UNIVERSIDADE PORTUCALENSE INFANTE D. HENRIQUE

Fuel Consumption Through the Lens of Machine Learning

Tiago Pinto

DISSERTAÇÃO

Mestrado em Ciência dos Dados

Supervisor: Bruno Cunha

October 17, 2025

Fuel Consumption Through the Lens of Machine Learning

Tiago Pinto

Mestrado em Ciência dos Dados

October 17, 2025

Abstract

Predicting fuel consumption accurately in the real world is crucial for both individual consumers, fleet directors, and policy makers to reduce costs and environmental impact. This dissertation showcases the creation and evaluation of a platform that estimates the fuel consumption of light vehicles using technical specifications publicly available, as well as user-provided behavioral data, and advanced machine learning techniques.

The system was designed as a decision support system, allowing users to obtain approximate estimates of fuel consumption for vehicles they are considering. Methodologically, the project consolidates diverse data sources, including technical details scraped from the Web, responses to a user survey, and synthetic data to build a robust dataset.

The information gathered was processed in a medallion architecture, in a cloud data warehouse and analyzed using tree-based ensemble models and neural networks integrated through different frameworks. Model performance was then evaluated using standard metrics (MAE, MSE, R2) and the results demonstrate a significant improvement over the fuel consumption test estimates.

Keywords: Machine Learning; Fuel Consumption; XGBoost; Random Forest; Neural Network; Artificial Intelligence; Deep Neural Networks; Ensemble Models

Resumo

Prever com precisão o consumo de combustível real de um determinado veículo é importante tanto para consumidores individuais, diretores de frota e para políticos, com o objetivo de reduzir custos e impactos ambientais causados pela circulação dos mesmos. Esta dissertação apresenta o desenvolvimento e avaliação de uma plataforma que faz estimativas sobre o consumo de combustível de veículos de passageiros, utilizando especificações técnicas disponíveis publicamente, dados comportamentais fornecidos pelos utilizadores e técnicas avançadas de Machine Learning.

Este sistema foi desenhado como uma ferramenta de apoio à decisão, permitindo aos utilizadores obter estimativas aproximadas sobre os valores de consumo de combustível reais para veículos que estejam a analisar. Na sua metodologia, o projeto consolidou diferentes fontes de dados, incluindo detalhes técnicos recolhidos da web, respostas a um inquérito, assim como dados sintéticos, com o objetivo de construir um conjunto de dados robusto.

Os dados foram processados numa arquitetura medalhão na cloud e analisados usando modelos ensemble baseados em árvores e redes neuronais, integrados através de diferentes frameworks. A performance dos modelos foi avaliada com métricas padrão (MAE, MSE, R^2) e os resultados demonstram melhorias significativas face às estimativas obtidas em testes de consumo de combustível.

Keywords: Aprendizagem Automática; Consumo de Combustível; XGBoost; Floresta Aleatória; Rede Neuronal; Inteligência Artificial; Modelos Ensemble; Redes Neuronais Profundas

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Bruno Cunha, for his invaluable guidance, support, and encouragement throughout the course of this dissertation, as well as the challenge to take something dear to my heart further, and strive for greatness. His expertise and insightful feedback were instrumental in shaping this project.

I would also like to thank Universidade Portucalense and its research division REMIT for providing the essential resources that made this project possible.

I deeply appreciate all who took the time to respond to the survey. Your information and feedback on user experience was invaluable to the development and direction of this study.

Special thanks go to all my friends and family, who provided a much needed relief from all the responsibilites, and sharing my frustrations, offering their shoulder as support, or just a beer. All of this was made that much easier due to all your efforts.

Lastly, to the 3 pillars in my life:

To my mom, Sónia, for always believing in me, and helping me realize my potential, since I was a child. A big part of my drive to succeed comes from you, and it shapes a lot of how I approach challenging situations.

To my dad, Hugo, for teaching me strength and for being the reason this project exists, in a more philosophical standpoint. You taught me to love cars, and video games, and mix them both, in work, and in leisure.

And last, but not least, to my girlfriend, Bia. Without you this project would not have been complete. Thank you for living through it with me, celebrating my wins with me, and scratching your head alongside me in my losses, which there were many. If I didn't give up, it's because I knew you were there beside me, cheering me on, even when it was difficult to keep going.

Tiago Pinto

"I am the wisest man alive, for I know one thing, and that is that I know nothing."

Plato, The Republic

Contents

List of Figures

List of Tables

Listings

Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
MAE	Mean Absolute Error
MSE	Mean Squared Error
MAPE	Mean Absolute Percentage Error
R^2	Coefficient of Determination
RQ	Research Questions
OBJ	Objectives
NEDC	New European Driving Cycle
WLTP	Worldwide Harmonised Light Vehicles Test Procedure
SSL	Solid State Logic
TLS	Transport Layer Security
RF	Random Forest
NN	Neural Network
XGBoost	Extreme Gradient Boosting
IEA	International Energy Agency
GDPR	General Data Protection Regulation
ISO	International Organization for Standardization
LB-FT	Pound-Force Foot
NM	Newton Meter
DSS	Decision Support System
IEA	International Energy Agency
UNEP	United Nations Environment Program
SI	International System of Units
IQR	Interquartile Range
TPE	Tree-structured Parzen Estimator

Chapter 1

Introduction

This dissertation focuses on the development of a system that consists of a predictive tool that estimates the real-world fuel consumption of light vehicles based on publicly available car technical data, combined with contextual information provided by users. The system is designed as a pre-purchase Decision Support System (DSS), allowing potential car buyers and current owners to get realistic consumption estimates specific to their expected conditions.

1.1 Context and Motivation

Fuel consumption continues to be one of the most important factors in the transportation industry due to its economic and environmental implications. For individual buyers, it directly affects the total cost of owning a car. For society, it has an important impact on greenhouse gas emissions, air quality, and dependence on fossil fuels. In this context, an there has been an increasing amount of attention dedicated, in the past few years, to understand, model, and estimate accurately vehicle fuel consumption.

Studies have shown that the most important factors influencing fuel consumption usually include the vehicle's weight, aerodynamics, engine type, rolling resistance, and driving behaviour [142, 108, 153]. In a similar way, regulatory entities and researchers have highlighted the role of the route topography, traffic density, and weather conditions as important factors that can affect fuel consumption values [140, 58]. The U.S. Environmental Protection Agency (EPA), 2025 [29] and Fafoutellis, et. al, 2021 [35] have also demonstrated how driving economically can result in substantial fuel savings.

The International Energy Agency (IEA) Fuel Economy in Major Car Markets report serves as a practical reference for tracking how vehicle fuel efficiency changes across different regions, offering a global benchmark for comparison. This report shows that average fuel economy has improved in most major markets over the past two decades, however, it also notes a slowdown in gains and a persistent gap between laboratory results and real world performance. This gap remains even after the move from the New European Driving Cycle (NEDC) to the Worldwide

Harmonised Light Vehicles Test Procedure (WLTP), a protocol intended to mirror everyday driving more closely [76]. This is detailed in Appendix A.

Beyond test cycles, well-to-wheel energy analysis has seen its use increased because it takes into account the full path from fuel production to end use, which provides a more complete view of efficiency [21]. Additionally to this perspective, analysts and agencies now rely on two practical tools to estimate fuel use. Simulation models apply vehicle physics and defined drive cycles to estimate the consumption values under controlled conditions, which is useful for creating policies and certification studies. Data-driven approaches, on the other hand, using Machine Learning (ML) models, for example, learn from large collections of vehicle specifications and observed trips, using sources such as OBD records, GPS data, and road conditions, then return estimates specific to a particulare route, climate, or personal driving style.

However, many of these solutions are closed source, require substantial manual input (such as continuous fuel-up logging by the end-user), or are created for analyzing a car the user already owns rather than providing support on a future purchase. Furthermore, publicly funded initiatives such as the MILE21 EU project have aimed to serve as authoritative references for consumers, but have faced operational and security shortcomings; most notably, the security certificate of their website has not been updated for nearly a year, making it unsafe for general use, given cybersecurity concerns.[85, 31]

This project emerges as a response to these gaps. It proposes an intelligent system capable of estimating the fuel consumption of any potential vehicle purchase. Its aim is to serve as a single source of truth for average fuel consumption across vehicle versions, integrating structured technical specifications with ML to produce accurate and trustworthy predictions. Such a platform could guide purchase decisions and help drivers benchmark their own consumption against realistic averages.

1.2 Problem Statement

Consumers today face a persistent challenge when trying to estimate the real fuel consumption of a car they intend to purchase. Manufacturer-provided standard consumption values, while useful for regulatory compliance and market comparison, are often optimistic and fail to reflect real-world driving patterns due to variations in test procedures, environmental conditions, and user behaviour [86, 77].

Several existing tools attempt to address this gap by providing more realistic figures, but they suffer from major limitations. While such methods can provide accurate records for tracking one's own car, they are of little help to a consumer researching a vehicle they have not yet purchased, due to differences in driving style, experience, or conditions.

Furthermore, these datasets are often unstandardized, where differences in route profiles, driving styles, driver experience, and vehicle maintenance history can skew results, reducing their reliability for comparative analysis [35].

Projects like MILE21 attempted to consolidate both standardized test results and user-contributed data into a centralized platform. However, operational issues, in the form of an expired Solid State Logic (SSL)/Transport Layer Security (TLS) certificate, as of the time of writing, raise important cyber-security risks, making it impossible to recommend for public use [4].

There is, therefore, a clear need for a secure and standardized platform that can provide realistic fuel consumption estimates for vehicles before purchase. The platform should take information from a trusted and normalized dataset, effectively a single source of truth for each vehicle, while combining structured technical specifications, verified observations, and predictive modeling.

This dissertation aims to address that need by proposing the development of ML models capable of estimating potential fuel consumption under typical driving conditions. Using features like weight, power, aerodynamics, and vehicle age, combined with user submitted behavioral information, the models will be implemented and compared using PyTorch and Scikit-learn. The objective is to develop a robust prediction tool that gives consumers the power they need to be able to make informed purchase decisions, while contributing to the broader energy efficiency and emissions reduction goals.

This project builds directly on previous work that was done for the subject Estimating, Detection and Learning II of the Master's in Data Science. That initial study laid important groundwork by developing ML models to predict vehicle fuel consumption using data from sources such as *fueleefficiency.gov*. It involved a full pipeline of data collection, preprocessing, exploratory analysis, feature engineering, and modeling with algorithms including Extreme Gradient Boosing (XGBoost), Random Forest (RF), and Neural Networks (NN).

While the results were promising, the initial project was limited by a small amount of data sources, challenges in prediction accuracy results, and insufficient incorporation of driving contexts that affect fuel consumption. This made it difficult for the models to explain very different fuel consumption figures for the same car. Since completing that project, my goal was to expand and improve this dissertation by applying new techniques and integrating more diverse data sources, inspired by what I have learned since then.

Building on the limitations found, the objective of this continuation is to produce a more complete and reliable system that can give realistic fuel consumption estimates specific to different vehicles and driving conditions, providing consumers with reliable support to make informed decisions. To assess the predictive quality of the developed models and ensure these goals are met, model performance will be evaluated using key metrics, such as Mean Absolute Error (MAE), which measures the average magnitude of prediction errors and offers intuitive error units, Mean Squared Error (MSE), which emphasizes larger errors by squaring deviations and the Coefficient of Determination (R^2) which indicates the proportion of variance explained by the model, thus reflecting its overall fit and reliability.

1.3 Objectives and Research Questions

This section presents the main Research Questions and Objectives that guide the dissertation, setting expectations for the following chapters. The Research Questions are explored in the Literature Review, while the Objectives are addressed during the development and evaluation of the decision support system (DSS). This structure clarifies how each chapter contributes to the study.

1.3.1 Research Questions

This dissertation presents a data-driven solution designed to address the existing gap in fuel consumption reporting to obtain a realistic value based on different and relevant parameters. To guide the development of best practices and ensure methodological rigor, 3 Research Questions (RQ) were devised which collectively establish the foundation and direction of the project:

- **RQ1**: How can publicly available vehicle specifications, combined with contextual and environmental factors, be transformed into a reliable, data-driven prediction of real-world fuel consumption for a vehicle a user is considering purchasing?
- **RQ2**: What ML techniques and data integration strategies offer the best balance between accuracy, scalability, and usability for generating such predictions?
- **RQ3**: How can the tool be designed to support prospective car buyers in making informed decisions by providing realistic, road-relevant estimates rather than idealized laboratory test values, in line with the IEA identified goals to close the lab-to-road consumption gap?

1.3.2 Objectives

Although the main objective of this dissertation is to create a tool that correctly predicts actual fuel consumption in different scenarios, the project also includes additional objectives.

These objectives (OBJ) range from critically examining current state-of-the-art approaches, to implementing advanced analytical methods, and creating a robust methodological framework to support all stages of the solution:

- **OBJ1**: Develop a predictive platform that uses vehicle specifications and contextual conditions to estimate likely real-world fuel consumption, giving prospective buyers and users a practical and trustworthy reference point.
- **OBJ2**: Create a unified “single source of truth” for vehicle fuel consumption by consolidating and normalising data from different sources, including publicly available datasets and user submissions.
- **OBJ3**: Evaluate and compare the predictive performance of different ML models, using PyTorch, Scikit-learn or equivalent tools, to identify the approach offering the optimal trade-off between accuracy, reliability and efficiency.

- **OBJ4**: Examine the limitations of existing tools, including MILE21 and user-based fuel-up logging platforms, identifying how this project can address gaps in reliability, accessibility, and independence from ongoing manual input.
- **OBJ5**: Ensure the proposed solution successfully improves the base project, delivering more accurate results, and better performing models.

1.4 Dissertation Structure

This document is split into five chapters, which contain the following contents:

- **Chapter 1**: This chapter contains the introduction to the dissertation, highlighting the motivations and goals, as well as research questions and ethical considerations.
- **Chapter 2**: Presents the Literature Review, where existing research and publications are analyzed to answer the research questions separately while integrating these findings to form a cohesive understanding of the topic.
- **Chapter 3**: Details the Proposed Solution and Methodology. This chapter highlights the tools and technologies used to build the project, along with decisions and adaptations made during development that shaped the final solution.
- **Chapter 4**: Contains the Results Discussion, where details of the findings from implementing the proposed solution, presenting quantitative and qualitative results are shown. It includes an evaluation of model accuracy, performance metrics, user interaction feedback, comparison with the previous work, as well as two test cases to showcase the results applied to two specific vehicles. Insights drawn from the solution are then analyzed to interpret how well the solution meets the research objectives.
- **Chapter 5**: This chapter summarizes the insights and results obtained from the development, reflecting on the success and limitations of the project. It returns to the RQs, to show how they were addressed, to the OBJs, to validate their completion, and discusses the broader implications for both research and practical purposes. Additionally, it showcases different directions for future research and development, including improvements, scalability challenges, integration with newer technologies, and opportunities for extending the scope to other domains.

1.5 Ethical Considerations & AI

The recent evolution and adoption of Generative AI and GPT-like technologies (ie. Large Language Models (LLM)) in recent years, along with the complexity of web scraping, means there are ethical challenges related to the research and development of this dissertation. Throughout its development, ethical standards have guided every step, from data collection and processing

to analysis and deployment. Particular care was taken to respect data ownership, ensure transparency throughout, and avoid harm to source platforms and stakeholders. Ethical practices and considerations for AI use and web scraping are outlined in this chapter.

1.5.1 Web Scraping

All the data used in this project was obtained exclusively from public information, with extra attention to policies, including adherence to robots.txt guidelines, for example, and compliance with both copyright and privacy laws. Proxy services were used responsibly to minimize impact on server performance, ensuring an ethical and sustainable approach to data scraping. No unauthorized or private information was accessed at any point during development.

1.5.2 Synthetic Data

To address the gaps existent where real-world data was insufficient, synthetic data was generated and integrated with the remaining responses. The origins and limitations of these synthetic datasets are clearly acknowledged, and steps were taken to prevent the introduction or amplification of bias that could distort findings or lead to unfair conclusions.

1.5.3 Data Security and Privacy

All information, including sensitive user-submitted data, is securely stored on trusted cloud platforms (such as Streamlit and Snowflake). These environments follow strict encryption protocols and access controls that meet the relevant data protection standards: the General Data Protection Regulation (GDPR) and the International Organization for Standardization (ISO) 27001.

1.5.4 Purpose and Integrity

The goal of this dissertation is to contribute positively to environmental sustainability and provide users with better Decision Support Systems (DSS). To that end, all assumptions, methods, and limitations are openly documented to maintain scientific honesty and rigor.

1.5.5 Use of AI and Language Models

Artificial intelligence tools, including large language models, were utilized as aids to support the research and as a tool for code documentation. Importantly, this means that these technologies were not depended on to write code, generate original text, or do they represent any dependency in the research and development process.

This emphasizes the ethical standards and sense of social responsibility that have consistently guided this dissertation, particularly within the automotive data science and ML domains.

Chapter 2

Literature Review

To write this chapter, a detailed literature review was written to investigate and address the RQ's presented in the Introduction. The following sections are organized around each one of them, presenting relevant findings from the literature and drawing key conclusions. The review synthesizes key academic and institutional findings, establishing the foundation for the framework presented in the later chapters.

2.1 RQ1: Analysis of Factors Influencing Real-World Fuel Consumption

RQ1: How can publicly available vehicle specifications, combined with contextual and environmental factors, be transformed into a reliable, data-driven prediction of real-world fuel consumption for a vehicle a user is considering purchasing? Real-world fuel consumption results from the interplay between vehicle characteristics, environmental conditions, and driver behavior. Transforming correctly publicly available vehicle specifications and contextual factors into reliable fuel consumption estimations represents a major challenge in automotive research, with significant implications for consumer decision making as well as environmental sustainability.

To address RQ1, the literature review synthesizes recent and historical studies and integrates key insights from the IEA's Fuel Economy in Major Car Markets [58] report, alongside current ML approaches that demonstrate how data-driven methodologies can bridge the gap between official ratings and real-world performance.

The challenge lies not only in the inherent complexity of fuel consumption estimation but also in the gap between laboratory-tested values and actual performance. The IEA [58] indicates that this gap has widened over the past decade, with actual fuel consumption values reaching almost 50% above tested values in some markets. Other studies show that well-trained machine learning models using detailed datasets that combine vehicle characteristics, driving behavior, and environmental data can predict fuel use with high accuracy, often achieving R^2 values exceeding 0.95 in extensive trials [51, 108].

2.1.1 Vehicle-Related Factors

Vehicle-related factors such as engine size, power, weight, body design, and transmission type represent key information to estimate fuel consumption values. These characteristics affect how efficiently a vehicle converts energy into motion, contributing to the variation observed in real-world fuel economy across different models and categories.

2.1.1.1 Engine Power, Displacement, and Efficiency Technologies

Engine displacement and power continue to be strong indicators of fuel consumption values in global markets. According to the IEA [58], efficiency improvements in major regions have largely stemmed from engine downsizing, turbocharging, and weight reduction, even as safety and comfort features have added to overall vehicle mass. Hamed, M. A. et al. [51] found that parameters obtained from On-Board Diagnostics (OBD) data, such as Mass Air Flow, engine speed (RPM), and Throttle Position Sensor readings, are important inputs for ML models, which reached R^2 values of up to 0.97.

Modern ML approaches have identified engine displacement and cylinder count as among the most impactful variables to fuel consumption values. Sonkar, S., Danwani, S., et al. in 2021 [122] found that RF models incorporating these parameters achieved exceptional performance with R^2 scores of 0.91, MAE of 1.85, and MSE of 2.42. The author's feature importance analysis revealed that engine displacement, in combination with transmission type and fuel type provides the strongest predictive power for real-world fuel consumption.

Although hybrid and electric vehicles are increasing their market share, internal combustion engines still dominate global sales. As a result, the efficiency trends of these engines largely shape the overall fuel consumption trend direction [58]. Advanced technologies like turbochargers and variable valve timing represent significant opportunities to improve prediction accuracy when incorporated into machine learning models using publicly available specifications.

2.1.1.2 Vehicle Mass, Body Design, and Aerodynamic Considerations

There is a consistent linear connection between vehicle mass and fuel consumption across global fleets. In 2017, the IEA [59] reported that the growing popularity of SUVs and light trucks has stalled or even reversed improvements in the overall efficiency of vehicles. Figure 2.1 shows how the average fuel consumption of light trucks and SUVs has diverged considerably from that of passenger cars, showcasing that the increasing SUV sales have reduced overall efficiency by 10 to 12%.

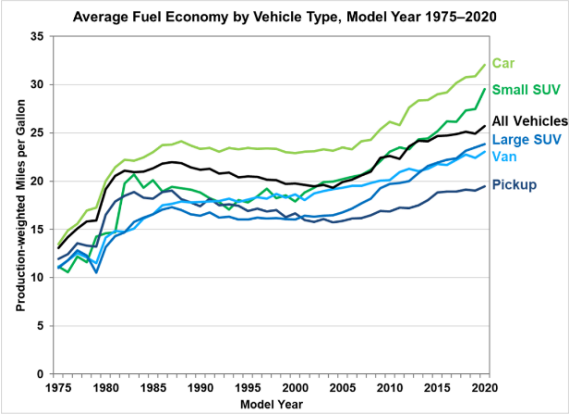

Figure 2.1: Average Fuel Economy trends from 1975 to 2020 by vehicle type [141]

The market share of sport-utility vehicles and pick-ups grew by 11 percentage points since 2014, representing nearly 40% of the global light-duty vehicle market by 2017, with particularly high rates in North Cyprus and China, as shown in Figure 2.2 [46].

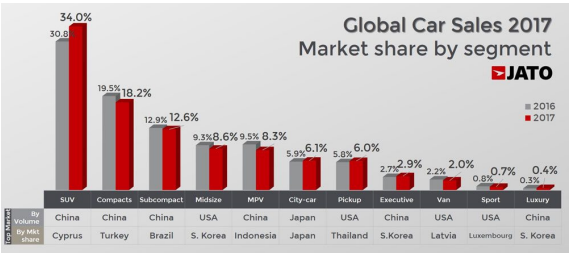

Figure 2.2: Car Sales by segment [46]

ML approaches have successfully quantified these relationships with high precision. In 2024, Manjunath and Ashok Kumar [138] reported strong linear regression results, with R^2 values reaching 99.63% when vehicle mass was included alongside other specifications. Their approach shows that integrating vehicle weight, body style, and dimensional features into prediction models can substantially improve accuracy without compromising efficiency.

Aerodynamic drag, rolling resistance, and drivetrain friction represent secondary but significant contributors to fuel economy variations. Research indicates that these factors, when combined with vehicle mass specifications, can account for substantial portions of real-world fuel consumption variance. One of the main challenges for data-driven methods is obtaining aerodynamic specifications, which are often proprietary. Turning these specifications from publicly available sources into useful prediction features can be difficult but is crucial to improve model accuracy.

2.1.1.3 Powertrain Technology and Electrification Integration

Hybrid and plug-in hybrid drivetrains can deliver significant real-world fuel savings, especially when drivers' charging habits match the vehicles' intended use. The IEA's 2019 report [58] notes

that while electric vehicle purchases are increasing, the impact on car consumption values is limited in most markets. Plug-in hybrid is largely dependent on how much drivers use the car's electric motors, leading, often, to much higher consumption and emissons values than reported by laboratory results.

Yoo, S. et al. [108] developed detailed frameworks using Extra Trees Regressor and Random Forest models to capture the complex, nonlinear relationships between electrified powertrain features and fuel efficiency. They applied SHAP [111] and LIME [89] techniques to enhance model interpretability, offering important insights into how various powertrain configurations influence real-world performance under different operating conditions.

The electrification of light vehicles is fundamental to reach the goal of fuel economy improvements, especially as diesel market shares decline in major European markets. Japan experienced the largest fuel economy improvements due to having the highest global market share for hybrid vehicles, followed by the United States with diverse electrified vehicle types, and China with rapidly growing battery electric and plug-in hybrid markets [58].

2.1.2 Environmental Factors and Contextual Dependencies

In addition to the technical specifications of each vehicle, a range of environmental factors contribute significantly to fuel consumption figures. Factors such as ambient temperature, road quality, weather conditions or topography can each have a distinct impact.

2.1.2.1 Ambient Temperature, Weather, and Regional Variations

IEA's report [58] indicates that there's a quadratic relationship between ambient temperature and fuel economy performance, with cold starts and car heating use increasing consumption in low temperatures, and air-conditioning loads doing the same in hot climates. Real world testing documentation (e.g., Hyundai i30 trials) captures these variations, as illustrated in Figure 2.3, highlighting how environmental conditions can produce a 9 to 20% underestimation of consumption if ignored [8].

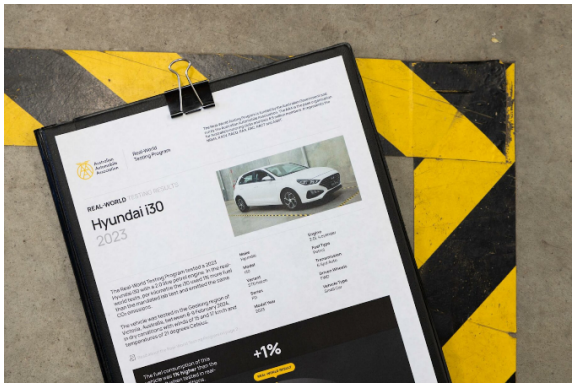

Figure 2.3: I30 test results [8]

There is a significant difference between laboratory-tested fuel consumption and real-world performance. As shown in Figure 2.4, on-road fuel consumption can be up to 50% higher than official figures, with most vehicles showing higher actual usage than their lab ratings.

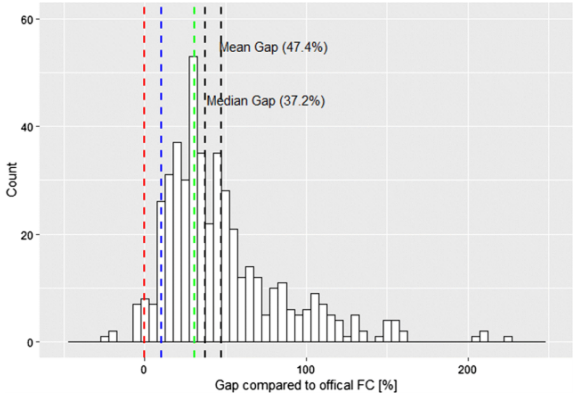

Figure 2.4: Histogram showcasing the difference between reported and actual values [99]

Advanced ML models have successfully incorporated these environmental dependencies. Research indicates that ignoring temperature variations can lead to fuel consumption estimates higher than advertised values by 6 to 11%, with seasonal effects exceeding 10% in several markets. Including weather and climate data in prediction models is therefore essential for producing accurate, location-specific fuel consumption estimates.

A recent study by Bagheri, E. et al. [9] examining driving behavior in Tehran showed that environmental conditions significantly influence the relationship between driving characteristic parameters and fuel consumption throughout different vehicle types. Their methodology utilizing K-means Clustering and Principal Component Analysis algorithms reveals driving patterns that more accurately reflect real-world environmental conditions.

2.1.2.2 Road Profile, Urbanization, and Surface Conditions

According to IEA, [58], road type distribution and urbanization patterns significantly affect national fuel economy averages. Urban stop-and-go traffic, road slopes, and surface conditions have a noticeable impact on fuel consumption. Research shows that these factors generally lower fuel efficiency compared to consistent highway driving.

Hanzl, J. et al. [52] studied the effect of road elevation on fuel consumption during acceleration, finding that going downhill reduced fuel use by 31.4% compared to flat surfaces. Their results suggest practical benefits for infrastructure design, such as elevated intersections or parking areas, which could save about 100 liters of fuel per week for every 100,000 vehicles passing through modified roundabouts. Adding topographical and road surface information into machine learning models offers a promising way to boost prediction accuracy [30]. Advances in GPS and mapping technologies provide detailed route data that, when combined with vehicle specifics, can improve consumption estimates for particular driving conditions.

2.1.2.3 Traffic Patterns, Utilization, and Geographic Diversity

In 2019, the IEA [58] reported that variations within a country, such as speed distributions and traffic congestion, can lead to fuel economy differences exceeding 10%. This indicates the importance of including geographic and traffic variables in predictive models to create reliable, location-specific fuel consumption estimates.

ML solutions have successfully showcased these dependencies by integrating real-world driving cycle data. Research using nineteen distinct driving cycles generated through K-means clustering shows that traffic-specific parameters significantly influence model performance across different vehicle configurations and powertrain types [9].

A big difficulty in using publicly available data is obtaining detailed traffic pattern databases and converting this information into effective prediction features. Recent methods that leverage telematics data, GPS tracking, and mobile device information offer more detailed views of real-world traffic, which can enhance prediction accuracy.

2.1.3 Driver-Related Factors and Behavioral Patterns

Driver-related factors, such as driving style, acceleration habits, auxilliary system use, and charging frequency strongly influence fuel consumption. Choices like aggressive acceleration, excessive speeding, and prolonged idling can significantly increase fuel use. Understanding these patterns is essential, as even the same vehicle can achieve markedly different fuel economy depending on how it is driven.

2.1.3.1 Driving Behavior and Style Dependencies

Driving style parameters, including acceleration patterns, speed variance, and gear use, are dominant drivers of variability in real-world fuel consumption [55]. Eco-driving programs in Japan and Europe have demonstrated reductions of 15 to 18% [58]. ML models trained on real-world consumption data, exemplified by performance metrics in Figure 2.5, successfully predict these behavioral impacts with R^2 values exceeding 0.95 [51].

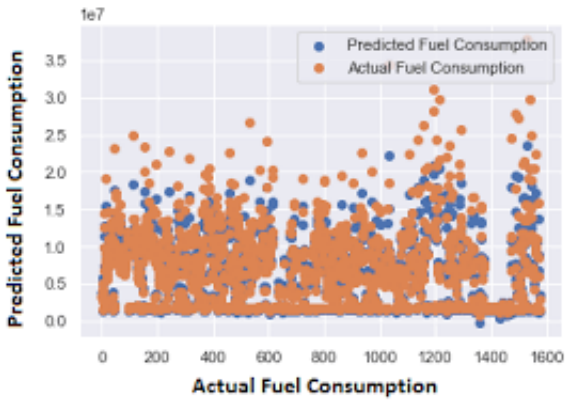

Figure 2.5: ML Performance using Real World Consumption Data [51]

Research by Yang, Z. et al. in 2024 [150] and Ashqar, H. I. et al. in 2024 [7] confirms these findings across different vehicle types and operational conditions.

ML models incorporating driving behavior parameters have achieved exceptional prediction accuracy. High performing multilayer perceptron architectures that integrate specification data with environmental and usage patterns have achieved R^2 values surpassing 0.95 in large fleet trials, demonstrating the critical importance of behavioral factors in fuel consumption prediction (research findings from analysis).

The challenge for consumer-oriented prediction systems lies in estimating individual driving behavior patterns from limited input data. Modern approaches utilize standardized driving profiles or allow users to specify their typical driving characteristics to personalize consumption predictions.

2.1.3.2 Trip Length, Start Conditions, and Auxiliary System Loads

Short trips and frequent cold starts disproportionately impact efficiency across all vehicle types. The IEA's [58] findings confirm that auxiliary loads such as heating, air conditioning, and more demanding infotainment systems account for increased total energy requirements, particularly in cars that are only partly electrified (Hybrids and Plug-in Hybrids). Research by Morawska, L. et al. [91] shows in detail these effects across different vehicle configurations.

Recent studies indicate that both trip length and the frequency of cold-starts play important roles in how accurately fuel consumption can be estimated by ML, meaning that models including these driving patterns perform better when tested outside controlled environments, compared to those relying only on vehicle specifications. Incorporating auxiliary load specs and usage habits effectively into ML models requires robust feature engineering to properly showcase and reflect the complex relationships between vehicle mechanisms, environmental factors, and driver behaviors [108].

2.1.3.3 Charging and Refueling Patterns for Electrified Vehicles

The IEA's [58] data shows that Plug-in Hybrids are often driven less in electric-only mode than assumed in laboratory tests, which leads to much higher than expected real-world fuel consumption and emissions values. This solidifies the critical role of charging habits in determining actual fuel economy outcomes for electrified vehicles.

Research on real-world plug-in hybrid usage patterns shows that actual fuel consumption and tailpipe CO_2 emissions are approximately two to four times higher than type-approval values [61]. The real-world share of electric driving for PHEVs averages significantly below regulatory assumptions, resulting in much lower emission savings compared to conventional vehicles than initially expected, shown in Figure 2.6.

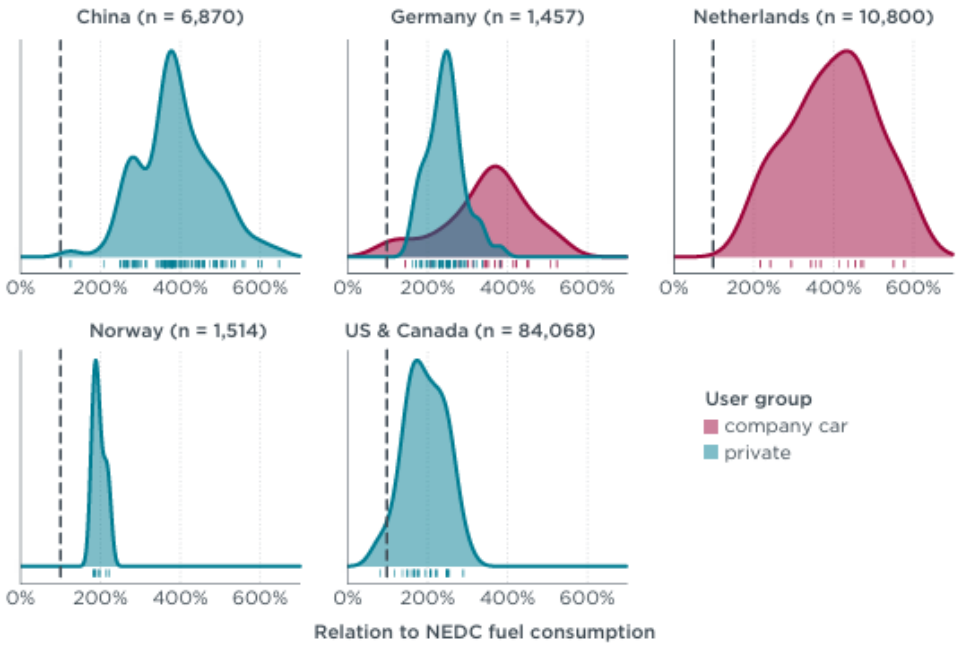

Figure 2.6: Comparison of Test Fuel Consumption and Actual Results for PHEV [61]

ML approaches must account for these behavioral dependencies when predicting fuel consumption for electrified vehicles. Models that incorporate charging frequency, trip length distribution, and regional charging infrastructure availability show significantly improved prediction accuracy compared to approaches relying solely on official specifications.

2.1.4 Summary

Table 2.1 presents a summary outlining the key factors affecting real-world vehicle fuel consumption, categorized by factor type, their primary drivers, quantified impacts on consumption, and key references supporting these findings, with a more in-depth overview present in Appendix B.

Factor Type	Primary Factors	Quantified Impact	References
Vehicle	Mass, engine power/displacement, technology trends	+7 to 17% per displacement/power increase; SUV adoption worsens avg. economy by 10 to 12%	[33, 63, 94]
Environmental	Temperature, road profile, surface conditions	6 to 11% underestimation if ignored; >10% seasonal swing	[33, 98]
Driver	Acceleration/braking style, trip length, auxiliary loads	10 to 20% variability; 15 to 18% reductions possible via eco-driving	[94, 16, 7]

Table 2.1: Factors influencing vehicle fuel consumption and their quantified impacts

Real-world fuel consumption originates from the combined effects of vehicle specifications, environmental conditions and driver behavior.

Vehicle details such as engine size, power, and weight strongly influence efficiency, with technological gains often offset by the growing share of heavier SUVs and light trucks (showing +10 to 12% average consumption). ML models show high predictive value from these parameters, but for electrified vehicles, incorporating real world charging habits is essential.

Environmental influences, including temperature, road profile, traffic, and geography, can shift fuel consumption by 4 to 15% seasonally. Integrating local climate, route, and topographic data significantly improves the accuracy of prediction [43].

Driver behavior, through acceleration style, travel length, and auxiliary loads, accounts for 10 to 20% variation in fuel consumption values. Eco driving can cut fuel use by up to 18%, while plug-in hybrids often record 2 to 4× higher fuel use than test values due to low electric driving only [136].

Overall, evidence supports a three-domain framework (vehicle-environment-driver) enriched by high resolution data. ML approaches that integrate these domains can achieve $R^2 > 0.95$, enabling accurate, user specific predictions that support better consumer choices, policy design, and environmental goals.

2.2 RQ2: ML Techniques and Data Integration Strategies for Predicting Real-World Automotive Fuel Consumption

RQ2: Which ML techniques and data integration strategies offer the best balance between accuracy, scalability, and usability for generating real-world vehicle fuel consumption predictions? The automotive industry has started to use ML to model and estimate fuel consumption values based on different data sources, from Engine Control Unit (ECU) data streams to environmental and driving behavior data. ML techniques (such as Gradient Boosting (notably XGBoost), RFs, and NNs) have shown strong ability to recognize complex patterns in fuel use that change over time and do not follow simple linear trends, and integrating these with data pipelines from sensors, for example, is important to achieve scalable and interpretable models capable of being applied in production environments.

2.2.1 The Models

This section presents the three machine learning models selected for investigation in this study: NNs, RFs, and XGBoost. These methods have been widely applied in vehicle fuel consumption prediction research due to their complementary strengths in modeling complex data relationships.

By comparing these models, the goal is to identify the best balance among accuracy, scalability, and practical usability for real-world fuel consumption prediction.

2.2.1.1 Gradient Boosting and XGBoost

XGBoost (Extreme Gradient Boosting) is an advanced implementation of gradient boosted decision trees first introduced by Tianqi Chen and Carlos Guestrin in 2016 [18].It has quickly become

one of the most popular machine learning algorithms due to its remarkable predictive accuracy, scalability, and flexibility, and has been widely adopted across domains, including automotive predictive analytics.

Gradient boosting itself is an ensemble technique that builds models sequentially by correcting the residual errors of prior models using gradient descent optimization [39]. XGBoost enhances traditional gradient boosting by incorporating system and algorithmic optimizations such as parallel processing, tree pruning, handling of missing data, and regularization, which reduce overfitting and improve generalization [18, 71]. Its overarching architecture is in Figure 2.7.

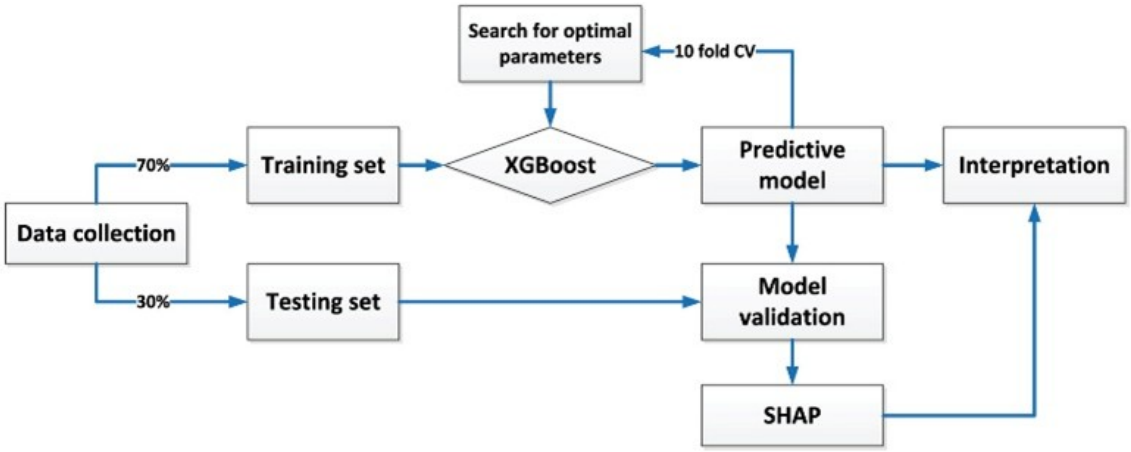

Figure 2.7: XGBoost Pipeline [37]

The core workflow of XGBoost involves iteratively training a series of simple decision trees, each focusing on modeling the residuals of its predecessors to minimize a defined loss function. The final prediction is the weighted sum of the outputs of these individual trees.

XGBoost is designed to speed up tree construction by running tasks concurrently and can take advantage of GPU acceleration, making it very efficient when working with large datasets [149]. This efficiency has been accelerated by recent advances in AI and ML hardware, especially Graphics Processing Units (GPUs). In addition to performance, understanding how the model makes decisions remains important. Tools like SHAP (SHapley Additive exPlanations) offer a consistent way to attribute the influence of each feature, which helps understand the impact of factors like engine load, vehicle speed, throttle position, and ambient temperature on the estimated fuel consumption. [82].

In the automotive industry, XGBoost has been extensively applied to estimate fuel consumption with high accuracy. Wang, G. in 2023 [147] reported R^2 values exceeding 0.95 in certain telematics-based analysis, showing near real-time predictive capability. Additionally, in applications involving light-duty trucks, XGBoost has been combined with optimization algorithms and deep learning frameworks like Echo State Networks to capture complex temporal dynamics, surpassing more conventional models in fuel consumption prediction accuracy [147].

This adaptability showcases XGBoost's capacity to handle diverse vehicle types and operational complexity across different data environments.

2.2.1.2 Random Forests

RF models consist in a robust ensemble learning method developed by Breiman in 2001 [12] that combines multiple decision trees to improve predictive accuracy and control overfitting. The algorithm generates a multitude of trees during training, each built from a different bootstrap sample of the data and considering random subsets of features when splitting nodes. Through this aggregation, RFs achieve noise resilience and enhanced generalization capabilities, as can be seen in Figure 2.8 [12, 80].

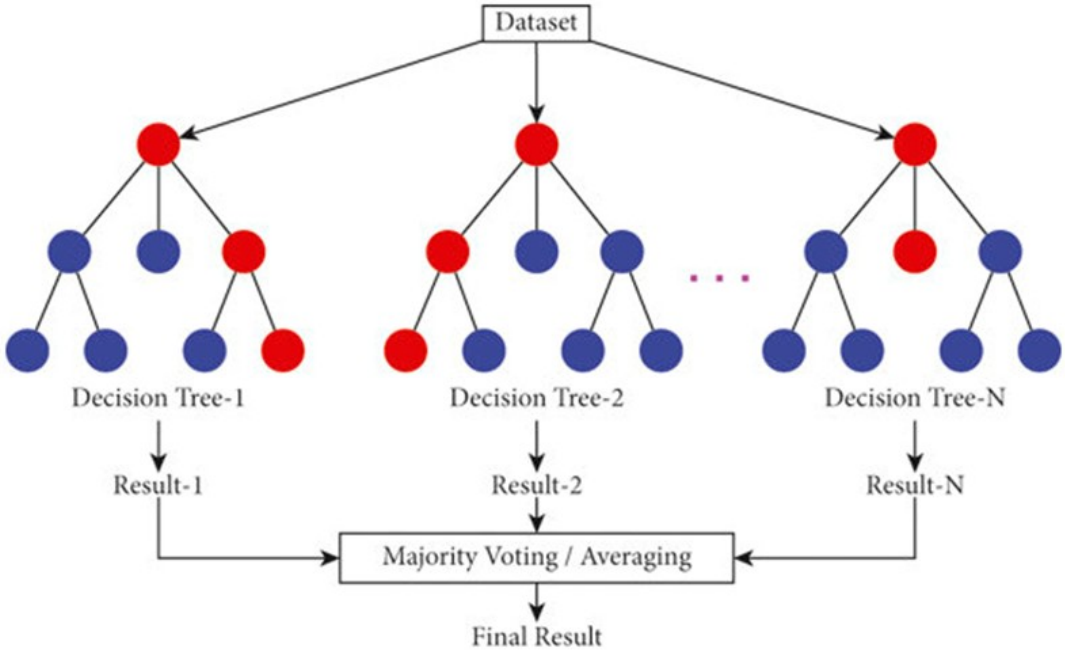

Figure 2.8: RF Model Architecture [157]

In the automotive domain, RFs have been widely applied for fuel consumption prediction, demonstrating reliable performance with reported values ranging from 0.85 to 0.91 in real-world datasets. For instance, Sonkar, S., Danwani, S., et al. [122] validated RF models on vehicle usage data, showing that engine displacement, cylinder count, and transmission type are key drivers of predictive accuracy. Their study further evidenced RF's superiority over Decision Tree and Linear Regression models, making it a practical choice when moderate to high accuracy with interpretability and robustness is needed for deployment. These differences can be seen in Figure 2.9.

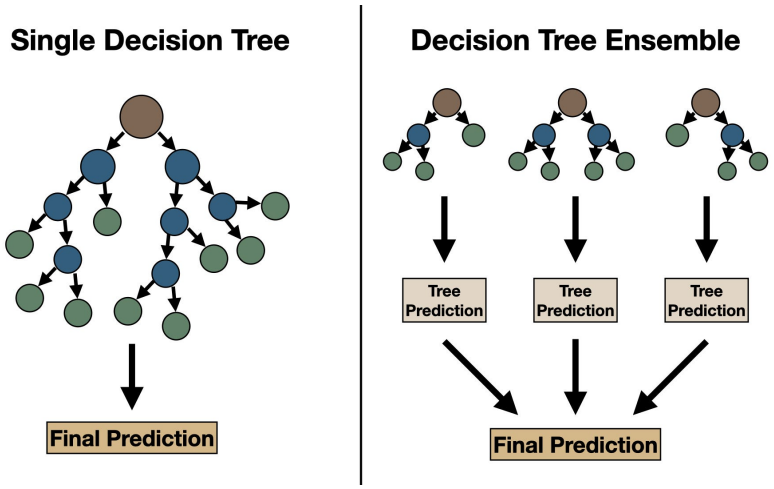

Figure 2.9: RF vs. Decision Trees [144]

An additional advantage of RFs lies in their built-in variable importance measures, which provide interpretable insights into feature contributions. This transparency is particularly valuable in automotive applications where understanding the impact of vehicle attributes, such as fuel type, engine characteristics, and emission ratings, on fuel consumption is crucial for both developers and policy stakeholders [108].

RF's balance of predictive power, interpretability, and robustness has made them a foundation for numerous real-time fuel consumption estimation systems, sometimes augmented with OBD data and driver behavior metrics to further improve accuracy [1].

2.2.1.3 Neural Networks: MLP, LSTM, and Hybrid Architectures

NNs offer powerful capabilities to model the complex, nonlinear relationships present in predicting real-world fuel consumption. For this dissertation, focused on accurately predicting fuel consumption across different scenarios and conditions, NNs provide a flexible framework to integrate multiple dynamic input streams. Three approaches are used widely, which are:

- Multi-Layer Perceptrons (MLPs) are feedforward NNs composed of an input layer, one or more fully connected hidden layers with nonlinear activations, and an output layer, typically trained via backpropagation to capture complex nonlinear relationships in tabular data. MLPs form the foundational architecture, effectively processing static vehicle specifications such as engine size, weight, and transmission type, as well as categorical driving behavior attributes like style classifications (e.g., highway, city, mixed). However, MLPs lack inherent temporal modeling capabilities, limiting their suitability for sequential driving data captured in telematics [154, 110].

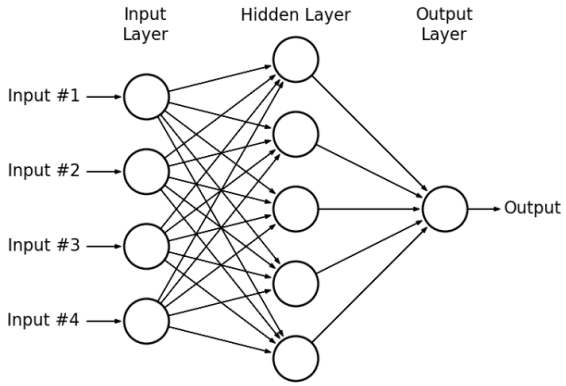

Figure 2.10: A multilayer structure of a NN [69]

- Long Short-Term Memory (LSTM) networks, pioneered by Hochreiter and Schmidhuber in 1997 [56] to overcome vanishing gradient issues in recurrent NNs, use memory cells with input, forget, and output gates to regulate information flow and preserve long-range temporal dependencies. They are capable of modeling time-series data such as acceleration patterns and speed fluctuations. Although they are ideally suited for incorporating continuous ECU signals and high-frequency telematics, practical constraints in this dissertation prevented the use of such data. Nevertheless, previous studies demonstrate LSTM's superior predictive accuracy and temporal modeling capabilities, which could enrich future model iterations when such data becomes available [147, 79].

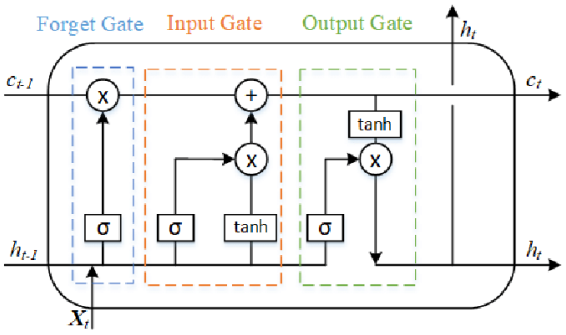

Figure 2.11: LSTM Cell Architecture [87]

- Hybrid CNN-LSTM architectures that combine convolutional feature extraction with sequential temporal learning have demonstrated exceptional R^2>0.95 in fleet telematics datasets. In comparison, literature on fuel consumption prediction indicates that models incorporating only vehicle specifications or simpler ML architectures typically achieve R^2 values between 0.75 and 0.92, highlighting the advantage of hybrid approaches for capturing both spatial and temporal patterns in real-world driving data. While not implemented here due to data limitations, they represent promising avenues for advanced, real-time fuel consumption prediction that include dynamic environmental and vehicle data [147].

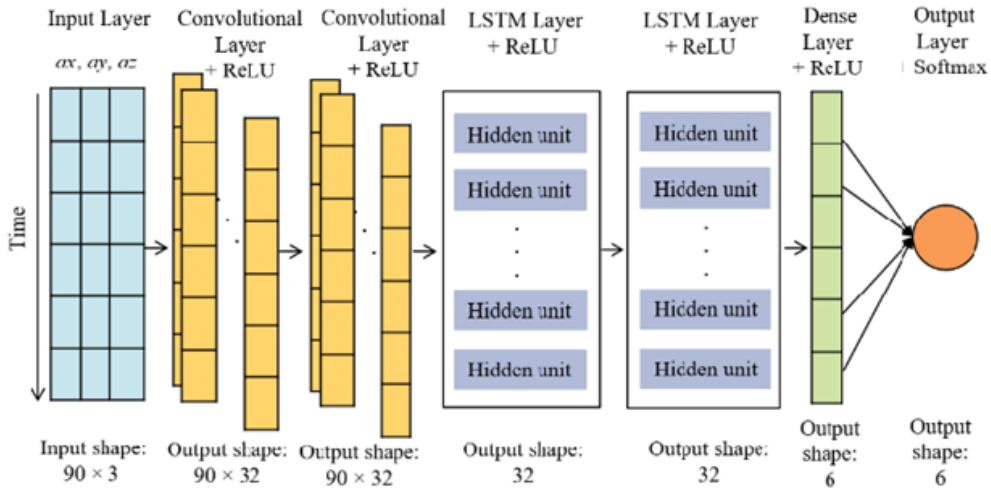

Figure 2.12: Hybrid NN - CNN + LSTM Layers [88]

Training those deep neural models requires extensive computational capacity and a rich set of annotated datasets, a challenge addressed here through cloud-based processing pipelines integrated with high-resolution vehicle and driver data. Although interpretability remains a concern, adopting explainability methods such as SHAP and LIME enables quantification of feature influences, critical for model trust and insights within automotive stakeholders [82].

This integration of neural architectures forms an important component of this dissertation's broader objective: to provide accurate, context-aware, and user-tailored predictions of fuel consumption combining vehicle specs and driver behavior in a unified modeling framework.

2.2.2 Data Integration Strategies

The superior performance of ML models depends on effective data integration, which combines multiple data sources and processing techniques.

High speed ECU data streams capture detailed engine dynamics and auxiliary system statuses, while environmental inputs, such as ambient temperature, road profile and traffic conditions collected via IoT sensors improve the accuracy of predictions. Cloud-based pipelines then process the inputs, allowing for near real-time estimations for fuel consumption values, for a given vehicle.

At the core of this workflow is a reliable preprocessing pipeline including noise filtering, feature selection and feature segmentation, serving as the base for an accurate and sacable model. Studies demonstrate that integrating vehicle specifications with these external contextual factors is essential for developing scalable, accurate and user tailored fuel consumption prediction systems [139, 151].

2.2.3 Summary

Table 2.2 contains the different ML models that were researched summarized, along with their respective R^2 values, and information on scalability, interpretability, and use cases:

Technique	R^2	Scalability	Interpretability	Use Cases
XGBoost	0.95 to 0.99+	Very High (Parallel/GPU)	High (via SHAP)	Leading performance; suited for large heterogeneous vehicle datasets
RF	0.85 to 0.91	High (Parallel)	Moderate	Robust, interpretable baseline; useful for fleet monitoring and prototyping
NNs (LSTM, Hybrid)	0.90 to 0.97+	High (Compute-Intensive)	Low–Moderate (Explainability)	Best at temporal modeling; requires large data and interpretable frameworks

Table 2.2: Comparison of ML techniques for vehicle fuel consumption prediction

Research specific to the automotive industry confirms that XGBoost currently provides the most effective balance between accuracy, scalability, and interpretability. RF models models remain reliable and produce strong results, particularly in conditions where data quality is uneven or processing power is limited, though their accuracy tends to be slightly lower. NN's, particularly LSTM and hybrid architectures, excel in capturing temporal driving patterns and dynamic behavior but require substantial data, extensive processing power, and additional tools to improve interpretability [148].

Emerging directions favor hybrid ensemble models combining the strengths of tree-based and neural methods integrated into scalable cloud pipelines with transparent interpretability systems, crucial for commercial deployments and policy applications aiming to reduce automotive fuel consumption effectively.

2.3 RQ3: Designing Tools to Provide Road-Relevant Fuel Consumption Estimates for Prospective Car Buyers

RQ3: How can the tool be designed to support prospective car buyers in making informed decisions by providing realistic, road-relevant estimates rather than idealised laboratory test values, in line with IEA-identified goals to close the lab-to-road consumption gap?

The widely documented "lab-to-road fuel consumption gap" remains one of the most significant barriers to informed decision-making by prospective car buyers [58, 137]. Standardized laboratory test cycles, such as WLTP, NEDC, or EPA methods are designed for regulatory comparability rather than replicating complex, real-world driving conditions. The gap, ranging from 10% to over 40%, as can be seen in Figure -, originates from varying driver behaviors, traffic congestion, terrain, environmental conditions, and vehicle loading [137, 129]. Closing this gap is critical for transparency, consumer trust and, ultimately, meeting global energy and emissions targets [58].

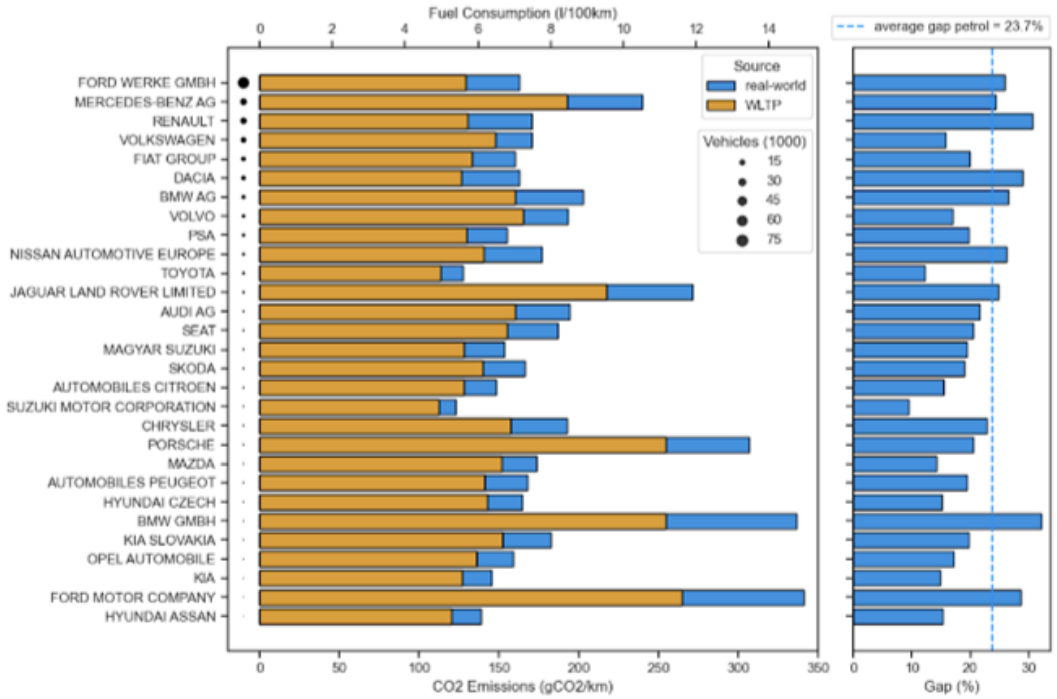

Figure 2.13: Real World vs Test Data for Fuel Consumption and CO2 Emission [32]

This section reviews the core elements and methodologies necessary to design a predictive tool that delivers realistic and personalized fuel consumption estimates, empowering buyers to make sound purchasing decisions aligned with the real demands of their daily driving conditions.

2.3.1 Understanding the Lab-to-Road Gap

Numerous studies converge on the conclusion that official fuel consumption values do not reliably reflect actual driving outcomes [40, 137]. The IEA highlights that the sources of discrepancy include:

- Driving Style and Traffic: Aggressive acceleration, braking, idling during congestion, and stop-start city driving substantially increase consumption [153, 7].

- Environmental Conditions: Weather extremes affect engine efficiency and auxiliary loads (for example, heating / cooling), significantly changing fuel economy [109].

- Vehicle Utilization: Differences in payload, tire conditions, maintenance, and drivetrain optimization also contribute [40].

This variability underscores the inadequacy of single, standardized consumption values for individual buyers seeking personalized estimates.

2.3.2 Elements of a Realistic Fuel Consumption Estimation Tool

2.3.2.1 Multi-Source, Real-World Data Collection

The broad consensus in the literature advocates the building of predictive tools based on diverse and real-world data sources beyond lab values. Incorporating crowd-sourced user submissions, telematics data, and environmental sensors ensures the model sees reality, not just simulations [7, 129].

Tools that harness public repositories of user-reported consumption (Spritmonitor, Fuelly) demonstrate higher predictive accuracy by anchoring predictions to real driving data [64, 137]. There is clear evidence that models trained on real-world user data predict fuel consumption more accurately than approaches relying on laboratory test data, as shown in ICCT's report [137] in 2019.

2.3.2.2 Personalization Through Behavioral and Contextual Inputs

Fuel consumption is highly sensitive to factors unique to each driver and their environment. Research suggests that models accounting for variables such as daily driving context (city, highway, mixed); driving style intensity (calm, average, sporty); ambient climate or road conditions; and vehicle load and usage patterns can significantly improve estimate accuracy [9, 150]. Figure 2.14 shows that tools utilizing these inputs retrieved from, for example, user forms, for personalized estimations achieve superior performance relative to models which only utilize generic manufacturer values.

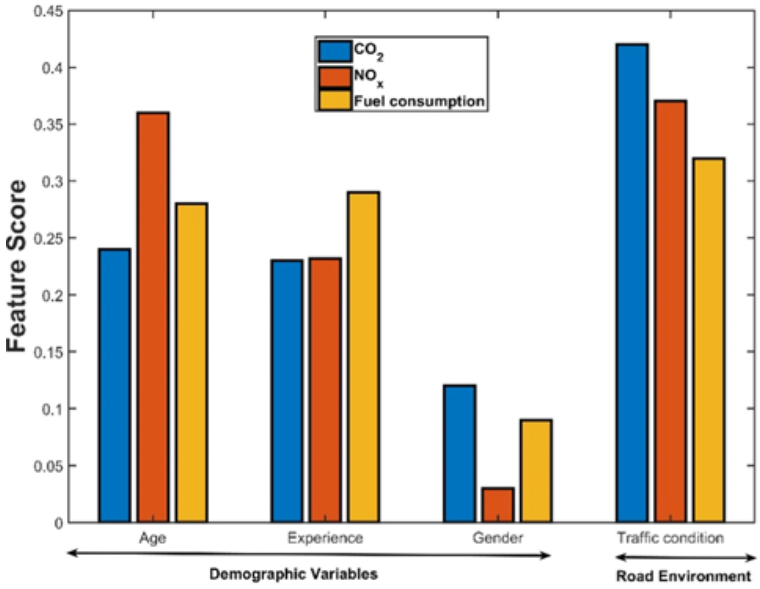

Figure 2.14: How driver specific variables impact fuel consumption values [120]

2.3.2.3 Integration of Advanced ML Models

Powerful ML models, including XGBoost, RFs, and NNs, have been demonstrated to capture non-linear interactions and temporal dynamics that influence consumption, often producing superior predictive performance over simplistic algorithms [147, 122, 108].

Studies emphasize combining predictive accuracy from gradient boosting and deep learning techniques and interpretability via tree-based models and explainability tools such as SHAP and LIME, providing users understandable insights into estimate drivers [73].

Such hybrid modeling enables transparent, precise predictions that are essential for consumer confidence.

2.3.2.4 User Experience and Transparent Result Presentation

Current research consistently shows the importance of interfaces that are both interactive and easy to understand. Dashboards built with modern frameworks (such as Streamlit) improve usability by allowing users to explore vehicle specifications and driving habits directly within the tool. They enable the comparison of different driving scenarios in real time and provide clear visual explanations of how key variables influence fuel consumption [146, 48]. This shift from static laboratory-style data toward dynamic, user-focused platforms contributes to greater engagement and creates trust in the predictive models.

2.3.2.5 Continuous Learning via Feedback and Data Enrichment

The development and enhancement of the system, in order to keep its relevance, requires constant incorporation of user-submitted consumption data into ongoing model training and validation cycles [9]. This feedback loop allows adaptive correction of model bias and adjustment to new vehicle models and technologies.

2.3.3 Summary

In Table 2.3, the best practices that were derived from analyzed literature are presented, along with their rationale, and literature that evidences such best practices:

Element	**Rationale**	**References**
Diverse real-world datasets	Anchors predictions in reality, reducing lab bias	[108, 152]
Behavioral and environmental inputs	Captures key consumption determinants beyond specs	[142, 65, 68]
Hybrid ML models	Balances accuracy and user trust by revealing "why" behind numbers	[41, 118]
Interactive, transparent UI	Enhances decision-making with scenario testing and clear visuals	[143, 118]
Continuous model updating	Maintains reliability across evolving fleets and driving patterns	[40, 108]

Table 2.3: Key elements for reliable and interpretable fuel consumption modeling with literature support

The literature reveals a clear pathway to build tools that deliver prospectively accurate, road-relevant fuel consumption estimates. By leveraging diverse real-world data, integrating user-specific behavioral inputs, employing advanced but interpretable ML algorithms, and fostering interactive transparency, designers can finally help buyers close the frustrating lab-to-road gap. This approach supports the IEA's global energy and climate goals and empowers consumers with the truthful, personalized insights they deserve.

2.4 Relation to Previous Work and Project Development

This dissertation builds on prior research conducted for the course Estimating, Detection and Learning II, as documented in the earlier report [102]. This initial study laid the groundwork by developing foundational ML models for vehicle fuel consumption prediction, relying mainly on data sourced from fuelefficiency.gov and focusing on key features such as engine size, vehicle weight, and traction. The methodology contained data collection, preprocessing, exploratory data analysis, feature engineering, model construction, and evaluation.

Although it achieved promising results and provided valuable insights into the application of ML techniques within the automotive industry, it was constrained by limitations in data scope and quality, as well as the predictive accuracy obtainable with the available resources and time. Specifically, the limited volume and diversity of data reduced the model's robustness, and incomplete data affected overall performance.

This dissertation represents a substantial revision and improvement, both in processes and architecture, of the previous project. It expands on the initial methodology by incorporating broader and more diverse data sources, employing improved preprocessing methods, and using advanced modeling frameworks supported by greater computational power and modern ML libraries. The objective is to address the shortcomings identified in the initial work and develop a more reliable, production-ready system capable of delivering improved prediction quality.

By explicitly extending the previous work's foundation, this work maintains continuity while advancing state-of-the-art predictive methodologies for actual vehicle fuel consumption values estimation.

2.5 Conclusion

This literature review has examined the various research questions introduced in the Introduction related to predicting vehicle fuel consumption, laying a solid foundation for the development and analysis of the tool discussed in this dissertation.

RQ1 showed that real-world fuel consumption is influenced by complex interactions among three main areas: vehicle characteristics, environmental conditions, and driver behavior.

Engine size, power, and weight remain the most important factors affecting fuel economy and recent studies indicate that the rising SUV sales have led to an approximately 10 to 12% drop in average fuel consumption values. Weather conditions, especially ambient temperature and road

elevation changes, can cause seasonal fuel consumption variations greater than 10%, while driving style also plays a significant role, causing fuel consumption to vary by 10 to 20%. The evidence strongly supports the creation of a ML model based on these three main areas, meaning that ML techniques that combine these factors can achieve R^2 values above 0.95.

Regarding RQ2, a comparison of machine learning methods finds that XGBoost balances accuracy (R^2 between 0.95 and over 0.99), scalability, and interpretability well, aided by tools like SHAP. RF models provide solid baseline accuracy (R^2 from 0.85 to 0.91) and easy interpretability, making them fit for early development and fleet-level monitoring. NNs, especially LSTM and hybrid models, do well with time-series data (R^2 from 0.90 to over 0.97), although they demand more computing power and specialized interpretability approaches.

The findings suggest that hybrid approaches mixing tree-based and neural methods hold the most promise for practical deployment.

Analysis of RQ3 identifies key design principles to close the gap between lab and real-world fuel consumption, which can be as large as 10 to 40%. The literature showcases five essential factors:

- Diverse real-world datasets that anchor predictions in actual driving conditions.
- Personalized behavioral and environmental inputs that capture individual usage patterns.
- Hybrid ML models with integrated explainability features.
- Interactive interfaces that improve a user's decision-making ability.
- Continuous learning systems that maintain accuracy across evolving vehicle technologies and driving patterns.

The review identifies several important gaps in current research that this dissertation aims to fill.

First, while many studies focus on specific prediction methods, there is little work on systems that combine multiple data sources alongside user interfaces to support create the predictions, and aid in decision making.

Second, much of the existing academic work relies heavily on controlled datasets, but this dissertation takes a different path by focusing on creating a scalable pipeline for gathering and processing directly from real driving conditions.

Lastly, a notable gap in the literature is the absence of comprehensive evaluation frameworks that effectively balance prediction accuracy with practical usability, which is crucial for end users.

Building on the foundational principles introduced in the Estimating, Detection and Learning II, as part of the Master's Degree, this research significantly broadens its scope by integrating a wider variety of data sources, using more advanced preprocessing techniques, and utilizing modern machine learning frameworks. Through a systematic process of analyzing these research questions and developing new tools, this dissertation contributes meaningful insights both to academic knowledge and to practical applications in fuel consumption prediction.

Chapter 3

Chapter 3 - Proposed Solution & Methodology

This dissertation aims to develop a scalable, interpretable system for predicting real-world fuel consumption across diverse automotive scenarios. Building upon foundational work originally presented in the author's earlier report for Estimating, Detection and Learning II [102], the solution draws on best-in-class data engineering and machine learning practices to integrate heterogeneous sources, including scraped technical specifications and survey responses containing behavior data, within an efficient cloud infrastructure to optimize results, performance, and scalability.

This chapter outlines the methodological advancements and choices that were made beyond the initial study, detailing the design choices and technical implementations that enabled the creation of a robust, production-ready prediction tool.

3.1 Data Acquisition and Web Scraping

From its initial concept in previous work to the solution proposed here, this dissertation is fundamentally dependent on acquiring quality data from public websites through web scraping techniques. Selecting the appropriate source website and web scraping tools was critical, as these choices influence the scope and quality of data collected, and set the foundation for the entire research process.

3.1.1 Data Source Selection

The core data foundation for this dissertation, the car specifications data, was retrieved from the website Ultimate Specs [123]. It provides up-to-date technical details for more than 50,000 car models, including information on engines, dimensions, and fuel-related parameters. Its extensive catalog covers the vast majority of vehicles currently found on the road, ensuring that the resulting analyses are both relevant and broadly applicable. This choice directly aligns with recent research highlighting the critical importance of detailed vehicle attributes in accurate fuel consumption modeling [9, 108].

While alternative data sources such as Automotive Data [19] and EncyCARpedia [28] exist and offer similarly broad coverage, a closer examination revealed substantial practical differences that favor Ultimate Specs, especially for large-scale, research-focused data extraction. Specifically:

- Ultimate Specs stands out for its completely public access, requiring no login or registration. The website's logical structure allows programmatic navigation through brands, models, and specific vehicle versions. For example, to extract the full dataset (with 263 brands, $\tilde{1}0$ models per brand, and an average of 19 versions per model, at the time of extracting), the extraction process involves a straightforward sequence:
  - 1 request for the brands page
  - 263 requests for brand-version lists
  - 49.700 requests for version details of each specific car.

  This totals 49.700 requests to cover all cars, brands, and versions, a feasible workload enabled by the predictable and scrape-friendly layout. The platform is maintained with frequent updates and does not impose paywalls, authentication, or aggressive anti-scraping measures, further supporting efficient and ethical data collection.
- Automotive Data, despite offering comprehensive data, presents significant hurdles. Most scalable access is locked behind a paid API, while its web interface is less amenable to automated extraction due to deeper menu structures, AJAX-based navigation, and pagination. As a result, obtaining an equivalent dataset would require several times more requests and involve higher risk of being rate-limited or blocked, not to mention incurring substantial costs if the API is used.
- EncyCARpedia requires users to sign in to access full vehicle listings and details. This authentication barrier complicates automated extraction, and navigation is structured around search and filter forms, meaning every lookup and detail view generates individual requests. Bulk dataset collection is not only slower and more fragmented but also exposes users to potential account restrictions or bans during aggressive use.

In summary, Ultimate Specs provided a uniquely open, up-to-date, and technically efficient source for vehicle data, making it exceptionally good for academic-scale modeling and analysis. The absence of commercial barriers, straightforward navigation, and a large, current dataset set it fundamentally apart from other leading alternatives.

3.1.2 Extraction Approach

A custom extraction pipeline was implemented using Scrapy [115], an asynchronous open-source framework optimized for large-scale data scraping tasks. Scrapy's architecture enables efficient and scalable data harvesting via direct HTTP interactions, which supports high-throughput extraction necessary for this dissertation [51].

To maintain uninterrupted scraping activity, the solution incorporates Zyte API proxies. This approach obscures scraping traffic and mitigates IP-blocking risks typical when accessing dynamic website content, ensuring continuous data availability [155, 7].

3.1.3 Tool Selection: Scrapy vs. Alternatives

UltimateSpecs loads as a complete page, meaning the text and links visible on it are also present in its source code, and the layout is not 'filled in' after loading, so there is no additional JavaScript constructing content after the initial load and the pages function even with JavaScript disabled.

Given this, the next step was selecting a scraping tool suited to static pages, prioritizing reliable HTML retrieval, resistance to blocking, and cost efficiency, rather than capabilities for rendering or executing JavaScript.

3.1.3.1 Selenium

Selenium is a widely used open-source web framework mainly designed to automate web application interactions, particularly in testing environments. It's the most popular tool in the test automation landscape, being used in approximately 75% to 80% of automation projects, which shows its broad adoption and reliability [6]. Selenium WebDriver, the framework's core component, emulates real interactions with browsers. It supports multiple programming languages such as Java, Python, C#, and JavaScript and works seamlessly across major browsers including Chrome, Firefox, Safari, and Edge [117].

This flexibility makes Selenium important not only for testing but also for automating general web-based tasks and browser operations [133]. Figure 3.1 shows an overview of its architecture. It provides excellent integration capabilities with testing frameworks like JUnit and TestNG, making it the main component of many CI/CD pipelines and delivery automation processes [105].

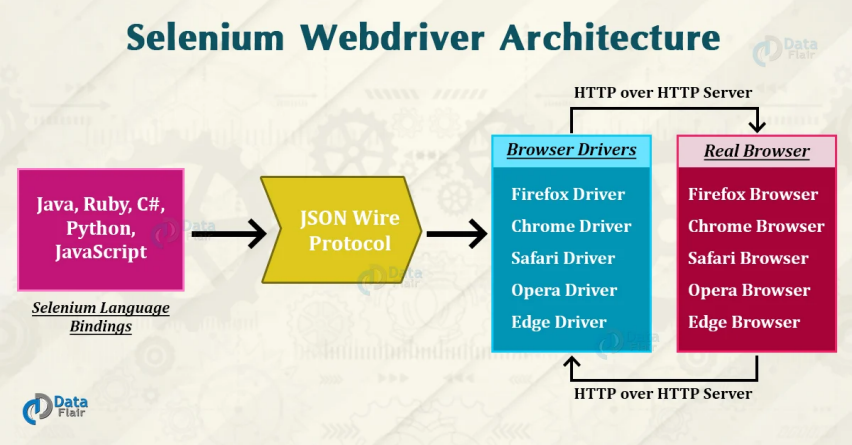

Figure 3.1: Selenium framework [20]

Selenium's primary advantages include the ability to automate tasks that replicate human interactions within browsers, resulting in realistic browsing experiences [117, 15]. It is well-suited

for interacting with websites that heavily rely on JavaScript, making it the most suited for dynamically loaded pages [117]. Furthermore, Selenium's cross-platform compatibility ensures consistent behavior between major browsers and operating systems such as Windows, macOS, and Linux [117, 135].

Its relatively simple API and strong community support make it beginner-friendly compared to more specialized tools [145]. Additionally, it allows writing automation scripts in various programming languages and integrates well with other frameworks and CI/CD pipelines [117, 133].

Nevertheless, Selenium also comes with certain drawbacks. It can be resource-intensive and slower because browser-based interactions require significant system resources and communication overhead between the test script and browser driver [15, 134]. While Selenium is suitable for basic scraping, it is not optimized for large data volumes or high-speed operations, which makes it less effective for big data extraction tasks [49, 132]. Setting up Selenium can also be complex, as it requires configuration of browser drivers and integration with external tools for reporting and parallel testing [117, 15]. Finally, Selenium tests may be unstable on dynamic and JavaScript-heavy pages, leading to flakiness due to timing issues and DOM changes [132, 24].

3.1.3.2 Scrapy

Scrapy is an open-source framework created specifically for large-scale web scraping and data extraction projects [10, 5]. Its asynchronous architecture supports handling multiple requests simultaneously, resulting in faster data extraction than browser automation tools. A diagram showing its architecture is displayed in Figure 3.2.

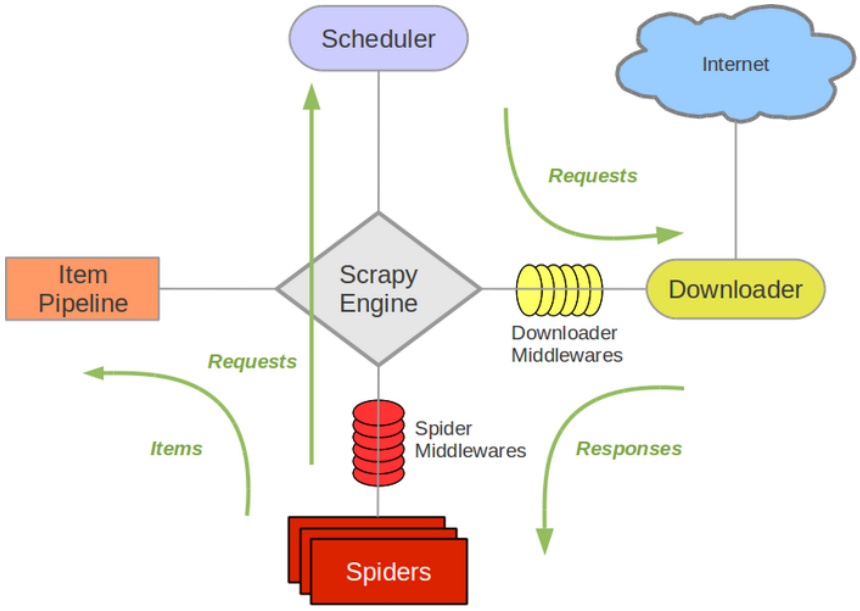

Figure 3.2: Scrapy architecture [116]

Scrapy is different from tools like Selenium, which focus on browser automation and user

interaction simulation. Instead, Scrapy is built for crawling and scraping at scale, operating without the overhead of rendering web pages [14, 104]. Its efficient resource usage and support for proxies and caching make it ideal for quickly collecting large amounts of static or semi-dynamic content [97].

Some of Scrapy's most important advantages include its optimization for high-speed, asynchronous HTTP request handling, making it particularly well-suited to large-scale data extraction tasks [38, 112]. It is also significantly less resource-intensive than browser-based automation tools [114, 156]. The built-in pipeline architecture of Scrapy streamlines data processing, cleaning, and storing, which is especially beneficial for complex scraping projects [115, 60]. Moreover, Scrapy's flexible architecture allows for custom middleware, complex navigation logic, and adaptation to various content types [38, 156].

On the other hand, the fact that Scrapy does not render pages imposes several limitations. It is less effective for JavaScript-heavy or dynamically loaded content, where tools like Selenium are more appropriate [97, 5]. This framework also means a steeper learning curve for beginners due to its setup and configuration requirements [112, 27]. Furthermore, Scrapy is not designed for automated or functional testing, thus its utility for quality assurance workflows is limited compared to Selenium [117, 133].

3.1.3.3 Requests

Requests is a Python library that simplifies communication over HTTP, offering a straightforward, readable syntax, that allows for rapid and convenient network communication with web servers [107, 97]. This makes it particularly well-suited for simple scraping and API interactions. Unlike Selenium or Scrapy, Requests does not provide browser emulation or page rendering, operating at the network level instead, as shown in Figure 3.3.

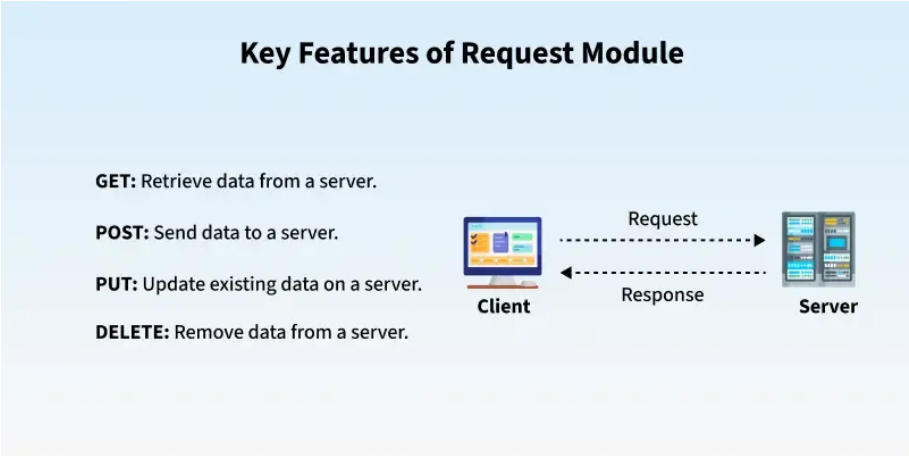

Figure 3.3: Requests architecture [42]

The primary advantages of Requests include, but are not limited to, its ease of use, which

makes sending HTTP requests accessible to beginners [107]. Requests is efficient for interacting with APIs, as it supports all main request methods, GET, POST, PUT, and DELETE [66]. The library also provides flexibility in modifying headers, cookies, and other request parameters, adapting well to a variety of scenarios [97]. Since it does not require extensive system resources, Requests is lightweight and particularly suitable for environments with limited capacity [42].

However, the limitations of Requests should be considered. It lacks support for rendering JavaScript, which limits its effectiveness for scraping websites with dynamic content [97]. While it suffices for simple tasks, Requests is not appropriate for large-scale data extraction [112]. Lastly, as Requests does not mimic a web browser, it cannot interact with page elements or dynamic content the same way as Selenium does [117].

3.1.3.4 Summary

In reviewing these web scraping tools, each option, Selenium, Scrapy, or Requests, has distinct strengths tailored to specific needs, as can be seen in Table 3.1.

Tool	Primary Use	Pros and Cons
Selenium	Browser automation and testing	+ Realistic user interaction; handles JavaScript and dynamic content; supports multiple browsers. – Resource-intensive and slower; complex setup; limited scalability for bulk scraping; flaky on JS changes.
Scrapy	High-performance static page scraping	+ Asynchronous requests for speed; low resource usage; built-in pipelines; flexible middleware; integration with other tools. – No browser rendering; steeper learning curve; not suited for testing workflows.
Requests	Simple HTTP requests and API calls	+ Lightweight; very easy syntax; ideal for API interactions; customizable headers. – Cannot render JavaScript; limited to simple scraping; no browser emulation.

Table 3.1: Comparison of scraping tools selected for static page extraction

Selenium is particularly useful for tasks that require simulating human interactions, such as testing and working with JavaScript-heavy sites.

Requests is a great choice for straightforward tasks and API interactions, known for its efficiency and small learning curve.

Extracting data on a larger scale, where speed, efficiency, and scalability are the pillars that shape the technologies used, Scrapy stands out as the top choice. Its architecture allows for fast, resource-efficient scraping with powerful data pipelines and flexible configurations, making Scrapy ideal for complex, high-volume scraping projects. For this dissertation, which requires both power and scalability, Scrapy is the clear choice.

To summarize this in a more visual manner, a decision flowchart is shown in Figure 3.4.

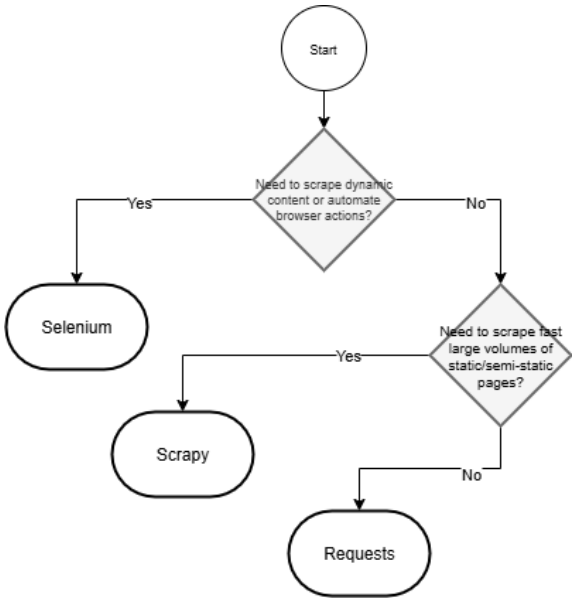

Figure 3.4: Simple scraping framework decision flowchart

3.1.4 API Selection for Web Data Extraction

To ensure consistent and efficient web data collection, particularly across sites with rate limiting, bot protection, or IP blacklisting, a resilient scraping API with rotating proxies was deemed essential. After evaluating multiple providers, Zyte API was selected for this dissertation based on its balance of usability, cost control, and technical reliability [155].

3.1.4.1 Justification for Zyte API Implementation

During early development, scraping attempts directly from the target platform (ultimatespecs.com) frequently encountered IP blocks, request throttling, and CAPTCHAs, severely limiting data collection rates and risking incomplete datasets.

These anti-bot mechanisms made it impossible to maintain a stable scraping workflow using vanilla approaches. The implementation of Zyte API was therefore deemed necessary to circumvent frequent platform-imposed blocks reliably and ethically. Zyte's managed proxy rotation and automatic ban detection provided the technical infrastructure needed to sustain uninterrupted access to the full vehicle database [155].

3.1.4.2 Advantages of Zyte

Zyte's prime advantages consist of:

- Pricing and Request Allowance: Zyte's entry-level plan (with a $5 credit limit) allows for around 40,000 successful requests per billing cycle. The dissertation has already required approximately 49,700 requests, and this number continues to grow as the dataset expands with new vehicle models and versions. To accommodate this, data extraction was split

across two separate free trial accounts, enabling the collection of the current dataset within the free quotas.

- Ease of Integration: Zyte is natively compatible with Scrapy and similar Python scraping frameworks [101]. Configuring the API requires minimal code, allowing fast integration and deployment.

- Reliability Features: The API automatically manages IP rotation, ban detection, and basic anti-bot challenges. Early stopping and failure handling logic were incorporated in the scraping pipeline to prevent wasting API credits on failed or redundant requests, which helped optimize the use of the free allocation.

3.1.4.3 Alternative Solutions

Other solutions, like Bright Data, ScraperAPI and Oxylabs are also available as an alternative, but were not chosen for the project due to the following reasons:

- Bright Data (Luminati): Offers enterprise-grade proxy services but comes with a significantly higher cost structure and no free tier for this use case due to trial periods being limited in request volume (often under 1,000 requests). This makes it impractical for datasets of this magnitude without substantial funding. Setup is also more complex, demanding manual proxy management [13, 74].

- ScraperAPI: Intended for developers and startups, ScraperAPI offers a free trial but with a cap of approximately 5,000 requests and restricts advanced features like JavaScript rendering to paid plans. This limits scalability during trial or budget-restricted phases [113, 48].

- Oxylabs: Targets enterprise clients and does not offer accessible free tiers. Its pricing models make it unsuitable for academic or self-funded projects requiring cost-effective, scalable solutions [74].

3.1.4.4 Summary

By using Zyte's free trial quotas and robust proxy infrastructure strategically, this dissertation achieved approximately 49,700 requests (and counting) with only two trial accounts, leaving enough margin for testing, efficiently collecting the dataset. The early stopping mechanisms and error handling in the data pipeline further maximized the effectiveness of the free allocation.

Most critically, Zyte's proxy management and anti-block technologies enabled stable and continuous scraping, overcoming the frequent direct platform blocks experienced prior to its integration [155].

In contrast, alternatives either lacked usable free tiers, imposed more restrictive quotas, or demanded prohibitively complex management and costs, as summarized in table 3.2 [13, 113, 74].

API/Service	Advantages	Drawbacks	Use Cases	References
Zyte API	Free trial (40,000 requests for $5); easy integration; reliable proxy rotation	Multiple trials for large projects	Cost-efficient scraping with proxy management	[155, 114]
Bright Data	Enterprise-grade, powerful proxies	Complex and expensive setup	Enterprise scale scraping	[14]
ScraperAPI	Simple API; good for developers	Low trial limits (5,000 requests); restricted features	Small projects, early development	[114]
Oxylabs	Robust, enterprise APIs	No free tier; costly; onboarding process	Enterprise data collection	[97]

Table 3.2: Comparison of API/Service Providers

Therefore, Zyte's combination of affordability, ease of implementation, scalability, and automatic proxy management uniquely met the dissertation's demands for a reliable, cost-neutral web data extraction tool, outperforming available alternatives under similar constraints. The decision flowchart for this API is represented in Figure 3.5:

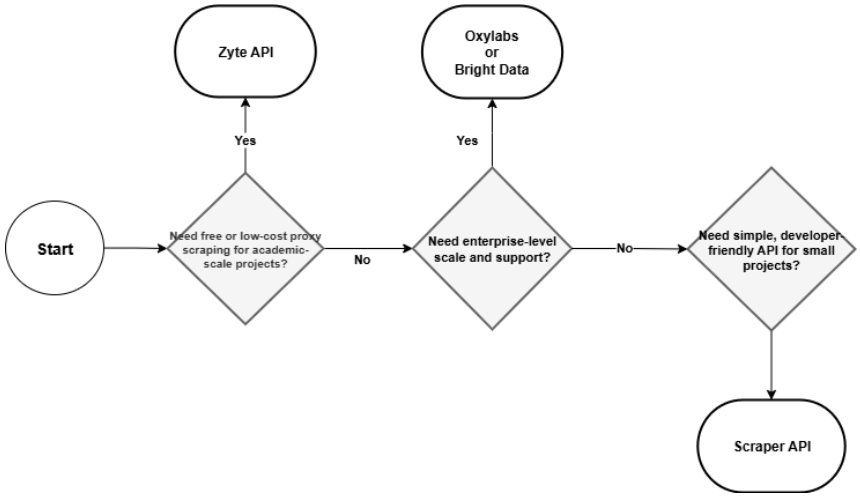

Figure 3.5: Flowchart outlining the decision process for the API

3.2 Data Processing and Data Warehouse

One of the main objectives for this dissertation to separate it from the previous work is the implementation of a modern cloud-based architecture for dataset storage. Modern data platforms offer various solutions for data analytics, with Snowflake, AWS, and Azure as the most viable options for this specific case.

Approaching this solution this way leads to improved performance, scalability, and maintainability compared to the typical, traditional on-premises solutions.

This section examines the different choices made, their rationale, and subsequent data processing methodologies used to support the fuel consumption prediction system.

3.2.1 Snowflake Data Lake Architecture

The processed data is systematically organized into Snowflake's cloud data warehouse via a bronze-silver-gold layering architecture:

- The bronze layer retains raw extracted data to preserve source integrity
- The silver layer comprises normalized, cleaned data with standardized units and feature engineering
- The gold layer contains fully integrated datasets enriched with derived variables designed to optimize ML model input quality [54]

The complete architecture diagram can be seen in Figure 3.6.

3.2.2 Selection Rationales: Snowflake vs. Azure Synapse, AWS Redshift

To choose the best cloud solution for this dissertation, Snowflake, AWS, and Azure were evaluated. Each solution was assessed based on its performance, cost, scalability, and how well they met the project's needs.

3.2.2.1 Snowflake

Snowflake is a cloud-native data warehousing platform recognized for its fully managed, scalable data storage and analytics spanning AWS, Azure, and Google Cloud [74, 45]. Its innovative multi-cluster shared data architecture, as seen in Figure 3.6, enables users to separate compute and storage workloads for efficient and elastic scaling, eliminating the infrastructure management overhead present in traditional systems [121]. Snowflake's architecture allows simultaneous analytical queries by multiple teams without resource contention, an advantage highlighted in recent cloud analytics literature [83].

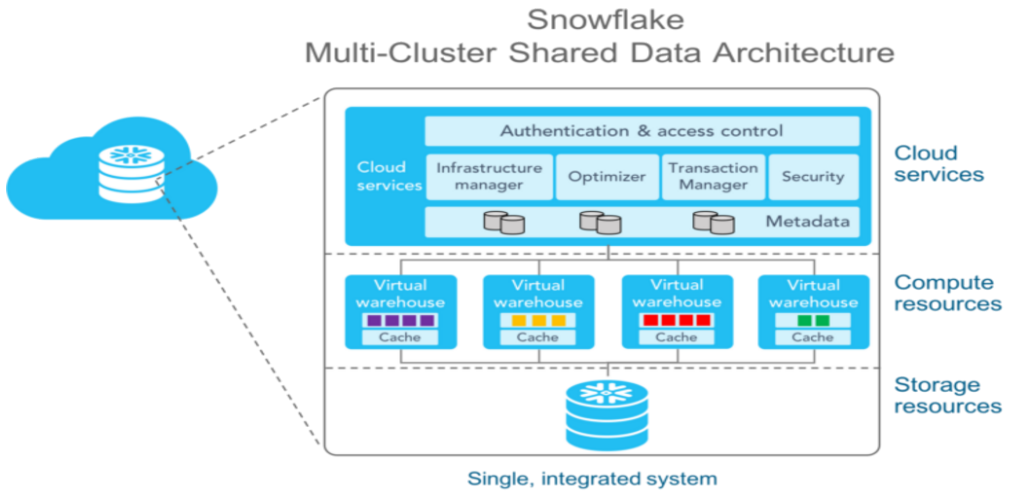

Figure 3.6: Overview of the Snowflake Architecture [121]

Snowflake's appeal for data science workflows is further strengthened by its broad SQL support, integration with Python and Spark, and advanced features such as data sharing, time travel for recovery, and automatic scaling [100, 54].

However, Snowflake's proprietary design, while powerful and highly optimized for cloud data warehousing, introduces significant vendor lock-in concerns. Its unique architecture and storage mechanisms are not easily transferable to other platforms, making migration costly and complex[26].

This creates long-term dependency on Snowflake's pricing models and update cycles, restricting flexibility and potentially increasing organizational risk as the cloud data landscape evolves [22, 128, 26]. Although Snowflake supports multi-cloud deployments, true data and workload portability remain limited, requiring careful planning to mitigate lock-in effects [127].

3.2.2.2 AWS Redshift

AWS Redshift is Amazon's cloud data warehouse solution, built on PostgreSQL and tightly integrated into the AWS ecosystem [3]. It offers fast query performance using columnar storage and parallel query execution, which is highly effective for analytics workloads within Amazon's cloud environment [83]. Its maturity and established user base make it a popular choice for organizations already using AWS services, and its architecture overview is shown in Figure 3.7.

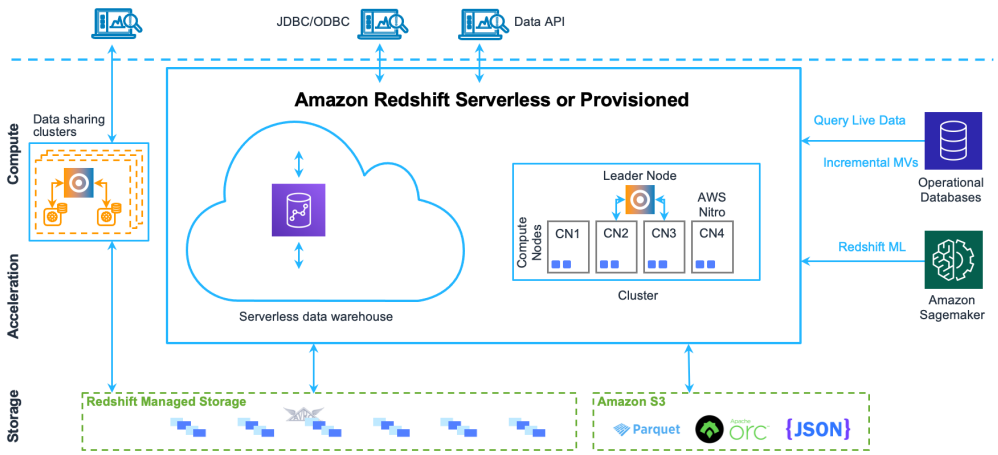

Figure 3.7: Overview of the Redshift Warehouse Architecture [2]

Despite its strengths, Redshift couples compute and storage resources, requiring manual cluster resizing, which can be disruptive [121], and this coupling can limit flexibility in scaling and cost control. Such architecture increases management complexity and reinforces dependence on AWS infrastructure and pricing strategies, contributing to vendor lock-in, particularly for organizations heavily invested in native AWS services.

While it offers strong compatibility within AWS, migrating to other cloud providers involves significant effort due to Redshift's proprietary features [74, 22, 128].

3.2.2.3 Azure Synapse Analytics

Azure Synapse Analytics, formerly known as SQL Data Warehouse, is Microsoft's unified analytics solution that combines data warehousing and big data capabilities [84], as shown in Figure 3.8. It integrates closely with the Azure ecosystem, including Power BI, Data Lake Storage, and Machine Learning Studio, offering both on-demand and provisioned resource allocation. Advanced security features such as Active Directory integration and encryption further strengthen its enterprise readiness [74].

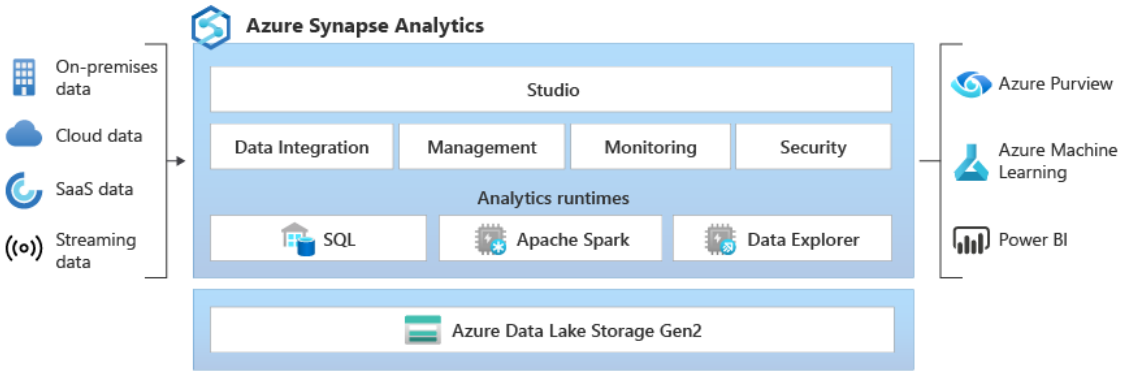

Figure 3.8: Overview of the Synapse Analytics Architecture [84]

Like AWS Redshift, Azure Synapse requires more configuration and tuning compared to Snowflake and is best suited for organizations already committed to the Azure cloud environment [83]. Its lack of broad multi-cloud support can be restrictive for organizations seeking flexibility to operate across multiple cloud platforms [45].

This tighter coupling to Azure services increases the risk of vendor lock-in, making migration away from the platform costly and complex [22, 128].

3.2.2.4 Summary

Snowflake's separation of compute and storage, cross-cloud support, and minimal operational management make it ideal for modern, scalable, and collaborative data science projects (Snowflake Inc., 2024). AWS Redshift and Azure Synapse are robust options within their ecosystems but require more manual intervention and present greater vendor lock-in risks [83, 45]. A decision flowchart, highlighting the process taken to arrive at this conclusion is displayed in Figure 3.9.

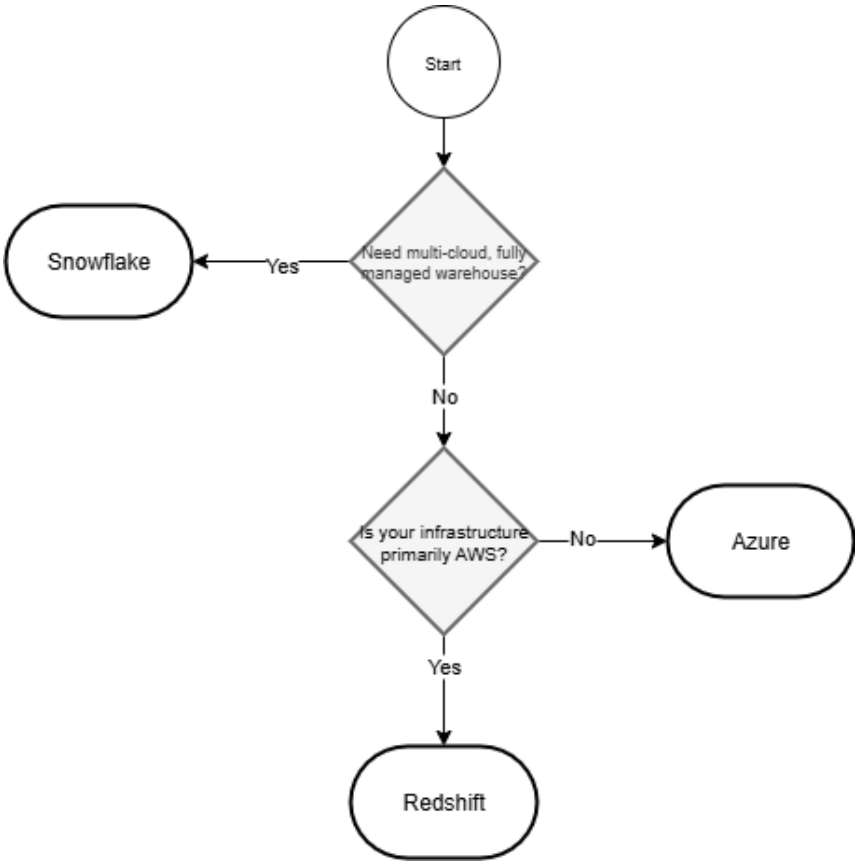

Figure 3.9: Data Warehouse Flowchart

3.3 Web Application for User Reported Fuel Consumption

A user-centered interface built on Streamlit [130] enabled dynamic exploration of vehicle specification selections, driver profile input, and real-time fuel consumption reporting. Streamlit's minimal-code framework supports rapid deployment and seamless integration with Python ML workflows. These features made it the perfect tool to gather responses and integrate them directly into Snowflake to gather real life data, by considering different uses of the same vehicles, painting a more reliable picture of the real fuel consumption.

3.3.1 Tool Selection: Streamlit vs Dash, Flask, Django

Streamlit, Dash, Flask, and Django were evaluated for their ability to build a user input form. The comparison focused on how easily each framework integrates with Snowflake, their simplicity for rapid development, and the expected time required to implement and deploy the form.

3.3.1.1 Streamlit

Streamlit is a Python-native, open-source framework expressly designed for rapidly building interactive forms and dashboards for data science applications [101, 130]. Its declarative API lets

developers transform simple Python scripts into dynamic web forms with live validation, widgets, and instant feedback, minimizing boilerplate code.

Streamlit uniquely supports seamless integration with pandas DataFrames, and with Snowflake, which, as the tool selected for warehousing, made this a good fit for the dissertation.

For the user fuel consumption form, Streamlit's advantages are clear:

- Instant form creation with st.form and live widgets
- All logic remains in Python, avoiding extensive HTML, CSS, or JavaScript
- Real-time feedback and session management, ideal for capturing user data and delivering predictions or downloadable receipts
- Minimal setup required for deployment and authentication
- Direct integration with backend ML models and databases (Gupta et al., 2022)

3.3.1.2 Dash

Dash, built by Plotly, is also Python-native and supports interactive HTML components and callbacks [103]. Dash permits more granular UI customization but demands deeper familiarity with component-based front-end development.

While forms can be implemented using core Dash components, the process often involves more code and design effort than Streamlit, especially for live, iterative form validation.

3.3.1.3 Flask

Flask is a minimalist Python web framework that excels at custom web applications and REST APIs [47]. Building a submission form in Flask requires manual HTML form templates, explicit validation, and session handling, tasks that slow development cycles for data science teams focused primarily on analytics rather than web engineering, which made it not suitable for this dissertation.

3.3.1.4 Django

Django is a comprehensive web framework containing a powerful form subsystem, ORM, and built-in admin interface [57]. However, its extensive setup (requiring explicit model definitions, template rendering, and user management) can be excessive for simple forms intended for lightweight research projects, falling into the same category as Flask, in that regard.

3.3.1.5 Summary

For the fuel consumption submission form of the dissertation, Streamlit's combination of instant form building driven by Python, live feedback, secure deployment, and simple integration with

Snowflake offers the efficiency and usability necessary to collect and process user submissions in an academic research context [101, 130, 48].

Dash, Flask, and Django, while powerful, would require substantially more effort and design for equivalent outcomes, as detailed in Table 3.3.

Framework	Advantages	Drawbacks	Use Cases	References
Streamlit	Rapid prototyping; minimal code for interactive dashboards	Basic UI customizability	Perfect for data science apps, quick ML dashboards	[101, 130]
Dash	Advanced UI components; high customization	Steep learning curve: more code	Custom analytic dashboards with rich interactivity	[103]
Flask	Lightweight, flexible	Requires full-stack knowledge, manual UI integration	Custom APIs and complex web apps	[47]
Django	Full web framework; authentication and admin tools	Complex setup: may be overkill for simple apps	Enterprise-grade web apps with complex logic	[57]

Table 3.3: Comparison of Web App Technologies

In addition, the decision flow chart to guide future reference is shown in Figure 3.10.

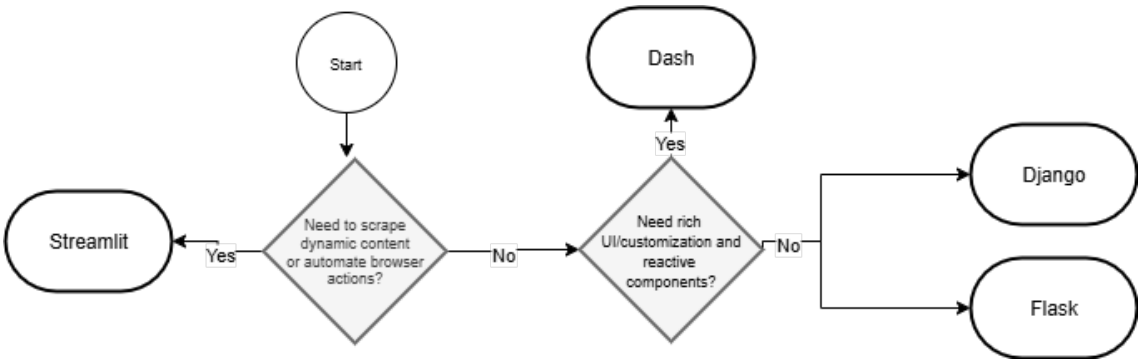

Figure 3.10: WebApp Framework Flowchart

3.4 Synthetic and Survey Data Integration

Due to time constraints for the dissertation, the Streamlit form was only live on the internet for 25 days, which meant that no substantial amount of responses were gathered, in order to provide an accurate estimation of fuel consumption values. Because of this, it was required to augment the existent data with context-aware synthetic data.

3.4.1 Sample Size Determination and Data Augmentation Strategy

To build a reliable fuel consumption prediction system, adequate data coverage is essential across all vehicle versions in the dataset. With aproximately 50.000 unique vehicle versions, estimating

accurate fuel consumption averages per version typically requires multiple user responses per version to capture variability and reduce statistical uncertainty.

3.4.2 How the 125,000 Target Was Determined

In statistical practice, estimating the mean of any variable (such as fuel consumption) with reasonable confidence and precision typically demands 30 to 50 observations per category or version to assure robustness [90, 119] Using the following as guidance:

For 30 responses per vehicle version, the total ideal sample size would be:

$$\mathbf{30 \times 50 = 1,500,000}$$

For 50 responses per vehicle version, the number of responses increases to:

$$\mathbf{50 \times 50 = 2,500,000}$$

However, collecting millions of organic responses is impractical within the dissertation's time and resource constraints. To find a feasible compromise, a proportional multiplier of 2.5 times the number of unique versions was chosen, which led to a target of:

$$\mathbf{2.5 \times 50 = 125,000}$$

This number represents a pragmatic balance, aiming to gather enough user responses to contribute meaningful variability and coverage, while keeping computational requirements manageable and enabling timely analysis [96].

3.4.3 Practical Data Collection and Need for Augmentation

In the 25 days that the form was live, about 25 responses were achieved, a much smaller number than the 125,000 minimum value defined, due to the ramping nature of these forms and lack of broad reception, even after publishing the App in a *Reddit* post and sharing with colleagues and family. This required that these answers be supplemented by synthetic responses based on organic data to meet the reliability and diversity goals.

To address this, a statistically grounded synthetic data generator was created, producing additional driver demographics and behavior profiles to fill gaps, capturing variability that the sparse organic data could not. By merging synthetic and real data, the system achieves broader coverage and reduces bias risks without compromising the integrity of the model's predictions.

This approach aligns with recommended practices in data-driven environmental modeling and machine learning applications [50].

3.5 Feature Engineering

Data extracted from scraping undergoes meticulous cleaning and normalization, addressing inconsistencies such as unit variations, missing values, and duplicated records through feature engineering. This process follows industry best practices in automotive data preparation ensuring data quality and relevance for modeling [78, 25]. The feature engineering workflow transformed inconsistent vehicle specification entries into a model-ready dataset through steps that are reproducible and verifiable. These were:

- Create a unique ID for each car
- Standardize measurement units to International System of Units (SI)
- Consolidate target value with origin, and calculations
- Harmonize powertrain values
- Unify electric and hybrid attributes
- Infer values based on industry knowledge
- Normalize key fields
- Fill in missing information

An overview of the engineered fields is outlined in Table C.1, their units, and derivatives is provided in the feature inventory table for quick reference.

To ensure that each vehicle appeared only once in the dataset and later in the database, the records were first deduplicated using an identifier present in the source URL structure. Duplicate rows were removed using this key, while placeholders were normalized to true missing values to allow consistent downstream handling.

This identity creation step anchors every subsequent transformation and is documented in the feature inventory for traceability, as seen in Table C.1.

Physical measurements and numeric fields were then standardized to enable comparability between sources. Imperial units were converted to SI, numeric values were parsed from text where applicable, and performance figures were normalized to common units.

Specific conversion formulas, rounding policies, and typing choices are summarized in the unit conversion map, which makes these measurement decisions explicit for replication and review. This mapping is shown in Table 3.4.

Field pattern	Target unit	Conversion, rounding, handling
X inches	meters	meters = inches / 39.37; round to 2 decimals; nullable float
X lbs	kilograms	kg = lbs×0.453; round to 1 decimal; nullable float
X mph	km/h	km/h = mph × 1.60934; integer rounding; nullable integer
X lb-ft	Nm	Nm = lb-ft × 1.3558; integer rounding; nullable float
X L	liters	Parse numbers before “L”; coerce to numeric keeping nulls

Table 3.4: Unit conversion map for physical and performance measurements

A single, unified target variable for combined fuel consumption was constructed to maximize coverage while preserving origin. When a direct combined value was present, it was retained. Otherwise, a weighted combination of city and highway values was calculated, and when only one component existed, that value was used.

Every value assignment recorded a source tag to ensure auditability of the target construction. The order of the rules and the corresponding origin tags are specified in Table 3.5.

Condition met	**Assigned target**	**Original column**
Combined consumption present	Use combined	combined
City and highway present	0.55×City+0.45×Highway	estimated_city_highway
Only city present	Use city value	city_only
Only highway present	Use highway value	highway_only
None present	Missing value	missing

Table 3.5: Target construction and origin tagging rules

Powertrain description values were harmonized to reduce skewness and strengthen interpretability. The "Aspiration" values were changed to a compact list of options, grouping varied raw descriptions into canonical classes that simplify modeling and downstream interpretation.

These classification rules, the indicative cues used to assign labels, and additional handling notes are detailed in Table 3.6.

Standard class	**Indicative cues**	**Notes**
Naturally Aspirated	strings starting with "n", value "8"	Handles noisy markers that denote "NA"
Turbo	turbo	Vendor mentions without "twin" default to Turbo
Turbo + Intercooler	"turbo" and "intercooler"	Combined forced induction
Variable Geometry Turbo	"variable geometry", "VGT", "TGV"	With "intercooler" yields combined label
Supercharger / Compressor	"supercharger", "compressor", "roots", "volumex"	Pure supercharging
Supercharger / Compressor + Turbo	"supercharger" and ("turbo" or "variable geometry")	Mixed forced induction
Twin / Bi-Turbo	"twin turbo", "biturbo", "bi-turbo", "twin"	Vendor cues plus "twin" yield "twin/bi"
Carburetor	"carb" substring	Legacy systems
Other / As-is	none matched	Returns cleaned original string

Table 3.6: Aspiration normalization classifications and cue patterns

Fuel system values were also simplified into domain-relevant classes to reduce complexity. Diverse raw strings were mapped into categories such as Direct Injection, Multi-Point Injection, Electronic Fuel Injection, and Carburetor, among others. The comprehensive mapping, including primary keywords and exceptions, is listed in Table 3.7.

Fuel System Class	Indicative Cues	Notes
Direct Injection	Keywords: "common rail," "CRDi," "CDTI," "TDI," "HDI," "GDI"	Includes diesel and petrol di systems
Multi-Point Injection	"MPI," "MPFI," "multi-point"	Multi-port injection types
Single-Point Injection	"SPI," "single-point"	Single body injection
Electronic Fuel Injection	"electronic," "EFI," "Digifant"	Electronic system variants
Mechanical Fuel Injection	"mechanical," "K-Jetronic," "CIS"	Mechanical systems
Throttle Body Injection	"TBI," "throttle body"	Single injector at throttle
Combined Injection	"combined injection," "direct+multipoint"	Hybrid injection systems
Carburetor	Contains "carburetor" and variants	Legacy systems
Unspecified - Euro Norm	Contains "euro"	Not an injection type
General Petrol System	Keywords: "petrol," "gasoline"	Vague petrol systems
None	Explicit "none" or blank	Absence of system
Other	No match	Fallback category

Table 3.7: Fuel system simplification classification

Similarly, transmission labels were normalized into two clear categories: "Automatic" and "Manual". Keyword detection rules classified entries mentioning "auto," "CVT," and related terms as Automatic, while strings explicitly noting "manual" or lacking distinct identifiers were labeled Manual. In ambiguous cases, the default was Manual to maintain consistency in modeling. Table 3.8 shows the classification approach with representative cues and default handling.

Transmission Class	Indicative Cues	Default Handling
Automatic	Contains "auto," "CVT," "dual," "sequential," "reduction," "EDC," "PDK"	Classified as Automatic
Manual	Contains "manual," or blank or ambiguous entries	Default to Manual if ambiguous

Table 3.8: Transmission normalization rules

Where native fields were missing or ambiguous, additional inferences completed the feature set. Fuel type was inferred from brand, model, and version indicators, the presence of a catalytic converter was inferred using fuel type and production year thresholds while respecting explicit negative entries, and body style was classified via a mapping supplemented by keyword heuristics, with an "Undefined" fallback in edge cases. These categorical inferences and their output classes are summarized in Table 3.9.

Field	Primary cues	Output classes and notes
Fuel Type	Version, model, brand keywords	Electric, Petrol or CNG, Other; EV cues include line names and energy terms
Catalytic Converter	Fuel type, production year	Y or N; respects explicit negative entries
Body Type	Curated mapping, keywords	Convertible, Coupe, Estate, Sedan, MPV, SUV, or Undefined if unresolved

Table 3.9: Additional categorical inferences

The electric and hybrid attributes were consolidated to capture the system capability consistently across sources. The types of electric motors were standardized in multiple columns, the

maximum output of the system for power and torque was derived by scanning all relevant fields, and the capacity, voltage, and chemistry of the battery were inferred using transparent rules driven by brand, model, hybrid type, production year, and version cues. This consolidation logic, including defaults and fallbacks, is summarized in Table 3.10.

Derived field	**Inputs scanned**	**Standardization or rule**
Electric Motor Type	Up to 4 electric motor type columns	Clean delimiters; standardize to Three Phase AC Induction Motor, AC Induction Motor, Electromagnet AC Motor, HSM; else join unique values
Max Horsepower	Conventional and electric power columns	Take maximum available numeric value
Max Torque	Conventional and electric torque columns	Normalize units then take maximum
Battery Capacity (kWh)	Battery capacity field and version string	Parse "kWh" from version; deterministic defaults for mild hybrids and non-electrified vehicles
Battery Voltage (V)	Battery voltage field	Default to 48 V for mild hybrids when missing; else 0.0
Battery Chemistry	Brand, model, hybrid type, year, version	BEV/PHEV assigned Lithium-ion; Toyota/Honda hybrids pre-transition use NiMH, post-transition use Lithium-ion; other brands mostly Lithium-ion post-2017; fallback to Lead Acid battery.

Table 3.10: Electric and hybrid attribute consolidation

Key performance fields were harmonized to support modeling and comparison. Acceleration metrics were consolidated into a single feature by setting 0 to 100 km/h values as the standard, while converting from 0 to 60 mph when necessary. The driving range metric was merged across test cycles with a priority classification strategy that is based on the vehicle's production year. The governing rules and notes appear in Table 3.11.

Feature	**Rule set**	**Notes**
Acceleration 0 to 100 km/h	Prefer 0 to 100; else 0 to 60mph × 1.04	Remove "<" and trailing "s"; cast to numeric
Range (merged)	Priority by production year across NEDC, WLTP, WLTC, EPA	Prefer WLTP after 2017; otherwise prefer NEDC; remove unit suffixes

Table 3.11: Performance field harmonization

Incomplete or missing fields were handled using a principled mixture of deterministic rules and imputation. Domain-specific features applied relevant defaults to preserve intent. For numeric features without such rules, a narrow, median-centered random imputation band preserved central tendency without overstating certainty. The contexts, methods, and rationales are summarized in Table 3.12.

Context	Method	Rationale
Numeric features without domain defaults	Sample uniformly around median	Preserve central tendency using a narrow band, avoid overconfident point imputation
Domain critical engineered fields	Deterministic rule based defaults	Predictable behavior and auditability for key features
Columns with no valid data	Leave unchanged and log	Failsafe behavior without speculative filling

Table 3.12: Missing data handling strategy

Finally, the dataset was prepared for modeling with clarity and consistency in mind. Meaningful ratios, such as power-to-weight, were introduced where they didn't exist, but were able to be calculated; categories that had a large amount of granularity were condensed to reduce resource requirements, whenever possible; names, types, and nullability were standardized across the feature matrix, and a schema mapping was generated for Snowflake to ensure consistent materialization in the analytical warehouse. The datatype correspondence used during schema creation is provided in Table 3.13.

Pandas dtype (string)	Snowflake type	Notes
object, string	STRING	Textual fields
int64, int	NUMBER or INT	Integer numerics
float64, float	FLOAT	Floating point numerics
bool	BOOLEAN	Logical fields
datetime64[ns]	TIMESTAMP_NTZ	Timestamps without timezone
timedelta[ns]	TIME	Duration like fields

Table 3.13: Datatype mapping for Snowflake schema materialization

3.6 Exploratory Data Analysis

The EDA phase involved analyzing and visualizing the scraped data to uncover key patterns and relationships, and assessing which features most strongly influence the machine learning model predictions. Extracting these valuable insights was crucial for the remainder of the development cycle.

3.6.1 Dataset Profiling

The EDA began with systematic profiling of the raw and refined datasets to capture their structural characteristics. The first steps involved obtaining the dataset dimensions and variable datatypes, together with obtaining unique value counts per column to assess granularity and coverage.

Descriptive statistics were created for numerical and categorical features to establish baseline data distributions and detect irregularities/outliers in the data. Examples of this type of EDA is illustrated in Figure 3.11.

```
[63]: df.describe(include = "all")
```

Out[63]:

	Rank	Sales	Average Check	State	Meals Served
count	100.000000	1.000000e+02	100.000000	100	100.000000
unique	NaN	NaN	NaN	19	NaN
top	NaN	NaN	NaN	N.Y.	NaN
freq	NaN	NaN	NaN	21	NaN
mean	50.500000	1.783343e+07	69.050000	NaN	317166.660000
std	29.011492	5.010408e+06	34.735181	NaN	192211.390011
min	1.000000	1.139168e+07	17.000000	NaN	87070.000000
25%	25.750000	1.409484e+07	39.000000	NaN	189492.500000
50%	50.500000	1.730078e+07	65.500000	NaN	257097.000000
75%	75.250000	1.990392e+07	95.000000	NaN	372079.000000
max	100.000000	3.908034e+07	194.000000	NaN	959026.000000

Figure 3.11: Example of dataset profiling, using describe, from Pandas, in a Jupyter Notebook

3.6.2 Missing Data Quantification

A thorough quantification of missing values was performed, feature-wise, calculating counts and proportions of absent data points. Visual tools such as horizontal bar charts were employed to differentiate features with low missingness, which required routine treatment from those with substantial data absence, potentially meriting exclusion or specialized imputation strategies. An example of this implementation is in Figure 3.12.

```
[20]: missing_count = df.isnull().sum() # the count of missing values
value_count = df.isnull().count() # the count of all values
missing_percentage = round(missing_count / value_count * 100,2) #the percentage of missing values
missing_df = pd.DataFrame({'count': missing_count, 'percentage': missing_percentage}) #create a dataframe
print(missing_df)
```

```
           count  percentage
title          0        0.00
score          0        0.00
id             0        0.00
url            0        0.00
comms_num      0        0.00
created        0        0.00
body       18134       49.45
timestamp      0        0.00
```

Figure 3.12: Example of logging missing values from each column of a dataset

3.6.3 Variable Uniqueness and Distributions

The uniqueness of the different variables was measured by performing counts of distinct values per feature, exposing overly granular categorical variables. Histograms and boxplots were generated for numeric columns to inspect distribution shape, central tendency, and outlier presence. Categorical variables were analyzed for level frequencies to inform grouping and encoding decisions during feature engineering, as exemplified in Figure 3.13

Name	Age	Location
Mark	27	USA
Juli	31	UK
Mark	31	NaN
Kevin	25	UK

Unique Value DataFrame

	Name	Age	Location
Total	4	4	3
Uniques	3	3	2

Figure 3.13: Example of counting unique values from each column of a dataset

3.6.4 Statistical Evaluation of Categorical Features

To evaluate the predictive relevance of categorical variables, statistical tests such as ANOVA F-tests were performed. These tests compared group-wise means of the target variable across categorical levels, allowing the identification of features with significant discriminatory power to be retained for predictive modeling. An implementation, as example, of a One Way test is outlined in Figure 3.14.

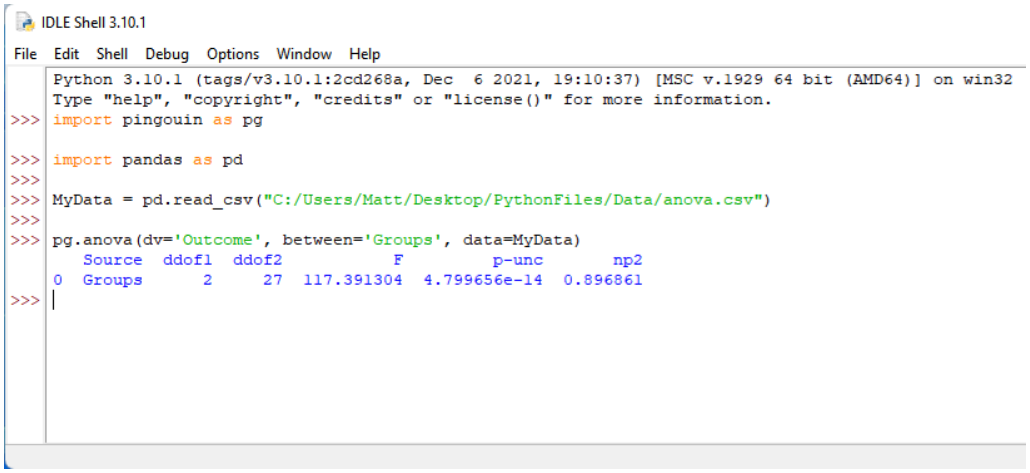

Figure 3.14: Example of ANOVA one way test, capturing the impact of categorical features, through hypothesis testing

3.6.5 Correlation and Multicollinearity Analysis

For numerical features, the coefficients of correlation were calculated to detect multicollinearity, a condition that may affect negatively the model's performance and interpretability. A Heatmap of correlation matrix provided a visual summary, highlighting strongly correlated variables that could be candidates for transformation or removal, as exemplified in Figure 3.15.

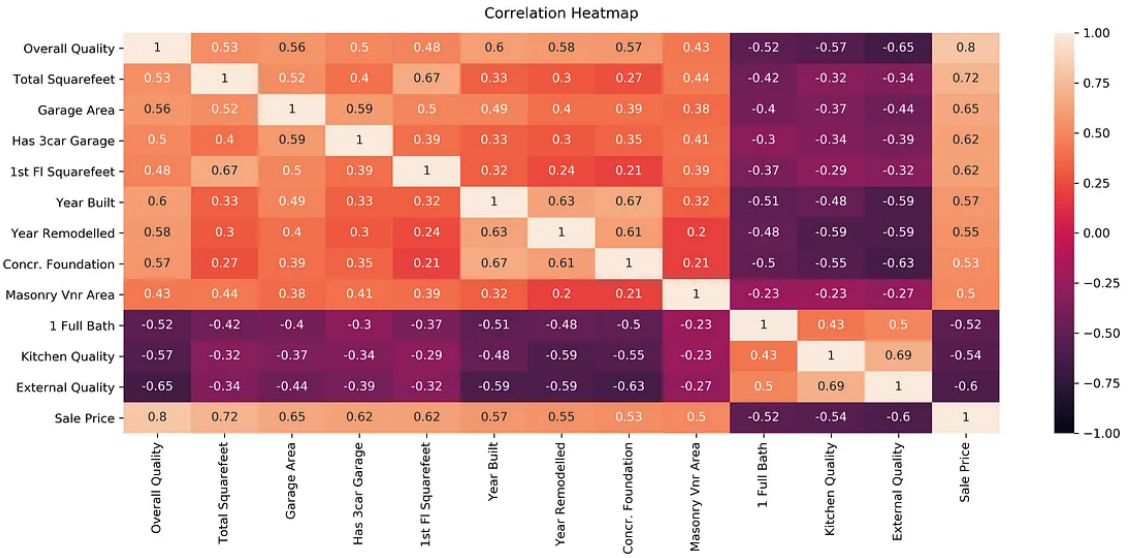

Figure 3.15: Example of a Correlation Matrix, capturing the impact of numerical features on other features

3.6.6 Outlier Detection and Treatment

Critical numeric features underwent visual inspection through boxplots and scatterplots to identify outliers and unusual patterns. Contextual judgment informed decisions on outlier treatment, including winsorization or exclusion, balancing data integrity with robust model training, as can be seen in Figure 3.16.

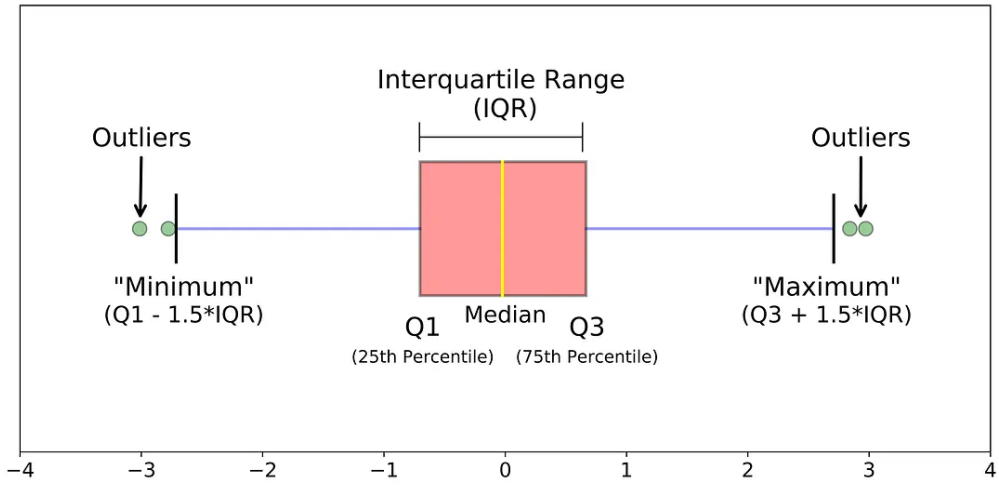

Figure 3.16: Example of a Boxplot, used to detect outliers in the data

3.6.7 Visualization and Documentation

During the EDA phase, different visualizations were used to help understand and document the data analysis process and results obtained. Bar charts, histograms, heatmaps, and boxplots were

used to gather different insights withing the data set and drive the next step.

Bar charts were used to analyze categorical data and compare how different groups behaved. Histograms helped visualize how numerical values were spread out, revealing where most records were concentrated and where the extremes were located. Heatmaps gave an overview of how variables interacted with each other, helping to spot strong relationships or lack thereof. Boxplots added another layer of analysis by showing how values varied inside each feature, marking the median, range, and any unusual observations that stood out.

All these visuals together gave a clear picture of the dataset and made it easier to check that results stayed consistent. They also helped share the findings more effectively and made the patterns emerging from the cleaned data easier to interpret. Figure 3.17 shows examples of the visual forms used in this phase.

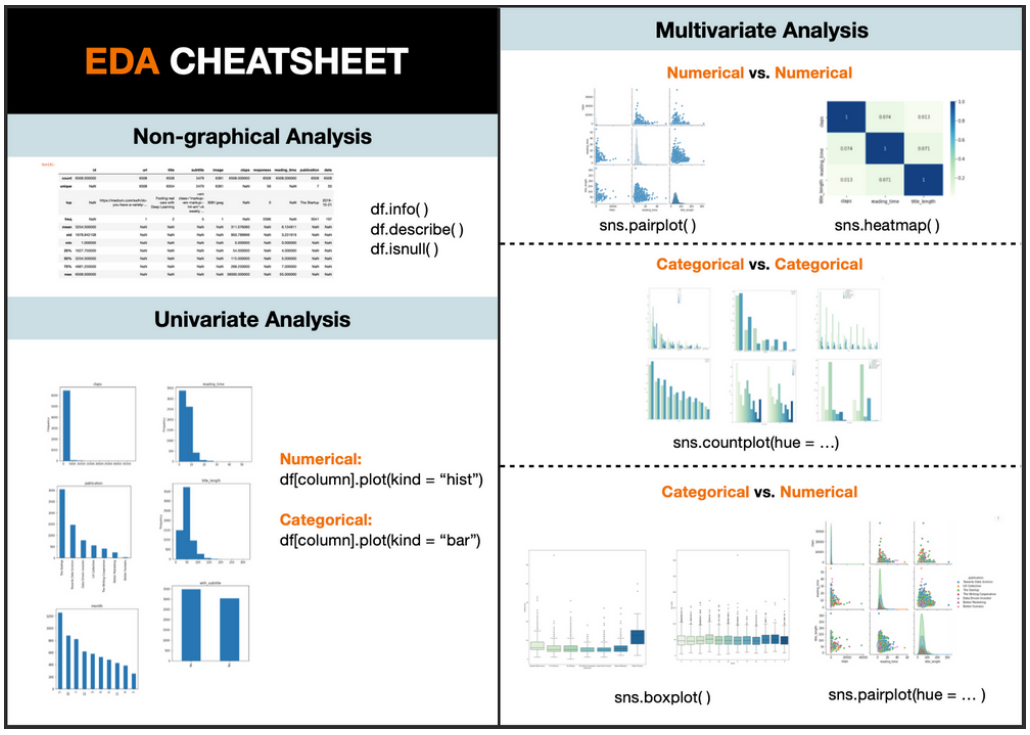

Figure 3.17: Cheatsheet of charts used for different types of analysis [44]

3.6.8 Summary

Following the analysis, features were selected for inclusion in the ML layer. This selection was based on criteria applied through both ANOVA and correlation tests to evaluate the statistical significance and domain relevance of each column.

Only features showing meaningful impact were kept, ensuring that the final dataset powering the ML Layer balanced minimalism with reliability.

3.7 Machine Learning Modeling Strategies

Following the creation of the final layer of the dataset, the ML layer, the subsequent phase involved building machine learning models to estimate fuel consumption. These models leverage both technical vehicle specifications and driver-related information as inputs. Based on established literature and state-of-the-art practices, three modeling approaches were selected for evaluation: NN, RF, and XGBoost.

To systematically compare their performance, dedicated implementations of each method were developed and applied to the ML layer dataset. This setup allowed for an empirical investigation of predictive accuracy, scalability, and suitability to address this dissertation's main objective.

3.7.1 Neural Networks

The first ML algorithm implemented was an NN, making use of custom multilayer perceptron models with categorical embeddings for categorical columns, with the advantage of being able to capture complex nonlinear relationships, which was evidenced in Chapter 2.

Optuna [95] was used as a guide for hyperparameter tuning, to find the best combination, by way of a study, going over the dataset multiple times testing different combinations to find the best one. Regularization techniques such as batch normalization and dropout were included to prevent overfitting, following current best practices in NN design [41].

Unlike the previous dissertation, which used more traditional network structures [102], this dissertation adopted a completely new approach for NNs, using more modern techniques and without reusing earlier code.

The NN model development used different components that worked together to ensure flexible, efficient, and successful training. Optuna automated the search for the best hyperparameters, including the number and size of layers, the dropout rates, the learning rate, the optimizer choice, the activation functions, and the batch size, using optimization and pruning methods to find the best performing configurations.

The dataset was then wrapped in custom PyTorch Dataset and DataLoader classes [106] that organized the numerical features, categorical embedding inputs, and target values into tensors suitable for model training.

PyTorch's DataLoader also managed batching, shuffling, and parallel data loading during training and evaluation, which improved the convergence speed and model stability.

When put together, these components formed a robust framework for training NNs costumized specifically to estimating fuel consumption, balancing flexibility, efficiency, and performance.

3.7.2 Random Forests

The RF model was used as a reliable baseline due to its robustness against noise, ability to handle complex feature interactions, and ease of interpreting feature importance, as reviews in Chapter 2.

This implementation was built upon the RF approach developed in the earlier project [102], serving as the foundation that was further refined in this dissertation to improve predictive accuracy and generalization.

Data was loaded from the Snowflake warehouse and preprocessed. Categorical features were encoded using ordinal encoding for compatibility with the model and numerical features and the target were scaled using MinMax normalization to stabilize training.

The dataset was split into training and testing subsets using a 70 to 30 split, as per industry standards. To evaluate model consistency and performance more robustly, repeated K-fold cross-validation was applied across multiple data folds. The RF was configured with 300 trees and a maximum depth of 12, selected to balance complexity and prevent overfitting.

After training on the training set, predictions were generated on the test set and then transformed back to their original scale for evaluation. Model performance was assessed using MAE, MSE, and R^2, quantifying both prediction error and explained variance.

This refined RF approach provided a strong, interpretable predictive model that extended and improved on the methodology initially developed in the previous work.

3.7.3 Extreme Gradient Boosting

The XGBoost model was used due to its efficiency, scalability, and strong predictive performance in regression tasks, as evidenced by the literature review. This approach builds upon the XGBoost implementation from the previous project [102], where it served as an initial framework.

In this dissertation, the earlier implementation was re-evaluated and adapted to handle updated and expanded datasets.

Data was loaded from Snowflake and preprocessed by separating features and the target variable. Categorical features were encoded using an ordinal encoder with provisions for unknown categories, while numerical features were optionally scaled using MinMax normalization, ensuring compatibility without negatively impacting model efficiency.

The model was tuned using a grid search together with repeated K-fold cross-validation. Using grid search to test different combinations of parameters, such as learning rate, maximum tree depth, minimum child weight, gamma, subsample ratio, and column sampling rate allowed the model to be as performant as possible, finding the right balance between complexity and generalization, while preventing both over and underfitting.

Repeated K-fold cross-validation was used to estimate model performance across different data splits and by running several rounds of training and testing, it gave a better sense of how the model would behave on unseen data. Evaluation focused on three key metrics: MAE, MSE, and R^2. The hyperparameter search aimed to minimize prediction errors (MAE and MSE) while maximizing R^2. The best parameters found by this process were then used to train the final model on the training split.

Finally, the trained model was evaluated on a test set with MAE, MSE, and R^2 metrics confirmed to exhibit improved accuracy relative to untuned models. The model benefits from GPU acceleration for faster training and efficient processing of large datasets.

Logging throughout the procedure tracked data loading, preprocessing, hyperparameter tuning, and evaluation steps, ensuring transparency and reproducibility.

This refined and revalidated XGBoost approach delivers a high-performance, scalable system for predicting vehicle fuel consumption, extending the foundational work undertaken in the previous project.

3.8 Summary of Key Technological Decisions

In table 3.14 an overview of the decisions made that shape the final solution for this dissertation is presented, highlighting the reasoning for each one of the choices:

Component	Technology	Justification
Web Scraping	Scrapy + Zyte	Superior for large-scale static data scraping with proxy support, ensuring throughput and reliability
Cloud Data Warehouse	Snowflake	Decoupled architecture, cross-cloud support, low maintenance overhead
Data Augmentation	Python	Complements real survey data to model driver heterogeneity
Feature Selection	ANOVA, Corr.	Ensures model parsimony and interpretability
ML Models	NN,RF,XGBoost	Captures complex relationships with trade-offs between accuracy and interpretability
User Interface	Streamlit	Rapid development of interactive apps with minimal overhead

Table 3.14: Decisions Made Throughout The Dissertation

Additionally, the architecture diagram for this implementation is detailed in Figure 3.18. It represents the connections between the different elements, as well as the steps that the data goes through from ingestion to consumption.

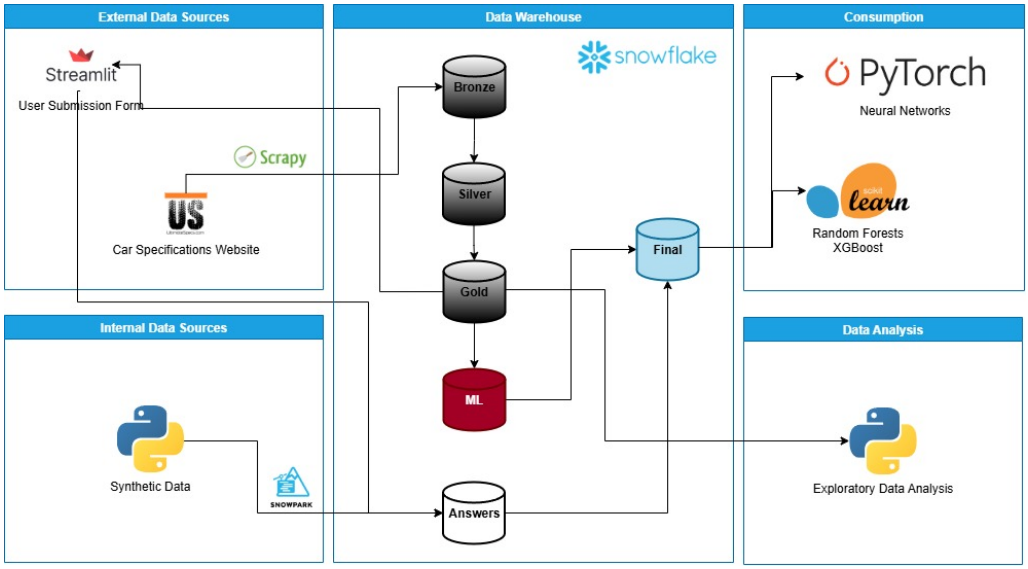

Figure 3.18: Architectural diagram of the project

3.9 Dissertation Timeline

The dissertation went through a sequence of phases starting in October of 2024 until of September 2025. This timeline aims not only to reflect planned activities but also the problem-solving iterations and techniques, as well as technical refinements necessary to build the vehicle data extraction pipeline, processing, and machine learning models. A visual representation of the progress and task time within the dissertation can be viewed in Figure F.1.

3.9.1 Planning and Initial Setup (October 2024)

The initial step focused on laying the groundwork, which consisted in creating clear research goals and selecting the ideal tech stack. A private GitHub repository was created to manage the codebase and documentation, and access to it granted to the supervision team.

At the same time, a development environment was configured, through a Python virtual environment to handle dependencies and ensure consistency across any machines that run the project's code.

During this phase, the first challenge was encountered, regarding dependency management. Conflicting library versions, especially within web scraping and data science packages, required careful resolution through version pinning and environment isolation. Since the goal was to have the dissertation replicable with minimal setup, this was a crucial step to get right.

3.9.2 Scraping Technology Selection and Pipeline Development (October 2024 to February 2025)

A review of web scraping frameworks was done to identify tools capable of handling large volumes of structured vehicle data that was contained within the website. From the alternatives, Scrapy was selected due to its asynchronous architecture, mature community support, and customizability, as evidenced in 2.

Originally, the data scraping pipeline was built to work in a hierarchy of operations: extract car brands, then models, generations, versions, and ultimately the detailed specifications. This multi-layer approach, while logically straightforward, posed considerable operational strain due to the expansive number of pages and requests involved.

Significant difficulties became evident early on:

- Website Anti-Bot Measures: The source website employed sophisticated defenses including IP blocking and Captcha challenges. This caused frequent interruptions requiring the scraper to be paused or restarted.
- Complex Page Structures: Vehicle specifications were presented in horizontally structured tables and varying HTML formats, complicating automated extraction logic.
- High Volume of Requests: The tiered scraping method resulted in a very large number of HTTP requests, increasing detection risk and prolonging extraction time.

To overcome these barriers, multiple tactics were used, in the form of random delays and retry strategies programmed into the spider to minimize detection, and incremental data saving, exporting results progressively to CSV files. This mechanism proved critical in resuming extraction without redundant processing after interruptions or losing progress due to external factors.

When analyzing the website for models that weren't appearing in the extracted results, there was an accidental discovery of a simplified scraping path allowing the spider to bypass intermediate steps (models and generations) by directly accessing versions and details pages by modifying URL patterns. This change reduced requests needed by more than 99%, dramatically improving both the extraction speed and the spider's reliability, minimizing risks and potential costs.

3.9.3 Refinement of Extraction Process and API Integration (February to April 2025)

Despite the new process and optimizations, constant blocks and the site's escalating anti-automation tactics required another solution. After research and consideration, it was decided to integrate the third-party scraping API Zyte, which provided proxy rotation, anonymization, and effective masking of automated requests, into the scraping pipeline, to overcome these issues.

This integration brought new challenges: spider requests had to be adjusted to work with Zyte's asynchronous API interface, and using the free trial provided by the platform, while maximizing results, required careful planning. In addittion to this, retry logic and error handling were incorporated to guarantee long extraction runs completed successfully, even when API quotas were reached.

3.9.4 Data Warehouse and Web Application Development (March to May 2025)

Simultaneously with the extraction improvements, the dissertation required analysis of different cloud warehouse providers, in order to have the base information securely available in the cloud, instead of flat files. Snowflake was selected for its elasticity, broad Python integration, and a free-trial framework allowing large workloads without any costs.

Security requirements and best practices guided the design of database schemas and user roles, which were created before starting the intensive data ingestion phase. The data warehouse allowed for the complete dataset to be stored and queried efficiently, serving as the foundation for all subsequent steps.

In parallel, to start collecting organic fuel consumption data, the development of the web application, in the shape of a Form, using Streamlit, started, after researching possible alternatives. The app provided an intuitive interface for drivers to submit real-world fuel consumption data tied to their vehicles. This required close coordination with backend data schema developments to ensure the app's search filters and input forms remained compatible with evolving data, as well as enhancing UX.

3.9.5 Data Cleaning and Multi-Layer Data Transformation (April to July 2025)

The raw scraped dataset displayed a high rate of null values, redundant fields, and inconsistent formats, necessitating a thorough data cleaning pipeline.

- The Bronze layer preserved data in its original, minimally processed form to maintain traceability.
- The Silver layer involved extensive transformations including unit standardization (e.g., metric conversions), column pruning, imputation of missing data, and harmonization of categorical variables based on domain expertise.
- The Gold layer utilized advanced SQL processing in Snowflake, applying window functions and median imputation within product groups to generate a normalized, enriched dataset ready for analytical and predictive modeling purposes.
- The ML layer utilized ANOVA tests and Correlation Tests to extract from the finalized dataset the most influential fields, further enhancing the impact of the data, while separating the dataset destined for consumption by the Web App and exploratory data analysis, and the one fit for the ML models.

This medallion-style approach ensured that data moving forward was consistent and accurate, separating the different stages of the dataset.

3.9.6 Synthetic Data Creation and ML Preparation (July 2025)

The limited number of user responses from the web form warranted the creation of synthetic data to improve the training and performance of the ML models.

Using statistical distributions retrieved from demographic and behavioral studies, synthetic profiles were created to reflect realistic ranges of age, gender, driving style, and driving context, adjusting fuel consumption values with a behavior-based multiplier to influence the final value. This significantly increased the size of the dataset, improving both the diversity and robustness of the training samples.

At the same time, statistical tests such as ANOVA and correlation analysis were applied to identify features with meaningful predictive power, to refine the final column list present in the dataset powering the ML models.

3.9.7 Modeling, Insights, and Finalization (July to October 2025)

The modeling phase started with the development of the regression models (NN, RF and XGBoost) which were then tuned and optimized through their own parameter adjustments and to maximize performance.

The resulting predictions and metrics were analyzed in detail. The insights gained from this analysis provided strong support for the dissertation's hypotheses and contributed to meaningful

conclusions. The dissertation concluded with thorough documentation and deployment for ongoing evaluation.

3.9.8 Final System Architecture

The culmination of the methodology is embodied in the dissertation's architecture, which unites data ingestion, transformation, storage, analysis, and machine learning into a streamlined and reproducible workflow. The architecture is visualized in Figure 3.18, and its components reflect the goals of modularity, scalability, and robust integration of both organic and synthetic data sources.

3.9.8.1 Data Sources

The system utilizes data from both external (scraped) and internal (generated) data sources.

External Data Sources include both the Streamlit user submission form, which gathers real-world usage data and feedback directly from users, and Scrapy, which extracts vehicle specifications from the chosen website to build the technical specifications dataset.

Internal Data Sources consist of a Python script that generates synthetic data, supplementing the organic user responses and enabling scenario modeling for enhanced analysis.

3.9.8.2 Data Warehouse: Snowflake & Medallion Architecture

Data flows into the Snowflake data warehouse, structured around the medallion architecture:

- Bronze Layer: Stores raw ingested data from Scrapy with minimal changes, preserving original fidelity for traceability.
- Silver Layer: Contains cleansed, validated, and harmonized data after deduplication and schema alignment.
- Gold Layer: Houses refined dataset optimized for analytics.
- ML Layer: Supports the preparation of the dataset for use in machine learning models, after removing less influential features.
- Answers Layer: Contains user responses and synthetic data generated.
- Final Layer: Delivers the curated, consumption-ready data for downstream applications. Contains the ML and Answers layers.

3.9.8.3 Consumption & Analysis

The consumption and analysis step was a key component of the system, combining EDA and ML model development to transform raw and processed data into precise fuel consumption estimations.

In the EDA stage, pandas, seaborn, and matplotlib were used to explore the dataset structure, visualize distributions, and detect important relationships or anomalies within the vehicle details and user values.

The main steps involved creating tables for summary statistics, histograms and boxplots to analyze the values of certain numerical features, creating heatmaps to identify correlations in the data, and performing ANOVA tests to evaluate the statistical influence of categorical variables.

After the EDA, three different, state of the art, ML algorithms were implemented and compared to determine their performance in predicting real-world fuel consumption values:

- A PyTorch-based NN framework was used to construct multilayer perceptrons integrating categorical embeddings, batch normalization, and dropout layers, allowing the model to capture nonlinear dependencies among the features.
- The RF model acted as a strong ensemble benchmark, taking advantage of its interpretability, resistance to overfitting, and built-in feature importance metrics to establish a reliable baseline.
- XGBoost was selected as the main model due to its scalability and high efficiency, using gradient-boosted trees with tuned hyperparameters to maximize prediction accuracy.

Training, validation, and testing were performed on the Snowflake ML layer dataset, and results were analyzed using the aforementioned success metrics. The integration of EDA with these models provided a complete process capable of balancing accuracy, interpretability, and scalability.

As a result, this DSS handled data ingestion, transformation, storage, and estimation cohesively, allowing for continuous development and performance improvement.

3.10 Conclusion

In conclusion, this approach represents a significant upgrade over the previous work. The data is now extracted dynamically, and there are features that allow it to continue expanding, as new cars are added to the website, the data is now stored in the cloud, and more features exist, to further enhance the ML models, allowing performance that was left on the table to be utilized.

The choice of models was deliberate, based on previous results, variety, but most importantly, according to the state of the art technologies, to determine, in this case, which one performs the best given the features and test conditions. The result is a complete system that handles ingestion, transformation, storage, analysis, and estimation, working in unison to provide as accurate as possible results on fuel consumption values.

With the methodology and architecture now established, the next steps are to evaluate the effectiveness of the DSS. The next chapter will present and analyze the results from both datasets and assess model performance across the success metrics and present two test cases that aim to confirm the tool's success. This discussion will highlight the strengths and limitations of the approach taken, providing insights on future applications.

Chapter 4

Chapter 4 - Results and Discussion

4.1 Introduction

This chapter presents the results obtained from implementing the fuel consumption prediction platform and provides a critical discussion of the findings. The objective is to evaluate how effectively the proposed solution addresses the research questions and objectives established in previous chapters. The results focus on the models developed (NNs, RF, and XGBoost) along with the evaluation metrics used, showing their predictive capabilities and performance characteristics on the dataset used in this study.

The chapter begins with an overview and analysis of the datasets utilized, displaying its composition and the most influential features, followed by a detailed evaluation of the performance of each ML model used, through by quantitative metrics and relevant visualizations. The conclusions are then synthesized and interpreted according to the research goals, with special attention to the model's ability to deliver practical, reliable estimates of real-world fuel consumption. Following the overview, two test cases are analyzed to show practical application of the ML models to vehicles present in the user survey. The goal is to apply them not only to the validation part of the original dataset, but to a random query, simulating real world use. The chapter finishes with a discussion of the broader implications these findings have for consumers, policy makers, and future research this domain.

4.2 Data Extraction and Ingestion

The crawl completed with 50,697 unique pages retrieved. There was a planned pause at 40,000 pages to switch the provider trial, after which the process resumed and finished without incident. The total elapsed time was 20 hours, which yields an overall mean throughput of approximately 42 pages per minute.

Throughput began at about 300 pages per minute and was progressively throttled to reduce the likelihood of blocks and bans over a long run. Four temporary bans were detected during the run, rotation and retry logic were engaged, and operational downtime was kept very close to zero.

Throttling, IP rotation, and automatic retries were managed by Zyte, which minimized manual anti-ban engineering and accelerated development by allowing effort to focus on parsing and quality assurance. The complete set of information is showcased in Table 4.1

Metric	Value	Notes
Pages retrieved (unique)	50,697	Final count after recovery of transient failures
Planned pause	At 40,000 pages	Provider trial switch, then resumed
Total elapsed time	20 hours	End to end, including pause and retries
Average throughput	$\tilde{4}2$ pages/min	Overall mean across the full run
Initial throughput	300 pages/min	Throttled gradually to reduce blocks and bans
Temporary bans	4 events	Rotation and retry eliminated practical downtime
Pages re-extracted	5	All recovered on retry, zero permanent failures
Operational downtime	Virtually 0	Managed through automated rotation and backoff
Anti-ban management	Provider managed (Zyte)	Throttling, IP rotation, and automatic retries handled externally
Development impact	Reduced overhead	Less custom anti-ban code, faster delivery of parsing and QA
Net success on target list	100%	All intended pages present in the final set

Table 4.1: Data extraction KPIs with provider-managed throttling, rotation, and retries

4.3 Data Overview

To accurately interpret the results presented in this chapter, a comprehensive understanding of the underlying datasets is essential. This requires a thorough and critical examination of the data to identify meaningful patterns and extract insights that reflect the current state of the automotive industry and driver behavior.

Each dataset offers unique and valuable information that directly influences the performance of the predictive models and shapes the final outcomes in different ways. Therefore, analyzing the data prior to creating the models is required, to develop reliable predictions and draw well-founded conclusions.

4.3.1 Car Details Dataset

4.3.1.1 Bronze Layer

Analyzing the raw data shows important information that is necessary to understand and process the data effectively. This layer, made up of the data as it was extracted from the website, contains

50,565 records and 163 columns, primarily of object data types, using 63 MB of memory.

The first observation is the variability in data completeness between columns. The most important columns (such as ID, URL, Years, Model, and Brand) are fully populated, acting as reliable identifiers for each record; however, many technical columns, particularly those related to electric engine specifications, show a substantial amount of empty values, with some columns missing data for nearly all entries.

This disparity indicates that the data source is missing critical information for many, and sometimes, all of its models. As shown in Figure 4.1, out of the 163 columns, only 24 have less than 10% missing values, while 109 columns have over 90% of their values empty. In Figure 3.12 are the columns with more than 90% missing values.

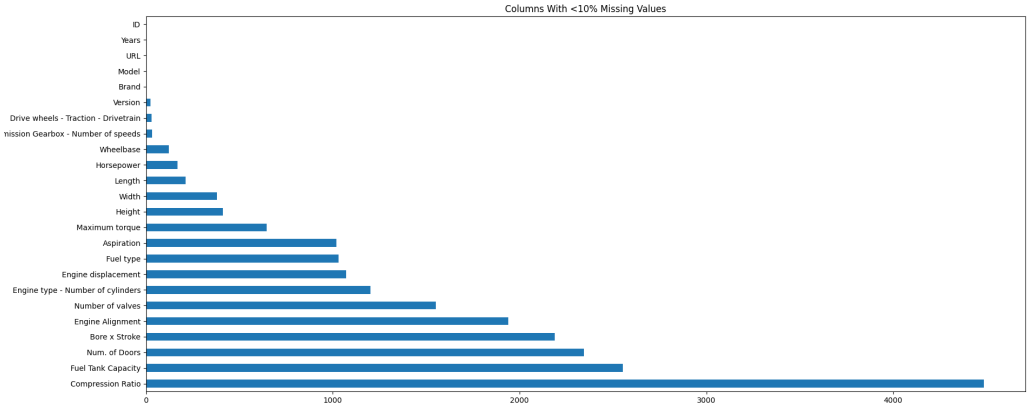

Figure 4.1: Raw Columns with Less than 10% Missing Values

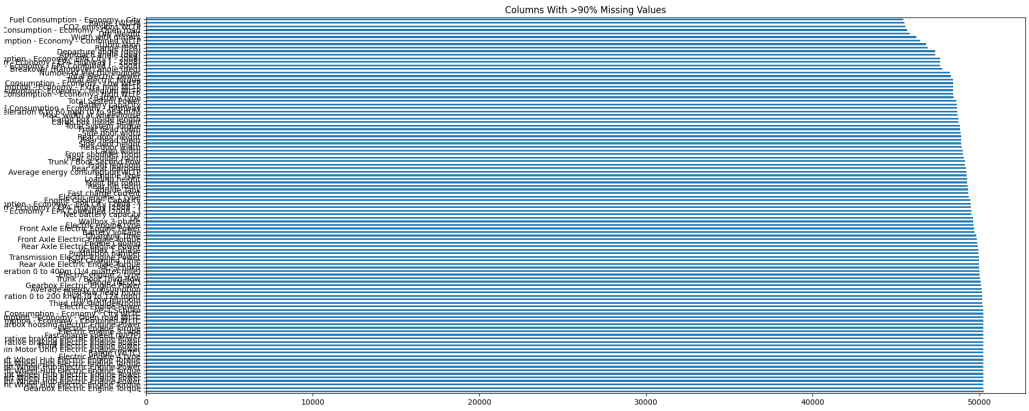

Figure 4.2: Raw Columns with More than 90% Missing Values

To showcase which columns contain the most varied information, Table 4.2 shows the columns with the highest number of unique values, such as “Model”, “Engine Code”, and “Version”. Table D contains a complete overview of every column and its granularity. This table provides a full breakdown of the raw data’s structure and level of detail. For example, the “Version” column is sometimes missing; for car models with only one version, it was set to the model name, so no information was lost. The amount of missing data in certain columns made extensive data cleaning and transformation necessary before being ready for analysis.

Column	**Unique values**
URL	50246
ID	50246
Version	25814
Model	5584
Euro NCAP	5329
Engine Code	3411
Bore x Stroke	2540
Lubrication	2339
Transmission Gearbox - Number of speeds	2247
Generation	2150

Table 4.2: Top 10 most granular features (by unique values)

In this layer, the only two transformations performed were removing entries where the vehicle had no valid information on the Production Dates, given that cars without this important information weren’t suitable for analysis. This resulted in the removal of 319 rows, as shown in Listing 4.1. The other transformation was the extraction of unique vehicle identifiers (IDs) from each car’s URL column, done using regular expressions specific to the URL formats, into a column called ID. This made the next step simpler, which was deduplication. Using database best practices, duplicate records weren’t allowed and were removed, deleting a total of 41 entries, as detailed in Listing 4.2.

```
1 INFO - Removed 319 rows with empty Production Dates
```

Listing 4.1: Cleaning log

```
1 INFO - Silver Layer Initial Row Count: 50287
2 INFO - Silver Layer Row Count After Deduplication: 50246
```

Listing 4.2: Deduplication log

4.3.1.2 Silver Layer

One of the techniques used to create the Silver Layer involved creating a separate mapping file that links specific vehicle models to their body types whenever this information was missing in the source table, which is displayed in detail in Appendix E. Although it does not cover all of the models, it represents a broad completion of what exists in the database.

Another technique that was used to create this layer was inference. Many columns had different, but fundamentally similar, values for the same category. For example, the column "Aspiration" had values such as "Turbo", "Naturally Aspirated", "Supercharged", but also had "Garrett", for example, which is a Turbo manufacturer, so a number of helper functions were created to engineer these features, with the goal of homogenizing the dataset. The result of this inference, for example, resulted in a significant reduction of different options, reducing the ML dataset size considerably, as can be seen in Listing 4.3.

```
1 INFO - Aspiration Unique Values Before Simplification: 174
2 INFO - Aspiration Unique Values After Simplification: 10
```

Listing 4.3: Aspiration inference results

Columns with many missing values, particularly those related to electric or plug-in vehicles, needed careful handling. Strategies such as imputing missing values, selectively excluding certain columns, or treating them individually were used to reduce their impact on later analysis and keep the database manageable.

For example, the original dataset included 21 columns for Fuel Consumption, each reflecting a different fuel consumption testing standard. To ensure that both the database and the data model remained accurate and comprehensive, these were consolidated into only 2 columns, using custom logic to match each car with the correct test procedure and to estimate combined consumption values when only city and highway figures were available, ensuring a unified target variable, thus preserving both variability and authenticity. This allowed context-aware estimating, by identifying the reported fuel consumption and its source.

Other examples of the different transformation steps taken contain:

- Unit conversions: Vehicle dimensions originally recorded in imperial units (inches, pounds) were converted to metric equivalents (meters, kilograms) for standardization, using custom conversion functions *(convert_inches_to_meters, convert_lbs_kg)*.

- Power and torque extraction: The numeric horsepower and torque values were extracted from free text columns, converting torque units from Pound-Force Foot (LB-FT) to Newton Meter (NM).

- Categorical simplification: Complex technical terms (such as fuel systems) were normalized by rule-based parsing to reduce sparsity and improve the interpretability of the model.

```
INFO - Fuel System Unique Values Before Simplification: 1828
INFO - Fuel System Unique Values After Simplification: 9
```

Listing 4.4: Fuel System simplification results

- Battery and electric motor data consolidation: Combined multiple attributes of electric motors into unified features, filling missing battery capacity and voltage data based on hybrid or electric vehicle classification.

4.3.1.3 Gold Layer

The Gold Layer brings together all the transformations into a single, analysis-ready view and represents the main source for modeling. In this layer, all necessary columns are enforced as non-null for modeling purposes (for example, horsepower, torque, fuel_consumption, and range). Auxiliary columns are standardized and completed wherever possible, and the schema is defined with explicit data types and column descriptions to ensure stability and follow best practices. This layer was built using a SQL query, shown in Listing F.1, which both defines the table and populates it from the Silver layer. The following steps were taken:

- `CAR_SPECS_GOLD` was created with explicit numeric precisions and column comments to clearly document units;
- Only rows with non-null values for `fuel_consumption_combined`, `torque`, `range`, and `horsepower` were included, while extreme `displacement` outliers were removed;
- Window functions with `IGNORE NULLS` were used to propagate geometry values, keeping them consistent across different versions of the same model;
- Median values for height and top speed were used to backfill missing data, reducing nulls and minimizing bias;
- Defaults were applied using `COALESCE` (for example, body set as Undefined, electric engine type as None) together with domain rules (e.g., *engine_type* set to Boxer for Porsche and Subaru, doors defaulted to 3). Derived indicators such as *hp_kg_ratio* were also calculated to support modeling.

Table 4.3 contains a comparison between the different dataset structures, for both the Raw Layer and the Gold Layer.

Aspect	Raw (Bronze)	Gold
Rows	50,565	38,780
Columns	163 (heterogeneous, sparse)	44 (typed, standardized)
Required columns enforced	No	Yes (HP, torque, combined consumption, range)
Units and rounding	Mixed	Harmonized scales and rounding
Deterministic backfilling	Minimal	By-model propagation + median backfill
Derived targets/ratios	Not consolidated	Combined target + hp_kg_ratio
Vocabulary normalization	Limited	Engine type rules; body/gearbox defaults
Schema documentation	None	Column comments and stable types

Table 4.3: Dataset composition comparison between the Raw (Bronze) and Gold layers

The numerical distribution chart shows a concise view of the tendency and dispersion of the central data after the transformations, and allows rapid inspection without resorting to per-feature tables. Many variables, such as weight, length, rear axle track and the hp / kg ratio, display approximately symmetric unimodal shapes that are consistent with near-normal behavior.

In contrast, Production Start is negatively skewed, with density concentrated in recent years, which indicates an acceleration in model introductions. Engine displacement is positively skewed, since observations cluster at small to mid-size engines and a long right tail captures large displacement vehicles; this pattern is compatible with downsizing under emissions constraints.

CO_2 emissions present a similar positive skew, with most vehicles below about 200 g/km and fewer high emission cases in the upper tail. Several remaining variables are multimodal or heavy-tailed, reflecting segment mixing and heterogeneous test practices. These results are detailed in Figure 4.3.

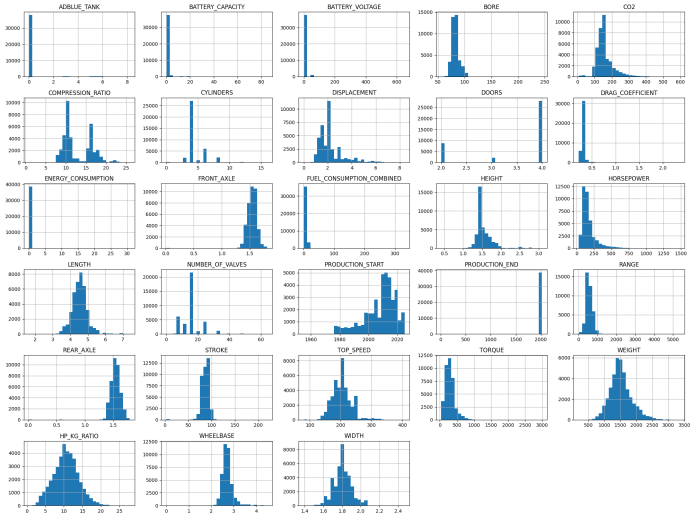

Figure 4.3: Gold Layer: numerical column distributions

Accordingly, boxplots 4.4 to 4.9 examine key columns in more detail, showing their trends, spread, and extreme values, in case there are any.

The Bore boxplot (Figure 4.4) shows a compact interquartile range, with most observations between approximately 77 mm and 86 mm, and a mean close to 83 mm; a small number of high values remain as outliers, which is consistent with performance-oriented engines or very antique ones.

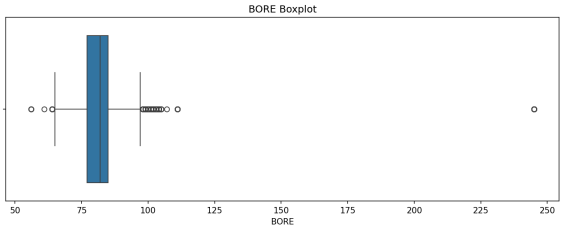

Figure 4.4: Gold Layer: BORE boxplot

The CO_2 boxplot (Figure 4.5) showcases a substantial dispersion, with its concentration between about 125 and 180 g/km, a mean near 150 g/km, and a long right tail which indicates high variance across the results.

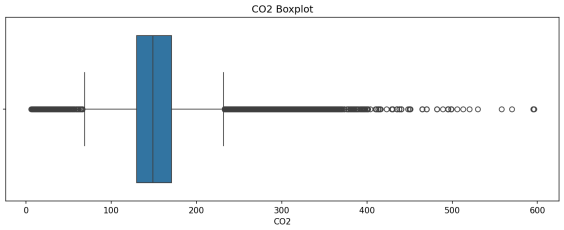

Figure 4.5: Gold Layer: CO_2 boxplot.

The Displacement boxplot (Figure 4.6) is right skewed, with a higher concentration of vehicles in the 1.5 to 2.5L range and a median close to 2.0L, while a thinner tail, in the form of outliers, captures larger engines. This pattern aligns with emissions driven engine downsizing and helps explain variance in fuel consumption.

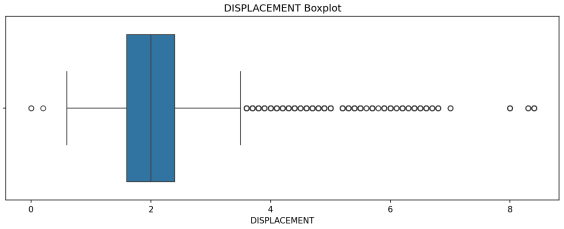

Figure 4.6: Gold Layer: DISPLACEMENT boxplot

The boxplot referring to the Top Speed centers around 200 km/h, while values above 250 km/h are sparse and characteristic of sports cars and hypercars, justifying the use of robust error metrics in the subsequent evaluation (Figure 4.7).

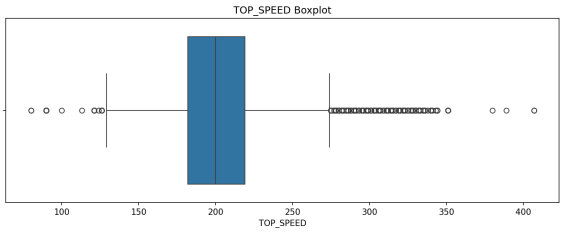

Figure 4.7: Gold Layer: TOP_SPEED boxplot

The Horsepower box plot (Figure 4.8) indicates that roughly seventy five percent of models are below 200 hp, reflecting the typical vehicle of the main car market and is consistent with lower fuel use and lower emissions, while the upper tail includes high-power sports cars and hypercars that can exceed 1,000 hp.

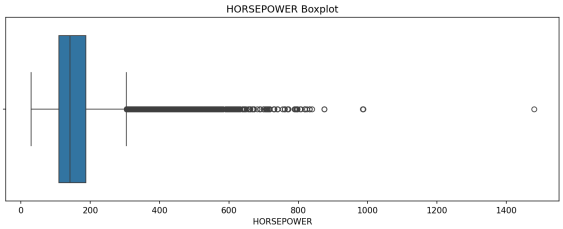

Figure 4.8: Gold Layer: HORSEPOWER boxplot

In the Fuel Consumption boxplot, most vehicles fall within the 5 to 10 L/100 km values, with an average close to 7.5 L/100 km. For robustness and noise in the evaluation, a small number of extreme values were retained.(Figure 4.9).

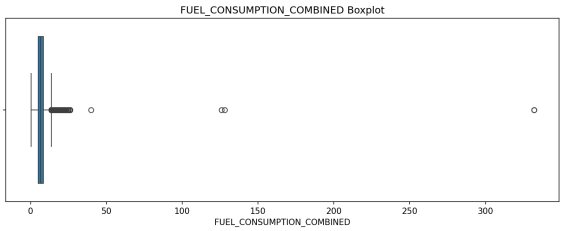

Figure 4.9: Gold Layer: combined fuel consumption (boxplot)

4.3.2 Form Responses Dataset

The dataset contains 125,020 responses. Only 20 of these are organic, while the rest were generated to complete the experimental design. Most respondents are female (68,058), and they most often report a Mixed driving context and an Average driving style. The mean fuel consumption across all responses is approximately 8.7 L/100 km, which is consistent with the midrange concentration visible in the distribution and bracket figures referenced. The most common vehicle is car_id 144966, a 2023 Mercedes-Benz GLA 250, which appears 12 times. Respondents have a median age of 31 years, an interquartile range of 24 to 43 years, and a mean of about 35.6 years,

with a median driving experience of 7 years and an interquartile range of 3 to 14 years. These results are present in Table 4.4.

Metric	**Value**
Total responses	125,020
Organic responses	20
Generated responses	125,000
Female respondents	68,058
Most common main context	Mixed
Most common driving style	Average
Mean fuel consumption	8.7 L/100 km
Most frequent vehicle (car_id)	144966
Vehicle description	2023 Mercedes-Benz GLA 250
Occurrences in responses	12
Age, median and IQR	31 years; IQR 24 to 43 years
Age, mean	35.6 years
Driving experience, median and IQR	7 years; IQR 3 to 14 years

Table 4.4: Summary of respondent profile and key consumption statistics from the form responses dataset

4.3.2.1 Fuel Consumption Distribution

Aggregating the target into operating ranges confirms that most responses fall within 5 to 10 L/100 km, with more than 40,000 reports in the 5–7.5 L/100 km band and a meaningful mass at or above 10 L/100 km, approximately 30,000 responses (Figure 4.10).

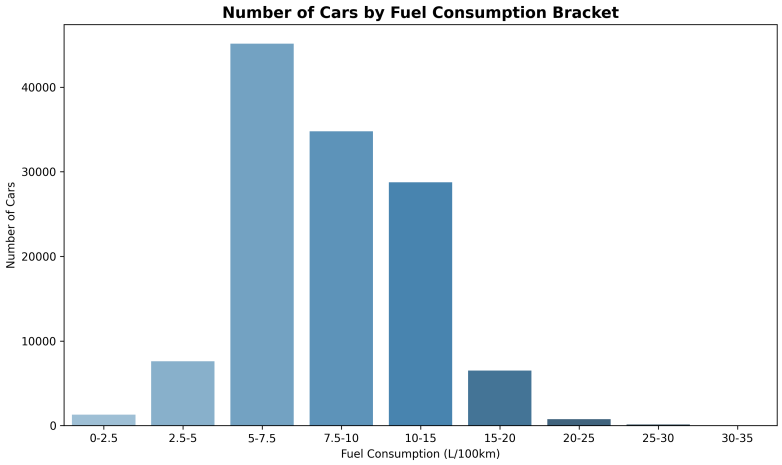

Figure 4.10: Number of responses by fuel consumption bracket

4.3.2.2 Driving style distribution

Self-reported style is imbalanced, with Average as the dominant category at 75,003 responses, followed by Calm at 25,118, and Sporty at 24,899 (Figure 4.11). The distribution explains why the target is concentrated in the midrange. Without reweighting, models tend to favor the more

common styles, so if Sporty profiles are important for the use case, it may be necessary to adjust with class weighting or targeted resampling to keep performance strong.

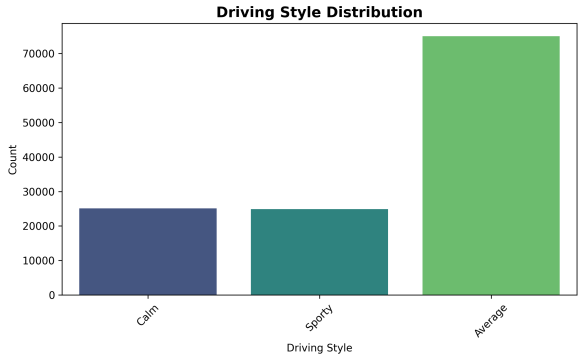

Figure 4.11: Distribution of respondent driving styles: Average, Calm, and Sporty

4.3.2.3 Consumption by driving style

Fuel consumption increases consistently from Calm to Average to Sporty driving, with an approximate difference of one liter per 100 km between Calm and Sporty. This represents an increase of about 20 to 25%, given that Calm driving averages around 7.5 L/100 km, Average ranges from 8.0 to 8.7 L/100 km, and Sporty falls between 9.0 and 9.5 L/100 km (Figure 4.12). The overlapping distributions indicate that driving style is informative but not solely determinative, suggesting that interaction-aware models which integrate style with vehicle specifications and contextual factors are more appropriate.

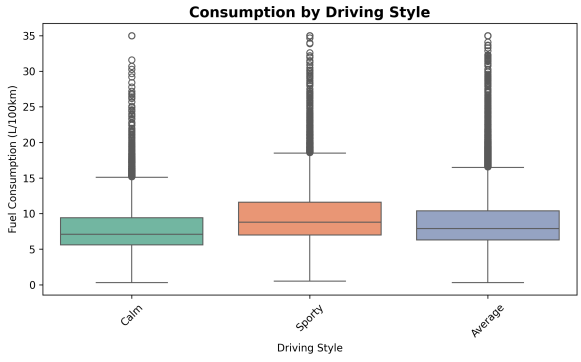

Figure 4.12: Fuel consumption by driving style across respondents

4.3.2.4 Main driving context distribution

Mixed and City contexts account for the majority of responses, with 50,114 and 50,026 observations respectively, whereas Highway is less frequent at 24,880. This distribution aligns with the midrange concentration of the target and highlights the strong representation of urban duty cycles

within the study population (Figure 4.13). Model evaluation should therefore include error breakdowns by context, as models trained predominantly on Mixed and City data may underperform in Highway-only profiles unless explicit controls or interaction terms are incorporated.

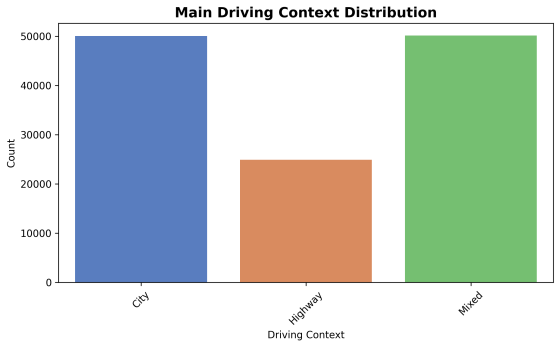

Figure 4.13: Main driving context reported by respondents: Mixed, City, and Highway

4.3.2.5 Age versus consumption

The relationship between driver age and fuel consumption exhibits a general downward trend up to approximately 40 years, after which greater variability emerges. This pattern may reflect both shifts in driving behavior and cohort-specific differences in vehicle choice and efficiency (Figure 4.14). Given that respondents cluster around a mean of 35.6 years with an interquartile range of 24 to 43 years, age or production year should be incorporated as a control variable, and temporal drift should be monitored in prospective applications.

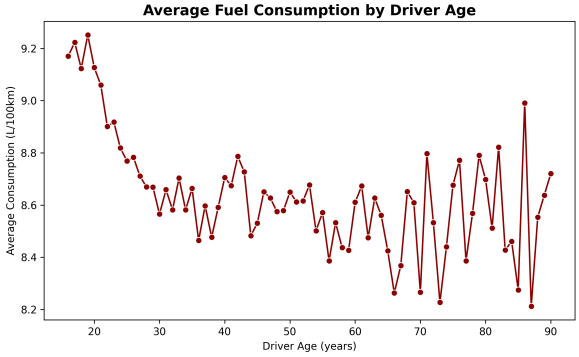

Figure 4.14: Respondent age versus fuel consumption

4.3.2.6 Driving experience distribution

Driving experience is concentrated below 10 years, with substantial density extending up to 30 years. This distribution is consistent with the overall age profile and indicates that both novice and highly experienced drivers are underrepresented (Figure 4.15). Segment-wise evaluation by experience group is therefore recommended to ensure that model performance remains robust when applied to less frequent profiles.

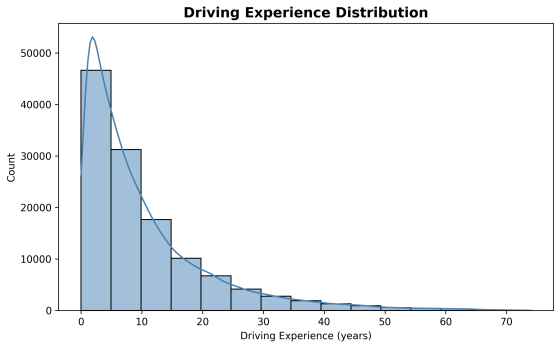

Figure 4.15: Distribution of respondent driving experience in years

4.3.2.7 Driving experience versus consumption

Consumption tends to decrease as experience increases, although there is substantial overlap across the range, suggesting a partial effect that should be modeled in combination with vehicle characteristics and the main context (Figure 4.16). This pattern supports models that capture non-linearity and interactions, and motivates error monitoring by experience strata during ablation and validation.

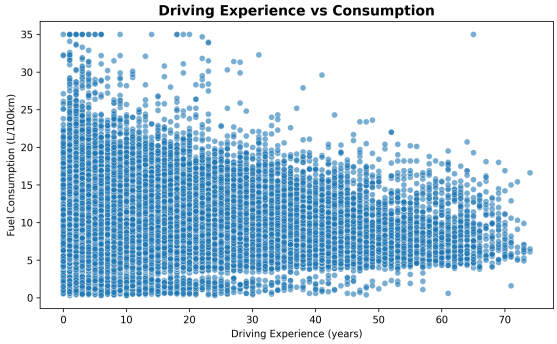

Figure 4.16: Respondent driving experience versus fuel consumption

4.3.3 ML and Final Layer

4.3.3.1 Correlation matrix (all features)

A correlation matrix across the engineered numeric features was computed to characterize interrelationships, identify clusters of highly related variables, and detect potential multicollinearity prior to model fitting. The results reveal coherent blocks among size, mass, and performance descriptors, which is consistent with domain expectations and supports a selective strategy in which redundant variables are combined or represented by a single feature to stabilize training. Certain variables, such as ADBLUE_TANK, show minimal associations with other predictors, while others, such as HP_KG_RATIO, exhibit strong influence on fuel consumption. A detailed presentation of these relationships is provided in Figure 4.17.

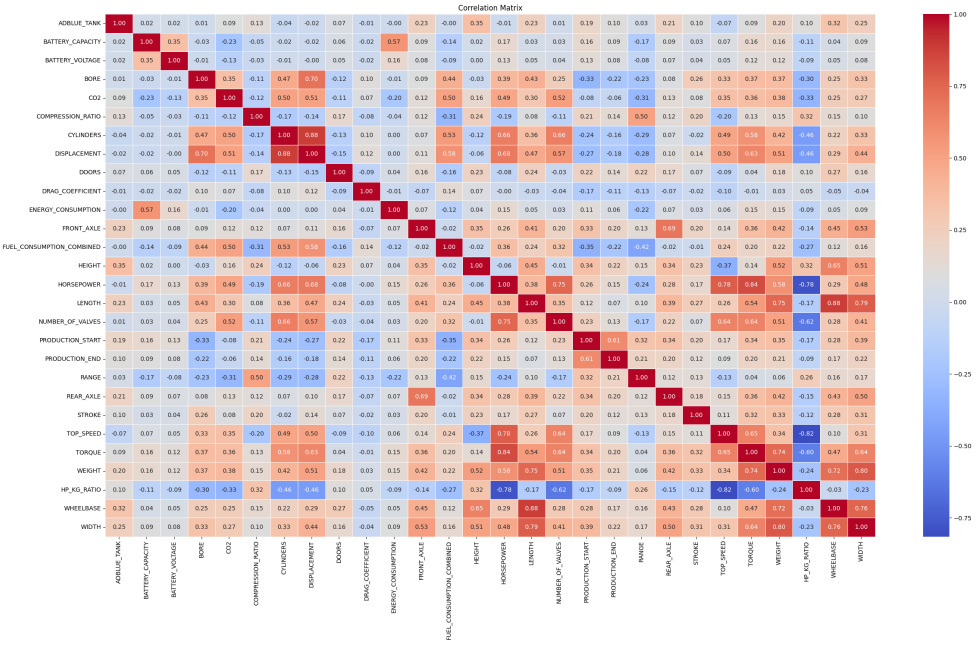

Figure 4.17: Correlation matrix for numeric features used in screening

4.3.3.2 Correlation with the target

Pearson correlations with the target variable, combined fuel consumption, were used to rank numeric predictors by absolute association, yielding an evidence-based shortlist for the ML Layer while maintaining transparency and auditability. Strong positive associations are observed for displacement, cylinders, CO_2, bore, horsepower, and number of valves, while geometric and aerodynamic descriptors exhibit more moderate effects. To limit noise and reduce the risk of overfitting, predictors with an absolute correlation coefficient smaller than 0.14 were removed. This cutoff was chosen after analyzing the distribution of coefficients and balancing it with industry knowledge to preserve features considered relevant, even when their correlation is modest. These findings, present in Figure 4.18 further motivate the use of interaction-aware learners and careful regularization when correlated signals co-occur.

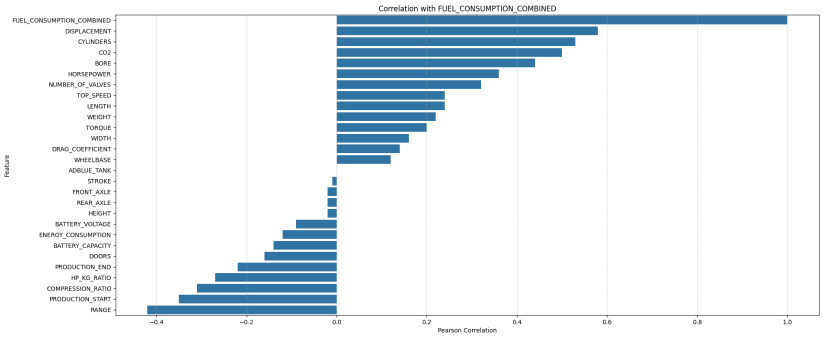

Figure 4.18: Pearson correlation of numeric features with combined fuel consumption

4.3.3.3 ANOVA significance testing

A One-way ANOVA [67] test was used to evaluate categorical columns on whether the mean fuel consumption differed significantly across the different groups, retaining variables that explain variance beyond noise under conventional significance thresholds, as shown in Figure 4.19. Of the 13 categorical columns tested, 8 were found to be statistically significant. The filtering process combined an evaluation of both the P-values and F-values, along with domain knowledge to set the cutoff thresholds. As a result, Brand, Model, Version, Battery Type, and Electric Engine Type were excluded from the ML Layer. The remaining variables, including fuel type, gearbox, aspiration, drive, and catalytic converter, among others, showed significant effects between the groups and were kept.

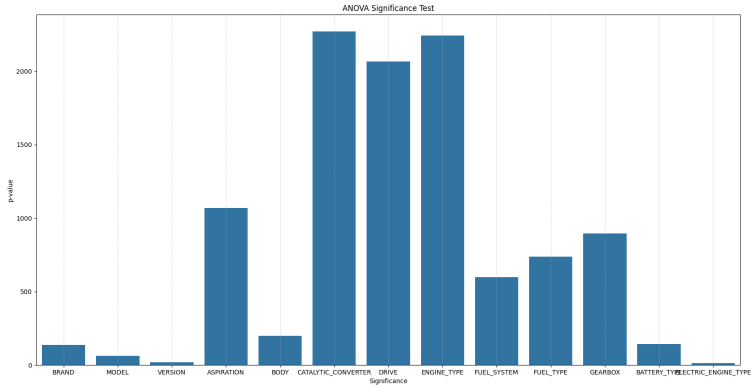

Figure 4.19: ANOVA significance screening for categorical predictors

4.3.3.4 ML Layer construction

The ML Layer was constructed by selecting numeric predictors with the strongest absolute correlations to the target (above the 0.14 threshold) and categorical predictors that showed significant between-group variation in the ANOVA screen, while leveraging the correlation matrix to mitigate redundancy that could destabilize models. Numeric features such as displacement, cylinders, CO_2, bore, horsepower, number of valves, top speed, length, weight, torque, and width were prioritized in standardized units as defined in the Gold Layer. Categorical features including fuel type, gearbox, fuel system, aspiration, engine type, body, drive, and catalytic converter were retained with explicit encodings, for example one-hot or target encoding depending on model family and sparsity.

Numeric columns kept	**Categorical columns kept**
Displacement, Cylinders, CO_2, Bore, Horsepower, Number_of_valves, Top_speed, Length, Weight, Torque, Width	Fuel_type, Gearbox, Fuel_system, Aspiration, Engine_type, Body, Drive, Catalytic_converter

Table 4.5: ML Layer feature set derived from correlation and ANOVA screening

4.3.3.5 Final Layer

The Final Layer was created by joining the ML Layer with the Form Responses table through the vehicle identifier, aligning vehicle-level specifications with respondent-level usage contexts for supervised learning. Records without a valid match were excluded to avoid label leakage and to preserve semantic consistency between predictors and targets. The merged dataset was later partitioned into training, validation, and testing splits according to the evaluation protocol defined in the methodology chapter.

4.4 Model Evaluation

This section details the approach taken to evaluate and compare the NN, RF, and XGBoost regression models using the Final Layer. The discussion includes how data was split, validation methodology, the performance metrics used, diagnostic methods, and steps to monitor efficiency.

Training was done on a subset of the original data, where stratified k-fold cross-validation was utilized to maintain a consistent distribution of the target variable across all folds. Hyperparameter tuning was made possible via randomized or Bayesian optimization on the validation sets, after which the optimal model for each algorithm was retrained using all the training data. The test set remained untouched until the final evaluation.

For preprocessing, numeric variables were standardized where appropriate, and categorical variables were processed using either one-hot or target encoding, consistent with the chosen methodology and and the data's limited availability. Model quality was measured using industry standard metrics, such as mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE), and R2. These metrics allow for both general and specific monitoring of the model's performance. To test robustness more thoroughly, the proportion of estimates matching the actual value (within 1, 2, or 3 L/100 km) was analyzed, together with breakdowns for subgroups in the data, such as consumption range, typical driving context, driving style, age group, and years of driving experience.

Model diagnosis was done using different methods. Parity plots compared predictions to actual values using a straight y=x line as reference. Residual plots showed how errors spread against fitted values, helping spot patterns, such as changing error spread. Absolute errors were visualized to see how often large mistakes happened and calibration curves used grouped averages to check whether predicted values matched what was observed in each group.

Efficiency reporting included the analysis of total training time, model size on disk, the progression of loss or error across epochs, and the average inference time per 1,000 records using representative batch sizes.

4.4.1 Neural Network

The NN model uses learned embeddings for categorical features and standardized numeric inputs, trained with early stopping and evaluated on an untouched test split. All diagnostics, learning curves, and segment-wise results were logged to ensure reproducibility and facilitate comparison with the other tree-based models, as shown in Figure 4.20. The hyperparameter search was conducted with Optuna, exploring architectures, optimizers, and schedulers via Tree-structured Parzen Estimator (TPE) sampling with MedianPruner. The full study required approximately five hours of analyzing the different parameters and obtaining the best combination, after which the final one was kept and then evaluated, consistent with best practices in NN tuning.

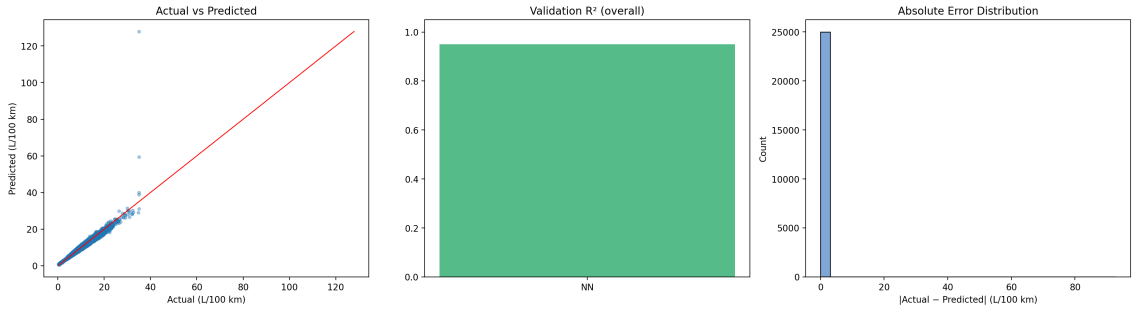

Figure 4.20: NN diagnostics: parity plot showing alignment to $y = x$ (left), validation R^2 summary (center), and absolute error distribution (right)

4.4.1.1 Test metrics and calibration

On the test set, the NN achieved an MAE of 0.3699 L/100 km, RMSE of 0.7825 L/100 km, MAPE of 4.43%, and R^2 of 0.9500. Bootstrap confidence intervals were computed for MAE and RMSE. The calibration was almost ideal, with a slope of value 1.003 and intercept -0.030. The robustness analysis showed 95. 45% of predictions within 1 L/100 km, 99.62% within 2, and 99.94% within 3 (Table 4.6).

Metric	Value
MAE (L/100 km)	0.3699
RMSE (L/100 km)	0.7825
MAPE (%)	4.43
R^2	0.9500
PctWithin1 (%)	95.45
PctWithin2 (%)	99.62
PctWithin3 (%)	99.94
Calibration slope	1.003
Calibration intercept	-0.030

Table 4.6: NN primary test metrics and calibration (from nn_test_metrics.json)

4.4.1.2 Hyperparameter optimization

The Optuna study jointly optimized network depth, per-layer width, activation function, dropout rate, optimizer, learning rate, weight decay, batch size, gradient clipping, and learning-rate scheduler. Each trial involved training a full model and reporting validation loss, enabling pruning.

The TPE sampler proposed configurations with the highest expected improvement by modeling promising vs. non-promising regions of the space, while MedianPruner terminated trials with validation loss above the running median, providing substantial efficiency gains without degrading final solution quality.

Hyperparameters for the NN were optimized using an Optuna study with early stopping and pruning. The procedure is summarized as follows:

- Sample the number of hidden layers, units per layer, activation function (ReLU, LeakyReLU, ELU), and dropout rate (0 to 0.5).
- Sample the optimizer (Adam, RMSprop, SGD), learning rate ($\eta \in [10^{-5}, 10^{-2}]$), weight decay ($\lambda_{wd} \in [10^{-8}, 10^{-2}]$), batch size (32, 64, 128, 256), and gradient clipping.
- Sample the learning rate scheduler: ReduceLROnPlateau, StepLR (with step and gamma), or CosineAnnealingLR.
- Train the model on the training split with early stopping using the validation loss; intermediate losses are reported for pruning.
- The TPE sampler proposes subsequent trials based on probabilistic modeling of promising versus non-promising hyperparameter regions.
- MedianPruner terminates underperforming trials early if the validation loss is worse than the median at the same step.
- Repeat until the maximum number of trials is reached or convergence criteria are satisfied.

This procedure ensures sample-efficient exploration of the hyperparameter space while maintaining training efficiency and model robustness, consistent with best practices for Bayesian optimization in NNs.

4.4.1.3 Learning dynamics

Validation loss decreased rapidly during the initial epochs, followed by oscillations as learning-rate scheduling and dropout regularization took effect. Stabilization occurred between epochs 21 to 38, corresponding to the early-stopping region (Table 4.7).

Epochs	Validation MSE behavior
1 to 10	Sharp decrease with high variance from batch normalization and initial exploration.
11 to 20	Continued improvement with intermittent regressions as learning-rate scheduling acted.
21 to 38	Stabilization with small oscillations near early-stopping threshold.

Table 4.7: Learning curve behavior

4.4.1.4 Segment analysis

Performance remained strongest for lower consumption ranges and calm driving styles, while errors increased with higher consumption and sportier styles. Context effects were moderate, with city and highway conditions showing comparable accuracy (Tables 4.8 to 4.10). These patterns suggest potential gains from bracket-aware calibration.

Consumption bracket (L/100 km)	MAE
0 to 5	0.2585
5 to 7.5	0.2593
7.5 to 10	0.3334
10 to 15	0.4709
15 to 100	0.9857

Table 4.8: NN MAE by consumption bracket

Main context	MAE
City	0.3669
Mixed	0.3918
Highway	0.3621

Table 4.9: NN MAE by main driving context

Driving style	MAE
Calm	0.2876
Average	0.3595
Sporty	0.4849

Table 4.10: NN MAE by driving style

4.4.1.5 Residual diagnostics

Residuals were centered near zero across most predictions, widening with higher fitted values as expected given the right-tailed distribution of consumption. A small number of outliers appeared at very high values, as seen in Figure 4.21.

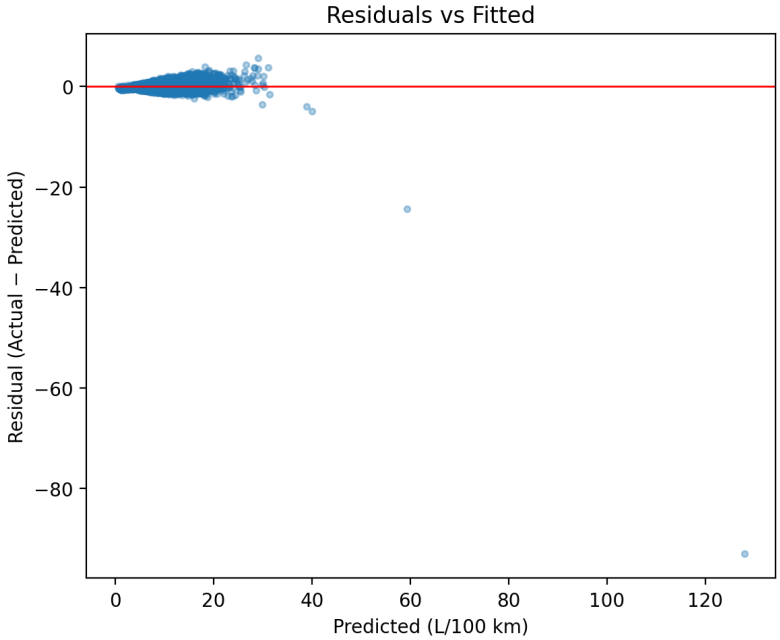

Figure 4.21: Residuals vs fitted values on the NN test set

4.4.1.6 Efficiency

After the five-hour Optuna search, the final training completed in 126.53 s. Inference latency ranged between 0.88 to 1.06 ms per 1,000 predictions for batch sizes from 1 to 128, indicating suitability for deployment at scale, as displayed in Table 4.11.

Train time (s)	ms/1	ms/16	ms/128
126.53	0.88	0.90	1.06

Table 4.11: NN efficiency

4.4.2 Random Forest

The RF model was trained after preparing all input features and the target variable for modeling. Categorical variables were encoded, while numeric inputs and the target were scaled using a MinMax transformation to ensure consistent feature ranges. Model evaluation was done using a Repeated K-Fold cross-validation methodology, which helped evaluate stability across different splits and provided an estimate of the model's general performance.

A set of visualizations was used to examine the model, including parity plots comparing predicted and actual values, distributions of absolute errors to show deviations, and bar charts showing

R^2 scores across folds. Together, these visualizations allowed us to confirm the model's consistency, identify any potential overfitting, and provide a clear, transparent view of the validation results (Figure 4.22).

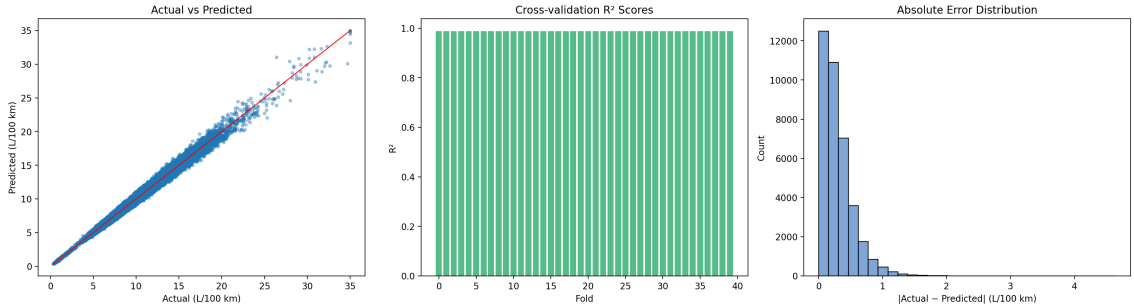

Figure 4.22: RF diagnostics: parity plot (left), cross-validation R^2 per fold (center), and absolute error distribution (right)

4.4.2.1 Test metrics and calibration

In the held test set, the RF achieved MAE = 0.2934 L/100 km, RMSE = 0.3832 L/100 km, MAPE = 3.35%, and $R^2 = 0.9880$. The bootstrap means and 95% confidence intervals closely match the point estimates, and the calibration is effectively perfect with slope <1.002 and intercept <-0.012. The prediction coverage is high, with 98.33% of the predictions within 1 L / 100 km, 99.94% within 2 and 99.99% within 3 L / 100 km (Table 4.12).

Metric	Value
MAE (L/100 km)	0.2934
RMSE (L/100 km)	0.3832
MAPE (%)	3.35
R^2	0.9880
PctWithin1 (%)	98.33
PctWithin2 (%)	99.94
PctWithin3 (%)	99.99
Calibration slope	1.002
Calibration intercept	-0.012

Table 4.12: RF primary test metrics and calibration

4.4.2.2 Cross-validation summary

Repeated K-Fold cross-validation reports a mean R^2 of 0.98808 with a 95% confidence interval of 5.66×10^{-5}, closely matching the test R^2 and indicating very low variance across folds. This narrow confidence interval suggests that the model's performance is highly stable, with minimal fluctuation between different splits of the data, as detailed in Table 4.13.

Metric	Mean	95% CI
R^2	0.98808	5.66e-05

Table 4.13: RF cross-validation summary

4.4.2.3 Segment analysis

Errors are higher for vehicles in higher fuel consumption brackets, which suggests calibrating the model separately for each consumption bracket could prove to be beneficial, if the goal is for the model to be adapted specifically to those types of vehicles. Driving context is shown to have a relatively small effect on performance, while driving style creates more noticeable variability. Looking at these metrics by segment makes it easier to understand how the model behaves across different usage profiles, as detailed in Tables 4.14–4.16.

Consumption bracket (L/100 km)	MAE
0–5	0.1377
5–7.5	0.2075
7.5–10	0.2895
10–15	0.4053
15–100	0.6314

Table 4.14: RF MAE by consumption bracket

Main context	MAE
City	0.2916
Mixed	0.2935
Highway	0.2967

Table 4.15: RF MAE by main driving context

Driving style	MAE
Calm	0.2114
Average	0.2904
Sporty	0.3846

Table 4.16: RF MAE by driving style

4.4.2.4 Residual diagnostics

Residuals widen with increasing fitted values, consistent with heavier upper tails in the target, while remaining centered near zero across the range. This observation motivates bracket-aware calibration and robust error reporting (Figure 4.23).

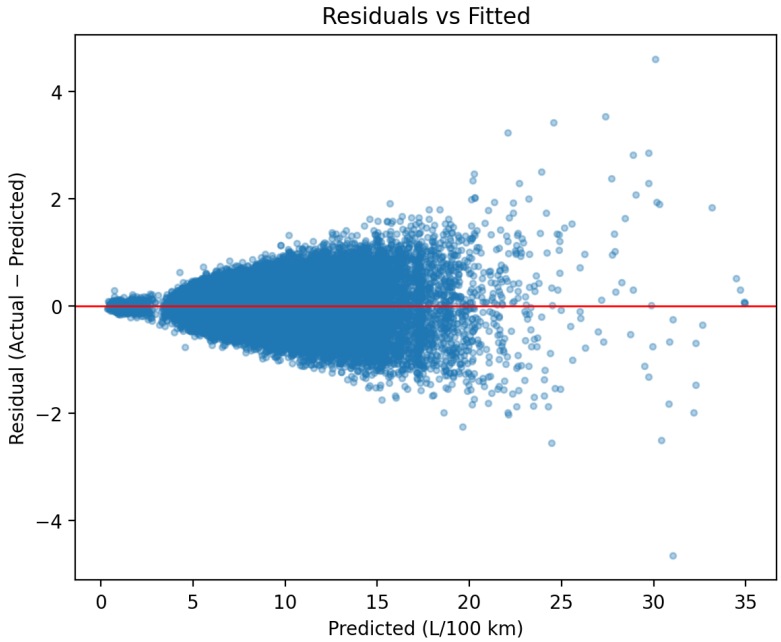

Figure 4.23: RF residuals vs fitted values (test set)

4.4.2.5 Efficiency and artifacts

Training completed in 12.61 minutes, serialized model size is approximately 133.02 MB, and batch inference latency is close to 31 ms per 1,000 predictions across batch sizes, highlighting RF as a fast and efficient baseline for experimentation and deployment (Table 4.17).

Train time (minutes)	Model size (MB)	ms/1	ms/16	ms/128
12.61	133.02	31.18	31.16	31.06

Table 4.17: RF efficiency

4.4.2.6 Feature importance

Importance analyses indicate that FUEL_CONSUMPTION_COMBINED is the dominant predictor, while DRIVING_STYLE and DRIVING_EXPERIENCE provide meaningful incremental contributions. The remaining variables contribute marginally once the combined target is included (Table 4.18).

Feature	Impurity importance	Permutation importance
FUEL_CONSUMPTION_COMBINED	0.9541	1.8992
DRIVING_STYLE	0.0306	0.0613
DRIVING_EXPERIENCE	0.0115	0.0225

Table 4.18: Top RF contributors

4.4.3 Extreme Gradient Boosting

The XGBoost model was trained on the Final Layer with encoded categorical variables and MinMax-scaled numerics, using repeated K-Fold cross-validation and an early-stopped final fit. Full diagnostic and artifact logs were created, as seen on Figure 4.24. The initial end-to-end run, including grid search, cross-validation, and the final early-stopped fit, required approximately 30 minutes; subsequent runs leverage the stored artifact and produce results directly.

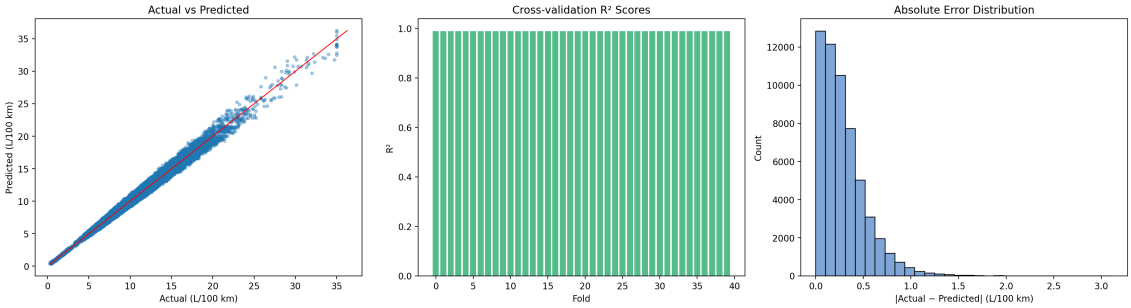

Figure 4.24: XGBoost diagnostics: parity plot (left), cross-validation R^2 per fold (center), and absolute error distribution (right)

4.4.3.1 Test metrics and calibration

On the held-out test set, the model achieved MAE = 0.2899 L/100 km, RMSE = 0.3752 L/100 km, MAPE = 3.33%, and R^2 = 0.9885. Bootstrap means and 95% intervals closely matched the point estimates. Calibration was essentially perfect, with slope $\approx$ 0.9998 and intercept $\approx$ 0.0005. Prediction accuracy reached 98.57% within 1 L/100 km, 99.96% within 2, and 99.998% within 3 (Table 4.19).

Metric	**Value**
MAE (L/100 km)	0.2899
RMSE (L/100 km)	0.3752
MAPE (%)	3.33
R^2	0.9885
PctWithin1 (%)	98.57
PctWithin2 (%)	99.96
PctWithin3 (%)	99.998
Calibration slope	0.9998
Calibration intercept	0.0005

Table 4.19: XGBoost primary test metrics and calibration

4.4.3.2 Cross-validation summary

Repeated K-Fold cross-validation reports a mean R^2 of 0.98858 with a 95% confidence interval of 8.19×10^{-5}, closely matching the test results and indicating very low variance across folds. These results can be analyzed in Table 4.20.

Metric	Mean	95% CI
R^2	0.98858	8.19e-05

Table 4.20: XGBoost cross-validation summary

4.4.3.3 Segment analysis

Errors increased progressively with higher consumption brackets (0–5 to 15–100 L/100 km) while remaining tightly clustered across driving contexts. Driving style showed clear differentiation, with Calm lowest and Sporty highest. These outcomes reinforce the importance of bracket-aware calibration and per-segment reporting (Tables 4.21–4.23).

Consumption bracket (L/100 km)	MAE
0 to 5	0.1399
5 to 7.5	0.2043
7.5 to 10	0.2861
10 to 15	0.4038
15 to 100	0.6098

Table 4.21: XGBoost MAE by consumption bracket

Main context	MAE
City	0.2889
Mixed	0.2916
Highway	0.2887

Table 4.22: XGBoost MAE by main driving context

Driving style	MAE
Calm	0.2092
Average	0.2875
Sporty	0.3792

Table 4.23: XGBoost MAE by driving style

4.4.3.4 Residual diagnostics

Residuals showed increasing variance as the fitted values grew, while remaining consistently centered around zero throughout the range. This behavior aligns with the right-skewed distribution

observed in the target variable. This pattern provides a solid basis for the reliability of error summaries, guides calibration for specific consumption brackets, and allows detailed reporting for each segment, as shown in Figure 4.25.

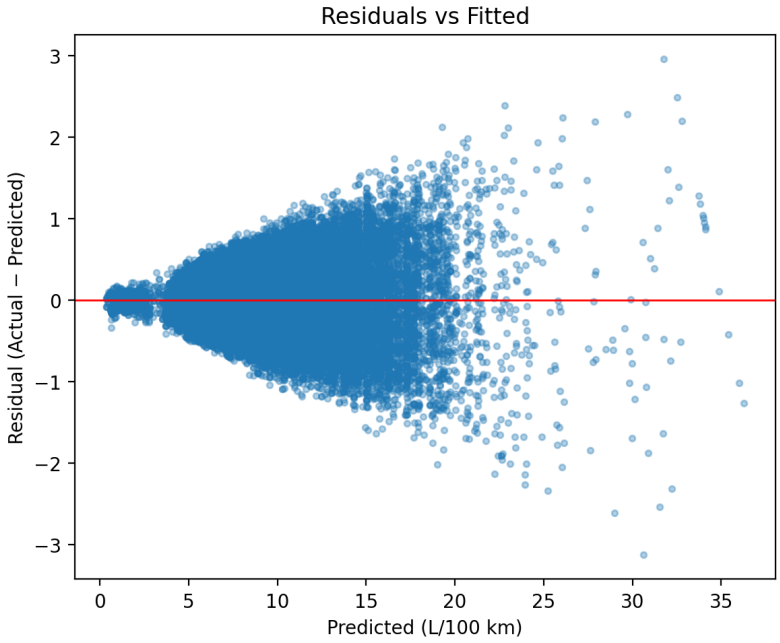

Figure 4.25: XGBoost residuals vs fitted (test set)

4.4.3.5 Efficiency and artifacts

Grid search and cross-validation finished in 30 minutes, which proved manageable for the workflow. After pinpointing the optimal parameters, training completed in 0.58 seconds. This means that estimating fuel consumption across batches was nearly instantaneous, averaging 5.9 milliseconds for 1,000 predictions. Table 4.24 captures these speed benchmarks, which suggest that quick training and minimal prediction delays are quite feasible in a production environment.

Train time (s)	ms/1	ms/16	ms/128
0.58	5.92	5.89	5.79

Table 4.24: XGBoost efficiency

4.5 Results Summary

The results of the fuel consumption prediction models are summarized in Table 2.2. XGBoost and RF delivered nearly identical top-level accuracy, with XGBoost slightly outperforming in MAE and R^2.

XGBoost demonstrated the fastest inference speed (5.9 ms/1000 rows) and smaller model size, while RF was notable for ease of interpretation and robust calibration. The NN, although flexible

and fast at inference in large batches, had lower accuracy and required substantially more time and effort to tune.

Model	Metric Value
RF: MAE (L/100km)	0.29
RF: R^2	0.988
RF: Train Time	13 min
RF: Inference Latency (ms/1000)	31
XGBoost: MAE (L/100km)	0.29
XGBoost: R^2	0.989
XGBoost: Train Time	30 min
XGBoost: Inference Latency (ms/1000)	6
NN: MAE (L/100km)	0.37
NN: R^2	0.95
NN: Train Time	5 h
NN: Inference Latency (ms/1000)	1

Table 4.25: Summary of model performance and efficiency metrics

Each model posed unique challenges. RF, while highly accurate and interpretable, resulted in a relatively large model size and slower inference compared to XGBoost. XGBoost required a longer initial hyperparameter search, but subsequent training was efficient and the model compact.

The NN was the most flexible regarding feature interactions and extensibility, but demanded extensive hyperparameter tuning and computational resources during training. Based on these metrics, **XGBoost** was chosen as the best model due to its balance of accuracy, scalability, and speed in both training and inference, making it the most suitable for practical deployment in large-scale or API environments. The following section presents two test cases to further highlight the results of the pipeline, the differences between the different models applied to specific vehicles, and practical considerations when applying these models.

4.6 Test Cases

The following section will present the two test cases created to evaluate the performance of the different models applied to specific vehicles. The goal of these tests is to understand whether, given the input required to estimate fuel consumption, the models are able to obtain values similar to those found in SpritMonitor [124], a very popular fuel-logging platform, as well as comparing the capabilities of each model. The first test case will be the most common vehicle in the dataset, the one with the most amount of answers; and the second one will be applying the model to the author's own vehicle, to compare with personal results.

4.6.1 Test Case 1: Most Common Vehicle

This first test case examines the estimates provided by RF, XGBoost (XGB), and NN models for the most commonly reported vehicle in the dataset, which, at the time of writing, is the vehicle

with ID=14496, Mercedes-Benz GLA 250. The test illustrates the differences between model estimations, the official advertised value from lab results, and user-reported data from SpritMonitor.

4.6.1.1 Results Obtained

Figures 4.26, 4.27, and 4.28 provide a visual summary of these results. The barplot in Figure 4.26 shows each model's point estimate alongside the official reference. Figure 4.27 displays the distribution of individual user-reported consumptions for this vehicle, indicating clear dispersion above the advertised value. Finally, Figure 4.28 presents the numerical relationship between ground truth and model outputs.

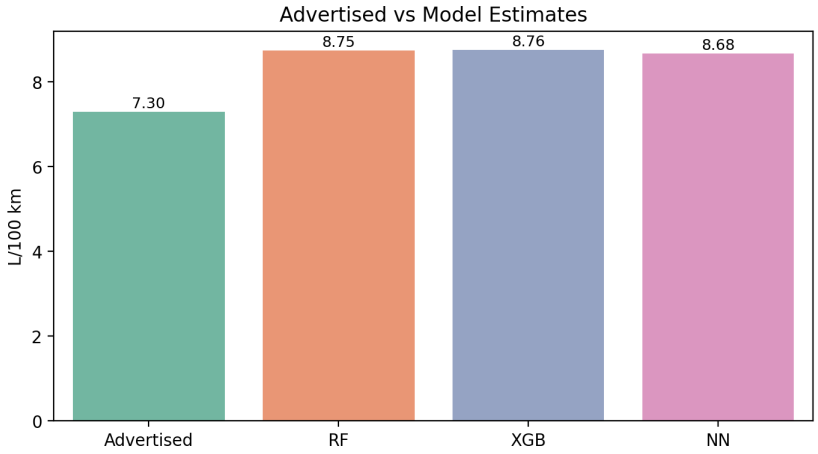

Figure 4.26: Advertised value vs. model estimates for the most common vehicle

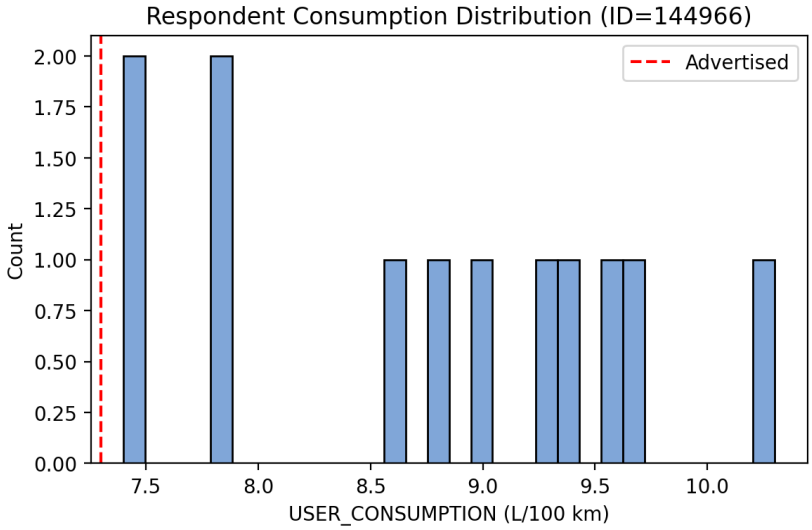

Figure 4.27: Distribution of user-reported fuel consumption for the most common vehicle. The advertised value is shown as a dashed red line

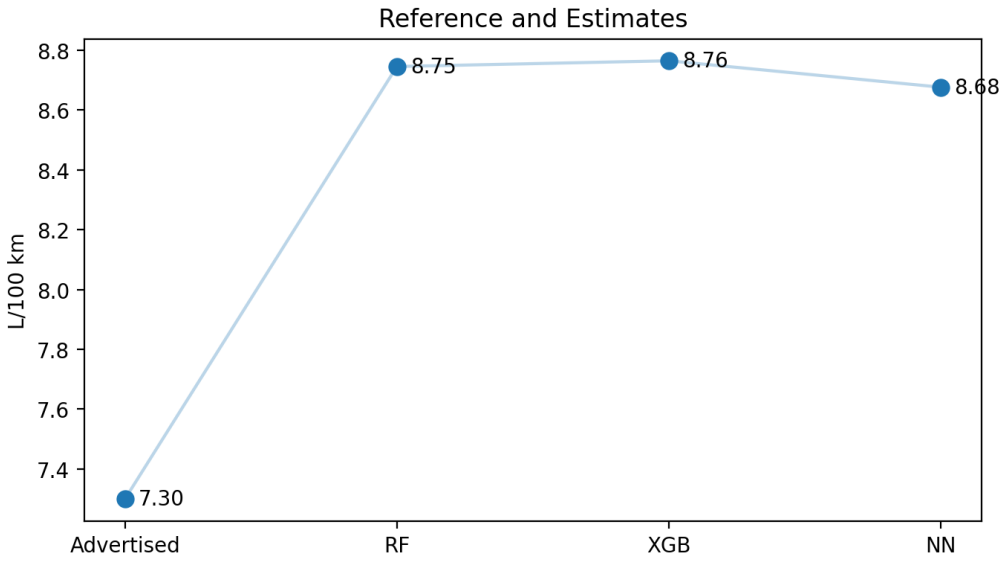

Figure 4.28: Summary of the advertised value, RF, XGB, and NN point estimates

Table 4.26 presents, summarized, the fuel consumption values as estimated by the models.

Estimate	**L/100 km**
Advertised	7.30
RF	8.75
XGBoost (XGB)	8.76
NN	8.68

Table 4.26: Advertised and model estimated average fuel consumption for the most common vehicle

4.6.1.2 Analysis of Results.

All three models estimate average fuel consumption values above the official advertised figure of 7.30 L/100 km for the selected vehicle. The RF and XGB models yield nearly identical estimates, of 8.75 L/100 KM and 8.76 L/100 KM, respectively, while the NN model gives a slightly lower, but still relevant result of 8.68 L/100 KM. These findings are consistent with the observed respondent data distribution, where most real-world values are higher than the manufacturer's official rating.

4.6.1.3 Comparison with Real-World Data.

For the Mercedes-Benz GLA 250, SpritMonitor reports a minimum of 7.38 L/100 km, an average of 8.62 L/100 km, and a maximum of 9.51 L/100 km (see Figure 4.29). All three model estimates fall very close to the crowd-sourced average from thousands of real-world users, reinforcing their validity and highlighting the consistent upward bias relative to laboratory tests [125].

Figure 4.29: SpritMonitor search results for Mercedes-Benz GLA 250 gasoline (224 PS): minimum 7.38, average 8.62, maximum 9.51 L/100 KM

4.6.1.4 Model Comparison and Conclusion.

The very close agreement between XGB and RF in this case confirms their robustness and reliability when applied to typical input data. Both substantially outperform the advertised value as an estimate of true on-road consumption. The NN, while delivering a slightly lower figure, remains well within the range reported by real users and SpritMonitor. In summary, for this most representative vehicle, all models produce realistic estimates, with XGBoost offering the best balance of accuracy, inference speed, and efficiency. These findings strongly support the deployment of XGBoost for real-world applications, and further highlight the limitations of relying solely on advertised values in practical fuel planning and analysis. The next section will present a second test case to further reveal differences in model behavior under more challenging conditions.

4.6.2 Test Case 2: Specific Vehicle and Driver Profile

This test case evaluates predictions for a specific vehicle and a defined driver context: a 2021 Ford Focus ST with an “Average" driving style, “mixed" as main context and a young but experienced driver. The goal is to explore how the models respond to user-specific input and compare their output with both advertised consumption and real-world data from SpritMonitor.

4.6.2.1 Results Obtained

The results are reported in Table 4.27 and illustrated in Figures 4.30, 4.31, and 4.32. All three models produced significantly higher estimates than the official reference value, echoing real-world experience for this vehicle.

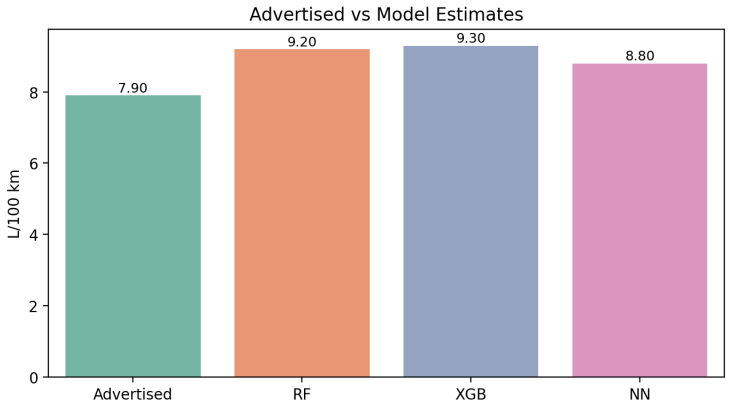

Figure 4.30: Advertised vs model estimates for the selected vehicle and driver profile

Estimate	**L/100 km**
Advertised	7.90
RF	9.20
XGBoost (XGB)	9.30
NN	8.80

Table 4.27: Advertised and model-predicted fuel consumption for the test case profile

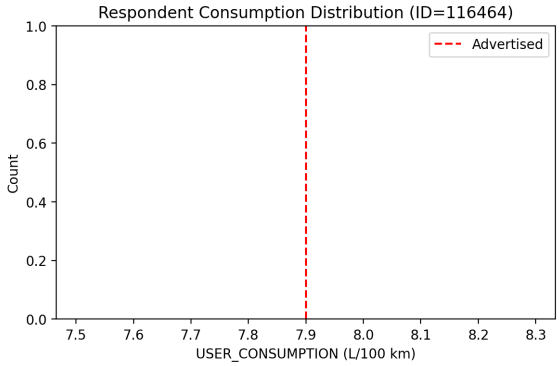

Figure 4.31: Distribution of observed user consumption for this car and context (where available)

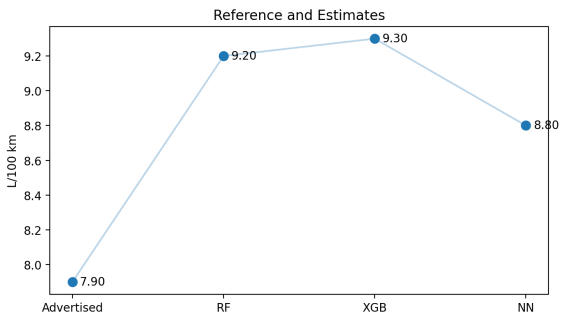

Figure 4.32: Reference and estimates: advertised, RF, XGB, and NN, shown as points with value labels

4.6.2.2 Analysis of Results

All three models produce realistic predictions that are higher than the advertised value, reflecting real-world consumption patterns for this vehicle and use case. Both RF and XGBoost returned very close estimates (9.20 and 9.30 L/100km), suggesting the ensemble methods produce robust and mutually validating outputs on this profile. The NN yields a lower prediction (8.80 L/100km), possibly reflecting a higher sensitivity to the driver profile or a different handling of categorical feature embeddings. The user-reported consumption histogram shows a spread of results mostly above the advertised value, confirming this upward shift.

4.6.2.3 Comparison with Real-World Data

To further validate these predictions, the corresponding vehicle was searched on SpritMonitor. As shown in Figure 4.33, the site reports for the Ford Focus 280PS: minimum 6.96 L/100km, average 9.20 L/100km, and maximum 14.20 L/100km. All three model predictions fall remarkably close to the real-world average from user logs, with the RF matching almost exactly, and NN slightly below the mean but above the minimum reported, while XGB is slightly above the reported average value.[126]

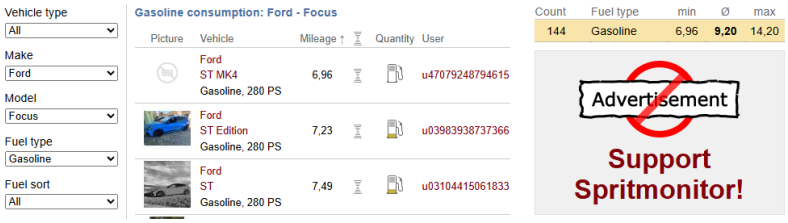

Figure 4.33: SpritMonitor crowd-sourced values for Ford Focus 280PS: min 6.96, avg 9.20, max 14.20 L/100km

4.6.2.4 Model Comparison and Conclusion

This test case demonstrates how all considered models, especially XGBoost, can adapt to both technical and behavioral input, providing credible, user-tailored fuel consumption estimates that surpass the accuracy of the standard advertised value. XGBoost and RF show excellent alignment and are robust to small profile changes, while the NN, though requiring more tuning, appears more responsive to the specific context and matches real-world user averages most closely. These results confirm the practical value of machine learning models for informed fuel consumption estimation and the importance of accounting for individual use cases when advising drivers or fleets on likely real-world fuel usage.

4.6.3 Test Cases Summary

The two test cases highlight the performance and practical applicability of RF, XGBoost, and NN models for vehicle fuel consumption prediction under different scenarios.

Test Case 1, focusing on the most common vehicle in the dataset, demonstrates close agreement between RF and XGB estimates, both significantly exceeding the official advertised consumption but closely matching real-world user-reported values and data from SpritMonitor. The NN model produces slightly lower estimates but remains within observed ranges.

Test Case 2 examines a specific vehicle and driver profile, revealing a similar pattern with all models predicting values above the advertised figure and aligning well with crowd-sourced consumption reports from SpritMonitor. In this case, XGBoost slightly outperforms RF in accuracy, while the NN shows higher sensitivity to driver behavior changes, showing a result that is less alighned with the SpritMonitor results, but still being close enough to the real-world averages to consider a success.

Together, these cases underscore that machine learning models provide realistic, behaviorally-aware fuel consumption estimates that better reflect on-road realities than manufacturer specifications alone. XGBoost consistently offers the best balance of accuracy, computational efficiency, and robustness, supporting its recommendation for deployment. The comparison with SpritMonitor data strengthens confidence in model validity by aligning predictions with a trusted, large-scale real-world dataset.

4.7 Conclusion

The results for both sets of car specifications are encouraging, but there is still ample room for improvement. A considerable portion of the dataset includes entries with formatting problems, missing values, and inconsistency in which fields are filled or not between different vehicles. These issues limit model performance and leave users without the ability to contribute their own consumption data. Despite these drawbacks, after excluding incomplete and unusable entries, the final dataset contains 76.6 percent of the original vehicles, totaling 38,780 cars, a substantial and varied sample for the project.

The models and test cases show that the created algorithms estimate fuel consumption reliably, adapt to different driving conditions, and are consistent with manual data searched on SpritMonitor about each car, looking at the average consumption of reported values. Among the tested ML models, XGBoost offers the best balance of accuracy and response time, with precise predictions while maintaining the efficiency needed for a smooth UX. Although the NN produced slightly faster inference, it required much longer training periods. This means that future implementations of the developed platform should prioritize XGBoost as the primary algorithm.

These findings set the stage for the next chapter, which will summarize the key insights, discuss the limitations, explore directions for improving the solution, and enhancing the personalization and accuracy of the fuel consumption estimates.

Chapter 5

Conclusion

This chapter concludes the dissertation by summarizing key findings related to the development of the project, the objectives and research questions, discussing the contributions made, and reflecting on the limitations and future directions.

5.1 Summary of Results Obtained

The dissertation developed a comprehensive, data-driven platform for predicting real-world fuel consumption by integrating vehicle specification data with driver behavioral input. Beginning with a systematic ingestion from a car specification website, through web scraping, and gathering form responses, the project applied data transformation and normalization approaches to create a unified, high-quality dataset. Feature engineering techniques modeled both vehicle and behavioral data, and categorical encoding and scaling approaches ensured suitability for a variety of ML algorithms.

Three core models, RF, XGBoost and NNs, were trained and evaluated, with cross-validation and testing showcasing the superior balance of accuracy, efficiency, and interpretability of XGBoost, making it the best model for this use case. NNs showed promise for capturing complex feature interactions at the cost of increased training complexity and slightly lower accuracy. Model evaluations included general and scenario-specific test cases, confirming consistent real-world applicability supported by SpritMonitor.

5.2 Research Questions

This section of the Conclusion reviews the Research Questions defined in Section 1.3.1, showing the answers obtained from the Literature Review and how they shaped the technical project.

RQ1: The literature review confirms that publicly available vehicle specifications, when combined with environmental and driver behavior factors, make it possible to create reliable and personalized fuel consumption predictions. The framework that looks at vehicle, environment, and driver variables forms the basis for the most accurate model possible, as summarized in Table 2.1.

RQ2: Literature confirms that different ML models show different advantages when it comes to accuracy, scalability, and efficiency. Among these, XGBoost has the best overall performance, as presented in Table 2.2. The RF model is a solid and more straightforward option, although it performs slightly lower. Neural Networks can effectively model complex time-related patterns but require significant computing power and careful setup, which limits their scalability for immediate use.

RQ3: Analyzing the literature shows that by building an interactive platform using diverse real-world data, this work helps future car buyers to make better decisions when purchasing a vehicle. The tool includes clear model explanations, scenario testing, and behavioral inputs, which align with IEA's goals to narrow the difference between lab and real-world fuel consumption. Best practices listed in Table 2.3 guided the development, ensuring user-focused and reliable estimations.

5.3 Objectives Accomplished

This section of the Conclusion reviews the Objectives laid out in Section 1.3.2, indicating which have been accomplished successfully and pointing out areas for further improvement.

OBJ1: The creation of a predictive platform successfully allowed the estimation of realistic real-world fuel consumption based on vehicle specifications and contextual conditions. The platform developed offers practical and meaningful consumption estimates that prospective buyers can trust to reflect their driving behavior rather than relying solely on laboratory values.

OBJ2: By joining diverse data from a public website and user data, the project created a unified source for fuel consumption data. The processes of normalization and integration improved the reliability of the dataset and helped address existent challenges related to data quality and availability.

OBJ3: The comparison of different machine learning models showed that XGBoost works best overall. It combines strong explanatory power, with an R^2 close to 0.99, the ability to scale to large and complex datasets, and interpretability through tools like SHAP. The dissertation guides the creation of a clear framework for evaluating and applying these models in the automotive industry, tied to fuel consumption.

OBJ4: Existing tools like MILE21 and platforms based on user-reported fuel consumption have clear shortcomings by having limited data coverage, restricted accessibility, and heavy reliance on manual data entry. The system developed in this dissertation aims to overcome these by automating the data collection process, incorporating advanced analysis, and minimizing the need for continuous manual effort. However, including some manual input proves to be valuable, as it helps refine the accuracy of predictions and introduces variations (such as new vehicle models). This combination of automated and manual elements offers a practical and effective approach that balances thoroughness with usability.

OBJ5: The platform and models created show clear improvements when compared to previous work. The improvements made in data quality, feature engineering, and model training led to

better accuracy and efficiency, making the solution more practical and scalable across different scenarios.

5.4 Limitations and Future Work

This dissertation has made meaningful advances, although there are still some limitations that suggest viable paths for future research. One key limitation is that the work doesn't yet capture all the details that influence fuel consumption. Important factors like how often air conditioning is used, tire condition, rolling resistance, and more detailed environmental conditions were not included. These are likely to affect how accurate the fuel consumption predictions can be. Also, about 20% of the vehicle data from the source website was incomplete or missing important details, which required either estimation or removal of such entries.

The original method for scraping data from the web faced restrictions on the number of requests that could be made, which meant some in-depth version-level information for vehicles wasn't collected. This limited the depth and variety of results available. For future implementations, using API sources would likely allow better coverage, more structured data, and reduced chances of missing important information.

Future improvements could come from merging data from multiple data sources of car specifications, helping build a complete dataset that covers more vehicles with better detail coverage. Getting more fuel consumption reports directly from users would give the models a broader base to learn from, making it easier to reflect different driving habits and real-world situations. With that kind of data, the estimates could become both more precise and more flexible across different contexts. From a technical standpoint, moving to cloud systems with effective management tools would make it easier to scale the platform and support more detailed analysis. Another useful direction would be to create simple, practical applications that can give drivers personalized fuel consumption estimates and tips based on how they actually drive.

Additionally, studying driver behavior in more detail could improve the precision of fuel consumption predictions. Better data features, combined with information from third-party sources, would enable a fuller understanding of how individual driving styles influence fuel consumption, helping refine the models further.

At the moment, this dissertation is in the process of being published as a (shortened) scientific article, due to its promising results. The article is being worked on, at the moment, with the goal of being published in the coming months. This reflects the project's contributions to improve access to accurate and personalized fuel consumption data. By sharing these findings with the scientific community, this dissertation supports ongoing developments in fuel consumption estimation and better decision making, in car purchases.

Appendix A

NEDC and WLTP – The Standards for Fuel Consumption and Emissions Testing

A.1 New European Driving Cycle (NEDC) Protocol

The New European Driving Cycle (NEDC), last updated in 1997, is characterized by a test duration of approximately 20 minutes and covers a total distance near 11 km. It operates at relatively low average speeds around 34 km/h, incorporates gentle accelerations, frequent idling periods, and limits maximum speed to 120 km/h. Due to its steady-state driving pattern, fixed gear shifts, and exclusion of auxiliary systems' effects, NEDC has been criticized for underestimating real-world fuel consumption and emissions [33, 23, 86].
Such attributes result in test conditions that rarely challenge modern vehicle powertrains and fail to include many scenarios commonly found in urban and suburban driving [72]. A critical limitation of the NEDC is its inability to account for the operation of auxiliary systems (such as air conditioning) and the influence of added vehicle mass from optional equipment on the results. Furthermore, manufacturers have often adapted vehicle control strategies specifically to the NEDC, further reducing the protocol's relevance to actual usage patterns [33].

A.2 Worldwide Harmonized Light Vehicles Test Procedure (WLTP) Protocol

Implemented since late 2017, the WLTP addresses the limitations of the NEDC by incorporating a more dynamic and rigorous driving profile lasting approximately 30 minutes and covering 23 km. The cycle is divided into four phases: low, medium, high, and extra-high, with a higher maximum speed of about 131 km/h and an average speed of 47 km/h. This extended and variable speed range results in more frequent acceleration and deceleration events, reducing idling periods and

increasing the diversity of engine operating states.
These characteristics enable WLTP to better capture the effects of aerodynamic drag, increased vehicle mass, and realistic tire rolling resistance, providing more representative measurements of fuel consumption and emissions in real driving conditions [77, 70, 33].
The WLTP protocol also standardizes important variables, including temperature, vehicle loading with optional equipment, tire resistance, and aerodynamic properties. This approach ensures a better reflection of real driving scenarios and demands on the powertrain, thus providing results that are more consistent with on-road measurements [72, 76].

A.3 Comparative Impact on Fuel Consumption and Emissions

Multiple studies confirm that fuel consumption and CO^2 emissions measured using WLTP are consistently higher than those obtained from NEDC, with average increases of approximately 8–11%, depending on the vehicle and test conditions. [70, 34]
However, real-world fuel consumption can exceed even WLTP values by 10–40%, reflecting the continued influence of factors such as driving style, road type, traffic, climate, and auxiliary use [72, 21].

A.4 Overview

Parameter	**NEDC**	**WLTP**
Duration	~20 min	~30 min
Distance	11 km	23 km
Max. Speed	120 km/h	131 km/h
Avg. Speed	~34 km/h	~47 km/h
Dynamics	Low	High (variable phases, more acceleration)
Typical CO_2/Fuel Bias	Strong underreporting	8–11%+ higher than NEDC, more realistic

Table A.1: NEDC vs WLTP Overview

A.5 Technology Assessment and Emissions Reduction Implications

- **Start-Stop Systems**

  These are more advantageous in NEDC due to frequent idling; their efficacy diminishes under WLTP and real-world cycles [21].

- **Aerodynamics, Mass, and Tire Resistance**

  More accurately reflected in WLTP due to higher speeds and protocol requirements; this enhances the representativeness of measured values [77].

- **Cycle Optimization**

  Manufacturers are now encouraged to develop solutions that perform reliably across a broader spectrum of real driving conditions, compared to cycle-specific strategies under NEDC.

A.6 NEDC vs. WLTP Fuel Consumption Test Protocols

The transition from NEDC to WLTP in late 2017 represents a major step forward in vehicle fuel consumption and emission testing, designed to reduce the gap between laboratory measurements and real-world driving outcomes.

This change aims to address widespread criticism of the NEDC's inability to replicate real-life driving conditions, which often led to underestimations of fuel consumption and CO^2 emissions [62, 34].

A.7 Limitations and Future Directions

While WLTP markedly reduces the gap, neither protocol fully accounts for all variables influencing fuel consumption and emissions. Efforts such as the Real Driving Emissions (RDE) initiative and ongoing development of portable emissions measurement systems (PEMS) aim to further bridge the divide between laboratory and real-world testing [72, 21].

Appendix B

Factors Influencing Fuel Consumption - An Overview

B.1 Vehicle Weight

Vehicle weight is one of the most decisive factors in fuel consumption. The energy required to move a vehicle is directly proportional to its mass, since greater weight demands more engine effort to overcome inertia and movement-related resistances such as tire rolling resistance. Studies show that a 10% increase in vehicle weight can lead to a 6% to 8% increase in fuel consumption [153].

In addition to this, weight impacts braking performance and safety, but from an energy efficiency perspective, it is a critical factor, especially in urban driving, where frequent stops and starts increase acceleration demands, also increasing fuel consumption. In electric and hybrid vehicles, the impact of weight is partially offset by energy recovery systems, though not entirely eliminated [153].

Therefore, reducing weight through the use of lighter materials or optimized design significantly contributes to lowering fuel consumption and pollutant emissions, positioning it as a crucial strategy for the automotive industry in its transition toward more sustainable vehicles [17].

B.2 Engine

The engine is the heart of the vehicle, and its technical characteristics have a direct and substantial impact on fuel consumption. Engines with larger displacement, higher power, and more torque usually consume more fuel, as they require more energy. However, the type of engine technology, such as direct injection, turbocharging, or hybrid systems, can significantly enhance energy efficiency.

Modern engines incorporate advanced electronic management systems that optimize air-fuel mixture and ignition timing, reducing consumption and emissions. In contrast, older engines with

carburetors or less efficient systems tend to consume more fuel, especially under variable driving conditions.

Additionally, engine maintenance, including checking key components such as filters, spark plugs, and the exhaust system, is essential to maintain efficiency and prevent increased fuel use due to wear and dirt buildup.[81]

B.3 Vehicle Age

Vehicle age is an important factor in determining fuel consumption, as natural wear of components reduces engine and auxiliary system efficiency. Older vehicles, typically equipped with less advanced technologies and more degradation-prone mechanical systems, often consume more fuel than newer models.

Moreover, the evolution of environmental standards means that older vehicles, with less efficient or nonexistent emissions control systems, emit more pollutants and consume more fuel. Lack of proper maintenance compounds these effects, leading to failures in ignition, injection, and exhaust systems, which in turn reduce energy performance.

Therefore, to maintain energy efficiency, it is essential to ensure rigorous maintenance and consider replacing older vehicles with newer models that incorporate more effective and environmentally friendly technologies[131].

B.4 Tires

Tires play a crucial role in fuel efficiency, as they are the vehicle's only contact point with the road and directly affect rolling resistance. Tires with pressure below the recommended level significantly increase resistance, forcing the engine to work harder and, consequently, increasing fuel consumption [92].

Besides pressure, tire width, compound type, and tread pattern also influence consumption. Wider tires or those made with softer compounds generate more friction with the pavement, increasing rolling resistance. The industry has developed low rolling resistance tires that meaningfully reduce fuel consumption and emissions. Additionally, proper tire maintenance, including regular inflation checks, alignment, and balancing, is essential to optimize energy efficiency and ensure driving safety [11].

B.5 Aerodynamics

Vehicle aerodynamics play a key role in fuel efficiency, especially at higher speeds, where air resistance increases exponentially. Aerodynamic drag forces the engine to use more energy to maintain speed and can account for up to 50% of total energy consumption at speeds above 55 mph [75].

Vehicles with aerodynamic designs, featuring smooth surfaces and specific elements (like spoilers and side skirts) to reduce drag, can substantially improve fuel efficiency. Removing unnecessary external accessories, such as roof racks, also helps reduce aerodynamic resistance.

Driver behavior also impacts this factor: driving at steady speeds and anticipating traffic reduces the resistance caused by sudden speed changes, contributing to improved energy efficiency.

B.6 Route

The route taken influences fuel consumption due to physical characteristics and traffic patterns. Urban routes, with frequent stops and starts, significantly increase fuel use compared to highway routes with steady speeds [53, 93].

The terrain also has a considerable effect: uphill sections increase engine load and consumption, while downhill stretches allow for reduced energy demand. Road conditions, such as surface irregularities and pavement type, also affect rolling resistance and vehicle component wear.

Optimizing the route, choosing paths with less traffic and more favorable terrain, as well as using smart navigation systems are effective strategies for reducing fuel consumption.

B.7 Driving Style

Driving style is one of the most impactful factors on fuel consumption. Drivers who accelerate abruptly, brake harshly, and keep engine RPMs high significantly increase fuel usage. Studies indicate that aggressive driving can raise fuel consumption by up to 33% in city driving and 5% on highways [29].

Excessive speeds increase air resistance, while very low speeds with inefficient engine operation also reduce efficiency. Efficient driving techniques, such as gradual acceleration, anticipating traffic, and maintaining a consistent speed, can reduce fuel consumption by around 20% [35].

Systems like start-stop, which shut off the engine at stops and restart it automatically, also help reduce fuel use in urban traffic.

B.8 Environment

Environmental conditions affect fuel consumption in several ways. Low temperatures increase oil viscosity and air density, leading to higher fuel use, especially in the first few minutes after a cold start.

High altitudes reduce the oxygen available for combustion, requiring more engine effort to maintain performance and increasing consumption. Adverse weather conditions, such as strong headwinds and rain, also raise movement resistance and fuel use. Additionally, fuel quality affects combustion efficiency: adulterated fuels or those with lower octane ratings can degrade performance and increase consumption [58].

B.9 Interactions Between Factors

Fuel consumption results from a complex interaction of various factors, which often amplify each other. For example, a heavy vehicle with underinflated tires and an old engine will consume more fuel than expected from the simple sum of individual impacts [36].

Aerodynamics and driving style are also interconnected: a car with excellent aerodynamic design loses that advantage if driven aggressively, with sudden accelerations and high speeds [136].

Similarly, route and environmental conditions combine effects: a mountainous route in a cold region imposes greater demands on the engine and increases consumption.

Therefore, an integrated analysis of these factors is essential to understand and optimize fuel use, and computational models have been developed to evaluate these interactions and guide design and driving strategies.

Appendix C

Factors Influencing Fuel Consumption - An Overview

Field	Type	Unit	Source or Derivation and Notes
ID	string	—	Extracted from URL using Regex; used to deduplicate records
Fuel Cons. Combined	float	L/100 km	Direct combined value, or $0.55xCity + 0.45xHighway$; or single component when only one exists; origin tagged separately
Fuel Cons. Source	String	-	One of Combined, Estimated, City, Highway, Missing
Horsepower (max)	int	hp	Maximum across conventional and electric power columns
Torque (max)	int	Nm	Maximum across torque columns after unit normalization to Nm
Acceleration 0-100 km/h	float	s	Prefer 0-100 km/h; else convert 0-60 mph with factor 1.04 after cleaning symbols
Top Speed	int	km/h	Converted from mph with factor 1.60934 when needed
Range (merged)	string	KM	Priority across cycles by production year; WLTP favored after 2017, NEDC otherwise; remove unit suffixes
Displacement (L)	float	L	Direct or parsed from version; defaults to 0 for EVs
Aspiration (simplified)	string	—	Rule based mapping to canonical classes such as Turbo, Turbo + Intercooler, Naturally Aspirated, etc.
Fuel System (simplified)	string	—	Rule based mapping to Direct Injection, Multi-Point Injection, Single-Point Injection, EFI, Mechanical FI, etc.
Transmission (normalized)	string	—	Keyword rules to Automatic or Manual; conservative default to Manual
Fuel Type (inferred)	string	—	Inferred from brand, model, version cues
Catalytic Converter (inferred)	string	—	Y/N inferred from fuel type and production year thresholds while respecting explicit entries
Body Type (inferred)	string	—	Curated mapping plus keyword heuristics; defaults to Undefined if unresolved
Electric Motor Type	string	—	Consolidated across up to four columns; standardized labels or joined unique cleaned values
Battery Capacity	float	kWh	Parsed from version text when present; Default otherwise
Battery Voltage	float	V	Default to 48V for mild hybrids when missing; else 0.0
Battery Chemistry	string	—	Inferred via brand, model, hybrid type, year and version cues;
Power-to-Weight	float	hp/kg	Derived from standardized horsepower and mass

Table C.1: Feature inventory and derivations used to construct the model-ready dataset

Appendix D

Unique Values Raw Layer

Column	Unique values
ID	50246
-	0
AC-1 Schuko	4
AC-Schuko	7
Acceleration 0 to 100 km/h (0 to 62 mph)	412
Acceleration 0 to 1000m	213
Acceleration 0 to 200 km/h (0 to 124 mph)	52
Acceleration 0 to 400m (1/4 quarter mile)	91
Acceleration 0 to 60 mph (0 to 96 Km/h)	308
AdBlue Tank	22
Aerodynamic drag coefficient - Cx	115
Approach angle (deg)	203
Aspiration	173
Average energy consumption	60
Average energy consumption WLTP	169
Battery capacity	308
Battery type	6
Battery voltage	71
Body	29
Bore x Stroke	2540
Brand	268
Breakover (Rampover) angle (deg)	177
CO2 emissions	519
CO2 emissions WLTP	310
Cargo box inside height	80
Cargo box inside length	97
Cargo width	38
Catalytic converter	16
Charging Time	128
Clutch Type	8

Table D.1: Unique values per raw layer column - 1

Column	**Unique values**
Compression Ratio	218
Curb Weight	2028
DC	56
Departure angle (deg)	196
Drive wheels - Traction - Drivetrain	4
Drives front wheels, regenerative braking Electric Engine Power	1
Drives rear wheels, regenerative braking Electric Engine Power	1
Dry Weight	980
Electric Engine Power	27
Electric Engine Torque	4
Electric engine 1 type	10
Electric engine 2 type	6
Electric engine 3 type	2
Electric engine 4 type	1
Electric engine type	5
Emission standard	104
Engine Alignment	2
Engine Code	3411
Engine Cooling	4
Engine Cooling - Capacity	80
Engine Position	3
Engine Type	1
Engine displacement	1164
Engine type - Number of cylinders	34
Euro NCAP	5329
Fast Charging Time	51
Fast charge current	114
Fast charge speed (WLTP)	5
Front Axle	510
Front Axle Electric Engine Power	94
Front Axle Electric Engine Torque	59
Front Electric Engine Power	1
Front Left Wheel Hub Electric Engine Power	1
Front Left Wheel Hub Electric Engine Torque	1
Front Right Wheel Hub Electric Engine Power	1
Front Right Wheel Hub Electric Engine Torque	1
Front head room	95
Front hip room	82
Front legroom	66
Front shoulder room	103

Table D.2: Unique values per raw layer column - 2

Column	Unique values
Fuel Consumption - Economy - City	215
Fuel Consumption - Economy - City NEDC	261
Fuel Consumption - Economy - City WLTC	10
Fuel Consumption - Economy - Combined	177
Fuel Consumption - Economy - Combined NEDC	190
Fuel Consumption - Economy - Combined WLTC	10
Fuel Consumption - Economy - Combined WLTP	155
Fuel Consumption - Economy - EPA City (- 2008)	38
Fuel Consumption - Economy - EPA City (2008 -)	33
Fuel Consumption - Economy - EPA Combined (- 2008)	38
Fuel Consumption - Economy - EPA Combined (2008 -)	34
Fuel Consumption - Economy - EPA Highway (- 2008)	42
Fuel Consumption - Economy - EPA Highway (2008 -)	35
Fuel Consumption - Economy - Extra high WLTP	101
Fuel Consumption - Economy - High WLTP	93
Fuel Consumption - Economy - Highway	111
Fuel Consumption - Economy - Low WLTP	195
Fuel Consumption - Economy - Medium WLTP	116
Fuel Consumption - Economy - Open road	111
Fuel Consumption - Economy - Open road NEDC	129
Fuel Consumption - Economy - Open road WLTC	9
Fuel System	1827
Fuel Tank Capacity	254
Fuel type	12
Gearbox Electric Engine Power	27
Gearbox Electric Engine Torque	1
Gearbox housing Electric Engine Power	5
Generation	2150
Ground clearance	211
Height	898
Horsepower	590
Length	1687
Loading height	53
Lubrication	2339
Max. Towing Capacity Weight	206
Max. width at wheelhouse	43
Maximum torque	514
Model	5584
Net battery capacity	157
Num. of Doors	6
Num. of Seats	22
Number of electric engines	4
Number of valves	23
Production number	202

Table D.3: Unique values per raw layer column - 3

Column	**Unique values**
Range	757
Range (EPA)	251
Range (NEDC)	70
Range (WLTC)	2
Range (WLTP)	654
Rear (Twin Motor Unit) Electric Engine Power	1
Rear Axle	507
Rear Axle Electric Engine Power	74
Rear Axle Electric Engine Torque	46
Rear Left Wheel Hub Electric Engine Power	1
Rear Left Wheel Hub Electric Engine Torque	1
Rear Right Wheel Hub Electric Engine Power	1
Rear Right Wheel Hub Electric Engine Torque	1
Rear door height	42
Rear door width	22
Rear head room	98
Rear hip room	85
Rear seat legroom	91
Rear shoulder room	105
Seat distribution	21
Side door height	30
Side door width	33
Third row head room	10
Third row legroom	11
Third row shoulder room	12
Top Speed	194
Total System Power	205
Total System Torque	112
Total electric power	242
Total electric torque	194
Transmission Electric Engine Power	42
Transmission Gearbox - Number of speeds	2247
Trunk / Boot First Row	565
Trunk / Boot Second Row	146
Trunk / Boot Third Row	27
Trunk / Boot capacity	769
URL	50246
Version	25814
Wallbox 1-phase	17
Wallbox 3-phase	7
Weight/Power Output Ratio	296
Wheelbase	829
Width	581
Width with mirrors	229
Years	770

Table D.4: Unique values per raw layer column - 4

Appendix E

Body Types Dictionary

Make	Model	Body Type
Alfa Romeo	Giulietta II	Hatchback
Alfa Romeo	MiTo	Hatchback
Alfa Romeo	Crosswagon	Estate
Alfa Romeo	156	Sedan
Alfa Romeo	159	Sedan
Alfa Romeo	GT	Coupe
Alfa Romeo	Brera	Coupe

Table E.1: Appendix 6 - Body Type Mapping: Alfa Romeo

Make	Model	Body Type
Aston Martin	DBS Superleggera	Coupe
Aston Martin	DB9	Coupe
Aston Martin	DB11	Coupe
Aston Martin	Vantage	Coupe
Aston Martin	Vanquish	Coupe
Aston Martin	Rapide	Sedan

Table E.2: Appendix 6 - Body Type Mapping: Aston Martin

Make	Model	Body Type
Audi	A1	Hatchback
Audi	A3	Hatchback
Audi	A3 Sedan	Sedan
Audi	A4	Sedan
Audi	A4 Avant	Estate
Audi	A5	Coupe
Audi	A5 Sportback	Sedan
Audi	A5 Cabriolet	Convertible
Audi	A6	Sedan
Audi	A6 Avant	Estate
Audi	A7	Sedan
Audi	A8	Sedan
Audi	Q2	SUV
Audi	Q3	SUV
Audi	Q5	SUV
Audi	Q7	SUV
Audi	Q8	SUV
Audi	TT	Coupe
Audi	R8	Coupe

Table E.3: Appendix 6 - Body Type Mapping: Audi

Make	Model	Body Type
BMW	1 Series	Hatchback
BMW	1 Series Coupe	Coupe
BMW	2 Series	Coupe
BMW	2 Series Gran Coupe	Sedan
BMW	3 Series	Sedan
BMW	3 Series Touring	Estate
BMW	4 Series Coupe	Coupe
BMW	4 Series Gran Coupe	Sedan
BMW	5 Series	Sedan
BMW	5 Series Touring	Estate
BMW	6 Series Coupe	Coupe
BMW	6 Series Gran Coupe	Sedan
BMW	7 Series	Sedan
BMW	8 Series	Coupe
BMW	X1	SUV
BMW	X3	SUV
BMW	X5	SUV
BMW	X6	SUV
BMW	Z4	Convertible

Table E.4: Appendix 6 - Body Type Mapping: BMW

Make	Model	Body Type
Bugatti	Centodieci	Coupe
Bugatti	Divo	Coupe
Bugatti	Chiron	Coupe
Bugatti	Veyron	Coupe

Table E.5: Appendix 6 - Body Type Mapping: Bugatti

Make	Model	Body Type
Citro"en	C1	Hatchback
Citro"en	C2	Hatchback
Citro"en	C3	Hatchback
Citro"en	C3 Picasso	MPV
Citro"en	C4	Hatchback
Citro"en	C4 Picasso	MPV
Citro"en	C5	Sedan
Citro"en	C5 Tourer	Estate
Citro"en	C6	Sedan
Citro"en	C8	MPV

Table E.6: Appendix 6 - Body Type Mapping: Citroën

Make	Model	Body Type
Ferrari	458 Italia	Coupe
Ferrari	488 GTB	Coupe
Ferrari	F8 Tributo	Coupe
Ferrari	812 Superfast	Coupe
Ferrari	Portofino	Convertible
Ferrari	California	Convertible
Ferrari	GTC4Lusso	Coupe
Ferrari	F355	Coupe
Ferrari	F430	Coupe

Table E.7: Appendix 6 - Body Type Mapping: Ferrari

Make	Model	Body Type
Fiat	500	Hatchback
Fiat	Panda	Hatchback
Fiat	Punto	Hatchback
Fiat	Bravo	Hatchback
Fiat	Stilo	Hatchback
Fiat	Tipo	Sedan
Fiat	Linea	Sedan
Fiat	Croma	Estate
Fiat	Multipla	MPV
Fiat	Sedici	SUV
Fiat	Fiorino	Van

Table E.8: Appendix 6 - Body Type Mapping: Fiat

Make	Model	Body Type
Ford	Fiesta	Hatchback
Ford	Focus	Hatchback
Ford	Mondeo	Sedan
Ford	Mustang	Coupe
Ford	Kuga	SUV
Ford	Explorer	SUV
Ford	Edge	SUV

Table E.9: Appendix 6 - Body Type Mapping: Ford

Make	Model	Body Type
Honda	Accord	Sedan
Honda	Civic	Hatchback
Honda	CR-V	SUV
Honda	Jazz	Hatchback

Table E.10: Appendix 6 - Body Type Mapping: Honda

Make	Model	Body Type
Hyundai	i10	Hatchback
Hyundai	i20	Hatchback
Hyundai	i30	Hatchback
Hyundai	Tucson	SUV
Hyundai	Santa Fe	SUV

Table E.11: Appendix 6 - Body Type Mapping: Hyundai

Make	Model	Body Type
Jaguar	F-Type	Coupe
Jaguar	XJ	Sedan
Jaguar	XF	Sedan
Jaguar	XE	Sedan

Table E.12: Appendix 6 - Body Type Mapping: Jaguar

Make	Model	Body Type
Jeep	Wrangler	SUV
Jeep	Grand Cherokee	SUV
Jeep	Renegade	SUV

Table E.13: Appendix 6 - Body Type Mapping: Jeep

Make	Model	Body Type
Kia	Picanto	Hatchback
Kia	Rio	Hatchback
Kia	Ceed	Hatchback
Kia	Sportage	SUV
Kia	Sorento	SUV

Table E.14: Appendix 6 - Body Type Mapping: Kia

Make	Model	Body Type
Land Rover	Defender	SUV
Land Rover	Discovery	SUV
Land Rover	Range Rover	SUV

Table E.15: Appendix 6 - Body Type Mapping: Land Rover

Make	Model	Body Type
Lamborghini	Gallardo	Coupe
Lamborghini	Huracan	Coupe
Lamborghini	Aventador	Coupe
Lamborghini	Urus	SUV

Table E.16: Appendix 6 - Body Type Mapping: Lamborghini

Make	Model	Body Type
Lexus	IS	Sedan
Lexus	RX	SUV
Lexus	NX	SUV
Lexus	GS	Sedan

Table E.17: Appendix 6 - Body Type Mapping: Lexus

Make	Model	Body Type
Maserati	Ghibli	Sedan
Maserati	Quattroporte	Sedan
Maserati	GranTurismo	Coupe

Table E.18: Appendix 6 - Body Type Mapping: Maserati

Make	Model	Body Type
Mazda	2	Hatchback
Mazda	3	Sedan
Mazda	6	Sedan
Mazda	CX-3	SUV
Mazda	CX-5	SUV

Table E.19: Appendix 6 - Body Type Mapping: Mazda

Make	Model	Body Type
McLaren	P1	Coupe
McLaren	570S	Coupe
McLaren	720S	Coupe
McLaren	Artura	Coupe

Table E.20: Appendix 6 - Body Type Mapping: McLaren

Make	Model	Body Type
Mercedes Benz	A-Class	Hatchback
Mercedes Benz	C-Class	Sedan
Mercedes Benz	CLA	Sedan
Mercedes Benz	E-Class	Sedan
Mercedes Benz	GLA	SUV
Mercedes Benz	GLC	SUV
Mercedes Benz	GLE	SUV
Mercedes Benz	S-Class	Sedan

Table E.21: Appendix 6 - Body Type Mapping: Mercedes Benz

Make	Model	Body Type
Mini	Countryman	SUV
Mini	Clubman	Estate
Mini	Cooper	Hatchback

Table E.22: Appendix 6 - Body Type Mapping: Mini

Make	Model	Body Type
Mitsubishi	Colt	Hatchback
Mitsubishi	Lancer	Sedan
Mitsubishi	Outlander	SUV

Table E.23: Appendix 6 - Body Type Mapping: Mitsubishi

Make	Model	Body Type
Nissan	350Z	Coupe
Nissan	GT-R	Coupe
Nissan	Juke	SUV
Nissan	Leaf	Hatchback
Nissan	Micra	Hatchback
Nissan	Note	Hatchback
Nissan	Qashqai	SUV
Nissan	X-Trail	SUV

Table E.24: Appendix 6 - Body Type Mapping: Nissan

Make	Model	Body Type
Opel	Astra	Hatchback
Opel	Corsa	Hatchback
Opel	Insignia	Sedan
Opel	Meriva	MPV
Opel	Mokka	SUV
Opel	Vectra	Sedan
Opel	Zafira	MPV

Table E.25: Appendix 6 - Body Type Mapping: Opel

Make	Model	Body Type
Peugeot	207	Hatchback
Peugeot	208	Hatchback
Peugeot	3008	SUV
Peugeot	306	Sedan
Peugeot	307	Hatchback
Peugeot	308	Hatchback
Peugeot	4007	SUV
Peugeot	405	Sedan
Peugeot	406	Sedan
Peugeot	407	Sedan
Peugeot	5008	SUV
Peugeot	RCZ	Coupe

Table E.26: Appendix 6 - Body Type Mapping: Peugeot

Make	Model	Body Type
Porsche	911	Coupe
Porsche	Boxster	Convertible
Porsche	Cayman	Coupe
Porsche	Cayenne	SUV
Porsche	Macan	SUV
Porsche	Panamera	Sedan
Porsche	Taycan	Sedan

Table E.27: Appendix 6 - Body Type Mapping: Porsche

Make	Model	Body Type
Renault	Captur	SUV
Renault	Clio	Hatchback
Renault	Clio RS	Hatchback
Renault	Espace	MPV
Renault	Grand Scenic	MPV
Renault	Kadjar	SUV
Renault	Koleos	SUV
Renault	Laguna	Sedan
Renault	Laguna Coupe	Coupe
Renault	Laguna Estate	Estate
Renault	Megane	Hatchback
Renault	Megane RS	Hatchback
Renault	Megane Estate	Estate
Renault	Scenic	MPV
Renault	Talisman	Sedan
Renault	Talisman Estate	Estate

Table E.28: Appendix 6 - Body Type Mapping: Renault

Make	Model	Body Type
Saab	900	Hatchback
Saab	9000	Sedan
Saab	9-3	Sedan
Saab	9-5	Sedan

Table E.29: Appendix 6 - Body Type Mapping: Saab

Make	Model	Body Type
Seat	Alhambra	MPV
Seat	Altea	Hatchback
Seat	Ateca	SUV
Seat	Arona	SUV
Seat	Ibiza	Hatchback
Seat	Leon	Hatchback
Seat	Toledo	Sedan

Table E.30: Appendix 6 - Body Type Mapping: Seat

Make	Model	Body Type
Škoda	Fabia	Hatchback
Škoda	Karoq	SUV
Škoda	Kodiaq	SUV
Škoda	Octavia	Sedan
Škoda	Superb	Sedan

Table E.31: Appendix 6 - Body Type Mapping: vSkoda

Make	Model	Body Type
Smart	Fortwo	Hatchback
Smart	Forfour	Hatchback

Table E.32: Appendix 6 - Body Type Mapping: Smart

Make	Model	Body Type
Subaru	Forester	SUV
Subaru	Impreza	Hatchback
Subaru	Legacy	Sedan
Subaru	Outback	Estate

Table E.33: Appendix 6 - Body Type Mapping: Subaru

Make	Model	Body Type
Suzuki	SX4	SUV
Suzuki	Swift	Hatchback
Suzuki	Vitara	SUV

Table E.34: Appendix 6 - Body Type Mapping: Suzuki

Make	Model	Body Type
Toyota	Aygo	Hatchback
Toyota	Corolla	Sedan
Toyota	Auris	Hatchback
Toyota	Hilux	Pickup
Toyota	Land Cruiser	SUV
Toyota	Prius	Hatchback
Toyota	RAV4	SUV
Toyota	Yaris	Hatchback

Table E.35: Appendix 6 - Body Type Mapping: Toyota

Make	Model	Body Type
Volkswagen	Beetle	Hatchback
Volkswagen	Golf	Hatchback
Volkswagen	Golf Variant	Estate
Volkswagen	Passat	Sedan
Volkswagen	Passat Variant	Estate
Volkswagen	Polo	Hatchback
Volkswagen	Scirocco	Coupe
Volkswagen	Sharan	MPV
Volkswagen	Tiguan	SUV
Volkswagen	Touareg	SUV
Volkswagen	Touran	MPV

Table E.36: Appendix 6 - Body Type Mapping: Volkswagen

Make	Model	Body Type
Vauxhall	Astra	Hatchback
Vauxhall	Corsa	Hatchback
Vauxhall	Insignia	Sedan
Vauxhall	Meriva	MPV
Vauxhall	Mokka	SUV
Vauxhall	Vectra	Sedan
Vauxhall	Zafira	MPV

Table E.37: Appendix 6 - Body Type Mapping: Vauxhall

Make	**Model**	**Body Type**
Volvo	240	Sedan
Volvo	740	Sedan
Volvo	850	Sedan
Volvo	940	Sedan
Volvo	S40	Sedan
Volvo	S60	Sedan
Volvo	S70	Sedan
Volvo	S80	Sedan
Volvo	V40	Estate
Volvo	V50	Estate
Volvo	V60	Estate
Volvo	V70	Estate
Volvo	XC40	SUV
Volvo	XC60	SUV
Volvo	XC70	Estate
Volvo	XC90	SUV

Table E.38: Appendix 6 - Body Type Mapping: Volvo

Appendix F

Project Timeline Gantt

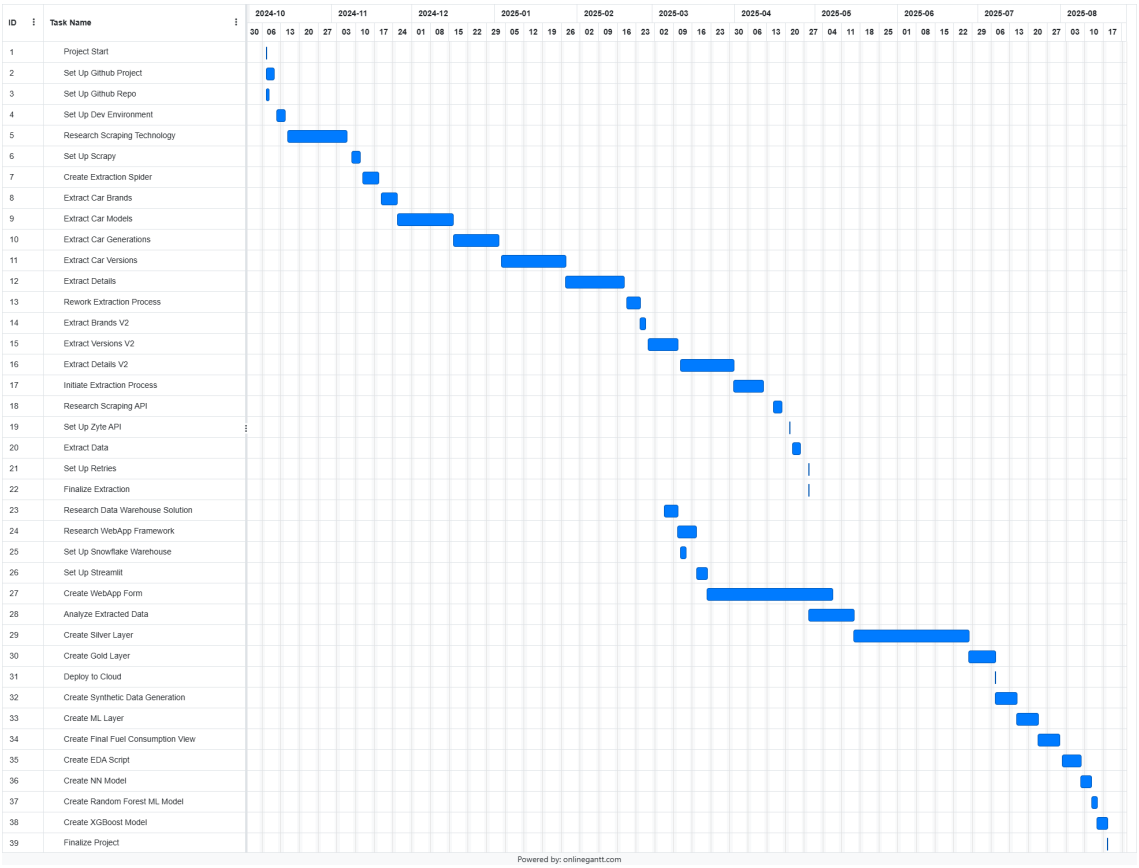

Figure F.1: Gantt graph highlighting the project timeline

```
INSERT INTO CAR_SPECS_GOLD (ID, brand, model, version, adblue_tank, aspiration,
    battery_capacity,
    battery_type,
    battery_voltage, body, bore, catalytic_converter, co2,
    compression_ratio, cylinders,
    displacement, doors, drag_coefficient, drive, electric_engine_type,
    energy_consumption,
    engine_type, front_axle, fuel_consumption_combined,
    fuel_consumption_source, fuel_system,
    fuel_type, height, horsepower, length, number_of_valves,
    production_start, production_end,
    range, rear_axle, stroke, top_speed, torque, gearbox, url, weight,
    hp_kg_ratio, wheelbase, width
)
WITH model_normalized as (
    SELECT
        ID
        ,brand
        ,model
        ,version
        ,adblue_tank
        ,aspiration
        ,battery_capacity
        ,battery_type
        ,battery_voltage
        ,body
        ,bore
        ,catalytic_converter
        ,co2
        ,round(compression_ratio,2) compression_ratio
        ,cylinders
        ,displacement
        ,doors
        ,drag_coefficient
        ,drive
        ,electric_engine_type
        ,energy_consumption
        ,engine_type
        ,FIRST_VALUE(front_axle) IGNORE NULLS OVER (PARTITION BY model ORDER BY
            model) as front_axle
        ,fuel_consumption_combined
        ,fuel_consumption_source
        ,fuel_system
        ,fuel_type
        ,FIRST_VALUE(height) IGNORE NULLS OVER (PARTITION BY model ORDER BY model)
            as height
        ,horsepower
        ,FIRST_VALUE(length) IGNORE NULLS OVER (PARTITION BY model ORDER BY model)
            AS length
```

```
        ,number_of_valves
        ,production_start
        ,production_end
        ,range
        ,FIRST_VALUE(rear_axle) IGNORE NULLS OVER (PARTITION BY model ORDER BY
            model) as rear_axle
        ,stroke
        ,time_to_100
        ,top_speed
        ,torque
        ,"TRANSMISSION_GEARBOX_-_NUMBER_OF_SPEEDS" gearbox
        ,url
        ,weight
        ,weight_power_ratio -- calculate
        ,FIRST_VALUE(wheelbase) IGNORE NULLS OVER (PARTITION BY model ORDER BY
            model) AS wheelbase
        ,FIRST_VALUE(width) IGNORE NULLS OVER (PARTITION BY model ORDER BY model)
            AS width

    FROM CAR_SPECS_SILVER
    WHERE 1=1
    AND horsepower is not null
    AND torque is not null
    AND fuel_consumption_combined is not null
    and range is not null
    and displacement < 60),
    medians AS (
    SELECT
        MEDIAN(height) AS median_height,
        MEDIAN(top_speed) AS median_top_speed
    FROM model_normalized)
SELECT ID,
 brand,
 model,
 version,
 adblue_tank,
 aspiration,
 battery_capacity,
 battery_type,
 battery_voltage,
 COALESCE(body, 'Undefined') AS body,
 ROUND(bore, 0)  AS bore,
 catalytic_converter,
 ROUND(co2, 0)   AS co2,
 compression_ratio,
 cylinders,
 displacement,
 ROUND(COALESCE(doors, 3), 0) AS doors,
 ROUND(drag_coefficient, 2) AS drag_coefficient,
```

```
92  drive,
93  COALESCE(electric_engine_type, 'None') AS electric_engine_type,
94  COALESCE(energy_consumption, 0) AS energy_consumption,
95  CASE
96      WHEN engine_type IS NOT NULL THEN engine_type
97      WHEN brand = 'Porsche' THEN 'Boxer'
98      WHEN brand = 'Subaru' THEN 'Boxer'
99      ELSE 'Inline'
100 END AS engine_type,
101 front_axle,
102 fuel_consumption_combined,
103 fuel_consumption_source,
104 fuel_system,
105 fuel_type,
106 COALESCE(height, md.median_height) AS height,
107 horsepower,
108 length,
109 case
110    when number_of_valves = 'null' then 4*cylinders
111    when number_of_valves is null then 4*cylinders
112    else number_of_valves
113 end as number_of_valves,
114 production_start,
115 production_end,
116 range,
117 rear_axle,
118 stroke,
119 COALESCE(top_speed, md.median_top_speed) AS top_speed,
120 torque,
121 gearbox,
122 url,
123 weight,
124 ROUND(weight / horsepower, 1) AS hp_kg_ratio,
125 wheelbase,
126 width
127 FROM model_normalized mn
128 CROSS JOIN medians md
```

Listing F.1: Consumption view code

References

[1] H Abediasl et al. Real-time vehicular fuel consumption estimation using machine learning and obd data. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 2024. Accessed 2025-05-26.

[2] Amazon Web Services. Data warehouse system architecture — amazon redshift, 2025. URL https://docs.aws.amazon.com/redshift/latest/dg/c_high_level_system_architecture.html. Accessed 2025-09-08.

[3] Amazon Web Services. Amazon redshift. https://aws.amazon.com/redshift/, 2025. Accessed 2025-06-21.

[4] Amazon Web Services (AWS). O que é um certificado ssl? – explicação sobre certificados ssl/tls – aws, 2025. URL https://aws.amazon.com/what-is/ssl-certificate/. Accessed: 2025-02-07.

[5] Apify Blog. Scrapy vs selenium: when to use them for web scraping. https://blog.apify.com/scrapy-vs-selenium/, 2025. Accessed 2025-06-11.

[6] Applitools Blog. 2020 most popular front end automation testing tools. https://applitools.com/blog/2020-front-end-automation-testing/, 2020. Accessed 2025-06-07.

[7] Huthaifa I. Ashqar, Mahmoud Obaid, Ahmed Jaber, Rashed Ashqar, Nour O. Khanfar, and Mohammed Elhenawy. Incorporating driving behavior into vehicle fuel consumption prediction: methodology development and testing. *Discover Sustainability*, 5:34, 2024. doi: 10.1007/s43621-024-00511-z. Accessed 2025-05-18.

[8] Australian Automobile Association. New tests identify real-world fuel consumption gaps. https://www.aaa.asn.au/2024/02/new-tests-identify-real-world-fuel-consumption-gaps/, 2024. Accessed 2025-08-14.

[9] E. Bagheri, M. M. Tehrani, M. Azadi, and A. Moosavian. Impact of driving characteristic parameters and vehicle type on fuel consumption and emissions performance over real driving cycles. *PLoS ONE*, 20(1):e0317098, 2025. doi: 10.1371/journal.pone.0317098. Accessed 2025-05-17.

[10] BlazeMeter Blog. Selenium vs scrapy: Which one should you choose for web scraping? https://www.blazemeter.com/blog/scrapy-vs-selenium, 2022. Accessed 2025-06-10.

[11] Michelin Brasil. Como os pneus podem aumentar a quilometragem rodada e ajudar na economia de combustível? https://www.michelin.com.br/auto/conselhos/escolher-pneus/aumentar-quilometragem-rodada-economia-de-combustivel, 2025. Accessed 2025-07-19.

[12] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. Accessed 2025-05-25.

[13] Bright Data. Residential proxies, 2025. URL https://brightdata.com/products/residential-proxy. Accessed 2025-06-17.

[14] BrightData. Scrapy vs. selenium for web scraping. https://brightdata.com/blog/web-data/scrapy-vs-selenium, 2025. Accessed 2025-06-13.

[15] BrowserStack. Top 10 advantages of selenium for web application testing. https://www.browserstack.com/guide/selenium-framework, 2025. Accessed 2025-06-09.

[16] Rafael Canal, Felipe K. Riffel, and Giovani Gracioli. Machine learning for real-time fuel consumption prediction and driving profile classification based on ecu data. *IEEE Transactions on Intelligent Transportation Systems*, 2023. doi: 10.1109/TITS.2023.3261459. Accessed 2025-05-21.

[17] S. Cecchel, D. Chindamo, E. Turrini, C. Carnevale, G. Cornacchia, M. Gadola, A. Panvini, M. Volta, D. Ferrario, and R. Golimbioschi. Impact of reduced mass of light commercial vehicles on fuel consumption, co2 emissions, air quality, and socio-economic costs. *Science of The Total Environment*, 613-614:409–417, 2018. doi: https://doi.org/10.1016/j.scitotenv.2017.09.081. Accessed 2025-07-19.

[18] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016. Accessed 2025-05-21.

[19] Automotive Data. Automotive data, 2025. URL https://www.auto-data.net. Accessed 2024-10-05.

[20] DataFlair. Everything about selenium webdriver architecture and its components, 05 2021. URL https://data-flair.training/blogs/selenium-webdriver-architecture/. Accessed 2025-09-08.

[21] G. D'Errico, C. Cintolesi, F. Di Genova, S. Mastrogiovanni, G. Polverino, M. Reale, and R. Vellone. Real drive well-to-wheel energy analysis of conventional and electrified car powertrains. *Energies*, 13(18):4788, 2020. doi: 10.3390/en13184788. Accessed 2025-05-11.

[22] Design Gurus. What are the disadvantages of snowflake? https://www.designgurus.io/answers/detail/what-are-the-disadvantages-of-snowflake, 2024. Accessed 2025-06-19.

[23] DieselNet. Ece 15 + eudc / nedc - emission test cycles, 1999. URL https://dieselnet.com/standards/cycles/ece_eudc.php. Accessed 2025-08-14.

[24] Dmitry Baraishuk. Playwright vs. selenium in 2025: Key differences for test automation. https://dev.to/dmitrybaraishuk/

playwright-vs-selenium-in-2025-key-differences-for-test-automation-8ji, 2025. Accessed 2025-06-10.

[25] A. Doruk and M. A. Bayram. Predicting vehicle fuel efficiency: A comparative analysis of machine learning models on the auto mpg dataset. *Index Copernicus International*, 2023. Accessed 2025-06-23.

[26] Dremio. Getting locked-in and locked-out with snowflake. https://www.dremio.com/blog/getting-locked-in-and-locked-out-with-snowflake/, 2024. Accessed 2025-06-21.

[27] Educative.io. Scrapy vs. selenium. https://www.educative.io/answers/scrapy-vs-selenium, 2023. Accessed 2025-06-16.

[28] EncyCarPedia. Encycarpedia, 2025. URL https://www.encycarpedia.com. Accessed 2024-10-05.

[29] Environmental Protection Agency. Data on cars used for testing fuel economy. https://www.epa.gov/compliance-and-fuel-economy-data/data-cars-used-testing-fuel-economy, 2025. Accessed 2025-05-08.

[30] Perttu Anttila et al. Analyzing the fundamental factors affecting vehicle fuel consumption: A global study. *Transportation Research Interdisciplinary Perspectives*, 13:100534, 2022. doi: 10.1016/j.trip.2022.100534. Accessed 2025-05-18.

[31] EuroConsumers. Mile21 project site, 2022. URL https://www.euroconsumers.org/projects/mile-21/. Accessed 2024-12-15.

[32] European Commission. Report under article 12(3) of regulation (eu) 2019/631 on the evolution of the real-world co2 emissions gap for passenger cars and light commercial vehicles, March 2024. URL https://climate.ec.europa.eu/document/download/b644dafe-1385-4b56-98d9-21e7e9f3601b_en?filename=report.pdf. Accessed: 2025-10-05.

[33] European Commission Joint Research Centre. Fuel consumption and co_2 emissions of passenger cars over the type-approval and real-world driving regimes. Technical report, European Commission, 2017. URL https://publications.jrc.ec.europa.eu/repository/bitstream/JRC107662/kjna28724enn.pdf. Accessed 2025-05-20.

[34] European Commission Joint Research Centre. From nedc to wltp: Effect on the type-approval co_2 emissions of passenger cars in europe. Technical report, European Commission, 2020. URL https://publications.jrc.ec.europa.eu/repository/bitstream/JRC107662/kjna28724enn.pdf. Accessed 2025-08-14.

[35] Panagiotis Fafoutellis, Eleni G. Mantouka, and Eleni I. Vlahogianni. Eco-driving and its impacts on fuel efficiency: An overview of technologies and data-driven methods. *Sustainability*, 13(1):226, 2020. doi: 10.3390/su13010226. Accessed 2025-05-08.

[36] Waleed F. Faris, Hesham A. Rakha, Raed I. Kafafy, Moumen Idres, and Salah Elmoselhy. Vehicle fuel consumption and emission modelling: an in-depth literature review. *International Journal of Vehicle Systems Modelling and Testing*, 6(3/4):318–395, 2011. doi: 10.1504/IJVSMT.2011.044232. Accessed 2025-08-14.

[37] De-Cheng Feng, Wen-Jie Wang, Sujith Mangalathu, and Ertugrul Taciroglu. Machine learning-based framework for assessing seismic performance of reinforced concrete frame structures. *Journal of Structural Engineering*, 147(11), August 2021. doi: 10.1061/(ASCE)ST.1943-541X.0003115. URL https://doi.org/10.1061/(ASCE)ST.1943-541X.0003115.

[38] Firecrawl. Best open-source web scraping libraries in 2025. https://www.firecrawl.dev/blog/best-open-source-web-scraping-libraries, 2025. Accessed 2025-06-14.

[39] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001. Accessed 2025-05-21.

[40] Chao Gao, Jinliang Xu, Miao Jia, and Zhenhua Sun. Correlation between carbon emissions, fuel consumption of vehicles and speed limit on expressway. *Transportation Research Interdisciplinary Perspectives*, 16:100873, 2024. doi: 10.1016/j.trip.2024.100873. Accessed 2025-05-31.

[41] Sichen Gao, Yuhua Zong, Fei Ju, Qun Wang, Weiwei Huo, Liangmo Wang, and Tao Wang. Scenario-oriented adaptive ecms using speed prediction for fuel cell vehicles in real-world driving. *Energy*, 329:126927, 2024. doi: 10.1016/j.energy.2024.126927. Accessed 2025-06-03.

[42] GeeksforGeeks. Python requests tutorial. https://www.geeksforgeeks.org/python/python-requests-tutorial/, 2025. Accessed 2025-06-17.

[43] Barouch Giechaskiel, Dimitrios Komnos, and Georgios Fontaras. Impacts of extreme ambient temperatures and road gradient on energy consumption and co_2 emissions of a euro 6d-temp gasoline vehicle. *Energies*, 14(19):6195, 2021. doi: 10.3390/en14196195. URL https://www.mdpi.com/1996-1073/14/19/6195. Accessed 2025-08-14.

[44] Destin Gong. Semi-automated exploratory data analysis (eda) in python. https://medium.com/data-science/semi-automated-exploratory-data-analysis-eda-in-python-7f96042c9809, 2021. Accessed 2025-10-11.

[45] Sidalina Gonçalves, José Biléu Ventura, Orlando Lima Rua, Rui Dias, and Rosa Galvão. Big data as an emerging paradigm in organisations' management: A bibliometric analysis. *Journal of Ecohumanism*, 1(1), 2022. Accessed 2025-06-18.

[46] GoodCarBadCar.net. Global car sales analysis 2017, 2017. URL https://www.goodcarbadcar.net/global-car-sales-analysis-2017/. Accessed: 2025-06-18.

[47] M. Grinberg. *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, 2nd edition, 2018. Accessed 2025-06-22.

[48] A. Gupta, R. Raskar, and S. Shah. Comparative review of python frameworks for rapid machine learning dashboard deployment. *Journal of Software Engineering and Applications*, 15(3):120–133, 2022. Accessed 2025-06-01.

[49] H2Kinfosys. What are the drawbacks and challenges of using selenium? https://www.h2kinfosys.com/blog/what-are-the-drawbacks-and-challenges-of-using-selenium/, 2025. Accessed 2025-06-09.

[50] S. S. Haghshenas, M. Ostadrahimi, and S. Mollaei. Fuel consumption and co2 emissions prediction in road transport using a hybrid deep learning approach. *Transportation Engineering*, 10:100254, 2025. doi: 10.1016/j.treng.2025.100254. Accessed 2025-06-23.

[51] Mohamed A. HAMED, Mohammed H.Khafagy, and Rasha M.Badry. Fuel consumption prediction model using machine learning. *International Journal of Advanced Computer Science and Applications*, 12(11), 2021. doi: 10.14569/IJACSA.2021.0121146. Accessed 2025-05-12.

[52] J. Hanzl, B. Šarkan, and A. Kuranc. Research on the effect of road height profile on fuel consumption during vehicle acceleration. *Technologies*, 10(6):128, 2022. doi: 10.3390/technologies10060128. Accessed 2025-05-18.

[53] Maria Vaíres Nunes Silva Hartmann. Análise do consumo de combustível em relação às características da via e à condição superficial do pavimento: estudo de um trecho da rodovia br-116. Dissertação de mestrado, Universidade Federal do Ceará, Fortaleza, agosto 2024. URL https://repositorio.ufc.br/bitstream/riufc/80600/3/2024_dis_mvnshartmann.pdf. Accessed 2025-08-14.

[54] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, 47:98–115, 2015. doi: 10.1016/j.is.2014.07.006. Accessed 2025-06-17.

[55] Yongming He, Jia Kang, Yulong Pei, Bin Ran, and Yuting Song. Research on influencing factors of fuel consumption on superhighway based on dematel-ism model. *Energy Policy*, 158:112562, 2021. doi: 10.1016/j.enpol.2021.112562. Accessed 2025-05-18.

[56] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. Accessed 2025-05-28.

[57] A. Holovaty and J. Kaplan-Moss. *The Definitive Guide to Django: Web Development Done Right*. Apress, 4th edition, 2022. Accessed 2025-06-22.

[58] IEA. Fuel economy in major car markets: Technology and policy drivers 2005–2017. Working paper, International Energy Agency, 2019. URL https://iea.blob.core.windows.net/assets/66965fb0-87c9-4bc7-990d-a509a1646956/Fuel_Economy_in_Major_Car_Markets.pdf. Accessed 2025-05-08.

[59] IEA and (UNEP). International comparison of light-duty vehicle fuel economy 2005-2015: Ten years of fuel economy benchmarking. Technical report, OECD/IEA and UNEP, 2017. URL https://iea.blob.core.windows.net/assets/6a9d5d7b-89cc-4b7e-97ef-dd40551fef26/wp15ldvcomparison.pdf. Working Paper 15; Accessed: 2025-08-05.

[60] Infomineo. The 10 best web scraping tools for 2025. https://infomineo.com/blog/best-web-scraping-tools-in-2025-top-picks-for-data-extraction/, 2025. Accessed 2025-06-16.

[61] International Council on Clean Transportation. Real-world usage of plug-in hybrid electric vehicles: Fuel consumption, electric driving, and co2 emissions. White paper, ICCT, 2020. URL https://theicct.org/sites/default/files/publications/PHEV-white%20paper-sept2020-0.pdf. Accessed 2025-05-20.

[62] International Council on Clean Transportation. On the way to 'real-world' co_2 values: The european passenger car market in transition, 2020. URL https://theicct.org/sites/default/files/publications/On-the-way-to-real-world-WLTP_May2020.pdf. Accessed 2025-08-14.

[63] International Council on Clean Transportation. Size or mass? the technical rationale behind attribute-based standards for vehicle efficiency, 2021. URL https://theicct.org/wp-content/uploads/2021/06/ICCTpaper_sizewt_final.pdf. Accessed 2025-05-20.

[64] International Council on Clean Transportation (ICCT). Laboratory to real-world fuel consumption and co2 emissions of eu passenger cars – an update. https://theicct.org/publications/laboratory-to-real-world-fuel-consumption-eu-passenger-cars-update/, 2021. Accessed 2025-06-01.

[65] International Transport Forum (ITF). Transport outlook 2017. Technical report, Organisation for Economic Co-operation and Development (OECD), Paris, France, 2017. URL https://www.itf-oecd.org/transport-outlook-2017. Accessed 2025-06-02.

[66] IPRoyal Blog. Python requests library (2025 guide). https://iproyal.com/blog/python-requests-library/, 2025. Accessed 2025-06-16.

[67] JMP Statistics Knowledge Portal. One-way anova. https://www.jmp.com/en/statistics-knowledge-portal/one-way-anova, 2025. Accessed 2025-06-29.

[68] MCA K. Madhusudhan Reddy and Cherukuri Haleema Bebe. Machine learning for real-time prediction of fuel consumption and classification of driving behavior using ecu data. *International Journal of Research Publication and Reviews*, 6(5):8633–8637, 2025. Accessed 2025-06-02.

[69] E.B. Gueguim Kana, Julius Oloke, Agbaje Lateef, and M.O. Adesiyan. Modeling and optimization of biogas production on saw dust and other co-substrates using artificial neural network and genetic algorithm. *Renewable Energy*, 46:276–281, 2012. doi: 10.1016/j.renene.2012.03.027. URL https://doi.org/10.1016/j.renene.2012.03.027. Accessed 2025-8-11.

[70] M. I. Karamangil and M. Tekin. Comparison of fuel consumption and recoverable energy according to nedc and wltp cycles of a vehicle. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 37(2):123–134, 2022. URL http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0122-53832022000200031. Accessed 2025-08-14.

[71] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017. Accessed 2025-05-21.

[72] Grzegorz Koszałka, Andrzej Szczotka, and Andrzej Suchecki. Comparison of fuel consumption and exhaust emissions in wltp and nedc procedures. *Combustion Engines*, 179 (4):186–191, 2019. doi: 10.19206/CE-2019-431. Accessed 2025-08-14.

[73] M. Kuhn and K. Johnson. *Feature engineering and selection: A practical approach for predictive models*. CRC Press, 2019. Accessed 2025-06-01.

[74] S. Kumar, R. Sharma, and R. Paul. Comparative study of cloud data warehouse solutions in modern enterprises. *Journal of Cloud Computing*, 11(3):201–210, 2023. Accessed 2025-06-17.

[75] Jérôme L. Importância da aerodinâmica na redução do consumo de combustível. https://www.automotores-rev.com/pt/importancia-da-aerodinamica-na-reducao-do-consumo-de-combustivel/, 2024. Accessed 2024-07-27.

[76] Jakub Lasocki. The wltc vs nedc: A case study on the impacts of driving cycle on engine performance and fuel consumption. *International Journal of Automotive and Mechanical Engineering*, 18:9071–9081, 09 2021. doi: 10.15282/ijame.18.3.2021.19.0696. Accessed 2025-05-09.

[77] Hyeonjik Lee and Kihyung Lee. Comparative evaluation of the effect of vehicle parameters on fuel consumption under nedc and wltp. *Energies*, 13(18):4788, 2020. doi: 10.3390/en13184788. Accessed 2025-05-12.

[78] Lori Lemazurier, Neeraj Shidore, Namdoo Kim, and Ayman Moawad. Impact of advanced engine and powertrain technologies on engine operation and fuel consumption for future vehicles. *SAE International Journal of Engines*, 8(3):1557–1570, 2015. Accessed 2025-06-23.

[79] Y. Li, W. Xu, C. Liao, K. K. Hoi, L. Nie, N. Wang, J. Wu, Y. Hu, and D. Zeng. High-precision light trucks fuel consumption prediction using xgboost-ipoa-deepesn. In *Advances in Computer Science and Ubiquitous Computing*. Atlantis Press, 2024. doi: 10.2991/978-94-6463-514-0_83. Accessed 2025-05-28.

[80] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2 (3):18–22, 2002. Accessed 2025-05-25.

[81] Yang Lui. The role of technology in improving fuel economy in automobiles. *Advances in Automobile Engineering*, 13:319, 2024. doi: 10.35248/2167-7670.24.13.319. Accessed 2025-07-19.

[82] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017. Accessed 2025-05-23.

[83] S. Madden, C. Li, and M. Stonebraker. The changing landscape of big data analytics: Cloud, parallel, and shared-nothing databases. *Communications of the ACM*, 66(8):78–87, 2023. Accessed 2025-06-19.

[84] Microsoft. What is azure synapse analytics? https://docs.microsoft.com/en-us/azure/synapse-analytics/, 2024. Accessed 2025-06-21.

[85] Mile21. Mile21 official site, 2024. URL https://www.mile21.eu/choose-your-country. Accessed 2024-12-15.

[86] Mitsubishi Electric RCE. Real-life vs. standard driving cycles and implications on ev power consumption, 2020. URL https://www.mitsubishielectric-rce.eu/wp-content/uploads/2020/06/Degrenne_IECON16.pdf. Accessed 2025-05-12.

[87] Malte Mittendorf, Ulrik D. Nielsen, and Harry B. Bingham. The prediction of sea state parameters by deep learning techniques using ship motion data. In *Proceedings of the 7th World Maritime Technology Conference (WMTC'22)*, Copenhagen, Denmark, 2022. URL https://backend.orbit.dtu.dk/ws/portalfiles/portal/274844961/The_Prediction_of_Sea_State_Parameters_by_Deep_Learning_Techniques_using_Ship_Motion_Data.pdf. Accessed 2025-10-11.

[88] Saeed Mohsen, Ahmed Elkaseer, and Steffen G. Scholz. Industry 4.0-oriented deep learning models for human activity recognition. *Sensors*, 21(22):7485, 2021. doi: 10.3390/s21227485. URL https://doi.org/10.3390/s21227485. Accessed 2025-10-18.

[89] Christoph Molnar. 14 lime. In *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Christoph Molnar, 2025. URL https://christophm.github.io/interpretable-ml-book/lime.html. Accessed: 2025-07-05.

[90] Caterina Beatrice Monti, Federico Ambrogi, and Francesco Sardanelli. Sample size calculation for data reliability and diagnostic performance: a go-to review. *European Radiology*, 2024. doi: 10.1007/s00330-024-10971-w. Accessed 2025-06-22.

[91] L. Morawska, G. R. Johnson, and Z. D. Ristovski. Auxiliary load impacts on vehicle fuel consumption. *Environmental Science & Technology*, 55(14):9876–9885, 2021. doi: 10.1021/acs.est.1c01551. Accessed 2025-05-20.

[92] Widmen 1° Auto Center Multimarcas. Como os pneus afetam o consumo de combustível. https://widmen.com.br/como-os-pneus-afetam-o-consumo-de-combustivel/, 2024. Accessed 2024-07-29.

[93] Bruno Baeta Nanni. Estudo da minimização do consumo de combustível em comboio de veículos pesados por meio da otimização dos perfis de velocidade. Master's thesis, Centro Universitário FEI, 2020. URL https://repositorio.fei.edu.br/items/f3c69ae1-56c9-4542-b441-41baf80119a8. Accessed 2025-08-14.

[94] Natural Resources Canada. Autosmart - learn the facts: Weight affects fuel consumption, 2016. URL https://natural-resources.canada.ca/sites/www.nrcan.gc.ca/files/oee/pdf/transportation/fuel-efficient-technologies/autosmart_factsheet_16_e.pdf. Accessed 2025-05-20.

[95] Optuna Development Team. Optuna: Hyperparameter optimization framework. https://optuna.org/, 2025. Accessed 2025-07-25.

[96] Dave Osborn. Don't automatically settle for a 30 piece capability study, 2022. URL https://blog.minitab.com/en/30-piece-capability-study-automotive. Accessed 2025-06-23.

[97] Oxylabs Blog. Scrapy vs selenium: Which one is better? https://oxylabs.io/blog/scrapy-vs-selenium, 2024. Accessed 2025-06-13.

[98] J. Pavlovic, A. Marotta, and B. Ciuffo. Co_2 emissions and energy demands of vehicles tested under the nedc and the new wltp type approval test procedures. *Applied Energy*, 205: 927–935, 2017. doi: 10.1016/j.apenergy.2017.08.115. Accessed 2025-05-21.

[99] Jelica Pavlovic, Georgios Fontaras, Markos Ktistakis, K. Anagnostopoulos, Dimitrios Komnos, Biagio Ciuffo, Michael Clairotte, and Victor Valverde. Understanding the origins and variability of the fuel consumption gap: lessons learned from laboratory tests and a real-driving campaign. *Environmental Sciences Europe*, 32, 04 2020. doi: 10.1186/s12302-020-00338-1. Accessed 2025-08-14.

[100] R. Perez and M. Clemens. Seamless data science pipelines using cloud data warehouses: A streamlit and snowflake case study. *Data Science Journal*, 20:1–11, 2021. Accessed 2025-06-19.

[101] R. Perez and M. Clemens. Seamless data science pipelines using cloud data warehouses: A streamlit and snowflake case study. *Data Science Journal*, 20:1–11, 2021. Accessed 2025-06-17.

[102] Tiago Pinto. Previsão de consumo de combustível na indústria automóvel: Um estudo de caso. Mestrado em ciência de dados, Universidade Portucalense, 07 2024. Accessed 2024-10-09.

[103] Plotly Technologies. Dash user guide. https://dash.plotly.com/, 2022. Accessed 2025-06-22.

[104] Proxyway. Scrapy vs beautiful soup vs selenium – which one to use? https://proxyway.com/guides/scrapy-vs-beautiful-soup-vs-selenium, 2025. Accessed 2025-06-13.

[105] Python in Plain English. End-to-end testing in django with selenium. https://python.plainenglish.io/, 2024. Accessed 2025-06-07.

[106] PyTorch Documentation. Data loading and processing tutorial (beginner). https://docs.pytorch.org/tutorials/beginner/basics/data_tutorial.html, 2025. Accessed 2025-07-23.

[107] Real Python. Python's requests library (guide). https://realpython.com/python-requests/, 2025. Accessed 2025-06-16.

[108] So rin Yoo, Jae woo Shin, and Seoung-Ho Choi. Machine learning vehicle fuel efficiency prediction. *Scientific Reports*, 15:8691, 2025. doi: 10.1038/s41598-025-96999-0. Accessed 2025-05-07.

[109] Mbelle Bisong Samuel, Paune Felix, Youmene Nongosso Miguel, Tambere Samam A. Cyrille, and Pierre Kisito Talla. Study and simulation of the fuel consumption of an internal combustion engine. *International Journal of Engineering and Technical Management Research (IJETMR)*, 8(1):41–54, 2020. Accessed 2025-05-31.

[110] scikit-learn developers. 1.17. neural network models (supervised) — scikit-learn 1.7.1 documentation, 2025. URL https://scikit-learn.org/stable/modules/neural_networks_supervised.html. Accessed 2025-05-08.

[111] Scott Lundberg. Shap: Shapley additive explanations, 2025. URL https://shap.readthedocs.io/en/latest/. Accessed: 2025-04-12.

[112] ScrapeOps. Web scraping with scrapy: The complete guide in 2025. https://scrapfly.io/blog/posts/web-scraping-with-scrapy, 2025. Accessed 2025-06-14.

[113] ScraperAPI. Scraper api documentation. https://www.scraperapi.com/docs/, 2025. Accessed 2025-06-17.

[114] ScraperAPI. 14 best web scraping tools in 2025 (pros, cons, pricing). https://www.scraperapi.com/web-scraping/tools/, 2025. Accessed 2025-06-14.

[115] Scrapy Documentation. Scrapy at a glance. https://docs.scrapy.org/en/latest/intro/overview.html, 2025. Accessed 2025-06-16.

[116] Scrapy Documentation Team. Scrapy architecture overview. https://docs.scrapy.org/en/0.24/topics/architecture.html, 2024. Accessed 2024-11-22.

[117] Selenium Project. Selenium webdriver. https://www.selenium.dev, 2023. Accessed 2025-06-07.

[118] Semanticscholar.org. Explainability in machine learning models for transportation, 2024. Accessed 2025-06-03.

[119] Anika Seufert, Florian Wamser, Stefan Wunderer, Andrew Hall, and Tobias Hoßfeld. Trust but verify: Crowdsourced mobile network measurements and statistical validity measures. In *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, June 2021. Accessed 2025-06-22.

[120] G. M. Hasan Shahariar, Timothy A. Bodisco, Nicholas Surawski, Md Mostafizur Rahman Komol, Mojibul Sajjad, Thuy Chu-Van, Zoran Ristovski, and Richard J. Brown. Real-driving co2, nox and fuel consumption estimation using machine learning models. *Journal of Environmental Management*, 345:116302, 2023. doi: 10.1016/j.jenvman.2023.116302. URL https://www.sciencedirect.com/science/article/pii/S2949821X23000595. Accessed: 2025-04-12.

[121] Snowflake Inc. Snowflake architecture and features. https://www.snowflake.com/, 2024. Accessed 2025-06-19.

[122] S. Sonkar, S. Danwani, Y. K. Sahu, and B. Sahu. Fuel efficiency prediction using machine learning. *International Journal of Research Publication and Reviews*, 6(5):3428–3432, 2025. Accessed 2025-05-14.

[123] Ultimate Specs. Ultimate specs, 2025. URL https://www.ultimatespecs.com. Accessed 2024-10-05.

[124] SpritMonitor. Spritmonitor – fahrzeug- und verbrauchsportal. https://spritmonitor.de/, 2025. Accessed 2025-10-02.

[125] SpritMonitor. Gasoline consumption: Mercedes-benz - gla-klasse - spritmonitor.de, 09 2025. URL https://www.spritmonitor.de/en/overview/28-Mercedes-Benz/1368-GLA-Klasse.html?fueltype=2&constyear_s=2023&power_s=221&power_e=225&powerunit=2. Accessed 2025-09-29.

[126] SpritMonitor. Gasoline consumption: Ford - focus - spritmonitor.de, 09 2025. URL https://www.spritmonitor.de/en/overview/17-Ford/148-Focus.html?fueltype=2&constyear_s=2019&constyear_e=2021&power_s=200&exactmodel=ST&powerunit=2. Accessed 2025-09-30.

[127] Starburst. Vendor lock-in: Tco and cloud data warehouses. https://www.starburst.io/blog/vendor-lock-in/, 2023. Accessed 2025-06-21.

[128] StatusNeo. Snowflake vs. data warehouse: Clash of innovation. https://statusneo.com/snowflake-vs-data-warehouse-clash-of-innovation/, 2025. Accessed 2025-06-19.

[129] K. Stewart. Consumer attitudes and priorities in vehicle fuel economy data. *Energy Policy*, 154:112285, 2021. Accessed 2025-05-31.

[130] Streamlit Inc. Streamlit documentation and deployment best practices. https://docs.streamlit.io/, 2023. Accessed 2025-06-21.

[131] Planton – Soluções Sustentáveis. Poluição automotiva: O impacto da idade dos veículos. https://www.planton.eco.br/poluicao-automotiva-o-impacto-da-idade-dos-veiculos/, 2025. Accessed 2025-07-19.

[132] Testomat Blog. Playwright vs selenium: The evolution of dominance—can selenium make a comeback? https://testomat.io/blog/, 2024. Accessed 2025-06-09.

[133] TestQuality. Playwright vs selenium: A 2025 test automation guide. https://testquality.com/playwright-vs-selenium-ultimate-guide-test-automation/, 2025. Accessed 2025-06-07.

[134] testRigor Blog. Why selenium sucks for end-to-end testing in 2025. https://testrigor.com/blog/why-selenium-sucks-for-end-to-end-testing/, 2025. Accessed 2025-06-09.

[135] Testsigma Blog. Pros and cons of selenium as an automation testing tool. https://testsigma.com/blog/selenium-automation-testing-pros-cons/, 2019. Accessed 2025-06-09.

[136] John Thomas, Jeffrey Gonder, Eric Wood, and William Sparks. Fuel consumption sensitivity of conventional and hybrid electric light-duty gasoline vehicles to driving style. Technical report, Oak Ridge National Laboratory, U.S. Department of Energy, 2017. URL https://afdc.energy.gov/files/u/publication/fuel_consumption_sensitivity_style.pdf. Accessed 2025-08-14.

[137] U. Tietge, P. Mock, V. Franco, and J. German. From laboratory to road: A comparison of official and "real-world" fuel consumption and co2 emission values for cars in europe. Technical report, International Council on Clean Transportation, 2017. Accessed 2025-05-29.

[138] Manjunath TK and Ashok Kumar PS. Fuel prediction model for driving patterns using machine learning techniques. *Journal of Computer Science*, 20(3):291–302, 2024. doi: 10.3844/jcssp.2024.291.302. Accessed 2025-05-17.

[139] Chien-Ming Tseng and Chi-Kin Chau. Personalized prediction of driving energy consumption based on participatory sensing. *arXiv preprint arXiv:1610.00171*, 2016. URL https://arxiv.org/abs/1610.00171. Version 2, latest update 2017-02-20.

[140] US Department of Energy. Fuel economy technologies and trends, 2023. Accessed 2025-05-07.

[141] U.S. Environmental Protection Agency. The 2020 epa automotive trends report: Greenhouse gas emissions, fuel economy, and technology since 1975, January 2021. URL https://www.epa.gov/sites/default/files/2021-01/documents/420r21003.pdf. Accessed 2025-06-11.

[142] Jaime Valencia, Mario Camargo, Edgar Mariño, Javier Jiménez, and Daniel Olarte. Eco-driving key factors that influence fuel consumption in heavy-truck fleets: A colombian case. *Transportation Research Part D: Transport and Environment*, 56:258–270, 2017. doi: 10.1016/j.trd.2017.08.012. Accessed 2025-05-07.

[143] Flore Vallet, Mostepha Khouadjia, Ahmed Amrani, and Juliette Pouzet. Designing a data visualisation and analysis tool for supporting decision-making with public transportation network. *Procedia Computer Science*, 181:999–1008, 2021. Accessed 2025-06-03.

[144] Marco Varalla. Machine learning techniques for the estimation of soil moisture from satellite data. Master's thesis in space engineering, Politecnico di Milano, 2023. URL https://www.politesi.polimi.it/retrieve/d6b53e72-dfb2-4e27-98ed-1e51ea9e69ca/Thesis.pdf. Accessed 2025-08-03.

[145] Vinoth QA Academy. Advantages and disadvantages of selenium webdriver. https://vinothqaacademy.com/docs/advantages-and-disadvantages-of-selenium-webdriver/, 2025. Accessed 2024-10-24.

[146] Visure Solutions. Machine learning in the automotive industry. https://visuresolutions.com/automotive/machine-learning/, 2025. Accessed 2025-06-01.

[147] G. Wang. Predictability of vehicle fuel consumption using lstm. *ASCE Journal*, 2023. Accessed 2025-05-23.

[148] Yicheng Wang. Price prediction of ford cars applying multiple machine learning methods. In *Proceedings of the 1st International Conference on E-commerce and Artificial Intelligence (ECAI 2024)*, page 280–285, 2024. doi: 10.5220/0013214800004568. URL https://www.scitepress.org/Papers/2024/132148/132148.pdf. Accessed 2025-08-06.

[149] Zeyi Wen, Jiashuai Shi, Bingsheng He, Jian Chen, Kotagiri Ramamohanarao, and Qinbin Li. Exploiting gpus for efficient gradient boosting decision tree training. *IEEE Transactions on Parallel and Distributed Systems*, 2019. Accessed 2025-05-22.

[150] Zhiwei Yang, Zuduo Zheng, Jiwon Kim, and Hesham A. Rakha. Eco-driving strategies using reinforcement learning for mixed traffic in the vicinity of signalized intersections. *Transportation Research Part C: Emerging Technologies*, 153:103031, 2024. doi: 10.1016/j.trc.2024.103031. Accessed 2025-05-18.

[151] Ying Yao, Xiaohua Zhao, Chang Liu, and Jian Rong. Vehicle fuel consumption prediction method based on driving behavior data collected from smartphones. *International Journal of Distributed Sensor Networks*, 16:1550147720959732, 2020. doi: 10.1177/1550147720959732. Accessed 2025-05-29.

[152] S. Zahid. Data-driven machine learning techniques for fuel economy prediction in sustainable transportation systems. *Sustainable Energy Technologies and Assessments*, page 101658, 2025. doi: 10.1016/j.seta.2025.101658. Accessed 2025-06-02.

[153] Haoyu Zhu, Lian Xie, Xiaozhen Zhang, Han Wang, Hongyang Chen, Hong Kun Xu, and Tao Xu. Influence of driving style on traffic flow fuel consumption and emissions based on the field data. *Physica A: Statistical Mechanics and its Applications*, 600:127494, 2022. doi: 10.1016/j.physa.2022.127494. Accessed 2025-05-07.

[154] M. Zimakowska-Laskowska and O. Orynycz. Integrating experimental data and neural computation for fuel consumption prediction. *Advances in Science and Technology Research Journal*, 19(9):452–468, 2025. doi: 10.12913/22998624/208172. Accessed 2025-05-26.

[155] Zyte. Zyte api documentation. https://www.zyte.com/api/, 2025. Accessed 2025-06-07.

[156] Zyte Blog. The future of scrapy: Smarter, faster and ready for ai-powered scraping. https://www.zyte.com/blog/the-future-of-scrapy/, 2024. Accessed 2025-06-14.

[157] Şevket Ay, Ekin Ekinci, and Zeynep Garip. A comparative analysis of meta-heuristic optimization algorithms for feature selection on ml-based classification of heart-related diseases. *The Journal of Supercomputing*, 79:11797–11826, 2023. doi: 10.1007/s11227-023-05132-3. URL https://link.springer.com/article/10.1007/s11227-023-05132-3. Accessed 2025-08-14.