THE NEW FRONTIERS OF E-LEARNING: SHAPING THE FUTURE OF EDUCATION

VOLUME 10, 2025
e-ISBN : 978-629-98755-7-4

SIG CS@e-Learning
Unit Penerbitan
Jabatan Sains Komputer & Matematik
Universiti Teknologi MARA Cawangan Pulau Pinang

e-ISBN : 978-629-98755-7-4
SIG : e-Learning@CS
https://appspenang.uitm.edu.my/sigcs/
Publication Date : 26 – Sep - 2025

DATA ANALYTICS FOR ELECTRIC VEHICLE SPECIFICATIONS AND PRICES

*Syarifah Adilah Mohamed Yusoff[1], Siti Nuralia Natashya Hamidan[2], Madihah Mokhtar[3] and Nurdiyana Hanisa Roslan Zaidi[4]

**syarifah.adilah@uitm.edu.my[1]*

[1]Jabatan Sains Komputer & Matematik (JSKM),
Universiti Teknologi MARA Cawangan Pulau Pinang, Malaysia

[2,3,4]Faculty of Computer and Mathematical Sciences,
Universiti Teknologi MARA Cawangan Arau, Perlis, Malaysia

ABSTRACT

The worldwide surge in Electric Vehicle adoption underscores the importance of not only production and sales but also the characteristics that shape consumer choices, such as price, battery capacity, range, charging time, and performance. While macro-level data reveals the overall market trajectory, micro-level analysis of vehicle specifications provides insight into how product attributes and pricing influence adoption across regions. This study examines the Electric Vehicle Specifications and Price dataset, which comprises both macro- and micro-level data, to be evaluated using data analytics in Python. The study emphasises the establishment of protocols for data management quality before doing any descriptive analysis.

Keywords: electric vehicle; data analytics, macro-level, micro-level, data quality

Introduction

Electric vehicles are a fast-expanding area of the automotive industry, with significant importance in sustainability, innovation, and customer interest. The EV now have shifted from niche to mainstream, driven by rapidly decrease cost, better emission policies, and extending model variation in the market. EV sales exceed 17 million in 2024 worldwide, accounting for 20 percent of all new cars; China remained the epicenter, with almost 11 million EV sold which nearly two-thirds of global sales (ahay38,2025). Analysts expect momentum to continue in 2025, with 22 million EVs projected worldwide as battery costs fall and more affordable model arrive (IEA, 2025).

Beyond sales, the supply side has scaled dramatically within 2024 production reached almost 17.3 million EV and 70 percent of which were built in China. Capacity build-out in batteries is running ahead of demand, with 2025 cell manufacturing capacity estimated in the multi-terawatt-hour range, indicating continued cost pressure and availability (IEA, 2025).

Asia anchors today's EV market, with China's share hovering around 50 percent of new car sales in 2024, while Europe averaged roughly one in four in early 2025, and the United States grew, albeit more slowly than the previous year. Focusing in Southeast Asia, adaptation is accelerating from a low base as governments roll out incentives and charging networks. As ASEAN uptake rises year-on-year, Thailand currently leads the region's transition.

Malaysia is entering a phase of faster adoption, supported by tax incentives and a growing model line-up. EV as BEV penetration remains modest but rising, with BEVs accounting for almost 3.4 percent of new car sales in 1H 2025, a sharp increase from 2024 levels (Lye, 2025). Policymakers target 10,000 public charging points under the Low Carbon Mobility Blueprint. As of October 2024, about 3,354 chargers had been installed, and authorities continue to emphasize network expansion to sustain uptake (Shahrilmd, 2025; *PLANMalaysia - MEVnet*, 2025).

This study employs the Electrical Vehicle Specifications and Price dataset (Fatih İlhan, 2023), which compiles detailed information on EV models worldwide. By analyzing specifications (e.g., battery size, range, acceleration, top speed) alongside price data, we gain a better understanding of the trade-offs consumers face and identify trends that differ globally across Asia and within Malaysia's emerging markets. Such granular analysis can reveal whether high upfront costs or lower costs drive adoption, and how these factors align with Malaysia's policy targets and infrastructure readiness.

Problem Statements

Despite the rapid growth of EV worldwide, adoption rate remains uneven across regions and market segments. While global data highlights rising sales volumes, less attention is given to micro-level factors such as price, battery capacity, range and performance that directly influence consumer adoption.

In Asia, focusing on Malaysia, EV penetration is still modest compared to global leaders like China and Europe. High upfront costs, limited model variety, and concerns over charging infrastructure are often known as barriers. However, there is limited empirical research of quantitatively links EV specifications and pricing to adoption trends in Malaysia context. This lack of insight leads to poor incentives and market strategies by policymakers and industries that align to consumer preferences and technological readiness. // write para to align the problem with summary of dataset.

Data Management Towards Quality Data

Data quality management is important to ensure it is useful for business purposes in term of accuracy, completeness, validity, uniqueness, timeliness and consistency. This empowers the business for decision making, understand spending, enhance customer experiences and drive business growth. Hence, several careful steps were taken to process the dataset of EV for better analysis and followed as elaborated in the next subsection.

Data Definition Installing and Important Libraries

The code begins by installing and importing necessary libraries, including pandas for data manipulation, matplotlib.pyplot, and seaborn for data visualization. Figure 1 shows process of loads the electric vehicle dataset (EV_cars.csv) into a DataFrame called df_ev. The column Price.DE., which most likely shows vehicle prices in Euros, is converted to Malaysian Ringgit (RM) at an exchange rate of around 4.7. Finally, the columns name is changed from Price.DE. to RM for clarity and consistency in the local context.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df_ev = pd.read_csv('EV_cars.csv')
df_ev['Price.DE.'] = df_ev['Price.DE.']*4.7
df_ev.rename(columns={'Price.DE.': 'RM'}, inplace=True)
```

Figure 1: Process of importing important libraries for excessing the dataset as data frame

The next initial preparation after the data was loaded and to get prepared for quality assessments were profiling the data, parsing and standardization to make sure all the data is standard across all records such as 'Tesla Model 3' vs 'tesla model 3", generalization or also known as cleansing null content, marching and check for inconsistent and misspelling,

Data Quality Assessment

1. Format check

The coding in Figure 2 describes output that include a notice for each column that indicates whether there are any formatting concerns. If a column contains entries that cannot be translated to numerical values, it will show the number of problematic entries that require attention. Columns with no difficulties will simply indicate that there are no formatting concerns. This makes it easier to identify columns that need extra inquiry or cleaning, resulting in more effective error detection and data analysis.

```
for col in df_ev.select_dtypes(include='object'):
    cleaned = pd.to_numeric(df_ev[col].str.replace(',', '').str.replace(' ', ''), errors='coerce')

    num_errors = cleaned.isna().sum()

    if 0 < num_errors < len(df_ev):
        print(f" Possible formatting errors in column '{col}': {num_errors} entries could not be converted to numeric.")
    else:
        print(f" No formatting issues in column '{col}'.")
```

```
No formatting issues in column 'Car_name'.
No formatting issues in column 'Car_name_link'.
No formatting issues in column 'Speed_Flag'.
```

Figure 2: *Process of Handling Formatting Issues*

2. Completeness checks

For the purposes of continuous data quality monitoring, completeness was verified by evaluating the following columns: Car_name, Efficiency, Fast_charge, RM, and Range. These fields are essential for conducting accurate analyses of electric vehicles. In Figure 3, a script was used to calculate the number of missing values in each of these columns, along with the total number of rows containing at least one missing value across these key attributes. This approach helps to identify incomplete records that could impact analysis or require human validation. Only a summary of missing data is displayed to streamline auditing and support targeted evaluation.

```
# List of essential columns
essential_cols = ['Car_name', 'Efficiency', 'Fast_charge', 'RM', 'Range']

# Check for missing values in these columns
missing_values = df_ev[essential_cols].isnull().sum()

# Display the results
print("Missing values in essential columns:")
print(missing_values)

# Check if there are any rows with missing values in these essential columns
rows_with_missing = df_ev[essential_cols].isnull().any(axis=1).sum()

print(f"\nTotal rows with missing values in essential columns: {rows_with_missing}")
```

```
Missing values in essential columns:
Car_name       0
Efficiency     0
Fast_charge    0
RM             0
Range          0
dtype: int64

Total rows with missing values in essential columns: 0
```

Figure 3: *Detecting Missing Values Through Completeness Checks*

3. Reasonableness checks

As part of the reasonable checks for ensuring data validity, two key evaluations were performed. First, in Figure 4 shows vehicle entries priced between RM 290,000 and RM 400,000 were filtered to identify high-end models that may require verification for pricing accuracy; second in Figure 5, a correlation analysis between Battery size and Range was performed to validate the logical relationship that larger batteries typically support longer driving distances. These tests guarantee that individual values fit within expected bounds and that associated features follow consistent patterns, which contributes to the dataset's overall integrity.

```
# Check for price values within the range (RM 290,000 to RM 400,000)
outliers = df_ev[(df_ev['RM'] >= 290000) & (df_ev['RM'] <= 400000)]

# Display the rows with outlier prices in a DataFrame format
outliers
```

	Battery	Car_name	Car_name_link	Efficiency	Fast_charge	RM	Range	Top_speed	acceleration..0.100.	Speed_Flag
6	71.0	BMW IX XDRIVE40	https://ev-database.org/car/1472/BMW-iX-xDrive40	197	480.0	363310.0	360	200	6.1	OK
14	75.0	TESLA MODEL Y PERFORMANCE	https://ev-database.org/car/1183/Tesla-Model-Y...	181	640.0	299234.9	415	250	3.7	OK
18	74.0	BMW IX3	https://ev-database.org/car/1535/BMW-iX3	192	520.0	316310.0	385	180	6.8	OK
21	85.4	BYD HAN	https://ev-database.org/car/1784/BYD-HAN	180	450.0	332783.5	475	180	3.9	OK
26	81.2	BMW I5 EDRIVE40 SEDAN	https://ev-database.org/car/1906/BMW-i5-eDrive...	171	710.0	329940.0	475	193	6.0	OK
27	80.7	BMW I4 M50	https://ev-database.org/car/1519/BMW-i4-M50	179	700.0	332760.0	450	225	3.9	OK
46	96.0	KIA EV9 99.8 KWH AWD	https://ev-database.org/car/1835/Kia-EV9-998-k...	226	740.0	359503.0	425	200	6.0	OK
51	86.4	BYD TANG	https://ev-database.org/car/1783/BYD-TANG	240	330.0	335580.0	360	180	4.6	OK
55	74.0	KIA EV6 GT	https://ev-database.org/car/1471/Kia-EV6-GT	206	940.0	343053.0	360	260	3.5	OK

Figure 4: *Price Range for Electric Vehicles*

```
print(df_ev[['Battery', 'Range']].corr())
```

```
          Battery     Range
Battery  1.000000  0.881937
Range    0.881937  1.000000
```

Figure 5: *Battery size and Range Analysis*

4. Limit Checks

To execute limit checks as part of data validation, two conditions were used to identify entries that may exceed usual performance bounds for electric vehicles. First, in Figure 6, the dataset was filtered to find EVs with a driving range more than 600 km, which could indicate exceptional performance or probable data entry errors. Second in Figure 7, acceleration values (ranging from 0 to 100 km/h) were calculated, and any records with values less than 1 second or greater than 15 seconds were noted. These limitations assist keep acceleration numbers within realistic expectations. Both filters produce simplified outputs that include only important columns for targeted human inspection and verification of extreme or potentially inaccurate values.

```
# Filter the DataFrame for cars with a range greater than 600 km
result = df_ev[df_ev['Range'] > 600][['Car_name', 'Range']]

# Display the result as a DataFrame
result
```

	Car_name	Range
67	Lucid Air Dream Edition R	685
86	Mercedes EQS 450+	635
131	Lucid Air Grand Touring	665
147	Lucid Air Dream Edition P	665
226	Mercedes EQS 450 4MATIC	610

Figure 6: *Cars with a Range Greater than 600 km*

```
df_ev[(df_ev['acceleration..0.100.'] < 1) | (df_ev['acceleration..0.100.'] > 15)]
```

	Battery	Car_name	Car_name_link	Efficiency	Fast_charge	Price.DE.	Range	Top_speed	acceleration..0.100.	Brand
25	25.0	Dacia Spring Electric 45	https://ev-database.org/car/1705/Dacia-Spring-...	152	180.0	22550.0	165	125	19.1	Dacia

Figure 7: *Acceleration Calculation 1s or Greater Than 15s*

5. Review of the data to identify outlier

The purpose of reviewing the data to identify outliers is to detect any data points that significantly differ from the rest of the dataset, which may indicate errors, rare cases, or exceptional values. In the context of electric vehicle data, identifying outliers in top speed can help analysts spot cars that are unusually fast or slow compared to the majority. The data is then sorted in descending order based on the. Top_speed column, allowing the user to quickly identify vehicles with exceptionally high or low speeds. The output displays the car names alongside their respective top speeds, highlighting models such as the Maserati GranTurismo Folgore with a top speed of 320 km/h and the Tesla Model S Plaid at 282 km/h, which represent the upper extreme. Conversely, vehicles like the Dacia Spring Electric 45, with a top speed of just 125 km/h, represent the lower extreme. Figure 8 visualize inspection aids in spotting potential outliers within the dataset, which may require further statistical analysis or consideration during data pre-processing.

```
# Load the dataset
df_ev = pd.read_csv('EV_cars.csv')

# Sort by top speed to check for unusually high or low values
print(df[['Car_name', 'Top_speed']].sort_values(by='Top_speed', ascending=False))
```

```
                          Car_name  Top_speed
215       Maserati GranTurismo Folgore        320
17                  Tesla Model S Plaid        282
131            Lucid Air Grand Touring        270
147           Lucid Air Dream Edition P        270
67            Lucid Air Dream Edition R        270
..                                 ...        ...
334      Toyota PROACE Shuttle L 50 kWh        130
336    Citroen e-SpaceTourer XL 75 kWh        130
359             Fiat E-Ulysse L3 50 kWh        130
122   Dacia Spring Electric 65 Extreme        125
25              Dacia Spring Electric 45        125

[360 rows x 2 columns]
```

Figure 8: *Review of The Data to Identify Outliers*

6. Assessment of data by subject area experts

Assessment of data by subject area experts refers to the process of having professionals who are knowledgeable in a specific field review and validate the dataset used for analysis. These experts evaluate whether the data is accurate, complete, consistent, and relevant for the intended purpose. The output shown the result of a filter applied to identify electric vehicles (EVs) with battery capacities less than 20 kWh. After performing the necessary data cleaning and conversion, the script checks for EVs that meet this condition. However, Figure 9 indicates the output displays an empty data frame, which means that no vehicles in the dataset have battery capacities below 20 kWh.

```
df_ev.columns = df_ev.columns.str.strip()

# Convert to numeric
df_ev['Battery'] = pd.to_numeric(df_ev['Battery'], errors='coerce')

# Identify small battery capacity
small_battery_ev = df_ev[df_ev['Battery'] < 20]

print("EVs with battery capacity < 20 kWh:")
print(small_battery_ev[['Car_name', 'Battery']])
```

```
EVs with battery capacity < 20 kWh:
Empty DataFrame
Columns: [Car_name, Battery]
Index: []
```

Figure 9: *Assessment of Data by Subject Area Experts*

7. Missing values

Missing values refer to the absence of data in one or more fields of a dataset. In the context of data analysis, missing values can occur for many reasons, such as human error during data entry, equipment malfunctions during data collection, or inconsistencies in data formats when merging datasets. These gaps in the data can affect the quality of analysis and the accuracy of machine learning models if not properly addressed. Figure 10 shows the count of missing values for each column then it will filter the column that have missing values.

```
# Load the data
df_ev = pd.read_csv("EV_cars.csv")

# Strip spaces in column names (optional good practice)
df_ev.columns = df_ev.columns.str.strip()

# Count missing values in each column
missing_counts = df_ev.isnull().sum()

# Filter columns with missing values
missing_columns = missing_counts[missing_counts > 0]

print("Columns with missing values:")
print(missing_columns)
```

```
Columns with missing values:
Fast_charge      2
Price.DE.       51
dtype: int64
```

Figure10: *Detecting Column with Missing Values*

8. Smoothing noisy data

Smoothing noisy data is a crucial pre-processing step in data analysis that involves reducing random variations or fluctuations, often referred to as "noise," in order to highlight the underlying patterns or trends within the dataset. By doing this in Figure 11, it shows the smoothed battery capacity using a moving average.

```
# Load data
df_ev = pd.read_csv("EV_cars.csv")
df_ev.columns = df_ev.columns.str.strip()

# Convert to numeric (in case there are errors or strings)
df_ev['Battery'] = pd.to_numeric(df_ev['Battery'], errors='coerce')

# Apply a 3-point moving average
df_ev['BatteryCapacity_Smoothed'] = df_ev['Battery'].rolling(window=3).mean()

# Show a few rows
print(df_ev[['Battery', 'BatteryCapacity_Smoothed']].head(10))
```

```
   Battery  BatteryCapacity_Smoothed
0     75.0                       NaN
1     57.5                       NaN
2     60.5                 64.333333
3     61.7                 59.900000
4     75.0                 65.733333
5     57.5                 64.733333
6     71.0                 67.833333
7     64.0                 64.166667
8     44.0                 59.666667
9     82.5                 63.500000
```

Figure11: *Analysis of Smooth Noisy Data of Battery Capacity*

9. Identify and remove outlier

First in Figure 12, extra spaces in column names are stripped for consistency. Then in Figure 13, it filters the dataset to find these unusually high or low EV prices and prints the names and RM prices of the outlier vehicles. In the output, cars like the BMW i7 xDrive60 and Audi e-tron GT RS are identified as having prices outside the normal range. This process is essential in identifying pricing anomalies that could affect data analysis or decision-making.

```
import pandas as pd

# Read CSV and clean column names
df_ev = pd.read_csv('EV_cars.csv')
df_ev.columns = df_ev.columns.str.strip()

# Convert Euro to RM and rename column
df_ev['Price.DE.'] = pd.to_numeric(df_ev['Price.DE.'], errors='coerce')
df_ev['RM'] = df_ev['Price.DE.'] * 4.7
df_ev.drop(columns='Price.DE.', inplace=True)

# IQR calculation
Q1 = df_ev['RM'].quantile(0.25)
Q3 = df_ev['RM'].quantile(0.75)
IQR = Q3 - Q1

# Bounds
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Detect outliers
price_outliers = df_ev[(df_ev['RM'] < lower_bound) | (df_ev['RM'] > upper_bound)]

# Display
print("EVs with unusually high or low prices (in RM):")
print(price_outliers[['Car_name', 'RM']])
```

Figure12: *Identify Price Outliers*

```
EVs with unusually high or low prices (in RM):
                                 Car_name          RM
31                           BMW i7 xDrive60   657530.0
48                         Audi e-tron GT RS   686435.0
67                 Lucid Air Dream Edition R  1024600.0
107                     Tesla Model X Plaid   545059.0
116                              BMW iX M60   639670.0
125              Mercedes EQS SUV 580 4MATIC   636539.8
131                   Lucid Air Grand Touring   747300.0
147                Lucid Air Dream Edition P  1024600.0
170                           Lotus Eletre R   709653.0
183                        Lucid Air Touring   606300.0
192                  Mercedes EQS 580 4MATIC   666013.5
205                          BMW i7 eDrive50   543790.0
207                      Porsche Taycan GTS   657558.2
210                        BMW i7 M70 xDrive   854460.0
231                   Porsche Taycan Turbo S   929378.0
254              Mercedes EQS AMG 53 4MATIC+   728542.3
262              Mercedes EQS SUV 500 4MATIC   585361.5
268                  Mercedes EQS 500 4MATIC   589276.6
269                   Porsche Taycan 4S Plus   564380.7
273              Mercedes EQS SUV 450 4MATIC   538662.3
286           Mercedes EQE SUV AMG 43 4MATIC   587124.0
289                        Porsche Taycan 4S   538427.3
296          Mercedes EQE SUV AMG 53 4MATIC+   655358.6
306                     Porsche Taycan Turbo   772774.0
311          Porsche Taycan 4S Cross Turismo   563595.8
324         Porsche Taycan GTS Sport Turismo   662032.6
325       Porsche Taycan Turbo Cross Turismo   779485.6
329     Porsche Taycan Turbo S Cross Turismo   936089.6
335     Porsche Taycan Turbo S Sport Turismo   933852.4
341     Porsche Taycan 4S Plus Sport Turismo   564380.7
344          Porsche Taycan 4S Sport Turismo   538427.3
349       Porsche Taycan Turbo Sport Turismo   777248.4
```

Figure13: *The Output of EV's With High or Low Prices*

10. Resolve inconsistencies

The output displays in Figure 14 shows the cleaned dataset after missing values in the Range and Top_speed columns were removed. The code fills in any missing values with the mean of each column, which is a popular imputation strategy for ensuring data completeness. Alternatively, rows with missing values in these columns could be removed. The cleaned dataset, as stated by the head() method, demonstrates that no NaN values exist in the Range and Top_speed columns. Each row now has detailed information on various EV models, including battery capacity, car name, efficiency, quick charging capability, and acceleration. This pre-processing phase prepares the dataset for subsequent analysis or modeling. Furthermore, the method corrects errors in the dataset to maintain uniformity and accuracy.

```
# Fill missing values with the mean (for numerical columns like Range or Top_speed)
df_ev['Range'] = df_ev['Range'].fillna(df_ev['Range'].mean())
df_ev['Top_speed'] = df_ev['Top_speed'].fillna(df_ev['Top_speed'].mean())
# Alternatively, drop rows with missing values in specific columns
df_ev = df_ev.dropna(subset=['Range', 'Top_speed'])
# Print the cleaned dataset
print(df_ev.head())
```

```
   Battery                             Car_name  \
0     75.0  Tesla Model Y Long Range Dual Motor
1     57.5                        Tesla Model 3
2     60.5                           BYD ATTO 3
3     61.7                MG MG4 Electric 64 kWh
4     75.0  Tesla Model 3 Long Range Dual Motor

                                     Car_name_link  Efficiency  Fast_charge  \
0  https://ev-database.org/car/1619/Tesla-Model-Y...         172        670.0
1     https://ev-database.org/car/1991/Tesla-Model-3         137        700.0
2        https://ev-database.org/car/1782/BYD-ATTO-3         183        370.0
3  https://ev-database.org/car/1708/MG-MG4-Electr...         171        630.0
4  https://ev-database.org/car/1992/Tesla-Model-3...         149        780.0

   Range  Top_speed  acceleration..0.100.          RM
0    435        217                   5.0  1303685.53
1    420        201                   6.1  1020999.80
2    330        160                   7.3   985766.25
3    360        160                   7.9   883379.10
4    505        201                   4.4  1219809.80
```

Figure14: *Resolving Inconsistencies of The Data*

Result of Comprehensive Visualization

This bar chart in Figure 15 displays the top ten electric vehicle (EV) brands with the most model listings in the dataset. To do this, we first extracted the brand name from the Car_name column by extracting the first word (for example, "Tesla" from "Tesla Model 3"). We then counted how many EV models each brand provides and chose the top ten brands based on model count. The horizontal bar chart clearly shows which brands are most prevalent in the EV market, according to the dataset. For example, if Tesla is at the top, it means that it has the most diverse EV line-up among the entries. This type of analysis helps us understand brand dominance and market diversity in the EV business.

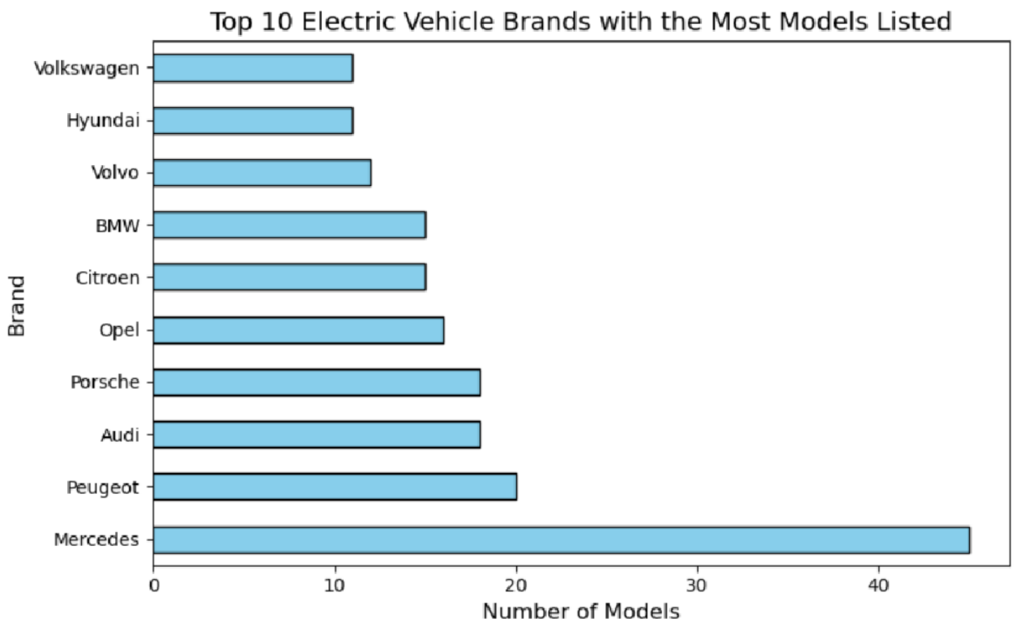

Figure15: *Bar Chart of Top 10 Electric Vehicle Brands with the Most Models Listed*

The pie chart in Figure 16 illustrates the distribution of electric vehicles based on their 0-100 km/h acceleration times, which are categorized into four speed ranges: 0-5 seconds, 5-10 seconds, 10-15 seconds, and 15-20 seconds. To create this graph, we divided the vehicles into bins based on their acceleration from 0 to 100 km/h, using the acceleration column from 0.100. Each bin represents a performance category, and the chart depicts the percentage of vehicles in each group.

The generated visualisation demonstrates how performance varies among the EVs in the dataset. For example, suppose a substantial percentage of the chart falls in the "5-10 sec" category. In that case, it means that most EVs have a moderate acceleration time, which is common in both consumer and mid-range performance models.

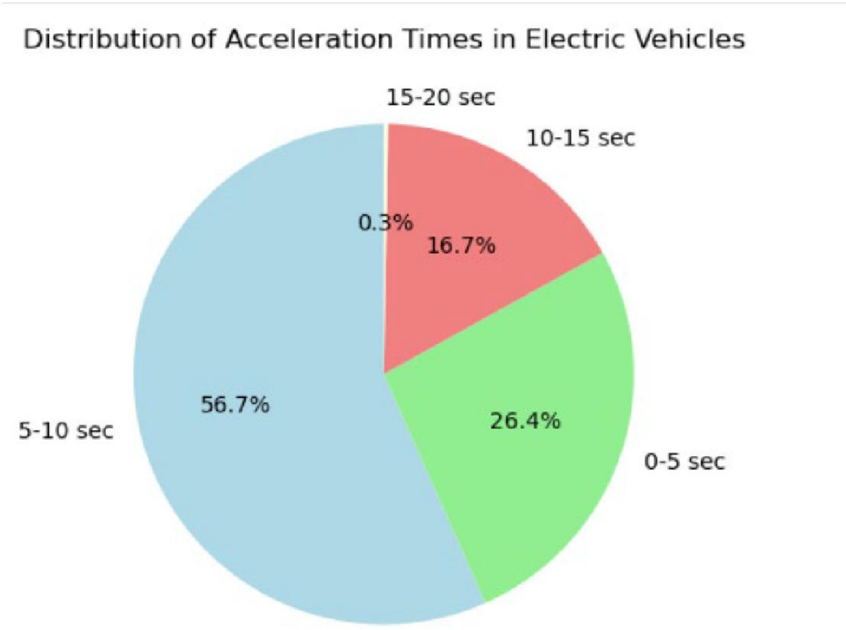

Figure 16: *Pie Chart of Distribution of Acceleration Times in Electric Vehicles*

The distribution of top speeds (in km/h) for electric vehicles in the dataset is displayed in the histogram in Figure 17. The cars maximum speed is shown by the x-axis, while the number of vehicles falling within each speed range is displayed by the y-axis. To visualise the general distribution form, we enabled the kernel density estimate (KDE) line and used Seaborns 15-bin histplot function to arrange the speeds into intervals.

The graphic shows us which EV models have the most common speed ranges. A peak in the histogram between 150 and 200 km/h, for instance, indicates that the majority of the EVs in the dataset are built to run in that range. The KDE line offers a smooth curve that illustrates the overall distribution of the data.

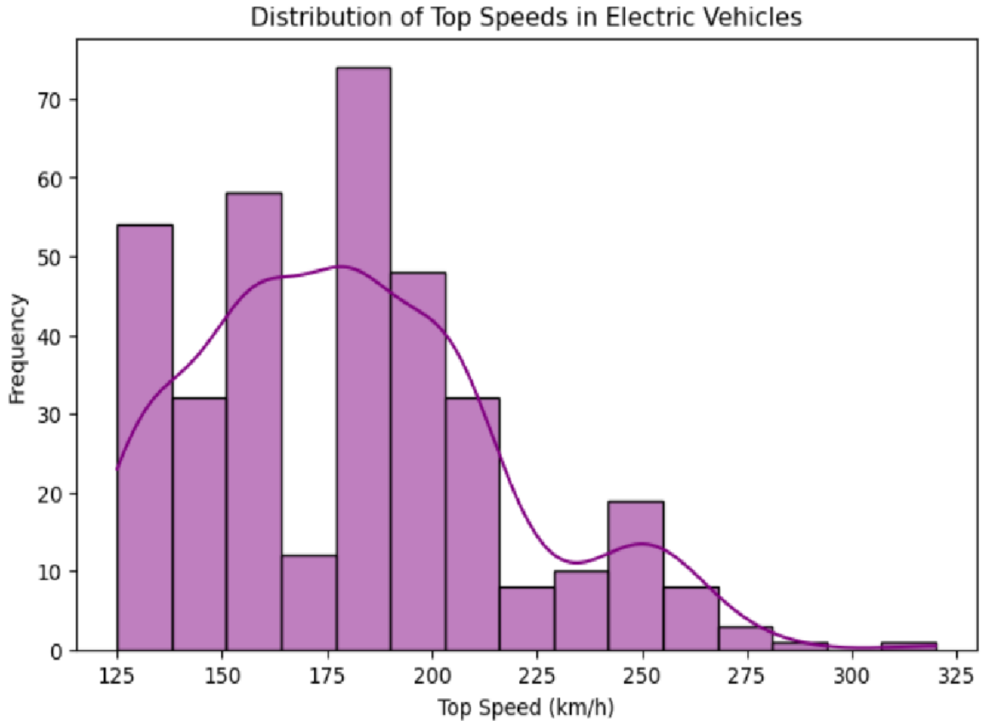

Figure 17: *Histogram of Distribution of Top Speeds in Electric Vehicles*

The top 10 brands of electric vehicles are listed in the bar chart in Figure 18, ranked by their average car pricing in Malaysian Ringgit (RM). Each brand's average price was determined by taking the mean of the RM price values and applying the groupby() function to the Brand column.

Which EV brands fall into the premium or luxury market sector can be determined with the aid of the chart. While the brands not displayed here (with lower averages) might prioritise affordability, those with higher average pricing are probably offering automobiles with additional features or high performance.

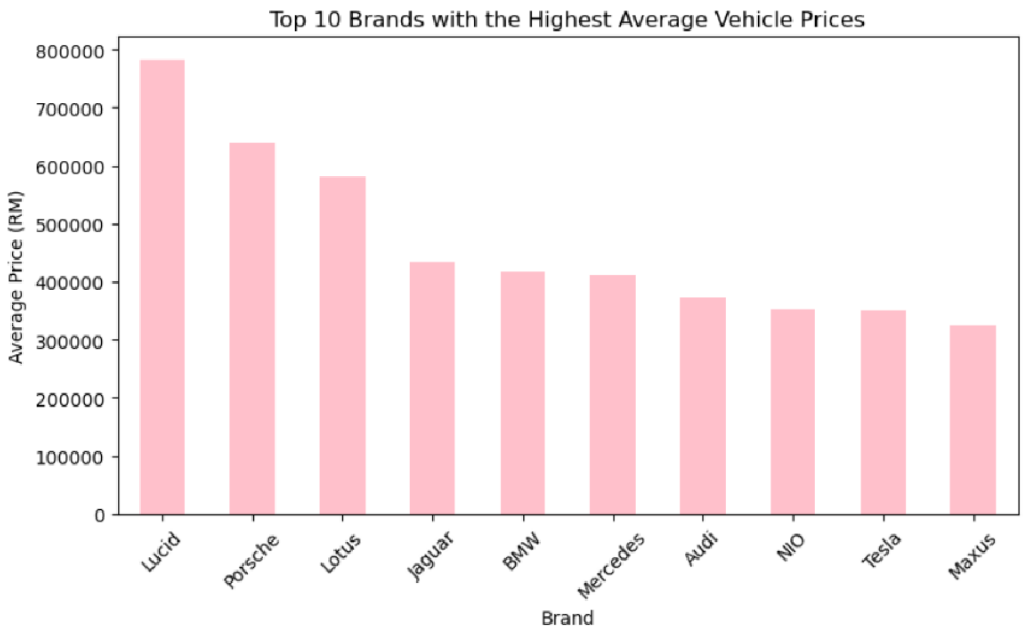

Figure 18: *Bar Chart of Top 10 Brands with the Highest Average Vehicle Prices*

The code in Figure 19 is designed to produce a bar chart that displays the proportion of electric vehicles that support fast charging versus those that do not. The number of vehicles that support fast charging is represented by a green bar, while the number of vehicles that do not is represented by a red bar. The number of cars in each category is shown by the height of each bar.

```
plt.figure(figsize=(6, 4))
plt.bar(['Supports Fast Charging', 'Does Not Support'], [fast_charge_support, no_fast_charge], color=['green', 'red'])
plt.title('Electric Vehicles Fast Charging Support')
plt.ylabel('Number of Vehicles')
plt.tight_layout()
plt.show()
```

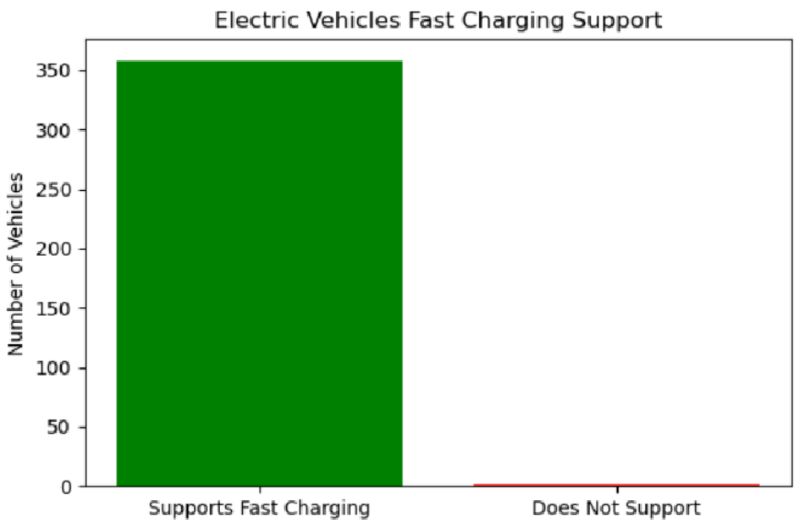

Figure 19: *Bar Chart of Electric Vehicles Fast Charging Support*

Conclusion

The descriptive analysis elaborated in the previous section, along with data management for quality evaluation, illustrates the extensive process of transforming raw data into high-quality data before analysis, ensuring substantial validity. In the subsequent attempt, the data will undergo further analysis using predictive analytics, which utilises machine learning techniques, as well as prescriptive analytics employing optimisation algorithms and simulation.

References

ahay38. (2025, June 18). *Global Electric Vehicle Sales Set for Record-Breaking Year, Even as US Market Slows Sharply, BloombergNEF Finds | BloombergNEF*. BloombergNEF. https://about.bnef.com/insights/clean-transport/global-electric-vehicle-sales-set-for-record-breaking-year-even-as-us-market-slows-sharply-bloombergnef-finds/?utm_source=chatgpt.com

IEA. (2025). *Trends in electric car markets – Global EV Outlook 2025 – Analysis - IEA*. IEA. https://www.iea.org/reports/global-ev-outlook-2025/trends-in-electric-car-markets-2?utm_source=chatgpt.com

Lye, G. (2025, July 17). *EV sales up 91.4% in H1 2025, but full electric vehicles only make up around 3.4% of new car sales in Malaysia*. Paul Tan's Automotive News. https://paultan.org/2025/07/17/maa-ev-sales-data-1h-2025/?utm_source=chatgpt.com

shahrilmd. (2025, February 3). *Powering the Future: Accelerating Malaysia's EV Charging Revolution for Sustainable Mobility*. MIDA | Malaysian Investment Development Authority. https://www.mida.gov.my/powering-the-future-accelerating-malaysias-ev-charging-revolution-for-sustainable-mobility/?utm_source=chatgpt.com

PLANMalaysia - MEVnet. (2025). Planmalaysia.gov.my. https://www.planmalaysia.gov.my/mevnet/?utm_source=chatgpt.com

Fatih İlhan. (2023). *Electric Vehicle Specifications and Prices*. Kaggle.com. https://www.kaggle.com/datasets/fatihilhan/electric-vehicle-specifications-and-prices

SIG CS@e-Learning
Unit Penerbitan
Jabatan Sains Komputer & Matematik
Universiti Teknologi MARA Cawangan Pulau Pinang

e-ISBN : 978-629-98755-7-4

Design of the cover powered by
https://www.free-powerpoint-templates-design.com/