BSD2333 DATA WRANGLING 2022/2023
SEMESTER 2

PROJECT TITLE:
AUTOHUB CAR DEALER

PREPARED FOR:
MOHD KHAIRUL BAZLI BIN MOHD AZIZ

PREPARED BY:

MATRIC ID	NAME	SECTION
SD22003	NUR ATIEKA RAFIEKAH BINTI RAZAK	01G
SD22047	NURUL FAQIHAH BINTI MAZLI AMRAN	01G
SD22066	SHAHIRA BINTI MOHAIDEEN MEERA	01G
SD22043	HAWA HUMAIRA BINTI HAMUZAN	01G
SD22032	NUR A'RIFAH AKMAL BINTI HUSSIN	01G

TABLE OF CONTENT

1.0 INTRODUCTION

The automotive industry has regularly been a subject of research and analysis, given its crucial role in transportation and economic development. Within this industry, the market for pre-owned, first-owner, or second-owner automobiles has received a lot of attention, as buyers look for more affordable and environmentally friendly alternatives (Kurko, nd).

Ownership, costs, consistency, and off-road capabilities are all important considerations when purchasing a car. Furthermore, the availability of detailed information about car characteristics, condition, and pricing has enabled prospective purchasers to make educated judgments(Mazda digital, n.d.).

By using the given information, which includes parameters such as vehicle condition, mileage, seller ratings, interior condition, drivetrain, efficiency of fuel, engine size, and ownership history, this study aims to identify significant insights about the structure of the transportation market. An analysis of these aspects may offer insight into customer preferences, price patterns, and the influence of various car attributes on demand within this specific sector of the automobile business.

1.1 Description Assignment

Consumers aim to make decisions such as what to purchase which can either be a new car or a second-hand car, which brand to go for. What this means is that even when there is a difference in price where you expect to get value for your money in the used automobile than in a new one, this is not true (Koons Mazda Silver Spring, n. d.). Likewise, owning a new car may be wonderful, but like the effect of depreciation which set in even faster than the appreciation of real estate values (Feldman & Smith, 2019).

Still, if one can find a used automobile, whose previous owner has a good maintenance record, and the car is in reasonable condition, they are a good investment (Koons Mazda Silver Spring, n. d.). Perhaps, it might be difficult to diagnose such a problem from the normal naked-eyed, superficial check but to owners' relief, PUSPAKOM stipulates compulsory fitness on tests on the name changes (Kurnia Insurans, 2024). This pushes the former owner to ensure he assesses the car well before passing it to the new owner. Used car also sometimes carry a warranty. arguably the most common question asked when buying a used automobile is on warranties. Many used automobiles today are still under the manufacturer warranties (Kurnia Insurans, 2024). Most recipients of auto leases return their automobiles after three years of use, while the warranty of the car still has two years on its five year guarantee. Most of the used automobiles and Certified Pre-Owned Vehicles on the lot have been tested and in case you bring them in, they will be under warranty (Koons Mazda Silver Spring, n. d.).

However, new cars also have their advantages. These include the following: As much as new cars can be costly, they come with a lot of advantages as described below. If one acquires a new car, then it is a new automobile whereby one purchases a new car (Sales Advisor, 2022). A new car may not be treated roughly and this ensures that there are minimal instances of damaging crucial items (2022). Owing to this, one may never be so certain about the kind of treatment the previous owner gave to the car. They probably had worn out the clutch or applied too much pressure on the brakes and accelerator (Sales Advisor, 2022). You may spend more money than required on purchasing the new automobile instead of replacing compartments in a second-hand one (written communication, February 11, 2022).

The purpose of this project is to assist the one with the decision, by evaluating and comparing the brand performance of the car and its MPg, the mileage, and the condition of the car. This will assist the customer make a decision to arrive at while having the best experience as discussed either when purchasing their first car from their estimated budget or from the experience that they assumed they would gain.

1.2 Problem Statement

A car dealership in Malaysia, "AutoHub", has been experiencing stagnant sales for the past few quarters. They have a diverse inventory of new and used cars from various brands and production years. To improve their sales strategy, they decide to conduct a comprehensive analysis of their car sales data.

The main objective of this case study is to identify patterns and trends in the car sales data that could help AutoHub enhance its sales and marketing strategies. The dealership aims to understand the factors that influence a customer's car buying decision and optimize their inventory accordingly. There are a few approaches that are used by their sales and marketing teams;

1. Car Condition Analysis: AutoHub will compare the condition and pricing of used and new cars in their inventory. They will also evaluate the reliability and longevity of different car brands based on their condition and historical maintenance records.

2. Production Year Analysis: AutoHub will examine the impact of the production year on the desirability and market value of the cars. They will analyze the depreciation rates and demand for vintage cars compared to newer models.

3. Price Analysis: AutoHub will investigate the distribution of car prices across different brands, conditions, and production years. They will also study the relationship between car prices and mileage.

4. Fuel Efficiency and Environmental Impact: AutoHub will analyze the fuel efficiency ratings of cars across different fuel types and drivetrains. They will also assess the environmental impact of different car models based on their fuel type and engine size.

By conducting this case study, AutoHub expects to gain valuable insights that will help them optimize their sales and marketing strategies, improve customer satisfaction, and ultimately increase their sales.

1.3 Question to be answered

1. How can data wrangling techniques be utilized to clean and preprocess the car sales dataset effectively, ensuring accurate analysis?

2. What are the key factors influencing customer purchasing decisions in the automotive industry, and how can these insights be leveraged to inform AutoHub's sales and marketing strategies?

3. Based on the insights derived from the data analysis, what strategies can AutoHub implement to enhance customer satisfaction, improve sales performance, and drive business growth?

1.4 Objectives

1. To apply data wrangling techniques using Python to clean and process.

2. To analyze and extract meaningful insights from the data to inform AutoHub's sales and marketing strategies.

3. To propose strategies to enhance customer satisfaction, improve sales performance, and drive business growth.

1.5 Description of Data

This dataset provides comprehensive information about various aspects of vehicles listed on the online marketplace, allowing for detailed analysis and comparison for potential buyers.

VARIABLES	DATA TYPE	DESCRIPTION
Make	String	The brand or manufacturer of the vehicle
Year	Integer	The manufacturing year of the vehicle.
Condition	String	The condition of the vehicle
Mileage	Integer	The total distance the vehicle has traveled, measured in miles.
Price	Integer	The listed price of the vehicle.
Seller Rating	Float	The rating of the seller is based on customer feedback or reviews.
Seller Rating Count	Integer	The number of ratings or reviews the seller has received
Interior Color	String	The color of the interior of the vehicle.
Drivetrain	String	The type of drivetrain system of the vehicle.
Min MPG	Float	The minimum miles per gallon (MPG) rating of the vehicle.
Max MPG	Float	The maximum miles per gallon (MPG) rating of the vehicle.
Fuel Type	String	The type of fuel the vehicle uses.
Engine Size (L)	Float	The displacement volume of the engine is measured in

		liters.
Accidents or damage	Boolean	Indicates whether the vehicle has been involved in accidents or has any damage history.
1-owner vehicle	Boolean	Indicates whether the vehicle has had only one owner.
Personal use only	Boolean	Indicates whether the vehicle has been used for personal purposes only.
Clean title	Boolean	Indicates whether the vehicle has a clean title without any liens or legal issues.
Open Recall	Boolean	Indicates whether there are any open recalls or safety issues associated with the vehicle.
Comfort_score	Integer	A rating or score indicating the comfort level of the vehicle.
Interior_score	Integer	A rating or score indicating the overall quality or condition of the vehicle's interior.
Performance_score	Integer	A rating or score indicating the performance capabilities of the vehicle.
Value_score	Integer	A rating or score indicating the overall value proposition of the vehicle.
Exterior score	Integer	A rating or score indicating the overall appearance or condition of the vehicle's exterior.
Reliability_score	Integer	A rating or score indicating the reliability or durability of the vehicle.

Table 1: Description of datasets

2.0 PACKAGE REQUIRED

1. Matplotlib

- Description: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations

```
import matplotlib.pyplot as plt
```

2. Numpy

- Description: NumPy is the fundamental package for scientific computing in Python. It provides support for arrays, mathematical functions, and linear algebra operations.

```
import numpy as np
```

3. Pandas

- Description: Pandas is a powerful and flexible data manipulation library that provides data structures such as DataFrames, which are essential for data wrangling tasks.

```
import pandas as pd
```

4. SciPy

- Description: SciPy is a Python library used for scientific and technical computing. It builds on NumPy and provides a large number of higher-level functions that operate on NumPy arrays.

```
from scipy import stats
```

5. Seaborn

- Description: Seaborn is a statistical data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics

```
import seaborn as sns
```

6. Plotly Express

- Description: Used for creating interactive visualizations easily.

```
import plotly.express as px
```

7. ipywidgets

- Description: Used for creating interactive widgets.

```
from ipywidgets import interact, widgets
```

These packages collectively enable efficient data wrangling, analysis, and visualization, facilitating the extraction of meaningful insights and the development of actionable strategies.

3.0 DATA PREPARATION

Figure 1: Flowchart of Data Preparation

1. **Start**
    - This is the initiation of the data preparation process.

2. **Import data & library**
    - In this step, the car dataset and necessary libraries are imported. Libraries include pandas for data manipulation, numpy for numerical operations, and matplotlib or seaborn for data visualization.

3. **Data understanding**
    - Here, we will explore the dataset to understand its structure and contents. This includes inspecting the data types, checking for missing values, and getting a sense of the data distribution.

4. **Data cleaning**
    - This step involves cleaning the data to ensure it is in a suitable format for analysis. It might include handling missing values, removing duplicates, correcting data types, and dealing with outliers.

5. **Data preview**
    - Once the data is cleaned, a preview of the dataset is obtained to ensure the cleaning process was successful. This might involve displaying the first few rows using df,head () and a summary of the dataset to verify changes.

6. **Data Description**
    - In this step, the data is described in more detail. This includes generating summary statistics, exploring categorical features, and understanding the relationships between different variables.

7. **Data Visualization**
    - Finally, data visualization is used to represent the data graphically. This helps in identifying patterns, trends, and insights that are not easily noticeable in raw data. Tools like matplotlib, seaborn, and plotly can be used to create various charts and graphs.

8. **End**
    - This marks the end of the data preparation process. By this point, this data should be well-understood, clean, and ready for further analysis or modeling.

3.1 Data Import

1. Data Source:

The dataset used in this assignment is provided from Kaggle and contains comprehensive car sales data, including information on car brands, models, production years, mileages, conditions (new or used), prices, fuel efficiency, maintenance history, and seller information. The data is stored in a CSV file named car_sales.csv.

2. Importing the Data:

To import the data into Python for analysis, we use the Pandas library, which is well-suited for handling tabular data. The pd.read_csv function is used to read the CSV file and load its contents into a Pandas DataFrame named df.

```
[2]  import pandas as pd
```

Pandas Library: The Pandas library is essential for data manipulation and analysis, providing powerful tools to load, inspect, and process data.

```
#Load the data
car = pd.read_csv("/content/Car_sales.csv")
```

CSV Format: The CSV (Comma Separated Values) format is commonly used for storing tabular data and is easy to read and write with Pandas.

3. Initial Data Inspection:

After importing the data, apply inspection data and places it in the database.It is essential to inspect the initial few rows and gather summary information about the DataFrame to ensure the data has been loaded correctly and to understand its structure (Maushart, 2023). This involves using the head() method to display the first five rows and the info() method to get an overview of the DataFrame, including the number of entries, column names, data types, and memory usage.

head() Method: This method displays the first five rows of the DataFrame, allowing you to verify that the data has been loaded correctly and to get a preliminary understanding of its structure.

```
[ ]  car.head()
```

3.2 Data Understanding

Data understanding is important to get an overall comprehensive understanding the data. We begin with a quick summary or description of the dataset. We check the number of rows and columns using shape. We can observe there are 9016 rows and 24 columns in total, and display all column names. Then we explore the data type of each column to ensure they are appropriate for analysis. Attributes make, condition, interior color, drivetrain, fuel type, 1-owner vehicles, personal use only, and clean title are objects or string data types. Meanwhile, year, price, and open recall are integers and the rest 13 attributes are float data types.

Additional information about the dataset is obtained using the .info() method, which provides a summary of the data frame consist the column name nonnull count and the datatype. For the categorical attributes, we identify the unique values and count each of them. This step is crucial to understand the distribution of the categorical data within the dataset. This step is to ensure the dataset is well prepared and we understand our data better by having the base for each attribute.

3.3 Data Cleaning

1. Count Total Missing value in each column

```
# Checking for missing values in each column
print(car.isnull().sum())
```

```
Make                    0
Year                    0
Condition               0
Mileage               149
Price                   0
Seller Rating        1355
Seller Rating Count  1355
Interior color          0
Drivetrain              0
Min MPG              1595
Max MPG              1595
Fuel type             575
Engine Size (L)       765
Accidents or damage  3616
1-owner vehicle      3642
Personal use only    3616
Clean title          7685
Open recall             0
Comfort_score        4630
Interior_score       4630
Performance_score    4630
Value_score          4630
Exterior_score       4630
Reliability_score    4706
dtype: int64
```

Interpret: This code represents the missing value in each column of the data that we have. This can show the count of missing values so it makes it easier for us to clean the unclean data

2. Dealing the missing/ null values in mileage column

```
[18] car[car['Condition'] == 'New']['Mileage'].isnull().sum()
```

```
149
```

```
[19] car.loc[(car['Condition'] == 'New') & (car['Mileage'].isnull()), 'Mileage'] = 0
     car.head()
```

	Make	Year	Condition	Mileage	Price	Seller Rating	Seller Rating Count	Interior color	Drivetrain	Min MPG	...	1-owner vehicle	Personal use only	Clean title	Open recall	Comfort_score	Interior_score	Performance_score	Value_
0	Lexus	2024	New	0.0	112865.0	4.7	1261.0	Black	Rear-wheel Drive	15.0	...	NaN	NaN	NaN	0	NaN	NaN	NaN	
1	Acura	2007	Used	61110.0	11295.0	4.2	440.0	–	Front-wheel Drive	22.0	...	True	True	NaN	0	NaN	NaN	NaN	
2	McLaren	2016	Used	6305.0	219997.0	3.1	421.0	Carbon Black	Rear-wheel Drive	NaN	...	False	False	NaN	0	5.0	5.0	5.0	
3	Audi	2016	Used	65715.0	23999.0	3.6	123.0	Black	All-wheel Drive	23.0	...	False	True	True	0	3.0	5.0	5.0	
4	BMW	2018	Used	19830.0	39799.0	4.4	91.0	Black	All-wheel Drive	0.0	...	True	False	True	0	5.0	5.0	5.0	

5 rows × 24 columns

```
# Verify that there are no missing values left in 'Mileage'
car['Mileage'].isnull().sum()
```

```
0
```

Interpret: The first code is to know the total of the missing data on the exact column that we need. We have been proven that missing 'Mileage' values are associated with 'New' conditions. Since new cars typically have 0 mileage, replace NaN values with 0 for 'New' cars.

3. Seller Rating, Seller Rating Count and Engine Size Columns

```
[21] # Sort the 'make' column alphabetically
     car = car.sort_values(by='Make')
     car.head()
```

	Make	Year	Condition	Mileage	Price	Seller Rating	Seller Rating Count	Interior color	Drivetrain	Min MPG	...	1-owner vehicle	Personal use only	Clean title	Open recall	Comfort_score	Interior_score	Performance_score	Value_
6285	Acura	2014	Used	108932.0	16995.0	4.1	46.0	Tan	All-wheel Drive	18.0	...	False	True	NaN	0	NaN	NaN	NaN	
4849	Acura	2019	Used	61472.0	24818.0	NaN	NaN	–	All-wheel Drive	21.0	...	False	True	NaN	1	NaN	NaN	NaN	
7254	Acura	2002	Used	73000.0	9500.0	NaN	NaN	Beige	Front-wheel Drive	NaN	...	False	True	True	1	NaN	NaN	NaN	
1890	Acura	2021	Used	17595.0	33585.0	4.9	1852.0	Ebony	All-wheel Drive	21.0	...	True	True	NaN	1	NaN	NaN	NaN	
8275	Acura	2016	Used	78000.0	17500.0	NaN	NaN	Black	All-wheel Drive	NaN	...	False	False	True	0	NaN	NaN	NaN	

5 rows × 24 columns

```
# Sort the 'condition' column in the order of 'new', 'used', 'certified'
# Convert 'condition' to a categorical type with specified order
car['Condition'] = pd.Categorical(car['Condition'], categories=['New', 'Used', 'Certified'], ordered=True)
car = car.sort_values(by='Condition')
car.head()
```

	Make	Year	Condition	Mileage	Price	Seller Rating	Seller Rating Count	Interior color	Drivetrain	Min MPG	...	1-owner vehicle	Personal use only	Clean title	Open recall	Comfort_score	Interior_score	Performance_score	Value_s
4867	Jeep	2024	New	90.0	55791.0	4.2	1632.0	Global Black	Four-wheel Drive	0.0	...	NaN	NaN	NaN	0	NaN	NaN	NaN	
4251	Jeep	2024	New	4.0	71224.0	4.4	1991.0	Tupelo	Four-wheel Drive	18.0	...	NaN	NaN	NaN	0	5.0	5.0	5.0	
2172	Jeep	2024	New	10.0	32992.0	4.8	3218.0	Black	Four-wheel Drive	24.0	...	NaN	NaN	NaN	0	5.0	5.0	5.0	
3159	Jeep	2023	New	30.0	48509.0	4.3	875.0	Black	Four-wheel Drive	16.0	...	NaN	NaN	NaN	0	5.0	5.0	5.0	
4491	Jeep	2024	New	0.0	52118.0	4.3	741.0	Global Black	Four-wheel Drive	18.0	...	NaN	NaN	NaN	0	5.0	5.0	5.0	

5 rows × 24 columns

```
[23] # Function to replace empty strings with NA
     def replace_empty_with_na(column):
         column.replace('', pd.NA, inplace=True)
         column.replace(' ', pd.NA, inplace=True)

     # Apply the function to the columns
     columns_to_clean = ['Seller Rating', 'Seller Rating Count', 'Engine Size (L)']
     for column in columns_to_clean:
         replace_empty_with_na(car[column])

     # Function to replace NA with mean, median, and mode
     def replace_na_with_statistics(column):
         mean_value = column.mean()
         median_value = column.median()
         mode_value = column.mode()[0]

         # Replace NA with mean
         column.fillna(mean_value, inplace=True)
         # Replace remaining NA with median
         column.fillna(median_value, inplace=True)
         # Replace remaining NA with mode
         column.fillna(mode_value, inplace=True)

     # Replace NA in the specified columns
     for column in columns_to_clean:
         replace_na_with_statistics(car[column])

     # Display the cleaned DataFrame
     print("Cleaned DataFrame:")
     print(car.head())
```

```
Cleaned DataFrame:
      Make  Year Condition  Mileage    Price  Seller Rating  \
4867  Jeep  2024       New     90.0  55791.0            4.2
4251  Jeep  2024       New      4.0  71224.0            4.4
2172  Jeep  2024       New     10.0  32992.0            4.8
3159  Jeep  2023       New     30.0  48509.0            4.3
4491  Jeep  2024       New      0.0  52118.0            4.3

      Seller Rating Count Interior color         Drivetrain  Min MPG  ...  \
4867               1632.0  Global Black   Four-wheel Drive       0.0  ...
4251               1991.0         Tupelo   Four-wheel Drive      18.0  ...
2172               3218.0          Black   Four-wheel Drive      24.0  ...
3159                875.0          Black   Four-wheel Drive      16.0  ...
4491                741.0  Global Black   Four-wheel Drive      18.0  ...

      1-owner vehicle Personal use only  Clean title  Open recall  \
4867              NaN               NaN          NaN            0
4251              NaN               NaN          NaN            0
2172              NaN               NaN          NaN            0
3159              NaN               NaN          NaN            0
4491              NaN               NaN          NaN            0

     Comfort_score Interior_score Performance_score  Value_score  \
4867           NaN            NaN               NaN          NaN
4251           5.0            5.0               5.0          5.0
2172           5.0            5.0               5.0          5.0
3159           5.0            5.0               5.0          5.0
4491           5.0            5.0               5.0          5.0

      Exterior_score  Reliability_score
4867             NaN                NaN
4251             5.0                5.0
2172             5.0                5.0
3159             5.0                5.0
```

```
[24] # Verify that there are no missing values left in 'Seller Rating'
     car['Seller Rating'].isnull().sum()
```

```
0
```

```
     # Verify that there are no missing values left in 'Seller Rating Count'
     print(car['Seller Rating Count'].isnull().sum())
```

```
0
```

```
[26] # Verify that there are no missing values left in 'Engine Size (L)'
     print(car['Engine Size (L)'].isnull().sum())
```

```
0
```

Interpret: Sorting the 'make' column alphabetically helps us arrange the dataset so that all the brands of cars are listed in order. This makes it easier to find specific brands and compare them.

Sorting the 'condition' column in the order of 'new', 'used', and 'certified' means we're organizing cars based on their condition. It's like putting them in groups: new cars, used cars, and certified cars. This makes it simpler to see what kinds of conditions the cars are in, making it easier to study or make decisions based on their condition.

4. Interior Colour, 1-Owner Vehicle and Personal Use Only Columns

```
car['Interior color'] = car['Interior color'].fillna('others')
car.head()
```

	Make	Year	Condition	Mileage	Price	Seller Rating	Seller Rating Count	Interior color	Drivetrain	Min MPG	...	1-owner vehicle	Personal use only	Clean title	Open recall	Comfort_score	Interior_score	Performance_score	Value_s
4867	Jeep	2024	New	90.0	55791.0	4.2	1632.0	Global Black	Four-wheel Drive	0.0	...	NaN	NaN	NaN	0	NaN	NaN	NaN	
4251	Jeep	2024	New	4.0	71224.0	4.4	1991.0	Tupelo	Four-wheel Drive	18.0	...	NaN	NaN	NaN	0	5.0	5.0	5.0	
2172	Jeep	2024	New	10.0	32992.0	4.8	3218.0	Black	Four-wheel Drive	24.0	...	NaN	NaN	NaN	0	5.0	5.0	5.0	
3159	Jeep	2023	New	30.0	48509.0	4.3	875.0	Black	Four-wheel Drive	16.0	...	NaN	NaN	NaN	0	5.0	5.0	5.0	
4491	Jeep	2024	New	0.0	52118.0	4.3	741.0	Global Black	Four-wheel Drive	18.0	...	NaN	NaN	NaN	0	5.0	5.0	5.0	

5 rows × 24 columns

Interpret: In this condition we replace the character "â€"" with "others" since the character does not mean anything related to the theme that the column provides. We cannot replace it with before or after data too because the interior should be different by any car and different brands have different ways to tell each color. That is why we choose the word "others" to tell the customer the color without confusing the customer.

```
car['1-owner vehicle'] = car['1-owner vehicle'].fillna('others')
car.head()
```

	Make	Year	Condition	Mileage	Price	Seller Rating	Seller Rating Count	Interior color	Drivetrain	Min MPG	...	1-owner vehicle	Personal use only	Clean title	Open recall	Comfort_score	Interior_score	Performance_score	Value_s
4867	Jeep	2024	New	90.0	55791.0	4.2	1632.0	Global Black	Four-wheel Drive	0.0	...	others	NaN	NaN	0	NaN	NaN	NaN	
4251	Jeep	2024	New	4.0	71224.0	4.4	1991.0	Tupelo	Four-wheel Drive	18.0	...	others	NaN	NaN	0	5.0	5.0	5.0	
2172	Jeep	2024	New	10.0	32992.0	4.8	3218.0	Black	Four-wheel Drive	24.0	...	others	NaN	NaN	0	5.0	5.0	5.0	
3159	Jeep	2023	New	30.0	48509.0	4.3	875.0	Black	Four-wheel Drive	16.0	...	others	NaN	NaN	0	5.0	5.0	5.0	
4491	Jeep	2024	New	0.0	52118.0	4.3	741.0	Global Black	Four-wheel Drive	18.0	...	others	NaN	NaN	0	5.0	5.0	5.0	

5 rows × 24 columns

Interpret: Since the new car still has no owner, the null values in this column represent the nonrelation of the values with the objective of the data that the column represents. so the null value here is replaced by "others" to prevent confusion for the customers.

```
car['Personal use only'] = car['Personal use only'].fillna('Not Relate')
car.head()
```

	Make	Year	Condition	Mileage	Price	Seller Rating	Seller Rating Count	Interior color	Drivetrain	Min MPG	...	1-owner vehicle	Personal use only	Clean title	Open recall	Comfort_score	Interior_score	Performance_score	Value_
4867	Jeep	2024	New	90.0	55791.0	4.2	1632.0	Global Black	Four-wheel Drive	0.0	...	others	Not Relate	NaN	0	NaN	NaN	NaN	
4251	Jeep	2024	New	4.0	71224.0	4.4	1991.0	Tupelo	Four-wheel Drive	18.0	...	others	Not Relate	NaN	0	5.0	5.0	5.0	
2172	Jeep	2024	New	10.0	32992.0	4.8	3218.0	Black	Four-wheel Drive	24.0	...	others	Not Relate	NaN	0	5.0	5.0	5.0	
3159	Jeep	2023	New	30.0	48509.0	4.3	875.0	Black	Four-wheel Drive	16.0	...	others	Not Relate	NaN	0	5.0	5.0	5.0	
4491	Jeep	2024	New	0.0	52118.0	4.3	741.0	Global Black	Four-wheel Drive	18.0	...	others	Not Relate	NaN	0	5.0	5.0	5.0	

5 rows × 24 columns

It is the same with the above problem. The personal use did not relate to the new car. so the null values there is replaced by not related to prevent confusion to customers.

```
[33] # Verify that there are no missing values left in 'Seller Rating Count'
     print(car['Interior color'].isnull().sum())

     0

[34] # Verify that there are no missing values left in 'Seller Rating Count'
     print(car['1-owner vehicle'].isnull().sum())

     0

     # Verify that there are no missing values left in 'Seller Rating Count'
     print(car['Personal use only'].isnull().sum())

     0
```

Interpret: The isnull().sum printed the sum value of missing data from the column to check whether there are still missing values there or not.

5. Drivetrain Column

```
[39] print(car['Drivetrain'].unique())

     ['4WD' 'AWD' 'FWD' 'RWD' '-' 'unknown']

     # CLEAN '-' AND 'UNKNOWN'
     car['Drivetrain'].replace({'-': 'others', 'unknown': 'others'}, inplace=True)
     car.head()
```

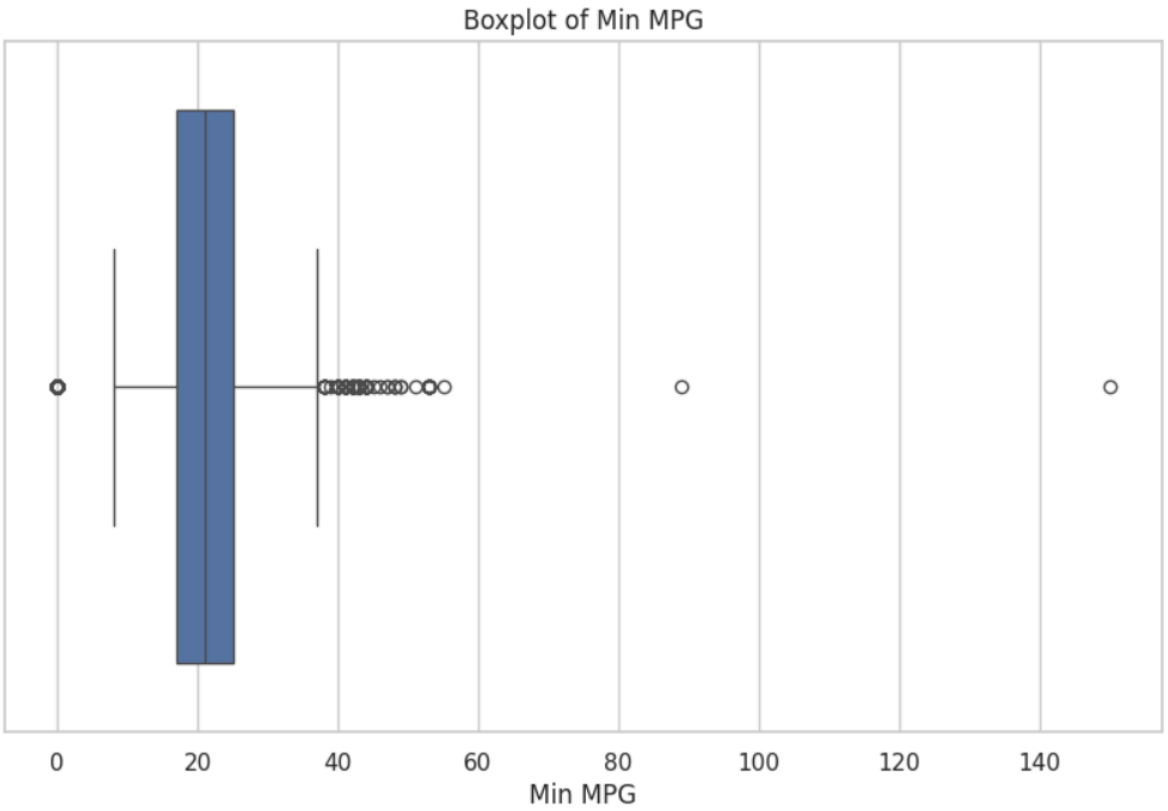
	Make	Year	Condition	Mileage	Price	Seller Rating	Seller Rating Count	Interior color	Drivetrain	Min MPG	...	1-owner vehicle	Personal use only	Clean title	Open recall	Comfort_score	Interior_score	Performance_score	Value_s
4867	Jeep	2024	New	90.0	55791.0	4.2	1632.0	Global Black	4WD	0.0	...	others	Not Relate	NaN	0	NaN	NaN	NaN	
4251	Jeep	2024	New	4.0	71224.0	4.4	1991.0	Tupelo	4WD	18.0	...	others	Not Relate	NaN	0	5.0	5.0	5.0	
2172	Jeep	2024	New	10.0	32992.0	4.8	3218.0	Black	4WD	24.0	...	others	Not Relate	NaN	0	5.0	5.0	5.0	
3159	Jeep	2023	New	30.0	48509.0	4.3	875.0	Black	4WD	16.0	...	others	Not Relate	NaN	0	5.0	5.0	5.0	
4491	Jeep	2024	New	0.0	52118.0	4.3	741.0	Global Black	4WD	18.0	...	others	Not Relate	NaN	0	5.0	5.0	5.0	

5 rows × 24 columns

Interpret: Replacing '-' and 'unknown' with 'others' in the 'Drivetrain' column. consistency ensures all data users use the same format. To handle the missing data we provide a clear placeholder for missing or uncertain values. To simplify the analysis, the groups that are less common categories are easier to analyze. The data must maintain its consistency and clarity throughout the dataset to preserve data integrity.

6. Min MPG column

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Set the style of the visualization
sns.set(style="whitegrid")

# Create a figure for the boxplot
plt.figure(figsize=(10, 6))

# Create a boxplot for Min MPG
sns.boxplot(data=car, x='Min MPG')

# Set the title and labels
plt.title('Boxplot of Min MPG')
plt.xlabel('Min MPG')

# Display the plot
plt.show()
```



```
[42] # Calculate the median for 'Min MPG'
     min_mpg_median = car['Min MPG'].median()
     min_mpg_median
```

```
21.0
```

```
[43] # Fill missing values with median
     car['Min MPG'].fillna(min_mpg_median, inplace=True)
```

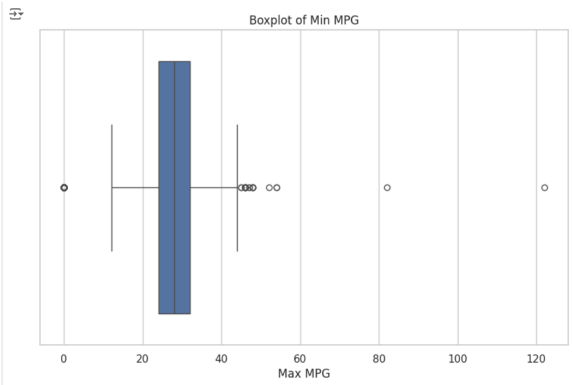
```
[44] # Verify that there are no missing values left in 'Min MPG'
     print(car['Min MPG'].isnull().sum())
```

```
0
```

Interpret: The median is the most appropriate choice to replace the missing values because it will not be skewed by the extreme values. the data will be in a way that maintains the central tendency of the data without being influenced by outliers.

7. Max MPG column

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Set the style of the visualization
sns.set(style="whitegrid")

# Create a figure for the boxplot
plt.figure(figsize=(10, 6))

# Create a boxplot for Max MPG
sns.boxplot(data=car, x='Max MPG')

# Set the title and labels
plt.title('Boxplot of Min MPG')
plt.xlabel('Max MPG')

# Display the plot
plt.show()
```




```python
[46] # Calculate the median for 'Max MPG'
     max_mpg_median = car['Max MPG'].median()
     max_mpg_median
```

```
28.0
```

```python
[47] # Fill missing values with median for 'Max MPG'
     car['Max MPG'].fillna(max_mpg_median, inplace=True)
```

```python
# Verify that there are no missing values left in 'Max MPG'
print(car['Max MPG'].isnull().sum())
```

```
0
```

Interpret: The median is the most appropriate choice to replace the missing values because it will not be skewed by the extreme values. the data will be in a way that maintains the central tendency of the data without being influenced by outliers.

8. Fuel by column

```
unique_fueltype = car['Fuel type'].unique()
unique_fueltype
```

```
array(['Gasoline ', 'Hybrid ', 'Diesel ', nan, '- ', 'E85 Flex Fuel ',
       'Plug-In Hybrid ', 'Gas/Electric Hyb ', 'Flexible Fuel '],
      dtype=object)
```

```
car['Fuel type'].replace('-', 'others', inplace=True)
```

```
car['Fuel type'].fillna('others', inplace=True)
```

```
# Verify that there are no missing values left in 'Fuel type'
print(car['Fuel type'].isnull().sum())
```

```
0
```

Interpret: unique() is used to extract all unique values from Fuel type. Any occurrence of the '-' value in the 'Fuel type' column is replaced with the string 'others'. The inplace=True parameter ensures that the replacement is made directly in the original DataFrame without needing to reassign it. fillna() fills any missing (NaN) values in the 'Fuel type' column with the string 'others'. isnull().sum() counts the number of missing values (NaNs) in the column. If everything has been correctly filled, the result should be 0.

9. Accident or damage

```
[53] car['Accidents or damage'].isnull().sum()
```

```
3616
```

```
# Fill missing values with 0
car['Accidents or damage'].fillna(0, inplace=True)
```

```
[55] # Verify that there are no missing values left in 'Accidents or damage'
     print(car['Accidents or damage'].isnull().sum())
```

```
0
```

Interpret: We chose to replace the null values with 0 because they represent where no accidents or damage occurred. Given that the 'Accidents or damage' column pertains to the condition of new vehicles, it makes sense filling missing values with 0 indicate new cars typically have no reported accidents or damage

10. Open recall Column

```
car = car.drop(columns=['Open recall'])
print("\nDropped 'Open recall' column.")
```

```
Dropped 'Open recall' column.
```

```
[109] # Display the cleaned dataset
      print("\nCleaned dataset:")
      car.head()
```

eller ating	Seller Rating Count	Interior color	Drivetrain	Min MPG	...	Accidents or damage	1-owner vehicle	Personal use only	Clean title	Comfort_score	Interior_score	Performance_score	Value_score	Exterior_score	Reliability_score
4.2	1632.0	Global Black	4WD	0.0	...	0.0	others	Not Relate	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4.4	1991.0	Tupelo	4WD	18.0	...	0.0	others	Not Relate	NaN	5.0	5.0	5.0	5.0	5.0	5.0
4.8	3218.0	Black	4WD	24.0	...	0.0	others	Not Relate	NaN	5.0	5.0	5.0	5.0	5.0	5.0
4.3	875.0	Black	4WD	16.0	...	0.0	others	Not Relate	NaN	5.0	5.0	5.0	5.0	5.0	5.0
4.3	741.0	Global Black	4WD	18.0	...	0.0	others	Not Relate	NaN	5.0	5.0	5.0	5.0	5.0	5.0

Interpret: The 'Open recall' column is not needed for interpretation or further analysis, so it is removed to clean up the dataset. So, the dataset no longer contains the 'Open recall' column.

11. Clean title,Comfort_score, Interior_score, Performance_score, Value_score, Exterior_score and Reliability_score Columns

```
num_rows = car.shape[0]
num_nulls = car['Clean title'].isnull().sum()

print(f"\nNumber of rows: {num_rows}")
print(f"Number of null values in 'clean title': {num_nulls}")
```

```
Number of rows: 9016
Number of null values in 'clean title': 7685
```

```
[111] # Drop the 'Clean title' column if more than half of the values are null
if num_nulls > (num_rows / 2):
    car = car.drop(columns=['Clean title'])
    print("\n'Dropped 'Clean title' column due to excessive null values.")
else:
    # Fill missing values if the column is not dropped
    car['clean title'] = car['clean title'].fillna('Default Value')  # Replace 'Default Value' with appropriate value
    print("\nFilled missing values in 'clean title' column.")
```

```
'Dropped 'Clean title' column due to excessive null values.
```

```
# Drop the 'Clean title' column if more than half of the values are null
if num_nulls > (num_rows / 2):
    car = car.drop(columns=['Clean title'])
    print("\n'Dropped 'Clean title' column due to excessive null values.")
else:
    # Fill missing values if the column is not dropped
    car['clean title'] = car['clean title'].fillna('Default Value')  # Replace 'Default Value' with appropriate value
    print("\nFilled missing values in 'clean title' column.")
```

```
'Dropped 'Clean title' column due to excessive null values.
```

```
[112] num_nulls = car['Comfort_score'].isnull().sum()

# Drop the 'Comfort_score' column if more than half of the values are null
if num_nulls > (num_rows / 2):
    car = car.drop(columns=['Comfort_score'])
    print("\n'Dropped 'Comfort_score' column due to excessive null values.")
else:
    # Fill missing values if the column is not dropped
    car['Comfort_score'] = car['Comfort_score'].fillna('Default Value')  # Replace 'Default Value' with appropriate value
    print("\nFilled missing values in 'Comfort_score' column.")
```

```
'Dropped 'Comfort_score' column due to excessive null values.
```

```
[113] num_nulls = car['Interior_score'].isnull().sum()
      # Drop the 'Interior_score' column if more than half of the values are null
      if num_nulls > (num_rows / 2):
          car = car.drop(columns=['Interior_score'])
          print("\n'Dropped 'Interior_score' column due to excessive null values.")
      else:
          # Fill missing values if the column is not dropped
          car['Interior_score'] = car['Interior_score'].fillna('Default Value')  # Replace 'Default Value' with appropriate value
          print("\nFilled missing values in 'Interior_score' column.")
```

```
'Dropped 'Interior_score' column due to excessive null values.
```

```
num_nulls = car['Performance_score'].isnull().sum()
# Drop the 'Performance_score' column if more than half of the values are null
if num_nulls > (num_rows / 2):
    car = car.drop(columns=['Performance_score'])
    print("\n'Dropped 'Performance_score' column due to excessive null values.")
else:
    # Fill missing values if the column is not dropped
    car['Performance_score'] = car['Performance_score'].fillna('Default Value')  # Replace 'Default Value' with appropriate value
    print("\nFilled missing values in 'Performance_score' column.")
```

```
'Dropped 'Performance_score' column due to excessive null values.
```

```
num_nulls = car['Reliability_score'].isnull().sum()
# Drop the 'Reliability_score  ' column if more than half of the values are null
if num_nulls > (num_rows / 2):
    car = car.drop(columns=['Reliability_score'])
    print("\n'Dropped 'Reliability_score   ' column due to excessive null values.")
else:
    # Fill missing values if the column is not dropped
    car['Reliability_score'] = car['Reliability_score'].fillna('Default Value')  # Replace 'Default Value' with appropriate value
    print("\nFilled missing values in 'Reliability_score' column.")
```

```
'Dropped 'Reliability_score   ' column due to excessive null values.
```

```
# Display the cleaned dataset
print("\nCleaned dataset:")
car.head()
```

```
Cleaned dataset:
```

	Make	Year	Condition	Mileage	Price	Seller Rating	Seller Rating Count	Interior color	Drivetrain	Min MPG	Max MPG	Fuel type	Engine Size (L)	Accidents or damage	1-owner vehicle	Personal use only
4867	Jeep	2024	New	90.0	55791.0	4.2	1632.0	Global Black	4WD	0.0	0.0	Gasoline	2.0	0.0	others	Not Relate
4251	Jeep	2024	New	4.0	71224.0	4.4	1991.0	Tupelo	4WD	18.0	25.0	Gasoline	5.7	0.0	others	Not Relate
2172	Jeep	2024	New	10.0	32992.0	4.8	3218.0	Black	4WD	24.0	32.0	Gasoline	2.0	0.0	others	Not Relate
3159	Jeep	2023	New	30.0	48509.0	4.3	875.0	Black	4WD	16.0	23.0	Gasoline	3.6	0.0	others	Not Relate
4491	Jeep	2024	New	0.0	52118.0	4.3	741.0	Global Black	4WD	18.0	25.0	Gasoline	3.6	0.0	others	Not Relate

Interpret: Columns with more than 50% missing data are often dropped because the remaining data may not be reliable for analysis. Otherwise, missing values are filled to maintain the integrity of the column. The final dataset is cleaner and more manageable, with unnecessary columns removed and missing values handled, making it ready for further analysis or modeling.

12.Checking for missing value

```
# Checking for missing values in each column
print(car.isnull().sum())
```

```
Make                  0
Year                  0
Condition             0
Mileage               0
Price                 0
Seller Rating         0
Seller Rating Count   0
Interior color        0
Drivetrain            0
Min MPG               0
Max MPG               0
Fuel type             0
Engine Size (L)       0
Accidents or damage   0
1-owner vehicle       0
Personal use only     0
dtype: int64
```

Interpret: Handling missing data, such as replacing NA values, is essential for ensuring the integrity and reliability of datasets. Failing to address missing values can lead to biased analyses, compromise model performance, and hinder interpretation. By appropriately managing missing data, we maintain data quality, improve the accuracy of analyses and models, and ensure that results are trustworthy and actionable results.

3.4 Data Preview

Data preview is the initial phase in the data analysis process where we explore and summarize the dataset. Data preview is important to a better understanding of the structure, content, and quality of the dataset. It also gives an overview of the data and helps to identify any missing or dirty values in the dataset.

Figure 2 shows the first 5 rows of the car_clean dataset. From the dataset, we can get a quick overview of the structure of the dataset. Then we explore the total number of rows and columns using the shape function in Python which results in 9016 rows and 16 columns. In 16 columns we used columns and dtypes, 7 of them have data type objects which are Make, Condition, Interior color, Drivetrain, Fuel type, 1-owner vehicle, and Personal use only, other data types have seven attributes which are float that include Mileage, Seller Rating, Seller Rating Count, min/max MPG, Engine Size (L), Accidents or damage attributes and the rest two attributes which are year and price are integer data types.

Besides, we are going deeper to know the overall distribution and characteristics of the car dataset. We used the describe function that can provide central tendency and dispersion of the variables in the car dataset. The count row for all attributes is the same which means a total number of observations of all attributes filled with 9016 rows. The mean row displays the average value for each attribute and std represents the standard deviation that measures the spread of the data. In the describe function also we know the min and max value for each attribute which we can see the range value of. Lastly, it provides the first, second, and third percentile for each attribute.

Lastly, we check the missing value for each column and count the unique value for each attribute. When we know the count of unique values it is easy for us to do the analysis relationship and what graph is suitable for the large distinct value and less distinct value. The codes for other operations have been shown in the appendix.

```
[7] car_clean.head()
```

	Make	Year	Condition	Mileage	Price	Seller Rating	Seller Rating Count	Interior color	Drivetrain	Min MPG	Max MPG	Fuel type	Engine Size (L)	Accidents or damage	1-owner vehicle	Personal use only
0	Jeep	2024	New	90.0	55791	4.2	1632.0	Global Black	4WD	0.0	0.0	Gasoline	2.0	0.0	others	Not Relate
1	Jeep	2024	New	4.0	71224	4.4	1991.0	Tupelo	4WD	18.0	25.0	Gasoline	5.7	0.0	others	Not Relate
2	Jeep	2024	New	10.0	32992	4.8	3218.0	Black	4WD	24.0	32.0	Gasoline	2.0	0.0	others	Not Relate
3	Jeep	2023	New	30.0	48509	4.3	875.0	Black	4WD	16.0	23.0	Gasoline	3.6	0.0	others	Not Relate
4	Jeep	2024	New	0.0	52118	4.3	741.0	Global Black	4WD	18.0	25.0	Gasoline	3.6	0.0	others	Not Relate

Figure 2: The result shows that the first 5 rows of the car_clean dataset

3.5 Data Description

Based on the problem statement and objectives of the study, we can identify and describe the most useful columns/attributes in the dataset to accomplish the objectives of the project. Here are the selected columns and their relevance:

Attributes	**Use**	**Objective**
Make	Analyze the popularity, reliability, and resale value of car brands	Understand brand preferences for better inventory management and marketing
Year	Examine the impact of production year on car value, depreciation, and desirability	Identify trends related to the production year for optimizing inventory and sales strategies
Condition	Compare used and new car conditions, assess impact on pricing and customer satisfaction	Enhance sales strategies by highlighting well-maintained used cars and their value propositions
Mileage	Analyze how mileage affects car pricing, desirability, and longevity	Provide insights into pricing strategies and customer preferences based on mileage

Price	Investigate price distribution across car brands, conditions, and production years	Develop competitive pricing strategies to attract customers and boost sales
Seller Rating & Seller Rating Count	Evaluate the trustworthiness and reputation of sellers, correlate with car condition, price, and customer satisfaction	Enhance customer trust and satisfaction by promoting highly-rated sellers
Drivetrain	Assess the impact of drivetrain types on fuel efficiency and customer preferences	Inform customers about drivetrain options and their benefits
Min MPG & Max MPG	Analyze fuel efficiency of different cars and impact on customer buying decisions	Promote fuel-efficient cars to environmentally-conscious buyers and optimize marketing strategies
Fuel Type	Assess the environmental impact and fuel efficiency of cars with different fuel types	Highlight eco-friendly options and inform customers about fuel-related benefits
Engine Size (L)	Analyze the relationship between engine size, performance, and fuel efficiency	Provide insights into performance-related attributes that influence customer preferences
Accidents or damages	Evaluate the impact of accident history on car value and customer trust	Ensure transparency and build customer trust by disclosing accident histories
1-Owner Vehicle	Assess the impact of single ownership on car condition and value	Promote single-owner vehicles as more reliable and well-maintained options

Personal-use only	To determine whether a vehicle has been used solely for personal purposes as opposed to commercial or other uses. This can impact the condition and maintenance history of the car.	Highlighting cars that have only been used for personal purposes, which are often perceived to be in better condition and more carefully maintained. This can help in marketing these vehicles as more reliable and appealing options for customers.

Table 2: Description of datasets (cleaned)

By focusing on these columns, we can effectively clean, preprocess, and analyze the dataset to extract meaningful insights and propose strategies that align with the objectives of improving AutoHub's sales performance, customer satisfaction, and overall business growth.

4.0 EXPLORATORY DATA ANALYSIS

Boxplot: Maximum Miles Per Gallon (Mpg) By Drivetrain Type

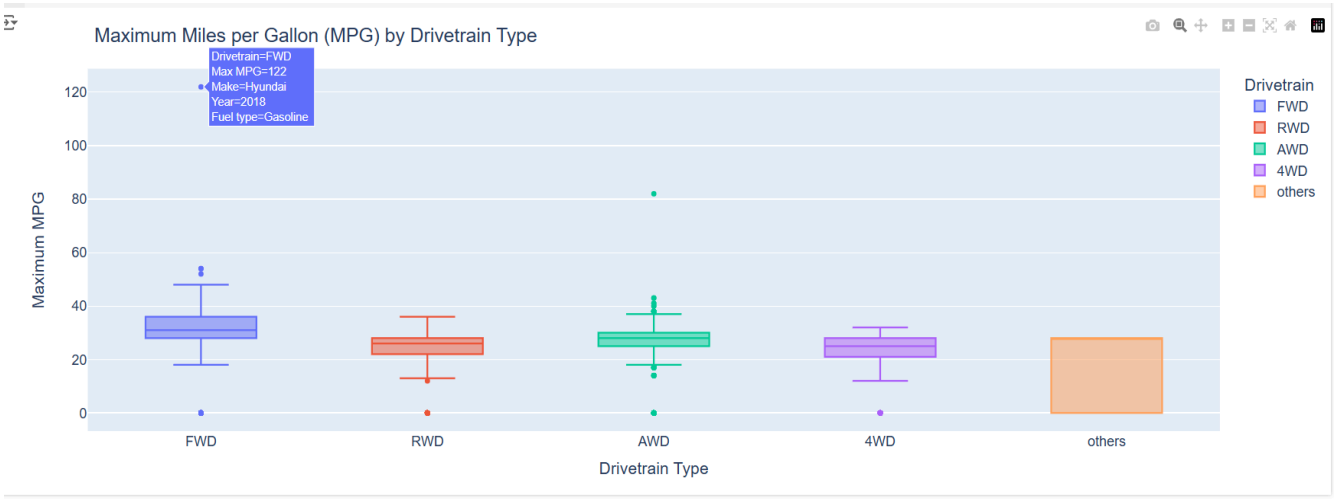


Figure 3: The boxplot Graph Between Maximum Miles per Gallon by Drivetrain Type

Based on Figure 2, the boxplot graph shows the maximum miles per gallon (MPG) by drivetrain type. We have five types of drivetrains which are Front-Wheel Drive(FWD), Rear-Wheel Drive(RWD), All-Wheel Drive (AWD), Four-Wheel Drive(4WD) and others. The boxplot represents the interquartile range of the maximum MPG value for each drivetrain type.

Front-wheel drive (FWD) cars generally have the highest maximum MPG in this dataset. The IQR for FWD cars is entirely above the IQRs for the other drivetrain types, and the median MPG for FWD cars is also the highest. This suggests that FWD cars tend to be more fuel-efficient than cars with other drivetrain types. Compared to All-wheel drive (AWD) cars that have the lowest maximum MPG.The IQR for AWD cars is the lowest, and the median MPG for AWD cars is also the lowest. This suggests that AWD cars tend to be less fuel-efficient than cars with other drivetrain types

Scatter Plot : Minimum and Maximum MPG by Engine Size for {brand}

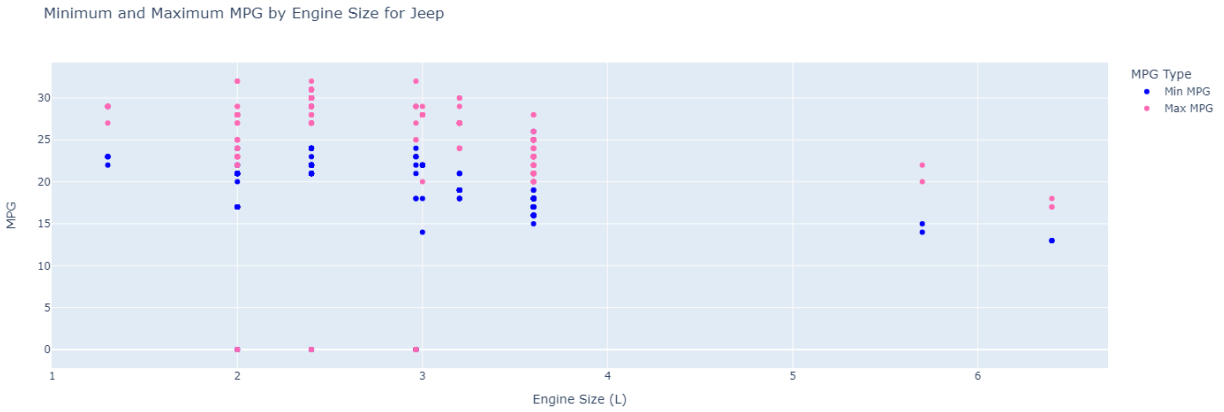


Figure 4: Maximum and Minimum MPG by Engine Size for Jeep Cars

The scatter plot graph of engine size against maximum and minimum miles per gallon (MPG) reveals a clear inverse relationship between engine size and fuel efficiency. As the engine size increases, both maximum and minimum MPG values tend to decrease. This indicates that larger engines, while generally more powerful, are less fuel-efficient compared to smaller engines.The mpg here divided by two which is min mpg and max mpg .Which min mpg is for city drive and max mpg is more to highway drive.

Generally, there is an inverse relationship between engine size and MPG. Larger engines tend to have lower MPG (both maximum and minimum) because they require more fuel to generate power. Conversely, smaller engines tend to have higher MPG due to their fuel efficiency.

So for a longer drive we would suggest a moderate engine size which is 2.0L to 3.0L. For long-distance driving, the graph underscores the advantage of selecting vehicles with smaller to moderate engine sizes, as they offer superior fuel efficiency. This can lead to significant cost savings on fuel and reduce environmental impact. Additionally, understanding the trend helps in making informed decisions based on driving needs and priorities. Advanced technologies that improve fuel efficiency in larger engines can also be considered for those requiring higher power output without severely sacrificing fuel economy.

Horizontal Bar Chart:

Maximum/Minimum/Average Price Of Car Make By Fuel Type

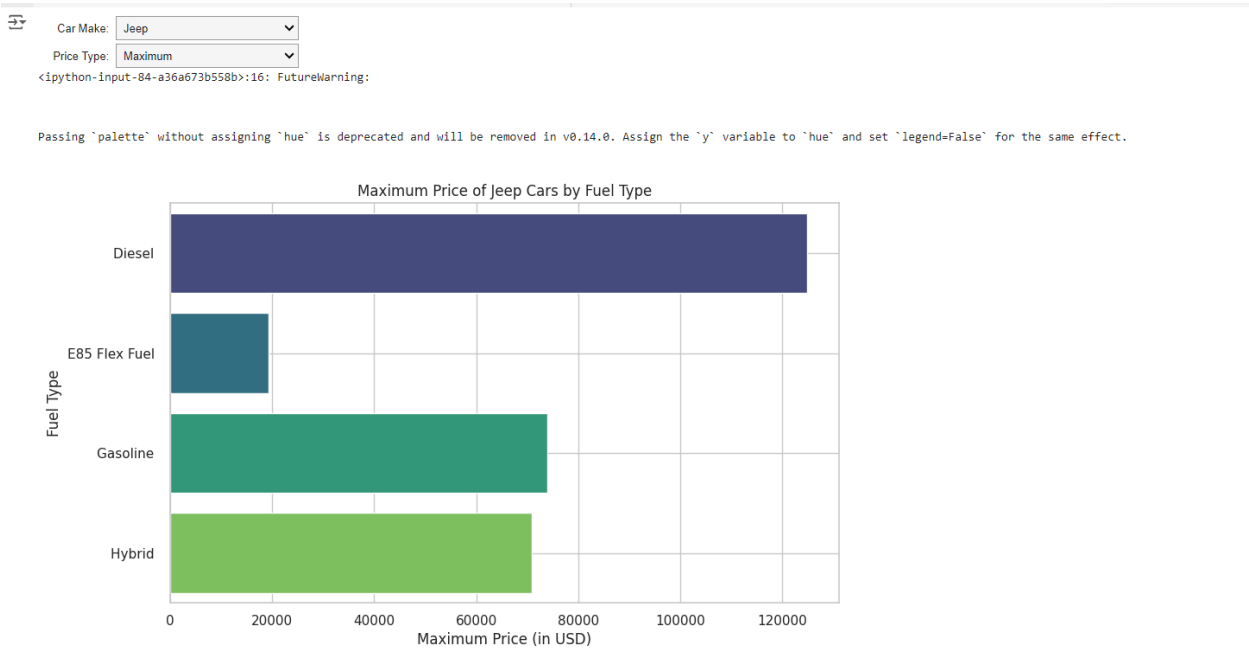


Figure 5: Maximum Price of Jeep Cars by Fuel Type

The horizontal bar graph titled “Maximum Price of Jeep Cars by Fuel Type” illustrates the maximum prices for Jeep cars across four distinct fuel types: Diesel, E85 Flex Fuel, Gasoline, and Hybrid. The x-axis, labeled “Maximum Price in USD,” ranges from 0 to 120,000 USD, indicating the scale of maximum price values.

The graph shows that E85 Flex Fuel Jeep cars have a maximum price of approximately less than 20,000 USD, suggesting a lower price range compared to other fuel types. Jeep cars with Gasoline have a maximum price of around 70,000-80,000 USD. Hybrid Jeep cars reach a maximum price of about 70,000 USD, indicating a significant decrease compared to Diesel models. Diesel-powered Jeep cars have the highest maximum price at approximately more than 120,000 USD, highlighting that these models are the most expensive among the fuel types listed.

This variability in maximum prices can be attributed to factors such as technology, fuel efficiency, and market demand. The data suggests a market trend favoring more environmentally

friendly and technologically advanced vehicles, as reflected by the higher prices of Hybrid models. Understanding these price ranges helps potential buyers make informed decisions based on budget and fuel type preference, while dealers can use this insight for inventory and pricing strategies. This analysis provides a comprehensive understanding of how Jeep car prices vary according to their fuel types, offering valuable insights for consumers and industry stakeholders alike.

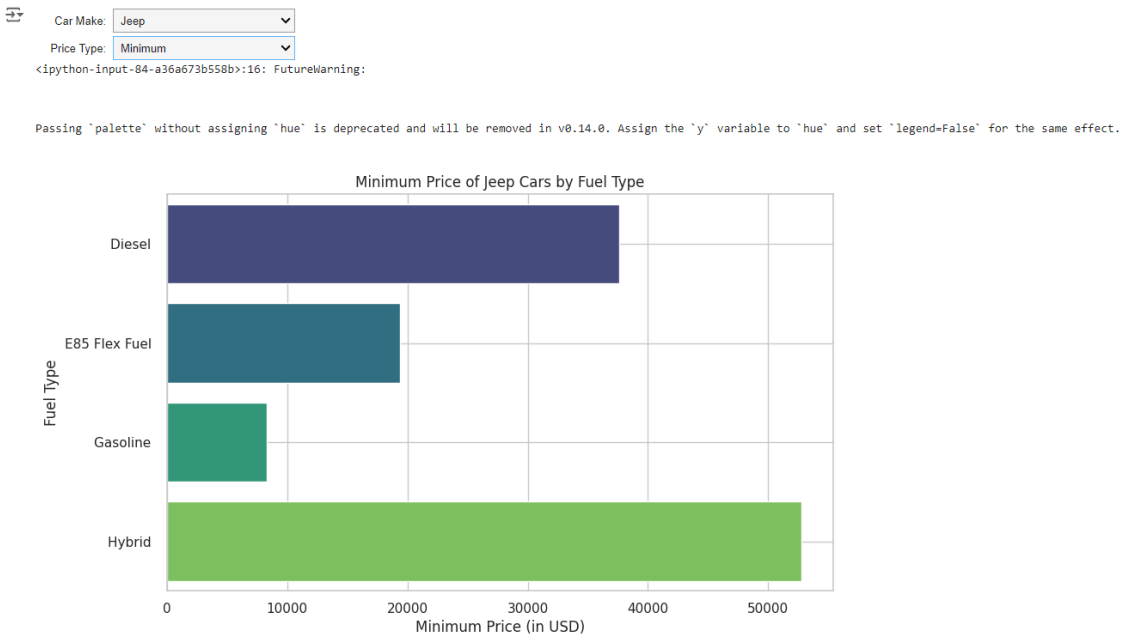


Figure 6: Minimum Price of Jeep Cars by Fuel Type

The bar graph above presents a comparative analysis of minimum prices across four distinct fuel types: Diesel, E85 Flex Fuel, Gasoline, and Hybrid. Each bar on the graph represents the minimum price (in USD) associated with a specific fuel type. The x-axis, labeled "Minimum Price in USD," provides a scale ranging from 0 to 50,000 USD, with markers at intervals of 10,000 USD.

Among the fuel types, Hybrid exhibits the highest minimum price, extending beyond the 50,000 mark on the horizontal axis. Diesel follows with a shorter bar compared to Hybrid but longer than E85 Flex Fuel and Gasoline. E85 Flex Fuel has a significantly shorter bar than Diesel and Hybrid but surpasses Gasoline in minimum price. Conversely, Gasoline shows the shortest bar, indicating the lowest minimum price among the listed fuel types.

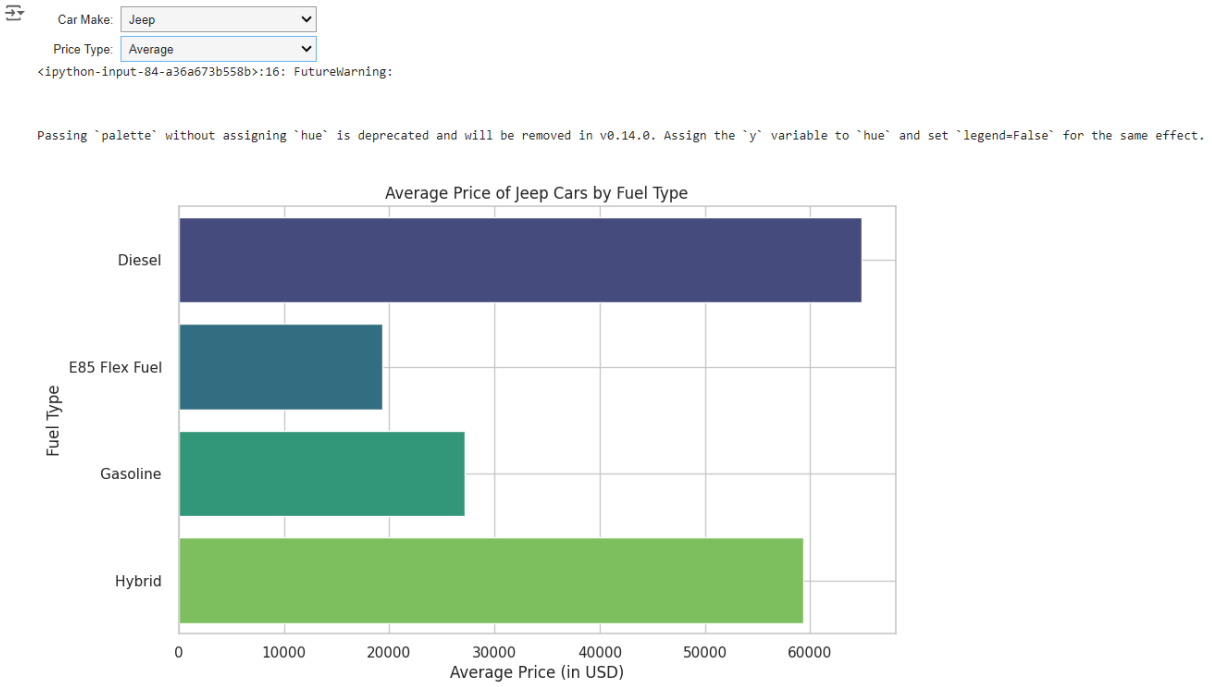


Figure 7: Average Price of Jeep Cars by Fuel Type

The bar graph titled "Average Price of Jeep Cars by Fuel Type" offers a comparative analysis of average prices across four distinct fuel types: Diesel, E85 Flex Fuel, and Gasoline. Each fuel type is represented by a horizontal bar on the graph, positioned along the y-axis. The x-axis delineates the price range from 0 to 60,000 USD, with markers at intervals of 10,000 USD, providing a clear scale for interpreting the average price values.

Diesel emerges with the highest average price, as evidenced by the longest bar on the graph. Following Diesel, E85 Flex Fuel exhibits a slightly lower average price, indicated by a moderately shorter bar. E85 Flex Fuel, on the other hand, displays the shortest bar among the three fuel types, suggesting the lowest average price for Jeep cars associated with this fuel type.

The observations drawn from this graph shed light on how different fuel types influence the average pricing of Jeep cars. Hybrid-powered models command a higher average price compared to E85 Flex Fuel and Gasoline counterparts, potentially reflecting factors such as

engine efficiency, market demand, and production costs. E85 Flex Fuel-powered Jeep cars, characterized by the shortest bar, demonstrate a comparatively lower average price, implying affordability relative to the other fuel types.

The implications of this visualization extend beyond mere price comparisons, offering valuable insights for economic and environmental analyses. Understanding the relationship between fuel type and average pricing enables stakeholders to make informed decisions regarding vehicle purchases, considering both financial constraints and environmental concerns. Moreover, this analysis contributes to a broader understanding of market dynamics within the automotive industry, facilitating strategic planning and decision-making processes.

Line Chart : Average Price Trend Over Production Years by Car Make and Condition

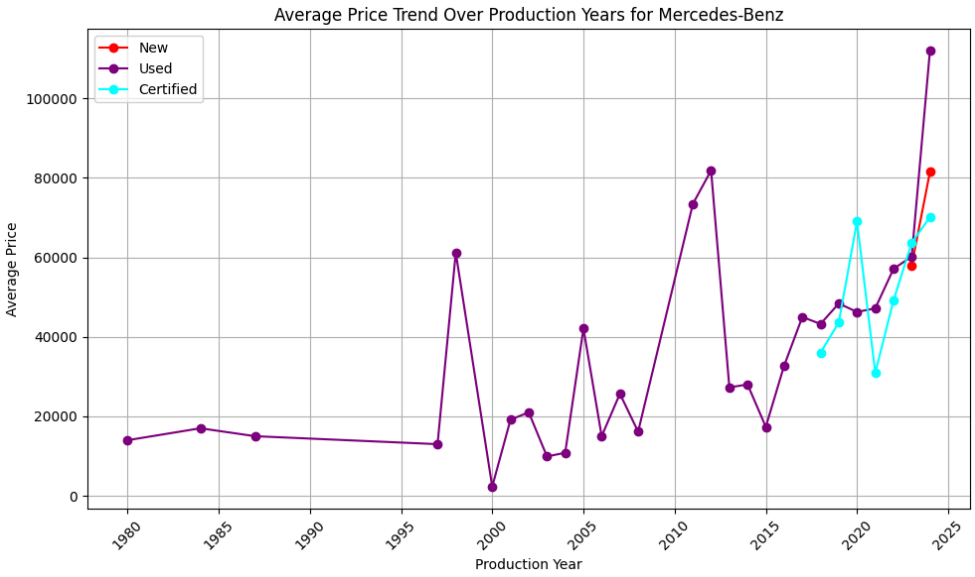


Figure 8 : The Line Plot between Average price, Years, Car make and Condition

The graph represents the average price trend across production years for the different makes and condition of cars—New, Used, or Certified. Each line will correspond to a make and condition of a car, thus giving a total comparison amongst brands and statuses. The x-axis points out the production years, meaning the age of the vehicle, while on the y-axis, the average price is shown.

The position of the lines vertically at different points along the x-axis can demonstrate the average price levels for every car make and condition in time. The comparative analysis of average price trends between Mercedes-Benz and Toyota vehicles identifies some characteristic patterns that define the position of brands and express consumer preferences within the automotive market. Meanwhile, Mercedes-Benz almost always is very high in average price due to its premium brand perception and luxury vehicle offerings. In contrast, Toyota is relatively more stable or on an upward trend in average price due to sustained demand and loyalty to the brand from the mass segment. This comparison is critically important as it would provide very valuable insights into how pricing dynamics differ across a luxury and a mainstream segment and allow for the formation of appropriate pricing strategies and inventory management and positioning in the market by relevant stakeholders to optimize their sales performance and competitiveness in the automotive industry.

Therefore, it have direct link to the objectives of the assignment because data visualization in the graph holds some important insights from the given car sales dataset. In particular, the insights from the graph would help in analyzing which factor influence the decision made by the customers in buying a car or a vehicle from the showroom of AutoHub. It includes brand perception and price trend of the vehicles. The graph also compares the average price trend between Mercedes Benz and Toyota which in turns shows how each brands is positioned in the market and what customers prefer to buy. This would help AutoHub sales and marketing team to understand where they stand in comparison to competitors and what customers prefer. Moreover, the insights from the data visualization of the graph could also be provided to the management of AutoHub to recommend a strategy that could increase in customer satisfaction as well as enhance sales performance. This in turn would be beneficial for business growth. Thus, the graph has a strong connection to the overall tasks of the assignment and by using the insights, objectives and goals of the assignment could be achieved efficiently with the operations of AutoHub.

Bar Chart: Distribution Of Car Condition For Selected Car Make

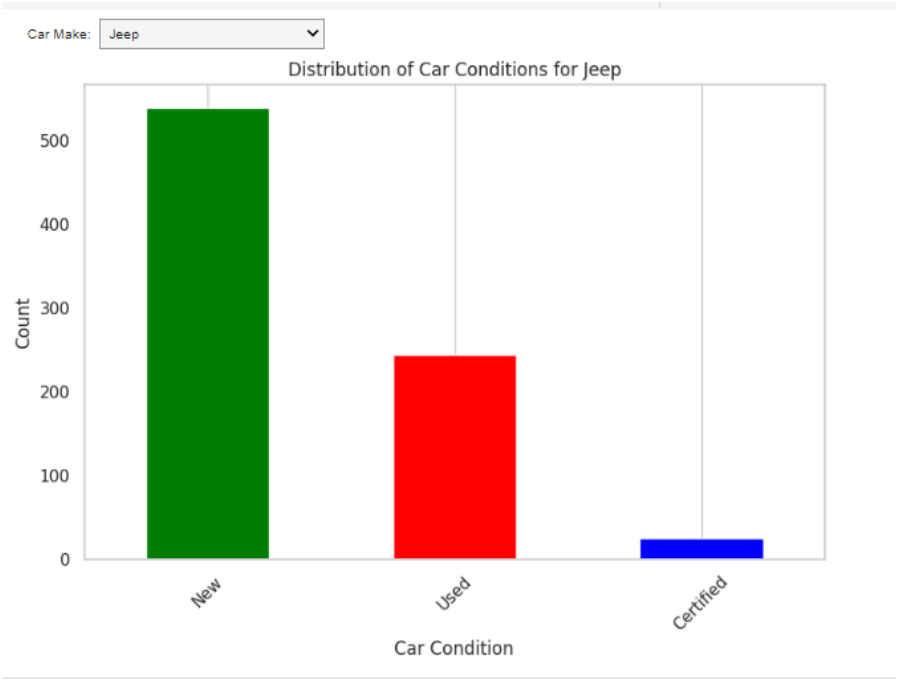


Figure 9: The Bar Chart Distribution of Car Condition for Selected Car Make

The bar chart illustrates the distribution of car conditions (New, Used, Certified) for a selected car make. Each bar represents a different condition of the car, providing a visual comparison of the number of cars in each category. The x-axis shows the car conditions, while the y-axis indicates the count of cars in each condition.

The height of the bars across the x-axis reveals the prevalence of each car condition for the selected make, offering valuable insights into consumer preferences and market dynamics. For instance, when selecting Mercedes-Benz, the bar chart may indicate a higher count of Certified cars compared to New or Used ones, suggesting that consumers value the assurance and warranty that come with certified pre-owned vehicles in the luxury segment. This preference for certified cars reflects a desire for quality and reliability while still saving on costs compared to buying new.

In contrast, for a mainstream brand like Toyota, the chart might display a more balanced distribution between New and Used cars, indicating a strong market presence in both new and pre-owned segments. The substantial number of New Toyota cars highlights the brand's appeal to buyers looking for the latest models with the newest features. Meanwhile, the considerable count of Used Toyota cars suggests robust demand among cost-conscious consumers seeking reliability at a lower price point. The presence of Certified Toyota cars, although perhaps lower in count compared to New and Used, still indicates a significant market for certified pre-owned vehicles, appealing to buyers who want the benefits of a used car with added quality assurances.

These insights are critical for AutoHub as they highlight the different market dynamics and consumer behaviors associated with luxury and mainstream brands. Understanding these preferences allows AutoHub to tailor its inventory and sales strategies effectively. For instance, maintaining a robust stock of Certified Mercedes-Benz cars can attract luxury buyers looking for reliability and value, while ensuring a good mix of New and Used Toyota cars can cater to a broader customer base. Additionally, these insights inform marketing strategies, enabling targeted campaigns that emphasize the strengths of each car condition and brand, thereby enhancing customer satisfaction and driving sales performance.

5.0 CONCLUSION

In conclusion, this project undertakes a comprehensive analysis of the car sales data from AutoHub, a car dealership in Malaysia, to uncover valuable insights and trends that can inform and enhance their sales and marketing strategies. The study delves into various factors such as vehicle condition, mileage, seller ratings, interior condition, drivetrain, fuel efficiency, engine size, and ownership history. By examining these parameters, the analysis aims to shed light on customer preferences, price patterns, and the impact of different car attributes on demand within the automotive market.

The comparative study between new and used cars highlights the unique advantages and considerations associated with each option. While new cars offer the benefit of being in pristine condition with no previous ownership history, they also come with higher depreciation rates. On the other hand, used cars provide a cost-effective alternative, often with warranties and thorough inspections, but may carry uncertainties regarding their previous usage. This project aims to assist customers in making informed decisions by evaluating the performance, brand reputation, fuel efficiency, mileage, and overall condition of the cars available at AutoHub. This enables customers to choose the best option within their budget and expectations.

The insights derived from this analysis are crucial for AutoHub. By understanding the factors that influence customer decisions, AutoHub can optimize its inventory, ensuring a balanced stock of new and used cars that align with market demand. Moreover, tailored marketing strategies can be developed to highlight the strengths of each car category, attracting potential buyers and boosting sales performance. Ultimately, this data-driven approach empowers AutoHub to enhance customer satisfaction, improve sales performance, and achieve sustained business growth in the competitive automotive industry.

REFERENCES

Bortel, I., Vávra, J., & Takáts, M. (2019). Effect of HVO fuel mixtures on emissions and performance of a passenger car size diesel engine. Renewable Energy, 140, 680–691. https://doi.org/10.1016/j.renene.2019.03.067

CompareHero . (05 May , 2021). Retrieved from #NewNormal: Top 3 Factors To Consider Before Buying A Car In Malaysia: https://www.comparehero.my/transportation/articles/what-to-consider-when-buying-car-in-malaysia

GeeksforGeeks. (28 Feb, 2024). Retrieved from What is Data Preparation?: https://www.geeksforgeeks.org/what-is-data-preparation/

Kurnia Insurans. (07 Feb, 2024). Retrieved from Buying a Car in Malaysia? Here are 11 Things to Consider: https://www.kurnia.com/blog/how-to-buy-a-car-in-malaysia

Narayanan, R. (22 August, 2022). *The Financial Express*. Retrieved from Types of drive train explained: 2WD, 4WD, FWD, RWD, AWD: https://www.financialexpress.com/auto/car-news/types-of-drive-train-explained-2wd-4wd-fwd-rwd-awd/2639856/

Sabhadiya, J. (n.d.). *Engineering Choice*. Retrieved from What Is Drivetrain? | Powertrain VS Drivetrain: https://www.engineeringchoice.com/what-is-drivetrain/

TiresPlus. (25 Jan, 2021). Retrieved from WHAT IS THE DRIVETRAIN AND WHAT DOES IT DO?: https://www.tiresplus.com/blog/maintenance/what-is-drivetrain/

10 advantages of buying a new car. Sales Advisor. (2022, April 14). https://salesadvisor.my/info/10-advantages-of-buying-a-new-car/

8 advantages of buying a used car: Koons Silver Spring Mazda: MD. Koons Mazda Silver Spring. (n.d.). https://www.koonsmazda.com/buying-used-car.html

Kurko, M. (n.d.). *Best used car websites of 2024*. Investopedia. https://www.investopedia.com/best-used-car-sites-5094153

Automatic Inspection Data Import for CAQ.net. (n.d.). Www.caq.de. Retrieved June 13, 2024, from https://www.caq.de/en/features/automatic-data-gathering

What factors affect the price of a second-hand car. (n.d.). HDFCErgo. https://www.hdfcergo.com/blogs/car-insurance/what-factors-affect-the-price-of-a-second-hand-car

APPENDIX

Detail dataset: Prediction of car prices using XGBoost (kaggle.com)

Detail code:
https://colab.research.google.com/drive/1pL8NyvS_b4vScB2yNLKEtmPU5uRZli-C?usp=sharing

Detail slide:
https://www.canva.com/design/DAGHPJCFiu4/wN6Klrh8vlQbUW8E4PjsAg/edit?utm_content=DAGHPJCFiu4&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton