Feynn Labs Internship

(January, 2023)

Project Report on

<u>EV Market Segmentation Analysis</u>

Submitted By,

Arya Shah

Batch-2-23-SB MLI

Table of Contents

Background

The electric vehicle market in India is currently experiencing a period of significant growth and development. The Indian government has set ambitious targets for the adoption of electric vehicles, with a goal of achieving 30% electric vehicle penetration by 2030. To support this goal, the government has implemented a range of policies and initiatives, including tax incentives, subsidies, and funding for research and development.

The electric vehicle market in India is largely driven by the demand for environmentally-friendly and sustainable transportation options. The Indian population is becoming increasingly aware of the negative impact of traditional gasoline and diesel-powered vehicles on the environment, and as a result, there is a growing interest in electric vehicles as a cleaner and more sustainable alternative.

There are a number of factors that are driving the growth of the electric vehicle market in India. One of the key factors is the increasing availability of charging infrastructure. The Indian government has launched a number of initiatives to promote the development of charging infrastructure across the country, and private companies are also investing in this area.

Another important factor is the declining cost of electric vehicles. In recent years, the cost of electric vehicles has been steadily decreasing, making them more accessible to a wider range of consumers. The Indian government has also implemented a number of policies and incentives to make electric vehicles more affordable, such as tax exemptions and subsidies.

Despite these positive trends, there are still several challenges facing the electric vehicle market in India. One of the biggest challenges is the lack of awareness and education among consumers about electric vehicles. Many consumers are still unfamiliar with the technology and may not understand the benefits of electric vehicles.

To summarize, we can say that the electric vehicle market in India is poised for significant growth in the coming years. A thorough segmentation analysis can help businesses and policymakers better understand the key drivers and challenges of the market, and develop effective strategies for capturing market share and promoting sustainable transportation options.

Problem Statement

The electric vehicle market in India is experiencing significant growth and development, driven by a range of factors including government policies, environmental concerns, and declining costs. However, despite these positive trends, the market still faces several challenges, including a lack of consumer awareness and education about electric vehicles. The problem statement for this report is to conduct a segmentation analysis of the electric vehicle market in India in order to identify key consumer segments and understand their attitudes, behaviours, and preferences towards electric vehicles. This analysis will help businesses and policymakers develop targeted strategies for promoting electric vehicle adoption and addressing the challenges facing the market.

In this report we analyse the Electric Vehicles Market in India using segments such as price, top speed, range, safety, battery capacity, fuel types, fast charging, boot space and much more.

Data Collection

The data collection step for the segmentation analysis of the electric vehicle market in India will involve gathering information from a variety of sources. One important source of data will be websites that provide information about electric vehicles and the Indian automotive market.

To collect data for different bases of segmentation, we will scrape information from websites that cater to different segments of the market. For example, to understand the attitudes and preferences of environmentally conscious consumers, we may scrape information from websites that focus on sustainability and eco-friendly living. Similarly, to understand the needs and preferences of consumers in different geographic regions, we may scrape information from local news sites and automotive forums.

In addition to scraping information from websites, we may also collect data from surveys and interviews with key stakeholders in the electric vehicle market, including consumers, dealers, and manufacturers. This will help us gather more detailed and specific information about consumer attitudes and preferences, as well as industry trends and challenges.

Once we have collected a sufficient amount of data, we will use statistical analysis techniques to identify meaningful segments within the market. These segments may be based on factors such as geographic location, income level, age, or lifestyle, and will help us better understand the different needs and preferences of consumers in the electric vehicle market.

So, Data was scraped from the website https://e-amrit.niti.gov.in/home.

e-AMRIT (**Accelerated e-Mobility Revolution for India's**

Transportation) is portal for creating awareness about electric mobility in India.

Also for some specification of Electrical Vehicle we gathered from https://www.cardekho.com/.

The data is partly used for visualization purpose and partly for clustering.

Code & Documentation

The complete code along with the dataset is available at the following GitHub Links:

Main Link: https://github.com/aryashah2k/Feynn-Labs

Assignment Specific Link:

Name	Link
Dataset	https://github.com/aryashah2k/Feynn-Labs/tree/main/EV%20Market%20Segmentation%20Analysis/Datasets
Notebook	https://github.com/aryashah2k/Feynn-Labs/blob/main/EV%20Market%20Segmentation%20Analysis/EV%20Market%20Segmentation%20Analysis.ipynb

Name: Arya Shah

Batch-2-23-SB MLI

Feynn Labs Internship

EV Market Segmentation Analysis

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

```
data=pd.read_csv('Final EV data.csv')
data.head()
```

```
       Vehicle full name    Manufacturing      Model  Top speed (km/hr)  \
0           Revolt RV400    Revolt Motors      RV400               85.0
1           Revolt RV300    Revolt Motors      RV300               65.0
2   Tork Motors(Kratos )      Tork Motors     Kratos              100.0
3  Tork Motors(Kratos R)      Tork Motors   Kratos R              105.0
4              Oben Rorr  Kabira Mobility  Oben Rorr              100.0

   Price (INR) Fuel Type Wheelers type  Battery capacity [kWh]  \
0     134000.0  Electric   Two wheeler                     4.0
1      94999.0  Electric   Two wheeler                     2.7
2     192499.0  Electric   Two wheeler                     4.0
3     207499.0  Electric   Two wheeler                     4.0
4     102999.0  Electric   Two wheeler                     4.4

   Full charging time (HR)  Kerb weight (KG)  Range (km/hr) Fast Charging  \
0                      4.5             108.0          150.0           YES
1                      4.2             101.0          180.0           YES
2                      5.0               NaN          180.0            NO
3                      5.0               NaN          180.0           YES
4                      2.0             110.0          200.0           YES

   Drive Type  Number of Seats  boot space (L)  Number of Airbags  \
0  Belt Drive                2             NaN                NaN
1   Hub Drive                2             NaN                NaN
2         NaN                2             NaN                NaN
3         NaN                2             NaN                NaN
4  Belt Drive                2             NaN                NaN
```

```
  Type of brakes  Max Torque (N-M) Type of Vehicle
0           Disc             170.0    Motor cycles
1           Disc               NaN    Motor cycles
2           Disc              28.0    Motor cycles
3           Disc              38.0    Motor cycles
4           Disc               NaN    Motor cycles
```

Description Of Columns

- Vehicle full name - Name of vehicle
- Manufacturing - Manufacturing company of vehicle
- Model - Model of vehicle
- Top speed (km/hr) - Maximum speed of vehicle in (km/hr)
- Price (INR) - Price of vehicle
- Fuel Type - Type of fuel (Electrical, Hybrid)
- Wheelers type - Type of wheelers(Two,Three,Four wheelers)
- Battery capacity [kWh] - Capacity of battery in (kwh)
- Full charging time (HR) - Total charging time 100% in (hr)
- Kerb weight (KG) - Total weight of vehicle in (kg)
- Range (km/hr) - Maximum kilometer covered per charging in (km/hr)
- Fast Charging - Vehicle have fast charging or not
- Drive Type - Type of Drive
- Number of Seats - Number of Seats in vehicle
- boot space (L) - Space for luggages in (Liter)
- Number of Airbags - Airbags for safety
- Type of brakes - Type of brakes
- Max Torque (N-M) - Max torque (n-m)
- Type of Vehicle - Vehicle types (Scooter, Cars,etc.)
- Income - Price range of vehicle (Thousands, Lakhs, Crore)

```
charging_station=pd.read_excel('charging_station.xlsx')
charging_station.head()
```

```
       State wise  Number of Electric Vehicle Charging Sanctioned
0     Maharashtra                                             317
```

```
1  Andhra Pradesh                                                266
2      Tamil Nadu                                                256
3         Gujarat                                                228
4   Uttar Pradesh                                                207
```

```
sales=pd.read_excel('EV_sales.xlsx')
sales.head()
```

```
       Years  Two Wheeler  Three Wheeler  Four Wheeler
0  Year 2020       152000         140683        168300
1  Year 2021       143837          88378        134821
2  Year 2022       231338         384215        429217
```

Data Preprocessing

Steps taken to preprocess the raw data scraped:

1. Dealing with different variables names but having the same information in columns, so we replace it.

```
data['Wheelers type']=data['Wheelers type'].replace('four wheeler','Four Wheeler')
data['Wheelers type']=data['Wheelers type'].replace('Four Wheeler','Four wheeler')
data['Fast Charging']=data['Fast Charging'].replace('NO','No')
data['Fast Charging']=data['Fast Charging'].replace('YES','Yes')
data['Fuel Type']=data['Fuel Type'].replace('electric','Electric')
```

1. Create Income feature for range between Low(Thousands), Medium(Lakhs), High(Crore).

```
def income(price):
    if price <= 100000:
        return 'Low (Thousands)'
    elif price>100000 and price<10000000:
        return 'medium (Lakhs)'
    else:
        return 'High(Crore)'
```

```
data['Income'] = data['Price (INR)'].apply(income)
```

1. Deals Null values in the dataset by filling them with mean values.

```
data['Top speed (km/hr)']=data['Top speed (km/hr)'].fillna(data['Top speed (km/hr)'].mean())
data['Price (INR)']=data['Price (INR)'].fillna(data['Price (INR)'].mean())
data['Battery capacity [kWh]']=data['Battery capacity [kWh]'].fillna(data['Battery capacity
data['Kerb weight (KG)']=data['Kerb weight (KG)'].fillna(data['Kerb weight (KG)'].mean())
data['Max Torque (N-M)']=data['Max Torque (N-M)'].fillna(data['Max Torque (N-M)'].mean())
data['Full charging time (HR)']=data['Full charging time (HR)'].fillna(data['Full charging t
data['Range (km/hr)']=data['Range (km/hr)'].fillna(data['Range (km/hr)'].mean())
```

```
data[' Drive Type']=data[' Drive Type'].fillna(data[' Drive Type'].mode()[0])
data['Type of brakes']=data['Type of brakes'].fillna(data['Type of brakes'].mode()[0])
```

```
data['Type of brakes'].mode()[0]
```

```
'disc (front + rear)'
```

Exploratory Data Analysis

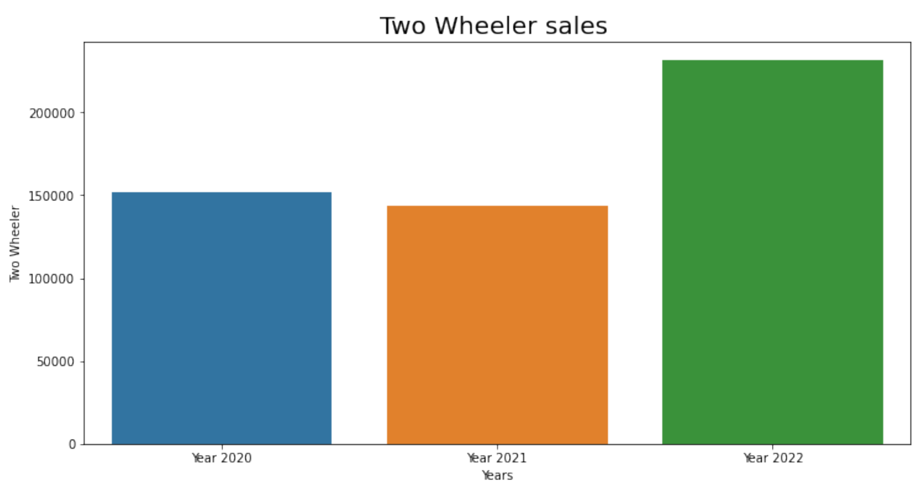
Exploratory Data Analysis (EDA) is the process of describing the data by means of statistical and visualization techniques in order to bring important aspects of that data into focus for further analysis.

For analysis, we took some features for visualization from our dataset as shown below:

```
plt.figure(figsize=(12,6))
print(sns.barplot(y=sales['Two Wheeler'],x=sales['Years']))
plt.title('Two Wheeler sales ',fontsize = 20)
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```

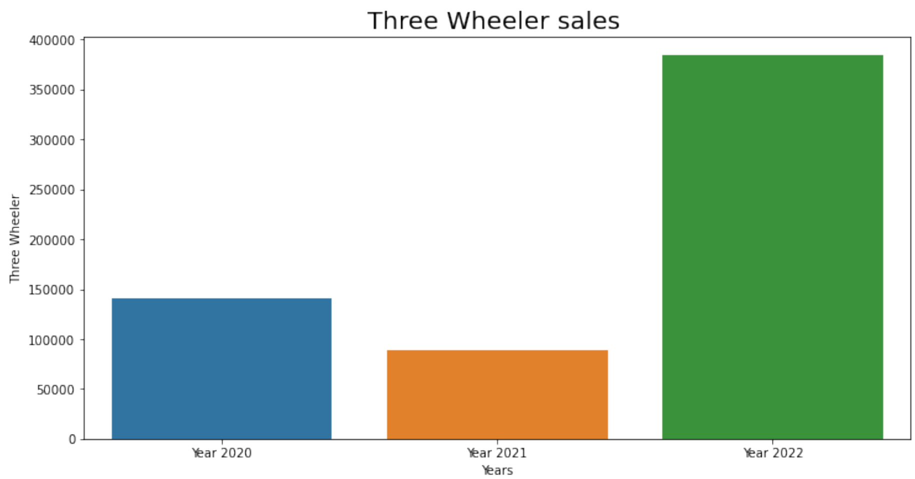
```
Text(0.5, 1.0, 'Two Wheeler sales ')
```




```
plt.figure(figsize=(12,6))
print(sns.barplot(y=sales['Three Wheeler'],x=sales['Years']))
plt.title('Three Wheeler sales ',fontsize = 20)
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```

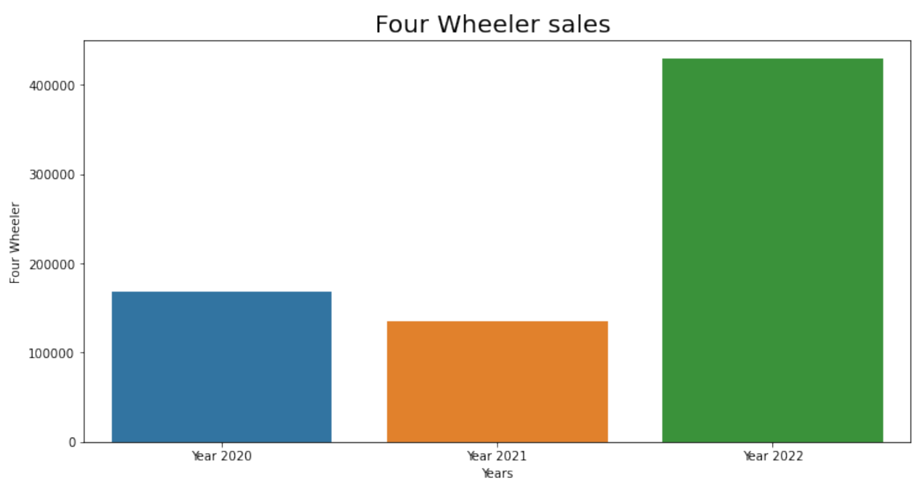
```
Text(0.5, 1.0, 'Three Wheeler sales ')
```




```
plt.figure(figsize=(12,6))
print(sns.barplot(y=sales['Four Wheeler'],x=sales['Years']))
plt.title('Four Wheeler sales ',fontsize = 20)
```

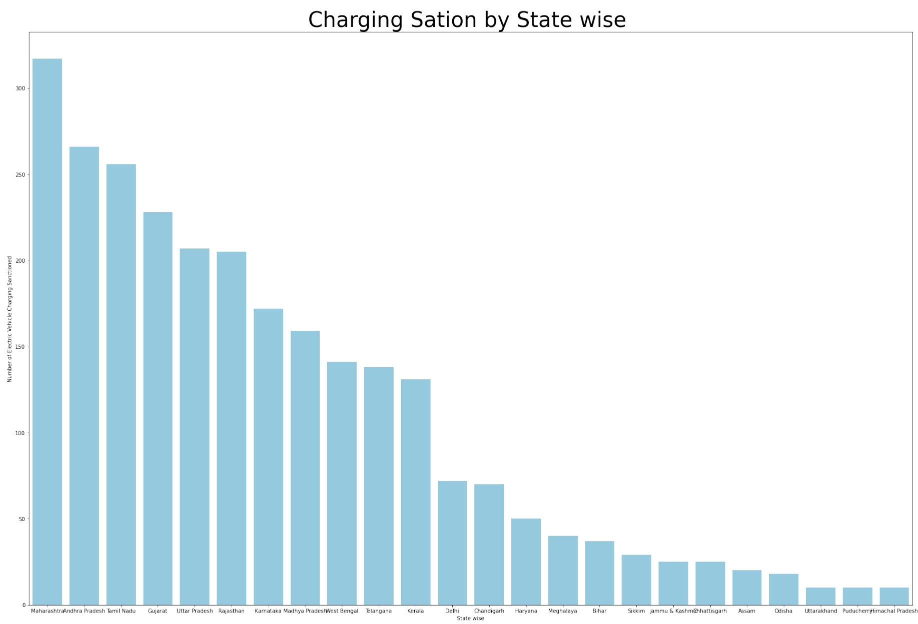
```
AxesSubplot(0.125,0.125;0.775x0.755)
```

```
Text(0.5, 1.0, 'Four Wheeler sales ')
```




```
plt.figure(figsize=(30,20))
sns.barplot(charging_station['State wise'],x=charging_station['State wise'],
    y=charging_station['Number of Electric Vehicle Charging Sanctioned'],color='skyblue')
plt.title('Charging Sation by State wise ',fontsize = 40)
```

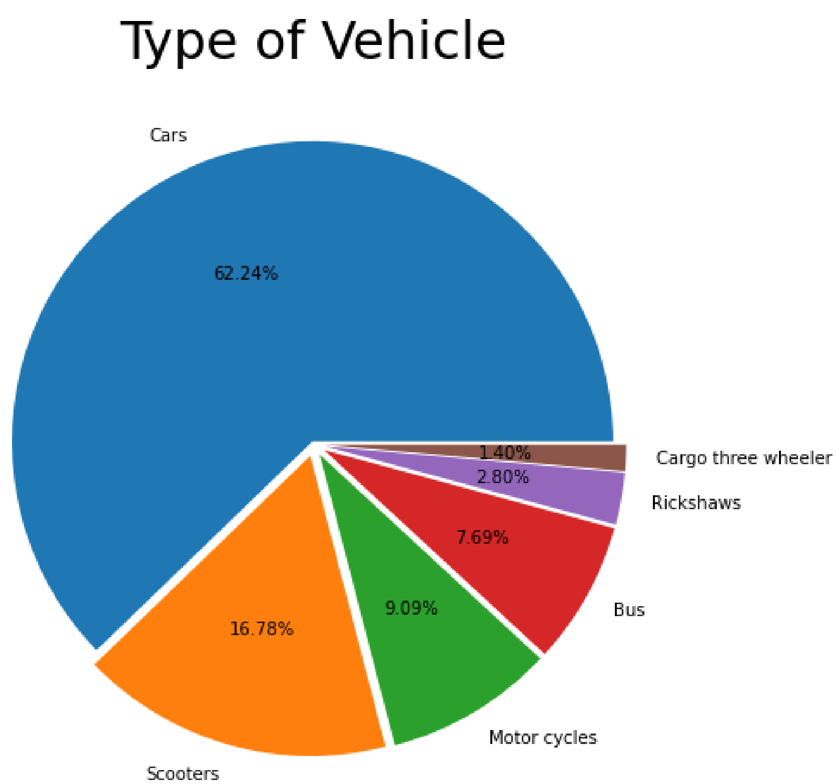
```
Text(0.5, 1.0, 'Charging Sation by State wise ')
```



We can see numbers of charging stations present in India as per states. The maximum number of charging stations present in Maharashtra and lowest in Himachal Pradesh.

```
plt.figure(figsize=(25,8))
explode = [0.01,0.04,0.04,0.04,0.04,0.04]
labels=['Cars','Scooters','Motor cycles','Bus','Rickshaws','Cargo three wheeler']
plt.pie(data['Type of Vehicle'].value_counts(),
        labels=labels,autopct = '%.2f%%',explode=explode)
plt.title('Type of Vehicle', fontsize = 30)
```

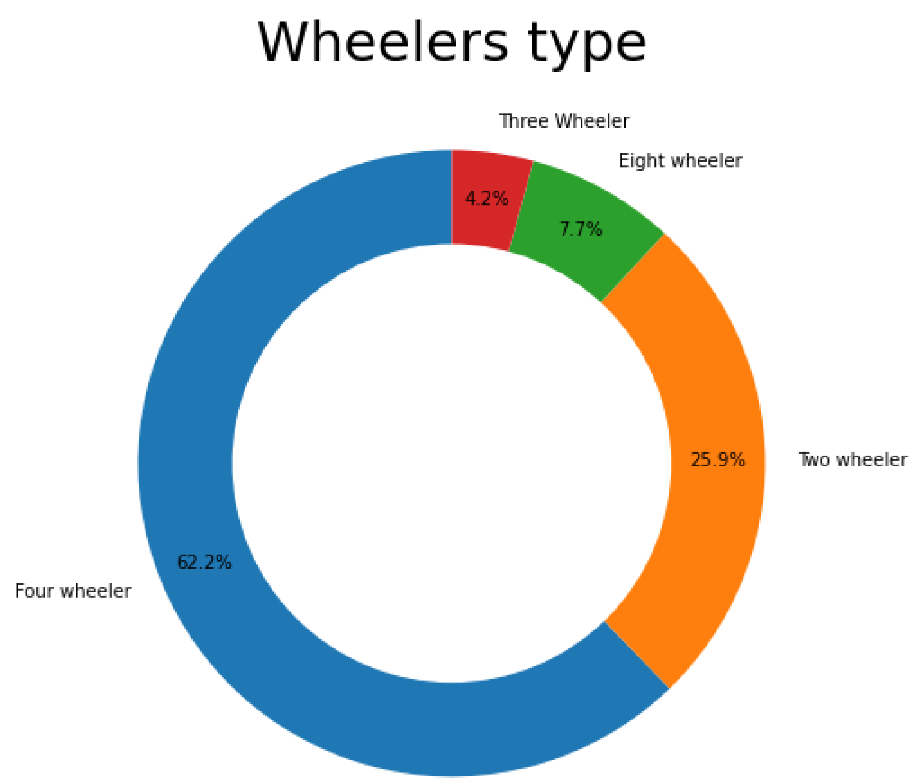
```
Text(0.5, 1.0, 'Type of Vehicle')
```



Above figure shows percentages of Electrical vehicles types in india. Basically it shows a manufacturing market percentage of every type of vehicle. In that we can see that the market of Cars is high. A lot of EV startup companies are manufacturing or focused on only Electricals Cars. Also there is less market for Cargo and Rickshaws. A very less number of companies are focusing on Cargo and Rickshaws.

```
plt.figure(figsize=(25,8))
labels=['Four wheeler','Two wheeler','Eight wheeler','Three Wheeler']
plt.pie(data['Wheelers type'].value_counts(),labels=labels, autopct='%1.1f%%',
         startangle=90, pctdistance=0.85,)
plt.title('Wheelers type', fontsize = 30)
centre_circle = plt.Circle((0,0),0.70,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
```

```
<matplotlib.patches.Circle at 0x1e57fc25e20>
```




```
data.head()
```

```
      Vehicle full name    Manufacturing      Model  Top speed (km/hr)  \
0          Revolt RV400    Revolt Motors      RV400               85.0
1          Revolt RV300    Revolt Motors      RV300               65.0
2   Tork Motors(Kratos )     Tork Motors     Kratos              100.0
3  Tork Motors(Kratos R)     Tork Motors   Kratos R              105.0
4             Oben Rorr  Kabira Mobility  Oben Rorr              100.0

   Price (INR) Fuel Type Wheelers type  Battery capacity [kWh]  \
0     134000.0  Electric   Two wheeler                     4.0
1      94999.0  Electric   Two wheeler                     2.7
2     192499.0  Electric   Two wheeler                     4.0
3     207499.0  Electric   Two wheeler                     4.0
4     102999.0  Electric   Two wheeler                     4.4

   Full charging time (HR)  Kerb weight (KG)  Range (km/hr) Fast Charging  \
0                      4.5        108.000000          150.0           Yes
```

```
1                     4.2       101.000000         180.0           Yes
2                     5.0      1506.382114         180.0            No
3                     5.0      1506.382114         180.0           Yes
4                     2.0       110.000000         200.0           Yes

   Drive Type   Number of Seats  boot space (L)   Number of Airbags  \
0  Belt Drive                 2             NaN                 NaN
1   Hub Drive                 2             NaN                 NaN
2         FWD                 2             NaN                 NaN
3         FWD                 2             NaN                 NaN
4  Belt Drive                 2             NaN                 NaN

  Type of brakes  Max Torque (N-M) Type of Vehicle           Income
0           Disc         170.00000    Motor cycles   medium (Lakhs)
1           Disc         346.74958    Motor cycles  Low (Thousands)
2           Disc          28.00000    Motor cycles   medium (Lakhs)
3           Disc          38.00000    Motor cycles   medium (Lakhs)
4           Disc         346.74958    Motor cycles   medium (Lakhs)
```

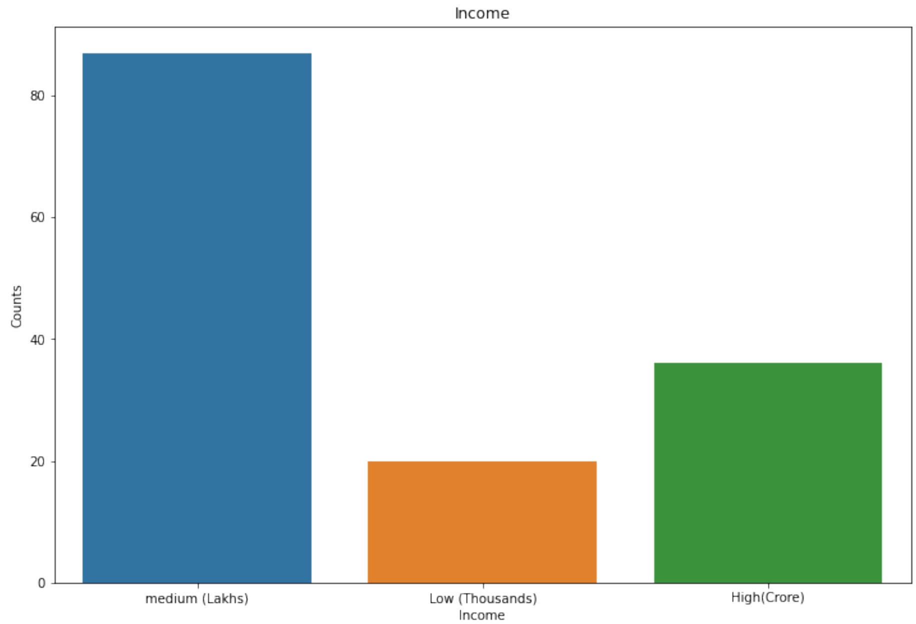
```
final=['Top speed (km/hr)','Price (INR)', 'Full charging time (HR)','Fuel Type','Battery cap
       'Kerb weight (KG)','Fast Charging', ' Drive Type','Wheelers type',' Number of Seats',
      ]
new_data=data.loc[:,final]
new_data
```

```
     Top speed (km/hr)   Price (INR)  Full charging time (HR) Fuel Type  \
0             85.00000  1.340000e+05                 4.500000  Electric
1             65.00000  9.499900e+04                 4.200000  Electric
2            100.00000  1.924990e+05                 5.000000  Electric
3            105.00000  2.074990e+05                 5.000000  Electric
4            100.00000  1.029990e+05                 2.000000  Electric
..                 ...           ...                      ...       ...
138           65.00000  3.893761e+06                 3.000000  Electric
139           75.00000  1.600000e+07                 2.500000  Electric
140           70.00000  1.500000e+07                 4.500000  Electric
141          129.76259  3.893761e+06                 7.344911  Electric
142          129.76259  3.893761e+06                 7.344911  Electric

     Battery capacity [kWh]  Range (km/hr)  Kerb weight (KG) Fast Charging  \
0                  4.000000     150.000000        108.000000           Yes
1                  2.700000     180.000000        101.000000           Yes
2                  4.000000     180.000000       1506.382114            No
3                  4.000000     180.000000       1506.382114           Yes
4                  4.400000     200.000000        110.000000           Yes
..                      ...            ...               ...           ...
138              250.000000     200.000000       1506.382114           Yes
139              124.000000     150.000000       1506.382114           Yes
```

```
140              41.355385     300.000000       1506.382114          Yes
141              41.355385     293.126929       1506.382114          Yes
142              41.355385     293.126929       1506.382114          Yes

     Drive Type  Wheelers type   Number of Seats       Type of brakes  \
0    Belt Drive    Two wheeler                 2                 Disc
1     Hub Drive    Two wheeler                 2                 Disc
2           FWD    Two wheeler                 2                 Disc
3           FWD    Two wheeler                 2                 Disc
4    Belt Drive    Two wheeler                 2                 Disc
..          ...            ...               ...                  ...
138         FWD  Eight wheeler                31  disc (front + rear)
139         FWD  Eight wheeler                31    front disc brakes
140         FWD  Eight wheeler                39  disc (front + rear)
141         FWD  Eight wheeler                43  disc (front + rear)
142         FWD  Eight wheeler                35  disc (front + rear)

     Max Torque (N-M)           Income
0           170.00000   medium (Lakhs)
1           346.74958  Low (Thousands)
2            28.00000   medium (Lakhs)
3            38.00000   medium (Lakhs)
4           346.74958   medium (Lakhs)
..                ...              ...
138         346.74958      High(Crore)
139        3000.00000      High(Crore)
140         800.00000      High(Crore)
141         346.74958      High(Crore)
142         346.74958      High(Crore)

[143 rows x 14 columns]
```

```
#Income Feature
plt.figure(figsize=(12,8))
sns.countplot(new_data['Income'])
plt.title('Income')
plt.ylabel('Counts')
```

```
Text(0, 0.5, 'Counts')
```



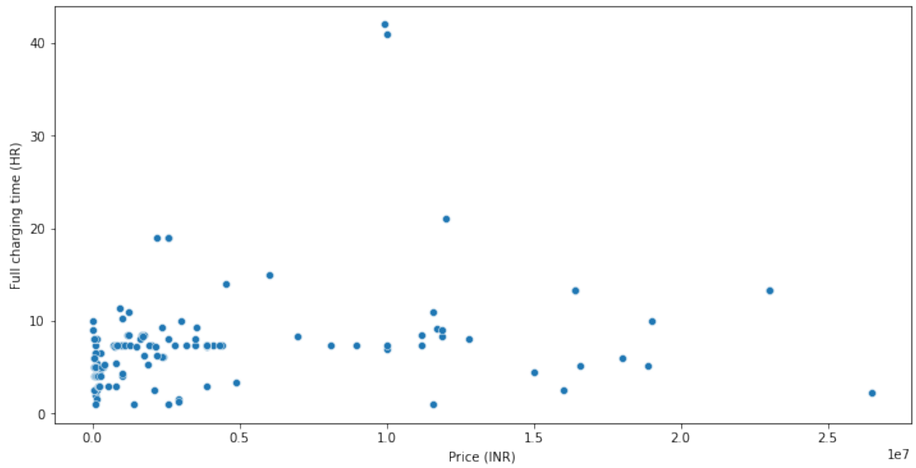
Above figure Shows a plot of information about Income feature. we categorized Income features in three different types as, first in Low means the price of EV is in thousands rupees (Less than 1 lakhs), second in Medium means the price of EV is in lakhs (Between 1 lakh to 1 crore) and Third in High means the price of EV is in crore (Greater than 1 crore). As from countplot we can conclude that the maximum EV's price is in lakhs (Medium).

```
sales.head()
```

	Years	Two Wheeler	Three Wheeler	Four Wheeler
0	Year 2020	152000	140683	168300
1	Year 2021	143837	88378	134821
2	Year 2022	231338	384215	429217

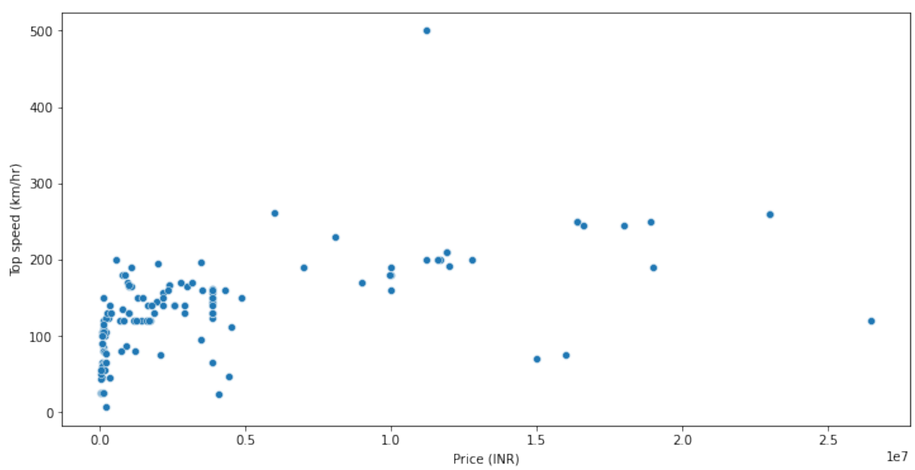
```
plt.figure(figsize=(12,6))
sns.scatterplot(x='Price (INR)',y='Full charging time (HR)',data=data)
```

```
<AxesSubplot:xlabel='Price (INR)', ylabel='Full charging time (HR)'>
```




```
#Scatter plot between Price and Top speed
plt.figure(figsize=(12,6))
sns.scatterplot(x='Price (INR)',y='Top speed (km/hr)',data=new_data)
```

```
<AxesSubplot:xlabel='Price (INR)', ylabel='Top speed (km/hr)'>
```

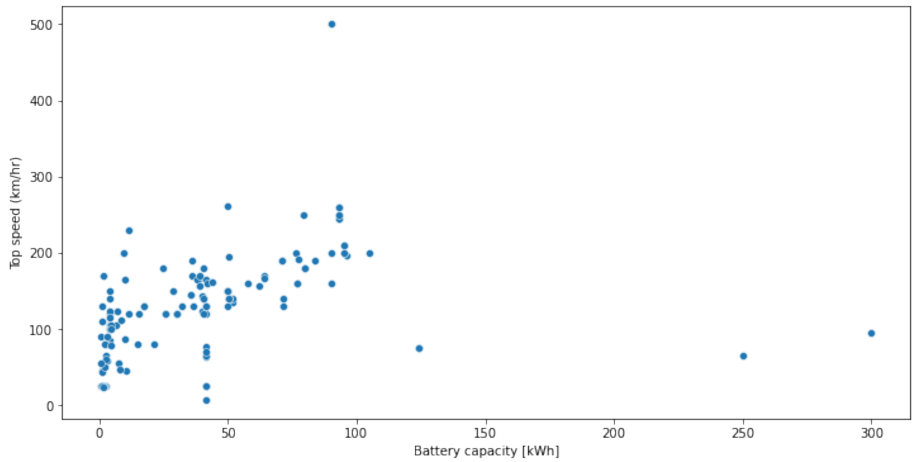


Above figure shows a scatter plot between Top speed vs Price to see the relation between them. As from this scatter plot ,we can conclude that if the Top Speed of EV is increasing then the Price of EV is also increasing.

Both are directly proportional to each other.

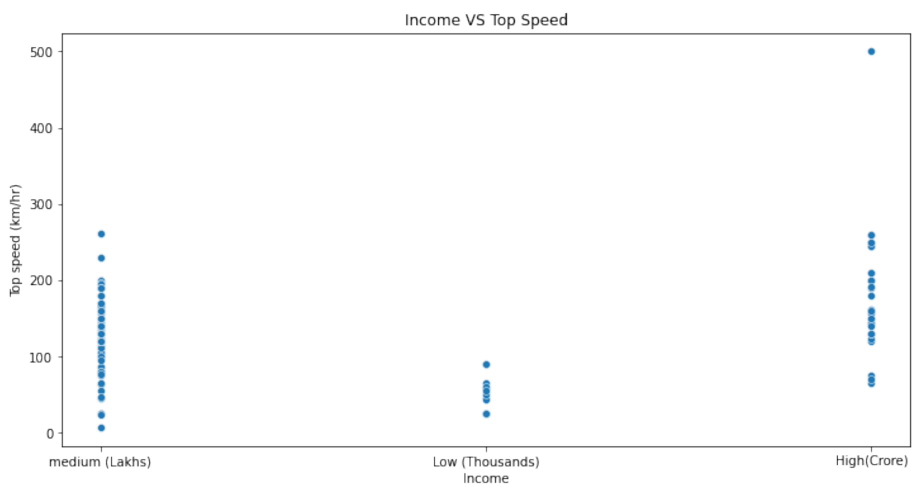
```
plt.figure(figsize=(12,6))
sns.scatterplot(x='Battery capacity [kWh]',y='Top speed (km/hr)',data=new_data)
```

```
<AxesSubplot:xlabel='Battery capacity [kWh]', ylabel='Top speed (km/hr)'>
```




```
#Scatter plot between Income and Top speed
plt.figure(figsize=(12,6))
sns.scatterplot(x='Income',y='Top speed (km/hr)',data=new_data)
plt.title('Income VS Top Speed')
```

```
Text(0.5, 1.0, 'Income VS Top Speed')
```



This figure shows a relationship between Income and Top speed. We can see that if the price of EV in Low (thousands) then your top speed lies within 0-110 km/hr. As the price increases your vehicle's top speed also increases

```
new_data.isna().sum()
```

```
Top speed (km/hr)          0
Price (INR)                0
Full charging time (HR)    0
```

```
Fuel Type                 0
Battery capacity [kWh]    0
Range (km/hr)             0
Kerb weight (KG)          0
Fast Charging             0
 Drive Type               0
Wheelers type             0
 Number of Seats          0
Type of brakes            0
Max Torque (N-M)          0
Income                    0
dtype: int64
```

```python
from sklearn.preprocessing import LabelEncoder

features =['Wheelers type', ' Drive Type', 'Type of brakes','Fast Charging','Income','Fuel T

for i in features:
     new_data[i] =LabelEncoder().fit_transform(new_data[i])
new_data
```

```
     Top speed (km/hr)   Price (INR)  Full charging time (HR)  Fuel Type  \
0             85.00000  1.340000e+05                 4.500000          0
1             65.00000  9.499900e+04                 4.200000          0
2            100.00000  1.924990e+05                 5.000000          0
3            105.00000  2.074990e+05                 5.000000          0
4            100.00000  1.029990e+05                 2.000000          0
..                 ...           ...                      ...        ...
138           65.00000  3.893761e+06                 3.000000          0
139           75.00000  1.600000e+07                 2.500000          0
140           70.00000  1.500000e+07                 4.500000          0
141          129.76259  3.893761e+06                 7.344911          0
142          129.76259  3.893761e+06                 7.344911          0

     Battery capacity [kWh]  Range (km/hr)  Kerb weight (KG)  Fast Charging  \
0                  4.000000     150.000000        108.000000              1
1                  2.700000     180.000000        101.000000              1
2                  4.000000     180.000000       1506.382114              0
3                  4.000000     180.000000       1506.382114              1
4                  4.400000     200.000000        110.000000              1
..                      ...            ...               ...            ...
138              250.000000     200.000000       1506.382114              1
139              124.000000     150.000000       1506.382114              1
140               41.355385     300.000000       1506.382114              1
141               41.355385     293.126929       1506.382114              1
142               41.355385     293.126929       1506.382114              1
```

```
     Drive Type  Wheelers type   Number of Seats  Type of brakes  \
0            11              3                 2               1
1            15              3                 2               1
2            14              3                 2               1
3            14              3                 2               1
4            11              3                 2               1
..          ...            ...               ...             ...
138          14              0                31               2
139          14              0                31               4
140          14              0                39               2
141          14              0                43               2
142          14              0                35               2

     Max Torque (N-M)  Income
0           170.00000       2
1           346.74958       1
2            28.00000       2
3            38.00000       2
4           346.74958       2
..                ...     ...
138         346.74958       0
139        3000.00000       0
140         800.00000       0
141         346.74958       0
142         346.74958       0

[143 rows x 14 columns]
```

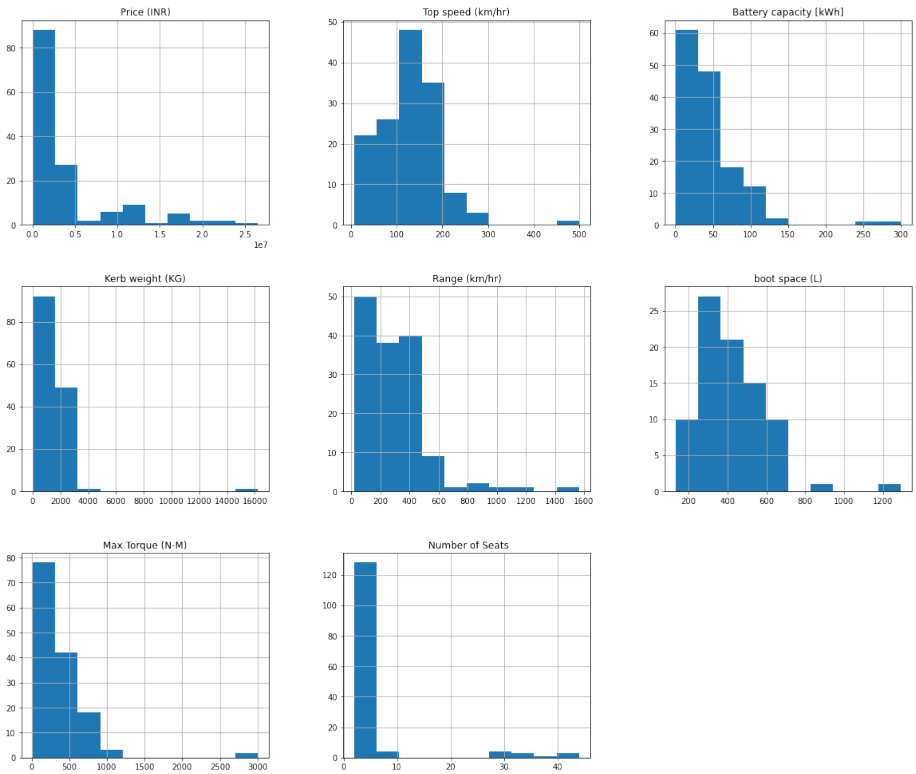
```
data.head()
```

```
       Vehicle full name    Manufacturing      Model  Top speed (km/hr)  \
0           Revolt RV400    Revolt Motors      RV400               85.0
1           Revolt RV300    Revolt Motors      RV300               65.0
2   Tork Motors(Kratos )      Tork Motors     Kratos              100.0
3  Tork Motors(Kratos R)      Tork Motors   Kratos R              105.0
4              Oben Rorr  Kabira Mobility  Oben Rorr              100.0

   Price (INR) Fuel Type Wheelers type  Battery capacity [kWh]  \
0     134000.0  Electric   Two wheeler                     4.0
1      94999.0  Electric   Two wheeler                     2.7
2     192499.0  Electric   Two wheeler                     4.0
3     207499.0  Electric   Two wheeler                     4.0
4     102999.0  Electric   Two wheeler                     4.4

   Full charging time (HR)  Kerb weight (KG)  Range (km/hr) Fast Charging  \
0                      4.5        108.000000          150.0           Yes
1                      4.2        101.000000          180.0           Yes
```

```
2                     5.0      1506.382114         180.0            No
3                     5.0      1506.382114         180.0           Yes
4                     2.0       110.000000         200.0           Yes

   Drive Type   Number of Seats  boot space (L)   Number of Airbags  \
0  Belt Drive                2             NaN                 NaN
1   Hub Drive                2             NaN                 NaN
2         FWD                2             NaN                 NaN
3         FWD                2             NaN                 NaN
4  Belt Drive                2             NaN                 NaN

  Type of brakes  Max Torque (N-M) Type of Vehicle           Income
0           Disc         170.00000    Motor cycles   medium (Lakhs)
1           Disc         346.74958    Motor cycles  Low (Thousands)
2           Disc          28.00000    Motor cycles   medium (Lakhs)
3           Disc          38.00000    Motor cycles   medium (Lakhs)
4           Disc         346.74958    Motor cycles   medium (Lakhs)
```

```
#Histogram
plt.rcParams['figure.figsize']=(20,17)
data.hist(['Price (INR)','Top speed (km/hr)','Battery capacity [kWh]','Kerb weight (KG)','Ra
           'boot space (L)','Max Torque (N-M)',' Number of Seats'])
```

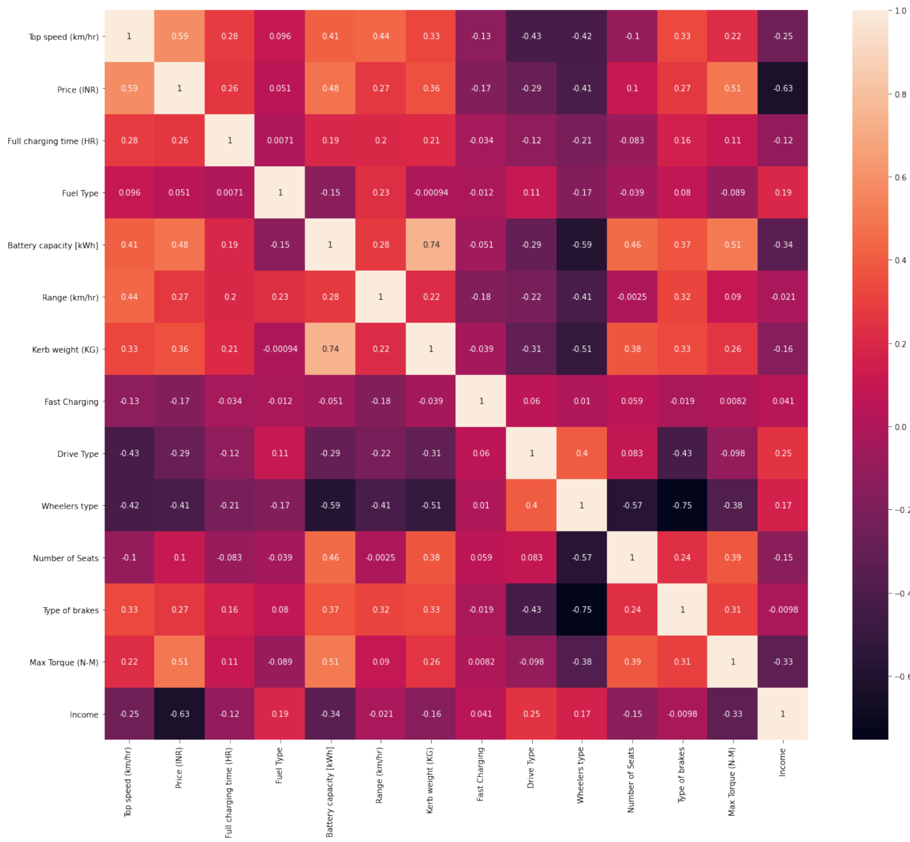
```
array([[<AxesSubplot:title={'center':'Price (INR)'}>,
        <AxesSubplot:title={'center':'Top speed (km/hr)'}>,
        <AxesSubplot:title={'center':'Battery capacity [kWh]'}>],
       [<AxesSubplot:title={'center':'Kerb weight (KG)'}>,
        <AxesSubplot:title={'center':'Range (km/hr)'}>,
        <AxesSubplot:title={'center':'boot space (L)'}>],
       [<AxesSubplot:title={'center':'Max Torque (N-M)'}>,
        <AxesSubplot:title={'center':' Number of Seats'}>,
        <AxesSubplot:>]], dtype=object)
```



In the above figure we plot histograms of every single feature. As from that we can see that mostly Price ranges between thousands to lakhs. In Top speed maximum average value is around 150km/hr, same as for Battery capacity ranges around 0-50Kwh. As a Kerb weight it averages at 0-2000kg. Most EVhas Range between 0-100km/hr. For boot space we can conclude that most EVs have 300 liter boot space. Also for maximum Torque and number of Seats, we can see torque lies between 0-400 and average EVs have 5 seats.

```
#Heatmap for chechking correlations
sns.heatmap(new_data.corr(),annot=True)
```

```
<AxesSubplot:>
```



Above figure shows the correlation between every individual variable. We can see that Kerb weight and Battery capacity have the highest correlation. Meaning if we want more battery capacity our EV weight will increase.

```
new_data.isna().sum()
```

```
Top speed (km/hr)         0
Price (INR)               0
Full charging time (HR)   0
Fuel Type                 0
Battery capacity [kWh]    0
Range (km/hr)             0
Kerb weight (KG)          0
Fast Charging             0
 Drive Type               0
Wheelers type             0
 Number of Seats          0
Type of brakes            0
Max Torque (N-M)          0
Income                    0
```

```
dtype: int64
```

```
x = new_data.loc[:,final].values
x
```

```
array([[8.50000000e+01, 1.34000000e+05, 4.50000000e+00, ...,
        1.00000000e+00, 1.70000000e+02, 2.00000000e+00],
       [6.50000000e+01, 9.49990000e+04, 4.20000000e+00, ...,
        1.00000000e+00, 3.46749580e+02, 1.00000000e+00],
       [1.00000000e+02, 1.92499000e+05, 5.00000000e+00, ...,
        1.00000000e+00, 2.80000000e+01, 2.00000000e+00],
       ...,
       [7.00000000e+01, 1.50000000e+07, 4.50000000e+00, ...,
        2.00000000e+00, 8.00000000e+02, 0.00000000e+00],
       [1.29762590e+02, 3.89376089e+06, 7.34491071e+00, ...,
        2.00000000e+00, 3.46749580e+02, 0.00000000e+00],
       [1.29762590e+02, 3.89376089e+06, 7.34491071e+00, ...,
        2.00000000e+00, 3.46749580e+02, 0.00000000e+00]])
```

Principal component analysis

```
#Principal component analysis

from sklearn.decomposition import PCA
from sklearn import preprocessing

pca_data = preprocessing.scale(x)

pca = PCA(n_components=13)
pc = pca.fit_transform(x)
names = ['pc1','pc2','pc3','pc4','pc5','pc6','pc7','pc8','pc9','pc10','pc11','pc12','pc13']
pf = pd.DataFrame(data = pc, columns = names)
pf
```

```
              pc1          pc2          pc3        pc4         pc5  \
0   -3.759761e+06 -1031.676497   -12.795436 -82.651999   -4.139288
1   -3.798762e+06 -1030.126184   162.762464 -40.313961  -24.227808
2   -3.701262e+06   357.275408  -190.168112 -96.207267   -2.523818
3   -3.686262e+06   356.050970  -180.809000 -95.151492    2.477828
4   -3.790762e+06 -1021.264643   159.982353 -17.550615    8.698158
..            ...          ...          ...        ...         ...
138 -7.077936e-04     1.543989    15.534895 -94.193375  -40.399071
139  1.210624e+07 -1142.792554  2268.255021 -83.524449  -75.618769
140  1.110624e+07 -1093.870469    95.359154 -96.522639 -125.663079
141  5.641733e-06     0.080624     0.344553  -0.072516   -1.370983
142  4.324157e-06     0.062608     0.269780  -0.057712   -1.082852
```

```
            pc6        pc7       pc8       pc9      pc10      pc11      pc12  \
0     -2.241033  -1.711551 -1.851184  1.939037 -0.623458  0.922368 -0.130822
1     -8.440665  -2.974352 -1.655089 -1.645485 -1.026680 -0.187068 -0.079442
2    -22.356337  -0.881796 -1.466224 -2.267081 -0.698099  0.640194 -0.349540
3    -23.082612  -0.705079 -1.472555 -2.362056 -0.709881  0.582187 -0.227120
4     -9.717611  -1.579581 -4.542867  1.175677 -0.754629  0.877735 -0.116837
..          ...        ...       ...       ...       ...       ...       ...
138  215.053664   5.409572 -0.712451 -0.584748  0.292914 -0.074643  0.324397
139   -5.083500   2.247052 -4.298553 -0.282006  1.023209 -0.461074 -0.703183
140  -10.251712  28.556664  1.417211  2.937555 -0.276379  0.363943 -0.185341
141    3.015294  35.666821  6.542323  3.414835 -1.245348 -0.492611 -0.301332
142    2.357722  27.928745  5.210369  2.175178 -0.906757 -0.811800 -0.077199

         pc13
0   -0.078987
1   -0.071603
2   -0.127859
3   -0.003067
4   -0.057154
..        ...
138 -0.436049
139 -0.445991
140  0.152629
141 -0.084651
142  0.056288

[143 rows x 13 columns]
```

```
#Proportion of Variance (from PC1 to PC11)
pca.explained_variance_ratio_
```

```
array([9.99999928e-01, 6.62858673e-08, 3.63440277e-09, 1.47816729e-09,
       8.42315226e-11, 1.86847581e-11, 1.63760362e-12, 8.14794549e-13,
       5.53857821e-13, 1.89147953e-14, 9.98663198e-15, 3.33581149e-15,
       1.87075367e-15])
```

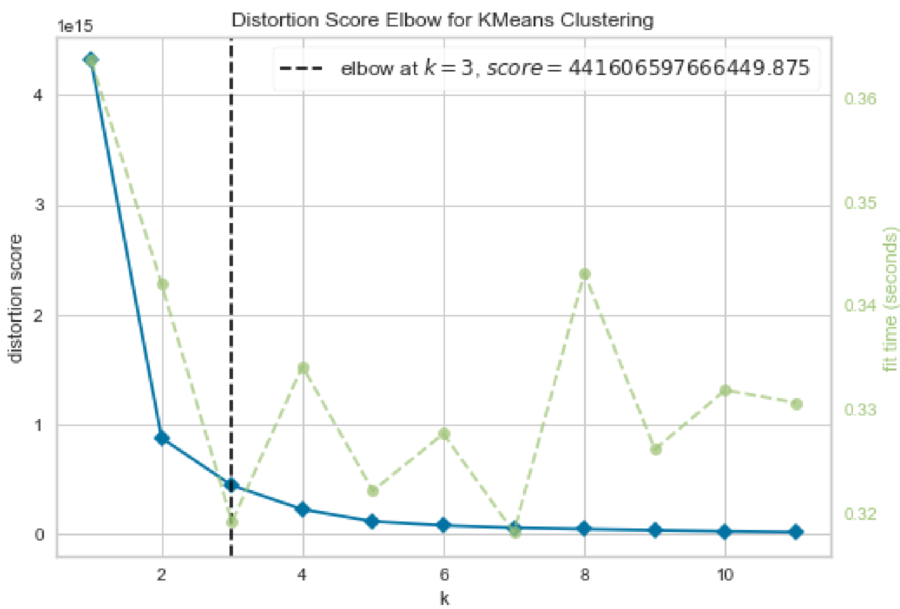
```
loadings = pca.components_
num_pc = pca.n_features_
```

K-Means clustering analysis

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters. The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

We start by pre-processing the data and cleaning it. This essentially involves null-handling ,label encoding and dummies variables in the ordinal parameters of the data. The data is then passed into the Scikit-Learn K-Means Clustering model to obtain the elbow curve for the ideal number of clusters. Using the "elbow" or "knee of a curve" as a cutoff point is a common heuristic in mathematical optimization to choose a point where diminishing returns are no longer worth the additional cost.

```python
#Extracting segments
#Using k-means clustering analysis
from sklearn.cluster import KMeans
from yellowbrick.cluster import KElbowVisualizer
model = KMeans()
visualizer = KElbowVisualizer(model, k=(1,12)).fit(x)
visualizer.show()
```



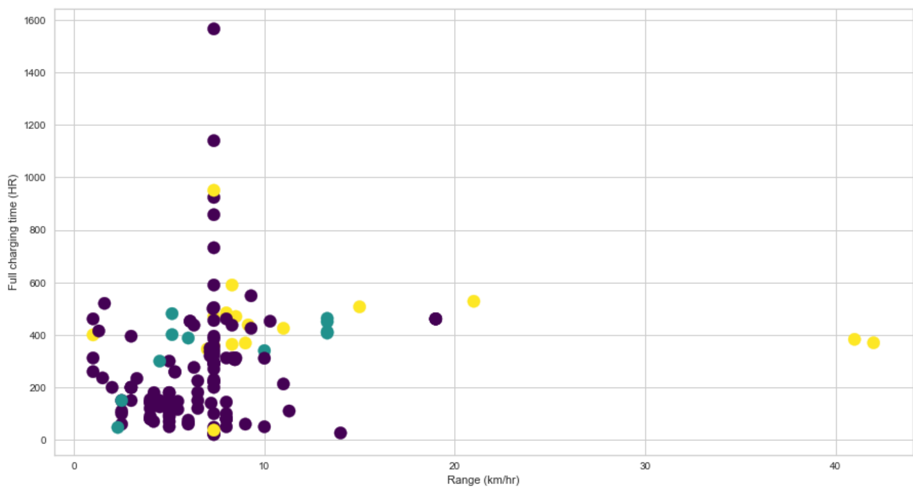

```
<AxesSubplot:title={'center':'Distortion Score Elbow for KMeans Clustering'}, xlabel='k', y
```

Based on the elbow curve, we assume the number of clusters to be optimally around 3. In clustering, this means one should choose a few clusters so that adding another cluster doesn't give much better modeling of the data. The intuition is that increasing the number of clusters will naturally improve the fit (explain more of the variation), since there are more parameters (more clusters) to use, but that at some point this is over-fitting, and the elbow reflects this.

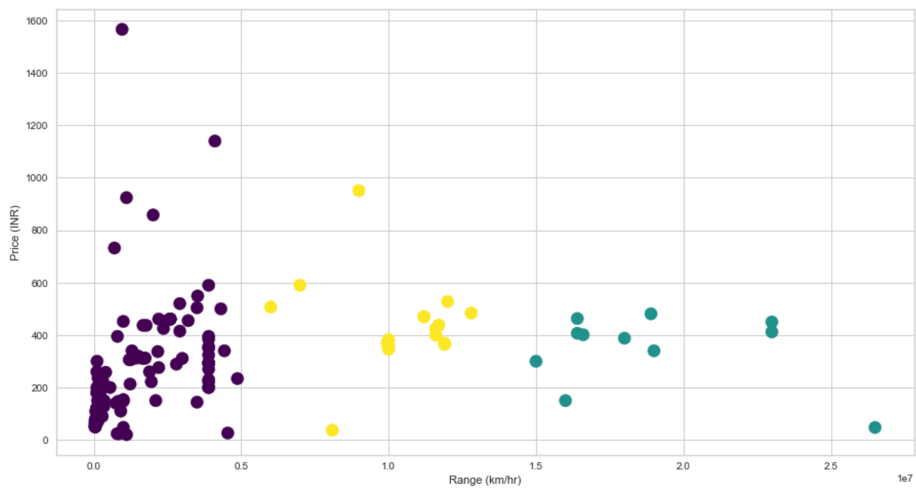
```
data['Range (km/hr)'].shape
```

```
(143,)
```

```
#create model
kmeans = KMeans(n_clusters=3)
data_predict = kmeans.fit_predict(new_data)
data_predict.shape
plt.figure(figsize=(15,8))
plt.scatter( y ='Range (km/hr)' ,x = 'Full charging time (HR)', data = data , c = data_predi
plt.xlabel('Range (km/hr)')
plt.ylabel('Full charging time (HR)')
plt.show()
```




```
plt.figure(figsize=(15,8))
plt.scatter( y ='Range (km/hr)' ,x = 'Price (INR)', data = data , c = data_predict , s =150,
plt.xlabel('Range (km/hr)')
plt.ylabel('Price (INR)')
plt.show()
```




```
#K-means clustering

kmeans = KMeans(n_clusters=3, init='k-means++', random_state=0).fit(x)
data['cluster_num'] = kmeans.labels_ #adding to df
print (kmeans.labels_) #Label assigned for each data point
print (kmeans.inertia_) #gives within-cluster sum of squares.
print(kmeans.n_iter_) #number of iterations that k-means algorithm runs to get a minimum wi
print(kmeans.cluster_centers_) #Location of the centroids on each cluster.
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 2 1 0 2 2 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 2 1
 2 1 2 2 2 2 2 2 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
 0 0 0 0 1 1 2 0 2 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 0 0 0 1 1 0 0]
441606597666449.4
2
[[1.10156960e+02 1.49641572e+06 6.45993556e+00 9.56521739e-02
  3.29301271e+01 2.65971311e+02 1.29536953e+03 9.65217391e-01
  1.20000000e+01 1.61739130e+00 6.38260870e+00 1.78260870e+00
  2.53453825e+02 1.58260870e+00]
 [2.01363636e+02 1.89818182e+07 8.07272727e+00 9.09090909e-02
  8.06868531e+01 3.49040909e+02 2.29964967e+03 8.18181818e-01
  1.01818182e+01 8.18181818e-01 1.02727273e+01 2.18181818e+00
  9.03636364e+02 0.00000000e+00]
 [2.16058824e+02 1.03482353e+07 1.28605672e+01 1.17647059e-01
  7.29000000e+01 4.40647059e+02 2.42052941e+03 8.23529412e-01
  9.29411765e+00 1.00000000e+00 5.17647059e+00 2.00000000e+00
  6.17529412e+02 7.05882353e-01]]
```

```
from collections import Counter
Counter(kmeans.labels_)
```

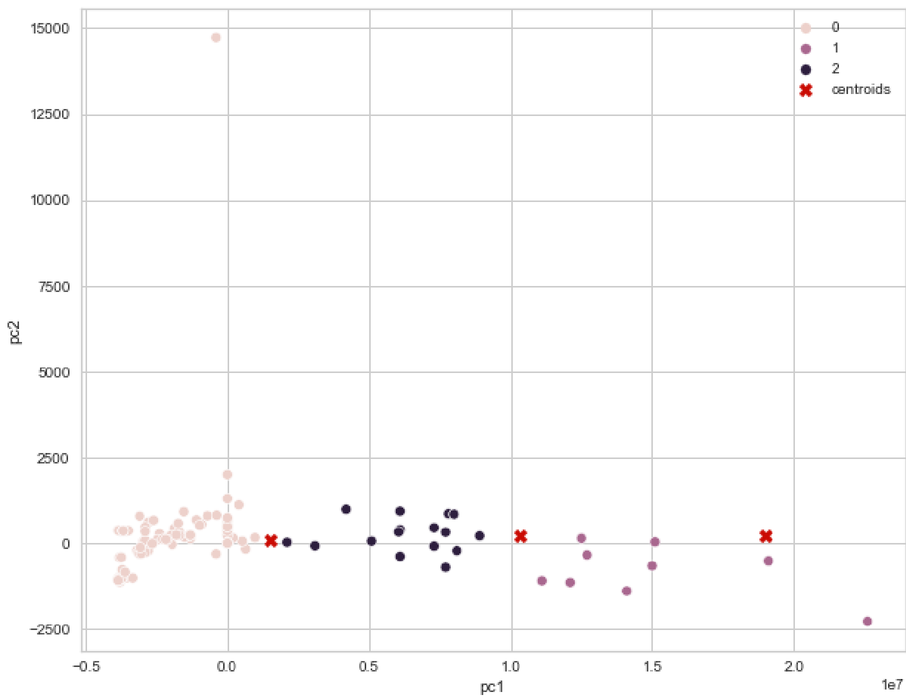
```
Counter({0: 115, 2: 17, 1: 11})
```

```
kmeans.cluster_centers_[:,1]
```

```
array([ 1496415.71722278, 18981818.18181818, 10348235.29411765])
```

```
plt.figure(figsize=(10,8))
sns.scatterplot(data=pf, x="pc1", y="pc2", hue=kmeans.labels_)
plt.scatter(kmeans.cluster_centers_[:,1], kmeans.cluster_centers_[:,0],
            marker="X", c="r", s=80, label="centroids")
plt.legend()
```

```
<matplotlib.legend.Legend at 0x1e50c307a90>
```



In the above figure we create 3 clusters by using K-Means Clustering and visualize for better understanding with Centroids.

```
data['Fuel Type']
```

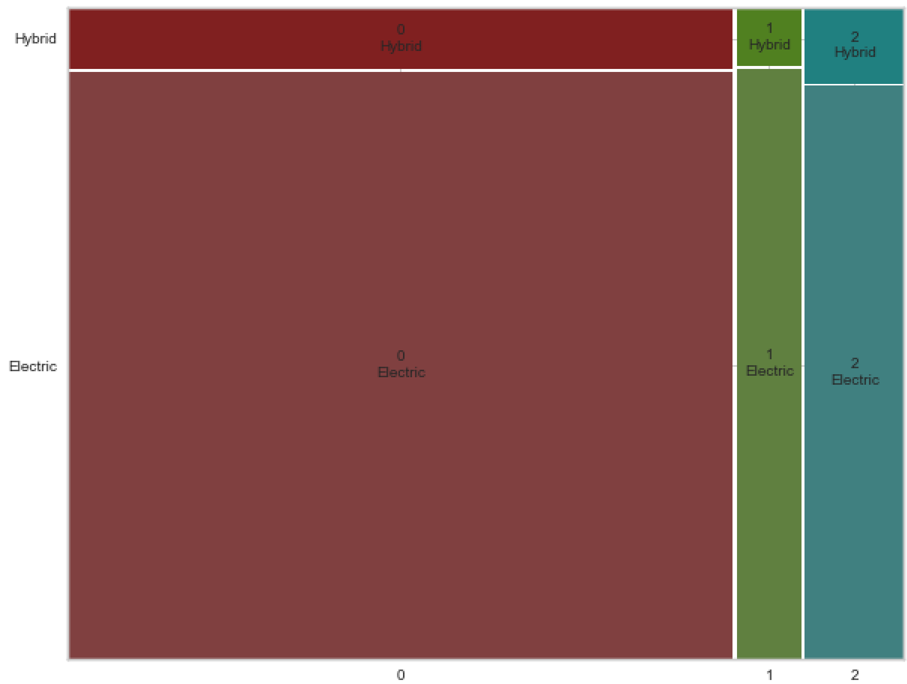
```
0      Electric
1      Electric
2      Electric
3      Electric
4      Electric
         ...
138    Electric
```

```
139    Electric
140    Electric
141    Electric
142    Electric
Name: Fuel Type, Length: 143, dtype: object
```

```
#DESCRIBING SEGMENTS
from statsmodels.graphics.mosaicplot import mosaic
from itertools import product

crosstab =pd.crosstab(data['cluster_num'],data['Fuel Type'])
#Reordering cols
crosstab1 = crosstab[['Electric','Hybrid']]
crosstab1
```

```
Fuel Type    Electric  Hybrid
cluster_num
0                 104      11
1                  10       1
2                  15       2
```

```
#MOSAIC PLOT
plt.rcParams['figure.figsize'] = (10,8)
mosaic(crosstab1.stack())
plt.show()
```

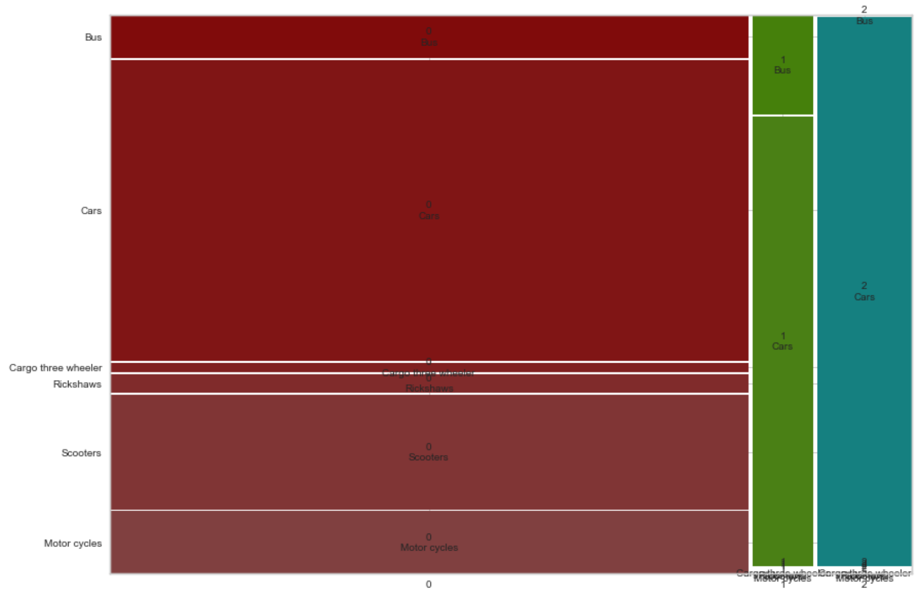



```
#DESCRIBING SEGMENTS
from statsmodels.graphics.mosaicplot import mosaic
from itertools import product

crosstab =pd.crosstab(data['cluster_num'],data['Type of Vehicle'])
#Reordering cols
crosstab2 = crosstab[['Motor cycles', 'Scooters', 'Rickshaws', 'Cargo three wheeler',
       'Cars', 'Bus']]
crosstab2
```

```
Type of Vehicle  Motor cycles  Scooters  Rickshaws  Cargo three wheeler  Cars  \
cluster_num
0                          13        24          4                    2    63
1                           0         0          0                    0     9
2                           0         0          0                    0    17

Type of Vehicle  Bus
cluster_num
0                  9
1                  2
2                  0
```

```
#MOSAIC PLOT
plt.rcParams['figure.figsize'] = (14,10)
```

```
mosaic(crosstab2.stack())
plt.show()
```




```
# DESCRIBING SEGMENTS
from statsmodels.graphics.mosaicplot import mosaic
from itertools import product

crosstab =pd.crosstab(data['cluster_num'],data[' Number of Seats'])
#Reordering cols
crosstab3 = crosstab[[ 2,  4,  6,  5,  7, 44, 30, 31, 40, 35, 39, 43]]
crosstab3
```

Number of Seats	2	4	6	5	7	44	30	31	40	35	39	43
cluster_num												
0	39	9	2	54	2	1	1	2	1	3	0	1
1	0	2	0	7	0	0	0	1	0	0	1	0
2	0	1	0	14	2	0	0	0	0	0	0	0

```
# DESCRIBING SEGMENTS
from statsmodels.graphics.mosaicplot import mosaic
from itertools import product

crosstab =pd.crosstab(data['cluster_num'],data['Manufacturing'])
#Reordering cols
crosstab4 = crosstab[['Revolt Motors', 'Tork Motors', 'Kabira Mobility',
       'Kabira Mobility KM 4000', 'SVM Prana', 'Earth Energy ',
       ' Earth Energy', 'Ultraviolette Automotive', 'Emflux Motors',
```

```
       'Ather Energy', 'Bajaj ', 'Simple Energy', 'Hero Electric',
       'Okinawa Praise', 'Yakuza Rubie', 'Lactrix Motors', 'Evolet Pony',
       'Omjay Eeve', 'Battre loev', 'BattRE Electric', 'PURE EV ',
       'Ampere', 'Ola', 'TVS', 'Amo Mobility', 'Lectrix EV',
       'Entice Impex', 'Lohia', 'Mahindra ', 'Kerala Automobiles',
       'Omega Seiki Mobility', 'Ele ', 'Tata', 'MG ZS', 'Hyundai',
       'Jaguar', 'Audi ', 'E6', 'Mercedes-Benz', 'BMW ', 'Mahindra',
       'Mercedes Benz', 'Pravaig Dynamics', 'MG', 'Toyota', 'Honda',
       'MG ', 'Maruti Suzuki', 'Maruti Suzuki ', 'Toyota ', 'Volvo',
       'BMW', 'Audi', 'Citroen', 'Kia', 'MIni', 'Nissan', 'Opel',
       'Peugeot', 'Porsche', 'Renault', 'Skoda', 'Smart', 'Volkswagen',
       'Citroën', 'BYD', 'Tesla', 'Ashok Leyland', 'JBM Auto Limited\xa0',
       'Tata Motors', 'Olectra Greentech Limited\xa0',
       'Deccan Auto Limited\xa0\xa0', 'Eicher Motors Limited\xa0']]
crosstab4
```

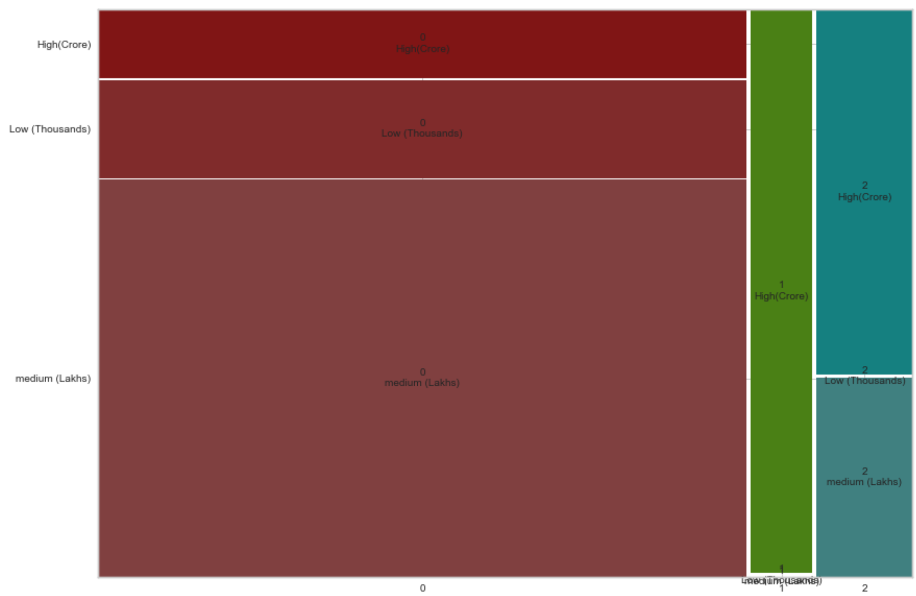
```
Manufacturing  Revolt Motors  Tork Motors  Kabira Mobility  \
cluster_num
0                          2            2                2
1                          0            0                0
2                          0            0                0

Manufacturing  Kabira Mobility KM 4000  SVM Prana  Earth Energy   \
cluster_num
0                                    1          2              1
1                                    0          0              0
2                                    0          0              0

Manufacturing   Earth Energy  Ultraviolette Automotive  Emflux Motors  \
cluster_num
0                          1                         1              1
1                          0                         0              0
2                          0                         0              0

Manufacturing  Ather Energy  ...  Volkswagen  Citroën  BYD  Tesla  \
cluster_num                  ...
0                         2  ...           4        1    1      0
1                         0  ...           0        0    0      0
2                         0  ...           0        0    0      1

Manufacturing  Ashok Leyland  JBM Auto Limited   Tata Motors  \
cluster_num
0                          1                  1            5
1                          0                  0            1
2                          0                  0            0
```

```
Manufacturing  Olectra Greentech Limited   Deccan Auto Limited    \
cluster_num
0                                      0                     1
1                                      1                     0
2                                      0                     0

Manufacturing  Eicher Motors Limited
cluster_num
0                                  1
1                                  0
2                                  0

[3 rows x 73 columns]
```

```
# DESCRIBING SEGMENTS
from statsmodels.graphics.mosaicplot import mosaic
from itertools import product

crosstab =pd.crosstab(data['cluster_num'],data['Income'])
#Reordering cols
crosstab5 = crosstab[['medium (Lakhs)', 'Low (Thousands)', 'High(Crore)']]
crosstab5
```

```
Income       medium (Lakhs)  Low (Thousands)  High(Crore)
cluster_num
0                        81               20           14
1                         0                0           11
2                         6                0           11
```

```
# MOSAIC PLOT
plt.rcParams['figure.figsize'] = (14,10)
mosaic(crosstab5.stack())
plt.show()
```




```
# Calculating the mean
# Fuel Type
data['Fuel Type']= LabelEncoder().fit_transform(data['Fuel Type'])
Fuel_Type = data.groupby('cluster_num')['Fuel Type'].mean()
Fuel_Type = Fuel_Type.to_frame().reset_index()
Fuel_Type
```

```
   cluster_num  Fuel Type
0            0   0.095652
1            1   0.090909
2            2   0.117647
```

```
# Calculating the mean
# Type_of_Vehicle
data['Type of Vehicle']= LabelEncoder().fit_transform(data['Type of Vehicle'])
Type_of_Vehicle = data.groupby('cluster_num')['Type of Vehicle'].mean()
Type_of_Vehicle = Type_of_Vehicle.to_frame().reset_index()
Type_of_Vehicle
```

```
   cluster_num  Type of Vehicle
0            0         2.634783
1            1         1.636364
2            2         2.000000
```

```
# Calculating the mean
# Number_of_Seats
data[' Number of Seats']= LabelEncoder().fit_transform(data[' Number of Seats'])
Number_of_Seats= data.groupby('cluster_num')[' Number of Seats'].mean()
```

```
Number_of_Seats = Number_of_Seats.to_frame().reset_index()
Number_of_Seats
```

```
   cluster_num   Number of Seats
0           0          1.730435
1           1          2.727273
2           2          2.176471
```

```
# Calculating the mean
# Income
data['Income']= LabelEncoder().fit_transform(data['Income'])
Income= data.groupby('cluster_num')['Income'].mean()
Income = Income.to_frame().reset_index()
Income
```

```
   cluster_num    Income
0           0  1.582609
1           1  0.000000
2           2  0.705882
```

```
data['Full charging time (HR)']
```

```
0      4.500000
1      4.200000
2      5.000000
3      5.000000
4      2.000000
         ...
138    3.000000
139    2.500000
140    4.500000
141    7.344911
142    7.344911
Name: Full charging time (HR), Length: 143, dtype: float64
```

```
# Calculating the mean
# Full_charging_time_(HR)
# data['Full charging time (HR)']= LabelEncoder().fit_transform(data['Full charging time (H
Full_charging_time=data.groupby('cluster_num')['Full charging time (HR)'].mean()
Full_charging_time=Full_charging_time.to_frame().reset_index()

Full_charging_time
```

```
   cluster_num  Full charging time (HR)
0           0                 6.459936
1           1                 8.072727
2           2                12.860567
```

```
# Calculating the mean
# Full_charging_time_(HR)
```

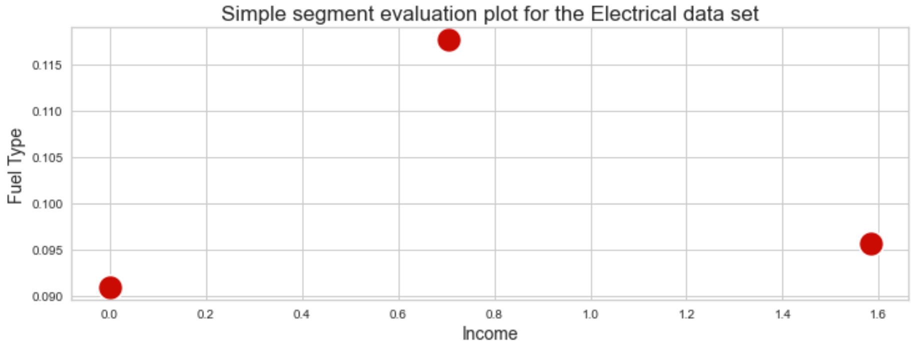
```
# data['Full charging time (HR)']= LabelEncoder().fit_transform(data['Full charging time (HI
ranges=data.groupby('cluster_num')['Range (km/hr)'].mean()
ranges=ranges.to_frame().reset_index()

ranges
```

```
   cluster_num  Range (km/hr)
0            0     265.971311
1            1     349.040909
2            2     440.647059
```

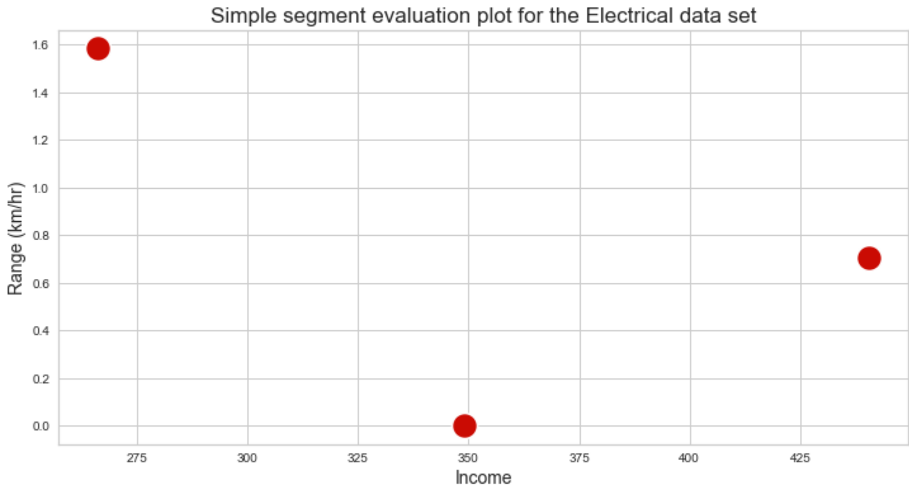
```
# Segment
segment = Income.merge(Type_of_Vehicle, on='cluster_num', how='left').merge(Fuel_Type, on='c
how='left').merge(ranges,on='cluster_num', how='left').merge(Full_charging_time, on='cluster
segment
```

```
   cluster_num    Income  Type of Vehicle  Fuel Type  Range (km/hr)  \
0            0  1.582609         2.634783   0.095652     265.971311
1            1  0.000000         1.636364   0.090909     349.040909
2            2  0.705882         2.000000   0.117647     440.647059

   Full charging time (HR)
0                 6.459936
1                 8.072727
2                12.860567
```

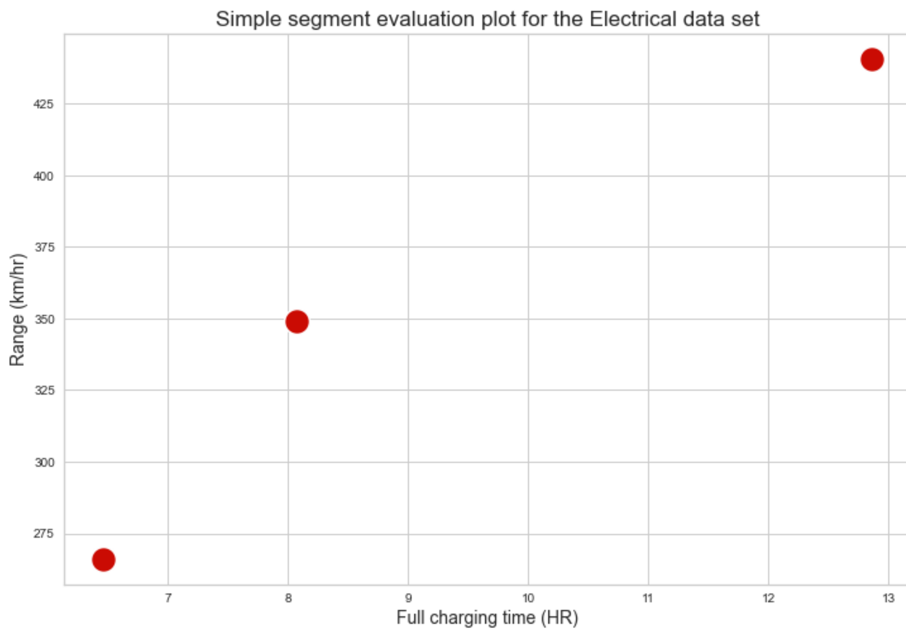
```
# Target segments

plt.figure(figsize = (12,4))
sns.scatterplot(x = "Income", y = "Fuel Type",data=segment,s=400, color="r")
plt.title("Simple segment evaluation plot for the Electrical data set",
          fontsize = 17)
plt.xlabel("Income", fontsize = 14)
plt.ylabel("Fuel Type", fontsize = 14)
plt.show()
```



```python
# Target segments

plt.figure(figsize = (12,6))
sns.scatterplot(x = "Range (km/hr)", y = "Income",data=segment,s=400, color="r")
plt.title("Simple segment evaluation plot for the Electrical data set",
          fontsize = 17)
plt.xlabel("Income", fontsize = 14)
plt.ylabel("Range (km/hr)", fontsize = 14)
plt.show()
```




```python
# Target segments
plt.figure(figsize = (12,8))
sns.scatterplot(x ='Full charging time (HR)' , y ="Range (km/hr)",data=segment,s=400, color=
plt.title("Simple segment evaluation plot for the Electrical data set",
          fontsize = 17)
plt.xlabel("Full charging time (HR)", fontsize = 14)
plt.ylabel("Range (km/hr)", fontsize = 14)
plt.show()
```



Analysing Market Segments

There are several different variables by which segmentation is done:

1. Geographic segmentation

Geographic segmentation consists of creating different groups of customers based on geographic boundaries. The needs and interests of potential customers vary according to their geographic location, climate and region, and understanding this allows you to determine where to sell and advertise a brand, as well as where to expand a business.

- Charging station by State wise: State wise charging station will become a significant effect on consumer purchasing decisions. Those states with more charging stations may prefer to buy an EV and vice versa.

1. Demographic segmentation

Demographic segmentation consists of dividing the market through different variables such as age, gender, nationality, education level, family size, occupation, income, etc. This is one of the most widely used forms of market segmentation, since it is based on knowing how customers use your products and services and how much they are willing to pay for them.

- Income: Income levels have a significant effect on consumer purchasing decisions. Those with higher-income levels may prefer luxury vehicles.

Conversely, individuals with lower income levels may prefer to get vehicles at the best deal and are likely to choose inexpensive products/services.

- Family size: Family size also determines consumers' purchase decisions. Those who have large family members may choose four wheelers and those who have less family members will choose two wheelers.

1. Psychographic segmentation

Psychographic segmentation consists of grouping the target audience based on their behavior, lifestyle, attitudes and interests. To understand the target audience, market research methods such as focus groups, surveys, interviews and case studies can be successful in compiling this type of conclusion.

- Lifestyle: A consumer whose profession is more time consuming than other average consumers , that consumer may select a vehicle who takes less time to charge a vehicle. This group of consumers only focus on the time required to charge an EV.

- Interests : Some consumers may have interest in particular manufacturing companies. Some consumers may like only vehicles made by the Tata company.

- Behavior : Behavior of consumers is the most important factor in the market segment. It shows what exactly consumers want from us?. Some consumers may want an EV who will cover far distance per a charging. Customizing the Market Mix The marketing mix refers to the set of actions, or tactics, that a company uses to promote its brand or product in the market.

The 4Ps make up a typicalmarketing mix - Price, Product, Promotion and Place.

- Price: Refers to the value that is put for a product. It depends on costs of production, segment targeted, ability of the market to pay, supply - demand and a host of other direct and indirect factors. There can be several types of pricing strategies, each tied in with an overall business plan.

- Product: Refers to the item actually being sold. The product must deliver a minimum level of performance; otherwise even the best work on the other elements of the marketing mix won't do any good.

- Place: Refers to the point of sale. In every industry, catching the eye of the consumer and making it easy for her to buy it is the main aim of a good distribution or 'place' strategy. Retailers pay a premium for the right location. In fact, the mantra of a successful retail business is 'location, location, location’.

- Promotion: This refers to all the activities undertaken to make the product or service known to the user and trade. This can include advertising, word of mouth, press reports, incentives, commissions and awards to the trade. It can also include consumer schemes, direct marketing, contests and prizes.

All the elements of the marketing mix influence each other. They make up the business plan for a company and handle it right, and can give it great success. The marketing mix needs a lot of understanding, market research and consultation with several people, from users to trade to manufacturing and several others.

Target Segment

Target marketing involves breaking a market into segments and then concentrating your marketing efforts on one or a few key segments consisting of the customers whose needs and desires most closely match your product or service offerings. It can be the key to attracting new business, increasing sales, and making your business a success.

It can be concluded from above figures that Range, Top Speed, Full charging time, Income and Types of Vehicles can be the most important segment categories for consumer purchasing decisions. These are the key factors who make markets different and similar at the same time. This segments have formed with distinct features which may indicate that their preferences for EVs are motivated by different factors.

Recommendations and Learnings

The penetration of EV in India has Increased Significantly in the last five years as they are more efficient. In addition, growing fuel prices are further helping to boost substantial growth in the product adoption, mainly due to their extended range and efficiency.

The global Electric Vehicle Market size is projected to grow from 8,151 thousand units in 2022 to 39,208 thousand units by 2030, at a CAGR of 21.7%. Factors such as growing demand for low emission commuting and governments supporting long range, zero emission vehicles through subsidies & tax rebates have compelled the manufacturers to provide electric vehicles around the world.

Increasing investments by governments across the globe to develop EV charging stations and Hydrogen fueling stations along with incentives offered to buyers will create opportunities for OEMs to expand their revenue stream and geographical presence.

From this analysis we create different types of segments to affect consumers' purchasing decisions. Geographic segmentation is about places, cities, states that where consumers live will affect market sales. Like if a consumer lives in a rural area there may be less possibility of having charging stations and vice versa in urban areas. Now in 2022 yet we have only 1742 public charging stations available.

So if a consumer is from those states who have more available charging stations ,the probability of buying is more as compared to others who have less charging stations in their states. Demographic segmentation focuses on education level, family size, occupation, income, etc. since it is based on knowing how customers use your products and services and how much they are willing to pay for them.

That depends on consumers' education, Financial status and purpose of buying EV's. If a customer's purpose is to buy an EV for transporting goods in different cities or states, that customer will focus on the boot space and maximum range of a vehicle. On a psychological segment some customers may go for a product which gives them satisfaction and others may go with a product who is cheaper in cost and their other factors are average.