# EV Market Segmentation

**I have to analyse the Electric Vehicle market in India using Segmentation analysis and come up with a feasible strategy to enter the market, targeting the segments most likely to use Electric vehicles.**

By

Solo

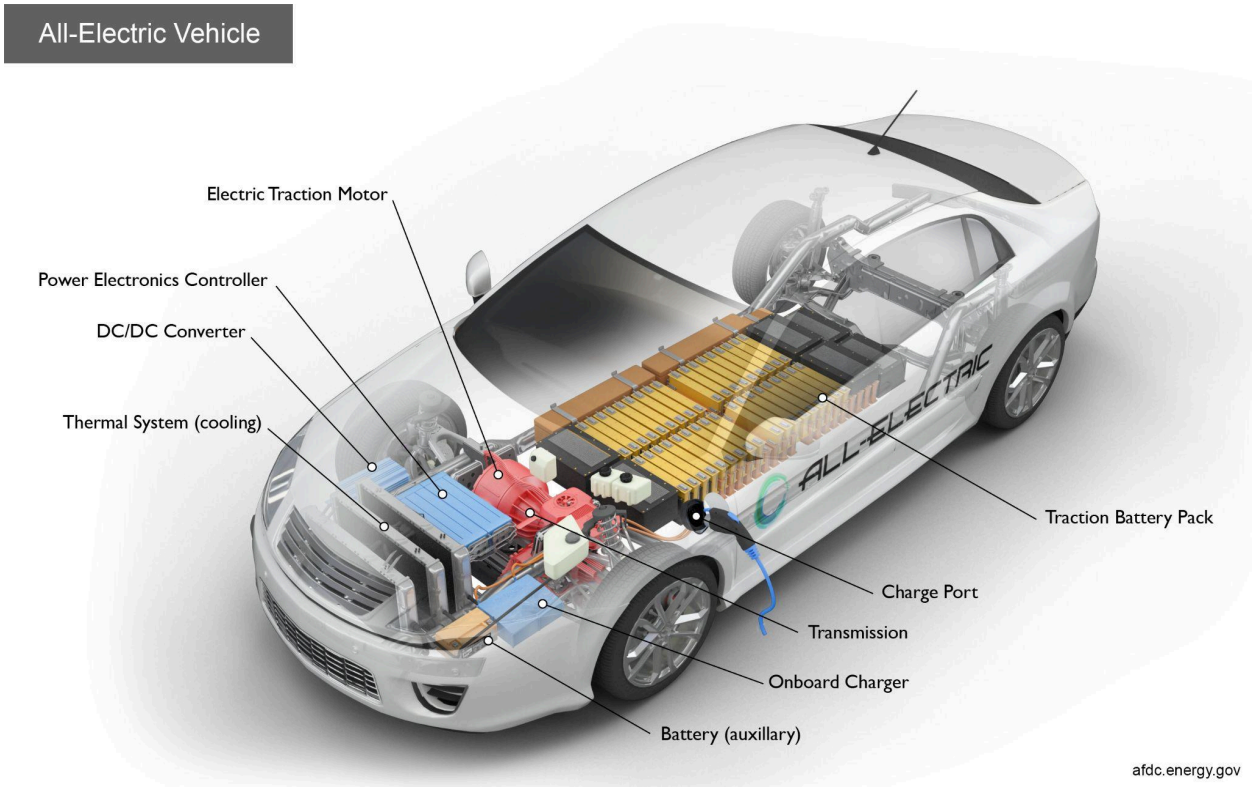
**V.Surya Vamsi**

**Problem Statement ( EV Market) :**

You are a team working under an Electric Vehicle Startup. The Startup is still deciding in which vehicle/customer space it will be develop its EVs. You have to analyse the Electric Vehicle market in India using Segmentation analysis and come up with a feasible strategy to enter the market, targeting the segments most likely to use Electric vehicles.

(CUSTOMER/VEHICLE/B2B) SEGMENTS: Apart from Geographic, Demographic, Psychographic, Behavioral segments, teams can consider different CATEGORY of Segments for the Segmentation Tasks, based on AVAILABILITY OF DATA. Market Segmentation comes with wide scope of possibility and Segments created can change based on different datasets collected.

DO NOTE that not every MARKET has Geographic, Demographic, Psychographic, Behavioral data available easily and there is going to :be lot of research required in DATA Collection Tasks.



1. **I am going to analysis EV cars, Collecting  data sets to analyse the market strategies and represent visually to show graphs by using python code language.**
2. **startup is analyzing the market for both EV bikes and EV cars. You'll need to assess customer segments for both types of vehicles and develop a strategy for entering the market, targeting the most profitable segments.**

>>> EV Cars Market Segmentation Project.
**# Importing Libraries:(these libraries are used to manipulating data , visualization , preprocessing data etc.**

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

## Step -1 Data collecting:

**Mock Data**: Create a dataset of EV cars with attributes like price, range, charging stations, top speed, and customer ratings.
**Real Data**: Replace mock data with actual EV market data collected from online sources, government reports, or EV manufacturers.

Look for free, publicly available datasets on trusted platforms:

- **Kaggle**: Search for EV-related datasets on Kaggle. Link: https://www.kaggle.com/
- Data set link (Ev Market Data): https://www.kaggle.com/datasets/patricklford/global-ev-sales-2010-2024
- Data set link (General type Vehicle https://www.kaggle.com/code/yossefazam/car-price-prediction-with-75?scriptVersionId=160664868&cellId=1
- Data set link (Vehicle market data):

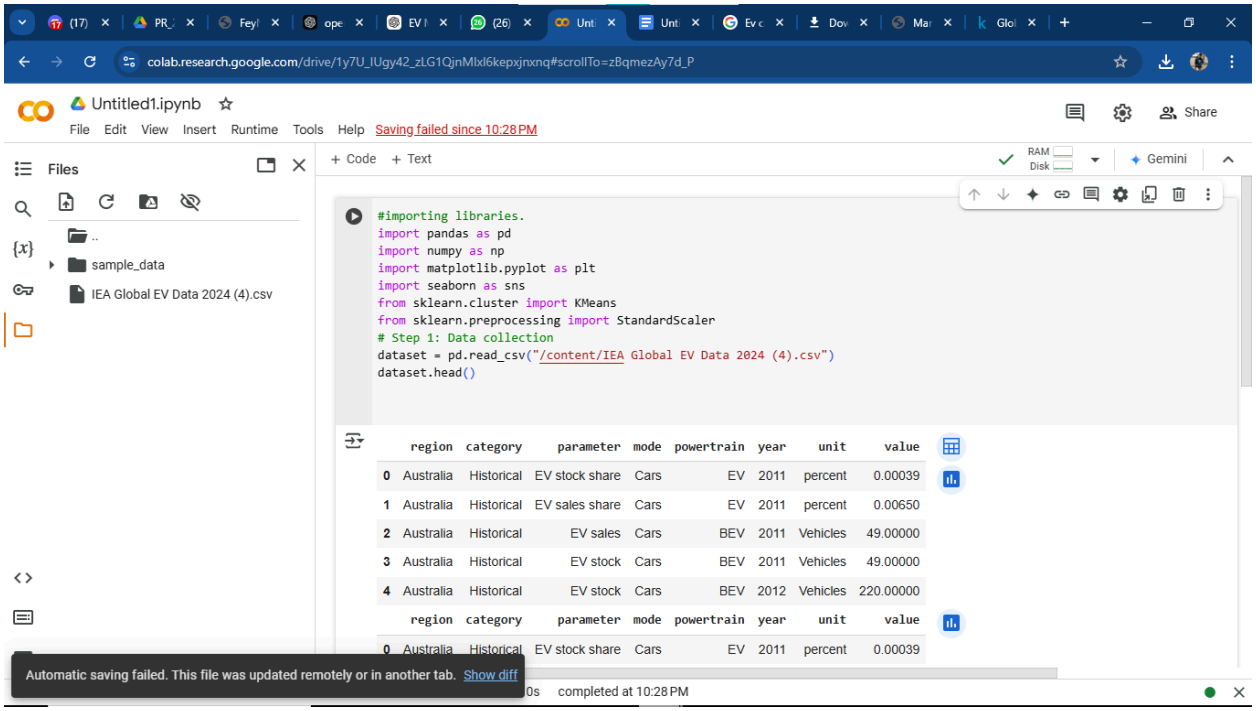
  https://www.kaggle.com/code/vencerlanz09/electric-cars-eda-with-feature-engineering?scriptVersionId=127869315&cellId=1

. Data set link(Charging Station):

https://www.kaggle.com/code/amitvkulkarni/ev-charging-eda-insights?scriptVersionId=158920465&cellId=1 (charging stations data)

. **Data set link( Vehicle usage statistics cities):**

https://www.kaggle.com/code/chiragmakhija2601/electric-vehicle-dataset-analysis?scriptVersionId=187261725&cellId=1



An also converting data frame.

**Web Scraping:**

Use web scraping techniques to collect data from online sources:

3. **Car Review Sites**: Scrape specifications, prices, and reviews from sites like Edmunds or CarDekho.
4. **E-commerce for EVs**: Sites like Amazon (for EV accessories) or dedicated EV platforms.
5. **Tools for Scraping**: Python libraries like `BeautifulSoup` and `Selenium`

## Step :2 Data preprocessing:

1.Import necessary libraries like `pandas`, `numpy`, `sklearn`, and visualization tools like `matplotlib` and `seaborn`.

2.Scale the numeric features (e.g., price, range) using `StandardScaler` to ensure they contribute equally to clustering.

3.Before analyzing, clean and preprocess the data to ensure quality and consistency.

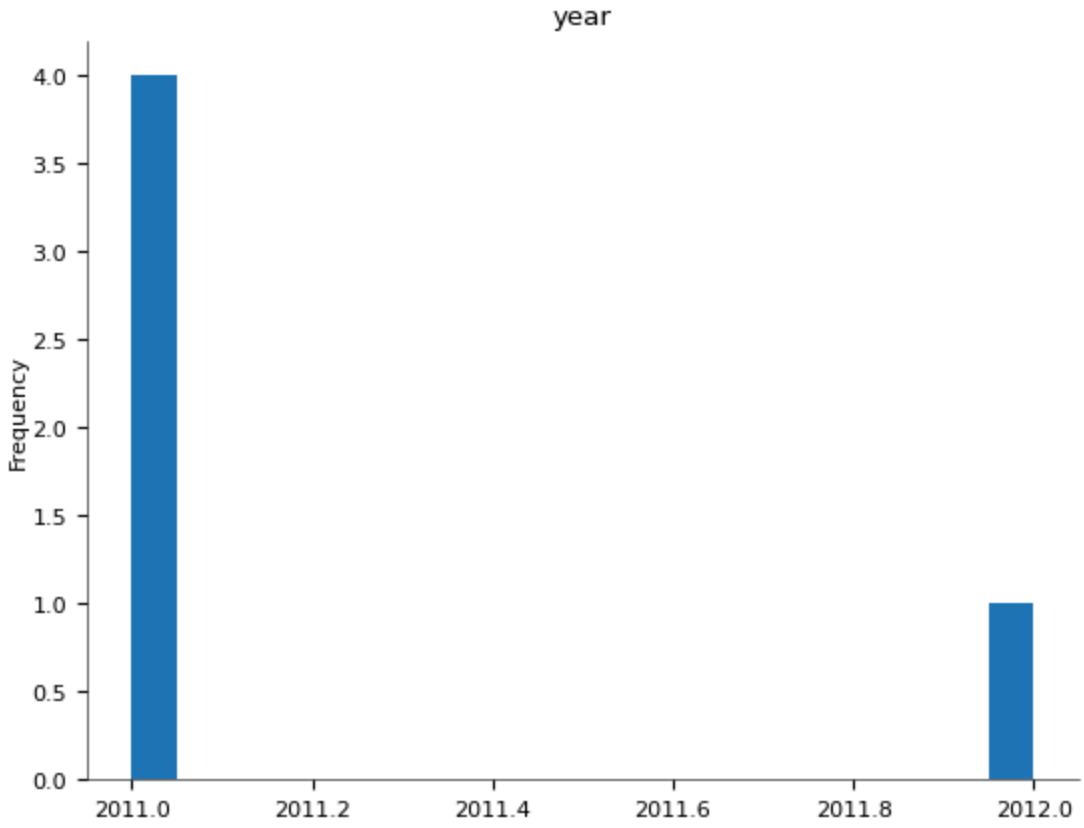
**Data Preprocessing Steps:**

1. **Remove Missing Data:** Handle missing values using imputation or by removing incomplete records.
2. **Normalize/Scale Data:** Ensure numerical data is standardized (e.g., scaling income levels, population).
3. **Categorize Data:** Convert categorical data (e.g., cities, professions) into numerical formats using encoding techniques.
4. **Feature Engineering:** Create new features that may be useful, such as "distance to nearest charging station."

# Scaling numerical features

scaler = StandardScaler()

ev_data_scaled = scaler.fit_transform(ev_data[[“index”, “region”, “Category”,parameter”, “mode”, “Power train”,” Year “, “unit”, “value”]])




This bar chart is distributions

1. I Can find outing the year of frequency..

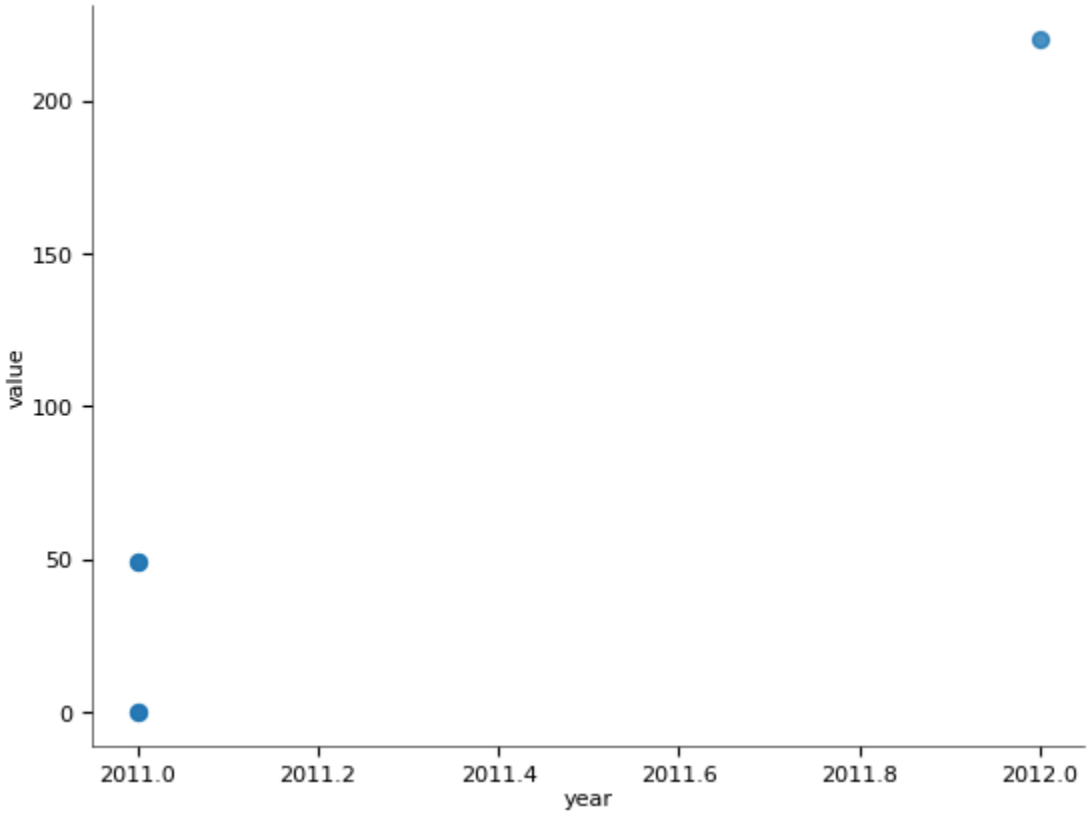
## 1. Geographic Analysis:

Determine which locations in India are most suitable for early market adoption based on the **Technology Adoption Life Cycle**:

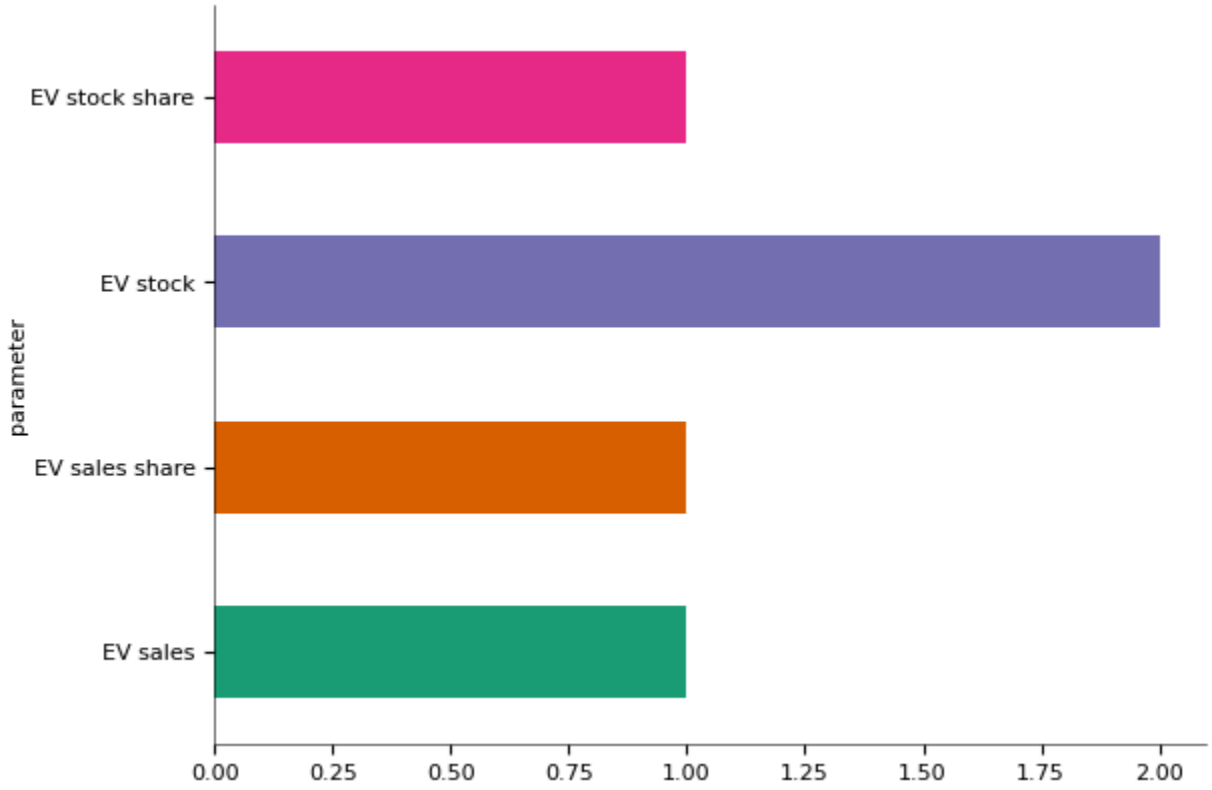
Focus on **urban areas** with:

- High population density.
- Existing EV infrastructure (charging stations).
- Favorable government policies (subsidies, tax incentives).

## This chart is second distribution chart



In this chart focusing on categorical data




**Collect the necessary data** to back up your segmentation analysis. You can gather data from a variety of sources to ensure robustness.

**Data Collection Categories:**

- **Geographic Data:** Identify regions in India with high vehicle usage, pollution levels, and interest in sustainable solutions.
- **Demographic Data:** Collect data on age groups, income levels, and occupations to identify target demographics for EV adoption.
- **Psychographic Data:** Look for data on people's interest in sustainability, eco-friendly products, and early adoption trends.
- **Behavioral Data:** Gather data on driving habits, travel frequency, willingness to switch to EVs, and interest in EV technology.
- **Charging Stations Data:** Information on the current availability and planned locations of EV charging stations.

Demographic Demographic segmentation groups customers and potential customers together by focusing on certain traits such as age, gender, income, occupation & family status. ... There are plenty of ways to segment markets using demographics. The most

commonly used demographic segmentation factors are: Age.Age, Gender, Education, Ethnicity, Income, Employment, hobbies, etc.

Pros of demographic segmentation 1. Easy to nd: It is accessible and easy to use. For example, government censuses are available in most countries. 2. Identify potential market: When an organization looks at the demographic segmentation, it focuses on the people who are most likely to buy a product. This helps in identifying the target market. 3. Marketing: Demographic customer segmentation helps organizations in developing market outreach for better marketing strategies. Cons of demographic segmentation 1. Assumptions: Demographic segmentation is based on the assumption that consumers in the same demographic group will have similar needs. For example; not everyone in their 30s will have the same needs when it comes to buying a mobile phone. Some might want a high-resolution camera, others might want storage capabilities, better speakers, etc

2. Constant changes: Behavior is never constant. Marketers cannot collect demographic data once, and use that same information for years. Because the population changes. Census data is updated every year. This information must be collected constantly to get a real scenario

.**Geographic** It is a component that competently complements a marketing strategy to target products or services on the basis of where their consumers reside. Division in terms of countries, states, regions, cities, colleges or Areas is done to understand the audience and market a product/service accordingly

## Advantages of Geographic Segmentation

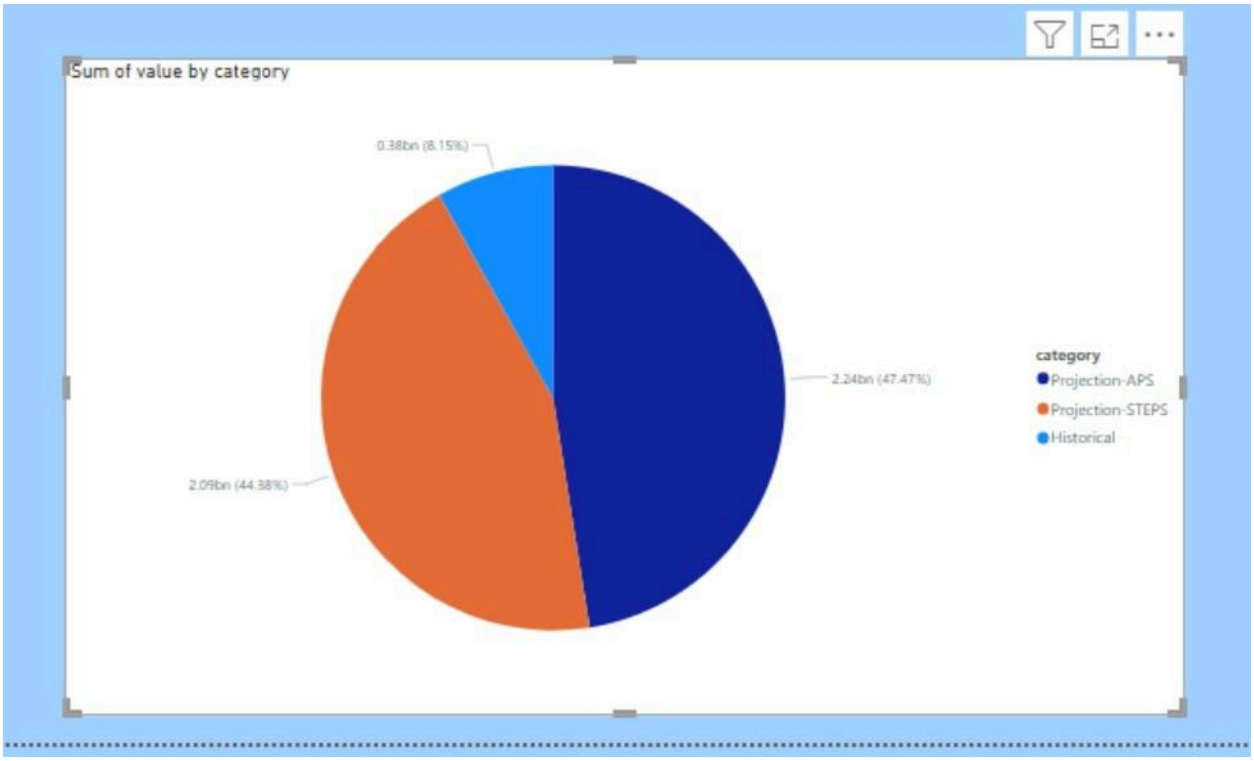
● Enhanced focus due to targeting: Geographic segmentation is an effective method to improve focus on target audience. As a division based on geographical characteristics is involved, organizations tend to create more focused marketing strategies to convert local consumers into successful customers.

● Immediate market growth: In situations where an organization has a marketing strategy for a particular location, it becomes additionally convenient for this organization to apply the same strategy to neighboring locations which demonstrate similar geographical characteristics. Expanding marketing operations and developing corresponding strategies for locations with unknown characteristics is much more time and resource consuming than expanding to locations that indicate traits similar to the existent target market

. ● Improved communication: As targeting is based on geography and the traits that change with a change in geographies, marketing and promotional communications for

local audiences need to be according to the specialized nature of this geographical segment. Better communication happens when there's clarity in regards to what the audience expects out of a product/service. ● Increase prots: Geographical segmentation gives an organization an essential early competitive edge in localized markets, increases brand recall value and also helps in providing better customer service which in turn leads to better customer retention rates. For organizations that have limited reach, geographic segmentation is a strategy to focus their resources on accurate target audiences and receive better

## Pie chart it represents( sum value by category):

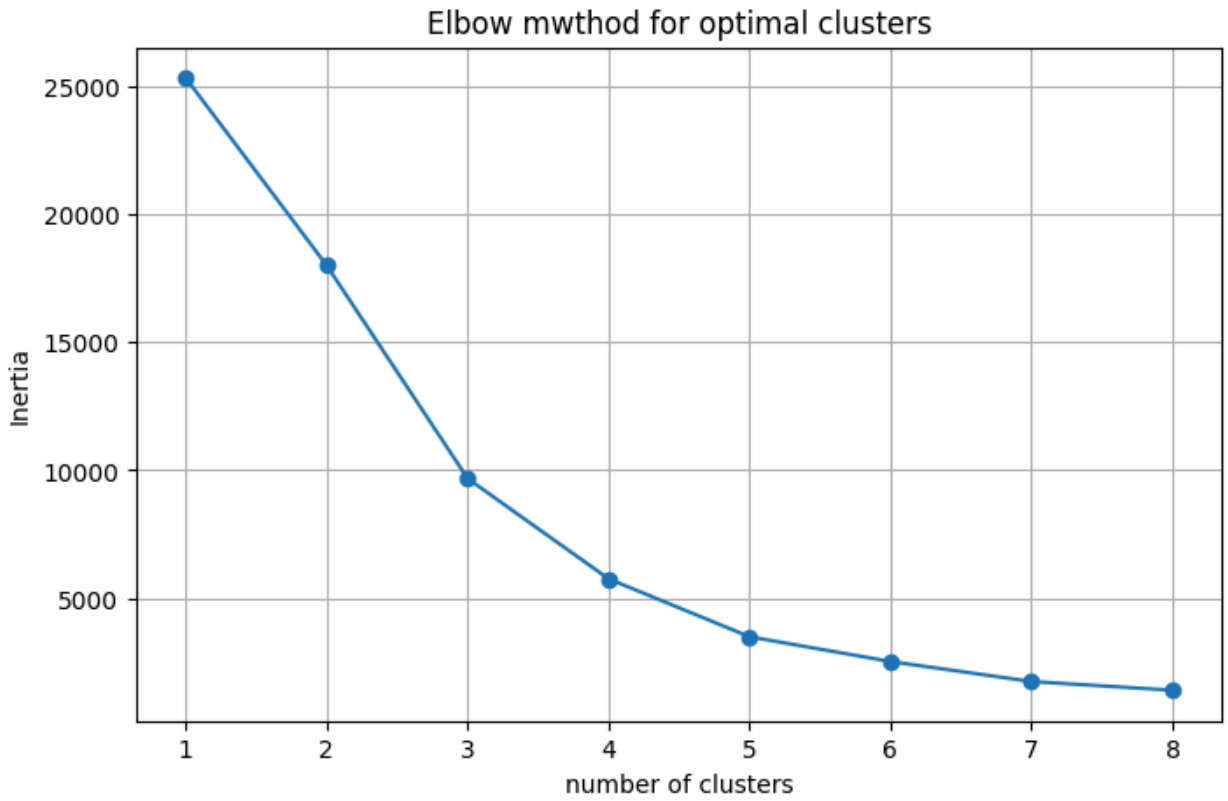



### Step 3: Determine Optimal Number of Clusters (Elbow Method):

1. Use the Elbow Method to determine the optimal number of clusters for segmentation.
    - Compute the inertia (sum of squared distances to the cluster center) for different k values (e.g., 1–10).
    - Plot an Elbow Curve to visualize the optimal cluster count.

---

```
inertia = []

k_values = range(1,9)

for k in k_values:

  kmeans = KMeans(n_clusters=k, random_state=42)

  kmeans.fit(ev_data_scaled)

  inertia.append(kmeans.inertia_)

#pllot the elbow curve

plt.figure(figsize=(8,5))

plt.plot(k_values, inertia, marker='o')

plt.title('Elbow mwthod for optimal clusters')

plt.xlabel('number of clusters')

plt.ylabel('Inertia')

plt.grid(True)

plt.show()
```




**Apply Clustering**

1. **Use K-Means Clustering with the determined number of clusters (e.g., 4 clusters).**
2. **Assign each EV car data point to a specific cluster.**

```
#Apply K-Means Clustering

optimal_clusters = 4

kmeans = KMeans(n_clusters=optimal_clusters, random_state=42)

ev_data['Cluster'] = kmeans.fit_predict(ev_data_scaled)

Analyze and Profile Clusters

# Cluster-wise Mean Values

cluster_profiles = ev_data.groupby('Cluster').mean()
```

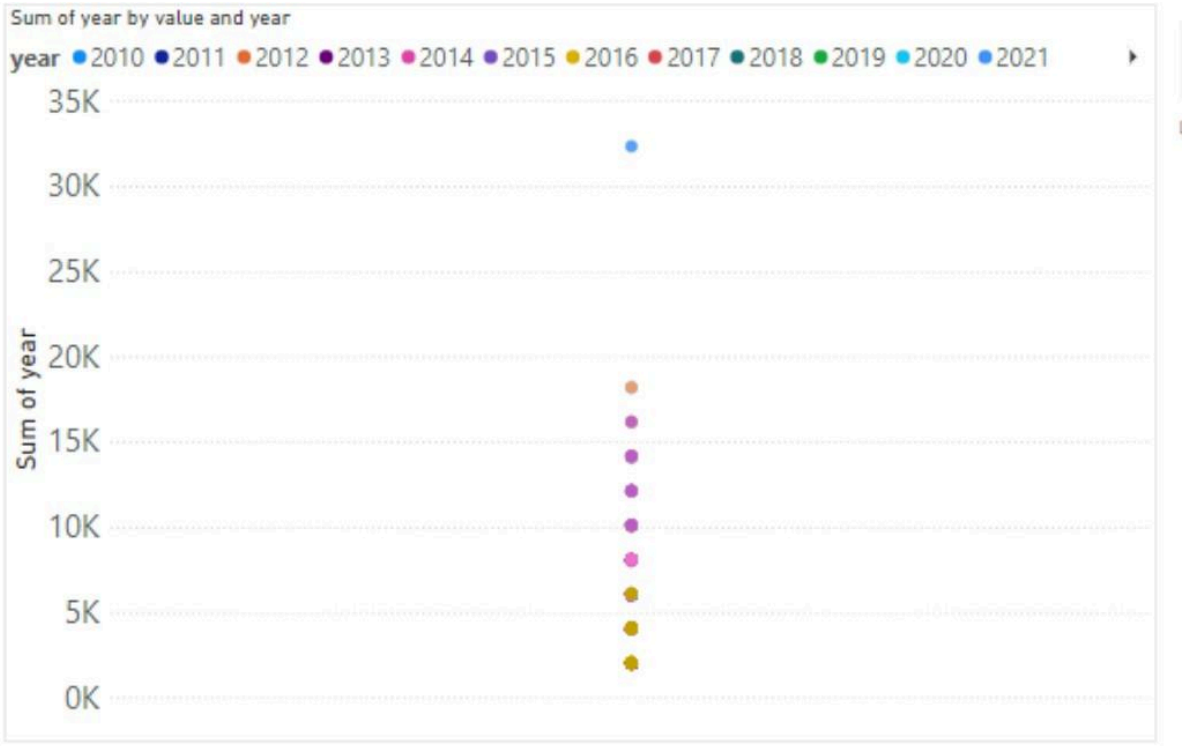
```
print("Cluster Profiles:")

print(cluster_profiles)
```

**Step 4 Visualization:**

```
# Step 6: Visualizations

# Scatter Plot of Clusters

plt.figure(figsize=(10, 6))

sns.scatterplot(x=ev_data['Price'], y=ev_data['Range_km'], hue=ev_data['Cluster'],
palette='viridis', s=100)

plt.title('Price vs Range by Cluster')

plt.xlabel('Price (in INR)')

plt.ylabel('Range (in km)')

plt.legend(title='Cluster')

plt.grid(True)

plt.show()


# Bar Plot for Cluster Profiles

cluster_profiles[['Price', 'Range_km']].plot(kind='bar', figsize=(10, 6))

plt.title('Cluster Profiles: Price and Range')
```

```
plt.ylabel('Average Value')

plt.grid(axis='y')

plt.show()
```

## Scattering plot chart:




---

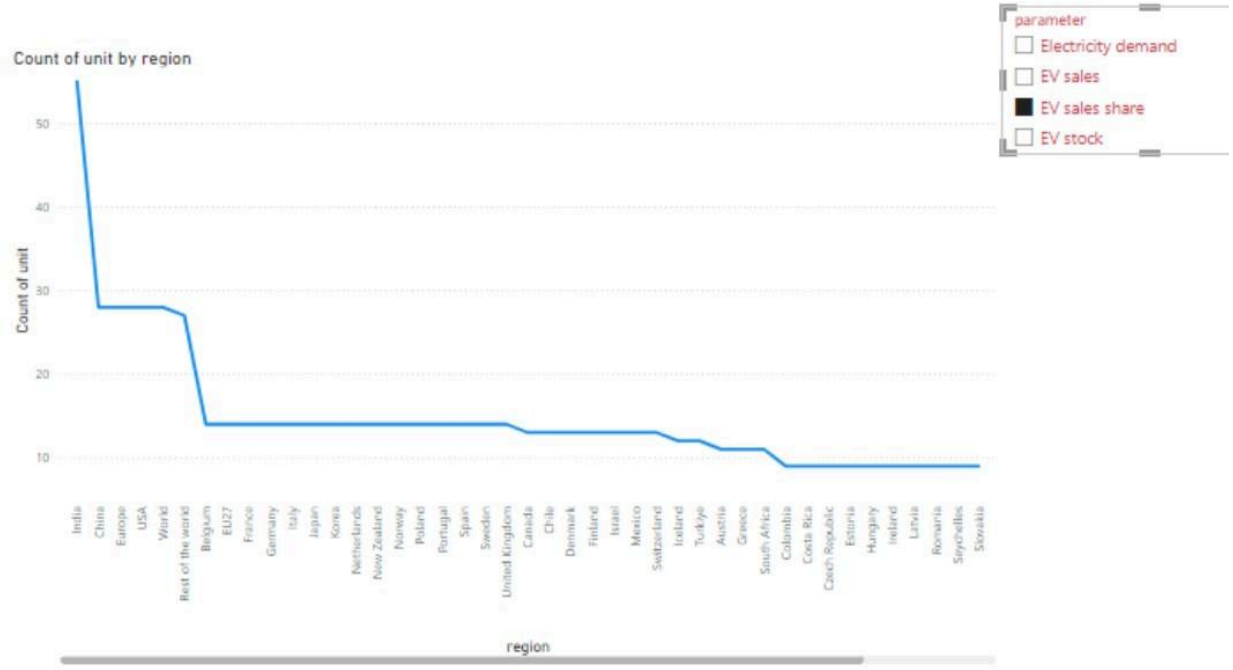
## >>In this visualization line charts through power Bi using to create :

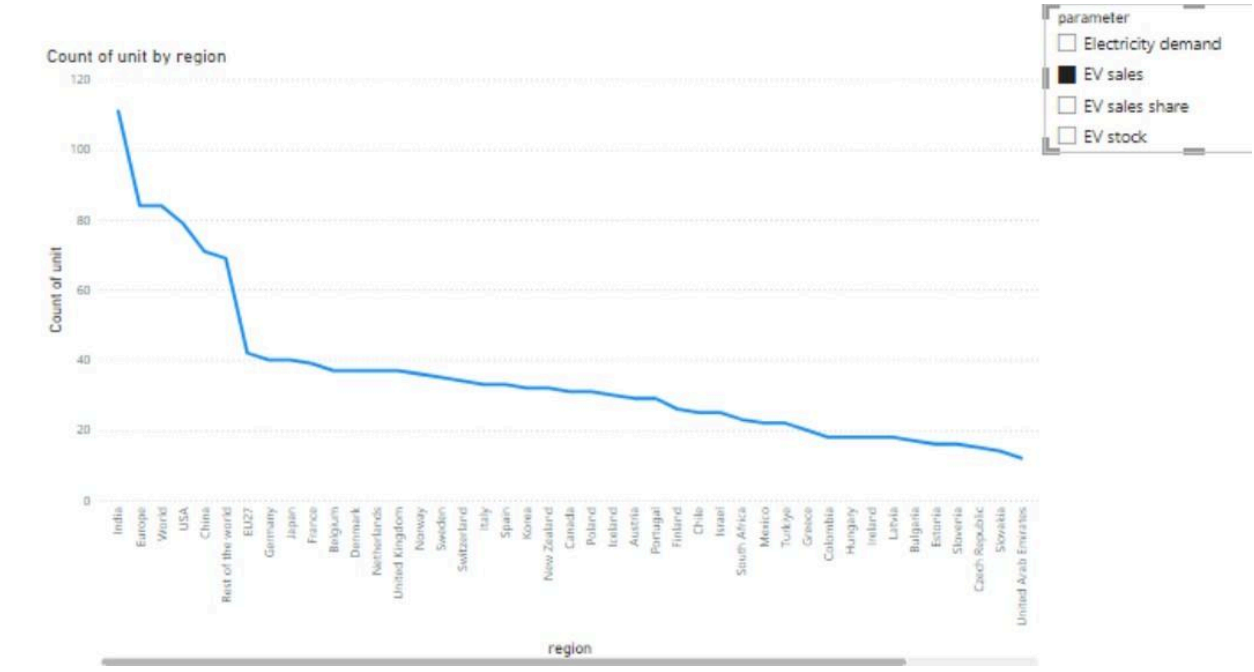
1.In this line charts iam using some data columns Like ( category, values, brand)

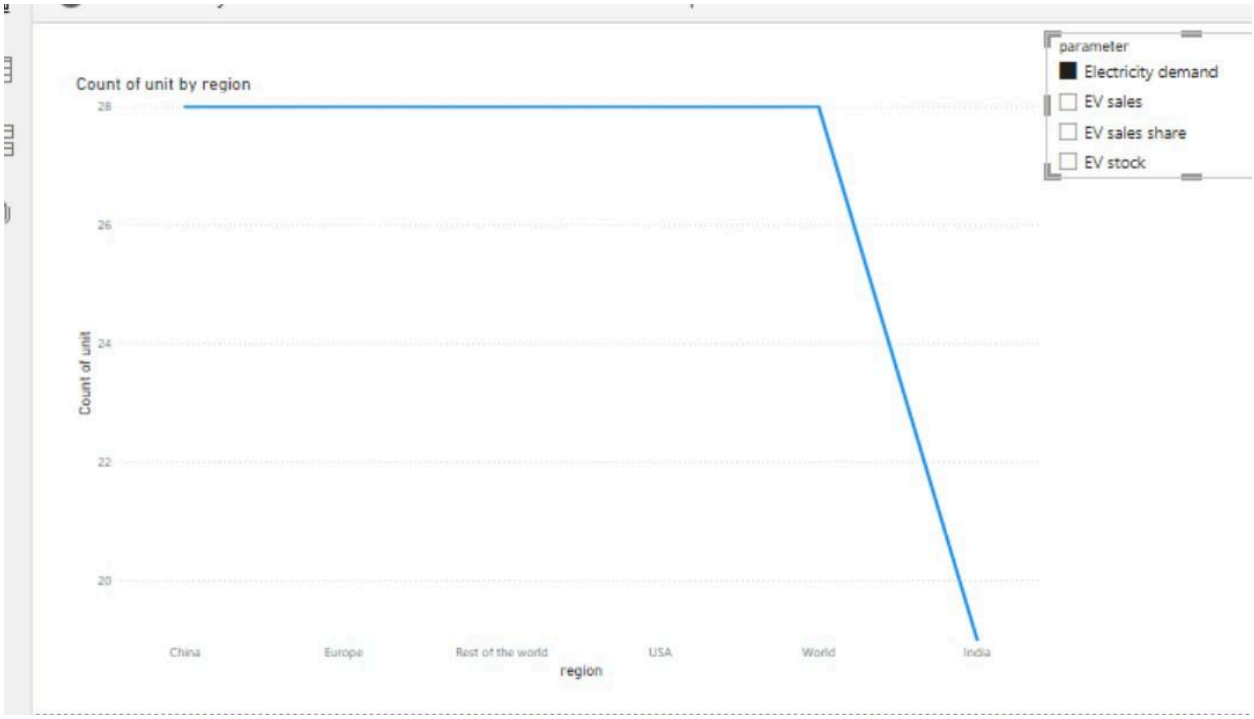
**This line charts  are visualizing( Ev Stock chart), (Ev Share sales),**

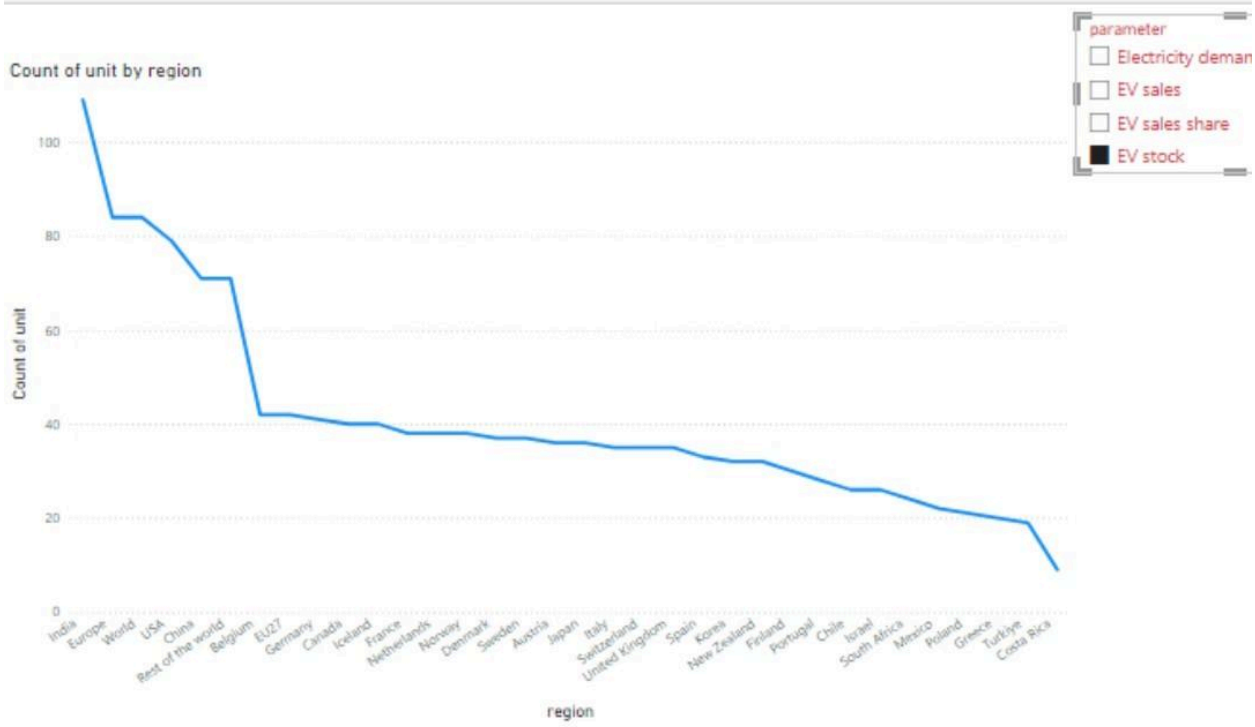
**(Ev sales), (Ev Electricity demeand):**

A line chart in Power BI is a visual representation of data points connected by lines, primarily used to show trends and changes over time, where the x-axis typically represents time and the y-axis displays the measured value, allowing users to easily identify patterns, increases, decreases, and fluctuations within a dataset.




Count of unit by region
parameter
Electricity demand
EV sales
EV sales share
EV stock
Count of unit
120
100
80
60
40
20
0
India
Europe
World
USA
China
Rest of the world
EU27
Germany
Japan
France
Belgium
Denmark
Netherlands
United Kingdom
Norway
Sweden
Switzerland
Italy
Spain
Korea
New Zealand
Canada
Poland
Iceland
Austria
Portugal
Finland
Chile
Israel
South Africa
Mexico
Turkye
Greece
Colombia
Hungary
Ireland
Latvia
Bulgaria
Estonia
Slovenia
Czech Republic
Slovakia
United Arab Emirates
region


Count of unit by region
parameter
Electricity demand
EV sales
EV sales share
EV stock
Count of unit
28
26
24
22
20
China
Europe
Rest of the world
USA
World
India
region


Count of unit by region
parameter
Electricity deman
EV sales
EV sales share
EV stock
Count of unit
100
80
60
40
20
0
India
Europe
World
USA
China
Rest of the world
Belgium
EU27
Germany
Canada
Iceland
France
Netherlands
Norway
Denmark
Sweden
Austria
Japan
Italy
Switzerland
United Kingdom
Spain
Korea
New Zealand
Finland
Portugal
Chile
Israel
South Africa
Mexico
Poland
Greece
Turkiye
Costa Rica
region

# Problem statement(Online vehicle Booking Market):

Here is a step-by-step explanation of how to prepare a project for **Online Cab Booking Statistics** using Python. We'll replace the EV car context with cab booking market analysis.

---

## Project Steps: Online Cab Booking Market Analysis

---

#### Step 1: Define Project Objective

- The goal is to analyze the online cab booking market in India and identify potential segments using clustering techniques.
- Key focus: Pricing, demand, location preferences, and customer behavior to improve targeting.

---

#### Step 2: Data Collection

1. **Gather Relevant Data:**
    - Use mock data or collect datasets from public sources like government transportation data, online booking portals, or Kaggle.
2. **Suggested Features:**
    - Trip Fare, Distance, Booking Time, Waiting Time, Ratings, Number of Trips, City/Region, Peak Hours.

---

#### Step 3: Data Preprocessing

1. **Clean the Data:**
    - Handle missing values, duplicates, and inconsistent entries.
2. **Standardize Features:**
    - Normalize numerical features (e.g., trip fare, distance)

---

#### Step 4: Determine Optimal Number of Clusters

1. Apply the **Elbow Method**:

- Plot the Elbow Curve to identify the optimal number of clusters (where the curve bends).

---

**Step 5: Apply Clustering**

1. Perform K-Means Clustering:
   - Cluster the data into meaningful groups based on attributes like trip fare, distance, and peak hours.
2. Assign a cluster label to each data point.

---

**Step 6: Analyze and Profile Clusters**

1. **Cluster Summary:**
   - Compute average values for features (e.g., average fare, average distance).
   - Understand the characteristics of each cluster (e.g., budget rides, long-distance peak-hour rides).
2. **Choose Target Segment:**
   - Identify the cluster that aligns with business goals (e.g., high-profit rides, frequent rides).

---

**Step 7: Visualizations**

1. **Scatter Plot:**
   - Plot relationships like trip fare vs. distance with clusters highlighted.
2. **Bar Plot:**
   - Compare average fares, distances, or trip counts across clusters.

---

**Step 8: Market Analysis**

1. Estimate Potential Market:
   - Use Fermi estimation to calculate the number of potential customers and revenue for the target segment.
2. Compute Profit:

- Multiply potential customers by the average fare of the target cluster.

---

**Step 9: Save Results**

1. Export the clustered dataset to a CSV file for further analysis.
2. Ensure all visuals and findings are documented.

---

**Step 10: Insights and Recommendations**

1. **Provide Key Insights:**
   - Highlight the target segment's characteristics (e.g., affordable rides during peak hours).
2. **Recommendations:**
   - Tailor marketing strategies and pricing models based on cluster behavior.

```
# Online Cab Booking Market Analysis Project


# Importing Libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler


# Step 1: Data Collection (Mock Data)

data = {

    'City': ['Delhi', 'Mumbai', 'Bangalore', 'Hyderabad', 'Chennai',

             'Kolkata', 'Pune', 'Ahmedabad', 'Jaipur', 'Surat'],
```

```
    'Trip_Fare': [250, 300, 200, 350, 150, 220, 270, 180, 260, 310],
    'Distance_km': [10, 15, 8, 20, 6, 12, 14, 7, 10, 16],
    'Peak_Hours_Trips': [20, 25, 18, 30, 12, 22, 24, 15, 20, 28],
    'Customer_Ratings': [4.5, 4.8, 4.6, 4.7, 4.4, 4.5, 4.7, 4.3, 4.6, 4.8]
}

# Create a DataFrame
cab_data = pd.DataFrame(data)

# Step 2: Data Preprocessing
# Scaling numerical features
scaler = StandardScaler()
cab_data_scaled = scaler.fit_transform(cab_data[['Trip_Fare', 'Distance_km',
'Peak_Hours_Trips', 'Customer_Ratings']])

# Step 3: Determine Optimal Number of Clusters (Elbow Method)
inertia = []
k_values = range(1, 11)
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(cab_data_scaled)
    inertia.append(kmeans.inertia_)

# Plot the Elbow Curve
```

```
plt.figure(figsize=(8, 5))
plt.plot(k_values, inertia, marker='o')
plt.title('Elbow Method for Optimal Clusters')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.grid(True)
plt.show()

# Step 4: Apply K-Means Clustering
optimal_clusters = 3  # Choose based on the elbow curve
kmeans = KMeans(n_clusters=optimal_clusters, random_state=42)
cab_data['Cluster'] = kmeans.fit_predict(cab_data_scaled)

# Step 5: Analyze and Profile Clusters
# Cluster-wise Mean Values
cluster_profiles = cab_data.groupby('Cluster').mean()
print("Cluster Profiles:")
print(cluster_profiles)

# Step 6: Visualizations
# Scatter Plot of Clusters
plt.figure(figsize=(10, 6))
sns.scatterplot(x=cab_data['Trip_Fare'], y=cab_data['Distance_km'],
hue=cab_data['Cluster'], palette='viridis', s=100)
```

```
plt.title('Trip Fare vs Distance by Cluster')
plt.xlabel('Trip Fare (INR)')
plt.ylabel('Distance (km)')
plt.legend(title='Cluster')
plt.grid(True)
plt.show()

# Bar Plot for Cluster Profiles
cluster_profiles[['Trip_Fare', 'Distance_km']].plot(kind='bar', figsize=(10, 6))
plt.title('Cluster Profiles: Trip Fare and Distance')
plt.ylabel('Average Value')
plt.grid(axis='y')
plt.show()

# Step 7: Potential Market Analysis
# Assuming target segment is Cluster 1
target_cluster = 1
potential_customers = 10000  # Example number of potential customers
average_fare = cluster_profiles.loc[target_cluster, 'Trip_Fare']
potential_profit = potential_customers * average_fare
print(f"Potential Profit from Target Segment (Cluster {target_cluster}): INR
{potential_profit:,.2f}")

# Step 8: Summary
```

```
# Export the final DataFrame with clusters

cab_data.to_csv('cab_booking_clustered_data.csv', index=False)

print("Final clustered data has been saved as 'cab_booking_clustered_data.csv'.")
```