DRIVING TOWARDS SUSTAINABILITY: PREDICTIVE ANALYTICS FOR VEHICLES' CARBON DIOXIDE EMISSIONS

FARHAH SYAHMINA

JANUARY 2025

DRIVING TOWARDS SUSTAINABILITY: PREDICTIVE ANALYTICS FOR VEHICLES' CARBON DIOXIDE EMISSIONS

FARHAH SYAHMINA

A Project Report Submitted to the
College of Computing and Informatics
Universiti Tenaga Nasional
in Partial Fulfilment of the Requirements for the
Bachelor of Information Technology (Information Systems) (Hons.)

JANUARY 2025

DECLARATION

I hereby declare that the final year project is my original work except for quotations and citations have been duly acknowledged. I also declare that has not been previously and is not concurrently submitted for any degree program at Universiti Tenaga Nasional or at any other institutions. This final year project may be made available within the university library and may be borrowed, consulted, copied, or reproduced in accordance with the provision of the UNITEN Library Regulations from time to time made by the Library Committee.

FARHAH SYAHMINA

26/01/2025

APPROVAL PAGE

This thesis entitled:

"DRIVING TOWARDS SUSTAINABILITY: PREDICITVE ANALYTICS FOR VEHICLE'S CARBON DIOXIDE EMISSIONS"

submitted by:

FARHAH SYAHMINA

In requirement for the degree of **Bachelor of Information Technology (Information Systems) (Hons.)** College of Computing and Informatics, Universiti Tenaga Nasional has been accepted.

Supervisor:
Signature: …….
Date: 26/01/2025

ACKNOWLEDGMENT

I would like to acknowledge the outstanding individuals who have kindled the flame of inspiration and encouraged me throughout the entire project. I owe an abundance of gratitude to my beloved parents, Sulfeeza and Mohd Faizal whose continuous support and encouragement have served as the foundation of my achievements. Your invincible faith in me has been a source of resilience and motivation throughout. I am grateful to my respected lecturers, especially my supervisor, for the priceless guidance and scholarly wisdom which illuminated my journey in completing this project. To my dear friends, which I am eternally grateful, whose constant morale has brought me comfort and joy during the long path of research and writing. And to the special someone who has been by my side through every twist and turn, providing unshaken support, love and encouragement, you are my rock and inspiration. Your faith in me has been the driving force behind this project. All of you have played significant roles in this journey, and I am deeply grateful for your constant encouragement. Thank you.

EXECUTIVE SUMMARY

Carbon dioxide (CO2) is a natural gas in our atmosphere, produced by humans through respiration and other living organisms. This project seeks to understand the rising emissions of CO2 from vehicles, a critical concern for global climate change. An investigation regarding the relationships between variables such as vehicle class, engine size, fuel type and driving conditions will be done to determine their impact on the emissions. By employing strategic research and analysis techniques, the project aims to identify key factors contributing to CO2 emissions and develop a predictive analytics system of CO2 emissions. The project will use the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology, a proven and organized approach to planning data mining projects. CRISP-DM provides a structured framework for each phase of the data analytics lifecycle, explaining the tasks involved and their interrelationships. This method ensures all the processes are done systematically and analysis is made using appropriate analytics models to identify key emission factors. Several benefits are expected to be gained such as a predictive analytics system for car manufacturers to build a less CO2 emitting vehicle and it can also assist manufacturers in improving engine designs, increasing fuel efficiency, and meeting demanding environmental regulations more effectively. The project will produce a model that accurately predicts CO2 emissions based on the variables. The performance of this model will be evaluated to ensure its reliability and effectiveness. The insights generated will be conveyed through a detailed dashboard. The dashboard will be crucial for car manufacturers to make informed decisions on the factors considered manufacturing or producing a new vehicle. The outcomes will assist automotive manufacturers in advancing sustainable transportation solutions, focusing on reducing vehicle CO2 emissions and contributing to efforts to tackle the broader challenges of global climate change.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AdaBoost	Adaptive Boosting Regressor
CatBoost	Categorical Boosting
CO2	Carbon dioxide

CHAPTER 1
INTRODUCTION

1.1 Project Background

Carbon dioxide (CO2) is a gas in Earth's atmosphere, made from one carbon atom and two oxygen atoms (The Editors of Encyclopaedia Britannica, 2024). It is important for the carbon cycle, which regulates our climate. However, burning fossil fuels in vehicles releases CO2 which contributes to climate change. CO2 traps heat in the atmosphere, worsening the greenhouse effect and leading to global warming. This causes problems like changing weather, rising sea levels, disrupted ecosystems, agricultural issues and health risks. To reduce vehicle CO2 emissions, cleaner technologies are needed like electric cars, hybrids and fuel-efficient engines and to use more public transport and renewable energy sources (Vehicle Emissions | Green Vehicle Guide, n.d.).

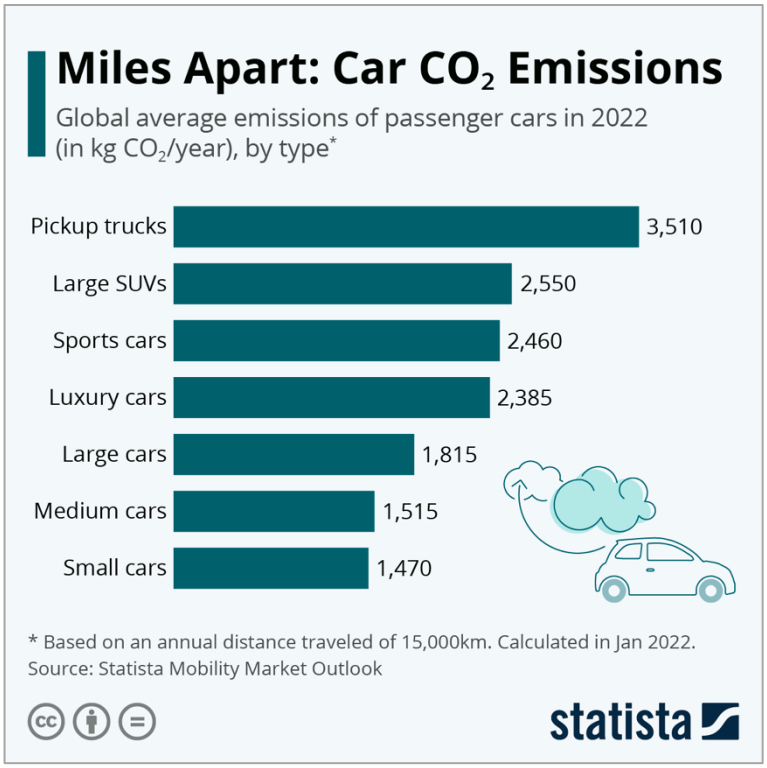

Figure 1.1: Average Emissions of Passenger Cars in 2022

Based on the figure above, the bar chart analyses the global average CO2 emissions of passenger vehicles according to vehicle type. Pickup trucks, a common mode of personal transportation in many areas, appear as the most emissions-intensive category, generating an astounding 3,510 kg of CO2 per year on average. Large SUVs and sports cars are also among the top emitters, with annual emissions of 2,550 kg and 2,460 kg, respectively. In comparison, smaller vehicles, such as medium and compact cars, release significantly less CO2, at 1,515 kg and 1,470 kg per year, respectively. This figure confirms the importance of personal vehicle choices in determining an individual's carbon footprint, emphasizing the necessity for customers to choose more environmentally friendly options.

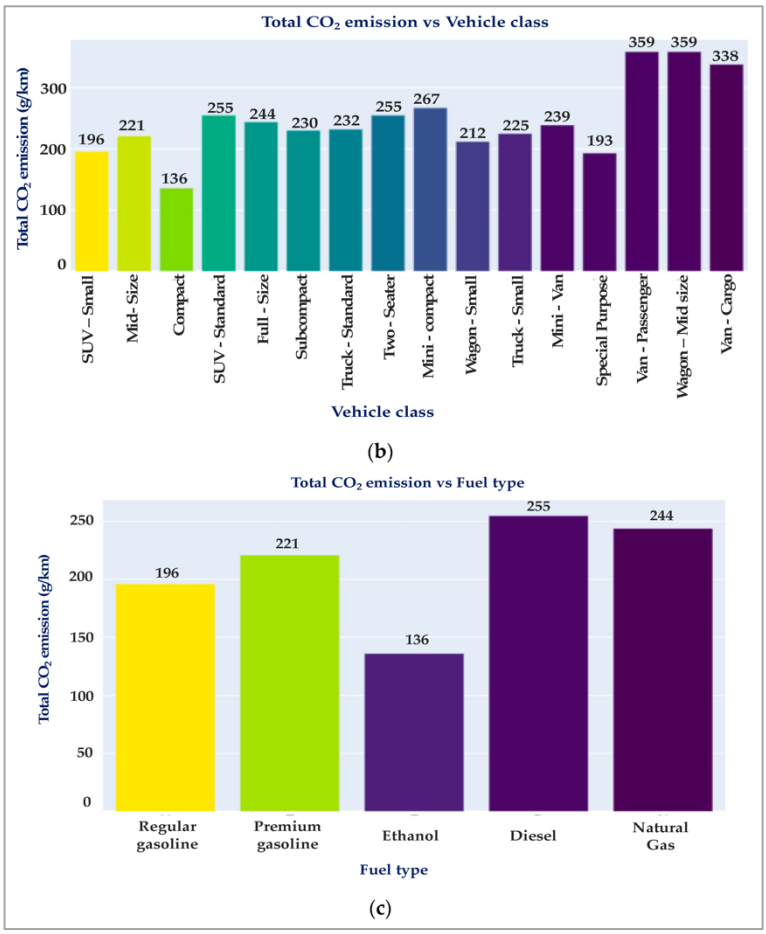

Figure 1.2: Total CO2 Emissions vs Vehicle Class/Fuel Type

Figure 1.2 indicates the environmental impact of vehicles by showing the increased CO2 emissions connected with larger vehicle classes and fuel types frequently used in passenger vehicles. The bar graph shows that SUVs, vans and larger cargo vans produce significantly more CO2 than smaller passenger cars. Furthermore, the second graph demonstrates that diesel and natural gas vehicles, which are common in the personal vehicle market, emit more pollutants than gasoline-powered vehicles. This information emphasizes the significance of vehicle size and fuel type when assessing the environmental effect of personal transportation options.

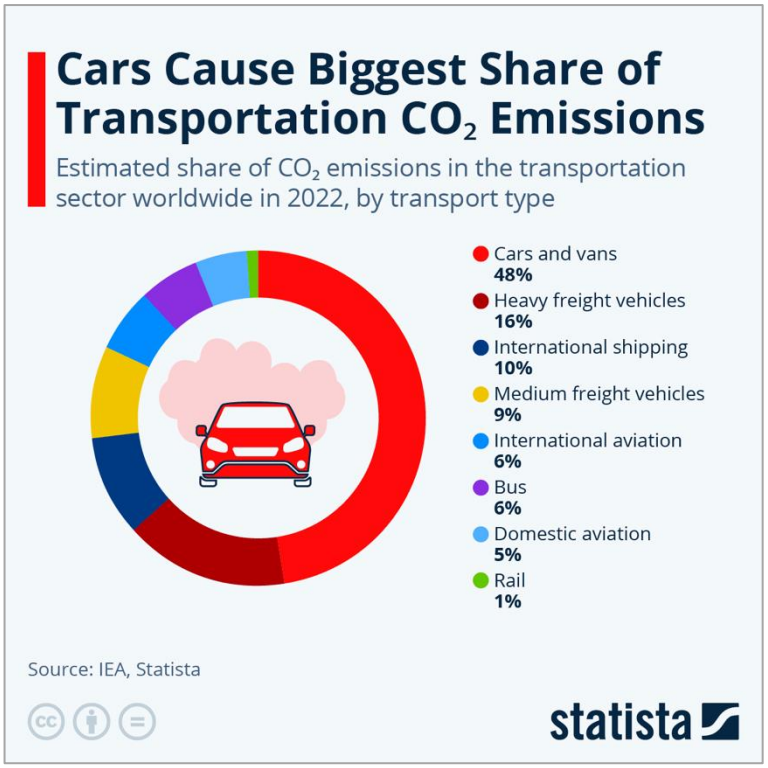

Figure 1.3: Cars Cause Biggest Share of Transportation CO2 Emissions

The figure above highlights cars and vans' significant contribution to global transportation-related CO2 emissions. Personal vehicles contribute 48% of total emissions in this sector. This remarkable percentage exceeds other types of transportation such as heavy freight vehicles (16%), international shipping (10%) and medium freight vehicles (9%). While international and domestic air travel produces 15% of emissions and buses for 6%, personal vehicles and vans have the largest

contribution. This emphasizes the need for action to reduce the carbon footprint of personal vehicles by implementing more sustainable transportation alternatives as well as developing cleaner vehicle technologies.

1.2 Problem Statement

The acceleration of global climate change has emphasized vehicle carbon dioxide (CO2) emissions as a crucial environmental challenge, seeking detailed knowledge of how to reduce their impact. Despite efforts to migrate to cleaner energy sources, the transportation sector remains a major CO2 emitter, which requires extensive analysis to discover emission trends and advise targeted responses. This includes investigating the complex relationship of variables such as vehicle class, engine size, fuel type and driving conditions to determine their proportional contributions and connections. To produce environmentally friendly transportation solutions, car manufacturers must make informed decisions that include an extensive knowledge of emission trends as well as successful mitigation measures. Collaboration among key stakeholders, including government agencies, research institutions and manufacturers, is critical for promoting innovation, sharing knowledge and adopting advanced emission-reducing technologies. By utilizing these data, manufacturers may design vehicles that meet environmental requirements while also meeting the growing demand for sustainability.

1.3 Objective

1. To identify the suitable supervised learning algorithm to predict vehicles' CO2 emissions.
2. To develop an analytics model using machine learning techniques to predict vehicles' CO2 emissions.
3. To develop dashboard to display the patterns and trends of vehicles' CO2 emissions.

1.4 Scope

The scope of this project involves utilizing an open dataset titled "Fuel Consumption Ratings", downloaded as a CSV file from the Government of Canada Open Data website. The project employs supervised machine learning methods, specifically linear regression, to predict CO2 emissions across different vehicle classes. By analyzing key features within the dataset, the project aims to identify the factors that influence CO2 emissions for each class of vehicles. Technologies such as Python, Jupyter Notebook and relevant libraries will be used to perform data analysis and model development. The outcomes of this project are intended for automotive manufacturers to support informed decisions in producing vehicles with reduced CO2 emissions and promoting sustainable practices within the automotive industry.

Table 1.1:Data Sources

Dataset Name	Source Link
2015-2019 Fuel Consumption Ratings	Government of Canada Open Data
2020 Fuel Consumption Ratings	
2021 Fuel Consumption Ratings	
2022 Fuel Consumption Ratings	
2023 Fuel Consumption Ratings	
2024 Fuel Consumption Ratings	

Table 1.2: Tools & Technologies

Tool/Technology	Tool/Technology Usage
Excel	Modelling & Dashboard Development
Python Libraries	Modelling
Jupyter Notebook	Modelling
Power BI	Dashboard Development

1.5 Project Timeline

Table 1.3: Project Timeline for Project 1

Milestone description	Week														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Proposal & Project Understanding															
Propose Project Title															
Define Project Objectives															
Milestone 1: Proposal	◆														
Data Understanding															
Identify datasets															
Understand data															
Describe data															
Explore the data															
Literature Review															
Search Related Works															
Study the Related Works															
Compare the Related Works															
Milestone 2A: Oral Presentation				◆											
Data Preparation															
Select data															
Clean data															
Integrate data															
Format data															
Milestone 2B: Draft Report								◆							
Modeling															
Select modeling techniques															
Design test model															
Build the model															
Assess the model															
Milestone 3A: Report												◆			
Evaluation															
Evaluate results															
Review the process															
Determine the next steps															
Milestone 3B: Dashboard Demo Showcase															◆

Table 1.4: Project Timeline for Project 2

Milestone description	Week																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Milestone 1A: System Demo						◆												
Modeling																		
Select modeling techniques																		
Design test model																		
Build the model																		
Assess the model																		
Dashboard																		
Sketch the overall dashboard design																		
Milestone 1B: System Demo									◆									
Dashboard																		
Produce descriptive visualizations																		
Design the descriptive visualizations page																		
Produce predictive visualizations																		
Design the predictive visualizations page																		
Milestone 2A: System Demo & System Testing												◆						
Milestone 2B: Submission of Report, Logbook, System, Demo Video														◆				
Panel Presentation																		◆

CHAPTER 2

PRELIMINARY STUDY

2.1 Introduction to the Domain

The focus of this project is the environmental impact of carbon dioxide emitted by vehicles during fuel combustion. This is a crucial area of study because transport is one of the biggest contributors to global greenhouse gas emissions, with road vehicles accounting for 72% of the global emissions (Wang & Ge, 2019). Understanding and regulating these emissions is critical to tackling climate change and encouraging sustainable development. This domain's primary features are fuel type, transmission type, engine size and vehicle class, all of which influence a vehicle's CO2 emissions. Other relevant features are fuel consumption measures, such as liters per 100 kilometers (L/100 km) and CO2 emissions values, which measure the amount of CO2 emitted per unit of fuel burned.

This domain's core theories are based on principles of energy conversion and thermodynamics. Internal combustion engines switch chemical energy from fuel to mechanical energy, producing CO2 as a byproduct. The amount of CO2 emitted is determined by the fuel's carbon content and engine performance. Improving vehicle design, fuel composition and alternative engine designs, such as electric or hybrid-electric vehicles, represent significant initiatives for lowering emissions.

The recent landscape of vehicle CO2 emissions is influenced by global initiatives to lower greenhouse gas emissions through enhanced regulations and technological innovation. Governments and organizations have established emission objectives, such as the European Union's 95gCO2/km limit for passenger vehicles (*CO2 emissions*

performance of New Passenger Cars in Europe 2024). Standard methods involve operating vehicles through standardized driving cycles to evaluate emissions and using telematics data for real-time emission monitoring. On-Board Diagnostics (OBD) and simulation models are commonly used for emissions analysis, nevertheless techniques based on machine learning such as CatBoost and Linear Regression are gaining significance for forecasting and regulating CO2 emissions.

Despite progress, significant reductions in CO2 emissions remain difficult to achieve. These include limited insights into how certain vehicle features contribute to emissions, as well as a lack of accessible, actionable data that manufacturers may use to create low-emission vehicles. This project intends to close this gap by creating a dashboard that compiles CO2 emissions data and predictions. The dashboard will enable manufacturers to analyze trends, identify high-emission features and make informed decisions during vehicle design, resulting in the manufacturing of vehicles with a lesser amount CO2 emissions.

2.2 Review of Related Work

Table 2.1: Literature Review

Literature	Aim	Data Source	Machine Learning Model/Techniques	Findings
Artificial Intelligence-based CO2 Emission Predictive Analysis System (Yeasmin, S., Syed, S. N. J., Shmais, L. A., & Dubayyan, R. A. (2020))	To propose a CO2 emission predictive analysis system using Artificial Intelligence (AI) that calculates and predicts the amounts of CO2 to be emitted from a cargo van in advance.	- data.gov.uk	- Multivariable Linear Regression	1. The researchers developed an AI system that uses Multivariable Linear Regression to analyse and predict emissions. 2. The model achieved high accuracy, with an R-squared score close to 1, meaning its predictions were very reliable. 3. Distance travelled in kilometres had the greatest impact on CO2 emissions predictions. 4. The researchers used 80% of their dataset for training the model and 20% for testing, which helped improve accuracy.
The driving factors and mitigation strategy of CO2 emissions from China's passenger vehicle sector towards carbon neutrality (Gao, Z., Zhang, Q., Liu, B., Liu, J., Wang, G., Ni, R., & Yang, K. (2024))	To understand the driving factors behind the changes in CO2 emissions from China's passenger vehicle sector and propose emission reduction strategies towards carbon neutrality.	- China Association of Automobile Manufacturer - China Automotive Energy Consumption Query Platform - National Bureau of Statistics	- Vehicle Fleet-based Fuel Cycle Carbon Emissions Accounting Model - Logarithmic Mean Divisia Index (LMDI) - Scenario-based Prediction Method	1. CO2 emissions from passenger vehicles in China increased from 228.4 Mt in 2012 to 459.9 Mt in 2022. 2. Gasoline vehicles account for about 95% of total emissions, though this proportion is decreasing. 3. Growth in vehicle ownership levels was the second largest factor driving emissions up. 4. Population growth had a smaller but still positive effect on emissions.
Deep Learning Model Based CO2 Emissions Prediction Using Vehicle Telematics Sensors Data	To propose a scalable vehicle CO2 emission prediction model using vehicle On-Board	- Vehicle On-Board Diagnostics (OBD-II) port data	- Recurrent Neural Network (RNN) - Long Short-Term Memory (LSTM)	1. Deep learning models are effective in handling time-series data and nonlinear relationships in CO2 emissions. 2. Incorporating real-time telematics data improves model reliability.

(Singh, M., & Dubey, R. (2023))	Diagnostics (OBD-II) port data.			3. LSTM outperforms simpler regression models in accuracy for time-series predictions.
Predicting CO2 Emissions from Traffic Vehicles for Sustainable and Smart Environment Using a Deep Learning Model (Al-Nefaie, A. H., & Aldhyani, T. H. H. (2023))	To model and predict carbon dioxide (CO2) emissions from vehicles using advanced artificial intelligence techniques, specifically deep learning models such as Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM).	- Kaggle.com	- Long Short-Term Memory (LSTM) - Bidirectional LSTM (BiLSTM)	1. They developed and compared two types of deep learning models, Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) 2. The BiLSTM model showed superior performance, achieved higher prediction accuracy, reached a determination coefficient (R2) of 93.78% and performed better on MSE and RMSE metrics 3. The researchers used correlation analysis (R%) to identify which vehicle characteristics had the strongest relationship with CO2 emissions.
Forecasting Carbon Dioxide Emissions of Light-Duty Vehicles with Different Machine Learning Algorithms (Natarajan, Y., Wadhwa, G., Preethaa, K. R. S., & Paul, A. (2023))	To develop predictive models for carbon dioxide (CO2) emissions from light-duty vehicles, leveraging a large sensor-based dataset collected by the government of Canada.	- A large sensor-based dataset collected by the government of Canada	- Categorical boosting (Catboost) model	1. The researchers developed a new approach using boosting and regression models to predict vehicle CO2 emissions. 2. Their main innovation was using a Categorical Boosting (Catboost) model 3. The data came from sensor-based measurements collected by the Canadian government, focusing on light-duty vehicles. 4. The model provides practical insights for vehicle users making purchase decisions and manufacturers developing new vehicles

Machine learning-based Time Series Models for Effective CO2 Emission prediction in India (Singh, S. K., & Kumari, S. (2022, January))	To predict India's CO2 emissions for the next ten years (2020-2030) using various time series models, including statistical models (ARIMA, SARIMAX, Holt-Winter), machine learning models (Linear Regression, Random Forest) and a deep learning model (LSTM).	- CAIT	- Auto-Regressive Integrated Moving Average (ARIMA) - Seasonal Autoregressive Integrated Moving Average with an Exogenous Variable (SARIMAX) - Holt-Winter's Model - Linear Regression - Random Forest Regressor - Long Short-Term Memory (LSTM)	1. LSTM, SARIMAX and Holt-Winter models performed better than other models for CO2 emission forecasting. 2. Among all models, LSTM showed the best performance across nine different performance metrics. 3. Random Forest and ARIMA models performed poorly for this type of time series forecasting. 4. Linear Regression showed similar patterns to actual emissions but wasn't reliable due to poor performance metric values. 5. The comparative forecasting showed that LSTM could provide the most accurate predictions for future CO2 emissions in India.

2.3 Review of Machine Learning Methods

2.3.1 Multivariable Linear Regression

Multivariable Linear Regression predicts a dependent variable by modelling a linear relationship with multiple independent variables (Yeasmin et al., 2020). It is easy to implement and interpret, making it a good starting point for predictive analysis when the relationship between variables is linear. However, it struggles with non-linear relationships, outliers and multicollinearity, which occurs when two or more predictor variables in a regression model are highly correlated with each other, this can compromise accuracy in complex datasets. The study by Yeasmin et al. (2020) proposes a system that uses multivariate linear regression on fleet vehicle data to predict CO2 emissions, to optimize delivery operations for sustainability.

2.3.2 Vehicle Fleet-based Fuel Cycle Carbon Emissions Accounting Model

This method evaluates carbon emissions throughout a vehicle fleet's lifecycle, from production to disposal. It provides a comprehensive assessment, useful for identifying areas to reduce emissions and informing policy decisions. However, it requires extensive data collection and is time-consuming to implement, making it challenging for large-scale or incomplete datasets.

2.3.3 Logarithmic Mean Divisia Index (LMDI) Method

The LMDI method breaks down carbon emissions into contributing factors , such as fuel efficiency and vehicle usage, to analyze their impact over time. It is particularly effective for understanding trends and providing detailed insights into emission drivers. However, it is not predictive and cannot handle non-linear relationships or future forecasting. Gao et al. (2024) employs an integrated analytical framework using the

Logarithmic Mean Divisia Index (LMDI) method and scenario-based predictions to analyze historical trends and drivers of CO2 emissions from passenger vehicles in China and evaluate emission reduction strategies.

2.3.4 Scenario-based Prediction Method

This method forecasts future emissions by modelling different scenarios based on varying assumptions, such as policy changes or advancements in technology. It is flexible, allowing "what-if" analysis. However, it is highly dependent on the accuracy of the assumptions, which can make it less reliable for long-term predictions. Developing realistic scenarios can also be time intensive.

2.3.5 Recurrent Neural Network (RNN)

RNNs are deep learning models designed for sequential data, such as time-series emissions data, as they retain information from previous inputs to improve predictions. They are effective for capturing short-term dependencies, but they can struggle with long-term memory due to the vanishing gradient problem and require large datasets and computational resources.

2.3.6 Long Short-Term Memory (LSTM)

LSTM, a variant of RNN, uses memory cells to overcome the vanishing gradient problem, making it capable of learning long-term dependencies in sequential data. It is highly effective for time-series forecasting but it requires significant computational power and is more complex to fine-tune and interpret compared to simpler models. Singh & Dubey (2023) utilizes a Long Short-Term Memory (LSTM) neural network model to predict CO2 emissions from individual vehicles in real-time using On-Board Diagnostics (OBD-II) port data.

2.3.7 Bidirectional LSTM (BiLSTM)

BiLSTM improves upon LSTM by processing data in both forward and backward directions, capturing context from both past and future sequences. This makes it particularly effective for time-series tasks where future context enhances predictions. However, it is even more computationally demanding than LSTM and can be overfit with small datasets. The use of LSTM and Bidirectional LSTM (BiLSTM) deep learning models to predict CO2 emissions from vehicles based on various vehicle parameters is proposed in the study done by Al-Nefaie & Aldhyani (2023), aiming to support policymakers in developing effective environmental policies.

2.3.8 Categorical Boosting (CatBoost) Model

CatBoost is a gradient boosting algorithm optimized for datasets with categorical features, such as vehicle type or fuel type, without extensive preprocessing. It offers high accuracy and efficient training but requires careful parameter tuning and more computational resources than simpler models like regression. It is less prone to overfitting, making it suitable for complex data. In the study made by Natarajan et al. (2023), boosting and regression models are developed, particularly CatBoost, to predict CO2 emissions from light-duty vehicles using a large sensor-based dataset collected by the government of Canada, supporting efforts to mitigate transportation-related air pollution.

2.3.9 Comparison of Machine Learning Methods

Table 2.2: Comparison of Machine Learning Methods

Machine Learning Method	Description	Strengths	Weaknesses
Multivariable Linear Regression	This method models the relationship between multiple independent variables (features) and a dependent variable by fitting a linear equation.	• Easy to implement and interpret. • Works well for linear relationships. • Requires minimal computational power.	• Assumes a linear relationship, which may not hold in real-world data. • Sensitive to outliers.
Vehicle Fleet-based Fuel Cycle Carbon Emissions Accounting Model	This method evaluates carbon emissions across the entire lifecycle of a vehicle fleet, including production, fuel consumption and disposal.	• Comprehensive, considering all emission sources. • Useful for policymaking and lifecycle assessments.	• Requires extensive data collection. • Complex and time-consuming to implement.
Logarithmic Mean Divisia Index method (LMDI)	A decomposition method that analyses the contribution of different variables (fuel efficiency, vehicle type) to differences in carbon emissions over time.	• Offers detailed insights into specific contributors to emissions. • Handles data with varying units.	• Not predictive; focuses on past trends. • Limited in dealing with nonlinear relationships.
Scenario-based Prediction Method	This method forecasts future emissions by creating multiple scenarios based on different assumptions such as changes in policy, technology or behaviour.	• Flexible and allows for "what-if" analysis. • Incorporates expert knowledge.	• Results depend heavily on assumptions, which may be inaccurate. • Time-intensive to create realistic scenarios.
Recurrent Neural Network (RNN)	A deep learning model designed to handle sequential data by using feedback loops to remember previous inputs, such as time-series data for vehicle emissions.	• Handles sequential and time-dependent data well. • Captures patterns in historical data.	• Prone to vanishing gradients, leading to difficulty learning long-term dependencies. • Requires large datasets and computational resources.
Long Short-term Memory network (LSTM)	An advanced type of RNN that solves the vanishing gradient problem by using memory cells to retain important information over long sequences.	• Excels at learning long-term dependencies in sequential data. • Effective for time-series forecasting.	• Computationally expensive. • Complex to tune and interpret compared to simpler models.
Bidirectional LSTM (BiLSTM)	An extension of LSTM that processes sequential data in both forward and backward directions, providing more context for predictions.	• Captures both past and future dependencies in data. • More accurate for certain time-series tasks.	• Even more computationally demanding than LSTM. • Overfitting risk with small datasets.
Categorical boosting (Catboost) model	A gradient boosting algorithm optimized for categorical data, making it ideal for datasets with non-numeric features like vehicle type or fuel type.	• Handles categorical data without extensive preprocessing. • High accuracy and efficient training. • Less prone to overfitting compared to other boosting methods.	• Requires parameter tuning for optimal performance. • Computationally intensive compared to simpler models like linear regression.

2.3.10 Selection of Ideal Method

The multivariable linear regression method was chosen not only for its efficiency and scalability, but also because it is simple to use and does not require advanced technical skills to implement. Its simplicity makes it extremely accessible, allowing it to quickly develop models without requiring complicated programming or machine learning knowledge. Unlike complex deep learning models, linear regression is simple to analyze variable correlations and understand how predictions are made. Furthermore, it does not require tuning several hyperparameters or creating advanced structures, decreasing the complexity of the modelling process. These qualities make multivariable linear regression an ideal choice for the system I would like to develop which requires a machine learning model that is simple and requires low knowledge.

2.4 Methodology Used

There are many distinct types of methodologies used in information systems development, such as software-focused ones like agile software development (Highsmith & Cockburn, 2001), which prioritize flexibility, collaboration and input from clients. Other than that, waterfall methodology is a systematic and sequential process in which every stage needs to be finished before proceeding to the next. They seem to be less common in data science and business analytics. Every method has advantages, disadvantages, and works well in various project situations. Waterfall offers a clear, linear path while Agile offers flexibility and adaptation and CRISP-DM's structured approach is perfect for data mining projects. The requirements and limitations of the project determine the methodology to be used.

This project will be using CRISP-DM, which stands for Cross Industry Standard Process for Data Mining, as the system development method. CRISP-DM is a reliable and well-proven approach that provides an organized method to plan a data mining project. It is a data analytics lifecycle which explains the relationships between the tasks involved in each phase, as well as descriptions of the common phases of a project. Using CRISP-DM, there are six phases to be applied and done, which are:

2.4.1 Business Understanding

In this phase, the goal is to understand the project's objectives and requirements, and the background of this project. Carbon dioxide (CO2) is a component of Earth's atmosphere, created by the bonding of one carbon atom with two oxygen atoms. It is an essential component of the carbon cycle, which naturally regulates our planet's climate. However, in the domain of transportation, CO2 takes the spotlight because of burning fossil fuels such as gasoline and diesel in internal combustion engines.

The concern over car CO2 emissions arises from its critical role in increasing climate change and resulting global warming. CO2 acts as a greenhouse gas, absorbing and re-emitting infrared radiation, thus trapping heat within the Earth's atmosphere. While this is necessary for maintaining a stable climate, however, due to human activities particularly the widespread use of fossil fuels in transportation, have significantly increased CO2 levels, worsening the effect. Furthermore, initiatives aimed at reducing dependency on fossil fuels, increasing public transportation and encouraging the use of renewable energy sources are critical in tackling the threats posed by CO2 emissions to our climate and atmosphere.

Given that this project is about the study of carbon dioxide emissions produced by vehicles, three objectives have been established which are to identify the suitable supervised learning algorithm to predict vehicles' CO2 emissions, to develop an analytics model, and to develop a dashboard. Furthermore, it is needed to know the hardware and software requirements to complete this task. This project requires the development of a dashboard, the hardware required is a laptop and the software required is the programs or applications that will be employed to build the system. Information will be gathered regarding other studies on CO2 emissions from vehicles by reviewing journals from

Google Scholar, ScienceDirect and others. After completing all of this, the following phase will involve understanding the data that has been obtained.

2.4.2 Data Understanding

For the data understanding phase, the retrieved dataset must be evaluated and analyzed. The data set on carbon dioxide emissions by vehicle is acquired from Government of Canada Open Data website as a csv file, also each attribute and value are identified. To accurately identify significant variables, the data set must have variables related to issues. Aside from that, the quality of the attributes and values must be assessed, particularly the identification of missing values and the spelling of attributes with values. For this project, the dataset chosen to be analyzed is CO2 emissions by vehicles in Canada. This dataset provides a comprehensive repository of critical information about a wide range of vehicle features. It encompasses an extensive variety of vehicle classes, including sedans, SUVs, and pickup trucks, with brands like Dodge, BMW and Audi and models like the Challenger, 320i Sedan and A3. Engine sizes can range from 1.5L to 3.5L, while cylinder counts vary from 2 to 12. It also includes significant information about fuel types, fuel consumption in liters per 100 kilometers (*L*/100km) and CO2 emissions in grams per kilometer. Such thorough data is essential in determining and analyzing the environmental impact of various vehicles. By utilizing this project, car manufacturers can identify patterns and develop vehicles which better fulfil the regulations to reduce CO2 emissions. Finally, this knowledge serves as the basis for informed decision-making in the vehicle sector, promoting a more sustainable and environmentally friendly future.

2.4.3 Data Preparation

The data preparation phase consists of several tasks intended to modify the dataset obtained from Government of Canada Open Data website into the final dataset that satisfies the objectives and requirements. Cleaning the dataset involves removing any missing values, which reduces the number of rows and allows for more accurate analysis. Next, the data will be properly formatted by rearranging attributes, reordering records, and renaming attributes and values. This phase will be explained further in Chapter 3.

2.4.4 Modelling

This is the phase in which the most suitable modelling techniques to use in this project will be discovered. There are two forms of machine learning: supervised and unsupervised learning. These two machine learning methods encompass a variety of techniques, including classification, regression, association rules and clustering. Supervised learning will be the focus for this project. Some of these techniques and the one selected will be discussed in Chapter 3. A test design, which outlines the training, testing and evaluation of the models using appropriate methods, will be created. The selected techniques will then be applied to the input data set, and the parameter settings used to create the model will be recorded. Finally, the results analyzed are based on evaluation criteria. Detailed explanation of this phase will be done in Chapter 4.

2.4.5 Evaluation

The process will be evaluated by reviewing all the steps involved in building the model to ensure that it can meet the project objectives that I made. The goal is to determine whether an essential project issue has not been adequately addressed. From here, a solution to the problems can be figured out.

CHAPTER 3

DATA COLLECTION AND PREPARATION

3.1 Data Sources

The carbon dioxide emissions by vehicle datasets were obtained by downloading it as a CSV file called “Fuel consumption ratings” from Government of Canada Open Data website. This dataset contains a wide range of features that are essential for analyzing the emissions produced by vehicles.

Vehicle types such as 4WD (four-wheel drive), AWD (all-wheel drive), FFV (flexible-fuel vehicle), SWB (short wheelbase), LWB (long wheelbase) and EWB (extended wheelbase) are stored in the “Model” field. This classification assists in understanding how various car designs influence CO2 emissions.

There is also a feature named “Class” where various vehicle classes like Compact, Mid-size, Full-size, Pickup truck and Minivan are stored. Utilizing this feature can furthermore understand the relationship between vehicle class and CO2 emissions as well as can be used for predictive modelling.

The “Transmission” field encompasses a range of transmission types, including continuously variable (AV), manual (M), automated manual (AM), automated with select shift (AS) and automatic (A). Moreover, 3-10 are the number of gears. This is essential to understanding how diverse transmission types affect the emissions produced.

The dataset includes a field called “Fuel Type” which provides information on the type of fuel used such as natural gas (N), diesel (D), ethanol (E85), regular gasoline (X) and premium gasoline (Z). To assess each fuel type's environmental impact and the efficiency of alternative fuels in lowering CO2 emissions, it is crucial to understand the several types of fuel.

As for fuel consumption, there are three fields on this, “Fuel Consumption City”, “Fuel Consumption Highway” and “Fuel Consumption Combined” which provide consumption ratings for city and highway driving in liters per 100 kilometers (L/100 km), as well as a combined rating expressed as L/100 km.

As for the “CO2 Emissions” feature, this provides the tailpipe emissions of carbon dioxide, measured in grams per kilometer (g/km), for combined city and highway driving. This data is critical for identifying trends and patterns in emissions.

The last feature “CO2 Rating” provides the vehicle’s tailpipe emissions of carbon dioxide are rated on a scale from 1 (worst) to 10 (best). As shown in the figure below is the data dictionary of the dataset.

Table 3.1: Data Dictionary

Model	**AWD** = All-wheel drive (vehicle designed to operate with all wheels powered)
	4WD/4X4 = Four-wheel drive (vehicle designed to operate with either two wheels or four wheels powered)
	FFV = Flexible-fuel vehicle (vehicle designed to operate on gasoline and ethanol blends of up to 85% ethanol (E85))
	CNG = Compressed natural gas
	NGV = Natural gas vehicle
	SWB = Short wheelbase
	LWB = Long wheelbase
	EWB = Extended wheelbase
Class	**Two-seater**
	Minicompact
	Subcompact
	Compact
	Mid-size
	Full-size
	Station wagon: Small
	Station wagon: Mid-size
	Pickup truck: Small
	Pickup truck: Standard
	Sport utility vehicle: Small
	Sport utility vehicle: Standard
	Minivan
	Van: Cargo
	Van: Passenger
	Special purpose vehicle
Engine size	Total displacement of all cylinders (in litres [*L*])
Cylinders	Number of engine cylinders
Transmission	**A** = Automatic
	AM = Automated manual
	AS = Automatic with select shift)
	AV = Continuously variable
	M = Manual
	1-10 = Number of gears/speeds
Fuel type	**X** = Regular gasoline
	Z = Premium gasoline
	D = Diesel
	E = E85
	N = Natural Gas
Fuel consumption	Fuel consumption ratings are shown in litres per 100 kilometres (L/100 km) **City rating** – represents urban driving in stop-and-go traffic **Highway rating** – represents a mix of open highway and rural road driving, typical of longer trips **Combined rating** – reflects 55% city driving and 45% highway driving
CO2 emissions	The vehicle's tailpipe emissions of carbon dioxide (CO2) are shown in grams per kilometre for combined city and highway driving. For PHEVs, CO2 emissions values reflect a mix of electric mode and gasoline-only operation.
CO2 rating	The vehicle's tailpipe emissions of carbon dioxide are rated on a scale from 1 (worst) to 10 (best).

3.2 Data Preparation

A crucial step in data preparation is the process of transforming raw data to clean and understandable formatted data which makes the dataset suitable and can be used easily for analysis. The process ensures that the dataset is free of errors, inconsistencies, and redundant data, which may significantly improve the quality of insights produced by it.

Data Preparation

Combining all datasets into one CSV file

```
import os
import pandas as pd

# Directory containing all the CSV files
csv_directory = r"Datasets"

# Output CSV file
output_file = "2019-2024_CO2Emissions.csv"

# Initialize an empty list to store DataFrames
dataframes = []
```

```
# Loop through all CSV files in the directory
for file_name in os.listdir(csv_directory):
    if file_name.endswith(".csv"):
        file_path = os.path.join(csv_directory, file_name)
        try:
            # Try reading the file with utf-8 encoding
            try:
                df = pd.read_csv(file_path, encoding='utf-8')
            except UnicodeDecodeError:
                # Fallback to latin-1 encoding
                df = pd.read_csv(file_path, encoding='latin-1')

            # Ensure the columns match the structure
            expected_columns = [
                "Model year", "Make", "Model", "Vehicle class", "Engine size (L)", "Cylinders", "Transmission",
                "Fuel type", "City (L/100 km)", "Highway (L/100 km)", "Combined (L/100 km)", "Combined (mpg)",
                "CO2 emissions (g/km)", "CO2 rating", "Smog rating"
            ]

            if set(expected_columns).issubset(df.columns):
                # Keep only the relevant columns
                df = df[expected_columns]
                dataframes.append(df)
            else:
                print(f"Warning: {file_name} does not have the expected columns. Skipping.")
        except Exception as e:
            print(f"Error reading {file_name}: {e}")

# Combine all DataFrames into one
if dataframes:
    combined_df = pd.concat(dataframes, ignore_index=True)

    # Save the combined DataFrame to a CSV file
    combined_df.to_csv(output_file, index=False)
    print(f"Combined CSV file saved to {output_file}")
else:
    print("No valid CSV files found to combine.")
```

```
Combined CSV file saved to 2019-2024_CO2Emissions.csv
```

Figure 3.1: Combining Multiple Datasets into one Dataset

Before starting to do any data cleaning steps, for easier analysis merging several CSV datasets into one CSV file was done. The code above reads each CSV file from a folder that contains the datasets and determines whether it contains the necessary

columns. To provide compatibility with various formats, files are read using UTF-8 encoding, with a fallback to Latin-1 if necessary. Files containing mistakes or missing necessary columns are skipped and warnings are shown. Then, the code saves the data into a CSV file named “2019-2024_CO2Emissions.csv” after combining all the other datasets into a single dataset. The code notifies if no valid files are identified. Working with various datasets is made easier with this data preparation step, which also ensures consistency throughout all data.

Load Dataset

```
df = pd.read_csv("2019-2024_CO2Emissions.csv")

print("Original Dataset: \n")
df.head()
```

```
Original Dataset:
```

	Model year	Make	Model	Vehicle class	Engine size (L)	Cylinders	Transmission	Fuel type	City (L/100 km)	Highway (L/100 km)	Combined (L/100 km)	Combined (mpg)	CO2 emissions (g/km)	CO2 rating	Smog rating
0	2015	Acura	ILX	Compact	2.0	4	AS5	Z	9.7	6.7	8.3	34	191	NaN	NaN
1	2015	Acura	ILX	Compact	2.4	4	M6	Z	10.8	7.4	9.3	30	214	NaN	NaN
2	2015	Acura	ILX Hybrid	Compact	1.5	4	AV7	Z	6.0	6.1	6.1	46	140	NaN	NaN
3	2015	Acura	MDX SH-AWD	Sport utility vehicle: Small	3.5	6	AS6	Z	12.7	9.1	11.1	25	255	NaN	NaN
4	2015	Acura	RDX AWD	Sport utility vehicle: Small	3.5	6	AS6	Z	12.1	8.7	10.6	27	244	NaN	NaN

Figure 3.2: Loading Dataset into Jupyter Notebook

After ensuring all the needed libraries are imported, the dataset is then loaded into a pandas DataFrame for easy manipulation and analysis. This process reads the CSV file named “2019-2024_CO2Emissions.csv” and displays the first few rows to provide an initial glimpse of the dataset.

Dataset Information

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9960 entries, 0 to 9959
Data columns (total 15 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Model year            9960 non-null   int64
 1   Make                  9960 non-null   object
 2   Model                 9960 non-null   object
 3   Vehicle class         9960 non-null   object
 4   Engine size (L)       9960 non-null   float64
 5   Cylinders             9960 non-null   int64
 6   Transmission          9960 non-null   object
 7   Fuel type             9960 non-null   object
 8   City (L/100 km)       9960 non-null   float64
 9   Highway (L/100 km)    9960 non-null   float64
 10  Combined (L/100 km)   9960 non-null   float64
 11  Combined (mpg)        9960 non-null   int64
 12  CO2 emissions (g/km)  9960 non-null   int64
 13  CO2 rating            8832 non-null   float64
 14  Smog rating           7726 non-null   float64
dtypes: float64(6), int64(4), object(5)
memory usage: 1.1+ MB
```

Figure 3.3: Display of Dataset Information

Information about the dataset is displayed to understand its structure, including the number of entries, column names, data types and any non-null counts. The output gives a comprehensive overview of the dataset, helping to identify potential issues such as missing values or incorrect data types. Once the dataset information is understood, proceed to the next step of detecting missing values and null values.

Missing Values Detection

```
missing_values_flag = any(('?' in df[col].values) or ('' in df[col].values) for col in df.columns)

print("Missing values:", missing_values_flag)
```

```
Missing values: False
```

Null Values Detection

```
null_values_flag = df.isnull().values.any()

print("Actual null values:", null_values_flag)
```

```
Actual null values: True
```

Figure 3.4: Missing and Null Values Detection Process

For “Missing Values Detection” code, it iterates over each column in the DataFrame and checks if it contains any missing values represented by “?” or blank cells. The any() function returns “True” if any column has missing values and “False” if it does

not. Afterwards, the result of whether missing values represented by "?" or blank cells are found in the DataFrame will be printed.

As for "Null Values Detection" code, it checks if there are any actual null values (NaN) in the DataFrame using the isnull() method. Then, it checks if any of the resulting values are "True" using the any() method. The result of whether there are any actual null values in the DataFrame will be printed. Since the results showed "True" above for the "Null Values Detection", therefore the data has null values. However, there are no missing values as displayed above.

Columns Removal & Rename

```
print("\nColumn names:")
print(df.columns)
```

```
Column names:
Index(['Model year', 'Make', 'Model', 'Vehicle class', 'Engine size (L)',
       'Cylinders', 'Transmission', 'Fuel type', 'City (L/100 km)',
       'Highway (L/100 km)', 'Combined (L/100 km)', 'Combined (mpg)',
       'CO2 emissions (g/km)', 'CO2 rating', 'Smog rating'],
      dtype='object')
```

```
delete_columns = [
    'Combined (mpg)', 'Smog rating'
]

delete_columns = [col for col in delete_columns if col in df.columns]

df.drop(columns=delete_columns, inplace=True)

print("\nDataset after deleting specified columns: \n")
df.head()
```

```
Dataset after deleting specified columns:
```

	Model year	Make	Model	Vehicle class	Engine size (L)	Cylinders	Transmission	Fuel type	City (L/100 km)	Highway (L/100 km)	Combined (L/100 km)	CO2 emissions (g/km)	CO2 rating
0	2015	Acura	ILX	Compact	2.0	4	AS5	Z	9.7	6.7	8.3	191	NaN
1	2015	Acura	ILX	Compact	2.4	4	M6	Z	10.8	7.4	9.3	214	NaN
2	2015	Acura	ILX Hybrid	Compact	1.5	4	AV7	Z	6.0	6.1	6.1	140	NaN
3	2015	Acura	MDX SH-AWD	Sport utility vehicle: Small	3.5	6	AS6	Z	12.7	9.1	11.1	255	NaN
4	2015	Acura	RDX AWD	Sport utility vehicle: Small	3.5	6	AS6	Z	12.1	8.7	10.6	244	NaN

Figure 3.5: Removing Columns in the Dataset

```
df.rename(columns = {
    'Model year':'Model_Year',
    'Make':'Vehicle_Make',
    'Model':'Vehicle_Model',
    'Vehicle class':'Vehicle_Class',
    'Engine size (L)':'Engine_Size',
    'Fuel type':'Fuel_Type',
    'City (L/100 km)':'Fuel_Consumption_City',
    'Highway (L/100 km)':'Fuel_Consumption_Highway',
    'Combined (L/100 km)':'Fuel_Consumption_Combined',
    'CO2 emissions (g/km)':'CO2_Emissions',
    'CO2 rating':'CO2_Rating',
}, inplace = True)

print("\nDataset after renaming columns: \n")
df.head()
```

```
Dataset after renaming columns:
```

	Model_Year	Vehicle_Make	Vehicle_Model	Vehicle_Class	Engine_Size	Cylinders	Transmission	Fuel_Type	Fuel_Consumption_City	Fuel_Consumption_Highway	Fuel_Cc
0	2015	Acura	ILX	Compact	2.0	4	AS5	Z	9.7	6.7	
1	2015	Acura	ILX	Compact	2.4	4	M6	Z	10.8	7.4	
2	2015	Acura	ILX Hybrid	Compact	1.5	4	AV7	Z	6.0	6.1	
3	2015	Acura	MDX SH-AWD	Sport utility vehicle: Small	3.5	6	AS6	Z	12.7	9.1	
4	2015	Acura	RDX AWD	Sport utility vehicle: Small	3.5	6	AS6	Z	12.1	8.7	

Figure 3.6: Renaming Columns in the Dataset

Before proceeding to the columns removal and rename process, it is advisable to check the column names in the DataFrame to ensure it matches the intended columns. The drop method is used to delete the specified columns from the DataFrame. The "delete_columns" variable is to list the columns names to be deleted, then the "columns" parameter takes the list and "inplace=True" ensures that the changes are made directly to the DataFrame. If a column name does not exist, an error will be raised. To do that, filter the list of columns to delete based on the actual columns in the DataFrame to avoid it. After the deletion process is completed, the new dataset is displayed.

Breaking down Transmission to Transmission_Type & Gears

```
# Function to determine transmission type
def get_transmission_type(code):
    if 'M' in code and not 'A' in code:
        return 'Manual'
    elif any(x in code for x in ['A', 'S', 'V']):
        return 'Automatic'
    return 'Unknown'

# Function to extract number of gears
def get_gears(code):
    digits = ''.join(filter(str.isdigit, code))
    return int(digits) if digits else 0  # Default to 0 if no digits found

# Apply functions to create new columns
df['Transmission_Type'] = df['Transmission'].apply(get_transmission_type)
df['Gears'] = df['Transmission'].apply(get_gears)

df.head()
```

	Model_Year	Vehicle_Make	Vehicle_Model	Vehicle_Class	Engine_Size	Cylinders	Transmission	Fuel_Type	Fuel_Consumption_City	Fuel_Consumption_Highway	Fuel_Cc
0	2015	Acura	ILX	Compact	2.0	4	AS5	Z	9.7	6.7	
1	2015	Acura	ILX	Compact	2.4	4	M6	Z	10.8	7.4	
2	2015	Acura	ILX Hybrid	Compact	1.5	4	AV7	Z	6.0	6.1	
3	2015	Acura	MDX SH-AWD	Sport utility vehicle: Small	3.5	6	AS6	Z	12.7	9.1	
4	2015	Acura	RDX AWD	Sport utility vehicle: Small	3.5	6	AS6	Z	12.1	8.7	

Figure 3.7: Breaking down "Transmission" column into "Transmission_Type" and "Gears" columns in the Dataset

For "Transmission" column, the dataset is modified by splitting the "Transmission" field into two new columns, "Transmission_Type" and "Gears". A function is defined to classify transmission types as "Manual", "Automatic" or "Unknown" based on the "Transmission" column. Another function takes the number of gears from the same column, returning 0 if no digits are detected. These functions are applied to the "Transmission" column, resulting in new columns for further analysis.

This transformation improves the dataset by providing more detailed information about each vehicle's transmission and gears. For example, rather than just having a designation like "AS6," we now know it is an automatic transmission with six gears. This step makes the dataset more interpretable and suitable for analysis.

Columns Rearrange

```
df = df[['Model_Year',
         'Vehicle_Make',
          'Vehicle_Model',
          'Vehicle_Class',
          'Engine_Size',
          'Cylinders',
          'Fuel_Type',
          'Fuel_Consumption_City',
          'Fuel_Consumption_Highway',
          'Fuel_Consumption_Combined',
          'Transmission',
          'Transmission_Type',
          'Gears',
          'CO2_Emissions',
          'CO2_Rating']]

print("\nDataset after rearranging columns: \n")
df.head()
```

```
Dataset after rearranging columns:
```

	Model_Year	Vehicle_Make	Vehicle_Model	Vehicle_Class	Engine_Size	Cylinders	Fuel_Type	Fuel_Consumption_City	Fuel_Consumption_Highway	Fuel_Consumption_Co
0	2015	Acura	ILX	Compact	2.0	4	Z	9.7	6.7	
1	2015	Acura	ILX	Compact	2.4	4	Z	10.8	7.4	
2	2015	Acura	ILX Hybrid	Compact	1.5	4	Z	6.0	6.1	
3	2015	Acura	MDX SH-AWD	Sport utility vehicle: Small	3.5	6	Z	12.7	9.1	
4	2015	Acura	RDX AWD	Sport utility vehicle: Small	3.5	6	Z	12.1	8.7	

Cleaned Dataset Saved

```
file_path = "Cleaned CO2 Emissions.csv"

df.to_csv(file_path, index=False)

print("Cleaned dataset saved successfully to:", file_path)
```

```
Cleaned dataset saved successfully to: Cleaned CO2 Emissions.csv
```

Figure 3.8: Rearranging Columns and Saving the Cleaned Dataset

Once all data preparation steps are completed, the columns are rearranged and then the cleaned dataset needs to be saved. To do that, create a new file named “Cleaned CO2 Emissions.csv” then set a file path. The df.to_csv() function saves the cleaned DataFrame to the provided file directory. The “index=False” parameter prevents row indices from being included in the CSV file. The print() statement prints a message stating that the cleaned dataset was successfully saved to the given file directory.

3.3 Data Exploration

Exploratory Data Analysis

Statistics of Continuous Variables

```
df.describe()
```

	Model_Year	Engine_Size	Cylinders	Fuel_Consumption_City	Fuel_Consumption_Highway	Fuel_Consumption_Combined	Gears	CO2_Emissions	CO2_Ratin
count	9960.000000	9960.000000	9960.000000	9960.000000	9960.000000	9960.000000	9960.000000	9960.000000	8832.00000
mean	2019.202209	3.143695	5.614558	12.468665	9.128926	10.965492	7.046084	253.383133	4.62567
std	2.819836	1.344787	1.881500	3.415055	2.183227	2.816342	1.976513	60.623033	1.58081
min	2015.000000	0.900000	3.000000	4.000000	3.900000	4.000000	0.000000	94.000000	1.00000
25%	2017.000000	2.000000	4.000000	10.100000	7.600000	9.000000	6.000000	210.000000	4.00000
50%	2019.000000	3.000000	6.000000	12.100000	8.800000	10.600000	7.000000	249.000000	5.00000
75%	2022.000000	3.700000	6.000000	14.500000	10.300000	12.600000	8.000000	292.000000	5.00000
max	2024.000000	8.400000	16.000000	30.700000	20.900000	26.100000	10.000000	608.000000	10.00000

Statistics of Continuous & Categorical Variables

```
df.describe(include='all')
```

	Model_Year	Vehicle_Make	Vehicle_Model	Vehicle_Class	Engine_Size	Cylinders	Fuel_Type	Fuel_Consumption_City	Fuel_Consumption_Highway	Fuel_Consum
count	9960.000000	9960	9960	9960	9960.000000	9960.000000	9960	9960.000000	9960.000000	
unique	NaN	42	2070	15	NaN	NaN	5	NaN	NaN	
top	NaN	Ford	Mustang	Sport utility vehicle: Small	NaN	NaN	X	NaN	NaN	
freq	NaN	862	48	1863	NaN	NaN	4751	NaN	NaN	
mean	2019.202209	NaN	NaN	NaN	3.143695	5.614558	NaN	12.468665	9.128926	
std	2.819836	NaN	NaN	NaN	1.344787	1.881500	NaN	3.415055	2.183227	
min	2015.000000	NaN	NaN	NaN	0.900000	3.000000	NaN	4.000000	3.900000	
25%	2017.000000	NaN	NaN	NaN	2.000000	4.000000	NaN	10.100000	7.600000	
50%	2019.000000	NaN	NaN	NaN	3.000000	6.000000	NaN	12.100000	8.800000	
75%	2022.000000	NaN	NaN	NaN	3.700000	6.000000	NaN	14.500000	10.300000	
max	2024.000000	NaN	NaN	NaN	8.400000	16.000000	NaN	30.700000	20.900000	

Figure 3.9: Computed Variables Statistics

By utilizing the “df.describe ()” function, which computes essential statistics for all continuous variables, the characteristics of the dataset can be thoroughly understood. This generates a summary that includes several key metrics:

- Count (count): The number of non-null entries in each column.
- Mean (mean): The average value of the data in each column.
- Standard Deviation (std): A measure of the amount of variation or dispersion in the dataset.
- Minimum Value (min): The smallest value in the column.

- Interquartile Range (IQR): This includes the 25th percentile (Q1), the median or 50th percentile (Q2), and the 75th percentile (Q3), providing a sense of the data distribution.
- Maximum Value (max): The largest value in the column.

These statistics provide a full understanding of the central tendency, dispersion and overall range of continuous variables, allowing quick understanding of the dataset's distribution and anomalies or outliers can be figured out.

Following that, the analysis becomes broader by executing the “df.describe ()” function with the “include='all'” parameter. This enables the computation of basic statistics for all continuous variables, as well as some fundamental statistics for categorical variables. In addition to the statistics provided above, this extended summary will include:

- Unique (unique): The number of unique values in each categorical column.
- Top (top): The most frequent value in each categorical column.
- Frequency (freq): The count of occurrences of the most frequent value.

This expanded summary of the dataset provides useful insights into the categorical data, revealing the variety of categories, the most common categories and the distribution of categorical variables. By combining the statistics of continuous and categorical variables, a more comprehensive picture of the dataset is acquired, which is critical for future analysis.

Visualizations are essential for effective data exploration and analysis because it allows quick identification of patterns, distributions and potential concerns within the data. Visualizations are required to help understanding the relationship between features. These visualizations assist in detecting outliers, abnormalities or unexpected trends that may necessitate more study or data cleansing.

This exploratory investigation yielded insights that directly influenced the dashboard's design and functionality. Understanding the value and distribution of each feature allows important data points to be determined, including which feature should be prominently presented and prioritized. Furthermore, the visualizations can help with the selection of appropriate charting methods, interactive and the general structure and organization of the dashboard.

Figure 3.10: Outliers Visualization

Figure above shows a code snippet is used to create and display box plots for each numeric feature in the DataFrame. Before starting the process of visualizing the data, the required libraries such as the "matplotlib" and the "seaborn" library need to be imported. The "pyplot' module is imported and aliased as plt. This allows the creation of static, animated and interactive Python visualizations. The "seaborn" library offers a high-level interface for creating visually appealing and useful statistical visualizations. Furthermore, the variable "num_feat" is made to store a list of column names that correspond to numeric features whereas the variable "cat_feat" is to store categorical features. However, this variable is not used in this visualization, but it will be used later.

A loop is initiated to iterate over each column name in the "num_feat" list. A new figure is produced for each box plot with a specified dimension of 10 inches wide by 5 inches tall. This ensures that each plot is adequate in size to be readable. Then, a box plot

is created by using the “seaborn” library, with further codes to specify the data source, set the x-axis to the numeric feature currently being processed in the loop and the usage of color palette for the box plots is specified. After that, the title of the box plot is set to indicate which feature is being visualized and the plot is finally displayed.

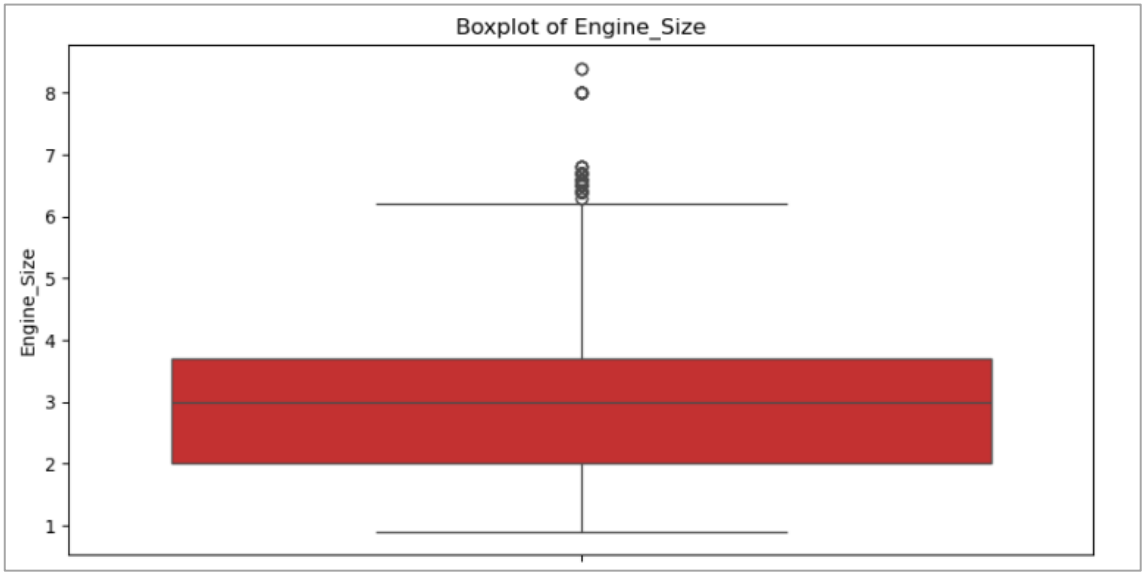

Figure 3.11: Boxplot of Engine_Size Outliers

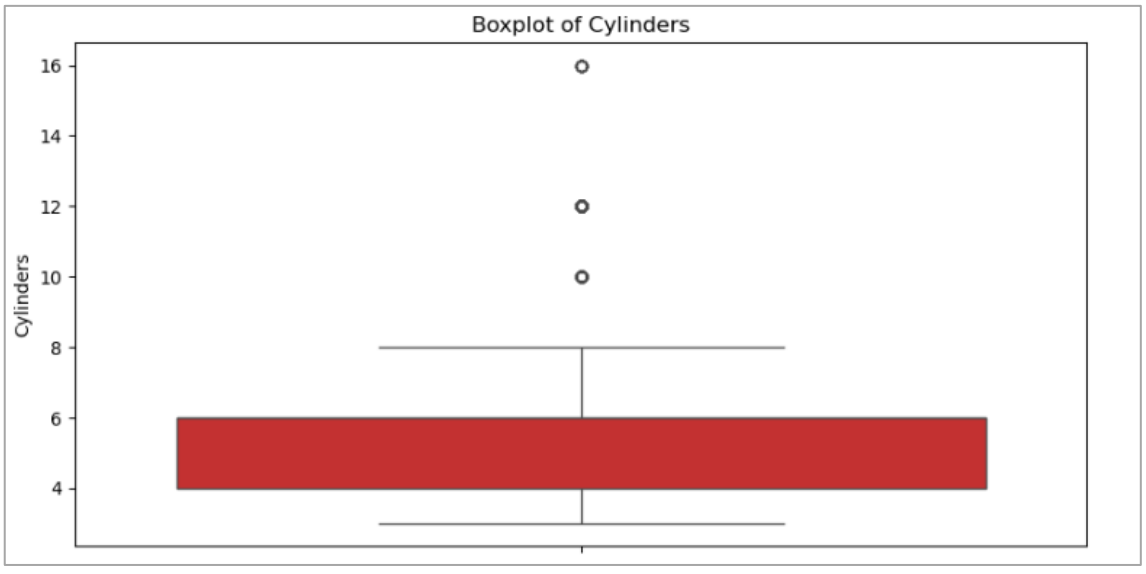

Figure 3.12: Boxplot of Cylinders Outliers

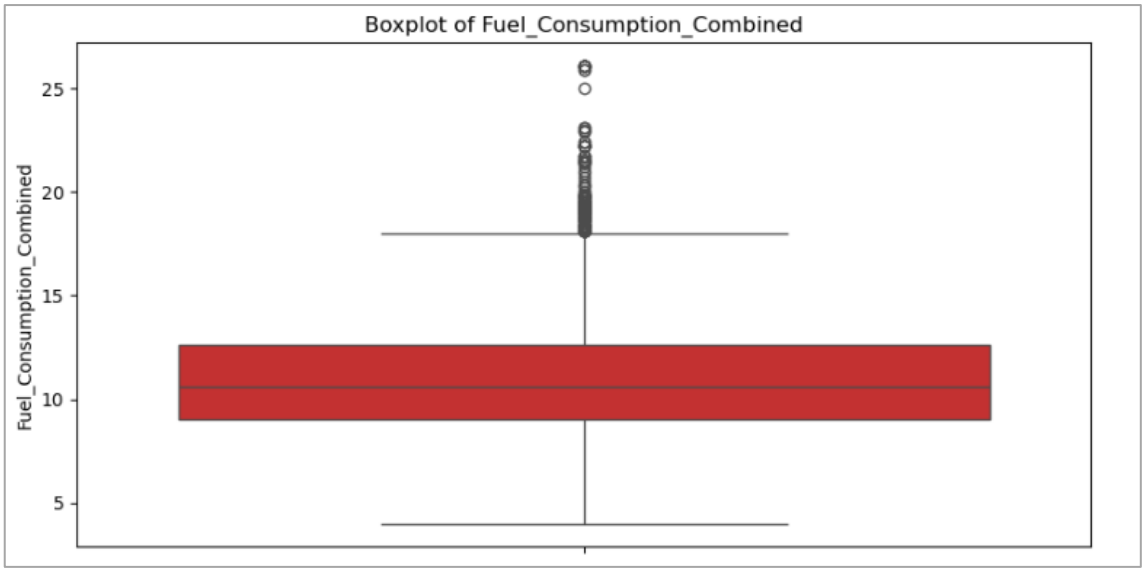

Figure 3.13: Boxplot of Fuel_Consumption_Combined Outliers

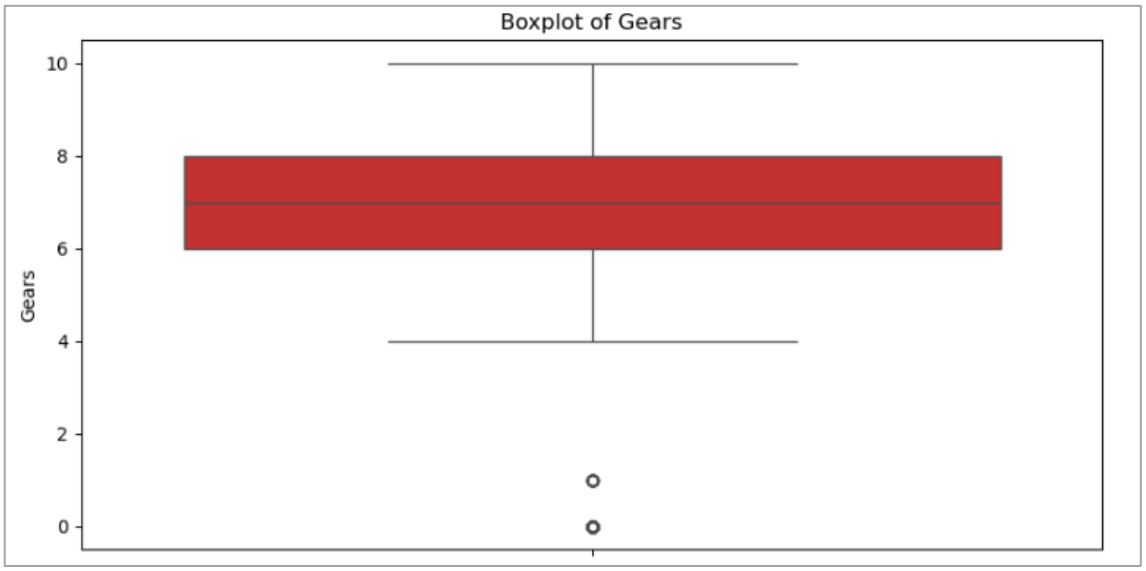

Figure 3.14: Boxplot of Gears Outliers

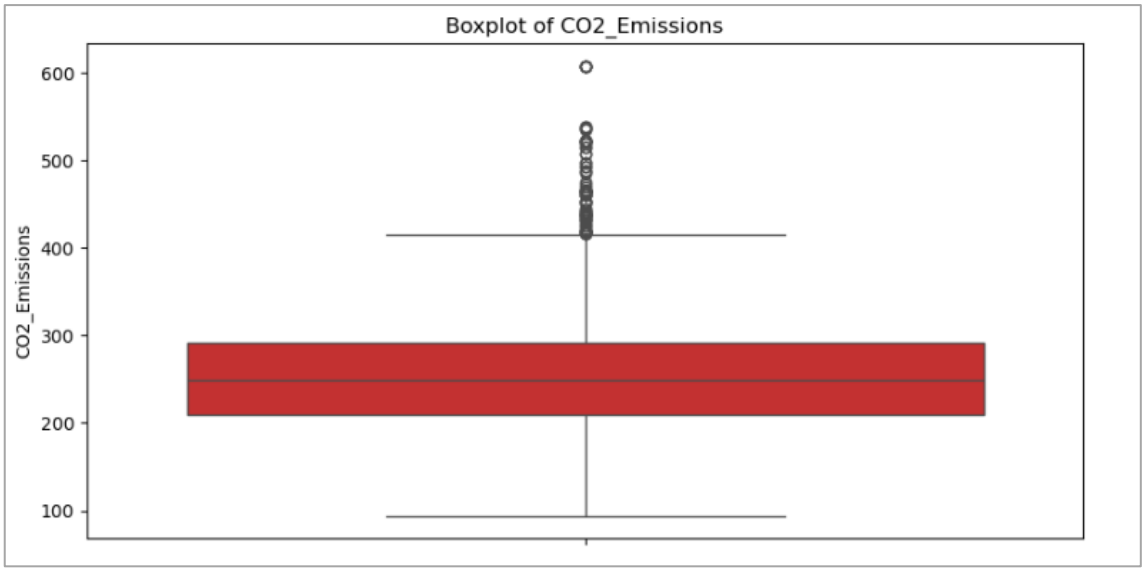

Figure 3.15: Boxplot of CO2_Emissions Outliers

Univariate analysis is a necessary phase in exploratory data analysis as it entails analyzing each variable by itself to get insight into its characteristics and distribution. For categorical features, univariate analysis generally involves computing the frequency or count of each category, as well as the relative percentage or proportion of observations falling into each category. Bar charts, pie charts and frequency tables are widely used for visualizing and conveying categorical variable patterns. As for numerical features, on the other hand, concentrates on measures of central tendency (mean, median, mode), dispersion (range, variance, standard deviation) and distribution form (skewness, kurtosis). Visualizations such as histograms, box plots and density plots are particularly useful for determining the distribution, outliers and mode of numerical data.

Univariate Analysis

Categorical

```
for feature in cat_feat:

    if feature in df.columns:

        count = df[feature].value_counts()
        percent = 100 * df[feature].value_counts(normalize=True)

        data = pd.DataFrame({'Sample': count, 'Percent': percent.round(1)})

        print(f"{feature}:")
        print(data)

        plt.figure(figsize=(15, 5))
        sns.countplot(x=feature, data=df, palette='gist_rainbow', order=df[feature].value_counts().index)
        plt.title(f'Count plot of {feature}')
        plt.show()
    else:
        print(f"The feature '{feature}' is not present in the DataFrame.")
```

Figure 3.16: Univariate Analysis on Categorical Features

Figure above shows a code that performs univariate analysis on categorical features to examine each variable in the dataset individually to summarize and find patterns in the data. Firstly, the for loop iterates over each feature in “cat_feat”, which is a list of categorical feature names, and it processes one feature at a time. For each feature, it checks if the feature exists in the columns of the DataFrame. If it exists, the value counts for that feature will be calculated. Then, a series containing counts of unique values in descending order is returned. It also calculates the percentage of each value by dividing the counts by the total number of rows and multiplying by 100. This percentage calculation provides a clearer understanding of the distribution of each category relative to the whole dataset.

The value counts and percentages are combined into a new DataFrame “data” with columns “Sample” and “Percent”. The percentages are rounded to one decimal place using “round(1)”. This provides a concise summary of the categorical variable, showing both the count and percentage of each category. The feature’s name and the data are printed to allow for immediate visual inspection of the results, helping to quickly identify the distribution of each category. A bar chart is created using functions from the

“matplotlib” and “seaborn” libraries. The chart shows the count of each value in the categorical feature, providing a visual representation of the distribution. The chart title is set using “plt.title()” with the feature name, and the chart is displayed using “plt.show()”. This makes the chart informative and easy to interpret, clearly indicating which feature is being analyzed. If the feature is not present in the DataFrame columns, a message is printed to indicate that it is not present.

However, this code is not used to develop visualizations as shown in Figure 3.17 and Figure 3.18 as the x-axis shown in the chart is not readable, therefore Power BI is used to create bar charts which are like the produced by the code. Figure 3.19 and Figure 3.20 utilize this code to produce the visualizations shown below as there are no issues with its readability.

Vehicle_Make:

Vehicle_Make	Sample	Percent
Ford	862	8.7
Chevrolet	782	7.9
BMW	691	6.9
Mercedes-Benz	616	6.2
Porsche	563	5.7
Toyota	481	4.8
GMC	469	4.7
Audi	408	4.1
Jeep	337	3.4
Nissan	321	3.2
Kia	289	2.9
Hyundai	289	2.9
Honda	282	2.8
Dodge	280	2.8
Lexus	264	2.7
MINI	253	2.5
Mazda	244	2.4
Cadillac	226	2.3
Volkswagen	226	2.3
Jaguar	192	1.9
Subaru	185	1.9
Land Rover	159	1.6
Volvo	158	1.6
Ram	130	1.3
Buick	117	1.2
Maserati	115	1.2
Infiniti	113	1.1
Lincoln	112	1.1
Mitsubishi	104	1.0
Acura	96	1.0
Chrysler	92	0.9
Rolls-Royce	78	0.8
Bentley	73	0.7
Lamborghini	72	0.7
FIAT	66	0.7
Aston Martin	63	0.6
Genesis	60	0.6
Alfa Romeo	55	0.6
Scion	13	0.1
Bugatti	13	0.1
Ferrari	6	0.1
smart	5	0.1

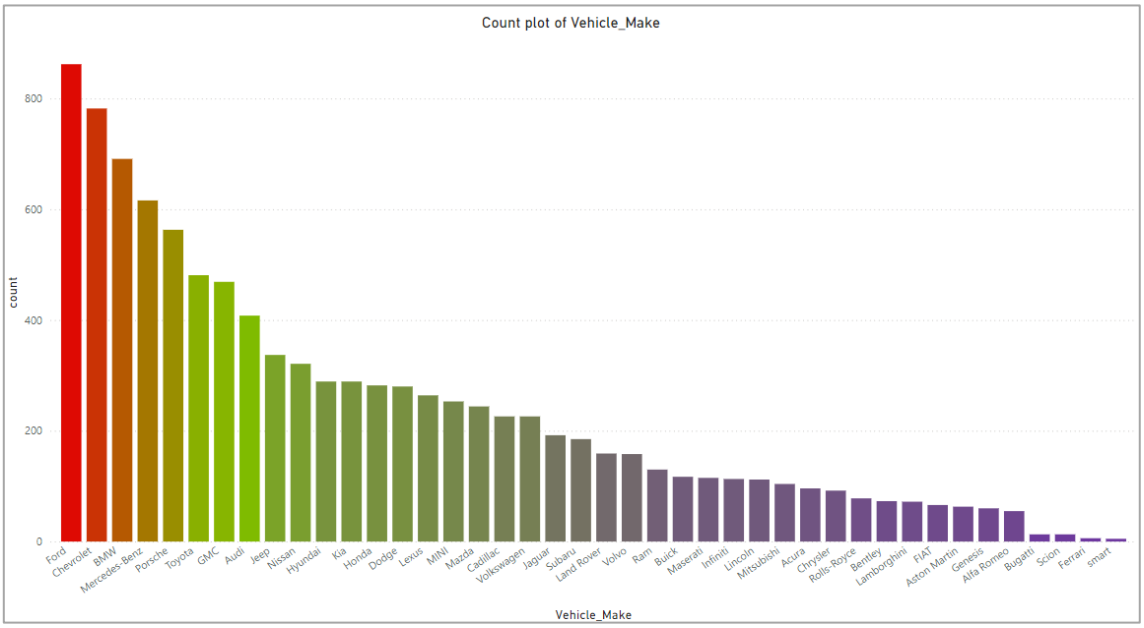

Figure 3.17: Bar Chart displaying Count plot of Vehicle_Make

Vehicle_Class:

Vehicle_Class	Sample	Percent
Sport utility vehicle: Small	1863	18.7
Mid-size	1393	14.0
Sport utility vehicle: Standard	1211	12.2
Compact	1131	11.4
Pickup truck: Standard	874	8.8
Subcompact	850	8.5
Full-size	767	7.7
Two-seater	611	6.1
Minicompact	433	4.3
Station wagon: Small	282	2.8
Pickup truck: Small	220	2.2
Special purpose vehicle	103	1.0
Minivan	95	1.0
Station wagon: Mid-size	82	0.8
Van: Passenger	45	0.5

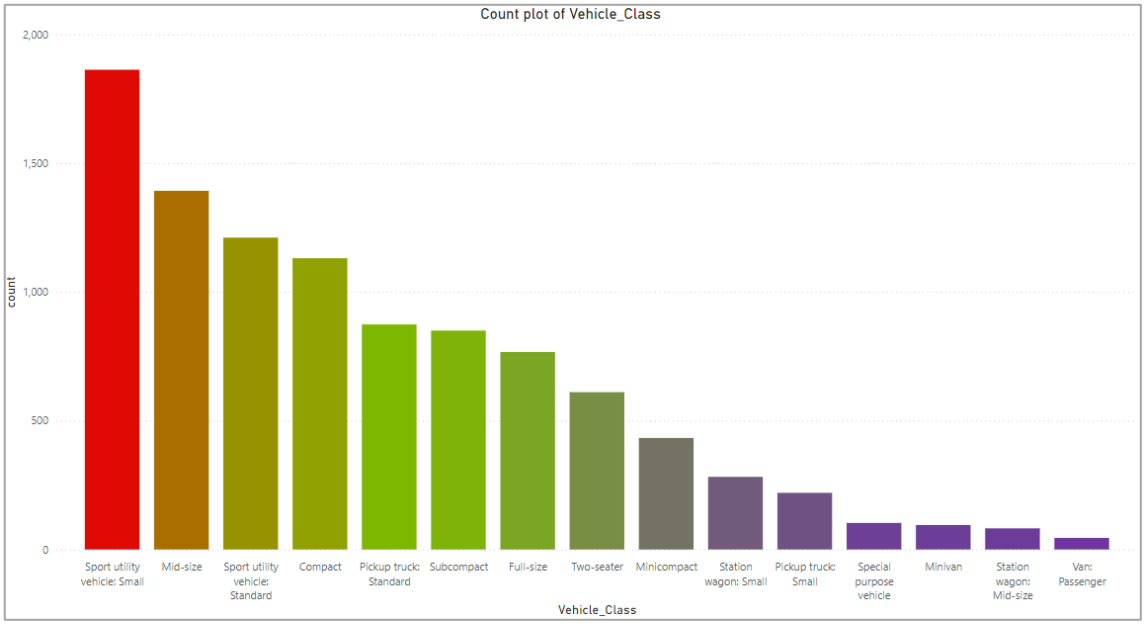

Figure 3.18: Bar Chart displaying Count plot of Vehicle_Class

Transmission:

Transmission	Sample	Percent
AS8	1995	20.0
AS6	1280	12.9
M6	1020	10.2
A8	793	8.0
AM7	678	6.8
A9	642	6.4
A6	633	6.4
AS10	506	5.1
AV	410	4.1
A10	279	2.8
AS7	265	2.7
AM8	232	2.3
AM6	162	1.6
M5	159	1.6
AS9	156	1.6
AV7	148	1.5
AV6	141	1.4
M7	130	1.3
AV8	120	1.2
A5	54	0.5
A7	46	0.5
AV10	39	0.4
AS5	24	0.2
A4	20	0.2
AV1	19	0.2
AM9	6	0.1
AM5	2	0.0
AS4	1	0.0

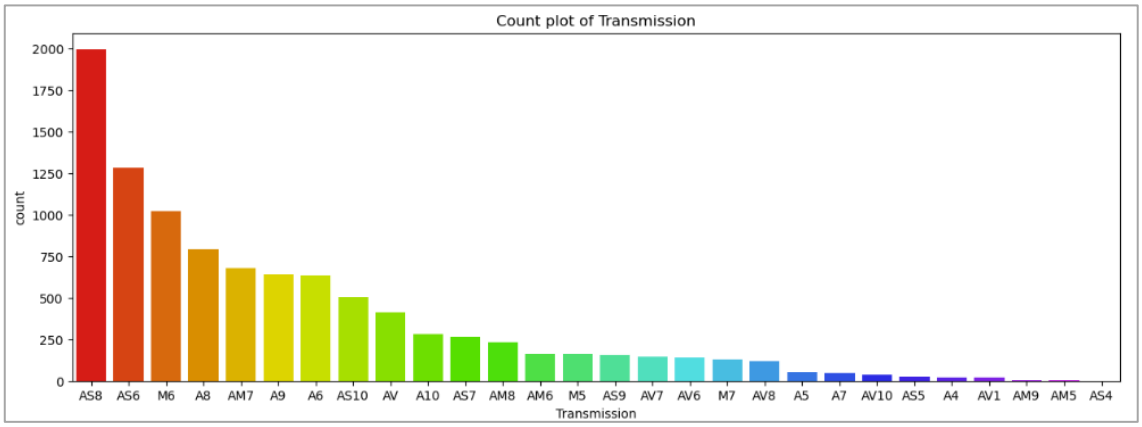

Figure 3.19: Bar Chart displaying Count plot of Transmission

```
Fuel_Type:
           Sample  Percent
Fuel_Type
X            4751     47.7
Z            4644     46.6
E             328      3.3
D             236      2.4
N               1      0.0
```

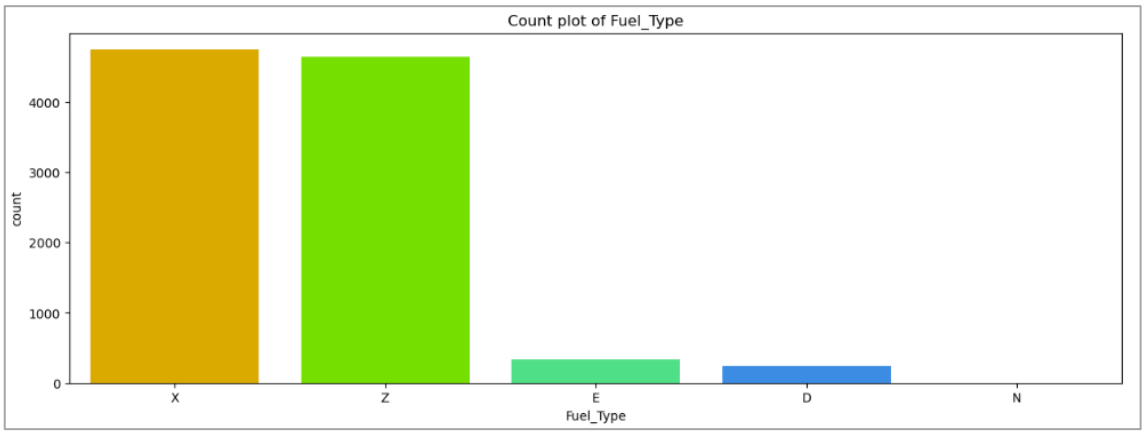

Figure 3.20: Bar Chart displaying Count plot of Fuel_Type

Numerical

```
df.hist(bins = 20, figsize = (15, 15), color = "crimson", grid = False)
plt.show()
```

Figure 3.21: Univariate Analysis on Numerical Features

The figure above shows a code snippet which creates and displays histograms for each numeric column in the DataFrame, using matplotlib for visualization. Next, histograms are generated for all numeric columns using “df.hist(...)”. The parameter “bins=20” indicates how many bins (intervals) to divide the data into for each histogram. A larger number of bins produces a more detailed distribution, whereas fewer bins produce a more general perspective. “figsize” determines the size of the finished figure in inches, which causes the figure to be 15 inches wide and 15 inches tall, making the histograms easily viewable. The parameter “color” sets the color of the bars in the histograms. "Crimson" is used to make the bars a deep red color. After color is set, then the grid lines are removed from the histograms, making the plot cleaner and drawing

emphasis to the data itself. The histogram is then displayed using the “plt.show ()” function. Without this, the histograms will be generated but not displayed in the output.

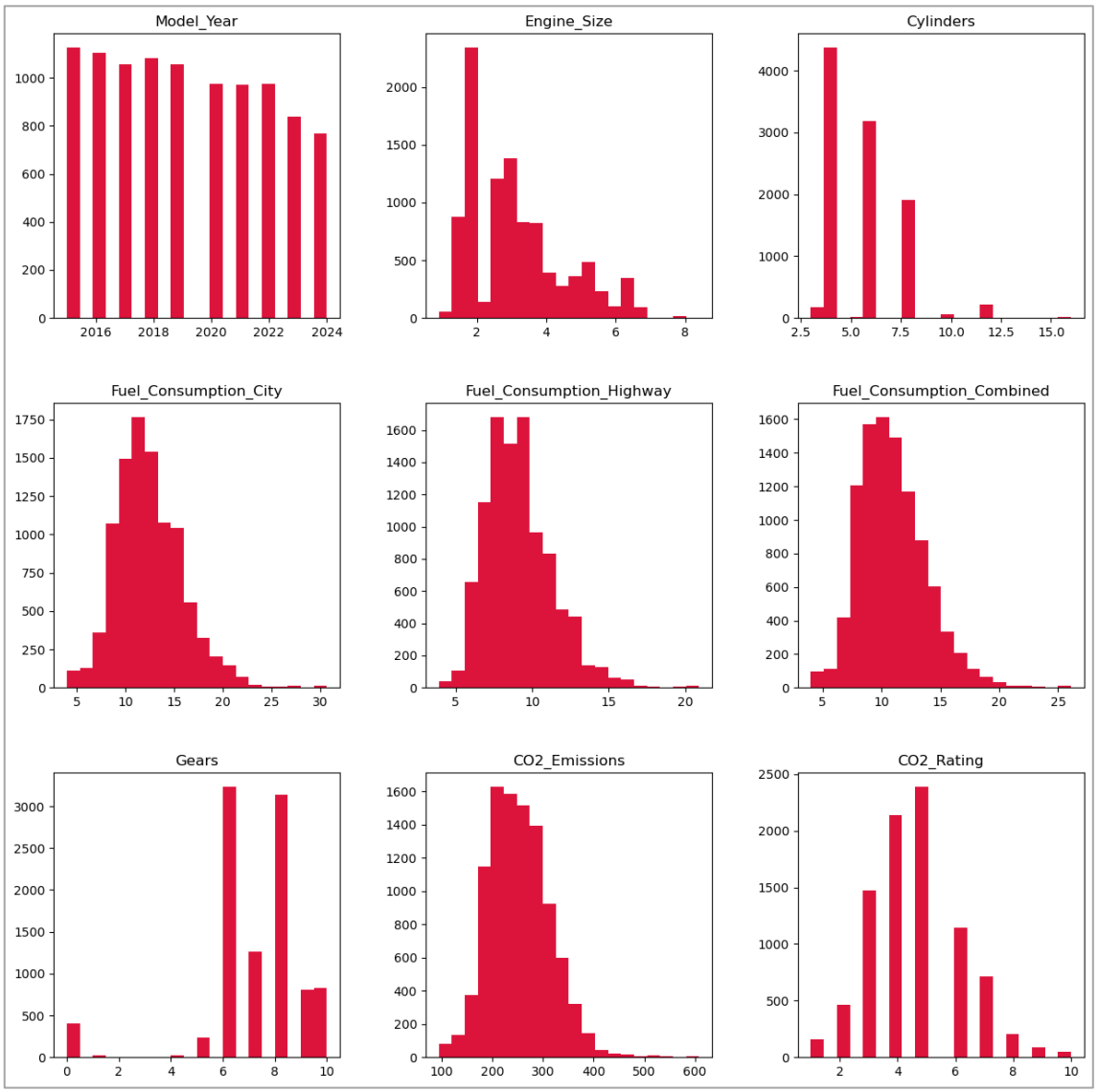

Figure 3.22: Histogram of Model_Year, Engine_Size, Cylinders, Fuel_Consumption_City, Fuel_Consumption_Highway, Fuel_Consumption_Combined, Gears, CO2_Emissions & CO2_Rating

Multivariate analysis is an analysis of the relationships between many variables in a dataset. For categorical features, we can look at how a numerical variable's meaning varies across multiple categories. Visual representations are created to help identify possible connections in CO2 emissions across categories. For numerical features, the standard method is to generate a scatter plot matrix or pair plot that shows scatter plots for each pair of numerical variables in the data. Therefore, a pair plot with regression lines is created and placed over the scatter plots. This representation aids in the identification of linear or nonlinear relationships, correlations and potential patterns among numerical data. It can also identify probable outliers or clusters in the data.

Figure 3.23: Multivariate Analysis on Categorical Features

The presented code snippet in Figure 3.23 above helps to visualize the correlation between categorical variables and CO2 emissions in the dataset with bar charts. The function “df.select_dtypes(include='object')” selects all columns in the DataFrame with the data type “object”, which is often used for categorical data. Then, the column names change into a list, so that they can be assigned to the categorical variable. A loop is initiated that iterates through the column names in the categorized list. Each iteration will process one categorical feature. Categorical charts are generated by using “sns.catplot(...)” function from the Seaborn library. In this case, it is utilized to create bar graphs. Afterwards, the x-axis and y-axis are set to indicate that CO2 emissions want to

be compared against multiple features. By using the parameter “kind=bar”, it sets that the plot should be a bar chart. Also, the height and aspect ratio of the plot is determined. The data source is specified along with the colour palette. the title of each plot is displayed by using the function “plt.title("Mean CO2 Emissions - {}".format(col))”, indicating which categorical variable is being examined. It uses the col variable to dynamically add the name of the current category column to the title.

However, since the x-axis in the chart is unreadable, this code is not utilized to construct the visualizations displayed in Figures 3.20 and 3.21; instead, Power BI is used to build bar charts identical to those produced by the code. Figures 3.22 and 3.23 use this code to generate the visualizations displayed below since it is easily readable.

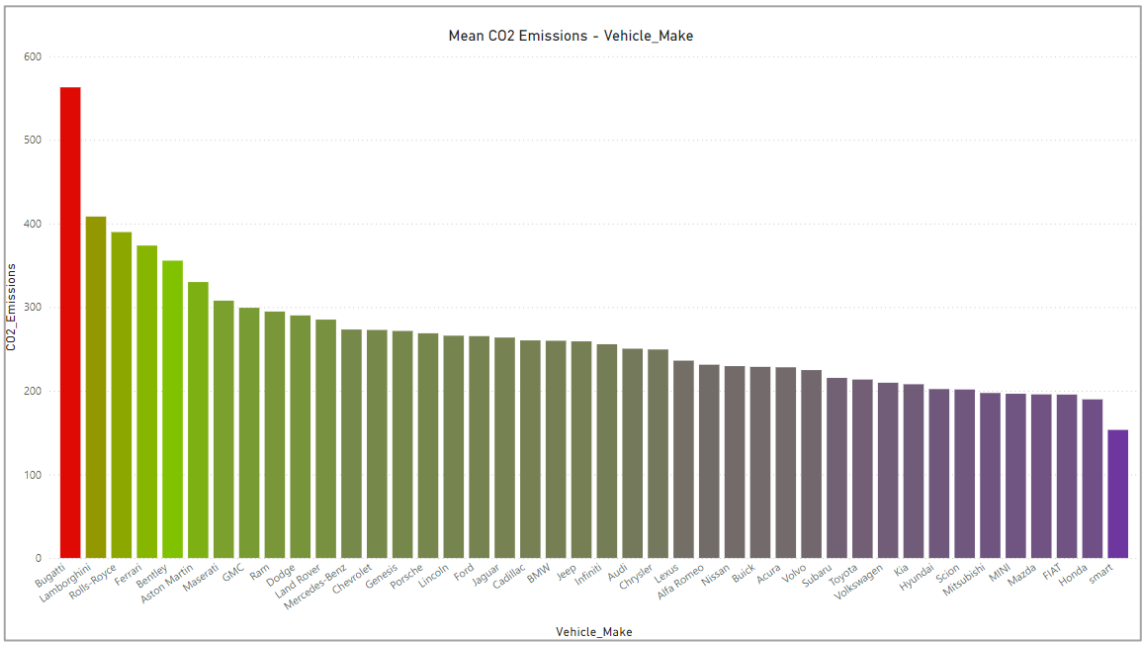

Figure 3.24: Bar Chart displaying Mean CO2 Emissions of Vehicle_Make

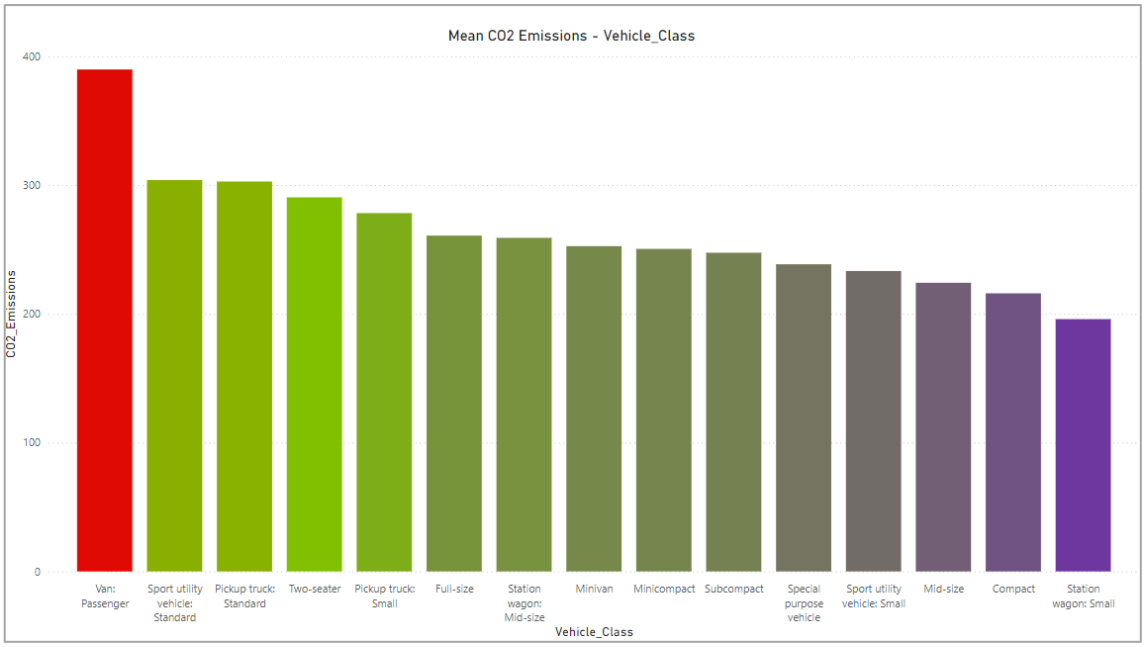

Figure 3.25: Bar Chart displaying Mean CO2 Emissions of Vehicle_Class

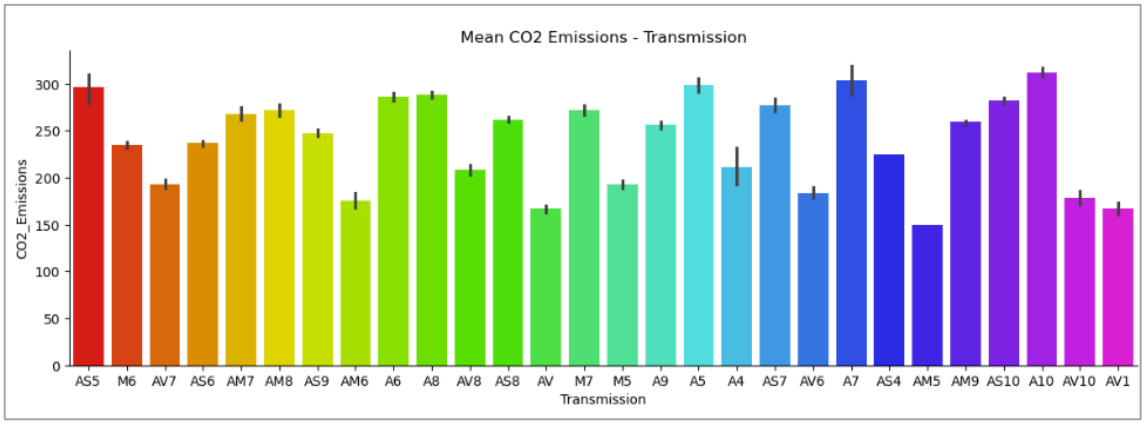

Figure 3.26: Bar Chart displaying Mean CO2 Emissions of Transmission

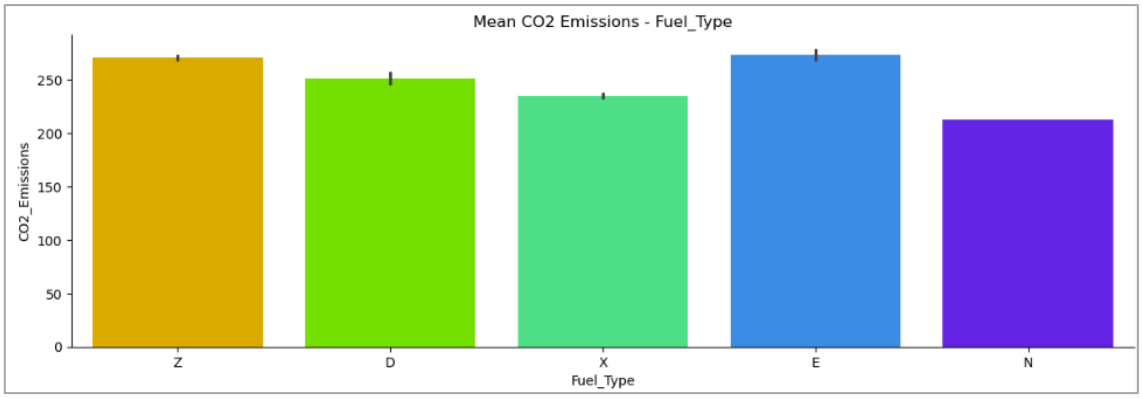

Figure 3.27: Bar Chart displaying Mean CO2 Emissions of Fuel_Type

Figure 3.28: Multivariate Analysis on Numerical Features

Above figure presents a code snippet which creates a pair plot of the DataFrame, including regression lines for each pair of variables. “sns.pairplot(df, kind="reg")” function is from the seaborn library that generates a pair plot, often known as a scatterplot matrix, which displays pairwise associations between variables. The parameter “df” is the source of data to use for plotting. Then, the next parameter “kind="reg"” is to ensure that the charts include regression lines. This means that not only scatter plots will be displayed for the correlations between variables, but a regression line will also be shown too. Lastly, “plt.show ()” function from the matplotlib library is used to show the plots. Without this, the plots will be generated but not displayed in the output.

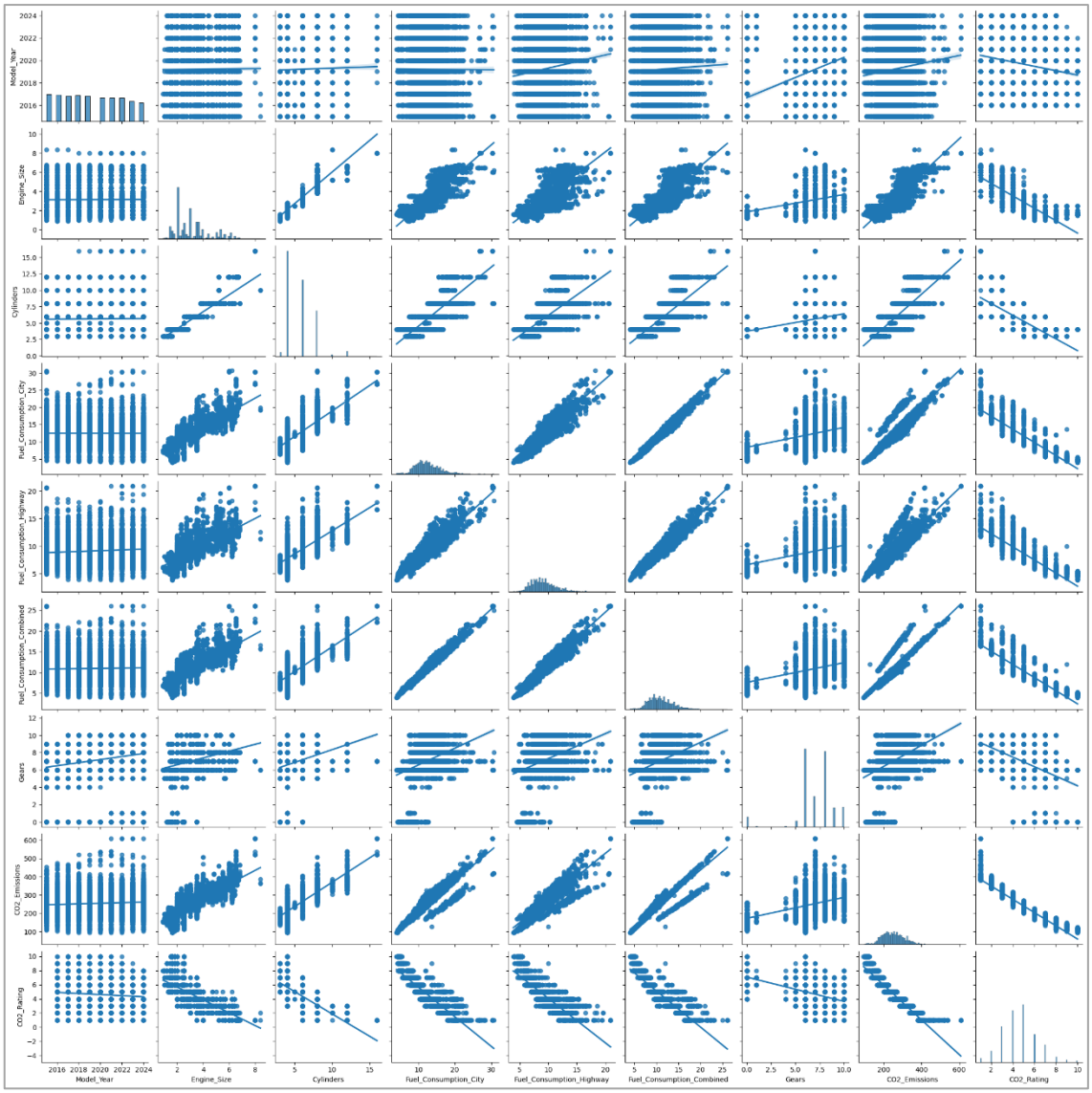

Figure 3.29: Scatter Plot and Regression Lines between Model_Year, Engine_Size, Cylinders, Fuel_Consumption_City, Fuel_Consumption_Highway, Fuel_Consumption_Combined, Gears, CO2_Emissions & CO2_Rating

Based on Figure 3.27 above, comparing the features with the CO2_Emissions which is the last row, it can be concluded that Engine_Size and Cylinders have a strong positive correlation. Also, Fuel_Consumption_Combined will be considered due to it is the combination of both highway and city fuel consumption.

Correlation Score

```
plt.figure(figsize=(10, 8))

numeric_df = df[num_feat]

correlation_matrix = numeric_df.corr().round(2)

sns.heatmap(data=correlation_matrix, annot=True, cmap='coolwarm')

plt.title("Matrix Correlation for Numeric Features", size=14)
plt.show()
```

Figure 3.30: Correlation Scores of Engine_Size, Cylinders, Fuel_Consumption_City, Fuel_Consumption_Highway, Fuel_Consumption_Combined, Gears & CO2_Emissions

A code snippet which generates and displays a heatmap of the correlation matrix for numeric features in a DataFrame is shown in the above figure. This matplotlib function “plt.figure(figsize=(10, 8))” creates a new figure for plotting, setting the size to 10 inches wide and 8 inches height for readability. This “numeric_df = df[num_feat]” produces a new DataFrame numeric_df that solely contains the numeric properties of the original DataFrame df. The names of these numeric columns can be found in the “num_feat” list. Furthermore, the correlation matrix for numeric_df is generated, which displays pairwise correlations between numerical features. The coefficients are rounded to two decimal places for easier reading. By using a function from the seaborn library, a heatmap is generated to represent the correlation matrix. It annotates heatmap cells with correlation values and utilizes the “coolwarm” colormap to express correlation strength and direction (warm colors for positive correlations, cool colors for negative correlations). Then, the title of the heatmap to "Matrix Correlation for Numeric Features" with a font size of 14 points is set. At last, the heatmap is displayed as well as the title.

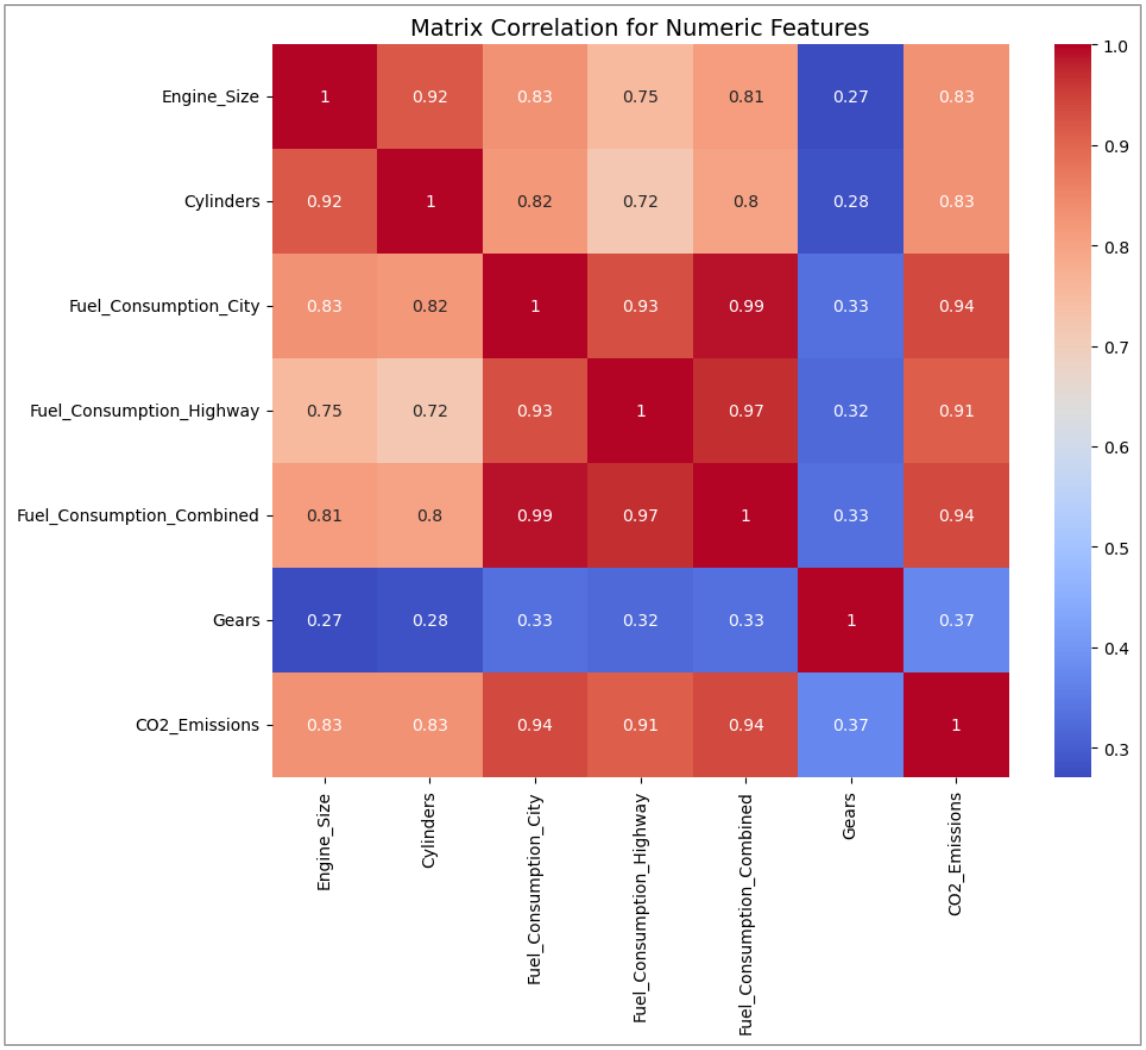

Figure 3.31: Heat Map of Correlation Scores between Engine_Size, Cylinders, Fuel_Consumption_City, Fuel_Consumption_Highway, Fuel_Consumption_Combined, Gears & CO2_Emissions

3.4 Dashboard Preliminary Sketches

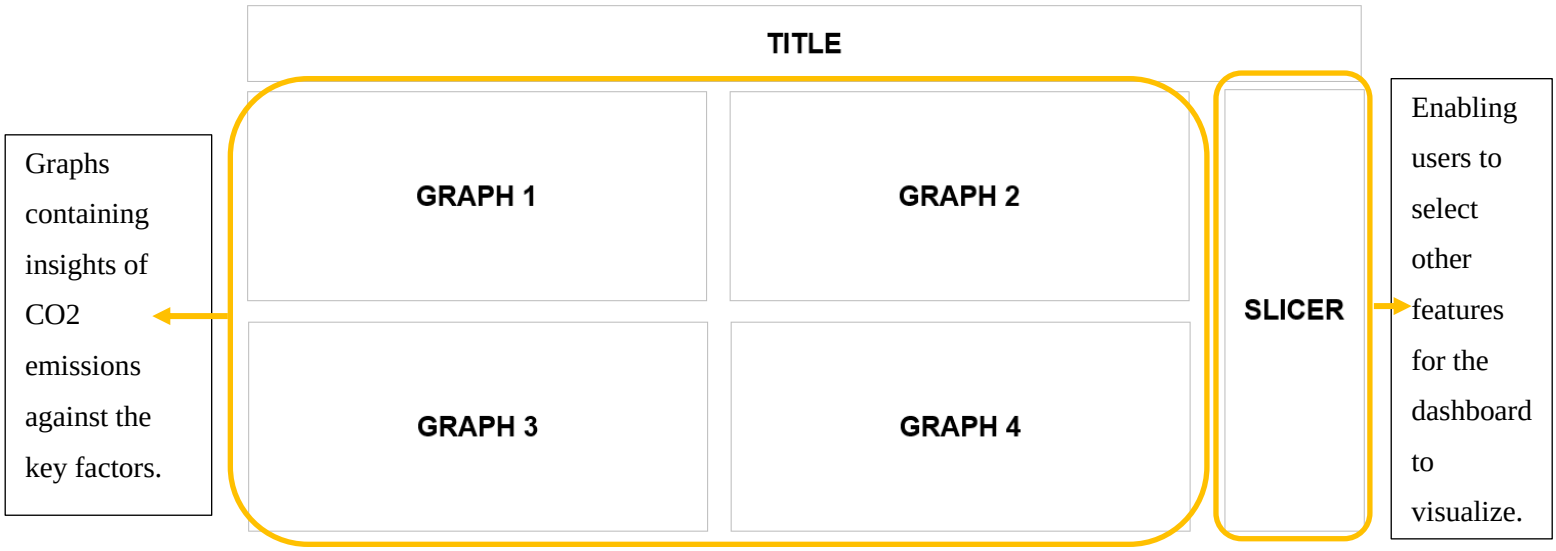

Figure 3.32: Sketch of Descriptive Visualizations Dashboard

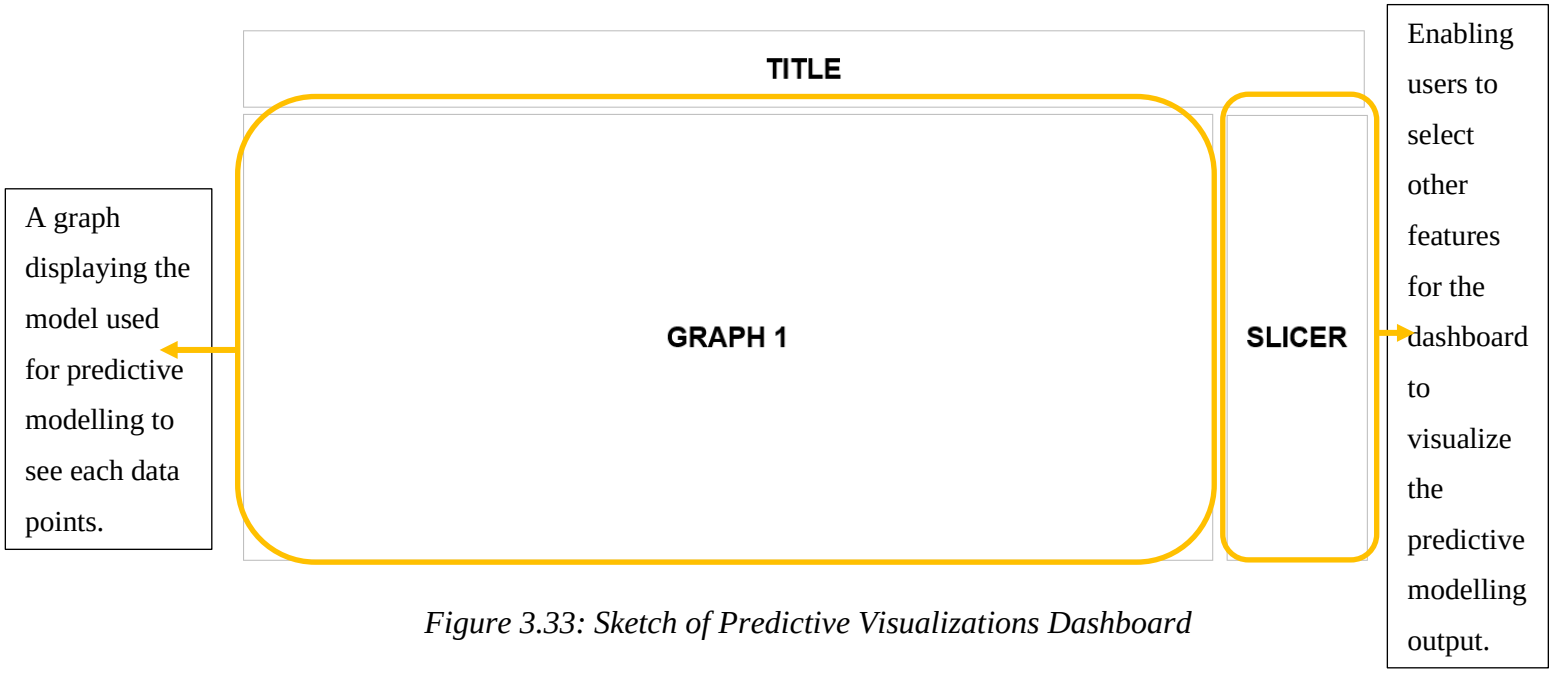

Figure 3.33: Sketch of Predictive Visualizations Dashboard

CHAPTER 4

MODEL DEVELOPMENT AND EVALUATION

4.1 Model Development

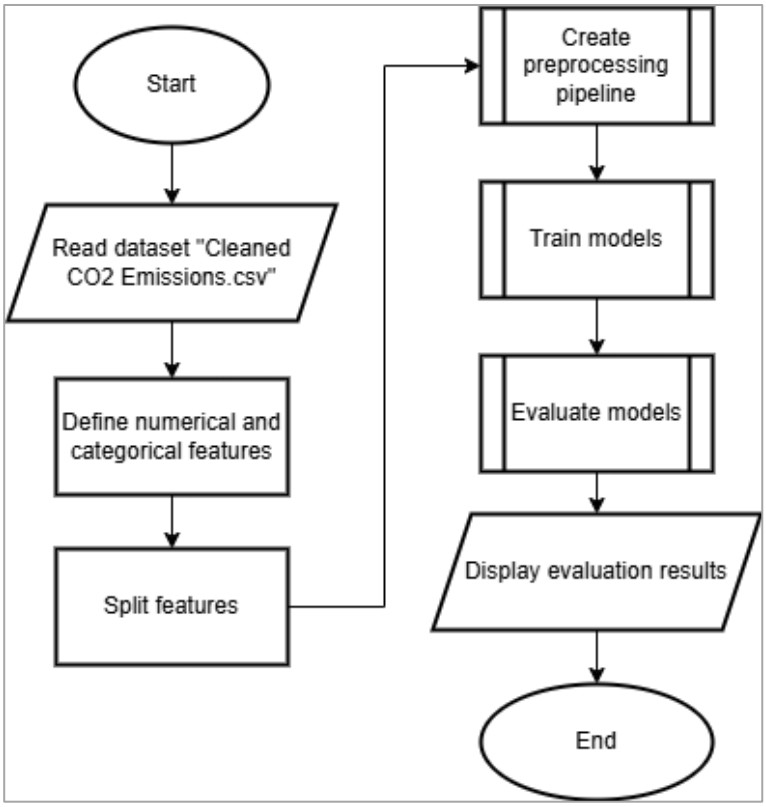

Figure 4.1: Flowchart of the Model Development

The model development begins with loading the dataset, which reads the input file "Cleaned CO2 Emissions.csv" into memory. This dataset includes data regarding engine size, fuel consumption metrics, transmission type as well as the target variable, CO2 emissions. The objective is to predict CO2 emissions using existing features, ensuring that the data is ready for processing and analysis.

The next step is to define numerical and categorical features. Numerical features include engine size, cylinders and a variety of fuel consumption metrics. Categorical features include qualitative variables such as vehicle make, model, class, transmission

type and fuel type. This separation ensures that the proper preprocessing processes may be applied to each feature group.

Once features have been identified, the dataset is divided into features (X) and target (y). The features, X, include all columns except the CO2 emissions column, which is used for the target variable, y. This separation enables the model to learn patterns from the predictors while comparing performance with actual emission values.

A preprocessing pipeline is then developed to get the data ready for modelling. Standardization is done to numerical features by normalizing the values with a StandardScaler, ensuring that all variables are on the same scale. One-hot encoding transforms categorical values into binary columns, allowing models to process them. These transformations are combined using a ColumnTransformer, which systematically applies the proper preprocessing to the various feature types.

Two models for predictions are then trained, which are Adaptive Boosting (AdaBoost) Regressor and Linear Regression. Each model is integrated into a pipeline through the preprocessing phase, which ensures that the data is consistently transformed before training. AdaBoost is a boosting technique that creates several weak learners to improve prediction accuracy, whereas Linear Regression is a simpler, more interpretable approach to modelling.

Each model is evaluated using 5-fold cross-validation. During this process, the dataset is divided into five subsets, with each subset functioning as a validation set once and the remaining subsets used for training. This ensures reliable performance measurements by averaging the results across multiple folds. Model accuracy and goodness-of-fit are evaluated using key metrics such as Root Mean Squared Error (RMSE) and coefficient of determination (R^2).

Finally, the evaluation results are presented in a properly structured table that highlights the performance of each model. This result enables for an easy comparison of the AdaBoost Regressor and Linear Regression, assisting in determining the best-performing model for predicting CO2 emissions.

4.2 Model Evaluation

Table 4.1: Cross-Validation Results

Models	RMSE	R^2
AdaBoost Regressor	13.499560	0.945954
Linear Regression	5.456686	0.990946

Table 4.2: Cross-Validation Prediction Results

Models	RMSE	R^2
AdaBoost Regressor	13.560223	0.946285
Linear Regression	5.635654	0.990722

As shown in Table 4.1 and Table 4.2, the model evaluation results demonstrate the performance of two regression models, Adaptive Boosting Regressor (AdaBoost) and Linear Regression, using cross-validation and prediction metrics. The evaluation measures used are Root Mean Squared Error (RMSE) and coefficient of determination (R^2). The RMSE measures the average extent of prediction errors, with lower values indicating more accuracy. R^2 is a measure of how well the model explains the range in the target variable, with values closer to 1 indicating a better fit.

The Adaptive Boosting Regressor has a cross-validation RMSE of 13.50 and a R^2 of 0.946. The model predictions to have a R^2 of 0.946 and an RMSE of 13.56. These findings show that AdaBoost captures patterns reasonably well, explaining around 94.6% of the variance in CO2 emissions. However, the model's somewhat high RMSE indicates that its predictions have a higher average error than the Linear Regression model.

The Linear Regression model executes well, with cross-validation results of RMSE 5.46 and R^2 0.991. The prediction results are equally accurate, with RMSE 5.64 and R^2 0.991. These results show that Linear Regression not only explains 99.1% of the variance in CO2 emissions but also makes predictions with much lower average error. The consistency of results across both evaluation phases highlights Linear Regression's reliability and precision.

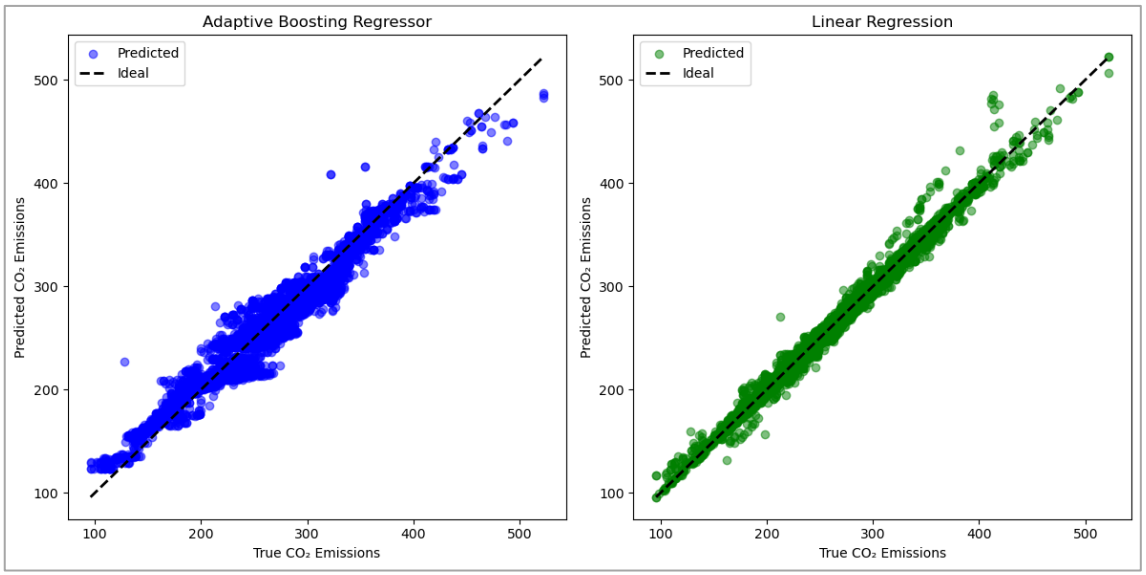

Figure 4.2: Scatter Plots of Actual vs Predicted CO2 Emissions Values for AdaBoost Regressor and Linear Regression

The scatter plots in Figure 4.2 evaluate how two models, Adaptive Boosting Regressor and Linear Regression, predicted CO2 emissions. Both graphs compare actual CO2 emissions on the x-axis with predicted plots on the y-axis, with an ideal prediction line, the dashed black line, when the predictions exactly match actual values. Both

models have high predictive performance since their points cluster closely around the ideal line but there are several significant differences. The Linear Regression model in green color, looks to have a slightly tighter clustering of points around the ideal line, especially in the middle emission range (200-400), implying more consistent predictions. The Adaptive Boosting Regressor in blue color, exhibits slightly more scattering and gaps from the ideal line, particularly at higher emission values.

Based on the evaluation measures shown in Table 4.1 and 4.2 as well as the scatter plots in Figure 4.2, Linear Regression outperforms Adaptive Boosting Regressor in terms of accuracy and simplicity. Linear Regression has lower RMSE and higher R^2, indicating better predictions with less computing complexity. Furthermore, its alignment with the project's objectives makes it the best choice. Linear regression is simple to set up, requires minimal technical knowledge and skills as well as scalable for large datasets. Furthermore, its manageable system allows it to be used efficiently even without strong machine learning experience. As a result, Linear Regression has been selected as the best-performing model, satisfying the project's objectives of ease of use and accurate CO2 emissions prediction.

4.3 Selected Model Development

The chosen model is developed using the steps mentioned in Section 4.1, with some additional phases integrated. These steps involve calculating the residuals, which indicate the differences between the model's predicted and actual values and then saving both the predicted and residual values to a CSV file. This extra step ensures that these values are easily accessible for later usage in dashboard development.

CHAPTER 5

DASHBOARD DEVELOPMENT AND TESTING

5.1 Dashboard Design and Development

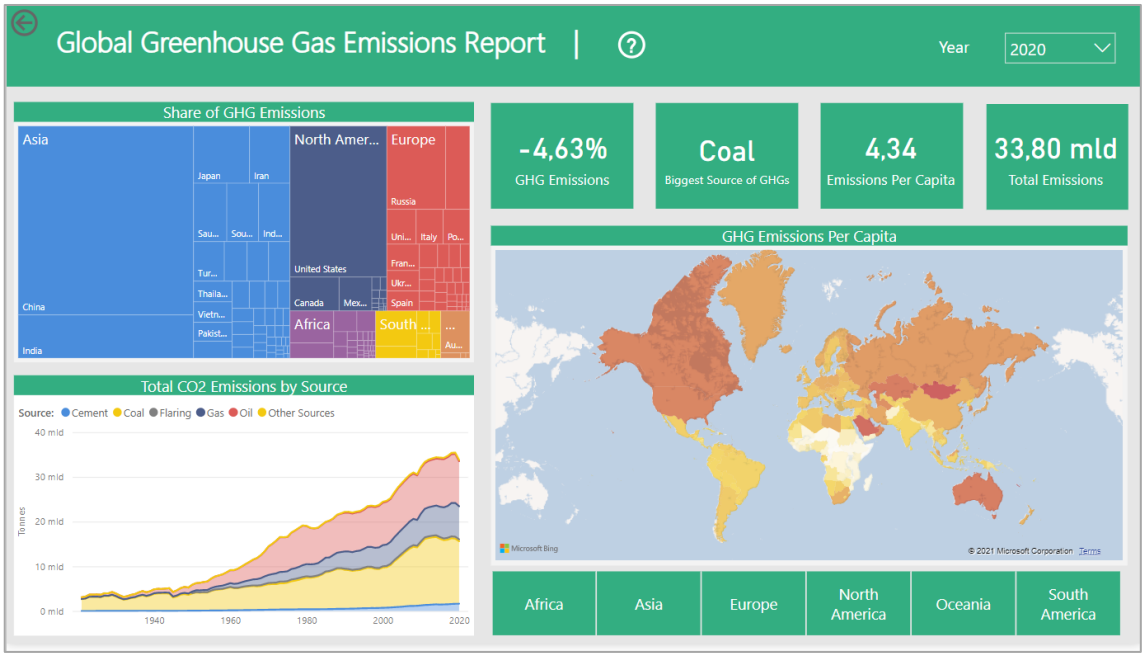

Figure 5.1: Example 1 of Existing Dashboard

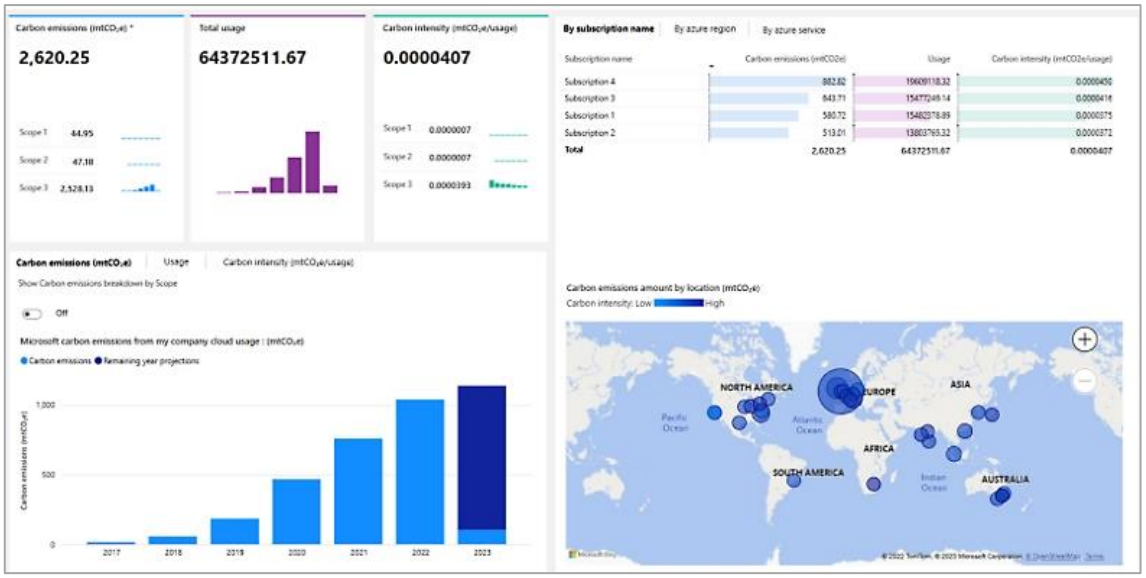

Figure 5.2: Example 2 of Existing Dashboard

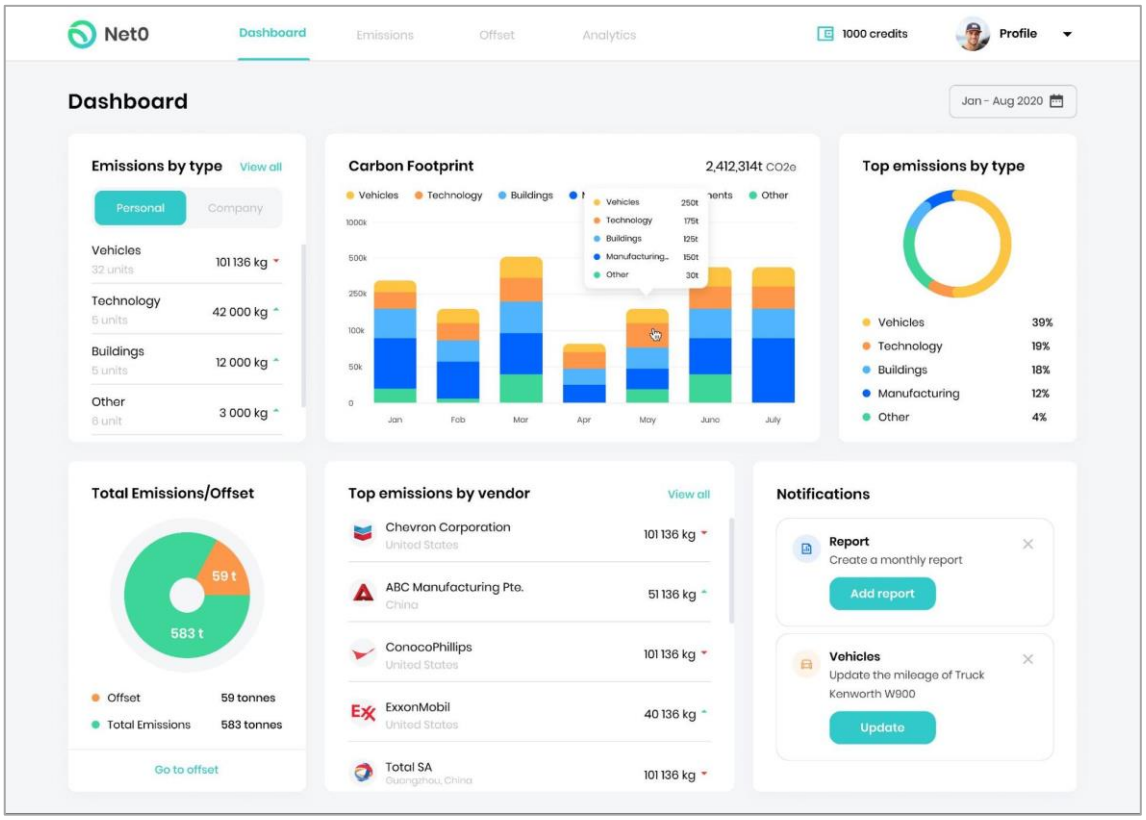

Figure 5.3: Example 3 of Existing Dashboard

The analytics dashboard was created after an extensive review of existing dashboards to identify the essential visualization needs that best support decision-making processes. These existing dashboards, as shown in the figures above, provide important insights into the types of visualizations that should be included on the coming dashboard's descriptive and predictive pages. Figure 5.1, for example, highlights the importance of visualizing CO2 emissions in relation to essential features by displaying Total CO2 Emissions by Source using a stacked area graph. Similarly, Figure 5.2 shows how a geological map can efficiently visualize CO2 emissions by location, providing a geographical perspective on emission data.

In addition to these examples, the reviewed dashboards highlight the significance of providing specific insights, such as identifying the top contributors to CO2 emissions. For example, Figure 5.3 shows a visualization of the Top CO2 Emissions by Vendor, allowing users to identify the major sources of emissions and focus on areas which need attention. These examples demonstrate the importance of incorporating diverse and relevant visualizations that address various analytical objectives.

By reviewing these examples and understanding their strengths, the process of generating meaningful and effective visualizations for CO2 emissions along with related features becomes much more informative. The resulting dashboard can be organized to not only display important trends and patterns but also deliver actionable insights which help in decision-making by looking at the curated graphs. This conscious and informed approach assures that the dashboard efficiently fulfils its intended function in both descriptive and predictive analytics visualizations.

The next step in the dashboard development process is to turn data into an operating and visually appealing dashboard. This starts with constructing initial wireframes, such as those shown in Figures 3.30 and 3.31, which outline the basic structure and arrangement of components. These wireframes act as blueprints, indicating where important components such as slicers, graphs and other graphic elements will be placed to ensure simple navigation and an easy-to-use interface.

The process then moves on to develop mock-ups, as shown in Figures 5.4 and 5.5, in which several graph types were tried using trial and error to determine their ability in representing data. For example, graphs displaying CO2 emissions against key features such as engine size, fuel type or vehicle class were thoroughly reviewed for clarity and relevance. This iterative method enabling discovery and enhancement of the visuals to best convey the intended insights.

Aside from graph selection, other important design decisions were taken early in the process to develop a consistent visual character for the dashboard. The color palette, fonts and overall design were thoughtfully selected to make the dashboard visually appealing and accessible to the intended users. Consistent colors were used to differentiate across data types and readable fonts were used to improve comprehension. Furthermore, slicer placement was carefully considered, ensuring that they were intuitively situated so that users could easily filter data.

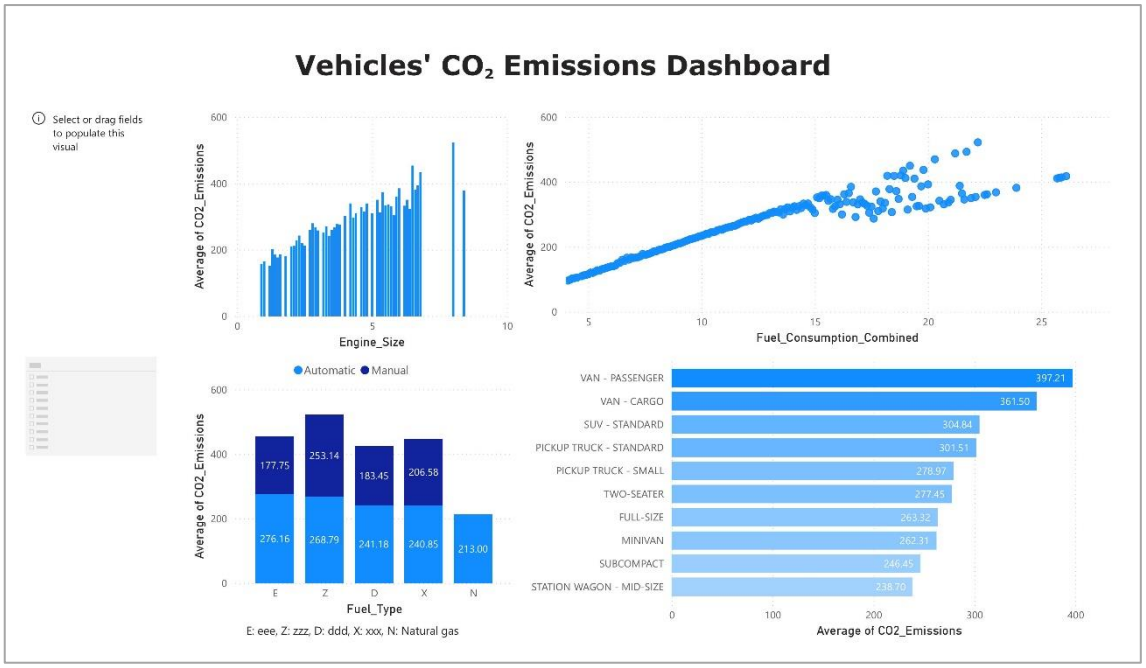

Figure 5.4: Design Mock-up of the Descriptive Visualizations Dashboard

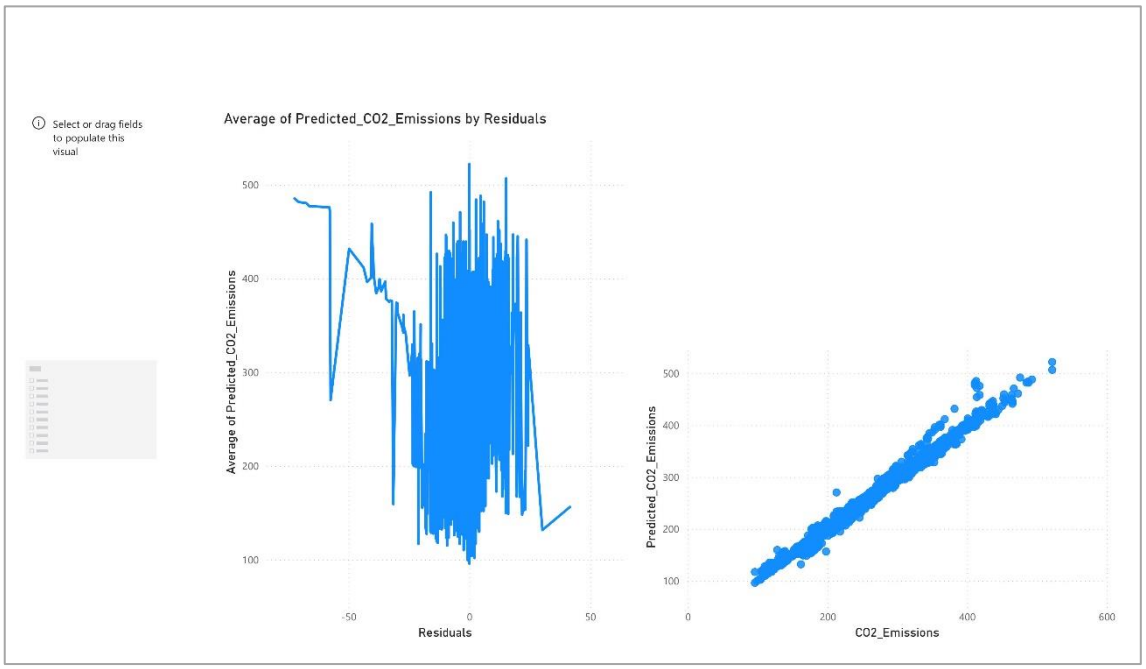

Figure 5.5: Design Mock-up of the Predictive Visualizations Dashboard

After these mock-ups were looked at and modified, the dashboard started the implementation phase, creating the first version of the dashboard as displayed in Figure 5.6 and 5.7. The feedback from the mock-ups as well as the first version motivated me to further develop and redesign, resulting in improved element placement and

arrangement. To simplify user interactions, slicer buttons were intentionally positioned at one location on the page.

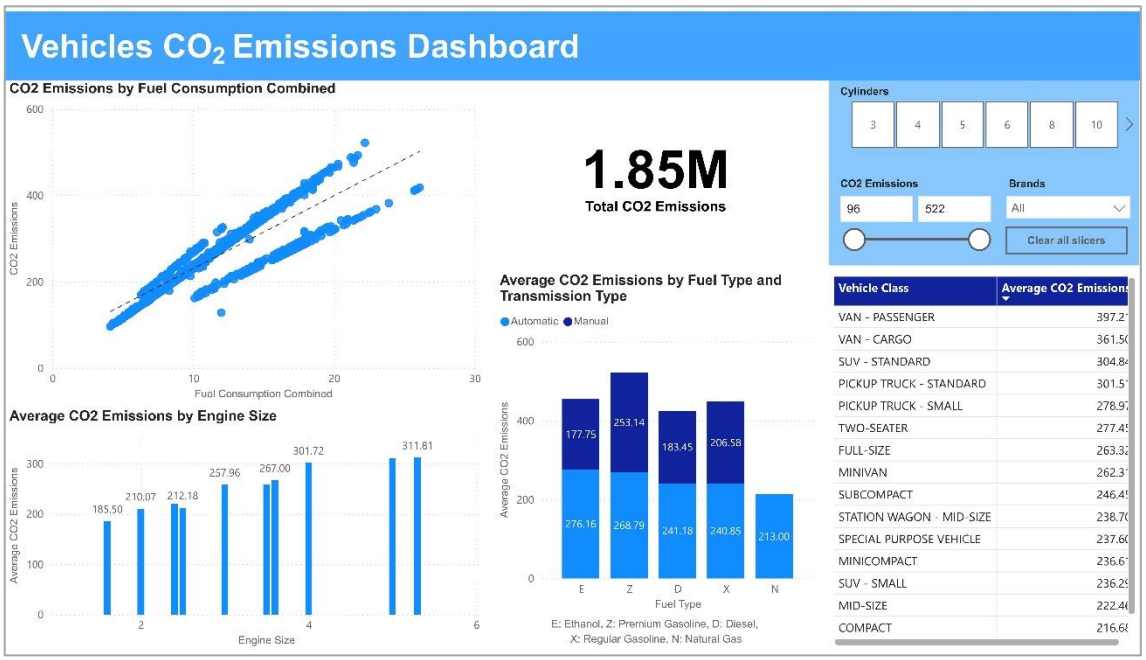

Figure 5.6: Descriptive Visualizations Dashboard (1st Version)

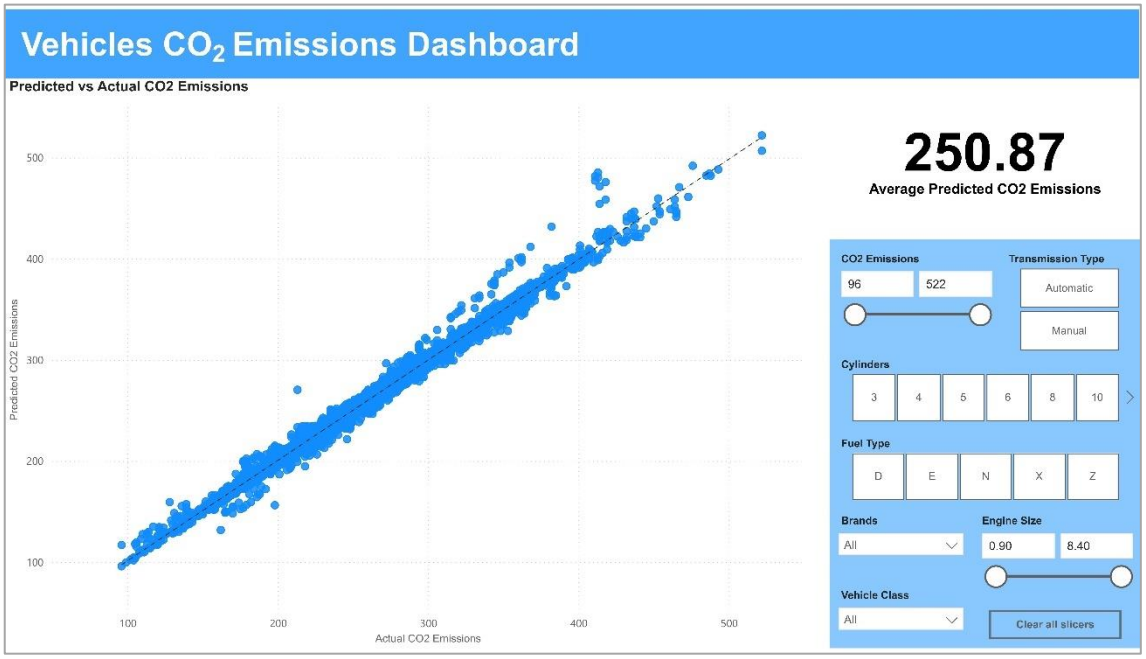

Figure 5.7: Predictive Visualizations Dashboard (1st Version)

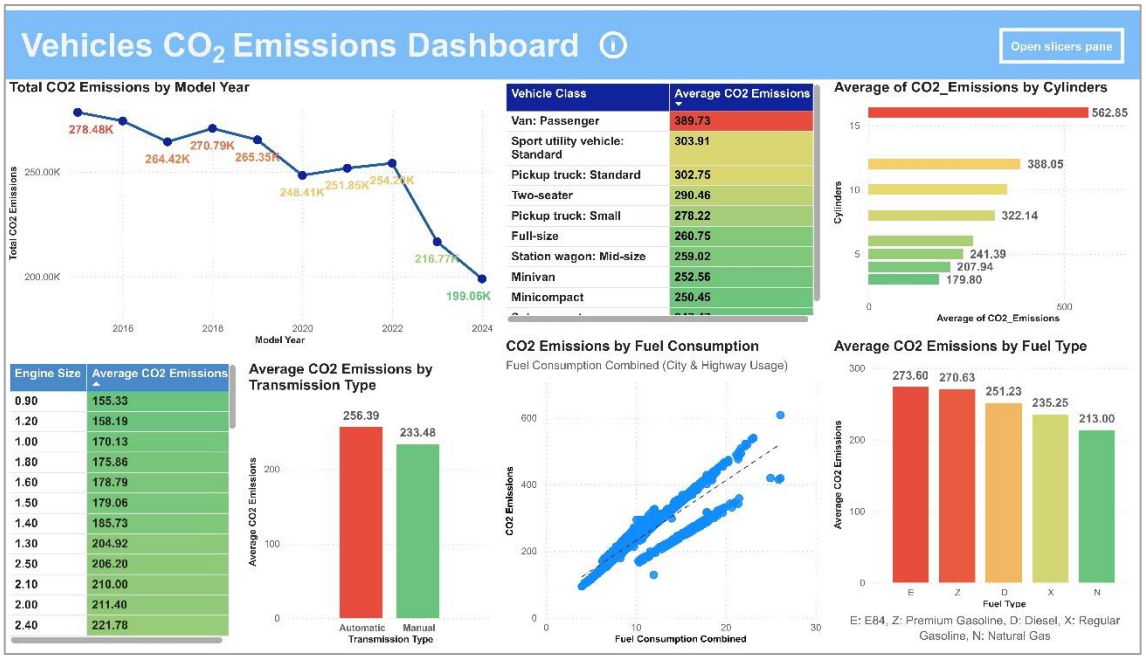

Figure 5.8: Descriptive Visualizations Dashboard (Finalized Version)

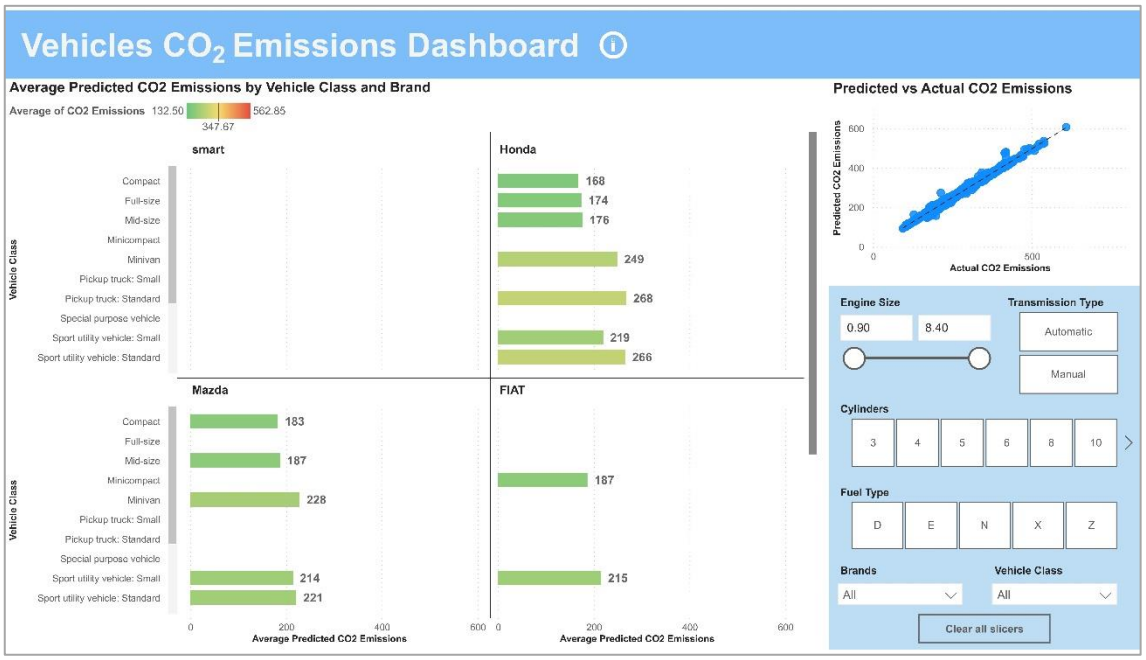

Figure 5.9: Predictive Visualizations Dashboard (Finalized Version)

Figures 5.8 and 5.9 show the final versions of the dashboard, which represent the outcome of these efforts. They include all the enhanced design components, resulting in a well-organized, visually appealing and highly functional tool for analyzing both descriptive and predictive CO2 emissions data. This arranged development process

emphasizes the need for iterative design, user-focused changes and careful integration of crucial elements in creating an effective analytical dashboard.

The dashboard was created using Power BI, an innovative and user-friendly platform for data visualization and analysis. Power BI is especially useful because it enables easy dataset connection and offers a variety of customization choices for creating interactive dashboards.

In order to use the analytics model built in Chapter 4, the model's outputs were initially saved as a CSV file. This step ensures that the data is in a format that allows for easy uploading into Power BI. When the CSV file was complete, it was put into Power BI as a dataset. The dashboard is built around this dataset, which contains the model's predictions and the other key features.

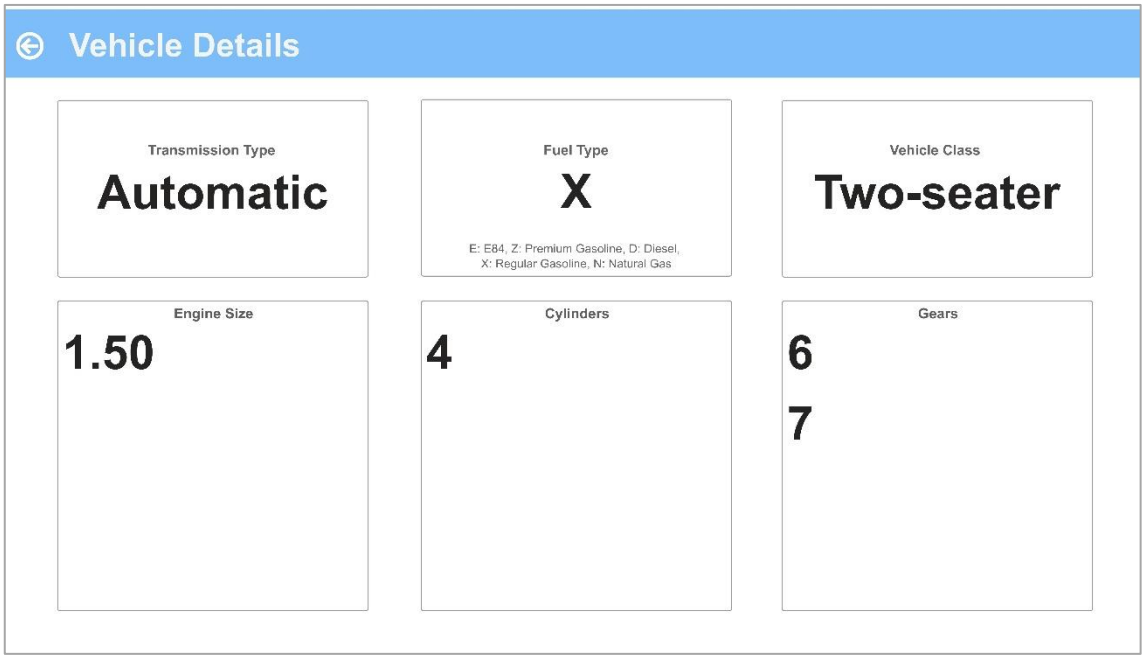

Figure 5.10: Predictive Visualizations Drill-through Page (Finalized Version)

Figure 5.10 is a drill-through page of Figure 5.9, allowing car manufacturers to dive deeper into the features that cause the vehicle brand and class to have a low average CO2 emissions compared to others.

In Power BI, the model's data was linked to various visualizations ensuring that the results from the model were clearly represented. Predictive values, such as predicted CO2 emissions based on the features in the dataset, are presented alongside descriptive visualizations, such as emission trends by vehicle type or fuel consumption. This integration ensures that the manufacturers can engage with the dashboard for exploring both current data and future predictions, allowing for informed decision-making and actionable insights.

5.2 Dashboard Testing

5.2.1 Targeted User 1's User Acceptance Test

System	Vehicles CO2 Emissions Dashboard			
Purpose	To test the functionality and interactivity of all elements for all pages.			
Test Case Details				
Page: Descriptive				
Step	Description	Expected Result	Actual Result	Comment/ Remark
1	Test slicer functionality • To ensure all data is shown based on selected feature in the slicer.	All graphs display data on the selected feature.	As expected.	Add Transmission Type as a feature in the slicer.
2	Test graph interactivity • To ensure if clicking a column any of the graphs changes relevant data in other graphs.	Related graphs change and correspond according to the data of the clicked column.	As expected.	Add Cylinders and Transmission Type as one of the graphs.
Page: Predictive				
1	Test slicer functionality • To ensure all data is shown based on selected feature in the slicer.	All graphs display data of the selected feature.	As expected.	
2	Test slicer reset functionality • To ensure the "Clear all slicers" button reset to the original data.	All graphs and slicers reset to display the original data.	As expected.	
3	Test graph interactivity • To ensure if clicking a column any of the graphs changes relevant data in other graphs.	Related graphs change and correspond according to the data of the clicked column.	As expected.	Sort the graph based on CO2 emissions (lowest to highest).

Test by	
Date	17/01/2025
Pass/Fail	Pass
Comment/ Remark	

Acceptance of Testing (Signatories)	
Verify by	Accepted by
Name:	Name: Farhah Syahmina
Date: 17/01/2025	Date: 17/01/2025

For dashboard testing, my examiner will carry out the user acceptance test to check that all elements and graphs in the dashboard work as intended.

CHAPTER 6

CONCLUSION

6.1 Project Outcome

The objectives guide the project's outcomes, which all contribute to the overall success and impact. The project uses linear regression as its supervised learning method to predict vehicle CO2 emissions. This method is intended for producing a predictive model that serve as an important resource for accurately measuring and predicting emission levels, which is used to develop predictive visualizations.

Second, the project aims to able to display descriptive and predictive visualizations on the dashboard. These visualizations show CO2 emissions patterns and trends for different vehicle classes, providing a clear understanding of emission differences. By displaying these data, the project helps identify key features which causes the CO2 emissions.

Finally, the project desires to provide the necessary data to car manufacturers, encouraging to develop and produce cleaner, more environmental-friendly vehicles. These insights can motivate manufacturers to align their products with regulatory standards and environmental sustainability goals, ultimately contributing to reducing CO2 emissions.

6.2 Strengths and Weaknesses

The project showcased several strengths that contributed to its successful completion. As the project was completed on top of other academic responsibilities, effective time management was critical. The availability of information, such as research

articles and industry reports, was another major advantage. Previous research gave useful insights on CO2 emissions, machine learning techniques and best practices for dashboard development, forming a solid foundation for the project's design and implementation.

However, the project was not without challenges. A significant shortcoming was a lack of proficient skills and knowledge in machine learning techniques and data visualization tools. This limitation affected efforts at times, especially during the predictive model development and interactive dashboard feature design. To overcome these issues in the future, further training, workshops or collaboration with data science and visualization specialists may be necessary. Addressing these areas can boost the project's impact, generating a more comprehensive and effective tool for assessing and predicting vehicle CO2 emissions.

6.3 Future Recommendations

Future work on the Vehicles CO2 Emissions dashboard can improve both its descriptive and predictive capabilities. For the descriptive section, combining real-time data updates and interactive filtering functions can provide more dynamic insights, allowing users to explore emission patterns across many features such as geographic regions and vehicle ages. Advanced machine learning algorithms such as Gradient Boosting or Neural Networks may improve the predictive model's accuracy and reliability. Additionally, including other variables such as weather conditions or traffic patterns, could improve predictions and enhance the model. Expanding the dashboard's usability by offering personalized emission-reduction recommendations based on specific vehicle profiles would also be beneficial.

REFERENCES

Armstrong, M. (2022, April 14). Miles apart: car CO2 emissions. *Statista Daily Data*. https://www.statista.com/chart/27253/car-CO2-emissions-by-size-type-statista-mmo/

Ding, S., Shen, X., Zhang, H., Cai, Z., & Wang, Y. (2024). An innovative data-feature-driven approach for CO2 emission predictive analytics: A perspective from seasonality and nonlinearity characteristics. *Computers & Industrial Engineering*, 110195. https://doi.org/10.1016/j.cie.2024.110195

Fleck, A. (2023, September 22). Cars cause biggest share of transportation CO2 emissions. *Statista Daily Data*. https://www.statista.com/chart/30890/estimated-share-of-CO2-emissions-in-the-transportation-sector/

Highsmith, J., & Cockburn, A. (2001). Agile software development: the business of innovation. *Computer*, *34*(9), 120–127. https://doi.org/10.1109/2.947100

Hindle, G. A., & Vidgen, R. (2018). Developing a business analytics methodology: A case study in the foodbank sector. *European Journal of Operational Research*, *268*(3), 836–851. https://doi.org/10.1016/j.ejor.2017.06.031

SoobiaEtAl, S. (2019). Analysis of software development methodologies. *International Journal of Computing and Digital System/International Journal of Computing and Digital Systems*, *8*(5), 445–460. https://doi.org/10.12785/ijcds/080502

The Editors of Encyclopaedia Britannica. (2024, May 15). *Carbon dioxide | Definition, Formula, Uses, & Facts*. Encyclopedia Britannica. https://www.britannica.com/science/carbon-dioxide

Unsw. (2020, January 29). *Descriptive, Predictive & Prescriptive Analytics: What are the differences?* https://studyonline.unsw.edu.au/blog/descriptive-predictive-prescriptive-analytics

Vehicle emissions | Green Vehicle Guide. (n.d.). https://www.greenvehicleguide.gov.au/pages/UnderstandingEmissions/VehicleEmissions

Chadha, A. S., Shinde, Y., Sharma, N., & De, P. K. (2022). Predicting CO2 emissions by vehicles using Machine Learning. Lecture Notes on Data Engineering and Communications Technologies, 197–207. https://doi.org/10.1007/978-981-19-2600-6_14

European Environment Agency. (2024, December 16). CO2 emissions performance of New Passenger Cars in Europe. European Environment Agency's home page. https://www.eea.europa.eu/en/analysis/indicators/co2-performance-of-new-passenger#:~:text=To%20achieve%20climate%20neutrality%2C%20the,by%20diesel%20cars%20(17%25).

Khajavi, H., & Rastgoo, A. (2023). Predicting the carbon dioxide emission caused by road transport using a random forest (RF) model combined by meta-heuristic algorithms. Sustainable Cities and Society, 93, 104503. https://doi.org/10.1016/j.scs.2023.104503

Singh, S. K., & Kumari, S. (2022, January). (PDF) Machine Learning-based time series models for effective CO2 emission prediction in India. Predicting CO2 Emissions by Vehicles Using Machine Learning. https://www.researchgate.net/publication/358419604_Machine_learning-based_Time_Series_Models_for_Effective_CO2_Emission_prediction_in_India

Wang, S., & Ge, M. (2019, October 16). Everything you need to know about the fastest-growing source of global emissions: Transport. World Resources Institute. https://www.wri.org/insights/everything-you-need-know-about-fastest-growing-source-global-emissions-

transport#:~:text=Emissions%20from%20the%20transport%20sector%20are%20a,of%20CO2%20emissions%20from%20burning%20fossil%20fuels.