University of Essex
Department of Mathematical Sciences

MA981: DISSERTATION

Predicting Automobile Carbon Emissions Based on Vehicle Characteristics and Usage Patterns

Siddhant Patil
2201538

Supervisor: Dr.Ariyo Oludare S.

August 24, 2023

Colchester

Abstract

Globally all are concerns about rising global warming. For this CO2 emission through vehicles are important contributors to this problem. To find the solution for this problem, all need to be understand the essential factors which driving vehicle emissions. This study is about exploring the complex relationship between vehicle emissions and various vehicle components and their usage behaviours. Dataset used in this study from the Canadian Government that includes a wide range of vehicle details such as brand, type, year, engine specifications, and fuel type. With the help of data analysis, some key results came forward which helped in model building process . Vehicle size, fuel type, and transmission system like these components strongly impact both fuel consumption and CO2 emission. Between 2000-2010, there is a continuously increased in the emission of CO2 gas through vehicles and after this,till now a days is increasing gradually. To tackle this, one need to predict this emission rate to save our future. By applying advanced statistical methods, including Linear Regression, Random Forest Regressor, and XG Boost Regressor, tried built models to predict carbon emissions, ensuring reliable results through cross validation. The models' accuracy, as indicated by their R-squared values, further confirms their effectiveness in capturing the variability in carbon emissions. The Linear Regression model achieved an R-squared of 0.79, showing its predictive accuracy. The Random Forest Regressor and XG Boost Regressor models outperformed with higher R-squared values of 0.88, highlighting their enhanced predictive capabilities. The results uncover valuable insights such as vehicle size, fuel used, transmission used in vehicle which affects the co2 emission. Looking forward, there are exciting possibilities for further investigation.

Keywords— CO2 emission, Linear Regression , fuel consumption

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

As global concerns about climate change and its impact on our planet continue to rise, understanding where greenhouse gas emissions comes from becomes incredibly important.The global atmosphere experiences global warming due to the heat-trapping effect of greenhouse gases like carbon dioxide (CO2), methane (CH4), and nitrous oxide (N2O). These gases are released due to different things humans do on daily basis, like burning fossil fuels for electricity and heat, producing things in factories, cutting down trees, and farming. Among these activities, the way we use transportation is a big source of CO2 emissions. Vehicles like cars, trucks, and buses release CO2 into the air when they burn fuels like gasoline and diesel. How much CO2 comes out depends on a few things, like how good the vehicle is at using fuel, how big its engine is, what kind of fuel it uses, and how people drive it. [1]

It's really important to understand why CO2 emissions are a problem and need immediate attention. CO2 is a major greenhouse gas, which means it helps keep heat in the atmosphere, which causes to global warming and changes in the climate. When we have more CO2 and other greenhouse gases in the air, it makes the planet hotter, melts ice on the poles, raises sea levels, and causes extreme weather like storms and floods. These changes can have a big impact on nature, animals, farming, and people all around the world. Because vehicles are a big reason for CO2 emissions, we have to study what things make these emissions happen. Figuring this out is really important for finding ways to fight against climate change. By looking at how different things about vehicles affect CO2 emissions, we can find new ideas for making transportation better for the planet. This will help lower the harmful effects of CO2 emissions on our environment.This project will help people who make rules, companies, and regular people make choices that are better for the Environment. The main purpose of this data science project is to figure out what things about vehicles make them release CO2 into the air. It has to know how different things about vehicles, like the kind of vehicle it is, how it

shifts gears, what fuel it uses, and how big its engine is, change the amount of CO2 it makes.A lot of information using in this project is from the Canadian Government. They have a big set of data that tells us about many different vehicles, like when they were made, what kind they are, and how they work.

Using some statistical analysis like descriptive statistics, inferential statistics to find out the distribution of data, relation between various attributes of vehicles and other insights from data and math techniques, like Linear Regression, Random Forest Regressor, and XG Boost Regressor, to make computer models that can predict how much CO2 a vehicle will make. Need to be select models to make sure they are good at guessing. The numbers from the models show that they're pretty good at figuring out how much CO2 a vehicle will make.The multi linear regreesor model is machine learning model uses the other column to predict the output.From this model the names of vehicle variables come forward which are definatly affects on the emission.Other two models Random forest and XG Boost models are advance models,they performs a lot of models (decision tress) in back and give combined result. A lot of things found about a vehicle through this study, like what it's made of, can make its CO2 emissions higher or lower. Also, shows that how a vehicle is used and what fuel it uses can also change how much CO2 it makes. In the future, it would be interesting to look at other places and see if vehicles in different parts of the world make different amounts of CO2 emissions.

CHAPTER 2

Literature Review

During the examination of the available research to learn how researchers dealt with the problem of vehicle pollution on the environment. With literature review the different methods they have used to research and reduce the negative effects of emissions.

It is essential to address the harmful impacts caused by fuel consumption of vehicles, and its emissions, mostly by passenger cars, are now coming to light due to the increasing speed of climate change. The effect of these emissions can be generally grouped into two main categories: emissions that negatively impact air pollution and human health and emissions that play a role in the more significant problem of climate change. Of these emission gases, co2 is the most significant contributor to greenhouse gas emissions and the most effective for the temperature rise. For example, in the European Union, road transport generates about one-fifth of the total carbon dioxide emissions, with passenger cars causing nearly 75 percentage of this share. [2] The link between fuel consumption and CO2 emissions is straightforward and problematic. [3]

Electric vehicles have emerged as a crucial step in the industry's seek for carbon reductions to address the growing environmental challenges. But according to the International Energy Agency, to keep global warming below the crucial 2 Â°C limit by 2030, at least 20 Percentage of all road transport vehicles or approximately 300 million units must be electrically powered [4]. Light-duty vehicles using low carbon degrees have been set up to be important during this transition phase.Looking at all the above research and results, it is essential to consider the effect of co2 emission from vehicles. We need to study this problem in detail and make some practical suggestions.

The Second paper talks about how lockdowns that happened because of COVID-19 affected three big cities in the US. The authors noted the two significant effects of lockdown; one is bad as a hard phase for the economy, and the second is good, clean air. For this, they used different information

sources to understand how this happened.A lot of different information like how people were moving, how the economy was doing, and pictures from space to figure out what was happening because of the lockdowns, is mentioned in this paper. [5]

Due to the lockdown, when people stop moving and how the people become poorer. At the same time, the lockdowns made the air better by reducing a type of pollution called NO2. The authors show that when people moved around less because of the lockdowns, the economy went down too, which could make more people poor. At the same time, the lockdowns improved the air by reducing pollution called NO2. This is important because pollution like NO2 can make lung problems worse and increase the chances of getting sick from COVID-19. This study is important because it discusses how lockdowns impact the environment and the economy. They showed how lockdowns may improve the air quality by using NO2 as an example. But they noticed that the lockdowns caused different changes in air pollution in different cities. This indicates that there many considerations to be made when studying lockdowns and their effects.The study also discusses how lockdowns impacted people of various communities. They discovered that because of the lockdowns, poorer communities felt more issues. This is significant because it shows that different people are affected in different ways. [5]

At last, the authors state that lockdowns can be useful and also harmful. They not only restricted the economy but also helped with preventing the virus. To see how things are changing, they suggest analysing data and images taken from space. In the future, this will direct our decisions and keep people safe while boosting the economy. [5]

The Third paper presents a way to predict how much fuel will use by a big truck or vehicle. They made a computer model that looks at seven things: how many times the vehicle stops, how long it stops for, how fast it goes, how it accelerates, how the air pushes against it, and changes in energy. These things help understand how the vehicle moves. They used this model to predict fuel use for different distances travelled, instead of using a specific time. This makes sense because how far a vehicle goes is related to how much fuel it uses. The model was accurate, almost like other models that use physics. [6]

They tried different settings for the model, like looking at 1 kilometre or 2 kilometres of travel. They found that looking at 1 kilometre was the best, especially for vehicles that travel in cities or short distances. For long trips, like on highways, looking at 5 kilometres was okay. In the future, they want to study more things, like how heavy a vehicle is and how old it is. They will add more information to the model to make it even better. They also want to figure out how often they need to update the model to keep it working well. This study helps me understand how vehicles use fuel, and the new model they made is good at predicting it. This can help me look for vehicles that use less fuel and

create less pollution. [6]

The Fourth study is about figuring out how much pollution vehicles create especially smaller ones like cars and trucks. People use computer models to estimate this pollution. The two most common kinds are MOBILE and EMFAC, but they are more successful in large compared to small areas. So as to better predict pollution for smaller places the researchers developed a new model called CMEM which focuses on how cars travel. [7]

The usefulness of CMEM was determined by the researchers. They compared real measurements of car pollution with the assumptions from CMEM. They found that CMEM works quite well in predicting pollution, in particular at medium speeds. At extremely low or very high speeds, it becomes less accurate. [7]

Also, they evaluated CMEM with the previous MOBILE and EMFAC models. They found while the CMEM is like these models in some respects, it performs better in others. For example, becoming familiar with CMEM is helpful. [7]

The researchers also talked about how they developed CMEM over a few years. They made sure it could predict pollution for different types of vehicles and situations. They used measurements from real vehicles to check if CMEM was right.

In the end, this study helps me to understand how to predict pollution from vehicles better. The computational models are good at it, especially for certain situations. With the help of this we can move forward in the path of computational model to predict the co2 emission.This is important for making our air cleaner and planning ways to reduce pollution from vehicles. [7]

CHAPTER 3

Dataset Description

The dataset was taken from the official open data portal of the Canadian government website.This data from one of the open data sources Kaggle. [8]

Here is the link of dataset:

https://www.kaggle.com/datasets/abhikdas2809/canadacaremissions. [8]

This dataset contains vehicle data from 1995 to 2022 which is around 27 years. This dataset provides a lot information about vehicles in detailed like company, model, engine size, number of cylinder, co2 emission rate etc. some of the key points about dataset are:

1. Dataset description:

- The dataset has a total of 26,075 entries (rows).

- The index is of type 'Int64Index' with values ranging from 1 to 26,075.

2. Columns information:

This dataset consists of 15 columns (attributes), each representing different information:. [8]

- MODEL YEAR': The year of the vehicle model.
- 'MAKE': The manufacturer of the vehicle.
- MODEL (high output engine) ': The model name of the vehicle
- 'VEHICLE CLASS': The class/category of the vehicle
- 'ENGINE SIZE (L)': The engine size in litres.
- 'CYLINDERS': The number of cylinders in the engine

- ‘TRANSMISSION‘: The type of transmission
- ‘FUEL TYPE‘: The type of fuel used.
- ‘FUEL CONSUMPTION CITY (L/100) ‘: Fuel consumption in city driving.
- ‘FUEL CONSUMPTION HWY (L/100) ‘: Fuel consumption in highway driving
- ‘COMB (L/100 km) ‘: Combined fuel consumption
- ‘COMB (mpg)‘: Combined fuel consumption in miles per gallon
- ‘CO2 Rating‘: CO2 emission rating
- ‘Smog Rating‘: Smog rating

3.Missing Values:

- The ‘CO2 Rating‘ column has 7,171 non-null values and 18,904 missing values.
- The ‘Smog Rating‘ column has 6,061 non-null values and 20,014 missing values.

.

This dataset contains information about various components of vehicles, such as their model year, manufacturer, engine specifications, fuel consumption, CO2 emissions, and emission ratings. ‘CO2 Rating‘ and ‘Smog Rating‘ have missing values that I need to deal with them in my future analysis to get better results.. [8]

CHAPTER 4

Methodology

In this section, need to be describe the step-by-step procedure of how to proceed into this dissertation analysis and model building.This section is all about the explanation of each method, process,model and give information about why to use these methods.

4.1 Handling missing values

After importing the data, first need to checked the shape of the dataset, which indicates that it has 26075 rows and 15 columns. In the first step, need to checked for duplicate rows in the dataset to ensure data integrity. Fortunately, there were no duplicated rows in this data, which is a positive indication of data cleanliness. So, each row represents a unique entry, and each column contains specific information about those entries.

Next was examining the information for each column in the dataset.The first 13 columns had complete data without any missing values.After looking at the 14th column named 'CO2 Rating' and the 15th column named 'Smog Rating',noticed that there were some missing values. Specifically, there were 7171 missing values in the 'CO2 Rating' column, and 6061 missing values in the 'Smog Rating' column. These missing values can create challenges during data analysis and might affect the accuracy of predicting CO2 emissions from vehicles. Therefore, this need to find an appropriate method to handle these missing values effectively.

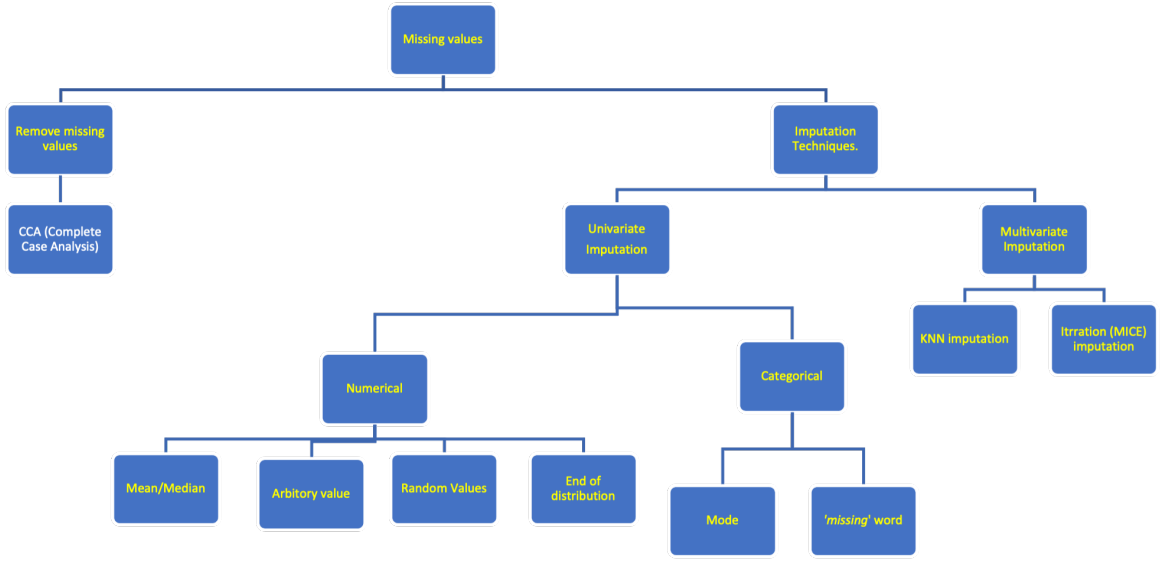

Figure 4.1: Handling Techniques of Missing Values

[9]

To fill these missing values,there are various methods to handle these gaps. Initially, one possible approach was to remove rows with missing values. One common technique for handling missing data is complete case analysis, where we consider only those rows that have complete information and exclude rows with any missing values. However, this method assumes that the missing data is randomly distributed across the dataset. In this case, the missing values were not randomly scattered throughout the data; instead, they appeared in clusters or bunches within a specific section of the dataset. Due to this pattern, complete case analysis would not be suitable for handling the missing values in this dataset. However, this option would lead to a significant loss of valuable data, especially because approximately 70 percentage of the data was missing in these columns. Therefore,not to use this complete case analysis and eliminated it as an option for handling the missing values. As a result, need to shifted towards using imputation techniques, which involve making educated estimates for the missing values based on the available data. [9]

There are two types of imputation techniques one is univariate and other is multivariate.In this case,Univariate imputation technique is eliminated due to the following disadvantages:

- Changes to Distribution Shape: Univariate imputation methods, such as mean, median, or random value imputation, can changes the original distribution shape of the data.
- Generation of Outliers: Univariate imputation may introduce outliers that do not reflect the true characteristics of the data. When using extreme values from the distribution to fill in missing values, it can create artificial outliers that are not representative of the underlying data.
- Altered Covariance/Correlations: Univariate imputation does not consider the relationships between variables. As a result, imputing missing values solely based on the characteristics of individual columns can lead to altered covariance and correlation structures in the data.
- Inaccurate Representations: Univariate imputation treats each column independently, disregarding any potential interactions or dependencies between variables. This approach may not accurately represent the complex relationships that exist in real-world datasets, leading to less reliable predictions and analyses.

Because of these drawbacks of univariate imputation, this is not useful in this case. Next is multivariate imputation methods, such as K-nearest neighbours (KNN) imputation or alternative MICE (Multiple Imputation by Chained Equations). These techniques consider the relationships between variables and provide more accurate estimations for the missing data, preserving the underlying structure of the dataset and improving the reliability of my CO2 emission prediction analysis from vehicle data. [9]

In KNN imputation, we use the similarity between columns to fill in the values in our dataset that are missing. The idea is to use the information that is available to find similar data points and then use the values from those similar data points to estimate the values that are missing. On the other hand, in Multivariate Imputation by Chained Equations (MICE), we utilize information from other columns to predict the missing values. This technique considers the relationships between different variables in the dataset. Since the CO2 rating in our dataset is related to other columns, using MICE can provide more accurate estimates for the missing values compared to KNN imputation. Therefore, using MICE over KNN imputation in this analysis. By using the relationships between variables, MICE help to capture the complexity of the data and generate more reliable and precise predictions for the missing values in the CO2 rating column. [9]

4.2 Descriptive Statistical Analysis

Understanding the summary, data distribution, and their visualisation requires descriptive statistical analysis. In this research process, exploratory data analysis (EDA) becomes an important step. One can easily identify complicated and hidden patterns in the data with the aid of EDA.

4.2.1 Density Curve

For all of the columns, the PDF curves are used to gain additional information from the data. A continuous random variable's data distribution is graphically shown by a PDF curve. It offers insights into the chances of detecting particular data ranges and aids in visualising how data points are distributed throughout various values. We may use the Probability Density Function (PDF) curve, a useful tool, to understand how data in a continuous random variable are distributed. We can identify trends, outliers, and issues with data quality by depicting the structure and properties of the data. This helps us to make better decisions and get more information from the dataset.

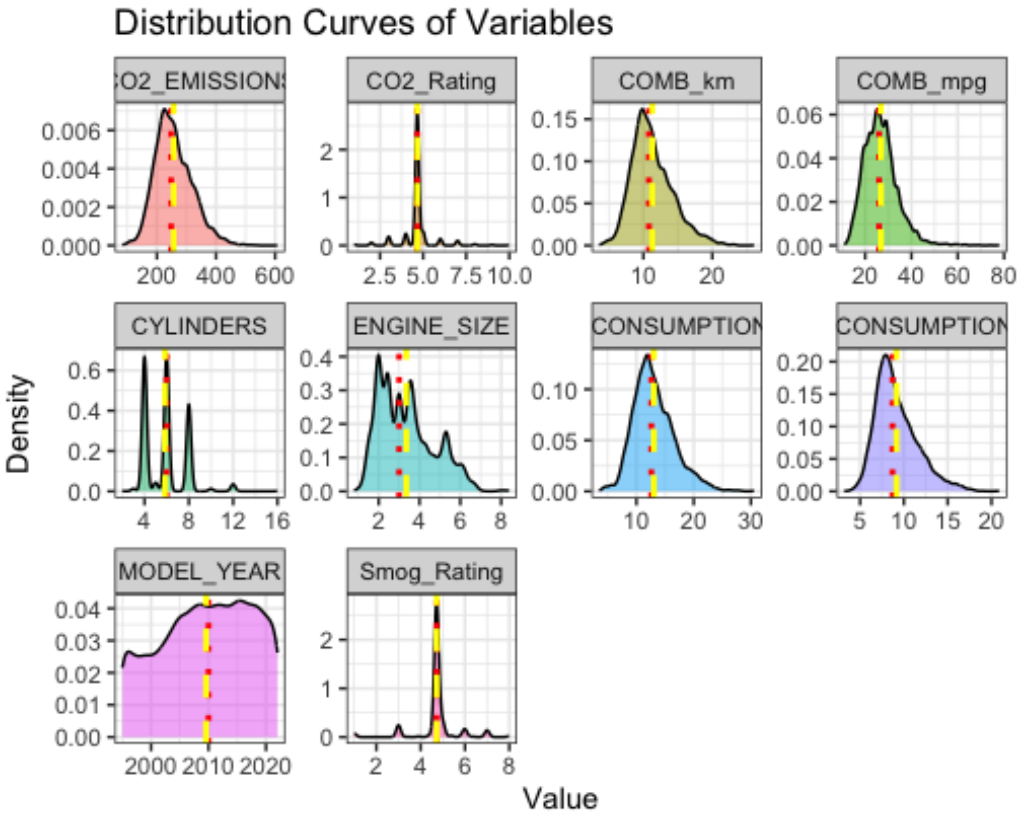

Figure 4.2: Density Curve

There are two types of PDF curves based on the type of random variable:

Table 4.1: Summary of Numeric Data

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
MODEL_YEAR	1995	2004	2010	2010	2016	2022
ENGINE_SIZE	0.800	2.300	3.000	3.355	4.200	8.400
CYLINDERS	2.000	4.000	6.000	5.844	8.000	16.000
FUEL_CONSUMPTION_CITY	3.50	10.60	12.60	13.04	15.10	30.60
FUEL_CONSUMPTION_HWY	3.200	7.500	8.700	9.163	10.500	20.900
COMB_km	3.60	9.20	10.80	11.29	13.00	26.10
COMB_mpg	11.00	22.00	26.00	26.78	31.00	78.00
CO2_EMISSIONS	83.0	212.0	248.0	256.7	297.0	608.0
CO2_Rating	1.00	4.66	4.66	4.66	4.66	10.00
Smog_Rating	1.000	4.720	4.720	4.721	4.720	8.000

- **Discrete Random Variable:** This type of random variable takes on distinct, separate values with gaps between them. The PDF curve for a discrete random variable consists of a series of spikes or bars representing the probabilities of each discrete value.

- **Continuous Random Variable:** This type of random variable can take on any value within a specific range. The PDF curve for a continuous random variable is a smooth, continuous line that estimates the probability density across the entire range.

Summary of all the numerical columns of the dataset is shown below:

MODEL YEAR:

The "MODEL YEAR" column represents a discrete random variable, showing the distribution of vehicle data rows for each year from 1995 to 2020 in Canada. It tells us how many vehicles were recorded in each year. Before the year 2000, the concentration of vehicles in the dataset is relatively low, indicating that there were fewer vehicle models recorded in the earlier years. However, after the year 2000, the concentration of vehicles gradually increases, suggesting that more vehicle models were introduced and recorded in the dataset during this period. The concentration appears to decrease slightly towards the end for the year 2020. This could be due to the data collection process, where the dataset may not yet be fully updated with all the vehicle models for the most recent year.

ENGINE SIZE:

The "ENGINE SIZE" column represents a continuous random variable showing the engine sizes of the vehicles. The engine sizes range from 0.8 to 8.4 liters, with an average engine size of about 3.36 liters. This means that most vehicles in the dataset have engine sizes around 3.36 liters, but there are also smaller and larger engine sizes represented in the data. The distribution of engine sizes is likely to be slightly right skewed. This means that there may be a few vehicles with larger engine sizes that are driving the average engine size higher than the majority of vehicles with smaller engine sizes.

CYLINDERS:

The "CYLINDERS" column represents a discrete random variable indicating the number of cylinders in the vehicles' engines. The number of cylinders ranges from 2 to 16, with an average of approximately 6 cylinders. This shows that most vehicles in the dataset have around 6 cylinders, but there are also vehicles with 2, 4, 8, 12, and 16 cylinders.

FUEL CONSUMPTION CITY:

The "FUEL CONSUMPTION CITY" column is a continuous random variable shows the fuel consumption of vehicles in the city. The fuel consumption ranges from 3.5 to 30.6 liters per 100 kilometers, with an average consumption of about 13.04 liters per 100 kilometers. This means that most vehicles in the dataset consume around 13.04 liters of fuel per 100 kilometers in city driving, but there are also vehicles with lower and higher fuel consumption values. The distribution of fuel consumption in the city is likely to be right-skewed. This means that there may be a few vehicles with higher fuel consumption values that are driving the average fuel consumption higher than the majority of vehicles with lower fuel consumption in the city.

FUEL CONSUMPTION HWY:

The "FUEL CONSUMPTION HWY" column is a continuous random variable shows the fuel consumption of vehicles on the highway. The fuel consumption on the highway ranges from 3.2 to 20.9 liters per 100 kilometers, with an average consumption of about 9.16 liters per 100 kilometers. This means that most vehicles in the dataset consume around 9.16 liters of fuel per 100 kilometers on the highway, but there are also vehicles with lower and higher highway fuel consumption values. The distribution of fuel consumption on the highway is slightly right-skewed. This means that there may be a few vehicles with higher highway fuel consumption values that are driving the average fuel consumption higher than most vehicles with lower fuel consumption on the highway.

COMB km:

The "COMB km" column is a continuous random variable shows the combined fuel consumption of vehicles in kilometers per liter. The combined fuel consumption ranges from 3.6 to 26.1 kilometers per liter, with an average of around 11.29 kilometers per liter. This means that most vehicles in the dataset can travel around 11.29 kilometers with one liter of fuel, but there are also vehicles with lower and higher combined fuel efficiency. The distribution of combined fuel consumption in kilometers per liter may be right-skewed. This means that there may be a few vehicles with higher fuel efficiency that are driving the average fuel efficiency higher than many vehicles with lower fuel efficiency.

COMB mpg:

The "COMB mpg" column is a continuous random variable indicating the combined fuel consumption of vehicles in miles per gallon. The combined fuel consumption ranges from 11.00 to 78.00 miles per gallon, with an average of approximately 26.78 miles per gallon. This means that most vehicles in the dataset can achieve around 26.78 miles per gallon, but there are also vehicles with lower and higher combined fuel efficiency in miles per gallon. The distribution of combined fuel consumption in miles per gallon may be left-skewed. This means that there may be a few vehicles with lower fuel efficiency (higher miles per gallon) that are driving the average fuel efficiency higher than most vehicles with higher fuel efficiency.

CO2 EMISSIONS:

The "CO2 EMISSIONS" column is a continuous random variable representing the CO2 emissions of vehicles in grams per kilometer. The CO2 emissions range from 83.0 to 608.0 grams per kilometer, with an average of about 256.7 grams per kilometer. The distribution of CO2 emissions is right-skewed, meaning there are fewer vehicles with higher emissions. The distribution of CO2 emissions is right-skewed, indicating that there are relatively fewer vehicles with higher emissions compared to many vehicles with lower emissions.

CO2 Rating:

This column is a discrete random variable representing the CO2 ratings of the vehicles.The CO2 ratings range from 1.00 to 10.00. The distribution of CO2 ratings indicates that most vehicles in the dataset have CO2 ratings around 4.66, as shown by the first quartile. CO2 ratings lower than 4.66 suggest lower CO2 emission. On the other hand, ratings higher than 4.66 indicate higher CO2 emissions, reflecting a higher environmental impact .

SMOG Rating:

The "Smog Rating" column is a discrete random variable representing the smog ratings of the vehicles. Smog ratings measure the environmental impact of vehicles in terms of smog formation. The smog ratings range from 1.00 to 8.00. The distribution of smog ratings shows that most vehicles in the dataset have smog ratings around 4.72, as indicated by the first quartile. Smog ratings lower than 4.72 suggest better environmental performance in terms of smog formation, while ratings higher than 4.72 indicate a higher environmental impact related to smog formation.

4.2.2 Collinearity Representation

Correlation refers to the link connecting two variables, acting as a measuring tool to understand the statistical relationship between them. To analyse this relationship,statistical metric known as the correlation coefficient is used, which operates within a scale ranging from -1 to 1.

When this coefficient falls between 0 and 1, it signifies a positive relationship between the variables. A correlation coefficient between -1 and 0 indicates a negative relationship. For in-depth analysis, we can utilize this correlation concept specifically between two columns of data. In this study,focus on relationships with an intensity of +/- 0.5 or higher. This threshold is chosen because it points us to strong correlations, which play a crucial role in our efforts to build regression models. This correlation represents the pairwise relationships between different columns (variables) in my dataset. Each cell in the matrix displays the correlation coefficient, which quantifies the strength and direction of the linear relationship between two variables.

Strong Positive Correlations

1. "FUEL CONSUMPTION CITY" and "FUEL CONSUMPTION HWY" have a correlation coefficient of approximately 0.94. âFUEL CONSUMPTION CITY" and "FUEL CONSUMPTION HWY." These variables represent the fuel consumption of vehicles under distinct conditions â city driving and highway driving, respectively. This strong correlation can be attributed to the fact that both variables essentially capture the same underlying parameterr: fuel efficiency of the vehicle. The only difference is in the context of measurement â one takes on city scenarios, while the other on highway conditions.
2. "FUEL CONSUMPTION HWY" and "COMB km" share a correlation of about 0.98, indicating a strong positive correlation between highway fuel consumption and combined fuel consumption. It’s kind of like they’re talking about the same thing â how efficiently the car uses fuel.

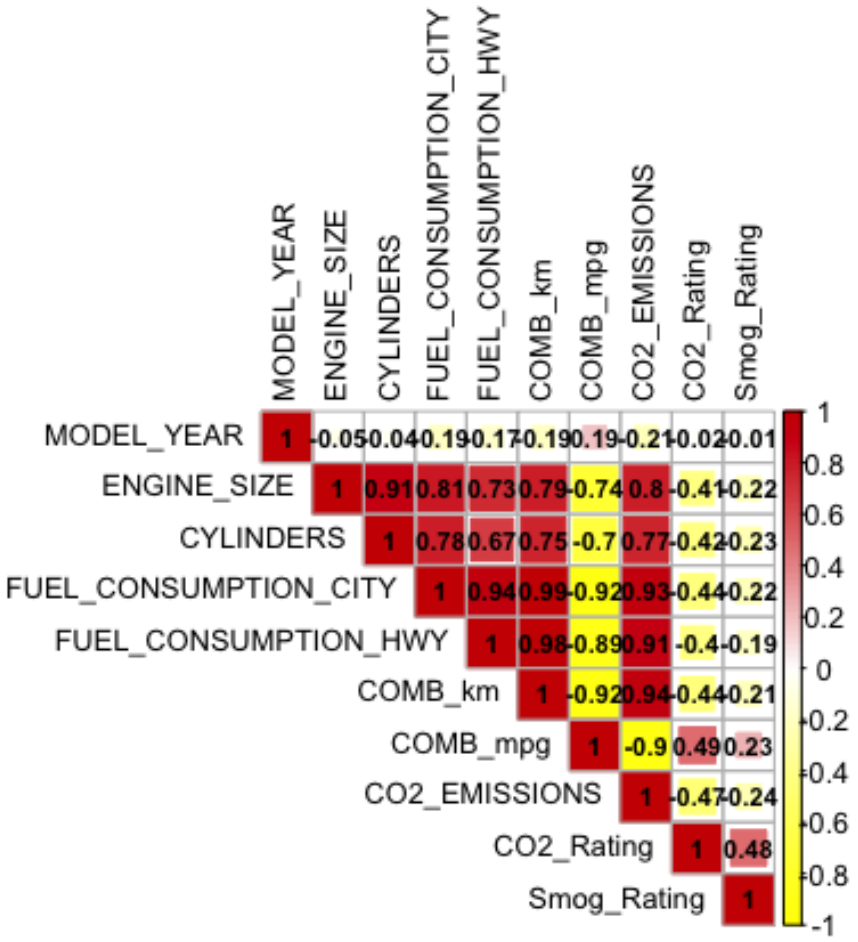

Figure 4.3: Correlation Matrix

3. "ENGINE SIZE" and "CYLINDERS" have a correlation coefficient of approximately 0.91, indicating a strong positive relationship. As engine size increases, the number of cylinders tends to increase, and vice versa. And the other way around, if the engine has more cylinders, it's often because the engine size is bigger.

4. "FUEL CONSUMPTION CITY" and "CO2 EMISSIONS" have a high positive correlation of around 0.93. Higher city fuel consumption is associated with higher carbon dioxide emissions. When a car uses more fuel in the city ("FUEL CONSUMPTION CITY"), it tends to release more carbon dioxide into the air ("CO2 EMISSIONS").

Strong Negative Correlation

"COMB mpg" and "CO2 Rating" exhibit a correlation coefficient of about -0.47. This indicates a moderate negative correlation between fuel efficiency (miles per gallon) and the CO2 emissions rating. As fuel efficiency improves, the CO2 emissions rating tends to decrease.

4.2.3 Combined Scatter Plot

In the third part of the methodology, combined scatter plots is used to gain insights into the multi-collinearity structure among the continuous numerical columns in this dataset. Multi-collinearity is the relation between predictor variables, can impact the stability and interpretability of regression coefficients. By visualizing the relationships between these variables through scatter plots, aimed to assess the degree of multi-collinearity and understand how it might influence in predictive model.

The combined scatter plots allowed to observe how pairs of continuous numerical variables move in relation to each other. By detecting multicollinearity through scatter plots, it could helps for the feature selection and model fitting.

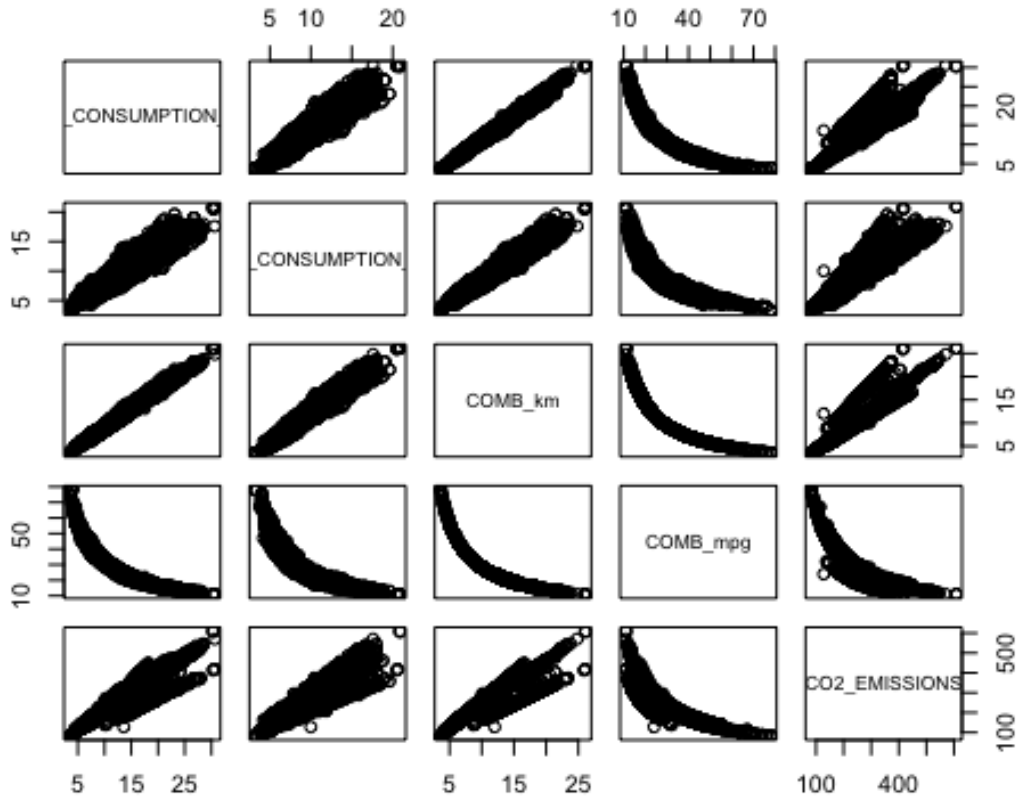

Figure 4.4: Scatter Plot

This step helps to focused on a specific group of things we can measure: "FUEL CONSUMPTION CITY," "FUEL CONSUMPTION HWY," "COMB km," "COMB mpg," and "CO2 EMISSIONS." These things helps because they might affect how much CO2 a vehicle releases.

An an interesting discovery while studying this measurements. "COMB mpg," gives us an idea

of how far a car can travel using a single gallon of fuel. When "COMB mpg" increased, the values of the other measurements went in the opposite direction â they decreased. However, looking at the other measurements like "FUEL CONSUMPTION CITY," "FUEL CONSUMPTION HWY," and "CO2 EMISSIONS," they showed a different behaviour. These measurements changed together, like points on a straight line.

This discovery is important for predicting CO2 emissions. When a car is more efficient (higher "COMB mpg"), it tends to produce less CO2. On the other hand, when "FUEL CONSUMPTION CITY," "FUEL CONSUMPTION HWY," and "CO2 EMISSIONS" are considered together, they show a consistent and predictable link. Also, it's good to know that the things are related to each other. This relationship might affect how well our models predict CO2 emissions.

4.2.4 Line graph:(CO2 EMISSIONS VS year)

"CO2 EMISSIONS" is the predictable variable. To effectively build and validate predictive model, it's essential to understand the historical context and trends of CO2 emissions over the years. By plotting the total emission data, one can get a valuable insight as how emissions have evolved and fluctuated in the past. A line graph is a particularly suitable choice for representing the total emission data over time. As we can see mostly for the all-time series analysis, generally âline graphâ has used. The best example is Share Market. This type of graph allows you to easily identify patterns, fluctuations, and overall tendencies in the dataset.

The line graph shows the relationship between years and total CO2 emissions. The x-axis represents the years, ranging from 1995 to 2022, while the y-axis represents the total emissions.

1. Early Fluctuations (1995 to 1998):

- In the mid 1990s, a moderate amount of CO2 emissions, around 258,130 units in 1995.

- Over the next few years, from 1996 to 1998, there was a slight decrease in emissions. We were producing a bit less CO2 during this period.

2. Slight Uptick (1999):

- In 1999, there was a small increase in emissions, reaching about 203,347 units. It's like emissions took a tiny step up.

3. Significant Decrease (Early 2000s):

- From around 2000 to 2002, there was a sharp decline in emissions. The lowest point was in 2000, with only about 165,592 units of CO2 emissions.

- During these years, we managed to reduce our emissions significantly compared to the previous years.

4. Steady Increase (Mid 2000s to Late 2000s):

- Starting from 2003, the emissions began to rise again each year.

- This increase continued until around 2008, reaching the highest point at 273,824 units in that year.

- During this period, our emissions were on the rise, and they climbed to their peak.

5. Stabilization (Late 2000s to 2022):

- After the peak in 2008, the graph shows a levelling off. Emissions didn't rise dramatically like before, but they also didn't drop back to the very low levels of the early 2000s.

- The graph suggests that emissions have become more stable and consistent during these years.

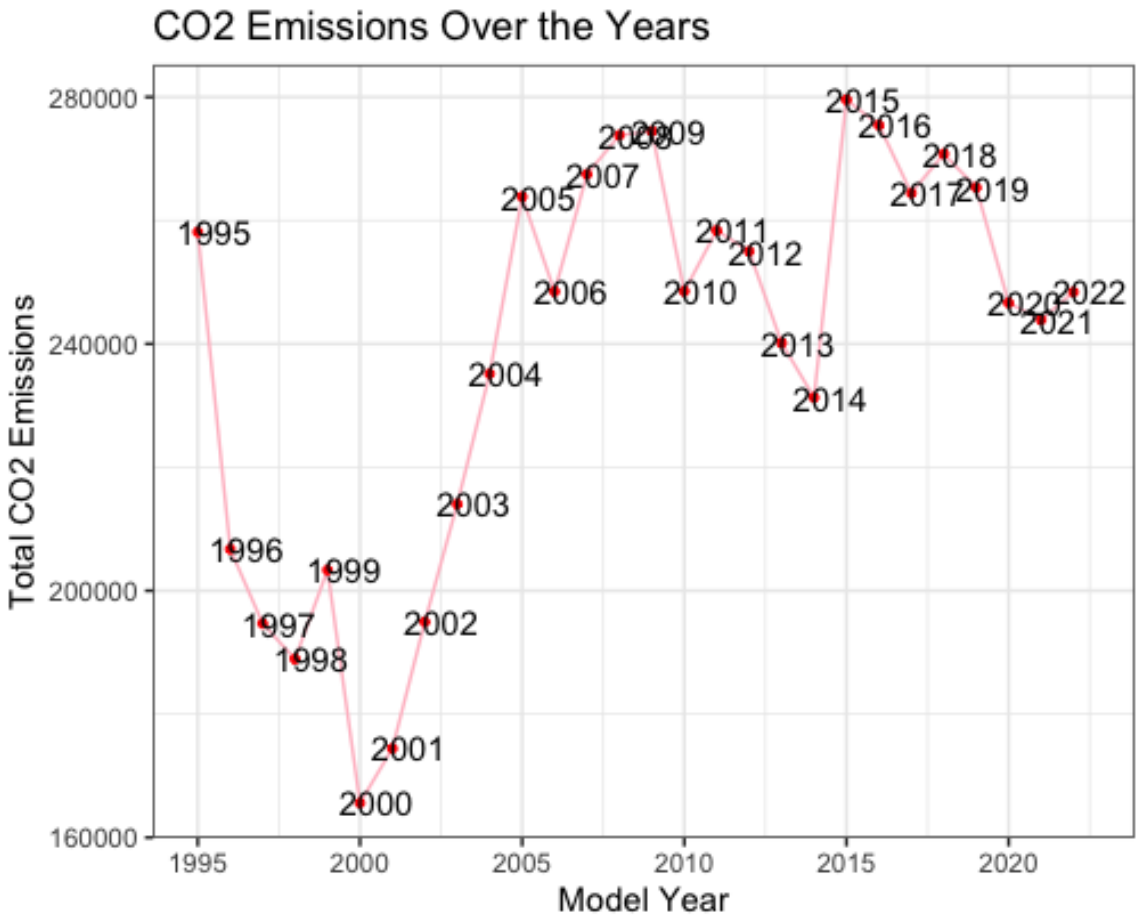

Figure 4.5: Time Series Year

Over the years, the line graph indicates fluctuating patterns, a minor dip in the mid 1990s, a modest rise in 1999, a notable early 2000s emission drop, followed by steady growth until around 2009, and eventual stabilization in emissions from the late 2000s onward.

4.2.5 Comparative line graph:

This comparative line graph is a powerful tool for understanding how the fuel consumption of vehicles differs between city and highway driving across various years.

The upper line representing "FUEL CONSUMPTION CITY" means the amount of fuel that vehicles typically consume when navigating through city areas. It reflects the efficiency of vehicles in stop-and-go traffic, where frequent accelerations and decelerations occur.

The other lower line, corresponding to "FUEL CONSUMPTION HWY," illustrates the fuel consumption of vehicles during highway journeys, when they maintain a relatively steady speed over longer distances.

1. In this data,the line representing city fuel consumption is higher than the line for highway fuel consumption. This suggests that vehicles use more fuel in city settings compared to highway driving.

2. Both the city and highway lines show a very similar trend. They both decrease from 1995 to 2014 and then start to increase again. This pattern of decreasing and then reversing is quite consistent in both cases.

This graph provides a comprehensive visual representation of how vehicles utilize fuel in distinct driving environments.

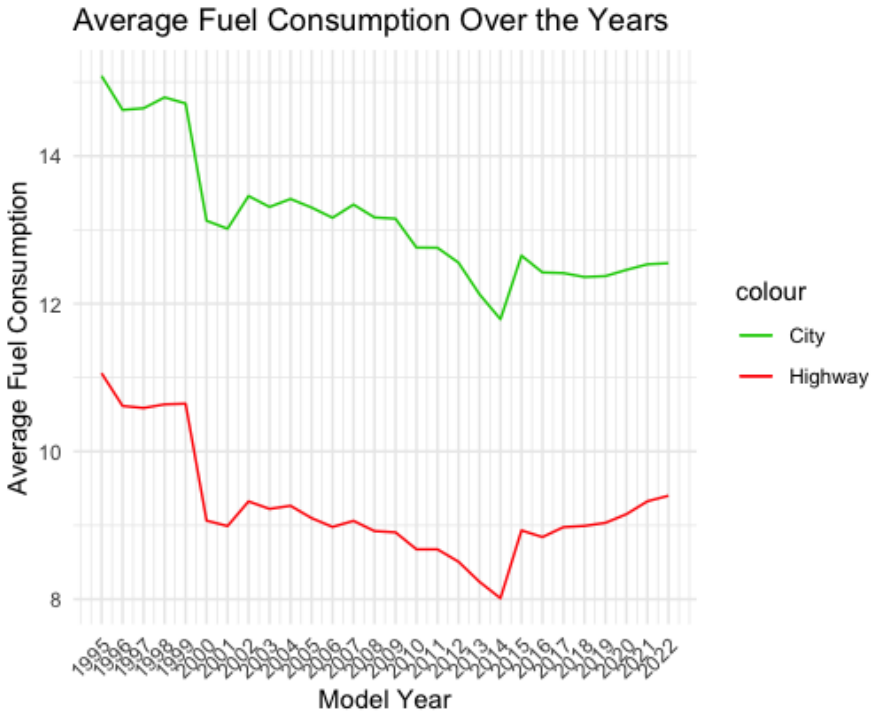

Figure 4.6: Comparative Line Graph

4.2.6 Comparison of Fuel Type:

For this project, understanding the distribution of CO2 emissions by different fuel types is crucial for several reasons. Different fuel types have different levels of CO2 emissions, which directly contribute to environmental pollution and climate change. Analysing these distributions helps to identify which fuel types have a more favourable environmental footprint

A violin plot is used to visualize the distribution of CO2 emissions across different fuel types. The violin plot combines a box plot and a kernel density plot. Each "violin" represents a different fuel type. The width of the violin shows the density of data at different emission levels, while the central "box" represents the interquartile range (IQR) of the data. The "tails" extending from the box show the entire range of data distribution.

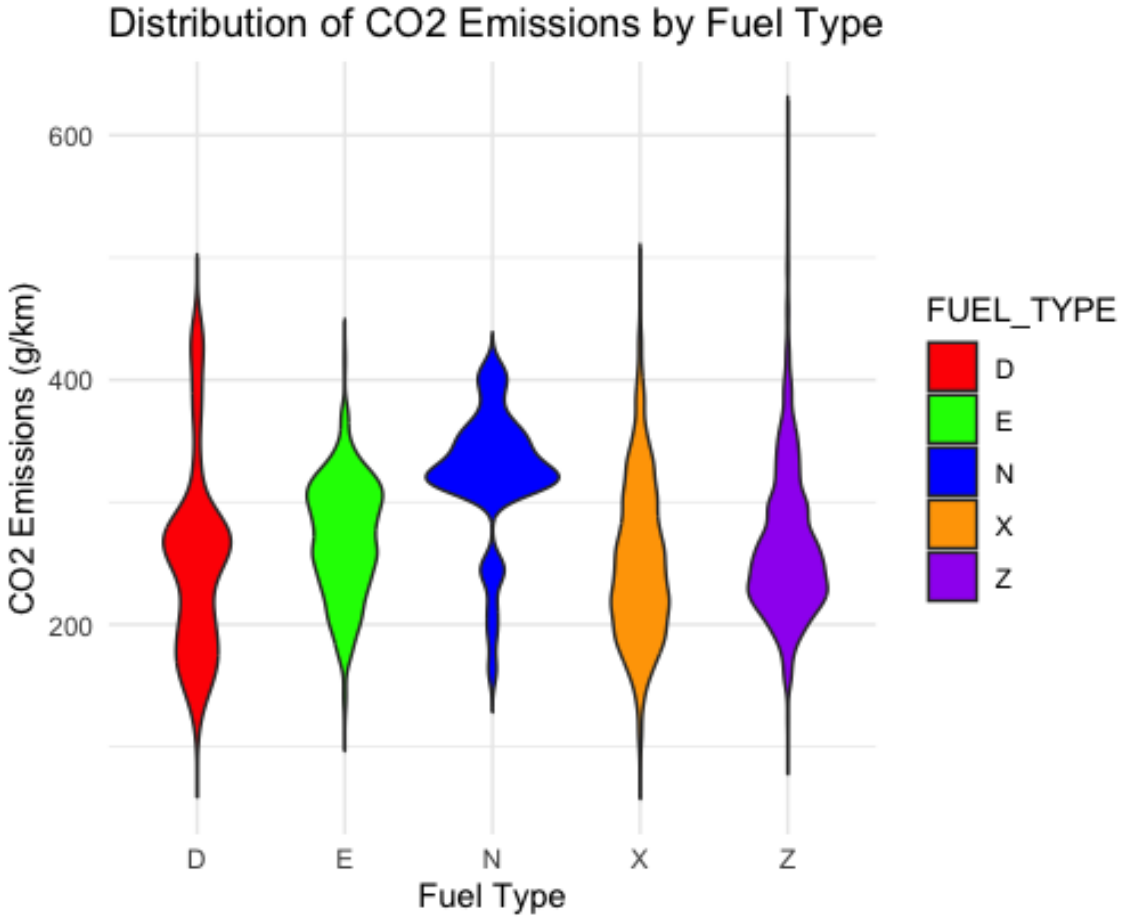

Figure 4.7: Fuel Type

Fuel Type D

- The emissions for Fuel Type D are most common between 0 to 500 grams per kilometer (g/km).

- Many vehicles in this category have emissions clustered around 300 g/km, with some slightly below.

- The amount of emissions can vary quite a bit, meaning some vehicles are more efficient while others use more fuel. On average, it might be around 300 g/km.

Fuel Type E:

- For Fuel Type E, emissions are often found between 0 to 400 g/km.

- Most vehicles have emissions close to 300 g/km, and there are fewer that go lower or higher.

- Overall, the emissions tend to be more consistent, with vehicles showing similar efficiency. The average could be about 300 g/km.

Fuel Type N:

- Fuel Type N is like E, with emissions mostly above 300 g/km, and not much below that.

- The amount of emissions doesn't vary a lot, similar to Fuel Type E.

- On average, these vehicles might also be around 300 g/km.

Fuel Type X:

- Vehicles in Fuel Type X emit the most between 0 to 500 g/km, but many are closer to 200 g/km.

- As emissions go higher than 200 g/km, there are fewer vehicles with those levels.

- This means some vehicles are more efficient (closer to 200 g/km), while others use more fuel. Average emissions could be around 350 g/km.

Fuel Type Z:

- Fuel Type Z has a big range of emissions, mostly above 200 g/km and below 300 g/km.

- There are a few vehicles with very high emissions, going beyond 600 g/km.

- This means there's a wide variety in how much fuel different vehicles use, from typical to occasional outliers. The average might be between 250 to 300 g/km.

Fuel Type D show a broad range, shows a mix of vehicle efficiencies. In contrast, Fuel Types E and N exhibit more consistent emissions around 300 g/km, indicating relatively uniform efficiency levels within these categories. Fuel Type E displays emissions concentrated around 300 g/km, underscoring a dependable and foreseeable efficiency pattern.

4.2.7 Comparative Boxplot for vehicle class:

Analysing CO2 emissions across different vehicle classes can provide valuable insights for my project. By comparing CO2 emissions across different vehicle classes, you can assess whether the vehicle class variable (categorical) is a significant predictor of CO2 emissions. If certain vehicle classes consistently have higher emissions than others, it indicates that vehicle class is an important factor to consider in your predictive model. Also, by comparing CO2 emissions among various vehicle classes, you can identify trends, outliers, and potential areas for improvement. A boxplot is an effective choice for comparing CO2 emissions across vehicle classes due to its ability to provide a visual summary of key statistics and distribution characteristics. Boxplots display the median, quartiles, potential outliers, and overall spread of data, making it easy to compare the central tendency, variability, and skewness of CO2 emissions in different vehicle classes.

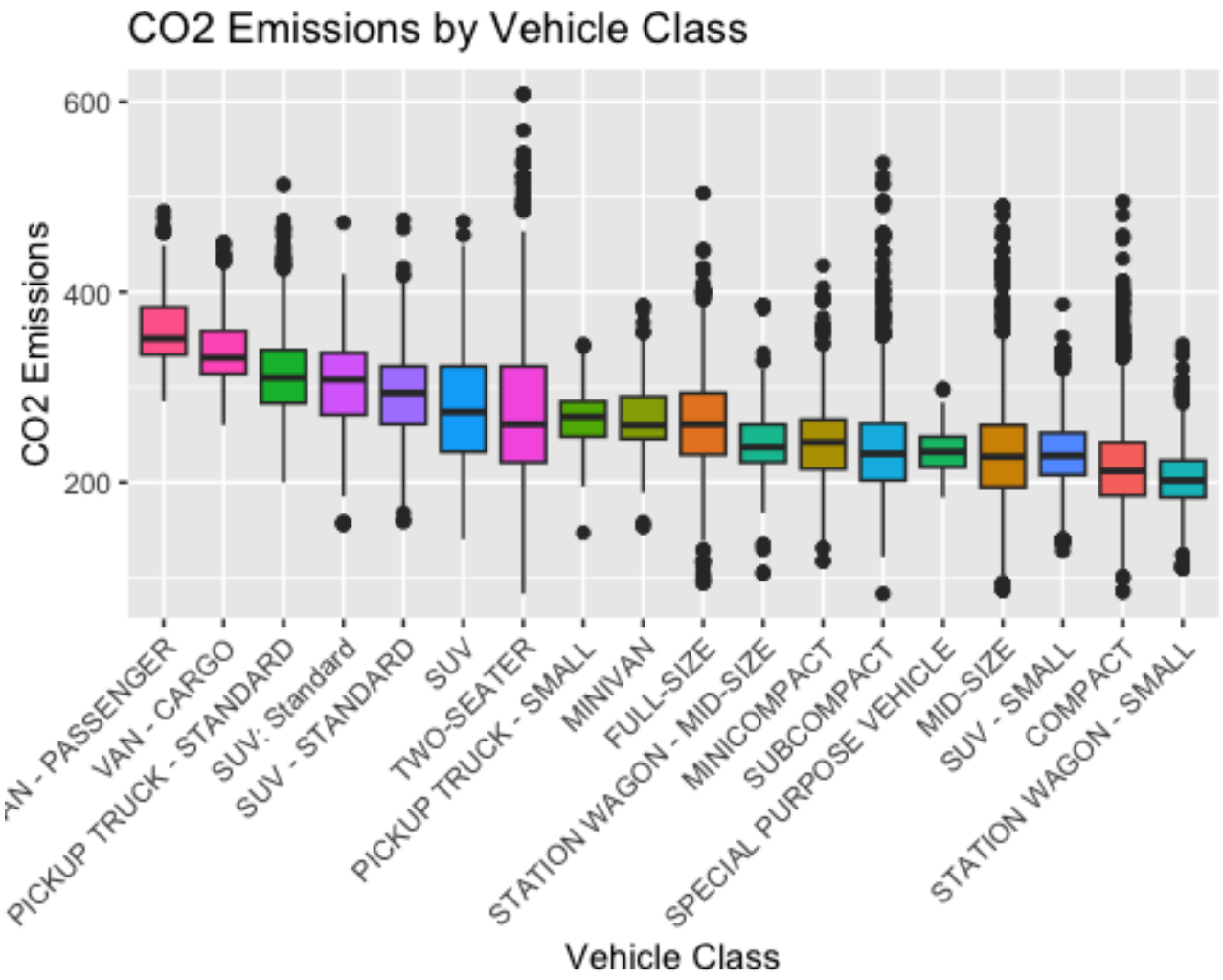

Figure 4.8: BoxPlot For Vechical Class

Some are the key insights from graph:

1. Larger vehicles like VAN - PASSENGER and VAN - CARGO emit more CO2 due to their size and weight. Smaller vehicles like TWO-SEATER and SUBCOMPACT have moderate emissions influenced by size and engine efficiency.

2. Vans used for businesses (VAN - PASSENGER, VAN - CARGO) have higher emissions due to frequent use.

3. SUVs designed for personal use emit medium to high emissions, striking a balance between comfort and emissions.

4. Pickup trucks (PICKUP TRUCK - STANDARD) emit moderate emissions as they're used for light work.

5. Minivans (MINIVAN) for families emit moderate emissions, tied to passenger comfort and space.

6. Different SUV types (SUV: Standard, SUV - STANDARD, SUV - SMALL) exhibit varying emissions due to differences in size, engine, and technology.

7. Compact cars (FULL-SIZE, MID-SIZE, MINICOMPACT, COMPACT, SUBCOMPACT) emit moderate emissions, with smaller ones being more fuel-efficient.

8. Wagons (STATION WAGON - SMALL, STATION WAGON - MID-SIZE) with extra space emit low to medium emissions due to their efficient design.

9. SPECIAL PURPOSE VEHICLE, designed for specific uses, shows moderate emissions. This reflects the diversity of purposes, suggesting that specific design features for various activities still maintain a balance in emissions.

10. TWO-SEATER, often associated with sporty performance, has moderate emissions. This suggests a trade-off between performance-oriented features and emissions.

The result from this analysis shows that vehicle size, purpose, and design all play roles in determining CO2 emissions. While larger vehicles and those used for commercial purposes tend to have higher emissions, there are variations within each class due to factors such as engine efficiency, technology, and design. Consumers looking to minimize emissions could consider smaller and more fuel-efficient classes

4.2.8 Comparative Boxplot for transmission class:

Comparing CO2 emissions across different types of transmissions is essential for understanding the environmental impact of different vehicle technologies. It helps assess how different transmission systems contribute to emissions and informs decisions aimed at reducing the carbon footprint of vehicles.

The data includes various types of transmissions:

- AV1, AV10, A10, AS10, AM9, AM8, AS9, A9, AM5, AV6, AV8, AV7: Automatic transmissions with varying gears

- M7, M4, M6, M5: Manual transmissions with different gear options

- AS8, AS7, AS5, AS6, AS4: Semi-automatic transmissions

- A3, A5, A8, A9, A4, A6, A7: Automatic transmissions

Manual transmissions (M7, M4, M6, M5) generally show moderately lower emissions compared to many automatic transmissions. This could indicate that manual transmissions offer better control over gear shifting, potentially leading to more efficient fuel consumption.

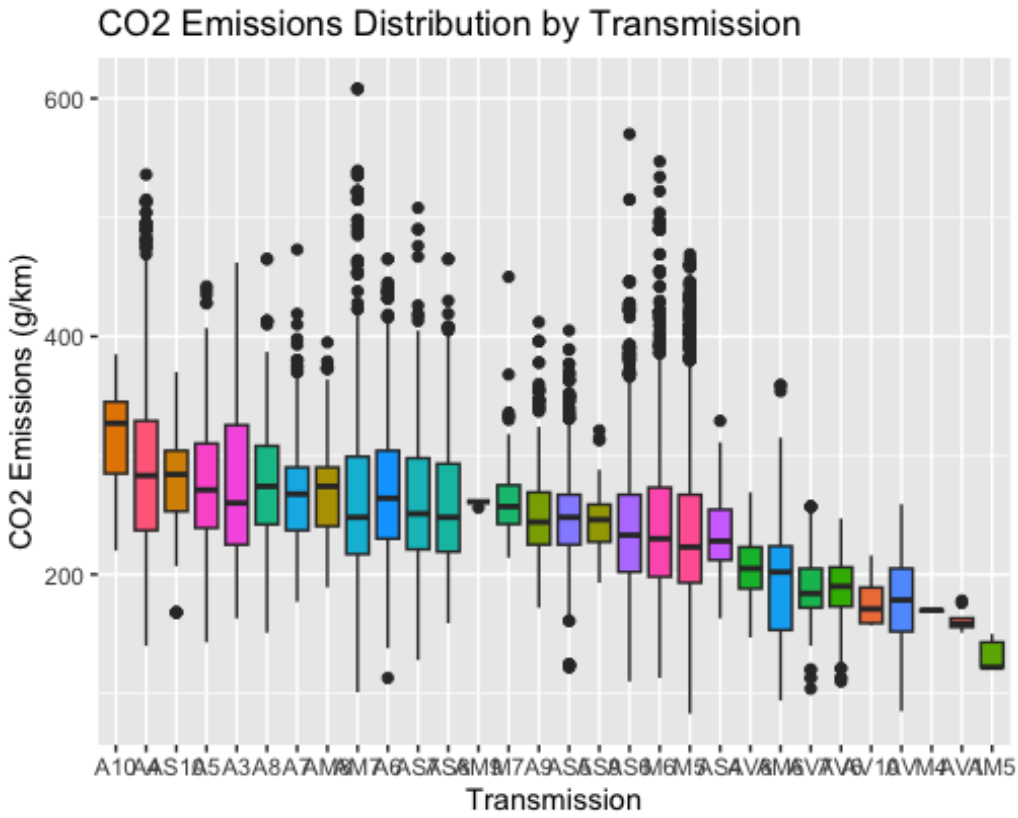

Figure 4.9: Boxplot Transmission

Semi-automatic transmissions (AS8, AS7, AS5, AS6, AS4) have a range of emissions. This suggests that while semi-automatic systems can be more efficient in some cases, they still have variations based on other factors.

Among automatic transmissions, the emissions vary across different gear options (e.g., AV1, AV10, A10). This highlights the importance of specific automatic transmission designs and technologies in influencing emissions.

combined graphs After categorizing all transmissions into three types: automatic, manual, and semi-automatic, need to do a comparison of fuel consumption and CO2 emissions for these categories.

- Auto transmission vehicles have the highest values for both CO2 emissions and fuel consumption. - Semi-auto transmission vehicles come next with intermediate values. - Manual transmission vehicles have the lowest values for both CO2 emissions and fuel consumption.

This analysis highlights that auto transmission vehicles consume more fuel and emit more CO2 compared to the other two types. Conversely, manual transmission vehicles demonstrate better fuel efficiency and lower CO2 emissions. These findings emphasize the potential environmental benefits of using manual transmission vehicles in terms of reduced fuel consumption and minimized carbon emissions.

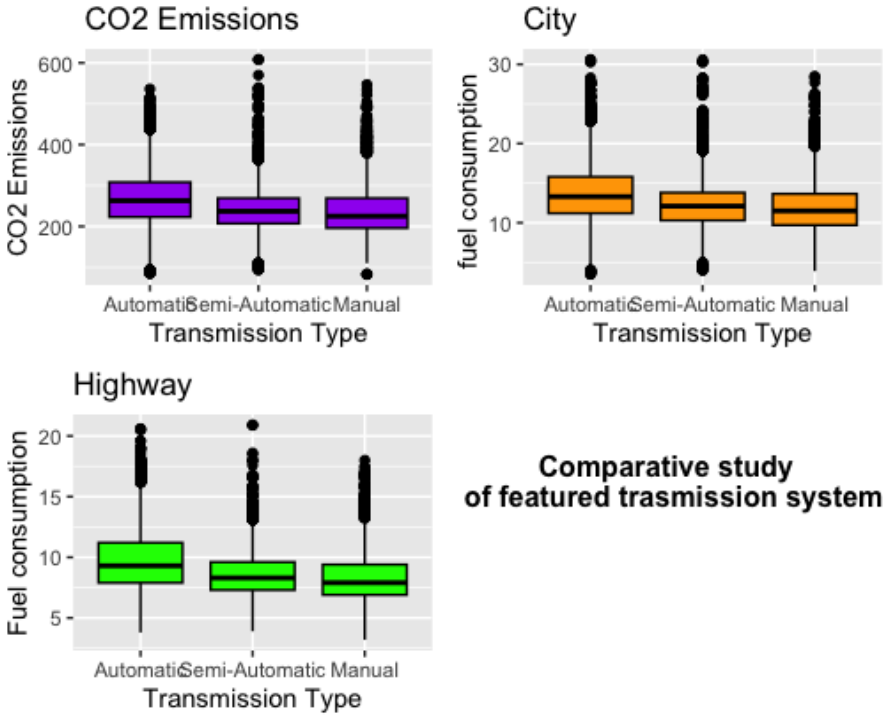

Figure 4.10: Boxplot Combined

4.2.9 Top performance Vehicle Types:

The bar chart represents the average CO2 emissions of top-performing vehicles from different companies, categorized by their respective vehicle classes.

In the chart:

Each bar corresponds to a company's top-performing vehicle(s) in terms of CO2 emissions. The x-axis displays the names of companies. The y-axis represents the average CO2 emissions. Different colours represent various vehicle classes within each company.

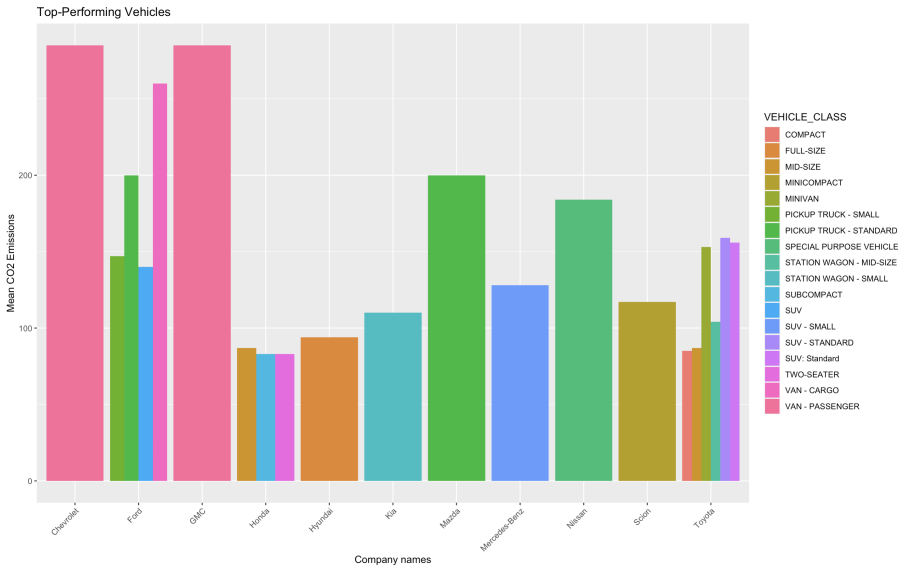

Figure 4.11: Top Performance

Common Vehicle Classes:

MID-SIZE and COMPACT classes are present among multiple companies. SUV: Standard and SUV - STANDARD classes are seen in Toyota and Scion. STATION WAGON - SMALL is common among Kia, Hyundai, and Scion.

Each bar on the graph represents a different company, such as Honda, Chevrolet, and Toyota. The varying heights of the bars shows the distinct average CO2 emissions of the top-performing vehicles for each company. Within each company's bar, the different colours represent specific vehicle classes: For example, within Honda's bar, we can see a colourful array of classes including SUBCOMPACT, TWO-SEATER, and PICKUP TRUCK - STANDARD.

Examining the tallest sections within each bar, shows the vehicle classes contributing to higher average CO2 emissions for each company.

For Example, the taller sections within Honda's bar are linked to vehicle classes like TWO-SEATER, which signifies that this class contributes more to emissions within Honda's lineup. On the other

hand, the shorter sections, like SUBCOMPACT, reflect the classes that excel in terms of lower CO2 emissions.

Observing common vehicle classes across multiple companies, such as SUV: Standard and MID-SIZE, provides valuable insights into segments that resonate across the industry. Conversely, the unique vehicle classes within a specific company's bar, such as PICKUP TRUCK - SMALL for Ford, unveil their distinctive approach and specialization within certain vehicle categories.

This also shows the analysis of specific vehicle classes that significantly contribute to a company's environmentally friendly image. For instance, Honda's TWO-SEATER class emerges as a "green star," showcasing their success in producing vehicles that lead to lower CO2 emissions.

The graph doesn't just highlight successes but also indicates potential areas of improvement. By examining vehicle classes associated with higher CO2 emissions, like SUV for Ford or MINIVAN for Toyota, companies can pinpoint segments where enhancing environmental impact could be a priority.

4.3 Inferantial Statical analysis

4.3.1 Hypothesis test

After completion of Exploratory Data Analysis (EDA) and gained a preliminary understanding of this data,next step need to go deeper and find out additional insights using hypothesis testing. EDA allows to uncover initial patterns and trends within the data. Hypothesis testing introduces a systematic and statistical approach to validate, refine, and discover hidden relationships or patterns that might have been overlooked during the initial exploration. [10] So, after using EDA for this data, usage of hypothesis testing to double-check what we found and maybe find some more secret things by this test. Hypothesis testing is like a tool that helps me be super sure about discoveries and maybe find even more deep analysis.

Hypothesis testing involves several steps to help us make informed decisions based on data. Here's a simple breakdown of these steps:

- Null Hypothesis (H0):

  - This is like starting with a default assumption that there's no important connection between the things we're looking at.

  - We use the null hypothesis to say,"nothing new is happening.â

  - It's the baseline we compare against, kind of like saying "let's see if anything interesting stands out."

- Alternative Hypothesis (Ha):

  - This one is the opposite of the null hypothesis.

  - It's like saying "there's actually something going on here!"

  - We use the alternative hypothesis to suggest that there's a meaningful relationship between the things we're studying.

The whole idea of hypothesis testing is to gather evidence, usually in the form of data, to see if we should stick with the null hypothesis or switch to the alternative hypothesis. We're trying to figure out if the things we see in our data are real or if they could have just happened by chance. Once we do some math and stats, we'll either reject the null hypothesis and go with the alternative, or we'll keep the null hypothesis if the evidence isn't strong enough for that factor. In simpler terms, the test helps you go beyond just observing a difference and provides you with a solid statistical foundation to confidently. [10]

The Welch Two Sample t-test:

This test is particularly useful when the assumptions of equal variances and normal distribution might not hold true for your data. It considers the potentially unequal variability between the two groups (city and highway fuel consumption) and helps you decide if the observed difference in means is statistically significant. [10]

The Welch Two Sample t-test is a statistical method used to compare the means of two independent groups when the assumptions of equal variances and normal distribution are not met. It's a variation of the standard t-test that adjusts for unequal variances between the groups.

The null and alternative hypotheses for the Welch Two Sample t-test to compare the average fuel consumption in the city (FUEL CONSUMPTION CITY) with the average fuel consumption on the highway (FUEL CONSUMPTION HWY):

Null Hypothesis (H0): The average fuel consumption in the city is equal to the average fuel consumption on the highway.

$$\mu city = \mu highway \tag{4.3.1}$$

Alternative Hypothesis (Ha): The average fuel consumption in the city is not equal to the average fuel consumption on the highway.

$$\mu city \neq \mu highway \tag{4.3.2}$$

Table 4.2: Summary of Welch Two Sample t-test Results

Test:	Welch Two Sample t-test
Data:	df $ FUEL_CONSUMPTION_CITY, df $FUEL_CONSUMPTION_HWY
t-value:	145.69
Degrees of Freedom:	45693
p-value:	$< 2.2 \times 10^{-16}$
Alternative Hypothesis:	True difference in means is not equal to 0
95% Confidence Interval:	3.821207 to 3.925425
Sample Estimates:	
Mean of x:	13.036092
Mean of y:	9.162776

Set significance level is 0.05, which means I'm willing to accept a 5 percentage chance of making a Type I error, that is, wrongly rejecting a true null hypothesis.

The p-value obtained is much smaller than 0.05 (p < 2.2e-16), which suggests strong evidence against the null hypothesis. This means that the difference in fuel consumption between city and highway driving is highly unlikely to have occurred randomly. The t-value of 145.69 represents how much the average fuel consumption in the city differs from that on the highway.

The 95 percentage confidence interval (3.821207 to 3.925425) gives a range within which one can be quite confident that the true difference in average fuel consumption between city and highway driving lies. It shows that, on average, fuel consumption in the city is between 3.82 and 3.93 gallons higher compared to highway driving.

Considering the sample estimates, the mean fuel consumption in the city is about 13.04 gallons, while on the highway, it's approximately 9.16 gallons.

Using the Welch Two Sample t-test provides strong evidence that there is indeed a significant difference in fuel consumption between city and highway driving. This supports what we observed during Exploratory Data Analysis and gives me a solid statistical basis for confirming this difference.

One-Way ANNOVA:

To validate above findings from the Exploratory Data Analysis (EDA) regarding the relationship between fuel type and CO2 emissions. A One-Way ANOVA test is used for this.

A statistical technique known as One-Way Analysis of Variance (ANOVA) is used to go deeper into the relationship between different fuel types and their impact on CO2 emissions. ANOVA is a powerful tool that allows for the comparison of means among multiple groups, helping me determine if there are significant differences in average CO2 emissions across various fuel types namely, E, D, N, X, and Z. [10]

Null Hypothesis (H0): The mean CO2 emissions are equal across all fuel types.

$$\mu E = \mu D = \mu N = \mu X = \mu Z \tag{4.3.3}$$

Table 4.3: One-way ANOVA on "CO2_EMISSIONS" among different fuel types

Source of Variation	**Df**	**Sum Sq**	**Mean Sq**	**F value**	**Pr(>F)**
FUEL_TYPE	4	1542380	385595	98.41	$< 2 \times 10^{-16}$ ***
Residuals	26067	102139312	3918		

Alternative Hypothesis (Ha): The mean CO2 emissions are not equal across at least one pair of fuel types. At least one Î¼i is different from the others, where i represents the fuel type (E, D, N, X, or Z).

The analysis revealed a strong connection between CO2 emissions and the type of fuel used, with a p-value of < 2.2e-16. This small p-value indicates that the observed relationship is highly unlikely to have occurred by chance alone. Therefore, we rejected the null hypothesis. The calculated F-statistic of 98.41 was notably higher than what we would expect by chance. This suggests that the variations in CO2 emissions among different fuel types are significant and not merely random fluctuations.

The Tukey HSD test:

After the initial One-Way ANOVA test that looked at how different fuel types affect CO2 emissions, need to go deeper with the Tukey Honestly Significant Difference (HSD) test. [10]

The reason behind using the Tukey HSD test is we wanted to go beyond the big picture from the ANOVA test. While the ANOVA test told us that there were overall differences in emissions, the Tukey HSD test zoomed in to tell us exactly which pairs of fuel types were different and how much. It helped us pinpoint the specific fuel types that stood out and contributed the most to the differences we observed.

- Null Hypothesis (H0):

  There is no significant difference in the average CO2 emissions between any pairs of fuel types.

- Alternative Hypothesis (Ha):

  There are significant differences in the average CO2 emissions between at least one pair of fuel types.

Table 4.4: Tukey Multiple Comparisons of Means

Comparison	**Difference**	**Lower CI**	**Upper CI**	**p adj**
E-D	19.076245	9.131026	29.0214634	0.0000017
N-D	70.163319	41.538413	98.7882254	0.0000000
X-D	-1.267671	-9.859895	7.3245532	0.9944869
Z-D	12.618350	3.975253	21.2614462	0.0006518
N-E	51.087075	23.254180	78.9199695	0.0000055
X-E	-20.343916	-25.739387	-14.9484449	0.0000000
Z-E	-6.457895	-11.934017	-0.9817735	0.0113459
X-N	-71.430990	-98.809613	-44.0523678	0.0000000
Z-N	-57.544970	-84.939600	-30.1503392	0.0000001
Z-X	13.886021	11.670880	16.1011617	0.0000000

The Tukey HSD test results for the different fuel types and their impact on CO2 emissions:

1. Gasoline (E) vs. Diesel (D): There is a significant difference in average CO2 emissions between vehicles using gasoline and diesel ($p < 0.0000017$). On average, vehicles running on gasoline emit about 19 units more CO2 than those using diesel.

2. Natural Gas (N) vs. Diesel (D): The difference in average CO2 emissions between vehicles powered by natural gas and diesel is also significant ($p < 0.0000000$). Vehicles using natural gas tend to emit around 70 units more CO2 on average compared to those using diesel.

3. Diesel (D) vs. Biodiesel (X): There is no significant difference in CO2 emissions between vehicles using diesel and those using biodiesel ($p > 0.9944869$). This suggests that the CO2 emissions for these two fuel types are likely to be similar.

4. Diesel (D) vs. Ethanol (Z): The difference in average CO2 emissions between vehicles fuelled by diesel and those using ethanol is significant ($p < 0.0006518$). On average, ethanol-fuelled vehicles emit approximately 12 units more CO2 than diesel-fuelled ones.

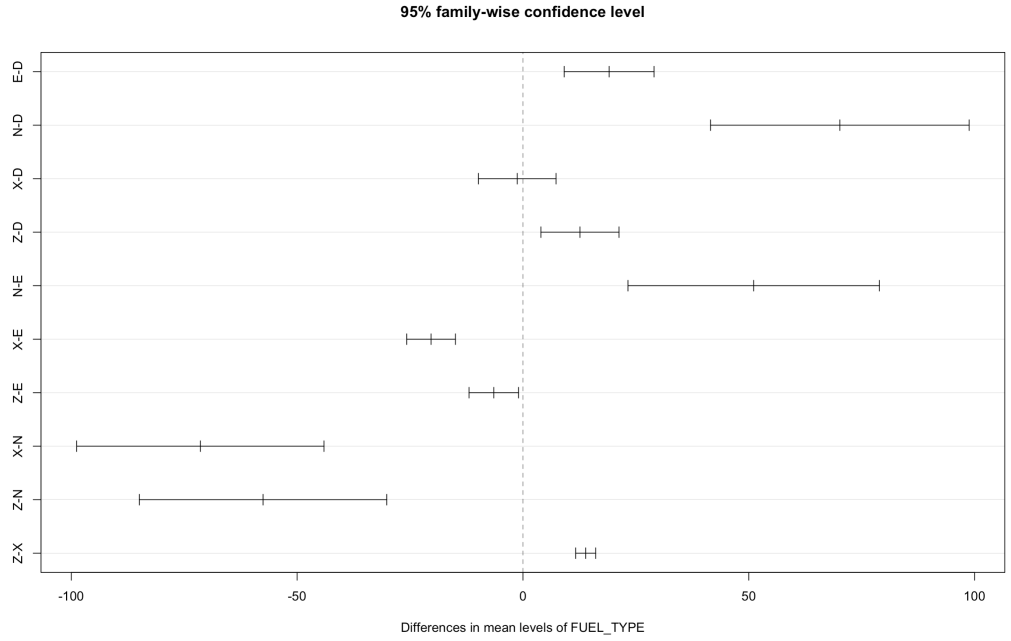

Figure 4.12: Tukey Test CO2 Emission

fuel consumption combined and fuel type:

This one-way ANOVA tests on fuel consumption and for combined (city and highway) driving conditions.

Table 4.5: Summary of ANOVA Analysis for FUEL_CONSUMPTION

Source of Variation	**Df**	**Sum Sq**	**Mean Sq**	**F value**	**Pr(>F)**
FUEL_TYPE	4	38498	9625	1282	$< 2 \times 10^{-16}$ ***
Residuals	26067	195758	8		

Since the p-value is much smaller than considerable threshold like 0.05, we have enough evidence to reject the null hypothesis. The null hypothesis in this case suggests that the type of fuel doesn't really have an impact on fuel consumption. However, the low p-value tells us that we can't just attribute the observed differences in fuel consumption to random variability. The F-statistic (1282) adds evidence for the same results.

Table 4.6: Tukey Multiple Comparisons of Means for COMB_km

Comparison	Difference	Lower CI	Upper CI	p adj
E-D	7.4482818	7.0128924	7.8836713	0.0000000
N-D	7.7212896	6.4681264	8.9744528	0.0000000
X-D	1.4985652	1.1224082	1.8747222	0.0000000
Z-D	2.0833032	1.7049190	2.4616873	0.0000000
N-E	0.2730078	-0.9454821	1.4914977	0.9733826
X-E	-5.9497166	-6.1859237	-5.7135095	0.0000000
Z-E	-5.3649786	-5.6047165	-5.1252408	0.0000000
X-N	-6.2227244	-7.4213269	-5.0241220	0.0000000
Z-N	-5.6379865	-6.8372897	-4.4386832	0.0000000
Z-X	0.5847380	0.4877618	0.6817141	0.0000000

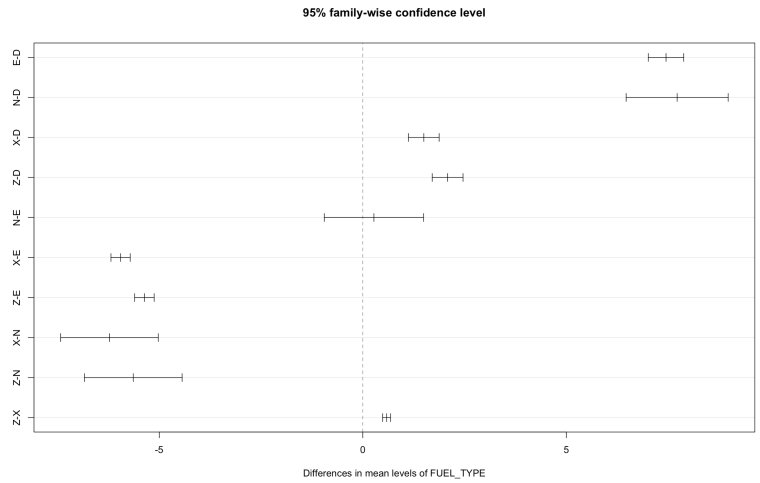

Figure 4.13: Fuel Consumption

The Tukey Honestly Significant Difference (HSD) test is used for to check the differences in how much fuel different types of vehicles consume. some key results are:

Gasoline (E) versus Diesel (D): A significant difference ($p < 0.0000000$) emerged, indicating that gasoline-fuelled vehicles tend to consume about 7.45 units more fuel than diesel-fuelled ones. Natural Gas (N) versus Diesel (D): Another significant difference ($p < 0.0000000$) emerged, implying that natural gas fuelled vehicles consume around 7.72 units more fuel on average compared to diesel-fuelled ones.

4.4 Feature Engineering:

To make model more effective to predict the things right that called "feature engineering".Feature engineering includes different things like creating new columns from existing columns and with this provide more meaningful information to machine learning algorithms.

frequency encoding:

In frequency encoding, we transform categorical data into numbers. We count how many times each category present and then replace that categories with these counts. This way, the model can understand the importance of that categories when making predictions. It is like giving a numbers to each category based on how common or rare it is present in the data. This helps the model to learn better from the patterns in the data and make better predictions. [11]

In this case, focus is on the "MAKE" attribute, which represents the car's manufacturer. To improve its usefulness, one need to calculated the frequency of each car manufacturer in the dataset and divided it by the total number of data rows. This result saved in a new attribute called "MAKE NEW," which indicates the relative presence of each car manufacturer in this dataset. By applying the same Frequency Encoding to the "MODEL" column, indicates whether the car has a high-output engine. To enhance its representation, this calculate the frequency of each unique value in the "MODEL" attribute and divided it by the total number of data points. The result is a new attribute named "MODEL NEW," which quantifies the prevalence of each "MODEL" category.

The purpose of creating these new attributes, "MAKE NEW" and "MODEL NEW," is to provide the machine learning model with additional insights. By converting categorical information into numerical values based on their frequencies, the model can potentially discern patterns related to specific manufacturers and high-output engine models and their impact on the target variable.

one-hot encoding:

After the statistical analysis phase which is to understand what affects on CO2 emission predictions, in the next step it have been realized that some categories like "FUEL TYPE," "VEHICLE CLASS," and "TRANSMISSION" were really important. These categories seemed to matter a lot when it comes to predicting CO2 emissions. To make the most of this useful information and make sure it fits well into prediction models, it is beneficiary to use something called one-hot encoding.

In one-hot encoding, we need to transform categorical columns into numerical/binarily a format like structure that a machine learning model can understand. Each category becomes a separate

column, and if an example belongs to that category, the column gets a "1"; otherwise, it gets a "0". Imagine you have different categories, like colours red, blue, and green. We create separate boxes for each colour. If an item is red, the "red" box gets a checkmark (1); if it's blue or green, the corresponding boxes get empty (0). This helps the model treat each colour as unique and avoids thinking one colour is better than another. It's a way to represent categorical information in a language that the model can work with. [12]

To do this, need to gathered the important categories "FUEL TYPE," "VEHICLE CLASS," and "TRANSMISSION" into a special table called 'Categorical df'. But this table had some extra numbers in it that could be confusing. So, we cleaned it up by starting the counting from zero and getting rid of the old numbers.This table, 'Categorical df', is important now. As used one-hot encoding on it, which is like giving models a secret code to understand the categories better. This helps models catch all the tricky connections between these categories and CO2 emissions. By doing this,prediction models becomes smarter and better at figuring out CO2 emissions.

Using one-hot encoding and organizing the categories in 'Categorical df' makes sure that "FUEL TYPE," "VEHICLE CLASS," and "TRANSMISSION" count in predictions. This way, models become more powerful and reliable when it comes to predicting CO2 emissions accurately.

Drop columns:

During the process of statistical analysis, after carefully examined the attributes in the dataset to determine which ones would be most valuable for predicting CO2 emissions accurately. After analysing the data, we identified several columns that could be dropped to enhance the quality and effectiveness of predictive models. [13]

- Model Year: This attribute showed a low correlation value of -0.205244 with CO2 emissions. Since it had a weak connection to the target prediction, so decided to remove it from consideration.
- . FUEL CONSUMPTION CITY (L/100): This column shows the signs of multicollinearity, which means it was closely related to other attributes. To avoid confusion and redundancy, so need to exclude it from the dataset.
- FUEL CONSUMPTION HWY (L/100): Similar to the previous case, this column also shows the multicollinearity. To maintain the clarity and effectiveness of models, option chose to eliminate it.
- COMB (mpg): This column is a duplicate column, containing similar information to other columns. To ensure a streamlined and efficient dataset, need to removed this redundant column.

- CO2 rating: With a low correlation value of -0.47 and a significant percentage of missing values (72.5), this attribute did not strongly contribute to predicting CO2 emissions. As a result, decision made to drop it from the dataset.

- Smog rating: This column displayed a correlation value of 0.41, which was not particularly strong. Additionally, a substantial portion of the data (77) was initially missing. Considering these factors, I chose to eliminate this attribute to improve the focus and accuracy of my models.

By this process carefully chose which columns to remove from the data to make models better at predicting CO2 emissions. This helps my models focus on the most important things for accurate predictions. The "MAKE NEW" and "MODEL NEW" attributes give numbers that show how often different car makers and special engine models appear in the data. This helps the models understand and use this important info to make better predictions.

Scaling:

Scaling is important preprocessing step that involves transforming the columns values into common range which helps to reduce the computational power requirement and reduce time for computation.Using scaling technique all the columns value range reduce to one range which becomes more easy to compare columns with each other. The speed of model training and predicting becomes more fast because all column values lie in one range.Earlier,some columns values are single digit and other columns are more than one digit.We can make the scaling process using many techniques.

Scaling Techniques [14]:

Min-Max Scaling (Normalization):

Formula:

$$X_{\text{normalized}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

[14] This scales the features to a range between 0 and 1, preserving the relative relationships between data points. It's suitable for algorithms sensitive to feature magnitudes, like k-means or neural networks.

Standardization (Z-score Scaling):

Formula:

$$X_{\text{standardized}} = \frac{X - \mu}{\sigma}$$

This scales the features to have zero mean and unit variance. It maintains the distribution shape and is suitable for algorithms that assume Gaussian-distributed data, like linear regression.

Robust Scaling:

Formula:

$$X_{\text{robust}} = \frac{X - Q_1}{Q_3 - Q_1}$$

This scales the features by using the interquartile range (IQR) instead of the standard deviation, making it robust to outliers.

Log Transformation:

Formula:

$$X_{\log} = \log(X)$$

[15] This applies a logarithmic transformation to the data, which can help in reducing the effect of extreme values and making the data more symmetric. [15]

Box-Cox Transformation:

Formula:

$$X_{\text{transformed}} = \begin{cases} \frac{X^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(X) & \text{if } \lambda = 0 \end{cases}$$

[15] The Box-Cox transformation tries to stabilize variance and make the data more Gaussian-like.

Quantile Transformation:

Formula:

$$X_{\text{quantile}} = F^{-1}(X)$$

This maps the data to a specified distribution, typically a uniform or Gaussian distribution, using the inverse cumulative distribution function (CDF).

Max Absolute Scaling:

Formula:

$$X_{\text{max\_abs}} = \frac{X}{\max(|X|)}$$

This scales the features to the range between -1 and 1, preserving the sign of the data while ignoring the outliers.

Why to choose regression model to predict co2 emission?

Usage of regression models to predict CO2 emissions because they are good at helping to understand how different things about a vehicle (like its features and how it's used) are related to how much CO2 it produces. These models are like tools that can show us how changes in one thing can lead to changes in another thing, like how changes in a car's engine size or how often it's driven can affect how much CO2 it emits.

Important information is learned from the Exploratory Data Analysis (EDA) process, such as the identification of various factors in the data that affects the amount of CO2 emitted by a vehicle.Ensuring this reliability of the result is achieved by using hypothesis testing to confirm the conclusions

Regression models are suitable for this study because they're designed to work with numbers and can help us make predictions about CO2 emissions based on the information we have about the vehicles. Regression models provide a robust framework for modelling continuous numerical outcomes, making them an ideal choice for this study's objectives.

In the context of predicting CO2 emissions, regression models offer several key advantages:

- Regression models offer a quantitative understanding of how changes in independent variables influence the dependent variable.
- Regression models provide interpretable coefficients that indicate the direction and magnitude of the relationship between each independent variable and the dependent variable. This enables a clear understanding of which factors contribute to higher or lower CO2 emissions.
- By using historical data to train the model, regression techniques can learn the complex patterns and trends that govern CO2 emissions
- Regression models can handle various types of independent variables, including categorical and continuous variables. This flexibility is crucial when dealing with a dataset that includes different vehicle characteristics and usage patterns.

Model 1: Multilinear Regression Model:

With the help of literature review and looking at data columns we need to predict the co2 emission.It can be easily identified. This kind of learning is called supervised machine learning. Since the predictive column is a number (like CO2 emissions), it falls under a category called regression.

Now, in this dataset,the predictive column is CO2 emissions, and all other columns that might affect on it, like vehicle characteristics and how the vehicles are used. Since I'm dealing with multiple

factors, we need a type of regression that can handle this complexity. That's where multilinear regression comes in.

Multilinear regression is a tool that lets me consider all these different factors together. It helps me understand how each factor, like vehicle traits and usage patterns, contributes to the final CO2 emissions number. By using multilinear regression, One can see how these factors work together and get a clearer picture of their combined impact. [16]

To predict CO2 emissions using various vehicle characteristics, we create a graph where the vertical line (y axis) represents the CO2 emissions we want to predict, and the horizontal lines (x axis) represent different vehicle variables that could influence emissions, like engine size, weight, and fuel efficiency.

In the context of multilinear regression model, the slope (m) and intercept (c) play pivotal roles in shaping our prediction line. The slope (m) signifies the extent of change in CO2 emissions for a unit alteration in each vehicle variable, while the intercept (c) establishes the starting point of the line. By fine tuning these parameters, we mold our prediction line to closely align with the linear regression line that accurately represents the data trend. This process reduces the disparity between our forecasted emissions and the actual values.

With multiple input variables at play, the multilinear regression equation takes a more comprehensive form [17]:

$$Y = \beta 0 + \beta 1 * X1 + \beta 2 * X2 + ... + \beta n * Xn + e \tag{4.4.1}$$

Here:

- Y represents the predicted CO2 emissions.

- b 0 represents the intercept coefficient.

- b 1, b 2, ..., bn are the beta coefficients corresponding to each predictor variable X1, X2, ..., Xn.

-e error term

In this multilinear regression model, there are 69 different beta values, each corresponding to a specific vehicle characteristic or parameter (X1, X2, ..., X69). These beta values will tell me how much each parameter influences the predicted CO2 emissions (Y) and there is a single intercept term (c) that sets the initial point of the prediction line on the CO2 emissions scale.

So, the equation for predicting CO2 emissions (Y) using all these parameters will look like this:

$$CO2Emissions(Y) = \beta 0 + \beta 1 * Parameter(X1) + \beta 2 * Parameter2(X2) + ... + \beta 69 * Parameter69(X69) + e \tag{4.4.2}$$

where,

-Y represents the predicted CO2 emissions.

- b0 represents the intercept coefficient.

- b1, b2, ..., b69 are the beta coefficients corresponding to each predictor column of my data set X1, X2, ..., X69.

-e represents error term

The equation captures the combined impact of all predictor variables on the CO2 emissions outcome. Each beta coefficient quantifies the change in emissions associated with a unit change in the respective predictor variable, while keeping other variables constant. This equation considers the effects of all 69 parameters and their respective beta values, along with the intercept, to estimate the CO2 emissions for a given set of vehicle characteristics.

As the results of the multilinear regression,gaining more valuable insights as how different vehicle parameters affect CO2 emissions. The beta values associated with each parameter provides a clear picture. When a beta value is positive, it means that increasing that specific parameter, like engine size or cylinders, will lead to higher CO2 emissions. On the other hand, if the beta value is negative, it suggests that an increase in that parameter results in lower CO2 emissions.

We have two techniques to find these values:

1. Closed Form Solution (OLS - Ordinary Least Squares): This technique involves using specific formulas to directly calculate the values of m and c. It doesn't require complex mathematical operations like integration or derivatives. Instead, it focuses on minimizing the overall squared difference between our prediction line and the actual data points. This method provides precise solutions that minimize errors and give us the best-fit line.

The least squares method is commonly used to estimate the coefficients that minimize the sum of squared residuals:

$$\hat{\beta} = (X^\top X)^{-1} X^\top y$$

Where:

$\hat{\beta}$: Represents the estimated coefficients

$X^\top$: Transpose of the predictor matrix X

y : Response variable

2.Non-closed Form Solution (Gradient Descent): This approach involves an iterative process where we start with initial values for m and c and gradually adjust them to minimize prediction errors. It's like finding the lowest point in a valley by taking steps in the steepest downhill direction. While it

may require more computational effort, gradient descent is versatile and can handle complex models where closed-form solutions are challenging.

The choice to use the Ordinary Least Squares (OLS) technique for my predictive model stems from careful consideration of the characteristics of my dataset and the complexity of the problem at hand. With a relatively small number of input parameters and a moderate-sized dataset, OLS offers a suitable and efficient approach to achieve accurate results.

Given the limited complexity of the model and the manageable dataset size, OLS provides an effective method to estimate the relationships between the input parameters and the CO2 emissions. By minimizing the squared differences between the predicted values and the actual emissions, OLS ensures a balanced fit of the model to the data points

Model 2: Random Forest Regression Model:

The multilinear regression model has some restrictions even though it is a helpful instrument for calculating CO2 emissions based on different vehicle attributes. Its assumption of linearity is one of its main limitations. In multilinear regression, the output variable (CO2 emissions) and the input variables (such as engine size, cylinder count, and fuel efficiency) are assumed to have an effect that is strictly linear. These factors' connection to one another may be complex and nonlinear.

Multilinear regression has some problems like when the input variables interact or depend on one another. It makes the false assumption that each input variable influences the result in a separate way. For example, depending on the number of cylinders, the impact of engine size on CO2 emissions could vary.

Random forest words are divided into two words: Random, like an unbiased sampling technique, and forest, a group of trees (models). As this dataset has a regression problem, we need to selected the random forest regression model. In the regression model, there are n numbers of decision tree models are connected parallelly with each other. When unbiased data provide to these sub-models, they learn from these provided data and give their respective results. Then this regression model takes all the results and calculates the average of all results, and this is the final output of this model. [18]

Decision trees are a fundamental part of Random Forest. In regression Random forest,Each decision tree predicts the target value based on a series of binary splits. The mathematical formulation involves to determining the best split point at each node to minimize losses measure,as Mean Squared Error (MSE) for regression:

$$MSE = \frac{1}{N_m} \sum_{i \in D_m} (y_i - \bar{y}_m)^2$$

Where:

$$N_m : \text{Number of data points in node } m$$

$$D_m : \text{Set of data points in node } m$$

$$y_i : \text{Target value of data point } i$$

$$\bar{y}_m : \text{Mean target value of node } m$$

In Bootstrapping,Random forest regression uses random sampling with replacement for dividing dataset in samll subset and provide this to each decision tree. [18] The formula for bootstrapping is :

$$\text{Bootstrap Sample} = \{(x_i, y_i)\}$$

At each split point of a decision tree, a subset of features is randomly selected. This helps introduce diversity and prevent overfitting.

For Random Forest Regression, the final prediction is the average of predictions from all decision trees:

$$\text{Random Forest Prediction} = \frac{1}{T} \sum_{t=1}^{T} \text{Decision Tree}_t(x)$$

Where:

$$T : \text{Total number of decision trees}$$

$$\text{Decision Tree}_t(x) : \text{Prediction of the } t\text{-th decision tree for input } x$$

Variable importance is often measured based on the decrease in impurity achieved by each feature's splits across all decision trees. The importance of feature j can be computed as the average over all trees of the decrease in impurity when feature j is used for splitting:

$$\text{Variable Importance}_j = \frac{1}{T} \sum_{t=1}^{T} \text{Decrease in Impurity}_t(j)$$

This measure provides insight into how much each feature contributes to the model's performance.

Using the random forest regression model,successfully detailed the problem of Biased, Variance trade-off. As we all know, we need to put the Biased and Variance factor as low as possible for every model performance because these two are associated with model accuracy. The biased is the error when model makes a mistake while learning training data. The Variance is an error that occurs on the test data when the model tries to memorize the results. In multilinear regression model, got good results, we need to make it better by applying different techniques like random forest to reduce the variance and Biased errors.

Model 3: XG Boost regression:

Extreme Gradient Boosting is a advance machine learning algorithm for both regression and classification problems. It works on boosting algorithms. XG Boost has ability to handle complex data patterns. XG Boost's key advantages include its flexibility, speed, and ability to handle a variety of data types.

This model works on the most common loss functions include "mean squared error" for regression and "logarithmic loss" (cross entropy) for classification.In this regression model, there. A series of sub-models (decision trees) are connected serially. It works on the principle of âlearn from the partâ. The data was provided to the first sub-model. After working on the data, this model provides its result and the error to the next connected model to reduce that error and improve the result, and so on. Finally, we can get a better result and less error output from this model. XG Boost's key advantages include its flexibility, speed, and ability to handle a variety of data types.

XG Boost aims to minimize an objective function, which is the sum of the loss function and a regularization term that reduced overfitting. For regression, the objective function formulated as [19]:

$$\text{Objective} = \sum_{i=1}^{n} \left(\frac{1}{2}(y_i - \hat{y}_i)^2 + \lambda \Omega(f_t) \right)$$

Where:

$\hat{y}_i$: Prediction of the model at iteration t

$\Omega(f_t)$: Regularization term that penalizes the complexity of the model at iteration t

λ : Regularization parameter that controls the strength of regularization

With the help of the XG Boost model, we can tackle different problems faces with the multilinear regression model. Some of the problems are outlier effects on the prediction result as the XG Boost works on an ensembled technique; it easily deals with complex data structure and outliers. It also reduces the effect of the Biased problem, which occurs during the model training process. This model also can deal with Variance.

Regression metrics:

To assess the effectiveness of regression models, here we are using a comprehensive evaluation approach by calculating key regression metrics for each model. These five metrics provide valuable insights into the performance of the models, allow to gauge their accuracy and the extent of their predictive capabilities.

1. MAE (Mean Absolute Error): MAE represents the average difference between my predicted CO2 emissions and the actual values. This helps me understand, on average, how close or far my

predictions are from the actual emissions. For instance, if we predict a specific CO2 emission level, the MAE shows how much typically deviate from the actual value. [20]

$$MAE = \sum |ActualCO2 - PredictedCO2|/NumberofDataPoints \tag{4.4.3}$$

$$MAE = \sum |Y(actual) - Y(predicted)|/N \tag{4.4.4}$$

In this equation:

Y(actual) represents the actual CO2 emissions value for a data point. Y(predicted) represents the CO2 emissions value predicted by the model for the same data point.

N is the total number of data points in your dataset.

Advantages of MAE:

- Direct Interpretation: One notable advantage of Mean Absolute Error (MAE) is its unit consistency. As it shares the same unit as the predicted and actual values (Y), any error calculated using MAE is intuitively relatable and easily interpretable in the context of the original data. This aids in conveying the magnitude of prediction errors in a straightforward manner.

- Robustness to Outliers: MAE is less sensitive to outliers compared to other error metrics, such as the Mean Squared Error (MSE). Outliers have a limited impact on the overall calculation, making MAE a reliable choice when dealing with datasets that may contain extreme or unusual data points.

Disadvantages of MAE:

- Non-Differentiable at Zero: MAE employs the modulus function, which is not differentiable at zero.

2. MSE (Mean Squared Error):

Similar to MAE, MSE looks at the differences between predictions and the actual emissions. However, it squares these differences before averaging them. It gives more weight to larger errors. In my CO2 context, it's like considering the average of the squared differences between my predictions and the real emissions. [20]

$$Formula : MSE = \ sum(ActualCO2 - PredictedCO2)^2/NumberofDataPoints \tag{4.4.5}$$

$$MAE = \sum (Y(actual) - Y(predicted))^2/N \tag{4.4.6}$$

In this equation:

Y(actual) represents the actual CO2 emissions value for a data point. Y(predicted) represents the CO2 emissions value predicted by the model for the same data point.

N is the total number of data points in your dataset.

Advantages of MSE:

- Differentiability: Mean Squared Error (MSE) overcomes the differentiability issue present in MAE. The squared nature of the loss function ensures differentiability at all points, including zero, facilitating the use of various optimization algorithms.

Disadvantages of MSE:

- Unit Difference: Unlike MAE, MSE does not share the same unit as the original data (Y) because of square terms.

- Not robust to Outliers: MSE creates large errors due to the squaring operation in its formula.

3. RMSE (Root Mean Squared Error):

RMSE is the square root of MSE. Since it's in the same units as CO2 emissions, it's easier to grasp. It indicates how much, on average, my predictions deviate from the actual values, considering both small and large deviations.

$$Formula : RMSE = \sqrt{MSE} \tag{4.4.7}$$

$$MAE = \sqrt{(\sum (Y(actual) - Y(predicted))^2 / N)} \tag{4.4.8}$$

In this equation:

Y(actual) represents the actual CO2 emissions value for a data point. Y(predicted) represents the CO2 emissions value predicted by the model for the same data point.

N is the total number of data points in your dataset.

Advantages:

- Same Unit as Y Prediction: A significant advantage of RMSE is that it shares the same unit as the original CO2 emissions data (Y). This characteristic allows us to interpret the error directly within the context of CO2 emissions.

Disadvantage:

- Not Highly Robust to Outliers.

4. R-squared (Coefficient of Determination):

R-squared (Coefficient of Determination) is a vital measure that offers insights into the effectiveness of our linear regression model in capturing and explaining the variability in CO2 emissions based on the chosen vehicle characteristics. It is often referred to as the "goodness of fit" metric. By comparing the performance of our linear regression line against a simple mean line for CO2 emissions, R-squared helps us understand how well our model is fitting the actual data.

R-squared = 1 - (SSR / SSM)

Where:

- SSR represents the sum of squared errors in our regression line, reflecting how well our model's predictions align with the actual CO2 emissions values.

- SSM stands for the sum of squared errors in the mean line, signifying the variability between the actual CO2 emissions and their average value.

$$R - squared = 1 - \frac{[(\sum(Y(actual) - Y(predicted))^2 regressionline)}{(\sum(Y(actual) - Y(predicted))^2 meanline)]} \tag{4.4.9}$$

Where: - Y(actual) represents the actual observed CO2 emissions values.

- Y(predicted) corresponds to the CO2 emissions values predicted by our regression model.

5. CV R Squared (Cross Validated R squared):

CV R-squared is like R squared but with added dependability. It tests if model consistently works well on new data subsets it has yet to see. It ensures that my predictions are consistently accurate.

When these metrics are analyzed, the objective is to get numerical values for MAE, MSE, and RMSE, indicating a close comparison between the predictions and the actual CO2 emissions. As for R squared and CV R squared, the aim is looking for larger values, closer to 1, to show that selected variables are effectively explaining the variations in CO2 emissions.

PCA (Principal Component Analysis):

PCA is a technic where we reduce the excess distance between points of multi-dimensions dataset. Due to the multiple dimensions, the distance between the two points is extended than the normal distance, which creates a problem of missing or false information about the data point. We need to reduce that distance to get exact information about the point and make a better model. PCA helps to reduce dimensions and get the required information. PCA starts with centring the data by subtracting the mean from all data points. Then find out the covariance matrix to see the relationship between all features. Then the eigen vectors and eigen values need to find from the covariance

matrix. The eigenvectors represent the direction in which all points vary the most, and the eigenvalues represent the magnitude of variation. Then sorting the eigenvectors by indexing the eigenvalues. After this, to reduce the dimensions of data, we need to select the subset or top component from all PCA components.Using following equation step wise PCA works as feature exaction technique. [21]

1. Mean-Centered Data

$$\text{Mean-Centered Feature} = \text{Original Feature} - \text{Mean of Original Feature} \tag{4.4.10}$$

Where:

$$\text{Mean-Centered Feature} : \text{Transformed feature after mean-centering}$$

$$\text{Original Feature} : \text{Original feature value}$$

$$\text{Mean of Original Feature} : \text{Mean value of the original feature}$$

2. Covariance Matrix

The covariance matrix measures the relationships between two features in the data. For a dataset with n samples and p features, the covariance matrix $\mathbf{C}$ is given by:

$$\mathbf{C} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top \tag{4.4.11}$$

Where:

$$\mathbf{C} : \text{Covariance matrix}$$

$$n : \text{Number of samples}$$

$$\mathbf{x}_i : \text{Mean-centered data point}$$

$$\bar{\mathbf{x}} : \text{Mean of mean-centered data points}$$

3. Eigenvalue Decomposition

In PCA,there are the eigenvectors and the eigenvalues of the covariance matrix. The eigenvectors shows the directions (principal components) of maximum variance, and the corresponding eigenvalues indicate the amount of variance explained by each principal component.

$$\mathbf{C}\mathbf{v} = \lambda \mathbf{v} \tag{4.4.12}$$

Where:

$\mathbf{C}$: Covariance matrix

$\mathbf{v}$: Eigenvector

λ : Eigenvalue

4. Explained Variance Ratio

The explained variance ratio shows the proportion of the total variance explained by each principal component. It is calculated by dividing each eigenvalue by the sum of all eigenvalues. [21]

$$\text{Explained Variance Ratio} = \frac{\lambda_i}{\sum_{j=1}^{p} \lambda_j} \tag{4.4.13}$$

Where:

Explained Variance Ratio : Proportion of variance explained by a principal component

λ_i : Eigenvalue of the i-th principal component

$\sum_{j=1}^{p} \lambda_j$: Sum of all eigenvalues

5. Projection onto Principal Components

Once the principal components are determined, the original data can be projected onto the lower-dimensional space spanned by the principal components.

$$\text{Projected Data Point} = \mathbf{x}^\top \mathbf{v} \tag{4.4.14}$$

Where:

Projected Data Point : Data point in the lower-dimensional space

$\mathbf{x}$: Original mean-centered data point

$\mathbf{v}$: Principal component (eigenvector)

CHAPTER 5

Results And Discussion

After doing all the required steps in data pre-processing, the next step is model building and analysis of the performance of the model. As mentioned earlier, three regression models need to be performed on this data. One is a simple machine learning model multi-linear regressor, and the other two are advanced machine learning models, Random Forest regressor and XG boost regressor.

During the very first trial, data set was given to these models without any feature engineering process. The results for these first trials are not very good. All three models are trying to mug up the training data, called an overfitting problem. This is easy to identify with train test validation.

To solve this problem, there is one technique is important called as feature engineering.In this technique, there are mainly two methods: one is feature selection, and the other one is featuring exaction. For this data set, a total of 69 columns are present for prediction;it is suitable to use the feature extraction technique instead of feature selection because one is a computation problem as we have 69 columns and the other lacks in-depth domain knowledge. The second technique features extraction, which extracts as much as possible information from the data and makes a prediction model with less requirement of computational power.

Using the selected number of components, we perform the model operation.To get optimum PCA numbers, all the results of the regression model were cheeked, and it decided to take the top performance PCA number and use that number for all model performance. It is also required to checked the percentage of data gathered through that PCA number. It comes to around 84 per cent, which is good. With this percentage of data, models are performing as similar as full models with less computation. PCA 48 is the top performance, and all the next model building was done based on PCA 48. With the help of PCA, the problem of overfitting was fixed. As the components were reduced, only the required and imported information was delivered to the model as training data.

Following are the results of all the three models:

5.1 Models first trail result:

An initial all three models were developed to predict CO2 emissions using a dataset comprising a total of 69 variables. Following some results of all three models:

5.1.1 Interpreting Coefficients for Multilinear model:

Each coefficient represents the estimated change in CO2 emissions for a one-unit change in the corresponding predictor variable, while holding all other factors constant. This allows us to understand how specific attributes contribute to the overall picture of emissions

The coefficient of 6.94 for the "ENGINE SIZE (L)" attribute. This coefficient indicates that, on average, when the engine size of a vehicle increases by one unit, the model predicts an increase of approximately 6.94 grams per kilometre (g/km) in CO2 emissions. simply, larger engine sizes tend to correlate with higher CO2 emissions, assuming other factors remain constant. In contrast, at the coefficient of -3.17 associated with "COMB (L/100 km)," a measure of combined fuel consumption. This negative coefficient suggests that a one unit increase in combined fuel consumption, indicating poorer fuel efficiency, is linked to a decrease of around 3.17 g/km in CO2 emissions. That means, vehicles with better fuel efficiency, which consume less fuel per distance travelled, tend to exhibit lower CO2 emissions.Positive coefficients, such as those around 2.08 and 1.60, indicate that certain vehicle classes, like "VEHICLE CLASS FULL SIZE" and "VEHICLE CLASS MID SIZE," or transmission types, like "TRANSMISSION A8" and "TRANSMISSION AM6," are associated with higher CO2 emissions when they increase. On the flip side, negative coefficients like -7.63 and -7.35 linked to attributes like "FUEL TYPE E" and "CYLINDERS" suggest that specific fuel types or a greater number of cylinders tend to result in lower CO2 emissions.

5.1.2 Models performance:

Upon assessment of the initial all three models performance, issue with overfitting was identified. This was evidenced by the observation of a negative test R-squared value of -1.11 for multi linear model and very high accuracy for Random forest and XG Boost model around 95 and 96 respectively.This was indication of the models not performing good on unseen data.All three models were trying to mug up the input data and tried to connect all the feature points present in the input data which means it can work better on only input data but not on new data.

To tackle the overfitting issue and enhance the performance of the all the regression models, the decision was made to implement Principal Component Analysis (PCA).Selection of number of PCA vectors was done by performing the model fitting method with 1 to 69 PCA and selecting top performance from them. PCA was conducted on the dataset, leading to a reduction in dimensionality from 69 variables to 48 principal components. This reduction aimed to simplify the dataset's complexity while preserving a significant amount of information around 80-90 percentage.

5.2 Models second trail results:

After selecting the PCA 48 again all three model were performed on this transformed data. Results for the second trial is as follows:

5.2.1 Multiple linear regression model:

The model accuracy r2 score for both test and train got 0.75 and 0.79 which are close enough which means model is performance is good.

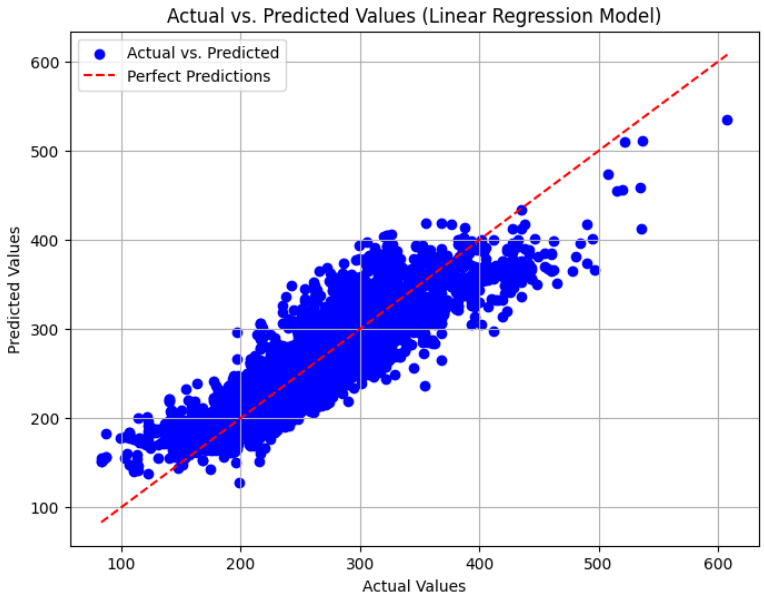

Figure 5.1: Predicated Linear Line

Key Statistical Results Explained:

1. R squared Value :

The model's effectiveness in explaining the differences in CO2 emissions becomes evident through the R-squared value, standing at 0.75 , approximately 75 percentage

2. MAE (Mean Absolute Error):

This model's predictions varies from the actual CO2 emissions by approximately 21.45 grams per kilometres.

3. MSE (Mean Squared Error):

The linear regression model's predictions have a squared error of about 806.02. This is the average squared difference between predicted values and actual values.

4. RMSE (Root Mean Squared Error):

The square root of MSE is approximately 28.39. It gives a mean of the error. This tells how well the model's predictions match the actual values.

5. CV R squared (Cross-Validated R-squared):

Through cross-validation, the linear regression model achieves a R-squared value of approximately79.65 percentage on average across different subsets of the data.

5.3 Random Forest Regression Model:

1. R squared Value :

An R-squared score of approximately 0.86 indicates that the model predicting round 86.4 percentage of the variability in CO2 emissions, suggesting a strong ability to fit the data.

2. MAE (Mean Absolute Error):

The random forest regressor model's predictions have an average absolute error of around 11.7 grams per kilometer. This is a measure of the average absolute difference between predicted and actual values.

3. MSE (Mean Squared Error):

The squared error of the random forest regressor model's predictions is approximately 296.89. It gives an idea of the average squared difference between predicted and actual values.

4. RMSE (Root Mean Squared Error):

The MSE score is about 7.23, representing the average magnitude of the error in the model's predictions.

5. CV R squared (Cross-Validated R-squared):

Through cross-validation, the random forest regressor achieves an average R-squared of around 85.56 percentage.

5.4 XG Boost Regression Model:

1. R squared Value :

An R-squared score of approximately 0.8817 for the XG Boost model.

2. MAE (Mean Absolute Error):

This model's predictions have an average absolute error of around 1.70 grams per kilometer.

3. MSE (Mean Squared Error):

: The squared error of the random forest regressor model's predictions is 296.8.

4. RMSE (Root Mean Squared Error):

The square root of MSE is about 17.23, representing the average magnitude of the error in the model's predictions.

5. CV R squared (Cross-Validated R-squared):

Through cross-validation, the random forest regressor achieves an average R-squared of around 87.34.

	Model	MAE	MSE	RMSE	R-squared	CV R-squared
1	Linear Regression	21.475	808.388	28.432	0.7526	0.7965
2	RandomForest Regressor	11.710	296.893	17.240	0.8644	0.8556
3	XGB Regressor	11.709	296.893	17.230	0.8817	0.8734

Table 5.1: Comparative Matrix

5.5 Q-Q Plot for all Three models:

Quantile-Quantile plots, called as QQ plots. In this a type of graph used in statistics to figure out whether a particular data distribution follows a theoretical distribution, usually the Gaussian (normal) distribution. It helps to decide if the quantiles (ordered values) of the data match with what we could have expected from a particular theoretical distribution. This comparison is crucial to figure out how well the observed data and theoretical distribution fit together. Comparing all the three Q-Q plots of above three models,the random forest model and the XG Boost model have a more number of points that are closer to the theoretical line (Q-Q line), which means that these model predictions try to come

closer to theoretical results. For the multilinear regression model, the prediction lines are less close to the Q-Q line compared to the other two models. These points look scatter-type. Both the random forest and XG boost regression models perform well than the linear regression model.

Reviewing all three model results, regression metrics and Q-Q plot results, XG Boost regression perform 87 percenntage best in all three models which were used. Then the Random forest regrssion is perform slightly less than the XG Boost one and then the multilinear regression model with 75 percentage of accuracy.

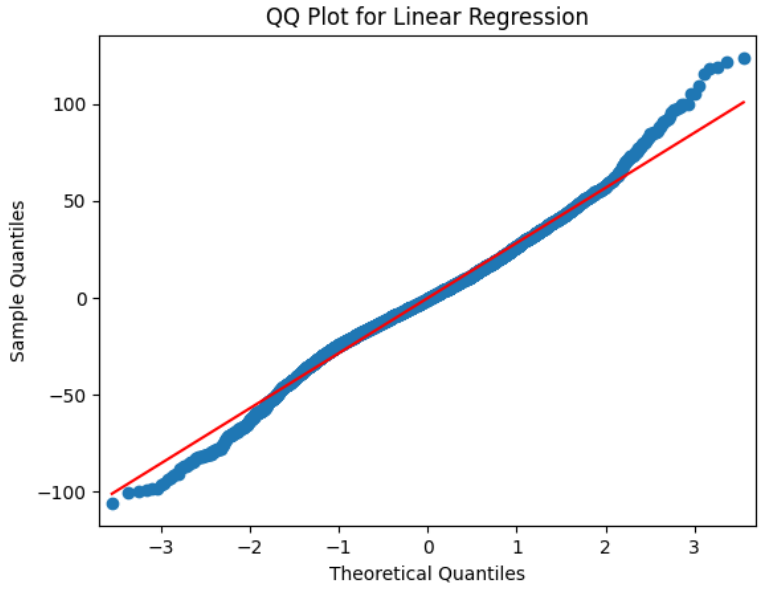

Figure 5.2: QQ Plot Multi Linear Regression

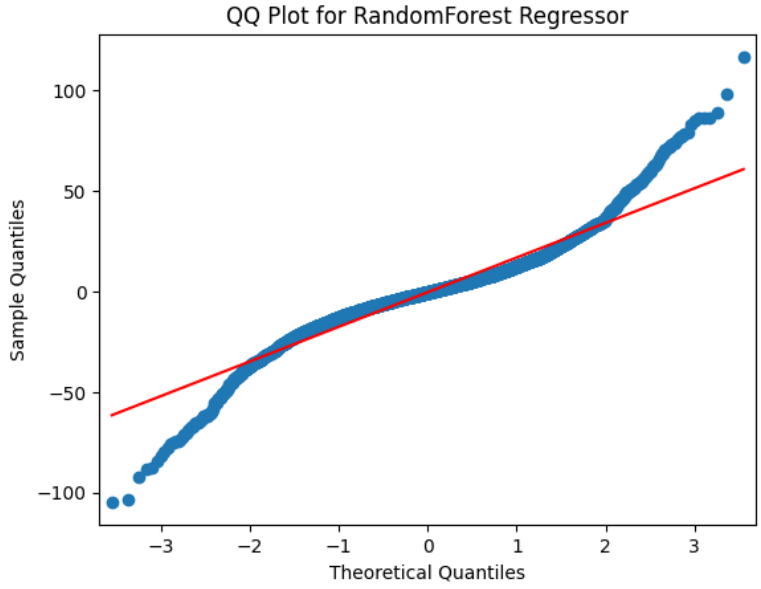

Figure 5.3: QQ Plot Random Forest

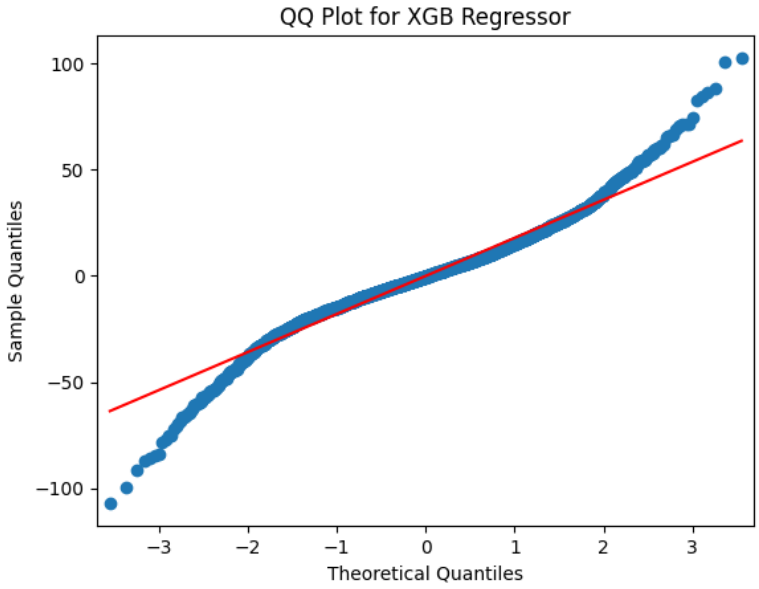

Figure 5.4: QQ Plot XG Boost

5.6 Discussion

- Model Complexity: Random Forest and XG Boost are ensemble methods that combine multiple decision trees. They can capture nonlinear relationships more effectively than Linear Regression, which works only on linear relationship between features and the target.

- Regularization: Both Random Forest and XG Boost incorporate regularization techniques that help prevent overfitting and improve generalization. This enhances their performance on unseen data compared to Linear Regression.

- Tree Growth Strategy: XG Boost employs a leaf-wise growth strategy, making it capable of creating deeper and more complex trees. This allows XG Boost to capture intricate patterns that Linear Regression might miss.

- Handling Outliers: Random Forest and XG Boost are less sensitive to outliers compared to Linear Regression, which can lead to improved performance in real-world datasets.

For accurate and resilient predictions of CO2 emissions, the XG Boost regression model stands as the top choice, closely followed by the Random forest regression model. While the Multilinear Regression model offers good accuracy, its limitations in handling complex relationships and managing high variance errors make it less suitable for this specific prediction task.

CHAPTER 6

Conclusion

In this dissertation,statistical analysis and machine learning models used to analyse the data in-depth and predict the CO2 emission through the vehicle.

In the statistical analysis,two approaches, one is descriptive, and the other one is inferential. Through the descriptive analysis, getting information about mostly all the numerical columns in the data follow the normal distribution precisely. The Time series analysis showed the rise of CO2 emission throughout the past decades, which is in an uptrend. Big engine size, passenger vehicles and fuel type N and D have comparative more impact on the CO2 emission, and this has been verified with the help of inferential statistical technical called hypothesis tests.

To make a prediction model,there were three machine learning techniques: multilinear regression,XG Boost regression model and Random Forest regression. Among the three models, XG Boost regression gave an exceptional performance, with a massive 88.17 percentage accuracy across all three models. Following closely, Random Forest regression performance slightly reduced accuracy than XG Boost regression, while the multilinear regression model predicts with 75 percentage accuracy rate. As the Random Forest and XG Boost are advanced machine learning models, they could tackle the problem of Variance better than the multilinear model. For the multilinear regression model, OLS technic were used to run the model because the Pythonâs library works on the same method. But we can use the Gradient descent technique for more data size and better results

6.0.1 Future Scope:

In a future analysis of this topic and this dataset, we can use deep learning models like Artificial Neural Networks with the help of more data. The deep learning models are performed better on big-size data.As reference to the literature review, CMEM model can be used on this dataset. In conclusion, this dissertation is based on predicting CO2 emissions by applying advanced machine learning models. The impressive performance of the XG Boost model indicates its potential for accurate prediction. At the same time, Random forest model , and multilinear regression also contribute valuable insights, and in future, we maybe get better results with the help of deep learning with additional data.

Bibliography

[1] https://courses.seas.harvard.edu/climate/eli/Courses/globalchangedebates/Sources/CO2saturation/more/ZhongHaigh2013.pdf. The greenhouse effect and carbon dioxide:wenyi zhong and joanna d. haigh.

[2] https://essex.primo.exlibrisgroup.com/discovery/. European environment agency. final energy consumption by sector and fuel; european environment agency: Brussels, belgium, 2015.

[3] https://essex.primo.exlibrisgroup.com/discovery/. Ntziachristos, l.; mellios, g.; tsokolis, d.; keller, m.; hausberger, s.; ligterink, n.; dilara, p. in-use vs. type-approval fuel consumption of current passenger cars in europe. energy policy 2014, 67, 403â411..

[4] https://www.unep.org/explore-topics/transport/ what-we-do/electric-mobility/electric-light-duty vehicles. Un environment, electric light duty vehicles. unep. 2021.

[5] https://essex.primo.exlibrisgroup.com/discovery/. Straka, w. et al. (2021) examining the economic and environmental impacts of covid-19 using earth observation data.

[6] https://essex.primo.exlibrisgroup.com/discovery/. Schoen, a. et al. (2019) a machine learning model for average fuel consumption in heavy vehicles. ieee transactions on vehicular technology.

[7] https://essex.primo.exlibrisgroup.com/discovery/. Barth, m. et al. (2001) recent validation efforts for a comprehensive modal emissions model.

[8] https://www.kaggle.com/datasets/abhikdas2809/canadacaremissions. Kaggle dataset.

[9] https://stefvanbuuren.name/fimd/ch introduction.html. Multiple imputation by stef van buuren.

[10] https://essex.primo.exlibrisgroup.com/discovery/. Verma, j. p. abdel-salam, a.-s. g. (2019) testing statistical assumptions in research.

[11] https://essex.primo.exlibrisgroup.com/discovery/. Zheng, a. casari, a. (2018) feature engineering for machine learningâ¯: principles and techniques for data scientists. first edition. beijing: Oâreilly.

[12] https://www.diva portal.org/smash/get/diva2:1259073/FULLTEXT01.pdf. An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing cedric seger.

[13] https://essex.primo.exlibrisgroup.com/discovery/. MÃ¼ller, andreas c, and sarah guido. introduction to machine learning with python: A guide for data scientists. sebastopol: Oâreilly media, incorporated, 2016. print.

[14] https://journals.icapsr.com/index.php/ijgasr/article/view/4/11. A study on data scaling methodsformachinelearningvinod sharmaresearch scholar, jiwaji university, gwalior.

[15] https://ieeexplore.ieee.org/abstract/document/4383455casatoken. Z. liu, j. almhana, r. mcgorman. "approximating lognormal sum distributions with power lognormal distributions", ieee transactions on vehicular technology, 2008.

[16] https://essex.primo.exlibrisgroup.com/discovery/. Beyad, y. maeder, m. (2013) multivariate linear regression with missing values. analytica chimica acta.

[17] https://essex.primo.exlibrisgroup.com/discovery/. Introduction to linear regression analysis douglas c. montgomery, elizabeth a. peck, g. geoffrey vining.

[18] https://essex.primo.exlibrisgroup.com/discovery/. Random forests leo breiman.

[19] http://www.ijrsset.org/pdfs/v7 i12/5.pdf. A scalable tree boosting system: Xg boost mounika nalluri1, mounika pentela1, nageswara rao eluri2.

[20] https://essex.primo.exlibrisgroup.com/discovery/. Alamri, s. khan, s. (2023) artificial intelligence based modelling for predicting co2 emission for climate change mitigation in saudi arabia. international journal of advanced computer science applications.

[21] https://www.sciencedirect.com/science/article/abs/pii/009830049390090Rvia3Dihub. Principal components analysis (pca)andrzej maÄkiewicz 1, waldemar ratajczak 2.

APPENDIX A

Python code

```
#importing the required library
import sys
assert sys.version_info >= (3, 5)
import numpy as np
import os
import tarfile
import urllib
import pandas as pd
import urllib.request
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error,
    ↪ r2_score
from tabulate import tabulate
```

```
#connect the dataset to google colab from google drive
from google.colab import drive
drive.mount('/content/gdrive', force_remount=True)


import os

GOOGLE_DRIVE_PATH_AFTER_MYDRIVE = os.path.join('./dissertation/') #
    ↪ setting the google drive path
GOOGLE_DRIVE_PATH = os.path.join('gdrive', 'MyDrive',
    ↪ GOOGLE_DRIVE_PATH_AFTER_MYDRIVE)
print('List␣files:␣', os.listdir(GOOGLE_DRIVE_PATH))


train = os.path.join(GOOGLE_DRIVE_PATH, 'CanadaCarEmissions.xlsx') # This
    ↪  is 100% of data

#read the data
data = pd.read_excel(train)
data.head()

#drop the first line which is showing 0 in each coloumns
data.drop(0)
data.info()

#changing the data types of some columns
data = data.dropna(subset=['CYLINDERS'])
data['CYLINDERS'] = data['CYLINDERS'].astype(int)

data = data.dropna(subset=['MODEL␣YEAR'])
data['MODEL␣YEAR'] = pd.to_datetime(data['MODEL␣YEAR'], format='%Y').dt.
    ↪ year

data = data.rename(columns={'MODEL(#␣=␣high␣output␣engine)': 'MODEL'})
```

```
data['MODEL'] = data['MODEL'].astype('category')


import pandas as pd
!pip install fancyimpute

from fancyimpute import IterativeImputer

#
import pandas as pd
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer

,
data = pd.DataFrame(data)

# Columns for MICE imputation
columns_to_impute = ['CO2 Rating', 'Smog Rating']

# Perform MICE imputation
imputer = IterativeImputer(max_iter=100, random_state=0)
data[columns_to_impute] = imputer.fit_transform(data[columns_to_impute])

data.head()



data['CO2 Rating'] = data['CO2 Rating'].round(2)
data['Smog Rating'] = data['Smog Rating'].round(2)

print(data)

data["VEHICLE CLASS"].value_counts()
```

```
y_make = data.groupby('MAKE').size()/len(data)
data['MAKE_NEW'] = data['MAKE'].apply(lambda x_make : y_make[x_make])
data.drop(['MAKE'], axis=1, inplace=True)


data['MAKE_NEW'].head()

y_model = data.groupby('MODEL').size()/len(data)
data['MODEL_NEW'] = data['MODEL'].apply(lambda x_model : y_model[x_model
    ↪ ])
data.drop(['MODEL'], axis=1, inplace=True)


data.head()

Categorical_df = data[['FUEL TYPE','VEHICLE CLASS','TRANSMISSION']]
Categorical_df.reset_index(inplace=True)
Categorical_df.drop(['index'], axis=1, inplace = True)
Categorical_df.head()


# code for one hot encoding
from sklearn.preprocessing import OneHotEncoder

encoder = OneHotEncoder(drop='first', sparse=False)

# Fit and transform the encoder
df_cat_encoded = encoder.fit_transform(Categorical_df)

# column names for the encoded features
encoded_columns = encoder.get_feature_names_out(input_features=
    ↪ Categorical_df.columns)
```

```
# Create a new DataFrame with the encoded categorical features and the
    ↪ column names
df_cat = pd.DataFrame(df_cat_encoded, columns=encoded_columns)
df_cat.head()


print(df_cat.shape)

df = pd.concat([data,df_cat], axis=1)
df.head()

df.drop(['VEHICLE CLASS','TRANSMISSION','FUEL TYPE'], axis =1, inplace =
    ↪ True)
df.dropna(inplace=True)

df.drop(['MODEL YEAR', 'FUEL CONSUMPTION CITY (L/100)', 'FUEL CONSUMPTION
    ↪  HWY (L/100)', 'COMB (mpg)', 'Smog Rating'], axis=1, inplace=True)

df.head()

modeldata = df.copy()
modeldata.drop(['CO2 Rating'], axis=1, inplace=True)
modeldata.head()

temp = modeldata[['CO2 EMISSIONS (g/km)']]
modeldata.drop(['CO2 EMISSIONS (g/km)'], axis=1, inplace=True)
modeldata = pd.concat([modeldata, temp], axis=1)
modeldata.head()

modeldata.info()

#split the data columns in x as input and y as output
X= modeldata.iloc[:,:69].values
y= modeldata.iloc[:,-1].values
```

```
#split␣the␣data␣into␣train␣and␣test
from␣sklearn.model_selection␣import␣train_test_split
X_train,␣X_test,␣y_train,␣y_test␣=␣train_test_split(X,␣y,␣test_size␣=␣0.2,
   ↪ ␣random_state␣=␣0)

#␣perform␣␣sandarded␣scaling␣on␣data
from␣sklearn.preprocessing␣import␣StandardScaler
sc␣=␣StandardScaler()
X_train␣=␣sc.fit_transform(X_train)
X_test␣=␣sc.transform(X_test)



print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)

#model␣1␣on␣plan␣data
from␣sklearn.linear_model␣import␣LinearRegression
regressor_mlr␣=␣LinearRegression()
regressor_mlr.fit(X_train,␣y_train)

from␣sklearn.model_selection␣import␣cross_val_score

#␣Calculate␣R-squared␣scores␣using␣cross-validation␣for␣training␣data
train_r2_scores␣=␣cross_val_score(regressor_mlr,␣X_train,␣y_train,␣cv=5,␣
   ↪ scoring='r2')
print("Training␣R-squared␣scores:",␣np.mean(train_r2_scores))

#␣Calculate␣R-squared␣scores␣using␣cross-validation␣for␣testing␣data
test_r2_scores␣=␣cross_val_score(regressor_mlr,␣X_test,␣y_test,␣cv=5,␣
   ↪ scoring='r2')
print("Testing␣R-squared␣scores:",␣np.mean(test_r2_scores))
```

```
#model1 trial1
from sklearn.ensemble import RandomForestRegressor
regressor_rf1 = RandomForestRegressor(n_estimators = 10, random_state =
    ↪ 0)
regressor_rf1.fit(X_train, y_train)


# Calculate R-squared scores using cross-validation for training data
train_r2_scores = cross_val_score(regressor_rf1, X_train, y_train, cv=5,
    ↪ scoring='r2')
print("Training R-squared scores:", np.mean(train_r2_scores))

# Calculate R-squared scores using cross-validation for testing data
test_r2_scores = cross_val_score(regressor_rf1, X_test, y_test, cv=5,
    ↪ scoring='r2')
print("Testing R-squared scores:", np.mean(test_r2_scores))

# XGB Regressor Model
import xgboost as xgb
xgb_model1 = XGBRegressor()
xgb_model1.fit(X_train, y_train)


# Calculate R-squared scores using cross-validation for training data
train_r2_scores = cross_val_score(xgb_model1, X_train, y_train, cv=5,
    ↪ scoring='r2')
print("Training R-squared scores:", np.mean(train_r2_scores))

# Calculate R-squared scores using cross-validation for testing data
test_r2_scores = cross_val_score(xgb_model1, X_test, y_test, cv=5,
    ↪ scoring='r2')
print("Testing R-squared scores:", np.mean(test_r2_scores))
```

```
# PCA
from sklearn. decomposition import PCA
pca = PCA (n_components=None)
X_train_pca1 = pca.fit_transform(X_train)
X_test_pca1 = pca.transform(X_test)
X_train_pca1.shape


#find_optimal_pca_components

from sklearn.decomposition import PCA
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_val_score

def find_optimal_pca_components(X_train, X_test, y_train, y_test,
    ↪ max_components=69):
    results = {'Num Components': [], 'Train R-squared': [], 'Test R-
    ↪ squared': []}

    for num_components in range(1, max_components + 1):
        pca = PCA(n_components=num_components)
        X_train_pca = pca.fit_transform(X_train)
        X_test_pca = pca.transform(X_test)

        regressor = LinearRegression()
        train_r2_scores = cross_val_score(regressor, X_train_pca, y_train,
        ↪  cv=5, scoring='r2')
        test_r2_scores = cross_val_score(regressor, X_test_pca, y_test,
        ↪ cv=5, scoring='r2')

        results['Num Components'].append(num_components)
        results['Train R-squared'].append(np.mean(train_r2_scores))
        results['Test R-squared'].append(np.mean(test_r2_scores))
```

```
    return results

# Example usage
optimal_pca_results = find_optimal_pca_components(X_train, X_test,
    ↪ y_train, y_test, max_components=69)

# Get the indices of the top 3 components with highest Test R-squared
    ↪ scores
top_3_train_indices = np.argsort(optimal_pca_results['Train R-squared'])
    ↪ [-3:]
# Get the indices of the top 3 components with highest Test R-squared
    ↪ scores
top_3_indices = np.argsort(optimal_pca_results['Test R-squared'])[-3:]

# Print the top 3 component indices and their scores
print("Top 3 Component Indices and Scores:")
for idx in top_3_indices:
    print(f"Component Index: {optimal_pca_results['Num Components'][idx]},
    ↪ Test R-squared: {optimal_pca_results['Test R-squared'][idx]},,
    ↪ Train R-squared: {optimal_pca_results['Train R-squared'][idx]}")

pca.explained_variance_ratio_
np.cumsum(pca.explained_variance_ratio_)

import matplotlib.pyplot as plt
import numpy as np



# Create a cumulative explained variance ratio plot
plt.plot(np.cumsum(pca.explained_variance_ratio_))

# Add a red horizontal line at y = 0.80
```

```
plt.axhline(y=0.80,␣color='red',␣linestyle='--')

#␣Find␣the␣index␣where␣cumulative␣explained␣variance␣ratio␣reaches␣0.90
cumulative_variance␣=␣np.cumsum(pca.explained_variance_ratio_)
index_90␣=␣np.argmax(cumulative_variance␣>=␣0.80)

#␣Add␣a␣red␣vertical␣line␣at␣the␣corresponding␣x-coordinate
plt.axvline(x=index_90,␣color='red',␣linestyle='--')

#␣Show␣the␣plot
plt.show()



#␣PCA
from␣sklearn.␣decomposition␣import␣PCA
pca␣=␣PCA␣(n_components=48)
X_train_pca7␣=␣pca.fit_transform(X_train)
X_test_pca7␣=␣pca.transform(X_test)
X_train_pca7.shape

#model␣2␣trial␣2␣final
from␣sklearn.linear_model␣import␣LinearRegression
regressor_mlr␣=␣LinearRegression()
regressor_mlr.fit(X_train_pca7,␣y_train)

from␣sklearn.model_selection␣import␣cross_val_score

#␣Calculate␣R-squared␣scores␣using␣cross-validation␣for␣training␣data
train_r2_scores␣=␣cross_val_score(regressor_mlr,␣X_train_pca7,␣y_train,␣
   ↪ cv=5,␣scoring='r2')
print("Training␣R-squared␣scores:",␣np.mean(train_r2_scores))

#␣Calculate␣R-squared␣scores␣using␣cross-validation␣for␣testing␣data
test_r2_scores␣=␣cross_val_score(regressor_mlr,␣X_test_pca7,␣y_test,␣cv=5,
```

```
   ↪ ␣scoring='r2')
print("Testing␣R-squared␣scores:",␣np.mean(test_r2_scores))



regressor_mlr.coef_


#␣Make␣predictions␣on␣the␣training␣data
y_pred_test␣=␣regressor_mlr.predict(X_test_pca7)

#␣Calculate␣evaluation␣metrics
mae_test␣=␣mean_absolute_error(y_test,␣y_pred_test)
mse_test␣=␣mean_squared_error(y_test,␣y_pred_test)
rmse_test␣=␣np.sqrt(mse_test)


#␣Print␣the␣evaluation␣metrics
print("Mean␣Absolute␣Error␣(MAE)␣on␣training␣data:",␣mae_test)
print("Mean␣Squared␣Error␣(MSE)␣on␣training␣data:",␣mse_test)
print("Root␣Mean␣Squared␣Error␣(RMSE)␣on␣training␣data:",␣rmse_test)


#␣Calculate␣R-squared␣scores␣using␣cross-validation␣for␣training␣data
test_r2_scores␣=␣cross_val_score(regressor_mlr,␣X_test_pca7,␣y_test,␣cv=5,
   ↪ ␣scoring='r2')
print("Cross-validated␣R-squared␣scores␣on␣training␣data:",␣np.mean(
   ↪ train_r2_scores))


y_pred␣=␣regressor_mlr.predict(X_test_pca7)


import␣matplotlib.pyplot␣as␣plt
```

```
import␣numpy␣as␣np


#␣Create␣a␣scatter␣plot␣of␣the␣actual␣vs.␣predicted␣values
plt.figure(figsize=(8,␣6))
plt.scatter(y_test,␣y_pred,␣c='blue',␣marker='o',␣label='Actual vs.
    ↪ Predicted')

#␣Plot␣a␣diagonal␣line␣to␣represent␣perfect␣predictions
plt.plot([min(y_test),␣max(y_test)],␣[min(y_test),␣max(y_test)],␣c='red',
    ↪ ␣linestyle='--',␣label='Perfect Predictions')

plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values (Linear Regression Model)')
plt.legend()
plt.grid(True)
plt.show()


#model␣2␣Random␣forest␣regrssion
from␣sklearn.ensemble␣import␣RandomForestRegressor
regressor_rf7␣=␣RandomForestRegressor(n_estimators␣=␣10,␣random_state␣=␣
    ↪ 0)
regressor_rf7.fit(X_train_pca7,␣y_train)

#␣Calculate␣R-squared␣scores␣using␣cross-validation␣for␣training␣data
train_r2_scores␣=␣cross_val_score(regressor_rf7,␣X_train_pca7,␣y_train,␣
    ↪ cv=5,␣scoring='r2')
print("Training␣R-squared␣scores:",␣np.mean(train_r2_scores))

#␣Calculate␣R-squared␣scores␣using␣cross-validation␣for␣testing␣data
test_r2_scores␣=␣cross_val_score(regressor_rf7,␣X_test_pca7,␣y_test,␣cv=5,
    ↪ ␣scoring='r2')
```

```
print("Testing␣R-squared␣scores:",␣np.mean(test_r2_scores))


#␣Make␣predictions␣on␣the␣training␣data
y_pred_test␣=␣regressor_rf7.predict(X_test_pca7)
#␣Calculate␣evaluation␣metrics
mae_test␣=␣mean_absolute_error(y_test,␣y_pred_test)
mse_test␣=␣mean_squared_error(y_test,␣y_pred_test)
rmse_test␣=␣np.sqrt(mse_test)

#␣Print␣the␣evaluation␣metrics
print("Mean␣Absolute␣Error␣(MAE)␣on␣testing␣data:",␣mae_test)
print("Mean␣Squared␣Error␣(MSE)␣on␣testing␣data:",␣mse_test)
print("Root␣Mean␣Squared␣Error␣(RMSE)␣on␣testing␣data:",␣rmse_test)



#␣XGB␣Regressor␣Model
import␣xgboost␣as␣xgb
xgb_model7␣=␣XGBRegressor()
xgb_model7.fit(X_train_pca7,␣y_train)


#␣Calculate␣R-squared␣scores␣using␣cross-validation␣for␣training␣data
train_r2_scores␣=␣cross_val_score(xgb_model7,␣X_train_pca7,␣y_train,␣cv=5,
   ↪ ␣scoring='r2')
print("Training␣R-squared␣scores:",␣np.mean(train_r2_scores))

#␣Calculate␣R-squared␣scores␣using␣cross-validation␣for␣testing␣data
test_r2_scores␣=␣cross_val_score(xgb_model7,␣X_test_pca7,␣y_test,␣cv=5,␣
   ↪ scoring='r2')
print("Testing␣R-squared␣scores:",␣np.mean(test_r2_scores))

#␣Make␣predictions␣on␣the␣training␣data
```

```
y_pred_train = xgb_model7.predict(X_train_pca7)

# Calculate evaluation metrics
mae_test = mean_absolute_error(y_test, y_pred_test)
mse_test = mean_squared_error(y_test, y_pred_test)
rmse_test = np.sqrt(mse_test)

# Print the evaluation metrics
print("Mean Absolute Error (MAE) on testing data:", mae_test)
print("Mean Squared Error (MSE) on testing data:", mse_test)
print("Root Mean Squared Error (RMSE) on testing data:", rmse_test)



# QQ Plot for each model
import statsmodels.api as sm
def plot_qq_plot(model, X, y, title):
    y_pred = model.predict(X)
    residuals = y - y_pred
    sm.qqplot(residuals, line='s')
    plt.title(title)
    plt.show()

plot_qq_plot(regressor_mlr, X_test_pca7, y_test, 'QQ Plot for Linear
    ↪ Regression')
plot_qq_plot(regressor_rf7, X_test_pca7, y_test, 'QQ Plot for
    ↪ RandomForest Regressor')
plot_qq_plot(xgb_model7, X_test_pca7, y_test, 'QQ Plot for XGB Regressor'
    ↪ )
```

APPENDIX B

R code

```
library(readxl)
df <- read_excel("~/Desktop/karoyamaro/canada_emissions_trial/
   ↪ imputed_data_0.xlsx")
View(df)
head(df)
# Rename the column names
colnames(df) <- c("MODEL_YEAR", "COMPANY_NAME", "MODEL_NAME","
   ↪ VEHICLE_CLASS", "ENGINE_SIZE","CYLINDERS","TRANSMISSION","FUEL_TYPE
   ↪ ","FUEL_CONSUMPTION_CITY","FUEL_CONSUMPTION_HWY","COMB_km","
   ↪ COMB_mpg","CO2_EMISSIONS","CO2_Rating","Smog_Rating")

# Check the data types of all columns
data_types <- sapply(df, class)
print(data_types)

# Convert specific columns to numeric data type
df$ENGINE_SIZE <- as.numeric(df$ENGINE_SIZE)
df$CYLINDERS <- as.numeric(df$CYLINDERS)
df$FUEL_CONSUMPTION_CITY <- as.numeric(df$FUEL_CONSUMPTION_CITY)
df$FUEL_CONSUMPTION_HWY <- as.numeric(df$FUEL_CONSUMPTION_HWY)
```

```
df$COMB_km <- as.numeric(df$COMB_km)
df$COMB_mpg <- as.numeric(df$COMB_mpg)
df$CO2_EMISSIONS <- as.numeric(df$CO2_EMISSIONS)
df$CO2_Rating <- as.numeric(df$CO2_Rating)
df$Smog_Rating <- as.numeric(df$Smog_Rating)

# Verify the updated data types
data_types <- sapply(df, class)
print(data_types)



library(lubridate)



duplicates <- duplicated(df)
df <- subset(df, !duplicates)

missing_values <- is.na(df)
df <- df[complete.cases(df), ]

library(dplyr)


# Count the levels in a column
column_counts <- table(df$VEHICLE_CLASS)
column_counts

df$VEHICLE_CLASS <- gsub("Compact","COMPACT", df$VEHICLE_CLASS, ignore.
   ↪ case = TRUE)
df$VEHICLE_CLASS <- gsub("Full-size", "FULL-SIZE", df$VEHICLE_CLASS,
   ↪ ignore.case = TRUE)
df$VEHICLE_CLASS <- gsub("Mid-size","MID-SIZE", df$VEHICLE_CLASS, ignore.
```

```
   ↪ case = TRUE)
df$VEHICLE_CLASS <- gsub("MiniCompact", "MINICOMPACT", df$VEHICLE_CLASS,
   ↪ ignore.case = TRUE)
df$VEHICLE_CLASS <- gsub("Minivan", "MINIVAN", df$VEHICLE_CLASS, ignore.
   ↪ case = TRUE)
df$VEHICLE_CLASS <- gsub("Special purpose vehicle", "SPECIAL PURPOSE
   ↪ VEHICLE", df$VEHICLE_CLASS, ignore.case = TRUE)
df$VEHICLE_CLASS <- gsub("Station wagon: Small", "STATION WAGON - SMALL",
   ↪  df$VEHICLE_CLASS, ignore.case = TRUE)
df$VEHICLE_CLASS <- gsub("Station wagon: Mid-size", "STATION WAGON - Mid-
   ↪ size", df$VEHICLE_CLASS, ignore.case = TRUE)
df$VEHICLE_CLASS <- gsub("STATION WAGON - Mid-size", "STATION WAGON - MID-
   ↪ SIZE", df$VEHICLE_CLASS, ignore.case = TRUE)
df$VEHICLE_CLASS <- gsub("SUV: Small", "SUV - SMALL", df$VEHICLE_CLASS,
   ↪ ignore.case = TRUE)
df$VEHICLE_CLASS <- gsub("Two-seater", "TWO-SEATER", df$VEHICLE_CLASS,
   ↪ ignore.case = TRUE)
df$VEHICLE_CLASS <- gsub("Van: Passenger", "VAN - PASSENGER",
   ↪ df$VEHICLE_CLASS, ignore.case = TRUE)
df$VEHICLE_CLASS <- gsub("Pickup truck: Standard","PICKUP TRUCK -
   ↪ STANDARD", df$VEHICLE_CLASS, ignore.case = TRUE)
df$VEHICLE_CLASS <- gsub( "Pickup truck: Small","PICKUP TRUCK - SMALL",
   ↪ df$VEHICLE_CLASS, ignore.case = TRUE)
df$VEHICLE_CLASS <- gsub("SubCompact", "SUBCOMPACT", df$VEHICLE_CLASS,
   ↪ ignore.case = TRUE)
table(df$VEHICLE_CLASS )

head(df)


df$COMPANY_NAME <- recode(df$COMPANY_NAME,
                     "ACURA" = "Acura",
                     "ALFA ROMEO" = "Alfa Romeo",
                     "ASTON MARTIN" = "Aston Martin",
```

```
"AUDI" = "Audi",
"BENTLEY" = "Bentley",
"BUGATTI" = "Bugatti",
"BUICK" = "Buick",
"CADILLAC" = "Cadillac",
"CHEVROLET" = "Chevrolet",
"CHRYSLER" = "Chrysler",
"DODGE" = "Dodge",
"FERRARI" = "Ferrari",
"FIAT" = "Fiat",
"FORD" = "Ford",
"GENESIS" = "Genesis",
"GEO" = "Geo",
"GMC" = "GMC",
"HONDA" = "Honda",
"HUMMER" = "Hummer",
"HYUNDAI" = "Hyundai",
"INFINITI" = "Infiniti",
"ISUZU" = "Isuzu",
"JAGUAR" = "Jaguar",
"JEEP" = "Jeep",
"KIA" = "Kia",
"LAMBORGHINI" = "Lamborghini",
"LAND ROVER" = "Land Rover",
"LEXUS" = "Lexus",
"LINCOLN" = "Lincoln",
"MASERATI" = "Maserati",
"MAZDA" = "Mazda",
"MERCEDES-BENZ" = "Mercedes-Benz",
"MERCURY" = "Mercury",
"MINI" = "Mini",
"MITSUBISHI" = "Mitsubishi",
"NISSAN" = "Nissan",
"OLDSMOBILE" = "Oldsmobile",
```

```
                    "PLYMOUTH" = "Plymouth",
                    "PONTIAC" = "Pontiac",
                    "PORSCHE" = "Porsche",
                    "RAM" = "Ram",
                    "ROLLS-ROYCE" = "Rolls-Royce",
                    "SAAB" = "Saab",
                    "SATURN" = "Saturn",
                    "SCION" = "Scion",
                    "SMART" = "Smart",
                    "SRT" = "Srt",
                    "SUBARU" = "Subaru",
                    "SUZUKI" = "Suzuki",
                    "TOYOTA" = "Toyota",
                    "VOLKSWAGEN" = "Volkswagen",
                    "VOLVO" = "Volvo"
)

# Check the updated company names
table(df$COMPANY_NAME)



head(df)
colnames(df)
head(df)
class(df)
################################################################################
   ↪


# Required packages
library(ggplot2)
library(dplyr)
library(tidyr)
```

```
# Get the numerical columns in the dataset
numeric_columns <- df %>% select_if(is.numeric)

summary(numeric_columns)

# Reshape the dataset to long format
df_long <- numeric_columns %>%
  tidyr::gather(key = "Column", value = "Value")

# Calculate median and mean values for each variable
summary_values <- df_long %>%
  group_by(Column) %>%
  summarize(Median = median(Value), Mean = mean(Value))

# Plotting the distribution curves with median and mean lines
ggplot(df_long, aes(x = Value, fill = Column)) +
  geom_density(alpha = 0.5) +
  facet_wrap(~ Column, scales = "free") +
  labs(x = "Value", y = "Density") +
  ggtitle("Distribution Curves of Variables") +
  geom_vline(data = summary_values, aes(xintercept = Median),
             color = "red", linetype = "dotted", size = 1) +
  geom_vline(data = summary_values, aes(xintercept = Mean),
             color = "yellow", linetype = "dashed", size = 1) +
  theme_bw() +
  theme(legend.position = "none") # Remove the legend from the graph



# Calculate correlation matrix
cor_matrix <- cor(numeric_columns)
```

```
# Plot correlation matrix with half size, numbers, and circles
corrplot::corrplot(cor_matrix, method = "circle", type = "upper",
                   tl.cex = 0.8, tl.col = "black",
                   addCoef.col = "black", number.cex = 0.7,
                   pch = 21, bg = "white")


# Define a new color palette
new_color_palette <- colorRampPalette(c("yellow", "white", "red3"))(80)

# Plot correlation matrix with modified color scheme
corrplot::corrplot(cor_matrix, method = "square", type = "upper",
                   tl.cex = 0.8, tl.col = "black",
                   addCoef.col = "black", number.cex = 0.7,
                   pch = 21, bg = "white",
                   col = new_color_palette) # Specify the new color palette

##############################################################################
    ↪

names(df)
df_s<-df[,9:13]

plot(df_s)


names(df)

# Group the data by model year and calculate the sum of CO2 emissions
# Convert CO2_EMISSIONS column to numeric data type
df$CO2_EMISSIONS <- as.numeric(df$CO2_EMISSIONS)

# Group the data by model year and calculate the sum of CO2 emissions
```

```
co2_emissions <- df %>%
  group_by(MODEL_YEAR) %>%
  summarise(total_co2 = sum(CO2_EMISSIONS))

# Print the resulting data
print(co2_emissions)
View(co2_emissions)


# Convert the 'MODEL YEAR' column to numeric
co2_emissions$`MODEL_YEAR` <- as.numeric(as.character(co2_emissions$`
   ↪ MODEL_YEAR`))


library(ggplot2)

# Create a line graph
ggplot(co2_emissions, aes(x = MODEL_YEAR, y = total_co2)) +
  geom_line(color = "pink") +
  geom_point(size = 1, color = "red") +
  geom_text(aes(label = `MODEL_YEAR`), nudge_x =0.5, color = "grey2") +
  xlab("Model Year") +
  ylab("Total CO2 Emissions") +
  ggtitle("CO2 Emissions Over the Years") +
  theme_bw()


# Calculate average fuel consumption per year for city and highway
avg_fuel_data <- aggregate(cbind(FUEL_CONSUMPTION_CITY,
   ↪ FUEL_CONSUMPTION_HWY) ~ MODEL_YEAR, df, mean)

# Create the plot
ggplot(avg_fuel_data, aes(x = MODEL_YEAR)) +
  geom_line(aes(y = FUEL_CONSUMPTION_CITY, color = "City")) +
```

```
  geom_line(aes(y = FUEL_CONSUMPTION_HWY, color = "Highway")) +
  labs(title = "Average Fuel Consumption Over the Years",
       x = "Model Year",
       y = "Average Fuel Consumption") +
  scale_color_manual(values = c("City" = "green3", "Highway" = "red")) +
  theme_minimal() +
  scale_x_continuous(breaks = seq(min(avg_fuel_data$MODEL_YEAR), max(
     ↪ avg_fuel_data$MODEL_YEAR), by = 1))+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))



# Required Libraries
library(ggplot2)


table(df$FUEL_TYPE)
w<- df %>%select(FUEL_TYPE,CO2_EMISSIONS) %>% group_by(FUEL_TYPE)
summary(w)

# Required Libraries
library(ggplot2)

# Create the plot
ggplot(df, aes(x = FUEL_TYPE, y = CO2_EMISSIONS, fill = FUEL_TYPE)) +
  geom_violin(trim = FALSE) +
  labs(title = "Distribution of CO2 Emissions by Fuel Type",
       x = "Fuel Type",
       y = "CO2 Emissions (g/km)") +
  scale_fill_manual(values = c("D" = "red", "E" = "green", "N" = "blue", "
     ↪ X" = "orange", "Z" = "purple")) +
  theme_minimal()

# Load required libraries
```

```
library(ggplot2)

# Calculate mean CO2 emissions by vehicle class
mean_co2 <- aggregate(CO2_EMISSIONS ~ VEHICLE_CLASS, df, mean)
mean_co2 <- mean_co2[order(mean_co2$CO2_EMISSIONS, decreasing = TRUE), ]


# Create the box plot with angled x-axis labels and a different theme
ggplot(df, aes(x = factor(VEHICLE_CLASS, levels = mean_co2$VEHICLE_CLASS),
   ↪  y = CO2_EMISSIONS, fill = VEHICLE_CLASS)) +
  geom_boxplot() +
  labs(title = "CO2 Emissions by Vehicle Class",
       x = "Vehicle Class",
       y = "CO2 Emissions") +
  theme_gray() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+theme(legend.
     ↪ position = "none")

# Arrange levels of TRANSMISSION variable in descending order
df$TRANSMISSION <- factor(df$TRANSMISSION, levels = rev(unique(
   ↪ df$TRANSMISSION)))
table(df$TRANSMISSION)

# Calculate the average CO2 emission per transmission type
avg_co2_by_transmission <- df %>%
  group_by(TRANSMISSION) %>%
  summarize(avg_CO2_Emission = mean(CO2_EMISSIONS))

View(avg_co2_by_transmission)
# Create the boxplot with descending order of TRANSMISSION
ggplot(df, aes(x = reorder(TRANSMISSION, -CO2_EMISSIONS), y =
   ↪ CO2_EMISSIONS, fill = TRANSMISSION)) +
  geom_boxplot() +
  labs(x = "Transmission", y = "CO2 Emissions (g/km)") +
```

```
  ggtitle("CO2 Emissions Distribution by Transmission") +
  theme_gray()+theme(legend.position = "none")

# Recode the 'TRANSMISSION' column into three categories based on
   ↪ specific mappings
df1 <- df1 %>%
  mutate(TRANSMISSION_TYPE = recode(TRANSMISSION,
                                    "A4" = "Automatic",
                                    "M5" = "Manual",
                                    "AS9" = "Semi-Automatic",
                                    # Add other mappings for all unique
                                       ↪ transmission types
                                    "AV" = "Automatic",
                                    "A6" = "Automatic",
                                    "AM6" = "Semi-Automatic",
                                    "A7" = "Automatic",
                                    "AM7" = "Semi-Automatic",
                                    "AS7" = "Automatic",
                                    "AS8" = "Automatic",
                                    "A8" = "Automatic",
                                    "M7" = "Manual",
                                    "AV7" = "Automatic",
                                    "AV8" = "Automatic",
                                    "AV6" = "Automatic",
                                    "AM5" = "Semi-Automatic",
                                    "A9" = "Automatic",
                                    "AS9" = "Semi-Automatic",
                                    "AM8" = "Semi-Automatic",
                                    "AM9" = "Semi-Automatic",
                                    "AS10" = "Automatic",
                                    "A10" = "Automatic",
                                    "AV10" = "Automatic",
                                    "AV1" = "Automatic",
                                    "A3" = "Automatic",
```

```
                              "A5" = "Automatic",
                              "AS4" = "Semi-Automatic",
                              "AS5" = "Semi-Automatic",
                              "AS6" = "Semi-Automatic",
                              "AS4" = "Semi-Automatic",
                              "M4" = "Manual",
                              "M6" = "Manual"
  ))

# View the updated dataset with the new column
table(df1$TRANSMISSION_TYPE)

names(df1)

# Create the boxplot
x<-ggplot(df1, aes(x = TRANSMISSION_TYPE, y = CO2_EMISSIONS)) +
  geom_boxplot(fill = "purple", color = "black") +
  labs(title = "CO2 Emissions ",
       x = "Transmission Type",
       y = "CO2 Emissions")

# Create the boxplot
y<-ggplot(df1, aes(x = TRANSMISSION_TYPE, y = FUEL_CONSUMPTION_CITY)) +
  geom_boxplot(fill = "orange", color = "black") +
  labs(title = "City",
       x = "Transmission Type",
       y = "Fuel consumption ")

# Create the boxplot
z<-ggplot(df1, aes(x = TRANSMISSION_TYPE, y = FUEL_CONSUMPTION_HWY)) +
  geom_boxplot(fill = "green", color = "black") +
  labs(title = "Highway",
       x = "Transmission Type",
       y = "Fuel consumption")
```

```
library(gridExtra)

library(gridExtra)
library(cowplot) # Make sure cowplot is installed and loaded

# Assuming x, y, and z are your plots
combined_plot <- grid.arrange(x, y, z, ncol = 2)

# Add title using cowplot
combined_plot_with_title <- ggdraw() +
  draw_plot(combined_plot) +
  draw_label("Comparative study \n of featured trasmission system", x =
    ↪ 0.75, y = 0.3, hjust = 0.5, vjust = 0, fontface = 'bold', size =
    ↪ 12)

# Print the combined plot with the title
print(combined_plot_with_title)




# Graph - Bar plot for Research Question 10
top_performing <- df %>%
  group_by(VEHICLE_CLASS) %>%
  top_n(-1, CO2_EMISSIONS)
View(top_performing[,c(2,4,13)])
ggplot(top_performing, aes(x = COMPANY_NAME, y = CO2_EMISSIONS, fill =
   ↪ VEHICLE_CLASS)) +
  geom_bar(stat = "summary", fun = "mean", position = "dodge") +
  labs(x = "Company names", y = "Mean CO2 Emissions") +
  ggtitle("Top-Performing Vehicles")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
```

```
##############################################################################################
   ↪
##############################################################################################
   ↪


#hypothesis for fuel consuption:

#1.a Welch Two Sample t-test

t.test(df$FUEL_CONSUMPTION_CITY, df$FUEL_CONSUMPTION_HWY, paired = FALSE)



#2.# One-way ANOVA on "FUEL_CONSUMPTION_CITY" among different fuel types
fit <- aov(CO2_EMISSIONS ~ FUEL_TYPE, data = df)
summary(fit)
# Post-hoc tests for "FUEL_CONSUMPTION_CITY"
posthoc<- TukeyHSD(fit)
posthoc
plot(posthoc)


#3.# One-way ANOVA on "Combined" among different fuel types
fit_city <- aov(COMB_km ~ FUEL_TYPE, data = df)
summary(fit_city)
# Post-hoc tests for "FUEL_CONSUMPTION_CITY"
posthoc_city <- TukeyHSD(fit_city)
posthoc_city
plot(posthoc_city)
```