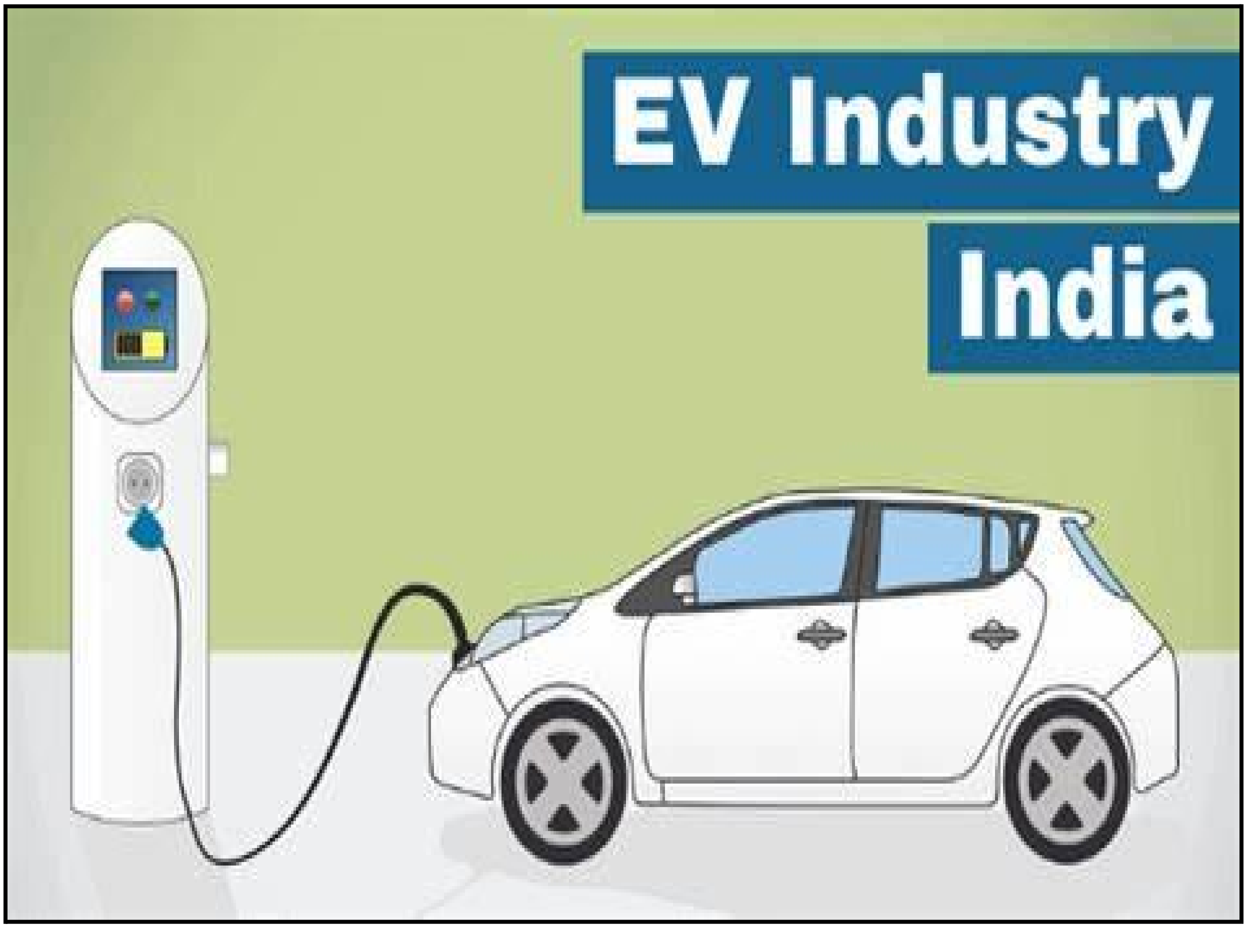
<u>ELECTRIC VEHICLE MARKET ANALYSIS</u>

11/07/2023



<u>Team</u>

Pooja Parmar (Team Leader)
Srinidhi Tarigoppula
Tirath Das
Harish.B

Problem statement: The electronic vehicle market faces challenges such as limited charging infrastructure, high vehicle cost, limited driving range, battery technology concerns, consumer awareness gaps, inadequate government policies, and recycling and sustainable materials requirements. Overcoming these challenges is crucial for widespread EV adoption and market growth.

Github links:-

1.	Pooja Parmar	Pooja-dotc/Electric_Vehicle (github.com)
2.	Srinidhi Tarigoppula	srinidhi-tarigoppula/EV-Market-Segmentation-Analysis (github.com)
3.	Tirath Das	Tirath-01/EV_market_segmentation_analysis: Electric Vehicle Market Segmentation analysis (github.com)
4.	Harish.B	Craziprogrammerharish/Feynnlab-project-ev (github.com)

Abstract:

This project focuses on addressing the challenges faced by the electronic vehicle (EV) market to facilitate its widespread adoption and growth. The primary challenges identified include limited charging infrastructure, high vehicle cost, limited driving range, battery technology and durability concerns, consumer awareness and education gaps, government policies and incentives, and recycling and sustainable materials requirements.

To overcome these challenges, the project proposes several key strategies. Firstly, the development of an extensive and robust charging infrastructure network is crucial to alleviate range anxiety and

enhance the convenience of EV ownership. Secondly, efforts should be made to reduce the upfront cost of EVs through advancements in battery technology and economies of scale in production. Additionally, improving driving range capabilities and addressing battery technology concerns will boost consumer confidence.

Moreover, comprehensive consumer awareness campaigns and educational initiatives are essential to dispel misconceptions and promote the advantages of EVs. Government support, in the form of favorable policies, incentives, and investments in infrastructure, is crucial for market growth and attracting manufacturers to invest in EV production. Lastly, a focus on recycling methods and sustainable materials will minimize the environmental impact associated with EVs.

By addressing these challenges and implementing the proposed strategies, the project aims to accelerate the adoption of EVs, promote sustainable transportation, and contribute to a cleaner and greener future.

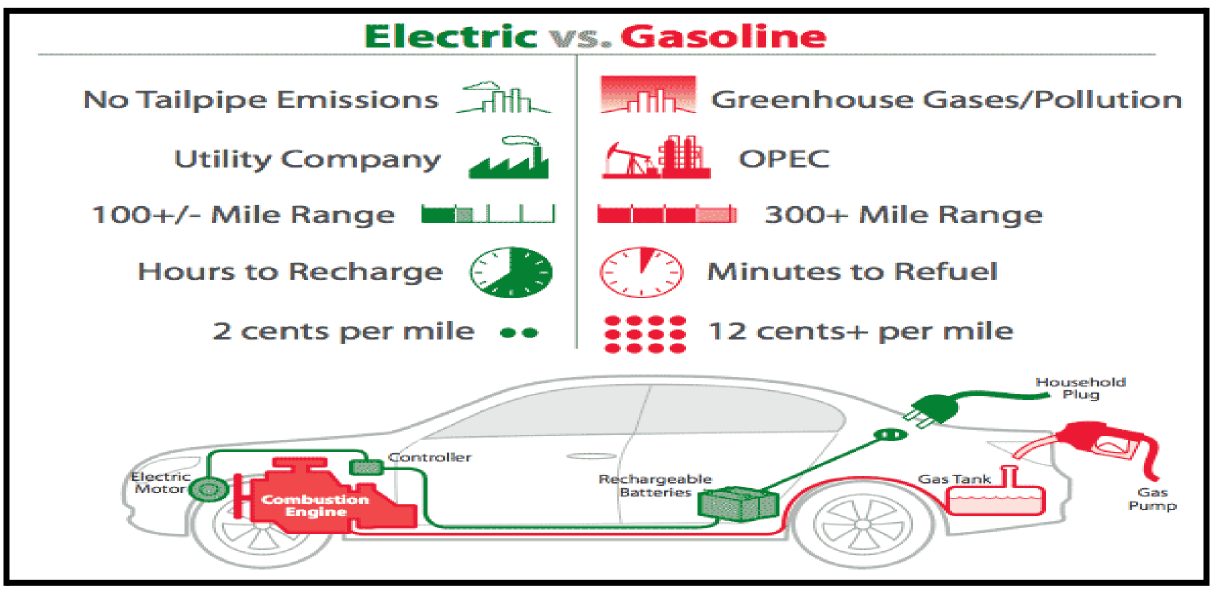


Fig.1

Market Overview :

The electronic vehicle (EV) market is experiencing rapid growth and positive momentum driven by factors such as rising environmental awareness, technological advancements in battery technology, supportive government policies and incentives, expanding charging infrastructure, increased investment and competition, cost reduction, and positive consumer perception of EV benefits. The market's outlook is promising, with continued expansion expected as technology improves, costs decrease, infrastructure expands, and consumer awareness grows. The transition to EVs presents opportunities for economic growth and a sustainable future.

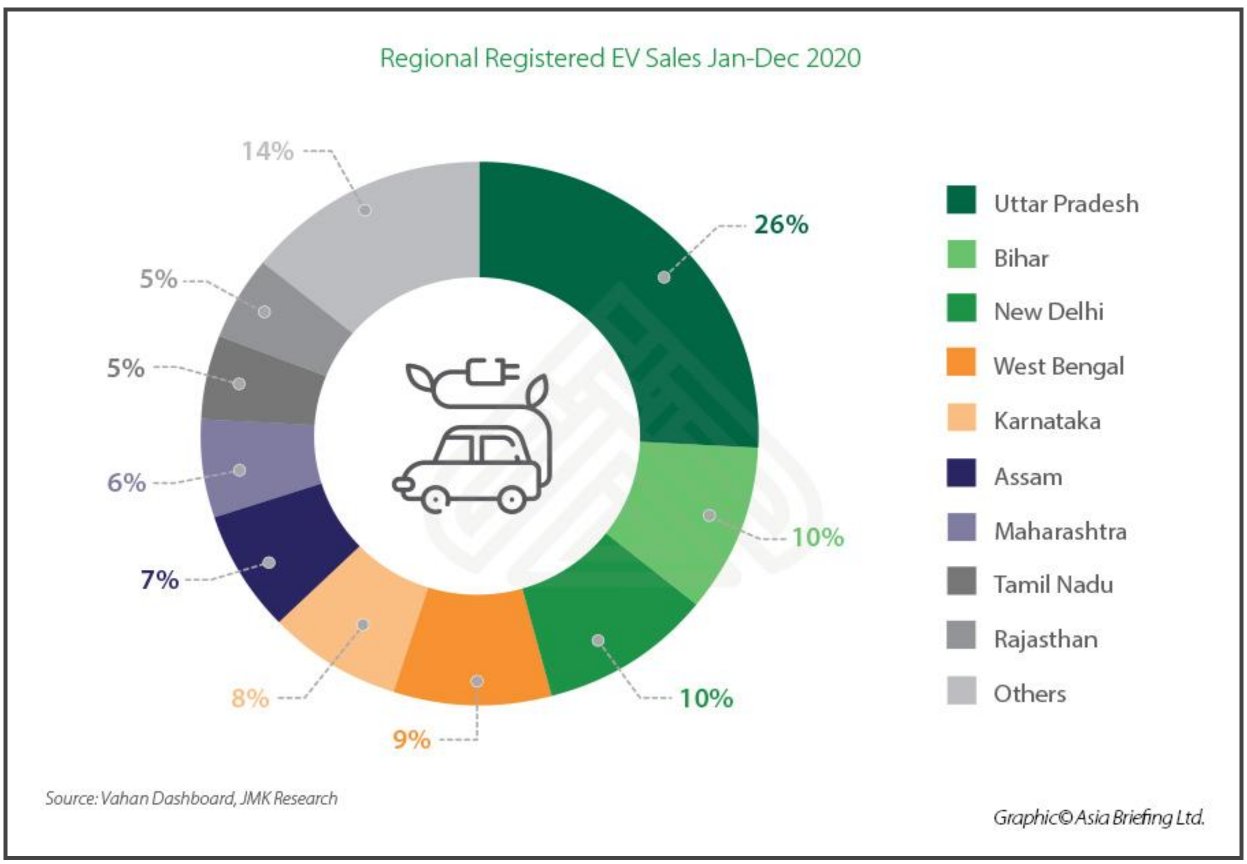


Fig.2

About the data :

We have gathered data manually from different sources. And each analysis is based on the dataset which you can find on the above GitHub links. Each of us has worked on a different dataset so there will be vast analysis of this Electric Vehicle market.

Data Preprocessing and analysis :

Data pre-processing refers to the steps and techniques applied to raw data before it can be used for analysis or machine learning tasks. It involves transforming and cleaning the data to ensure its quality, consistency, and suitability for further processing.

Analysis 1

In this article, we will use Python and its different libraries to analyze the Electric Vehicle Data.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Importing Libraries The analysis will be done using the following libraries :

Pandas: This library helps to load the data frame in a 2D array format and has multiple functions to perform analysis tasks in one go.

Numpy: Numpy arrays are very fast and can perform large computations in a very short time.

Matplotlib / Seaborn: This library is used to draw visualizations. To importing all these libraries, we can use the below code :

Importing Dataset

Fig.3

This dataset shows details about electric cars as per the brand names. As we can see the details about battery capacity, top speed, max power, charging points, price etc.
Let’s analysis this dataset for better visualization:

```python
dataset = pd.read_csv("EV_CARS _INDIA.csv")
dataset.head()
```

	Brand Name	Battery Capacity(kWh)	Acceleration(sec)	TopSpeed(km/h)	Range(km)	Max Power(kW)	Max Torque(Nm)	Transmission	No. of Seats	Charging T(h)	No. of Airbags	Drive Type	Price(Lh)
0	Audi RS e-tron GT	93.4	3.3	250	480	500	830	Automatic	5	9	Yes	AWD	204
1	Audi e-tron GT	93.4	4.1	245	500	523	630	Automatic	5	9	Yes	AWD	179
2	Audi e-tron	95.0	5.7	200	484	300	664	Automatic	5	9	Yes	AWD	123
3	Tata Nexon EV	30.2	9.9	180	312	96	245	Automatic	5	9	Yes	FWD	17
4	Tata Tigor EV	26.0	5.7	120	306	55	170	Automatic	5	9	Yes	FWD	14

Fig.4

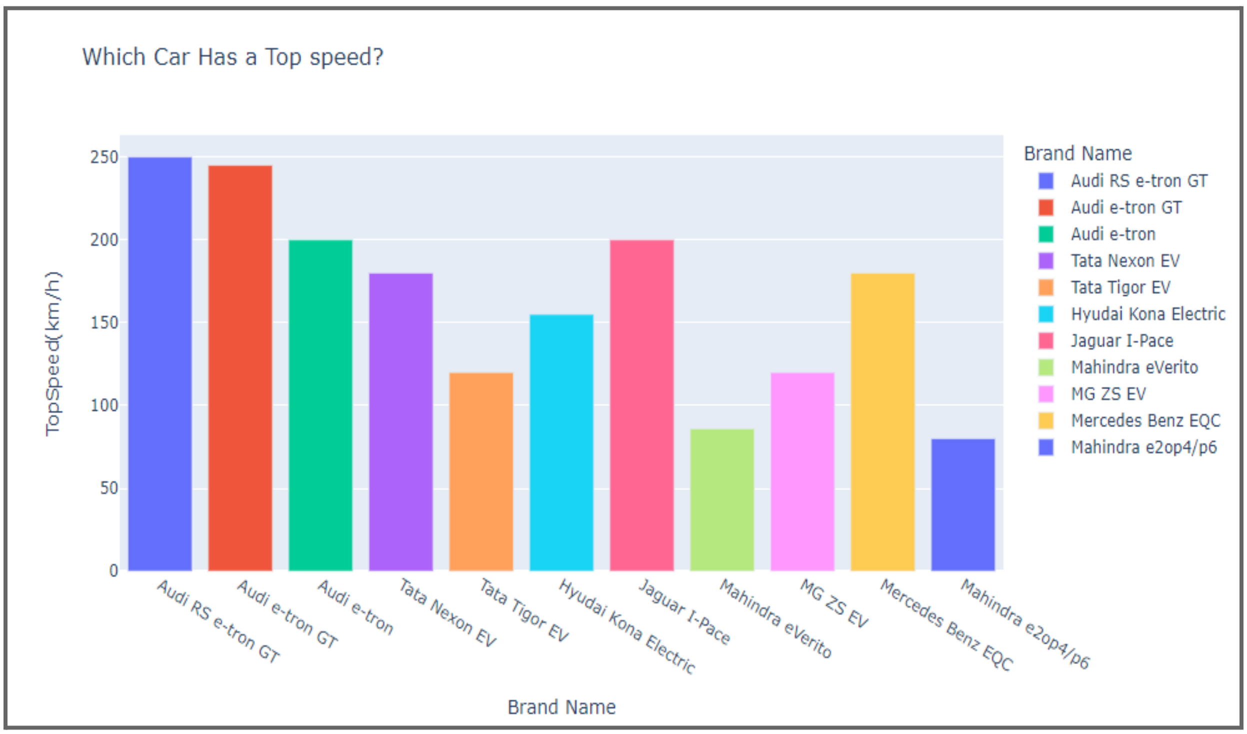
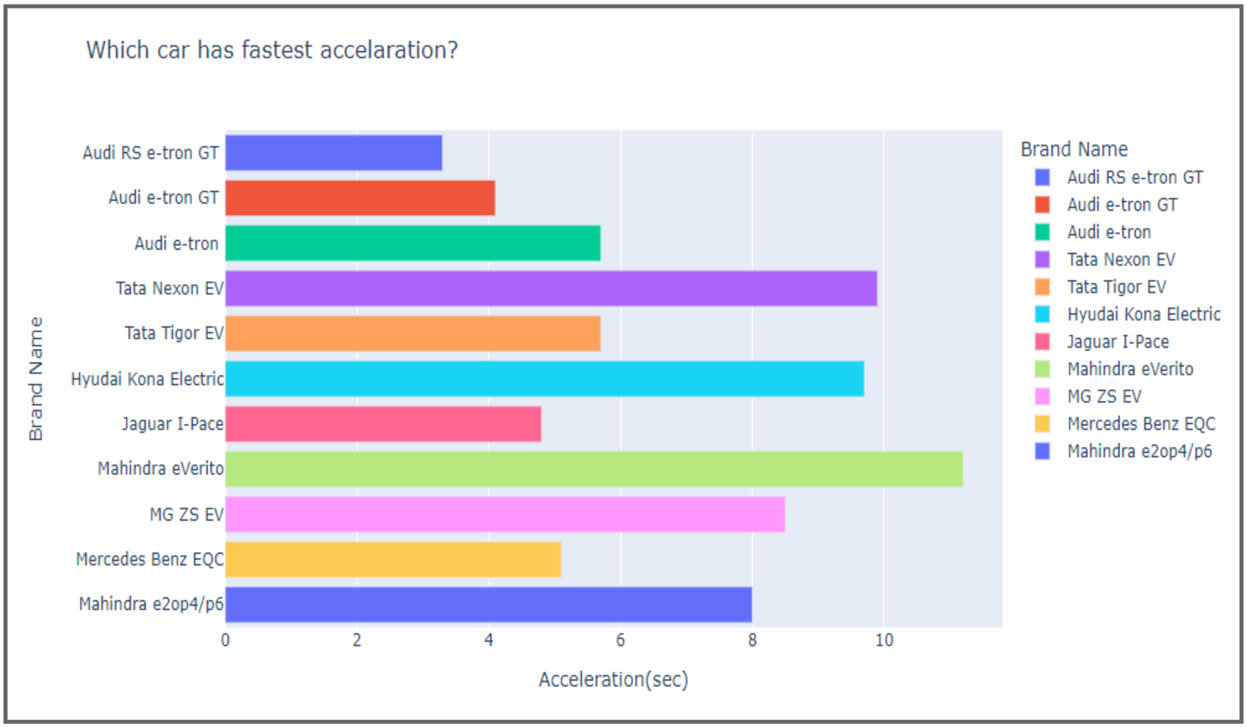
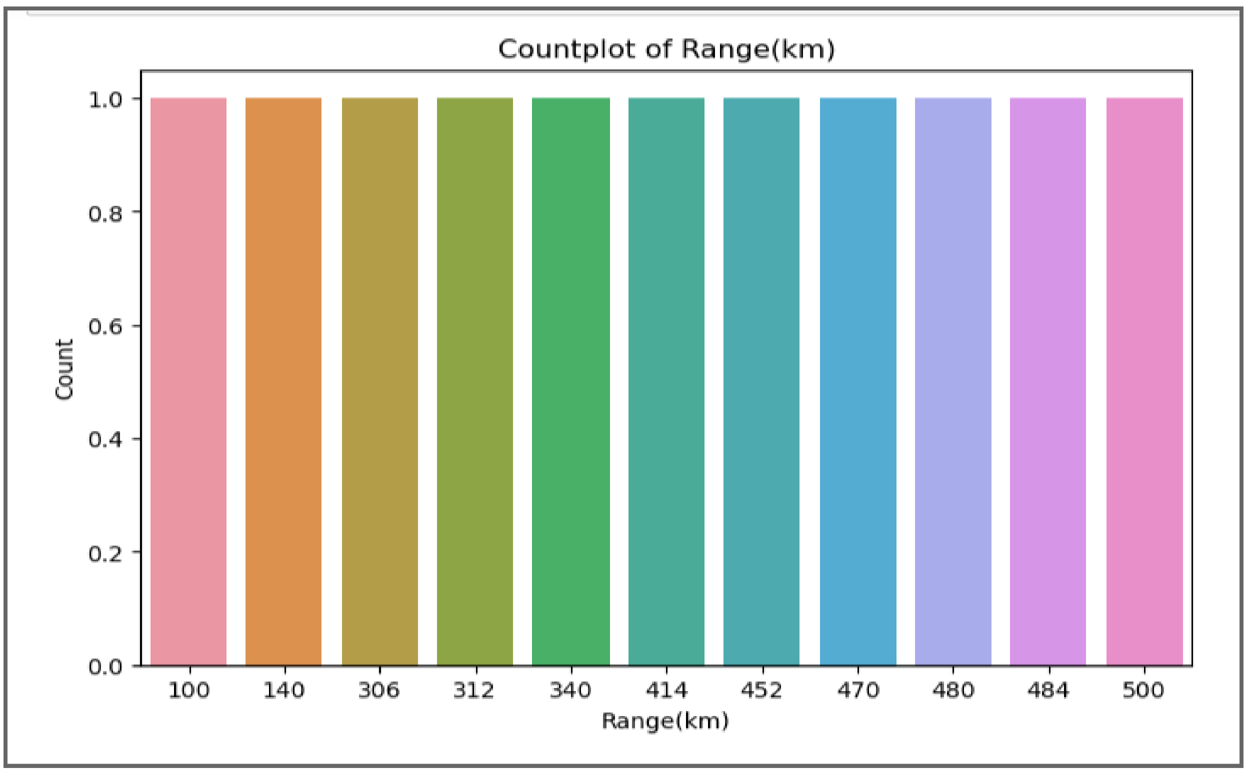


Fig.5



Which car has fastest accelaration?
Brand Name
Audi RS e-tron GT
Audi e-tron GT
Audi e-tron
Tata Nexon EV
Tata Tigor EV
Hyudai Kona Electric
Jaguar I-Pace
Mahindra eVerito
MG ZS EV
Mercedes Benz EQC
Mahindra e2op4/p6
0
2
4
6
8
10
Acceleration(sec)

Fig.6



Countplot of Range(km)
Count
1.0
0.8
0.6
0.4
0.2
0.0
100
140
306
312
340
414
452
470
480
484
500
Range(km)

Fig.7

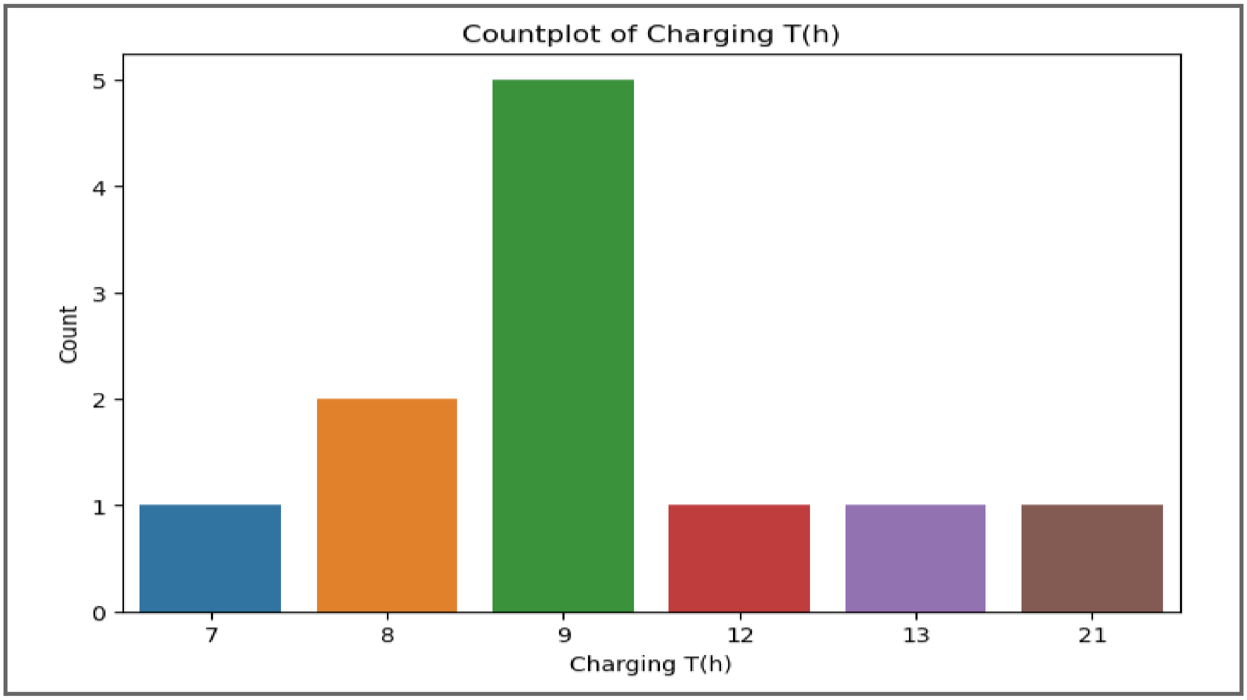


Fig.8

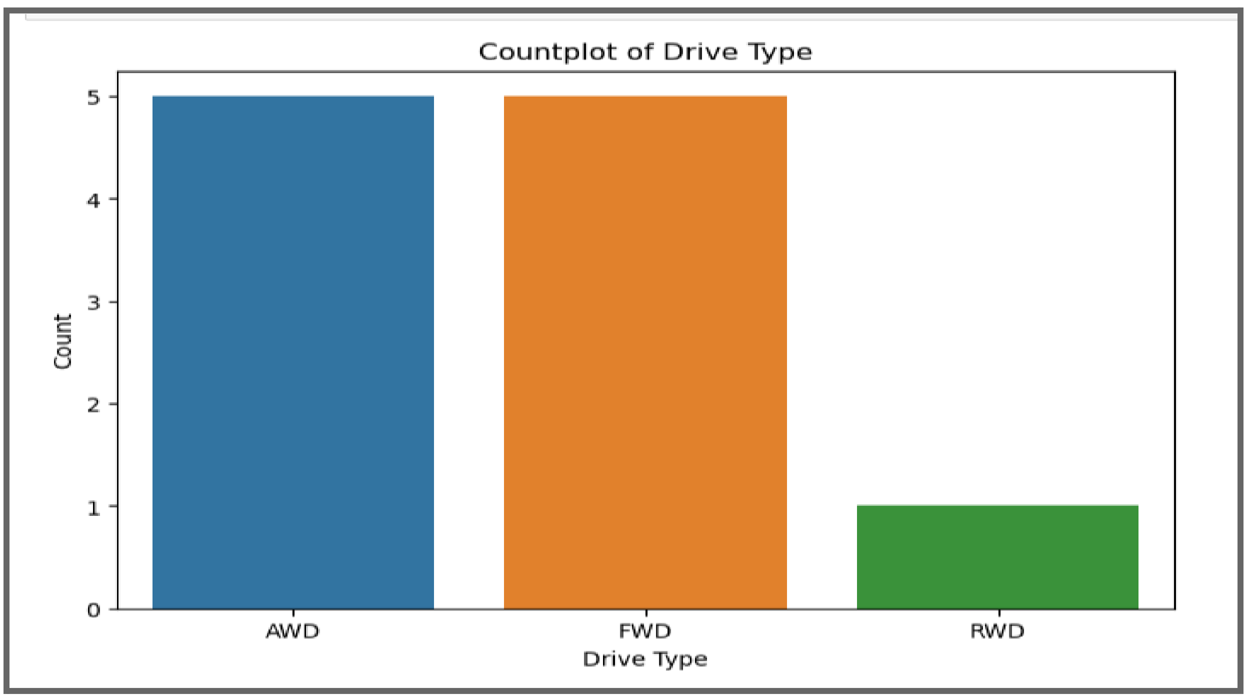


Fig.9

As we can see there are 3 types of Drive in Electric Vehicle
(1). AWD, (2). FWD, (3). RWD

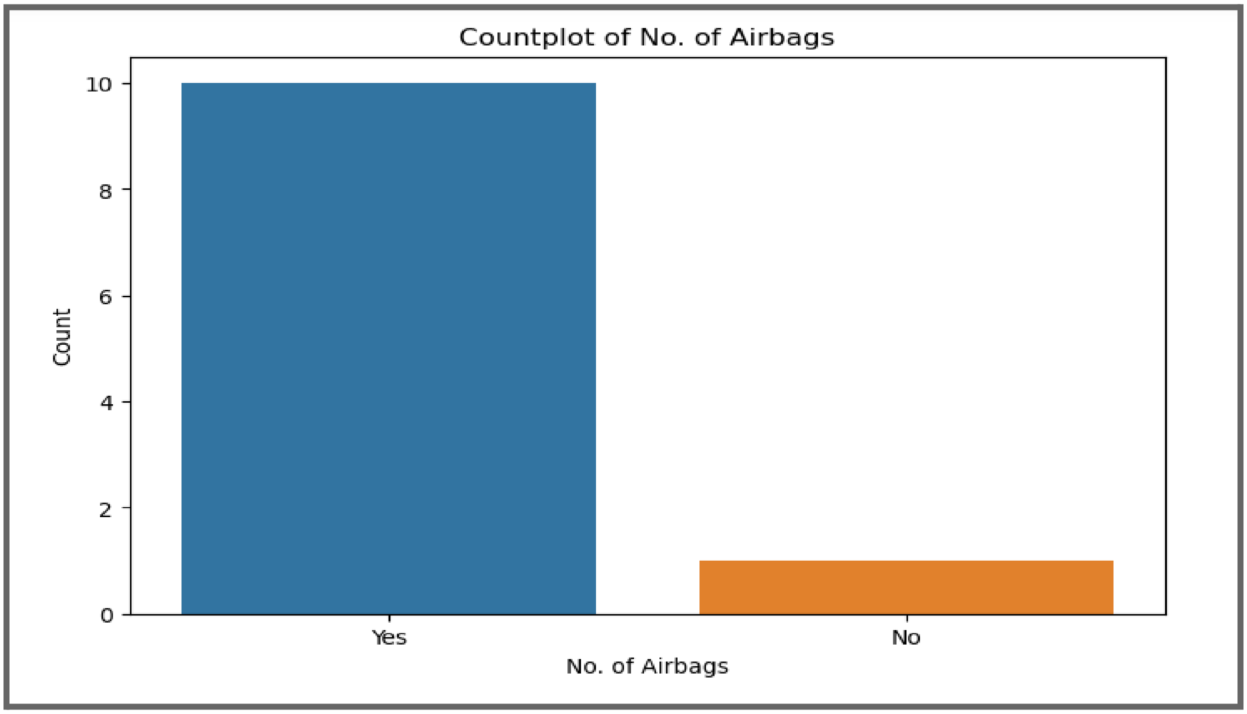

Countplot of No. of Airbags
10
8
6
4
2
0
Count
Yes
No
No. of Airbags

Fig.10

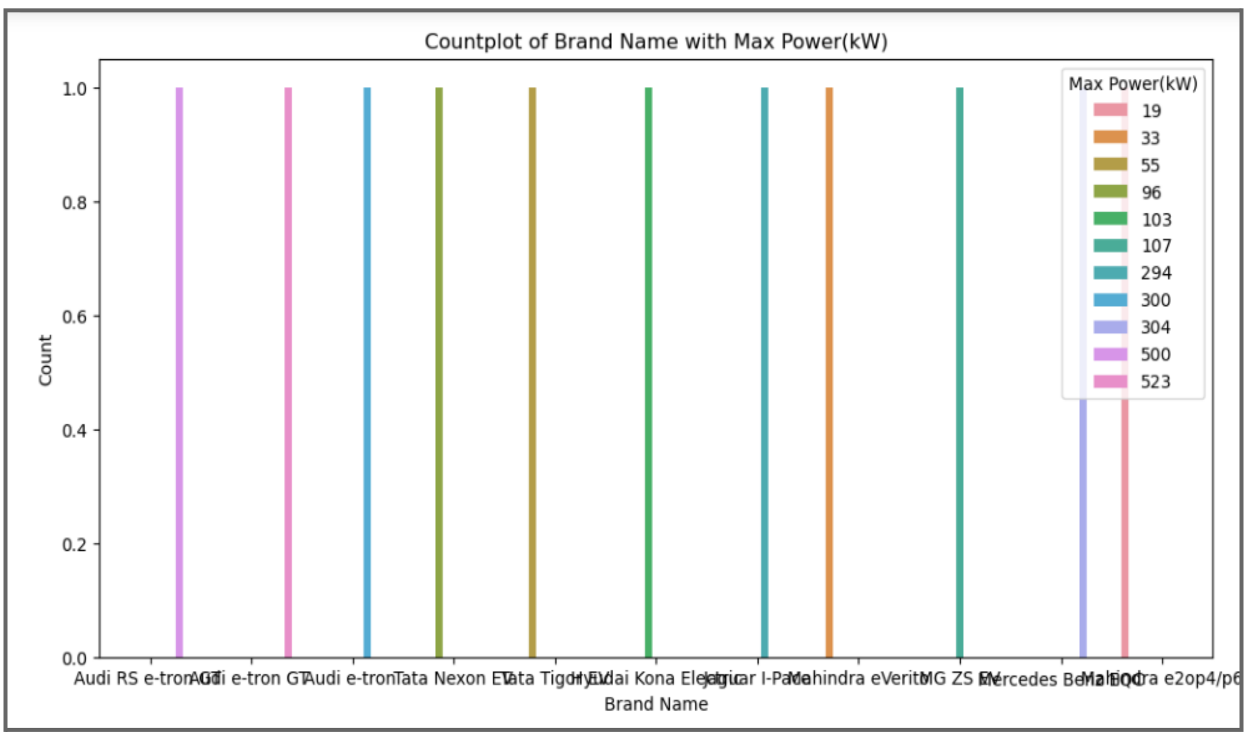

Countplot of Brand Name with Max Power(kW)
1.0
0.8
0.6
0.4
0.2
0.0
Count
Max Power(kW)
19
33
55
96
103
107
294
300
304
500
523
Brand Name

Fig.11

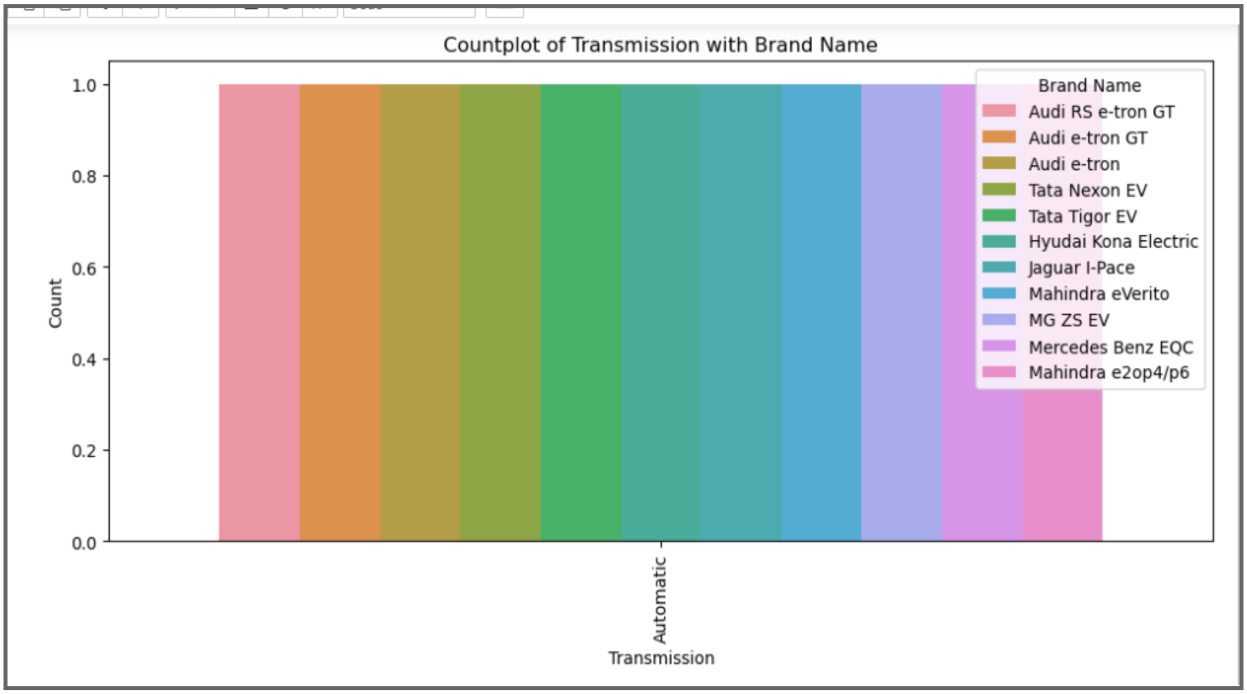

Countplot of Transmission with Brand Name
Count
Transmission
Automatic
Brand Name
Audi RS e-tron GT
Audi e-tron GT
Audi e-tron
Tata Nexon EV
Tata Tigor EV
Hyudai Kona Electric
Jaguar I-Pace
Mahindra eVerito
MG ZS EV
Mercedes Benz EQC
Mahindra e2op4/p6

Fig.12

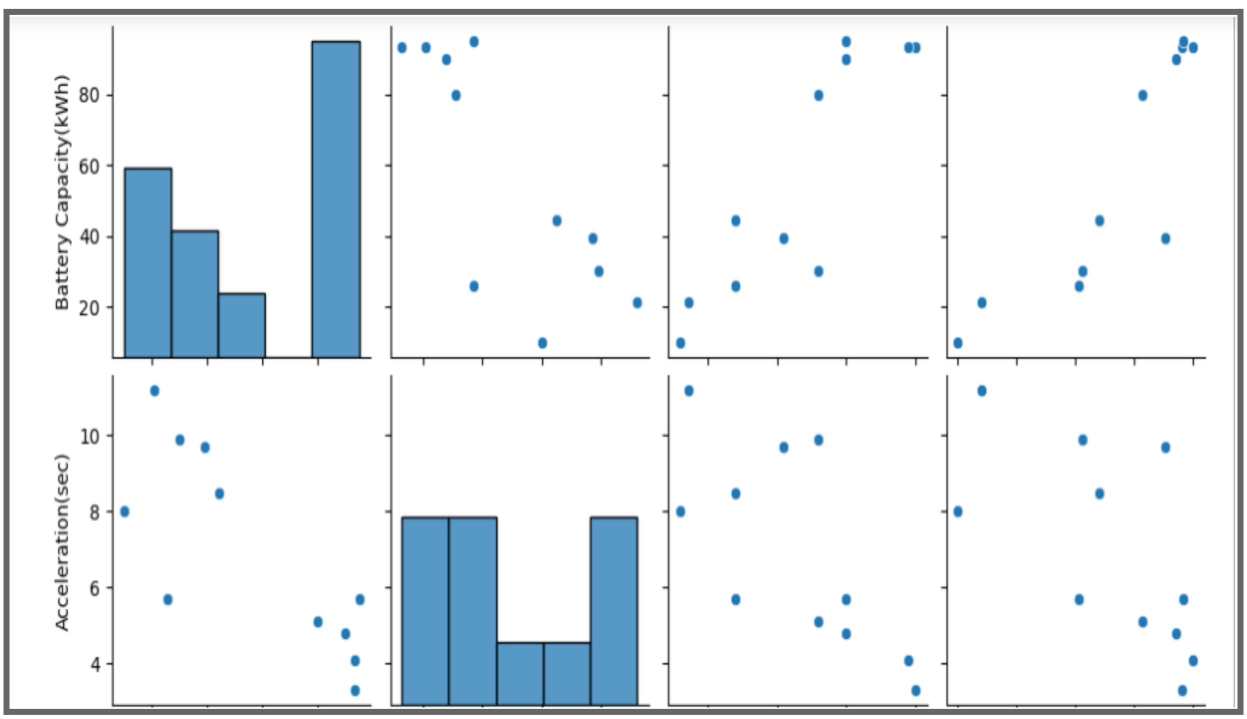

Battery Capacity(kWh)
Acceleration(sec)

Fig.13

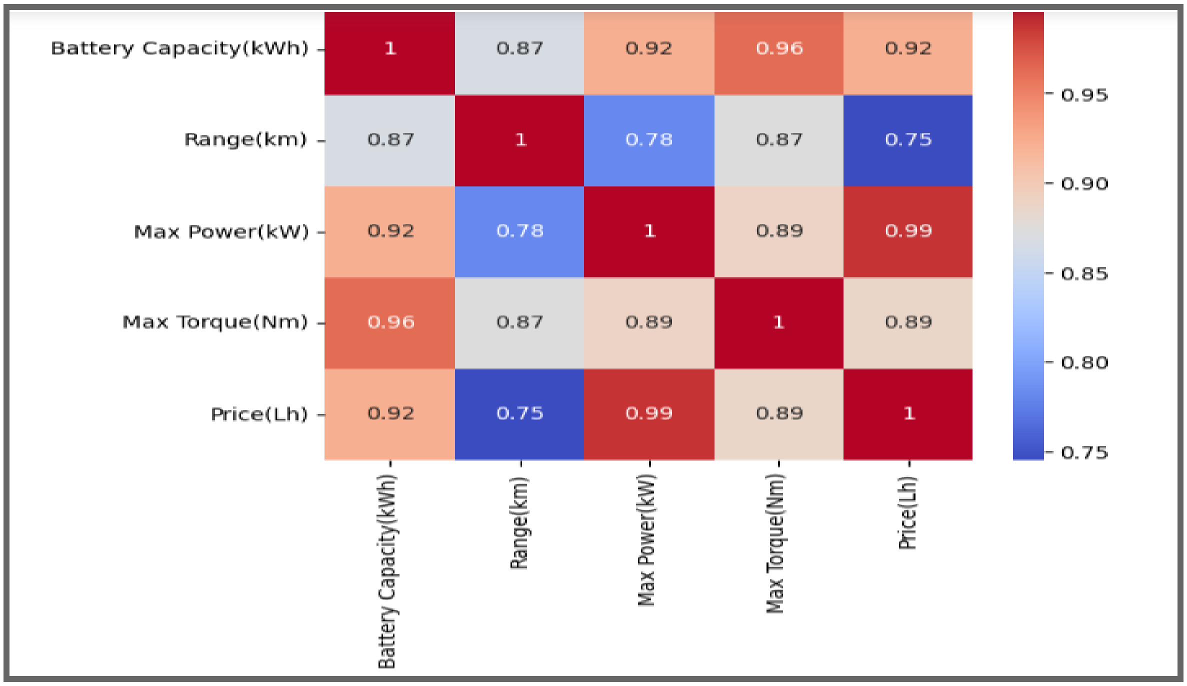


Fig.14

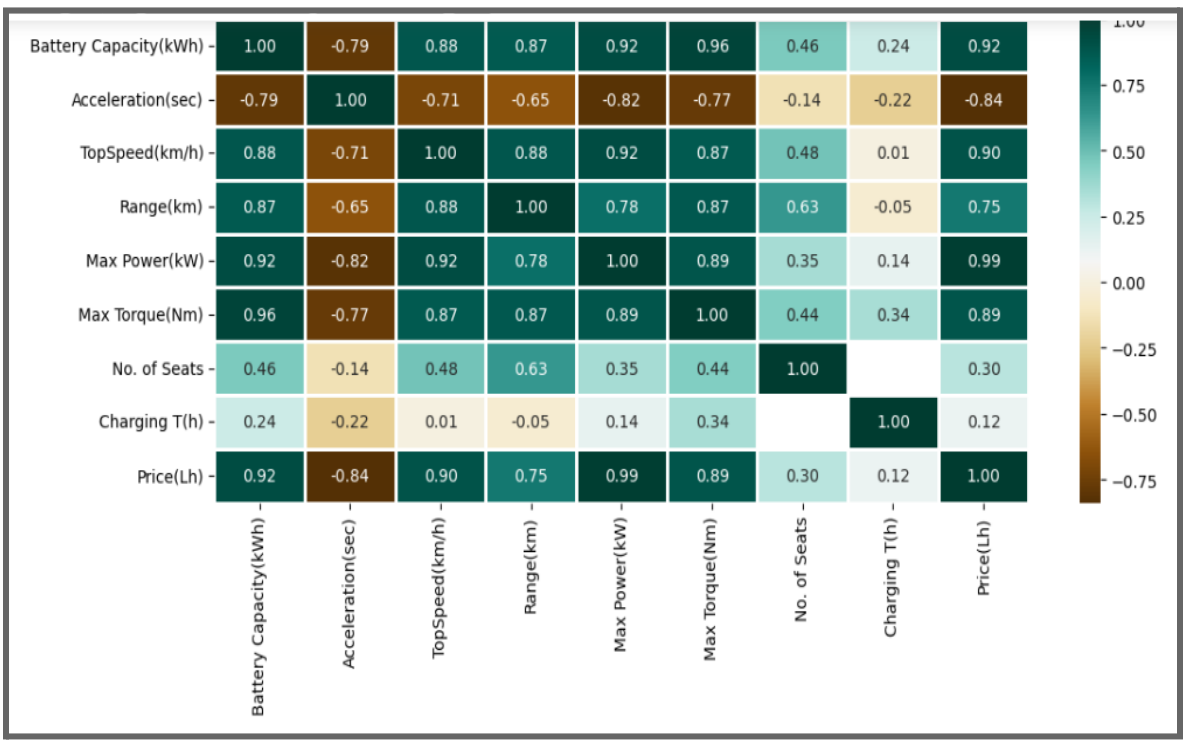


Fig.15

As we can see there are Highly negative correlation between top speed(km/h) and acceleration(sec)

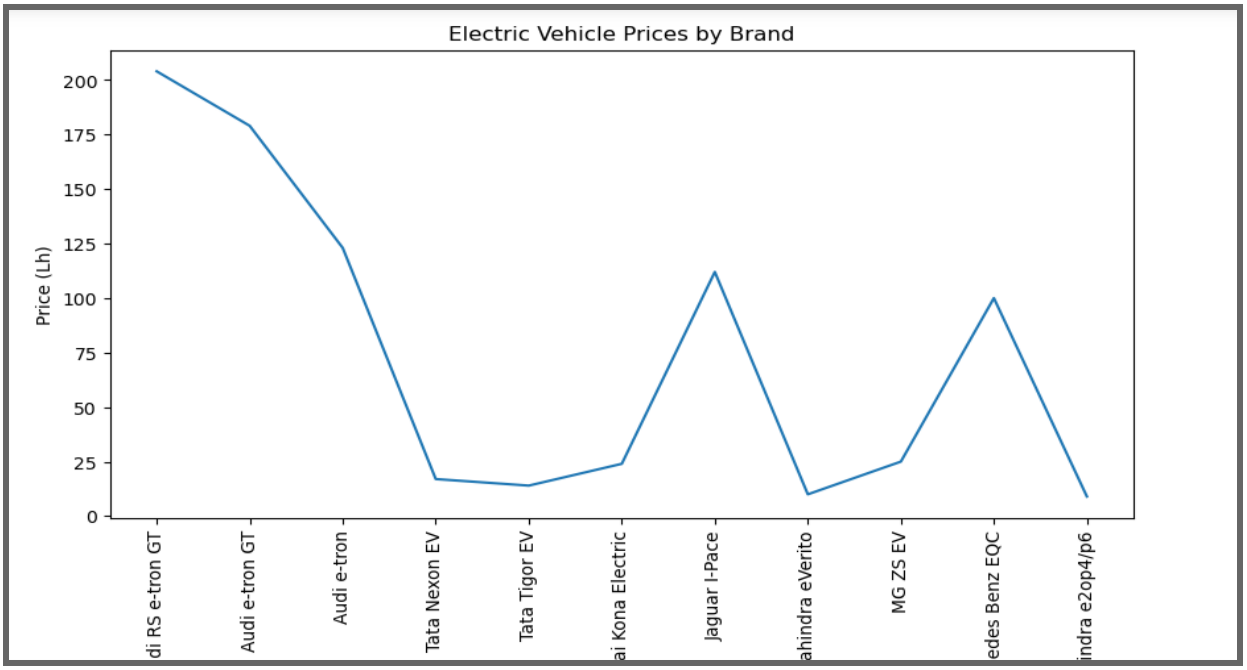


Fig.16

The Prices are very irregular.
Still it’s very clear that the Prices are very less for some brands which are affordable for every person.

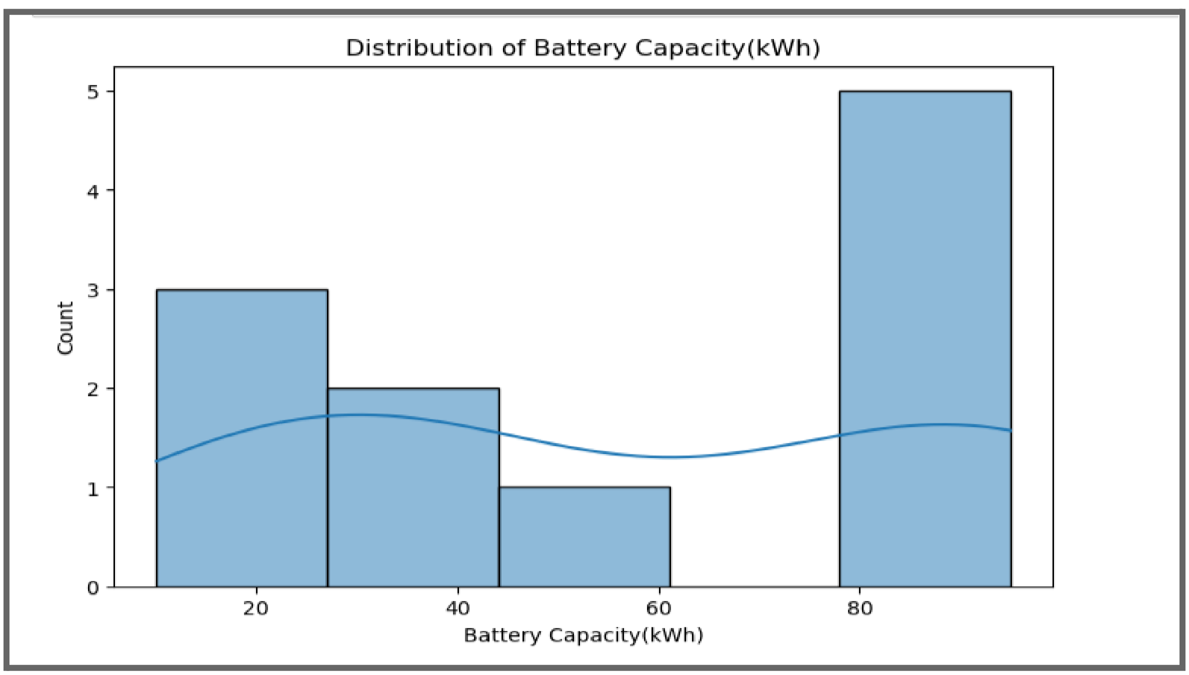


Fig.17

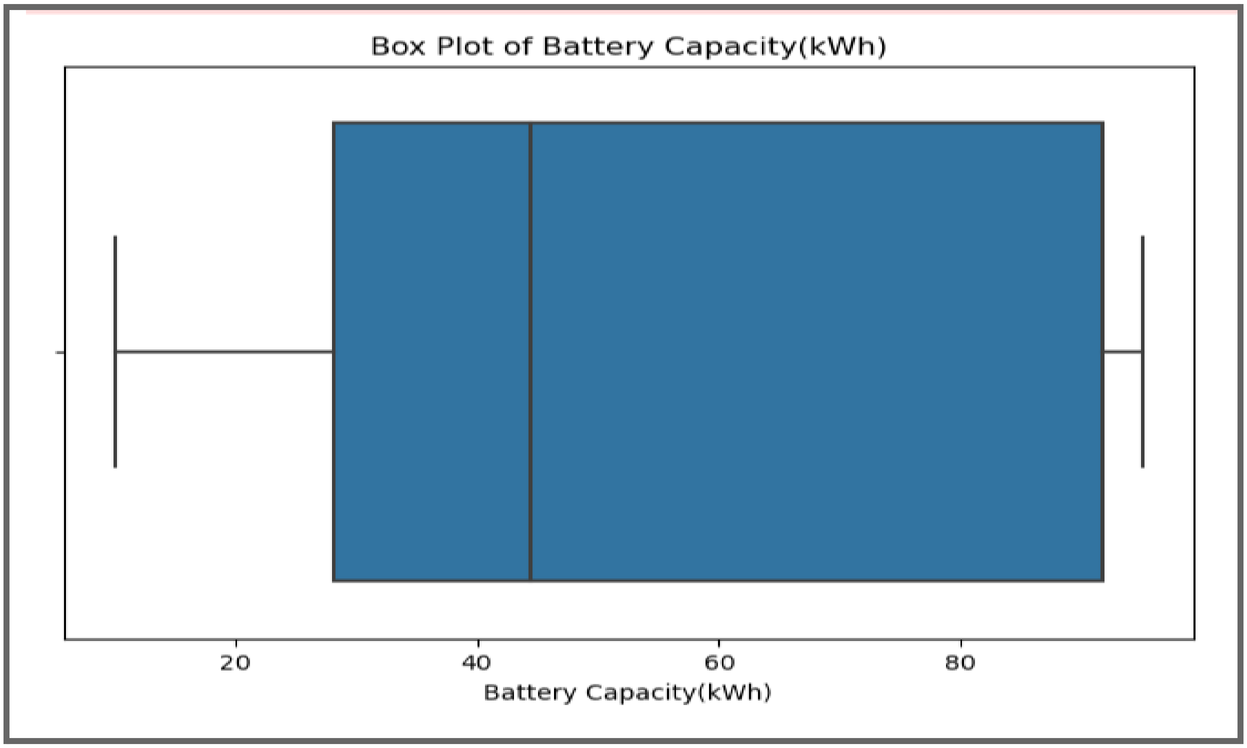

Box Plot of Battery Capacity(kWh)
20
40
60
80
Battery Capacity(kWh)

Fig.18

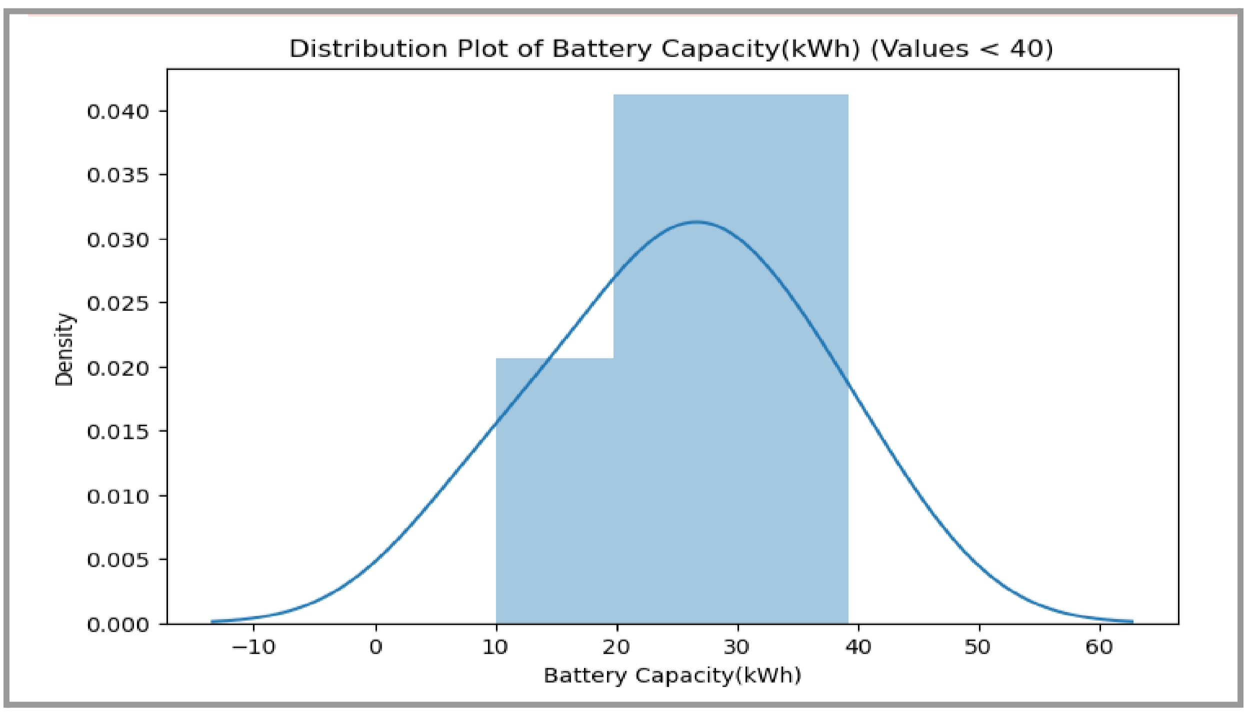

Distribution Plot of Battery Capacity(kWh) (Values < 40)
Density
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
−10
0
10
20
30
40
50
60
Battery Capacity(kWh)

Fig.19

Analysis 2

Importing the required libraries

```python
import numpy as np
import pandas as pd
import seaborn as sns
sns.set(style="darkgrid")
import matplotlib.pyplot as plt
import sklearn
```

Fig.20

Let's further analyze the dataset.

```python
df = pd.read_csv('EV_Data.csv')
```

```python
df.head()
```

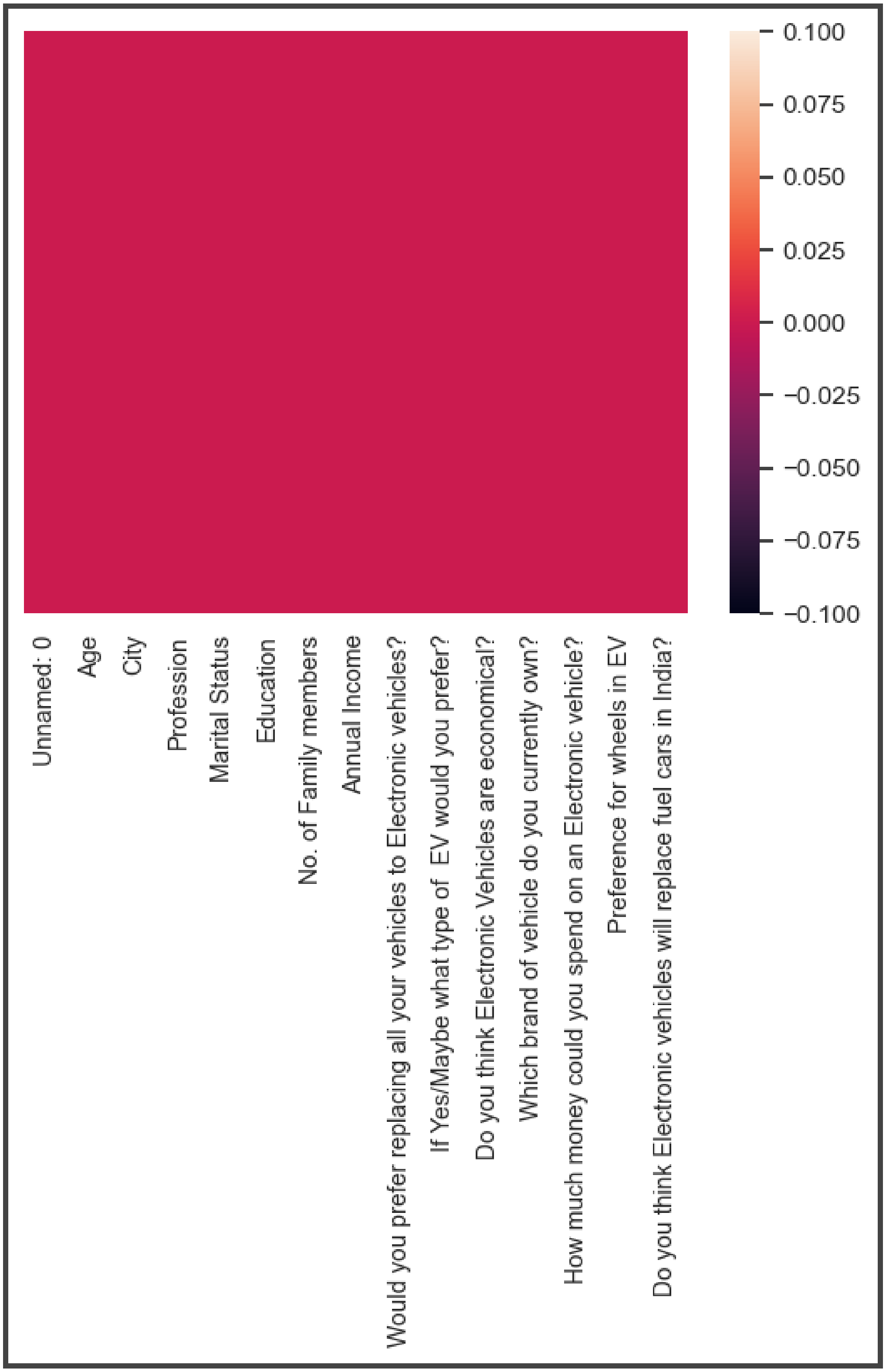
named: 0	Age	City	Profession	Marital Status	Education	No. of Family members	Annual Income	Would you prefer replacing all your vehicles to Electronic vehicles?	If Yes/Maybe what type of EV would you prefer?	Do you think Electronic Vehicles are economical?	Which brand of vehicle do you currently own?	How much money could you spend on an Electronic vehicle?	Preference for wheels in EV	Do you think Electronic vehicles will replace fuel cars in India?
0	30	Nabha	None	Single	Graduate	5	1.193876e+06	Maybe	SUV	Yes	Hyundai	<5 lakhs	2	I don't think so
1	27	Pune	None	Single	Graduate	4	1.844540e+06	Yes	SUV	Yes	Honda	<15 lakhs	4	Yes, in <20years
2	32	Kashipur	None	Single	Graduate	4	2.948150e+06	Yes	Hatchback	Yes	KIA	<15 lakhs	4	Yes, in <20years
3	55	Pune	Business	Single	Graduate	3	2.832380e+06	Maybe	Hatchback	No	Hyundai	<5 lakhs	4	Yes, in <10 years
4	26	Satara	None	Single	Graduate	4	2.638751e+06	Yes	Sedan	Yes	McLaren	<15 lakhs	4	Yes, in <20years

Fig.21

```
df.isnull().sum()
```

```
Unnamed: 0                                                            0
Age                                                                   0
City                                                                  0
Profession                                                            0
Marital Status                                                        0
Education                                                             0
No. of Family members                                                 0
Annual Income                                                         0
Would you prefer replacing all your vehicles to Electronic vehicles?  0
If Yes/Maybe what type of  EV would you prefer?                       0
Do you think Electronic Vehicles are economical?                      0
Which brand of vehicle do you currently own?                          0
How much money could you spend on an Electronic vehicle?              0
Preference for wheels in EV                                           0
Do you think Electronic vehicles will replace fuel cars in India?     0
dtype: int64
```

Fig.22



0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
Unnamed: 0
Age
City
Profession
Marital Status
Education
No. of Family members
Annual Income
Would you prefer replacing all your vehicles to Electronic vehicles?
If Yes/Maybe what type of EV would you prefer?
Do you think Electronic Vehicles are economical?
Which brand of vehicle do you currently own?
How much money could you spend on an Electronic vehicle?
Preference for wheels in EV
Do you think Electronic vehicles will replace fuel cars in India?

Fig.23

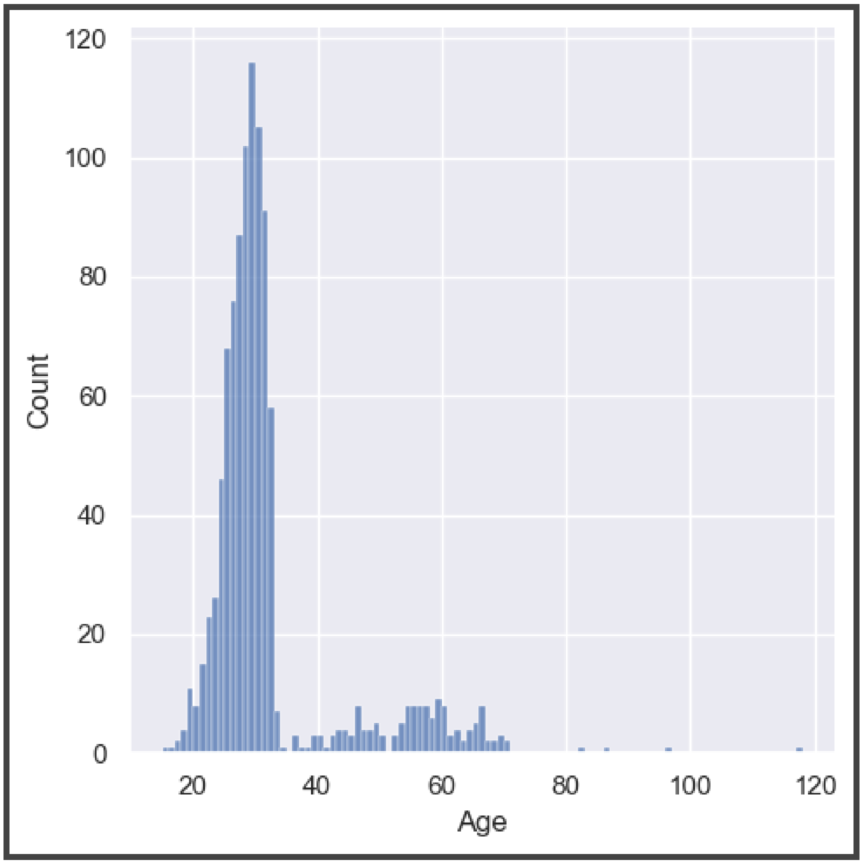


Fig.24

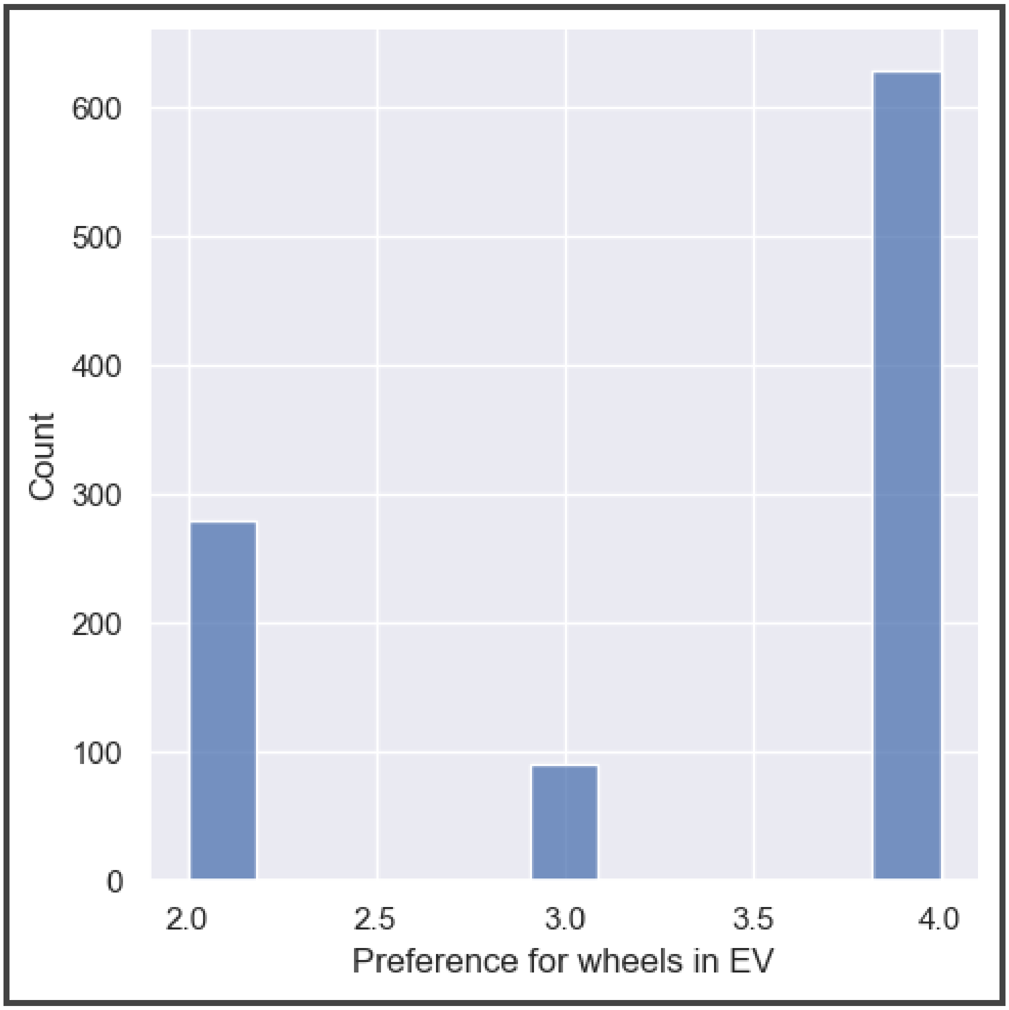

600
500
400
300
200
100
0
Count
2.0
2.5
3.0
3.5
4.0
Preference for wheels in EV

Fig.25

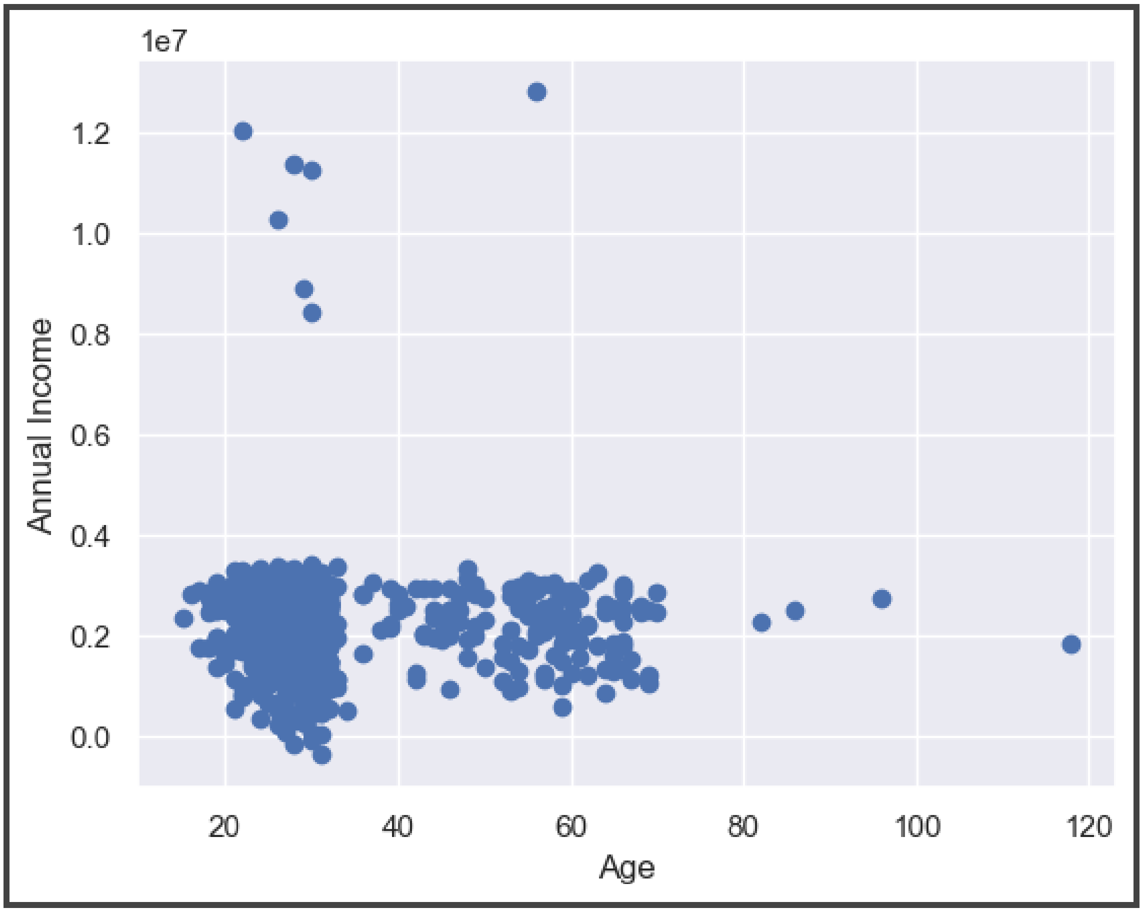

1e7
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Annual Income
20
40
60
80
100
120
Age

Fig.26

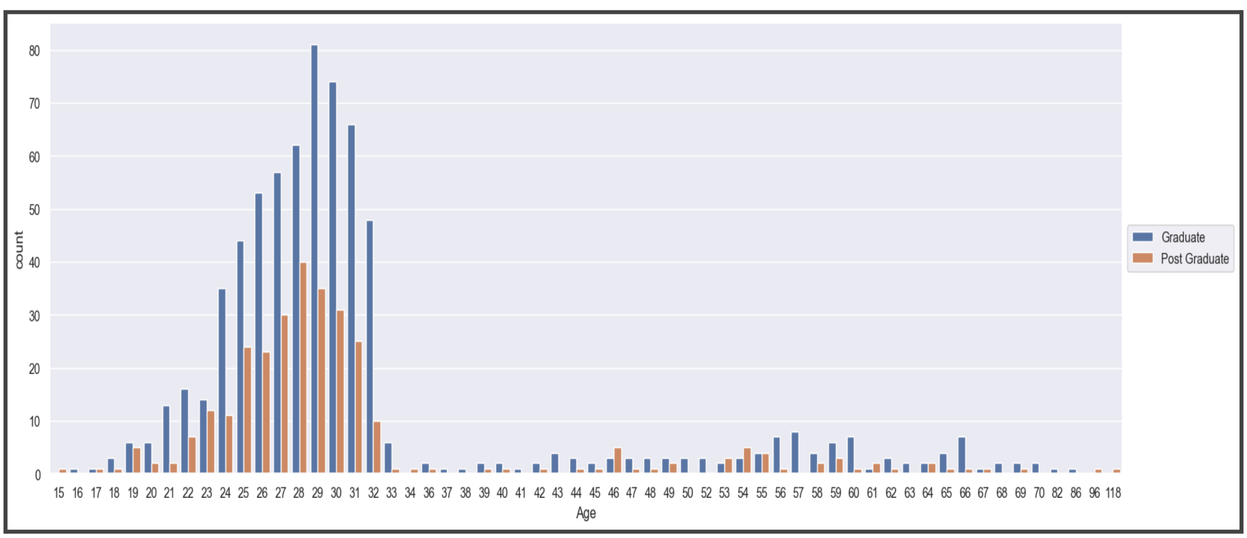

80
70
60
50
40
30
20
10
0
count
15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 82 86 96 118
Age
Graduate
Post Graduate

Fig.27


Which brand of vehicle do you currently own?
BMW
Audi
Nexa
Maruti Suzuki
Maruti
Maruti
Suzuki
Chevrolet
Nissan
Tata
MG
Volkswagen
McLaren
KIA
Honda
Hyundai
None
Business
Working Professional
Salaried
Profession

Fig.28

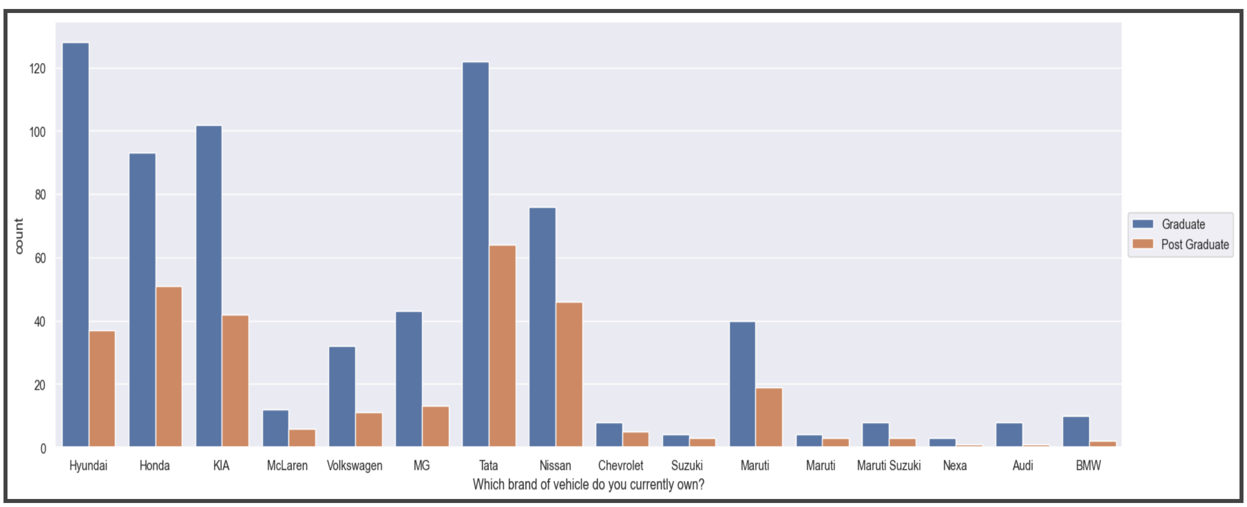

count
0
20
40
60
80
100
120
Graduate
Post Graduate
Hyundai
Honda
KIA
McLaren
Volkswagen
MG
Tata
Nissan
Chevrolet
Suzuki
Maruti
Maruti
Maruti Suzuki
Nexa
Audi
BMW
Which brand of vehicle do you currently own?

Fig.29

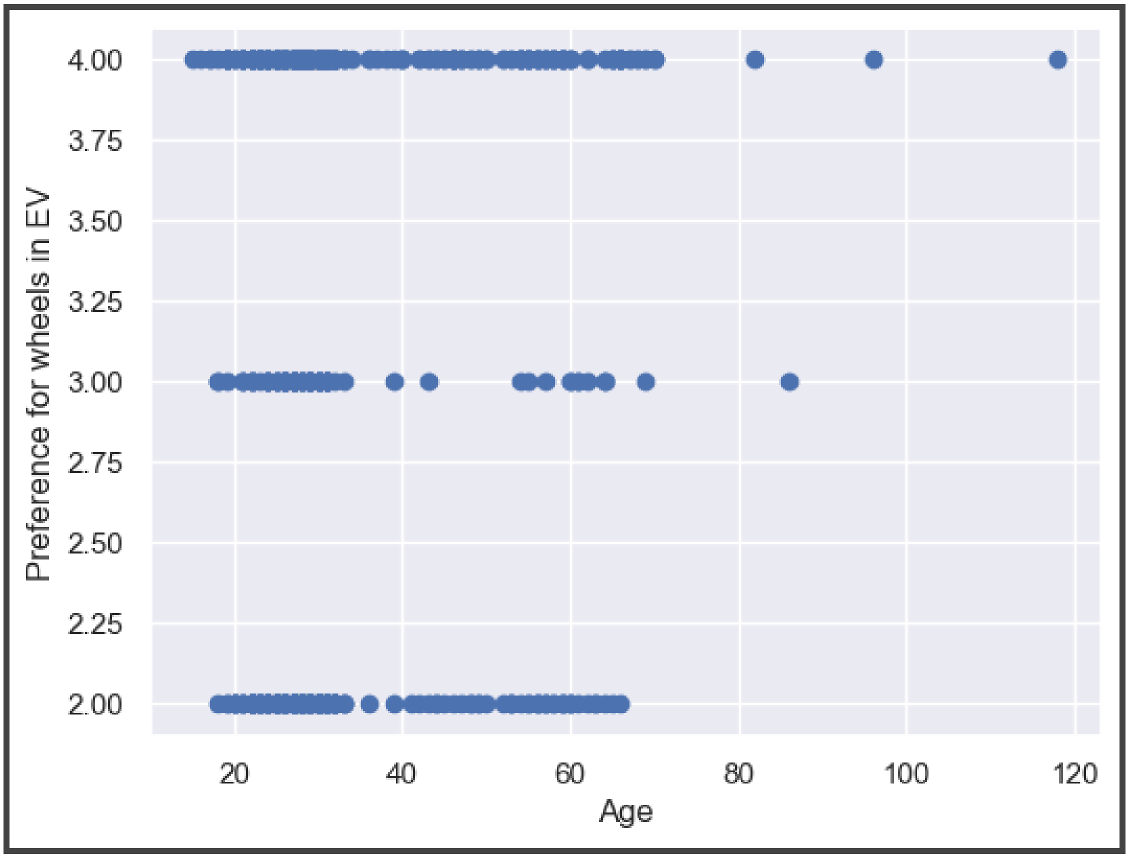

Preference for wheels in EV
4.00
3.75
3.50
3.25
3.00
2.75
2.50
2.25
2.00
20
40
60
80
100
120
Age

Fig.30

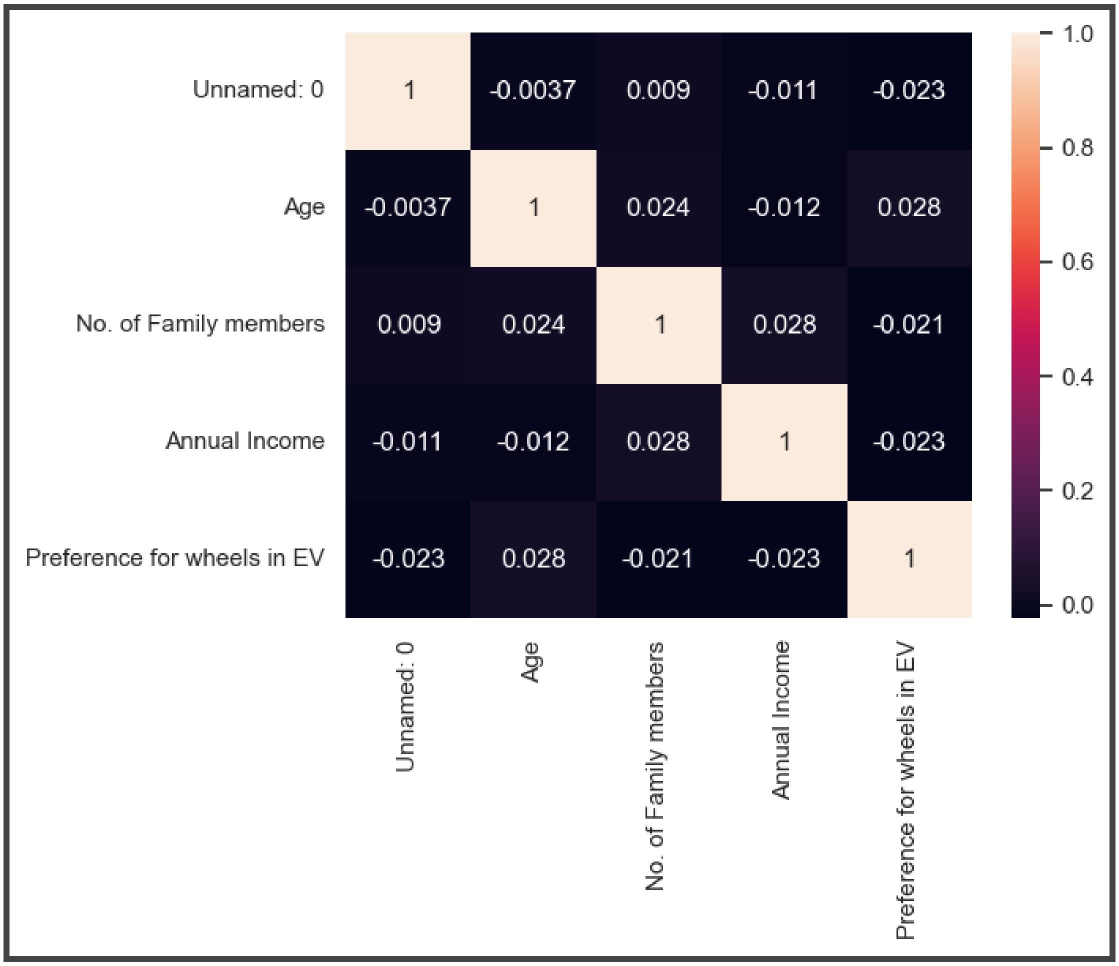
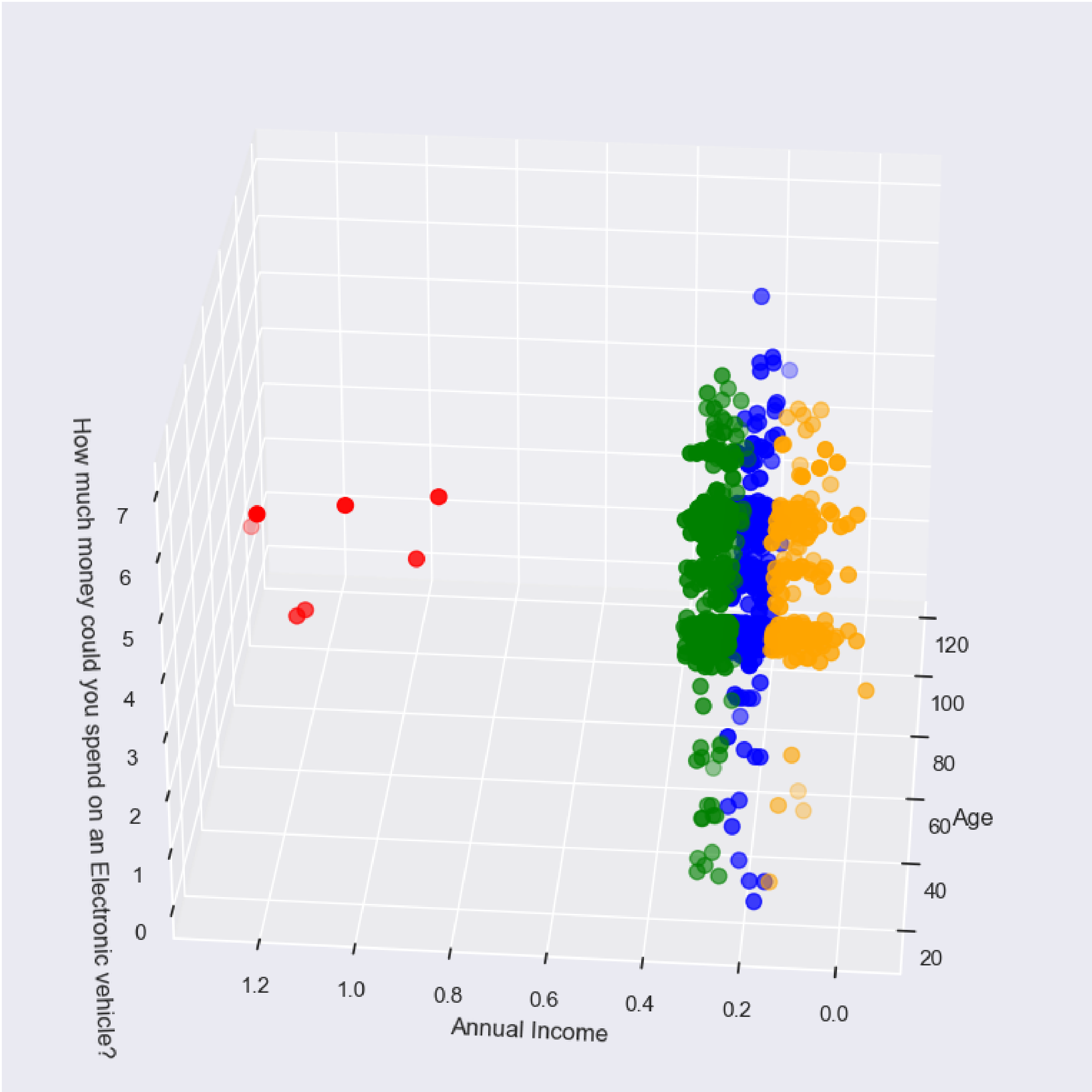


Fig.31

The clusters identified were:-



How much money could you spend on an Electronic vehicle?
7
6
5
4
3
2
1
0
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Annual Income
120
100
80
60
Age
40
20

Analysis-3

Implementation:-

Packages Used:

1. **Numpy**
2. **Pandas**
3. **SKLearn**

Importing Libraries:-

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tqdm import tqdm
import seaborn as sb
import statsmodels.api as sm
import plotly.express as px
from google.colab import files
import kaleido
from sklearn.preprocessing import StandardScaler,PowerTransformer
from sklearn.decomposition import PCA
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import KMeans, MeanShift, estimate_bandwidth
from sklearn.datasets import make_blobs
from yellowbrick.cluster import KElbowVisualizer, SilhouetteVisualizer, InterclusterDistance
from collections import Counter
from sklearn.model_selection import cross_validate,train_test_split
from sklearn.linear_model import LinearRegression,LogisticRegression
from sklearn import metrics
from sklearn.metrics import r2_score,silhouette_score,confusion_matrix,accuracy_score
pd.set_option("display.precision",3)
np.set_printoptions(precision=5, suppress=True)
pd.options.display.float_format = '{:.4f}'.format
import plotly.io as pio

pio.renderers.default = "svg"
```

The data collected is compact and is partly used for visualization purposes and partly for clustering. Python libraries such as NumPy, Pandas, Scikit-Learn, and SciPy are used for the workflow, and the results obtained are ensured to be reproducible.

Data Analysis:-

Step used to visualize the data

1.Imported necessary libraries

2.Load the data into dataframe

3.Understanding the data using datatype,examining a few rows and viewing summary statistics.

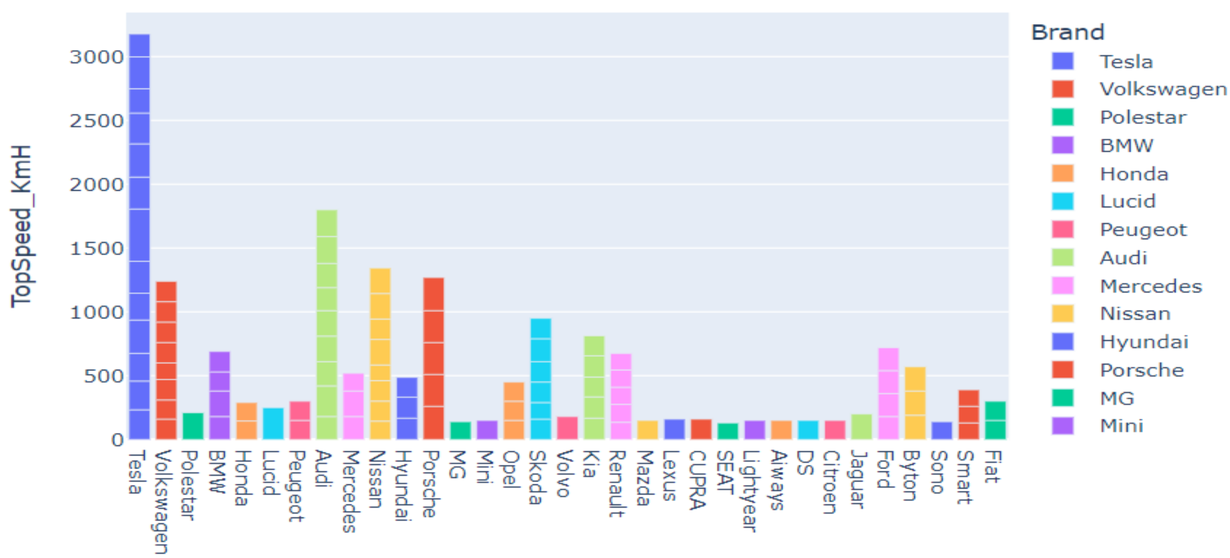
4.Handling missing and null values by dropping rows and imputing them.

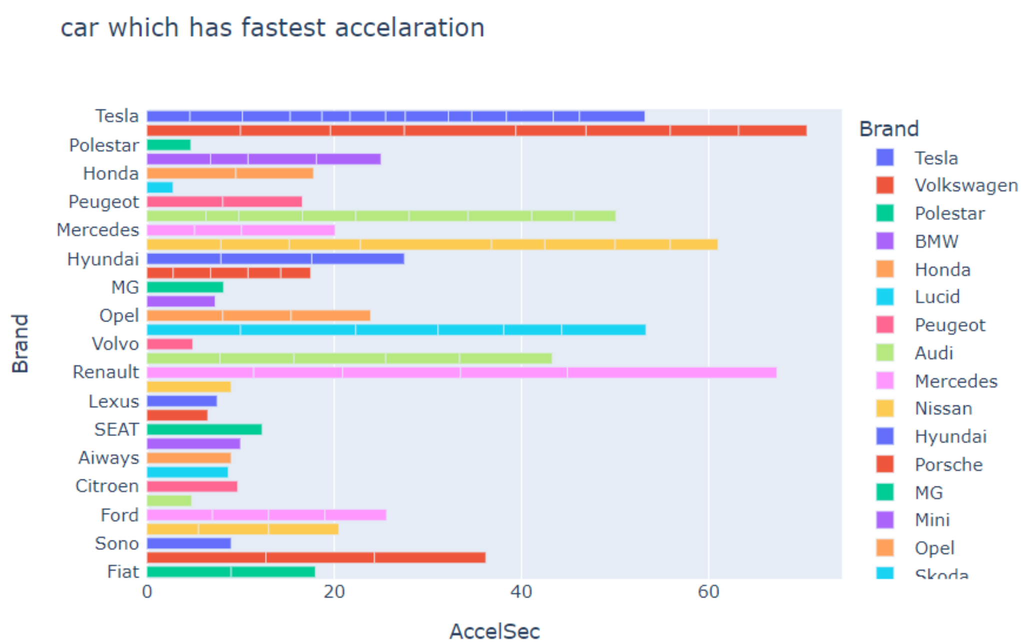
5.Visualize the data using some advance technique like histograms,correlation matrix etc

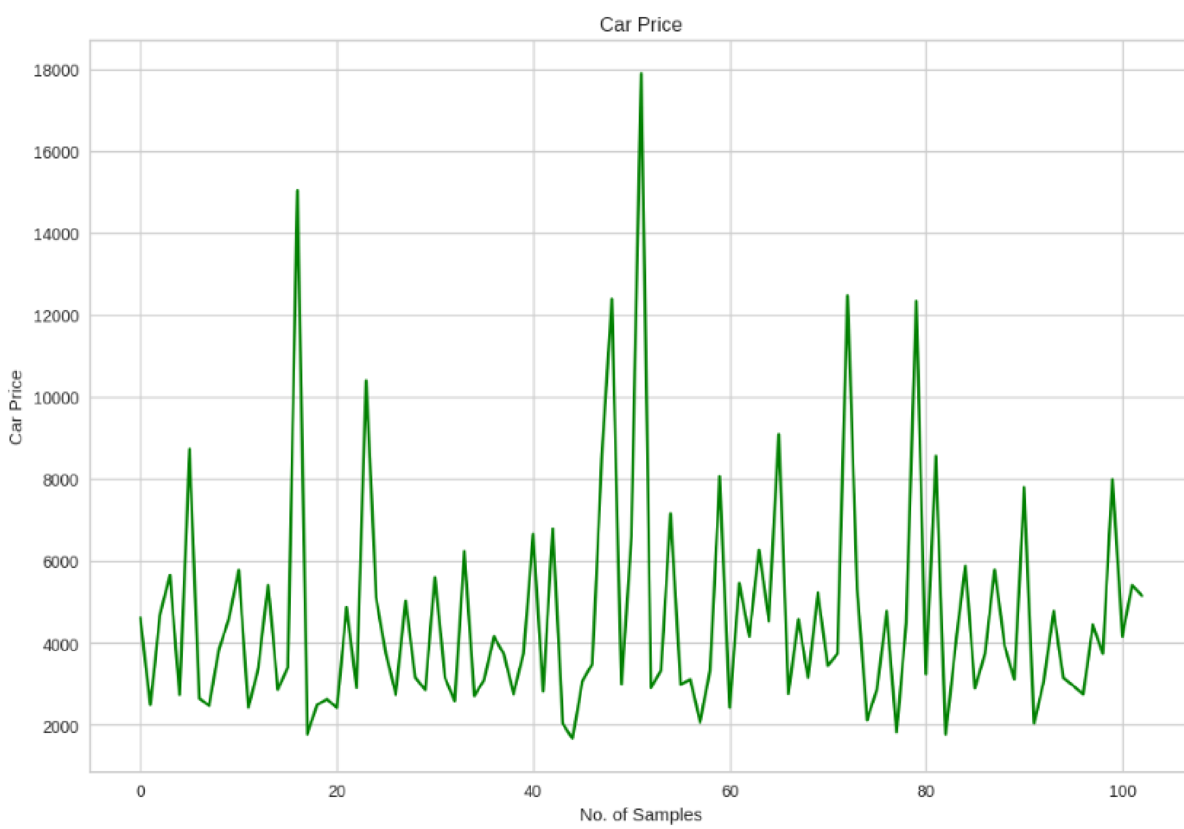
6. Analyze relationships between variables through scatter plots and heatmaps.

To perform data analysis(EDA) we used histograms,scatter plot,box plot,heatmaps , Correlation matrix etc.

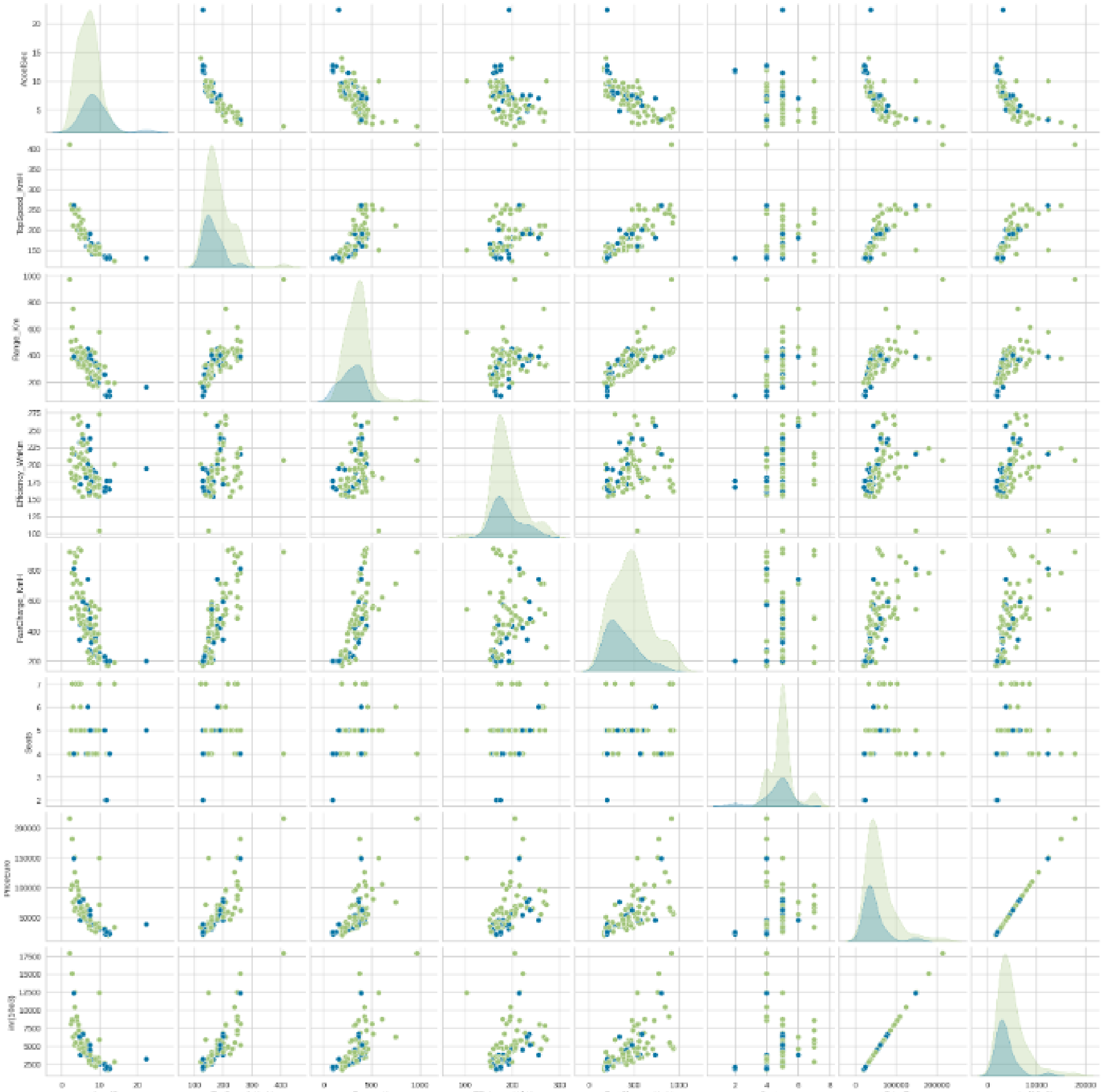
Car which has a top speed is visualized below-




car which has fastest accelaration
Brand
Tesla
Polestar
Honda
Peugeot
Mercedes
Hyundai
MG
Opel
Volvo
Renault
Lexus
SEAT
Aiways
Citroen
Ford
Sono
Fiat
0
20
40
60
AccelSec
Brand
Tesla
Volkswagen
Polestar
BMW
Honda
Lucid
Peugeot
Audi
Mercedes
Nissan
Hyundai
Porsche
MG
Mini
Opel

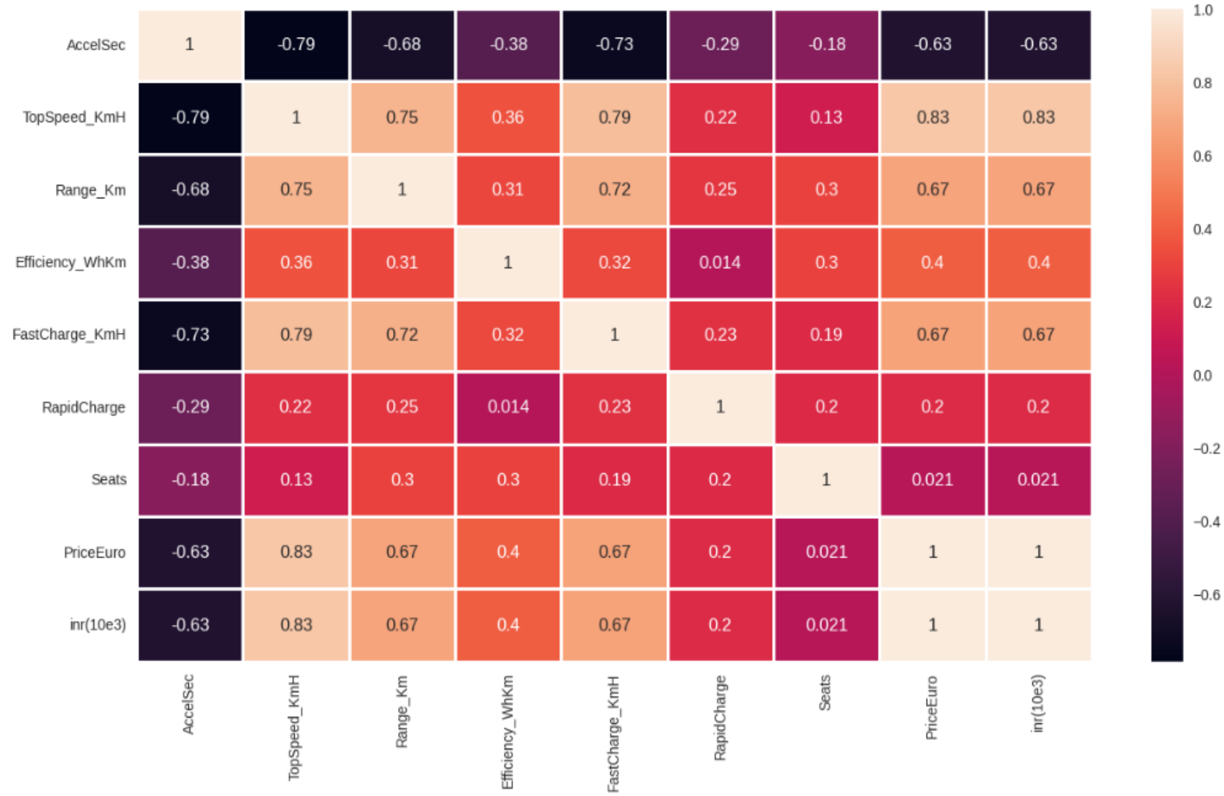

Car Price
18000
16000
14000
12000
10000
8000
6000
4000
2000
Car Price
0
20
40
60
80
100
No. of Samples

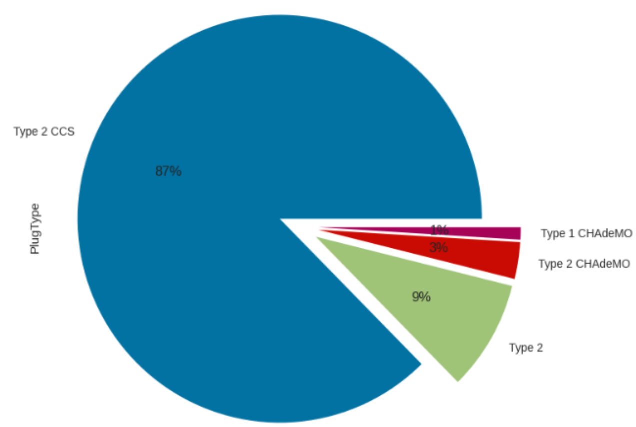
Pairplot of all the columns based on Rapid Charger presence:-



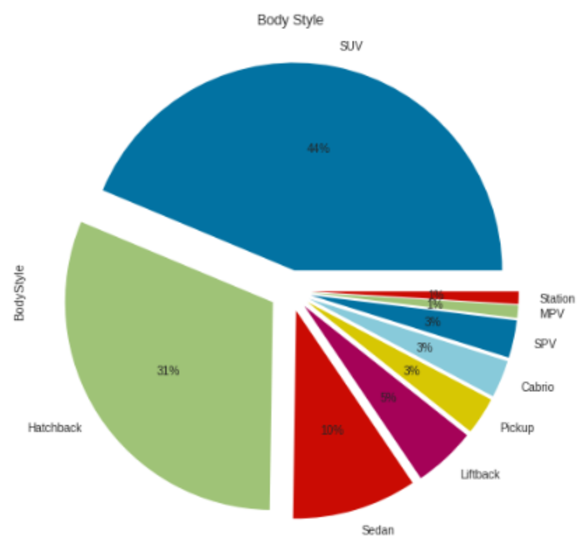
Correlation matrix:-

A correlation matrix is simply a table that displays the correlation. It is best used in variables that demonstrate a linear relationship between each other. Coefficients for different variables. The matrix depicts the correlation between all the possible pairs of values through the heatmap in the below figure. The relationship between two variables is usually considered strong when their correlation coefficient value is larger than 0.7.



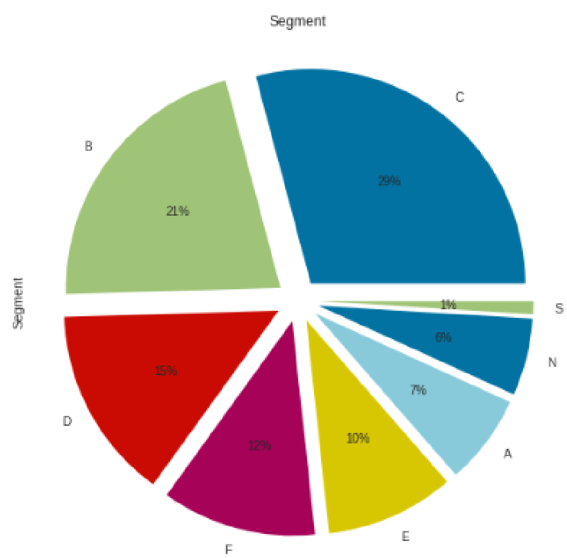


Type of Plug used for charging



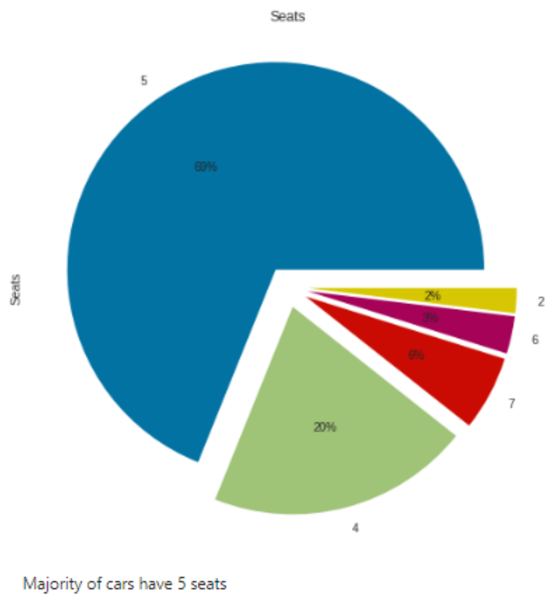
Most cars are eiher SUV or Hatchback

Cars and their body style



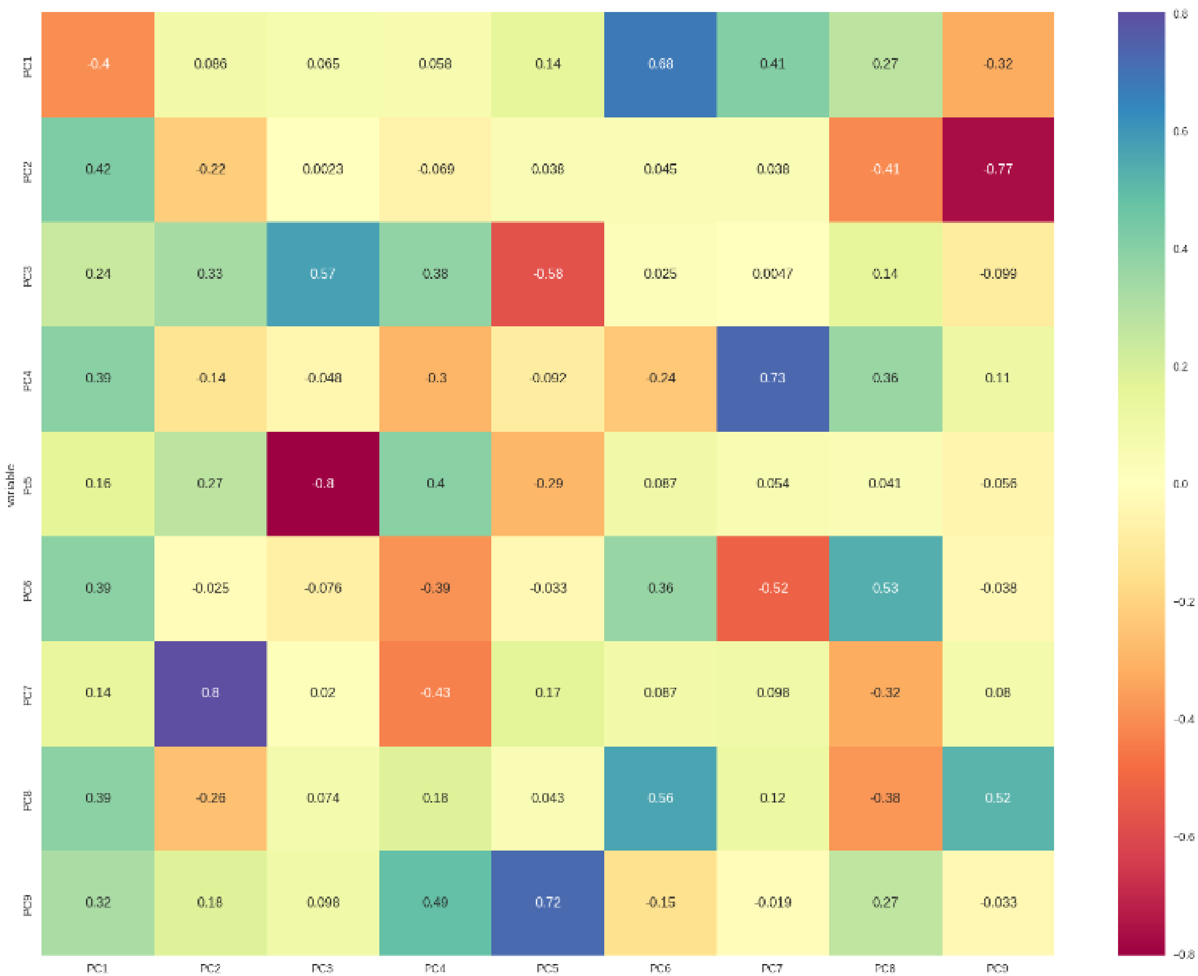
Most cars are either C or B type

Segment in which the cars fall under



Number of Seats

Now we can see that the requirements of what type of cars are most needed for customers and from the past 10 years there is a rapid growth of Electric vehicles usage in India.

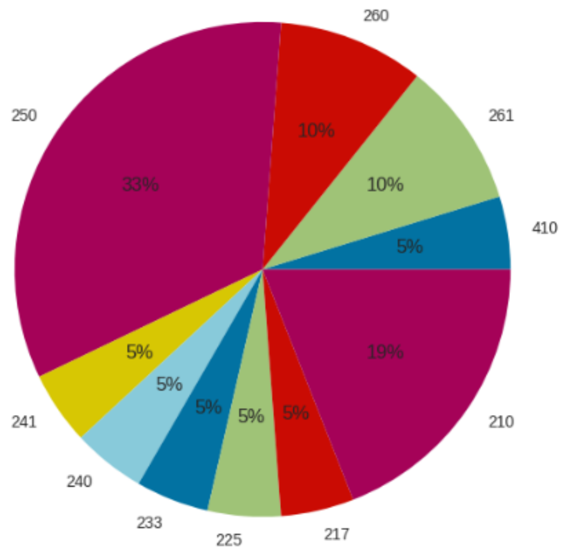


Correlation matrix plot for loading

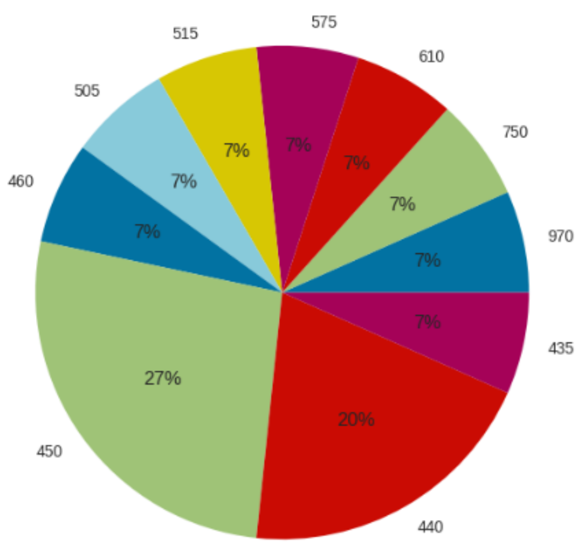
Profiling and Describing the Segments:-

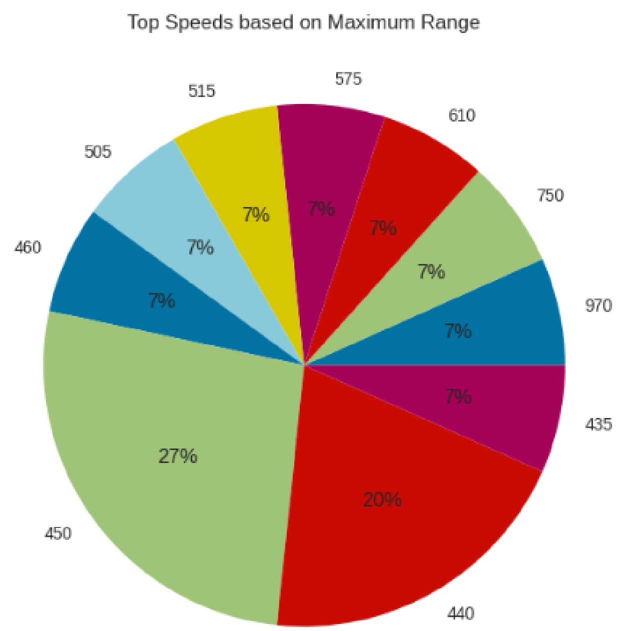
Sorting the Top Speeds and Maximum Range in accordance to the Price with head () we can view the Pie Chart.

Cost based on top speed



Cost based on Maximum Range

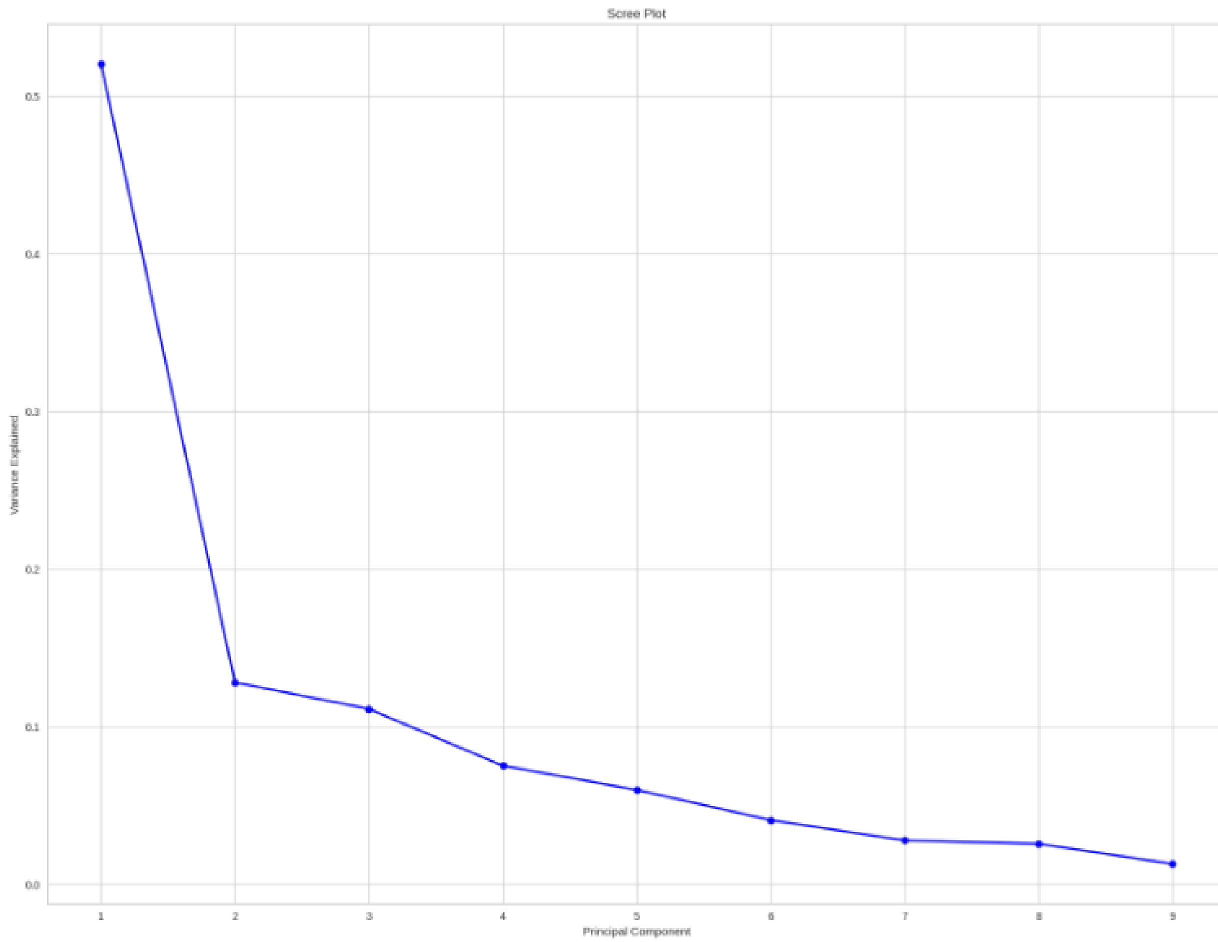




Scree Plot:-

It is a common method for determining the number of PCs to be retained via graphical representation. It is a simple line segment plot that shows the eigenvalues for each individual PC. It shows the eigenvalues on the y-axis and the number of factors on the x-axis. It always displays a downward curve. Most scree plots look broadly similar in shape, starting high on the left, falling rather quickly, and then flattening out at some point. This is because the first component usually explains much of the variability, the next few components explain a moderate amount, and the latter components only explain a small fraction of the overall variability. The scree plot criterion looks for the “elbow” in the curve and selects all components just before the line flattens out. The proportion of variance plot: The selected PCs should be able to describe at least 80% of the variance.

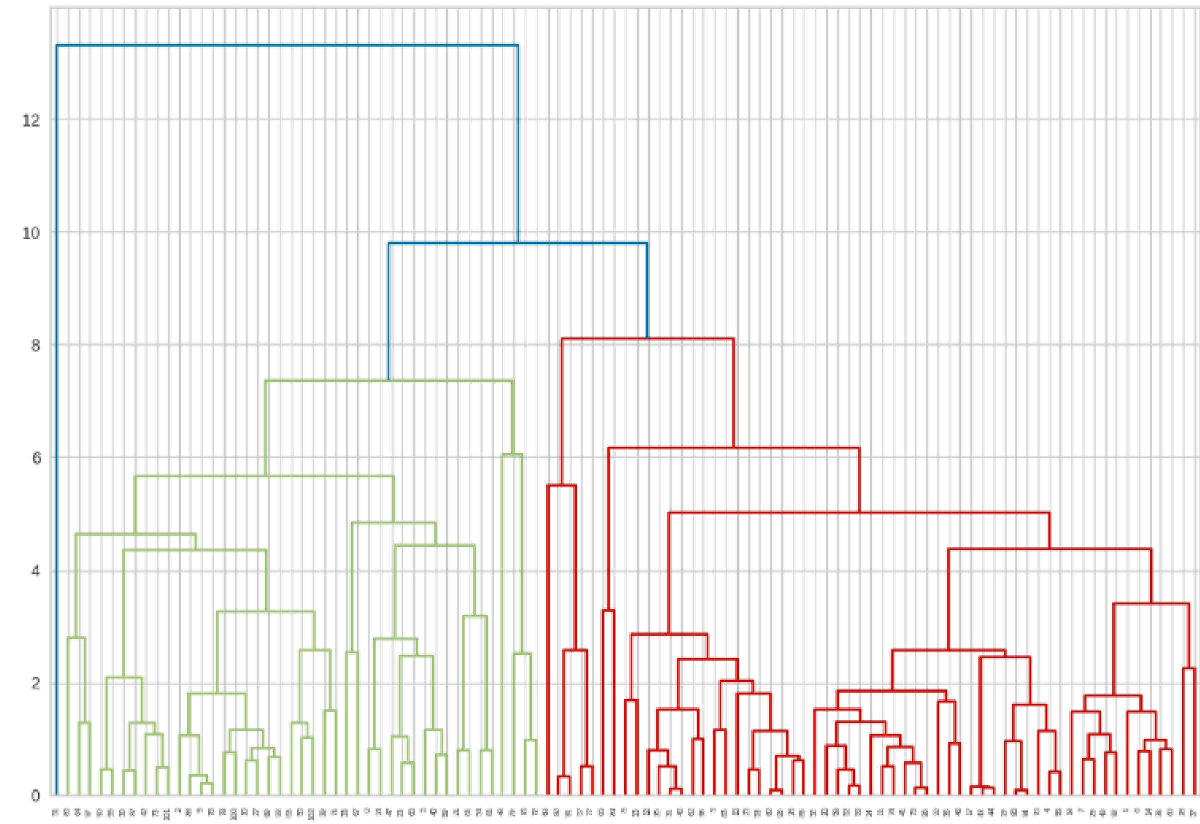
```python
PC_values = np.arange(pca.n_components_) + 1
plt.plot(PC_values, pca.explained_variance_ratio_, 'o-', linewidth=2, color='blue')
plt.title('Scree Plot')
plt.xlabel('Principal Component')
plt.ylabel('Variance Explained')
plt.show()
```



Dendrogram:

This technique is specific to the agglomerative hierarchical method of clustering. The agglomerative hierarchical method of clustering starts by considering each point as a separate cluster and starts joining points to clusters in a hierarchical fashion based on their distances. To get the optimal number of clusters for hierarchical clustering, we make use of a dendrogram which is a tree-like chart that shows the sequences of merges or splits of clusters. If two clusters are merged, the dendrogram will join them in a graph and the height of the join will be the distance between those clusters. As shown in Figure, we can choose the optimal number of clusters based on the hierarchical structure of the dendrogram. As highlighted by other cluster validation metrics, four to five clusters can be considered for the agglomerative hierarchical as well.

```
linked = linkage(data2, 'complete')
plt.figure(figsize=(13, 9))
dendrogram(linked, orientation='top')
plt.show()
```



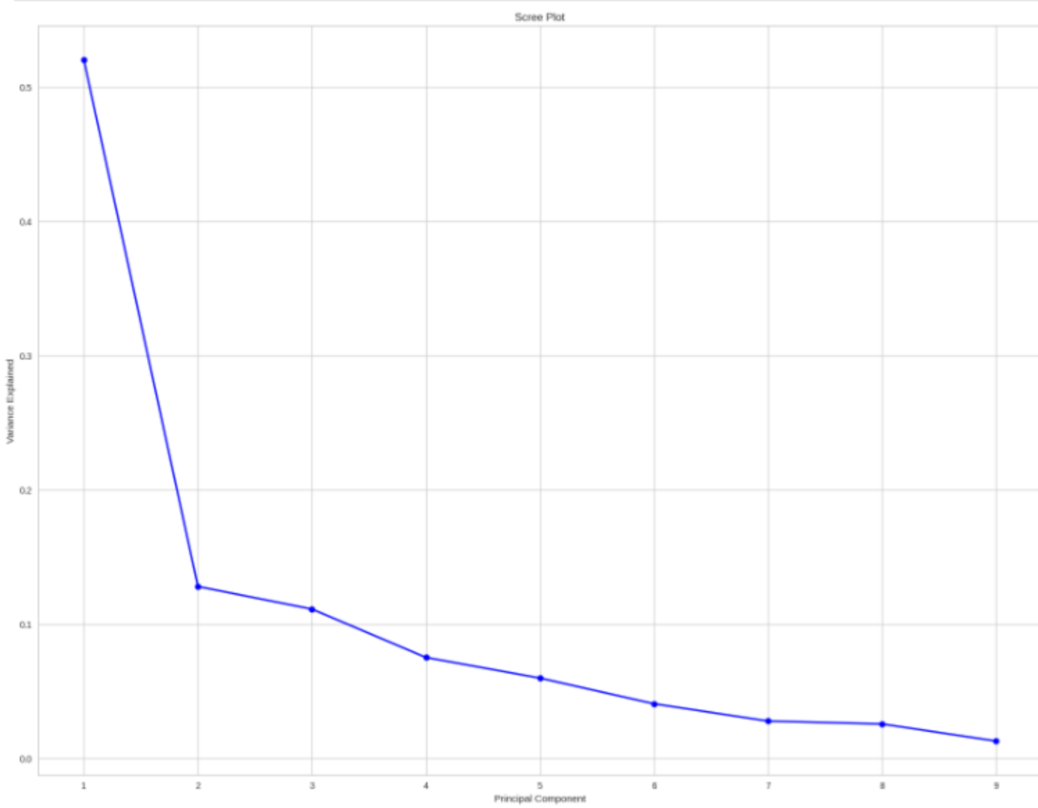
Dendrogram plot for our dataset

Elbow Method:-

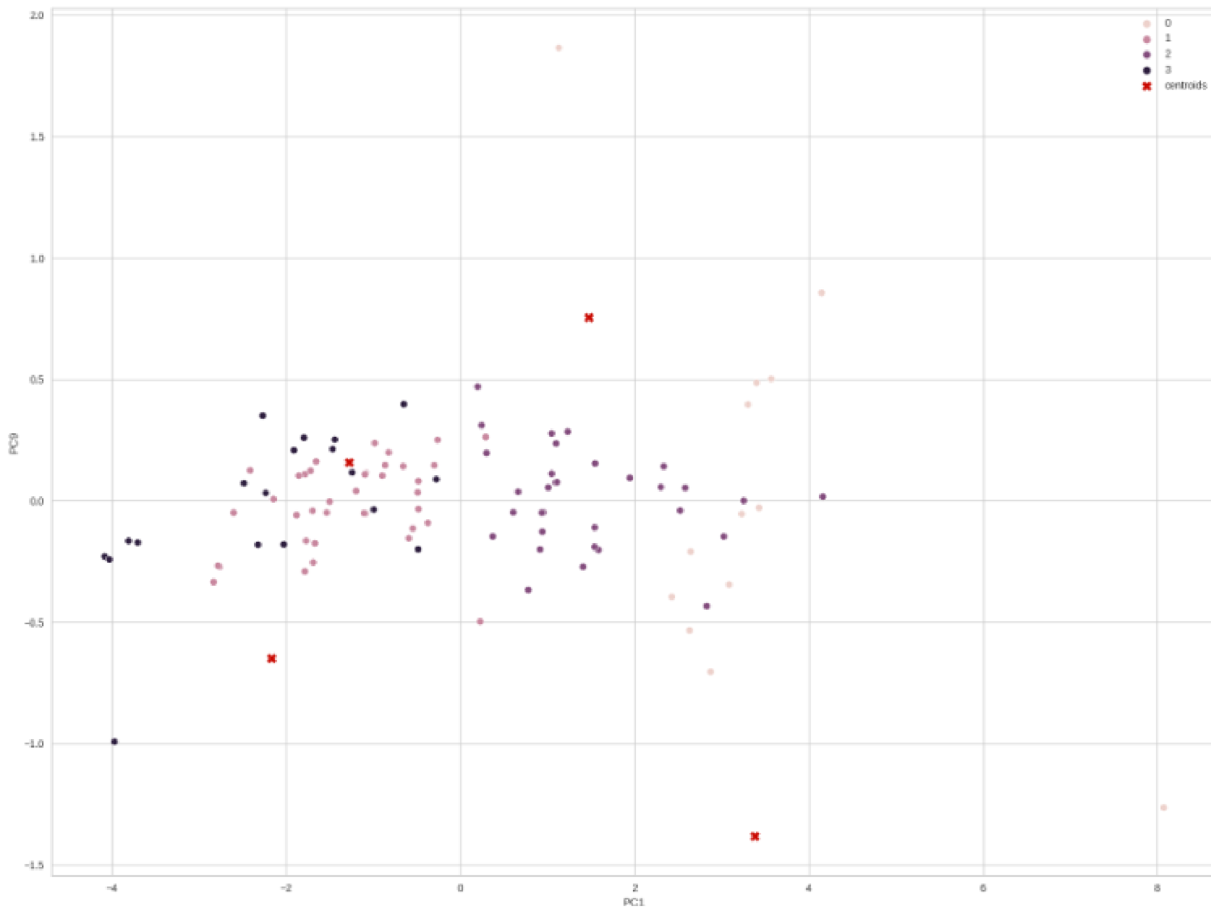
The Elbow method is a popular method for determining the optimal number of clusters. The method is based on calculating the

Within-Cluster-Sum of Squared Errors (WSS) for a different number of clusters (k) and selecting the k for which change in WSS first starts to diminish. The idea behind the elbow method is that the explained variation changes rapidly for a small number of clusters and then it slows down leading to an elbow formation in the curve. The elbow point is the number of clusters we can use for our clustering algorithm. The KElbowVisualizer function fits the KMeans model for a range of clusters values between 2 to 8. As shown in Figure, the elbow point is achieved which is highlighted by the function itself. The function also informs us about how much time was needed to plot models for various numbers of clusters through the green line.

```
PC_values = np.arange(pca.n_components_) + 1
plt.plot(PC_values, pca.explained_variance_ratio_, 'o-', linewidth=2, color='blue')
plt.title('Scree Plot')
plt.xlabel('Principal Component')
plt.ylabel('Variance Explained')
plt.show()
```

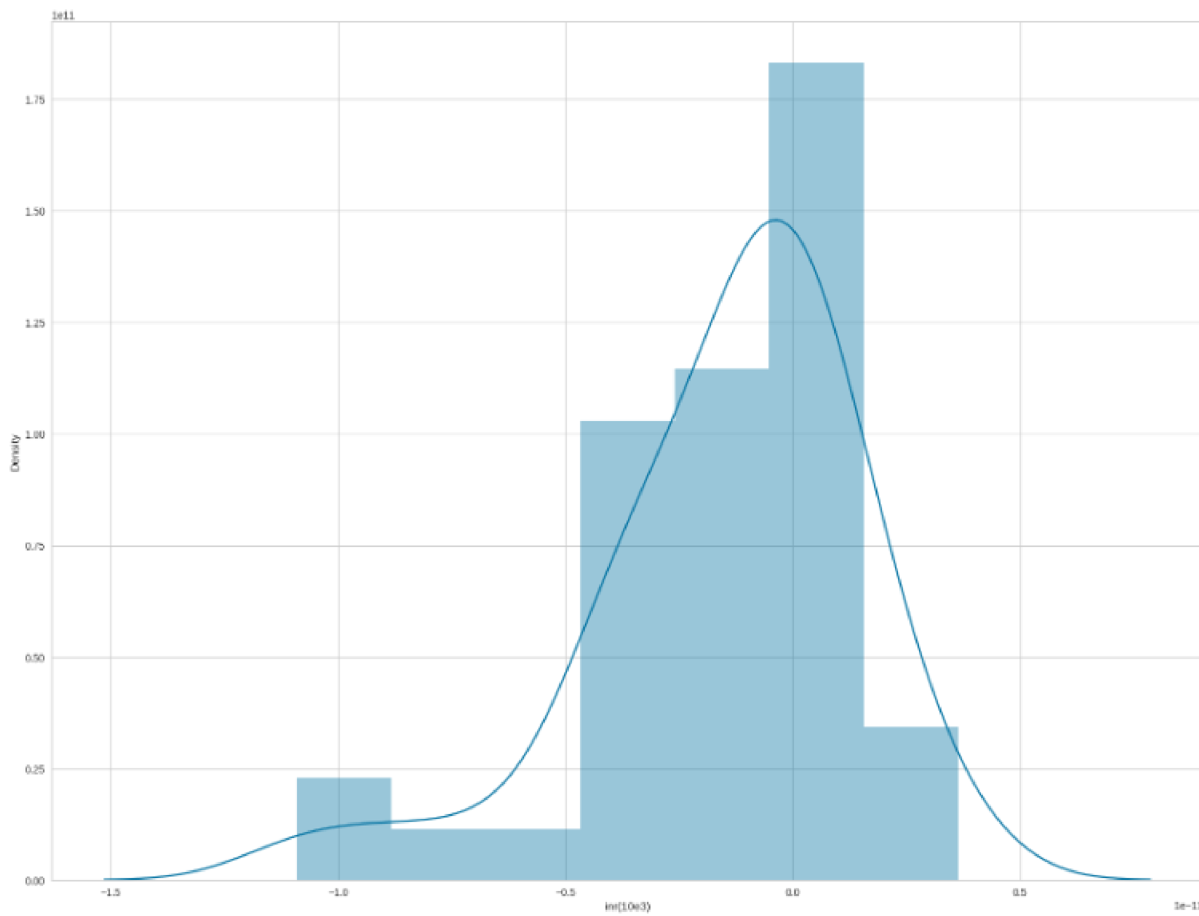


```
sb.scatterplot(data=data2, x="PC1", y="PC9", hue=kmeans.labels_)
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1],
            marker="X", c="r", s=80, label="centroids")
plt.legend()
plt.show()
```



LinearRegression().fit(Xtrain,ytrain) command is used to fit the data set into the model. The values of intercept, coefficient, and cumulative distribution function (CDF) are described in the figure.

After completion of training the model process, we test the remaining 60% of data on the model. The obtained results are checked using a scatter plot between predicted values and the original test data set for the dependent variable and acquired similar to a straight line as shown in the figure and the density function is also normally distributed

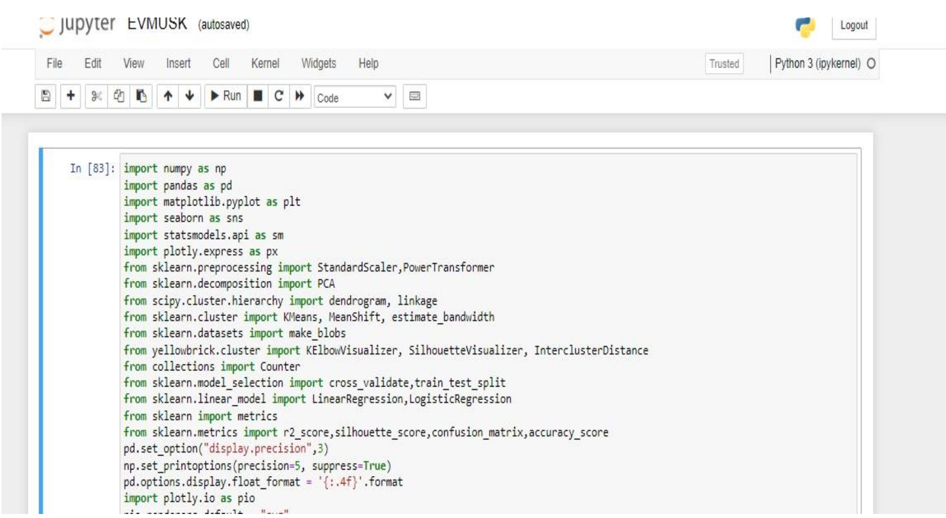


The metrics of the algorithm, Mean absolute error, Mean squared error and mean square root error are described in the below figure:-

```
In [80]: print('MAE:',metrics.mean_absolute_error(y_test,predictions))
         print('MSE:',metrics.mean_squared_error(y_test,predictions))
         print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test,predictions)))

MAE: 2.2629094365540715e-12
MSE: 1.0768119045556378e-23
RMSE: 3.281481227366138e-12

In [81]: metrics.mean_squared_error(y_test,predictions)

Out[81]: 1.0768119045556378e-23

In [82]: metrics.mean_absolute_error(y_test,predictions)

Out[82]: 2.2629094365540715e-12

In [83]: np.sqrt(metrics.mean_squared_error(y_test,predictions))

Out[83]: 3.281481227366138e-12
```

ANALYSIS - 4

Import required libraries.




```
In [83]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
import plotly.express as px
from sklearn.preprocessing import StandardScaler,PowerTransformer
from sklearn.decomposition import PCA
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import KMeans, MeanShift, estimate_bandwidth
from sklearn.datasets import make_blobs
from yellowbrick.cluster import KElbowVisualizer, SilhouetteVisualizer, InterclusterDistance
from collections import Counter
from sklearn.model_selection import cross_validate,train_test_split
from sklearn.linear_model import LinearRegression,LogisticRegression
from sklearn import metrics
from sklearn.metrics import r2_score,silhouette_score,confusion_matrix,accuracy_score
pd.set_option("display.precision",3)
np.set_printoptions(precision=5, suppress=True)
pd.options.display.float_format = '{:.4f}'.format
import plotly.io as pio
```

THEN LOAD THE DATA SET:

jupyter EVMUSK (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

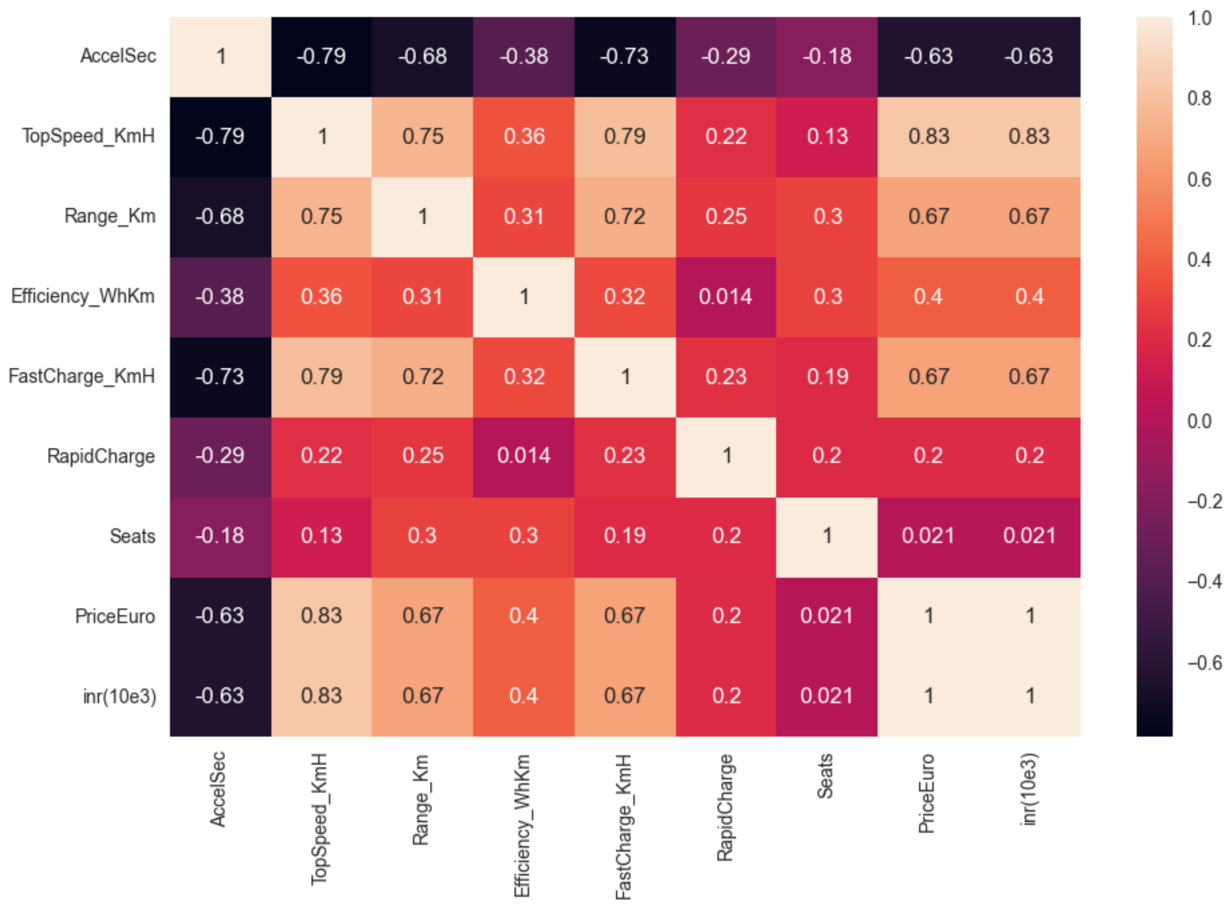
Trusted | Python 3 (ipykernel)

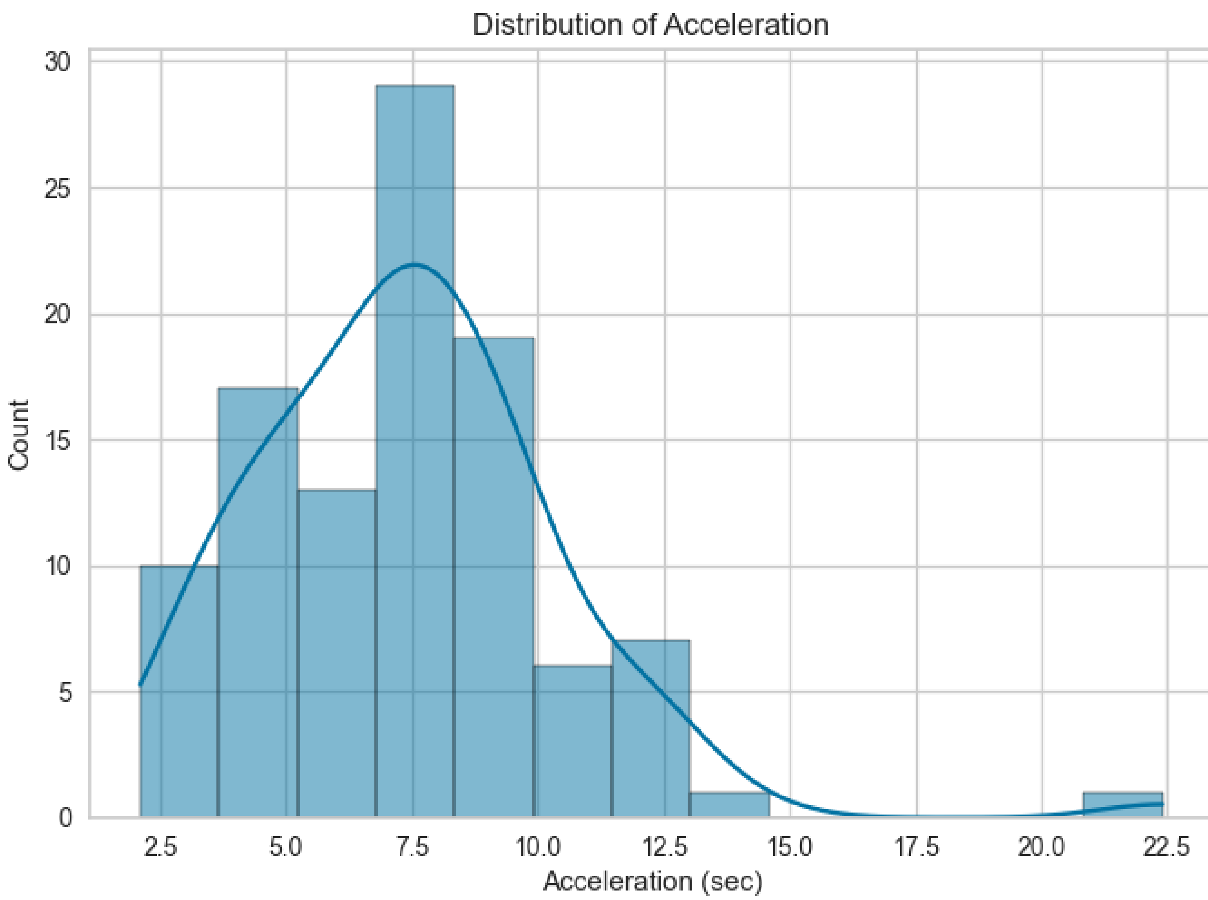
Run Code

In [7]:
```python
df = pd.read_csv("data.csv")
df.drop('Unnamed: 0', axis=1, inplace=True)
df['inr(10e3)'] = df['PriceEuro']*0.08320
df['RapidCharge'].replace(to_replace=['No','Yes'],value=[0, 1],inplace=True)
df
```

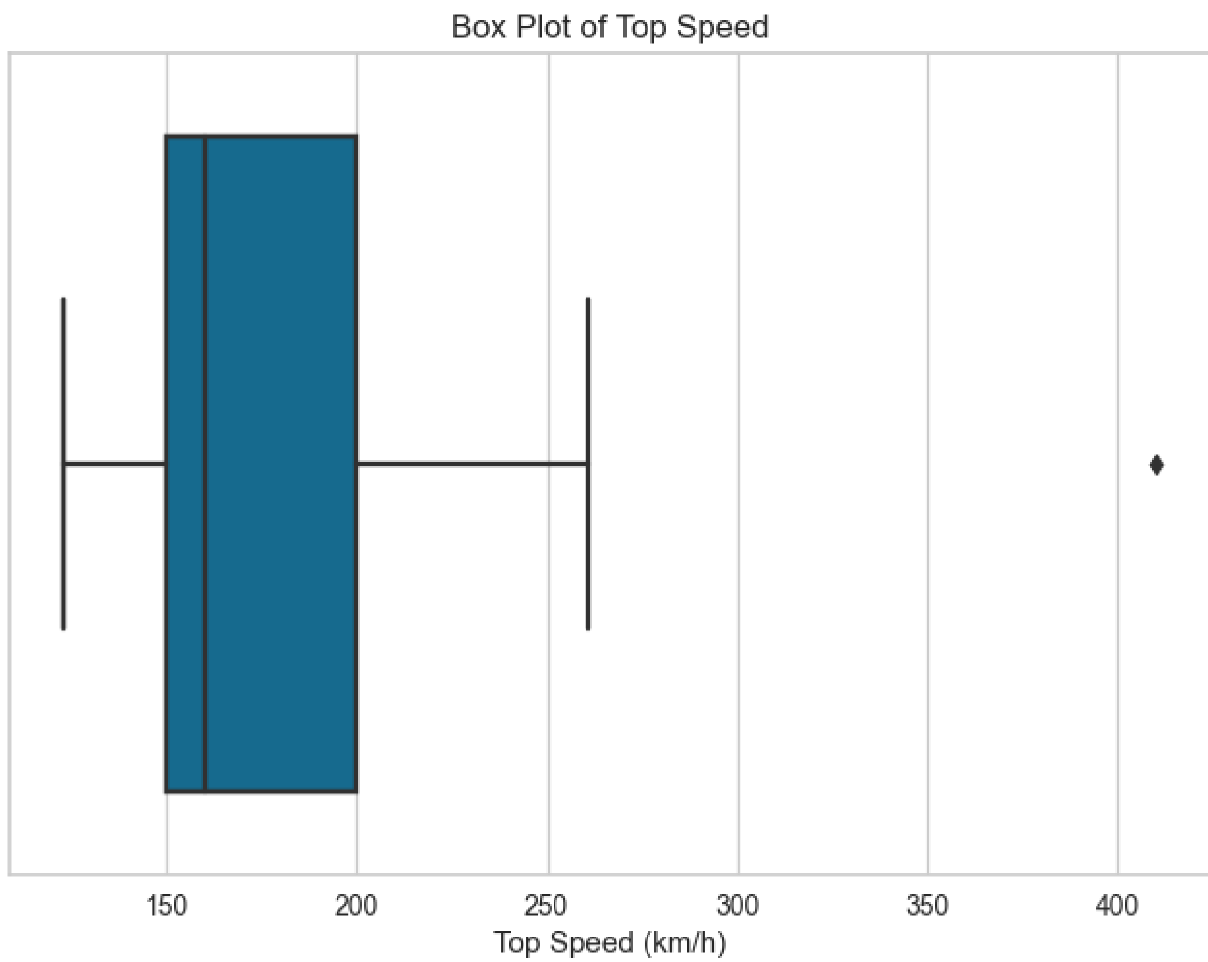
Out[7]:

	Brand	Model	AccelSec	TopSpeed_KmH	Range_Km	Efficiency_WhKm	FastCharge_KmH	RapidCharge	PowerTrain	PlugType	BodyStyle	Segr
0	Tesla	Model 3 Long Range Dual Motor	4.6000	233	450	161	940	1	AWD	Type 2 CCS	Sedan	
1	Volkswagen	ID.3 Pure	10.0000	160	270	167	250	0	RWD	Type 2 CCS	Hatchback	
2	Polestar	2	4.7000	210	400	181	620	1	AWD	Type 2 CCS	Liftback	
3	BMW	iX3	6.8000	180	360	206	560	1	RWD	Type 2 CCS	SUV	
4	Honda	e	9.5000	145	170	168	190	1	RWD	Type 2 CCS	Hatchback	
...	...	...	...	...	...	...	...	...	...	...	...	
98	Nissan	Ariya 63kWh	7.5000	160	330	191	440	1	FWD	Type 2 CCS	Hatchback	
99	Audi	e-tron S Sportback 55 quattro	4.5000	210	335	258	540	1	AWD	Type 2 CCS	SUV	
100	Nissan	Ariya e-4ORCE 63kWh	5.9000	200	325	194	440	1	AWD	Type 2 CCS	Hatchback	
101	Nissan	Ariya e-4ORCE 87kWh Performance	5.1000	200	375	232	450	1	AWD	Type 2 CCS	Hatchback	



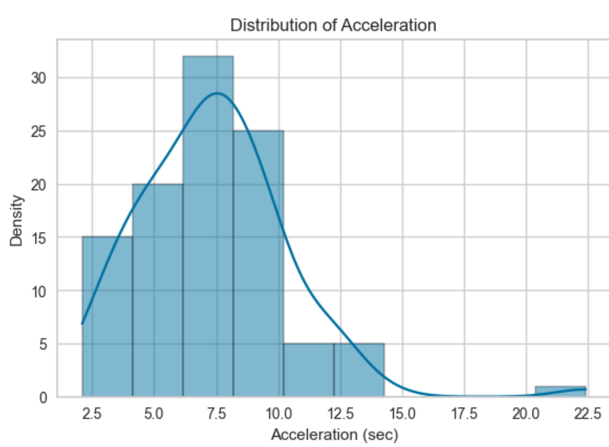


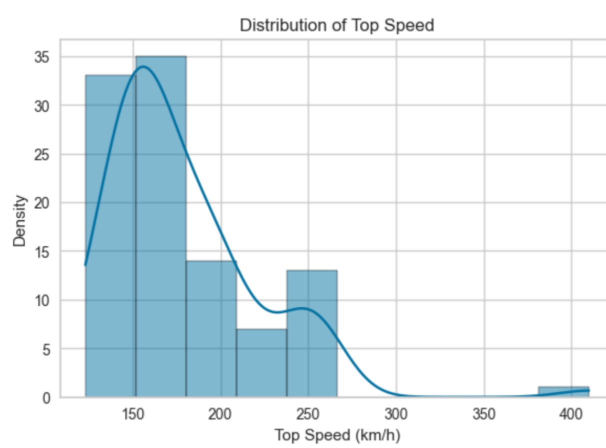
Through the histplot we can see the visualization among count and acceleration.



This box plot shows that top speed has a outlier as more then 400 km speed .

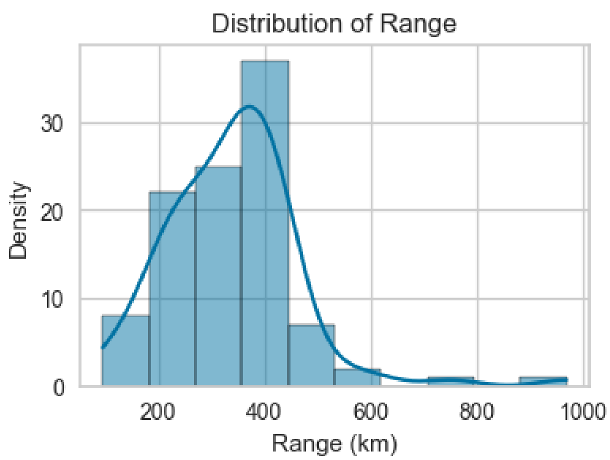
RANGE VS EFFICIENCY BY PT

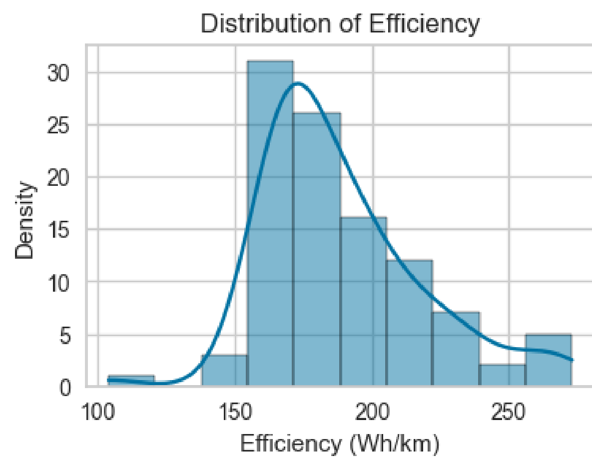




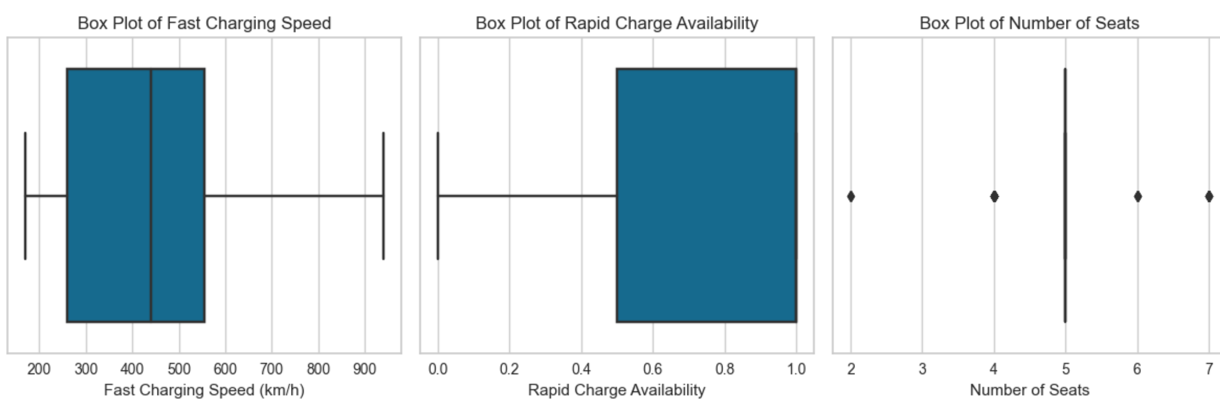
DISTRIBUTION OF ACCELERATION AND

DISTRIBUTION OF TOP SPEED





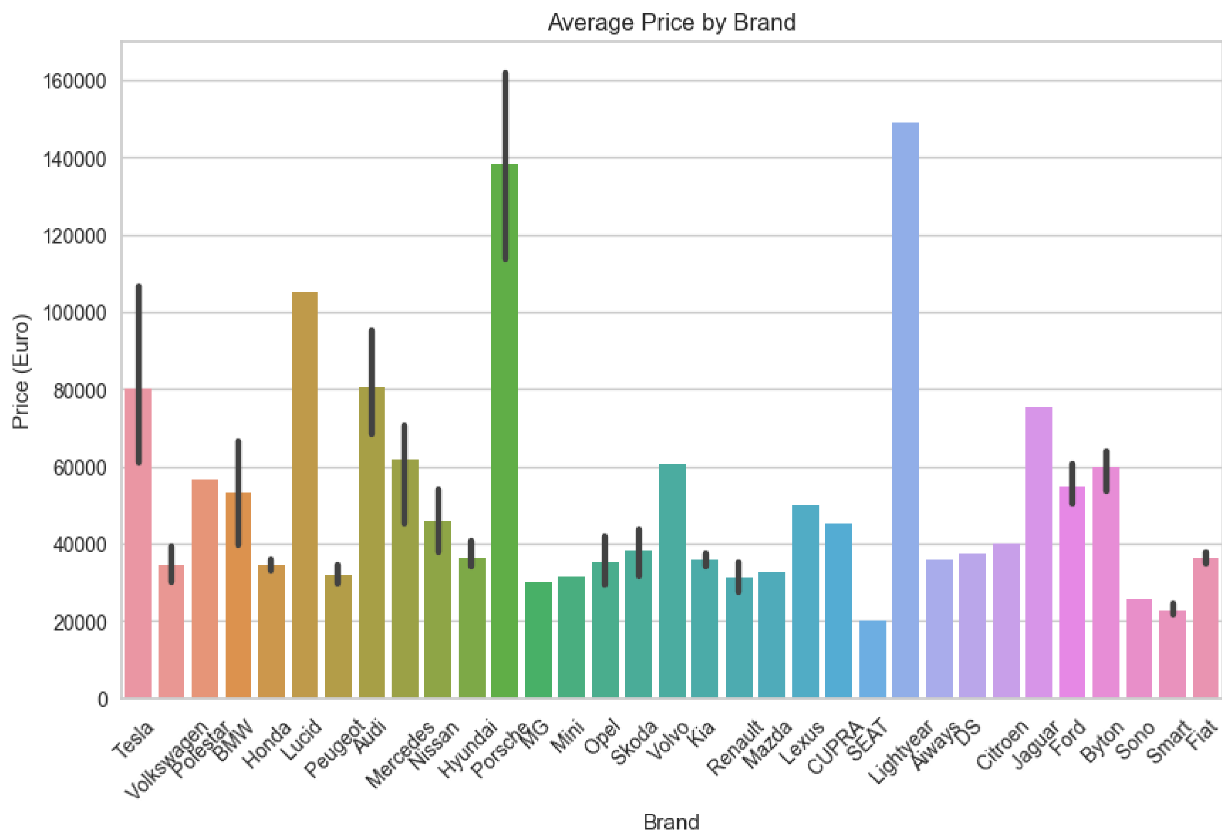
DISTRIBUTION OF RANGE **DISTRIBUTION OF EFFICIENCY**



THROUGH THIS BOX PLOT WE CAN SEE THE OUTLIERS EXIST

BUT Outlier exist on number of seats
Some cars on ev have 2 seats some have more than 5 seats

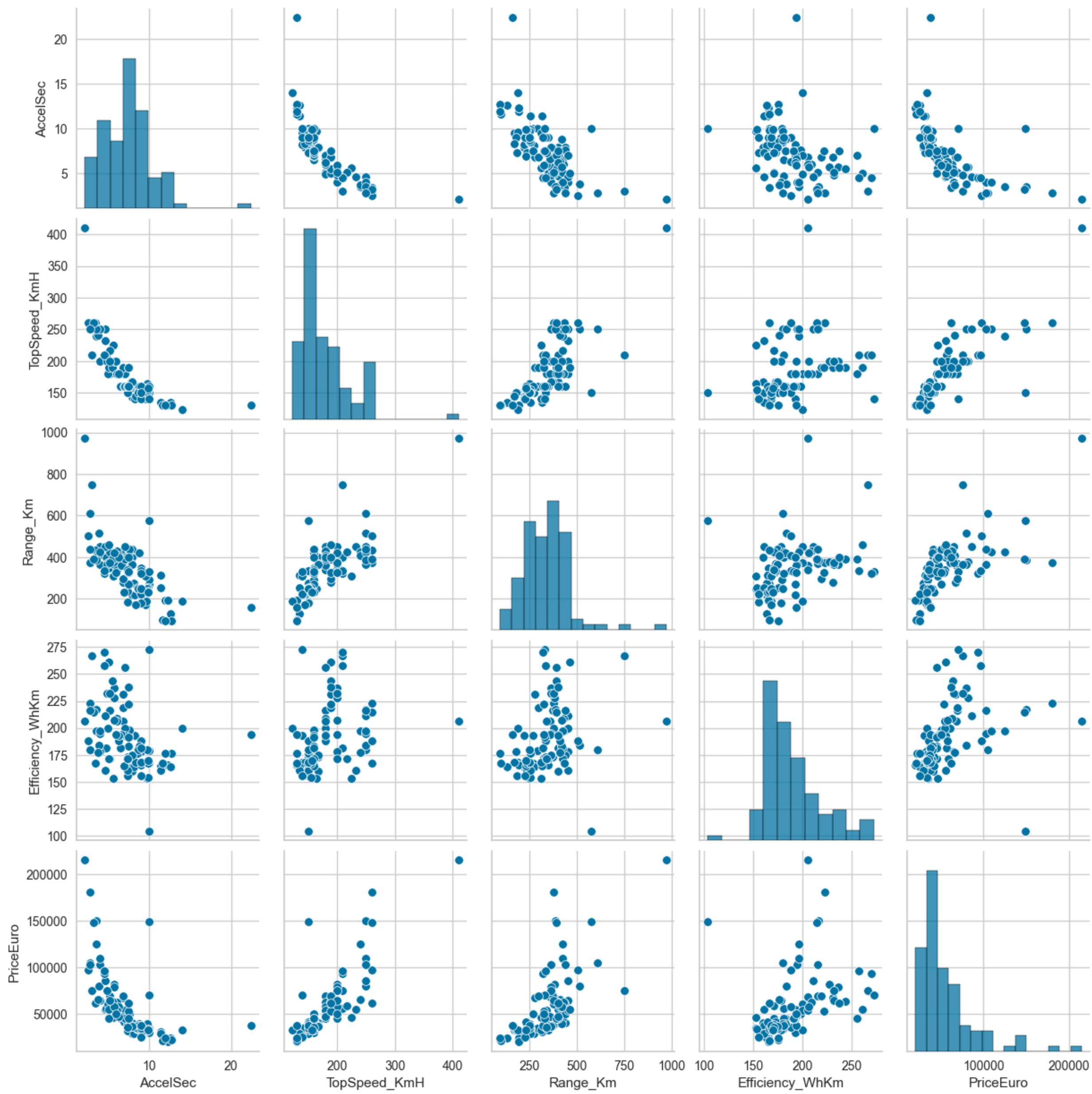
On other side Fast charge and Rapid charge does not have outliers.



	Tesla
1	**Volkswagen**
2	**Polestar**
3	**BMW**
4	**Honda**
	...
98	**Nissan**
99	**Audi**
100	**Nissan**
101	**Nissan**
102	**Byron**

Those and 102 car exist on brands which can be see through barplot and their price .

In given dataset PORSCHE has high price .



AccelSec
TopSpeed_KmH
Range_Km
Efficiency_WhKm
PriceEuro