FEYNN Labs: Project 2

Market Segment Analysis Of EV Startup in India

Contributors

1. S.V.Gautham
2. Sweta Patil
3. Saad Bebal
4. Punya S

Github Link

1. https://github.com/SVGautham/EV_Market_Analysis
2. https://github.com/SwetaPatil555/EV_Market_Analysis
3. https://github.com/SaadBebal/EV_Stations_India
4. https://github.com/Punya2011/FL--Project2-EV_Market_Segment.git

Problem Statement

The problem statement focuses on determining the most suitable strategy for an Electric Vehicle (EV) startup in India by analyzing market segments and identifying the optimal entry point. The EV startup needs to conduct a segmentation analysis of the Indian EV market to identify key customer or vehicle segments. This involves evaluating geographic, demographic, psychographic, behavioral, and other relevant market factors based on data availability. The analysis will focus on finding the most appropriate location for early market adoption in line with the **Innovation Adoption Life Cycle**, understanding consumer needs, and estimating market demand.

The team must overcome challenges related to data availability and quality, including researching and collecting data on EV usage, vehicle categories, charging stations, and vehicle statistics. The strategy will incorporate findings from data analysis, identify key market segments, and recommend a pricing strategy suitable for early adopters. Additionally, the report will profile potential segments, perform market forecasting, and estimate the potential profit based on the chosen segment and price range. The deliverables include a detailed analysis report, data sources, segment extraction using machine learning techniques, and a strategic plan for entering the market, along with the technical documentation on GitHub.

Datasets used

By S.V.Gautham

- EV_INDIA_CARS.csv
- Vehicle_EV_DATAset_Count.csv
- EV_Count_State.csv
- Charging_st_count.csv

By Sweta Patil

- EV_model_spec.csv
- SMEV_data.csv

By Saad Bebal

- EV-charging-stations-india.csv
- EV_india.csv

By Punya S

- car details v4.csv
- Electric_Vehicle_Population_Data.csv

Data Collection

Dataset 1 - EV_INDIA_CARS.csv
Car_name - Name of the car model with brand (OBJECT)
Car_price - price of the car in lakhs (FLOAT)
Batter_cap - capacity of battery in kWh (OBJECT)
Drive_range - full charge range of car in km (FLOAT)
Power - power of car in Bhp (FLOAT)
Charge_time - time to charge the car in hours (FLOAT)
Transmission - in liters and automatic (OBJECT)
Boot_space - automatic, nil (FLOAT)
Top_speed - nil and 200kmph (FLOAT)

Dataset 2 - Vehicle_EV_DATAset_Count.csv
Sl no - serial number
Vehicle category - 2 or 3 or 4 wheeler
No of electric vehicle - count of each type of wheeler

Dataset 3 -EV_Count_State
State - All States and UT of India
No of EV sanctioned - Count of EV vehicles

Dataset 4 -Charging_st_count
Category - type
City/Highway - Name of city or highway
Charging stations - count of stations

Dataset 5 -SMEV_data.csv
Vehicle category -Electric 4-Wheeler, Electric 3-Wheeler, Electric 2-Wheeler.
No of electric vehicle - count of each type of wheeler
EV Industries
EV Market

Dataset 6 -EV_model_spec.csv
Model name
Prices
Riding Range
Battery Charging

Dataset 7- EV-charging-stations-india.csv
Charging Stations
State
City
Address
Latitude
Longitude
category- type

Dataset 8 - EV_india.csv
State name
Total electric vehicle
EV adoption level

Dataset 9 - car details v4.csv
Vehicle Make
Model
Price
Year
Transmission
fuel type

Dataset 10 -Electric_Vehicle_Population_Data.csv
Vehicle range
Vehicle make
Range
Year
City

EDA Dataset 1

https://github.com/SVGautham/EV_Market_Analysis/blob/main/Jupyter%20Notebook%20Files/EV_CARS_INDIA_ANALYSIS%20(1).ipynb

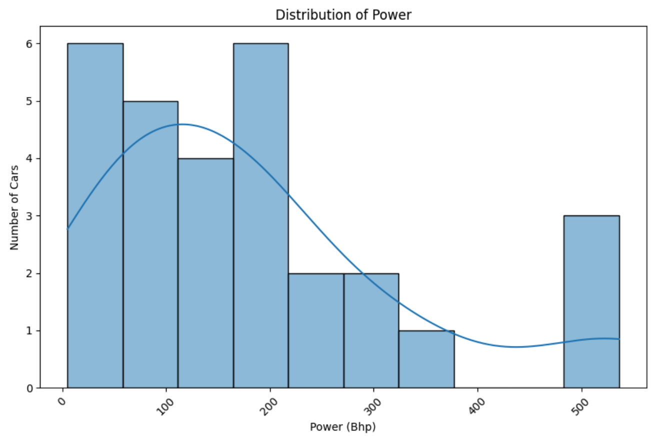
```
# Basic information
df.info()

# Summary statistics for numerical columns
df.describe()
```

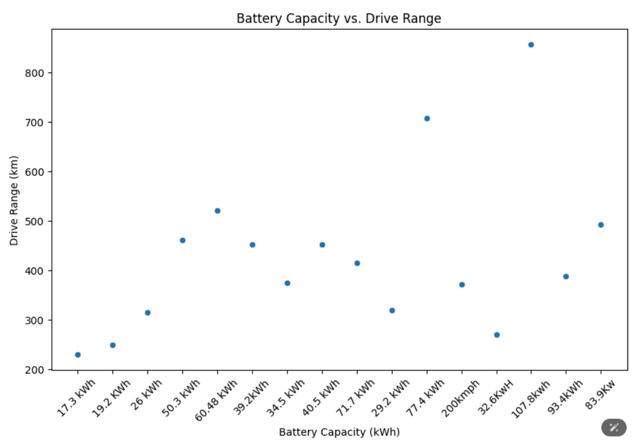
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Car_name      30 non-null     object
 1   Car_price     30 non-null     object
 2   Batter_cap    27 non-null     object
 3   Drive_range   30 non-null     object
 4   Power         30 non-null     object
 5   Charge_time   30 non-null     object
 6   transmission  28 non-null     object
 7   Boot_space    12 non-null     object
 8   Top_speed     6 non-null      object
dtypes: object(9)
```

	Car_name	Car_price	Batter_cap	Drive_range	Power	Charge_time	transmission	Boot_space	Top_speed
count	30	30	27	30	30	30	28	12	6
unique	30	29	16	18	18	14	12	2	4
top	MG Comet EV	1.9 cr	17.3 kWh	230 km/full charge	41.42 Bhp	Automatic	Automatic	Automatic	200kmph
freq	1	2	2	2	2	7	12	10	3

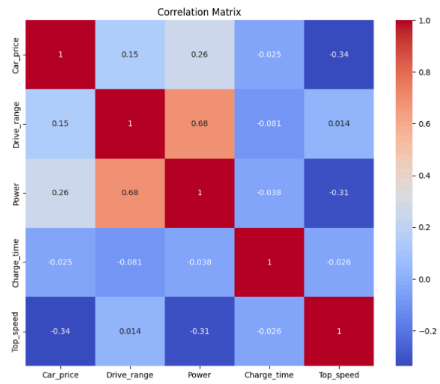
```python
# Checking for missing values
df.isnull().sum()
```

```
Car_name         0
Car_price        0
Batter_cap       3
Drive_range      0
Power            0
Charge_time      0
transmission     2
Boot_space      18
Top_speed       24
dtype: int64
```





```
In [26]: # Compute correlation matrix
corr = df[['Car_price', 'Drive_range', 'Power', 'Charge_time', 'Top_speed']].corr()

# Plot heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```

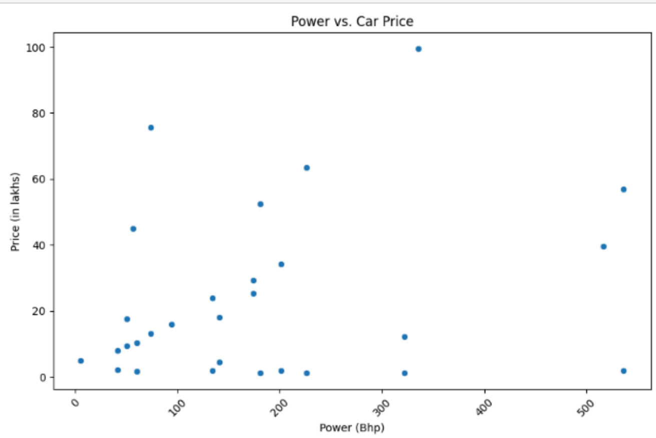


The matrix provides a numerical value between -1 and 1 for each pair of variables:

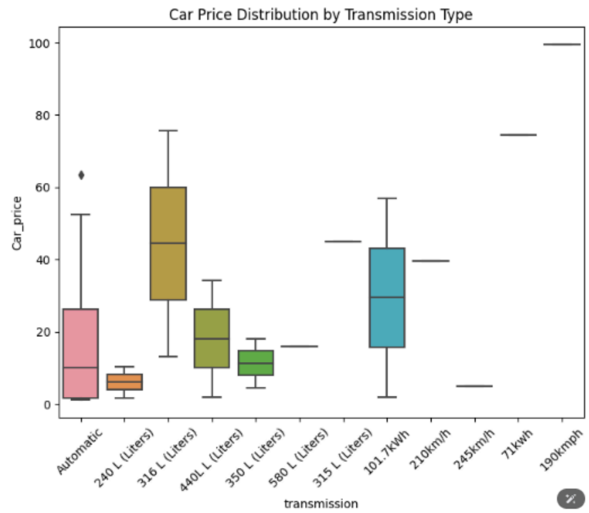
- 1 indicates a perfect positive correlation (as one variable increases, the other also increases).
- -1 indicates a perfect negative correlation (as one variable increases, the other decreases).
- 0 indicates no correlation.

A high positive correlation between variables like Power and Top_speed would suggest that cars with higher power generally have higher top speeds. A negative correlation between Car_price and Charge_time would suggest that more expensive cars tend to have shorter charging times.

```
plt.figure(figsize=(10, 6))
df_sorted2 = df.sort_values(by='Car_price', ascending=True)
bx=sns.scatterplot(x='Power', y='Car_price', data=df_sorted2)
plt.xticks(rotation=45)
plt.title('Power vs. Car Price')
plt.xlabel('Power (Bhp)')
plt.ylabel('Price (in lakhs)')

plt.show()
```

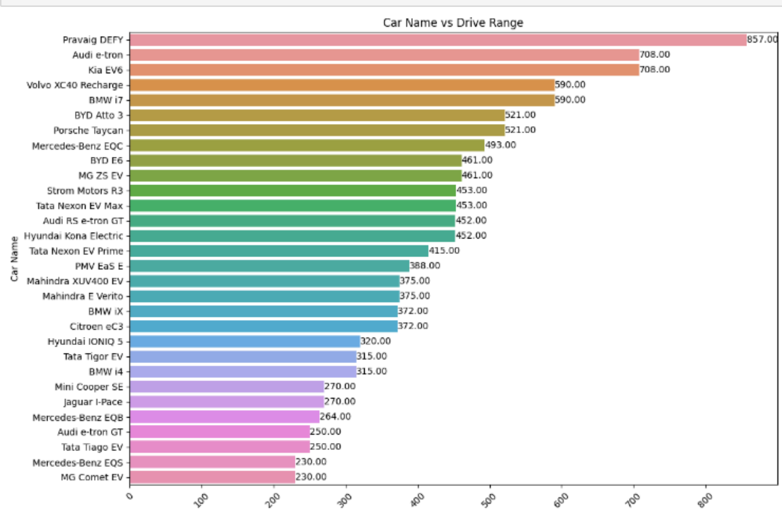


```python
# Boxplot for Car Price by Transmission
plt.figure(figsize=(8, 6))
plt.xticks(rotation=45)  # Adjust angle as needed

sns.boxplot(x='transmission', y='Car_price', data=df)
plt.title('Car Price Distribution by Transmission Type')
plt.show()
```



The median price for automatic cars seems to be around the mid-range, and there is a large spread in the price values, indicating variability. There is one noticeable outlier, where a car with automatic transmission has a much lower price.
316L has a higher median price than **240L**, with slightly more spread.
The prices vary significantly across different transmission or fuel capacity types. Vehicles with larger capacities or higher performance metrics (such as electric vehicles with more kWh or higher speeds) tend to have higher price ranges.

```python
# Sorting the data by Drive_range for better visualization
df_sorted = df.sort_values(by='Drive_range', ascending=False)

# Bar plot for Car Name vs Drive Range
plt.figure(figsize=(12, 8))
bx=sns.barplot(x='Drive_range', y='Car_name', data=df_sorted)
plt.title('Car Name vs Drive Range')
plt.xlabel('Drive Range (km/full charge)')
plt.ylabel('Car Name')
plt.xticks(rotation=45)
plt.tight_layout()
for index, value in enumerate(df_sorted['Drive_range']):
    bx.text(value, index, f'{value:.2f}', color='black', va="center")
plt.show()
```



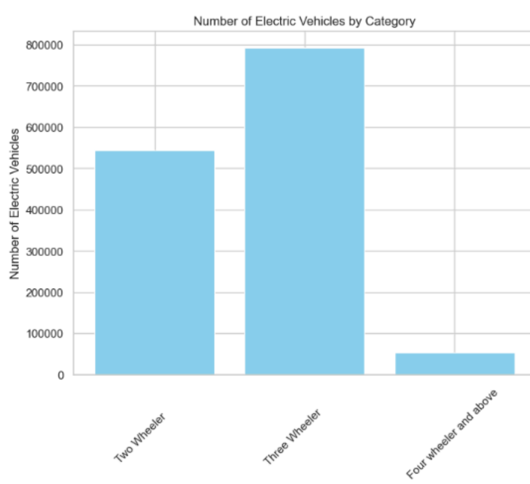
Pravaig DEFY has the highest drive range at 857 km, standing out from the other vehicles.
Luxury EVs like those from Audi, BMW, and Mercedes-Benz generally have longer ranges, reflecting their premium status and advanced battery technologies.
Mid-range EVs from Tata Motors, MG, and BYD provide a balance between affordability and performance, offering ranges suitable for daily commutes and short trips.
Low-range EVs, such as the MG Comet EV and Mercedes-Benz EQS, are more likely to be urban vehicles, suitable for shorter daily travel.

EDA Dataset 2

https://github.com/SVGautham/EV_Market_Analysis/blob/main/Jupyter%20Notebook%20Files/Vehicle_EV_DATASet_Analysis.ipynb

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load the CSV dataset
file_path = 'Vehicle_EV_DATAset_Count.csv'
df = pd.read_csv(file_path)

# Strip whitespace from column names
df.columns = df.columns.str.strip()

# Data Cleaning: Remove the 'Grand Total' row if present
df_cleaned = df[df['Vehicle Category'] != 'Grand Total']

# Basic Stats
print("Basic Statistics of the dataset:")
print(df_cleaned.describe())

# Check for missing values
print("Missing Values per Column:")
print(df_cleaned.isnull().sum())

# Bar plot of Electric Vehicles by Category
plt.figure(figsize=(8,6))
plt.bar(df_cleaned['Vehicle Category'], df_cleaned['No. of electric vehicles'], color='skyblue')
plt.xlabel('Vehicle Category')
plt.ylabel('Number of Electric Vehicles')
plt.title('Number of Electric Vehicles by Category')
plt.xticks(rotation=45)
plt.show()
```

This shows us that the count of the 3 wheelers is most in the category and is the largest segment.




```
Largest Vehicle Segment:

  Sl. No.        Vehicle Category  No. of electric vehicles
1       2  Three Wheeler                             793370
```

```python
# Segment Extraction: Determine the largest segment
largest_segment = df_cleaned[df_cleaned['No. of electric vehicles'] == df_cleaned['No. of electric vehicles'].max()]
print("Largest Vehicle Segment:")
print(largest_segment)
```

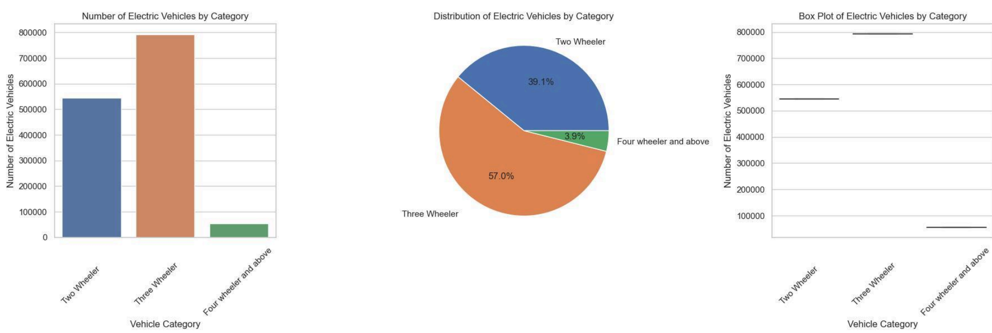
```
Basic Statistics of the dataset:

       No. of electric vehicles
count                  3.000000
mean              464088.333333
std               376085.954567
min                54252.000000
25%               299447.500000
50%               544643.000000
75%               669006.500000
max               793370.000000

Missing Values per Column:

Sl. No.                     0
Vehicle Category            0
No. of electric vehicles    0
dtype: int64
```

In [166]:

```python
# Pie Chart for Distribution of Electric Vehicles by Category
plt.figure(figsize=(8, 6))
plt.pie(df_cleaned['No. of electric vehicles'], labels=df_cleaned['Vehicle Category'], autopct='%1.1f%%', startangle=140)
plt.title('Distribution of Electric Vehicles by Category')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()

# Horizontal Bar Plot for Electric Vehicles by Category
plt.figure(figsize=(8, 6))
plt.barh(df_cleaned['Vehicle Category'], df_cleaned['No. of electric vehicles'], color='lightgreen')
plt.xlabel('Number of Electric Vehicles')
plt.title('Number of Electric Vehicles by Category')
plt.show()

# Box Plot for No. of Electric Vehicles
plt.figure(figsize=(8, 6))
plt.boxplot(df_cleaned['No. of electric vehicles'], vert=False)
plt.xlabel('Number of Electric Vehicles')
plt.title('Box Plot of Electric Vehicles')
plt.show()

# Correlation Matrix (though we only have one feature, this is more relevant with larger datasets)
import seaborn as sns

# Create a correlation matrix (if applicable)
correlation_matrix = df_cleaned[['No. of electric vehicles']].corr()

# Heatmap of the correlation matrix
plt.figure(figsize=(6, 4))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', center=0)
plt.title('Correlation Matrix')
plt.show()
```

In [167]:

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

# Set the style for seaborn
sns.set(style="whitegrid")

# Create a 1x3 subplot layout (or adjust as needed)
fig, axs = plt.subplots(1, 3, figsize=(18, 6))

# Bar plot for the number of electric vehicles per category
sns.barplot(x='Vehicle Category', y='No. of electric vehicles', data=df_cleaned, ax=axs[0])
axs[0].set_title('Number of Electric Vehicles by Category')
axs[0].set_xlabel('Vehicle Category')
axs[0].set_ylabel('Number of Electric Vehicles')
axs[0].tick_params(axis='x', rotation=45)

# Pie chart for the distribution of electric vehicles by category
axs[1].pie(df_cleaned['No. of electric vehicles'], labels=df_cleaned['Vehicle Category'], autopct='%1.1f%%')
axs[1].set_title('Distribution of Electric Vehicles by Category')

# Box plot (if you want to show distribution)
sns.boxplot(x='Vehicle Category', y='No. of electric vehicles', data=df_cleaned, ax=axs[2])
axs[2].set_title('Box Plot of Electric Vehicles by Category')
axs[2].set_xlabel('Vehicle Category')
axs[2].set_ylabel('Number of Electric Vehicles')
axs[2].tick_params(axis='x', rotation=45)

plt.tight_layout()
plt.show()
```



This shows us that the three wheeler category has dominated everywhere in the distribution of data.

EDA Dataset 3

https://github.com/SVGautham/EV_Market_Analysis/blob/main/Jupyter%20Notebook%20Files/EV_Count_State_analysis.ipynb

```python
import pandas as pd
import numpy as np
# Load the dataset
df_ev_chargers = pd.read_csv('EV_Count_State.csv')

# Remove total row and check for missing data
df_ev_chargers = df_ev_chargers[df_ev_chargers['State/UT'] != 'Total']
df_ev_chargers.dropna(inplace=True)

# Preview the dataset
print(df_ev_chargers.head())
```

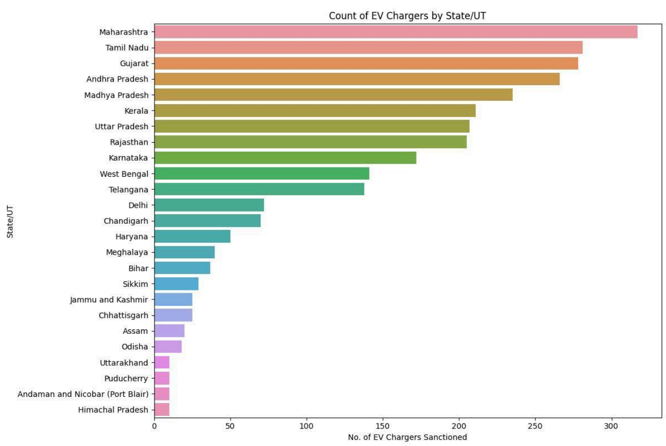
```
         State/UT  No. of EV Chargers Sanctioned
0     Maharashtra                            317
1  Andhra Pradesh                            266
2      Tamil Nadu                            281
3         Gujarat                            278
4   Uttar Pradesh                            207
```

```python
# Basic statistical summary of the dataset
print(df_ev_chargers['No. of EV Chargers Sanctioned'].describe())
```

```
count     25.000000
mean     115.080000
std      105.731941
min       10.000000
25%       25.000000
50%       70.000000
75%      207.000000
max      317.000000
Name: No. of EV Chargers Sanctioned, dtype: float64
```

This result shows us the value of head in the dataset and then calculates the basic required mathematics for the dataset.

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

plt.figure(figsize=(12, 8))
sns.barplot(x='No. of EV Chargers Sanctioned', y='State/UT', data=df_ev_chargers.sort_values(by='No. of EV Chargers Sanctioned',
plt.title('Count of EV Chargers by State/UT')
plt.xlabel('No. of EV Chargers Sanctioned')
plt.ylabel('State/UT')
plt.tight_layout()
plt.show()
```



This visualization above shows us the State-wise arrangement Count of Ev charges across the country and Maharashtra leads the Dataset followed by Tamil Nadu.

```python
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

# Scaling the data
scaler = StandardScaler()
df_ev_chargers['Chargers_Scaled'] = scaler.fit_transform(df_ev_chargers[['No. of EV Chargers Sanctioned']])

# Perform k-means clustering
kmeans = KMeans(n_clusters=3, random_state=42)
df_ev_chargers['Cluster'] = kmeans.fit_predict(df_ev_chargers[['Chargers_Scaled']])

# Plot clusters
plt.figure(figsize=(10, 6))
sns.scatterplot(x='No. of EV Chargers Sanctioned', y='State/UT', hue='Cluster', data=df_ev_chargers, palette='Set1')
plt.title('State Segments Based on EV Charger Count')
plt.xlabel('No. of EV Chargers Sanctioned')
plt.ylabel('State/UT')
plt.tight_layout()
plt.savefig("c.jpg")
plt.show()
```

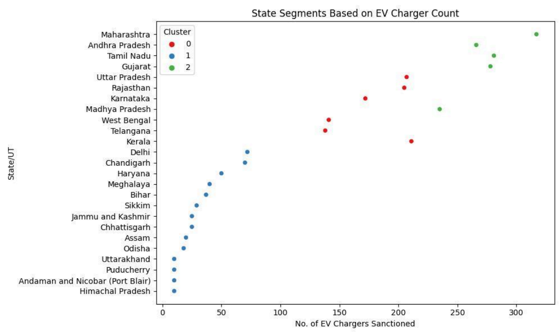
The results indicate that some states have significantly higher counts of EV chargers, clustering together, while others with lower counts form their own group. This clustering can help identify regions with similar infrastructure levels for EV charging, which is useful for targeted policy making and resource allocation.

```python
# Summarize each cluster's average charger count
segment_summary = df_ev_chargers.groupby('Cluster')['No. of EV Chargers Sanctioned'].mean()
print(segment_summary)
```

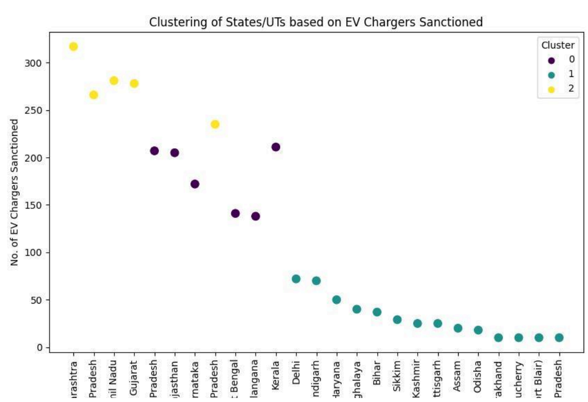
```
Cluster
0    179.000000
1     30.428571
2    275.400000
Name: No. of EV Chargers Sanctioned, dtype: float64
```

```python
sns.pairplot(df_ev_chargers)
plt.suptitle("Pairplot of EV Chargers Data", y=1.02)
plt.show()
```

Overall, the analysis shows a disparity in the average number of EV chargers sanctioned across the clusters, with Cluster 2 demonstrating the highest capacity for EV charger deployment.




```python
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

# Standardize the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df_ev_chargers[['No. of EV Chargers Sanctioned']])

# Apply KMeans clustering
kmeans = KMeans(n_clusters=3, random_state=42)
df_ev_chargers['Cluster'] = kmeans.fit_predict(scaled_data)

# Visualize the clusters
plt.figure(figsize=(10, 6))
sns.scatterplot(x='State/UT', y='No. of EV Chargers Sanctioned', hue='Cluster', palette='viridis', data=df_ev_chargers, s=100)
plt.title('Clustering of States/UTs based on EV Chargers Sanctioned')
plt.xticks(rotation=90)
plt.savefig('e.jpg')
plt.show()

# Print out the cluster centers
print("Cluster Centers (Scaled):\n", kmeans.cluster_centers_)
```



States like Maharashtra, Andhra Pradesh, Tamil Nadu fall in the Cluster 2 (Yellow), indicating they have sanctioned a significantly higher number of EV chargers.

Cluster 1 (Green) has states like Madhya Pradesh, Uttar Pradesh, Delhi, suggesting a middle ground in terms of EV charger distribution.
Cluster 0 (Purple) includes states like West Bengal, Haryana, Gujarat, which have fewer EV chargers sanctioned.

EDA Dataset 4

https://github.com/SVGautham/EV_Market_Analysis/blob/main/Jupyter%20Notebook%20Files/EV_Count_State_analysis.ipynb

```python
import pandas as pd

# Load the dataset
df_charging_stations = pd.read_csv('Charging_st_count.csv')

# Basic information about the dataset
print(df_charging_stations.info())

# Check the first few rows
df_charging_stations.head()
```

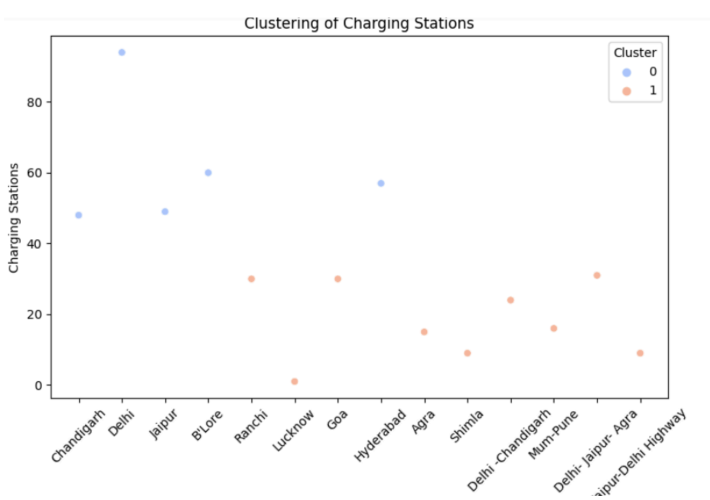
```python
# Check for missing values
print(df_charging_stations.isnull().sum())

# Clean any 'Total' rows if needed
df_cleaned = df_charging_stations[df_charging_stations['City/Highway'] != 'Total']
```

```
Category             0
City/Highway         0
Charging Stations    0
dtype: int64
```

```python
# Summary statistics for the number of charging stations
df_cleaned['Charging Stations'].describe()
```

```
count    14.000000
mean     33.785714
std      25.404962
min       1.000000
25%      15.250000
50%      30.000000
75%      48.750000
max      94.000000
Name: Charging Stations, dtype: float64
```

```python
from sklearn.cluster import KMeans

# Extract features for clustering
X = df_cleaned[['Charging Stations']].values

# Apply KMeans Clustering
kmeans = KMeans(n_clusters=2, random_state=0)
df_cleaned['Cluster'] = kmeans.fit_predict(X)

# Visualize the clusters
plt.figure(figsize=(8,6))
sns.scatterplot(x='City/Highway', y='Charging Stations', hue='Cluster', data=df_cleaned, palette='coolwarm')
plt.title('Clustering of Charging Stations')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



It seems that larger cities like Chandigarh, Delhi, and Bangalore are in Cluster 0 (Blue), indicating that they have a higher concentration of charging stations compared to other locations.
Highways such as Delhi-Jaipur and Delhi-Chandigarh Highway fall into Cluster 1 (Red), suggesting a need for increased infrastructure development in these areas.

EDA Dataset 5

https://github.com/SwetaPatil555/EV_Market_Analysis/blob/main/Jupyter%20Notebook%20Files/EVMarket_Segmentation1.ipynb

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
import nltk
from nltk.sentiment import SentimentIntensityAnalyzer
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
import warnings
```

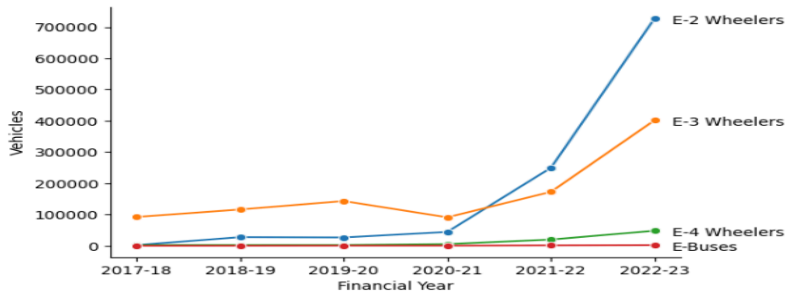
```python
warnings.filterwarnings("ignore")
pd.options.display.max_columns = None
```

```python
data_smev = pd.read_excel("smev_data.xlsx", sheet_name=None)
```

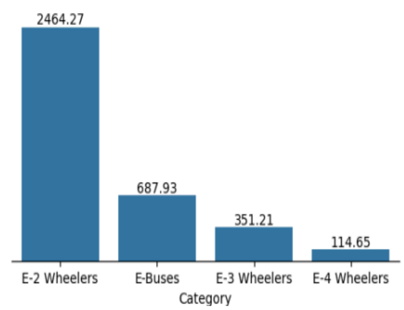
```python
data_smev['EV Industries']
```

	category	financial_year	num_vehicles
0	E-2 Wheelers	2017-18	1981
1	E-2 Wheelers	2018-19	27478
2	E-2 Wheelers	2019-20	26512
3	E-2 Wheelers	2020-21	44294
4	E-2 Wheelers	2021-22	249615
5	E-2 Wheelers	2022-23	728090
6	E-3 Wheelers	2017-18	91970
7	E-3 Wheelers	2018-19	116031
8	E-3 Wheelers	2019-20	143051
9	E-3 Wheelers	2020-21	90898
10	E-3 Wheelers	2021-22	172543
11	E-3 Wheelers	2022-23	401882
12	E-4 Wheelers	2017-18	2242
13	E-4 Wheelers	2018-19	2407
14	E-4 Wheelers	2019-20	2404
15	E-4 Wheelers	2020-21	5201
16	E-4 Wheelers	2021-22	19782
17	E-4 Wheelers	2022-23	48105
18	E-Buses	2017-18	35
19	E-Buses	2018-19	75
20	E-Buses	2019-20	369
21	E-Buses	2020-21	373
22	E-Buses	2021-22	1198
23	E-Buses	2022-23	1917

```
fig, ax = plt.subplots(figsize=(6, 4))
ax = sns.lineplot(data=data_smev['EV Industries'], x='financial_year', y='num_vehicles', hue='category', marker='o', palette='tab10')
plt.xlabel("Financial Year")
plt.ylabel("Vehicles")
plt.legend(title='Category')

# Annotate the last data point
for col in data_smev['EV Industries']['category'].unique():
    last_point = data_smev['EV Industries'][data_smev['EV Industries']['category'] == col].iloc[-1]
    plt.annotate(f"{last_point['category']}",
                 (last_point['financial_year'], last_point['num_vehicles']),
                 textcoords="offset points",
                 xytext=(10, -5),
                 ha='left')
ax.spines[['right', 'top']].set_visible(False)
ax.get_legend().set_visible(False)
plt.show()
```




```
fig, ax = plt.subplots(figsize = (6, 3))
ax = sns.barplot(data_smev['EV Market'], x = 'Category', y = 'Amount INR Cr')
ax.bar_label(ax.containers[0])
plt.yticks([])
plt.ylabel("")
ax.spines[['right', 'top', 'left']].set_visible(False)
plt.show()
```

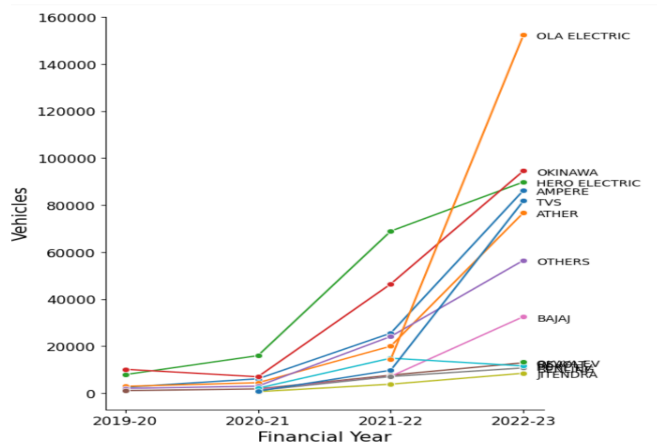


Above Figure delved into the market's financial perspective, representing the industry's total value in crores. Notably, two-wheelers emerged as the primary revenue generators, highlighting their economic significance.

```
np.sort(data_ev2w['maker'].unique())
```

```
array(['AMPERE', 'AMPERE VEHICLES', 'ATHER', 'ATHER ENERGY', 'BAJAJ',
       'BEING INDIA', 'BENLING', 'HERO ELECTRIC', 'JITENDRA',
       'JITENDRA NEW EV', 'OKAYA EV', 'OKINAWA', 'OKINAWA AUTOTECH',
       'OLA ELECTRIC', 'OTHERS', 'PURE EV', 'REVOLT', 'TVS'], dtype=object)
```

```python
fig, ax = plt.subplots(figsize=(6,8))
ax = sns.lineplot(data=data_ev2w_year, x='financial_year', y='num_vehicles', hue='maker', marker='o', palette='tab10')
plt.xlabel("Financial Year", fontsize = 15)
plt.ylabel("Vehicles", fontsize = 15)
plt.xticks(fontsize = 12)
plt.yticks(fontsize = 12)
plt.legend(title='Maker')

# Annotate the last data point
for col in data_ev2w_year['maker'].unique()[::1]:
    last_point = data_ev2w_year[data_ev2w_year['maker'] == col].iloc[-1]
    plt.annotate(f"{last_point['maker']}",
                 (last_point['financial_year'], last_point['num_vehicles']),
                 textcoords="offset points",
                 xytext=(10, -5),
                 ha='left')
ax.spines[['right', 'top']].set_visible(False)
ax.get_legend().set_visible(False)
plt.show()
```



EDA Dataset 6

EV Market Segmentation

https://github.com/SwetaPatil555/EV_Market_Analysis/blob/main/Jupyter%20Notebook%20Files/EV_Market_Segmentation2.ipynb

```python
data_model = pd.read_csv("ev_model_spec.csv")
```

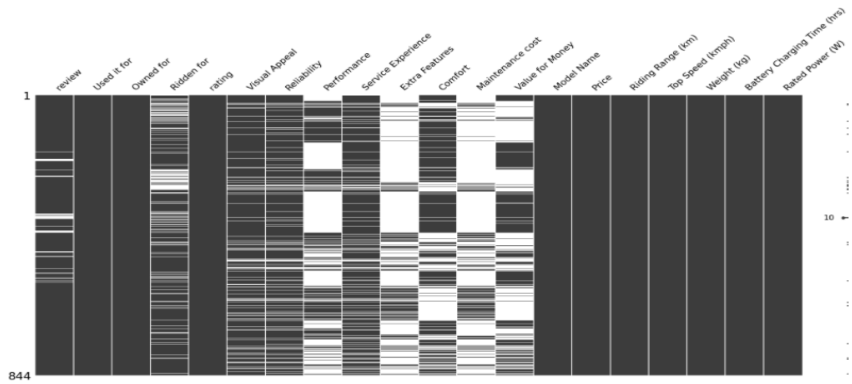
```python
data.head()
```

	review	Used it for	Owned for	Ridden for	rating	Visual Appeal	Reliability	Performance	Service Experience
0	We all checked the bike's capacity to be 150 k...	Daily Commute	Never owned	NaN	1	3.0	4.0	NaN	NaN
1	Performance is very poor on this bike. The cha...	Everything	> 1 yr	< 5000 kms	1	3.0	1.0	NaN	1.0
2	I purchased this in April 2022 and the sales s...	Daily Commute	< 3 months	< 5000 kms	3	4.0	4.0	NaN	2.0
3	If any issues come in scooty parts not availab...	Daily Commute	6 months-1 yr	5000-10000 kms	1	1.0	1.0	NaN	1.0
4	Don't buy this vehicle unless you have a near ...	Daily Commute	6 months-1 yr	< 5000 kms	1	3.0	4.0	NaN	1.0

Extra Features	Comfort	Maintenance cost	Value for Money	Model Name	Price	Riding Range (km)	Top Speed (kmph)	Weight (kg)	Battery Charging Time (hrs)	Rated Power (W)
NaN	4.0	NaN	1.0	TVS iQube	137890	100	78	117	5	3000
NaN	3.0	NaN	3.0	TVS iQube	137890	100	78	117	5	3000
NaN	5.0	NaN	2.0	TVS iQube	137890	100	78	117	5	3000
NaN	1.0	NaN	1.0	TVS iQube	137890	100	78	117	5	3000
NaN	3.0	NaN	2.0	TVS iQube	137890	100	78	117	5	3000

```
msno.matrix(data)
```

<Axes: >




```
pca = PCA(random_state = 42)
pca.fit(data_scaled)
```

```
PCA
PCA(random_state=42)
```

```
data_pca = pca.transform(data_scaled)
df_pca = pd.DataFrame(data_pca, columns = [f'PC{x +1}' for x in range(len(data_segment.columns))])
df_pca.head()
```

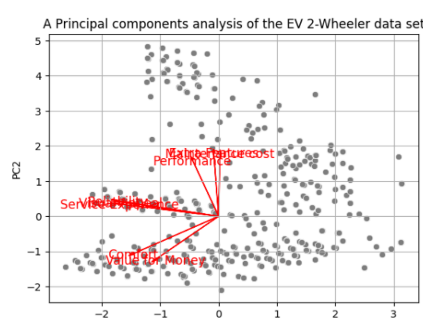
	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
0	0.291227	-1.038055	0.354864	-0.623469	-1.102720	-0.169287	-0.438012	0.957827
1	0.710801	-1.394405	-0.360466	-0.621671	0.320899	-0.086053	-0.426279	-0.149917
2	-0.849149	-1.189765	0.167683	-0.410898	-0.409054	-0.191904	-0.329993	0.830738
3	1.967022	-0.878935	-0.100197	0.330003	-0.075822	0.069599	0.013068	0.011328
4	0.078940	-1.017161	-0.079210	-0.324132	-0.961554	0.005517	-0.107260	0.427972

```
pca_summary
```

	Standard Deviation	Proportion of Variance	Cumultive Proportion
PC1	1.845017	0.425007	0.425007
PC2	1.736646	0.376546	0.801553
PC3	0.903486	0.101915	0.903468
PC4	0.517750	0.033468	0.936936
PC5	0.405239	0.020503	0.957439
PC6	0.379558	0.017987	0.975426
PC7	0.337883	0.014254	0.989680
PC8	0.287510	0.010320	1.000000

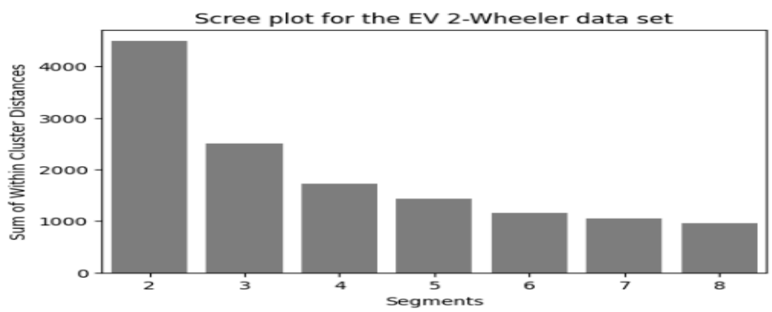
```
factor_loadings = pd.DataFrame(pca.components_, columns = data_segment.columns, index = df_pca.columns).T
factor_loadings.style.background_gradient(cmap = 'Blues')
```

```
# Calculate centroid
index_names = factor_loadings.index

# Plot data points
sns.scatterplot(df_pca, x = 'PC1', y = 'PC2', color = 'grey')

# Plot arrows from centroid to provided coordinates with index names
for i in range(len(factor_loadings['PC1'])):
    plt.arrow(0, 0, factor_loadings['PC1'][i] * 3.5, factor_loadings['PC2'][i] * 3.5, head_width=0.05, head_length=0.1, fc
='red', ec='red')
    plt.text((factor_loadings['PC1'][i] * 3.5), factor_loadings['PC2'][i]* 3.5, index_names[i], fontsize=12, ha='center', c
olor = 'red', va = 'center_baseline')

# Set labels and legend
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.title('A Principal components analysis of the EV 2-Wheeler data set')
plt.grid(True)
plt.show()
```



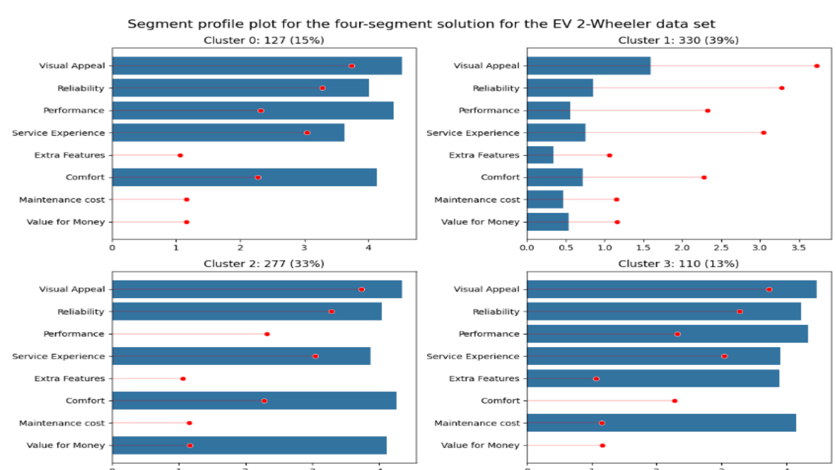
	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Visual Appeal	-0.480170	0.117814	0.063320	-0.730598	0.247014	0.105903	0.375474	0.067539
Reliability	-0.494758	0.124910	-0.002776	0.152447	-0.819319	0.060484	0.117211	0.166384
Performance	-0.128721	0.459145	0.574833	-0.005549	-0.019902	-0.025704	-0.288468	-0.598232
Service Experience	-0.486499	0.100691	-0.054176	0.653781	0.470391	0.052432	0.311210	-0.044129
Extra Features	-0.024373	0.519633	-0.364578	-0.023208	0.116821	0.559390	-0.456829	0.246323
Comfort	-0.418255	-0.304266	0.249807	-0.020111	0.172621	-0.296656	-0.623271	0.404238
Maintenance cost	0.005912	0.513208	-0.386495	-0.054822	0.020302	-0.762039	-0.003360	0.055435
Value for Money	-0.309572	-0.351548	-0.563840	-0.107598	-0.046688	0.009572	-0.260855	-0.617065

```
fig = plt.figure(figsize = (6,4))
sns.barplot(x = list(range(2, 9)), y = wcss, color = 'grey')
plt.xlabel("Segments")
plt.ylabel('Sum of Within Cluster Distances')
plt.title("Scree plot for the EV 2-Wheeler data set")
plt.show()
```



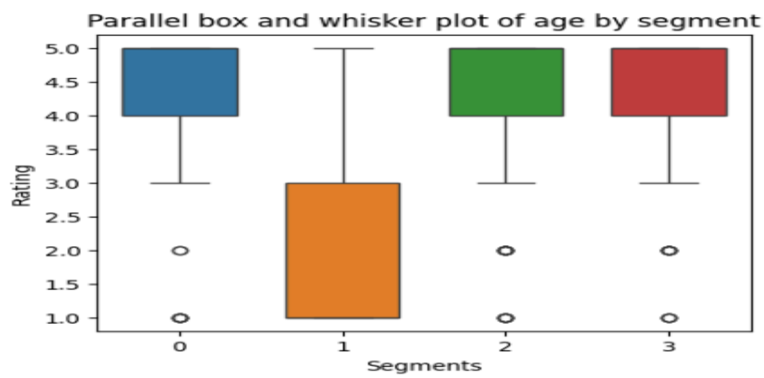
Profiling Segments

```
plt.figure(figsize = (12, 9))
for i in range(4):
    plt.subplot(2, 2, i+1)
    sns.barplot(data_pivot, x = i, y = data_pivot.index)
    sns.scatterplot(data_pivot_mean, x = 'Value', y = 'Variable', color = 'red')
    for index, row in data_pivot_mean.iterrows():
        plt.hlines(y=row['Variable'], xmin=0, xmax=row['Value'], colors='red', alpha = 0.2)
    plt.ylabel("")
    plt.xlabel("")
    plt.title(f"Cluster {i}: {data_profile['cluster'].value_counts()[i]} ({data_profile['cluster'].value_counts()[i]*
100/len(data_profile):.0f}%)")
plt.suptitle("Segment profile plot for the four-segment solution for the EV 2-Wheeler data set", fontsize = 15)
plt.tight_layout()
plt.show()
```



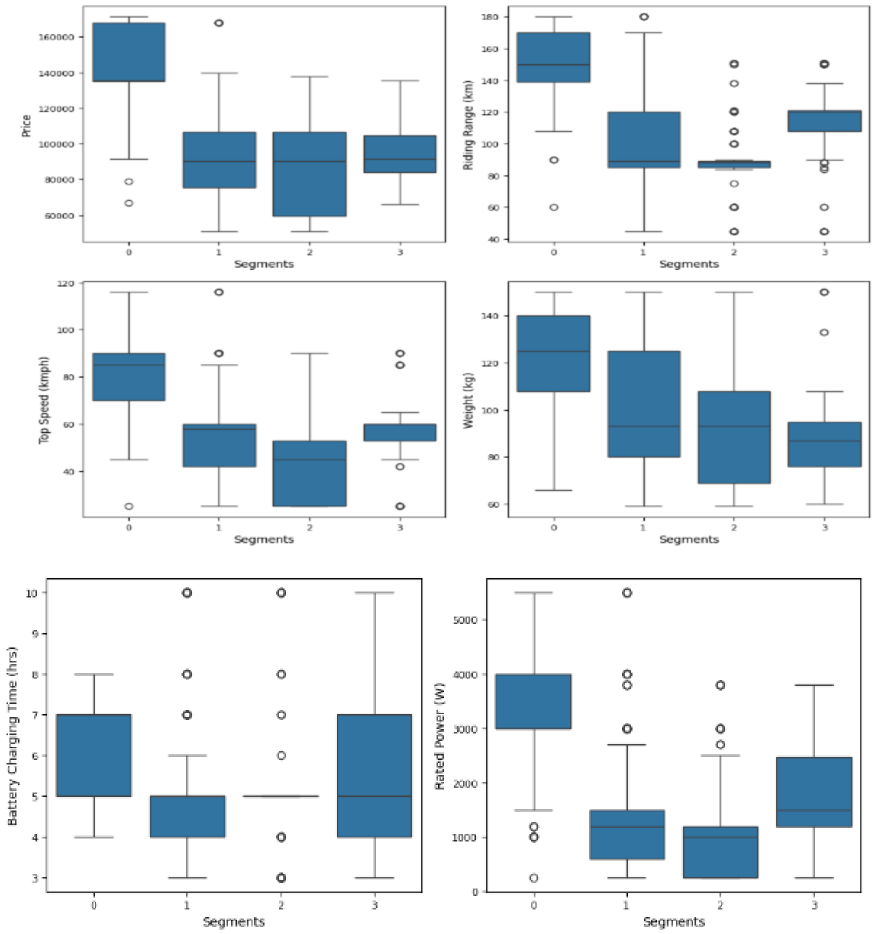
Above graph visually captures the diverse perceptions among different segments. Segment 0, representing 15% of consumers, values the electric two-wheeler vehicle for its visual appeal, reliability, performance, service experience, and comfort. Conversely, Segment 1 (39% of consumers) expresses dissatisfaction across all aspects, marking them as the largest but least satisfied group. Segment 2 (33% of consumers) appreciates visual appeal, reliability, service experience, comfort, and notably, perceives a strong value for money. Lastly, Segment 3 (13% of consumers), the smallest segment, values visual appeal, reliability, performance, service experience, extra features, and maintenance cost, showcasing distinct perceptions, particularly on features and costs.

```
# Number of cluster along the x-axis and rating along the y-axis
plt.figure(figsize = (5,4))
sns.boxplot(data_desc, x= 'cluster', y = 'rating', palette = 'tab10', width = 0.7)
plt.xlabel("Segments")
plt.ylabel("Rating")
plt.title("Parallel box and whisker plot of age by segment")
plt.savefig("rating.png")
plt.show()
```



Above parallel box and whisker plot, emphasizes significant differences in average ratings among segments. Specifically, Segment 1 consumers express dissatisfaction across all perceptions, leading to lower overall ratings.

```
plt.figure(figsize = (10, 12))
for i, col in enumerate(data_desc.columns[-8:-2]):
    plt.subplot(3, 2, i+1)
    sns.boxplot(data_desc, x = 'cluster', y =col)
    plt.xlabel("Segments", fontsize= 10)
    plt.ylabel(col, fontsize = 10)
    plt.xticks(fontsize = 8)
    plt.yticks(fontsize = 8)
plt.tight_layout()
plt.show()
```



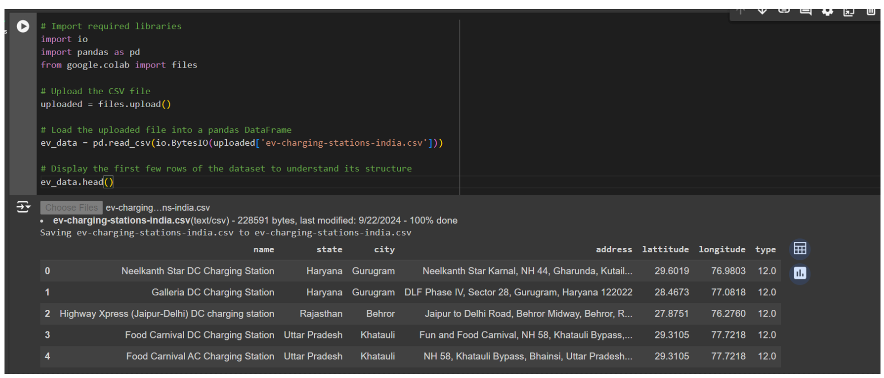
In analyzing technical specification of electric vehicles across different segments, distinct patterns emerge. Segment 0 prefers premium EVs with a higher price range and extended riding range, emphasizing consumer preference for luxury and long-distance travel. Segment 1 focuses on budget-friendly options with lower prices and moderate riding ranges, suitable for daily commuting. Segment 2 and Segment 3 prioritize affordability, with slight differences in riding range and speed preferences. Weight preferences vary, with Segment 0 and Segment 1 favoring heavier vehicles, while Segment 2 and Segment 3 prefer lighter options. Charging time also differs, with Segment 0 and Segment 3 opting for longer durations for overnight charging, while Segment 1 and Segment 2 prioritize faster charging for quick turnaround times. These nuanced preferences shape the electric vehicle market in India.

In summary, our in-depth analysis of India's electric vehicle market led us to identify Segment 1 as the optimal target. With a significant 39% consumer base, this segment represents a substantial market opportunity. By tailoring our electric two-wheeler specifications to meet the preferences of this segment, we ensure our products align seamlessly with the demands of a large customer base. This strategic decision is grounded in a thorough understanding of market segmentation, consumer behavior, and technical specifications. These insights provide a clear direction for our market entry, emphasizing precision and relevance in both product development and marketing strategies. Moving forward, this approach equips us with a solid foundation, ensuring our offerings resonate effectively within India's evolving electric vehicle landscape.

EDA Dataset 7

https://github.com/SaadBebal/EV_Stations_India/blob/main/EV_CHARGINGS_STATIONS_INDIA.ipynb

The project focuses on analyzing the distribution of EV charging stations across India. The dataset, provided in a CSV format, includes details such as geographical coordinates, station type, state, city, and address. After loading the data, an initial exploration was conducted to understand its structure and identify issues like missing values and incorrect column labels. Key data cleaning steps involved correcting the latitude and longitude fields, filling missing values, and handling inconsistencies. The cleaned data was then grouped by state, city, and type of charging station to gain insights into the distribution patterns, followed by visualizations to highlight the geographical spread and concentration of EV infrastructure across the country.

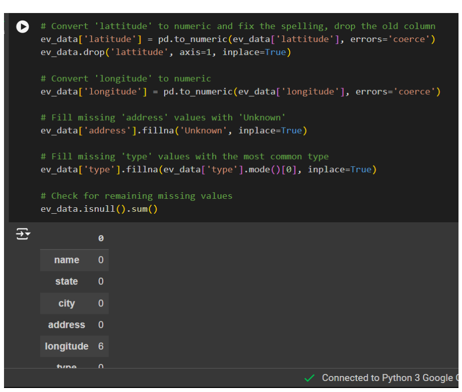


```python
# Import required libraries
import io
import pandas as pd
from google.colab import files

# Upload the CSV file
uploaded = files.upload()

# Load the uploaded file into a pandas DataFrame
ev_data = pd.read_csv(io.BytesIO(uploaded['ev-charging-stations-india.csv']))

# Display the first few rows of the dataset to understand its structure
ev_data.head()
```

Choose Files ev-charging...ns-india.csv
- **ev-charging-stations-india.csv**(text/csv) - 228591 bytes, last modified: 9/22/2024 - 100% done

Saving ev-charging-stations-india.csv to ev-charging-stations-india.csv

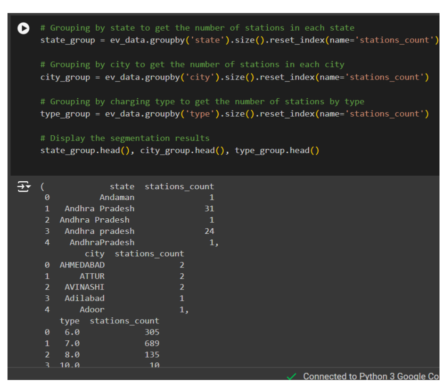
	name	state	city	address	lattitude	longitude	type
0	Neelkanth Star DC Charging Station	Haryana	Gurugram	Neelkanth Star Karnal, NH 44, Gharunda, Kutail...	29.6019	76.9803	12.0
1	Galleria DC Charging Station	Haryana	Gurugram	DLF Phase IV, Sector 28, Gurugram, Haryana 122022	28.4673	77.0818	12.0
2	Highway Xpress (Jaipur-Delhi) DC charging station	Rajasthan	Behror	Jaipur to Delhi Road, Behror Midway, Behror, R...	27.8751	76.2760	12.0
3	Food Carnival DC Charging Station	Uttar Pradesh	Khatauli	Fun and Food Carnival, NH 58, Khatauli Bypass,...	29.3105	77.7218	12.0
4	Food Carnival AC Charging Station	Uttar Pradesh	Khatauli	NH 58, Khatauli Bypass, Bhainsi, Uttar Pradesh...	29.3105	77.7218	12.0



```python
# Convert 'lattitude' to numeric and fix the spelling, drop the old column
ev_data['latitude'] = pd.to_numeric(ev_data['lattitude'], errors='coerce')
ev_data.drop('lattitude', axis=1, inplace=True)

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

# Fill missing 'address' values with 'Unknown'
ev_data['address'].fillna('Unknown', inplace=True)

# Fill missing 'type' values with the most common type
ev_data['type'].fillna(ev_data['type'].mode()[0], inplace=True)

# Check for remaining missing values
ev_data.isnull().sum()
```

	0
name	0
state	0
city	0
address	0
longitude	6
type	0

Connected to Python 3 Google C

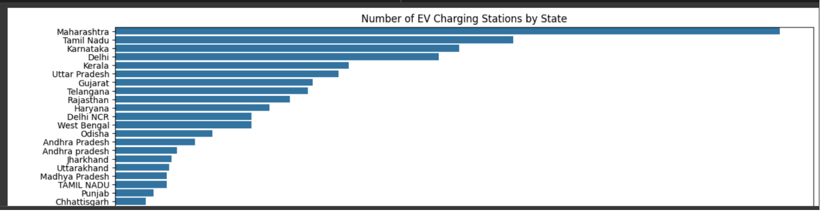


```python
# Grouping by state to get the number of stations in each state
state_group = ev_data.groupby('state').size().reset_index(name='stations_count')

# Grouping by city to get the number of stations in each city
city_group = ev_data.groupby('city').size().reset_index(name='stations_count')

# Grouping by charging type to get the number of stations by type
type_group = ev_data.groupby('type').size().reset_index(name='stations_count')

# Display the segmentation results
state_group.head(), city_group.head(), type_group.head()
```

```
(             state  stations_count
 0          Andaman               1
 1   Andhra Pradesh              31
 2  Andhra Pradesh                1
 3   Andhra pradesh              24
 4    AndhraPradesh               1,
        city  stations_count
 0  AHMEDABAD               2
 1      ATTUR               2
 2   AVINASHI               2
 3   Adilabad               1
 4      Adoor               1,
    type  stations_count
 0   6.0             305
 1   7.0             689
 2   8.0             135
 3  10.0              10
```

Connected to Python 3 Google Co

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

# Bar plot showing the number of charging stations in each state
plt.figure(figsize=(15, 11))
sns.barplot(x='stations_count', y='state', data=state_group.sort_values(by='stations_count', ascending=False))
plt.title('Number of EV Charging Stations by State')
plt.xlabel('Number of Stations')
plt.ylabel('State')
plt.show()
```




```python
# Scatter plot of EV charging stations based on their geographical coordinates
plt.figure(figsize=(10, 6))
plt.scatter(ev_data['longitude'], ev_data['latitude'], c=ev_data['type'], cmap='viridis', alpha=0.6)
plt.colorbar(label='Charging Station Type')
plt.title('EV Charging Stations in India')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.show()
```

Connected to Python 3 Google Compute Engine backend

This block provides a summary of the overall insights gathered from the analysis, which can be useful in drawing conclusions about the distribution and types of charging stations in India.

EDA Dataset 8

https://github.com/SaadBebal/EV_Stations_India/blob/main/EV_INDIA.ipynb

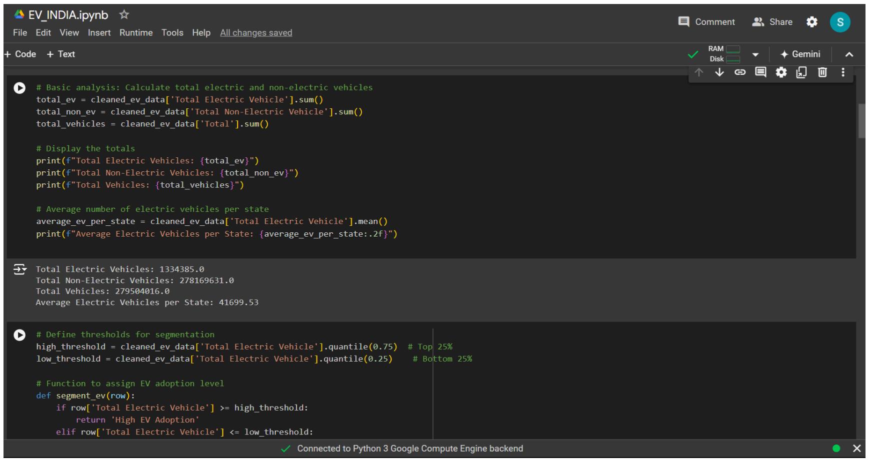
In this section, I provided an introduction to the project, explaining the significance of analyzing electric vehicle (EV) data in India. I highlighted the growing emphasis on sustainable transportation and the need for insights into EV adoption trends.

```python
# Ensure you have pandas imported
import pandas as pd

# Load the data
file_path = '/EV_India.csv'  # Update this with your file path
ev_data = pd.read_csv(file_path)

# Data cleaning: Remove commas, extract numeric values, and convert to float
ev_data['Total Electric Vehicle'] = ev_data['Total Electric Vehicle'].str.replace(',', '').str.extract('(\d+)').astype(float)
ev_data['Total Non-Electric Vehicle'] = ev_data['Total Non-Electric Vehicle'].str.replace(',', '').str.extract('(\d+)').astype(float)
ev_data['Total'] = ev_data['Total'].str.replace(',', '').str.extract('(\d+)').astype(float)

# Drop rows with missing values in critical columns
cleaned_ev_data = ev_data.dropna(subset=['Total Electric Vehicle', 'Total Non-Electric Vehicle'])

# Display cleaned data to verify
cleaned_ev_data.head()
```

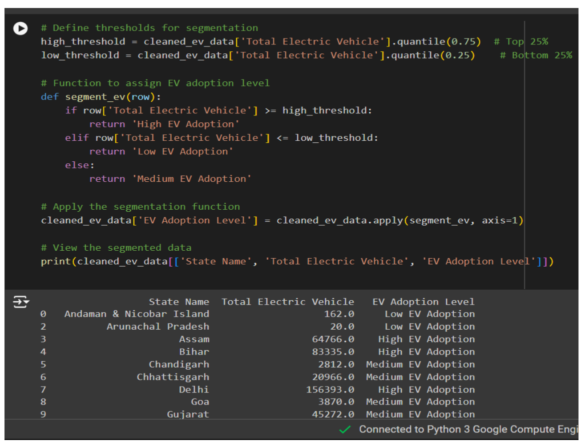
	Sr. No.	State Name	Total Electric Vehicle	Total Non-Electric Vehicle	Total
0	1	Andaman & Nicobar Island	162.0	146945.0	147107.0
2	3	Arunachal Pradesh	20.0	252965.0	252985.0
3	4	Assam	64766.0	4677053.0	4741819.0
4	5	Bihar	83335.0	10407078.0	10490413.0
5	6	Chandigarh	2812.0	746881.0	749693.0

Connected to Python 3 Google Compute Engine backend

I summarized the main objectives and findings of the project, including the overall EV adoption rates, state segmentation based on electric vehicle counts, and the insights gained from various visualizations. This serves as a quick reference for stakeholders to understand the project’s goals and outcomes.



Here, I described the EV_India.csv dataset in detail, listing the columns included in the dataset such as state names and vehicle counts. I also mentioned the source of the data to ensure transparency and reliability.

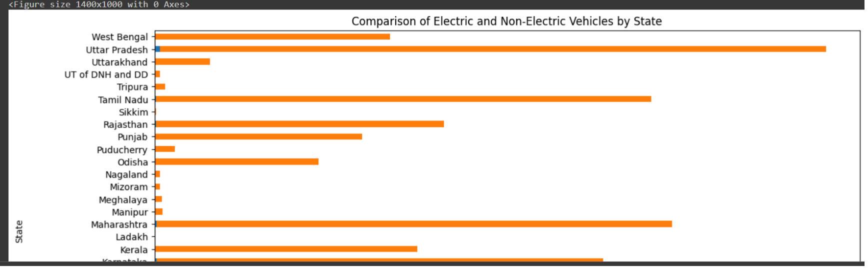


I provided step-by-step instructions for setting up the environment required to run the analysis. This includes listing the necessary libraries (`pandas`, `matplotlib`, `seaborn`) and installation commands, ensuring that anyone interested in the project can easily replicate the setup.

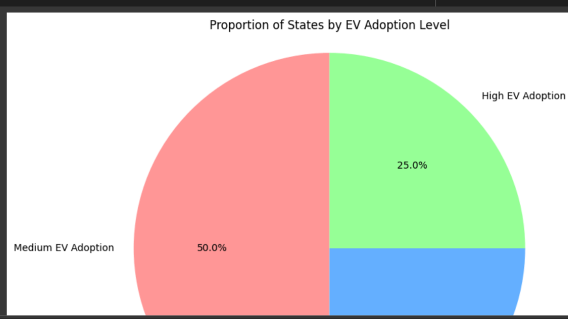
In this part, I elaborated on the analysis performed on the dataset, which includes calculating total and average electric vehicles, as well as segmenting states based on their EV counts. This section provides the technical details of the analysis for those interested in the data-driven aspects of the project
I documented the visualizations created to present the findings effectively. This includes details about the types of visualizations used (e.g., bar plots, pie charts, heatmaps) and the rationale behind each choice. Visualizations help to convey complex information in an easily digestible format.

```python
# Plot comparison between Electric and Non-Electric Vehicles
plt.figure(figsize=(14, 10))
cleaned_ev_data.plot(kind='barh', x='State Name', y=['Total Electric Vehicle', 'Total Non-Electric Vehicle'], stacked=True, figsize=(14, 8))
plt.title('Comparison of Electric and Non-Electric Vehicles by State')
plt.xlabel('Number of Vehicles')
plt.ylabel('State')
plt.legend(['Electric Vehicles', 'Non-Electric Vehicles'])
plt.show()
```

```
<Figure size 1400x1000 with 0 Axes>
```




```python
plt.figure(figsize=(8, 8))
plt.pie(adoption_counts, labels=adoption_counts.index, autopct='%1.1f%%', colors=['#ff9999','#66b3ff','#99ff99'], startangle=90)
plt.title('Proportion of States by EV Adoption Level')
plt.axis('equal')  # Equal aspect ratio ensures the pie is drawn as a circle.
plt.show()
```



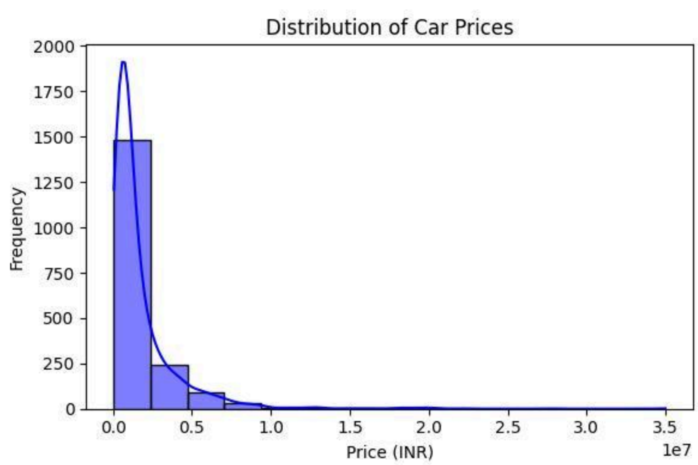
EDA DATASET 9
EV Market Segmentation
https://github.com/Punya2011/FL--Project2-EV_Market_Segment.git

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

# Load the dataset
df_car_details = pd.read_csv('/content/drive/MyDrive/car details v4.csv')

# Remove rows with missing values
df_car_details.dropna(inplace=True)
```

```python
# Preview the dataset
print(df_car_details.head())
```

```
           Make                          Model    Price  Year  Kilometer  \
0         Honda             Amaze 1.2 VX i-VTEC   505000  2017      87150
1  Maruti Suzuki                Swift DZire VDI   450000  2014      75000
2       Hyundai           i10 Magna 1.2 Kappa2   220000  2011      67000
3        Toyota                       Glanza G   799000  2019      37500
4        Toyota  Innova 2.4 VX 7 STR [2016-2020]  1950000  2018      69000

  Fuel Type Transmission   Location   Color   Owner Seller Type   Engine  \
0    Petrol       Manual       Pune    Grey   First   Corporate  1198 cc
1    Diesel       Manual   Ludhiana   White  Second  Individual  1248 cc
2    Petrol       Manual    Lucknow  Maroon   First  Individual  1197 cc
3    Petrol       Manual  Mangalore     Red   First  Individual  1197 cc
4    Diesel       Manual     Mumbai    Grey   First  Individual  2393 cc

           Max Power              Max Torque Drivetrain  Length   Width  \
0   87 bhp @ 6000 rpm      109 Nm @ 4500 rpm        FWD  3990.0  1680.0
1   74 bhp @ 4000 rpm      190 Nm @ 2000 rpm        FWD  3995.0  1695.0
2   79 bhp @ 6000 rpm  112.7619 Nm @ 4000 rpm        FWD  3585.0  1595.0
3   82 bhp @ 6000 rpm      113 Nm @ 4200 rpm        FWD  3995.0  1745.0
4  148 bhp @ 3400 rpm      343 Nm @ 1400 rpm        RWD  4735.0  1830.0

   Height  Seating Capacity  Fuel Tank Capacity
0  1505.0               5.0                35.0
1  1555.0               5.0                42.0
2  1550.0               5.0                35.0
3  1510.0               5.0                37.0
4  1795.0               7.0                55.0
```

```python
# Histogram of Price
plt.figure(figsize=(6, 4))
sns.histplot(df_car_details['Price'], bins=15, kde=True, color='blue')
plt.title('Distribution of Car Prices')
plt.xlabel('Price (INR)')
plt.ylabel('Frequency')
plt.tight_layout()
plt.savefig("car_price_distribution.jpg")
plt.show()
```

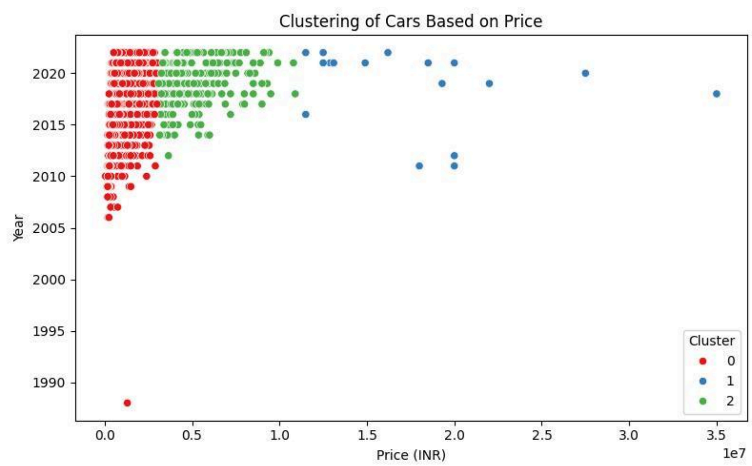


The histogram shows the distribution of car prices in a dataset. The x-axis represents the price in INR, and the y-axis represents the frequency of cars within each price range. The bars indicate the number of cars in each price bin, and the curve (if present) provides a smoother representation of the distribution. This visualization helps to understand the typical car prices in the dataset.

```python
[7] # Scaling the 'Price' for clustering
scaler = StandardScaler()
df_car_details['Price_Scaled'] = scaler.fit_transform(df_car_details[['Price']])
```

```python
[8] # Perform KMeans clustering based on the scaled price
kmeans = KMeans(n_clusters=3, random_state=42)
df_car_details['Cluster'] = kmeans.fit_predict(df_car_details[['Price_Scaled']])
```

```python
# Plot clusters
plt.figure(figsize=(8, 5))
sns.scatterplot(x='Price', y='Year', hue='Cluster', data=df_car_details, palette='Set1')
plt.title('Clustering of Cars Based on Price')
plt.xlabel('Price (INR)')
plt.ylabel('Year')
plt.tight_layout()
plt.savefig("car_clusters.jpg")
plt.show()
```

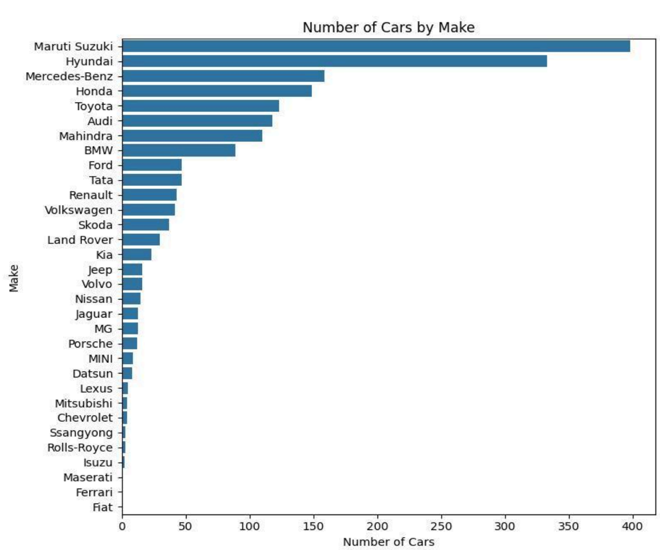


The scatterplot visualizes how cars are grouped based on their price and year. The x-axis represents the price in INR, and the y-axis represents the year. The points are colored according to their cluster, showing the relationship between these two variables within each cluster.

```
[12] # Summarize each cluster's average price
cluster_summary = df_car_details.groupby('Cluster')['Price'].mean()
print("Average Price for each Cluster:\n", cluster_summary)
```

```
Average Price for each Cluster:
 Cluster
0    9.570503e+05
1    1.773158e+07
2    5.121812e+06
Name: Price, dtype: float64
```

```
# Bar plot of Car Count by Make
plt.figure(figsize=(8, 7))
df_car_make_counts = df_car_details['Make'].value_counts()
sns.barplot(x=df_car_make_counts.values, y=df_car_make_counts.index)
plt.title('Number of Cars by Make')
plt.xlabel('Number of Cars')
plt.ylabel('Make')
plt.tight_layout()
plt.savefig("car_make_distribution.jpg")
plt.show()
```

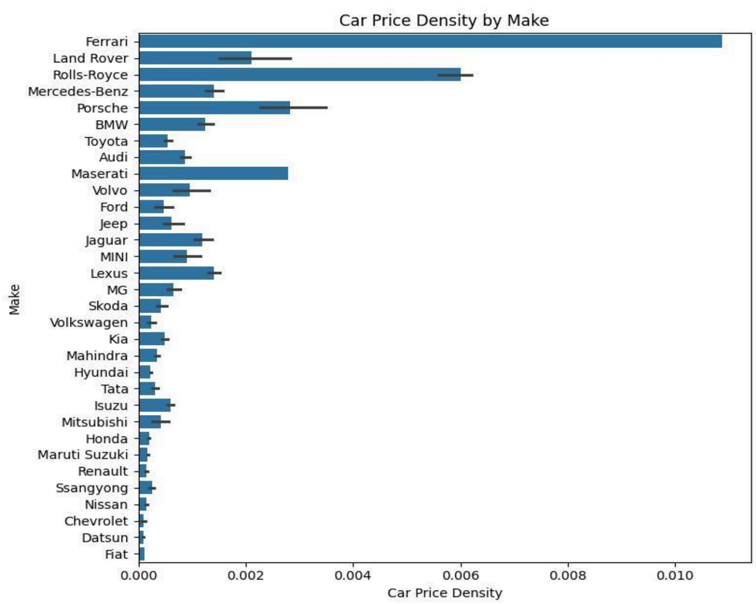


The bar graph shows the number of cars for each car made. The height of each bar represents the number of cars from a particular make, allowing for easy comparison of car popularity.

```
[15] # Add Car Price Density
     df_car_details['Price_Density'] = df_car_details['Price'] / df_car_details['Price'].sum()
```

```
[16] # Sort by density for better visualization
     df_sorted_density = df_car_details.sort_values(by='Price_Density', ascending=False)
```

```
# Plotting the Car Price Density by Make
plt.figure(figsize=(8, 7))
sns.barplot(x='Price_Density', y='Make', data=df_sorted_density)
plt.title('Car Price Density by Make')
plt.xlabel('Car Price Density')
plt.ylabel('Make')
plt.tight_layout()
plt.savefig("car_density_by_make.jpg")
plt.show()
```

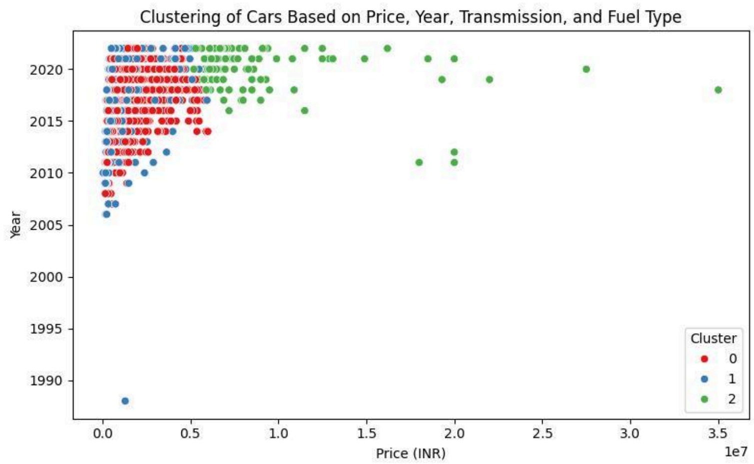


The bar graph shows the price density of cars for different car makers. The height of each bar represents the average price of cars from a particular make, allowing for easy comparison of car prices relative to their availability.

```
[24] # Encode categorical features: Transmission and Fuel Type
     label_encoder_transmission = LabelEncoder()
     label_encoder_fuel = LabelEncoder()

     df_car_details['Transmission_Encoded'] = label_encoder_transmission.fit_transform(df_car_details['Transmission'])
     df_car_details['Fuel_Type_Encoded'] = label_encoder_fuel.fit_transform(df_car_details['Fuel Type'])
```

```
[25] # Perform KMeans clustering based on the scaled features
     kmeans = KMeans(n_clusters=3, random_state=42)
     df_car_details['Cluster'] = kmeans.fit_predict(df_cluster_data)
```

```
# Plot clusters
plt.figure(figsize=(8,5))
sns.scatterplot(x='Price', y='Year', hue='Cluster', data=df_car_details, palette='Set1')
plt.title('Clustering of Cars Based on Price, Year, Transmission, and Fuel Type')
plt.xlabel('Price (INR)')
plt.ylabel('Year')
plt.tight_layout()
plt.savefig("car_clusters_with_additional_features.jpg")
plt.show()
```

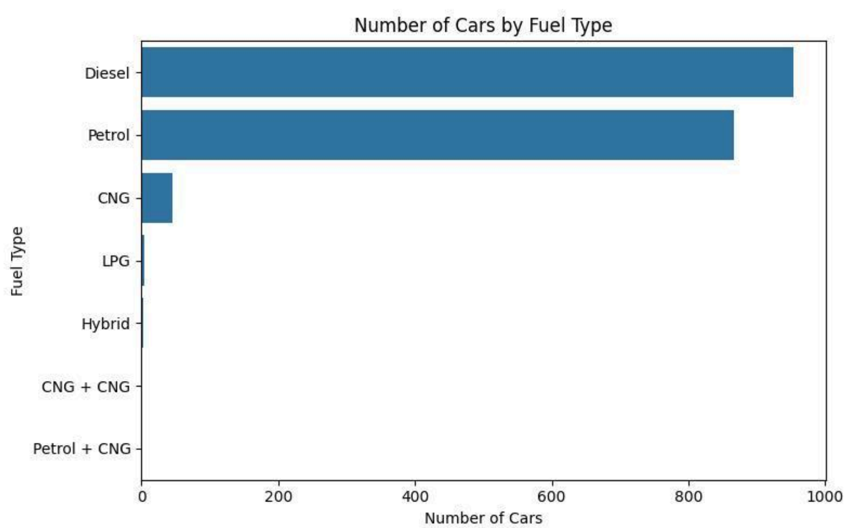


The scatterplot visualizes how cars are grouped based on price, year, transmission, and fuel type. The x-axis represents the price in INR, and the y-axis represents the year. The points are colored according to their cluster, showing the relationship between these variables within each cluster.

```
[28] # Summarize each cluster's average price and year
cluster_summary = df_car_details.groupby('Cluster')[['Price', 'Year']].mean()
print("Average Price and Year for each Cluster:\n", cluster_summary)
```

```
Average Price and Year for each Cluster:
                 Price         Year
Cluster
0        1.621124e+06  2016.442184
1        8.911992e+05  2016.608643
2        9.005196e+06  2019.897196
```

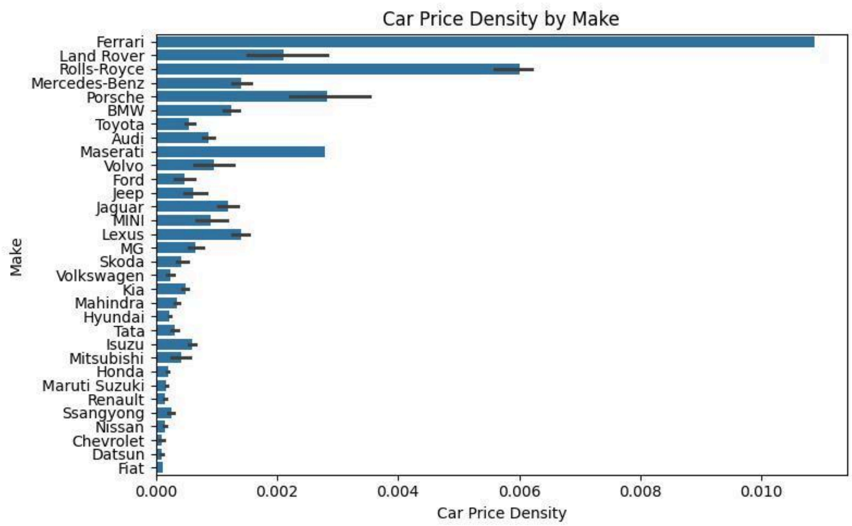
```
# Bar plot of Car Count by Fuel Type
plt.figure(figsize=(8, 5))
df_fuel_type_counts = df_car_details['Fuel Type'].value_counts()
sns.barplot(x=df_fuel_type_counts.values, y=df_fuel_type_counts.index)
plt.title('Number of Cars by Fuel Type')
plt.xlabel('Number of Cars')
plt.ylabel('Fuel Type')
plt.tight_layout()
plt.savefig("car_fuel_type_distribution.jpg")
plt.show()
```



The bar graph shows the number of cars for each fuel type. The height of each bar represents the number of cars using a particular fuel type, allowing for easy comparison of fuel type popularity.

```
# Add Price Density based on Price
df_car_details['Price_Density'] = df_car_details['Price'] / df_car_details['Price'].sum()

# Sort by density for better visualization
df_sorted_density = df_car_details.sort_values(by='Price_Density', ascending=False)
```

```
# Plotting the Car Price Density by Make
plt.figure(figsize=(8, 5))
sns.barplot(x='Price_Density', y='Make', data=df_sorted_density)
plt.title('Car Price Density by Make')
plt.xlabel('Car Price Density')
plt.ylabel('Make')
plt.tight_layout()
plt.savefig("car_density_by_make_with_transmission_fuel_year.jpg")
plt.show()
```



The bar graph shows the car price density for different car makers. The height of each bar represents the average price of cars from a particular make, allowing for easy comparison of car prices relative to their availability.

The dataset contains detailed information about electric vehicles, including their make, model, electric range, model year, and city of registration. The data is cleaned and analyzed to reveal trends such as the distribution of electric ranges, clustering of vehicles based on electric range, and the frequency of electric vehicles by make and city. Vehicles are segmented into high, moderate, and low range categories, with clusters reflecting distinct electric range groups. Visualizations show the distribution of electric range, the number of vehicles per make, and clustering patterns based on electric range and model year. The dataset provides valuable insights into electric vehicle usage and distribution across cities.

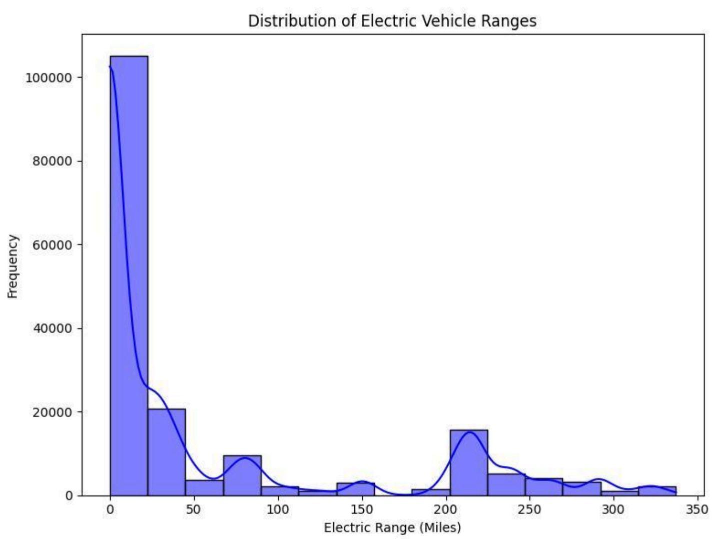
EDA DATASET 10
EV Market Segmentation
https://github.com/Punya2011/FL--Project2-EV_Market_Segment.git

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

# Load the dataset
df_ev_population = pd.read_csv('/content/drive/MyDrive/Electric_Vehicle_Population_Data.csv')

# Remove rows with missing values
df_ev_population.dropna(inplace=True)
```

```
# Preview the dataset with the additional columns
print(df_ev_population[['Make', 'Model', 'Electric Range', 'Model Year', 'City']].head())
```

```
    Make    Model  Electric Range  Model Year       City
0  TESLA  MODEL Y             291        2020    Seattle
1  TESLA  MODEL Y               0        2023    Bothell
2  TESLA  MODEL S             270        2019    Seattle
3  TESLA  MODEL S             210        2016   Issaquah
4  TESLA  MODEL Y               0        2021  Suquamish
```

```
# Histogram of Electric Range
plt.figure(figsize=(8, 6))
sns.histplot(df_ev_population['Electric Range'], bins=15, kde=True, color='blue')
plt.title('Distribution of Electric Vehicle Ranges')
plt.xlabel('Electric Range (Miles)')
plt.ylabel('Frequency')
plt.tight_layout()
plt.savefig("ev_range_distribution.jpg")
plt.show()
```

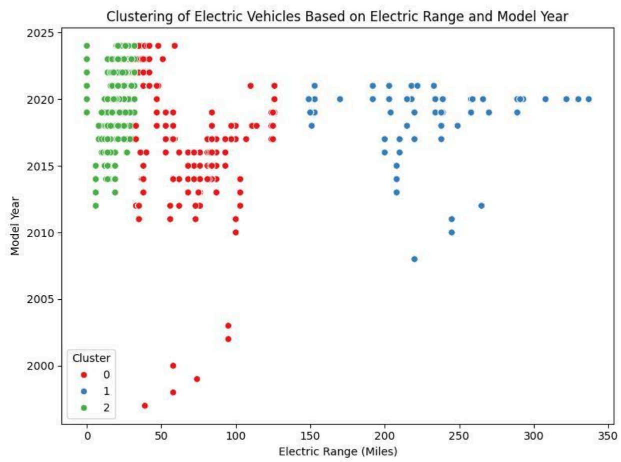


The histogram shows the distribution of electric vehicle ranges in a dataset. The x-axis represents the electric range in miles, and the y-axis represents the frequency of vehicles within each range. The bars indicate the number of vehicles in each range, and the curve provides a smoother representation of the distribution. This visualization helps to understand the typical range of electric vehicles in the dataset.

```
[38] # Scaling the 'Electric Range' for clustering
     scaler = StandardScaler()
     df_ev_population['Range_Scaled'] = scaler.fit_transform(df_ev_population[['Electric Range']])
```

```
[39] # Perform KMeans clustering based on the scaled electric range
     kmeans = KMeans(n_clusters=3, random_state=42)
     df_ev_population['Cluster'] = kmeans.fit_predict(df_ev_population[['Range_Scaled']])
```

```
# Plot clusters with Electric Range vs Model Year
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Electric Range', y='Model Year', hue='Cluster', data=df_ev_population, palette='Set1')
plt.title('Clustering of Electric Vehicles Based on Electric Range and Model Year')
plt.xlabel('Electric Range (Miles)')
plt.ylabel('Model Year')
plt.tight_layout()
plt.savefig("ev_clusters.jpg")
plt.show()
```

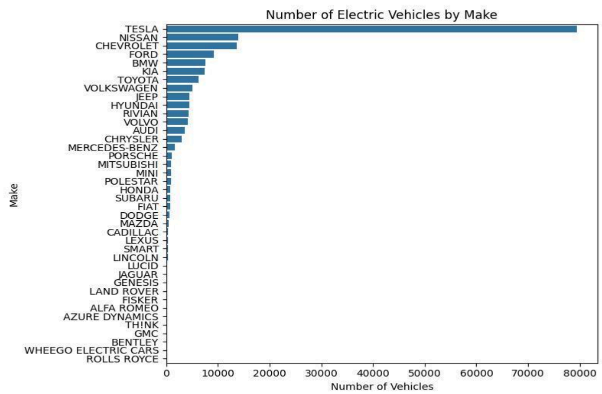


The scatterplot visualizes how electric vehicles are grouped based on their electric range and model year. The x-axis represents the electric range, and the y-axis represents the model year. The points are colored according to their cluster, showing the relationship between these two variables within each cluster.

```
[42] # Summarize each cluster's average electric range
     cluster_summary = df_ev_population.groupby('Cluster')['Electric Range'].mean()
     print("Average Electric Range for each Cluster:\n", cluster_summary)
```

```
Average Electric Range for each Cluster:
 Cluster
0     64.500718
1    232.295706
2      4.858689
Name: Electric Range, dtype: float64
```

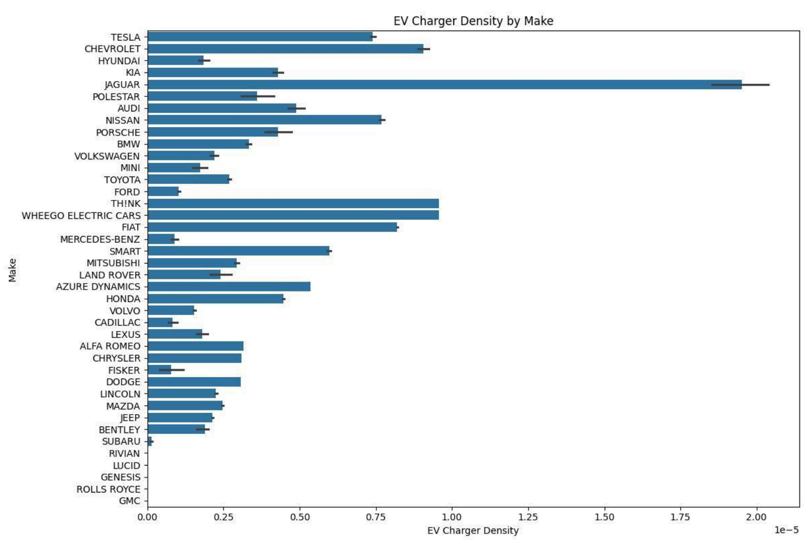
```
# Bar plot of EV Count by Make
plt.figure(figsize=(8, 6))
df_ev_make_counts = df_ev_population['Make'].value_counts()
sns.barplot(x=df_ev_make_counts.values, y=df_ev_make_counts.index)
plt.title('Number of Electric Vehicles by Make')
plt.xlabel('Number of Vehicles')
plt.ylabel('Make')
plt.tight_layout()
plt.savefig("ev_make_distribution.jpg")
plt.show()
```



The bar graph shows how many electric vehicles there are from each car manufacturer (Make) in your data. The height of each bar represents the number of electric vehicles, and the labels on the bottom (y-axis) indicate the different car makes. This helps you see which manufacturers have the most electric vehicles in your dataset.

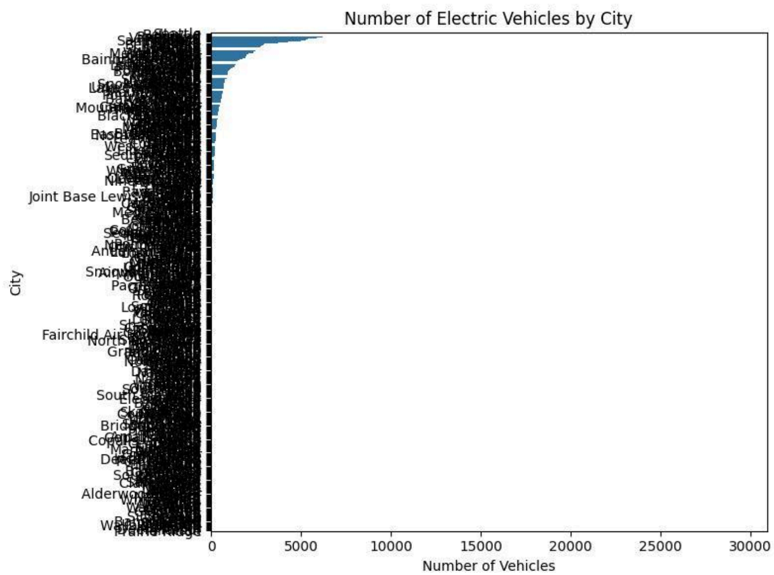
```
# Add EV Density based on Electric Range
df_ev_population['EV_Charge_Density'] = df_ev_population['Electric Range'] / df_ev_population['Electric Range'].sum()

# Sort by density for better visualization
df_sorted_density = df_ev_population.sort_values(by='EV_Charge_Density', ascending=False)
```

```
# Plotting the EV Charge Density by Make
plt.figure(figsize=(12, 8))
sns.barplot(x='EV_Charge_Density', y='Make', data=df_sorted_density)
plt.title('EV Charger Density by Make')
plt.xlabel('EV Charger Density')
plt.ylabel('Make')
plt.tight_layout()
plt.savefig("ev_density_by_make.jpg")
plt.show()
```

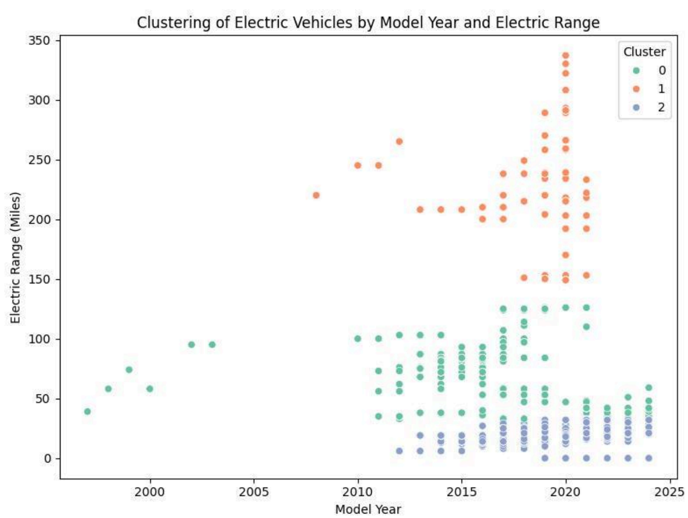


The bar graph shows the density of electric vehicle chargers for different car makers. The height of each bar represents the number of chargers per electric vehicle for that car maker, allowing for easy comparison of charger availability.

```python
# Plot EV distribution by City
plt.figure(figsize=(8, 6))
df_ev_city_counts = df_ev_population['City'].value_counts()
sns.barplot(x=df_ev_city_counts.values, y=df_ev_city_counts.index)
plt.title('Number of Electric Vehicles by City')
plt.xlabel('Number of Vehicles')
plt.ylabel('City')
plt.tight_layout()
plt.savefig("ev_city_distribution.jpg")
plt.show()
```



The bar graph shows the number of electric vehicles in different cities. The height of each bar represents the number of vehicles in a city, allowing for easy comparison of electric vehicle distribution across cities.



The scatterplot visualizes how electric vehicles are grouped based on their model year and electric range. The x-axis represents the model year, and the y-axis represents the electric range. The points are colored according to their cluster, showing the relationship between these two variables within each cluster.

This dataset explores the distribution and clustering of cars based on features such as price, make, city, model year, and model. After cleaning, the dataset is analyzed using KMeans clustering to group cars based on their prices, revealing three distinct price clusters. Visualizations include the distribution of car prices, car counts by make, and price density by make. Cars are also segmented into high, moderate, and low price categories. Additionally, insights are gained into the distribution of cars by city, and relationships between model year and price are explored through scatter plots. The analysis highlights price-driven patterns in car distribution across various cities and makes.

Analysis market segment

To effectively analyze the electric vehicle (EV) market segment in India, it is crucial to adopt a comprehensive and systematic approach. This process will involve identifying relevant data sources, defining segmentation criteria, employing analytical techniques, and ultimately formulating a robust market entry strategy.

1. Identify Data Sources

The first step is to gather relevant data that will inform the analysis. Key sources include:

- Market Reports: Utilize reports from organizations like NITI Aayog, which often provide insights into EV adoption trends, government policies, and market forecasts. Industry publications from research firms like McKinsey and Deloitte can also offer valuable perspectives.
- Government Data: Access statistics related to vehicle registrations, charging infrastructure, and EV-related incentives. The Ministry of Heavy Industries and Public Enterprises often publishes such data.
- Surveys and Studies: Conduct or refer to consumer surveys that capture attitudes towards EVs, preferences regarding vehicle features, and behavioral insights. These studies can highlight consumer motivations and barriers to adoption.

2. Segmentation Criteria

Segmentation is essential to tailor marketing efforts effectively. The following criteria will be considered:

- Geographic Segmentation: Focus on urban centers with high pollution levels and robust government incentives for EV adoption, such as:

    - Delhi: Known for severe air quality issues and strong EV policies.
    - Bengaluru: A tech hub with a growing infrastructure for EV charging.
    - Mumbai: High vehicle density and awareness of environmental issues.
- Analyze regions with developed charging infrastructure, which can significantly impact adoption rates.
- Demographic Segmentation: Target specific demographic groups based on characteristics such as:
    - Age: Focus on young professionals (ages 25-40) and families, as they are more likely to embrace new technologies.
    - Income Level: Target the middle to upper-middle-class segments, as they typically have higher disposable incomes to invest in EVs.
    - Education Level: Consider consumers with higher education levels, who are often more aware of environmental issues and technological advancements.
- Psychographic Segmentation: Explore consumer lifestyles and values:
    - Lifestyle: Target environmentally conscious consumers and tech-savvy individuals who prioritize sustainability.
    - Values: Focus on consumers who are committed to innovation and reducing their carbon footprint.
- Behavioral Segmentation: Analyze consumer behavior patterns:
    - Usage Rate: Identify segments based on vehicle usage frequency (e.g., daily commuters versus occasional users) to tailor marketing messages.
    - Loyalty: Assess brand loyalty among existing vehicle owners and their openness to exploring new EV brands.

3. Data Analysis Techniques

Employing appropriate analytical techniques will help derive meaningful insights from the data:

- Descriptive Statistics: Summarize key data characteristics to identify trends in EV adoption, consumer preferences, and market growth.
- Clustering Analysis: Utilize clustering techniques, such as K-means clustering, to group consumers with similar attributes. This approach can reveal distinct market segments based on their characteristics.
- Regression Analysis: Analyze the factors influencing EV adoption, such as price sensitivity, charging convenience, environmental awareness, and government

incentives. This will help predict how changes in these factors affect consumer behavior.

4. Profiling Segments

Once segments are identified, create detailed profiles for each segment. This profiling should include:

- Demographic information (age, income, education).
- Psychographic attributes (lifestyle choices, values).
- Behavioral insights (usage patterns, brand loyalty).

Assess each segment's size, growth potential, and readiness to adopt EVs. This profiling will facilitate targeted marketing efforts.

5. Selection of Target Segment

Evaluate the identified segments based on criteria such as potential profitability, strategic alignment with the startup's capabilities, and overall market dynamics. Prioritize segments that exhibit:

- High growth potential.
- Lower competition.
- Greater alignment with the startup's mission and product offerings.

6. Market Entry Strategy

Develop a comprehensive market entry strategy that includes:

- Pricing Strategy: Determine an optimal pricing range that considers production costs, competitor pricing, and consumer willingness to pay. This pricing strategy should reflect the perceived value of the EVs among the target demographic.
- Marketing Mix: Customize the marketing mix (product features, promotional strategies, distribution channels) to resonate with the target segments. Highlight unique selling propositions (USPs) that appeal to consumer values and preferences.
- Distribution Channels: Consider various distribution models, such as partnerships with local dealerships, online sales platforms, or direct-to-consumer approaches. A well-thought-out distribution strategy can enhance accessibility and convenience for consumers.

7. Forecasting Potential Sales

Estimate the potential customer base in the chosen target segment. This involves calculating expected profits using the formula:

Potential Profit=Potential Customer Base×Target Price Range

This forecasting will provide insights into the financial viability of entering the market.

Customizing market mix

To effectively penetrate the electric vehicle (EV) market in India, the marketing mix—commonly referred to as the 4Ps (Product, Price, Place, Promotion)—must be tailored to align with the needs and preferences of the target segments identified during market analysis. Here's how to customize each element:

1. Product

- Features and Specifications: Develop EV models with features that cater to the preferences of the target demographic. For example:
    - Battery Range: Offer models with varying battery ranges to address the concerns of daily commuters versus long-distance travelers.
    - Smart Technology: Integrate smart features like connectivity with mobile apps for monitoring battery health, charging status, and navigation to charging stations.
    - Sustainability: Highlight eco-friendly materials used in manufacturing to appeal to environmentally conscious consumers.
- Variety of Models: Provide a range of models to cater to different market segments:
    - Compact Cars: Target urban dwellers looking for cost-effective and efficient vehicles.
    - Luxury Models: Appeal to higher-income consumers with premium features and enhanced performance.
    - Commercial Vehicles: Consider electric variants for businesses, such as delivery vans or e-rickshaws, tapping into the B2B market.
- After-Sales Service: Emphasize robust after-sales support, including maintenance packages, warranty services, and a network of service centers to build consumer trust.

2. Price

- Pricing Strategy: Implement a pricing strategy that reflects the value offered while remaining competitive in the market:

    - Penetration Pricing: Introduce EVs at lower prices initially to attract early adopters and gain market share.
    - Incentives and Subsidies: Leverage government incentives for EV buyers, ensuring transparency about potential savings to encourage purchases.
- Financing Options: Provide flexible financing plans, including leasing options, low-interest loans, or installment payment plans to make EVs more accessible.
- Price Differentiation: Consider different pricing for various models based on features and target segments, ensuring that each price point corresponds to consumer expectations and perceived value.

3. Place

- Distribution Channels: Develop a multi-channel distribution strategy:
    - Online Sales: Offer direct-to-consumer online sales through a user-friendly website or app, allowing consumers to customize their vehicles and arrange home delivery.
    - Dealership Partnerships: Collaborate with existing automotive dealerships to expand reach and provide test drives, enhancing customer engagement.
- Charging Infrastructure: Partner with charging station providers to enhance accessibility. Consider creating a network of charging stations in key urban areas and along major highways to address range anxiety.
- Geographic Focus: Concentrate initial sales efforts in urban areas with higher EV adoption rates and supportive policies, expanding gradually to less saturated regions.

4. Promotion

- Awareness Campaigns: Develop campaigns to educate consumers about the benefits of EVs, focusing on environmental impact, cost savings, and technological advancements. Utilize digital platforms and social media for broad reach.
- Targeted Advertising: Create targeted ads that resonate with specific segments, such as highlighting family-friendly features for families or showcasing eco-friendly benefits for environmentally conscious consumers.
- Influencer Collaborations: Partner with influencers and industry experts to build credibility and trust. Their endorsements can effectively sway public perception and encourage adoption.
- Community Engagement: Organize events, workshops, and test-drive sessions to engage with potential customers directly. This hands-on approach helps demystify EV technology and fosters community connection.

Potential sales

Estimating potential sales is crucial for understanding the market viability and profitability of the electric vehicle (EV) startup. This estimation involves calculating the potential customer base, determining the target price range, and projecting revenue. Here's a structured approach to estimate potential sales:

1. Identify Target Market Segments

- Based on the market segmentation analysis, focus on the identified segments most likely to adopt EVs. For instance:
  - Urban Commuters: Young professionals and families living in major cities.
  - Corporate Fleets: Businesses looking for sustainable transportation solutions.
  - Environmentally Conscious Consumers: Individuals prioritizing eco-friendly options.

2. Estimate the Potential Customer Base

- Market Size: Research the total number of vehicles registered in target urban areas. For example, if a city like Delhi has 10 million registered vehicles and the EV market penetration is projected at 5%, the potential EV market size would be: Potential EV Market=10,000,000×0.05=500,000 EVs
- Adoption Rate: Estimate the percentage of the target segment likely to switch to EVs based on factors such as government incentives, charging infrastructure, and awareness campaigns. For example, if the estimated adoption rate is 20% among urban commuters:

3. Determine Target Price Range

- Analyze competitor pricing and consumer willingness to pay. Suppose the EV models are priced between ₹10,00,000 to ₹15,00,000. The average target price can be calculated as: Average Price=(10,00,000+15,00,000)/2=₹12,50,000

4. Calculate Potential Revenue

- With the estimated potential customer base and average target price, calculate potential sales revenue using the formula:

Potential Sales Revenue=Potential Customers×Average Price
For our example:
Potential Sales Revenue=100,000×12,50,000=₹1,250,000,000,000 (or ₹1.25 trillion)

5. Adjust for Market Dynamics

- Consider factors that may affect sales, such as:
    - Market Penetration Timeline: Estimate how quickly the startup can capture the market based on marketing strategies and production capacity.
    - Economic Conditions: Account for economic factors that may impact consumer purchasing power, such as inflation or economic downturns.
    - Competitive Landscape: Analyze the presence of competitors and potential market share adjustments.

6. Projection Over Time

- Project potential sales over a defined period (e.g., 3-5 years) to understand growth trajectories. Consider factors like:
    - Yearly sales growth rates based on market trends and company performance.
    - Incremental adoption rates as consumer awareness and infrastructure improve.

Market strategy

1. Target Market Segmentation

A comprehensive segmentation approach is essential for identifying and focusing on the most promising customer segments. The segmentation strategy will be based on:

- Geographic Segmentation: Focus on urban areas with higher adoption potential, like metropolitan cities (Delhi, Mumbai, Bengaluru, Chennai) that have government incentives, charging infrastructure, and high awareness of EV benefits.
- Demographic Segmentation:
    - Income Level: Target higher-income groups for premium EVs and middle-income groups for affordable, compact models.
    - Age: Focus on environmentally conscious millennials (aged 25-40) who are tech-savvy and more likely to adopt EVs.
    - Gender: Include women in the target demographic, particularly in urban areas, as safety and convenience are key selling points.
- Psychographic Segmentation: Target eco-conscious individuals who are motivated by environmental concerns, sustainability, and fuel cost savings.

- Behavioral Segmentation: Prioritize frequent urban commuters and businesses with last-mile delivery needs, as they seek efficient and cost-saving alternatives to traditional fuel vehicles.

2. Innovation Adoption Life Cycle

The strategy will align with the Technology Adoption Life Cycle, which includes:

- Innovators and Early Adopters: Focus on tech-savvy and environmentally conscious consumers. This group is critical for initial brand building and product acceptance.
- Early Majority: Gradually expand to mass-market consumers by offering practical and affordable EV models and promoting ease of use, low maintenance, and financial savings over time.

3. Product Development and Positioning

- Product Range: Offer a diverse range of EVs, including:
  - Compact Electric Cars: For urban commuters seeking affordability, convenience, and low operating costs.
  - Premium EV Models: For higher-income consumers desiring luxury, cutting-edge technology, and sustainability.
  - Commercial Vehicles: For B2B customers in logistics, delivery services, and urban transport.
- Battery Technology: Position vehicles as reliable and efficient with advanced battery technology ensuring high performance, faster charging, and extended range.
- Sustainability and Innovation: Highlight the eco-friendliness of EVs through marketing campaigns that focus on reducing carbon footprints, contributing to a cleaner environment, and driving innovation in the automotive sector.

4. Strategic Pricing

- Penetration Pricing: Offer competitive pricing for entry-level EVs to attract early adopters and penetrate the market quickly.
- Government Incentives: Leverage government subsidies, tax rebates, and incentives for EV buyers to lower the effective cost and make EVs more accessible.
- Flexible Financing Options: Provide easy financing plans, low down payments, and attractive loan terms to encourage customers to switch to EVs.
- Long-Term Savings: Promote the lower cost of ownership, focusing on savings in fuel costs, maintenance, and government incentives for charging infrastructure.

5. Distribution and Sales Channels

- Online Direct-to-Consumer Sales: Build a robust e-commerce platform to allow customers to research, customize, and purchase EVs online, reflecting changing consumer behavior towards online shopping.
- Dealership Partnerships: Establish strategic partnerships with dealerships for physical presence, test drives, and servicing.
- Charging Infrastructure Partnerships: Collaborate with charging station providers and local governments to enhance EV charging accessibility, particularly in key urban areas and highways.
- Geographic Focus: Initially focus on EV-friendly cities (like Delhi, Mumbai, and Bengaluru) with higher consumer awareness, developed charging infrastructure, and progressive government policies supporting EV adoption.