CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Personalized AI Driver Coaching in Heavy-Duty Trucks for Fuel-Efficient Driving

From Vehicle Telemetry to Eco-Driving Feedback through SHAP and Retrieval-Augmented LLMs

Master's thesis in Computer Science and Engineering

JOSEFINE KARLSSON

KASPER LUNDBERG

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2026

Master's thesis 2026

Personalized AI Driver Coaching in Heavy-Duty Trucks for Fuel-Efficient Driving

From Vehicle Telemetry to Eco-Driving Feedback through SHAP and Retrieval-Augmented LLMs

JOSEFINE KARLSSON KASPER LUNDBERG

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2026

Personalized AI Driver Coaching in Heavy-Duty Trucks for Fuel-Efficient Driving From Vehicle Telemetry to Eco-Driving Feedback through SHAP and Retrieval-Augmented LLMs
JOSEFINE KARLSSON, KASPER LUNDBERG

Supervisor: Xuechen (Hugh) Liu, Computer Science and Engineering
Advisor: Anders Eriksson, Volvo Group
Examiner: Kivanç Tatar, Computer Science and Engineering

Master's Thesis 2026
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2026

Personalized AI Driver Coaching in Heavy-Duty Trucks for Fuel-Efficient Driving From Vehicle Telemetry to Eco-Driving Feedback through SHAP and Retrieval-Augmented LLMs

JOSEFINE KARLSSON, KASPER LUNDBERG
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Fuel-efficient driving is important for reducing both operational costs and environmental impact in heavy-duty transport. Although existing driver coaching systems can support eco-driving, they often rely on predefined rules, generic feedback, or numerical scores that provide limited explanation of how drivers can improve. This thesis investigates how vehicle telemetry data can be used to generate personalized eco-driving feedback through an AI-based conversational driver coaching prototype.

Vehicle telemetry data from Volvo heavy-duty trucks operating in the Latin American market was analysed within two operational segments. Segment-specific LightGBM models were trained to predict fuel efficiency from driving behaviour variables, while SHAP values were used to identify the behaviours that most strongly influenced each prediction in each segment-specific model. These explanations were then translated into conversational coaching feedback using a large language model. To support trustworthy and personalized interaction, the prototype further incorporated retrieval-augmented generation for domain grounding and a persistent semantic memory layer for adapting responses to user-specific preferences, goals, and constraints.

The modelling results show that features related to engine torque usage, driving speed, and acceleration behaviour influence predicted fuel efficiency the most, with some variation between operational segments. The LightGBM models outperformed a ridge regression baseline, achieving mean cross-validated R^2 scores of 0.796 and 0.744 for the two analysed segments. SHAP analysis further showed that segment-specific models can interpret the same driving behaviour differently, supporting the relevance of context-aware modelling.

The conversational prototype was evaluated through an expert questionnaire with participants from Volvo Group. The results indicate that personalized and context-aware responses were generally preferred over non-personalized alternatives. However, the evaluation also showed that retrieval-augmented generation did not consistently improve perceived response quality, partly due to overly detailed or technically ambiguous feedback. These findings highlight both the potential and the challenges of combining explainable machine learning and large language models for driver coaching. Overall, the thesis demonstrates that telemetry-based explanations can be transformed into personalized conversational feedback, while emphasizing the need for careful grounding, concise presentation, and domain-specific validation.

Keywords: conversational AI, SHAP, heavy-duty trucks, machine learning, personalization, driver coaching.

Acknowledgements

We would first like to express our deepest gratitude to our academic supervisor, Dr. Xuechen (Hugh) Liu, for his invaluable guidance and constant support throughout this project. His enthusiasm and optimism have been greatly appreciated. In the spirit of Dr. William S. Clarks words, "Boys, be ambitious!", he encouraged us to aim high and to keep pushing the work forward!

A very big thanks to our advisor Anders Eriksson, who helped us throughout the thesis and to navigate through the "corporate jungle"! Tack!

We would also like to thank our examiner Dr. Kivanç Tatar for his continuous feedback on the project, helping to guide us and giving us insightful feedback.

Additionally, a special thanks goes out to the domain experts who participated in our evaluation questionnaire. Their insights were important for our results. And a big thanks to everyone else at Volvo who helped us throughout the thesis!

Last, but not least, we want to thank Juliana Viscenheski. Her help and guidance from start to finish of the project has been invaluable. Without her, we would not have been able to complete the project. We owe you many semlor! Obrigado!

Josefine Karlsson & Kasper Lundberg, Gothenburg, 2026-06-08

Contents

List of Acronyms

ADAS Advanced Driver Assistance Systems. 8, 9

AI Artificial Intelligence. 1–3, 7–11, 13, 15, 27, 31

API Application Programming Interface. 5, 24

CoT Chain of Thought. 26, 27, 50

GCW Gross Combined Weight. 32

HELM Holistic Evaluation of Language Models. 48

IR Information Retrieval. 22, 23

KPIs Key Performance Indicators. 3

LightGBM Light Gradient Boosting Machine. 37, 42, 43, 59, 60, 78

LIME Local Interpretable Model-Agnostic Explanations. 18

LLM Large Language Model. 1–4, 10, 15, 20–28, 45, 47, 48, 50–55, 70, 71, 78–80

MAE Mean Absolute Error. 41, 59, 60

ML Machine Learning. 3, 4, 10, 11, 13, 15, 16, 19, 20

MTEB Massive Text Embedding Benchmark. 46

NLP Natural Language Processing. 10, 11, 20, 55

RAG Retrieval-Augmented Generation. 1, 3, 4, 15, 22–25, 45, 46, 56, 66–71, 78–80

SHAP SHapley Additive exPlanations. 11, 18–20, 37, 42–45, 47, 49–51, 53, 54, 59–63, 76–78

UX User Experience. 7, 53

List of Figures

List of Tables

1

Introduction

Energy-efficient driving is an important focus area for both transport companies and society at large. Fuel consumption, vehicle wear and driving style have substantial environmental and economic consequences, particularly for heavy-duty vehicles operated in commercial transport [1]. In this context, energy-efficient driving is often referred to as eco-driving, encompassing driving practices aimed at reducing fuel consumption and emissions [2]. As a result, many fleets deploy driver coaching systems to help drivers adopt more energy-efficient driving habits. These systems aim to translate vehicle data into actionable insights that support sustainable driving and reduce operational costs.

Despite their widespread use, much of the existing research on eco-driving feedback and driver coaching systems is grounded in predefined, rule-based strategies that provide largely generic guidance [2]. Previous studies showed that drivers often perceive existing coaches as unclear, impersonal and disconnected from real driving situations [3]. As a consequence, coaching systems are frequently ignored and thus fail to create meaningful long-term behaviour change [4]. This highlights a gap between the potential of driver coaching and the actual application in practice of current systems.

Advances in Artificial Intelligence (AI), particularly Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), and data-driven behaviour modelling methods such as clustering, create new opportunities for more interactive and personalized driver coaching. In this context, the driving situation is represented through observable vehicle and trip-level data, such as speed profiles, acceleration, braking patterns and operational context. Clustering can be used to identify recurring patterns or segments in driving behaviour, while LLMs can generate conversational feedback grounded through RAG in authoritative sources such as vehicle manuals or driving guidelines.

By clustering drivers based on their historical driving behaviour, the system can interpret their specific conditions, such as their typical driving context. By improving the relevance and clarity of the feedback through such data-driven personalization, the system may increase driver engagement and willingness to follow recommendations, potentially supporting more sustainable driving over time.

However, there is limited research exploring how LLM-based systems can be integrated with vehicle data and personalization strategies to form a practical and

trustworthy driver coach. Existing work often focuses on rule-based eco-driving feedback, leaving a gap in interactive, LLM-driven coaching systems [2], [5].

This thesis aims to address this gap by investigating how AI can be used in a driver coach to interpret vehicle telemetry, profile individual drivers and generate personalized and contextually grounded post-drive feedback to enhance energy-efficient driving. The overarching goal is thus to develop and evaluate an AI-assisted off-board driver coach application, and was conducted in collaboration with Volvo Group.

1.1 Research gap

While there are studies exploring the usage of personalized driver coaching, there appears to be a gap in current academic research. Many of published papers are within applications related to the auto-mobile industry (light vehicles) [6], overlooking the specific needs of the heavy trucking sector. Furthermore, while AI is already a powerful tool in route planning and scheduling within logistics [7], we believe its potential is underutilized in the general coaching of drivers' actual driving behaviour and its possibility to personalize its output.

1.1.1 Heavy-Duty Trucks as a Target Domain

Heavy-duty trucks operate under conditions that differ substantially from those of passenger vehicles, both in terms of vehicle dynamics and operational context. Their large mass, limited manoeuvrability and long braking distances mean that driving behaviour has a pronounced impact on fuel consumption [8]. Small variations in e.g. acceleration or speed can lead to significant differences in fuel efficiency and emissions, which makes driving behaviour particularly important in commercial transport. This is further underscored by global estimates indicating that heavy-duty trucks account for a disproportionately large share of road transport emissions, contributing around 25% of global road emissions despite representing only 1% of the total vehicle fleet [9].

In addition, truck drivers operate in professional settings that impose specific constraints and requirements. Long driving hours, repetitive routes, time pressure and varying operating environments shape both driving behaviour and the conditions under which feedback can be delivered [10]. As a result, coaching systems must be careful not to provide guidance that is poorly aligned with the current driving context.

These characteristics place higher demands on driver coaching systems in the heavy-duty domain. Feedback must be adaptable and trustworthy to different operational settings, as generic advice risks being ignored or even counterproductive. Consequently, methods developed for passenger vehicles may not directly transfer to heavy-duty trucks, which motivates a dedicated investigation of AI-driven, personalized driver coaching in this domain.

1.1.2 Personalized Driver Coaching Research

Studies have established that personalized and context-aware driver coaching shows positive user feedback, suggesting drivers prefer customized guidance over generic warnings [11], [12]. The potential for better communication has also been demonstrated, such as the improved interpretability achieved by the usage of conversational AI chatbots for post-driving feedback [13]. Yet, some of this research is limited to niche areas, such as performance driving on race tracks, offering limited applicability to commercial logistics contexts.

The historical application of Machine Learning (ML) and AI within driver coaching has predominantly relied on threshold-based and predictive classification models. Earlier systems often utilized predefined benchmarks to flag deviations such as speeding [14]. More advanced models incorporated the usage of Bayesian models or probabilistic methods to handle more complex feedback [11], [15], [16]. Furthermore, algorithms such as support vector machines have been successfully employed to assess individual driver safety based on historical telematics [13]. While these traditional approaches performed well at pattern recognition and risk assessment, they typically output numerical scores or alerts rather than human-readable and actionable advice.

Furthermore, existing research generally does not take driver profile or behaviour into context, often treating truck drivers as a relatively homogenous population [11]. We believe that this could be an oversight, as the optimal driving behaviour for example for a heavy-duty mining truck driving on uneven terrain differs vastly from that of a long-haul driver.

Finally, many existing systems provide rigid or opaque instructions, whereas conversational explanations have been shown to improve clarity and driver engagement [11]. Yet these interpretability benefits have rarely been combined with personalized behaviour modelling in the heavy-duty trucking domain.

Therefore, although individual components have been explored, no existing work integrates (1) heavy-duty telematics-based segmentation, (2) LLM-generated conversational feedback, and (3) RAG-based factual grounding into a unified driver coaching system.

1.2 Problem

Current driver coaching systems used in commercial transport are predominantly based on predefined Key Performance Indicators (KPIs) and generalized feedback strategies [2], [5]. While these systems are designed to encourage eco-driving, research shows that drivers often experience them as repetitive and irrelevant [3]. As a result, many drivers ignore or disable such systems, undermining their intended purpose and eliminating potential environmental and economic benefits [17]. Static KPIs-based coaches are unable to adapt to individual differences in driving style, experience level and preferences.

In Volvo trucks, the current driver coach provides a numerical score intended to reflect driving performance [18]. However, the underlying logic of these calculations remains opaque, providing drivers with neither the context of specific events nor actionable feedback on how to improve.

At the same time, modern vehicles generate large volumes of telemetry data that could support more dynamic, personalized coaching if interpreted effectively. Advances in ML and LLMs offer new opportunities to analyze complex data patterns, model driver-specific behaviour, and generate conversational context-aware feedback. However, little is known about how these technologies can be integrated into a coaching system.

The core technical problem is therefore to design a system that can:

- Interpret and model driver behaviour through utilizing available data clusters and driving data at Volvo Group
- Adapt feedback to individual trucks and contexts through personalization techniques, and
- Deliver trustworthy and conversational factually grounded coaching by utilizing LLMs and RAG

1.3 Research questions

The aim of this thesis is to investigate how vehicle telemetry data can be utilized to generate personalized eco-driving feedback through an AI-based conversational driver coaching system. To address this overarching objective, three supporting research questions are formulated, each targeting a specific aspect of the problem.

RQ1: Which driving behaviour variables most strongly influence fuel efficiency in different operational segments of heavy-duty truck fleet data?

RQ2: What context-aware eco-driving feedback can be derived from driving behaviour insights related to fuel efficiency?

RQ3: What framework can enable the delivery of factually grounded and personalized eco-driving feedback in a conversational AI prototype?

1.4 Purpose

The purpose of this thesis is to investigate how data-driven methods, LLM and RAG can be combined to provide personalized eco-driving feedback for heavy-duty truck drivers. Vehicle telemetry data is analysed utilizing pre-existing clusters to segment driver behaviour, alongside explainable ML models to isolate the driving parameters that most strongly influence fuel efficiency. These insights are then integrated into a prototype driver-coaching system that combines behavioural analysis with a chatbot interface, with the purpose of enabling drivers to receive understandable and context-

aware feedback. Through this approach, the thesis aims to explore how explainable and personalized AI systems can support more fuel-efficient driving.

1.5 Limitations

The following limitations should be considered when interpreting the results of this thesis:

- The project builds on existing operational clusters rather than creating or modifying the segmentation approach. As a result, the analysis depends on the assumptions and quality of the predefined clusters.
- The dataset consists of aggregated vehicle-level telemetry rather than data linked to individual drivers. This limits the ability to attribute observed driving patterns to specific driver behaviour, especially when multiple drivers may operate the same vehicle.
- The modelling approach is based on observational data. While SHAP values can explain model predictions, they do not establish causal relationships between driving behaviours and fuel efficiency.
- The driver coach was implemented as a prototype for internal testing rather than for deployment in a real-world operational setting. Therefore, the evaluation does not demonstrate actual behavioural change, long-term driver acceptance or measurable fuel savings.
- The evaluation relies on offline scenarios and a limited set of questionnaire responses. This may not fully capture the complexity or interaction dynamics of real-world driving and coaching contexts.
- The system uses pre-trained, Application Programming Interface (API)-based large language models, limiting control over generation behaviour and the transparency of the language generation process.

2

Background

This section reviews existing driver coaching systems across a range of vehicle domains with a particular focus on the techniques and design approaches they employ. The aim is to examine how concepts relevant to this thesis, such as eco-driving feedback, personalization and the use of data-driven or AI-based methods, are currently realized in practice. This broader perspective helps identify which mechanisms have been explored in other contexts and where gaps remain.

2.1 Conversational Assistants and UX

Conversational assistants, often referred to as chatbots or conversational agents, are intelligent conversational systems designed to mimic human conversation and provide automated support through natural language interaction [19]. Advances in AI have contributed to the increasing adoption of such systems across a wide range of application domains. From a human-computer interaction perspective, however, the effectiveness of conversational assistants depends not only on their technical capabilities but also on the User Experience (UX) they provide [20]. UX extends beyond traditional measures of usability and task performance and include broader aspects of how users perceive and experience interactions with a system.

Research has shown that users often expect conversational assistants to provide more than purely functional task execution. In a study investigating users' envisioned interactions with a "perfect" voice assistant, Völkel et al. found that participants described assistants that were interactive, proactive and knowledgeable about the user, rather than simple question-and-answer systems [21]. These findings suggest that users expect conversational assistants to provide interactions that go beyond simple task execution and are adapted to their individual needs. Such expectations are particularly relevant in coaching applications, where the system is expected not only to provide information, but also to explain and adapt feedback to the individual user.

As conversational assistants become responsible for more complex tasks, user experience factors such as trust and reliability become increasingly important. Baughan et al. found that failures in voice assistants can significantly affect user trust and willingness to use the system for future tasks [22]. Their results suggest that users may reduce their reliance on an assistant following repeated failures. These considerations are especially relevant in driving-related applications, where information

must be communicated in a manner that supports the driver without becoming distracting or overwhelming. Kaufman et al. showed that the type and modality of explanations provided by an AI driving coach can influence driving performance, trust and cognitive load [23]. Their findings highlight the importance of designing coaching systems that provide useful information while presenting it in a way that aligns with human cognitive processes. Together, these studies suggest that the effectiveness of conversational coaching systems depends not only on the capabilities of the underlying AI models, but also on how effectively information is communicated to the user.

2.2 Personalization in Driver Coaching Systems

Personalization in driver coaching systems can be understood as the adaptation of system behaviour and interaction to the needs and preferences of an individual driver [6]. More generally, personalization refers to making something suitable for the needs and preferences of a particular person, and is a concept that has gained increasing attention recently. In the automotive domain, personalization is motivated by the observation that drivers differ in their skills, preferences, and driving contexts, and that these factors may also vary over time. Adapting driver assistance systems accordingly has been shown to improve usability and overall effectiveness.

Existing research approaches personalization from several perspectives. Survey literature provides an overview of how personalization is currently implemented in Advanced Driver Assistance Systems (ADAS), often through implicit driver modelling and data-driven adaptation [6], [24]. Other work focuses on specific system implementations, demonstrating how personalized coaching can be realized in practice through context-aware architectures and adaptive feedback strategies [12], [25]. Additionally, research from a behavioural intervention perspective highlights the importance of tailoring feedback to individual drivers to increase effectiveness [26]. Together, these perspectives highlight that while personalization is increasingly explored, many approaches remain limited in terms of continuous adaptation and flexible feedback generation.

One survey by Hasenjäger et al. focused on personalization in ADAS and showed that most existing approaches rely on implicit personalization through driver models learned from observed driving behaviour [6]. Personalization is typically implemented by adapting model parameters to reflect individual driving styles, with the goal of improving system acceptance and usability. However, the authors noted that personalization is often treated as a one-time process and that most systems lack mechanisms for continuous, interactive adaptation. They further concluded that no existing ADAS prototypes fully integrate driver modelling, adaptive system behaviour, and human-machine interaction into a unified personalized architecture.

Complementing this view, Yi et al. highlighted recent advances in implicit personalized driving assistance and categorized existing work into personalized safe driving systems, driver monitoring systems and in-vehicle information systems [24]. The authors show that most driving assistance systems are still designed as generic solu-

tions, despite significant differences both between drivers and within the same driver over time in regards to both driving behaviour and preferences. Personalization is therefore increasingly realized implicitly through data-driven approaches that adapt system behaviour based on observed driving data and interaction history rather than explicit user input. The survey highlighted that many current systems treat personalization as a static process and lack mechanisms for continuous adaptation, and identified this as an open challenge for future personalized driving assistance systems.

Moving from surveys to concrete system implementations, Gilman et al., proposed a context-aware driving assistance architecture that integrates vehicle, driver and environmental data to provide post-trip coaching aimed at improving fuel-efficient driving [25]. The system relies on predefined, rule-based logic to detect aggressive driving behaviour and generate explanatory feedback, while also adapting the selection and frequency of feedback messages based on observed driver responses over time. Although the system incorporates elements of personalization and self-monitoring, the feedback generation remains rule-based.

Building on such system-level approaches, Orlovska et al., proposed and evaluated a Driver Coach application designed to improve drivers' understanding and use of semi-automated ADAS functions [12]. The system provides real-time context-aware and personalized feedback based on driver behaviour, driving context and system performance. It was evaluated in a field trial with 17 drivers over four months. The proposed architecture emphasizes continuous adaptation through monitoring driver responses and adjusting communication strategies accordingly. While the system demonstrates the potential of personalized and adaptive coaching in practice, feedback generation remains predefined and does not leverage data-driven natural language generation, highlighting an opportunity for more flexible and expressive AI-based coaching approaches.

Beyond vehicle-integrated systems, personalization has also been studied from a behavioural intervention perspective. Baig proposes a personalised intervention model for driving behaviour change, motivated by the observation that generic, one-size-fits-all feedback is often ineffective [26]. The work focused on tailoring feedback based on driver characteristics and preferences, informed through surveys, interviews and telematics data, rather than implementing a fully integrated in-vehicle coaching system. While primarily conceptual and user-centred in nature, the study supports the broader claim that personalised feedback mechanisms can improve driver engagement and the effectiveness of driving behaviour interventions.

Taken together, these studies illustrate that although personalization is increasingly recognized as a key factor in effective driver coaching, existing systems often rely on predefined logic or limited adaptation mechanisms. This highlights remaining challenges related to continuous personalization and flexible, expressive feedback generation.

2.3 Usage of AI and ML

Driver coaching systems have evolved alongside advances in vehicle telematics and data analytics capabilities. Early approaches to automated driver feedback relied primarily on threshold-based rules and predefined performance benchmarks [14]. These systems flagged deviations from prescribed behaviours such as exceeding speed limits, braking events surpassing certain thresholds, or extended periods of engine idling. Based on these metrics, they generated simplified reports intended to improve fuel efficiency. More sophisticated systems utilized Bayesian models, knowledge graphs, and probabilistic methods [11], [15], [16]. These "white box" systems allowed for complex feedback handling while ensuring engineers were able to trace the logic behind coaching recommendations. However, even these advanced traditional models struggle to capture the highly complex correlations in real-world driving environments, particularly when attempting to model the relationship between driver behaviour and operational outcomes across diverse conditions.

Currently, the commercial landscape of AI-driven coaching is predominantly oriented toward safety and risk management in passenger vehicles. Applications such as AutoCoach utilize machine learning algorithms, such as support vector machines, to assess individual driver safety scores by analysing historical driving data and behaviours like swerving or sudden braking [13]. By leveraging these ML and AI algorithms, such systems can provide more relevant feedback tailored to each driver's prior driving behaviour.

Recent advancements in Natural Language Processing (NLP) and LLMs suggest new possibilities for how coaching feedback might be delivered and personalized. Rather than relying solely on dashboard alerts or numerical scores, AI systems could engage drivers through conversational dialogue that explains complex driving strategies tailored to individual comprehension levels. In 2025, Costa et al. [11] developed a conversational AI coach for performance drivers to improve their skills. Feedback was generated post-drive where their comparison was feedback based solely on raw driver data versus feedback delivered through a conversational AI bot. Their results showed that the interpreted data via a conversational AI provided feedback that showed improvements in both driver satisfaction and performance.

Within the heavy-duty trucking sector specifically, research utilizing AI as a coaching tool has often focused on the fleet level [27], [28]. Applications have centred on predictive maintenance, instructing drivers or fleet owners when trucks should undergo maintenance work, as well as fuel economy prediction, emission estimates, and route optimization. As such, there remains a gap in research focused on individualized, personalized driver training in the heavy-duty context.

2.4 Model Interpretability and Feature Attribution

As ML models have grown in complexity, understanding why a model produces a particular prediction has become increasingly important. Feature attribution methods are a form of interpretability techniques that aim to explain model behaviour by assigning importance values to input features. These methods can be applied locally to explain individual predictions or globally to identify which variables are most influential across a dataset. By highlighting the factors that drive model outputs, feature attribution methods can increase transparency and improve user trust in AI-based systems.

Among feature attribution methods, SHapley Additive exPlanations (SHAP) has emerged as a widely adopted approach due to its strong theoretical foundation and ability to explain complex machine learning models. Recent research has demonstrated its applicability across a range of domains, including transportation and natural language processing. For example, Kusnawi et al. proposed a framework for open-pit mining trucks utilizing SHAP to analyse anomalies in fuel consumption [29]. Their system was able to distinguish whether a drop in fuel efficiency was caused by external environmental factors, such as terrain, or by specific behavioural inputs, like excessive speed and harsh acceleration. For a driver coaching system, this distinction ensures that operators are evaluated on the driving behaviours within their control, rather than being penalized for unavoidable operational conditions.

Furthermore, in the context of AI-driven coaches designed to generate readable feedback, feature attribution methods have been adapted to explain text-based models. In their review of SHAP-based explanation methods for NLP, Mosca et al. demonstrate how these frameworks can trace which specific tokens or words drive a model's output [30]. For example, they highlight how SHAP is used in text classification to isolate individual words in a sequence, showing which specific phrases or adjectives triggered a model's final prediction. Applying this level of token-specific interpretability to an AI coach ensures that the generated advice remains transparent, allowing developers to verify that the language model is relying on the correct underlying data rather than hallucinating feedback. However, while SHAP has been applied to both telemetry analysis and NLP transparency independently, its use as a bridge between the two remains unexplored. Specifically, the integration of SHAP-calculated driving features with AI chatbots to deliver explainable driver feedback represents a gap in the research.

2.5 Driver Segmentation and Profiling

Drivers exhibit different behavioural patterns and performance characteristics, and as such modern coaching systems employ segmentation or profiling techniques to categorize drivers into distinct groups. These segmentation approaches enable systems to move beyond treating all drivers identically, instead recognizing that different driver types may require different coaching strategies or evaluation criteria.

Currently, a large part of driver segmentation research focuses on safety assessment and risk management [31]. Systems cluster drivers according to behavioural indicators of aggressive or risky driving patterns, allowing for targeted interventions for high-risk individuals. For instance, AutoCoach employs unsupervised k-means clustering to segment drivers into a small number of safety personality types based on their driving behaviours [13]. Each personality type receives different feedback tailored to their characteristic driving patterns, and performance scoring is calibrated to their assigned personality type rather than against a universal standard. This allows drivers to receive credit for improvements within their own profile, potentially increasing engagement and motivation compared to generic benchmarking.

Driver segmentation has also been applied to fuel efficiency to a certain extent. Massoud et al. proposed a gamified application for scoring drivers on their eco-driving behaviour [32], assigning scores from 0-100 and categorizing drivers into three different styles: Saver, Typical, or Careless. While their results were not conclusive, the study indicates that segmentation research extends beyond safety scoring to encompass environmental impact as well.

Research suggests that professional truck drivers, compared to passenger vehicle drivers, are easier to segment, likely due to the more homogeneous operational conditions under which many drivers work [33]. Professional drivers usually undergo similar training, operate under comparable regulatory frameworks, and work within structured fleet environments that standardize certain aspects of driving behaviour. This relative homogeneity may simplify the identification of meaningful driver clusters and produce more stable segmentation results than those obtained from diverse passenger vehicle populations.

Although driver segmentation has been explored in heavy-duty trucking contexts, such as Seixas et al.'s application of k-means clustering to truck driver data [34], segmentation has, to our knowledge, not been systematically integrated with driver coaching systems to deliver personalized eco-driving feedback in trucking.

2.6 Focus on Eco-Driving

Several studies have explored different approaches to encouraging eco-driving adoption, focusing on interface design, motivation and contextual awareness. Addressing the first aspect of these, interface design, Henzler et al. explored the willingness of professional truck drivers to adopt eco-driving guidelines [35]. In their study, they evaluated driver responses to two different real-time feedback interfaces: a graphical dashboard display and a haptic accelerator pedal. Their findings indicated that truck drivers are generally positive regarding the adoption of fuel-efficient driving styles, provided the assistance system is intuitive. They found that by making the interface unobtrusive, specifically through haptic feedback that does not require visual attention, drivers were significantly more willing to adapt their behaviour compared to those using visual displays alone.

Massoud et al. [32] similarly investigated the challenge of driver motivation. The authors proposed a framework based on "Serious Games" in which driving perfor-

mance is gamified through scoring and profiling. Unlike systems that rely solely on output metrics like fuel consumption, Massoud et al. identified the Throttle Position Sensor as the most critical behavioural indicator, as it represents the direct control input of the driver independent of vehicle load. By converting physical inputs into a digital score (0-100), the system leverages competitive psychology to incentivize behavioural shifts, transforming eco-driving from a passive monitoring task into an active skill-development challenge.

Magaña and Muñoz-Organero [36] developed "Artemisa" a smartphone-based assistant that fuses vehicle telematics with mobile sensor data to detect road slope and traffic conditions in real-time. By applying classifiers to this contextual data, the system distinguishes between 'necessary' high-load events and inefficient driving behaviours. Their study demonstrated that this context-aware approach could yield fuel savings of approximately 11% without requiring expensive, proprietary hardware.

While these studies demonstrate valuable approaches to eco-driving assistance, it is notable that only Henzler et al. focused on professional truck drivers; both Massoud et al. and Magaña and Muñoz-Organero conducted their research with passenger vehicles. This reflects a broader pattern in eco-driving research, where heavy-duty trucking has received less attention despite its distinct operational contexts and potentially greater fuel-saving impact. Furthermore, these studies focused primarily on real-time feedback, providing guidance during the driving task, whereas post-trip analysis for long-term behavioural reflection remains less explored in these studies.

2.7 Summary

The reviewed literature shows that driver coaching systems have been explored from several perspectives. Research on conversational assistants highlights that user experience factors such as trust, reliability and information presentation play an important role in determining how users interact with and rely on AI-based systems. At the same time, existing eco-driving systems often focus on interface design, gamification or real-time feedback, while research on personalization frequently relies on predefined rules or limited adaptive mechanisms.

Similarly, although AI and ML methods have been applied in automotive contexts, their use has primarily focused on safety assessment, predictive maintenance or fleet-level optimization rather than individualized eco-driving coaching. Feature attribution methods have emerged as a promising approach for improving transparency in complex machine learning systems and have been successfully applied both to transportation-related analyses and to explaining AI-generated outputs.

At the same time, driver segmentation has been studied as a means of categorizing driving behaviour, yet it has rarely been integrated into driver coaching systems to tailor feedback to different behavioural profiles. Taken together, these observations suggest a gap in the current research landscape: the lack of a system that combines telemetry-driven analysis, interpretable machine learning and AI-based

conversational feedback to deliver personalized eco-driving coaching for heavy-duty truck drivers. Addressing this gap forms the central focus of this thesis.

3

Theory

This chapter presents the theoretical background underlying the main components of the thesis. It begins with eco-driving, which defines the application domain by outlining the importance of fuel-efficient driving in heavy-duty trucking and the behaviours that influence fuel consumption. It then introduces clustering, as the work builds on an existing segmentation of driving contexts and operating conditions. Model interpretation methods are thereafter presented, with particular emphasis on SHAP, since interpretable ML is central to explaining model predictions and transforming them into actionable coaching insights.

The chapter then shifts to the conversational AI aspects of the work. LLMs are introduced as the foundation for generating conversational feedback, followed by RAG as an approach for improving factual grounding and reducing unsupported outputs. Prompt engineering is subsequently discussed as a way to guide model behaviour and improve response quality. Finally, personalization in conversational AI is presented to motivate how user-specific preferences and contextual information can be represented and retained across interactions. Together, these topics provide the theoretical basis for the thesis by connecting eco-driving behaviour, data-driven analysis, interpretable modelling, grounded language generation, and personalization in a unified driver coaching framework.

3.1 Eco-Driving

Eco-driving encompasses a range of driving techniques and behaviours aimed at reducing fuel consumption and improving operational efficiency. In the heavy-duty trucking sector, there is a significant positive economic and environmental potential in utilizing better eco-driving behaviours. Specifically, adopting simple practices such as smoother acceleration and better anticipation of traffic can significantly reduce consumption.

Fuel costs represent a substantial portion of trucking operational expenses. According to the American Transportation Research Institute, approximately 40% of operational costs for trucking companies are attributable to the combination of fuel consumption, maintenance, repair, and tire replacement [37]. While vehicle manufacturers such as Volvo invest considerable resources in developing more fuel-efficient engines and drivetrains, driver behaviour remains a critical factor influencing total fuel consumption. Martin et al. [38] found that eco-driving behaviour reduces fuel

consumption by 5–10% on average, with reductions of 20–50% achievable for certain drivers. This suggests substantial potential for fleet operators to reduce fuel expenses and consequently overall operational costs through improved driver behaviour alone.

Effective eco-driving involves several techniques [39], [40]:

- **Smooth acceleration and deceleration**: Gradual throttle inputs allow automated transmissions to select optimal gears and minimize fuel consumption. Aggressive acceleration increases engine RPMs unnecessarily, leading to disproportionate fuel use.
- **Speed consistency**: Maintaining a steady speed, ideally at or slightly below posted limits, improves fuel efficiency. Fuel consumption increases disproportionately with speed, and frequent speed variations waste energy.
- **Anticipatory driving**: Looking ahead to anticipate traffic patterns, signal timing, and road conditions enables drivers to maintain momentum and avoid unnecessary braking or acceleration. "Stop-and-go" driving is energy inefficient.
- **Effective braking strategies**: Maximizing the use of engine braking and retarders while minimizing service brake application reduces both fuel consumption and brake wear.
- **Optimizing automated systems**: Modern trucks with automated manual transmissions benefit from drivers who utilize eco-mode settings and cruise control where appropriate, allowing vehicle systems to optimize gear selection and throttle response.
- **Minimizing auxiliary energy use**: Reducing unnecessary idling and limiting use of auxiliary systems (heating, cooling, lights) when not required decreases overall energy consumption.

3.2 Clustering

Clustering is an unsupervised ML method that aims to divide or partition a multidimensional set of data points into clusters based on their similarity. Given a set of data points, a clustering algorithm assigns each point to a cluster such that intra-cluster similarity is maximized while inter-cluster similarity is minimized. The ultimate goal is to uncover hidden patterns in multidimensional data.

Clustering methods vary in complexity, ranging from simple distance-based techniques to more advanced approaches. Clustering algorithms are normally categorized based on their underlying principles and assumptions. Typical clustering models include:

- **Centroid models**, which represent each cluster by a central point, commonly the mean vector, e.g., k-means algorithm.
- **Density models**, which identify clusters as regions of high data density, e.g., DBSCAN.

- **Connectivity models**, which are based on distance connectivity, e.g., hierarchical clustering.

This thesis builds upon cluster definitions generated by Seixas et al. [34], who applied the k-means clustering algorithm to heavy-duty vehicle telematics data in Latin America. The k-means algorithm partitions n data points into k distinct clusters by minimizing the within-cluster sum of squares, also known as variance. The optimization problem is defined as:

$$\arg \min_{\{C_1,\ldots,C_k\}} \sum_{i=1}^{k} \sum_{\mathbf{x}\in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \tag{3.1}$$

where the centroid of cluster i, denoted as $\boldsymbol{\mu}_i$, is calculated as the mean of all data points assigned to that cluster:

$$\boldsymbol{\mu}_i = \frac{1}{|C_i|} \sum_{\mathbf{x}\in C_i} \mathbf{x} \tag{3.2}$$

Where:

- k represents the predefined total number of clusters.
- C_i denotes the i-th cluster, and $\{C_1, \ldots, C_k\}$ represents the complete set of partitioned clusters.
- $\mathbf{x}$ represents an individual multidimensional data point assigned to cluster C_i.
- $\boldsymbol{\mu}_i$ is the centroid of cluster C_i.
- $\|\mathbf{x} - \boldsymbol{\mu}_i\|^2$ is the squared Euclidean distance between a data point $\mathbf{x}$ and its assigned cluster centroid $\boldsymbol{\mu}_i$, representing the variance or "error" that the algorithm seeks to minimize.
- $|C_i|$ represents the cardinality, or the total number of data points, currently assigned to cluster C_i.

A challenge in k-means clustering is determining the appropriate number of clusters k. Seixas et al. employed the elbow method to address this challenge [34]. The elbow method evaluates clustering quality as a function of the number of clusters by examining two complementary metrics. First, the within-cluster sum of squares is evaluated across different values of k. A low within-cluster sum of squares indicates tight grouping of data points within their clusters. Second, the sum of squares between clusters is examined as a function of k. A high sum of squares between clusters indicates clear separation between clusters. The optimal number of clusters is identified at the inflection point, or "elbow," of these curves, where a balance is struck between model complexity and error reduction. This point represents the number of clusters beyond which increasing k yields diminishing returns.

3.3 Model Interpretation Methods

When analysing a model's output, there is a significant trade-off between model accuracy and interpretability [41]. While complex algorithms, such as Deep Neural Networks, can achieve high predictive accuracy on complex multidimensional data, they often operate as "black boxes". The results are inherently difficult to interpret and often opaque. Simple models, such as linear approximations, provide clear explanations of the underlying mechanics but suffer from an inability to fully map complex, non-linear data structures.

Early approaches to post-hoc interpretability focused on local approximations. An early model in this area, introduced by Ribeiro et al., was Local Interpretable Model-Agnostic Explanations (LIME), which is able to explain the predictions of any classifier in what is described as an interpretable and faithful manner [42]. LIME isolates a single prediction from a black-box model and maps out its local decision boundary through a process of random neighbourhood sampling. The algorithm generates a dataset of synthetic, perturbed data points by randomly altering or masking features of the original input. These perturbed instances are weighted based on their proximity to the original data point, ensuring that closer variations have a greater influence. It then observes how the black-box model's prediction changes in response to the perturbations. LIME fits a simple interpretable model, such as linear regression, around that data point. This allows the user to understand which features were driving individual predictions, regardless of the underlying complex algorithm being used.

A drawback of LIME, and other similar models, is their lack of rigid mathematical guarantees. Due to LIME's reliance on random neighbourhood sampling, it suffers from a problem of robustness [43]. That is, two almost identical inputs should yield the same result or explanations, yet this is not always the case for LIME. This instability could lead to a user receiving different explanations for nearly the same event, thereby eroding the trustworthiness of the system.

This form of inconsistency was addressed by Lundberg and Lee with the creation of the SHAP framework [41]. The authors argued that, under the hood, many interpretability models, such as LIME, share the same underlying mathematics, which they defined as Additive Feature Attribution. These forms of models construct a simplified explanation model, denoted as g. This explanation model operates as a linear combination of binary variables, meaning it attributes an additive impact to each feature. The explanation model is defined as:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{3.3}$$

Where $z' \in \{0,1\}^M$ is a binary vector indicating whether a simplified feature is present or not, M represents the number of input features, ϕ_0 is the expected base value of the model's predictions, and ϕ_i is the allocated SHAP value for feature i. By summing the base value and the individual feature contributions, the explanation

model explicitly shows how a specific combination of features pushes the prediction away from the dataset's average.

However, previous models lacked a theoretical justification for how ϕ_0 was distributed. What SHAP added was a link between this mathematics and Shapley values, a concept within game theory. Introduced by Lloyd Shapley, its original purpose was to fairly distribute a total payout among a group of players based on their individual marginal contributions to a game's outcome. Within ML, this concept is adapted by redefining the core components: the "game" becomes the specific prediction task for a single instance, the "players" are the individual features within that data point, and the "payout" is the difference between the model's actual prediction for that instance and the expected prediction across the entire dataset.

These exact Shapley values, used as feature attributions, are calculated as:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)] \tag{3.4}$$

Where F is the set of all features, S is a subset of features excluding i, and $f_S(x_S)$ is the model's prediction using only the features in S. The term in brackets calculates the marginal contribution of feature i to subset S, while the preceding fraction acts as a weighting factor, accounting for all possible permutations of feature subsets.

By linking Additive Feature Attribution and Shapley Values, a feature's SHAP value represents its average marginal contribution calculated across all possible permutations of the available features. Hence, SHAP is, according to the authors, the only model that satisfies three core properties of explainability:

1. **Local accuracy:** The output of the explanation model must match the output of the original complex model. The explanation does not approximate the final number, instead, it reaches it exactly:

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \tag{3.5}$$

2. **Missingness:** If a feature is logically missing or its simplified input is absent, it must be attributed an impact of zero:

$$x'_i = 0 \Rightarrow \phi_i = 0 \tag{3.6}$$

3. **Consistency:** If the underlying machine learning model changes such that it relies more heavily on a specific feature, that feature's attributed importance must not decrease. Let $f_x(z') = f(h_x(z'))$ and $z' \setminus i$ denote $z'_i = 0$. For any two models f and f', if $f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i)$ for all inputs $z' \in \{0, 1\}^M$, then $\phi_i(f', x) \geq \phi_i(f, x)$.

Defined as Theorem 1 of Lundberg and Lee's work, only one explanation model g satisfies both the formula for additive feature attribution and the three properties above, which is:

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!}[f_x(z') - f_x(z' \setminus i)] \tag{3.7}$$

Where $|z'|$ is the number of non-zero entries in z', and $z' \subseteq x'$ represents all z' vectors where the non-zero entries are a subset of the non-zero entries in x'.

Despite its strong theoretical foundation, there was a major drawback with standard implementations of SHAP, such as KernelSHAP: the assumption of feature independence, as highlighted by Aas et al. [44]. Standard SHAP evaluates permutations by randomly sampling background data to simulate "missing" features. When features are highly correlated, the model will inevitably create unrealistic combinations. The resulting SHAP values can, as such, become highly unrealistic.

For tree-based ML models, this problem is resolved by TreeSHAP [45]. Designed explicitly for models such as Random Forests and Gradient Boosted Trees, TreeSHAP calculates exact Shapley values in polynomial time by tracing the conditional decision paths built directly into the trees. By restricting the calculations to actual routing paths formed during training, TreeSHAP respects the real-world dependencies of the data. This provides explanations that maintain SHAP's strict mathematical guarantees while remaining robust against the out-of-distribution errors that plague KernelSHAP.

3.4 Large Language Models

LLMs have emerged as a major advancement in NLP, driven by a combination of transformer-based architectures, large-scale training data and the availability of increased computational capacity [46]. These developments have enabled models that can generalize across a wide range of language tasks such as question answering, summarization, information synthesis and conversational interaction. As a result, LLMs have become increasingly relevant for systems that could benefit from flexible and natural language-based communication with users.

In the context of this thesis, LLMs are of interest due to their ability to generate adaptive feedback. However, despite their strong generative capabilities, LLMs are known to exhibit limitations related to factual reliability and consistency [46]. As a result, their direct application in safety-relevant domains, such as driving, requires careful consideration. This section therefore provides a high-level theoretical overview of LLMs, their core capabilities and known limitations that are relevant to the design choices explored in this work.

3.4.1 How LLMs work

Recent advances in NLP are largely driven by the observation that LLMs can acquire rich representations of language, context and factual knowledge by being trained to predict the next token in a sequence based on its preceding context, using large-scale text corpora [47]. This training is commonly referred to as pre-training, and

it enables models to learn general linguistic patterns and world knowledge. As a result, pre-trained LLMs exhibit strong performance across a wide range of natural language tasks, as the learned representations can be reused and adapted without task-specific training.

At inference time, an LLM takes a sequence of tokens as input and produces a probability distribution over possible next tokens [47]. Text generation is subsequently performed by sampling a token from this distribution and appending it to the existing context, after which the process is repeated. This iterative procedure results in an autoregressive generation process, where each generated token is conditioned on both the initial input and the model's previously generated output. The use of probabilistic sampling enables LLMs to generate flexible and context-sensitive text, but also introduces variability in the generated responses.

LLMs can be broadly categorized into three architectural variants: encoder-only models, decoder-only models, and encoder-decoder models [47]. Encoder-only models are typically trained using masked language modelling objectives and are commonly applied to tasks such as classification or representation learning. Encoder-decoder models map an input sequence to an output sequence and are frequently used in applications such as machine translation and speech recognition. Decoder-only models generate text auto-regressively by predicting the next token based solely on the preceding context. This architecture is particularly well-suited for open-ended text generation and conversational interaction. Most contemporary LLMs used for conversational and generative applications use the decoder-only architecture. The models considered in this thesis all follow the decoder-only architecture. For this reason, the subsequent discussion focuses on decoder-based language models, while other architectural variants are not considered further.

A useful perspective on LLMs is that many language-related tasks can be framed as conditional text generation [47]. In this setting, the model generates text conditioned on an input sequence, which is commonly referred to as a prompt. By conditioning generation on different forms of input text, such as questions or instructions, the same underlying model can be applied to a wide range of tasks. This formulation underlies the use of LLMs for conversational interaction and feedback generation, where the generated output is shaped by both the provided context and the model's previously generated tokens.

During generation, sampling strategies are typically employed to select tokens from the model's predicted probability distribution rather than always choosing the most probable option [47]. Sampling parameters such as temperature regulate the trade-off between deterministic and diverse generation by controlling the degree of randomness in the output. Lower temperature values bias the model toward higher-probability tokens, resulting in more consistent responses. Higher values instead increase variability at the cost of reduced reliability.

3.4.2 Hallucinations and Inconsistencies

Despite their strong generative capabilities, LLMs are known to produce factually incorrect or unsupported statements [47]. This phenomenon is commonly referred to as hallucination. It arises because LLMs are trained to generate text that is statistically likely and coherent given a context, rather than to verify the factual correctness of their outputs. As a result, models may generate plausible-sounding responses that are not grounded in reliable information. In addition to outright factual errors, outputs may also exhibit inconsistencies across similar inputs or across multiple generations for the same input. Such variability is a consequence of probabilistic text generation and can lead to unstable or contradictory responses.

Recent survey work has shown that hallucinations are a systematic and persistent challenge in LLMs, including instruction-following and conversational systems, and that increased model scale alone does not eliminate the problem [48]. Because hallucinated content is often expressed fluently and with high confidence, such errors can be difficult for users to detect.

Addressing hallucinations and improving factual grounding is therefore an important consideration when integrating LLMs into applied systems. In this work, the LLM is treated as a generative component whose outputs must be constrained and supported by additional system mechanisms, rather than relied upon as an authoritative source of truth. One common approach to mitigating hallucinations is to ground generation in external knowledge sources [47], an idea explored in the following section on RAG.

3.5 Retrieval Augmented Generation

LLMs are trained on static corpora and therefore lack access to information introduced after training. When used in knowledge-intensive settings, this can result in outdated or fabricated responses. In addition, hallucinations represent a well-documented failure mode of LLMs [47]. These limitations pose challenges for applications where factual correctness and transparency are important.

RAG addresses these limitations by combining LLMs with information retrieval techniques [47]. In a RAG-based system, relevant documents are retrieved from an external knowledge source based on a user query and provided as additional context to the language model during generation. By conditioning generation on retrieved passages, RAG systems aim to improve factual grounding and transparency compared to generation based solely on internal model parameters. This makes RAG suitable for applied systems where correctness are important.

The remainder of this section provides an overview of the RAG process and its main components.

3.5.1 Information Retrieval

Information Retrieval (IR) concerns the task of identifying documents that are relevant to a user's information need, typically expressed as a textual query [47]. IR

forms the foundation of a RAG system by selecting a subset of documents from a larger collection that are likely to contain information relevant to the current query. These retrieved documents serve as contextual evidence for the subsequent text generation.

Traditional IR systems rely on lexical matching techniques such as TFIDF or BM25, which score documents based on term overlap [47]. Modern RAG systems, however, commonly employ dense retrieval methods in which queries and documents are represented as continuous vector embeddings. By mapping both queries and documents into the same semantic vector space, dense retrieval enables similarity-based matching and mitigates vocabulary mismatch between queries and documents. At query time, the embedding of the input query is compared to pre-computed document or passage embeddings using similarity search, and the most similar passages are selected as retrieval candidates. This approach allows semantically relevant information to be retrieved even when surface wording differs.

3.5.2 Document Preprocessing and Chunking

In RAG systems, source documents are typically divided into smaller units prior to indexing [47]. This preprocessing step is commonly referred to as chunking. Chunking is necessary because full documents may exceed the context window of the language model and can also be inefficient to retrieve in their entirety. By splitting documents into manageable passages, retrieval works at a smaller scale which allows the system to return only the most relevant segments as context for the generation.

There are several different chunking strategies that are commonly used [49]. Token-level chunking divides text into fixed-length segments measured in tokens. This approach is computationally simple and ensures uniform chunk sizes, but may split sentences or semantic units across boundaries. Sentence-level chunking instead uses sentence delimiters to preserve linguistic structure, which provides more coherent segments at the potential cost of uneven lengths. Semantic-level chunking instead determines breakpoints by using LLMs. This is done by identifying semantically coherent units before splitting the text, which better preserves contextual integrity compared to fixed-length segmentation. However, it introduces additional computational overhead and increased complexity.

The choice of chunk size introduces a trade-off between retrieval precision and contextual completeness[49]. Smaller chunks increase retrieval granularity and can improve the likelihood that highly relevant passages are retrieved. However, they also risk fragmenting semantically coherent information across multiple chunks. Larger chunks preserve more surrounding context but may dilute relevance by including unnecessary information, potentially reducing retrieval precision.

Empirical evaluations show that moderate chunk sizes often provide the most favourable balance [49]. In controlled experiments, chunk sizes in the range of 256-512 tokens achieved the highest combined scores for faithfulness (i.e. factual consistency with retrieved text) and relevancy (i.e. alignment between query, context, and response). To further mitigate boundary effects, meaning situations where relevant informa-

tion is split across chunk borders and therefore not fully retrieved, overlapping or sliding-window strategies are commonly applied. By introducing partial overlap between adjacent chunks, these techniques preserve contextual continuity and improve retrieval robustness without substantially increasing retrieval noise.

3.5.3 Embeddings and Similarity search

After documents have been segmented into chunks, each chunk is transformed into a dense vector representation, commonly referred to as an embedding [47]. Embeddings map textual inputs into a continuous semantic vector space, where semantically similar texts are located closer to each other. In a RAG system, both document chunks and user queries are encoded using the same embedding model to ensure that they reside in a shared semantic space.

This use of vector representations is conceptually related to the clustering approach introduced in Section 3.2, where similarity between multidimensional data points is also used to identify meaningful structure in the data. However, while clustering groups telemetry observations into operational segments, embeddings in RAG are used to represent textual meaning and support retrieval based on semantic similarity.

The choice of embedding model is important, as it determines how effectively semantic similarity between queries and document chunks can be captured [49]. Modern RAG systems commonly employ dense encoders, often based on transformer architectures. Examples include open-source embedding models such as BGE or LLM-Embedder, as well as API-based embedding models.

Once embeddings have been computed, they are stored in a vector database that supports efficient similarity search. Vector databases such as FAISS, Milvus, Qdrant, or Weaviate implement approximate nearest neighbour (ANN) algorithms to retrieve the top-k document vectors most similar to a given query embedding [49]. The parameter k controls how many candidate chunks are retrieved, thereby influencing the trade-off between contextual coverage and input length constraints. These systems are designed to scale to large document collections and often provide multiple index structures to balance retrieval accuracy, latency, and memory usage.

At query time, the input query is encoded into an embedding and compared against stored document embeddings using similarity measures such as cosine similarity or dot product [50]. The most similar chunks are selected and passed to the language model as contextual grounding for generation.

3.5.4 Generation with Retrieved Context

Once relevant document chunks have been retrieved, the generation phase integrates this external information into the LLM's input, meaning that the model conditions on both the original user query and the retrieved passages [47]. This input formulation differs from standard LLM usage, where the model typically conditions only on the user prompt.

The introduction of retrieved documents changes the input distribution and may affect how the model interprets the query, particularly for smaller models [51]. For this reason, prior work highlights the importance of adapting or fine-tuning the generator to handle inputs of the form query + retrieved documents, ensuring that the model effectively utilizes the provided context rather than ignoring or misinterpreting it.

Fine-tuning strategies can focus on different components. Some approaches fine-tune the generator to better incorporate retrieved evidence and reduce hallucinations, while others adapt the retriever to provide more beneficial passages for downstream generation [49]. Holistic approaches treat RAG as an integrated system and jointly optimize both the retriever and the generator to improve overall performance, albeit at the cost of increased system complexity.

In practice, generation quality in RAG systems depends not only on retrieval accuracy but also on how effectively the language model incorporates retrieved evidence into its reasoning process. Consequently, model adaptation, context formatting, and system integration are central considerations when designing RAG-based architectures.

3.5.5 Limitations with RAG

While RAG improves the factual grounding of LLMs and reduces the risk of hallucinated outputs, it does not eliminate this issue entirely [47]. Generated responses may still contain inaccuracies, particularly when the retrieved context is incomplete, ambiguous or not effectively utilized by the language model.

Another limitation relates to the retrieval process itself. Effective retrieval depends on clear, accurate, and well-formulated queries [49]. In practice, semantic gaps may exist between a user's query and relevant documents, even when both are represented as embeddings. As a result, relevant information may not be retrieved, which can negatively affect the quality of the generated response.

In addition, retrieval systems may return irrelevant or noisy documents, especially in cases where multiple documents are semantically similar or the query is ambiguous [50]. Such noise can degrade generation quality by introducing misleading or unnecessary context. Similarly, poorly phrased queries may lead to missed relevant chunks, further limiting the effectiveness of the system. Techniques such as hybrid retrieval, which combine lexical and semantic matching, have been proposed to mitigate these issues [50].

3.6 Prompt Engineering

Prompt engineering is the technique of optimizing the input prompts to LLMs to generate higher-quality outputs. This section provides a general overview of the literature that has researched both general prompting principles and more specified and advanced prompting techniques.

Marvin et al. [52] outline five principles for optimizing prompts, emphasizing that

effective communication with LLMs requires structured constraints. The core principles include:

1. **Define a clear objective**: Establishing a specific goal minimizes semantic ambiguity. By explicitly stating the desired outcome, the user narrows the model's output, steering it away from irrelevant generations.
2. **Understand model capabilities**: Users must recognize the limitations of their chosen model, including its context window size and specific strengths (e.g., reasoning versus creative writing). Understanding these boundaries prevents prompting the model beyond its operational capacity.
3. **Format precisely**: LLMs are sensitive to syntax and delimiters. Utilizing a clear, precise, and logically sequenced format helps guide the model's attention mechanism, ensuring it appropriately weights the most critical instructions within the prompt.
4. **Provide context**: Providing relevant background information acts as a grounding mechanism. Contextual constraints anchor the model's latent space to a specific domain, significantly reducing the likelihood of hallucinations or reliance on generic, pre-trained data.
5. **Iterate and refine**: Prompt engineering is rarely a zero-shot success. It requires a systematic, empirical approach where the user evaluates the generated responses for failure modes (such as logical drift or formatting errors) and iteratively refines the input variables to optimize the output.

While these principles serve as a theoretical baseline, Marvin et al. note that their guidelines are highly contextual. The necessity of explicitly defining each parameter depends on the user's expertise. Users with significant experience engineering prompts for a specific LLM likely understand the model's functional boundaries without needing formal evaluation at each step. Ultimately, the authors emphasize that optimal prompt engineering is an interplay between the specific language model architecture, the user's objectives, and the intended audience.

Kong et al. [53] investigated four varying prompting strategies and compared which achieved the highest accuracy metrics across standardized datasets. The four compared strategies include:

- **Zero-Shot prompting:** Provide only the task without context, forcing the model to rely solely on its pre-trained knowledge.
- **Zero-Shot Chain of Thought (CoT):** Add instruction similar to "think step-by-step" to the zero-shot prompt, encouraging the model to provide the reasoning process before answering
- **Few-Shot CoT:** Include context alongside the task, such as providing similar example question with their corresponding answers, and also include CoT.
- **Role playing setting:** Assigning a specific persona to the model such as "You are an excellent maths teacher" to guide the tone and perspective of its response.

In their evaluation across twelve reasoning benchmarks, Kong et al. [53] found that the role-play prompting strategy outperformed the different approaches. The authors claim that role-playing functions as an implicit CoT trigger. By assigning a specific persona, the prompt provides the LLM with an identity and background context that naturally forces it into a more rigorous reasoning process. Their results demonstrated that this implicit trigger actually generated more effective reasoning steps than explicitly instructing the model to "explain step-by-step". For instance, the authors noted performance increases on specific benchmarks, such as accuracy on symbolic reasoning tasks jumping from 23.8% to 84.2%, highlighting that role-play prompting changes its underlying reasoning capabilities.

3.7 Personalization in Conversational AI

LLMs have expanded the capabilities of conversational agents by enabling flexible conversational interaction across a wide range of topics [54]. Unlike earlier dialogue systems that were typically designed for short interactions within specific domains, modern LLM-based systems can sustain longer and more open-ended conversations. While this flexibility enables new applications, it also introduces challenges for systems that interact repeatedly with the same user over time.

In such settings, effective interaction often requires personalization. Rather than generating identical responses for all users, conversational systems should adapt their behaviour based on user-specific preferences, goals, and interaction patterns [55]. Achieving this requires mechanisms that allow the system to retain and utilize information about individual users across multiple interactions. This section therefore introduces personalization in conversational AI, outlines key challenges associated with LLM-based personalization, and reviews memory-based approaches for representing and retaining user context.

3.7.1 User Modelling

Personalization in intelligent systems commonly relies on user modelling, which aims to infer and represent characteristics of individual users based on historical interactions or user-generated data [55]. User models (or user profiles) are commonly described as structured representations capturing user preferences, needs and behaviours, typically derived either from direct user feedback or inferred from interaction data [56]. User modelling can accordingly be viewed as the process of acquiring, extracting and representing such user characteristics in a form that supports personalized system behaviour.

A recurring theme in the user modelling literature is that effective profiles combine both static and dynamic user information [56]. Static attributes represent relatively stable characteristics (e.g., background information or constraints), whereas dynamic attributes capture evolving behaviours and preferences observed through ongoing interaction. Maintaining both aspects supports personalization that is responsive to short-term changes while remaining consistent with longer-term user needs.

User profile construction can further be approached at different levels of granularity. Individual profiling represents a single user based on their own data, while group profiling aggregates commonalities across multiple users who share similar behaviours or preferences [56]. This distinction is relevant in settings where group-level segmentation provides useful context, but individual-level adaptation is still required for personalized interaction.

In conversational systems, user models are often constructed from dialogue interactions [55]. During conversations, users may explicitly express preferences, goals or constraints, while other characteristics can be inferred indirectly from interaction patterns. Maintaining such user representations becomes particularly important in systems that interact repeatedly with the same user, where previously observed preferences and behaviours can inform future responses.

3.7.2 Challenges for LLM-Based Personalization

LLMs face several challenges when used for personalization. One challenge arises from the nature of conversational data itself. Relevant user information is often sparse and embedded within large amounts of otherwise generic interaction data [55]. Important signals about user preferences may only appear occasionally within a conversation, making them difficult to identify and prioritize during response generation.

Another challenge relates to the way LLMs process context [57]. Language models typically generate responses based on the tokens available in their context window, meaning that information not included in the prompt cannot directly influence generation. As conversations grow longer or span multiple sessions, it becomes increasingly difficult to ensure that relevant user information remains available within the model's input [54]. Consequently, models may produce responses that overlook previously stated preferences or revert to generalized behaviour[58].

These limitations highlight a central challenge for conversational personalization: while LLMs can generate coherent responses, they do not inherently maintain persistent representations of individual users. Personalization therefore requires mechanisms that allow systems to store and retrieve user-specific information across interactions.

3.7.3 Memory in Conversational Systems

Memory mechanisms are increasingly recognized as a defining component of LLM-based agents [54], [59]. While stand-alone language models generate responses conditioned only on the current prompt and a limited context window, conversational agents require mechanisms for retaining and utilizing information beyond the immediate interaction. Such memory modules enable agents to accumulate experience and adapt behaviour over time.

In conversational systems, memory is often categorized into short-term and long-term forms [59]. Short-term memory typically corresponds to information main-

tained within the current interaction session, such as the dialogue history. This supports local conversational coherence by allowing the system to reference previous turns. However, short-term memory is constrained by the model's context window and does not persist across sessions.

Long-term memory instead refers to mechanisms that store information across multiple interactions [59]. Such memory enables systems to retain knowledge about users, tasks or previous events and incorporate this information into future responses. Long-term memory is therefore particularly important for personalization, as it allows conversational agents to maintain persistent representations of user preferences, goals and behavioural patterns.

Memory systems can also differ in how information is represented. One distinction concerns parametric memory, where information is encoded within model parameters through fine-tuning, and textual memory, where information is explicitly stored as natural language or structured textual records [59]. Textual memory is widely used in conversational systems because it allows information to be easily updated and selectively retrieved when needed. This explicit representation also supports transparency in how user information is modelled, which has been identified as an important aspect of explainable user modelling [56]. In explainable user modelling, systems aim to make user profiles understandable and revisable by users, enabling greater transparency and control over how personal preferences are interpreted by the system.

In addition to storage format, memory mechanisms are characterized by operational processes such as writing, retrieving, updating and discarding stored information [59]. Naively storing complete dialogue histories may introduce noise and increase computational cost, making it difficult for the model to focus on relevant information. Therefore, many conversational systems instead extract salient information from interactions and store it in more compact representations.

3.7.4 Representing Personal Context

One approach to representing user-specific information is to construct explicit user profiles derived from conversational interactions [58]. Rather than conditioning generation on full dialogue histories, it can be beneficial to generate intermediate summaries that capture distinctive user attributes such as preferences or behavioural patterns. These profiles act as concise and interpretable descriptions of the user and can be incorporated into prompts to guide response generation.

Another approach involves memory-augmented conversational architectures in which salient user information is extracted from dialogue and stored in external repositories [57]. In these systems, important facts about the user are embedded into vector representations and stored in a database that supports similarity-based retrieval. During subsequent interactions, relevant stored information can be retrieved and incorporated into the prompt, allowing the system to maintain continuity across conversations while avoiding the need to process large dialogue histories.

3.7.5 Risks and Considerations

While memory-based personalization enables more adaptive conversational systems, it also introduces several challenges. Stored user information may become outdated over time, leading to incorrect assumptions about user preferences [57]. Errors in extracted or stored memory entries may influence future responses if not corrected, highlighting the importance of mechanisms for updating or removing outdated information [54]. In addition, storing and processing user-specific information raises privacy and data management concerns [55]. Designing memory mechanisms that remain accurate, transparent and privacy-conscious therefore remains an important challenge for personalized conversational systems.

4

Method

This project drew inspiration from a Design Science Research approach using a design-build-evaluate methodology [60]. The methodology was selected because it is well-suited to the purpose of the project: to develop and evaluate an artefact in the form of an AI-driven personalized driver coaching framework, while also generating generalizable insights into modelling and personalization strategies.

The research process was structured into three main phases: problem identification, solution design, and evaluation [60]. In the first phase, relevant literature on the different components of the project, such as driver profiling, personalization, and adaptive systems was reviewed to define and motivate the research problem. In the second phase, alternative approaches for driver modelling, predictive modelling and conversational feedback generation were designed. In the final phase, the proposed artefact was evaluated by expert evaluation in the form of a questionnaire.

Figure 4.6 provides an overview of the methodological pipeline used in this study to create the artefact, from data preparation and model development to explanation generation and the conversational driver coaching prototype.

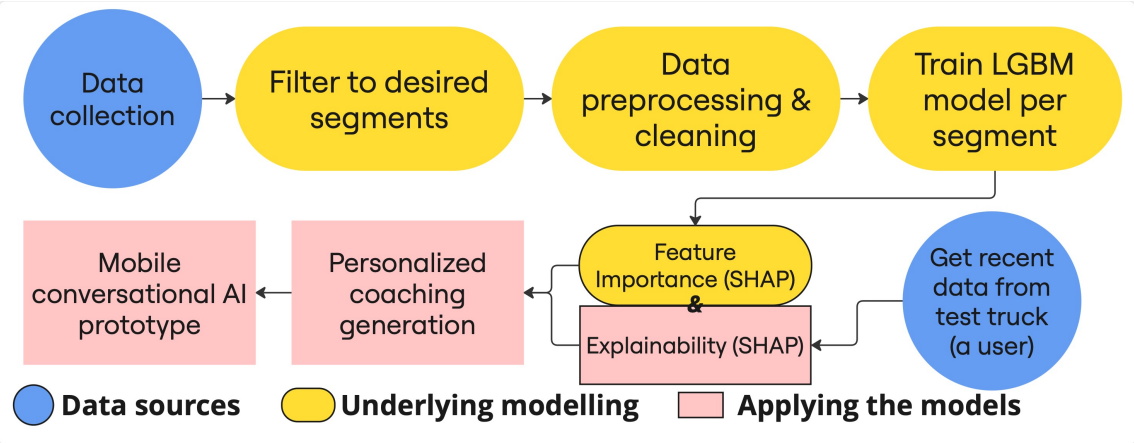


Figure 4.1: Flowchart visualization of the full pipeline used to develop the AI-driven driver coaching artefact.

For clarity and readability, the term *prototype* will be used interchangeably with artefact throughout the remainder of this thesis.

4.1 Data Description

This section outlines the dataset used for analysis, including its origin, the methods of data collection and a descriptive overview of the telemetry parameters.

4.1.1 Data Source

This project relied on telemetry data from a fleet of Volvo heavy-duty vehicles operating across the Latin American market. These data streams were generated by a set of on-board sensors that monitored different aspects of the vehicle during active service. This telemetry provided a high-dimensional view of vehicle performance, capturing variables such as engine load and fuel efficiency. The scope of the dataset spanned from 2024 to 2026. While the fleet primarily consisted of recently assembled vehicles, it included a small portion of older models equipped with earlier-generation powertrains.

Due to the differences of truck configurations and operational usage in this dataset, directly comparing global fleet averages is ineffective. To attempt to perform a fair evaluation of vehicle performance, we utilized a clustering framework rather than relying on global fleet averages. Comparing a heavily loaded vehicle with a light vehicle would yield inherently skewed results. For example, the heavier vehicle will operate with higher fuel consumption. Therefore, clustering allows for a more fair comparison by grouping vehicles with similar operational contexts.

The clustering methodology used in this thesis was derived from work by Seixas et al. [34], who applied a k-means clustering algorithm to histograms of Volvo truck operation over a defined temporal window. This process identified patterns in driving behaviour and environmental conditions, clustering the fleet into more homogenous groups based on three dimensions:

- **Slope:** Classifies the slope of terrain into four distinct levels.
- **Speed:** Vehicles are grouped into 11 different clusters based on their average operational velocity.
- **Gross Combined Weight (GCW):** Reflects the average load the vehicle carries, also divided into 11 distinct clusters.

Because every vehicle is simultaneously assigned to one category within each of these three dimensions, there are $4 \cdot 11 \cdot 11 = 484$ possible unique combinations.

To translate these combinations into real-world utilization, they are mapped into a set of "buckets". A bucket represents a collection of cluster combinations that correspond to a particular truck modification and its intended operational segment. For instance, a single bucket encapsulates vehicles that share a specific GCW cluster and a specific range of speeds, encompassing all variations of slope within those parameters.

For clarity and readability, the term *segment* will be used interchangeably with bucket throughout the remainder of this thesis, as each bucket represents a distinct

operational segment of the fleet.

While the dataset contains many of these buckets, this project focused on a subset. Based on the recommendation of a domain expert at Volvo, the investigation was narrowed to two specific operational buckets, 7XT and 5XT. These represent the primary mass ranges in which the vehicle operates in, 70–79 tons for 7XT and 50-59 tons for 5XT.

In the Latin American market, the 7XT bucket represents the most common bucket. The 5XT bucket was selected as a comparative segment proposed by the domain expert at Volvo. By comparing these two buckets it allows us to determine if there are any operational differences that exist. Ultimately, this approach could allow for more precise feedback in the end.

As shown in Table 4.1, the dataset also includes several variants of the Volvo D13 engine family across the two investigated buckets. Although these engines share the same base platform, they differ in power ratings and calibration (e.g., 460, 500 and 540 horsepower), which can influence fuel consumption characteristics under different operating conditions.

Table 4.1: Distribution of engine versions across investigated buckets.

Engine Version	Count
D13K540	11619
D13K500	45
D13K460	23

(a) Engine counts for 7XT

Engine Version	Count
D13K540	464
D13K500	728
D13K460	992

(b) Engine counts for 5XT

To ensure fairer comparisons between vehicles within each segment, trucks equipped with different engine variants were filtered so that only vehicles with the same engine configuration were included in the analysis. For each segment, the engine version with the largest number of observations was selected. This filtering step should reduce variability caused by potential mechanical differences between vehicles and helps ensure that observed differences in fuel efficiency are primarily attributable to driving behaviour rather than underlying hardware characteristics. A simple overview of how the data filtering is combined together can be seen in Figure 4.2.

4.1.2 Description of Telemetry Variables

The telemetry data contained approximately 300 parameters derived from truck telematics, describing different aspects of vehicle operation and driving behaviour. These parameters are generated by onboard vehicle systems and represent aggregated measures of vehicle usage and driving patterns.

For the purpose of this project, the variables were broadly categorized into two categories: actionable variables and non-actionable variables. Actionable variables describe aspects of driving style that are directly influenced by the driver, such as cruise control usage or idling time. Variables like these are particularly relevant for

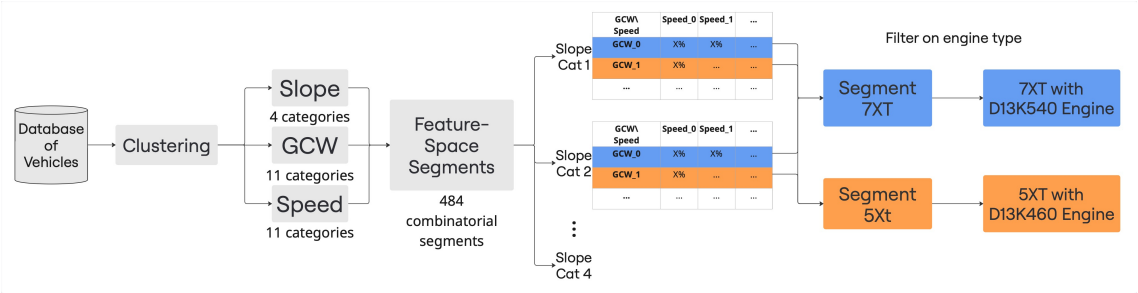


Figure 4.2: Flowchart visualization of the process of combining clusters into the selected segments.

eco-driving analysis, as they represent behaviours that can potentially be modified through driver coaching.

Non-actionable variables describe operational conditions under which the vehicle is driven, as well as internal vehicle states. Examples include factors like cargo weight or environmental conditions, such as road gradient, as well as variables describing internal system behaviour. While such variables often directly influence fuel efficiency, they are generally not directly controllable by the driver.

An illustrative subset of telemetry variables and their classification is shown in Table 4.2. These variables represent the detailed feature space used for analysis and should be distinguished from the higher-level clustering dimensions shown in Figure 4.2, which are derived from aggregated patterns of similar signals.

Table 4.2: Illustrative subset of telemetry variables used in the analysis and their classification as actionable or non-actionable.

Feature	**Description**	**Category**
IDLE_TIME	Percentage of total time spent idling	Actionable
SVC_BRAKE	Number of service brake activations per 100km driven	Actionable
CRUISE_TIME	Percentage of total time using cruise control	Actionable
TOTAL_DIST	Total distance driven in km	Non-actionable
SLOPE_6_9	Percentage of total driven distance occurring on road segments with a gradient between 6-9%	Non-actionable
AVG_GB_OIL_TEMPERATURE	Average gearbox oil temperature during operation (C)	Non-actionable

As shown in Table 4.2, non-actionable variables include both external driving conditions and internal vehicle characteristics, highlighting that this category extends beyond the variables used in the clustering framework.

All variables in the dataset were aggregated over predefined observation windows at the vehicle level. Aggregation primarily consisted of average values calculated over time periods ranging from weekly to monthly intervals. Each row in the dataset

therefore represents one aggregated observation for a specific vehicle over a given time window. As a result, multiple observations may exist for the same vehicle across different time periods.

4.2 Fuel Efficiency Modelling

This section describes the modelling framework used to analyse how different driving behaviours influence vehicle fuel efficiency. Based on the processed telemetry data described in the previous section, predictive models were developed to estimate fuel efficiency from aggregated driving behaviour variables. Particular emphasis was placed on using interpretable modelling techniques in order to identify driving behaviour patterns that are associated with higher or lower fuel efficiency. These insights form the basis for the context-aware eco-driving feedback explored later in the thesis.

4.2.1 Problem Formulation

The objective of the predictive modelling stage was to learn the relationship between driving behaviour and fuel efficiency based on the available vehicle telemetry data. The task can therefore be formulated as a supervised regression problem, where the goal is to predict the fuel efficiency of a vehicle based on a set of observed driving behaviour variables.

Let $\mathbf{X} = [x_1, x_2, ..., x_p]$ denote the vector of input features representing aggregated driving behaviour for a given observation window. Each feature x_i corresponds to one actionable parameter describing aspects of vehicle operation, as described in section 4.1.2.

The target variable y represents fuel efficiency, defined as kilometres driven per litre of fuel (km/L). Higher values of y therefore indicate more fuel-efficient driving.

Given a dataset consisting of n observations, the objective is to learn a function

$$f(\mathbf{X}) \rightarrow y \tag{4.1}$$

that maps the input features to the corresponding fuel efficiency value. The learned function f can then be used to estimate fuel efficiency for previously unseen observations based on their driving behaviour characteristics.

4.2.2 Data Cleaning and Preprocessing

Prior to model training, a series of preprocessing steps were applied to improve data quality and achieve comparability between observations. These steps included filtering of vehicles, feature selection, handling of missing values, removal of redundant variables, and treatment of extreme values.

The original telemetry dataset contained approximately 300 parameters, as described in Section 4.1.2. To reduce noise and avoid information leakage, a series of filtering steps were applied:

1. **Removal of features with excessive missing values.** Variables with more than 15% missing observations were removed in order to avoid excessive imputation and potential bias in the modelling stage. Previous similar work within Volvo had applied a threshold of 10%, but this threshold was increased to 15% in order to retain more potentially useful data.

2. **Removal of fuel-related variables.** Parameters directly related to fuel consumption, such as `TOTAL_FUEL`, `IDLE_FUEL`, and `TOTAL_CONSUMPTION`, were removed to prevent information leakage into the prediction of the target variable.

3. **Exclusion of non-actionable parameters.** Variables previously classified as non-actionable were excluded from the modelling stage, since the goal of the analysis was to focus on driving behaviours that can potentially be influenced through driver coaching.

4. **Removal of redundant representations.** Some behavioural parameters were available both as percentages of driving time and as percentages of driven distance (e.g., `CRUISE_TIME` and `CRUISE_DIST`). In such cases, only the time-based representation was retained in order to maintain consistency across features and reduce redundancy between highly correlated variables.

To further reduce redundancy among the remaining variables, a correlation analysis was performed across the remaining features. When two variables exhibited a Pearson correlation coefficient greater than 0.85, one of the variables was removed. The purpose of this step was to reduce multicollinearity and to prevent similar behavioural patterns from being overrepresented in the model. After applying these filtering steps, a total of 24 telemetry features remained for both the 5XT and the 7XT segments. These can be seen in Table 4.4.

In addition, extreme values were filtered to mitigate the impact of potential sensor noise or data recording errors. Observations falling outside the 1st and 99th percentiles of each numerical feature were removed. This trimming approach ensured that extreme sensor readings did not disproportionately influence the model training process. Vehicles that had driven less than 100 km for the time period was also removed, after recommendation from domain experts at Volvo. Table 4.3 provides an overview of the resulting datasets.

After preprocessing, the remaining variables were used to construct the input feature matrix $\mathbf{X}$ and the target vector y. Each observation corresponded to an aggregated driving period for a specific vehicle, with features representing aggregated driving behaviour metrics and the target variable representing fuel efficiency.

Table 4.3: Overview of the final datasets used for modelling after preprocessing.

Segment	Observations	Unique Vehicles	Features	Engine
7XT	74240	11159	24	D13K540
5XT	4531	910	24	D13K460

4.2.3 Model Selection

To model the relationship between driving behaviour variables and fuel efficiency, a tree-based gradient boosting model was selected. Specifically, the Light Gradient Boosting Machine (LightGBM) [61] algorithm was used for the predictive modelling task.

LightGBM is an ensemble learning method based on gradient boosting decision trees, where multiple weak learners are sequentially combined to improve predictive performance [61]. Tree-based boosting models are suited for tabular datasets, as they can effectively capture non-linear relationships and complex interactions between variables without requiring extensive feature engineering or feature scaling. These properties make LightGBM a suitable choice for modelling vehicle telemetry data, where relationships between driving behaviour and fuel efficiency may be non-linear. The LightGBM algorithm was implemented using the LightGBM Python package, specifically the `LGBMRegressor` implementation [62].

In addition to predictive performance, model interpretability was an important consideration in this study. Tree-based models are compatible with post-hoc explanation techniques such as SHAP, which allow the contribution of individual features to model predictions to be analysed (as described in Section 3.3). This capability is particularly relevant in the context of driver coaching, where understanding the influence of specific driving behaviours is essential for generating actionable feedback.

As a baseline comparison, a ridge regression model was also evaluated. Ridge regression provides a regularized linear modelling approach that can serve as a useful reference point for assessing the benefits of more flexible non-linear models. Since linear models are sensitive to feature scaling, the input features were standardized prior to training using `StandardScaler` from the scikit-learn library [63]. In addition, median imputation was applied using `SimpleImputer` also from the scikit-learn library to handle any remaining missing values in the data. The ridge model was implemented using the `Ridge` implementation from the scikit-learn library. However, tree-based gradient boosting models were ultimately selected as the primary modelling approach due to their ability to capture more complex relationships in the data.

4.2.4 Training Procedure

Predictive models were trained separately for each operational segment using the corresponding subset of vehicles. In this study, models were trained for both the 5XT and 7XT segments.

To evaluate model performance while avoiding information leakage, grouped cross-

Table 4.4: The remaining features used in the modelling stage

No.	Feature	Description
1	ACC_1250_1400_RPM	% of total time where the engine operates between 1250-1400 RPM while producing torque.
2	ACC_900_1250_RPM	% of total time where the engine operates between 900-1250 RPM while producing torque.
3	BRAKE_MODE_ACT	Number of activations of the Brake program.
4	BRAKE_TIME	% of total time spent in Brake mode.
5	COASTING_DIST	% of total distance spent coasting (vehicle moving without fuel injection).
6	COASTING_SPEED	Average vehicle speed during coasting periods.
7	CRUISE_TIME	% of total time spent using cruise control.
8	DRIVE_SPEED	Average driving speed, calculated as total driving distance divided by total driving time.
9	DRIVING_MODE_E_DIST	% of total distance driven in Economy mode.
10	ECOROLL_ACT_DIST	% of total distance where the Eco-Roll function is active.
11	ECOROLL_ENA_DIST	% of total distance where the Eco-Roll function is enabled.
12	GB_AUT_TIME	% of total time the gearbox operates in automatic mode.
13	GB_HIGH	% of total distance where the gearbox operates in the high gear range.
14	GB_MAN_TIME	% of total time the gearbox operates in manual mode.
15	IDLE_IROLL	% of total time spent in a low engine speed and torque state associated with I-Roll operation.
16	IDLE_TIME	% of total time spent idling (vehicle stationary with engine running).
17	KICKDOWN_MODE_ACT	Number of activations of the Kickdown mode.
18	KICKDOWN_MODE_TIME	% of total time the Kickdown mode is active.
19	MAX_ENG_SPEED	Maximum recorded engine speed during engine overspeed events.
20	PEDAL_TIME	% of total time spent in Pedal mode.
21	POWER_MODE_TIME	% of total time spent in Performance mode.
22	SVC_BRAKE	Number of service brake activations per 100 km driven.
23	TORQUE_MODE_3	% of total engine operating time spent in a specific engine speed and torque operating range.
24	TORQUE_MODE_30	% of total engine operating time spent in a specific engine speed and torque operating range.

validation was applied. Because the dataset contains multiple observations from the same vehicle across different time periods, a GroupKFold strategy was used with vehicle ID as the grouping variable. GroupKFold ensures that all observations belonging to the same group remain within a single fold, preventing observations from the same vehicle from appearing in both training and validation sets [64]. This reduces the risk of overly optimistic performance estimates caused by the model learning vehicle-specific patterns rather than general relationships in the data.

Cross-validation was performed using five folds. In each iteration, the model was trained on four folds and evaluated on the remaining fold. Final performance metrics were obtained by averaging the results across all folds.

The LightGBM models were implemented using the `LGBMRegressor` class from the LightGBM Python package. The ridge regression baseline was implemented using the `Ridge` estimator from the scikit-learn library within a preprocessing pipeline. Since linear models are sensitive to feature scaling, the ridge model pipeline included feature standardization using `StandardScaler` and median imputation of missing values using `SimpleImputer`.

For the ridge regression baseline, a regularization parameter of $\alpha = 1.0$ was used, corresponding to the default value in the scikit-learn implementation. Hyperparameters for the LightGBM models were selected through a targeted tuning procedure described in Section 4.2.5.

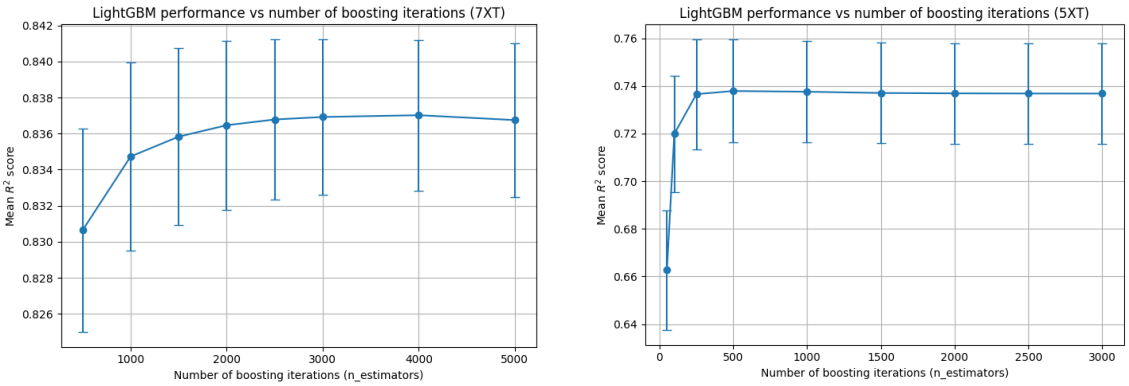
4.2.5 Hyperparameter Tuning

Hyperparameters of the LightGBM models were selected through a two-stage tuning procedure. The objective of this process was to identify model configurations that provided strong predictive performance while avoiding unnecessary model complexity.

In the first stage, the learning rate and number of boosting iterations were examined. Gradient boosting models build trees sequentially, and the number of boosting iterations therefore directly affects the capacity of the model. A small grid search was performed over different learning rates and numbers of estimators while evaluating predictive performance using the grouped cross-validation procedure described in Section 4.2.4.

Figures 4.3a and 4.3b show the cross-validated performance of the LightGBM model as a function of the number of boosting iterations for the 7XT and 5XT segments, respectively. In both segments, predictive performance improved as additional boosting iterations were added, but eventually reached a plateau where further increases provided only marginal improvements.

For the 7XT segment, which contains substantially more observations, the performance plateau occurred at approximately 3000 boosting iterations. For the 5XT segment, which contains fewer observations, the plateau occurred earlier at approximately 500 boosting iterations. Consequently, these values were selected as the final numbers of boosting iterations for the respective segment models. The learning rate



(a) 7XT segment

(b) 5XT segment

Figure 4.3: Cross-validated performance of the LightGBM model as a function of the number of boosting iterations. The plot shows the mean R^2 score across five GroupKFold validation folds for different values of `n_estimators`, with error bars indicating the standard deviation across folds.

selected through the grid search was 0.05 for both segments.

In the second stage, a targeted hyperparameter search was conducted to tune parameters controlling tree complexity and stochastic regularization. Specifically, the parameters `feature_fraction`, `bagging_fraction`, and `num_leaves` were explored through a grid search evaluated using the same grouped cross-validation procedure described in Section 4.2.4. The parameter `feature_fraction` determines the proportion of features randomly sampled for each boosting iteration, while `bagging_fraction` controls the proportion of training observations used when performing row subsampling. The parameter `num_leaves` controls the maximum number of leaves in each decision tree and therefore influences the complexity of the learned model.

Row subsampling was enabled by fixing the parameter `bagging_freq` to 1, ensuring that subsampling was applied at every boosting iteration when bagging was active. All remaining hyperparameters were kept at their default values in the LightGBM implementation.

The results of the hyperparameter search are illustrated in Figures 4.4 and 4.5, which visualize the mean cross-validated R^2 scores for different combinations of `feature_fraction` and `bagging_fraction` across the evaluated values of `num_leaves`. The results indicate that moderate levels of feature and observation subsampling generally improved predictive performance compared to using all features and observations. Additionally, smaller values of `num_leaves` tended to produce slightly better results, suggesting that limiting tree complexity helped reduce overfitting.

Based on these results, the final hyperparameter configurations summarized in Table 4.5 were selected for the LightGBM models in each segment.

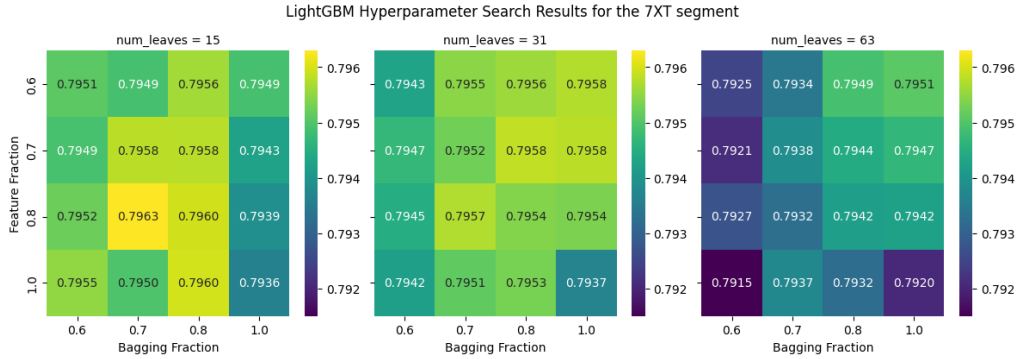


Figure 4.4: Results of the LightGBM hyperparameter search for the 7XT segment. The heatmaps show the mean cross-validated R^2 scores for combinations of `feature_fraction` and `bagging_fraction` for different values of `num_leaves`.

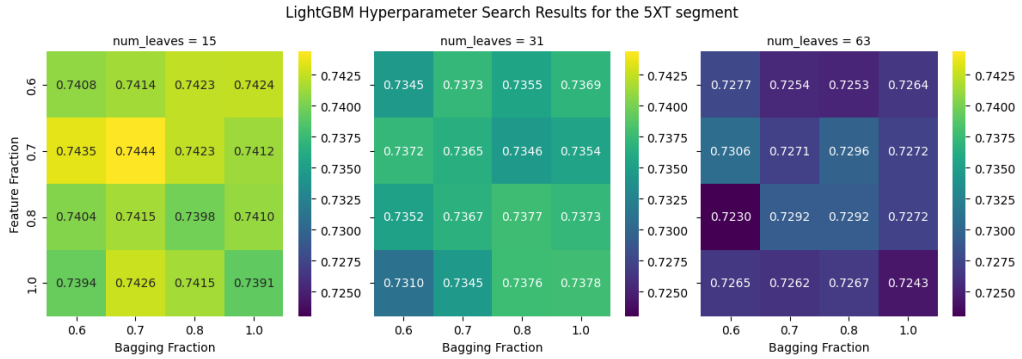


Figure 4.5: Results of the LightGBM hyperparameter search for the 5XT segment. The heatmaps show the mean cross-validated R^2 scores for combinations of `feature_fraction` and `bagging_fraction` for different values of `num_leaves`.

4.2.6 Model Evaluation

Model performance was evaluated using both the coefficient of determination (R^2) and the Mean Absolute Error (MAE), in order to capture complementary aspects of model performance.

The R^2 metric measures the proportion of variance in the target variable that is explained by the model predictions [65]. It is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{4.2}$$

where y_i represents the true target values, $\hat{y}_i$ the predicted values, and $\bar{y}$ the mean of the observed target variable. As such, R^2 provides an interpretable measure of how well the model captures the relationship between driving behaviour variables and fuel efficiency.

In contrast, the MAE quantifies the average magnitude of prediction errors in the original unit of the target variable [66]. It is defined as:

Table 4.5: Final LightGBM hyperparameters selected for each segment.

Segment	lr	n_estimators	feat_frac	bag_frac	leaves
7XT	0.05	3000	0.8	0.7	15
5XT	0.05	500	0.7	0.7	15

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{4.3}$$

providing a directly interpretable measure of how far predictions deviate from the true values on average.

The two metrics were used together to capture both explanatory power and predictive accuracy. While R^2 reflects the "goodness-of-fit" of the model [67], MAE provides insight into the practical magnitude of prediction errors. Given the objective of analysing relationships between driving behaviour and fuel efficiency, R^2 was used as the primary metric for model selection and hyperparameter tuning, as described in Section 4.2.5, while MAE served as a complementary evaluation metric.

Model performance was assessed using grouped cross-validation, as described in Section 4.2.4. For each fold, the model was trained on the training subset and evaluated on the validation subset, and the resulting R^2 and MAE scores were recorded. Final performance estimates were obtained by averaging the scores across the five cross-validation folds.

Model training and evaluation were implemented using the `cross_val_score` function from the scikit-learn library, with the scoring metrics set to `r2` and `neg_mean_absolute_error`.

4.3 Model Interpretation and Feature Importance

While the predictive models described in the previous section allow fuel efficiency to be estimated from driving behaviour variables, the objective of this study is not only to achieve accurate predictions but also to understand which driving behaviours influence fuel efficiency. Model interpretation methods are therefore required to analyse how individual input variables contribute to the predictions of the trained models. For this reason, SHAP values were computed for the trained LightGBM models.

4.3.1 SHAP Implementation

To interpret the predictions of the trained LightGBM models and identify which driving behaviour variables influence fuel efficiency, SHAP values were computed. As described in Section 3.3, SHAP provides local feature attributions that quantify how each input variable contributes to the model's prediction relative to the model's expected output.

Since the predictive models used in this study are based on gradient boosted decision trees, SHAP values were computed using the TreeSHAP algorithm implemented in the SHAP Python library. TreeSHAP is specifically designed for tree-based models and allows exact Shapley values to be calculated efficiently by leveraging the structure of the trained decision trees [45]. The SHAP values were obtained using the `TreeExplainer` class applied to the trained LightGBM models.

For each observation in the dataset, SHAP assigns a contribution value to every feature, indicating how that feature increases or decreases the predicted fuel efficiency relative to the model's expected value. The sum of all feature contributions together with the base value equals the model prediction for that observation.

To analyse global feature importance, SHAP values were aggregated across all observations by computing the mean absolute SHAP value for each feature. This measure reflects the overall influence of a variable on the model predictions across the dataset. In addition to global importance measures, SHAP summary plots were generated to visualize both the magnitude and direction of feature contributions. For selected observations, waterfall plots were also used to illustrate how individual driving behaviour variables contributed to the predicted fuel efficiency.

Because SHAP explanations depend on the background data used to represent the models expected prediction, different background sampling strategies were explored in this study. These strategies are described in the following subsection.

4.3.2 Background Dataset Selection

When computing SHAP values, feature contributions are calculated relative to a background dataset that represents the model's expected prediction. This background distribution is used to simulate missing features when evaluating the marginal contribution of individual variables. Consequently, the choice of background data influences how feature contributions are interpreted.

To account for potential dependencies between features, SHAP values were computed using the interventional feature perturbation strategy, which conditions feature perturbations on the provided background dataset. This approach allows feature contributions to be estimated while respecting the joint distribution represented by the background data [44].

In many applications, the background dataset is constructed by randomly sampling observations from the full dataset. This approach approximates the average data distribution and therefore results in feature contributions that describe how an observation differs from the typical behaviour observed in the dataset.

In this study, two different background sampling strategies were explored in order to analyse how the interpretation of driving behaviour contributions changes depending on the reference distribution.

1. **Fleet-average background distribution:** In the first approach, 200 observations were randomly sampled from the dataset to represent the overall

distribution of driving behaviour. This configuration provides explanations relative to the average driving behaviour within the analysed fleet segment.

2. **Fuel-efficient background distribution:** In the second approach, the background dataset was constructed by sampling 200 observations from the top 10% most fuel-efficient observations in the dataset. This alternative background distribution represents highly efficient driving behaviour and therefore allows feature contributions to be interpreted relative to efficient driving patterns rather than the fleet average.

The comparison of these two background datasets makes it possible to investigate how the importance and interpretation of driving behaviour variables change when predictions are explained relative to typical driving behaviour versus highly fuel-efficient driving behaviour. This distinction is especially relevant in the context of driver coaching, where the objective is not only to explain current behaviour but also to identify how drivers differ from more efficient driving patterns.

4.3.3 Feature Importance Analysis

To analyse how different driving behaviour variables influence the model predictions, SHAP values were examined at both the global and local levels. Global feature importance was assessed by aggregating SHAP values across all observations. For each feature, the mean absolute SHAP value was computed according to

$$\text{Importance}_j = \frac{1}{n} \sum_{i=1}^{n} |\phi_{ij}|, \tag{4.4}$$

where ϕ_{ij} represents the SHAP value of feature j for observation i. This measure reflects the overall influence of each feature on the model predictions across the dataset.

To visualise the distribution and direction of feature contributions, SHAP summary plots were generated. These plots display the magnitude of feature contributions for all observations while simultaneously showing how the value of each feature affects the predicted fuel efficiency.

In addition to global feature importance, local explanations were examined for individual observations using SHAP waterfall plots. These visualizations illustrate how specific driving behaviour variables contribute to the predicted fuel efficiency for a single observation relative to the models expected prediction.

Finally, to investigate the robustness of feature importance estimates, both the effect of different background datasets and the stability of feature importance across repeated model runs were considered. By comparing feature importance values obtained under different reference distributions and across multiple model initializations, the consistency of feature rankings can be assessed. In addition, segment-specific explanations were compared on held-out observations to assess whether identical driving behaviour was interpreted differently by the two segment models.

4.3.4 Cross-Segment Explanation Comparison

To further assess whether the segment-specific models interpret driving behaviour differently, an additional comparison was performed using held-out observations that were not included in the training data. The same observations were passed through both the 5XT and 7XT models, and SHAP explanations were generated for each model using the fuel-efficient background dataset. This made it possible to compare how identical driving behaviour was interpreted under different segment-specific models.

The comparison was conducted separately for held-out observations originating from the 5XT and 7XT segments. The held-out test set consisted of 32 observations from the 5XT segment and 36 observations from the 7XT segment, none of which were included in the model training process. For each observation, the predicted fuel efficiency from the two models was compared, and the corresponding SHAP explanations were analysed across the shared input features.

Several metrics were used to quantify the similarity between the two explanations. First, the prediction difference was computed as the difference between the fuel-efficiency predictions produced by the 5XT and 7XT models for the same observation. Second, cosine similarity was used to compare the full SHAP attribution vectors, providing a measure of how similar the overall direction of the explanations was [68]. Cosine similarity ranges from -1 to 1, where higher values indicate that the two models assign feature contributions in a more similar overall direction and relative pattern. It is formally defined in Eq. 4.5. Third, sign agreement was computed as the proportion of features for which the two models assigned contributions with the same sign. This indicates whether the models agree on whether each feature increases or decreases the predicted fuel efficiency. Finally, the strongest positive and strongest negative contributors were compared to assess whether the two models identified the same main beneficial and detrimental driving behaviours for each observation.

In addition to these observation-level metrics, feature-level paired comparisons were conducted for the most influential variables identified in the global SHAP analysis. For each selected feature, SHAP values generated by the 5XT and 7XT models for the same observations were compared using paired t-tests [69]. Since multiple features were tested simultaneously, p-values were adjusted using false discovery rate correction. These comparisons were used to evaluate whether differences in feature attribution between the segment-specific models were systematic across the held-out observations.

4.4 RAG Architecture

In an effort to ground the LLMs responses in factually correct documents, a RAG pipeline was implemented. The pipeline is structured into four core phases: ingestion and chunking, vectorization, retrieval, and context assembly. The pipeline begins by extracting raw text from Volvo documents. For this project, three different

documents were used:

- One Vehicle Manual from 2012. It consists of information mainly regarding the vehicle, e.g. how infotainment screen functions, but also shorter driver instructions.
- Two trainer-booklets, one from 2018 and one from 2022, containing driving tips for how to drive in the best ways.

Including the entire volume of these manuals in every query would make the cost unnecessarily expensive, and likely give poor results. Therefore, the text was partitioned into manageable segments known as chunks. Each chunk consists of a set amount of tokens, where each token represents approximately four characters. To ensure that sentences are not arbitrarily cut off, which can strip the text of its context, it is conventional that the system utilizes a sliding window approach with a defined token overlap. To determine the optimal chunk size, overlap length, and the number of chunks to retrieve (k), an evaluation environment was established using the Ragas framework [70]. Ragas ranks RAG architectures across four key metrics: precision, recall, faithfulness, and relevance. We evaluated across the four metrics, evaluating for different combinations of chunk sizes (64–256 tokens), overlaps (0–30 tokens), and retrieval counts (1–5 chunks). The tested configurations are seen in Table 4.6. The final system utilizes a fixed chunk size of 256 tokens, a 30-token overlap and a retrieval count of $k = 3$ chunks. While larger configuration yielded marginally higher metric scores, this specific setup provided a good balance between high-quality context, latency, and operational cost.

Table 4.6: Evaluation of RAG parameters across various metrics.

Chunk	**Overlap**	**Top K**	**Prec.**	**Recall**	**Faith.**	**Relev.**	**Time**	**Mean**
64	10	2	0.667	0.604	0.944	0.627	202	0.710
64	10	3	0.661	0.629	0.958	0.707	210	0.739
64	10	5	0.539	0.629	0.889	0.753	243	0.703
128	20	2	0.633	0.413	**1.000**	0.617	**155**	0.666
128	20	3	0.667	0.696	0.920	0.726	172	0.752
128	20	5	0.658	0.777	**1.000**	0.791	224	0.807
256	30	2	**0.833**	0.594	0.850	0.724	516	0.750
256	30	3	0.800	0.850	0.950	0.766	236	0.842
256	30	5	0.745	**0.975**	**1.000**	**0.861**	345	**0.895**

Once segmented, indexing is performed once prior to system startup to minimize runtime delays. Each text chunk is transformed into a vector representation. For this project, OpenAI's `text-embedding-3-small` model was selected via Volvos internal Azure AI Hub. As seen in Table 4.7, this model provides a balance of cost and performance, outperforming older architectures on the Massive Text Embedding Benchmark (MTEB). MTEB evaluates embedding models across tasks such as semantic similarity and information retrieval [71]. Strong performance on the MTEB

indicates that the model produces embeddings suited for the semantic search utilized in this system.

The chosen model maps the text into a vector space where geometric proximity reflects semantic similarity. Evaluating this proximity is done via Cosine Similarity, which calculates the cosine of the angle θ between two non-zero vectors, A and B:

$$\text{Cosine Similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} \tag{4.5}$$

Before the vectors are stored in the system's global state, L2 normalization is applied, scaling each vector to a unit length of one ($\|A\| = \|B\| = 1$). This transformation simplifies the retrieval phase, allowing the system to compute Cosine Similarity utilizing a dot product:

$$\text{Similarity} = A \cdot B = \sum_{i=1}^{n} A_i B_i \tag{4.6}$$

Table 4.7: Embedding models evaluated via Volvos Internal Azure AI Hub.

Embedding Model	**Provider**	**Cost (USD / 1M tokens)**	**MTEB Eval.**
text-embedding-ada-002	OpenAI	\$0.10	61.0%
text-embedding-3-large	OpenAI	\$0.13	**64.6%**
text-embedding-3-small	OpenAI	**\$0.02**	62.3%

During inference, the system employs a dual-retrieval approach to provide context. Using the same embedding model, the system vectorizes both the user's explicit query and a secondary query targeting the driver's most critical area for improvement (i.e., their lowest-scoring eco-driving metric). The system then calculates the dot product between these query vectors and the pre-computed document matrix to determine semantic proximity. These top-ranked snippets are then combined into a string which is injected into the prompt.

4.5 Coaching generation

Inspired by the work of Costa et al. [11], as mentioned in Section 2.3, the objective of the coaching system is not only to explain model predictions but to translate them into concise, actionable feedback that drivers can realistically apply during daily operations. To achieve this, the system combines model interpretability techniques with natural language generation. SHAP explanations are first used to identify the most behaviourally relevant driving variables for a trip. These signals are then contextualized using a structured knowledge base and translated into conversational coaching messages by an LLM.

4.5.1 Natural Language Generation

To generate the coaching feedback and handle driver inquiries, an LLM was integrated into the system pipeline. This ensures that truck data is translated into easily understandable and conversational responses for the user.

There is a large amount of different LLMs, with models often specalized for distinct use cases and architectural constraints. Therefore, finding an objective and fair comparative score between models is hard to come by. This study utilizes the Holistic Evaluation of Language Models (HELM) capabilities score developed by Liang et al.[72]. For the purpose of this thesis and to establish a common ground of comparison, the HELM Capabilities mean score serves as the primary evaluative baseline for model performance. Consequently, model selection was restricted to models currently ranked within the HELM framework.

To ensure a diverse pool of model providers, we limited models from the same model provider, apart from OpenAI which we explain further down. A comprehensive comparison of the investigated models is provided in Table 4.8. This table details each model's provider, HELM Capabilities mean score, Context window and the operational cost per million input tokens. Furthermore, it tracks licensing status (open-source vs closed-source), inference speed (measured as Time to First Token), CO_2 emissions and their availability on Volvo's Internal Azure AI Foundry. Because no provider publishes CO_2 emissions, these are estimated based on the Green Algorithms framework [73]. Information regarding context window, operational cost and inference speed are all gathered from Artificial Analysis [74].

Requirements for the project's LLM were divided into two primary categories: technical performance and corporate compliance. On the technical side, each model was evaluated across four metrics: reasoning capabilities, operational cost, inference speed and carbon footprint.

Because the application is designed for mobile use where user experience relies on quick interactions, low latency is important. Furthermore, since the model's task involves interpreting structured truck telemetry data rather than executing complex cognitive tasks (e.g., code generation), massive parameter models were deemed unnecessary. Generally, utilizing smaller models reduces the cost per token. Finally, since this project overall is focused on reducing fuel consumption of trucks and thus emissions, it would be contradictory to power the application via a model that is bad for the environment.

Beyond technical specifications, the model had to comply to Volvo's corporate data governance and privacy policies. This required the model to be available and pre-approved within Volvo's internal Azure AI Foundry. This constraint significantly influenced the selection process, leading to focus primarily on the OpenAI models.

Among the available candidates, OpenAI's GPT-5-Nano seemed as the best option fulfilling our initial requirements. According to the sources, it offered low latency and low operating costs. However, during implementation, practical performance deviated from these expectations. The latency was significantly higher than that of other available GPT models. This is likely because the local Azure AI Foundry

Table 4.8: LLMs investigated for this project.

Model	Provider	HLEM	Context	Cost/1M	Open	TTFT	CO_2e (g)	Azure
Claude 4 Opus (Ext.)	Anthropic	0.780	**1M**	$15.00	No	1.51s	∼4.2	No
DeepSeek-R1-0528	DeepSeek	0.699	130k	$1.35	Yes	0.95s	∼1.5	No
Gemini 3 Pro (Prev.)	Google	0.799	**1M**	$2.00	No	0.60s	∼1.8	No
GPT 4.1	OpenAI	0.727	**1M**	$2.00	No	0.85s	∼2.1	Yes
GPT 4.1 Mini	OpenAI	0.726	**1M**	$0.40	No	0.65s	∼0.5	Yes
GPT 4.1 Nano	OpenAI	0.616	**1M**	$0.10	No	0.48s	∼0.2	Yes
GPT-5 Mini	OpenAI	**0.819**	128k	$0.30	No	0.55s	∼0.4	No
GPT-5-Nano	OpenAI	0.748	128k	**$0.05**	No	**0.35s**	**∼0.1**	Yes
Grok 4	xAI	0.785	256k	$3.00	No	1.10s	∼3.5	No
Kimi K2 Instruct	Moonshot AI	0.768	256k	$0.57	Yes	0.75s	∼1.2	No
Llama 4 Mav. FP8	Meta	0.718	128k	-	Yes	0.45s	∼0.8	No
Mistral Small 3.1	Mistral	0.558	128k	$0.30	Yes	0.40s	∼0.3	Yes
Nova Premier	Amazon	0.637	**1M**	-	Yes	0.80s	∼1.6	No
Qwen3 235B A22B	Alibaba	0.798	256k	$0.071	Yes	0.38s	∼0.9	No

infrastructure does not prioritize this model, resulting in slower execution times compared to alternatives. As such, GPT-5-Nano was ultimately discarded.

Instead, OpenAI's GPT-4.1 Mini was selected for the final implementation. As shown in Table 4.8, it achieves a high HELM score while maintaining a relatively low operational cost. Furthermore, testing within the Azure AI Foundry environment showed that the latency was significantly lower than that of alternatives such as GPT-5-Nano, proving adequate for the project's requirements.

4.5.2 Translating SHAP Explanations into Feedback

To generate actionable coaching feedback, the SHAP explanations produced by the predictive model had to be translated into interpretable insights for the driver. SHAP values quantify how individual driving behaviour variables contribute to the predicted fuel efficiency relative to a chosen reference distribution. The choice of this reference baseline therefore directly influences how driver performance is framed.

Presenting all drivers with explanations relative to highly fuel-efficient driving patterns may highlight opportunities for improvement, but may also result in exclusively negative feedback for drivers whose performance is substantially below that level. Such feedback may reduce motivation and engagement, as drivers may perceive the improvement gap as unrealistic.

To address this, the coaching system adopted an adaptive baseline strategy. When a driver's predicted fuel efficiency exceeds the random average performance within the corresponding operational segment, explanations are generated relative to the fuel-efficient background distribution described in Section 4.3.2. This allows drivers to compare their behaviour with highly efficient driving patterns and identify further improvement opportunities. Conversely, when a driver's predicted efficiency falls below the fleet average, explanations are generated relative to the random average background distribution. This configuration should enable the system to highlight

both positive and negative behavioural contributions regardless of the driver's initial skill level, thereby supporting constructive feedback that reinforces existing efficient behaviours while still identifying areas for improvement.

After computing the SHAP explanations, the system identifies the feature with the largest positive contribution and the feature with the largest negative contribution. These two features represent the behaviours that most strongly improve or reduce the predicted fuel efficiency for the analysed trip. In order to avoid overwhelming the driver with excessive information, the feedback focuses only on these two signals. Finally, the model can decide whether or not the negative behaviour should be increased or decreased upon. The selected features, together with their corresponding feature values and SHAP contributions, are then forwarded to the coaching generation pipeline.

4.6 Dynamic Prompt Construction

The generation of feedback is based on dynamic prompt assembly, rather than a static instruction sent to the LLM. The prompt acts as the foundation; it provides the exact context and instructions required to formulate the feedback. The system constructs the prompt by combining four components: a persona, telemetric data via SHAP, persistent semantic memory and retrieved knowledge context.

4.6.1 Persona and Behavioural Guardrails

When reasoning about how the feedback from the LLM should be formatted for the driver, the structure and constraints of the system prompt are critical. Following the prompting strategies outlined by Kong et al. in Section 3.6, the model is first assigned a specific persona: `"an expert Volvo Truck driver instructor specializing in eco-driving and fuel efficiency."` The motivation for this design choice is due to their findings that assigning a persona serves as an implicit CoT trigger, thereby improving the model's reasoning capabilities.

This role-setting also provides significant operational benefits. As identified in the core prompting principles established by Marvin et al., in Section 3.6, the assignment of a persona provides the necessary context to limit the model's operational scope strictly to the heavy-trucking domain. This defines a clear objective within the system prompt, ensuring the generation of focused, domain-specific outputs. At the same time, the primary goal of the system is set: to deliver feedback tailored to the specific professional needs of truck drivers.

To govern the interaction model between the driver and the LLM, a set of behavioural guardrails was established. Given that the application operates within the professional context of heavy-duty trucking, the written output must reflect high levels of reliability and authority. The model is thus prompted to maintain a tone that is `"trustworthy, approachable, intelligent and positive,"` adopting a helpful and professional persona.

4.6.2 Targeting via SHAP

To structure the generated feedback, a specific format was established following discussions with stakeholders at Volvo: the LLM must highlight one positive driving behaviour and provide one constructive critique per trip. To fulfil this requirement, the system utilizes the driver's SHAP array (as detailed in Section 4.5.2) to isolate the highest and lowest scoring metrics. These two data points act as the targeted focus areas for the prompt. By constraining the LLM's attention to these specific metrics, the system guarantees that the resulting feedback is grounded in the driver's actual performance, while simultaneously setting the exact search parameters for the subsequent knowledge retrieval phase.

To ensure the LLM accurately interprets these metrics, a semantic lookup table formatted in JSON was utilized. This mapping translates internal feature names (from Table 4.4) into human-readable labels and short descriptions, giving the model the necessary context for feedback generation. For example, the variable `SVC_BRAKE` (Feature 22) is translated into the human-readable label "Service Brake," accompanied by the description: "Number of service brake activations per 100 km. The counter increases each time the brake pedal is actuated." These extracted features not only ground the feedback in observed driving behaviour, but also serve as structured inputs to the subsequent prompt construction process.

In addition to the static trip-based targeting, the system incorporates a lightweight form of temporal memory to support continuity in driver coaching. While the SHAP-based analysis identifies the most relevant behaviours for a given trip, it does not by itself capture how driver performance evolves over time.

To address this, previously generated coaching feedback is stored across reporting periods in a persistent structure indexed by vehicle identifier. Each record contains the previously targeted feature, the recommended improvement direction, and the observed feature value for that period. At runtime, the system retrieves the most recent prior record and compares it to the current observation, enabling the identification of whether the driver has improved, maintained, or worsened performance with respect to previously highlighted behaviours.

This information is incorporated into the prompt as a structured follow-up signal, allowing the LLM to acknowledge progress or reinforce areas that still require attention. In this way, the system moves beyond single-instance feedback and supports coaching over time. This aligns with the concept of long-term memory in conversational systems discussed in Section 3.7, where stored information across interactions enables more consistent and personalized system behaviour over time.

4.6.3 Persistent Semantic Memory Layer

A key limitation of LLMs, particularly when accessed through API-based services, is the absence of persistent memory. As discussed in Section 3.7, model outputs are conditioned solely on the current prompt and its context window. Consequently, user-specific information such as preferences, goals, and constraints is not retained

across interactions, leading to generic responses that may ignore previously expressed user needs.

This limitation is particularly problematic in driver coaching scenarios, where effective feedback relies on adapting to individual drivers over time. Personalization therefore requires mechanisms for storing and reusing user-specific information beyond a single interaction, corresponding to the concept of long-term memory in conversational systems [59].

To address this, a lightweight semantic memory layer was implemented to enable persistent personalization across sessions. While existing frameworks such as Mem0 [75] provide modular memory capabilities, a custom solution was developed in this work. This decision was motivated by practical considerations, including limitations in available versions and the need to maintain full control over stored data and system behaviour.

An initial rule-based approach using regular expressions was explored for extracting user-specific information. However, this approach proved too rigid to capture the variability of natural language input. Instead, an LLM-based extraction method was adopted, allowing the system to identify semantically meaningful information from user messages. This design is conceptually inspired by recent work on memory-augmented conversational agents, including Mem0, but represents a simplified implementation tailored to the application domain.

The memory layer stores a constrained set of structured user-specific attributes to support personalization. Each memory item is represented as a structured record consisting of a type, key, and value. The design is intentionally limited to a predefined schema in order to ensure interpretability and controllability of stored information.

The following memory types are used:

- **Preferences**: Describe how responses should be presented, including verbosity, tone, and formatting.
- **Goals**: Capture the primary long-term objective of the driver, such as improving fuel efficiency.
- **Constraints**: Represent explicit limitations or topics that should be avoided in generated responses.
- **Context**: Store stable background information, such as the driver's name or operational driving context.

Memory items are extracted using a lightweight LLM (GPT-4.1 Nano via Volvo GenAIHub) configured for deterministic behaviour. Given the current user message, together with limited conversational context, the model is prompted to identify stable and reusable information and return structured memory entries in a predefined JSON format. The full extraction prompt is provided in Appendix B.1.

Extracted memory items are stored on a per-user basis in a persistent JSON structure, enabling reuse across sessions. An example of the JSON store for one user

can be seen in Appendix B.2. The storage mechanism supports both overwriting and accumulation depending on the memory type. For example, preferences such as verbosity may be updated over time, whereas constraints are accumulated to reflect multiple user-defined limitations.

To ensure reliability, several constraints are applied during memory formation. These include restricting the number of extracted items, enforcing valid typekey combinations, and filtering invalid or placeholder values (e.g., "null" or "unknown"). Memory extraction is therefore treated as a binary decision process in which candidate items are either stored or discarded based on these validation rules.

At runtime, stored memory items are retrieved and incorporated into the system prompt to guide response generation. This allows the system to adapt responses to individual users while avoiding excessive prompt length. This memory representation enables the system to move beyond single-turn interactions and instead maintain continuity across sessions, supporting the individualized coaching strategies discussed in Section 3.7.

4.6.4 Context Assembly and Prompt Fusion

Before interacting with the primary generation LLM, the different system components (as described above) are combined into a single prompt. The system takes the retrieved manual snippets, the targeted SHAP telemetry, and the retrieved personalization context from the semantic memory layer. By directly injecting these modules into the prompt, the system leverages the LLM's existing reasoning capabilities while maintaining control over personalization and factual accuracy. This augmented instruction block is combined with the user's input within the processing pipeline, ensuring that responses are conditioned on both the current query and the constructed contextual information. A full excerpt of a prompt is provided in Appendix C.

4.7 Mobile Application Prototype

The prototype developed in this thesis was chosen to be a mobile phone application, to demonstrate its practical applicability. The selection of a mobile application as the interface was inspired by the existing research discussed in Section 2.6. As implemented by Magaña and Muñoz-Organero [36] and Massoud et al. [32], mobile platforms offer a versatile way of combining vehicle data with a user-centric feedback loop.

The purpose of the application was not to develop a production-ready solution, but to provide a functional environment in which data processing, predictive modelling, personalization, and language generation components could be integrated and evaluated as a cohesive system. The focus of the implementation was placed on the response generation, rather than on UX optimization. Consequently, the application was developed as a lightweight prototype intended to validate system functionality and component interoperability.

An overview of the complete system architecture and information flow is presented in Figure 4.6. The figure illustrates how the mobile application interacts with the back-end pipeline, where telemetry-derived insights, personalization components, historical coaching context, and retrieved external knowledge are combined through dynamic prompt construction before being processed by the LLM.

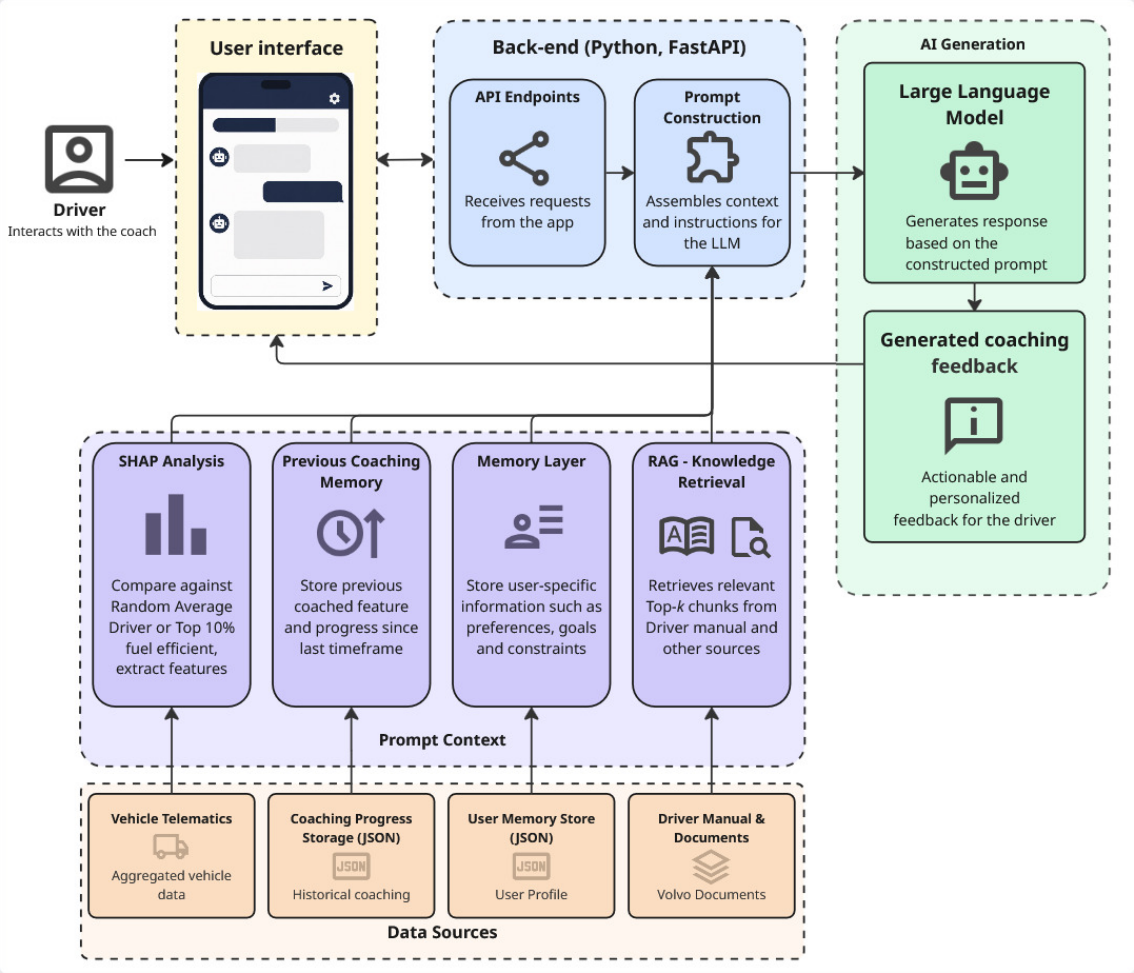


Figure 4.6: A visualization of the complete pipeline of the prototype

4.7.1 Back-end

The back-end architecture was implemented in Python due to its strong ecosystem for data processing, machine learning, and LLM-based applications. The back-end served as the orchestration layer of the system, integrating telemetry analysis, SHAP-based interpretation, semantic memory retrieval, prompt construction, retrieval-augmented generation, and response generation.

Communication between the mobile application and the back-end was exposed through REST-based API endpoints implemented using FastAPI. This enabled structured interaction between the client interface and the underlying conversational pipeline. Incoming requests were validated and processed before being passed through the prompt construction framework described in the previous sections.

The back-end further maintained persistent storage for personalization memory and historical coaching records using lightweight JSON-based persistence mechanisms. These components enabled personalization and temporal continuity across interactions without requiring external database infrastructure. The application was served using Uvicorn, a web server compatible with FastAPI, enabling lightweight deployment and real-time interaction during prototype testing.

4.7.2 Front-end

The prototype front-end was developed using Flutter to provide a mobile conversational interface for interacting with the coaching system. Flutter was selected due to its cross-platform capabilities and the development experience of the project team.

The application allowed users to interact with the system through both text and voice-based communication. In addition to presenting generated coaching feedback, the interface also exposed aspects of the personalization system through a dedicated settings view, where stored memory items could be inspected, edited, or removed. This was motivated by the transparency considerations discussed in Section 3.7, allowing users greater visibility and control over stored personalization data.

The front-end communicated with the back-end through REST-based API endpoints following a client-server architecture. All telemetry analysis, personalization logic, and LLM-based response generation were performed server-side, while the mobile application functioned primarily as an interaction and visualization layer. The implementation focused on validating system integration and demonstrating the generated eco-driving feedback rather than optimizing usability or interface design.

4.8 Expert Evaluation of Final Prototype

Evaluating NLP systems without access to labelled datasets presents a significant challenge. In the absence of an established ground truth, the quality and appropriateness of generated responses must instead be assessed through human judgment. Creating labelled datasets for conversational systems is both time-consuming and costly, particularly in specialized domains where there may not exist a single objectively correct response. As a scalable alternative, the paradigm of "LLM-as-a-judge" has emerged, where LLMs are used to evaluate the responses generated by other LLMs [76]. However, previous research indicates that in domain-specific settings, subject matter experts are generally better suited to identify factual inaccuracies and subtle quality differences in generated content.

For this project, no labelled dataset for eco-driving feedback existed. Therefore, the final evaluation of the prototype was conducted through expert evaluation. The recruitment criterion was domain expertise within Volvo Group, including experience related to truck operations, eco-driving, and data-driven analysis. A total of five experts participated in the evaluation. This is consistent with prior literature suggesting that even small groups of domain experts (2-3 participants) can provide valuable insights in qualitative expert evaluations [77]. While expert evaluation is

often associated with high financial cost, the primary challenge in this project was instead the limited availability of domain experts and their ability to allocate time for participation. Consequently, the evaluation was conducted in the form of a questionnaire rather than interviews, allowing participants to complete the evaluation at their own convenience.

Before participating, respondents were informed about the purpose of the study and that participation was voluntary. The consent form can be seen in Appendix D. No personally sensitive information was collected. In addition to the evaluation questions, background information regarding participant expertise and familiarity with eco-driving and AI-based tools was collected in order to contextualize the respondent group.

The questionnaire consisted of three main sections. The first section (Part 1) focused on evaluating the overall quality of the generated eco-driving feedback, as well as evaluating the impact of RAG on the generated responses. Participants were presented with ten scenario-based examples derived from telemetry-based driving insights, consisting of five RAG-based and five non-RAG responses, together with the corresponding feedback generated by the AI driver coach. Following established practices from previous literature [78], participants evaluated the responses across several predefined quality dimensions using a five-point Likert scale. The evaluation criteria are presented in Table 4.9.

Table 4.9: Definitions of the evaluation criteria used in the questionnaire.

Criterion	Description
USEFULNESS	The feedback would help improve the driver's fuel efficiency.
RELEVANCE	The feedback focuses on the driver's actual issues in the scenario.
ACTIONABILITY	The feedback provides advice that the driver can realistically act upon.
CLARITY	The feedback is clear and easy to understand.
TONE	The feedback is presented in a trustworthy, approachable, intelligent and positive manner.
SPECIFICITY	The feedback provides specific and concrete guidance rather than general or vague advice.

The second section (Part 2) focused on personalization and contextual adaptation. Participants were presented with pairs of generated responses corresponding to the same driving scenario, where one response incorporated additional contextual or historical driver information while the other did not. The participants were then asked to compare the responses and select which alternative they perceived as more appropriate and personalized for the given driver. This form of pairwise comparison is commonly referred to as A/B testing. The compared responses were presented without revealing which underlying system configuration generated them, as recommended in the literature [78]. The final section consisted of open-ended questions intended to capture qualitative reflections regarding the overall perception of the

conversational AI coach.

The quantitative results from the Likert-scale questions were analyzed using descriptive statistics, including mean values and standard deviations across the evaluated criteria. For the pairwise comparison questions, the number of preferences for each response alternative was summarized. The qualitative responses were analyzed using a lightweight thematic analysis approach, where recurring themes and observations were identified and grouped manually.

Because the questionnaire consisted of structurally similar scenarios that primarily differed in the specific driving behaviors being evaluated, a representative questionnaire template is provided in Appendix D.

5

Results

This chapter presents the results of the developed approach, covering both the data-driven modelling pipeline and the conversational driver coaching prototype. The results are structured in three parts. First, the predictive modelling results are presented. Second, the prototype results illustrate how these insights are translated into generated eco-driving feedback. Finally, the results from the expert evaluation questionnaire are presented.

5.1 Modelling Results

This section presents the results from the predictive modelling of fuel efficiency based on the driving behaviour variables. The primary objective of the modelling was to identify behavioural patterns associated with higher or lower fuel efficiency in the analysed fleet data. First, the predictive performance of the trained models is evaluated and compared against a baseline linear regression model. Subsequently, SHAP-based model interpretation is used to analyse how individual driving behaviour variables influence the model predictions. Together, these results provide the foundation for deriving behavioural insights and personalized eco-driving feedback.

5.1.1 Model Performance

The predictive performance of the models was evaluated using the cross-validation procedure described in Section 4.2.4. Table 5.1 summarizes the mean and standard deviation of R^2 and MAE across the five validation folds for both the LightGBM model and the Ridge Regression baseline.

Table 5.1: Cross-validated predictive performance of the evaluated models.

Segment	Model	Mean R^2	Std	Mean MAE	Std
7XT	Ridge Regression	0.642	0.005	0.128	0.002
7XT	LightGBM	0.796	0.007	0.100	0.001
5XT	Ridge Regression	0.670	0.027	0.136	0.005
5XT	LightGBM	0.744	0.020	0.121	0.005

The LightGBM model consistently achieved higher predictive performance than the Ridge Regression baseline across both analysed segments, both in terms of R^2 and

MAE. This indicates that the relationship between driving behaviour variables and fuel efficiency is not purely linear and that non-linear modelling approaches are well-suited for capturing the underlying patterns in the telemetry data.

The LightGBM models explain a considerable proportion of the variation in fuel efficiency in both analysed segments. In addition, the MAE values indicate that the prediction error is relatively small in the context of the target variable, with average deviations of approximately 0.10 km/L for the 7XT segment and 0.12 km/L for the 5XT segment. This suggests that the models provide reasonably precise predictions of fuel efficiency.

The slightly lower performance observed for the 5XT segment may partly be explained by the smaller dataset size available for this segment, which may limit the model's ability to capture complex behavioural patterns. The relatively small standard deviations across cross-validation folds suggest that the model performance is stable across different validation splits.

To further verify that the observed performance differences were not dependent on a single model initialization, the final LightGBM configuration for each segment, as described in Section 4.2.5, was evaluated across 20 runs using different random seeds. For each run, model performance was assessed using the same grouped cross-validation procedure described in Section 4.2.4. The random seed affects internal stochastic components of the LightGBM training process, such as feature subsampling and bagging.

Across the repeated runs, the variation in the mean R^2 scores was small, indicating that the predictive performance of the LightGBM models is stable with respect to random initialization. A paired t-test was conducted to statistically compare the LightGBM results with those of the ridge regression baseline. The results show that the LightGBM models significantly outperform the ridge regression baseline in both analysed segments (7XT: $p < 10^{-40}$, 5XT: $p < 10^{-30}$).

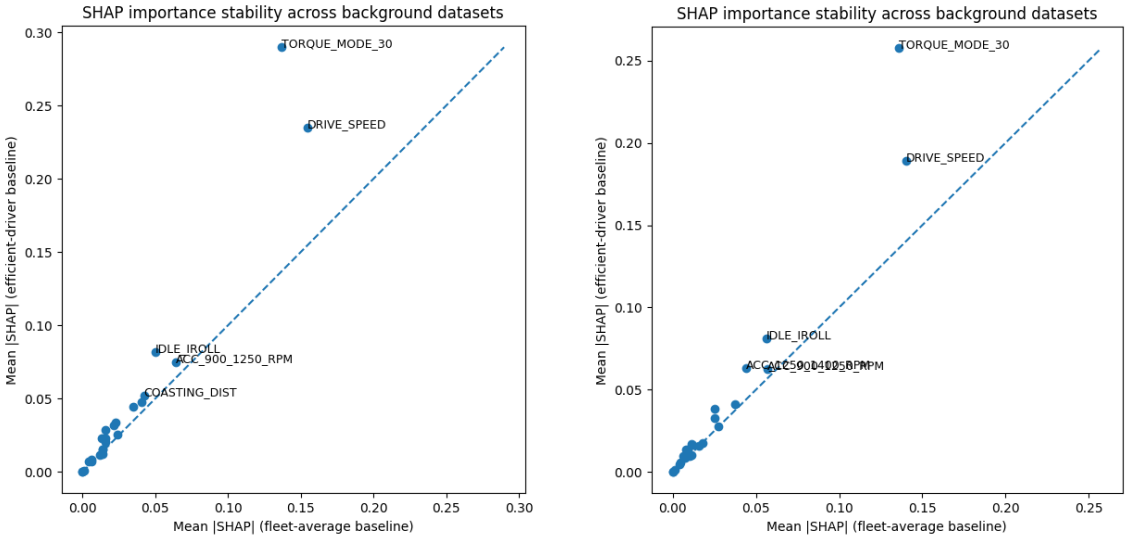
These results indicate that the selected driving behaviour variables contain meaningful information for explaining differences in fuel efficiency across vehicles.

5.1.2 Feature Importance (SHAP)

To understand which driving behaviour variables influence the fuel efficiency predictions, SHAP values were analysed for the trained LightGBM models. SHAP provides feature-level contributions that quantify how each input variable influences the models predicted fuel efficiency relative to the models expected prediction.

As previously discussed, SHAP explanations depend on a background dataset used to estimate the models expected prediction. To assess whether the resulting feature importance rankings depend strongly on this choice, two background datasets were evaluated for each segment: (1) a random sample of observations from the segment and (2) a sample consisting of the top 10% most fuel-efficient observations within the segment.

Figure 5.1 compares the resulting mean absolute SHAP importance values obtained



(a) 7XT segment

(b) 5XT segment

Figure 5.1: Comparison of mean absolute SHAP feature importance obtained using two different background datasets for the (a) 7XT and (b) 5XT segments. Each point represents a feature, and the diagonal line indicates identical importance under both background selections.

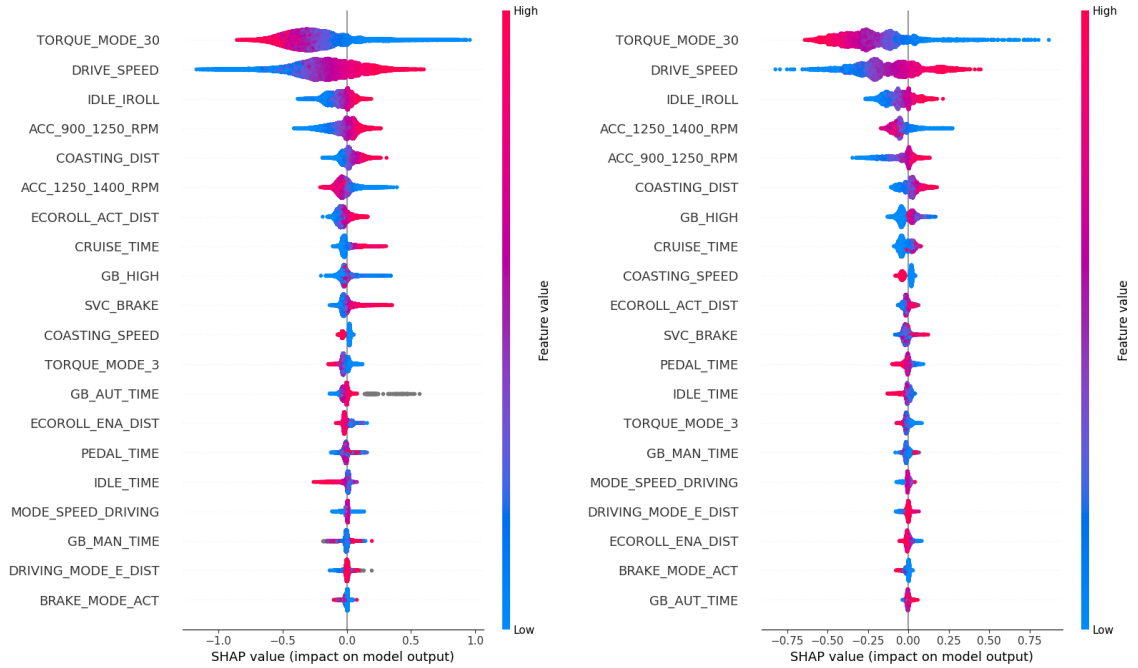
from these two background selections for both the 7XT and 5XT segments. The strong alignment of points along the diagonal indicates that the relative feature importance rankings remain largely consistent across both configurations. However, some features, particularly `TORQUE_MODE_30`, shift in magnitude depending on the selected background distribution. This suggests that the background dataset mainly changes the reference point of the explanation, while the main importance patterns remain relatively stable.

Figure 5.2 shows the SHAP summary plots for the 7XT and 5XT segments. These plots illustrate both the relative importance of features and the direction of their influence on the predicted fuel efficiency. Each point represents a single observation, where the horizontal position indicates the SHAP value and the colour indicates the magnitude of the feature value. Positive SHAP values indicate that the feature increases the predicted fuel efficiency, while negative values indicate a reduction in the prediction.

The resulting feature importance rankings are summarised in Table 5.2, which reports normalized mean absolute SHAP importance values for the most influential features in both segments and under both background datasets.

The SHAP analysis indicates that several driving behaviour variables consistently influence fuel efficiency predictions across both segments. In particular, `TORQUE_MODE_30` and `DRIVE_SPEED` are the two most influential variables in both segments, suggesting that engine load characteristics and vehicle operating speed play a central role in explaining variations in fuel efficiency.

Although the most important variables are broadly similar across segments, the



(a) 7XT segment

(b) 5XT segment

Figure 5.2: SHAP summary plots for the (a) 7XT and (b) 5XT segments using the efficient-driver background dataset.

relative importance of several features differs. For example, `ACC_1250_1400_RPM`, `GB_HIGH`, `IDLE_IROLL`, and `CRUISE_TIME` show higher relative importance in the 5XT segment, while `ECOROLL_ACT_DIST`, `GB_AUT_TIME`, `SVC_BRAKE`, and `COASTING_DIST` are more influential in the 7XT segment.

The repeated-run analysis shows low variability in feature importance estimates across random seeds, as indicated by the small standard deviations in Table 5.2. This suggests that the observed segment-level differences are not primarily driven by model stochasticity. However, the differences should still be interpreted as segment-specific associations between driving behaviour variables and predicted fuel efficiency, rather than direct causal effects.

5.1.3 Local Explanations for Individual Predictions

To illustrate how individual driving behaviours contribute to specific model predictions, SHAP waterfall plots were generated for representative observations from both segments. These examples are used to demonstrate how feature contributions vary depending on both the operational context (segment) and the selected reference baseline.

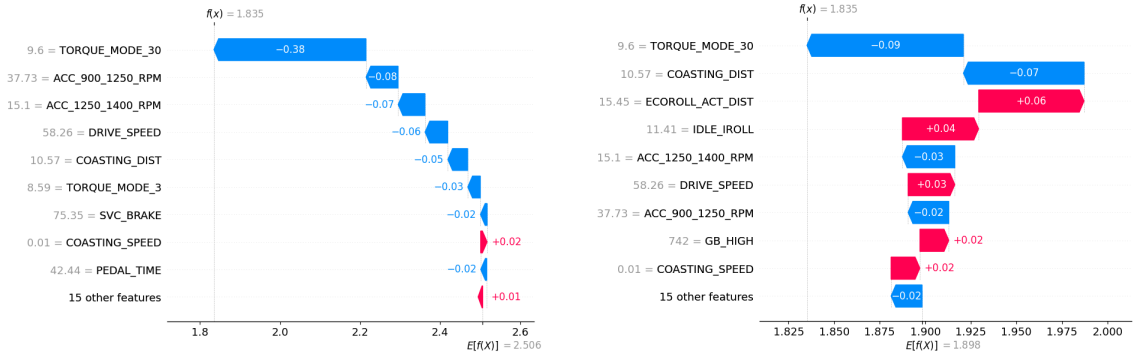
The selected observations were chosen to represent typical driving instances within each segment, rather than extreme outliers, in order to provide interpretable and realistic examples of model behaviour. The waterfall plots decompose the predicted fuel efficiency into additive contributions from individual features relative to the models expected prediction. Each bar therefore represents the contribution of a

Table 5.2: Comparison of normalized SHAP feature importance for the most influential variables under the efficient-driver and fleet-average background datasets. Values represent mean ± standard deviation across 20 model runs.

Feature	5XT Eff.	7XT Eff.	5XT Avg.	7XT Avg.
TORQUE_MODE_30	0.270 ± 0.003	0.266 ± 0.001	0.202 ± 0.002	0.188 ± 0.001
DRIVE_SPEED	0.201 ± 0.002	0.217 ± 0.001	0.208 ± 0.002	0.215 ± 0.001
IDLE_IROLL	0.085 ± 0.001	0.076 ± 0.001	0.082 ± 0.001	0.069 ± 0.001
ACC_900_1250_RPM	0.066 ± 0.002	0.068 ± 0.001	0.085 ± 0.001	0.088 ± 0.001
ACC_1250_1400_RPM	0.068 ± 0.002	0.043 ± 0.001	0.068 ± 0.002	0.056 ± 0.001
COASTING_DIST	0.043 ± 0.001	0.047 ± 0.001	0.054 ± 0.001	0.058 ± 0.001
GB_HIGH	0.042 ± 0.001	0.029 ± <0.001	0.040 ± 0.001	0.030 ± <0.001
CRUISE_TIME	0.036 ± 0.001	0.031 ± <0.001	0.039 ± 0.001	0.031 ± 0.001
ECOROLL_ACT_DIST	0.021 ± 0.001	0.040 ± 0.001	0.028 ± 0.002	0.048 ± 0.001
GB_AUT_TIME	0.007 ± 0.001	0.020 ± 0.001	0.007 ± 0.001	0.021 ± 0.001

specific feature to the final prediction for that observation.

The figures also allow comparison between the two background datasets used for SHAP estimation. Figure 5.3 shows the results for an example observation from the 7XT segment, while Figure 5.4 presents the an example from the 5XT segment.



(a) Fuel-efficient background

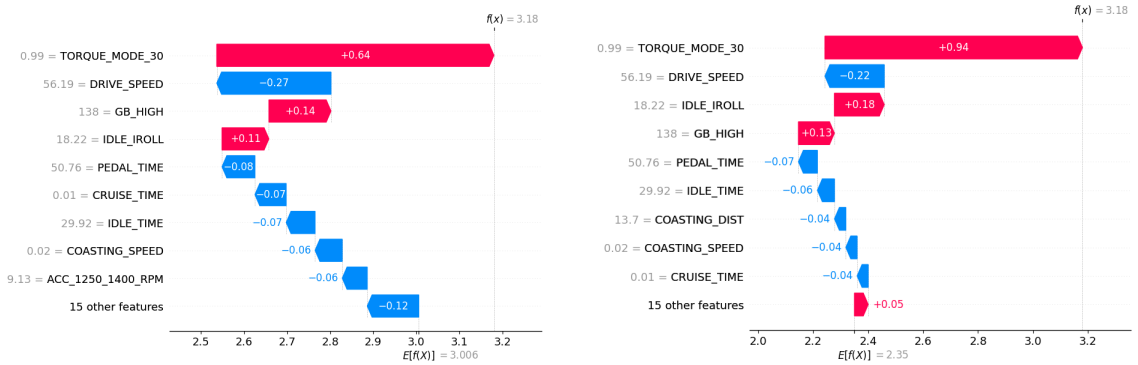
(b) Random background

Figure 5.3: SHAP waterfall plots for a representative observation in the 7XT segment using (a) fuel-efficient and (b) random observations as background data.

For the 7XT example, the predicted fuel efficiency of the selected observation is lower when compared to both reference baselines, although the difference is substantially larger when using the baseline derived from the most fuel-efficient drivers. When using the efficient-driver baseline, several features contribute negatively to the prediction, most notably `TORQUE_MODE_30`, followed by variables related to engine load and acceleration behaviour such as `ACC_900_1250_RPM` and `ACC_1250_1400_RPM`. Additional negative contributions are observed for `DRIVE_SPEED` and `COASTING_DIST`, indicating that the driving behaviour for this observation differs from the patterns observed among the most fuel-efficient drivers.

When using the random background dataset, the relative contributions of some variables change. In this configuration, features such as `ECOROLL_ACT_DIST` and

IDLE_IROLL contribute positively to the prediction, indicating that the driver's behaviour performs relatively well compared to the fleet average for these aspects. However, variables related to engine torque utilisation and acceleration behaviour still contribute negatively to the prediction. This demonstrates that the choice of background dataset primarily affects how deviations from a reference behaviour are interpreted, rather than the underlying direction of the feature contributions.



(a) Fuel-efficient background

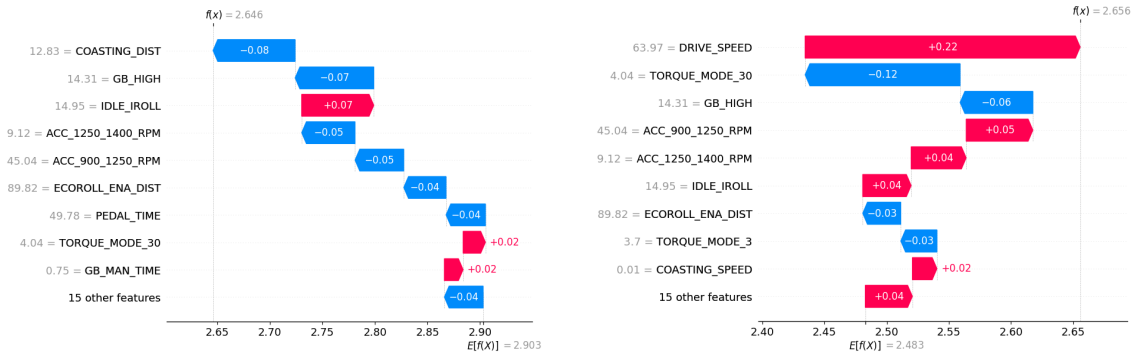
(b) Random background

Figure 5.4: SHAP waterfall plots for a representative observation in the 5XT segment using (a) fuel-efficient and (b) random observations as background data.

For the 5XT observation, the overall prediction remains relatively similar under both background configurations, although the magnitude of individual feature contributions varies. In this case, the predicted fuel efficiency is higher than the baseline derived from the most fuel-efficient drivers, indicating that the selected observation represents a highly efficient driving instance.

In this example, TORQUE_MODE_30 contributes strongly and positively to the predicted fuel efficiency, indicating favourable engine load behaviour. In contrast, features such as DRIVE_SPEED and PEDAL_TIME contribute negatively to the prediction, suggesting that improvements in speed management and accelerator usage could potentially increase fuel efficiency further.

To further investigate segment-specific effects, a randomly selected observation from the 5XT segment was analysed using both the 5XT and 7XT models. This comparison revealed differences in how identical driving behaviour is evaluated depending on the operational context. The resulting SHAP explanations using the fuel-efficient baseline are shown in Figure 5.5.



(a) 5XT model

(b) 7XT model

Figure 5.5: Comparison of SHAP explanations for the same observation when using segment-specific models.

The results of this example show that the same driving behaviour is interpreted differently depending on the model used. While some features contribute in a similar direction, the magnitude and relative importance of contributions differ between the two models.

To assess whether these differences extend beyond individual examples, the cross-segment comparison described in Section 4.3.4 was applied to held-out observations from both segments. The resulting similarity metrics are presented in Tables 5.3 and 5.4.

Table 5.3: Comparison of segment-specific SHAP explanations across held-out observations using the fuel-efficient background dataset. Values are reported as mean ± standard deviation unless otherwise stated.

Metric	**5XT observations**	**7XT observations**
Prediction difference	0.320 ± 0.172	0.411 ± 0.160
Cosine similarity	0.770 ± 0.206	0.862 ± 0.142
Sign agreement	0.759 ± 0.093	0.713 ± 0.078
Same strongest positive contributor	0.375	0.556
Same strongest negative contributor	0.656	0.806

The results presented in Table 5.3 indicate that the segment-specific models frequently produce different predictions and SHAP explanations for the same driving observations. The average prediction differences between the models are apparent for both observation sets, suggesting that the estimated fuel efficiency depends not only on the observed driving behaviour itself but also on the operational context represented by the segment-specific models.

At the same time, the cosine similarity and sign agreement metrics indicate that the overall explanation patterns remain partially consistent across models. In many cases, the models identify similar general behavioural trends, even if the magnitude of the feature contributions differs. This suggests that certain behavioural effects

generalise across segments, while the relative importance of these effects remains context-dependent.

The agreement analysis of the strongest contributing features further highlights this behaviour. The models more frequently agree on the strongest negative contributor than on the strongest positive contributor, particularly for the 7XT observations. This may indicate that inefficient driving patterns are more consistently identified across operational contexts, whereas efficient driving behaviour appears to be more segment-specific.

Table 5.4: Feature-level comparison of SHAP contributions between the 5XT and 7XT models across held-out observations using the fuel-efficient background dataset. Reported p-values correspond to paired t-tests with false discovery rate correction.

	5XT observations				7XT observations			
Feature	**5XT model**	**7XT model**	**Abs. diff**	p_{adj}	**5XT model**	**7XT model**	**Abs. diff**	p_{adj}
TORQUE_MODE_30	-0.110	-0.259	0.166	5.16×10^{-10}	-0.169	-0.393	0.224	3.23×10^{-23}
DRIVE_SPEED	-0.058	0.023	0.097	1.93×10^{-6}	-0.000	0.095	0.105	1.65×10^{-10}
IDLE_IROLL	0.013	-0.006	0.020	4.29×10^{-7}	-0.012	-0.033	0.023	8.28×10^{-6}
ACC_900_1250_RPM	-0.105	-0.025	0.083	4.56×10^{-13}	-0.083	-0.002	0.081	5.30×10^{-17}
ACC_1250_1400_RPM	-0.046	0.016	0.062	6.47×10^{-17}	-0.054	-0.011	0.045	5.44×10^{-10}
COASTING_DIST	-0.038	-0.022	0.040	0.049	-0.025	-0.008	0.026	2.80×10^{-4}
GB_HIGH	-0.031	-0.018	0.014	2.81×10^{-3}	-0.035	-0.031	0.015	0.322
CRUISE_TIME	-0.009	-0.003	0.018	0.159	0.016	0.039	0.027	2.11×10^{-6}
ECOROLL_ACT_DIST	-0.004	-0.022	0.028	5.03×10^{-3}	-0.007	-0.036	0.034	8.35×10^{-8}
GB_AUT_TIME	-0.002	-0.011	0.010	4.50×10^{-5}	-0.002	-0.017	0.017	5.36×10^{-6}

The feature-level comparison presented in Table 5.4 further demonstrates that several important driving behaviour variables are interpreted differently by the two segment-specific models. Features such as `TORQUE_MODE_30`, `DRIVE_SPEED`, and `ACC_900_1250_RPM` exhibit statistically significant differences in SHAP contribution across both observation sets.

These results indicate that the relationship between driving behaviour variables and fuel efficiency is context-dependent. While the models share several common behavioural patterns, the differences observed in both prediction outputs and SHAP explanations indicate that segment-specific modelling provides additional contextual information that would not be captured by a single global model.

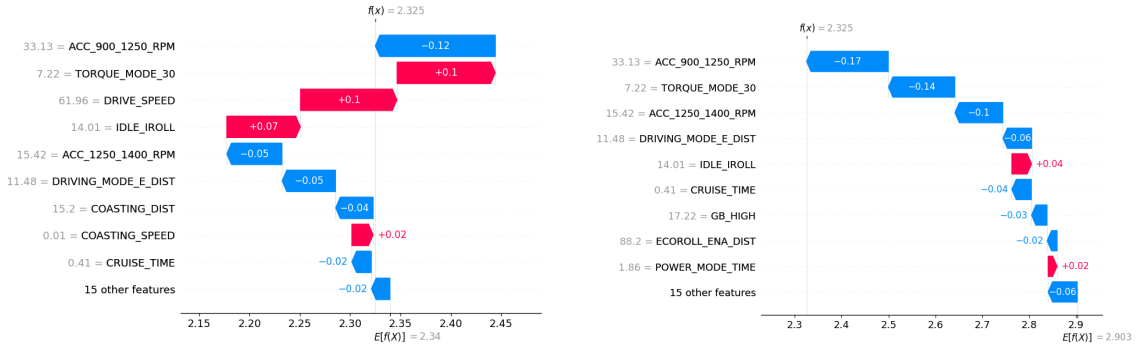
5.2 Prototype Results

This section presents the results from the prototype application. First, we provide an illustrative example of the system's pipeline in action. This includes presenting the SHAP prediction waterfall plots used to identify key driving features, followed by a direct comparison of the generated coaching feedback both without and with the utilization of RAG. Finally, we compare the coaching feedback without and with the utilization of a memory layer.

5.2.1 Feedback Generation Comparison

Provided below are the predicted results of a single truck from the 5XT segment when compared against the random average (Figure 5.6a) and efficient drivers (Fig-

ure 5.6b). Because the truck performs below average (2.325 km/L vs. an average of 2.34 km/L), it will be compared against the random average. Looking more closely at Figure 5.6a, we can see that the driver's most positive aspect is their utilization of `TORQUE_MODE_30`, or Maximum Torque Time. The driver's most negative feature, however, is `ACC_900_1250_RPM`. These are the two primary features the coach will highlight.



(a) Average random 5XT model

(b) Efficient 5XT model

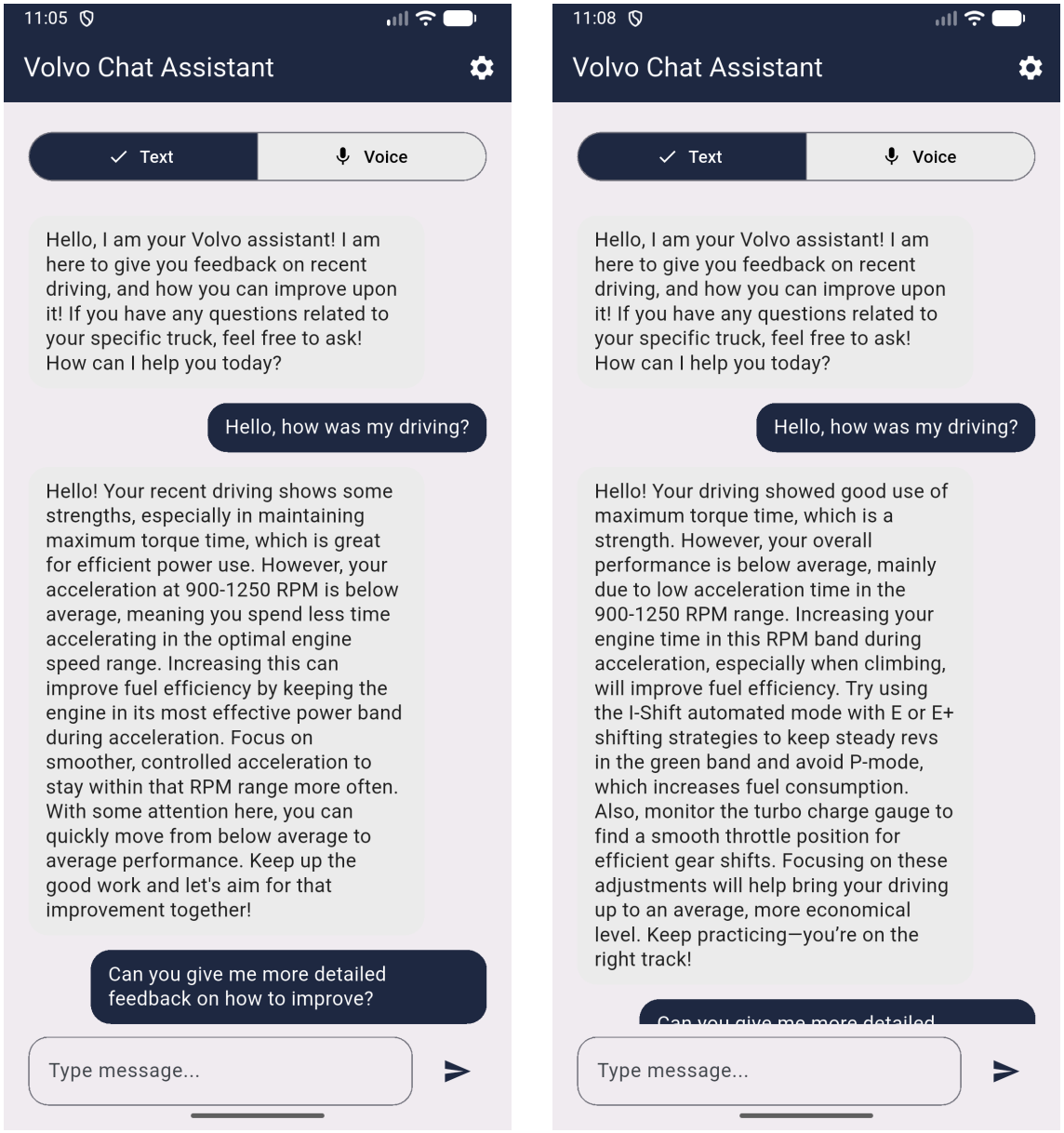
Figure 5.6: Comparison of SHAP explanations for the same truck observation when comparing against (a) random average and (b) efficient baseline

To compare the effect of the retrieval mechanism, the generated feedback for this specific scenario was evaluated under two conditions. The baseline response, generated without utilizing RAG, can be seen in Figure 5.7a. The corresponding response generated utilizing the full RAG pipeline is presented in Figure 5.7b. Both generated responses identify and highlight the positive and negative features derived from the SHAP analysis, praising the maximum torque time (`TORQUE_MODE_30`) while identifying the specific acceleration range as an area for improvement. However, a comparison of the output shows qualitative differences in how feedback is delivered.

When generating a response without the utilization of RAG (Figure 5.7a), the model relies on its pre-trained internal knowledge base. The resulting feedback is theoretically correct regarding general eco-driving principles. For example, it recommends *"smoother, controlled acceleration"* to maintain the optimal engine speed. However, the advice remains broad. Since the model lacks access to domain-specific documentation, the instructions are generic and lack concrete operational steps. The feedback is factually sound from a general driving standpoint, but it does not instruct the user on how to utilize the vehicle's specific hardware to achieve the desired outcome.

Comparatively, the response generated utilizing the RAG pipeline (Figure 5.7b) shows a integration of domain-specific context. While addressing the same underlying SHAP features, the RAG model translates the general need for improved acceleration into hardware-specific instructions. It advises the driver to utilize the *"I-Shift automated mode with E or E+ shifting strategies,"* warns against the use of the fuel-intensive "P-mode," and suggests monitoring the "turbo charge gauge" to optimize gear shifts.

Overall, the primary difference between the two outputs is the transition from generic advice to technical instruction. The addition of RAG enables the system to ground its feedback in the actual operational context of the truck, utilizing specific Volvo terminology to guide the driver's direct interaction with the vehicle's systems.



(a) Feedback output when not utilizing RAG

(b) Feedback output when utilizing RAG

Figure 5.7: Comparison of feedback when not utilizing and utilizing RAG

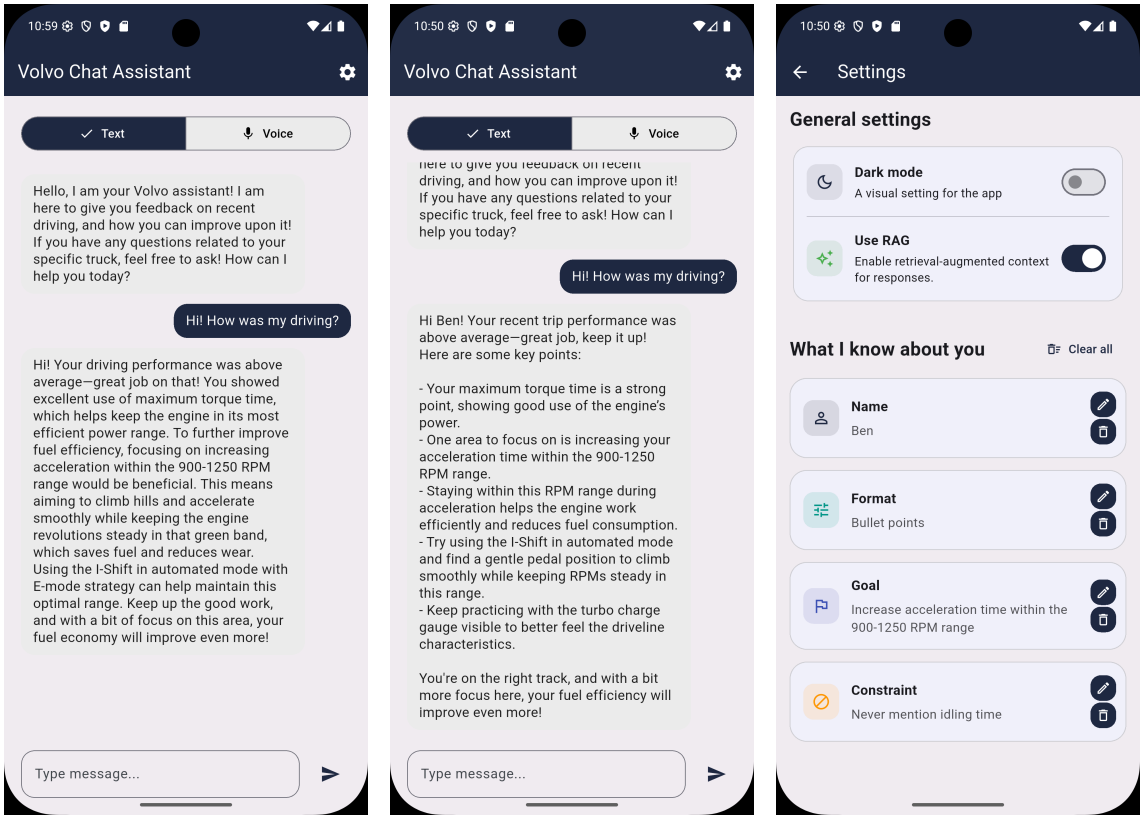
5.2.2 Memory Layer Demonstration

This section demonstrates how the persistent semantic memory layer is exposed through the interface and used during feedback generation. In addition to generating responses from telemetry-derived SHAP explanations and retrieved domain knowledge, the prototype allows stored user-specific information to influence the

generated feedback.

Figure 5.8 shows the same driving scenario under three interface states. Figure 5.8a shows the generated response without access to the memory layer, Figure 5.8b shows the corresponding response when stored memory items are included in the prompt, and Figure 5.8c shows the settings view where these memory items are presented to the user.

Comparing the two generated responses illustrates how these memory items affect the output. Without memory, the assistant provides relevant feedback based on the same driving scenario, but the response is generic in presentation. This example demonstrates how the memory layer contributes to personalization in the prototype. The stored information does not replace the telemetry-based coaching context or the domain knowledge retrieved through RAG; instead, it modifies how the generated feedback is framed and delivered. In this way, the prototype supports persistent user modelling while also giving the driver visibility and control over the information used for personalization.



(a) Response without memory. (b) Response with memory. (c) Stored memory items in the settings interface.

Figure 5.8: Prototype demonstration of the semantic memory layer. The left image shows a response generated without memory, the middle image shows the corresponding response with semantic memory, and the right image shows the stored memory items available in the settings interface.

5.3 Results from Expert Evaluation Questionnaire

This section presents the results from the expert evaluated questionnaire described in Section 4.8, which constituted the primary evaluation method for the developed artefact. The results include both quantitative assessments and qualitative feedback regarding the generated eco-driving responses, personalization and the overall perception of the conversational AI coach. A summary of the participant backgrounds and expertise is presented in Table 5.5.

Table 5.5: Summary of participant background information.

Characteristic	Summary
Number of participants	5 for Part 1, of which 4 also answered Part 2
Professional backgrounds	Data Engineering, Data Analytics and Application Engineering, Business Development, UX Strategy, and End User Function Ownership
Professional experience	Majority of participants had more than 6 years of professional experience
Eco-driving familiarity	Participants reported moderate to very high familiarity with eco-driving and fuel-efficient driving
Experience with eco-driving	All participants reported between 3–9 years of experience related to eco-driving
Formal eco-driving education	3 out of 5 participants had received formal eco-driving education or training
AI tool usage	All participants reported using AI-based tools such as ChatGPT, Claude, Copilot, or Gemini on a weekly or daily basis
General attitude toward AI	Participants mostly expressed positive attitudes toward AI and reported being comfortable using AI tools in their work

5.3.1 Results from Part 1

The first section of the questionnaire evaluated the impact of the RAG pipeline on the quality of the generated coaching feedback. Respondents were presented with 10 driving scenarios (five generated with the RAG pipeline and five without) and asked to evaluate the system's responses based on the six criteria defined previously in Table 4.9. Evaluations were conducted using a 5-point Likert scale, ranging from 1 ("Strongly Disagree") to 5 ("Strongly Agree"). The aggregated mean scores for these evaluations are presented in Table 5.6.

The baseline LLM (Without RAG) yielded slightly higher overall scores among the experts. As seen in Table 5.6, the non-RAG approach outperformed the RAG pipeline in three of the six metrics (Usability, Tone, and Specificity), resulting in a higher overall average (3.78 vs. 3.72). The RAG pipeline showed slight improvements only in Actionability and Relevance, while Clarity remained tied between the two approaches.

Table 5.6: Updated expert evaluation scores comparing generated feedback with and without utilizing RAG.

Metric	Without RAG	With RAG
Usability	3.65 ±0.57	3.48 ±0.47
Actionability	3.58 ±0.51	3.68 ±0.36
Tone	4.21 ±0.1	4.08 ±0.28
Clarity	3.66 ±0.28	3.66 ±0.26
Relevance	3.66 ±0.58	3.69 ±0.52
Specificity	3.94 ±0.39	3.74 ±0.32
Average	**3.78** ±0.18	**3.72** ±0.22

Analysing the free-form text responses provided context for these quantitative results. Initial trends indicated that while the RAG pipeline successfully generated detailed, hardware-specific advice, it often suffered from information overload. This caused the feedback to be perceived as overly dense and confusing when compared to the baseline LLM. Furthermore, respondents noted that the RAG-generated feedback occasionally contained contradictory instructions, an issue that would likely confuse a driver in a real-world scenario. Most notably, there were instances where the information retrieved and presented by the RAG system was factually inaccurate.

Finally, the qualitative feedback for Part 1 highlighted a disconnect regarding the presentation of the feature to improve upon. Respondents were unsure of what the specific features actually meant. Consequently, when the RAG system provided highly specific, technical advice based on these poorly understood features, respondents often perceived the feedback as unclear or irrelevant to actual driving behaviour.

5.3.2 Results from Part 2

The second part of the questionnaire evaluated the effect of contextual memory and personalization on the perceived quality of the generated responses. Participants compared pairs of responses corresponding to the same driving scenario, where one response incorporated contextual or historical driver information while the other did not. The participants then selected which response they preferred overall, as well as which response they perceived as more tailored to the driver.

The results from the A/B evaluation are summarized in Table 5.7. Overall, the results indicate that incorporating contextual memory and personalization generally improved the perceived quality and contextual adaptation of the generated feedback. In five out of six scenarios, the personalized response was preferred overall by the majority of participants.

However, the impact of personalization varied between scenarios. Several scenarios resulted in "No significant difference" responses regarding perceived personalization, suggesting that some contextual adaptations were either too subtle to substantially affect the perceived response quality, or that the baseline responses were already

Table 5.7: Summary of A/B evaluation results for personalization scenarios. Scenarios 1-3 tested previous coaching memory, while scenarios 4-6 tested the semantic memory layer.

Scenario	Personalized response	Preferred overall	Perceived as more tailored	No significant difference
1	B	4/4	3/4	1/4
2	A	3/4	2/4	2/4
3	A	3/4	2/4	1/4
4	A	3/4	2/4	1/4
5	B	1/4	1/4	1/4
6	B	4/4	4/4	0

sufficiently contextualized without the additional memory information.

One notable exception was Scenario 5, where the personalized response was only preferred by one participant. A respondent specifically commented that the personalized response contained technically questionable advice regarding low-rev acceleration behaviour (Quote: *"Option B says to avoid low-rev acceleration. I consider low-revs as 900-1250rpm, and this is exactly what he is doing right."*). This suggests that personalization alone does not necessarily improve perceived response quality if the incorporated contextual information introduces ambiguity or conflicts with domain-specific expectations.

Taken together, the results suggest that contextual memory and personalization mechanisms can improve the perceived appropriateness and contextual adaptation of conversational eco-driving feedback. At the same time, the findings indicate that contextual personalization does not automatically improve perceived response quality, but rather depends on whether the additional context is perceived as relevant and technically sound.

5.3.3 Qualitative Findings

In addition to the quantitative evaluation, respondents were given the opportunity to provide qualitative feedback in free text throughout the questionnaire. This included both optional comments after each presented scenario, as well as more general reflection questions at the end of the survey. This section summarizes the main recurring themes identified from the qualitative responses.

A recurring theme across the qualitative responses was the importance of concise and more easily scannable feedback. Multiple participants noted that the generated responses were often overly verbose and contained too much explanatory text. Several respondents suggested using bullet points, shorter summaries or clearer separation of information in order to improve the readability.

Another recurring theme concerned the importance of technically grounded and context-aware coaching advice. Several respondents highlighted that certain driving behaviours, such as coasting, I-Roll usage, cruise control and average speed, are

highly dependent on operational context and may not always be universally beneficial or harmful. Participants emphasized that overly simplified recommendations could become misleading if important contextual factors, such as road topology, legal speed limitations, traffic conditions or vehicle configuration are not considered.

Several participants also expressed concerns regarding the use of technical terminology and overly detailed explanations in the generated feedback. Concepts such as "I-Roll" and "green band directions" were described as potentially confusing for drivers unfamiliar with the underlying vehicle systems. Multiple respondents suggested that more general and intuitive explanations would improve comprehensibility and practical usefulness. For example, one respondent said *"Mixing I-Roll, Coasting and Momentum concepts seems to introduce more complexity than real improvement.."*.

Despite the identified limitations, the overall perception of the prototype was positive. Several respondents described the concept as promising and highlighted its potential practical usefulness in real-world eco-driving applications. One participant described the system as *"an excellent step towards using AI for driving tips.."* while another stated that they *"believe there is a real-world application to what you are developing."*.

6

Discussion

This chapter discusses the findings presented in the previous chapter in relation to the research goals of the thesis by revisiting the research questions and reflecting on how the results contribute to answering them. The chapter then considers broader ethical and environmental implications of using AI-based driver coaching in a heavy-duty transport context. Finally, potential directions for future work are outlined.

6.1 Research goals

In this section, the results are brought together and discussed in relation to the research questions. Each question is revisited and answered based on the findings presented in the previous chapter. Table 6.1 provides a quick overview of the findings.

Research Question	**Findings**
RQ1	Fuel efficiency is influenced by a core set of driving behaviour variables across both analysed segments, particularly `TORQUE_MODE_30` and `DRIVE_SPEED`, while the relative importance and interpretation of some additional variables vary moderately depending on operational context.
RQ2	Context-aware eco-driving feedback can be derived by translating local SHAP explanations into driver-facing advice, using segment-specific models, reference baselines, and user-specific conversational context to adapt the feedback.
RQ3	A framework combining telemetry-derived explanations, an LLM, conversational history, and a semantic memory layer can enable conversational and partially personalized eco-driving feedback, although the current RAG implementation did not consistently improve factual grounding.

Table 6.1: Summary of the research questions and their corresponding findings.

6.1.1 RQ1

Which driving behaviour variables most strongly influence fuel efficiency in different operational segments of heavy-duty truck fleet data?

The results show that several driving behaviour variables consistently exhibit a strong influence on fuel efficiency across both analysed segments. In particular, the variables `TORQUE_MODE_30` and `DRIVE_SPEED` are among the most influential features in both the 5XT and 7XT segments. This indicates that how the engine is utilised and the speed at which the vehicle is operated play a central role in determining fuel efficiency across both operational, which is to be expected.

In addition to these common drivers, the analysis reveals segment-specific differences in the relative importance of other variables. In the 5XT segment, features related to engine speed intervals and gearbox usage, such as `ACC_1250_1400_RPM` and `GB_HIGH`, as well as variables related to idle and cruising behaviour, show higher importance. In contrast, the 7XT segment places greater emphasis on features associated with coasting and automated driving behaviour, including `ECOROLL_ACT_DIST`, `GB_AUT_TIME`, and `SVC_BRAKE`.

These differences are consistent across both background datasets used for SHAP estimation and across repeated model runs, indicating that they are not primarily driven by model stochasticity or the choice of reference distribution. Furthermore, local explanation analysis demonstrates that identical driving behaviour can be evaluated differently depending on the segment-specific model used, reinforcing the conclusion that the relationship between driving behaviour and fuel efficiency is context-dependent.

However, these differences should be interpreted with some caution. Since the two analysed segments were not based on the same engine configuration, the observed differences between the 5XT and 7XT models may reflect differences in vehicle configuration, rather than operational segment effects alone. This means that the results should be interpreted as segment-specific associations in the analysed data, rather than as isolated effects of the segment definitions themselves.

In addition, the magnitude of the observed segment-specific differences is generally moderate rather than substantial. Many of the most influential variables remain highly important across both segments, and the overall feature importance rankings are broadly similar. One possible explanation for this is that the analysed segments are relatively broad and may not fully capture finer-grained operational differences between vehicles. As a result, some context-specific behavioural patterns may be averaged out within each segment, limiting the extent to which more distinct segment-specific effects can be observed.

The findings suggest that while a core set of driving behaviours influences fuel efficiency across all segments, the relative importance and interpretation of these behaviours vary to some extent depending on the operational context. This supports the use of segment-specific models to capture contextual differences, while also indicating that further refinement of segment definitions or the inclusion of additional

contextual variables could reveal more pronounced differences in how driving behaviour influences fuel efficiency.

6.1.2 RQ2

What context-aware eco-driving feedback can be derived from driving behaviour insights related to fuel efficiency?

The results show that telemetry-based driving behaviour insights can be translated into eco-driving feedback by identifying which variables contribute positively or negatively to predicted fuel efficiency for a specific observation. Local SHAP explanations provide the basis for this translation, as they indicate both which driving behaviour variables are most influential and whether each variable increases or decreases the predicted fuel efficiency relative to a selected reference baseline. This enables feedback to be generated from the driver's observed behaviour rather than from generic eco-driving principles alone.

The context-aware aspect of the feedback is supported in several ways. First, the use of segment-specific models allows feedback to be adapted to the operational context represented by each segment. The cross-segment explanation comparison shown in Section 5.1.3 showed that the same observations can receive different predictions and SHAP attributions depending on the segment-specific model used. This indicates that the interpretation of a driving behaviour is not entirely global, but depends partly on the operational context.

In addition, the use of different SHAP background datasets enables different feedback framings. The fleet-average background allows the driver to be compared against typical behaviour within the segment, which is useful for identifying behaviours that are better or worse than the average. The fuel-efficient background provides a more aspirational reference point by comparing the driver to high-performing observations.

Furthermore, the prototype supports additional personalisation through conversational context and the memory layer. This makes it possible to adapt feedback not only to the current telemetry-based explanation, but also to previously stated driver preferences, constraints or earlier coaching interactions. For example, if a driver has previously expressed a preference for receiving feedback in bullet-point format, this information can be used to adjust how future feedback is structured. In this sense, context-aware feedback is not limited to the operational segment, but also includes user-specific and conversational context.

However, the results also indicate that the level of context-awareness achieved through the current segmentation is limited. The differences between the 5XT and 7XT segments were present but generally moderate, and many of the most influential variables remained important across both segments. One possible explanation is that the analysed segments capture broad operational categories rather than more detailed driving contexts. As a result, variations related to for example route profile, vehicle load or driving environment may be partly averaged out within each segment,

reducing the specificity of the feedback that can be derived from segment-level explanations.

A possible extension would therefore be to construct SHAP explanations using more fine-grained reference groups, such as cluster-combination-specific background datasets, rather than relying only on segment-level explainers. This would allow the same trained model to generate explanations relative to a more specific operational context, potentially making the resulting feedback more context-sensitive without requiring separate LightGBM models for every small cluster. Such an approach could help bridge the gap between broad segment-level modelling and more specific driver coaching.

6.1.3 RQ3

What framework can enable the delivery of factually grounded and personalized eco-driving feedback in a conversational AI prototype?

The results from the questionnaire indicate that a framework combining telemetry-derived driving insights, an LLM and conversational memory mechanisms can enable the delivery of conversational and partially personalized eco-driving feedback in a conversational AI prototype. The generated responses were generally perceived as understandable and relevant to a certain extent by the participating experts. In particular, the incorporation of previous coaching history was often perceived positively and tended to improve the contextual adaptation of the generated feedback.

The semantic memory layer produced more mixed results. While some personalized responses were perceived as more tailored and context-aware, other scenarios showed little perceived difference, and in one case the personalized response was clearly disfavoured due to technically questionable advice. This suggests that personalization alone is insufficient unless the incorporated contextual information remains relevant and technically grounded.

Regarding factual grounding, the current RAG implementation did not consistently improve the generated feedback. Although the intention was to provide more detailed and technically precise coaching advice, the experts generally preferred the responses generated without RAG. The retrieved context occasionally introduced overly specific or confusing recommendations, which negatively affected the perceived quality of the responses.

From the start of the project, our hypothesis was that with the help of RAG, we would be able to retrieve more specific and detailed driving tips, which would make it easier for drivers to improve. Instead, the experts ultimately preferred the feedback generated solely by the LLM. We believe, upon reviewing the outputs, that when relying on the LLM, the system converses in broad terms. That is, it rarely goes into specific details, but as a result, it also avoids digging too deep and responding factually incorrectly. In contrast, the RAG implementation became overly specific and detailed, which made it confusing and lowered its scores. Therefore, based on the current implementation of the artefact, relying solely on the LLM proved to be the more reliable choice.

We believe this contrast with our hypothesis is due to the fact that the manuals used for the RAG database were rather irrelevant to our specific context. Relying on a vehicle manual from 2012 (which was mostly detailed regarding vehicle infotainment) and two driver manuals that are at least four years old and not necessarily directed at our target audience meant the generated feedback occasionally became disjointed. The limited size of the retrieved chunks also meant that some context was lost in how the model formulated its answers. For example, when a driver needed to improve their acceleration, the system often grounded its advice in acceleration techniques for driving on hills, which was a different context from what some of the evaluated trucks were actually experiencing. It essentially provided information for a scenario that was not applicable to the specific case.

Finally, we underestimated the importance of the interface. We repeatedly received feedback that the responses felt like a *"wall of text"* that was hard to interpret, which likely lowered the prototype's clarity and overall scores. This indicates that the system prompts regarding the structure of the generated text must be adjusted. For example, almost all respondents preferred bullet points or some form of general overview.

6.2 Ethical and Environmental Considerations

There are several ethical and environmental considerations that must be addressed to understand the broader impact if an actual implementation were to be deployed to the public. An ethical and safety aspect is the unreliability of LLMs in terms of the risk of hallucinations, which is discussed in Section 3.4.2. While we actively tried to limit this risk in this project, both through instructional constraints in the prompt and with the addition of RAG, it is difficult to guarantee the elimination of inaccurate output. In this particular context, hallucinations could lead to misleading or dangerous coaching advice. If the system misinterprets vehicle telemetry, it could put the driver at risk and reduce trust in the system. Rigorous testing of the RAG retrieval and relevant sources must be conducted to achieve reliable results.

In addition to the ethical implications of a potential deployment, ethical considerations were also relevant during the execution of this study. The vehicle telemetry data used in the project was provided by Volvo Group and remains the property of the company. Access to the data was granted solely for the purpose of conducting this thesis and was limited to the authors. The data was stored and processed using company-approved infrastructure and did not contain personally identifiable information about individual drivers.

Furthermore, the questionnaire participants were informed about the purpose of the study, how their responses would be used, and that participation was voluntary. Informed consent was obtained from all participants prior to participation. A copy of the consent form is provided in Appendix D.

In a real-world deployment, the continuous monitoring of a driver would also raise privacy and surveillance concerns. While commercial truck drivers are often already subject to logistical tracking by their fleet operators, introducing an application that

continuously evaluates and critiques driving performance creates a new dimension of surveillance. Intrusive monitoring could lead to increased stress and a deterioration of workplace trust. Furthermore, the conversational nature of the application introduces data security risks. Drivers interacting with the LLM coach might disclose sensitive personal or professional information within the chat. If a data breach were to occur either within the application's own infrastructure or through vulnerabilities in the underlying language model, this confidential data could be exposed.

Finally, if an application like this were to be deployed, it would introduce a potential environmental paradox. The overarching goal of the coaching system is to instruct drivers on eco-friendly techniques to minimize fuel consumption and, ultimately, reduce emissions. However, the LLM powering the chatbot consumes substantial amounts of electricity, possibly from non-renewable sources, and requires significant water for server cooling. This raises questions regarding the system's net environmental impact: if the emissions generated by operating the application exceed or are on par with the emissions saved through improved driving behaviour, the system would effectively worsen overall environmental outcomes. To ensure a net-positive impact, one must consider model selection, prioritizing smaller and more efficient models over larger ones. Furthermore, it is essential to verify that the hardware is hosted in data centres powered by renewable energy and situated in regions where water consumption does not strain local ecosystems or populations.

6.3 Future Work

There are several aspects that can be changed or improved upon if one wants to continue with this work. Overall, we believe that for an actual implementation, the underlying data, external knowledge base and features must be changed. First and foremost, the data must be changed from aggregated truck data to specific driver data. Since the utilized data is from an individual truck, which may be operated by multiple people, and not an individual driver, the provided feedback in its current form will likely be diluted. Secondly, the manuals for the RAG knowledge base must be changed to newer and more relevant ones. Thoroughly reviewed manuals must be chosen, so engineers are comfortable that the retrieved excerpts are relevant and correct. An increased size of chunks could be implemented, which might improve the context of the retrieved feedback. Thirdly, the associated features have to be refined and evaluated by domain experts. A clear understanding of what a driver can actually influence during their driving has to be established. Finally, relationships between the selected features should be modelled more explicitly. Discussions with domain experts indicated that some variables are closely connected, meaning that changes in one variable may influence another. Future implementations should therefore include constraints or rules that account for these dependencies, so that the generated feedback does not treat related variables as independent coaching targets.

Apart from the underlying data, there are several other aspects that can improve the work in the future. An interesting aspect is to further explore the utilization of even narrower or more specific segments. As previously mentioned, the segments

are generally quite broad in terms of their cluster types, where for example they encompass a wide range of vehicle speeds and slope profiles. By comparing drivers against a highly specific clustering combination, rather than a broad segment, one could likely retrieve even more contextual and relevant information. Initial testing at the end of the project gave promising results, but was nothing that could be fully investigated. To further improve specificity, one could include further context, such as customer context. One could specify more precisely what kind of environments or type of haulage a specific driver or customer operates in.

Finally, an alternative approach for future research is to shift the target audience of the application from the individual truck driver to the fleet owner. While fleet owners are not always the ones physically driving, they bear the financial responsibility for the vehicles, maintenance and fuel. Consequently, they often possess the strongest interest in reducing operational costs. Redirecting the system's feedback toward fleet managers would equip them with the insights needed to monitor individual drivers and deliver targeted coaching. This approach could facilitate more structured and systematic follow-up routines, which may prove more effective than placing the entire burden of performance improvement solely on the individual driver.

Bibliography

[1] K. Ayyildiz, F. Cavallaro, S. Nocera, and R. Willenbrock, "Reducing fuel consumption and carbon emissions through eco-drive training," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 46, pp. 96–110, Apr. 2017, ISSN: 1369-8478. DOI: 10.1016/j.trf.2017.01.006. Accessed: Feb. 12, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1369847816302212.

[2] N. Xu, X. Li, Q. Liu, and D. Zhao, "An Overview of Eco-Driving Theory, Capability Evaluation, and Training Applications," en, *Sensors*, vol. 21, no. 19, p. 6547, Jan. 2021, ISSN: 1424-8220. DOI: 10.3390/s21196547. Accessed: Jan. 29, 2026. [Online]. Available: https://www.mdpi.com/1424-8220/21/19/6547.

[3] J. Gunnarsson and V. Jansson, *A truck driver's guide to sustainable driving: A qualitative user experience evaluation of volvo trucks' on-board driver coaching system*, Bachelor's thesis, Report no. 2022:104, 2022.

[4] *Maintaining Ecodriving Benefits in the Long Term*, en-US. Accessed: Mar. 19, 2026. [Online]. Available: https://changing-transport.org/publications/maintaining-ecodriving-benefits-in-the-long-term/.

[5] R. Tu, J. Xu, T. Li, and H. Chen, "Effective and Acceptable Eco-Driving Guidance for Human-Driving Vehicles: A Review," en, *International Journal of Environmental Research and Public Health*, vol. 19, no. 12, p. 7310, Jan. 2022, ISSN: 1660-4601. DOI: 10.3390/ijerph19127310. Accessed: Jan. 29, 2026. [Online]. Available: https://www.mdpi.com/1660-4601/19/12/7310.

[6] M. Hasenjäger, M. Heckmann, and H. Wersing, "A Survey of Personalization for Advanced Driver Assistance Systems," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 2, pp. 335–344, Jun. 2020, ISSN: 2379-8904. DOI: 10.1109/TIV.2019.2955910. Accessed: Jan. 27, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/8918081.

[7] D. Loske and M. Klumpp, "Human-AI collaboration in route planning: An empirical efficiency-based analysis in retail logistics," *International Journal of Production Economics*, vol. 241, p. 108 236, Nov. 2021, ISSN: 0925-5273. DOI: 10.1016/j.ijpe.2021.108236. Accessed: Feb. 12, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925527321002127.

[8] Q. Luo, X. Lu, Z. Zang, H. Gong, X. Guo, and X. Chen, "A Real-Time Early Warning Framework for Multi-Dimensional Driving Risk of Heavy-Duty Trucks Using Trajectory Data," en, *Systems*, vol. 14, no. 2, p. 204, Feb. 2026,

ISSN: 2079-8954. DOI: 10.3390/systems14020204. Accessed: Mar. 17, 2026. [Online]. Available: https://www.mdpi.com/2079-8954/14/2/204.

[9] *How to decarbonize heavy-duty transport and make it affordable*, en, Aug. 2021. Accessed: Feb. 12, 2026. [Online]. Available: https://www.weforum.org/stories/2021/08/how-to-decarbonize-heavy-duty-transport-affordable/.

[10] M. Amoadu, J. O. Sarfo, and E. W. Ansah, "Working conditions of commercial drivers: A scoping review of psychosocial work factors, health outcomes, and interventions," *BMC Public Health*, vol. 24, p. 2944, Oct. 2024, ISSN: 1471-2458. DOI: 10.1186/s12889-024-20465-1. Accessed: Mar. 17, 2026. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11515491/.

[11] J. Costa et al., "From Dashboards to Dialogue: Evaluating a Conversational AI Coach for Performance Driving Skill Development," in *Proceedings of the 17th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, ser. AutomotiveUI '25, New York, NY, USA: Association for Computing Machinery, 2025, pp. 9–18, ISBN: 979-8-4007-2013-0. DOI: 10.1145/3744333.3747811. Accessed: Jan. 30, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/3744333.3747811.

[12] J. Orlovska, C. Wickman, R. Söderberg, D. Bark, C. Carlsson, and P. Gustavsson, "Design and implementation of driver coach application for pilot assist: A first validation study," *Transportation Research Interdisciplinary Perspectives*, vol. 25, p. 101 130, May 2024, ISSN: 2590-1982. DOI: 10.1016/j.trip.2024.101130. Accessed: Jan. 30, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590198224001167.

[13] Z. Marafie et al., "AutoCoach: An Intelligent Driver Behavior Feedback Agent with Personality-Based Driver Models," en, *Electronics*, vol. 10, no. 11, p. 1361, Jan. 2021, ISSN: 2079-9292. DOI: 10.3390/electronics10111361. Accessed: Jan. 30, 2026. [Online]. Available: https://www.mdpi.com/2079-9292/10/11/1361.

[14] M. Galvani, "History and future of driver assistance," *IEEE Instrumentation & Measurement Magazine*, vol. 22, no. 1, pp. 11–16, Feb. 2019, ISSN: 1941-0123. DOI: 10.1109/MIM.2019.8633345. Accessed: Feb. 9, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/8633345.

[15] E. Arroyo, S. Sullivan, and T. Selker, "CarCoach: A polite and effective driving coach," in *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '06, New York, NY, USA: Association for Computing Machinery, Apr. 2006, pp. 357–362, ISBN: 978-1-59593-298-3. DOI: 10.1145/1125451.1125529. Accessed: Feb. 9, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/1125451.1125529.

[16] J. Rehm, I. Reshodko, and O. E. Gundersen, "A virtual driving instructor that assesses driving performance on par with human experts," *Expert Systems with Applications*, vol. 248, p. 123 355, Aug. 2024, ISSN: 0957-4174. DOI: 10.1016/j.eswa.2024.123355. Accessed: Jan. 30, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424002203.

[17] Volvo Trucks, "How technology contributes to efficient driving," Volvo Trucks, White Paper, Jun. 2022, Accessed: Feb. 6, 2025. [Online]. Available: https:

//www.volvotrucks.com/content/dam/volvo-trucks/markets/master/home/news/insights/articles/2022/jun/how-technology-contributes-to-efficient-driving/How-technology-contribute-to-efficient-driving-White-paper.pdf.

[18] *Volvo Trucks Driver Guide*. Accessed: Mar. 17, 2026. [Online]. Available: https://driverguide.volvotrucks.com/lang/en/chassi/A00FH16/topic/370072/.

[19] G. Caldarini, S. Jaf, and K. McGarry, "A Literature Survey of Recent Advances in Chatbots," en, *Information*, vol. 13, no. 1, p. 41, Jan. 2022, arXiv:2201.06657 [cs.CL], ISSN: 2078-2489. DOI: 10.3390/info13010041. Accessed: Jun. 8, 2026. [Online]. Available: http://arxiv.org/abs/2201.06657.

[20] M. Hassenzahl and N. Tractinsky, "User experience - a research agenda," en, *Behaviour & Information Technology*, vol. 25, no. 2, pp. 91–97, Mar. 2006, ISSN: 0144-929X, 1362-3001. DOI: 10.1080/01449290500330331. Accessed: Jun. 8, 2026. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01449290500330331.

[21] S. T. Völkel, D. Buschek, M. Eiband, B. R. Cowan, and H. Hussmann, "Eliciting and Analysing Users Envisioned Dialogues with Perfect Voice Assistants," en, in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, Yokohama Japan: ACM, May 2021, pp. 1–15, ISBN: 978-1-4503-8096-6. DOI: 10.1145/3411764.3445536. Accessed: Jun. 8, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/3411764.3445536.

[22] A. Baughan, X. Wang, A. Liu, A. Mercurio, J. Chen, and X. Ma, "A Mixed-Methods Approach to Understanding User Trust after Voice Assistant Failures," en, in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, Hamburg Germany: ACM, Apr. 2023, pp. 1–16, ISBN: 978-1-4503-9421-5. DOI: 10.1145/3544548.3581152. Accessed: Jun. 8, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/3544548.3581152.

[23] R. Kaufman, J. Costa, and E. Kimani, "Effects of multimodal explanations for autonomous driving on driving performance, cognitive load, expertise, confidence, and trust," en, *Scientific Reports*, vol. 14, no. 1, p. 13 061, Jun. 2024, ISSN: 2045-2322. DOI: 10.1038/s41598-024-62052-9. Accessed: Jun. 8, 2026. [Online]. Available: https://www.nature.com/articles/s41598-024-62052-9.

[24] D. Yi et al., "Implicit Personalization in Driving Assistance: State-of-the-Art and Open Issues," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 3, pp. 397–413, Sep. 2020, ISSN: 2379-8904. DOI: 10.1109/TIV.2019.2960935. Accessed: Jan. 30, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8936867.

[25] E. Gilman, A. Keskinarkaus, S. Tamminen, S. Pirttikangas, J. Röning, and J. Riekki, "Personalised assistance for fuel-efficient driving," *Transportation Research Part C: Emerging Technologies*, Technologies to support green driving, vol. 58, pp. 681–705, Sep. 2015, ISSN: 0968-090X. DOI: 10.1016/j.trc.2015.02.007. Accessed: Jan. 30, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X15000480.

[26] J. Baig, "A Personalised Intervention Model for Improving the Effectiveness of Driving-Behaviour Apps," in *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, ser. UMAP '20, New York, NY, USA: Association for Computing Machinery, Jul. 2020, pp. 372–375, ISBN: 978-1-4503-6861-2. DOI: 10.1145/3340631.3398680. Accessed: Jan. 30, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/3340631.3398680.

[27] S. Katreddi, S. Kasani, and A. Thiruvengadam, "A Review of Applications of Artificial Intelligence in Heavy Duty Trucks," en, *Energies*, vol. 15, no. 20, p. 7457, Jan. 2022, ISSN: 1996-1073. DOI: 10.3390/en15207457. Accessed: Jan. 30, 2026. [Online]. Available: https://www.mdpi.com/1996-1073/15/20/7457.

[28] P. R. Potdar and S. M. Parikh, "Internet of things (IoT) and artificial intelligence (AI) enabled framework for smart fleet management," en, *OPSEARCH*, May 2025, ISSN: 0975-0320. DOI: 10.1007/s12597-025-00961-7. Accessed: Feb. 9, 2026. [Online]. Available: https://doi.org/10.1007/s12597-025-00961-7.

[29] Kusnawi, M. A. Wibowo, and Ridwan Sanjaya, "Real-Time Explainable Concept Drift Detection for Eco-Driving in Mining Trucks using KSWIN and Event-Triggered SHAP," *Journal of Information Systems and Informatics*, vol. 8, no. 2, pp. 1534–1556, Apr. 2026, ISSN: 2656-4882, 2656-5935. DOI: 10.63158/journalisi.v8i2.1551. Accessed: Jun. 8, 2026. [Online]. Available: https://journal-isi.org/index.php/isi/article/view/1551.

[30] E. Mosca, F. Szigeti, S. Tragianni, D. Gallagher, and G. Groh, "SHAP-based explanation methods: A review for NLP interpretability," in *Proceedings of the 29th International Conference on Computational Linguistics*, N. Calzolari et al., Eds., Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 4593–4603. [Online]. Available: https://aclanthology.org/2022.coling-1.406/.

[31] D. I. Tselentis and E. Papadimitriou, "Driver Profile and Driving Pattern Recognition for Road Safety Assessment: Main Challenges and Future Directions," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 4, pp. 83–100, 2023, ISSN: 2687-7813. DOI: 10.1109/OJITS.2023.3237177. Accessed: Feb. 9, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10018541.

[32] R. Massoud, F. Bellotti, R. Berta, A. De Gloria, and S. Poslad, "Eco-driving Profiling and Behavioral Shifts Using IoT Vehicular Sensors Combined with Serious Games," in *2019 IEEE Conference on Games (CoG)*, ISSN: 2325-4289, Aug. 2019, pp. 1–8. DOI: 10.1109/CIG.2019.8847992. Accessed: Feb. 9, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8847992.

[33] B. Higgs and M. Abbas, "Segmentation and Clustering of Car-Following Behavior: Recognition of Driving Patterns," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 81–90, Feb. 2015, ISSN: 1558-0016. DOI: 10.1109/TITS.2014.2326082. Accessed: Feb. 9, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6832510.

[34] L. D. Seixas, F. C. Corrêa, H. V. Siqueira, F. Trojan, and P. Afonso, "Vehicle Industry Big Data Analysis Using Clustering Approaches," en, in *Optimization, Learning Algorithms and Applications*, A. I. Pereira, A. Mendes, F. P. Fernandes, M. F. Pacheco, J. P. Coelho, and J. Lima, Eds., Cham: Springer Nature Switzerland, 2024, pp. 312–325, ISBN: 978-3-031-53036-4. DOI: 10.1007/978-3-031-53036-4_22.

[35] M. Henzler, A. Boller, M. Buchholz, and K. Dietmeyer, "Are Truck Drivers Ready to Save Fuel? The Objective and Subjective Effectiveness of an Ecological Driver Assistance System," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, ISSN: 2153-0017, Sep. 2015, pp. 2007–2012. DOI: 10.1109/ITSC.2015.325. Accessed: Feb. 10, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7313417.

[36] V. C. Magaña and M. Muñoz-Organero, "Artemisa: A Personal Driving Assistant for Fuel Saving," *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2437–2451, Oct. 2016, ISSN: 1558-0660. DOI: 10.1109/TMC.2015.2504976. Accessed: Feb. 10, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7345559.

[37] A. Leslie and D. Murray, "An Analysis of the Operational Costs of Trucking: 2023 Update," en,

[38] E. W. Martin, N. D. Chan, and S. A. Shaheen, "How Public Education on Ecodriving Can Reduce Both Fuel Use and Greenhouse Gas Emissions," EN, *Transportation Research Record*, vol. 2287, no. 1, pp. 163–173, Jan. 2012, ISSN: 0361-1981. DOI: 10.3141/2287-20. Accessed: Feb. 10, 2026. [Online]. Available: https://doi.org/10.3141/2287-20.

[39] J. N. Barkenbus, "Eco-driving: An overlooked climate change initiative," *Energy Policy*, vol. 38, no. 2, pp. 762–769, Feb. 2010, ISSN: 0301-4215. DOI: 10.1016/j.enpol.2009.10.021. Accessed: Feb. 10, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301421509007484.

[40] M. S. Young, S. A. Birrell, and N. A. Stanton, "Safe driving in a green world: A review of driver performance benchmarks and technologies to support smart driving," *Applied Ergonomics*, Applied Ergonomics and Transportation Safety, vol. 42, no. 4, pp. 533–539, May 2011, ISSN: 0003-6870. DOI: 10.1016/j.apergo.2010.08.012. Accessed: Feb. 10, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003687010001262.

[41] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017. Accessed: Mar. 12, 2026. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

[42] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, New York, NY, USA: Association for Computing Machinery, Aug. 2016, pp. 1135–1144, ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939778. Accessed: Mar. 12, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/2939672.2939778.

[43] D. Alvarez-Melis and T. S. Jaakkola, *On the Robustness of Interpretability Methods*, arXiv:1806.08049 [cs], Jun. 2018. DOI: 10.48550/arXiv.1806.08049. Accessed: Mar. 12, 2026. [Online]. Available: http://arxiv.org/abs/1806.08049.

[44] K. Aas, M. Jullum, and A. Løland, "Explaining individual predictions when features are dependent: More accurate approximations to Shapley values," *Artificial Intelligence*, vol. 298, p. 103 502, Sep. 2021, ISSN: 0004-3702. DOI: 10.1016/j.artint.2021.103502. Accessed: Mar. 12, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370221000539.

[45] S. M. Lundberg et al., "From local explanations to global understanding with explainable AI for trees," en, *Nature Machine Intelligence*, vol. 2, no. 1, pp. 56–67, Jan. 2020, ISSN: 2522-5839. DOI: 10.1038/s42256-019-0138-9. Accessed: Mar. 12, 2026. [Online]. Available: https://www.nature.com/articles/s42256-019-0138-9.

[46] H. Naveed et al., "A Comprehensive Overview of Large Language Models," *ACM Trans. Intell. Syst. Technol.*, vol. 16, no. 5, 106:1–106:72, Aug. 2025, ISSN: 2157-6904. DOI: 10.1145/3744746. Accessed: Feb. 6, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/3744746.

[47] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, with Language Models*, 3rd. 2026, Online manuscript released January 6, 2026. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/.

[48] Y. Zhang et al., "Sirens Song in the AI Ocean: A Survey on Hallucination in Large Language Models," *Computational Linguistics*, vol. 51, no. 4, pp. 1373–1418, Dec. 2025, ISSN: 0891-2017. DOI: 10.1162/COLI.a.16. Accessed: Feb. 9, 2026. [Online]. Available: https://doi.org/10.1162/COLI.a.16.

[49] X. Wang et al., "Searching for Best Practices in Retrieval-Augmented Generation," en, in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Miami, Florida, USA: Association for Computational Linguistics, 2024, pp. 17 716–17 736. DOI: 10.18653/v1/2024.emnlp-main.981. Accessed: Feb. 10, 2026. [Online]. Available: https://aclanthology.org/2024.emnlp-main.981.

[50] T. akar and H. Emekci, "Maximizing RAG efficiency: A comparative analysis of RAG methods," en, *Natural Language Processing*, vol. 31, no. 1, pp. 1–25, Jan. 2025, ISSN: 2977-0424. DOI: 10.1017/nlp.2024.53. Accessed: Feb. 10, 2026. [Online]. Available: https://www.cambridge.org/core/journals/natural-language-processing/article/maximizing-rag-efficiency-a-comparative-analysis-of-rag-methods/D7B259BCD35586E04358DF06006E0A85.

[51] Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2024. DOI: 10.48550/arXiv.2312.10997.

[52] G. Marvin, N. Hellen, D. Jjingo, and J. Nakatumba-Nabende, "Prompt Engineering in Large Language Models," en, in *Data Intelligence and Cognitive Informatics*, I. J. Jacob, S. Piramuthu, and P. Falkowski-Gilski, Eds., Sin-

gapore: Springer Nature, 2024, pp. 387–402, ISBN: 978-981-99-7962-2. DOI: 10.1007/978-981-99-7962-2_30.

[53] A. Kong et al., "Better Zero-Shot Reasoning with Role-Play Prompting," en, in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Mexico City, Mexico: Association for Computational Linguistics, 2024, pp. 4099–4113. DOI: 10.18653/v1/2024.naacl-long.228. Accessed: Mar. 4, 2026. [Online]. Available: https://aclanthology.org/2024.naacl-long.228.

[54] Z. Pan et al., *On Memory Construction and Retrieval for Personalized Conversational Agents*, en, arXiv:2502.05589 [cs], Mar. 2025. DOI: 10.48550/arXiv.2502.05589. Accessed: Mar. 3, 2026. [Online]. Available: http://arxiv.org/abs/2502.05589.

[55] Z. Tan and M. Jiang, *User Modeling in the Era of Large Language Models: Current Research and Future Directions*, en, arXiv:2312.11518 [cs], Dec. 2023. DOI: 10.48550/arXiv.2312.11518. Accessed: Mar. 3, 2026. [Online]. Available: http://arxiv.org/abs/2312.11518.

[56] E. Purificato, L. Boratto, and E. W. D. Luca, *User Modeling and User Profiling: A Comprehensive Survey*, en, arXiv:2402.09660 [cs], Feb. 2024. DOI: 10.48550/arXiv.2402.09660. Accessed: Mar. 5, 2026. [Online]. Available: http://arxiv.org/abs/2402.09660.

[57] K. Ko, T. T. Wai, and A. N. Oo, "Expanding Conversational AI Beyond Context Windows: A Memory-Augmented Approach," in *2025 6th International Conference on Advanced Information Technologies (ICAIT)*, Nov. 2025, pp. 1–6. DOI: 10.1109/ICAIT68809.2025.11236754. Accessed: Mar. 4, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/11236754.

[58] J. Zhang, *Guided Profile Generation Improves Personalization with LLMs*, en, arXiv:2409.13093 [cs.CL], Sep. 2024. DOI: 10.48550/arXiv.2409.13093. Accessed: May 19, 2026. [Online]. Available: http://arxiv.org/abs/2409.13093.

[59] Z. Zhang et al., *A Survey on the Memory Mechanism of Large Language Model based Agents*, en, arXiv:2404.13501 [cs], Apr. 2024. DOI: 10.48550/arXiv.2404.13501. Accessed: Mar. 4, 2026. [Online]. Available: http://arxiv.org/abs/2404.13501.

[60] P. Offermann, O. Levina, M. Schönherr, and U. Bub, "Outline of a design science research process," in *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, ser. DESRIST '09, New York, NY, USA: Association for Computing Machinery, May 2009, pp. 1–11, ISBN: 978-1-60558-408-9. DOI: 10.1145/1555619.1555629. Accessed: Mar. 12, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/1555619.1555629.

[61] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," en,

[62] *Welcome to LightGBMs documentation! LightGBM 4.6.0.99 documentation*. Accessed: Mar. 12, 2026. [Online]. Available: https://lightgbm.readthedocs.io/en/latest/.

[63] *Scikit-learn: Machine learning in Python scikit-learn 1.8.0 documentation*. Accessed: Mar. 13, 2026. [Online]. Available: https://scikit-learn.org/stable/index.html.

[64] *GroupKFold*, en. Accessed: Mar. 30, 2026. [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.model_selection.GroupKFold.html.

[65] J.-M. Dufour, "Coefcients of determination," en,

[66] *3.4. Metrics and scoring: Quantifying the quality of predictions*, en. Accessed: May 8, 2026. [Online]. Available: https://scikit-learn/stable/modules/model_evaluation.html.

[67] A. Botchkarev, *Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology*, 2018. arXiv: 1809.03006. [Online]. Available: https://arxiv.org/pdf/1809.03006.

[68] T. K. Jonker Alexandra, *What Is Cosine Similarity? | IBM*, en, Jul. 2025. Accessed: May 8, 2026. [Online]. Available: https://www.ibm.com/think/topics/cosine-similarity.

[69] J. S. Milton and J. C. Arnold, *Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences*, 4th. USA: McGraw-Hill, Inc., 2002, ISBN: 0071198598.

[70] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, *Ragas: Automated Evaluation of Retrieval Augmented Generation*, en, arXiv:2309.15217 [cs], Apr. 2025. DOI: 10.48550/arXiv.2309.15217. Accessed: Apr. 20, 2026. [Online]. Available: http://arxiv.org/abs/2309.15217.

[71] *MTEB Leaderboard - a Hugging Face Space by mteb*. Accessed: Mar. 30, 2026. [Online]. Available: https://huggingface.co/spaces/mteb/leaderboard.

[72] P. Liang et al., *Holistic Evaluation of Language Models*, en, arXiv:2211.09110 [cs], Oct. 2023. DOI: 10.48550/arXiv.2211.09110. Accessed: Mar. 31, 2026. [Online]. Available: http://arxiv.org/abs/2211.09110.

[73] L. Lannelongue, J. Grealey, and M. Inouye, "Green Algorithms: Quantifying the Carbon Footprint of Computation," en, *Advanced Science*, vol. 8, no. 12, p. 2 100 707, Jun. 2021, ISSN: 2198-3844, 2198-3844. DOI: 10.1002/advs.202100707. Accessed: Mar. 31, 2026. [Online]. Available: https://advanced.onlinelibrary.wiley.com/doi/10.1002/advs.202100707.

[74] *LLM Leaderboard - Comparison of over 100 AI models from OpenAI, Google, DeepSeek & others*, en. Accessed: Mar. 31, 2026. [Online]. Available: https://artificialanalysis.ai/leaderboards/models.

[75] P. Chhikara, D. Khant, S. Aryan, T. Singh, and D. Yadav, *Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory*, en, arXiv:2504.19413 [cs], Apr. 2025. DOI: 10.48550/arXiv.2504.19413. Accessed: Mar. 4, 2026. [Online]. Available: http://arxiv.org/abs/2504.19413.

[76] A. Szymanski, N. Ziems, H. A. Eicher-Miller, T. J.-J. Li, M. Jiang, and R. A. Metoyer, "Limitations of the LLM-as-a-Judge Approach for Evaluating LLM Outputs in Expert Knowledge Tasks," in *Proceedings of the 30th International*

Conference on Intelligent User Interfaces, ser. IUI '25, New York, NY, USA: Association for Computing Machinery, Mar. 2025, pp. 952–966, ISBN: 979-8-4007-1306-4. DOI: 10.1145/3708359.3712091. Accessed: Apr. 29, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/3708359.3712091.

[77] N. Beyranvand, H. Dastmalchi, A. An, H. Davoudi, W. Chan, and R. DiCarlantonio, "GEAR: A Scalable and Interpretable Evaluation Framework for RAG-Based Car Assistant Systems," en,

[78] C. Van Der Lee, A. Gatt, E. Van Miltenburg, S. Wubben, and E. Krahmer, "Best practices for the human evaluation of automatically generated text," en, in *Proceedings of the 12th International Conference on Natural Language Generation*, Tokyo, Japan: Association for Computational Linguistics, 2019, pp. 355–368. DOI: 10.18653/v1/W19-8643. Accessed: Apr. 29, 2026. [Online]. Available: https://aclanthology.org/W19-8643.

A

Feature Appendix

A.1 Feature context JSON

```
{
  "DRIVE_SPEED": {
    "name": "Average driving speed",
    "description": "Average vehicle speed during periods when the truck is moving."
    },

  "IDLE_IROLL": {
    "name": "Idle I-Roll operation",
    "description": "Proportion of total driving time spent in a low engine speed and low torque state associated with very low engine load conditions that may occur during rolling or I-Roll operation"
  },

    "ECOROLL_ACT_DIST": {
    "name": "Eco-Roll activated distance",
    "description": "Percentage of total driven distance during which the Eco-Roll function is active. Eco-Roll disengages the driveline and allows the truck to roll freely in order to reduce power losses and save energy."},

    "ECOROLL_ENA_DIST": {
    "name": "Eco-Roll enabled distance",
    "description": "Percentage of total driven distance during which the Eco-Roll system was enabled. Eco-Roll becomes enabled when the driver activates the Eco-Roll function, allowing the system to disengage the driveline when conditions permit."},

    "SVC_BRAKE": {
    "name": "Service brake activations",
    "description": "Number of service brake activations per 100 km. The counter increases each time the brake pedal is actuated." },

    "GB_MAN_TIME": {
    "name": "Manual gear mode time",
```

```
26    "description": "Percentage of total driving time during which the
      gearbox is operated in manual mode using the gear lever or control
      buttons." },
27
28    "GB_AUT_TIME": {
29    "name": "Automatic gear mode time",
30    "description": "Percentage of total driving time during which the
      gearbox operates in automatic mode." },
31
32    "COASTING_DIST": {
33    "name": "Coasting distance",
34    "description": "Percentage of total driven distance during which the
      vehicle is coasting. Coasting occurs when the vehicle is moving
      without fuel being injected into the engine."},
35
36    "COASTING_SPEED": {
37    "name": "Average coasting speed",
38    "description": "Average vehicle speed during periods when the vehicle
      is coasting (with no fuel injection)."},
39
40    "IDLE_TIME": {
41    "name": "Idling time",
42    "description": "Percentage of total time during which the vehicle is
      stationary while the engine is running and no auxiliary equipment (PTO
      ) is active."},
43
44    "TORQUE_MODE_3": {
45    "name": "Engine torque mode 3",
46    "description": "Percentage of total driving time during which the
      engine operates in a specific engine speed and torque operating range
      defined by the engine control system. Values are normalized between 0
      and 100, representing the share of total engine operating time spent
      in this mode."},
47
48    "TORQUE_MODE_30": {
49    "name": "Maximum torque time",
50    "description": "Percentage of total driving time during which the
      engine operated at the maximum torque the engine can deliver. Values
      are normalized between 0 and 100, representing the share of total
      engine operating time spent in this mode."},
51
52    "ACC_900_1250_RPM": {
53    "name": "Acceleration at 900-1250 RPM",
54    "description": "Percentage of total driving time where the engine
      operates in the 900-1250 RPM range while producing torque. This
      represents periods of acceleration or load where the engine speed is
      relatively low. Values are normalized between 0 and 100, representing
      the share of total engine operating time spent in this range."},
55
```

```
56    "ACC_1250_1400_RPM": {
57    "name": "Acceleration at 1250-1400 RPM",
58    "description": "Percentage of total driving time where the engine
      operates in the 1250-1400 RPM range while producing torque. This
      represents periods of acceleration where the engine runs at a
      moderately higher speed. Values are normalized between 0 and 100,
      representing the share of total engine operating time spent in this
      range."},
59
60    "PEDAL_TIME": {
61    "name": "Time with accelerator pedal engaged",
62    "description": "Percentage of total driving time where the accelerator
       pedal is actively pressed, representing the share of the trip during
      which the driver maintains throttle input."},
63
64    "GB_HIGH": {
65    "name": "High gear usage",
66    "description": "Percentage of total driving distance during which the
      gearbox operates in the high range. The high range represents the
      higher gear ratios of the transmission that are typically used at
      higher vehicle speeds."},
67
68    "BRAKE_MODE_ACT": {
69    "name": "Brake mode activations",
70    "description": "Number of times the driver activates the brake program
      ."},
71
72    "BRAKE_TIME": {
73    "name": "Braking time",
74    "description": "Percentage of total driving time during which the
      vehicle is braking."},
75
76    "CRUISE_TIME": {
77    "name": "Cruising time",
78    "description": "Percentage of total driving time during which cruise
      control is active. Cruise control maintains the vehicle speed
      automatically without continuous accelerator pedal input."},
79
80    "POWER_MODE_TIME": {
81    "name": "Performance drive mode time",
82    "description": "Percentage of total driving time during which the
      vehicle operates in Performance drive mode. This mode prioritizes
      vehicle responsiveness and engine power over fuel efficiency." },
83
84    "DRIVING_MODE_E_DIST": {
85    "name": "Economy drive mode distance",
86    "description": "Percentage of total driving distance during which the
      vehicle operates in Economy drive mode. In this mode the vehicle
      prioritizes fuel efficiency by optimizing engine response, gear
```

```
    shifting behaviour and predictive functions such as I-See." },
87
88      "MAX_ENG_SPEED": {
89      "name": "Maximum engine overspeed",
90      "description": "Maximum recorded engine speed (in RPM) during engine
        overspeed conditions. Engine overspeed occurs when the engine operates
         above its intended speed range."},
91
92      "KICKDOWN_MODE_TIME": {
93      "name": "Kickdown mode usage time",
94      "description": "Percentage of total driving time during which kickdown
         mode is active. Kickdown is triggered when the accelerator pedal is
        fully pressed, causing the transmission to downshift and the engine to
         deliver power."},
95
96      "KICKDOWN_MODE_ACT": {
97      "name": "Kickdown activations",
98      "description": "Number of times kickdown mode is activated during
        driving. Kickdown is triggered when the accelerator pedal is fully
        pressed, causing the transmission to downshift for maximum engine
        power."}
99  }
100 }
```

B

Memory Appendix

B.1 Prompt

```
101
102         system_prompt = (
103             "You are a memory extraction component for a driver coaching
    assistant.\n"
104             "Your job: decide if the user's message contains stable
    information worth remembering for future personalization.\n"
105             "Only extract: preferences, goals, constraints, or stable
    context.\n"
106             "A user may have MULTIPLE constraints over time; do not assume
     there is only one.\n"
107             "If the user states their name, extract it using key='name'.\n
    "
108             "You must use only valid mtype/key combinations.\n"
109             "Only extract a field if it is explicitly stated or strongly
    implied.\n"
110             "Do NOT guess or infer missing information.\n"
111             "If a field is not present, DO NOT create an item for it.\n"
112             "Do NOT include placeholder values such as 'null', 'none', or
    'unknown'.\n"
113             "Do NOT store transient mood, greetings, or sensitive personal
     data.\n"
114             "Output MUST be valid JSON only (no markdown, no extra text)."
115         )
116
117         user_prompt = (
118             "Extract up to 5 memory items from the conversation context
    below.\n"
119             "If there is nothing stable to store, return an empty list.\n\
    n"
120             "Important:\n"
121             "- Use the previous assistant message only to interpret the
    current user message.\n"
122             "- Do NOT store facts stated only by the assistant.\n"
123             "- Only store information that is confirmed, stated, or
    clearly implied by the current user message.\n\n"
```

```
            "JSON schema:\n"
            "{\n"
            '  "items": [\n'
            "    {\n"
            '      "mtype": "preference|goal|constraint|context",\n'
            '      "key": "verbosity|tone|format|goal_primary|constraint|name|driving_context",\n'
            '      "value": "string",\n'
            '      "source": "explicit|inferred",\n'
            "    }\n"
            "  ]\n"
            "}\n\n"
            "Key rules:\n"
            "- If mtype='preference', key must be one of: verbosity, tone, format.\n"
            "- If mtype='goal', key must be: goal_primary.\n"
            "- If mtype='constraint', key must be: constraint.\n"
            "- If mtype='context', key must be one of: name, driving_context.\n"
            "- 'clear and concise' or 'more details' are usually preferences about verbosity, not constraints.\n"
            "- Use key='constraint' for every constraint.\n"
            "- A message may contain more than one constraint.\n"
            "- If the user says their name, store it with key='name'.\n\n"
            "Previous assistant message:\n"
            f"{previous_assistant_text if previous_assistant_text else '[none]'}\n\n"
            "Current user message:\n"
            f"{text}"
        )

        payload = {
            "messages": [
                {"role": "system", "content": system_prompt},
                {"role": "user", "content": user_prompt},
            ],
            "max_tokens": 300,
            "temperature": 0.0,
        }
}
```

B.2 Example of stored JSON

```
    {
  "driver_1": {
    "name": {
      "mtype": "context",
      "key": "name",
```

```
      "value": "Josefine",
      "source": "explicit",
      "updated_at": "2026-04-28T10:43:38.481003+00:00"
    },
    "format": {
      "mtype": "preference",
      "key": "format",
      "value": "bullet points",
      "source": "explicit",
      "updated_at": "2026-05-06T08:32:34.437765+00:00"
    },
    "goal_primary": {
      "mtype": "goal",
      "key": "goal_primary",
      "value": "increase acceleration time within the 900-1250 RPM range",
      "source": "explicit",
      "updated_at": "2026-04-28T07:32:47.428786+00:00"
    },
    "verbosity": {
      "mtype": "preference",
      "key": "verbosity",
      "value": "concise and clear",
      "source": "explicit",
      "updated_at": "2026-05-06T08:50:51.774055+00:00"
    }
  }
}
}
```

C

Prompt Appendix

```
<persona>
### ROLE
You are a professional Volvo Truck driver coach specializing in eco-driving and fuel efficiency.

### OBJECTIVE
Help truck drivers understand their trip performance and improve fuel efficiency based on their driving behaviour.
Provide clear, constructive, and encouraging coaching.

### GUIDELINES
- Keep responses concise, typically 5-6 sentences. Respond with no more than 7 sentences.
- If personalization preferences exist (tone, verbosity, format), follow those instead of the default guideline.
- When relevant, briefly explain why a behaviour affects fuel consumption.

### SCOPE
- You specialize in truck driving behaviour and eco-driving.
- Always respond politely to greetings or small talk.
### TONE
The tone must be trustworthy, approachable, intelligent and positive.'</persona>

<user_context>
Personalization context for this driver (internal only):
Use this information to adapt tone, wording, and coaching style.
Rules:
- If a driver name is available, you may occasionally address the driver by name.
- Respect stored preferences (tone, verbosity, format).
- Stored constraints must always be respected and override conflicting suggestions.
Driver name:
- Kasper
Constraints:
- do not mention idling
</user_context>
```

```
223
224 Context for this driver's recent trip:
225 <driver_telemetry>
226 <driver_status>
227 BELOW AVERAGE
228 Driver performs below average. The tone should be more straightforward and
        focus on improving to average level
229 </driver_status>
230 <follow_up>
231 - Previously coached feature: ACC_900_1250_RPM
232 - Previous period: 2025-04
233 - Current period: 2025-05
234 - Status: improved
235 - Still a current issue: True
236 - Instruction: If the user asks about their recent driving performance,
      briefly mention this follow-up first. If the status is improved,
      acknowledge the progress. If the status is unchanged or worsened, say
      it still needs attention. Then focus mainly on the current period's
      feedback.
237 </follow_up>
238 <strengths>
239 - Average driving speed
240 </strengths>
241 <areas_for_improvement>
242 - Acceleration at 900-1250 RPM:
243   * What it is: Percentage of total driving time where the engine operates
        in the 900-1250 RPM range while producing torque. This represents
      periods of acceleration or load where the engine speed is relatively
      low. Values are normalized between 0 and 100, representing the share
      of total engine operating time spent in this range.
244   * Coaching Rule: increase
245
246   * RAG LOOKUP, PRIORITIZE THIS:
247 <official_volvo_rules_for_improvement>
248 --- MANUAL EXCERPT FOR: 'Acceleration at 900-1250 RPM' ---
249 [score 0.504] somewhere between 900  1500 rpm little bit depending on the
      engine size and Euro rating. Currently our engines tend to be better
      on the lower part of the green band but sometimes we need horsepower
      in order to climb. More horsepower we have on the upper part of the
      green band. So the best rule is to try to stay within green band while
       climbing. The easiest way to meet this is to stay with the I-Shift in
       the automated mode and control the gearbox by foot. Try to find a
      pedal position where you can climb with steady revolutions on as high
      gear as possible. E-mode should be used in paved road contidions. Use
      of P-mode will increase the fuel consumption since the drivetrain is
      aiming for maximum power, not for economical climb. 4.3.4.1. Volvo
      Trucks Guideline for Climbing  Best way to utilise correct gear for
      climbing is to use I-Shift in automated mode and E or E+ gear shifting
       strategy  The most economical way to climb up is to climb with engine
```

```
    revolutions on green band and with a gear that is closest possible to
    the direct gear  Gear 12 on direct drive  Gear 11
250
251 [score 0.492] 98)  For practicing nice and smooth acceleration you can
    pick turbo charge gauge to your favourite screen on drivers
    information display. Be gentle with the throttle on gears 1-6. Once
    you are on the high range 7-12, find a pedal position that is minimum
    position to keep turbo charge fully up in the end of the orange part
    of gauge. While accelerating, gear shifts should occur in lower part
    of green band. To learn the driveline characteristics it is good to
    keep the turbo charge gauge visible for at least three weeks on normal
     driving. After that you have probably learned, how the driveline
    behaves and you can replace it with something else instead. 4.1.1.5.
    Chart: Increasing Speed 0  50 km/h, Flat Road These charts are more
    for trainer use than to show drivers in training. Although they might
    be shown if a question pops up in the training and you can explain the
     matter by using the chart. Before showing this information to drivers,
     make sure that you know how to read it. Measurements were done in
    Hällered Proving Ground which is a testing center for Volvo Cars and
    Volvo Group close to
252
253 [score 0.485] km/h 100% pedal o 30  80 km/h with resume function of cruise
     control  Eco3, all o Same as Eco0 but now with three Eco-stars o
    Three Eco-stars have impact on gear shifting strategy so you can see
    the results in consumption both pedal and Cruise control resume.
    Manual, Every Gear o Gear shifting sequence 2-5-7-8-9-10-11-12 o Easy
    on gears 2 and 5 o Shifting point about 1200  1300 revolutions  Manual,
     Skip Gears o Gear shifting sequence 2-5-7-9-11-12 o Easy on gears 2
    and 5 o Shifting point between 1400  1500 revolutions 4.1.1.7. Chart:
    Increasing Speed 0  90 km/h, Flat Road Remarkable with this chart is,
    how much more fuel you need to accelerate to 90 km/h than to 80 km/h.
    We will talk about impact of speed to fuel consumption in later
    chapter but its also good to acknowledge that the cruising speed
    chosen will have its impact on accelerating efficiency as well.
    Accelerating
254
255 </official_volvo_rules_for_improvement>
256 </areas_for_improvement>
257 </driver_telemetry>
258
259 <constraints>
260 STRICT_TECHNICAL_REFUSAL:
261 - You currently have NO access to technical manuals.
262 - You are FORBIDDEN from using your internal training data to answer
    mechanical, operational, or technical truck questions.
263 - If asked a technical question, you MUST NOT say 'I think' or 'Maybe'.
264 - MANDATORY RESPONSE: 'I do not have access to the technical manuals right
     now.'
265 </constraints>
```

```
266 }
```

D

Survey Appendix

D.1 Consent form

Evaluation of an AI-Based Eco-Driving Coaching System

This project is conducted as part of a Masters thesis at Chalmers University of Technology in collaboration with Volvo Group.

The purpose of the study is to evaluate an AI-based conversational system designed to provide post-trip eco-driving feedback to truck drivers. Participation involves answering a questionnaire about the systems feedback and performance. The questionnaire takes approximately 60 minutes to complete. Your participation is voluntary, and you may stop at any time before submitting your responses without providing a reason. The responses collected will be used for academic research purposes only and will be reported in an anonymized form as part of the thesis. All data will be stored securely and will only be accessible to the researchers involved in this project.

By proceeding, you confirm that:

- You have read and understood the information above
- You understand that your responses will be used for academic research purposes
- You voluntarily agree to participate in this study
- You may withdraw at any time before submitting your responses

If you have any questions, please contact:

Josefine Karlsson and Kasper Lundberg

D.2 Examples from Questionnaire - Comparing RAG vs No RAG

Note: The full questionnaire consists of 10 scenarios evaluating different feature combinations. Because the layout, questions, and Likert scales remain identical across all scenarios, only one representative template is shown below to illustrate the testing structure.

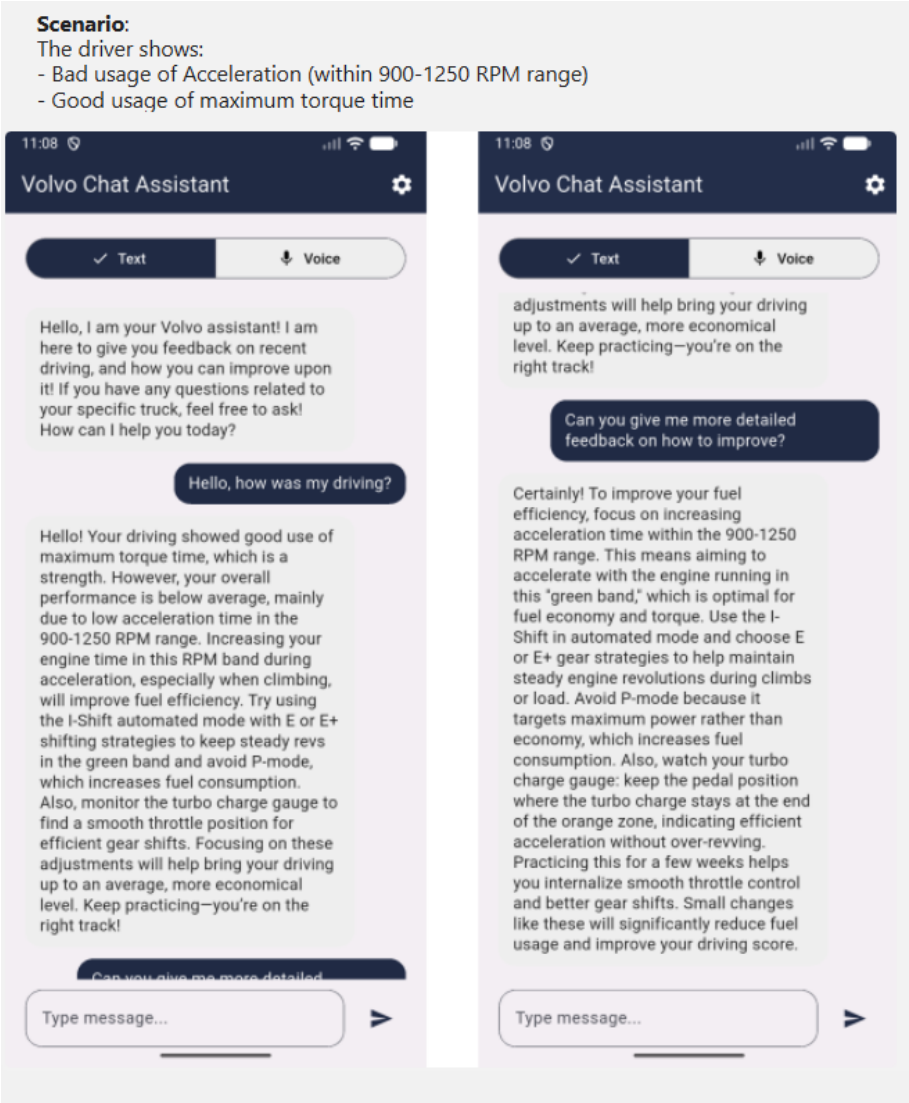


Figure D.1: An example of a scenario where RAG is not utilized

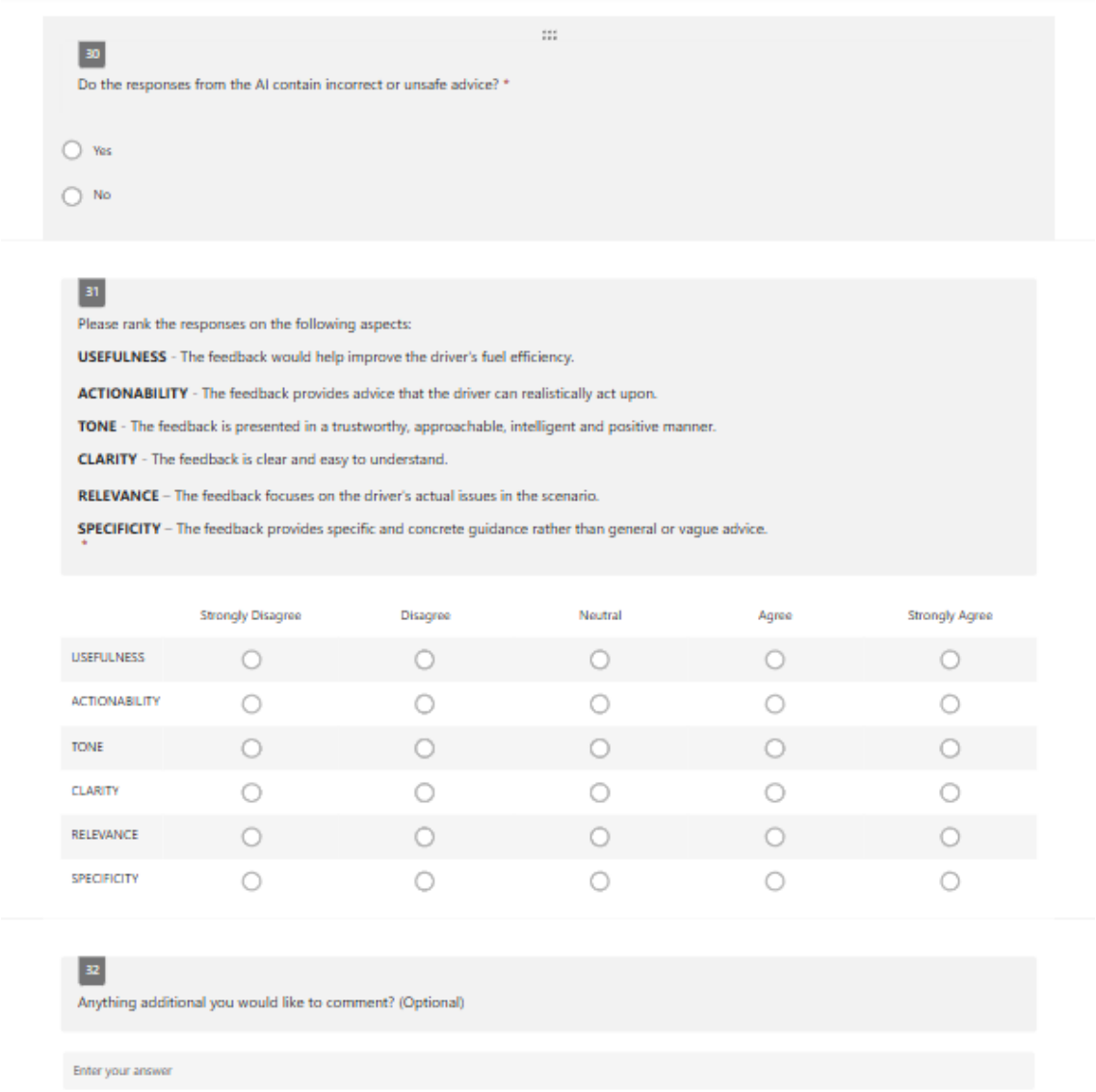


Figure D.2: Questions for follow up for each scenario

D.3 Examples from Questionnaire - Comparing no personalization vs with personalization

Note: The full questionnaire consists of 6 scenarios evaluating different feature combinations. Because the layout, questions, and remain identical across all scenarios, only one representative template is shown below to illustrate the testing structure.

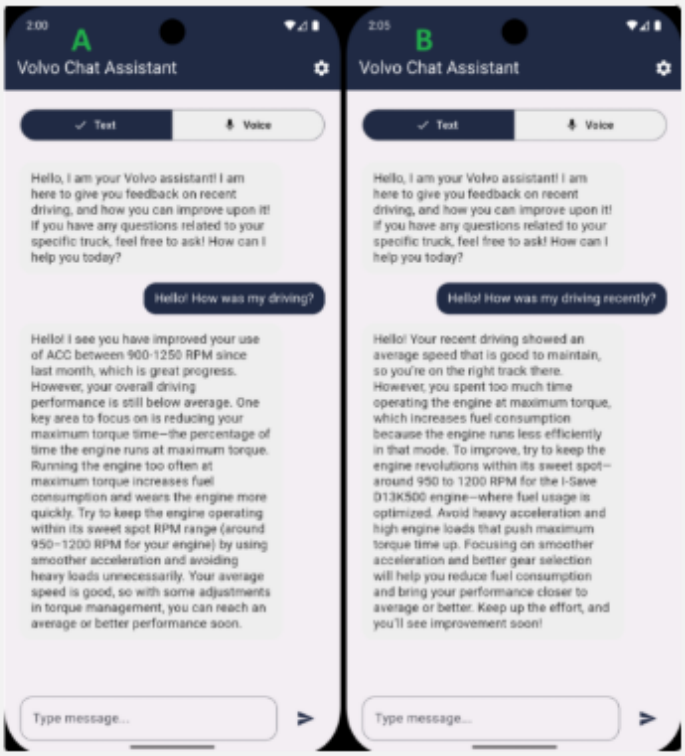
Scenario:
The driver shows:
- Poor use of maximum engine torque
- Good average speed

In the previous coaching session:
The driver had:
- Poor acceleration
- Good average speed

Please select the response you believe provides **better and more appropriate feedback** for the given driver

42

*



○ I prefer the response in A (left)

○ I prefer the response in B (right)

Figure D.3: An example of a comparison scenario

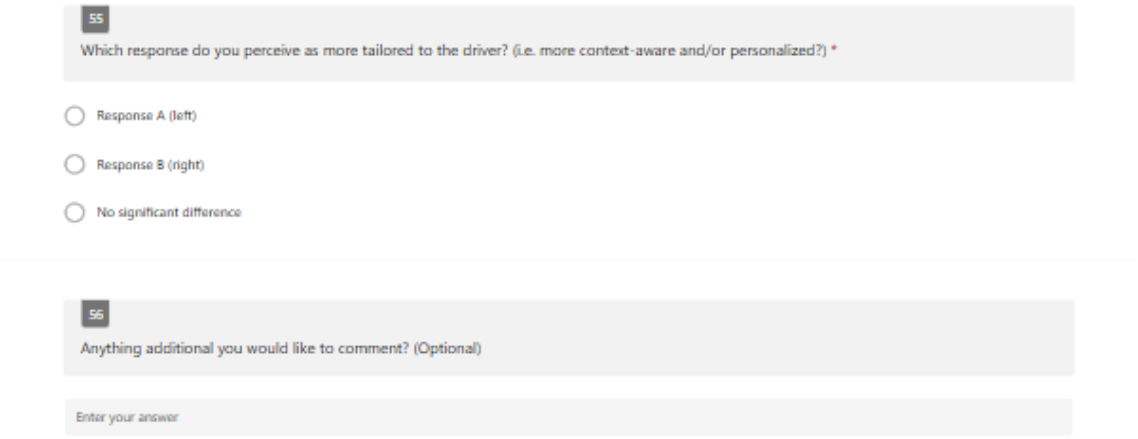


Figure D.4: Questions for feedback for comparison scenario

D.4 Questions regarding overall impression

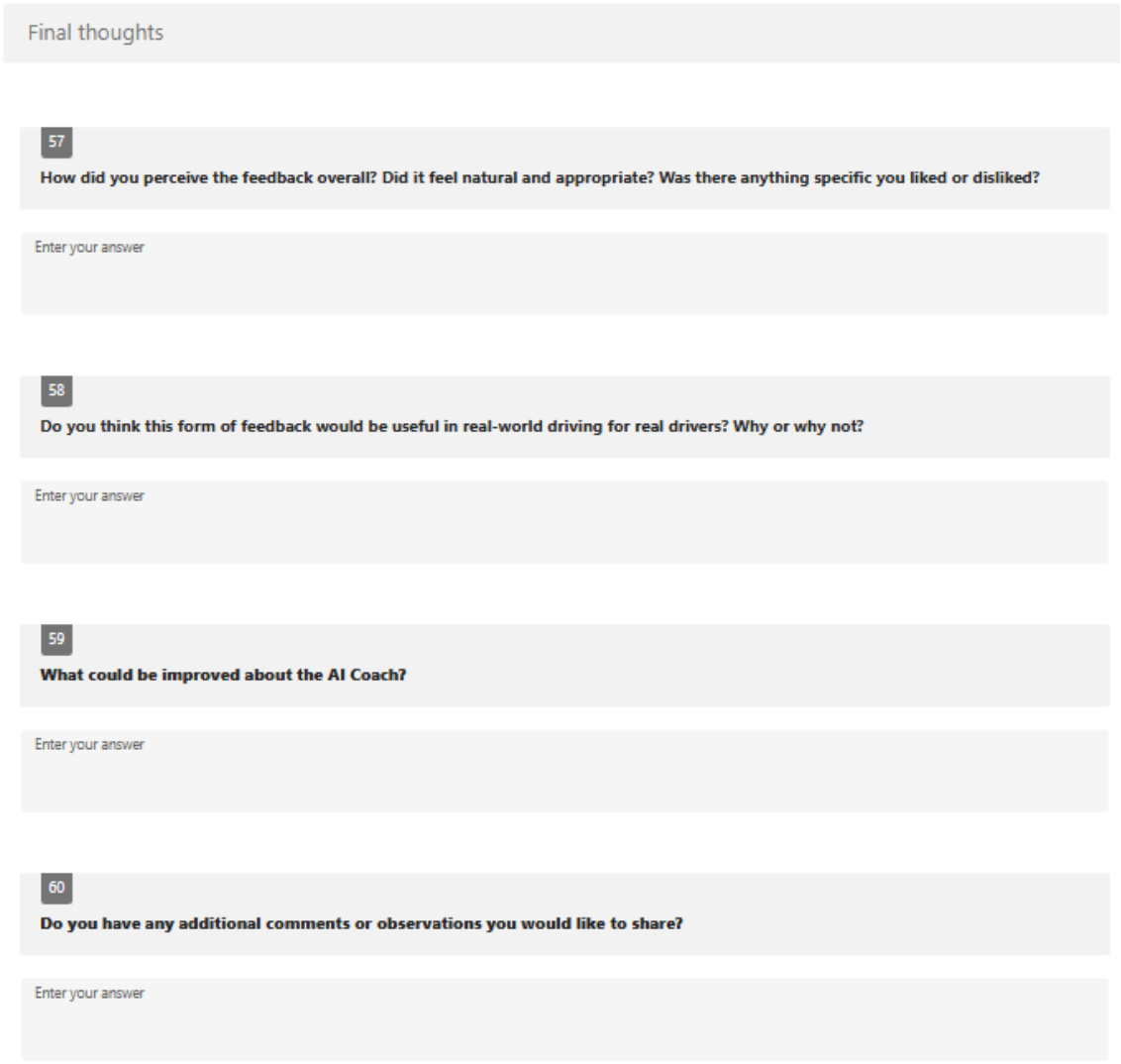


Figure D.5: Open ended free form questions for assessing the overall impression