' Decision . 34
5.3 A portion of the feature summary table for Multiple Linear Regression 37

6.1 Number of Car Settings included in the Ticket Description & Final Experts' Decision 62

1

Introduction

The modern production cycle of the automotive industry consists of several stages: concept design, industrialization, launch, the start of production (SOP), and maintenance. The cost of any potential software failure increases over time. However, there is a notable difference in who finds the problem before and after the SOP. Before production, engineers identify potential failures and solve them before delivering the car to customers. Once customers receive the car, they may encounter problems during their daily use. These can raise concerns about the car's quality and impact their overall satisfaction with it.

On top of customers finding problems, it might take time to transition feedback from customers to engineers. This delay can vary due to factors like the complex communication between workshops, retailers, manufacturers, and suppliers, as well as the shipping of the problem car parts. Moreover, when customers report symptoms, they often do not come with a clear root cause, making the problem ambiguous and challenging to address properly. These symptoms do not point to any specific group, and multiple teams may be involved, resulting in high team dependencies. At Volvo Cars, the problems are collected by a central quality team and then categorized into different units, for example, vehicle platform, vehicle tophat, and vehicle propulsion with the local *Quality & Launch* team. This team conducts the Root Cause Analysis (RCA). This is the process to find the actual main cause of the problem that has occurred. Due to the technical dependencies between teams [1], additional time and action are required for this team to link the car symptoms to software problems and pass the error to the actual team that can solve the issues. During this process, there is a reduction in the user experience which may raise questions about the quality of the car.

To address these issues, it is important to take proactive action by identifying the most affected group, for example, vehicle type A, using B type of fuel which has been assembled in year C and released in market D, and passing the group information to the Quality Engineer at an earlier phase. This approach will save time and enable the engineers to provide an early-stage solution and resolve the issue before it happens and affects a large number of cars. Studying previous Diagnose Trouble Code (DTC), which is a code used to alert of a specific problem in a vehicle can be beneficial in this case.

The primary objective of this study is to devise a systematic, data-driven approach for early problem detection with the help of DTC and ML techniques. Through the

application of these ML techniques and the development of the intended methodology, the goal is to benefit both Volvo and its customers. With the proposed automated method, we can explore the possibility of computer scientific methodologies and how they optimize the automotive industry by providing data-driven insight into part of their problem-identification process, thus reducing resolving time and warrant cost. Proactive action and shorter waiting time will enhance the user experience and build customer trust.

We employ a multi-method approach in this study, including two case studies and one laboratory experiment. Initially, we conduct a case study to interview industry experts from the problem identification and resolution field, including product owners, quality leaders, program managers, and scrum masters to senior engineers and specialists, to gain a better understanding of the current entire problem identification process. Second, we prepare the data for the upcoming laboratory experiment. The data are the raw anomaly DTCs collected from the workshop according to the cases discussed during the interviews. In total, five cases are selected. After formatting the data, a laboratory experiment is performed to investigate the possibility of employing ML to construct four models for early problem identification. The model is expected to identify and prioritize the most affected groups based on various factors, such as technical model year, vehicle type, market, fuel type, platform, assembly plant, assembly week, and software part number. Two Machine Learning algorithms, Linear Regression and K-means Clustering, are examined and compared between these models and the baseline. In the end, a follow-up case study is used to interview the industry experts again. Together, we compare and evaluate the performance of the models, figure out the possible approach that could integrate our model into their current workflow, and discuss the current model's influencing factors and limitations to provide direction for future studies.

The structure of this thesis is organized as follows. Chapter 2 reviews some related works in nowadays problem identification and fault detection fields. Chapter 3 explains the background knowledge for this thesis, such as troubleshooting, a brief introduction about Machine Learning and Machine Learning algorithms, and some knowledge that is related to the study. Chapter 4 formulates three research questions and introduces the research methods, including case study and laboratory experiment, that are employed in this thesis. Toward the end of this chapter, we talk about the threats to validity. Chapter 5 presents the results from the previous research methods and interprets them to address the research questions. Chapter 6 delves into the notable findings we encounter during the study and their impact on this thesis. Besides, it mentions the current model's advantages and limitations. Finally, Chapter 7 concludes this study and arises the discussion about potential research directions for future work to refine the current methodology.

2

Related Work

As industrial processes become larger and more complex nowadays, the difficulties in conducting early problem identification also increase. Combined with the need to reduce warrant costs and improve profitability, the current research leans toward applying machine learning techniques to provide more data insight into this process. The general goal is to identify the problem and take action at an early stage before the issue affects the customer. Here are some related works that we reviewed.

Wilhelm, Reimann, Gauchel, *et al.* mentioned research on hybrid approaches for fault detection and diagnosis, which involve the combination of data-driven, physics-based, and knowledge-based models [2]. These approaches are then categorized and provide a foundation for the integration of methodologies in early problem identification.

Al-Zeyadi, Andreu-Perez, Hagras, *et al.* applied a deep learning approach to improve fault diagnostic systems to overcome the challenges encountered in traditional rule-based diagnostic systems within the automotive industry [3]. A deep symptoms-based model was designed to estimate required services by predicting a range of faults within the vehicles. Pirasteh, Nowaczyk, Pashami, *et al.* utilized DTC to forecast the condition of components in heavy-duty trucks [4]. They employed machine learning techniques, such as random forest, to predict whether a component in the trucks is likely to fail in the future. This research generated its predictions on the analysis of the DTC-triggered situation of the component, aiming to foresee its future maintenance needs. Virk, Muhammad, and Martinez-Enriquez argued for the importance of fault monitoring. They explored machine learning techniques, such as artificial neural networks and fuzzy logic models, to develop a fault prediction service that forecasts the remaining life of vehicle components such as batteries [5]. Gong, Su, Chen, *et al.* underlined the significance of predicting vehicle faults, especially in autonomous unmanned vehicles [6]. Broadwell employed statistical techniques, including naive Bayes classification, to predict and prevent hardware failures [7].

The previous research has focused on predicting when and which component will fail. Our study distinguished itself by focusing on shortening the problem identification process with the help of analyzing which group will be affected most automatically before the problem actually influences at scale.

3

Background

This chapter aims to introduce the necessary theory behind the thesis to provide the reader with a brief understanding of the related background knowledge. Firstly, we discuss the modern way of troubleshooting and review the related work in this field. Secondly, we talk about machine learning, including supervised and unsupervised ML. We also mention possible algorithms that could contribute to the model construction.

3.1 Troubleshooting

DTC is the acronym for Diagnostic Trouble Code. It is produced by the Electronic Control Unit (ECU) and can be treated as an information carrier of the result for the build-in test in the ECU. This signal is widely utilized in the automotive industry to monitor car performance and analyze vehicle issues [8]. In 2012, SAE International standardized the format of DTC. The presentation format is an array composed of two parts. The first part is called Base DTC, which describes the failure car component or function. The second part is Failure Type Byte which identifies if it is a systematic, electrical, or mechanical problem and outlines the possible symptoms [4].

When a DTC is triggered in a vehicle, it takes a snapshot of the logs and then saves them in the ECU with other useful information, including status bits, timestamps, fault detection counter, operation cycle counter, and status indicators. When a customer experiences issues with a car, or simply for maintenance, they visit a workshop where the DTC data is extracted from the vehicle ECU and then forwarded to the vehicle manufacturer. In our case, it is Volvo Cars.

There are different purposes when using DTC. Some of them are designed to fail (get triggered), and some of them indicate failure situations. A triggered DTC does not imply that a vehicle has an issue, it simply states that a DTC has been triggered, it is by design and could be due to monitoring purposes or indications of potential issues that might require attention in the future. DTCs can have a co-relation with each other, but in this thesis, we treated each DTC individually. Moreover, a DTC alone does not determine the severity of the issue. A DTC can have a straightforward explanation in a simple situation, however, it can also function as an indicator for a serious incident in another case depending on the symptoms and environment settings [9]. Ferreira et al. mention in this paper that in practice, vehicles often run with some active DTCs and that it is not a threat at all, we need to cooperate

closely with industrial experts to identify the significant abnormal DTCs relating to the car symptoms.

After collecting useful DTCs, performing RCA is the next step in the troubleshooting process. RCA is a structural process to address the causal factors of a problem [10]. RCA is an important scientific method in many research areas. For example, in manufacturing, it is applied to reduce failure, enhance product quality, and further improve the entire manufacturing operation [11] [12]. It also shows a significant effect in medical research and healthcare areas that it can help to decrease the recurrence of avoidable adverse events [13] [14]. At Volvo Cars *Quality & Launch* team, once confirmed car symptoms are reported and relevant DTCs are found, an RCA will be performed to identify the reason that made the problem happen.

3.2 Machine Learning

Machine learning (ML) is a subset of artificial intelligence (AI), it focuses on the usage of data and algorithms to gradually learn and improve. In this thesis, we use historical time-series data as input for our algorithms. An ML algorithm is a computational process that given input data, wants to achieve a desired task without being specifically told what the outcome is supposed to be. They can resemble "soft code" in that sense [15]. Typically, an ML algorithm improves its results through numerous iterations and eventually creates a trained model to recognize patterns. During the training process, data is fed to algorithms to build a model. The model leverages the inputted historical information to make further decisions. ML can be categorized into two types, depending on whether the data is labeled: **supervised learning** and **unsupervised learning**. ML draws upon the fields of AI, probability, statistics, and computer science.

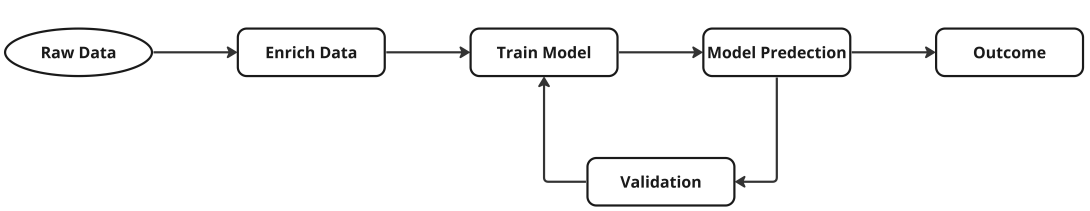

Figure 3.1: Supervised ML model

Figure 3.1 shows the generic flow of a supervised ML algorithm. It starts with raw data and proceeds to format, enrich, and adapt it to accommodate it into the ML model. The ML model then learns and makes some output. The output is a prediction that could be the predicted target, or it could need further adjustments to reach the predicted target.

In machine learning, there is a difference between training data and testing data. Training data is the data set used to train a model to predict an outcome. Test data is the data set that is used to measure the accuracy and efficiency of the algorithm

that is used to train the model. Training data is used to train the model and then validated through the use of testing data which is used to test how well the model performs based on the training data. It is two different sets of data, training data, and test data typically different, the test data is used as a complement to the training data to verify that the algorithm computes as expected.

3.2.1 Supervised learning

Supervised learning is a subset of machine learning where the training of models is supervised. Algorithms learn patterns and associations from data that is labeled to make predictions or decisions on new data. Typically this strategy involves providing model data sets that contain both input and corresponding output labels. There are two primary types of supervised learning, classification, and regression. With classification, the algorithms predict discrete categories or labels. Regression however involves predicting numerical values such as forecasts based on variables. During the training of supervised algorithms, they learn from the labeled data set by taking an iterative approach and adjusting its parameters to minimize the difference between the values that are being predicted and its actual values. A typical example of supervised learning is where historical data with known outcomes can be taken advantage of and predictions or classifications.

3.2.2 Unsupervised Learning

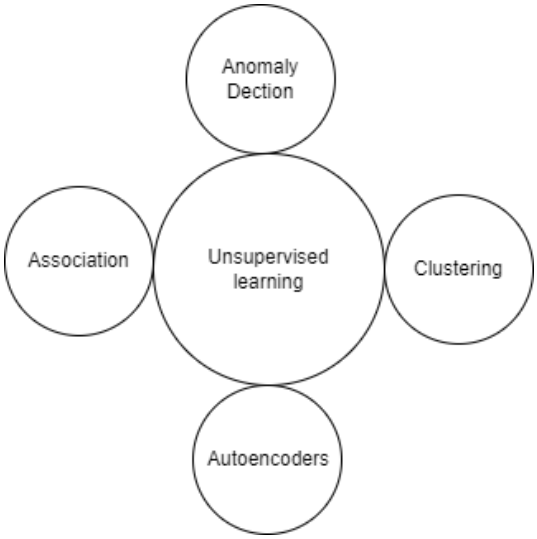

Figure 3.2: Unsupervised learning

Unsupervised learning is a type of machine learning where the algorithms purpose is to identify patterns and structures from the data without using specific guidance or labels. Unsupervised algorithms' typically work with unlabeled data which it then finds patterns in. These characteristics of unsupervised learning allow the algorithms to find relevant information such as the identification of clusters that otherwise might

not be apparent. The algorithms can dynamically adjust their parameters to capture data structures.

There are four types of unsupervised learning tasks as shown in figure 3.2. Rupali, Dixit, and Anil argue that clustering is the most important one and that it can be defined as "the process of organizing objects into groups whose members are similar in some way" [16]. Figure 3.3 displays a common clustering technique called K-means clustering and we also utilize this algorithm to our research.

3.2.3 Normalization

Normalization uses numeric feature scaling within specific ranges which are usually 0 to 1. It is achieved by transforming the values of a feature based on its min and max values. The purpose is to bring the features to a standard scale without altering their distributions. Normalization is meant to ensure fairness, guard against bias, and provide a smoother convergence during the training of a model.

3.2.4 Standardization

Standardization, also known as Z-score normalization is a data preprocessing technique used in machine learning. It uses the mean of 0 and standardizes the variance to 1 by adjusting the values based on the standard and mean deviation. It transforms the distribution of the features to have a mean value of 0 and a standard deviation value of 1. The goal is to transform the data set to share a common scale to make them compatible for model training and analysis. Standardization keeps the distribution shape of the data intact which preserves the relationship between the data points and ensures that the relative difference between the values remains consistent.

3.2.5 Correctness

Correctness is a quantitative assessment of how the model output aligns with the expected data. In our case, a larger correctness value represents a better similarity between the model result and the expert's decision. However, we can not only contribute to getting a statistically big value since it may also lead to over-fitting. The specific calculation is described in Section 5.2.2.

3.2.6 Thresholds

Thresholds are used to influence the algorithm's final result. In most situations, this factor is used to filter out unqualified or unnecessary data points. The purpose is to further enhance the quality of the result. In our case, we utilize several thresholds to control the data volume, remove the unnecessary groups, decide if a warning should be given, and identify the most affected group. The various usage of thresholds in different models can be found in Section 5.2.

3.2.7 Linear Regression

The simplest form of linear regression is an algorithm that provides a linear relationship between an independent variable and a dependent variable to predict the outcome of future events. The model aims to predict the value of the dependent variable which is the outcome of the model. With linear regression, there is an assumption that there is a linear relationship between the input variables and the output variable [17]. There is another form of linear regression called Multiple linear regression where several predictors contribute to predicting the said outcome variable. It is a statistical tool used to model the relationship between variables and to make predictions of the target variable.

3.2.8 K-means clustering

K-means have previously been used in the automotive industry, specifically in anomaly detection [18]. It requires low computational power and is both easy to understand and implement. K-means clustering is a clustering type of algorithm that is classified as unsupervised learning. The goal of k-means is to group similar data points and discover underlying patterns in the data set. K-means looks for a set number k of clusters in the data set. K-means performs an iterative calculation to optimize the position of the centroids which are randomly selected to start with.

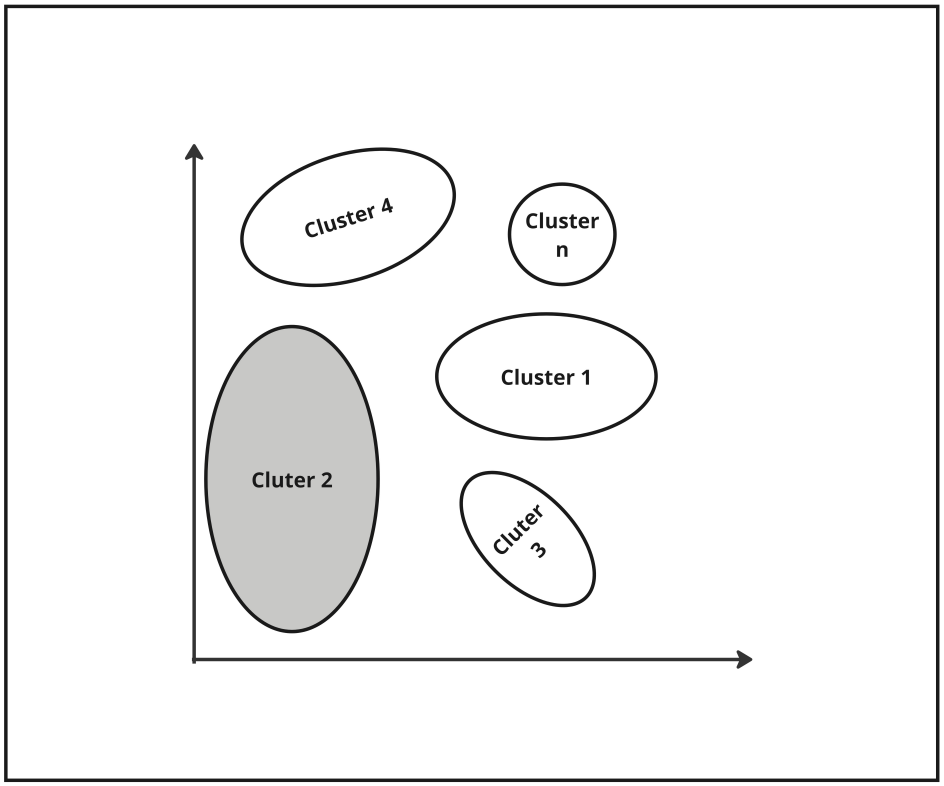

Figure 3.3: K-means clusters

Algorithm 1: K-means Clustering Algorithm

Input: Data points, K (number of clusters)
Output: Clusters
Parameters: K - Number of clusters

Step 1: Define the number "K" which is the number of clusters;
Step 2: Choose random K points or centroids;
Step 3: Assign the data points to their nearest centroid to create K clusters;
Step 4: Calculate the variance and enter a new centroid of each cluster;
Step 5: Repeat Step 3 and keep reassigning each data point to the closest centroid of the latest cluster;
while *any reassignment happens* **do**
    Step 6: Perform Step 4;
    Step 7: Check if any reassignment happens;
end
Step 8: End;

The result is a set of clusters which is represented by their centroids. The data points within those clusters indicate some sort of association.

3.2.8.1 Elbow Method

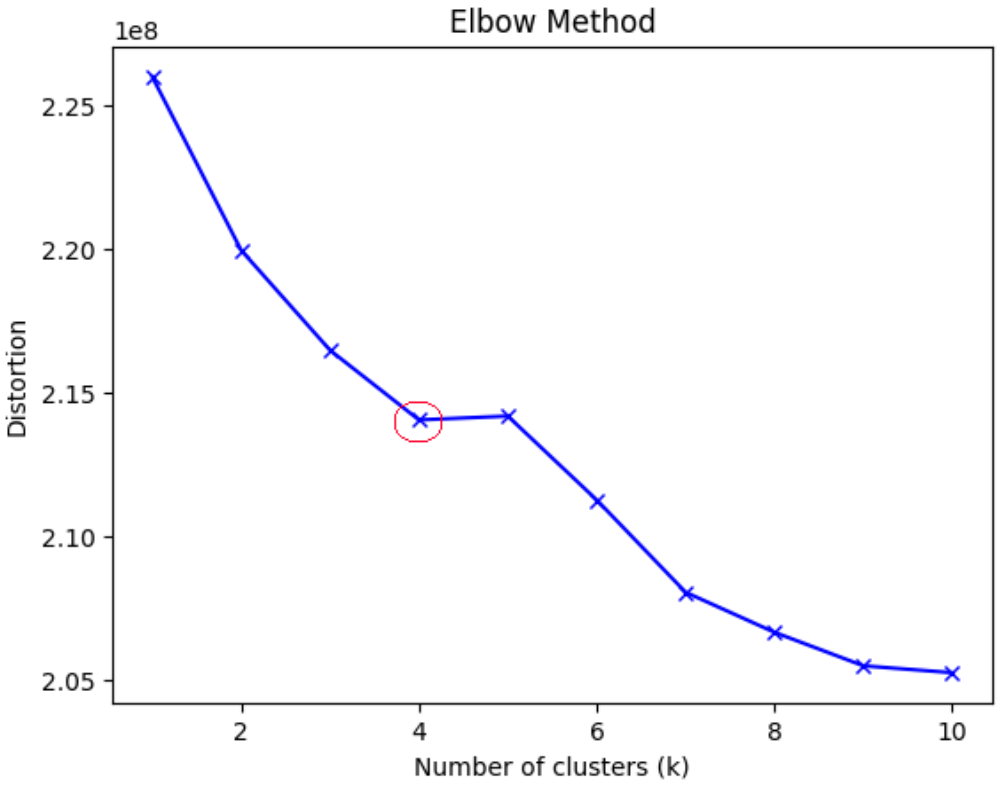

Figure 3.4: Elbow

Various methods calculate the optimal K value since the number of clusters affects the way that K-means work [19]. The elbow method is one of them that calculates the squared difference of different K values [20]. As seen in the example figure 3.4, the elbow is located at 4 which means the optimal K value is 4 in this case. In other words, the data set will be divided into 4 clusters.

4

Research Methods

This chapter discusses the applied research methods that have been conducted throughout the study. In the beginning, we present our research questions and how they relate to the goal we want to achieve. Then, we introduce the multi-method approach, including case studies and laboratory experiments to answer the questions and accomplish the purpose.

This thesis aims to shorten the problem identification process by identifying the most affected groups of possible issues earlier with the help of DTCs and ML. The group is classified by various car settings, including model year, vehicle type, energy source type, platform, assembly plant, assembly week, and software part number. We formulated three research questions that helped us to break down the main issue step by step. The first research question focused on understanding the entire problem-identification process. The second research question aimed at exploring and evaluating the potential new methods to solve the problem. The third research question intended to propose a solution that integrates the model construction into production and provides proactive capability.

- **RQ1: What is the current process of problem identification using DTCs, are there any limitations and challenges encountered in the case company?** By investigating the current process, we can learn from previous experience, and it can help pave the way for proposing a more effective alternative. This is adopted into a single case study with the help of interviews with the relevant personnel, to discuss the strengths, weaknesses, problems, limitations, and so on.
- **RQ2: Which techniques can be applied to overcome some of the addressed limitations and challenges of the current process for early problem identification based on DTCs?** After gaining some knowledge from addressing the previous research question, we aim to conduct a laboratory experiment to transfer this knowledge into practical approaches by exploring the application of ML. In the beginning, we construct a baseline model without using any ML algorithms. Besides, we apply two ML algorithms for constructing and refining three ML models. In the end, we compare the model performance between them and analyze the ability of each technique.
- **RQ3: How can the proposed solution be integrated into the current problem identification process in the case company?** This is the

validation process of the new methodology. We conduct a second round of case study with the industrial experts again aiming to receive feedback about our model construction, discuss the possibility of merging our algorithm with the current workflow, and figure out the best practice.

In the subsequent sections (4.1 and 4.2), we delve into the details of the research methods, dataset, processing methods, model construction, and evaluation process. These sections provide a comprehensive explanation of our methodology and address the missing details highlighted in this feedback.

4.1 First-round Case Study

Conducting a case study in collaboration with industrial experts in this field can help us quickly acquire empirical knowledge for the problem-identification process. In alignment with the five steps guided by Runeson and Höst, we will firstly design the case study, secondly, prepare the data collection, thirdly perform the data collection process, fourthly analyze the data we obtain, and finally report the results [21].

4.1.1 Data Collection

We collected data from various resources, including slides, documentation, transcripts from the semi-structured industry expert interviews, and tickets related to the problem identification process. The slides and documentation were provided by Volvo Cars which can be found on its intranet. Additionally, to gain insight and a better understanding of the current problem identification process in production, we conducted interviews with eight experienced engineers, ranging from product owners, quality leaders, program managers, and scrum masters to senior engineers and specialists, from two different teams. They have been working in this area from 4 to 12 years. In their daily work, they focus on various aspects of the actual production, including management problems, plant issues, car launches, three months of in-service car problem reports, and current model quality. They also work in three subsystems of the car, including the climatization system, car part A unit, and add-on system. Table 4.1 provides an overview of these interviewees. Each interview was conducted in English and lasted approximately 30 minutes, involving two interviewers and one interviewee. The interviews were conducted between September and October 2023. We performed a semi-structured interview with questions prepared for each interviewee based on their role, depending on their answers it led to follow-up questions as we had flexibility in our predefined questions.

4.1.2 Data Analysis

To analyze the data obtained from the interviews, we followed the six-phase process of thematic coding proposed by Braun and Clark [22]. Thematic analysis guided us in collecting and identifying the possible themes to shorten the problem-identification process. It also directed the subsequent laboratory experiment. The list below comprehensively describes various phases:

Interview	Role	Years of Experience
A	Product Owner	10
B	Product Quality Leader	5
C	Senior Quality Engineer	8
D	Warranty Specialist	12
E	Scrum Master	10
F	Program Manager, Quality & Launch Lead	10
G	Senior Quality Engineer	9
H	Product Owner	4

Table 4.1: Overview of the first-round interviewees

- The first phase consisted of getting familiar with the data. This was accomplished through interviews, internal documents, and transcripts of the conducted interviews
- The second phase involved generating the first set of code inspired by the first phase. Each concept and thought from the first phase could be treated as an individual code.
- The third phase discussed the potential themes for the codes and models. After collecting sufficient codes from the second phase, some of them were grouped and abstracted into themes. There were different levels' themes, which might have correlations with others.
- The fourth phase reviewed and refined the identified themes. Since the themes gathered from the third step were numerous and diverse, we needed to sort, combine, and elect them to a smaller amount. The theme's validity to the overall goal was also under consideration.
- The fifth phase analyzed each selected theme and generated the results and discussion points of the research questions. This phase also involved removing company-specific information which was sensitive data specific to Volvo Cars.
- The sixth and last phase was the publication, the results were presented, and the research questions were answered. It could also guide future research questions, or raise further discussion for future work.

4.2 Laboratory Experiment

Laboratory experiments are commonly used to transfer the expert's knowledge acquired from the previous case study to advanced practical methods. They allow the researcher to isolate the phenomenon from their context in a controlled environment [23]. In this study, we used this approach to simulate the most affected group, such as vehicle type, market, and assembly plant, based on the DTC and other informative data in a predefined configuration where every DTC was treated individually, they did not affect each other.

Due to confidentiality, we can not state the exact size of our dataset. Still, the data amount is appropriate and guaranteed to support the application of ML techniques that were used. Each row in our data set equals a vehicle and we have a large amount of unlabelled data provided by Volvo which are used for our laboratory experiments.

4.2.1 Data Preparation

In the first-round case study with eight engineers, we investigated five cases reported by customers due to irregular behaviors. These cases are related to three subsystems, including a climatization system, the car part A unit, and an add-on system. Their DTC-triggering situations have been recorded, and the related data has been forwarded from workshops to Volvo Cars.

This data collection spanned 6 months, capturing information six months before each case was reported. We collected the following raw data from Volvo's database for each case: Vehicle Number (VIN), Model Year, Vehicle Type, Energy Source type, Platform, Assembly Plant, Assembly Week, Software Part Number (SWPN), Readout Date, and if the car has triggered the related DTCs that match and explain the vehicle symptoms. Some issues only occurred in specific countries or markets, for these cases, we would also gather information on the market where the car was sold. It's important to note that VIN has been anonymized to ensure privacy. Additionally, Assembly Week refers to the week in which the vehicle was produced. The DTC is selected based on the car symptoms, the related Electronic Control Unit (ECU), and expert experience. Table 4.2 shows an overview of the cases. To enhance the data interpretability, we transfer the readout date to its corresponding week according to Volvo Cars' calendar, as they operate on a weekly basis. Each row of data that we collected was a vehicle with the columns previously mentioned. We can not disclose the amount of rows we had due to confidentiality other than it is an appropriate amount for ML purposes.

Case No.	Car Part	Number of DTC(s)
Case 1	Climatization System	4
Case 2	Car Part A Unit	1
Case 3	Car Part A Unit	4
Case 4	Add-on System	1
Case 5	Climatization System	1

Table 4.2: DTC Data Collection Table

4.2.2 Baseline Model

During the laboratory experiment, we constructed a baseline model to serve as a reference point for comparing the performance of more advanced and sophisticated models. It was a simple yet reliable benchmark that offered a basic level of predictive ability that the more advanced models were expected to surpass. We developed this model with straightforward and intentionally simple techniques such as group-by and threshold filtering. If a more complex model could not outperform the simplistic

approach of the baseline, it was a way to rethink the approach and reconsider the model. To better align the baseline model's result with the expert's decisions, we made 3 iterations of the baseline model. Adding thresholds each time further refined the model and resulted in a more appropriate and realistic output.

4.2.3 Process

We separated all five cases into two groups, the model construction group and the model evaluation group. Our grouping method also ensures that the affected car parts are evenly distributed among these two groups. The group designated for model construction consisted of Case 1, Case 3, and Case 4. This group included the vehicle climatization system, car part A unit, and an add-on system. The second group, the evaluation group, encompassed Case 2 and Case 5, including the vehicle climatization system and car part A unit. For each case, every DTC was treated individually. Each experiment contained only one DTC. As shown in Table 4.2, Case 1 included four DTCs, Case 3 included four DTCs and Case 4 included one DTC, in total, nine experiments were conducted for one model construction. For model evaluation, since both Case 2 and Case 5 included one DTC, two experiments were carried out. Besides, one calculation included data from five weeks, the current week, and the previous four weeks.

At first, we prepared and built a baseline model as the minimum acceptance condition. It simply identified the most affected group by outputting the group with the highest percentage of triggered-DTC vehicles divided by the total vehicles in one calculation circle. To align the model result with the experts' views, we conducted two more iterations to refine it using threshold controls and obtained the final baseline model with two threshold controls. The thresholds improved the models as a kind of filter, for example, we filtered out the groups that had less than 100 vehicles to reduce the system providing unnecessary warnings.

After the refined baseline development, we ran two selected ML algorithms and constructed another three models to explore and compare the ML feasibility for this case.

The first one was a foundational machine learning model using Linear Regression. This algorithm helped us find the best-fitting linear relationship between various variables. In our case, we treated various types of car settings as the independent variables and used the DTC-triggered situation to label the group. In this model, we use the p-value and coefficient value to select features. A low p-value rejects the null hypothesis and indicates a significant relationship, a high and positive coefficient value also represents a strong relationship between variables.

Secondly, inspired by Guerreiro et al. to use unsupervised ML algorithms for grouping similar components and performing anomaly detection in the automotive industry, we utilized one of their mentioned clustering techniques called K-means to construct an advanced model [24]. By employing this algorithm, we could cluster all the vehicles into several groups based on their performance and select the most affected ones by observing the DTC-triggering situation in each group. Differing from the

baseline model, which only outputted the group with the highest proportion of trigger-DTC cars among all cars (usually only including one group), this method helped us integrate more groups with various types of vehicles to further improve the coverage of outcomes.

In the end, we built a fusion model that combined K-means clustering and Linear Regression. The first half of the process was similar to constructing a single K-means model. However, after receiving the clustering result and outputting the groups included in the most affected cluster, we fitted all the features in each group to a Linear Regression model and filtered based on each feature's p-value performance. As we mentioned before, the p-value can determine whether the relationship between variables is statistically significant. we eliminated unnecessary features, outputted statistically influential features, and further improved identification accuracy.

We evaluated the models by comparing their results with the baseline model, the running time, the alignment situation with the experts' decision, and the earliest time that they were able to identify the most affected group.

4.3 Second-round Case Study

After the laboratory experiment, we conducted another case study. Similar to the first round, interviewing the industrial experts took up a large portion of this process. In total, six people were involved in the second-round case study, all of whom had participated in the previous round. The interviews spanned from the end of November to the middle of December 2023. Each interview was conducted in English and lasted from 25 to 40 minutes. Table 4.3 provides a summary of the participants. It is important to note that the correctness used in these interviews was taken from the cases after RCA had been applied which does not necessarily align with the correctness of the ticket when initially reported.

Interview	Role	Years of Experience
A	Product Owner	10
B	Product Quality Leader	5
C	Senior Quality Engineer	8
D	Warranty Specialist	12
E	Scrum Master	10
G	Senior Quality Engineer	9

Table 4.3: Overview of the second-round interviewees

The first part of the interview was to introduce our four model construction methods and present the results for each of them. We also showed the model output and created some comparison graphs to compare the correctness of each model result, and the earliest time that we were able to identify the most affected groups. The second part consisted of a question session where we asked about their opinions regarding the experiment settings. For example, we discussed whether car settings

should be treated equally, which car settings had a correlation with others, and if the thresholds had been properly selected. Additionally, we presented various scenarios to observe their reactions. The last part is to decide with the experts on the feasible and applicable solution that could integrate our model construction into their workflow and aid them in making better decisions.

4.4 Threats to Validity

We identified four validity threats in this study: Internal Validity, External Validity, Construct Validity, and Conclusion Validity [25]. In this section, we will discuss them and propose strategies to mitigate these threats in this study.

Internal Validity: This refers to the extent to which researchers can confirm that the outcome is influenced by the manipulation of the independent variable [26]. Confounding factors pose a significant challenge in software engineering studies and can undermine internal validity [27]. To reduce the testing effects and instrumentation in our study, we tried to conduct all research methods in a controlled and identical environment. For example, all the settings for the interview were the same, including the personnel composition, background introduction section, and the basic questions in the first half session. For the laboratory experiment, we constructed the model and simulated the results using the same data and experiment settings. In the end, we evaluated the models based on the same criteria.

External Validity: This refers to the ability to extend the findings of the current specific study to other general situations [28]. Factors such as selection bias, limited diversity of contexts, and strict study settings can threaten external validity. In this study, we made an effort to interview experts with different roles, and different years of experience and focus on different car parts to minimize the selection bias. We also tried to select different industrial incidents and abstract the common procedures in their problem-identification processes to generalize the future usability of our approach.

Construct Validity: It verifies if our scales, metrics, and instruments accurately measure the intended properties [29]. The major threats are the biased definition of the concept and the incorrect use of measurement indicators [30]. In our study, we cooperated with the industry supervisor, and experts from Volvo Cars to appropriately define the scope and verification measurements. During the interview session, we selected diverse participants working on this concept and utilized a semi-structured approach to keep the questions and answers on the correct track. Besides these, we had follow-up meetings with our supervisor once a week, reviewed meetings with a quality manager every second week, and maintained continuous email communication with the academic supervisor. All of these actions were aimed at enhancing construct validity.

Conclusion Validity: Strong conclusion validity suggests a stable correlation between the statistical methods, procedure, and conclusion [31]. França and Travassos also point out that inappropriate experiments, incorrect data, and unnoticed independence between factors can threaten the conclusion's validity. To mitigate these threats,

we interviewed industrial experts currently working in this area, used realistic production data provided by Volvo Cars formulated and discussed ML algorithms with the supervisors, developed verification criteria, and validated the approach in collaboration with the relevant participants.

5

Results

In this chapter, we present and interpret the results of the research methods, including two case studies and one laboratory experiment. At the beginning, we introduce the first-round case study and show the outcomes of the thematic analysis. Then, we summarize the current entire problem identification process and place additional emphasis on the limitations and challenges encountered by Volvo Cars and addressing **RQ1**. Secondly, we start by discussing the usage for five production cases gathered from the first-round case study. Three of these are involved in the development of four models, while the other two serve as the groups to compare and evaluate the models. By analyzing and comparing the models conducted in the laboratory experiment, we discuss the techniques that can be applied to overcome some of the mentioned limitations and challenges in the current problem-identification process based on DTCs. Therefore, we answer **RQ2**. Thirdly, we introduce and present the results from the second case study, including a follow-up interview. The purpose of this case study is to verify and validate the final methodology in collaboration with industry experts and figure out the best practices for integrating this method into the current workflow inside Volvo Cars. Hence, we respond to **RQ3**.

5.1 First-round Case Study

The objective of the case study is to understand the current problem-identification process and to gather information about cases that can be used for the laboratory experiment part.

5.1.1 Introduction

We aim to provide a brief introduction to how industrial experts explain the current process of problem identification.

Interviewee A explained the process as follows: "“*We got a vehicle report from the workshop, and the QE (Quality Engineer) started looking at the DTC. It could be a known DTC or it could be a new DTC. Either way, we analyzed the DTC to find differences or correlations.*”The vehicle report was the start of the problem identification process; it provided a list of DTCs that had been triggered, and the dealers could determine which DTCs were relevant or not.

Interviewee B explained and walked through the internal programs used to identify

the problem causing the DTCs. This process aligned with Interviewee A's description, but B specifically focused on the programs that were used to generate metadata to find the root cause of why the DTC had been triggered. Interviewee B attempted to isolate the symptoms to some certain DTCs. Once the DTCs had been confirmed, the corresponding teams could conduct to fix the issue. Interviewee B explained that symptoms could have consequences, meaning that there were critical DTCs and then there were consequences; on the timestamp, it was possible to filter out by seeing which triggered first. The process was explained as: "“*Look at the symptom, readout time, and date. Make a story out of it; every case is like a new story where we tried to connect the dots and find a red thread through it to see if it is plausible. Time and symptoms are very important.*”

Interviewee B also explained what happened when an unknown DTC had been triggered; they asked the dealer about the date and time to see if they could correlate them 'one to one.' Then they erased the DTC to see if the DTC was reproducible. If the internal tests kept failing and this DTC showed up again, it meant the DTC was confirmed. An important note was that once a test failed at least twice, it became confirmed; one failure did not necessarily mean that an issue was confirmed; it could be a glitch.

Interviewee C mentioned the same internal programs as Interviewee B, but he also added a problem-solving document: "“*This document is a template that is followed to perform the RCA and to fix it.*”If the car part came from a supplier, this document would be provided externally with cooperation and supervision by Volvo Cars' QE.

Interviewee D mentioned the same internal programs as C and B and explained that cases had different prioritization depending on the severity of the case. Some components had the potential to be of higher severity due to security reasons, making them more important to look at than others.

Remarkably, all interviewees emphasized that the knowledge and experience about the DTCs were a big factor in the problem identification process. They relied on their personal experience to determine what was critical and what was not.

Most of the interviewees mentioned the trend of DTCs. Currently, there was an internal system to visualize and alert the occurrence of the DTC data coming from the workshops in each market. The engineers from the *Quality & Launch Team* monitored and took action based on the increasing or decreasing trend. Interviewee A argued that the frequency of the triggered DTC was important. The situation might become serious and impact the customers if an obvious increase showed up in the diagram.

5.1.2 Thematic Analysis

We employed thematic analysis to summarize the interview. Figure 5.1 displays the result of thematic analysis phase 2, an initial thematic map. There are four themes extracted from these interviews:

- The Problem identification process can be time-consuming, right now, it

is a manual process that needs cooperation between internal and external departments, including customers, logistics, workshops, suppliers, and R&D.

- The Problem identification process is case-sensitive, the situation may vary from each other depending on vehicle types, symptoms, manufacturing plants, target markets, and other environment settings.
- However, the problem identification process is partly summarized based on empirical knowledge, the internal knowledge base, and a standard procedure.
- Problem identification is a helpful process because it solves customers' issues and mitigates shortcomings for future development.

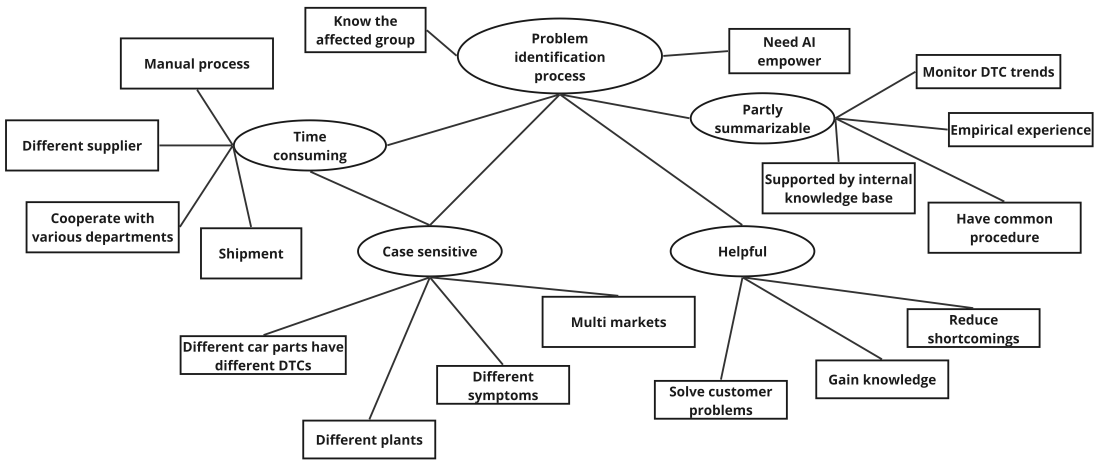

Figure 5.1: Thematic Analysis Phase 2

5.1.3 Problem Identification Process

Concluding from the interview, we can determine the problem-identifying process, figure 5.2 visualizes the whole process. It starts when the vehicle goes to the workshop. It can be the vehicle has some problem or just performs its periodic inspection. After this, for the problem car, a vehicle report with related DTCs and other useful information will be downloaded, documented, and sent to the Quality Engineer(QE). With the help of other internal systems, including the vehicle information system, DTC information system, and DTC trend visualization platform, this person will analyze the list of DTCs and determine whether they are the familiar ones that link to some well-explained root causes or they are the unfamiliar ones that require more investment. The decision is made based on QE's empirical knowledge and experience. QE also analyzes certain circumstances to evaluate and prioritize the case. For example, the car symptoms, DTC type, affected markets or countries, and amounts. It is important to look at the amount of the triggered DTC in all existing vehicles to inspect whether the failure rate is going up or down. After these, QE needs to identify if the problem is caused by the supplier or Volvo Cars. The result of this question influences the entity that conducts the problem identification. A part of the problem-identifying process is reproducing the issue. If it is reproducible then it

simplifies finding the solution for the problem. If not, a collaboration between Volvo Cars and the suppliers might be needed to stress test the units. There are multiple important internal programs used to proceed with the problem identification process. The software helps with finding the report, compiling it into metadata to see the logs of the DTC, and finding information about the symptoms and DTCs. It is hard to estimate the needed time for the problem-identification process since each case is different and the resolution depends highly on the knowledge of the responsible engineer.

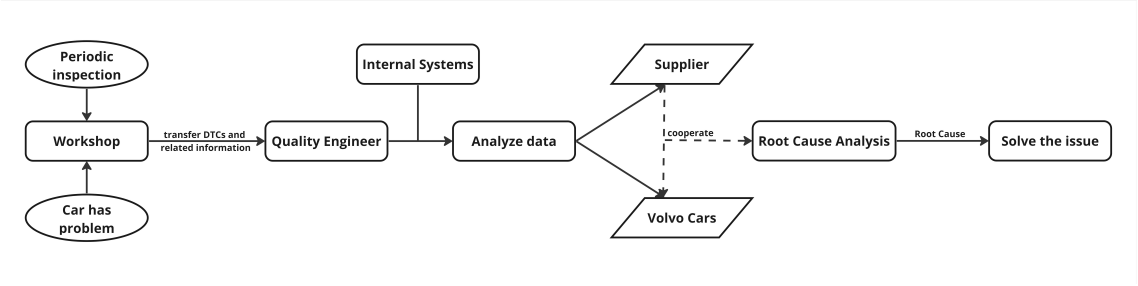

Figure 5.2: Problem Identification Process

5.1.4 Limitations & Challenges

Based on discussions with industrial experts and our insights, we identify some limitations and challenges in the current problem-identification process. In this section, we break down the entire process into five subsidiary processes and address the limitations and challenges encountered for each process separately. These sub-processes include data collection, data transformation, problem pre-investigation, root cause analysis, and problem closure.

Figure 5.3: Data Collection in Problem Identification Process

Data Collection This is the first step of the whole problem identification process, figure 5.3. The whole troubleshooting begins when the vehicle goes to the workshop. The customers can go to the workshop to perform periodic inspections. However, in most cases, they go there when their vehicles encounter some problems during the driving circle. In this way, all the actions performed subsequently by Volvo Cars are reactive. Customer experience might have already been reduced. Other than this, interviewee B mentioned "*Technicians at the workshop will first erase all the data stored in the ECU to see if the problem is reproducible*". Additionally, technicians will perform a first filtration to select the related DTCs based on their previous experience. All of these reduce the completeness of the data, and some useful ones

might get deleted which makes it more difficult when trace the root cause in the later process.

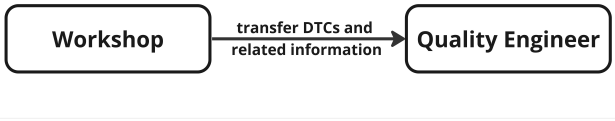

Figure 5.4: Data Transformation in Problem Identification Process

Data Transformation Figure 5.4 visualizes this process. After collecting the data from the workshop, they will be extracted, transferred, and loaded into Volvo Cars' internal database. For the problem vehicles, a vehicle report might also be sent along. According to the interviewees, this process takes time. There is a latency between the data collection date and the data receive date. It requires extra time for customers to wait. In addition, some data may be missing during this process due to unknown technique issues. It might also affect the completeness of the data.'

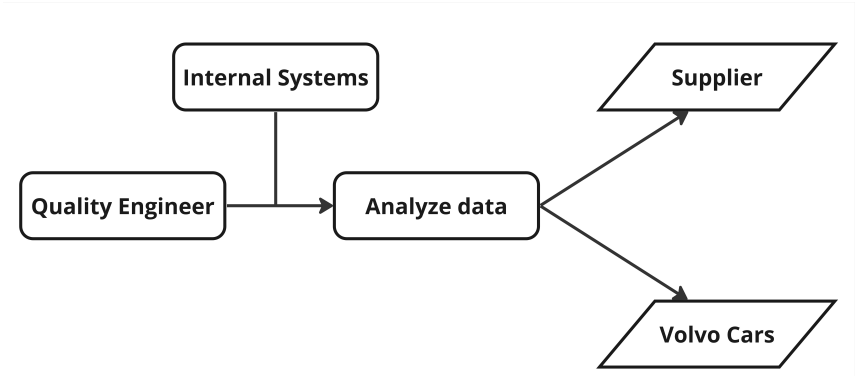

Figure 5.5: Problem Pre-investigation in Problem Identification Process

Problem Pre-investigation Figure 5.5 presents the problem pre-investigation process. Based on different car parts, different Quality Engineers will be assigned to pre-investigate the problem. Instead of figuring out the actual root cause, at this stage, QE will focus more on identifying if this is a familiar issue that happened before with a well-explained root cause, or if it is a new problem that requires more investment. QE will also determine whether the supplier or Volvo Cars should take more actions, or if collaborative efforts are needed to identify the root cause. Some internal systems facilitate this process. These four are the most commonly used: a system to visualize the DTC trends in various adjustable situations, an application that stores all the information about the problem cars, an application that introduces the DTCs and the possible connected vehicle symptoms, and a knowledge base that stores all the history of previous tickets. However, the process still heavily relies on the expertise of QEs, who make significant decisions based on their experience. The lack of knowledge may result in longer response times and incorrect problem-solving

directions, leading to extended waiting times for customers and a continuous decrease in customer experience. This, in turn, may contribute to higher warranty costs and a diminished reputation for Volvo Cars.

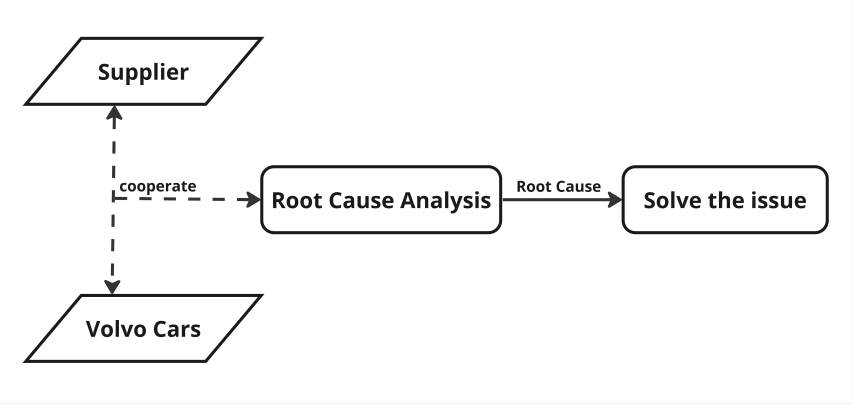

Figure 5.6: Root Cause Analysis in Problem Identification Process

Root Cause Analysis Figure 5.6 briefly describes the RCA process. Normally, this process involves participants from various internal and external departments. The communication and the knowledge gap between them further prolong the problem-identification process. Furthermore, in some cases, there will be an inspection performed on the problem car part. Based on the market where the problem car is located, the destination for inspection, and the shipping condition, the shipment may vary from a few days to several weeks, it is an unpredictable factor that can not be conquered. Additionally, all the interviewees agreed this process is highly case-sensitive and it is difficult to provide an estimated time. From their experiences, they have participated in the RCAs that lasted from one month to half a year. Sometimes the root cause is obvious but sometimes it is not. Two similar cases with nuances may also end up with totally different reasons. Aligning with the previous process, problem pre-investigation, RCA is also an expert-driven process. The professionalism of the engineers highly affects the overall resolving time.

Problem Closure The consummation of the RCA does not signify the end of the problem identification process. A well-documented transcript is needed. This can be beneficial when Volvo Cars encounters the same problem next time. More time can be saved. Also, it helps to refine the problem identification process by summarizing the advantages and disadvantages. It also facilitates newbies to gain as much experience as the experts. In the current process, a feedback activity is performed to avoid a similar issue in the upcoming new design. It is good enough for the current stage, but there still is a knowledge gap between teams and people may also cause bias, for example, the misunderstanding of the acronyms.

Therefore, we can summarize the limitations and challenges involved in these five sub-processes:

- The current problem identification process is reactive; it normally starts when a customer encounters some problems.

- During the data collection and data transformation process, the raw data may be incomplete, which may affect subsequent actions.
- The whole process is heavily expert-driven; experts' knowledge and experience may significantly influence the decision.
- Each case has its particularities, making it hard to estimate the required time.
- The historical issues are not standardized and documented, which may lead to further knowledge gaps or biased understanding.

After discussing and reviewing the themes from the thematic analysis and aiming to overcome some of the mentioned limitations and challenges, we conclude with the single practical theme for further laboratory experimentation. It involves exploring several algorithms, including ML algorithms, to construct a model that determines the most affected group based on DTCs and other attributes. Identifying the most affected group forward enables QEs to make further decisions on whether they will focus on other more important cases, monitor the car performance, or collaborate with other departments to proactively resolve the potential problem before it impacts customers at scale.

5.2 Laboratory Experiment

Throughout the first-round case study, we were presented with examples and cases. After discussions with industrial experts and supervisors, we chose five of them to perform our laboratory experiment. We gathered related data from an internal database. To align with Volvo Cars and keep data confidential, we anonymized several specific types of data. In total, we constructed four models based on these cases. In this section, we described each case and model construction in detail, presented and compared the models' results, and discussed the techniques that could help resolve some of the limitations and challenges encountered in the current problem identification process.

5.2.1 Reported Cases

There were five cases in total: two of them were about vehicle climatization systems (Case 1 & Case 5), two of them were related to car part A (Case 2 & Case 3), and one was associated with an add-on car part (Case 4). These incidents occurred in recent years. A more detailed description of each case is shown below:

Case 1 - In-car atmosphere changes intermittently
The customer came from one of the markets and stated that the climatization system functioned intermittently. According to their descriptions, after driving for a few minutes, the climatization system did not perform as expected.

Case 2 - Reduced functionality in car part A
The customer stated the car part A's functionality was not working as expected.

According to their descriptions, this problem happened right after they ignited the engine, a failure message was shown on the screen in front of the driver, Driver Information Module (DIM).

Case 3 - Car part A temporarily unavailable
The customer stated a system related to specifically car part A was not working as expected. After the driver performed a complete stop, this system's functionality was temporarily unavailable along with a DIM message stating the error.

Case 4 - Add-on system unavailable
The customer stated that an add-on mode was unresponsive. According to their descriptions, the add-on was not functioning, and an error message stating "Please contact the after-sales service for the add-on mode "appeared on the DIM.

Case 5 - Unexpected behavior in the climatization system
The report came from a specific market, and it stated that the climatization wasn't working as expected. During this period, the specific market experienced extreme heat scenarios, the temperatures were high. According to their descriptions, when facing stop-and-go traffic with low airflow, the climatization system would lose performance. It would resume if the car accelerated and the airflow increased.

We divided these five cases into two parts, development cases, and evaluation cases, and distributed the related car parts evenly. The grouping is as follows:

- Development Cases: Case 1, Case 3, and Case 4. These include the car part A unit, climatization system, and an add-on system.
- Evaluation Cases: Case 2 and Case 5. These include the car part A unit and climatization system.

Case No.	Model Year	Vehicle Type	Energy Source	Platform	Assembly Plant	Assembly Week	SWPN
1	3-5	10-15	3-5	<3	5-10	100-200	50-70
2	6-10	10-15	3-5	<3	5-10	100-200	50-70
3	6-10	10-15	3-5	<3	5-10	200-300	50-70
4	6-10	10-15	6-8	<3	5-10	> 400	10-30
5	6-10	10-15	6-8	<3	5-10	200-300	50-70

Table 5.1: Laboratory Experiment Data Summary

Table 5.1 provides a brief data summary for each case to enhance data clarity. The number in the table represents the range of types that are involved in the case. For example, Case 1 includes 10 to 15 types of vehicles that involve 3 to 5 Technical Model Years, 3 to 5 types of Energy Sources, and 2 to 4 types of Platforms. They have been assembled in 5 to 10 different Plants and constructed in 100 to 200 weeks. They share 50 to 70 different Software Part Numbers.

Besides, we also collect the experts' decisions for each case. This information was gathered from the internal ticket report system and inquired directly from the related engineers.

Below, we present the results from the laboratory experiments. Both Case 1 and Case 3 has four DTCs, and other cases have one DTC individually. Since we treat each DTC separately, a total of nine sets of experiments need to be conducted for each model. Four sets come from Case 1, four from Case 3, and one from Case 4. There are four different models: one baseline model, one foundational ML model using Linear Regression, and two advanced ML models, one utilizing K-means clustering individually and the other combining K-means clustering with Linear Regression. We will also discuss the advantages and disadvantages of each model and compare their performances with each other to select the final model construction methods.

5.2.2 Correctness Calculation

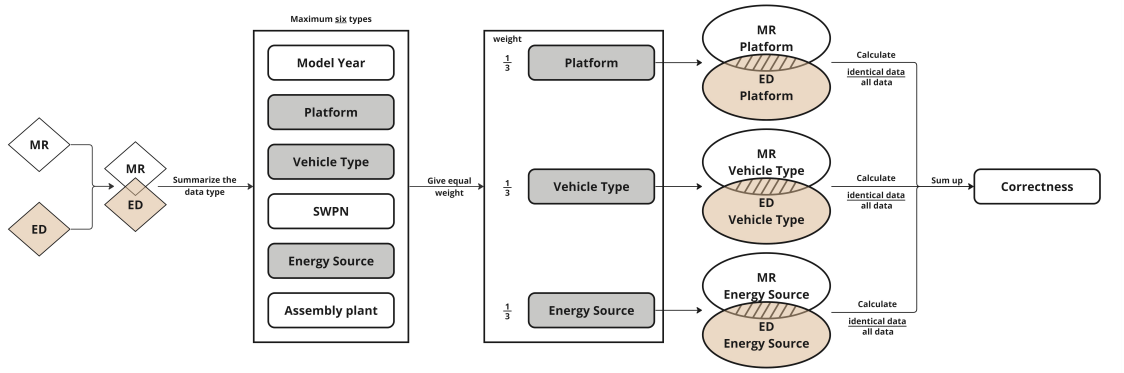

Figure 5.7: An example of correctness calculation

During the laboratory experiment, model construction, we defined a metric called **correctness**, this metric is used to measure the results of the models which allows us to compare the results in a numerical way and label them. It stood for the similarity between the results provided by the model and the experts' opinions. Figure 5.7 presents an example where the result of the model and the experts' decision includes three types of data: Platform, Vehicle Type, and Energy Source. A more specific calculation process is described as follows:

- 1. Collect the results given by the model (MR) in one calculation circle and the experts' decisions (ED).
- 2. Summarize the data types that appear in the collection of MR and ED. There are a **maximum six** types of data: Model Year, Vehicle Type, Energy Source, Platform, Assembly Plant, and SWPN.
- 3. Give equal weight to each covered data type.
- 4. For each data type, calculate the number of identical data in MR and ED as a percentage of all data.

- 5. Utilize this percentage multiplied by the weight to obtain the weighted percentage.
- 6. Sum the weighted percentages to calculate the final correctness.

5.2.3 Model Development - Baseline Model

The construction of the baseline model did not involve any ML techniques. We collected the data in a six-month period. The only factor in determining the most affected group was the percentage of triggered cars relative to the total number of cars of that specific type during a calculation cycle. This percentage is defined as trigger-total-percentage (TTP), and equation 5.1 describes how to calculate it. G means that this is a metric calculated for a group.

$$TTP(G) = \frac{\text{Triggered Vehicle Volume}}{\text{Total Vehicle Volume}} \tag{5.1}$$

Equation 4.1: Trigger-Total-Percentage (TTP) Calculation

One calculation cycle was five weeks, including the current week and the four previous weeks. All of the existing vehicles were grouped based on various car settings, including Model Year, Vehicle Type, Energy Source, Platform, Assembly Plant, Assembly Week, SWPN, and Market if this information was available. After this, we counted the TTP for each group. During a calculation cycle, the group that had the highest TTP value was determined as the most affected group for that cycle. In the end, we sorted the data in ascending order by date to present the most affected group with the car settings, the total DTC volume this group triggered, the total triggered vehicle amount, the total vehicle number in this setting, and the TTP value. Figure 5.9 presents the process for the first-round baseline model when identifying the most affected group in one calculation circle.

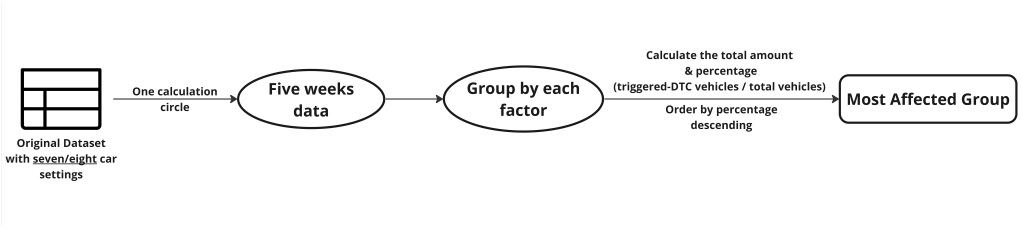

Figure 5.9: First-round Baseline Model Construction

However, the first-round result did not align with the actual situation and the experts' decision. The model ended up identifying some significantly different groups than the experts' decision. After observation, we found out that two car settings, Market and Assembly Week, divided the data very discretely which resulted in containing a very small amount of vehicles in one group. Another factor that could also have led to this result is if the Vehicle Type was too new or too old, a very small number of samples

could be collected. It further affected and reduced the interpretative ability of TTP. A high percentage did not efficiently point to the most affected group. For example, if we compared a group that had 50 triggered vehicles out of 100 with another group that only contained 1 vehicle and it triggered DTC. The former percentage was 50%, and the latter was 100%. But the first group should be emphasized more because more cars were affected. Due to this issue, we refined the baseline model by deleting Market and Assembly Week classification conditions and filtering out the group that had fewer than 100 vehicles. Figure 5.10 presents the second-round baseline model construction. Because we employed a 6-month range and used a 5-week calculation cycle for each case, the baseline model initially produced results by week 5 and continued to identify the most affected group through week 27 for all cases. However, it's important to note that some reported groups still displayed a relatively low TTP value. This suggested that no warning should be sent to Volvo Cars, and the engineer might not necessarily have needed to take immediate action, as there may not have been a significant problem.

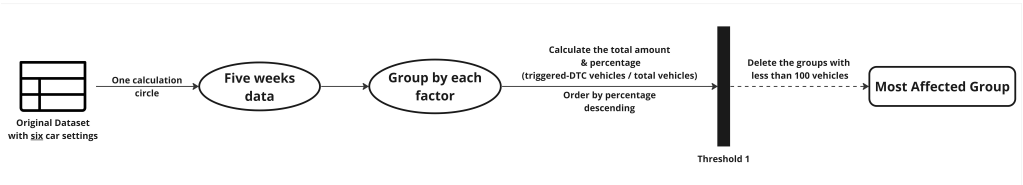

Figure 5.10: Second-round Baseline Model Construction

5.2.4 Model Development - Refined Baseline Model

To better refine the baseline model, we used another threshold of 1% TTP value to filter out the most affected group during a calculation circle. If the maximum TTP value for the current round was smaller than 1%, then no warning should have been raised. This action decreased the sensitivity of the warning system by reducing the possibility of detecting unnecessary problems. It also helped to specify the earliest warning date. 1% was a percentage defined and practiced in other departments at Volvo Cars. We placed this as a minimum level to classify if an issue happened or not. Figure 5.11 visualizes the refined baseline model construction process with this 1% TTP threshold.

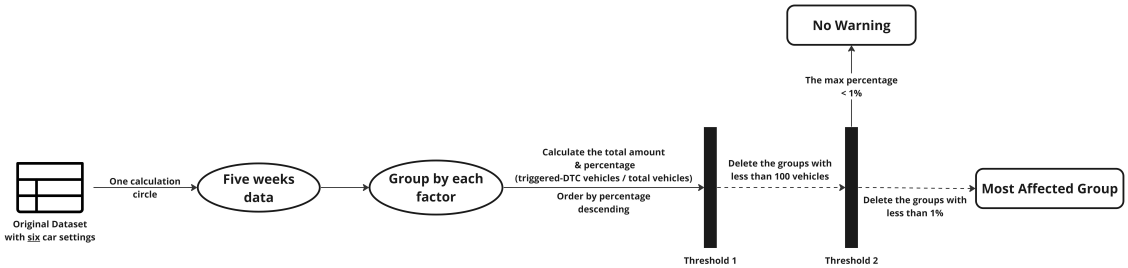

Figure 5.11: Refined Baseline Model Construction

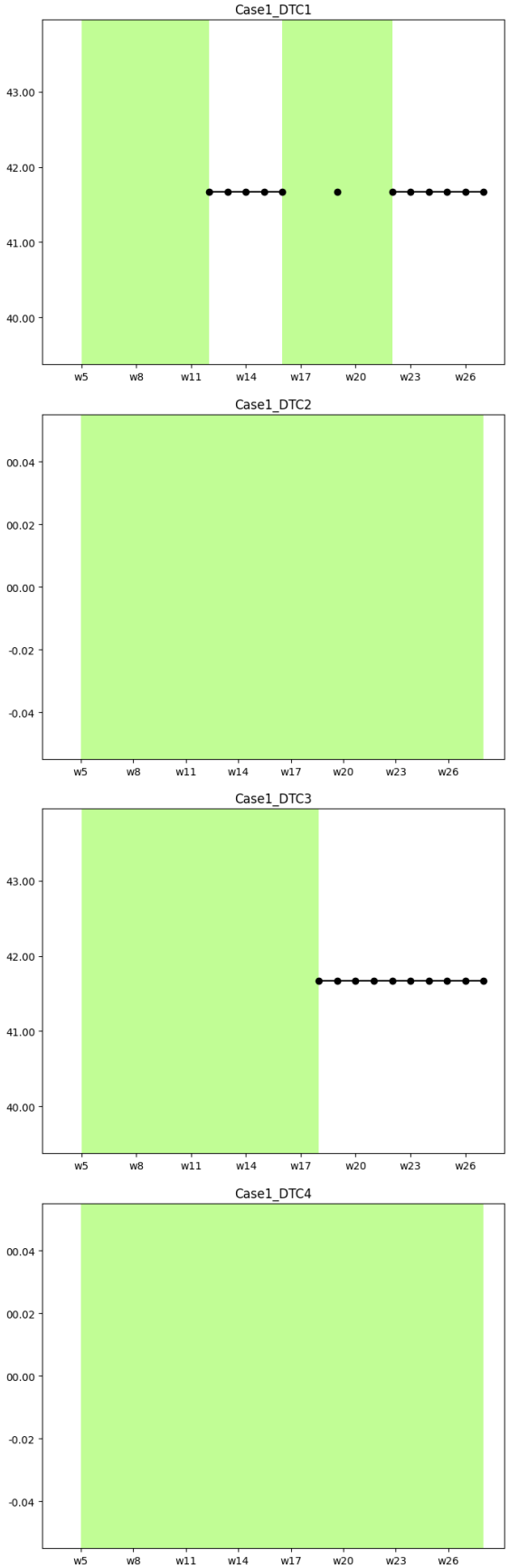

Figure 5.12: Baseline Refined Model Correctness (1)

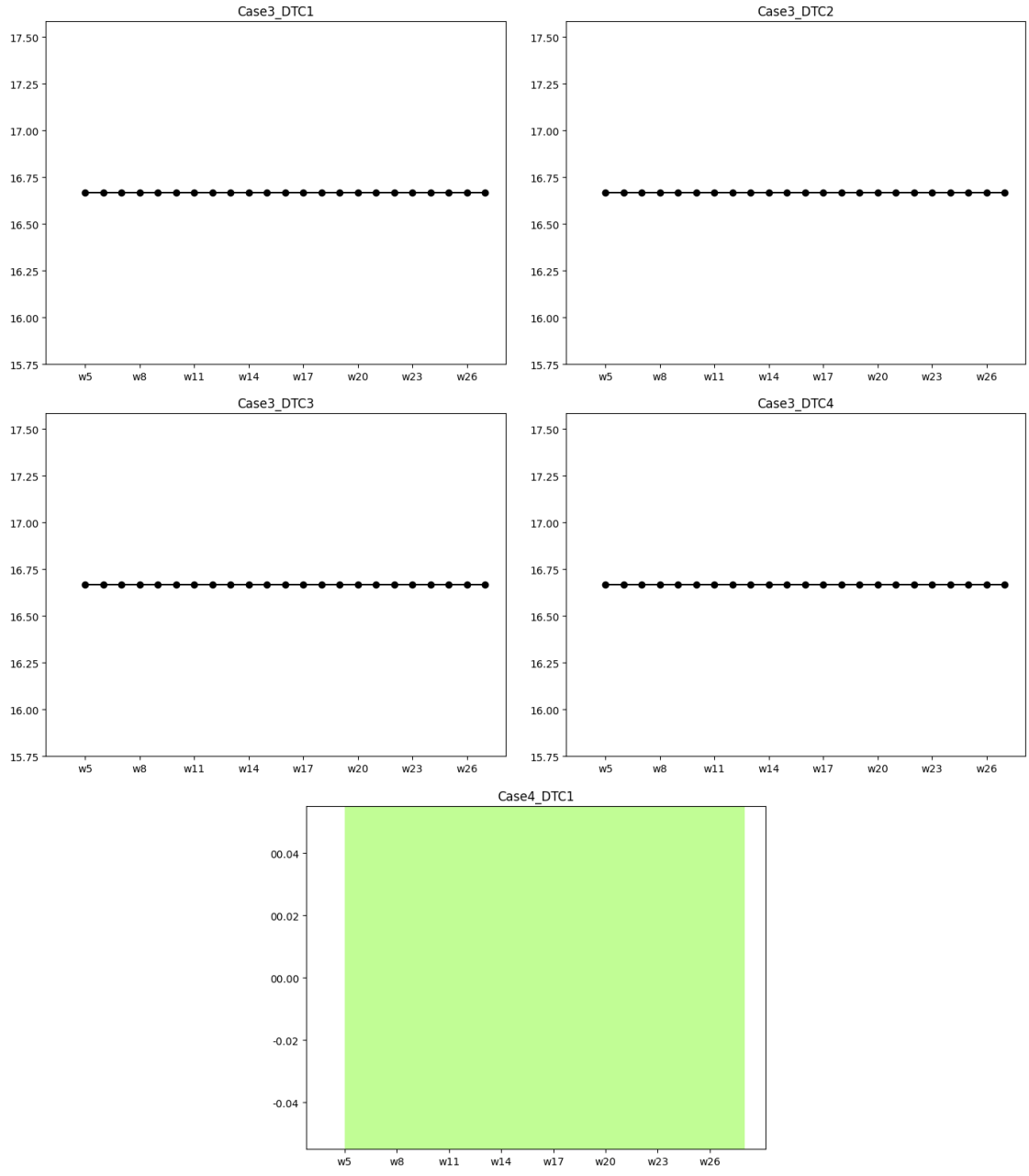

Figure 5.13: Baseline Refined Model Correctness (2)

We ran the baseline algorithm again and got the following correctness result. In Figure 5.12 & 5.13, the x-axis represents readout week, and the y-axis stands for correctness. If the background color is green, it means no warning has been detected during that week. The correctness was calculated based on the most affected group that the refined baseline model gave and the most affected group mentioned in the incident report ticket and interview.

It could be easily observed from the diagram that the predictions of the refined baseline model did not always align with the experts' views. For Case 1 DTC 1 and Case 1 DTC 3, correctness remained at 41.67% for several weeks. However, the baseline model did not detect any warning in Case 1 DTC 2 and Case 1 DTC 4. The most affected group for Case 1 was first discovered in week 11 by DTC 1. For Case 3, regardless of the DTC, the result provided by the refined baseline model was only 16.67% accurate. The model started to detect the most affected group from the first calculation cycle, week 5. However, the current baseline model gave a completely different result for Case 4. Because the TTP for this case consistently stayed below 1% each week, with a mean value of 0.40%, no most affected group was identified during this period.

The contained scope of the refined baseline model varied from the experts' choice. The refined baseline model always provided a smaller scope than the experts' decision. Table 5.2 provides a summary table comparing the types of classifications employed in the results of both models. The refined baseline model used a defined car setting of six different factors: Model Year, Vehicle Type, Energy Source, Platform, Assembly Plant, and SWPN. The result it gave always included these six types. However, the experts could use their knowledge to include or exclude some of the classification criteria based on the incident type, vehicle composition, and other experience-related factors. In their opinions, Case 1 included Vehicle Type and Energy Source; Case 2 included Vehicle Type, Electric, and Assembly Plant; Case 3 included Platform; Case 4 included Platform and SWPN; Case 5 included Vehicle Type.

Case No.	Refined Baseline Model	Experts' Decision
Case 1	6	3
Case 3	6	1
Case 4	6	2

Table 5.2: Car Settings Types in Refined Baseline Model vs. Experts' Decision

5.2.5 Model Development - Multiple Linear Regression

To better interpret the relationship between each factor and vehicle-triggered situations and to understand the necessity of utilizing advanced machine learning algorithms, we also employed another foundational algorithm, multiple linear regression, to construct a model. In accordance with the baseline algorithm, we took into consideration the same six factors: Model Year, Vehicle Type, Energy Source, Platform, Assembly Plant, and SWPN. We spanned a period of five weeks as well, which included the current week and the four weeks before it.

In one calculation circle, initially, we grouped the data set based on the six car settings and calculated data in three dimensions. The first dimension was the total number of vehicles in that specific group, the second was the total number of triggered DTC vehicles in the same group, and the last was TTP. Later, we applied a threshold to filter out groups with fewer than 100 vehicles included. After this, another threshold was employed to determine whether an error occurred. If TTP was less than 1%, no warning should have been reported for that calculation circle; otherwise, the algorithm processed to identify the most affected group using multiple linear regression, extracted the significant features, and outputted the group description.

Traditional linear regression is designed to work with numerical data. However, five car settings, including Vehicle Type, Energy Source, Platform, Assembly Plant, and SWPN, were recorded as textual descriptions. For Model Year, it was stored as an integer in the data set with certain limits and distributed discretely. To align with the design purpose of linear regression and improve its efficiency, we converted these six categorical data into numerical data using one-hot encoding [32]. It created binary columns for each category based on the original variable. These binary indicator variables were treated as the independent variables, while the dependent variable was the percentage of triggered vehicles divided by the total number of vehicles. The multiple linear regression model is represented by Equation 5.2, where MY stands for Model Year, ES stands for Energy Source, and SW stands for SWPN.

$$y = \beta_0 + \beta_{\mathrm{MY1}} x_{\mathrm{MY1}} + \beta_{\mathrm{MY2}} x_{\mathrm{MY2}} + \ldots + \beta_{\mathrm{ESn}} x_{\mathrm{ESn}} + \ldots + \beta_{\mathrm{SWn}} x_{\mathrm{SWn}} + \varepsilon \qquad (5.2)$$

Equation 4.2: Multiple Linear Regression

After constructing the multiple linear regression model, we needed to carefully choose the most affected features to formalize and describe the most affected groups. Six statistical values could be used to assess a feature's importance, including the estimated coefficient, standard error of the coefficient, t-statistic, P-value, and the 95% confidence interval. We employed the P-value and coefficient value as thresholds to identify the corresponding variables that had a statistically significant effect on the dependent variable. Table 5.3 presents a portion of the feature summary table, containing coefficient value and P-value, from one experiment. The P-value indicates the degree to which the data aligns with the null hypothesis, and 0.05 is a common threshold used in linear regression [33]. A positive coefficient reflects that this feature has a positive correlation with the probability of triggering DTC in the vehicle. Furthermore, a larger positive coefficient suggests a stronger positive relationship between the independent and dependent variables. For these reasons, we selected the features that had less than a 0.05 P-value and had a positive coefficient value. The algorithm outputted the top six features. If fewer than six features were left after the threshold filtering, then all features were output. Figure 5.15 visualizes the process for constructing the Multiple Linear Regression model.

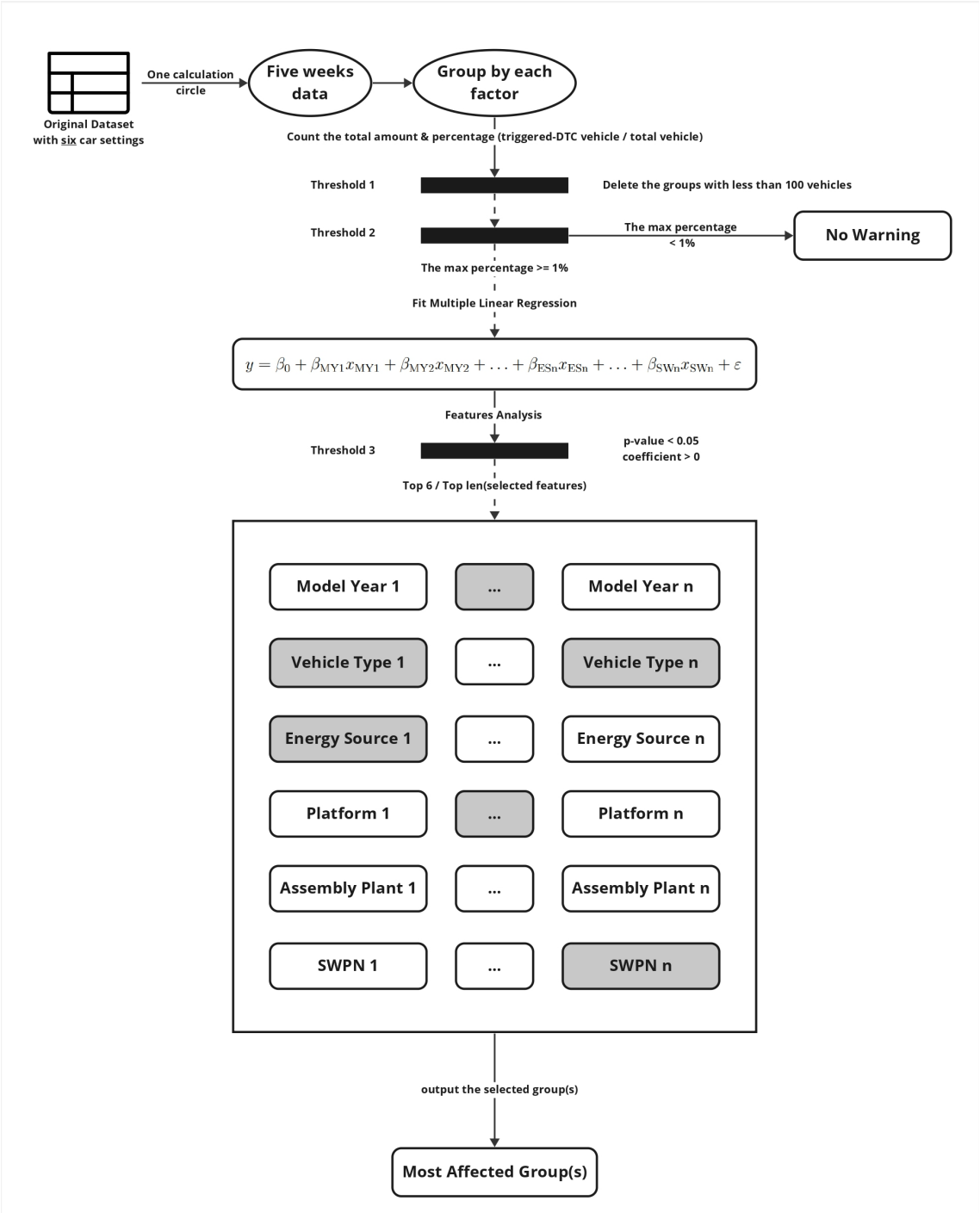

Figure 5.15: Multiple Linear Regression Model Construction

Feature Name	Coef	P>|t|
Vehicle Type 1	-0.5834	0.002
Vehicle Type 2	0.1022	0.428
...	...	...
Model Year 3	-0.7107	0.000
...	...	...
Energy Source 1	0.0378	0.538
Energy Source 2	0.1990	0.007
...	...	...
Platform 1	0.4273	0.000
...	...	...
Assembly Plant 1	0.4810	0.011
Assembly Plant 2	0.0801	0.728
Assembly Plant 3	0.1603	0.295
...	...	...
SWPN 7	0.0123	0.903
SWPN 8	0.1531	0.283
...	...	...

Table 5.3: A portion of the feature summary table for Multiple Linear Regression

5.2.6 Model Development - K-means

To explore the feasibility and performance of unsupervised ML algorithms in this context, we first integrated K-means clustering into the model construction. To align with the baseline model, we considered six factors: Model Year, Vehicle Type, Energy Source, Platform, Assembly Plant, and SWPN.

As mentioned before, the data set we provided included several categorical data such as Vehicle Type, Energy Source, Platform, Assembly Plant, and SWPN. Since the K-means algorithm also functions well with numeric values, we needed to convert the categorical data into numerical data to enhance its performance. Therefore, we utilized categorical feature encoding [34] [35]. Additionally, we implemented the Standard Score Normalization (Z-Score Normalization) technique to enhance clustering efficiency. Furthermore, we employed the Elbow method to dynamically choose and optimize the number of clusters [36] [20].

However, K-means clustering, like other clustering techniques, relies on a random component [37]. The initial centroid value is randomly determined in K-means clustering, leading to different outcomes with each run [38]. Fränti and Sieranoja suggested that correctness can be improved by repeating the algorithm several times [39]. Following this theory to reduce running time, we examined and set the number of repetitions to 25.

To maintain the same independent variable, we defined a calculation circle as five weeks, including the current week and the previous four weeks. After the algorithm separated all the vehicles into several groups in one iteration, we considered each individually. Two features were calculated and employed when deciding the most

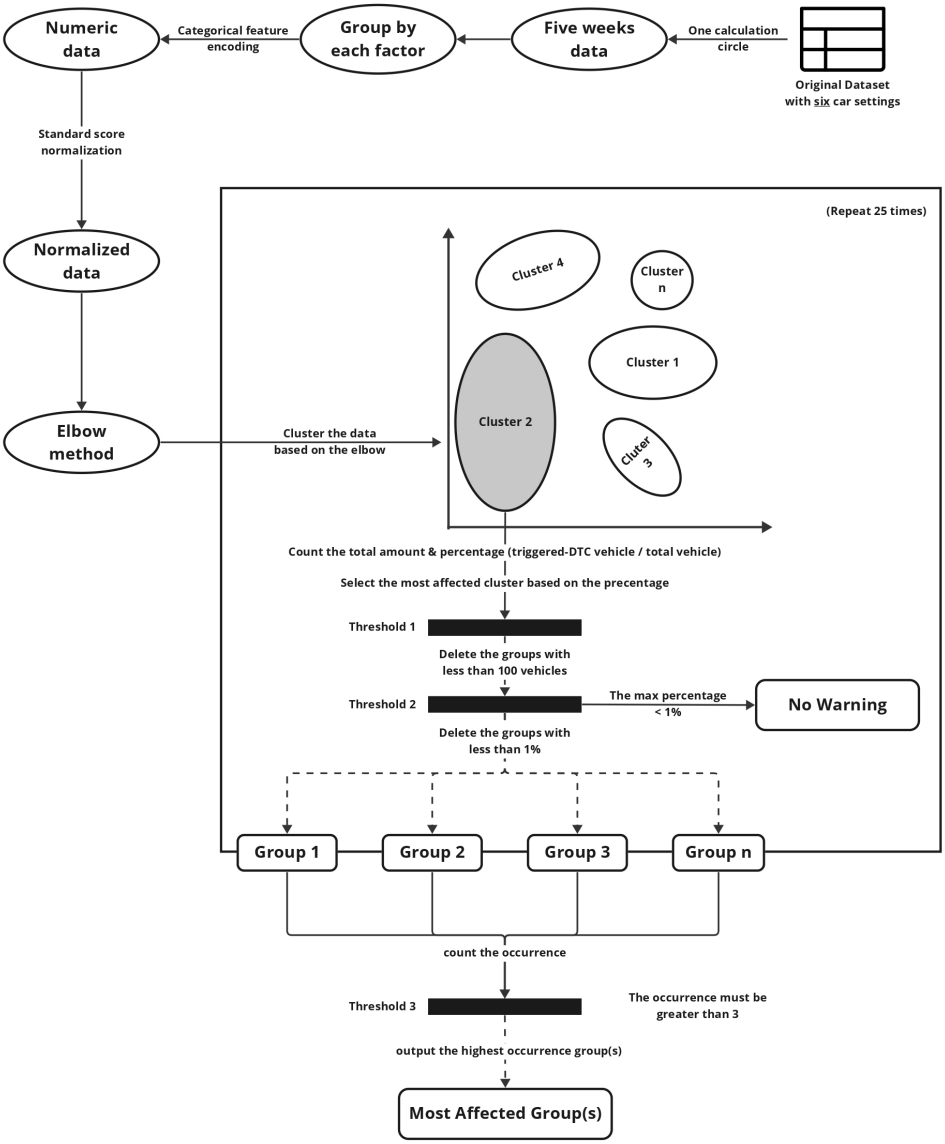

Figure 5.16: K-means Model construction with Elbow methods

affected cluster: the total number of vehicles in one cluster and the portion of triggered-DTC vehicles. To narrow down unnecessary groups, we applied two identical thresholds, consistent with those used in the baseline model. Clusters with fewer than 100 vehicles and TTP less than 1% were not treated as the most affected cluster due to their low severity. To specify this cluster concretely, we obtained six defined car settings for each vehicle and calculated the occurrences of each combination. The combination that appeared the most times was considered the most affected group. As mentioned earlier, one iteration was repeated 25 times, and we defined a threshold of a minimum of 3 occurrences to reduce the randomness caused by K-means clustering. We believed this approach could improve the robustness of the algorithm. It could have more than one most affected group in one circulation circle if these groups' occurrences were the same and the highest. Figure 5.16 specifies the process for developing this model.

5.2.7 Model Development - K-means & Linear regression

The K-means model presented better performance when predicting the most affected group based on the triggered DTC situation. However, the wideness of the result was still smaller than the experts' decision for every case. In K-means model construction, because we didn't perform feature selection, the result it provided always included six car settings: Model Year, Vehicle Type, Energy Source, Platform, Assembly Plant, and SWPN. To reduce the unnecessary filtering condition and further improve the prediction accuracy, we employed a feature selection based on the P-value given by the Linear Regression algorithm after the K-means clustering.

The first half of the process was similar to the K-means Model construction process. Initially, we gathered five weeks of data as a calculation set and grouped it by six factors. This resulting dataset was referred to as DS1. Subsequently, feature encoding was applied to DS1 to transform the characteristic data into numeric data. Following this, we employed the Standard Score Normalization technique to normalize the data, enhancing clustering efficiency for subsequent processes. Afterward, we determined the number of clusters based on the elbow method and performed K-means clustering.

For each cluster, we computed the percentage of the overall car volume relative to the number of vehicles that triggered DTC. The cluster with the highest percentage was selected as the most affected cluster. Therefore, we collected the group data included in this cluster. Two thresholds were applied to these groups to determine whether a warning should be issued and to identify which groups were considered the most affected within this cluster.

The first threshold was the total vehicle volume of the selected group. We excluded groups with fewer than 100 vehicles. The second threshold was TTP. We removed groups with less than 1% TTP. If no group remained after these two filters, no warning was issued. Otherwise, the retained group was considered the most affected group for that round.

To mitigate the randomness of K-means clustering, we ran it 25 times. The group that appears more than three times in these 25 rounds is treated as the most affected group

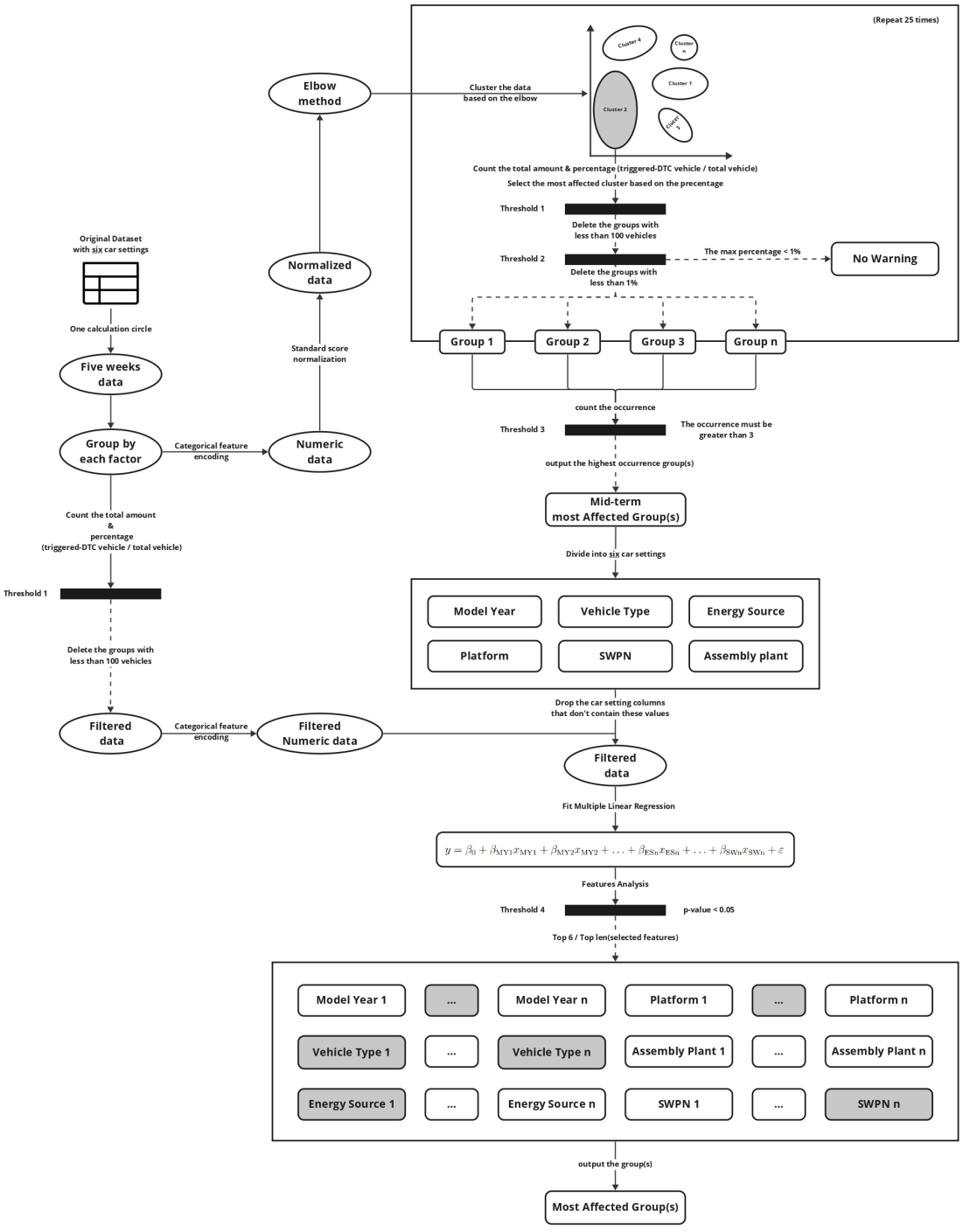

Figure 5.17: K-means & Linear Regression Model

for that calculation round. At this point, the most affected group still encompasses all six types of car setting mentioned above. After than this, we summarize and classify all the different car settings, for example, Model Year 1, Model Year 2, Platform 1, Platform 2, and so on to a data set called DS2.

The second half of the process is aligned to the construction of a Multiple Linear Regression Model. Initially, we calculate the total number of vehicles and TTP for each group in DS1, applying a threshold to exclude groups with fewer than 100 vehicles. Secondly, we employ the same feature encoding method on this filtered data to convert it from characteristic data to numerical data, resulting in another refined numerical dataset, DS3. Since our objective is to conduct feature selection based on the outcome of the K-means process, only the features of DS3 are presented in DS2 are remained, while others are eliminated. We construct a linear regression model using these remaining feature data and the corresponding TTP values. After having the P-value for each feature, we select the feature whose P-value is smaller than 0.05. These formed the final most affected group for the current calculation circle. Figure 5.17 visualizes this model construction process.

5.2.8 Model Comparison

We also followed the above grouping result during the model comparison process. Consequently, we divided this section into two parts. The model comparison for development cases involved assessing the results from Case 1, Case 3, and Case 4. For evaluation cases, we compared the outcomes of Case 2 and Case 5.

The results were presented based on the correctness of each case and each model, comparing their performance and analyzing the reasons for differences. In the correctness comparison section, we anonymized all time-related information for confidentiality reasons. Additionally, for better comparability, we used the same week range that spanned from week 5 (w5) to week 27 (w27) to align the time series for all cases. W27 marked the endpoint for data collection. Then all cases were documented in the internal ticket system. Because we used a five-week calculation cycle, the model started providing results from w5. Besides, The empty space in the figures indicated no warning at all that week, 0% correctness indicated that there was a warning but it did not align with the experts' decision. Here, we explain the abbreviations utilized in the legend:

- **M1**: Refined Baseline Model (blue line with dots)
- **M2**: Multiple Linear Regression Model (green line with triangles)
- **M3**: K-means Model (red line with stars)
- **M4**: K-means & Linear regression Model (purple line with squares)

5.2.8.1 Development Cases

As explained before, experiments were run based on individual DTCs. In total, nine experiments were carried out in the model development. A more detailed description for each case follows.

Case 1 This case describes a car issue related to the Climatization System. Based on vehicle reports, expert opinions, and internal knowledge base, four DTCs can be used as indicators of this problem. For each DTC, four models have been generated individually.

Figure 5.18 presents the result for the first DTC.

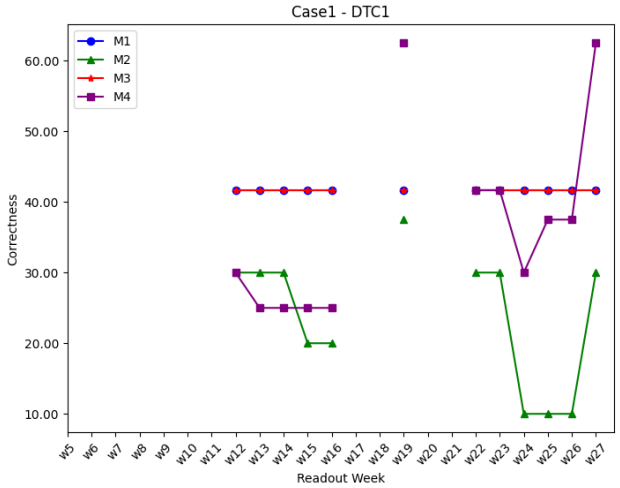

Figure 5.18: Case 1 DTC 1 - Correctness Comparison

As it shows, all models can detect and warn at week 12 with different correctness. M1 and M3 outshine the rest of the models at this early stage, however, at W19 we see a huge spike for case 4, which reaches around 61% at around 2 months early and beats the rest of the models which are around 35-40%. There are gaps in the weeks where no models can detect and make warnings at all, M1 and M3 are quite stable and are at a constant 41% correctness when they can notify and warn, M2 and M4 however are not as stable, they vary a lot. M2 for example ranges from 20% upwards to 30%, while M4 ranges from 25% to 61% so even though it detects the problem quite early and with a high correctness, it is not as stable as M1 and M3.

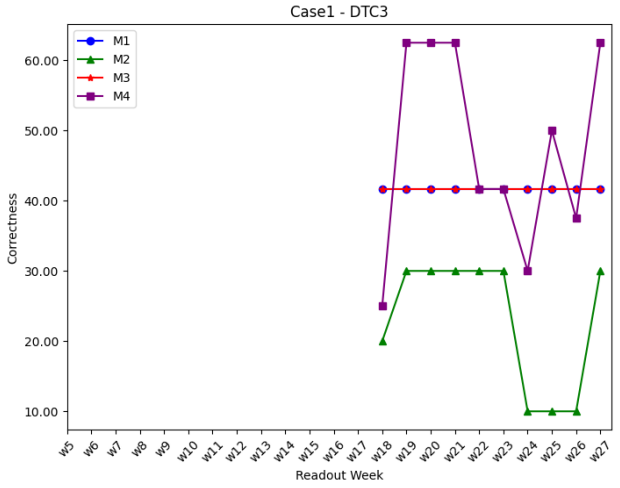

Figure 5.19: Case 1 DTC 3 - Correctness Comparison

For DTC3, All 4 models are able to detect a problem around 2 months early as seen in figure 5.19. Once again we have the stable models M1 & M3 at around 40% correctness. M3 shows worse results and is only able to reach 30% correctness as the maximum and 10% one month before the case was reported. M4 is not as stable but can produce great results, it can warn 2 months early with high correctness, and it reaches 62% which is quite high. It beats M1 which is the baseline model 5/10 times and is equal to it 2/10 times. Indicating great results from M4 in this case.

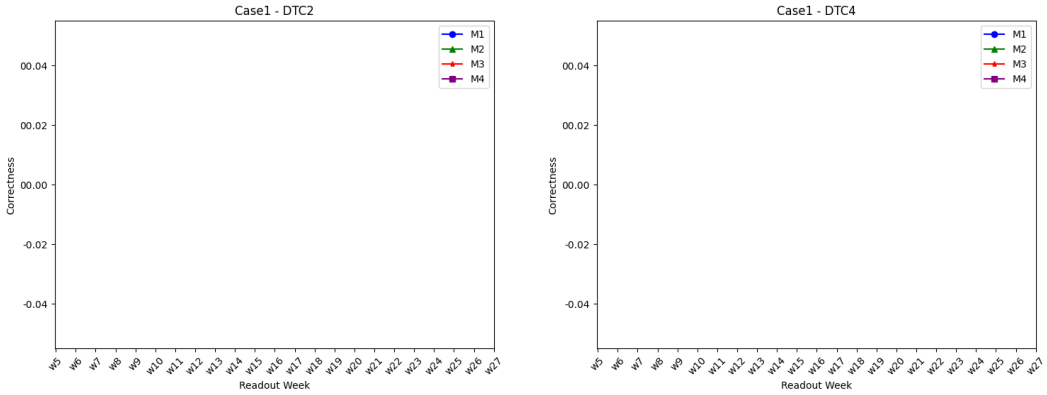

Figure 5.20: Case 1 (DTC 2 & DTC 4) - Correctness Comparison

In figure 5.20, both left and right are empty, this indicates that no model was able to make a warning in either DTC 2 or DTC 4 for Case 1. The models were not able to identify any affected group at all. The reason is, that for these two DTCs, the triggered situations are not as severe as DTC1 and DTC3. The data quantity is not enough to go beyond the threshold. "No Warning" has been shown during this whole period.

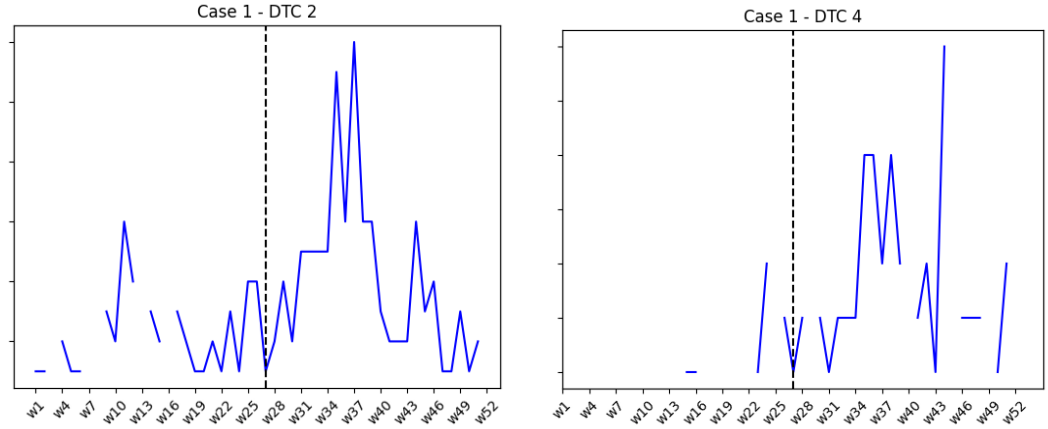

Figure 5.21: Case 1 (DTC 2 & DTC 4) - DTC reported date

Figure 5.21 shows the DTC-triggered situation. The dotted line is the reported date of Case 1, the x-axis stands for the experiment period, and the y-axis represents the DTC triggered amount. As seen in the figure, before the reported date, these DTCs

have not been activated frequently. Especially DTC 4, it was barely triggered before W27 and an upward trend appeared after that week.

Case 3 This case is related to the abnormal actions performed by the Car Part A Unit. Based on the initial round of case study, experts suggest four DTCs that are related to this issue and can be employed as predictable signs of the problem. However, each DTC provides unique results. To comprehensively describe each situation and analyze the reasons, we present the outcomes individually based on the corresponding DTCs.

Figure 5.22 shows the correctness result of Case 3 - DTC 1.

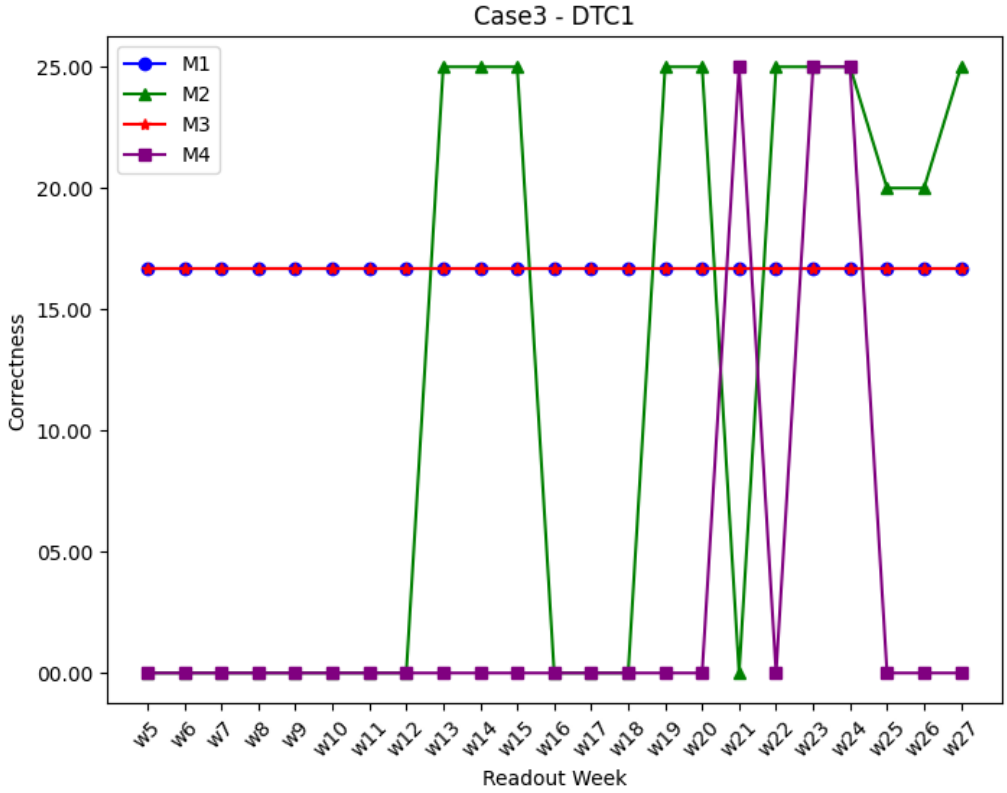

Figure 5.22: Case 3 DTC 1 - Correctness Comparison

As seen in the figure, all models can make a warning several weeks ahead of the reported time. Some with higher correctness than others. For both M1 and M3, we can see that they can predict the features with a 16% correctness 3-4 months ahead of the reported date. However, M2 and M4 can reach 25% correctness. M2 can reach 25% correctness 2-3 months ahead of the reported date of the case which is the highest out of all models and the earliest with that high correctness for this case. M4 reaches 25% correctness one month before the reported case, otherwise, it is not able to properly select the features of the reported case that the experts identified, leaving it at 0% correctness most weeks. It does however reach 25% which is the highest together with M2 for this case, the difference being that M2 reaches that level 1-2 months earlier.

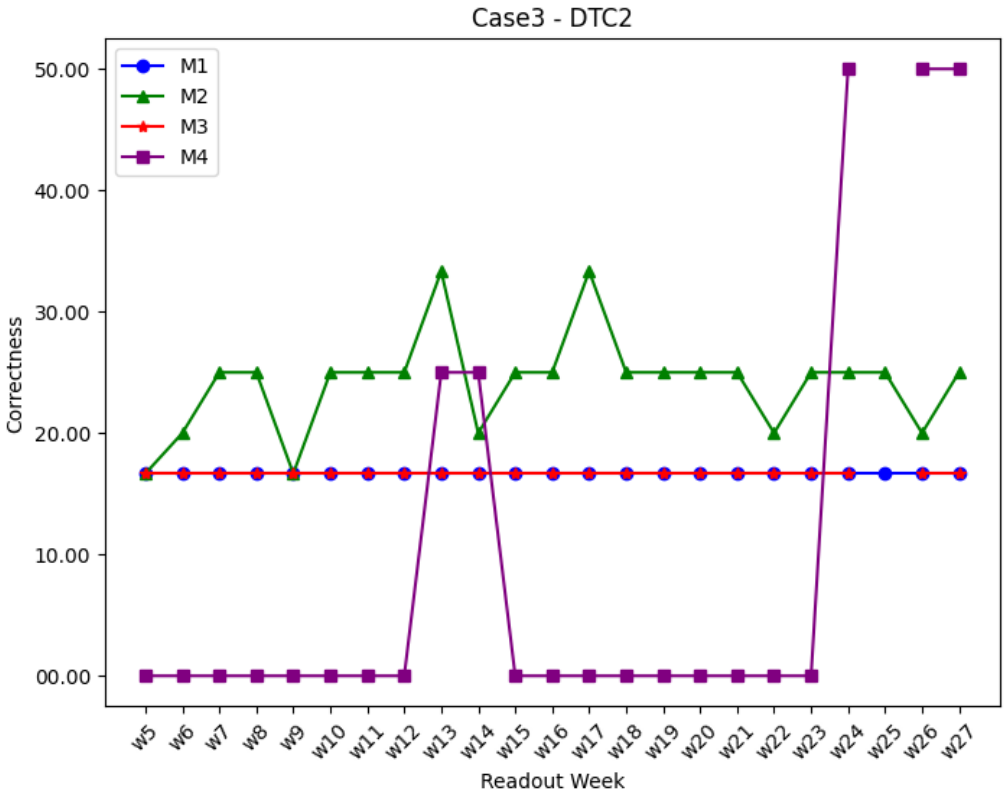

Figure 5.23: Case 3 DTC 2 - Correctness Comparison

Figure 5.23 has some interesting results. For DTC 2, All the models can find features most weeks. M1 and M3 have a constant correctness of about 16% except in week 24 where M3 is not able to select features. The M2 model however outperforms the M1 and M3, has a higher correctness, and is between 16% to 32%. It can predict as early as the baseline but outperforms it a big majority of the weeks. It performs the best in this case except for the last month where M4 can achieve a 50% correctness. M4 achieves the highest correctness one month before the reported date, other than that it isn't able to properly select the features that the experts can do. This case is a great example of whether high correctness but late warning is better than lower correctness but early warning or not.

Figure 5.24 presents the correctness comparison for the third DTC. From this figure, we can observe that M1 is very consistent, it has a constant correctness percentage of 16% throughout the weeks. This consistency indicates that the baseline model is stable. M2 however varies between 25-35%, it indicates a higher correctness but it is not as stable as M1. M3 as M1 has a constant correctness of 16% which implies stability but the correctness is not as high as M2. M4 is the model that fluctuates the most, it has varied correctness ranging from 0 to 34%. This could indicate adaptability but instability of some sort. From these models, we can conclude that for Case 3 DTC3, M2 has the best correctness as it has the highest and most stable result. From Figure 5.24, we can easily tell that M2 which is the multiple linear regression model outperforms the other models in this case. It constantly has the highest correctness out of all the models and unlike M4, the model can partially

select correct features every week.

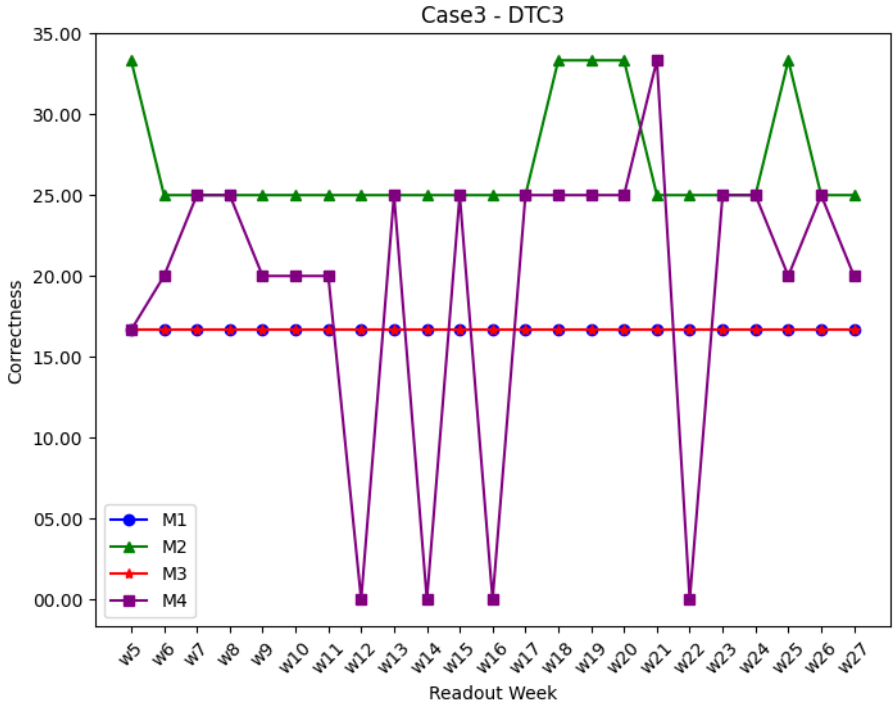

Figure 5.24: Case 3 DTC 3 - Correctness Comparison

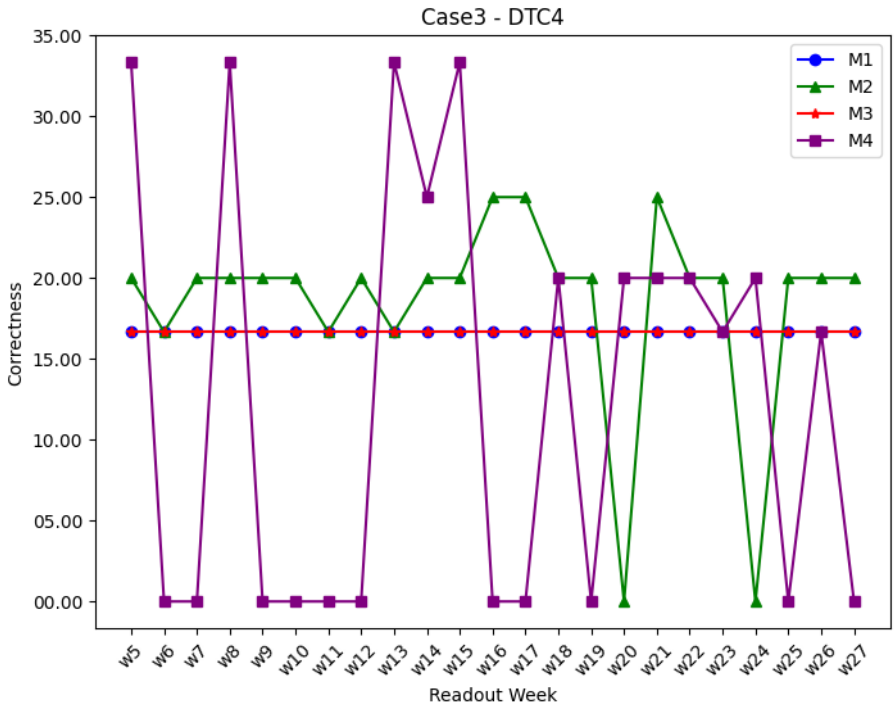

Figure 5.25: Case 3 DTC 4 - Correctness Comparison

As illustrated in Figure 5.25, the correctness comparison for DTC 4, M1 is once

again a constant correctness of 16%, the baseline model is stable and reliable in its results. M2 shows some variability in the correctness where it fluctuates ranging from 0 - 25%, however, most of the weeks it provides higher or the same correctness as M1. M3 is also a constant 16& which is a reliable model. M4 is the model that achieves the highest correctness in this case in a week, however, it also reaches the lowest frequently, the model seems to be unstable due to the correctness fluctuating so much weekly.

The reason for M4 and M2 having correctness some weeks down at 0% is because the features that the models selected those weeks are not correct at all. This means that the models selected features that the experts did not consider problematic in the reports thus resulting in 0% correctness.

M2 seems to provide the best results which is the multiple linear regression model, it has 2 weeks where the model fails to select any correct feature but most of the weeks, it can select features above the baseline model. M4 however, can reach a high correctness % based on these models quite early but it is not stable at all as it varies a lot throughout the weeks.

Case 4 This case is reported due an add-on system behaves differently than expected. According to the experts, one DTC can indicate the occurrence of this issue.

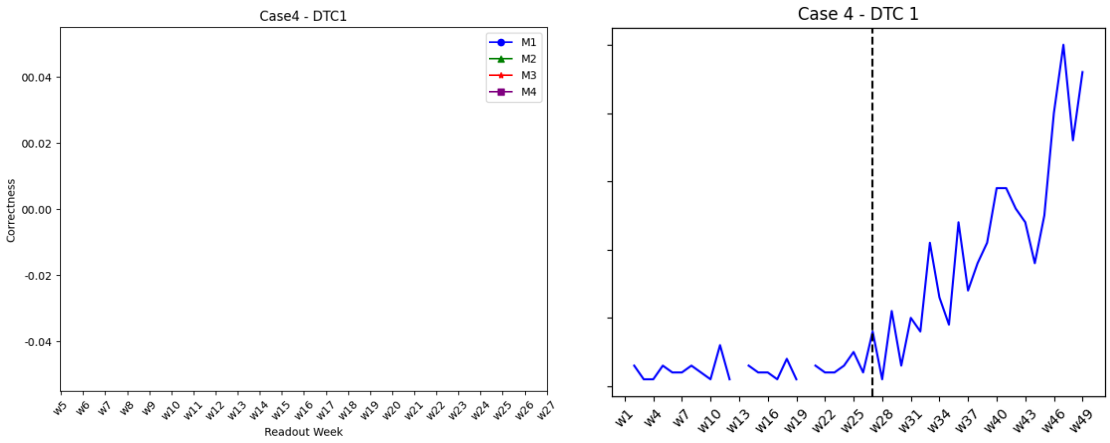

Figure 5.26: Case 4 - Correctness Comparison (left) & DTC trend (right)

The left part of Figure 5.26 presents the correctness results of four models. Represented as an empty diagram, it signifies that none of the models can predict the most affected groups at an early stage; instead, all of them output 'No Warning' in every calculation circle. To understand the reasons, we collected another six months of data after the case was reported. The right part of Figure 5.26 illustrates the DTC trend during this period. The blue line represents the actual DTC volume, and the dashed line marks the reported week. From this figure, we can easily observe that the DTC-triggered situations were at a low level before the case was reported. After week 27, the number of triggered DTCs increases. This distribution indicates that the case was identified at an early stage, and our models lacked predictive

ability in this instance. Another factor causing the models to output 'No Warning' is the TTP threshold. By decreasing this 1% to 0.25% or eliminating this filter, the models would be able to identify the relatively most affected group. However, this action increases the methods' sensitivity to an unnecessary level. Therefore, we have decided to maintain the TTP threshold at 1%.

For case 4, once again, the models are not able to select any features with the provided data. It is a result of reporting the problem before any upward trends. The case was reported early on which resulted in fewer data being able to be inputted into our models, in combination with the thresholds, they are not able to produce any results as seen in the first graph in figure 5.26

5.2.8.2 Evaluation Cases

After constructing the models using Case 1, 3 and 4, Case 2 and 5 were used to evaluate the models. In total, two experiments, one for each case are conducted during the evaluation of the model. A more detailed description of the cases:

Case 2 This case indicates an error associated with the Car Part A Unit. According to the expert's experience, one DTC can be used as a sign to decide whether the problem has occurred or not.

Figure 5.27 illustrates the correctness situation for each model.

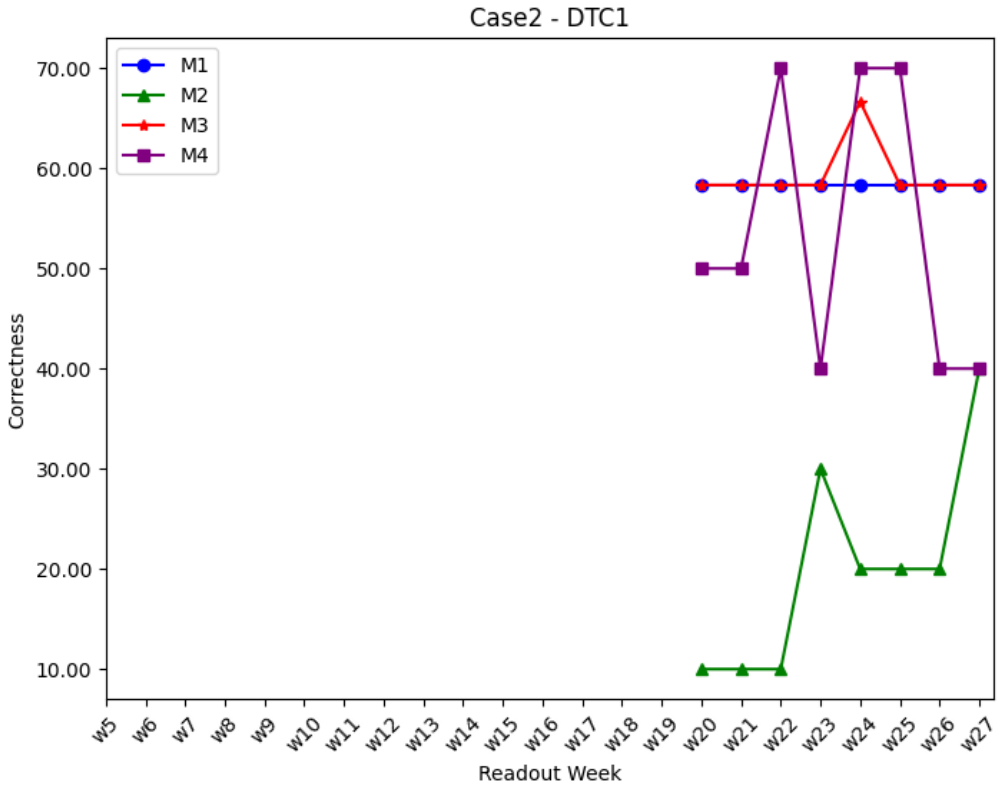

Figure 5.27: Case 2 - Correctness Comparison

As shown in the figure, all the models can warn about features with varying correctness. M1 which is the baseline model can produce a warning with 60% correctness 2 months early. The same goes for M3 but it even reaches 67%. M4 however, is not as stable and varies from 40% to 70%. It can reach 70% correctness at 1.5 months early which is pretty high and accurate. M2 varies from 10% correctness to 40%, it is not able to beat the baseline model at any point during this case and can select the proper groups later than the other models.

Case 5 This case outlines an irregular performance carried out by the Climatization System. One DTC can denote this abnormal action.

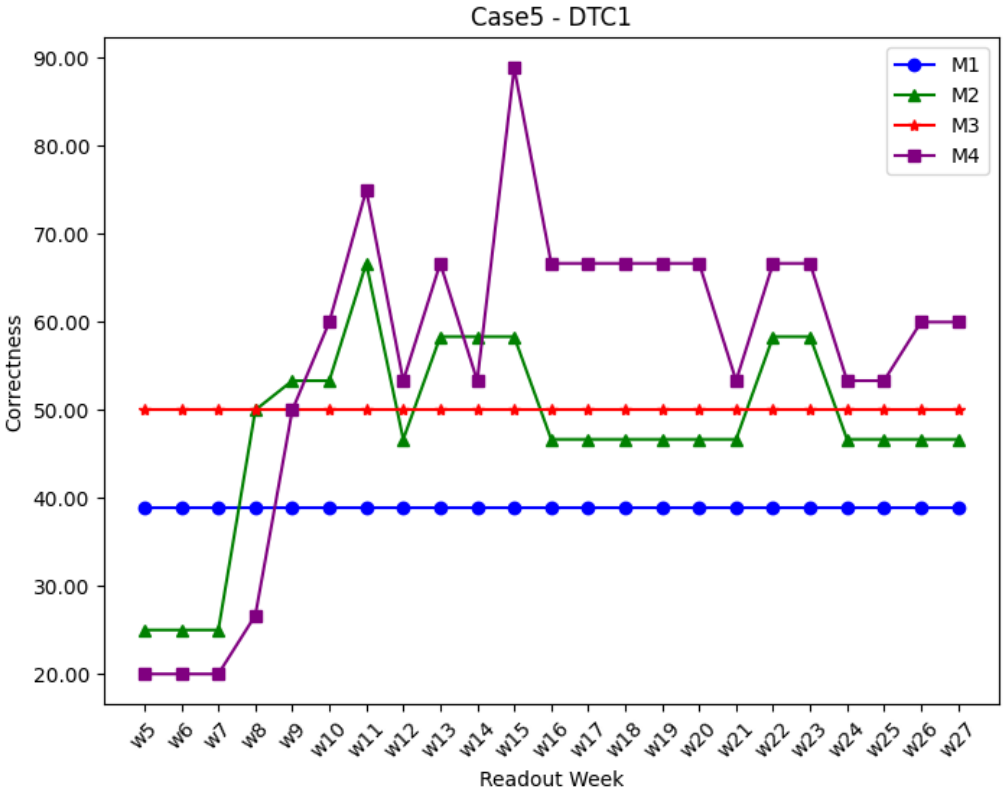

Figure 5.28: Case 5 - Correctness Comparison

Figure 5.28 describes the correctness of each model throughout the entire period. It indicates a great result. While the baseline model M1 is at a constant 39% and able to make a warning every week in the given time frame, M3 can reach 50% every single week. M2 reaches above the 50% mark which is a great result considering how early it can do it. However, M4 also reaches 50% very early, and it even continues up to nearly 90% which is an amazing result. After 4 weeks, it can stay above 50% correctness for the duration of the time frame, it is quite stable and varies from 50% to 88% during 4 months. The reason the models can produce such great results in this case is mainly due to the case being reported quite late which can be seen in Figure 5.29. There is a big upward trend which results in more data provided for our models to go through.

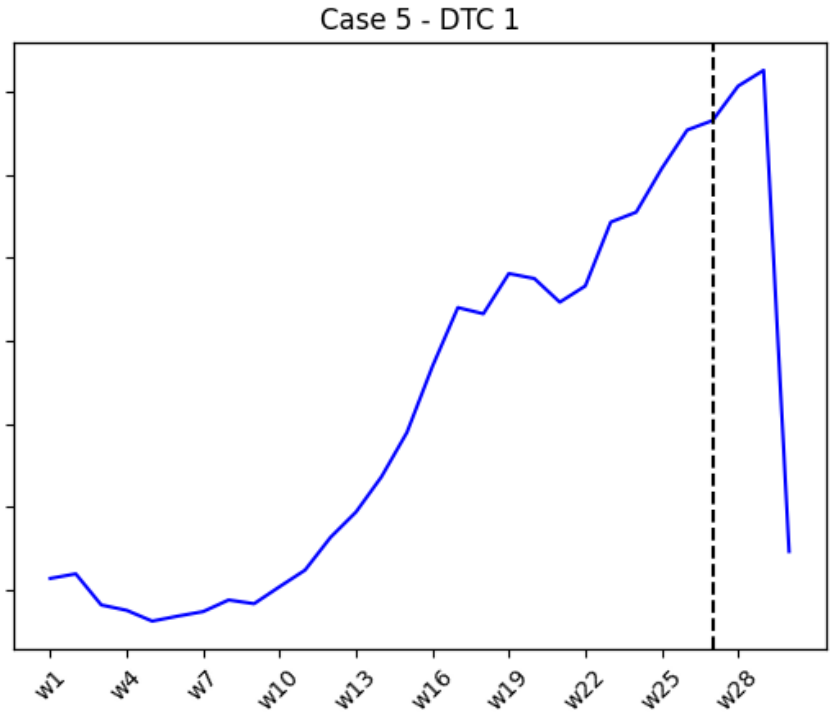

Figure 5.29: Case 5 - DTC trend

5.2.9 Techniques Comparison

During the current study, we constructed four models using the Linear Regression algorithm and K-means clustering: one baseline model and three ML models. All the experiments were conducted under equal environment settings. In this section, we analyze the results of each technique and the correctness of the output to assess their abilities to identify the most affected group at an early stage.

Threshold Control During the laboratory experiment, we employed various thresholds to select the calculation scope, reduce the system's sensitivity, and filter out unnecessary groups. Three thresholds were utilized at the beginning to select the affected groups. When comparing the results between the first round, second round, and the final baseline model, these threshold controls led to a significant increase in correctness. These three thresholds are:

- Calculation Circle Threshold: To ensure sufficient data for providing reliable results at a relatively early stage, we define one calculation as equal to five weeks. The data includes the current week's data and the previous four weeks' data.
- Group Volume Threshold: Any group containing less than 100 vehicles is excluded during the problem identification process.
- TTP Threshold: Groups with less than 1% of triggered DTC vehicles compared to the total number of vehicles are not issued warnings.

Other than these, various thresholds were applied for constructing different ML models:

- P-value Threshold: This threshold is used in linear regression model construction to select statistically significant features. Features with a p-value less than 0.05 are considered.
- Coefficient Threshold: This threshold is utilized when constructing the linear regression model. We select features with positive coefficient values to identify features that have a positive correlation with the probability of triggering DTC.
- Occurrence Threshold: In K-means model construction, the clustering situation has randomness. We repeat the K-means algorithm 25 times and select the groups that appear at least 3 times to ensure a stable result.

Group-by The refined baseline model was built using a single group-by function and outputted the group that occupied the highest TTP value. We divided groups based on their values in six car settings: Model Year, Vehicle Type, Energy Source, Platform, Assembly Plant, and SWPN. Based on the performance of this model, it consistently provided stable outputs that aligned with the experts' decisions. We trusted that the group-by technique applied to these six car settings was indispensable for the problem identification process, as it divided the total vehicles fairly and efficiently. Besides, it provided results at a fast speed. For all subsequent models, we also employed this technique at an early stage as part of the group identification.

K-means Clustering This algorithm is an unsupervised technique that helped us cluster the total vehicles into several sets based on their triggered DTC situations. We used the Elbow method to determine the number of clusters. Subsequently, we ran this algorithm, compared the TTP situation in each cluster, and selected the one with the highest value as the most affected cluster. Later on, we generated and outputted descriptions from six car settings types for the groups included in the most affected cluster, considering them as the most affected groups. There are four advantages to using this technique:

- This technique is straightforward and can handle a large amount of data. The current model construction only contains several DTCs, in the future, we will extend the scope to examine the total DTCs. The K-means algorithm is easy to scale up and satisfy the requirement.
- The results can contain more than one group. Different than the baseline model that only outputs the group that has the highest TTP value, the K-means model can output several groups if they have a similar triggered-DTC situation at the same time as the most affected groups.
- In most cases, this technique helps the model consistently provide a stable result that has a relatively high similarity to the experts' decision.
- The models that employ this technique can generate a warning at an early stage. For some cases, these models suggest the valid most affected group at the first calculation circle.

Linear Regression Algorithm This technique was used to select the features that might impact the triggered DTC situation. There were two models involving the usage of the Linear Regression algorithm: one of them used it in the whole range, and the other used it to select features based on the result given by K-means Clustering. Comparing the results given by these two models, we figured out that Linear Regression is more suitable for performing feature selection based on another model's output. It does not provide a relatively valid result if we use it solo. However, both results indicated that the Linear Regression algorithm is not the most capable technique in this model construction. For example, those two models could not provide a stable answer like the refined baseline model and the K-means model; the result correctness was so jumpy that it could vary from 0 to 70%. The reason that might have caused this issue is the correlations between car settings. In our experiment, we simply treated each vehicle and car set individually, but in real life, there are some inner connections between them. A more specific explanation will be introduced in sections 6.2.2 & 6.2.3.

5.3 Second-round Case Study

The purpose of the second round of case study is to present the models and results to the experts and to gather information regarding the results and their opinion on it to find out what it takes for them to act on an early warning. With the experts' opinion, it could provide us clarity on what correctness is ideal and how early they would like to be warned about a potential problem. More importantly, the follow-up interview can provide us with the answer to RQ3 which is how it could potentially be integrated into the engineer's workflow.

This case study focuses on arbitrary scenarios to gain an understanding of what is the acceptance level for correctness and at what stage. The second case study, which included a follow-up interview, was done with six of the eight people who had attended the first-round case study. We present the results of the interviews in a one-by-one fashion due to transperancy and real-time documentation. The interviews were conducted over a time period of 1-2 weeks with the results being written down the same day. This approach led us to provide a more authentic representation of their thoughts and results. While it resulted in a sequential representation, it reflects the chronological nature of the interview process.

In this section, the interviews are presented in the order in which they were conducted. The sequence of interviews differs from the first stage. However, it was done with the same people, as such, we used the same pseudonymization.

For all the interviewees, we started with a brief introduction about how we built the models and also explained the necessary metrics. Such as correctness and the TTP value which we utilized to interpret the accuracy of our models. The experts agreed on these metrics. We also proceeded with asking questions to find out how early contra how high correctness would be ideal and in which scenarios it could be applicable.

When asked about ideal correctness, interviewee E answered that 50% correctness

would be a nice number for him to act. However, if given the option between 3 months early warning with 50% correctness and 3 weeks early with 90% correctness, E chose 3 weeks early with much higher correctness. “3 months early with low correctness such as 20% would not cause much effort to be put into it, depending on the severity of the component or case it would however be a talking point and be subject to a historical analysis ”. The correctness has a direct impact on how early E would act, there were mentions as well of severity. If their severity is high, the acceptance for the correctness metric might go down.

As for interviewee A, A emphasized how the severity of the case impacts the desired balance between high correctness and early detection. However, when asked about the same scenarios as interviewee E, if the choice was between 50% correctness and 3 months early contra 3 weeks early with 90% correctness, the choice was the earlier detection with lesser correctness. Especially if the criticality of the case is high. A big difference between the interviewees is that A was willing to go down to 3-5% correctness if the case is very severe and 20% in a more general term if it is DTCs that do not directly impact the customer. The most important factor mentioned was the customer criticality, it directly impacts how lenient A would be with correctness. When asked if A would act on a warning if the models highlighted the same most affected group 3 weeks in a row the response was “Yes, I would act on it as it would indicate an increase in the trend which means it is a bigger deal and more likely to be safe to act on”.

The interview with interviewee C had a bit more focus on the technical aspects and the thresholds. For example, when asked about the 100 vehicles threshold for filtering the models, he raised a concern about specific cars that have a low quantity of production. These vehicles could potentially slip through the models due to having lower manufacturing numbers. When asked about different scenarios to find out which % correctness is ideal, C shared that 50% is the ideal correctness to take an action, the higher the better, however, if it is critical, that number goes down. An interesting comment that C mentioned though was that he prefers higher correctness later due to a concept known as ghost DTCs, which are DTCs that trigger without actually having an issue or symptom. If the correctness goes up, then it is more likely to have an issue rather than a ghost DTC.

All the interviewees were asked about their opinions on our proposed thresholds. Most of the answers aligned in some way, interviewee D however had a different view on the calculation cycle. When asked if 5 weeks is enough the direct response was “It depends on the case, if you have a problem that you want to analyze but it does not happen very often, then 5 weeks is most likely not enough data for that specific case ”. This once again emphasizes how different each case is and that the factors are case-sensitive in what the thresholds should be and the accepted correctness contra early warning.

We also discussed about how the models could be integrated into the current problem-identification process. We presented three different options:

- A ticket system plugin that automatically generates a ticket when a warning and the most affected group have been found. Subsequently, the plugin will

add the ticket to the project management dashboard.

- A Power BI plugin that provides warnings and the most affected group based on the DTC-triggered situation. It can be integrated into the current DTC trend visualization workplace.
- An individual standalone application that sends notifications and visualizes the most affected group based on the DTC-triggered situation.

Considering the diverse applications they had already been using in their work, none of them suggested creating another application. However, B was open to the automatic ticket plugin, while A and E shared their experiences when working with automatic ticket generation. They questioned and critiqued the sensitivity of ticket generation conditions. If the trigger condition was too easy, then a lot of unnecessary tickets might have been generated, disturbing their work. On the other hand, if the trigger condition was too difficult to achieve, some severe warnings might not have been found, and they could not have met the requirement to identify the most affected groups in advance.

Furthermore, the experts all appreciated the solution for developing a Power BI plugin to better facilitate their work. First, they could all use this platform proficiently, so no additional education was required. Second, based on the various teams they worked with, there were weekly or bi-weekly catch-up meetings for them to discuss the current DTC trend. They also recommended relating the DTC trend with the plugin proposed, as it could benefit them to have a better understanding of the current trigger situation.

In the end, A, C, E, and G expressed a desire to have the possibility to adjust the thresholds dynamically based on their experience and preferences. For example, the TTP threshold should be adjusted based on the severity of the problem car part. If it is a problem related to customer safety, 1% should be decreased to a lower level to avoid latency when identifying the most affected group.

5.3.1 Integration Solution

All of the experts lean to select the Power BI solution which embedded a warning system in the current DTC trend visualization workplace.

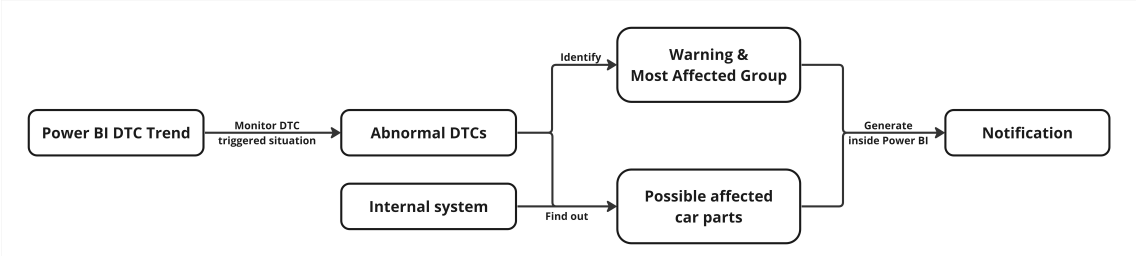

Figure 5.30: The Power BI plugin integration solution

Figure 5.30 provides a brief overview of this solution, illustrating how the results can

be leveraged and seamlessly integrated into the existing workflow to help the experts take proactive actions before the problem scales.

The entire process begins with monitoring the overall DTC trend under various conditions, such as different vehicle types, different markets, specific periods, and more. This trend is visualized using Power BI. By analyzing the DTC-triggered situations, abnormal DTCs are selected for further processing, resulting in the identification of three key objectives:

- Whether a warning should be provided (O1)
- Determining the most affected group (O2)
- Identifying the potentially affected car part (O3)

O1 and O2 are addressed by utilizing our method to construct a predicting model and output the most affected group. As we discussed in section 5.2.9, different techniques have different strengths and limitations. During the interview, participants emphasized the importance of identifying the most affected group early and correctly. Based on the severity of the possible issue, if the goal is to find it out as early as possible, then the K-means model should be adopted to enhance the prediction power of the methodology. If the goal is to figure the most affected group out with higher correctness, then the K-means plus Linear Regression Model should be picked to improve the identification power of the methodology.

Experts have the option to use default thresholds or adjust them based on their past experiences. This adjustment modifies the sensitivity of the identification algorithm. As for O3, experts currently rely on their knowledge to determine the potentially affected car part based on the abnormal DTC. They utilize their personal experience and other internal systems, such as the DTC's naming format and the system storing all DTCs and their associated vehicle problems. However, after discussing with industrial experts, we see the possibility of automatically correlating the abnormal DTC with the potentially affected car part in the future.

Once these three aspects are determined, a notification is generated within Power BI. Experts review these three aspects on a weekly basis and discuss them with colleagues during weekly or bi-weekly catch-up meetings to prioritize potential incidents and decide whether to investigate the root causes.

Currently, the DTC data that we monitor and analyze comes from the workshop. However, Volvo Cars is developing and examining solutions for transferring real-time DTC data directly to the internal database without relying on the workshop. If this is accomplished, both the earliest warning date and the identifying accuracy will be significantly improved.

6

Discussion

In this section, we first make a brief description to summarize the answers to the three research questions. Later, some influencing factors are discussed which might affect our research. For example, the vehicles that are related, the correlations between car settings, the internal connections between DTCs, and the significance of setting and adjusting thresholds. In the end, we explain the advantages and limitations of our current methodology.

6.1 Research Questions Summary

In Section 5.1.4, Section 5.2.9 and Section 5.3.1, we have answered the research questions. Here, we will revisit these questions and provide a summary of them respectively.

6.1.1 RQ1

RQ1 is about clarifying the current process of problem identification using DTCs in Volvo Cars and identifying if there are any limitations and challenges encountered in the case company. We answered this question by conducting and summarizing the first-round case study. Primarily, nowadays, the problem-identification process is a reactive approach that commences when customers encounter issues with their vehicles and consequently take their vehicles to the workshop for further inspection. During the vehicle examination, the DTC triggered situation and other related data get recorded and transferred to Volvo Cars. From then, the engineers start documenting the incident and perform root cause analysis to figure out the main reasons for causing such problems. The entire problem-identification process relies on experts' knowledge and their previous experience. It is an expert-driven approach that limits itself. Also, this process is very case-dependent. Even vehicles that show the same symptoms can have significantly different root causes. All of these limitations and challenges make problem identification a time-consuming process that disappoints the customers and affects the enterprise's reputation.

6.1.2 RQ2

RQ2 compares the techniques that can be used to conquer the mentioned limitations and challenges in the case company found in RQ1. To answer this research question,

we performed some laboratory experiments that constructed various models utilizing several techniques. For example:

- Threshold Control
- Group-by
- K-means Clustering
- Linear Regression Algorithm

In total, four models were built, including a simple baseline model, a linear regression model, a K-means model, and a model that combined K-means clustering and a linear regression algorithm. Different techniques showed different abilities by comparing the outcomes of the model output with the experts' decisions in the correctness dimension. In general, all the selected techniques can improve the problem identification process by identifying the most affected group before the incident documented date. However, due to the scope of applications and several environmental affecting factors, the predicting ability varies from each other. Depending on the severity of the potential issue, utilizing the K-means model early in the process enhances the model's predictive capabilities. If the goal is to accurately identify the most affected group, combining K-means with the Linear Regression Model improves the identification power of the methodology. A more comprehensive comparison for each technique can be found in Section 5.2.9.

6.1.3 RQ3

RQ3 discusses the integration solution by employing the techniques mentioned in RQ2 to the current problem identification workflow. We proposed three different approaches and communicated with the industrial experts during the second round of the case study to see their preferences. The Power BI plugin solution was the preferred choice according to the feedback. This methodology monitors abnormal DTCs and provides necessary warnings about the most affected group. The majority of the experts also expressed the desire for a weekly update of the most affected group so that they can check up with other engineers to make the proper prioritization. This solution was considered the best due to familiarity with the Power BI platform and the high degree of adaptation to their current workflows. The experts also wanted to dynamically adjust various thresholds in the model construction to optimize the output based on their personal preferences, knowledge, and experience.

6.2 Influencing Factors

Throughout the entire research, we strive to ensure that all experiments are conducted in the same environmental settings. For example, we define a calculation cycle as 5 weeks, group by the same vehicle setting types, and maintain group volume threshold and TTP threshold at the same level, among other factors. However, there are a few influencing factors that might affect the model construction process and identification accuracy. In this section, we summarize these influencing factors

into four categories: vehicle dependencies, car settings, DTC selection, and experts' decisions. A comprehensive description for each category follows.

6.2.1 Vehicle Dependencies

In the modern automation industry, the development of new vehicles is an iterative and continuous process. One approach is leveraging the successes from the previous development. In this way, the same design logic may be shared among various cars. This practice results in a notable lack of independence between different vehicle types, where the advantages and disadvantages of one model can significantly influence others. In other words, when an issue arises for one vehicle type, the failure can extend to others with a similar system design.

For instance, despite having distinct exterior and interior designs, two vehicles may share identical climatization systems. The hardware components for this system are provided by the same supplier. Both of the vehicles also utilize the same SWPN to perform version control. Consequently, when a climate-related problem shows up for one vehicle type, the other might also be impacted due to the corresponding system composition.

Industry experts occupying various experiences can navigate these interconnected complexities by inferring knowledge from previous cases. They can establish connections and foresee potential impacts on similar affected groups with limited data within such contexts. However, our current data-driven methodology does not have the ability to function as an expert due to the inadequacy of related data. Moreover, the whole identifying model is constructed with a precondition that every vehicle type is individual.

6.2.2 Car Settings

There are two types of car setting issues that might affect the research: insufficient environmental car settings and internal connections between car settings.

Our current data comes from the workshop and the car settings we select mostly describe the vehicle condition in a driving cycle. Initially, the study included eight car settings: Model Year, Vehicle Type, Energy Source Type, Platform, Assembly Plant, Assembly Week, SWPN, Readout Date, and Market. However, Assembly Week and Market split the total vehicle data so discretely that the volume is insufficient for performing threshold control and machine learning. Consequently, we have eliminated these two car settings and constructed the models using data collected from the remaining six. However, during the discussion with the industrial experts, they also mentioned some other environmental factors that might be required for some specific cases. For example, the outdoor temperature can be useful when investigating an issue related to the climatization system. The differences between temperatures can further affect the DTC-triggered situation. Adding more environmental car settings that describe the external aspects of a driving cycle can help analyze the situation more comprehensively.

The correlations between car settings also impact the accuracy of the results. Analyzing the outcomes of both the refined baseline model and the K-means model, we observe that the model output consistently narrows down the most affected group by including some unnecessary filtering conditions related to car settings. In our model construction, we employ a linear regression algorithm for feature selection to identify the most influential factors, thereby expanding the scope of the included group and enhancing the similarity with the expert decisions. While this approach proves effective in certain cases, it also noticeably affects the stability of the model's output. The correctness comparison diagrams in Section 5.2.8 reveal that models encompassing a linear regression algorithm often exhibit significant fluctuations. One of the contributing factors is the unaccounted interconnections between the six-car settings. For example, one vehicle type consistently corresponds to one specific platform, and certain types of electric vehicles are exclusively assembled in specific plants. One precondition of our experiments is treating each car setting separately. Thus, the inter-dependency among car settings reduces the effectiveness of the linear regression algorithm for feature screening.

6.2.3 DTC Selection

The DTCs involved in our study are collected from issue tickets and the first-round case study. We make a fair assumption that each DTC is a standalone entity, treating each DTC separately. Following this precondition, we conducted nine experiments from three cases (Case 1, Case 3, and Case 4) for model construction and two experiments from two cases (Case 2 and Case 5) for model evaluation.

Revisit the correctness results presented in Section 5.2.8. We can observe that different DTCs have varying identification capabilities. Taking four DTCs from Case 1 as an example, the models constructed by:

- DTC 1: Earliest warning week is w12 with an average correctness of 35.54
- DTC 2: Does not have the ability to give a warning.
- DTC 3: Earliest warning week is w18 with an average correctness of 38.48
- DTC 4: Does not have the ability to give a warning.

In conjunction with the discussions with industrial experts during the second round case study, they mentioned that there are interconnections and logical relationships between DTCs. Some can be identified as primary DTCs with the highest ability to foresee the occurrence of issues, while others are determined as subsidiary DTCs caused by the chain reactions triggering the primary DTCs. Subsidiary DTCs have a smaller ability to identify possible errors compared to the primary DTCs. There are various methods to distinguish them, including checking the DTC naming format, comparing the triggered time stored in the DTC snapshot, or inspecting vehicle symptoms related to DTCs. Sometimes, experts also observe the triggered situation by considering all the DTCs as a whole.

Taking inspiration from that, we facilitate an easy comparison of the correctness of the baseline model results for Case 1. This involves comparing two assumptions: the first

considers all DTCs as a whole, while the second treats each DTC individually. When considering all DTCs as a whole, the TTP is calculated regardless of which DTC the vehicle triggered. In Figure 6.1, the black line with dots represents the first situation and the red line with stars stands for the second situation. The figure illustrates that, although the correctness remains at the same level, the first condition can produce a more stable result. This observation highlights the significant impact that different approaches to handling associated DTCs can have on the modeling result, underscoring the importance of carefully considering the treatment of individual DTCs in the analytical process. This observation emphasizes that employing different approaches to handle associated DTCs can significantly impact the modeling result.

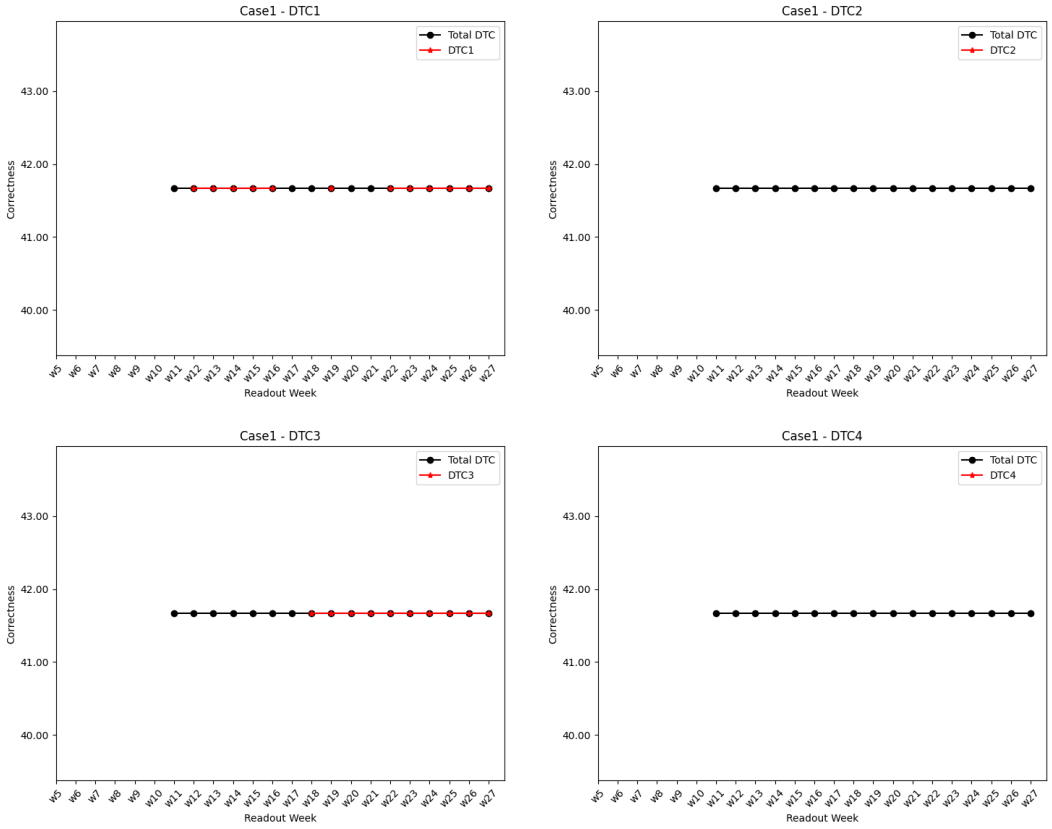

Figure 6.1: Case 1 Correctness Comparison - all DTCs & individual DTC

6.2.4 Expert Decision

The correctness of the output calculates the similarity between the model's result and the expert's decision. Our current expert decision is derived from summarizing information from the ticket and the initial case study with industrial experts. However, the correctness and comprehensiveness of the expert decision are not guaranteed to be 100%. They might change over time due to progress in identifying the root cause. Based on the experts' experience, knowledge, and professional level, they may also have different opinions.

For example, in Case 5, at the research starting point, only one vehicle type is described in the ticket. During the discussion with the industrial experts, they

only mentioned the same vehicle type that might have a problem. In contrast, the K-means model, at the beginning, identifies three possible most affected groups, including the one they emphasized. If we calculate the result's correctness in this situation, it is relatively low. However, after the QEs conducted further investigations on the issue, they discovered that the other two vehicle types could also be affected. Because of this, the actual result's correctness should be much higher.

This example illustrates that the dynamic changes in the expert decision can affect the correctness calculation, potentially leading to an underestimation of the model's performance. A low correctness score does not necessarily indicate a false result; instead, it may suggest the presence of unforeseen conditions that QEs have not yet realized but are discovered by the models. Furthermore, this indirectly validates the potential ability of our model to serve as a valuable tool for providing suggestions to industrial experts.

6.2.5 Correctness Calculation

In this study, we use correctness to evaluate our model result. However, it is not necessary to always acquire a high correctness since the maturity of the experts' decision can significantly vary the correctness. The current comparisons have been calculated regarding the cases as they have already been processed. This means that the RCA is already done, and an extensive amount of man-hours has been put into the case to find the most affected groups. The maturity of the experts' decision is high. However, when the ticket is initially created, the maturity of the experts' decision is low since not much effort has been put in. If we compare the correctness at this time, the correctness output does not align with our experiment result. When a ticket is reported, there is no real-time information about the most affected groups; the information that exists in the description is most likely "there is currently a problem that affects one type of vehicle". Subsequently, this type of vehicle will be treated as the most affected group. Table 6.1 presents the number of car settings included in the ticket description and the final experts' decision. For example, the ticket reporting the Case 1 incident initially mentioned two car settings. However, after conducting the RCA, the experts determined that the most affected group should contain three car settings. This difference in the most affected groups will affect the correctness calculation.

Case No.	Ticket Description	Final Experts' Decision
Case 1	2	3
Case 2	1	5
Case 3	1	1
Case 4	1	2
Case 5	2	4

Table 6.1: Number of Car Settings included in the Ticket Description & Final Experts' Decision

Comparing our models to the cases after RCA has been applied makes the correctness

levels more impressive and fair. It also indirectly proves the capability of our models since they will most likely output a more specific and comprehensive affected group that outperforms the initial manual report and facilitates their early problem identification.

6.3 Advantages & Limitations

During the second-round case study, the industrial experts validated and appreciated the performance of our models. They also expressed a desire to integrate them into their current workflow to enhance their capability to take proactive actions, reduce problem identification time, further enhance customer safety, improve user experience, and reduce Volvo Cars' warranty costs. In this section, we will discuss some of the advantages of our model and also mention some limitations that account for the current situation.

6.3.1 Continuous & Dynamic Output

In Sections 5.1.3 and 5.1.4, we discuss the current problem identification process and the associated limitations and challenges. In the Problem Pre-investigation process, QE indicates the direction of subsequent Root Cause Analysis. During the RCA process, Volvo Cars and suppliers collaborate to determine the primary cause of the issue. However, based on the related report tickets and two case studies with industry experts, we discovered that the current workflow does not fully support continuous tracking of the most affected group. Industrial experts must make well-informed and solid decisions based on diverse and extensive data.

Our data-driven methodology empowers problem identification by providing continuous and dynamic output as it runs weekly. It can also be adjusted to a shorter running cycle, but it requires a guarantee of sufficient data quantity for model training. The dynamically changing results reflect fluctuations in the most affected group over time. Additionally, constructing a model does not take much time, and it can be easily deployed to applications like Jenkins. This is an automation tool commonly used for continuous integration and continuous delivery in software development. In this way, we can construct the model and receive results automatically on a scheduled basis in future development. These periodically generated results can be valuable for industrial experts in foreseeing changes in the most affected group over time and facilitating discussions during their weekly or bi-weekly catch-up meetings.

6.3.2 Providing Data-driven insight

As explained in Section 5.1.4, the current problem identification process heavily relies on participants' experience and knowledge. Three subsidiary processes are particularly evident: workshop technicians use their knowledge to make a selection about the related DTC based on vehicle symptoms during Data Collection, and Quality Engineers (QEs) utilize their experience to figure out the primary issue during the Problem Pre-investigation and Root Cause Analysis processes. Expert-driven

methods have certain benefits; for instance, they can break down complex problems, learn from previous mistakes, and provide innovative solutions. However, there are noticeable drawbacks. Differences in knowledge and experience can introduce decision biases; relying too much on past experiences may lead to over-fitting; and when experts leave the current organization, knowledge transfer can be hard to secure.

Our approach aims to shift the process towards a more data-driven direction by analyzing abnormal DTC-triggered situations. We are dedicated to uncovering stories hidden in the data, such as providing necessary warnings and identifying the most affected groups. In the current situation, we utilize real workshop data to generate results. In the foreseeable future, Volvo Cars will begin to transform real-time DTC data into production, which intends to collect DTC data during the driving circle and insert it directly into the internal database in real time. There will be a noticeable increase in both efficiency and accuracy. Furthermore, with ongoing refinement in model construction and the ability to collect and integrate more data from related internal systems, such as those linking DTCs and possible vehicle symptoms, this methodology can be defined as an advanced data-driven approach.

6.3.3 Static Thresholds

Our model construction utilizes several thresholds, as detailed in Section 5.2.9, to adjust the sensitivity of our methods (determine whether a warning should be issued) and identify the most affected group. These thresholds are borrowed from successful practices in other departments and go through some careful selections and examinations. While they demonstrate solid performance in certain cases, during the second-round interview with industrial experts, they think this is a limitation of the current method and propose a need for dynamically changing these parameters.

This is mainly because each case should be treated individually. In their opinion, finding common thresholds that fit every situation is challenging; instead, they advocate for a case-sensitive approach. Case severity, cited by various experts, is a parameter that calculates how urgently a problem needs to be solved. The threshold values, especially the TTP threshold, should be adjusted accordingly. For example, a problem related to customer safety is always assigned the highest priority. The TTP threshold should be reduced from 1% to a lower level, such as 0.5% or even 0.1%, to provide warnings and identify the most affected group at an earlier stage. Another expert suggests adjusting the Group Volume Threshold for specific vehicle types with limited quantities. For these types, triggering 100 vehicles means the situation has become very serious. He would like to decrease this threshold to ensure the opportunity to early identify the most affected group is not lost.

6.3.4 Feature Selection

Two of our models involve using the linear regression algorithm to perform feature selection to car settings, as mentioned in Section 5.2.8, resulting in a noticeable amplitude in the correctness comparison diagram. We also analyze the dependencies between vehicle and car settings that can cause the data not to follow linear regression

in Section 6.2.1 & 6.2.2. This poses a limitation to our current methodology. In the present state, we rely on experts to evaluate the influencing car settings after providing them with the most affected group based on their knowledge, experience, DTC type, and related vehicle symptoms. However, some documents record the interconnections between vehicles and car settings, also most experts understand those. In the future, we would like to research more on how to transfer this specific knowledge into a data-driven method and how to formulate a better feature selection algorithm to reduce unnecessary filtering conditions.

7

Conclusion

This study is a step in the direction of using a data-driven approach to provide various insights as a complement to expert competence. We have explored the possibility of automated early problem identification based on DTCs to identify the most affected group before the problem scales. We end up with some promising results. Some models are better than others but each iteration provides an understanding which allows us to refine further and develop better models. In total, we develop four models using two ML techniques that can automate early problem detection to different degrees with varying results thus successfully validating the possibility of automated early problem identification using DTCs.

Our data-driven methodology can predict errors months in advance and achieve correctness levels that meet experts' expectations. In the second-round case study, the experts expressed great interest in integrating our data-driven approach into their current workflow. They hold a positive attitude, believing it significantly enhances their work capability.

This study provides insight into how we construct our models, what data we collect, and how we have utilized the data. However, it could benefit from further improvements, which will be mentioned in the following section.

7.1 Future Work

Based on the experience gained from this research and discussions with industrial experts and supervisors, we have identified several areas for further exploration.

Better model construction & feature selection ability We employed two types of ML algorithms in our study. However, there will always be a better and easier model construction approach that can provide a warning at an earlier stage and occupy a better correctness output. Especially the feature selection ability, the current linear regression algorithm can improve the correctness to a high degree for some cases, as the trade-off, the ability to provide stable results is reduced. The size of the evaluation set affects the construction of the models too. Having a larger data set could provide more accurate and reliable results which can be assessed. It could also change which models performed better as larger sets reduce the impact of randomness in the evaluation.

Dynamic Threshold From the interviews, we understand that problem identification is a very case-sensitive process and it is hard to find some static thresholds that suit every case. In the future, we would like to examine whether there is a way, for example employing ML to facilitate experts dynamically adjusting the thresholds based on different situations.

DTC automatic related In our proposed Power BI integration solution, after the system detects the abnormal DTC, experts need to relate the possible affected car parts and the possible vehicle symptoms with the DTC-triggered situation using their previous experience, knowledge, and with the help of other internal systems. It will be beneficial to construct a data-driven model that automatically relates them together. It can further shorten the problem-identification process.

Bibliography

[1] N. Sekitoleko, F. Evbota, E. Knauss, A. Sandberg, M. Chaudron, and H. H. Olsson, "Technical dependency challenges in large-scale agile software development," in *Agile Processes in Software Engineering and Extreme Programming: 15th International Conference, XP 2014, Rome, Italy, May 26-30, 2014. Proceedings 15*, Springer, 2014, pp. 46–61.

[2] Y. Wilhelm, P. Reimann, W. Gauchel, and B. Mitschang, "Overview on hybrid approaches to fault detection and diagnosis: Combining data-driven, physics-based and knowledge-based models," *Procedia CIRP*, vol. 99, pp. 278–283, 2021, 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020, ISSN: 2212-8271. DOI: https://doi.org/10.1016/j.procir.2021.03.041. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827121003152.

[3] M. Al-Zeyadi, J. Andreu-Perez, H. Hagras, *et al.*, "Deep learning towards intelligent vehicle fault diagnosis," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–7. DOI: 10.1109/IJCNN48605.2020.9206972.

[4] P. Pirasteh, S. Nowaczyk, S. Pashami, *et al.*, "Interactive feature extraction for diagnostic trouble codes in predictive maintenance: A case study from automotive domain," in *Proceedings of the Workshop on Interactive Data Mining*, ser. WIDM'19, Melbourne, VIC, Australia: Association for Computing Machinery, 2019, ISBN: 9781450362962. DOI: 10.1145/3304079.3310288. [Online]. Available: https://doi.org/10.1145/3304079.3310288.

[5] S. M. Virk, A. Muhammad, and A. M. Martinez-Enriquez, "Fault prediction using artificial neural network and fuzzy logic," in *2008 Seventh Mexican International Conference on Artificial Intelligence*, 2008, pp. 149–154. DOI: 10.1109/MICAI.2008.38.

[6] C.-S. A. Gong, C.-H. S. Su, Y.-H. Chen, and D.-Y. Guu, "How to implement automotive fault diagnosis using artificial intelligence scheme," *Micromachines*, vol. 13, no. 9, 2022, ISSN: 2072-666X. DOI: 10.3390/mi13091380. [Online]. Available: https://www.mdpi.com/2072-666X/13/9/1380.

[7] P. Broadwell, "Component failure prediction using supervised naıve bayes classification," *http. V/citeseerx. ist. psu. edu/viewdoc/summary*, 2002.

[8] D. G. Vrachkov and D. G. Todorov, "Automotive diagnostic trouble code (dtc) handling over the internet," in *2018 IX National Conference with International Participation (ELECTRONICA)*, IEEE, 2018, pp. 1–3.

[9] D. R. Ferreira, T. Scholz, and R. Prytz, "Importance weighting of diagnostic trouble codes for anomaly detection," in *Machine Learning, Optimization, and Data Science: 6th International Conference, LOD 2020, Siena, Italy, July 19–23, 2020, Revised Selected Papers, Part I 6*, Springer, 2020, pp. 410–421.

[10] D. Mahto and A. Kumar, "Application of root cause analysis in improvement of product quality and productivity," *Journal of Industrial Engineering and Management (JIEM)*, vol. 1, no. 2, pp. 16–53, 2008.

[11] E. e Oliveira, V. L. Miguéis, and J. L. Borges, "Automatic root cause analysis in manufacturing: An overview & conceptualization," *Journal of Intelligent Manufacturing*, vol. 34, no. 5, pp. 2061–2078, Jun. 2023, ISSN: 1572-8145. DOI: 10.1007/s10845-022-01914-3. [Online]. Available: https://doi.org/10.1007/s10845-022-01914-3.

[12] A. Lokrantz, E. Gustavsson, and M. Jirstrand, "Root cause analysis of failures and quality deviations in manufacturing using machine learning," *Procedia Cirp*, vol. 72, pp. 1057–1062, 2018.

[13] K. B. Percarpio, B. V. Watts, and W. B. Weeks, "The effectiveness of root cause analysis: What does the literature tell us?" *The Joint Commission Journal on Quality and Patient Safety*, vol. 34, no. 7, pp. 391–398, 2008.

[14] J. Martin-Delgado, A. Martınez-Garcıa, J. M. Aranaz, J. L. Valencia-Martın, and J. J. Mira, "How much of root cause analysis translates into improved patient safety: A systematic review," *Medical Principles and Practice*, vol. 29, no. 6, pp. 524–531, 2020.

[15] I. El Naqa and M. J. Murphy, "What is machine learning?" In *Machine Learning in Radiation Oncology: Theory and Applications*, I. El Naqa, R. Li, and M. J. Murphy, Eds. Springer International Publishing, 2015, pp. 3–11, ISBN: 978-3-319-18305-3. DOI: 10.1007/978-3-319-18305-3_1. [Online]. Available: https://doi.org/10.1007/978-3-319-18305-3_1.

[16] R. Bhardwaj, V. Dixit, and A. Upadhyay, "A fuzzy intra-clustering approach for load balancing in peer-to-peer system," *Journal of Information and Computing Science*, vol. Vol. 7, No. 1, pp. 019–024, Dec. 2011.

[17] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009, vol. 2.

[18] M. T. Guerreiro, E. M. A. Guerreiro, T. M. Barchi, *et al.*, "Anomaly detection in automotive industry using clustering methods—a case study," *Applied Sciences*, vol. 11, no. 21, 2021, ISSN: 2076-3417. DOI: 10.3390/app11219868. [Online]. Available: https://www.mdpi.com/2076-3417/11/21/9868.

[19] T. M. Kodinariya, P. R. Makwana, *et al.*, "Review on determining number of cluster in k-means clustering," *International Journal*, vol. 1, no. 6, pp. 90–95, 2013.

[20] M. Syakur, B. Khotimah, E. Rochman, and B. D. Satoto, "Integration k-means clustering method and elbow method for identification of the best customer profile cluster," in *IOP conference series: materials science and engineering*, IOP Publishing, vol. 336, 2018, p. 012 017.

[21] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical software engineering*, vol. 14, pp. 131–164, 2009.

[22] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, pp. 77–101, Jan. 2006. DOI: 10.1191/1478088706qp063oa.

[23] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting empirical methods for software engineering research," *Guide to advanced empirical software engineering*, pp. 285–311, 2008.

[24] M. T. Guerreiro, E. M. A. Guerreiro, T. M. Barchi, *et al.*, "Anomaly detection in automotive industry using clustering methods—a case study," *Applied Sciences*, vol. 11, no. 21, p. 9868, 2021.

[25] X. Zhou, Y. Jin, H. Zhang, S. Li, and X. Huang, "A map of threats to validity of systematic literature reviews in software engineering," in *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)*, IEEE, 2016, pp. 153–160.

[26] K. Cahit, "Internal validity: A must in research designs," *Educational Research and Reviews*, vol. 10, no. 2, pp. 111–118, 2015.

[27] H. K. Wright, M. Kim, and D. E. Perry, "Validity concerns in software engineering research," in *Proceedings of the FSE/SDP workshop on Future of software engineering research*, 2010, pp. 411–414.

[28] R. Feldt and A. Magazinius, "Validity threats in empirical software engineering research-an initial survey.," in *Seke*, 2010, pp. 374–379.

[29] P. Ralph and E. Tempero, "Construct validity in software engineering research and software metrics," in *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*, 2018, pp. 13–23.

[30] D. I. Sjøberg and G. R. Bergersen, "Construct validity in software engineering," *IEEE Transactions on Software Engineering*, vol. 49, no. 3, pp. 1374–1396, 2022.

[31] B. B. N. de França and G. H. Travassos, "Simulation based studies in software engineering: A matter of validity," *CLEI electronic journal*, vol. 18, no. 1, pp. 4–1, 2015.

[32] F. Pargent, B. Bischl, and J. Thomas, "A benchmark experiment on how to encode categorical features in predictive modeling," *München: Ludwig-Maximilians-Universität München*, 2019.

[33] G. Di Leo and F. Sardanelli, "Statistical significance: P value, 0.05 threshold, and applications to radiomics—reasons for a conservative approach," *European radiology experimental*, vol. 4, no. 1, pp. 1–8, 2020.

[34] N. Kosaraju, S. R. Sankepally, and K. Mallikharjuna Rao, "Categorical data: Need, encoding, selection of encoding method and its emergence in machine learning models—a practical review study on heart disease prediction dataset using pearson correlation," in *Proceedings of International Conference on Data Science and Applications: ICDSA 2022, Volume 1*, Springer, 2023, pp. 369–382.

[35] S. K. Patnaik, S. Sahoo, and D. K. Swain, "Clustering of categorical data by assigning rank through statistical approach," *International Journal of Computer Applications*, vol. 43, no. 2, pp. 1–3, 2012.

[36] S. Kumar *et al.*, "Efficient k-mean clustering algorithm for large datasets using data mining standard score normalization," *Int. J. Recent Innov. Trends Comput. Commun*, vol. 2, no. 10, pp. 3161–3166, 2014.

[37] L. I. Kuncheva and D. P. Vetrov, "Evaluation of stability of k-means cluster ensembles with respect to random initialization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 11, pp. 1798–1808, 2006.

[38] Z. Zhang, J. Zhang, and H. Xue, "Improved k-means clustering algorithm," in *2008 Congress on Image and Signal Processing*, IEEE, vol. 5, 2008, pp. 169–172.

[39] P. Fränti and S. Sieranoja, "How much can k-means be improved by using better initialization and repeats?" *Pattern Recognition*, vol. 93, pp. 95–112, 2019.