Market Segmentation

Electric Vehicle (EV) Market Analysis and Segmentation Strategy in India

by

Shweta Shaw

Introduction

This report focuses on identifying key segments and devising strategies for entering the Indian EV market. Using data analysis and market segmentation, it aims to provide insights into suitable geographic, demographic, psychographic, and behavioural segments for initial market penetration. The strategy aligns with the principles of the **Innovation Adoption Life Cycle** to ensure successful early adoption.

Objectives

- Analyse the Indian EV market to identify early adopters.
- Determine suitable segments using available data.
- Propose a data-driven entry strategy focusing on pricing, location, and customer needs.

Methodology

- **Data Sources**: Research on EV market trends, vehicle usage statistics, charging station availability, and consumer behaviour.
- **Tools Used**: Python, Pandas, NumPy, Matplotlib, and Seaborn for data analysis and visualization.
- **Key Metrics**: EV adoption rates, city-wise vehicle registration data, income distribution, infrastructure availability, and behavioural preferences.

Situational Analysis

4.1 *Environment*

To develop a successful marketing strategy for the EV market in India, a situation analysis is essential. The adoption of electric vehicles (EVs) is growing at an unprecedented rate due to increasing awareness of sustainability and government incentives. Urban areas are leading the transition, reflecting a need for enhanced charging infrastructure and user-friendly technology.

4.2 *Current Trends in the EV Market*

Increasing EV Adoption and Usage

- With rising fuel costs and environmental concerns, more consumers are opting for EVs, driving growth in the market. This trend is expected to accelerate as government policies and incentives support the transition.

Technology as a Growth Driver

- The EV sector relies heavily on innovations in battery efficiency, charging technology, and mobile connectivity. Smartphone apps play a key role in enhancing user experience by offering features like route optimization and real-time charging station availability.

Evolution of Charging Infrastructure Models

- The market is witnessing diversification in charging solutions, from public fast-charging stations to home-based setups, enabling more convenient and accessible charging.

Intense Competition Among EV Brands and Charging Networks

- Similar to disruptions in other industries, both established brands and new players are competing to provide better EV options and charging services, reshaping traditional business models.

Demographic Segmentation

- **Key Focus**: Urban middle-income groups with higher disposable incomes, as well as tech-savvy millennials and Gen Z individuals.
- **Geographic Target**: Urban and semi-urban areas where infrastructure and adoption rates are rapidly improving.

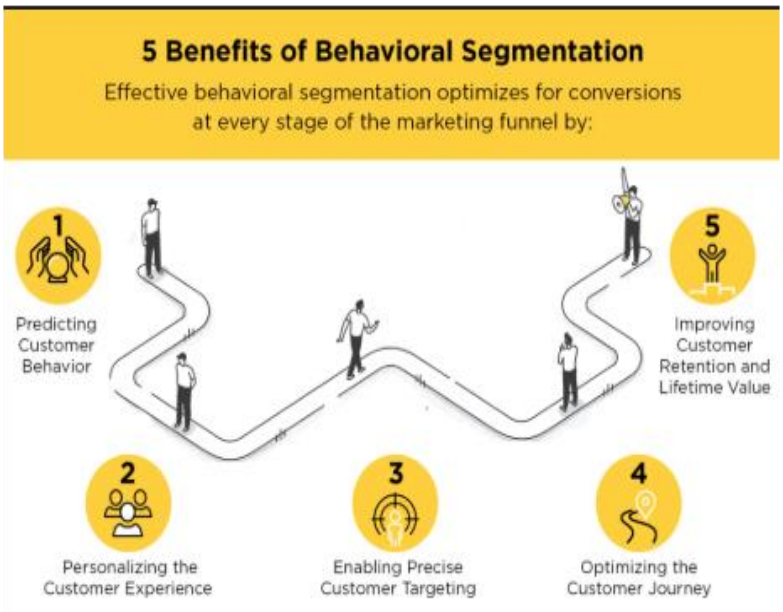
5.1 ***Pros****:*

- Identifies the most likely buyers, aiding precise targeting.
- Easy to gather demographic data for focused outreach.

5.2 ***Cons***:

- Relies on assumptions about uniformity within demographic groups.
- Requires constant updates to remain relevant.

Behavioural Segmentation



Advocates

- Customers who value environmental sustainability and actively promote EV adoption.

Pragmatic Users

- Cost-conscious individuals who switch brands or charging services based on price and efficiency.

Key Traits

- Tech-Savvy: Dependence on mobile apps for navigation, charging, and vehicle performance tracking.
- Experience-Oriented: Positive experiences with EVs and related services increase loyalty.

Psychographic Segmentation

Achievers

- Goal-oriented individuals who value convenience and time efficiency. They prefer pre-scheduled charging services to align with their busy schedules.

Experiencers

- Early adopters who are eager to try the latest EV technologies and trends. This group values cutting-edge features and aesthetics in EVs.

Geographic Segmentation

8.1. *Advantages*

- **Targeted Strategies**: Focuses on regions with high EV demand, such as metropolitan cities.
- **Market Expansion**: Easy to replicate successful strategies in neighbouring areas.
- **Localized Communication**: Ensures marketing messages are tailored to regional preferences and needs.
- **Profitability**: Increases brand recall and customer retention by addressing specific local challenges.

8.2. *Implementation*

- Group consumers based on location to deliver region-specific content and services, such as access to charging stations or subsidies.

Data Analysis

9.1*: About dataset:*

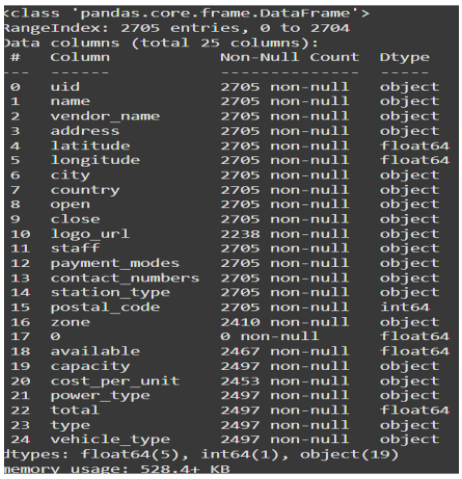
The df.info () method in Pandas provides a concise summary of a Data Frame, which includes details about:

1. **The class type of the object**: Confirms that df () is a Pandas Data Frame.
2. **The index range**: Displays the number of rows (entries) and their range.
3. **The number of columns and their names**.
4. **The number of non-null (non-missing) values in each column**.
5. **The data type (D type) of each column**.
6. **Memory usage** of the Data Frame.

Column Analysis:

For example:

7. **latitude and longitude**: Contain float64 values for geographic coordinates.
8. **logo_url**: Only 2238 non-null out of 2705, meaning 467 rows have missing values.
9. **0**: Contains 0 non-null, indicating the column is empty.
10. **total**: Contains 2497 non-null, so 208 rows have missing data.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2705 entries, 0 to 2704
Data columns (total 25 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   uid              2705 non-null   object
 1   name             2705 non-null   object
 2   vendor_name      2705 non-null   object
 3   address          2705 non-null   object
 4   latitude         2705 non-null   float64
 5   longitude        2705 non-null   float64
 6   city             2705 non-null   object
 7   country          2705 non-null   object
 8   open             2705 non-null   object
 9   close            2705 non-null   object
 10  logo_url         2238 non-null   object
 11  staff            2705 non-null   object
 12  payment_modes    2705 non-null   object
 13  contact_numbers  2705 non-null   object
 14  station_type     2705 non-null   object
 15  postal_code      2705 non-null   int64
 16  zone             2410 non-null   object
 17  0                0 non-null      float64
 18  available        2467 non-null   float64
 19  capacity         2497 non-null   object
 20  cost_per_unit    2453 non-null   object
 21  power_type       2497 non-null   object
 22  total            2497 non-null   float64
 23  type             2497 non-null   object
 24  vehicle_type     2497 non-null   object
dtypes: float64(5), int64(1), object(19)
memory usage: 528.4+ KB
```

Code Explanation:

Visualizing Missing Values

11.1**:** *Calculate Percentage of Missing Values:*

```python
missing_percentage = df.isnull().mean() * 100
```

- Computes the percentage of missing values for each column by dividing the count of NaN values by the total number of rows.
- Multiplies by 100 to express as a percentage.

11.2*: Create a Data Frame for Missing Information*

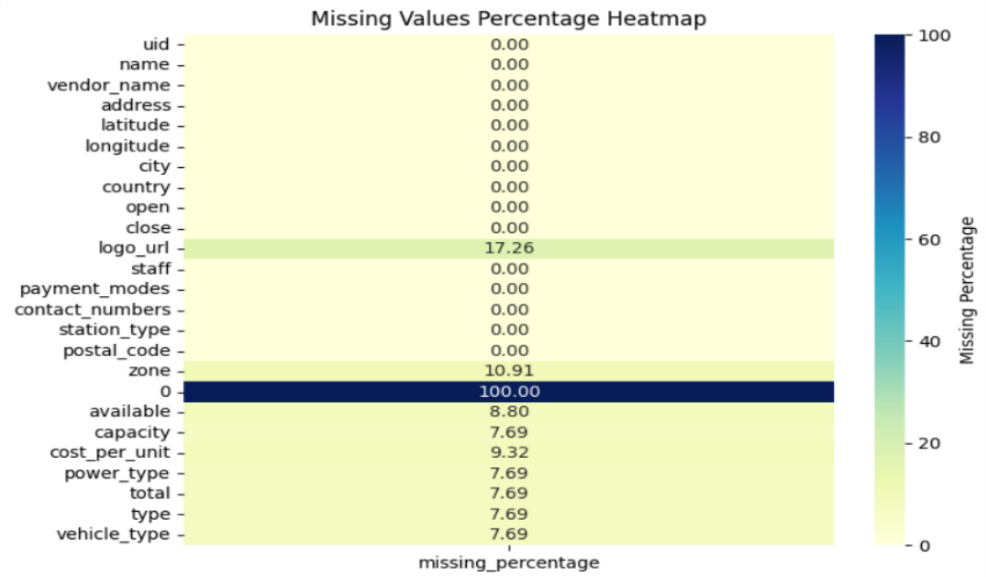
```
# Create a DataFrame with missing percentage information
missing_info = pd.DataFrame({'column': df.columns, 'missing_percentage': missing_percentage})
```

- Combines the column names and their respective missing percentages into a new Data Frame for better organization.

11.3*: Generate Heatmap for Missing Values*

```
# Create a heatmap grid highlighting missing values percentage
plt.figure(figsize=(8, 6))
sns.heatmap(missing_info[['missing_percentage']], cmap='YlGnBu', annot=True, fmt='.2f', cbar_kws={'label': 'Missing Percentage'})
plt.title('Missing Values Percentage Heatmap')
plt.show()
```

- **Heatmap Grid**: Visualizes the missing percentages using shades of color (Yl GnBu colormap).
- **Annotations**: Displays numeric values of missing percentages in each cell with two decimal points (fmt='.2f').



Time Based Analysis:

12.1: *Convert 'open' and 'close' Columns to Datetime*

```
# Time Based Analysis
# Convert 'open' and 'close' columns to datetime
df['open'] = pd.to_datetime(df['open'])
df['close'] = pd.to_datetime(df['close'])
```

- Converts the open and close columns to Pandas datetime format for time-based operations.

12.2: *Calculate Duration of Charging Station Operation*

```
# Calculate the time duration for which charging stations are open
df['duration'] = df['close'] - df['open']
```

- Calculates the duration for which each charging station is open by subtracting the open time from the close time.

12.3*: Display Duration Data*

```
# Display the first few rows of the DataFrame with the calculated duration
print(df[['open', 'close', 'duration']].head())
```

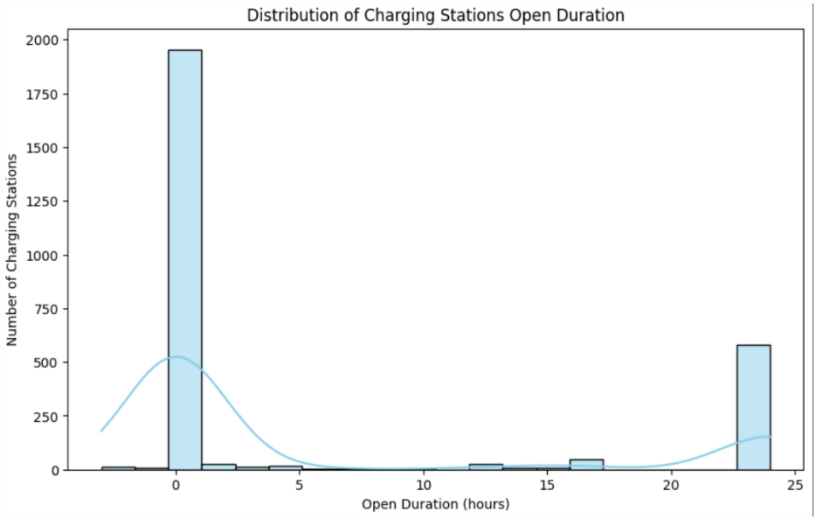
- Displays the first few rows of the Data Frame showing the open, close, and duration columns.

12.4: *Plot Distribution of Open Durations*

```
# Plot the distribution of time durations
plt.figure(figsize=(10, 6))
sns.histplot(df['duration'].dt.total_seconds() / 3600, bins=20, kde=True, color='skyblue')
plt.title('Distribution of Charging Stations Open Duration')
plt.xlabel('Open Duration (hours)')
plt.ylabel('Number of Charging Stations')
plt.show()
```

- **Histogram**: Visualizes the distribution of charging station open durations (converted to hours) using a histogram and kernel density estimation (KDE).
- **Labels and Title**: Adds appropriate labels and a title to the plot for clarity.

	open	close	duration
0	2024-12-10	2024-12-10 23:59:59	0 days 23:59:59
1	2024-12-10	2024-12-10 23:59:59	0 days 23:59:59
2	2024-12-10	2024-12-10 23:59:59	0 days 23:59:59
3	2024-12-10	2024-12-10 23:59:59	0 days 23:59:59
4	2024-12-10	2024-12-10 23:59:59	0 days 23:59:59



Capacity Analysis:

13.1**: *Count Occurrences of Each Capacity Value***

```
# Capacity Analysis

capacity_counts = df['capacity'].value_counts()
```

- Counts the occurrences of each unique value in the capacity column (e.g., different charging station capacities in kW).

13.2**:** ***Apply Logarithm Transformation***

```
# Apply logarithm transformation to count values
log_counts = np.log1p(capacity_counts)
```

- Applies a logarithmic transformation (log1p) to the capacity counts to reduce skewness and better visualize variations in the data.

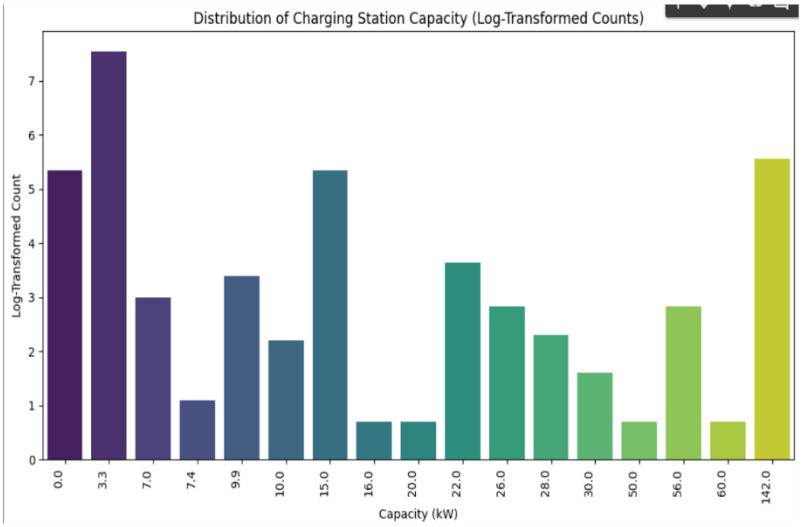
13.3**:** ***Plot Log-Transformed Capacity Distribution***

```
# Plot the distribution of charging station capacity with log-transformed counts
plt.figure(figsize=(10, 6))
sns.barplot(x=log_counts.index, y=log_counts.values, palette='viridis')
plt.title('Distribution of Charging Station Capacity (Log-Transformed Counts)')
plt.xlabel('Capacity (kW)')
plt.ylabel('Log-Transformed Count')
plt.xticks(rotation=90, ha='right')
```

- **Bar plot**: Plots a bar chart with log-transformed counts of charging station capacities.
- **Labels**: Adds titles and labels to the plot for clarity.
- **Rotation**: Rotates the x-axis labels to make them readable.

13.4***: Adjust Layout***

- Adjusts the plot layout for better visualization and displays the chart.



Zone Wise Analysis

14.1*: Count Occurrences of Each Zone*

```
# Zone Wise Analysis

# Calculate the counts for each zone
zone_counts = df['zone'].value_counts()
```

- Counts the occurrences of each unique value in the zone column (i.e., the number of charging stations in each zone).

14.2*: Sort Zones by Count*

```
# Sort the zones based on the counts
sorted_zones = zone_counts.sort_values(ascending=False).index
```

- Sorts the zones based on the counts in descending order to prioritize the zones with the most charging stations.

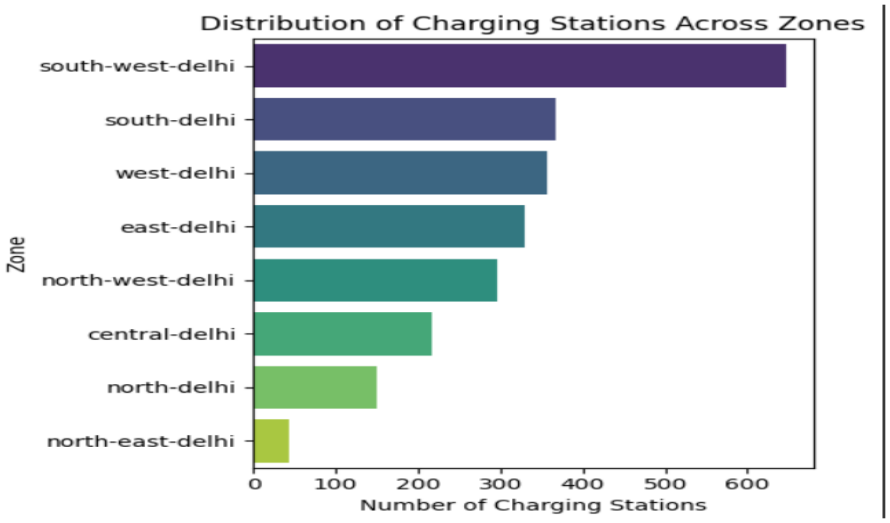
14.3**:** *Plot Distribution of Charging Stations Across Zones*

```
# Create a horizontal bar plot for the distribution of charging stations across different zones
plt.figure(figsize=(5, 5))
sns.countplot(y='zone', data=df, order=sorted_zones, palette='viridis')
plt.title('Distribution of Charging Stations Across Zones')
plt.ylabel('Zone')
plt.xlabel('Number of Charging Stations')
```

- **Count plot**: Creates a horizontal bar plot showing the distribution of charging stations across different zones, ordered by count.
- **Labels**: Adds titles and axis labels for clarity.

14.4*: Adjust Layout*

- Adjusts the plot layout to prevent overlap and displays the plot.



Pricing Strategy

- Base models: ₹7-10 lakhs (accessible to middle-class families).
- Premium models: ₹15-20 lakhs (targeting high-income groups).

Infrastructure and Collaboration

- Establish charging networks in Tier-1 and Tier-2 cities.
- Collaborate with renewable energy providers for sustainable charging solutions.

Data-Driven Decision Making

- Use real-time customer feedback to refine offerings.
- Develop predictive models to anticipate market trends.

In the Event of Data Unavailability

- Conduct **primary research** via surveys and interviews.
- Utilize proxy data from similar emerging markets.
- Apply **market simulation models** to estimate trends and behaviour.

Conclusion:

The Indian EV market presents significant opportunities, particularly in urban areas with existing infrastructure. By targeting eco-conscious, tech-savvy, and high-income consumers, the startup can effectively establish its foothold and grow as the market matures.