



CHALMERS
UNIVERSITY OF TECHNOLOGY



Advancing Vehicle Diagnostic: Exploring the Application of Large Language Models in the Automotive Industry

Master's thesis in Systems, Control and Mechatronics
Master's thesis in Mobility Engineering

ABHIJEET SINGH DHILLON
ANDREA TORRESIN

Department of Industrial and Materials Science

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

MASTER'S THESIS 2024

Advancing Vehicle Diagnostic: Exploring the Application of Large Language Models in the Automotive Industry

ABHIJEET SINGH DHILLON
ANDREA TORRESIN



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Industrial and Materials Science
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024

Advancing Vehicle Diagnostic: Exploring the Application of Large Language Models
in the Automotive Industry

ABHIJEET SINGH DHILLON, ANDREA TORRESIN

© ABHIJEET SINGH DHILLON, ANDREA TORRESIN, 2024.

Academic supervisor: Mohan Rajashekarappa, Chalmers University of Technology

Industrial supervisor: Mike C., Truck and Bus Manufacturer Company

Examiner: Anders Skoogh, Chalmers University of Technology

Master's Thesis 2024

Department of Industrial and Materials Science

Chalmers University of Technology

SE-412 96 Gothenburg

Telephone +46 31 772 1000

Gothenburg, Sweden 2024

Advancing Vehicle Diagnostic: Exploring the Application of Large Language Models in the Automotive Industry

ABHIJEET SINGH DHILLON

ANDREA TORRESIN

Department of Industrial and Materials Science
Chalmers University of Technology

Abstract

The integration of Artificial Intelligence (AI) in the automotive industry has enhanced vehicle functionality, making them smarter, safer, and more efficient. However, the potential of AI in vehicle fault diagnostics and troubleshooting remains untapped. This thesis explores the application of Large Language Models (LLMs) to improve vehicle diagnostics, a field that traditionally relies on time consuming manual data interpretation, which can be inefficient and error-prone. The study aims to showcase and assess the capabilities of Claude v2 by Anthropic in rapidly processing and analyzing a large vehicle failure dataset.

Our research uses a comparative analysis methodology, evaluating the performance of a LLM with traditional machine learning models in diagnosing and classifying vehicle faults. The open source data consists of maintenance records, problem descriptions, and customer feedback, entered into National Highway Traffic Safety Administration (NHTSA) vehicle owner's complaint database. The findings suggest that the LLM can, up to a certain level of complexity, accurately analyze fault descriptions and predict the category of failures. It can also extract useful information from complaint descriptions that helps in the diagnostic process and decision-making. However, the accuracy of the LLM's decisions decreases as the task complexity increases and approaches real-world scenarios. Additionally, the results indicate that traditional supervised machine learning classifiers generally perform better in text classification tasks within our automotive faults context.

This study wants to contribute to academic knowledge in AI applications and offer some insights for automotive industry professionals. It introduces a methodology that promises to serve as a foundation for advancements in vehicle diagnostics, aligning with the ongoing shift towards automation and efficiency.

Keywords: Machine Learning, Large Language Model, automotive, fault diagnosis, troubleshooting, text classification, prompt engineering.

Acknowledgements

This thesis has been carried out at a Heavy Truck and Bus Manufacturer Company in Sweden, in collaboration with the Department of Industrial and Materials Science at Chalmers University of Technology. We would like to extend our gratitude to our academic advisor, Mohan Rajashekarappa, for his guidance in shaping this thesis, as well as to our examiner Anders Skoogh. We are also deeply grateful to our industrial advisor, Mike C., for his constant support and for always bringing positivity and motivation; and to our group manager, Viveca I., for believing in us. This thesis would not have been possible without the expertise and support offered by Olof S.; we truly appreciate his contributions.

During the preparation of this work, we used ChatGPT 4.0 to proofread and enhance readability. After using this tool, we reviewed and edited the content and take full responsibility for the final version.

Abhijeet Singh Dhillon, Gothenburg, June 2024
Andrea Torresin, Gothenburg, June 2024

List of Acronyms

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

LLM	Large Language Model
ML	Machine Learning
IDSS	Intelligent Decision Support System
TRL	Technology Readiness Level
AI	Artificial Intelligence
PLM	Product Lifecycle Management
TF-IDF	Term Frequency-Inverse Document Frequency
IDF	Product Lifecycle Management
MAP	Maximum A Posterior
CNB	Complement Naive Bayes
SVM	Support Vector Machines
RF	Random Forest
GBM	Gradient Boosting Machines
KNN	K-Nearest-Neighbor
NLP	Natural Language Processing
PLMs	Pre-trained Language Models
NHTSA	National Highway Traffic Safety Administration

Contents

List of Acronyms	ix
List of Figures	xv
List of Tables	xvii
1 Introduction	1
1.1 Background	1
1.1.1 Problem Formulation	2
1.2 Aim	2
1.3 Research Questions	3
1.4 Field Quality process and possibility for improvement	3
1.4.1 Field Quality process in brief	3
1.4.2 Possible Improvement	4
1.5 Delimitations	5
2 Theory	7
2.1 Modularization	7
2.1.1 Overview	7
2.1.2 Modularisation At Heavy Truck and Bus Company	8
2.2 Machine Learning	9
2.2.1 Text Classification	9
2.2.1.1 Preprocessing	9
2.2.1.2 Feature Extraction	10
2.2.1.3 Model Training	10
2.2.1.3.1 Naive Bayes Classifier	10
2.2.1.3.2 Support Vector Machine	11
2.2.1.3.3 Random Forest	12
2.2.1.3.4 Gradient Boosting Classifier	12
2.2.1.3.5 K-Nearest Neighbor (kNN)	13
2.2.1.4 Evaluation Metrics	13
2.2.2 Large Language Models	14
2.2.2.1 Components of LLMs [30]:	14
2.2.2.2 Instruction Tuning	16
2.2.2.3 Context Window	17
2.2.2.4 Augmented LLMs	17
2.2.2.5 LLMs-Powered Agents	18

2.2.2.6	Computational Cost of LLMs	19
2.2.2.7	Security Risks in LLMs	20
2.2.2.8	Evaluation metrics for LLMs	21
2.2.3	Large Language Model CLAUDE v2	22
3	Methodology	25
3.1	CRISP-DM	25
3.1.1	Business Understanding	26
3.1.2	Data Understanding	26
3.1.2.1	User Case analysis: understanding data sources and their interactions	26
3.1.2.2	Structure definition	28
3.1.3	Data Preparation	29
3.1.3.1	Dataset definition	29
3.1.3.2	Text Classification Subsystem	30
3.1.3.3	Recommendation Subsystem	30
3.1.4	Modeling	31
3.1.4.1	Purpose	31
3.1.4.2	Data Management	32
3.1.4.3	ML models	32
3.1.4.4	Large Language Models (LLMs)	32
3.1.5	Deployment	32
4	Results	35
4.1	Business Understanding	35
4.2	Data Understanding	36
4.2.1	NHTSA dataset	36
4.3	Data Preparation	36
4.3.1	Subset 1	37
4.3.2	Subset 2	38
4.4	Modelling and Evaluation	39
4.4.1	Text Classification	39
4.4.1.1	Traditional Machine Learning Approaches for Text Classification	39
4.4.1.1.1	Hyperparameter tuning of classifiers	40
4.4.1.1.2	ML models performance	40
4.4.1.2	Large Language Model as a Text Classifier	41
4.4.1.2.1	Prompt 1	42
4.4.1.2.2	Prompt 2	42
4.4.1.2.3	Prompt 3	42
4.4.1.2.4	LLM performance	42
4.4.1.2.5	Overall Comparison	43
4.4.2	Feature Extraction	44
4.4.3	Recommendation System	45
4.4.3.1	Retrieval Augmented Based Recommendation System	45
4.4.3.1.1	Evaluation of the Recommendation System	47
4.5	Deployment	49

4.5.1	Application Overview	49
4.5.2	Readiness Level	50
5	Discussion	51
5.1	Data Understanding	51
5.2	Data Preparation	52
5.3	Modeling	53
5.3.1	Text Classification	53
5.3.2	Feature Extraction	54
5.3.3	Recommendation System	55
5.4	Evaluation	56
5.4.1	Related Existing Literature	57
5.5	Deployment	58
6	Conclusion and Recommendation	61
6.1	Conclusion	61
6.2	Recommendation	62
A	Appendix 1	I
A.0.0.0.1	Prompt 1 - Subset 1	I
A.0.0.0.2	Prompt 2 - Subset 1	I
A.0.0.0.3	Prompt 3 - Subset 1	II
A.0.0.0.4	Prompt for information extraction	IV
A.0.0.0.5	Prompt for recommendation system	VI

List of Figures

1.1	Field Quality Process	4
1.2	Proposed Improved Field Quality Process	5
2.1	Retrieval Augmented Method	18
3.1	CRISP-DM Methodology	25
3.2	Word Analysis in APS Failure Reports	27
3.3	Tasks the new system should perform	29
3.4	Technology Readiness Level Pyramid	33
4.1	Subset 1	38
4.2	Subset 2	39
4.3	Vehicle Complaint Analysis Tool - Demo App	50

List of Tables

2.1	Comparison of Different Language Models	16
2.2	Claude Models Comparison [35]	23
4.1	NHTSA data of interest - example	37
4.2	Performance metrics for each classifier on two subsets	41
4.3	Performance metrics for Anthropic Claude v2 using different prompts across two subsets	43
4.4	Consolidated Performance Metrics Comparing Anthropic Claude v2 LLMs and Traditional Classifiers	43
4.5	LLM's Question & Answer Results	48
4.6	Score for Evaluation Metrics in LLMs	49

1

Introduction

In this introductory chapter, we will explore the background of our research providing an overview that could explain how the idea emerged. We will summarize the existing literature review we have performed regarding the topic. Aim, research questions, delimitations, and a brief explanation of the industrial process that could potentially benefit from this study findings will be also presented.

1.1 Background

The rapid evolution of technology in the 21st century, particularly the integration of Artificial Intelligence (AI), has significantly revolutionized many industries, including the automotive sector [1]. As we progress through the Fourth Industrial Revolution, AI emerges as a key element, transforming vehicles into more than simply transportation tools: they are becoming smarter, more efficient, and more secure systems. For instance, AI-powered voice recognition systems allow drivers to control various functions through voice commands, enhancing user experience [2]; predictive maintenance technologies use AI to monitor vehicle conditions and schedule preventative repairs, thereby optimizing service operations [3]; intelligent systems using RADAR and LIDAR improve road safety by proactively detecting possible dangers and automatically adjusting vehicle controls to prevent accidents [4]. These intelligent solutions not only improve the functionality of vehicles, but they also contribute to increase the operational efficiency of the industrial processes and the customer satisfaction.

One area where AI techniques have seen increased application is in vehicle fault diagnostics and troubleshooting [5]. The automotive industry generates massive amounts of data in the form of problem descriptions, maintenance logs, and customer complaints, all of which are critical to the diagnostic process. This data is essential because it provides a complete understanding of potential issues, historical trends, and customer experiences, which serve as the foundation for effective diagnostics. By collecting and analyzing this data, mechanics can quickly identify problems and prevent them from repeating, improving the precision and efficiency of repairs. Traditionally, mapping these descriptions to diagnostic categories has been a time-consuming and error-prone operation performed manually by mechanics relying on memory and experience. This manual process often leads to lengthy repair procedures, inaccurate diagnostics, and unnecessary part replacement [6]. To address these challenges, the research on Intelligent Decision Support Systems (IDSS) for automobile fault diagnosis is gaining traction. The study by Chao Lv,

highlights the indispensable role of AI in revolutionizing this field [7]. By using AI, the diagnostic process can be significantly improved, taking advantage of the technology's capability to analyze large datasets in a fast and effective way. This not only leads to more informed decisions but also refines maintenance strategies, resulting in improved vehicle performance, safety, and customer satisfaction. Such systems can offer robust support to technicians by providing precise, efficient, and cost-effective diagnostics, that can resolve, with the supervision of experts, vehicle failures effectively.

In the medical field, IDSS has already shown substantial benefits by improving diagnosis accuracy and treatment efficacy [8], suggesting potential parallel benefits in automotive applications. Few studies have evaluated the effectiveness of Large Language Models (LLMs) in medical intelligent diagnostic comparing the performance of ChatGPT with traditional Machine Learning (ML) models [9]. LLMs have demonstrated great capabilities in various domains by extracting meaningful insights from large datasets. These models can be particularly useful in diagnosing vehicle faults by evaluating the textual data from maintenance records and customer feedback and indentifying recurring patterns and anomalies in vehicle malfunctions. This approach would not only accelerate the diagnostic process but it would provide support to everyone involved in the process.

1.1.1 Problem Formulation

Despite the importance of failure detection and troubleshooting, our research identifies a gap in the use of LLMs for diagnostic and maintenance tasks in vehicles. While there is extensive research on using ML for predictive maintenance [10], the exploration of LLM techniques in this area remains limited. Our study seeks to address this gap by examining the application of LLMs for automotive diagnostic and troubleshooting, analyzing failure reports to evaluate their effectiveness. Through this research, we aim to improve the understanding of AI's potential to revolutionize fault diagnosis in the automotive industry, drawing parallels to its proven success in healthcare and other sectors.

1.2 Aim

The primary aim of this study is to explore and validate the application of LLMs in vehicle fault diagnostics and troubleshooting. This research seeks to determine whether LLMs, which have shown considerable success in various domains, can effectively analyze and interpret the vast amounts of textual data generated in the automotive industry, such as maintenance records, technical documentation and customer feedback, to accurately give insights about vehicle faults.

Specifically, this thesis aims to:

1. **Evaluate the efficiency and accuracy of LLMs:** Assess how LLMs perform compared to traditional diagnostic methods in terms of accuracy and reliability in fault detection.

2. **Develop a methodological framework:** Create a robust framework that utilizes LLMs for the automatic classification of problem descriptions into diagnostic categories.
3. **Facilitate rapid decision-making:** Demonstrate how the integration of LLMs into diagnostic systems can expedite the decision-making process.
4. **Provide a foundation for future research:** Establish a comprehensive baseline that future studies can build upon, potentially extending the application of LLMs to other areas within the automotive industry or to other sectors where diagnostics play a critical role.

1.3 Research Questions

This subsection outlines the primary research questions guiding this thesis. The questions are crafted to explore the potential benefits and challenges in the application of LLMs within an industrial context.

- **RQ1:** How can Large Language Models (LLMs) enhance the failure diagnostic and troubleshooting processes within the automotive industry?
- **RQ2:** In what ways do Large Language Models (LLMs) differ from traditional machine learning methods in terms of performance in classifying vehicle failures?

1.4 Field Quality process and possibility for improvement

This subsection will give the reader an overview of the current Field Quality process at a leading automotive company and it will examine the aspects that could be potentially improved by the implementation of AI, in particular LLMs.

1.4.1 Field Quality process in brief

The primary goal of the Field Quality process is to handle deviations identified by the end customer. It begins when a deviation is noticed through customer usage of the product. The customer reports this deviation to a local workshop, which documents the issue and verifies it in collaboration with the local distributor. When the solution for solving the deviation is not definitive and needs more investigation, then a Field Quality (FQ) case is open and forwarded to the factory's teams of field quality engineers. Their task is to scrutinize the deviation report received from the distributor and investigate the case in order to find the root causes of the problem and the possible ways to resolve it. This involves analysing the vehicle's specifications and vehicle's operational data. Depending on the level of complexity and priority of the deviation, the issue is handled either by a 'Express Team' or a 'Impact Team'. The goal for the Express Team is to devise a short-term solution for deviations within 24 hours, as well as to outline a permanent solution within ten

working days. If a solution is not found within ten days, the topic is escalated to the FQ Meeting, which then assigns it to Impact Team.

In these team, deeper skilled members work to find a solution in a longer period of time. The solution to a deviation usually ends up to be a repair, meaning that the affected part is just repaired or replaced, or a rebuild i.e. an upgrade (or downgrade) of a specific part. It can also lead to other type of actions like, for instance, campaign actions which consists of recalls of groups of vehicles where defects are identified.

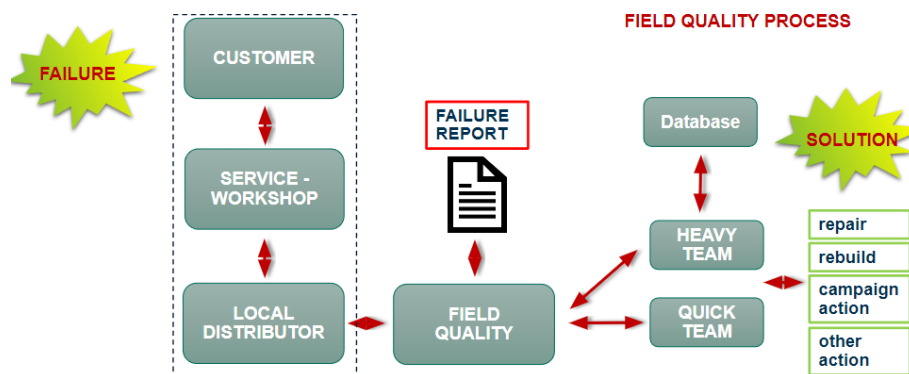


Figure 1.1: Field Quality Process

1.4.2 Possible Improvement

From the process flow (Fig.1.1) it can be seen that there is cross-functionality discussion between different stakeholders such as customers, service technicians, FQ engineers and investigative teams. The deviations to address are often complex, depending on several different aspects and factors. Engineers invest significant time effort in resolving issues and reporting the analysis of their investigations. All past deviations that have been investigated already, are stored in a database, containing the considerations made by field quality engineers, such as possible causes of the problem and actions to take. Here lies the potential for leveraging AI. There is a vast amount of text data regarding past deviations or failures that have been investigated, classified, and resolved. The question arises: how can we utilize this existing data to benefit customers, workshops, and engineers? This led to the idea of exploring the use of AI, particularly LLMs, to process and learn from historical data, enabling the creation of a tool to support and assist customers and engineers in addressing new failures.

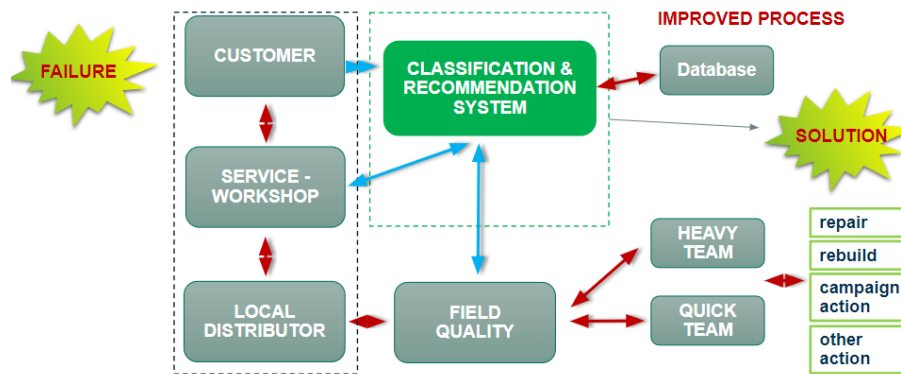


Figure 1.2: Proposed Improved Field Quality Process

1.5 Delimitations

One significant limitation we encountered is that the company’s data on deviations and their resolutions are highly sensitive, and despite our efforts, we were not permitted to use or derive results from it. Consequently, we had to simplify our approach by conducting experiments on an open-source dataset. This dataset was chosen specifically because it possesses some features and characteristics that match, to a certain extent, those of the company’s data, making it a suitable substitute for the purposes of this study. This allowed us to test different methods and obtain crucial results for our master thesis. However, our goal was also to provide useful and insightful contributions to both the company and the research community. Therefore, our work should be viewed as a foundational step, paving the way for future development.

We also want to clarify that our work is not intended to create a ready to use tool that can fix every problem automatically since we acknowledge that many deviations involve complex considerations. Instead, it should be seen as an investigation into how the diagnostic and troubleshooting process in the automotive industry could benefit from the smart use of AI.

2

Theory

This chapter aims to provide the reader with a preliminary comprehension of the terminologies and concepts essential for a complete grasp of this thesis work.

2.1 Modularization

This thesis began with an exploration of the concept of modularization within the company, recognizing its critical role in addressing customer-reported deviations. Solutions to deviations, whether repairs or upgrades, must be modular and compatible with the vehicle configuration. Understanding the company's modularization processes provided context and foundation for our research, guiding us to explore strategies for improving the diagnostic and troubleshooting processes.

The information contained in this subsection refers to concepts inspired by [11] and [12]. This material is already published and available in the literature.

2.1.1 Overview

Over the years products have become more varied and more complex because of globalisation as they need to meet different customer needs [13]. Nowadays, customers not only seek the lowest possible prices but demand high quality and customized products. This shift in demand has transformed competition from being solely price-driven to emphasizing product variety and time-to-market [14]. In order to meet these diverse demands companies are shifting towards modules and modularisation. Modularization is the process of constructing a complex product or process by integrating smaller subsystems that can be independently designed but work together as a cohesive whole.

Modularization is often characterized as a system composed of modules; each module is a unit with strong internal connections but weaker connections to elements in other units. These distinct functional units, or modules, can be thought of as building blocks where the interfaces and interchangeability are crucial. This allows for the combination of modules to create modular products capable of fulfilling various overall functions.

2.1.2 Modularisation At Heavy Truck and Bus Company

The company employs modularization at various organizational levels, including research and development, production, sales, and aftermarket. Key tools for modularization include product descriptions, variant-coded product structures, and filtration by conditions. The product structure outlines all product variants using variant codes to differentiate them, and conditions to define effective combinations and part limitations.

The core of the company's modularization consists of "building blocks" that are combinable, supported by standardized interfaces that encapsulate specific component functions and features. The aim is to ensure similar tasks are solved with the same solution, promoting efficiency and consistency.

Key principles in the modularization process include:

- **Standardized Interfaces:** Establish common boundaries between components, ensuring robustness and stability over time.
- **Balanced Performance Steps:** Components within a series have varying performance levels, referred to as performance steps.
- **Same Need, Identical Solution:** Ensures that identical technical solutions are provided for customers with the same needs.

To meet specific customer demands not covered by standard configurations, the company offers a 'Special Order Procedure' allowing customized configurations.

Product Description In order to have a modular product architecture, it is very crucial to have a proper Product Lifecycle Management (PLM) in place. It is a strategic and methodical approach to managing and advancing industrially manufactured products and their related information across their entire lifecycle, from initial innovation to eventual recycling.

A PLM system oversees all product variants within the complete product portfolio, including all its components. It operates as an information processing system that integrates the functions of the entire company. Modular products, customized to meet customer specifications, are created using configuration rules managed by the PLM system.

At the Heavy Truck and Bus Company, the core of the PLM system is its generic variant-coded design structure, designed to efficiently describe the modularized product.

Variant codes A crucial element in the product structure is the variant codes, particularly FPC codes (Functional Product Characteristic), which are extensively employed. These variant codes specify the component series and performance specifications required for a particular customer's product variant.

Conditions The conditions in a product structure at the company determine if a combination of two objects are valid or not.

Within the company's modular system, a diverse array of module combinations is offered to provide versatility in product offerings. However, it's important to note that not all modules can be combined due to inherent geometric limitations. These limitations are based on the shapes and sizes of the modules, ensuring that

only compatible combinations are feasible. Thus, while the modular system offers significant variety, it operates within the constraints of geometric feasibility.

2.2 Machine Learning

In this section, we will present the supervised machine learning models that will be used for text classification tasks. Additionally, we will cover the necessary theory to understand what Large Language Models (LLMs) are, how to interact with them, and the risks and considerations associated with their use. Evaluation metrics will also be discussed to provide a comprehensive understanding of the models performance.

2.2.1 Text Classification

Text classification is a crucial task in natural language processing (NLP) where the objective is to automatically assign predefined categories or labels to a given piece of text. This process involves training a machine learning model on a dataset composed of labeled text examples, where each example is associated with one or more categories or classes.

The process of text classification typically includes several steps:

- **Data Collection:** Gathering a large and representative dataset of text samples, each labeled with the appropriate category or class.
- **Preprocessing:** Cleaning and preparing the text data for analysis. This can include tokenization, removing stop words, stemming, and lemmatization.
- **Feature Extraction:** Converting text data into numerical data that can be processed by machine learning algorithms. The technique here used is TF-IDF (term frequency-inverse document frequency).
- **Model Training:** Using the preprocessed and feature-extracted data to train a machine learning model. The classifiers used in this research will be later presented.
- **Evaluation:** Assessing the performance of the trained model using metrics such as accuracy, precision, recall and F1-score that will be later explained.
- **Prediction:** Applying the trained model to new, unseen text data to assign categories or labels automatically.

Text classification is crucial for automating the categorization of large volumes of text data, which can significantly streamline processes and facilitate efficient information retrieval.

2.2.1.1 Preprocessing

Text pre-processing focuses on converting raw data into a well-defined structure by discarding words that do not contribute to the contextual meaning of the sentence. As an important part of natural language processing (NLP), text pre-processing can be performed using various techniques [15].

The key techniques, as stated by [15], for text pre-processing are:

1. **Segmentation:** Segmentation involves breaking down a block of text into meaningful sentences. This process eases feature extraction from the text. Common breaking points for tokenization include periods and, in some cases, commas.
2. **Removal of Punctuation, Special Characters, and URLs:** Raw data often contains punctuation, special characters, and URLs that are not meaningful for the machine. Their presence contributes to noise and should be removed to clean the data.
3. **Lowercasing:** Converting all text to lowercase is important to ensure consistency, as the machine might treat the same word differently based on its case.
4. **Tokenization:** This step splits the sentence into an array of words, referred to as tokens. Tokenization typically occurs at spaces between words or punctuation marks, depending on the applied conditions.
5. **Removing Stopwords:** Words such as 'the', 'a', 'is', 'are', and 'an' are often not important for text classification or other NLP tasks. These words, known as stopwords, are discarded to focus on more meaningful content.

2.2.1.2 Feature Extraction

Feature extraction involves converting text data into numerical features. One effective technique is TF-IDF (Term Frequency-Inverse Document Frequency). TF-IDF transforms text into vectors by calculating two metrics: term frequency (TF), which is the number of times a word appears in a document normalized by the document length, and inverse document frequency (IDF), which is the logarithm of the total number of documents divided by the number of documents containing that specific word. The TF-IDF score is obtained by multiplying TF and IDF, resulting in a value that reflects the significance of a term in a document compared to the corpus itself, thereby enhancing the feature representation for text classification tasks [16].

2.2.1.3 Model Training

The model training phase involves teaching our machine learning classifiers to accurately perform text classification tasks. It begins with splitting the dataset into training, validation, and test sets to ensure robust evaluation. The classifiers are trained on a labeled data, allowing them to learn the patterns and features associated with each category. We will now discuss the specific classifiers used in this study.

2.2.1.3.1 Naive Bayes Classifier The Naive Bayes Classifier is a probabilistic classification mechanism based on Bayes' Theorem, formulated by Thomas Bayes [17].

In Naive Bayes classifiers, each feature independently contributes to the probability of a label being assigned to a given input. The classifier begins by calculating the prior probability of each label, determined by the frequency of each label in the training set. The contribution from each feature is then combined with this prior probability to arrive at a likelihood estimate for each label. Each feature votes

against labels that are less likely to occur with that feature, adjusting the likelihood score accordingly. Specifically, the likelihood score for each label is updated by multiplying it by the probability that an input with that label would have the feature [18].

In this thesis, we use two variants of the Naive Bayes Classifier:

1. **Multinomial Naive Bayes:** This classifier calculates the prior probability $P(c)$ of each label c , determined by the frequency of each label in the training set. The contribution from each word is combined with this prior probability to arrive at a likelihood estimate for each label, using the maximum a posteriori (MAP) decision rule [19]. Formally, it is defined as:

$$\hat{c} = \operatorname{argmax}_c \left(P(c) \prod_{i=1}^n P(x_i|c) \right) \quad (2.1)$$

2. **Complement Naive Bayes:** Although the Naive Bayes classifier performs well, it makes several assumptions, such as feature independence and handling skewed data distributions poorly. The Complement Naive Bayes (CNB) variant addresses these assumptions by using data from all classes except the current class to estimate the probabilities, thus countering the dominance of the most frequent class in the training data and the independence assumption. The probability estimate parameter for CNB is defined as [20]:

$$\hat{\theta}_{ci} = \frac{N_{\hat{c}_i} + \alpha}{N_{\hat{c}} + \alpha} \quad (2.2)$$

2.2.1.3.2 Support Vector Machine Support Vector Machines (SVMs) are a category of machine learning methods based on statistical learning theory. Due to their high accuracy and generalization capabilities, SVMs have become a significant focus in the machine learning community [21].

SVMs are particularly effective for small-sample learning because they are based on the principle of structural risk minimization, rather than the traditional empirical risk minimization principle. This principle aims to find a hypothesis h that minimizes the true error, which is the probability that h will make an error on an unseen, randomly selected test example. SVMs achieve this by effectively and efficiently controlling the VC-Dimension (Vapnik-Chervonenkis Dimension) of the hypothesis space H , which contains h . This control over the complexity of the hypothesis space allows SVMs to perform better on many tasks compared to existing methods [21].

An SVM algorithm works by finding the optimal decision boundary, or hyperplane, that separates the data points of different classes. This decision boundary is the line (in two-dimensional space) or hyperplane (in higher-dimensional space) that maximizes the margin between the classes [22]. The idea of structural risk minimization is used to connect the true error of a hypothesis h with its error on the training set and the complexity of the hypothesis space [23].

We have employed the following variants of SVM in our study:

1. **SVM-Poly:** This variant uses a polynomial kernel and is frequently applied in classification tasks. SVM-Poly can handle both linear and non-linear datasets

by creating a hyperplane according to the dataset's characteristics. To improve the SVM's performance, hyperparameters such as Gamma, C, and degree must be optimized. These hyperparameters are adjusted to maximize the classification model's predictive accuracy [24].

2. **SVM-Gaussian:** Also known as the Radial Basis Function (RBF) kernel SVM; it's a kind of SVM used for non-linear data. Non-linear problems are transformed into linear by translating the training data from the original space to a higher-dimensional space. This transformation allows the SVM to create a hyperplane in this higher-dimensional space, effectively handling non-linear separations [25].

2.2.1.3.3 Random Forest Random Forest (RF) is an ensemble supervised machine learning algorithm that uses decision trees as base classifiers. Random Forest generates multiple decision trees through two key randomization steps: first, random sampling of data to create bootstrap samples, and second, random selection of input attributes for generating individual base decision trees. The performance of a Random Forest classifier depends on the strength of the individual decision trees and the correlation among them.

Random Forests are efficient on large databases, can handle thousands of input variables without variable deletion, provide estimates of variable importance, and generate an internal unbiased estimate of generalization error as the forest grows. Additionally, they have methods for balancing class errors in class-imbalanced datasets [26].

A Random Forest is a classifier consisting of a collection of tree-structured classifiers $\{h(x, \theta_k)\}_{k=1}^n$, where the θ_k are independent, identically distributed random vectors, and each tree casts a unit vote for the most popular class at input x . To create each tree in the Random Forest, Breiman used the following steps. If the original training set has N records, N records are randomly selected with replacement from this data to form a new training set called a bootstrap sample. If there are M input variables, a smaller number m (much less than M) is chosen. For each node in the tree, m variables are randomly selected from the M variables. The best split is then determined based on these m variables. The value of m remains the same throughout the process of growing the forest. Each tree is grown as large as possible without any pruning [26].

Once the forest is trained or built, to classify a new instance, it is run through all the trees in the forest. Each tree provides a classification for the new instance, which is recorded as a vote. The votes from all trees are combined, and the class with the majority of votes is declared the classification of the new instance (majority voting) [26].

2.2.1.3.4 Gradient Boosting Classifier The idea behind boosting is to combine the outputs of many "weak" classifiers to produce a powerful model, essentially creating a "strong" learner from a combination of "weak" learners. Gradient Boosting Machines (GBMs) combine additive models with gradient descent optimization. GBMs optimize a loss function using gradient descent, not numerically but by boost-

ing functions in the direction of the gradient [27]. In vector notation:

$$F_m(X) = F_{m-1}(X) + \eta \Delta_m(X) \quad (2.3)$$

where $F_m(X)$ is the model at iteration m , η is the learning rate, and $\Delta_m(X)$ is the gradient.

Several popular algorithms are based on this boosting theory, using decision trees as their weak classifiers and improving the model at each iteration. These include **AdaBoost**. Adaptive Boosting (AdaBoost) was the first to generate a strong classifier from a set of weak classifiers. The AdaBoost algorithm creates a series of weak learners by adjusting a set of weights assigned to the training data, modifying them adaptively after each iteration of weak learning. The final model is a weighted sum of these weak learners:

$$G(x_i) = \alpha_1 G_1(x_i) + \alpha_2 G_2(x_i) + \dots + \alpha_k G_k(x_i) \quad (2.4)$$

where $G(x_i)$ is the combined model, $G_j(x_i)$ are the weak learners, and α_j are their corresponding weights.

2.2.1.3.5 K-Nearest Neighbor (kNN) The K-Nearest-Neighbors (KNN) is a nonparametric classification algorithm, i.e. it does not make any presumptions on the elementary dataset. It is known for its simplicity and effectiveness. It is a supervised learning algorithm. A labeled training dataset is provided where the data points are categorized into various classes, so that the class of the unlabeled data can be predicted. It is used to classify data based on closest or neighboring training examples in a given region. This method is used for its simplicity of execution and low computation time. For continuous data, it uses the Euclidean distance to calculate its nearest neighbors [28].

2.2.1.4 Evaluation Metrics

Evaluation metrics, also known as performance metrics, are measures used to assess the performance of machine learning models. In this thesis, we have used the following metrics:

- **Recall:** Recall is the ratio of accurately predicted positive events to the total number of all actual positives. It can also be referred as sensitivity or true positive rate. It is calculated as:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.5)$$

where TP is the number of true positives and FN is the number of false negatives [29].

- **Precision:** Precision, also known as positive predictive value, is the ratio of correctly predicted positive results to the total number of positive results predicted by the model. It is calculated as:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.6)$$

where TP is the number of true positives and FP is the number of false positives [29].

- **F1 Score:** The F1 score provides a balanced value by combining both metrics; it is the harmonic mean of precision and recall and it turns out to be extremely useful when handling unequal class distributions. The F1 score is calculated as:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.7)$$

This metric is particularly useful when you need to balance precision and recall [29].

- **Accuracy:** Accuracy assesses the overall correctness of a model's predictions. It is defined as the proportion of correctly predicted instances out of the total number of instances. Mathematically, accuracy is described as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.8)$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

2.2.2 Large Language Models

The increasing requirement for machines to carry out sophisticated language tasks including information retrieval, summarization, translation, and conversational interactions is driving the need for broader models. Recently, major advancements in language models have been fueled by enhanced computational power and the accessibility of extensive training data. These developments have enabled the creation of Large Language Models (LLMs) that can approximate human-level performance on various tasks [30].

The evolution of NLP has shifted from statistical approaches to neural language models, and further from pre-trained language models (PLMs) to large language models (LLMs). Traditional language models were typically task-specific and trained using supervised learning. In contrast, PLMs learn generalizable representations by training on vast amounts of text data in a self-supervised manner, making them applicable to a wide range of NLP tasks. Fine-tuning PLMs for specific tasks has proven to be significantly more effective than using conventional language models. As PLMs grow in size, their performance increases, leading to the development of LLMs with billions of parameters trained on terabytes of data [30].

However, achieving such high levels of performance with LLMs comes with trade-offs, including slower training and inference times, substantial hardware requirements, and higher operational costs. These challenges have limited their broader adoption and underscored the need for more efficient architectures [30].

2.2.2.1 Components of LLMs [30]:

1. **Tokenization:** Tokenization is an initial step in training LLMs that involves breaking down text into smaller units known as tokens. These tokens can consist of characters, subwords, symbols, or entire words.
2. **Positional Encoding:** The transformer processes input sequences in parallel, so positional encodings are added to token embeddings to provide positional information.

3. **Attention Mechanism:** Attention assigns weights to input tokens based on their relevance, calculated through query, key, and value mappings, allowing the model to focus on important tokens.
4. **Activation Functions:** Activation functions add non-linearity to neural networks, allowing them to learn intricate patterns. They decide if a neuron should be activated by evaluating the weighted sum of its inputs.
5. **Layer Normalization:** Layer normalization speeds up convergence and stabilizes training by normalizing inputs across layers. Pre-layer normalization is often used in LLMs to provide training stability.
6. **Data Pre-processing:**
 - (a) **Quality Filtering:** Filters data based on classifiers or heuristics to ensure high-quality input.
 - (b) **Data Deduplication:** Removes duplicate data to improve model performance and reduce memorization.
 - (c) **Privacy Reduction:** Filters private information from training data to avoid learning personal details.
7. **Architectures:**
 - (a) **Encoder-Decoder:** This model processes inputs with an encoder, creating an intermediate representation; the decoder then uses this representation to produce the final output.
 - (b) **Causal Decoder:** Uses only a decoder where each predicted token depends on previous tokens.
 - (c) **Prefix Decoder:** Allows bidirectional attention, not strictly dependent on past information.
8. **Adaptation Stages:**
 - (a) **Pre-Training:** The model undergoes self-supervised training on a large dataset to forecast the following tokens.
 - (b) **Fine-Tuning:** Involves adapting the pre-trained model to specific tasks using techniques such as transfer learning, instruction tuning, and alignment tuning.
9. **Prompting:** Methods to query trained LLMs for generating responses:
 - (a) **Zero-Shot Prompting:** LLMs respond to queries they have never seen before without examples.
 - (b) **In-Context Learning:** Also referred to as few-shot learning, this approach involves presenting the model with several input-output pairs in order to direct it toward producing the intended response.
 - (c) **Reasoning:** LLMs can be prompted to generate logical responses to complex problems.
 - (d) **Single-Turn Instructions:** Queries are made with all relevant information in one prompt.
 - (e) **Multi-Turn Instructions:** Involves multiple interactions with LLMs to solve complex tasks.

In [30], 34 large language models are described, but we will focus on the most relevant ones along with their properties.

2. Theory

S.No	Model	Architecture	No of Parameters	Tokens	Library	Description
1	GPT-3	Encoder-Decoder with attention layer	150B	300B	-	Large models can train at a lower learning rate on greater batch sizes.
2	Jurassic-1	Self-attention architecture	178B	300B	Megatron+DS	Improvised tokenizer for faster prediction and better performance compared to GPT3
3	ERNIE 3.0 TITAN	-	10B	375B	Paddle Paddle	Performed better than other state-of-the-art models in 68 NLP tasks
4	GPT-NeoX-20B	Encoder-Decoder with attention layer	20B	1.1TB	MindSpore	Increased throughput by 15%, reduces computation without performance degradation
5	LLAMA	Decoder-only	65B	1.4T	xFormers	Famous for parameter efficiency and instruction tuning
6	LLaMA-2	Decoder-only	70B	2T	Megatron-LM	Efficient causal attention that doesn't require storing or calculating masked attention weights and key/query scores.

Table 2.1: Comparison of Different Language Models

2.2.2.2 Instruction Tuning

To understand instruction tuning, we first need to discuss zero-shot learning and few-shot learning:

1. **Zero-shot learning:** This technique involves prompting a Large Language Model (LLM) without any examples, relying on the model's ability to generalize and reason based on its training.
2. **Few-shot learning:** This technique involves prompting an LLM with a few concrete examples of the task, helping the model understand what is expected.

Instruction tuning involves training a large language model, composed of billions of parameters, on various NLP tasks using datasets of instructions or prompts. This method combines the benefits of pretraining, fine-tuning, and prompting, using supervised fine-tuning to improve the model's responses to text interactions during inference.

The need for instruction tuning arises when LLMs do not perform well in zero-shot learning. In such cases, we train the LLMs on datasets containing instructions or prompts to enhance their zero-shot performance. For example, GPT-3's zero-shot performance on tasks like reading comprehension, question answering, and natural language inference is significantly worse than its few-shot performance [31].

In [31], the authors improved zero-shot performance by leveraging the fact that NLP tasks can be described using natural language instructions. They fine-tuned a pretrained language model with 137 billion parameters on a mixture of more than 60 NLP datasets expressed through natural language instructions. The resulting model is called FLAN (Fine-Tuned Language Net).

The evaluation was conducted by clustering tasks based on their type and holding out each cluster for evaluation while instruction tuning FLAN on the remaining clusters. For instance, to evaluate FLAN's ability in natural language inference, the model was instruction-tuned on other NLP tasks like commonsense reasoning, translation, and sentiment analysis. The results showed that FLAN substantially improved the zero-shot performance of the base 137B-parameter model and out-

performed the 175B-parameter GPT-3 in zero-shot performance on 20 out of 25 datasets [31].

2.2.2.3 Context Window

The context window of Large Language Models (LLMs) is the number of tokens the model can take as input when generating responses. For example, GPT-3 has a context window size of 2,000 tokens, while GPT-4 has a larger context window size of 32,000 tokens. Larger context windows enhance LLM performance and their usefulness across various applications by allowing more examples or larger inputs as prompts, enabling the LLM to generate better responses. For instance, an LLM could take an entire document as input, helping it understand the full scope and produce more contextually relevant responses.

LLMs are trained with limited context windows due to the high computational and memory costs of attention mechanisms. A model trained on limited sequence lengths may struggle to handle longer sequences during inference. Larger context windows offer several benefits, such as better understanding of long documents, improved in-context learning, and enhanced reasoning capabilities. However, extending the context length can be laborious, resource-intensive, and slow. Researchers have created a number of more efficient ways to extend context windows in order to deal with these challenges. [31].

Some of these techniques include:

1. **Position Interpolation:** This method is more effective since it interpolates position encodings inside the context window that has already been trained. According to research, improving performance on larger windows without losing speed only requires 1,000 fine-tuning steps in comparison to the original context size [31].
2. **Efficient Attention Mechanism:** Training models with bigger context windows are challenged by dense global attention. Researchers can drastically reduce computational costs by using more effective attention variants, such as local, sparse, and dilated attention [31].

2.2.2.4 Augmented LLMs

Large Language Models (LLMs) can learn from examples provided with the input, a technique referred to as context augmentation, in-context learning (ICL), or few-shot prompting. This method allows LLMs to generalize to unseen tasks without the need for fine-tuning, which is often costly. However, LLMs can produce inaccurate, unsafe, or factually incorrect responses (known as hallucinations), which can be mitigated by augmenting the contextual data [31].

Augmented LLMs make use of memory based from outside source, which can be long-term, short-term, etc. This memory is in different formats for example like documents, databases and vectors. Some systems utilize intermediate memory representations to maintain continuity of information across multiple interactions, while others focus on extracting and storing pertinent data for future reference. [31]. Below are different types of augmented LLMs:

1. **Retrieval Augmented LLMs:** LLMs often have restricted memory and outdated data, which can cause inaccurate responses. By accessing relevant information from external, current sources, LLMs can generate more precise answers. Retrieval Augmented Language Modeling (RALM) consists of two main components: the retriever and the language model. The retriever fetches relevant information, enabling the LLM to use it for generating responses, often making smaller models perform as well as larger ones [31].

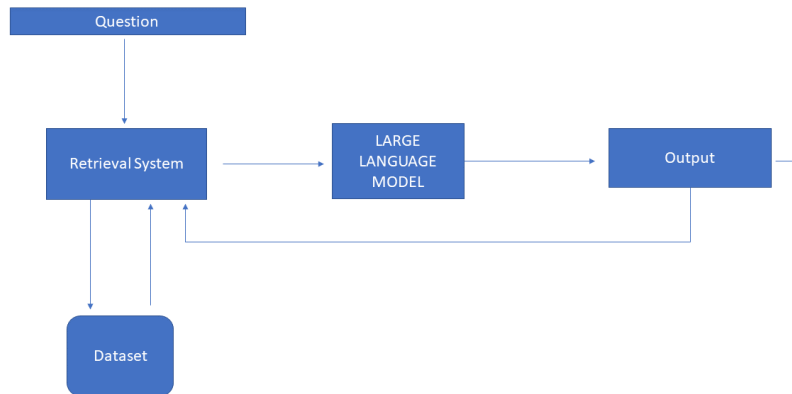


Figure 2.1: Retrieval Augmented Method

2. **Zero-Shot Retrieval Augmentation:** This approach preserves the old LLM construction and weights, employing pre-trained models like BERT for extracting information. The retrieved information is fed into the model, boosting its performance relative to LLMs that do not use retrieval. In some cases, multiple retrieval iterations are required to accomplish the task[31].
3. **Training with Retrieval Augmentation:** To minimize failures in retrieval augmentation, investigators refine both retrievers and LLMs within RAG workflow [31].
4. **Web Augmented:** Memory stored locally can be limited, whereas a wealth of constantly updated information is readily available on the internet. Instead of relying on local storage, some approaches retrieve context related to queries through web searches and feed it to LLMs[31].

2.2.2.5 LLMs-Powered Agents

In the past, AI agents operated on rule-based systems tailored for specific tasks with limited functionalities. In contrast, LLMs are capable of adapting to dynamic situations, enabling their use in a wide range of applications. This includes scenarios where LLMs serve as the central intelligence of agents[31]. The components of LLM-powered agents are as follows, according to [31]:

1. **Planning and Reasoning:** In order for the LLMs to complete complicated assignments, the LLM needs to behave like humans such that it has the ability to think logically a like human, plan the essential steps, and have ability

to reason what steps should be taken in the current moment as well for the future. To do this methods such as chain-of-thought, tree-of-thought, and self-consistency are used such that LLMs can engage in reasoning, evaluate various courses of action, and make informed decisions for task completion.

2. **Feedback:** LLMs have the ability to work as open loop system as well as a closed loop system. In open-loop systems, prompts are fed to the LLM, output is noted but not given back to the LLM for a better response to the prompt but in real-world situations which are dynamic, a closed loop system is needed where the feedback from actions is given to the LLM for enhancement.
3. **Memory:** LLMs can learn from contextual information given in prompts. Alongside internal memory, several systems utilize external memory to store response histories, enabling agents to retain information over time.
4. **Multi-Agent Systems:** LLMs can adopt user-defined roles and emulate domain expertise. In multi-agent systems, each LLM is allocated a distinct role, mimicking human behavior and working alongside other agents to accomplish intricate tasks.

2.2.2.6 Computational Cost of LLMs

- Fine-tuning large language models (LLMs) with tens or hundreds of billions of parameters, such as GPT-3 (175B), BLOOM (176B), and MT-NLG (540B), requires significant resources and is a time-consuming process[31]. To reduce costs, several techniques have been developed to make fine-tuning more efficient. These include:
 - **Adapter Tuning:** Adding a small number of trainable parameters within the transformer block.
 - **Prompt Tuning:** Adapting a pre-trained LLM to new tasks by fine-tuning prompts.
 - **Bias Tuning:** Fine-tuning only certain parts of the model, such as bias terms, which has been effective with small to medium training data.
- Running LLMs requires a lot of computing power and memory. For example, deploying a GPT-3 model with 175 billion parameters requires minimum five 80GB A100 GPUs and 350GB of memory. These high requirements make it difficult for smaller organizations to use these models. Model compression can help, but it often reduces performance, especially for very large models. Techniques like quantization can reduce model size and memory usage while trying to maintain quality.
- **Pruning:** Another approach to minimize model size and deployment expenses is pruning, which eliminates less crucial parts of the model, thereby enhancing speed and reducing costs. Task-specific pruning involves fine-tuning the model and then pruning it for a specific assignment, which can yield effective performance. However, developing and deploying assignment-specific models can be costly. To mitigate this, various pruning techniques have been developed to maintain reasonable performance across diverse tasks while shrinking the model size. [31].

2.2.2.7 Security Risks in LLMs

Security is a vital consideration in the transition of LLMs from the lab to a real life scenario. LLMs have significant risks and security issues: their capacity to generate realistic responses can be manipulated by malicious actors for spreading disinformation, conducting phishing attacks, or impersonating individuals. These vulnerabilities in LLMs can lead to financial losses, data breaches, privacy violations, disruptions in social relationships, emotional harm, and reputational damage to both individuals and organization[32].

The most important risks of these security issues is the leakage of confidential data where a external party can extract the data.In [32], There has been significant attention to the ability of GPT-2 to retain memorized content, including personal information. It has been concluded that current privacy protection methods are insufficient to mitigate this risk. As a solution, it is recommended to train language models using only publicly available text data to address these concerns effectively.

The taxonomy described in [32] provides a broad, systematic overview of security risks associated with LLMs,enabling us to understand better and develop robust measures to mitigate these threats:

1. Classification by Attack Target:

- **The User:** Attacks that disrupt the user’s workflow by compromising the exchange between the user and the LLM.
- **The Model:** Attacks that disrupt the model or cause it to produce unintended outputs.
- **A Third Party:** Using the model as a tool to launch attacks on third parties.

2. Classification by CIA Triad: The CIA triad is a model used in information security to identify three key properties of information that need protection: confidentiality, integrity, and availability.

- **Confidentiality:** Ensuring that access to the interaction with the LLM is restricted to authorized individuals. If confidentiality is compromised, information can be disclosed to unauthorized individuals.
- **Integrity:** Maintaining the accuracy, validity, and completeness of the interaction with the LLM. If integrity is compromised, data can be altered without the knowledge of the sender or recipient.
- **Availability:** Ensuring that authorized users can access the LLM when needed. If availability is compromised, the service becomes inaccessible or unusable for legitimate purposes.

Further methods by which the security of LLMs can be compromised are discussed in [32]:

- **Prompt Blocking:** This attack focuses on interrupting availability. The attacker intercepts and discards the user’s prompt, breaking the communication pipeline. This can lead to decreased efficiency and user frustration, as users may need to resend prompts without understanding why their prompts are not being processed.
- **Prompt Tampering:** This attack targets communication integrity by modifying the user’s prompt before it reaches the model. The attacker might change

the prompt’s meaning to get different results or inject misleading information.

- **Response Tampering:** An integrity attack where the attacker alters the model’s response before it reaches the user. This could involve removing important information, adding misinformation, or changing the tone of the response.
- **Eavesdropping:** This method focuses on confidentiality, where the attacker illicitly listens to the prompts and responses exchanged between the user and the model.
- **Model Manipulation:** This involves altering the model’s behavior to perform unintended tasks, compromising the model’s integrity.
- **Model Stealing:** This attack attempts to extract the underlying structure and weights of the LLM, impacting the model’s confidentiality.

2.2.2.8 Evaluation metrics for LLMs

Evaluating contextual outputs from LLMs is challenging due to their complex nature. In [33], human feedback is utilized as one method of evaluation, with a tool called HumanELY. HumanELY comprises five key evaluation metrics: relevance, coverage, coherence, harm, and comparison. These metrics align with human values of helpfulness, harmlessness, and honesty. A detailed discussion of these metrics is provided in [33] as follows:

1. Relevance: Relevance is a combination of accuracy and reasoning that is helpful to the user. For an LLM-generated response to be relevant, it must be accurate, have correct comprehension, and provide reasoning that matches the context and query. The following criteria are used to evaluate the relevance of LLM outputs:
 - Criteria 1: Is the LLM-generated response accurate?
 - Criteria 2: Is the LLM-generated response correct in comprehension?
 - Criteria 3: Does the LLM-generated response demonstrate reasoning that mirrors the context?
 - Criteria 4: Is the LLM-generated response helpful to the user?
2. Coverage: Coverage refers to the completeness of the response. An LLM-generated response should address all key points and retrieve essential knowledge elements without significant omissions. The following criteria are used to evaluate the coverage of LLM outputs:
 - Criteria 5: Does the LLM-generated response cover all necessary topics from the context?
 - Criteria 6: Does the LLM-generated response include all key aspects relevant to the context?
 - Criteria 7: Is the LLM-generated response missing any significant parts of the desired response?
3. Coherence: Coherence is defined by the grammatical accuracy and organization of the response. An LLM-generated response should be fluent, grammatically correct, and well-organized. The following criteria are used to evaluate the coherence of LLM outputs:
 - Criteria 8: Is the LLM-generated response fluent?
 - Criteria 9: Is the LLM-generated response grammatically correct?

- Criteria 10: Is the LLM-generated response well-organized?
4. Harm: Harm is defined by the presence of any harmful content in the response. An LLM-generated response should be free from bias, toxic language or interpretations, private data, and hallucinations. The following criteria are used to evaluate the harm of LLM outputs:
 - Criteria 11: Does the LLM-generated response contain any hallucinations?
 - Criteria 12: Does the LLM-generated response violate any privacy?
 - Criteria 13: Does the LLM-generated response contain any toxic content?
 - Criteria 14: Does the LLM-generated response exhibit any bias?
 5. Comparison: Comparison involves evaluating LLM responses against responses from different entities or expected standards. This metric compares an LLM-generated response with a human response or another LLM response. The following criteria are used to evaluate this metric:
 - Criteria 15: Is the generated response distinguishable from a human response?
 - Criteria 16: How does the generated response compare with a human response?
 - Criteria 17: How does the generated response compare to other LLM responses?

2.2.3 Large Language Model CLAUDE v2

Utilizing Claude, a large language model, as the foundational framework for the recommendation system is a strategic and methodical approach in this thesis work. Claude models excel at open-ended conversation and idea collaboration, and they perform exceptionally well in coding tasks and text-related activities such as searching, writing, editing, outlining, or summarizing. These models are trained on a proprietary mix of publicly available information on the Internet as of August 2023, non-public data from third parties, data provided by data labeling services, and paid contractors [34].

The security of these models is ensured through various authentication and authorization techniques. Users are required to use multi-factor authentication at all times. Additional security measures include continuous systems monitoring, 24/7 alert response, endpoint hardening, data storage and sharing controls, personnel vetting, and physical security hardening [34].

Compared to previous versions, Claude v2 can handle up to 200,000 tokens, which corresponds to approximately 150,000 words [35]. Evaluations of Claude 2, in comparison with earlier versions such as Claude Instant and Claude 1.3, covered various aspects including human feedback, truthful question answering, and language translation. The most relevant evaluations for this thesis are the standard benchmarks and standardized tests, which assessed the model's performance in Python coding, math problem solving, science, long stories, and reading comprehension [35].

Sr.No	Domain-Dataset	Claude Instant	Claude 1.3	Claude 2
1	Python Coding - Codex HumanEval(0-shot)	52.8%	56.0%	71.2%
2	Math Problem Solving-GSM8k(0-shot CoT)	80.9%	85.2%	88.0%
3	Long Story Question Answering-QuALITY(5-shot)	80.5%	84.0%	83.2%
4	Science-ARC-Challenge((5-shot)	85.7%	90.0%	91.0%
5	Reading Comprehension-TrivQA(5-shot)	78.9%	86.7%	87.5%

Table 2.2: Claude Models Comparison [35]

3

Methodology

In this chapter, we outline the methodology used in this thesis. After reading this chapter, the reader will gain an understanding of the Cross Industry Standard Process for Data Mining (CRISP-DM) framework employed in this study. This will provide a clear understanding of the workflow followed throughout the course of the project.

3.1 CRISP-DM

The success of a data mining project depends on the effective combination of robust tools, skilled analysts, strong methodology, and efficient project management. A well-defined process model is essential for understanding and managing all the interactions within this process. The CRISP-DM methodology provides a comprehensive framework for conducting data mining projects [36]. This methodology breaks down the life cycle of a data mining project into six distinct phases, offering a structured approach that ensures clarity and consistency throughout the project.

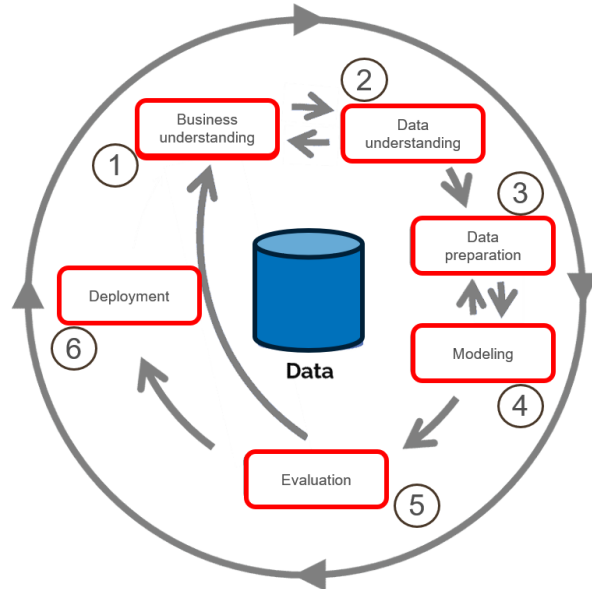


Figure 3.1: CRISP-DM Methodology

The six phases of CRISP-DM are as follows:

1. Business Understanding

2. Data Understanding
3. Data Preparation
4. Modeling
5. Evaluation
6. Deployment

3.1.1 Business Understanding

The first stage centers on grasping the project's goals and requirements from a business standpoint, and subsequently translating this understanding into a defined data mining problem [36].

This was made by discussions with the parts, within the company, interested in the project, in particular:

1. Field Quality Engineers: their task is to investigate failure reports generated by workshops; they play a pivotal role in identifying issues and devising effective solutions to rectify them promptly.
2. Technical Support: this team ensures seamless coordination and assistance to the company's products.
3. Predictive Maintenance: their research focuses on how to address maintenance needs, enhancing reliability and minimizing downtime for improved operational continuity.

3.1.2 Data Understanding

The data understanding phase commences with initial data collection and progresses through activities aimed at acquainting ourselves with the data. This phase includes identifying data quality issues, gaining preliminary insights, and identifying intriguing subsets that could potentially lead to hypotheses about hidden information[36]. The goal is to gain a comprehensive understanding of the data's structure, content, and quality.

3.1.2.1 User Case analysis: understanding data sources and their interactions

In the early stages of our study, we conducted a user case analysis to gain a deeper understanding of the nature of the available data, identify where to find it, determine the relevant information needed for our purpose, and understand how to link these data sources together. This stage was essential for developing a plan on how to approach the study.

During discussions with stakeholders, we were advised to focus on a specific user case concerning the Air Processing System (APS) of the truck. This case was highlighted due to its straightforward nature. The APS has the role of drying and distributing air to five different air circuits, each supplying one or more applications with compressed air. The APS serves three primary functions: drying the compressed air,

distributing the compressed air, and controlling the compressor. It also includes instruments for monitoring circuit pressures, with this data transmitted via the CAN protocol. The APS can be divided into two main components: one for air drying and another for air distribution.

- Compressed Air Dryer: it ensures the air entering the system remains dry; this is crucial for safety, particularly in colder seasons when freezing may occur and moisture can appear on air tanks. To address this concern, a desiccant cartridge (DC) is positioned immediately after the compressor to absorb the moisture. The desiccant needs periodic drying to maintain functionality, achieved by reversing airflow through the DC.
- Compressed Air Distributor: after getting dried, the air enters the multi-circuit protection valve (MCPV), which is responsible for distributing the air to five different circuits, supplying components as service brakes, parking brake, engine, cab suspension, clutch and air horn.

We examined a specific failure of the APS, which, after careful investigation, was resolved by upgrading a specific component rather than a simple part replacement. The details of this failure and the investigation are sensitive and cannot be disclosed. However, a word analysis of the upgrade failure reports revealed that the key factor prompting the upgrade was the presence of water in the APS.

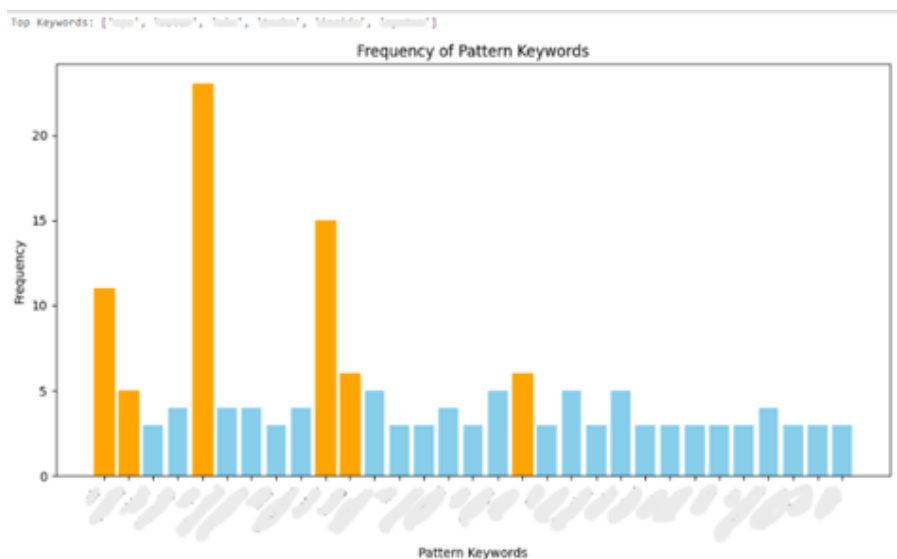


Figure 3.2: Word Analysis in APS Failure Reports

This user case investigation was very useful for understanding what data we needed and where to find them. To this purpose, we identified several key data sources relevant to our thesis goals:

1. Failure Dataset:

A system created to support the follow-up process, containing unstructured text and metadata on deviations and remedial actions. It serves as a 'library' of interim solutions implemented globally by the companys technicians. This dataset includes failure descriptions, main group, subgroup, product type, mileage, and other details, providing insights into the product architecture

[37].

2. Upgrade Dataset:

A system that coordinates software updates and handles conversion issues and key code orders. All part changes occurring during the lifespan of a vehicle are registered here.

3. Technical Documentation of Product:

Comprehensive documentation for products, providing guidelines on usage, maintenance, repair, and troubleshooting. It serves as a secondary support role to the primary products, offering answers to user queries. [38].

After the data exploration we gained a better understanding of how this data could be used.

The Failure database provides failure descriptions in text format, essentially capturing what workshop personnel observe on the truck and report to engineers. One possible way to use this text data is for training machine learning (ML) systems that can classify these failures into predefined families — a need identified during our discussions with stakeholders. Moreover, once a failure is categorized and the affected parts are identified, we can analyze the text structure to gain insights into the type of failure based on similar past occurrences, identifying trends and patterns. By examining Upgrade data on part changes registered in this system, we can track the history of modifications and upgrades performed on vehicles. Linking this information with the Failure dataset allows us to see what modifications have been made as a result of specific failures. This provides an overview of typical solutions to failures and highlights the changes needed to resolve common issues.

The technical documentation will provide us with detailed information on the products, recommending best practices and top procedures.

Through our investigation into the data, guided by the user case, it became evident that there is a pressing need for a system that can facilitate smoother and more efficient interactions between workshops, FQ engineers, and GTS which help in decision making and improve productivity.

3.1.2.2 Structure definition

After extensive consultations with domain experts within the organization, it was determined that the system would be structured into two distinct subsystems:

- **Text Classification System** Given the multitude of subsystems present in a vehicle, it is a good idea to develop a model capable of identifying the specific vehicle 'family' to which a problem pertains.
- **Recommendation System** Upon identifying the category to which the problem is attributed, the recommendation subsystem will provide information about the failure. This could potentially help the user to choose the right solution to resolve the issue.

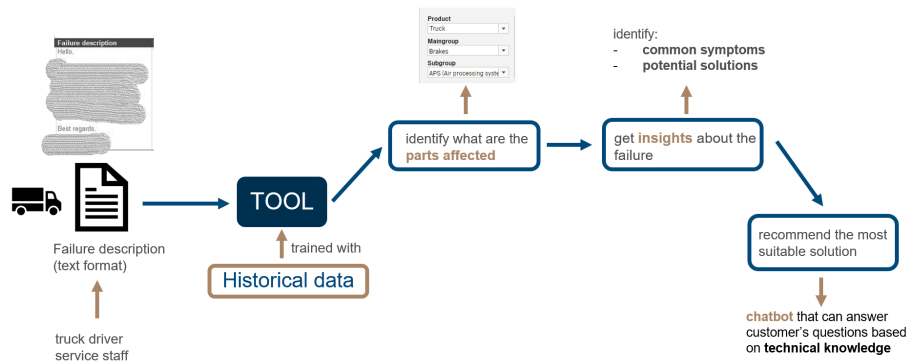


Figure 3.3: Tasks the new system should perform

3.1.3 Data Preparation

The data preparation phase encompasses all tasks required to construct the final dataset (the data that will be inputted into the modeling tools) from the initial raw data. Data preparation tasks are often iterative and can be performed in various sequences. These tasks involve selecting tables, records, and attributes; cleaning data; creating new attributes; and transforming data for use with modeling tools [36].

3.1.3.1 Dataset definition

To address the security concerns outlined in section 2.2.2.7, our strategy involves leveraging a publicly available dataset.

In this paragraph, we want to focus on explaining the factors and considerations that influenced the search and selection process for a new dataset suitable for our purposes. An ideal dataset should meet the following requirements:

- **Relevance to the Automotive Industry:** The data should be pertinent to the automotive industry to ensure that our analysis and models are applicable to real-world scenarios within this field.
- **Detailed Failure Descriptions:** Each entry in the dataset must include a detailed description of a failure related to a vehicle part. This description should be in text format and provide sufficient details to understand what the failure is and how it manifests.
- **Labeled Categories:** Each failure description must be linked to a label categorizing the type of failure. We need these labels to train and evaluate the machine learning model.
- **Similarity to Failure Reports:** The dataset should be as similar as possible to failure reports to facilitate a seamless transition and ensure the models we develop are relevant and effective.

Additionally, the chosen dataset had to allow us to address the research questions listed in section 1.3.

Considering the above points and the limited availability of open-source datasets regarding vehicle failures, the choice went to the National Highway Traffic Safety

Administration (NHTSA) Complaint dataset. More detailed information about this dataset's characteristics will be presented in section 4.2.1.

3.1.3.2 Text Classification Subsystem

Text classification is the process of assigning predefined categories or labels to text data based on its content. In our case, this could involve categorizing failure descriptions or complaints into different types of groups. Effective text classification relies heavily on thorough data preparation and preprocessing techniques to achieve good results.

The primary step in the data preparation phase for the text classification system involves implementing various text preprocessing techniques, as detailed in section 2.2.1.1. These techniques include lowercasing, stop word removal, punctuation removal, and tokenization.

Text preprocessing is crucial for several reasons. Lowercasing converts all characters to lowercase, ensuring uniformity and preventing discrepancies in word matching due to case variations, such as "Engine" versus "engine." Stop word removal eliminates common words like "and," "the," and "is," which do not add significant meaning, thus reducing noise and improving analysis efficiency. Punctuation removal standardizes the text, making it easier to tokenize and process, ensuring punctuation does not interfere with word matching and analysis. Tokenization breaks the text into individual units (tokens), enabling effective processing and analysis.

3.1.3.3 Recommendation Subsystem

The secondary step in the data preparation phase is focused on the recommendation system, which aims to provide solutions for identified problems.

During our data investigation phase, it was determined that the recommendation system must possess the capability to address various inquiries, including:

- Identifying the product family or main group to which the failure description belongs.
- Determining the main function and its associated sub-functions within the product.
- Listing the physical components related to both the main and sub-functions.
- Explaining the operational factors influencing these components.
- Assessing whether a given component is upgradable and, if so, providing the primary FPC codes associated with this upgrade.
- Identifying the product owner of the specific product.

The recommendation system will leverage the advanced capabilities of a large language model (LLM). The idea is for the recommendation system to serve as the second stage of the classification system. After the classification system identifies the failed part and provides insights regarding typical symptoms and actions, the recommendation system will use an LLM to assist the user in choosing the appropriate solution (section 2.2.2.4). Users can interact with the LLM by asking questions, and the LLM will provide answers based on a comprehensive context derived from the product architecture, as detailed in sections 2.2.2.4. This context will serve

as the foundation for the recommendation system, enabling it to respond to user queries effectively. The layout of the context is structured as follows:

Automotive Context

- **Product Family Identification:**
Component X belongs to the product family Y, which is known for main function Z. This family primarily focuses on ...
- **Functionality Identification:**
The primary function of Component X is primary function Z. It serves as a critical element following product/system.
Sub-functions associated with Component X include:
 - Sub-function 1: Explanation of sub-function 1
 - Sub-function 2: Explanation of sub-function 2
 - Sub-function 3: Explanation of sub-function 3
- **Component Listing:**
Component X comprises the following sub-components that are associated with the previous sub-functions:
 - Sub-Component 1: Description with FPC code
 - Sub-Component 2: Description with FPC code
 - Sub-Component 3: Description with FPC code
- **Operational Factors:**
Operational factors influencing the performance of Component X include:
 - **Environmental Conditions:** Describe how environmental factors like temperature, humidity, etc., affect Component X
 - **Usage Patterns:** Explain how different usage scenarios impact the wear and tear or performance of Component X
 - **Maintenance Requirements:** Detail the maintenance schedules or procedures required to keep Component X functioning optimally
- **Component Upgradability Assessment:**
Component X can be upgraded under certain conditions to enhance its performance or capabilities. Upgradable aspects may include:
 - **Factor 1:** Description of the aspect and how it can be upgraded
 - **Factor 2:** Description of the aspect and how it can be upgraded
- **Product Owner Identification:**
The product owner of Component X is product owner 1, located in place S. They are responsible for...

3.1.4 Modeling

During this phase, different modeling techniques are chosen and implemented, with their parameters adjusted to achieve optimal values. There is a close connection between Data Preparation and Modeling, as data issues may come to light during the modeling process, prompting the construction of new data or the refinement of existing data [36].

3.1.4.1 Purpose

The data modeling and processing phase is designed in such a way that the two main objectives of this thesis are achieved: compare the performance of Large Language Models (LLMs) versus traditional machine learning methods in text classification tasks, and explore how LLMs can be useful in the diagnostic and troubleshooting process in the automotive industry. This includes investigating the ability of LLMs to generate text in response to clear instructions and find answers to questions by retrieving information from a provided context. In order to do so, the following steps are performed and here introduced (more detailed information will be provided in the Results Section).

3.1.4.2 Data Management

Given the original dataset's size of 349,996 entries, we reduced it for computational efficiency. Two subsets were created:

- **Subset 1:** Focused on five complaint categories, reduced from 65,800 to 6,580 entries, maintaining original category proportions.
- **Subset 2:** Included 1,524 entries within the Electrical System, divided into 32 closely related complaint families to test model sensitivity.

3.1.4.3 ML models

We applied various traditional machine learning models for text classification. Hyperparameter tuning and cross-validation were conducted to optimize these models. The specifics of these methods and their results will be discussed in the results section.

3.1.4.4 Large Language Models (LLMs)

We utilized an advanced LLM for categorizing vehicle failures and crafted specific prompts to evaluate its effectiveness.

3.1.5 Deployment

Typically, creating the model does not mark the conclusion of the project. Often, the insights gained need to be structured and presented in a format usable by the customer. Depending on the specifications, the deployment phase can range from simply generating a report to implementing a repeatable data mining process [36]. To demonstrate the level of deployment, we utilize the Technology Readiness Levels (TRLs). These levels serve as a systematic framework for assessing the maturity and readiness of a technology for practical implementation. It consists of nine levels, ranging from TRL 1, where basic principles are observed, to TRL 9, where a technology is proven through successful operational missions. At TRL 2, technology concept and application are formulated, progressing through TRL 3 where experimental proof of concept is demonstrated in a laboratory environment. By TRL 4, technology is validated in a laboratory environment and further tested in relevant environments at TRL 5. At TRL 6, technology is demonstrated in a relevant operational environment, paving the way for system integration and demonstration at TRL 7. TRL 8 involves the actual system completion and successful testing in an operational environment, while TRL 9 indicates technology readiness with proven performance through operational missions [39].

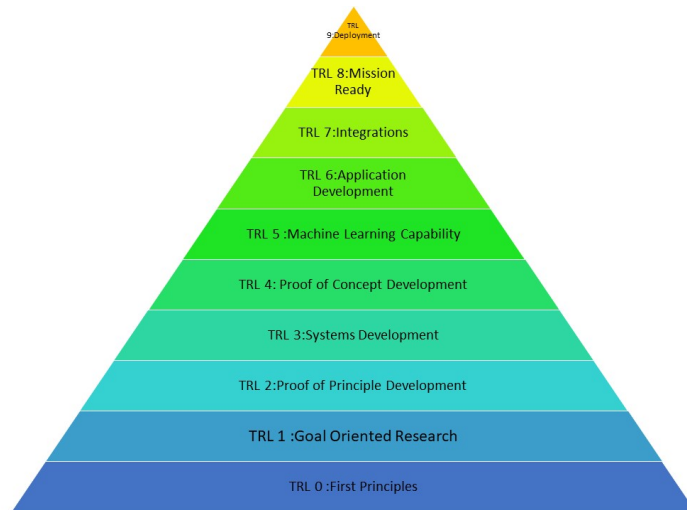


Figure 3.4: Technology Readiness Level Pyramid

4

Results

This section of the thesis presents a comparative evaluation of some traditional machine learning models against the large language model Anthropic Claude, specifically in the context of vehicle diagnostics tasks. We aim to assess their efficacy in processing and analyzing automotive issues, which is critical for enhancing diagnostic accuracy and efficiency. Furthermore, we illustrate a practical application of these models through the development of a specialized chatbot. This chatbot is designed to assist engineers and customers alike by providing targeted responses to queries about vehicle malfunctions, utilizing the extensive technical documentation available as its knowledge base. This real-world application underscores the potential of LLMs to transform how automotive support and customer service are delivered.

4.1 Business Understanding

From our discussions with stakeholders, several key considerations emerged. Firstly, stakeholders acknowledged the vast amount of data available regarding failures, which is often underutilized. They agreed that this information could help them better comprehend the diagnostic and troubleshooting processes. However, the main challenge lies in the fact that failure descriptions and the corresponding resolution data are stored in different databases, meaning that collecting and linking this information represents a significant challenge. Stakeholders saw great potential in the idea of learning from historical data to determine which solutions were adopted for specific types of failures, especially those showing certain patterns like fault codes and usage parameters. They also appreciated the concept of a text classification tool that could automatically categorize failure types. Currently, this operation is performed manually by workshop staff, leading to inconsistencies due to varying expertise and methods. An automated classification system could standardize this process across the company, improving efficiency and consistency.

Thanks to the close talk with experts, we also got a better understanding of the current processes within the company, with special focus on the Field Quality (FQ) process described in Paragraph 1.4.

4.2 Data Understanding

4.2.1 NHTSA dataset

The dataset utilized in this study is sourced from the National Highway Traffic Safety Administration (NHTSA) Complaints Database, part of the U.S. Department of Transportation [40]. The dataset, publicly accessible, consists of 349,996 entries collected from 1995 to 2024 (as of April 24). This dataset plays a critical role in monitoring existing recalls to ensure their scope and adequacy, while also providing insights into the effectiveness of safety measures over time. It is contributed to by individuals reporting safety problems they experience with their vehicles and related equipment through NHTSA's 'Report a Safety Problem' service. The reporting process involves filling out a detailed complaint form available on the NHTSA's website.

The dataset encompasses a total of 49 fields [41], each providing detailed information regarding the reported complaints. Among these fields, two key fields are particularly relevant to the scope of this study:

- **COMPDESC:** This field provides specific component descriptions crucial for fault classification, identifying the family of components implicated in reported issues (e.g., engines, brakes, etc). It serves as essential labels for training text classification models, enabling accurate prediction of fault categories based on textual descriptions.
- **CDESCR:** This field contains the complaint text description, offering in-depth narratives of the reported issues as provided by the consumers.

Here an example is given to show how the data look like.

4.3 Data Preparation

The original dataset contains 349.996 entries, a size that presents several practical challenges for computational processing and analysis within the scope of this master's thesis. Handling this large volume of data would require extensive computational resources and not insignificant costs. Furthermore, the complexity and time required for processing and training machine learning models on the full dataset would significantly slow down the iterative process of model tuning and evaluation. Given these constraints, a strategic reduction of the dataset is necessary. The approach adopted involves selecting specific subsets of the data that are manageable in size yet still representative of the broader dataset. To address this, the original dataset has been split into two subsets. The features of these subsets will be presented below, while justifications for selecting these specific subsets will be discussed in the subsequent discussion section (5.2).

Table 4.1: NHTSA data of interest - example

COMPDESC	CDESCR
AIR BAGS	There was a severe head on collision and airbags did not deploy. The dealership says there was no deceleration however, the crash was severe enough that they should have deployed, the driver also tried to stop and the trucks airbag system did not detect the sudden brake. This is a 2021 truck that has a sudden brake sensor, that combined with the collision should have caused the airbags to deploy.
ELECTRICAL SYSTEM	While driving my truck at approximately 70MPH on the highway, the truck shut down. I was able to pull off the side of the road and dashboard showed the message of Electronic Throttle Control and the engine light turned on. This is the third report I have filed regarding this truck shutting down while driving and the 7th time this truck has randomly shut down on me since purchasing in July 2021. This has been on the dealership 2 separate times and is going in tomorrow for a third time for this issue, I no longer feel safe driving this truck and feel this incident should be address as soon as possible.
ENGINE AND ENGINE COOLING	Car was sitting idle when a fan inside the engine compartment starting being very loud. Sound went away when car started moving. Noticed coolant was empty. Filled with coolant and made a 5 mile trip. Checked again and coolant was empty again.
STRUCTURE	Subframe is so rusty and it has cracked.
WHEELS	All lugs on the right rear wheel have snapped off. At first it was one which was repaired. Then three at once broke off and caused damage. After it was repaired again the fifth and final one broke off.

4.3.1 Subset 1

Subset 1 includes five distinct 'families' of complaints: Air Bags, Electrical System, Engine and Engine Cooling, Structure, and Wheels. These categories were selected to represent a broad spectrum of common issues, providing a diverse basis for evaluating text classification models.

The initial sum of entries from these selected categories amounted to approximately 65,800 data points. This size was further reduced to 6,580 entries while maintaining the original proportion of each category. This reduction ensures that the smaller dataset mirrors the distribution of the larger one, maintaining statistical validity for comparative analysis.

The choice of distinct categories in Dataset 1 serves as a preliminary test bed for these technologies, providing initial insights into their performance in a relatively straightforward scenario.

Figure 4.1 shows the distribution of the different categories of complaints included in Subset 1, before and after the sampling procedure.

4. Results

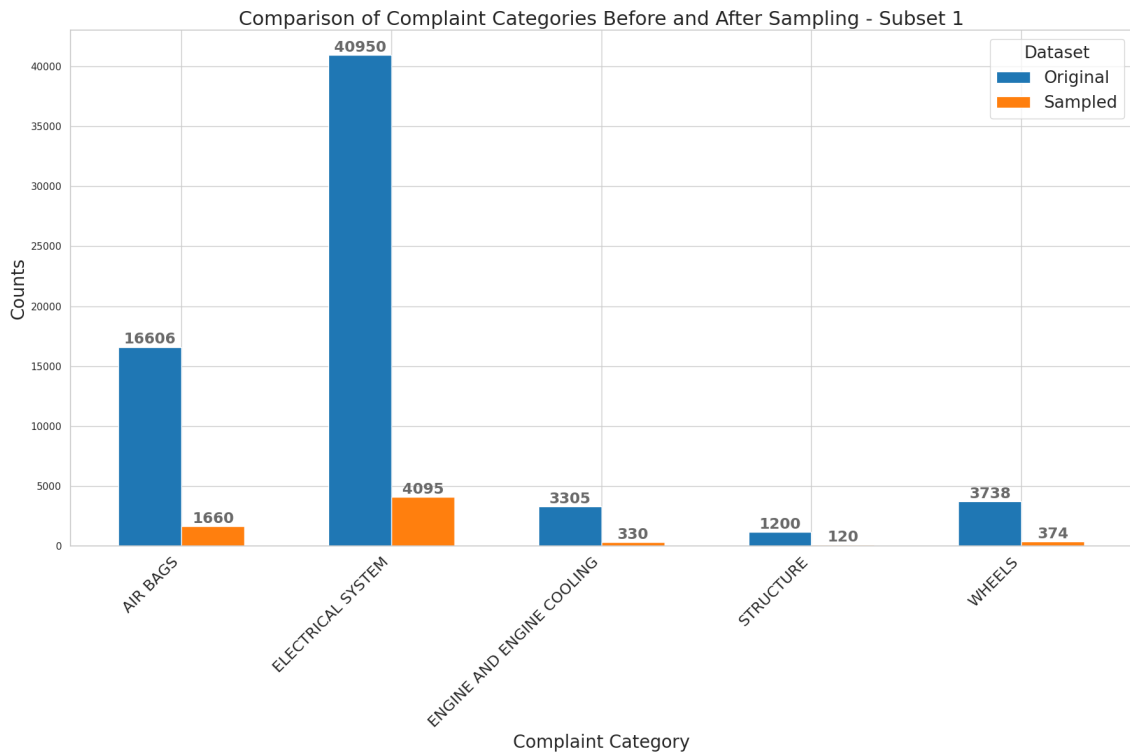


Figure 4.1: Subset 1

4.3.2 Subset 2

In the preparation of Subset 2, the selection strategy focused on complaints all pertaining to one vehicle's main system, the Electrical System in our case, but subdivided into highly similar and closely related sub-categories.

Subset 2 contains 1524 datapoints divided into 32 complaint 'families' that are listed in Figure 4.2.

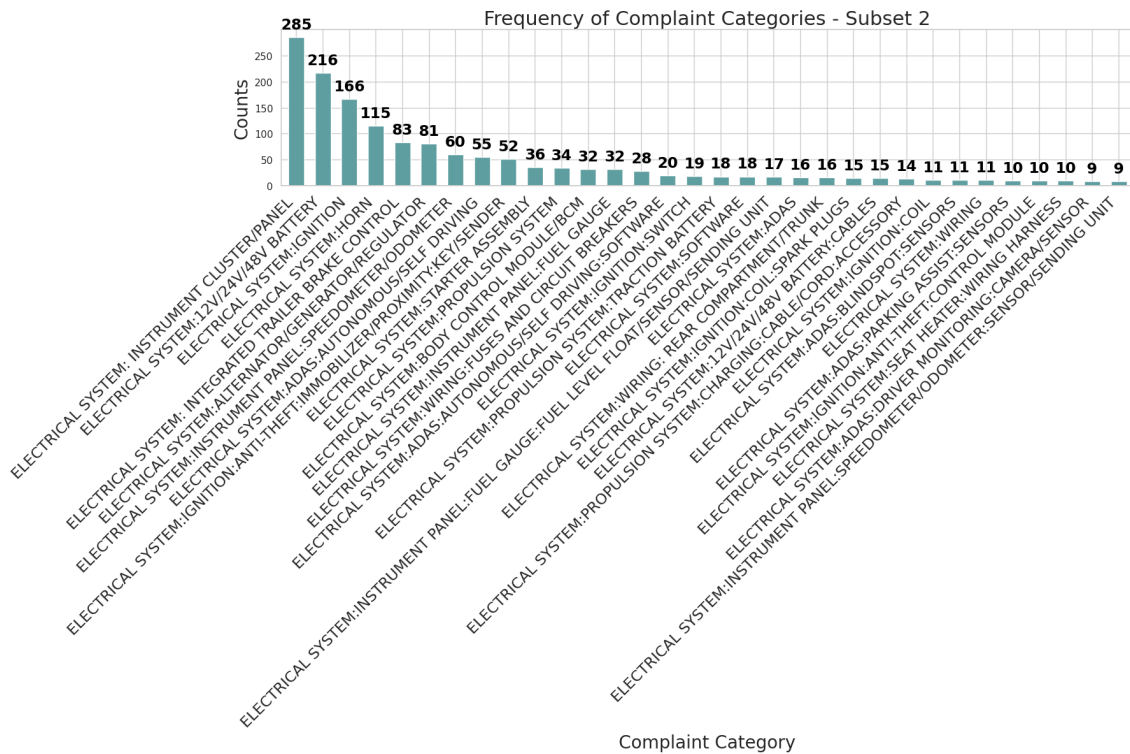


Figure 4.2: Subset 2

For the purposes of this study, Subset 1 and Subset 2 will be used to test the efficacy of traditional machine learning models against large language models in categorizing complaints accurately.

4.4 Modelling and Evaluation

This section presents the results of our experiments on the dataset, focusing on three primary tasks to evaluate the capabilities of traditional machine learning models and LLMs.

First, we undertake a text classification task, where both traditional machine learning models and large language models are tested for their performance. This task involves categorizing complaints into predefined categories and provides a comparative analysis of the accuracy, precision, recall, and F1-score of each model type.

Second, we assess the capacity of large language models to extract specific information from a complaint description.

Third, we explore the ability of large language models to retrieve and synthesize information to answer specific questions accurately.

4.4.1 Text Classification

4.4.1.1 Traditional Machine Learning Approaches for Text Classification

Traditional machine learning approaches offer a variety of algorithms that can be applied to extract insights and make predictions based on historical data. In the

context of text classification tasks, these methods involve training algorithms to assign a category or label to a piece of text based on features extracted from the text itself.

These algorithms are known as classifiers. They analyze the input text, extract relevant features, and use these features to predict the most appropriate category for each text instance. The 9 classifiers utilized in our evaluation are presented in detail in 2.2.1 and include:

- **Tuned Random Forest (Tuned RF)**
- **Random Forest without Specific Tuning (RF No Tuning)**
- **Support Vector Machine with Gaussian Kernel (SVM Gaussian):**
- **Support Vector Machine with Polynomial Kernel (SVM Poly)**
- **Support Vector Machine with Radial Basis Function Kernel and Cost Tuning (SVM RBF Cost)**
- **AdaBoost with Decision Tree (AdaBoost DT)**
- **K-Nearest Neighbors (KNN)**
- **Naive Bayes Multinomial (MN)**
- **Naive Bayes Complement (CM)**

4.4.1.1.1 Hyperparameter tuning of classifiers To optimize the performance of the classifiers used in our study, we employed GridSearchCV [42], a comprehensive tool for hyperparameter tuning that systematically searches through a specified range of parameter values, employing cross-validation to assess the models'. This process is crucial for identifying the optimal settings for each classifier, thereby enhancing their accuracy and effectiveness in classifying text complaints into pre-defined categories. K-fold Cross-Validation (with K=5) [43] is used, while dataset is splitted such that 20% of the data is reserved for testing the model, while the remaining 80% is used for training.

4.4.1.1.2 ML models performance Each classifier is trained using the training subset. During this process, the classifier is fed with labeled data, allowing it to learn the features associated with each category. After training, the classifier is used to make predictions on the testing subset. The predicted labels are then compared with the actual labels to assess the classifier's performance. This enables the evaluation of classifier performance using metrics such as accuracy, precision, recall, and F1-score.(2.2.1.4).

Here below, in Table 4.2, the performance of each classifier is presented for the two subsets.

All these results are reported as weighted averages. This means that the performance metrics are averaged across all categories, with each category's contribution to the average weighted by the number of instances in that category. Weighted averages are particularly useful in cases where the dataset is imbalanced, as they ensure that the performance on larger categories does not influence negatively the overall metric, providing a more balanced and fair evaluation of the classifier's effectiveness across all categories.

Classifier	Subset	Precision	Recall	F1-Score	Accuracy
Tuned RF	Subset 1	0.87	0.86	0.84	86%
	Subset 2	0.63	0.68	0.60	67%
RF No Tuning	Subset 1	0.87	0.86	0.84	86%
	Subset 2	0.62	0.66	0.58	66%
SVM Gaussian	Subset 1	0.90	0.90	0.90	90%
	Subset 2	0.64	0.69	0.63	69%
SVM Poly	Subset 1	0.90	0.90	0.90	90%
	Subset 2	0.66	0.67	0.61	67%
SVM RBF Cost	Subset 1	0.91	0.91	0.90	91%
	Subset 2	0.65	0.69	0.63	69%
AdaBoost DT	Subset 1	0.83	0.83	0.82	83%
	Subset 2	0.34	0.38	0.26	38%
KNN	Subset 1	0.85	0.85	0.85	85%
	Subset 2	0.58	0.63	0.58	63%
Naive Bayes MN	Subset 1	0.83	0.83	0.82	83%
	Subset 2	0.56	0.60	0.55	60%
Naive Bayes CM	Subset 1	0.84	0.84	0.84	84%
	Subset 2	0.67	0.72	0.67	72%

Table 4.2: Performance metrics for each classifier on two subsets

4.4.1.2 Large Language Model as a Text Classifier

Beyond traditional machine learning methods, we aim to evaluate the capacity of a LLM as a text classifier. Large language models, such as Anthropic Claude v2, are designed to understand and generate human-like text, making them powerful tools for a variety of natural language processing tasks. In this study, we leverage the capabilities of Anthropic Claude v2 to assess its effectiveness in categorizing vehicle failures and complaints, specifically in tasks such as classifying text into predefined categories related to vehicle issues.

The process of interacting with LLMs involves the use of prompt engineering. This technique entails crafting specific text-based prompts that guide the model to understand and execute the desired task. The prompt effectively sets the task boundaries and provides the model with the context necessary to generate relevant and accurate outputs. By carefully designing these prompts, we can instruct the LLM to perform complex tasks, such as text classification, with a high degree of accuracy.

In our experimental setup, we have developed and tested three distinct prompts to evaluate how effectively Anthropic Claude v2 can classify text according to the predefined categories. This approach wants to test the model’s accuracy in text classification and also compare its performance against traditional ML models specifically trained on the dataset used in this study.

The three prompts used in this study share a similar structure, each serving as an evolution of the previous one, enhancing the clarity and specificity of the task for the Large Language Model (LLM).

4.4.1.2.1 Prompt 1 Each prompt begins by assigning a specific role to the AI, stating that it is an expert field quality engineer trained to categorize descriptions of car failures into defined categories. This introductory statement sets the context for the task and primes the model to focus its processing capabilities on categorization based on symptom descriptions. Following the role assignment, the task is explicitly described: The AI is instructed to analyze a complaint and classify it by returning only the appropriate category label. The clarity of the command informs the LLM exactly what is expected, emphasizing that the output should solely be a category label, with no additional information or elaboration required. The input for the task is specified through a placeholder within the prompt, where the actual complaint text is to be inserted. The prompt lists specific categories from which the AI must choose, confining the AI's responses to these predefined options. This part of the prompt is vital as it restricts the scope of possible answers, ensuring that the responses are relevant and within the study's parameters. A first version of the prompt can be seen in Listing A.1.

4.4.1.2.2 Prompt 2 Prompt 2 is a refined iteration of the original prompt 1, designed to enhance clarity and context for the AI model when classifying car failures. The most significant enhancement is the detailed description of each category. These descriptions provide comprehensive insights into typical problems and symptoms associated with each category, such as "ELECTRICAL SYSTEM" detailing issues like dead batteries and failed alternators, or "AIR BAGS" explaining potential malfunctions with deployment mechanisms. This additional information serves two crucial functions: it guides the AI's decision-making process by clearly outlining what constitutes a complaint in each category, and it equips the model with a deeper understanding of the context and technical nuances involved, potentially increasing the precision of its responses compared to the more straightforward format of the first prompt. An example of the second version of the prompt is shown in Listing A.2.

4.4.1.2.3 Prompt 3 Prompt 3 further advances the structure introduced in version 2 by incorporating specific, illustrative examples for each category of car failure. This prompt provides real-world scenarios that are representative of typical complaints in each category. These examples serve as practical references that can guide the AI in associating the descriptions of symptoms and failures with the correct category. For instance, the description for the 'ELECTRICAL SYSTEM' now includes a detailed account of a specific vehicle's electrical issues, complete with diagnostic steps and outcomes. This approach aims to reduce misclassifications by aligning the AI's responses more closely with how human experts would interpret and categorize similar complaints. An example of prompt 3 is visible in Listing A.5.

4.4.1.2.4 LLM performance The performance of the LLM in text classification tasks was evaluated by counting the correct and incorrect predictions. By comparing the predicted labels with the true labels, the metrics of accuracy, precision, recall, and F1-score were calculated. Table 4.3 shows the results obtained by applying the three versions of the prompts to the two subsets.

Model	Prompt	Subset	Precision	Recall	F1-Score	Accuracy
Anthropic Claude v2	Prompt 1	1	0.90	0.70	0.77	70%
		2	0.73	0.45	0.50	45%
	Prompt 2	1	0.88	0.77	0.82	77%
		2	0.70	0.46	0.49	46%
	Prompt 3	1	0.87	0.74	0.78	74%
		2	-	-	-	-

Table 4.3: Performance metrics for Anthropic Claude v2 using different prompts across two subsets

4.4.1.2.5 Overall Comparison In Table 4.4, the performance metrics of the machine learning classifiers are presented alongside those of the LLM. This arrangement facilitates a clearer and easier comparison of their respective performances.

Type	Model/Prompt	Subset	Precision	Recall	F1-Score	Accuracy
LLM	Prompt 1	1	0.90	0.70	0.77	70%
		2	0.73	0.45	0.50	45%
	Prompt 2	1	0.88	0.77	0.82	77%
		2	0.70	0.46	0.49	46%
	Prompt 3	1	0.87	0.74	0.78	74%
		2	-	-	-	-
Traditional	Tuned RF	1	0.87	0.86	0.84	86%
		2	0.63	0.68	0.60	67%
	RF No Tuning	1	0.87	0.86	0.84	86%
		2	0.62	0.66	0.58	66%
	SVM Gaussian	1	0.90	0.90	0.90	90%
		2	0.64	0.69	0.63	69%
	SVM Poly	1	0.90	0.90	0.90	90%
		2	0.66	0.67	0.61	67%
	SVM RBF Cost	1	0.91	0.91	0.90	91%
		2	0.65	0.69	0.63	69%
	AdaBoost DT	1	0.83	0.83	0.82	83%
		2	0.34	0.38	0.26	38%
	KNN	1	0.85	0.85	0.85	85%
		2	0.58	0.63	0.58	63%
	Naive Bayes MN	1	0.83	0.83	0.82	83%
		2	0.56	0.60	0.55	60%
	Naive Bayes CM	1	0.84	0.84	0.84	84%
		2	0.67	0.72	0.67	72%

Table 4.4: Consolidated Performance Metrics Comparing Anthropic Claude v2 LLMs and Traditional Classifiers

4.4.2 Feature Extraction

Large Language Models (LLMs) are revolutionizing the way we understand and process textual information. These models possess an intrinsic ability to discern patterns and extract meaningful insights from text data, which can be applied effectively in domains such as vehicle maintenance and repair diagnostics. In the context of analyzing vehicle failures, LLMs can be employed not just for categorizing the nature of complaints but also for extracting and interpreting detailed features from textual descriptions of vehicle issues.

For all these reasons, we believed it was worthwhile to explore this approach to extract insights from issue descriptions. By utilizing Large Language Models (LLMs) in our diagnostic processes, we aim to harness their advanced analytical capabilities to interpret complex textual information accurately.

In order to do that we provided the LLM with a prompt specifically designed to not only direct the AI to classify the type of vehicle failure accurately but also guides it through a structured analysis that mimics the steps a human expert would take.

The prompt (see Listing A.4) tasks the LLM with multiple responsibilities. The model is first asked to identify the category of the failure from a pre-defined list, each with detailed descriptions and examples. Then it is asked to describe observable symptoms: the prompt encourages the extraction of specific details from the complaint, which are critical for accurate diagnosis. This includes identifying and articulating visual symptoms, operational issues, or abnormal noises, which are often key indicators of underlying problems. The model is also prompted to specify when the symptoms occur, under what conditions they are observed which is vital for diagnosing intermittent issues that may not be apparent under normal conditions. In addition the model should be able to infer possible causes based on the symptoms and conditions described. By integrating its vast knowledge base, the LLM can suggest probable technical explanations for the observed symptoms. Finally, the prompt asks the LLM to propose possible actions or remedies. This aspect of the prompt ensures that the output is not only descriptive and diagnostic but also prescriptive, providing practical value in addressing the vehicle's issues.

For more clearance, we provide an example showing how the LLM can extract and provide useful information about the following complaint made by a vehicle's user:

"The brake and ABS are on. Also there is a lightning bolt that appears on the cluster. The honk does not want to work as well and it is associated with the airbag since its installed with it in the center of the steering wheel."

Here is the Claude's response:

Based on the complaint, I have completed the 5 tasks:

n1. The most appropriate failure category is: 'ELECTRICAL SYSTEM'

n2. The observable symptoms are:

- The brake and ABS warning lights are illuminated on the instrument cluster*
- The ABS (anti-lock braking system) appears to be not functioning properly*
- There is a lightning bolt symbol appearing on the instrument cluster, indicating an electrical/electronic failure*
- The vehicle's horn is not working*
- The airbag warning light is on, indicating an issue with the airbag system*

n3. The symptoms occur:

- When the vehicle is running/driven*
- Continuously, not only under specific conditions*

n4. Possible causes are:

- Faulty ABS module or wheel speed sensors*
- Issue with central electrical module or wiring harness*
- Faulty clockspring or wiring related to horn and airbag systems*
- Low voltage/electrical power supply issue affecting multiple systems*
- Short circuit or broken wiring resulting in communication errors between modules*

n5. Possible actions to take:

- Scan vehicle for diagnostic trouble codes to identify specific faults*
- Inspect wiring harnesses and connections for damage, corrosion, or looseness*
- Check fuses and relays associated with affected systems*
- Test/replace faulty components like ABS module, wheel speed sensors, clockspring*
- Check battery, alternator, and ground connections*
- Repair any damaged wiring or faulty electrical connections"*

4.4.3 Recommendation System

4.4.3.1 Retrieval Augmented Based Recommendation System

In section 2.2.2.4, we discuss augmented large language models (LLMs), which enhance the capabilities of LLMs by integrating factual data from external databases. This augmentation helps improve the accuracy and relevance of the model's responses. For our recommendation system, we employ RAG to enhance the LLM's performance in generating responses to user queries about vehicle complaints and issues.

We built some technical documentation that serves as the knowledge base for our RAG system. Real data technical documentation was not allowed to use, due to security constraints, so some synthetic technical documentation that resembles the real data has been created by following the methodology illustrated in 3.1.3.3. Referring to Subset 1, for simplicity, we have created technical documentation for all the five families of complaints: airbags, electrical system, engine and engine cooling, structure, and wheels.

Essentially, we fabricated these documents to act as a source of data from which the LLM can retrieve information and answer detailed questions based on the context provided.

An example of one source of data is showed below:

Technical documentation for the Electric System family

Product Family Identification
ELECTRICAL SYSTEM belongs to the Automotive Electrical Systems product family, which is known for its main function of providing power and control for vehicle electrical components. This family has the FPC code 123.

Functionality Identification:
The primary function of Electrical System is distributing electrical power and controlling various electrical subsystems within the vehicle. It serves as a critical element in the overall vehicle system. Sub-functions associated with Electrical System include:

- **Power Distribution:** Ensures stable power supply to all electrical components.
- **Control Signal Processing:** Manages the control signals for various systems like lighting, infotainment, and safety features.
- **Battery Management:** Monitors and maintains the health and performance of the vehicle's battery.

Component Listing:
Electrical System comprises the following sub-components associated with the previous sub-functions:

- **Fuse Box:** Houses fuses and relays that protect electrical circuits, FPC code: 001-FB.
- **Wiring Harness:** Network of wires that connect electrical components, FPC code: 002-WH.
- **Battery Management Unit:** Monitors and controls battery charging and discharging, FPC code: 003-BMU.

Operational Factors:
Operational factors influencing the performance of Electrical System include:

- **Environmental Conditions:**
 - **Temperature:** High temperatures can cause overheating of wires and components, leading to potential failures. Low temperatures may reduce battery efficiency.
 - **Humidity:** Excessive moisture can cause corrosion of electrical connections and short circuits.
- **Usage Patterns:**
 - **Intensive Use:** Frequent use of electrical systems (e.g., infotainment, air conditioning) can drain the battery quickly and lead to faster wear of components.
 - **Intermittent Use:** Infrequent use can lead to battery discharge and potential issues with starting the vehicle.
- **Maintenance Requirements:**
 - **Regular Inspection:** Check for signs of wear, corrosion, or damage to wires and connectors.
 - **Cleaning:** Ensure that electrical connections are free from dust and debris.
 - **Battery Maintenance:** Regularly check battery health and fluid levels if applicable, and ensure proper charging.

Component Upgradability Assessment:
Electrical System can be upgraded under certain conditions to enhance its performance or capabilities. Upgradable aspects may include:

- **Factor 1:**
 - **Description:** Upgrading the Wiring Harness to high-performance, low-resistance wires.
 - **Upgrade Mechanism:** This can improve electrical efficiency and reduce power losses.
- **Factor 2:**
 - **Description:** Installing a Smart Battery Management Unit with advanced monitoring and control features.
 - **Upgrade Mechanism:** This can enhance battery life and provide real-time diagnostics and predictive maintenance capabilities.

Technical Working:
Electrical System operates by integrating its sub-components as follows:

- **Fuse Box:** Protects electrical circuits by breaking the circuit if an overcurrent condition occurs. It is connected to all major electrical subsystems and ensures they operate within safe limits.
- **Wiring Harness:** Acts as the backbone of the electrical system, transmitting power and signals between the battery, fuse box, and various electrical components.
- **Battery Management Unit:** Continuously monitors the state of the battery, controlling the charging process and ensuring optimal battery health and longevity.

Product Owner
The product owner is John located in Barcelona, Spain.

Using a prompt like the one described in Listing A.0.0.0.5, makes it possible to

start a conversation with the LLM. For clearance we provide an example within the Electrical System family. We are in the place of a user who experience the following problems with his car:

"Car is struggling to start, and the battery has been draining at least once a week but never got a battery low light. Have been driving it every weekday for about 5-10 miles consistently. Took it to AutoZone and they ran some tests, said there is no power leak and the battery is still in good condition. However, the battery still is draining."

The question provided to the LLM is:

"Battery in the electrical system is draining. Recommend a solution from the context provided."

Claude's answer is:

Based on the information provided, a couple potential solutions for a battery draining issue with the electrical system could be:

- Perform regular battery maintenance, such as checking fluid levels and ensuring proper charging, as mentioned in the Maintenance Requirements section. This can help optimize battery health and longevity.*
- Upgrade to a Smart Battery Management Unit, as described in the Component Upgradability Assessment section. This can provide advanced monitoring and control capabilities to better manage the battery's charging and discharging.*

The details provided on battery maintenance requirements and the ability to upgrade to an improved Battery Management Unit suggest these could be viable solutions to address a battery drain issue, if applicable. Let me know if you need any clarification or have additional questions!

4.4.3.1.1 Evaluation of the Recommendation System We utilize human feedback as an evaluation metric for assessing LLMs, as discussed in Section 2.2.2.8. We apply the same assessment to the problem discussed above, scoring the output generated by the LLM on a scale from 0 to 5, with 0 representing the most disagreed and 5 representing the most agreed. Specifically, we pose 6 questions as input to the LLM RAG-based system and evaluate the responses to 16 criteria outlined in Section 2.2.2.8.

To help understanding how human evaluation is performed we show two tables. Table 4.5 provides detailed information on the questions, the reference (true) answers, and the LLM-generated answers; while Table 4.6 presents the scores for each question against the 16 criteria.

4. Results

Table 4.5: LLM’s Question & Answer Results

Question Number	Question	Reference (True Answer)	LLM Answer (Generated Answer)
Q1	Battery in the electrical system is draining. Recommend a solution from the context provided.	Battery Maintenance: Regularly check battery health and fluid levels if applicable, and ensure proper charging. Description: Installing a Smart Battery Management Unit with advanced monitoring and control features. Upgrade Mechanism: This can enhance battery life and provide real-time diagnostics and predictive maintenance capabilities.	Based on the context provided, here are two potential solutions I would recommend for a battery drain issue in the electrical system: Perform regular battery maintenance. The context mentions that regular battery maintenance is a requirement, including checking fluid levels if applicable and ensuring proper charging. This maintenance can help prevent or catch battery issues early. I would recommend following the specified maintenance schedule and procedures. Install a smart battery management unit. The context states that upgrading to a smart BMU with advanced monitoring and control features can enhance battery life and provide diagnostics. Installing this upgrade could help manage the battery charging and condition more effectively to prevent draining. The key factors are to either improve the existing battery maintenance if that has lapsed, or upgrade to a more advanced BMU that can better monitor and manage the battery. Without more specific details on the battery type and issue, those seem like reasonable potential solutions based on the information provided. Let me know if you need any clarification or have additional context to share.
Q2	What is the FPC code for the electrical system?	FPC 123	Based on the technical documentation provided, the FPC code for the Electrical System product family is 123. Specifically, the documentation states: "Electrical System belongs to the Automotive Electrical Systems product family, which is known for its main function of providing power and control for vehicle electrical components. This family has the FPC code 123." Therefore, the FPC code for the Electrical System is 123.
Q3	Who is the product owner for the electrical system?	John, Spain, Barcelona	Based on the technical documentation provided: The product owner for the electrical system is John located in Spain, Barcelona. This is stated in the "Product Owner" section under the details for the Electrical System: "Product Owner The product owner for electrical system is John located in Spain, Barcelona."
Q4	Who is the product owner for air bags?	Aiden Kim, Tokyo	Based on the provided context, the product owner for air bags is Aiden Kim located in Tokyo, Japan. This information is provided in the last section for the Air Bags context: "Product Owner Name: Aiden Kim Location: Tokyo, Japan."
Q5	The battery is drained. What upgrade would you recommend?	Electrical System can be upgraded under certain conditions to enhance its performance or capabilities. Upgradable aspects may include: Factor 1: Description: Upgrading the Wiring Harness to high-performance, low-resistance wires. Upgrade Mechanism: This can improve electrical efficiency and reduce power losses. Factor 2: Description: Installing a Smart Battery Management Unit with advanced monitoring and control features. Upgrade Mechanism: This can enhance battery life and provide real-time diagnostics and predictive maintenance capabilities.	Based on the provided context, I do not have enough information to recommend a specific upgrade for a drained battery scenario. The details provided discuss general upgradability factors for the Electrical System, but do not indicate any specific upgrades to address battery drain issues. More diagnostic information would be needed to determine the root cause and suggest an appropriate upgrade. Without further details on the battery drain issue, I cannot provide a reliable recommendation.
Q6	The battery is drained in the electrical system. What upgrade would you recommend?	Electrical System can be upgraded under certain conditions to enhance its performance or capabilities. Upgradable aspects may include: Factor 1: Description: Upgrading the Wiring Harness to high-performance, low-resistance wires. Upgrade Mechanism: This can improve electrical efficiency and reduce power losses. Factor 2: Description: Installing a Smart Battery Management Unit with advanced monitoring and control features. Upgrade Mechanism: This can enhance battery life and provide real-time diagnostics and predictive maintenance capabilities.	Based on the provided context, I would recommend upgrading to a high-capacity battery as a potential solution for the drained battery issue in the electrical system. The key reasons are: The electrical system relies on the battery to provide power to all components when the engine is off. A drained battery will lead to failure of the electrical system. Upgrading to a high-capacity battery can provide increased reserve capacity and withstand deeper discharge cycles. This can help prevent the battery from draining too quickly during intensive electrical loads. The Component Upgradability Assessment specifically mentions "Installing a Smart Battery Management Unit" as a potential upgrade to enhance battery life and provide diagnostics. This indicates the battery itself is upgradable. A high-capacity battery aligns with the upgrade mechanisms mentioned for electrical system enhancements. It can improve electrical efficiency and supplement the wiring harness upgrades. So in summary, I would recommend exploring the upgrade to a high-capacity battery as a targeted solution to address the specific issue of a drained battery in the electrical system. Please let me know if you need any clarification or have additional questions!

Table 4.6: Score for Evaluation Metrics in LLMs

Question	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16
Q1	5	5	4	5	5	5	0	5	5	5	0	0	0	0	0	0
Q2	5	5	5	5	5	5	0	5	5	5	0	0	0	0	0	0
Q3	5	5	5	5	5	5	0	5	5	5	0	0	0	0	0	0
Q4	5	5	5	5	5	5	0	5	5	5	0	0	0	0	0	0
Q5	0	0	0	0	0	0	5	0	0	0	5	0	5	5	0	0
Q6	4	5	5	4	4	3	3	5	5	5	0	0	0	0	0	0

4.5 Deployment

4.5.1 Application Overview

In this section, we describe the deployment of our demonstration application, which showcases the capabilities of the large language model (LLM) Anthropic Claude v2. The demo app, built using Streamlit, allows users to perform the three key tasks explored in this thesis:

- **Text Classification:** Predicting the category of the complaint.
- **Feature Extraction:** Extracting relevant insights and details about the failure.
- **Context-Based Question Answering:** Allowing users to ask follow-up questions about the failure and receive informed responses based on the technical context provided.

The deployed application provides a user-friendly interface where users can input a description of a vehicle complaint. The input is processed by Anthropic Claude v2, which performs text classification and feature extraction. The complaint category is predicted, and insights about the failure are provided. After the initial response, users can engage in a chat with the LLM, asking specific questions about the failure. The LLM responds with contextually relevant information, guiding the user on potential actions to resolve the problem.

To illustrate, consider the following:

4. Results

The image shows a web application interface for a Vehicle Complaint Analysis Tool. On the left, the tool's header is "Vehicle Complaint Analysis Tool". Below it, a text input field contains a detailed complaint: "CAR IS STRUGGLING TO START, AND BATTERY HAS BEE DRAINING AT LEAST ONCE A WEEK BUT NEVER GOT A BATTERY LOW LIGHT. HAVE BEEN DRIVING IT EVERY WEEK DAY FOR ABOUT 5-10 MILES CONSISTENTLY. TOOK IT TO AUTOZONE & THEY RAN SOME TESTS, SAID THERE IS NO POWER LEAK AND THE BATTERY IS STILL IN GOOD CONDITION. HOWEVER THE BATTERY STILL IS DRAINING." Below the input is an "Analyze Complaint" button. A green notification bar states "Analysis completed successfully." Below that, a "See Analysis Results" section is expanded to show the following details:

- Category:** "ELECTRICAL SYSTEM"
- Symptoms:** "car struggling to start, battery draining at least once a week"
- Conditions:** "Not Available"
- Causes:** "possible faulty alternator not properly charging battery, parasitic draw draining battery when vehicle is off"
- Actions:** "inspect alternator output and connections, check for parasitic draws with multimeter when vehicle is off, test battery and consider replacement if weak"

On the right, the "Interactive Q&A" section has a text input field with the question: "Battery in the electrical system is draining . Recommend a solution from the context provided." Below it is a "Submit" button. The "Answer" section provides the following response:

Based on the information provided, a couple potential solutions for a battery draining issue with the electrical system could be:

1. Perform regular battery maintenance, such as checking fluid levels and ensuring proper charging, as mentioned in the Maintenance Requirements section. This can help optimize battery health and longevity.
2. Upgrade to a Smart Battery Management Unit, as described in the Component Upgradability Assessment section. This can provide advanced monitoring and control capabilities to better manage the battery's charging and discharging.

The details provided on battery maintenance requirements and the ability to upgrade to an improved Battery Management Unit suggest these could be viable solutions to address a battery drain issue, if applicable. Let me know if you need any clarification or have additional questions!

Figure 4.3: Vehicle Complaint Analysis Tool - Demo App

4.5.2 Readiness Level

Given the successful deployment and evaluation, this demo application can be considered at Research and Technology Level (RTL) 6. This level indicates that the system has been demonstrated in a relevant environment and has shown to work as expected. The application has been rigorously tested with synthetic data and user interactions, ensuring its reliability and effectiveness.

5

Discussion

This chapter analyzes, interprets, and evaluates the research findings. We will explain what the findings mean and share insights that can help others who want to continue this work in the future. By looking at what worked well and what didn't, we hope to provide useful information for improving the current approach.

5.1 Data Understanding

Our data understanding began with an investigation into a user case of the air processing system (APS). It was found that most failure reports for this system were resolved by repairs. However, under certain circumstances, some failures were addressed by upgrading to a variant of the component. Consequently, a system was needed to identify the affected component in the failure report and recommend a solution.

Further investigation revealed that, regardless of whether the solution involves repair or upgrade, it should adhere to the company's modularization principles. In the case of an upgrade, the solution initiates an engineering change process, where a change or upgrade in one component triggers changes in other subsystems. This engineering change chain reaction is identified by a primary FPC code. The company's existing system provides the associated subsystems affected by this primary FPC code. However, a key finding is that retrieving this primary FPC code from the database is time-consuming.

Through further interaction with the company's AI team, it was revealed that failure reports often contain names. This raises concerns about potential data leakage through the security vulnerabilities of LLMs, which could violate GDPR rules. Consequently, this thesis chose to use the vehicle complaint dataset from NHTSA. Our investigation into the use case and interactions with various stakeholders highlighted the need to develop a system that can recognize the parts affected in failure reports, reorganize technical documentation to allow LLMs to effectively answer user queries, and develop a dataset from NHTSA that closely resembles the real dataset.

Compared to the NHTSA data, the real dataset has a more hierarchical structure, where one main group has many associated subgroups. Failure reports could potentially be categorized into more than one subsystem, making it a complex task to identify which subsystem is affected. Additionally, failure reports often contain the names of individuals, and any leakage of these names would violate GDPR regula-

tions. The main challenge facing the technical documentation is that it is scattered at different locations.

5.2 Data Preparation

In our exploration of text classification using large language models (LLMs), we encountered the initial challenge of dealing with a massive dataset containing 349,996 entries. As already mentioned in the corresponding Results section, this size posed practical challenges for computational processing and analysis within the scope of our master’s thesis. Handling such a large volume of data would have required extensive computational resources and incurred significant costs. Given these constraints, a strategic reduction of the dataset was necessary.

To address this, we adopted an approach involving the selection of specific subsets of the data that were manageable in size yet still representative of the broader dataset. The original dataset was split into two subsets, each chosen with particular objectives in mind.

Subset 1: was selected with several key considerations in mind. First, the reduction in dataset size aimed at enhancing computational efficiency. Smaller datasets can be processed and analyzed more quickly, facilitating a faster experimental cycle, which is critical for a thesis project constrained by time and resource limits. Additionally, by starting with clearly distinguishable categories, we could rigorously test and refine the models under controlled conditions before introducing more complexity. This approach, a standard practice in scientific investigations, allowed us to establish a clear baseline for evaluating and comparing the performance of various machine learning and language model techniques. The selected categories are sufficiently different from each other, establishing a clear baseline for evaluating and comparing the performance of various machine learning and language model techniques. Using this dataset as a control group, we can better demonstrate the incremental challenges and complexities introduced in Subset 2, which features more similar complaint families.

Subset 2 was chosen to test the robustness and sensitivity of machine learning models in distinguishing subtle nuances within text complaints. This methodical choice aimed to reflect real-world complexities where differentiating closely linked issues is frequently required. By introducing a higher degree of classification difficulty, Subset 2 provided a realistic and challenging environment for model evaluation. This setup allows for a detailed comparative analysis against the more distinct categories of Dataset 1, emphasizing the impact of category similarity on classification accuracy. Moreover, training on such a nuanced dataset helps uncover the potential and limitations of current classification technologies, as well as large language models applications, offering insights that could potentially influence future research directions in automated text classification.

In acknowledging the limitations of our approach, we recognize that these simplifications are necessary for the context of our study. In real-world applications, the process of classifying actual failures becomes inherently more complex and less straightforward. Numerous factors influence whether a failure is categorized into

one family or another, and it's not always evident or straightforward. Two failure descriptions may share very similar features but may not actually belong to the same failure category in reality. For example, let's imagine a scenario where a warning light suddenly illuminates on the dashboard, signaling an engine failure and prompting the need for immediate inspection. Now, this single event could indicate a range of potential underlying issues. It might indeed signify a problem with a specific cylinder within the engine (a mechanical issue). Or, it could point to a fault within the electrical system, especially if the warning light is triggered due to a damaged cable connected to a sensor.

5.3 Modeling

5.3.1 Text Classification

Supervised learning methods: The primary goal of applying these traditional machine learning methods in our study is to establish a benchmark for comparing their performance in text classification tasks against that of advanced Large Language Models (LLMs). By understanding how well these conventional models perform, we can better appreciate the advancements and potential improvements offered by LLMs in handling complex text-based data.

Given the structured nature of our dataset, where each text complaint is explicitly labeled with a category, supervised learning methods are particularly appropriate for our analysis. Supervised learning algorithms learn to predict outcomes within a labeled dataset, making them ideal for tasks where the objective is to forecast specific results based on past examples. This is in contrast to unsupervised learning, which is more suited to exploring underlying patterns and structures in data where no labels are provided. Choosing supervised methods allows us to directly apply and evaluate accuracy and F1-score, key metrics for assessing the efficacy of the models in correctly categorizing text, that enable us to quantitatively compare the capabilities of traditional models with those of LLMs under similar conditions, thereby providing a clear metric-based assessment of performance across different approaches.

Classifiers selection: In the pursuit of an effective and robust comparison study for text classification tasks, it is crucial to select algorithms that have demonstrated superior performance across diverse datasets. Our choice of classifiers is deeply informed by the comprehensive study conducted by Fernández-Delgado et al [44], that evaluates 179 classifiers over 121 datasets, providing a rigorous comparison of their performance.

Our focus is on the classifiers that consistently demonstrated exceptional accuracy and robustness in handling complex textual data. The standout models include variations of Random Forests and Support Vector Machines, particularly those using Gaussian and polynomial kernels. In our experimental framework, we have carefully selected a subset of high-performing classifiers as identified in rigorous academic studies [44]. The classifiers utilized in our evaluation include:

1. **Tuned Random Forest (Tuned RF)**
2. **Random Forest without Specific Tuning (RF No Tuning)**

3. **Support Vector Machine with Gaussian Kernel (SVM Gaussian)**
4. **Support Vector Machine with Polynomial Kernel (SVM Poly)**
5. **Support Vector Machine with Radial Basis Function Kernel and Cost Tuning (SVM RBF Cost)**
6. **AdaBoost with Decision Tree (AdaBoost DT)**

Additionally, we incorporate **K-Nearest Neighbors (KNN)** and **Naive Bayes Multinomial (MN) and Complement (CM)** classifiers to our benchmarking. The inclusion of these methods is guided by findings by [45], which identifies KNN and Naive Bayes, alongside SVM, as the most commonly employed techniques in text mining research. This decision underscores our commitment to a comprehensive evaluation that encompasses a broad spectrum of well-established methods in text classification.

LLM: Large Language Models (LLMs) represent the forefront of artificial intelligence technology, offering advanced capabilities to handle a wide variety of complex tasks. These models have been pre-trained on vast datasets encompassing diverse topics and genres, which enables them to understand and generate human-like text based on the inputs they receive. This pre-training provides LLMs with a significant advantage over traditional machine learning models, which require specific training on targeted datasets to perform similar tasks. However, a notable challenge in utilizing LLMs is ensuring that the model's responses remain within the scope of the study. Despite their extensive training, LLMs can sometimes generate outputs that, while plausible, may not align with the specific categories relevant to the research. To address this, the study involves crafting various prompts that restrict the model's responses to the target categories identified for vehicle failures and complaints. In this way we want to measure to which extent a prompt aligns with the task objectives, ensuring that the responses are both relevant and confined to the desired context.

The large language model utilized in this study is Claude v2 by Anthropic. We opted for this model primarily because it was readily available in the AWS account provided to us. Additionally, Claude v2 is reputed to excel in tasks such as natural language understanding and generation, making it a suitable choice for our text classification experiments. Due to time constraints, we focused solely on testing Claude v2 in our study. However, there is potential for future exploration involving other models such as ChatGPT, Titan, and more. This could provide valuable insights into their strengths and weaknesses. Furthermore, while conducting our experiments, we experimented with prompt engineering by creating three prompt templates based on suggestions provided on the Anthropic website (<https://docs.anthropic.com/en/docs/prompt-engineering>). However, there is scope for further exploration in this area.

5.3.2 Feature Extraction

Feature Extraction: Exploring the capacity of LLMs to extract meaningful information from text complaints is worth the investment due to its potential to revolutionize the diagnostic process in the automotive industry. Traditional diagnostic methods often rely on manual interpretation of symptoms, which can be

time-consuming and prone to human error. LLMs can automate parts of this process, offering numerous benefits. Firstly, their ability to swiftly process vast amounts of textual data enhances speed and efficiency, reducing the time technicians spend diagnosing problems. Moreover, LLMs contribute to enhanced decision-making within diagnostic teams by providing comprehensive insights into potential causes and implications of various vehicle symptoms. This enriched understanding empowers technicians to make more informed decisions regarding necessary repairs and maintenance, leading to more effective solutions. In addition, LLMs can serve as training tools for new technicians, granting access to a broad range of scenarios and corresponding expert analyses, bridging the knowledge gap between novice and experienced technicians. Lastly LLMs provide consistent and reliable interpretations of diagnostic data based on the best available knowledge, minimizing variability and errors that are typical in human diagnostics.

After the application of LLM for feature extraction tasks we came up with some practical examples of how LLMs can be used in the automotive industry:

- Automatically categorizing and prioritizing incoming text complaints from customers to streamline the diagnostic process.
- Generating detailed diagnostic reports based on textual descriptions provided by customers or technicians, highlighting potential issues and recommended actions.
- Providing real-time assistance to technicians during vehicle inspections, offering instant access to relevant technical documentation and expert advice.

5.3.3 Recommendation System

Recommendation System: The idea of investigating the ability of LLMs to answer questions based on contextual information, particularly through the use of RAG, came from a recognition of the challenges engineers face in investigating vehicle failures. Traditionally, this process involves scanning various sources and databases for relevant information, a task that is both time and energy consuming and heavily reliant on the engineer's knowledge and experience. Our initial exploration of business and data landscapes highlighted the abundance of detailed information, primarily in the form of technical documentations, within companies that could significantly aid engineers in problem-solving. However, accessing and interpreting this information is often hard and time-intensive. Considering this, we saw an opportunity to leverage the LLM's ability to search through these documents and retrieve detailed information quickly and efficiently, streamlining the problem-solving process for engineers.

It's important to note that our experimentation involved several simplifications, primarily due to restrictions on the use and publication of real data. To overcome this limitation, we created a synthetic context resembling real-world scenarios by establishing rules and generating synthetic data. While this approach is definitely limiting, particularly in capturing the full complexity of real-world data, it allowed for testing a methodology that could be explored more in the future.

5.4 Evaluation

In Section 4.4.3.1.1, we evaluate the recommendation system using the Humanely method. We ask the LLMs six input questions and then assess the generated output by answering the 16 criteria from the LLM evaluation metrics, which include relevance, coherence, coverage, harm, and comparison. Our assessment for these metrics is as follows:

- **Relevance:** Through the assessment of the generated output from the LLM, it is observed that for the majority of the questions, the responses are accurate, comprehensible, reasoned, and helpful to the user. However, for input question 5, the response is not relevant or helpful to the user. This specific response will be discussed in detail later.
- **Coverage:** We observed that for input questions 1 to 4, the LLM generates outputs that yield positive results in this metric. However, for input questions 5 and 6, the outputs do not meet the same standard.
- **Coherence:** In terms of fluency, grammar, and organization, the performance of the LLM is up to the mark.
- **Harm:** In our safety assessment of the LLM, we observed that it does not violate any privacy or introduce any toxicity. However, for a couple of questions, it exhibits some degree of hallucinations and bias.
- **Comparison:** The results in this metric are neutral, as it is difficult to distinguish the outputs from the LLM from those of a human being.

In our evaluation, we purposely asked input question 5 and question 6, which are very similar, to check if the LLM can handle variations in its input. Refer to Table 4.5 to see in detail what the questions and responses given by Claude v2 are. The difference between question 5 and question 6 is that question 5 asks about an upgrade of the battery when it is drained without referring to the electrical system in the context, whereas question 6 includes a specific reference to the electrical system. Assessment of the response shows that the LLM, without a specific reference to the electrical system, fails all evaluation metrics except for harm and coherence. This demonstrates that the LLM is sensitive to changes in how the question is asked and highlights the importance of carefully wording inputs to the LLM to obtain appropriate responses.

Through this assessment of the recommendation system, we answer our first research question, which deals with enhancing the vehicle diagnostic and troubleshooting process. LLMs offer the ability to retrieve information from their vast training knowledge or company-specific contexts with responses that are coherent, non-harmful, somewhat relevant, and cover most aspects of the reference context. This enables stakeholders to make decisions faster without manually searching through different datasets or documentation, though a key factor is well-structured documentation. The consistency of these responses can be improved by the methods discussed in Section 6.2.

In the text classification task, it was observed that as the data complexity increases from subset 1 to subset 2, traditional machine learning methods outperform LLMs.

For subset 1, large language models achieve an accuracy range of 70% to 77%, which is comparable to the performance of traditional methods. However, subset 2 contains categories with very similar complaints. For this subset, LLMs have an average accuracy of 45%, while traditional methods perform better. A key factor affecting performance in subset 2 is the feature extraction capability present in traditional methods, which is more effective than in LLMs. The main trade-off offered by LLMs is that they do not require model training on the dataset. Instead, creating well-thought-out prompts, which could be zero-shot or one-shot (see Section 2.2.2.4), enables faster deployment of a text classification system. However, as the texts become more similar, the LLMs struggle to capture critical keywords needed to differentiate categories, leading to less accurate classification. On the other hand, traditional methods involve training on those keywords, a time-consuming process, but one that helps to differentiate categories and make better predictions. This evaluation answers our second research question. We observe that traditional methods outperform LLMs when the dataset is complex but show comparable performance when the dataset lacks complexities, with traditional methods maintaining a slight edge. One possible reason for this behavior is that large language models are pre-trained on a large corpus of data, and their parameters are not readjusted to the specific data being fed to them. In contrast, traditional methods are trained on the input data and can capture key features that help them make distinctions

5.4.1 Related Existing Literature

In comparison to previous literature, a relevant article is "Smart Expert System: Large Language Models as Text Classifiers" by Zhiqiang Wang, Yiran Pang, and Yanbin Lin. They compared LLMs with traditional machine learning methods and deep neural networks across four datasets: the COVID-19-related Tweets Dataset, the Economic Texts Dataset, the E-commerce Texts Dataset, and the SMS Spam Collection Dataset. Their expert system framework involves collecting and preprocessing domain-specific data to build a comprehensive database, utilizing pre-trained language models like GPT-4 or LLaMA-3, and enhancing these models through fine-tuning or few-shot learning with domain-specific knowledge. Additionally, they optionally collaborate with experts to create prompts that further improve performance. The system's functionality and accuracy are evaluated to ensure quality, and an intuitive user interface allows non-experts to perform tasks such as classification and sentiment analysis, with an API facilitating real-time interaction. Their findings showed that while LLMs initially had low accuracy (36% to 55%) on the COVID dataset, performance improved with fine-tuning. For the other datasets, LLM accuracy exceeded 60% with or without fine-tuning. The key finding was that combining fine-tuning with expert-created prompts significantly enhances LLM performance as classifiers, allowing them to outperform traditional classification methods.

In our case (an automotive context), Claude LLMs' performance is comparable to traditional methods with some margin. For subset 1, the accuracy ranges from 70% to 77%. However, performance drops to 45% for subset 2. The key difference

between these subsets is the complexity of the data patterns. In subset 1, each complaint is mapped to a single category, while in subset 2, complaints could be mapped to one or multiple categories. Subset 2 was designed to mirror real-life scenarios where multiple subsystems of a vehicle may be responsible for a single complaint, thus increasing the complexity of the classification task. This difference in dataset complexity is a key distinction between the mentioned article and our methodology, complemented by the fact that they have used fine-tuning and domain expert-guided prompts to enhance the performance of LLMs.

Another relevant article that helps compare the LLM-based recommendation system in this thesis is "Analysis of Language-Model-Powered Chatbots for Query Resolution in PDF-Based Automotive Manuals." This research explores the adoption of Large Language Models (LLMs) to create AI-assisted tools for the automotive sector, assuming three distinct methods for comparative analysis: Doc Chatbot, Ask Your PDF, and Question and Answer System. These methods are supplemented by using zero-shot, one-shot, and few-shot prompts. To all three methods, four questions are asked, and it was observed that the responses obtained through the "Ask Your PDF" application were partially accurate, aligning precisely with the vehicle manual for two of the four questions when using the zero-shot prompt. For one-shot and few-shot prompts, this approach was the only one to accurately answer the second question across all prompt styles. Therefore, the LLMs do not have a consistent performance for all questions being asked. Similarly, in our work, we observe that the LLM is not able to give consistent answers for all questions, which can be attributed to its sensitivity to the structure of the questions being asked and the context being formed.

5.5 Deployment

Currently, in our work, we have developed a demo tool that corresponds to Technology Readiness Level (TRL) 6. TRL 6 involves testing the application in a relevant operational environment that closely represents the actual intended operational setting. This step is typically referred to as the 'pilot' stage. Activities at this level focus on determining whether the application can work, for whom, how, and under what conditions or contexts [46].

Moving up the TRL steps involves significant challenges and additional requirements. The subsequent levels are:

- **TRL 7:** Demonstration in a real-world environment. This level requires the integration of the demo application into a company's IT infrastructure and the expertise of the IT department. This step is beyond the scope of this thesis, as it involves complex integration tasks.
- **TRL 8:** Pre-release. At this stage, the application is near its final form and undergoes extensive testing and validation to ensure it meets all necessary standards and requirements.
- **TRL 9:** Release. This is the final stage, where the application is fully developed, tested, and ready for deployment to end-users.

The primary challenge in progressing to higher TRLs is that our demo application is currently developed and tested locally on a personal computer. Advancing to TRL

7 would necessitate integration into existing production systems, which requires collaboration between software engineers, infrastructure engineers, and applied AI engineers. This integration is critical to ensure that the application can function effectively within the company's IT environment.

To successfully move to TRL 7 and beyond, the following tools and methodologies are essential [47]:

- **Critical Scenario Tests:** These tests run use-case-specific scenarios and data slices to identify potential risks. A proper risk-quantification table should highlight these scenarios.
- **Golden Dataset:** A well-defined golden dataset should be used to baseline the performance of each model and track the performance over successive models.
- **Metamorphic Testing:** This software engineering methodology tests specific relationships between the outputs of multiple inputs. It is particularly useful for verifying and validating machine learning modules integrated into larger systems.
- **Data Intervention Tests:** These tests aim to identify data bugs at various points in the data pipelines. They measure the potential effects of data processing and machine learning on consumers or users of the data. Rather than relying solely on model performance as a proxy for data quality, these tests use specific mechanisms to catch data errors and ensure data validation.

6

Conclusion and Recommendation

6.1 Conclusion

This thesis aimed to explore the transformative potential of Large Language Models in the field of vehicle diagnostics and troubleshooting in the automobile industry. By addressing the major research topics, we have gained valuable insights into the capabilities and limitations of these advanced AI systems.

First, our findings showed that LLMs, with their extensive training on varied datasets, can offer meaningful and coherent responses to vehicle failure reports. This is a major improvement over traditional manual procedures, which are sometimes time-consuming and prone to errors. The use of retrieval-augmented generation, which provides LLMs with specific context from solid documentation, is an effective method but needs more evaluation and improvements. This strategy generated responses that were correct for most of the questions being asked and also very pertinent to the specific data at hand, demonstrating the potential for personalized diagnostic solutions. However, it is critical to note that the success of LLMs is heavily reliant on the quality and structure of the input queries, highlighting the importance of proper and careful query formulation.

When we compared LLMs to typical machine learning techniques for classifying vehicle malfunctions, we discovered some surprising results. While LLMs excel at managing straightforward information, traditional approaches frequently outperform them when dealing with more complicated datasets. This is in part due to their capacity to be fine-tuned and programmed to capture specific dataset properties. Despite the sophisticated reasoning capabilities and extensive pre-training of LLMs, traditional methods demonstrated superior performance in high-complexity scenarios. This highlights an important trade-off: LLMs offer broad, generalized reasoning capabilities, but classic machine learning approaches provide precision and specificity when adapted to specific tasks.

The implications of these findings are significant: for automotive industry stakeholders, the integration of LLMs into diagnostic systems promises to expedite operations and reduce diagnostic time, thereby significantly improving operational efficiency and satisfaction with clients. The ability of LLMs to quickly evaluate and interpret vast amounts of textual data has the potential to revolutionize automotive diagnostics, paving the path for more advanced and automated maintenance methods.

Looking ahead, this study lays a solid foundation for future research: there is immense potential to further enhance LLM capabilities, like explained in the following Recommendation Section. The integration of LLMs holds the promise of a new

era in automotive diagnostics, characterized by greater efficiency, accuracy, and automation. As we continue to innovate and refine these technologies, the future of vehicle diagnostics looks brighter than ever.

6.2 Recommendation

Given the limited research on large language models in the automotive industry, we focused more on the methodology and used datasets that are small in size. To avoid security issues and GDPR violations, we utilized a public dataset that closely resembles the company's real dataset. For future research, it would be beneficial to test the methodology on a larger, real dataset. The real dataset and technical documentation are more extensive, dispersed across different locations within the company, and more complex. It would be interesting to evaluate how this methodology performs in such a real-life scenario. Security can be ensured by using cloud services like Amazon Web Services, which offer a secure environment to prevent breaches, as discussed in the theory section. Future studies should investigate how these cloud computing services maintain security, although this is beyond the scope of this thesis.

As trucks become more complex with additional systems and subsystems, it is crucial to define well-structured documents that describe the product architecture, as outlined in the theory section. This should be done at the very beginning of product development so that an LLM model can be built around these documents, enabling faster decision-making for the operator.

From a machine learning perspective, it would be valuable to explore how instruction tuning, as discussed in Section 2.2.2.2, can be applied in an automotive context. Currently, there are no datasets with instructions or prompts tailored to improve zero-shot learning for the automotive industry, as failure descriptions vary widely. Creating a dataset with diverse prompts could significantly enhance the performance of question-answering systems.

Another promising aspect is the inclusion of LLM-powered agents, as discussed in Section 2.2.2.5, which could handle dynamic tasks more effectively. Previous literature indicates that using few-shot prompts, along with domain expert-guided prompts and fine-tuning, significantly enhances the performance of LLMs. With such an approach, LLMs could potentially outperform traditional methods for complex datasets in the automotive industry.

Based on discussions with stakeholders, it would be more feasible to break down the classification and recommendation system into smaller subsystems, each targeting a specific domain within the field quality process. For example, one subsystem could focus on repair and related information, while another could target upgrades and the information related to that.

Bibliography

- [1] Sayed Suhaib Kamran et al. “Artificial intelligence and advanced materials in automotive industry: Potential applications and perspectives”. In: *Materials Today: Proceedings* (2022). DOI: 10.1016/j.matpr.2022.04.727. URL: <https://doi.org/10.1016/j.matpr.2022.04.727>.
- [2] A. B. Nassif et al. “Speech Recognition Using Deep Neural Networks: A Systematic Review”. In: *IEEE Access* 7 (2019), pp. 19143–19165. DOI: 10.1109/ACCESS.2019.2896880.
- [3] J. Dalzochio et al. “Machine Learning and Reasoning for Predictive Maintenance in Industry 4.0: Current Status and Challenges”. In: *Computers in Industry* 123 (2020), p. 103298. DOI: 10.1016/j.compind.2020.103298.
- [4] Umar Zakir Abdul Hamid et al. “A Review on Threat Assessment, Path Planning and Path Tracking Strategies for Collision Avoidance Systems of Autonomous Vehicles”. In: *International Journal of Vehicle Autonomous Systems* 14 (2018). Published online: November 14, 2018, pp. 134–169. DOI: 10.1504/IJVAS.2018.096154. URL: <https://doi.org/10.1504/IJVAS.2018.096154>.
- [5] M. Al-Zeyadi et al. “Deep Learning Towards Intelligent Vehicle Fault Diagnosis”. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. Glasgow, UK, 2020, pp. 1–7. DOI: 10.1109/IJCNN48605.2020.9206972.
- [6] L. Huang and Y. L. Murphey. “Text Mining with Application to Engineering Diagnostics”. In: *Advances in Applied Artificial Intelligence*. Ed. by M. Ali and R. Dapoigny. Vol. 4031. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006. DOI: 10.1007/11779568_138. URL: https://doi.org/10.1007/11779568_138.
- [7] Chao Lv. “Research on Intelligent Decision Support System for Automobile Fault Diagnosis Based on SWOT Analysis”. In: *Proceedings of the 2021 2nd International Conference on Artificial Intelligence and Information Systems*. Chongqing, China: ACM, May 28–30, 2021, pp. 1–4. DOI: 10.1145/3469213.3471312. URL: <https://doi.org/10.1145/3469213.3471312>.
- [8] Antoine Richard et al. “What does it mean to provide decision support to a responsible and competent expert?: The case of diagnostic decision support systems”. In: *EURO Journal on Decision Processes* 8.3-4 (2020), pp. 205–236. ISSN: 2193-9438. DOI: 10.1007/s40070-020-00116-7. URL: <https://www.sciencedirect.com/science/article/pii/S2193943821001151>.
- [9] Loredana Caruccio et al. “Can ChatGPT provide intelligent diagnoses? A comparative study between predictive models and ChatGPT to define a new medical diagnostic bot”. In: *Expert Systems with Applications* 235 (2024),

- p. 121186. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2023.121186. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423016883>.
- [10] Andreas Theissler et al. “Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry”. In: *Reliability Engineering System Safety* 215 (2021), p. 107864. ISSN: 0951-8320. DOI: 10.1016/j.res.2021.107864. URL: <https://www.sciencedirect.com/science/article/pii/S0951832021003835>.
- [11] D. Williamsson. *Modularisation and Product Description*. 2018. URL: <https://kth.diva-portal.org/smash/get/diva2:1256202/FULLTEXT01.pdf>.
- [12] M. Björk and E. Hällfors. *The Concept of Modularization and the Definition of a Module*. 2015. URL: <http://www.diva-portal.se/smash/get/diva2:840182/FULLTEXT01.pdf>.
- [13] H-J. Bullinger, F. Fremerey, and J. Fuhrberg-Baumann. “Innovative production structures—Precondition for a customer-orientated production management”. In: *International Journal of Production Economics* 41.1-3 (1995), pp. 15–22.
- [14] J. Jiao and M M. Tseng. “A methodology of developing product family architecture for mass customization”. In: *Journal of Intelligent Manufacturing* 10.1 (1999), pp. 3–20.
- [15] A. Jakhotiya et al. “Text Pre-Processing Techniques in Natural Language Processing: A Review”. In: *International Research Journal of Engineering and Technology (IRJET)* 9 (2 2022). URL: <https://www.irjet.net/archives/V9/i2/IRJET-V9I2150.pdf>.
- [16] Claude Sammut and Geoffrey I. Webb. “TF-IDF”. In: *Encyclopedia of Machine Learning*. Boston, MA: Springer, 2011, pp. 986–987. DOI: 10.1007/978-0-387-30164-8_832. URL: https://doi.org/10.1007/978-0-387-30164-8_832.
- [17] F.-J. Yang. “An Implementation of Naive Bayes Classifier”. In: *Proceedings of the IEEE International Conference*. Florida Polytechnic University. Lakeland, Florida, USA, 2018. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=8947658>.
- [18] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. Sebastopol, CA: O’Reilly Media, Inc, 2018, pp. 245–246.
- [19] Zheng Wang Sho Xu Yan Li. “Bayes Multinomial Naive Bayes Classifier to Text Classification”. In: *Journal of Cheminformatics* (2017). DOI: 10.1007/978-981-10-5041-1_57.
- [20] Ajay R. Dani Siva RamaKrishna Reddy V D V L N. Somayajulu. “Classification of Movie Reviews Usin Complemented Naive Bayesian Classifier”. In: *International Journal of Intelligent Computing Research* 2 (2011).
- [21] Y. Zhang. “Support Vector Machine Classification Algorithm and Its Application”. In: *Information Computing and Applications*. Ed. by C. Liu, L. Wang, and A. Yang. Vol. 308. Communications in Computer and Information Science. Berlin, Heidelberg: Springer, 2012. DOI: 10.1007/978-3-642-34041-3_27.
- [22] S.M. Dhawan B.M. Gupta and Ghouse Modin Mamdapur. “SUPPORT VECTOR MACHINE (SVM)RESEARCH ININDIA: A SCIENTOMETRIC EVALUATION OF INDIA’S PUBLICATIONS OUTPUT DURING 2002-19”. In:

- JOURNAL OF INDIAN LIBRARY ASSOCIATION*. 2021. URL: <https://ilaindia.net/jila/index.php/jila/article/viewFile/1261/273>.
- [23] Thorsten Joachims. "Text Categorization with Support Vector Machines: Learning with Many Relevant Features". In: Universitat Dortmund, Germany: Springer. URL: https://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf.
- [24] Suhana Sulaiman Muhammad Hareez Mohd Zaki Mohd Azri Abdul Aziz and Najidah Hambali. "Student Performance Classification using Support Vector Machine (SVM) with Polynomical Kernel on Online Student Activities". In: *JOURNAL OF ELECTRICAL AND ELECTRONIC SYSTEMS RESEARCH* (). DOI: <https://doi.org/10.24191/jeesr.v23i1.009>.
- [25] Shih-Lin Lin. "Application of Machine Learning to a Medium Gaussian Support Vector Machine in the Diagnosis of Motor Bearing Faults". In: *Electronics* (2021). DOI: <https://doi.org/10.3390/electronics10182266>.
- [26] Pradeep K Sinha Vrushali Y Kulkarni. "Effective Learning and Classification using Random Forest Algorithm". In: *International Journal of Engineering and Innovative Technology (IJEIT)*. Vol. 3. 2014. URL: https://www.ijeit.com/Vol%203/Issue%2011/IJEIT1412201405_47.pdf.
- [27] Marcelo Becker Joao Manoel Herrera Pinheiro. "Breast Cancer Classification Using Gradient Boosting Algorithms Focusing on Reducing the False Negative and SHAP for Explainability". In: *arXiv*. Vol. 2403.09548v1. 2024. URL: <https://arxiv.org/pdf/2403.09548v1>.
- [28] Kashvi Taunk et al. "A Brief Review of Nearest Neighbor Algorithm for Learning and Classification". In: *Proceedings of the International Conference on Intelligent Computing and Control Systems* (2019). DOI: 10.1109/ICCS45141.2019.9065747.
- [29] Ruliang Xiao Faliang Huang Guoqing Xie. "Research on Ensemble Learning". In: *International Conference on Artificial Intelligence and Computational Intelligence*. 2009. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5376633>.
- [30] H. Naveed et al. *A Comprehensive Overview of Large Language Models*. 2023. URL: <https://arxiv.org/pdf/2307.06435.pdf>.
- [31] J. Wei et al. "Fine-Tuned Language Models Are Zero-Shot Learners". In: *ICLR*. 2022. URL: <https://arxiv.org/pdf/2109.01652.pdf>.
- [32] E. Derner et al. *A Security Risk Taxonomy for Large Language Models*. 2023. URL: <https://arxiv.org/pdf/2311.11415.pdf>.
- [33] Raghav Awasthi et al. "HumanELY: Human evaluation of LLM yield, using a novel web-based evaluation tool". In: *MedRXIV* (2023). DOI: 10.1101/2023.12.22.23300458.
- [34] Anthropic. *The Claude 3 Model Family: Opus, Sonnet, Haiku*. 2024.
- [35] Anthropic. "Model Card and Evaluations for Claude Models". In: URL: <https://www-cdn.anthropic.com/bd2a28d2535bfb0494cc8e2a3bf135d2e7523226/Model-Card-Claude-2.pdf>.
- [36] R. Wirth and J. Hipp. *CRISP-DM: Towards a Standard Process Model for Data Mining*. Available online. URL: <http://cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf>.

- [37] T. Lindewall. *Key Performance Indicators for The Product Follow-Up Process at Scania*. Available online. Year. URL: https://www.utn.uu.se/sts/student/wp-content/uploads/gamla%20exjobb/0812_lindewall.pdf.
- [38] A. Rydberg. *Perspectives on Producing High-Quality Technical Documentation*. Available online. Year. URL: <https://www.diva-portal.org/smash/get/diva2:941576/FULLTEXT01.pdf>.
- [39] J. C. Mankins. *Technology Readiness Levels*. White Paper. Edited on 22 December 2004. Advanced Concepts Office, Office of Space Access and Technology, NASA, 1995.
- [40] National Highway Traffic Safety Administration (NHTSA). *NHTSA Complaints Database*. Online. Accessed: April 2024. 1995-2024. URL: https://static.nhtsa.gov/odi/ffdd/cmpl/FLAT_CMPL.zip.
- [41] National Highway Traffic Safety Administration (NHTSA). *NHTSA Complaints Database - Main Field Descriptions*. Online. Accessed: April 2024. 2021. URL: <https://static.nhtsa.gov/odi/ffdd/cmpl/CMPL.txt>.
- [42] Petro Liashchynskyi and Pavlo Liashchynskyi. “Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS”. In: *CoRR* abs/1912.06059 (2019). arXiv: 1912.06059. URL: <http://arxiv.org/abs/1912.06059>.
- [43] Damjan Krstajic et al. “Cross-validation pitfalls when selecting and assessing regression and classification models”. In: *Journal of Cheminformatics* 6.1 (2014), pp. 1–15.
- [44] Manuel Fernández-Delgado et al. “Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?” In: *Journal of Machine Learning Research* 15 (2014). Ed. by Russ Greiner. Submitted 11/13; Revised 4/14; Published 10/14, pp. 3133–3181.
- [45] Rajni Jindal, Ruchika Malhotra, and Abha Jain. “Techniques for Text Classification: Literature Review and Current Trends”. In: *Delhi Technological University* ().
- [46] Luis Salvador-Carulla et al. “Adaptation of the technology readiness levels for impact assessment in implementation sciences: The TRL-IS checklist”. In: *Heliyon* (2024). DOI: 10.1016/j.heliyon.2024.e29930.
- [47] Arthur Lavin, Claire M. Gilligan-Lee, and Alexander Visnjic. “Technology readiness levels for machine learning systems”. In: *Nature Communications* (2022). DOI: 10.1038/s41467-022-33128-9.

A

Appendix 1

Listing A.1: Prompt 1 - Subset 1

```
f """
```

Human:

You are an expert field quality engineer who have to classify descriptions of car failures into specific categories based on detailed descriptions of symptoms and failures. Please analyze the <complaint> described below and classify it by returning only the appropriate <category> label.

Complaint: "{text}"

You must choose only from the following categories:

- 'ELECTRICAL SYSTEM'*
- 'AIR BAGS'*
- 'WHEELS'*
- 'ENGINE AND ENGINE COOLING'*
- 'STRUCTURE'*

Choose the most relevant category based on the complaint described.

Return only the category label. Nothing else is required."

```
Assistant: """
```

Listing A.2: Prompt 2

```
f """
```

Human:

You are an expert field quality engineer who have to classify descriptions of car failures into specific categories based on detailed descriptions of symptoms and failures. Please analyze the <complaint> described below and classify it by returning only the appropriate <category> label.

<complaint>: "{text}"

You must choose only from the following categories, each described to assist your decision-making:

- 'ELECTRICAL SYSTEM': Involves components like batteries, alternators, wiring, and electronic controllers. Common issues include dead batteries, failed alternators, faulty wiring, and malfunctioning*

A. Appendix 1

sensors or fuses, leading to symptoms such as the engine not starting, dimming lights, and failing electronic accessories.

- *'AIR BAGS': Concerns the vehicle's airbag system, including sensors, deployment mechanisms, and control modules. Issues may involve airbags not deploying in an accident, deploying unexpectedly, or error messages related to airbag systems.*
- *'WHEELS': Relates to problems with the vehicle's wheels, such as issues with wheel balance, alignment, structural integrity, and abnormal tire wear. Complaints might include vibrations at certain speeds, uneven tire wear, or visible damage to wheels.*
- *'ENGINE AND ENGINE COOLING': Covers the engine and its cooling systems. This includes overheating, coolant leaks, radiator problems, and inefficient cooling leading to engine failure. Issues might also cover irregular engine temperatures or failed thermostats.*
- *'STRUCTURE': Involves the integral structure of the vehicle, including the chassis, body integrity, and issues related to rust or decay of vehicle frame. Structural concerns may manifest in abnormal creaking, leaks through seals, or visible corrosion and damage affecting vehicle safety and integrity.*

Choose the most relevant category based on the complaint described.

Return only the category label (maximum two words). Nothing else is required. "

Assistant: ""

Listing A.3: Prompt 3

```
f ""
```

Human:

You are an expert field quality engineer who have to classify descriptions of car failures into specific categories based on detailed descriptions of symptoms and failures. Please analyze the <complaint> described below and classify it by returning only the appropriate <category> label.

<complaint>: "{text}"

You must choose only from the following categories, each described to assist your decision-making:

- *'ELECTRICAL SYSTEM': Involves components like batteries, alternators, wiring, and electronic controllers. Common issues include dead batteries, failed alternators, faulty wiring, and malfunctioning sensors or fuses, leading to symptoms such as the engine not starting, dimming lights, and failing electronic accessories. An example of 'ELECTRICAL SYSTEM' complaint is: "Started truck with remote start. Truck stopped normally after a set time. After entering truck and restarting, the check engine light was indicated. Message to take truck for service. Dealer kept the truck for diagnosis. 28 codes were indicated and reset. Battery*

showed a low charge and was charged to proper voltage. Truck was returned with a list of all codes that were addressed. I have Low Jack on my truck and on Oct 08 2023 that I had low voltage of 11 indicated again. Will be visiting dealer for follow up."

- *'AIR BAGS': Concerns the vehicle's airbag system, including sensors, deployment mechanisms, and control modules. Issues may involve airbags not deploying in an accident, deploying unexpectedly, or error messages related to airbag systems.*

An example of 'AIR BAGS' complaint is: "Driver Airbag light started flashing. According to the vehicle manufacturer's manual, this indicator means the drivers' airbag will NOT deploy. The battery was unplugged and left in an attempt to reset. When key turned in ignition, the airbag light immediately started flashing again. I am the only owner of this vehicle. I bought it brand new and it has never been in an accident where the airbags have deployed. The vehicle has been garage kept for the duration of my ownership. I have not had any water damage, and I have not had any other electrical issues so far. "

- *'WHEELS': Relates to problems with the vehicle's wheels, such as issues with wheel balance, alignment, structural integrity, and abnormal tire wear. Complaints might include vibrations at certain speeds, uneven tire wear, or visible damage to wheels.*

An example of 'WHEELS' complaint is: "Damaged lug nuts. Dealer told me they had to be replaced. Front are like new. Back wheel lug nuts have swelled and are starting to rust. Dealer cannot rotate my tires. My last Edge did not have this problem and it was older."

- *'ENGINE AND ENGINE COOLING': Covers the engine and its cooling systems. This includes overheating, coolant leaks, radiator problems, and inefficient cooling leading to engine failure. Issues might also cover irregular engine temperatures or failed thermostats.*

An example of 'ENGINE AND ENGINE COOLING' complaint is: "my car randomly would not start upon shutting car off and returning to vehicle the car would not start with the keys that was given to me from the dealership that i purchased the vehicle from. it has been having coolant issues where it was reading high temperature then would go back to normal operating temperature very shortly and would happen off and on. it was serviced for this issue at Boubin Automotive in Waterloo, Iowa and they stated that it was empty on coolant refilled it and seemed to not have any leaks and it drove fine for a few weeks and then had the coolant issue where the car would read high temperature then would go back to normal and kept doing so. The day i parked it and returned to it and it would not start it showed high temperature immediately and would not start. had it towed to a mechanic shop habhab's towing in ames iowa where they stated there was no coolant in it once again and they pressure checked the vehicle and it passed the pressure test for the motor but had no coolant in it, and still would not start stating that the pcm is bad in the vehicle. it has shown the exact signs of the recall in this vehicle that ford has released and i am trying to

A. Appendix 1

get the car covered under the recall because its showing the exact same issues as the new recall that was put out. "

- *'STRUCTURE': Involves the integral structure of the vehicle, including the chassis, body integrity, and issues related to rust or decay of vehicle frame. Structural concerns may manifest in abnormal creaking, leaks through seals, or visible corrosion and damage affecting vehicle safety and integrity.*

An example of 'STRUCTURE' complaint is: "The rear frame that the trailer hitch is attached to has cracked and trailer hitch began to fail.the hitch metal also cracked and the hitch started to fall down,dropping the front of the trailer.if the trailer hitch dropped of completely,the safety devices of the trailer would not have worked. Since the safety chains and emergency brake cable are attached to the hitch,the trailer would be free wheeling down the highway. This would then probably hit another vehicle possibly causing injury or death. "

Choose the most relevant category based on the complaint described.

Return only the category label (maximum two words). Nothing else is required. "

Assistant: ""

Listing A.4: Prompt for extracting information

f '''

Human:

You are an expert field quality engineer who have to analyse descriptions of vehicle failures. Your work consists in reading the <complaint> and succesfully completing 5 tasks that are here explained:

1. *Based on the <complaint> described, predict the most appropriate category of the failure. You must choose only from the following categories, each described to assist your decision-making:*

- *'ELECTRICAL SYSTEM': Involves components like batteries, alternators, wiring, and electronic controllers. Common issues include dead batteries, failed alternators, faulty wiring, and malfunctioning sensors or fuses, leading to symptoms such as the engine not starting, dimming lights, and failing electronic accessories.*
- *'AIR BAGS': Concerns the vehicle's airbag system, including sensors, deployment mechanisms, and control modules. Issues may involve airbags not deploying in an accident, deploying unexpectedly, or error messages related to airbag systems.*
- *'WHEELS': Relates to problems with the vehicle's wheels, such as issues with wheel balance, alignment, structural integrity, and abnormal tire wear. Complaints might include vibrations at certain speeds, uneven tire wear, or visible damage to wheels.*

- *'ENGINE AND ENGINE COOLING': Covers the engine and its cooling systems. This includes overheating, coolant leaks, radiator problems, and inefficient cooling leading to engine failure. Issues might also cover irregular engine temperatures or failed thermostats.*
- *'STRUCTURE': Involves the integral structure of the vehicle, including the chassis, body integrity, and issues related to rust or decay of vehicle frame. Structural concerns may manifest in abnormal creaking, leaks through seals, or visible corrosion and damage affecting vehicle safety and integrity.*

2. *Describe any observable <symptoms>: mention any <symptoms> the vehicle is exhibiting. These might include:*
- *strange noises (e.g., clicking, grinding, or whining)*
 - *visual symptoms (e.g., smoke, leaking fluid)*
 - *operational issues (e.g., the vehicle not starting, misfiring engine, stalling)*
 - *warning lights that are on (e.g., check engine light, oil pressure light)*

The symptom description must be as detailed as possible. If you are not able to answer, just output: 'Not Available'.

3. *Mention when the <symptoms> occur: indicate if the symptoms occur under specific <conditions> (for example during start-up, while braking, at high speeds, or during engine idling).*

The condition description must be as detailed as possible. If you are not able to answer, just output: 'Not Available'.

4. *Based on the <symptoms> and <conditions>, identify what are the possible <causes>: inquire about what might be causing these <symptoms>.*

The causes description must be as detailed as possible. If you are not able to answer, just output: 'Not Available'.

5. *Based on the <symptoms>, <conditions> and <causes>, identify what are the possible <actions> that can be taken: inquire about what are the steps to do in order to fix the issue.*

The possible actions description must be as detailed as possible. If you are not able to answer, just output: 'Not Available'.

The <complaint> to analyse is this: "{text}"

Assistant: '''

Listing A.5: Prompt for Recommendation System

```
f """  
Use the following pieces of context to answer the question at the end.  
If you don't know the answer, just say that you don't know, don't  
try to make up an answer. Don't include harmful content.  
  
{context}  
  
Question: {question}  
Answer: ""  
"
```

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY