MACHINE LEARNING FOR EV STARTUP MARKET SEGMENTATION

J Suganya

29th April , 2024

Abstract

This study aims to conduct a segmentation analysis of the Indian Electric Vehicle (EV) market to identify promising target segments for EV adoption. India's EV landscape is dynamic, influenced by diverse demographics, infrastructure challenges, and evolving consumer preferences. The dataset used for analysis comprises geographic information on EV charging stations across India, providing insights into the distribution and accessibility of charging infrastructure. By identifying distinct market segments, this analysis seeks to uncover demographic and geographical clusters that exhibit high potential for embracing electric mobility. The outcomes of this analysis will guide decision-making for an Electric Vehicle Startup, assisting in formulating a targeted market entry strategy. By understanding and prioritizing segments with the greatest propensity for EV adoption, the startup can optimize resource allocation, tailor product offerings, and devise marketing strategies that resonate with the identified market segments. Ultimately, this approach aims to enhance the startup's competitiveness, maximize market penetration, and contribute to the sustainable growth of electric mobility in India's transportation ecosystem.

1.0 Introduction

Electric Vehicles (EVs) in India are gaining momentum due to government initiatives, infrastructure development, technological advancements, rising consumer awareness, and industry collaboration. These factors are driving the adoption of EVs as a sustainable transportation solution. Challenges include affordability, charging infrastructure accessibility, and regulatory consistency. Despite challenges, EVs hold promise for transforming India's transportation sector and contributing to a cleaner environment.

2.0 Context of the problem

In the rapidly evolving landscape of the Indian Electric Vehicle (EV) market, our Electric Vehicle Startup stands at a crucial juncture, aiming to strategically enter the market by identifying and targeting the most promising customer segments. Market segmentation is a pivotal tool in this endeavor, allowing us to divide the diverse market into distinct and manageable segments based on various criteria such as geographic, demographic, psychographic, and behavioral factors.
Apart from these traditional segmentation dimensions, we also have the opportunity to explore different categories of segments based on the availability of data.This segmentation analysis aims to delve into the complexities of the Indian EV market, leveraging data-driven insights to identify segments that are not only viable but also aligned with our strategic objectives and core competencies. By understanding the unique needs, preferences, and behaviors of each segment, we can tailor our product offerings, marketing strategies, and business models to maximize market penetration and long-term success.
Our goal is not only to enter the market but also to establish a strong foothold and contribute significantly to the growth and adoption of electric mobility in India. This requires a comprehensive understanding of the market dynamics, competition landscape, regulatory environment, and most importantly, the diverse spectrum of customer segments that drive the demand for electric vehicles. Through strategic segmentation analysis, we aim to pave the way for a successful market entry strategy that positions us as a leader in the Indian EV market while promoting sustainable transportation solutions for a greener future.

3.0 Business Objective

- Identify Optimal EV Type : Determine the ideal electric vehicle (EV) model for launch by analyzing market trends,consumer preferences , and technological feasibility.
- Segment Customer Base : Utilize Machine learing to identify distinct customer segments within the EV market based on demographics , behaviour and psychographics.

- Based on geographic ,our analysis focuses on understanding how geographical factors such as urbanization levels, infrastructure development, population density, and regulatory frameworks impact the demand for electric vehicles across different regions of India. By delving into the geographic nuances, we aim to identify the most promising geographic segments that are conducive to embracing electric mobility solutions.

3.0 Data Collection and Preprocessing:

To kickstart our EV startup's market segmentation analysis for the upcoming launch in India . I began by focusing on data acquisition . This involved extensive research across multiple online sources to gather pertinent and suitable data for our project . This thorough data gathering process forms the foundation for the next pivotal phase: highlighting the most lucrative segment to ensure a successful entry into India's dynamic and burgeoning EV market.

Resources used for research:

- https://data.gov.in/
- https://www.kaggle.com/
- https://www.nature.com/articles/s41597-024-02942-9
- https://www.statista.com/statistics/1395109/india-public-ev-charging-stations-by-top-states/
- https://www.kaggle.com/datasets/edsonmarin/historic-sales-of-electric-vehicles
- https://cea.nic.in/electric-vehicle-charging-reports/?lang=en

https://www.kaggle.com/datasets/saketpradhan/electric-vehicle-charging-stations-in-india

This dataset provides information about electric vehicle (EV) charging stations in India. The dataset contains 7 columns and 1547 rows .

- Name: The name of the charging station.
- State: The state in which the charging station is located.
- City: The city in which the charging station is situated.
- Address: The specific address or location of the charging station.
- Latitude: The latitude coordinates of the charging station's location.
- Longitude: The longitude coordinates of the charging station's location.
- Type: The type of charging station, which could indicate the charging technology or power rating (e.g., DC,AC charging station).

```
df=pd.read_csv('ev-charging-stations-india.csv')
df.head(5)
```

	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

4.0 Exploring data

- **Descriptive Statistics**

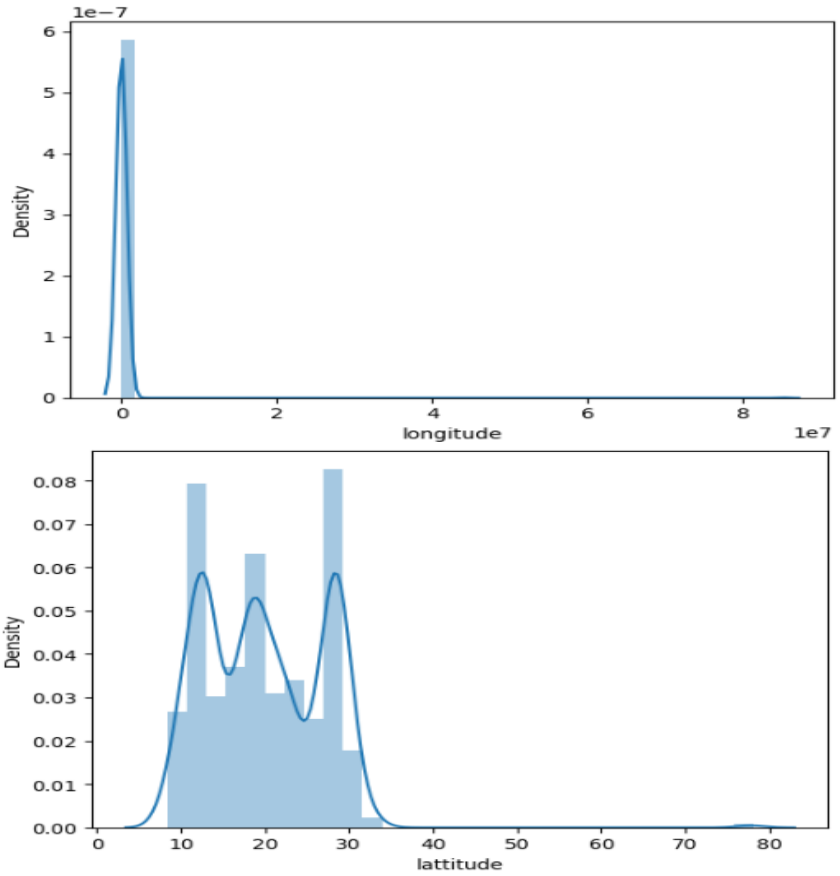
	lattitude	longitude	type
count	1541.000000	1.541000e+03	1539.000000
mean	19.979926	1.105323e+05	9.020793
std	7.125371	3.064996e+06	4.136436
min	8.390198	8.058454e+00	6.000000
25%	13.041390	7.562036e+01	7.000000
50%	19.106317	7.721257e+01	7.000000
75%	26.900894	7.847983e+01	11.000000
max	78.065400	8.510551e+07	24.000000

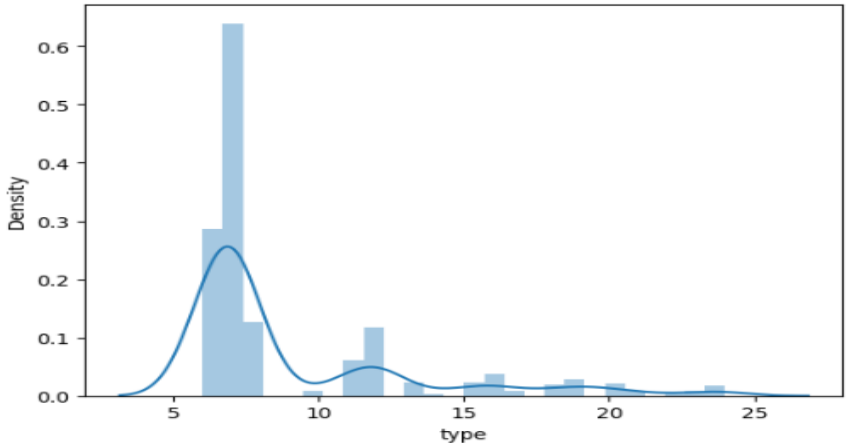
	name	state	city	address
count	1547	1547	1547	1507
unique	1144	60	362	1180
top	Tata Power	Maharashtra	Delhi	Outside Chelmsford Club/ Opposite CSIR Buildin...
freq	58	259	72	6

Here in this we could see the Common descriptive statistics include measures such as mean, median, mode, standard deviation, variance, range, and percentiles of the numeric and categorical columns.

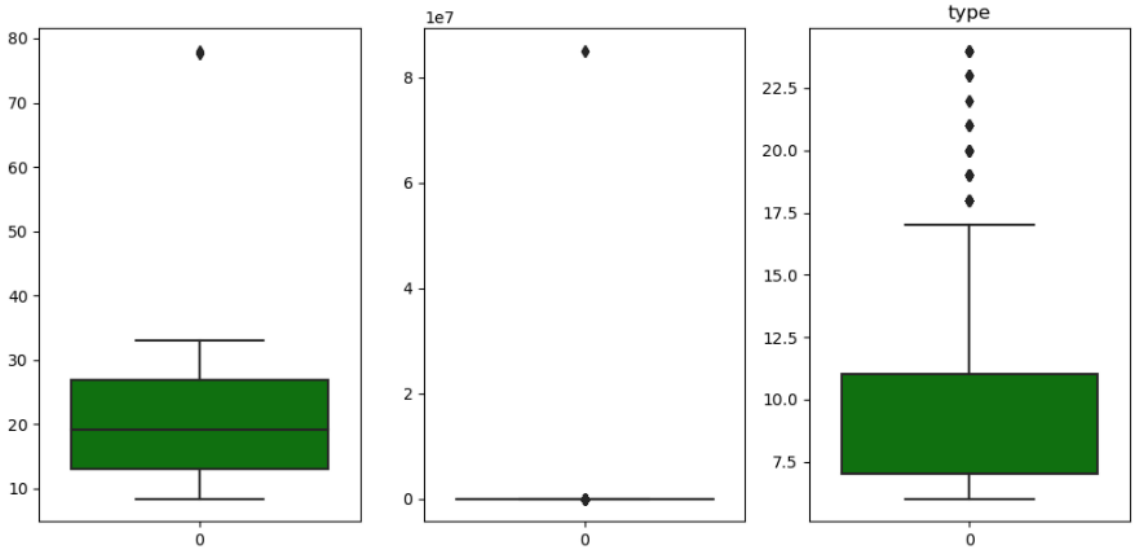
- ## Univariant Analysis

  visualize the distribution of all the numerical columns

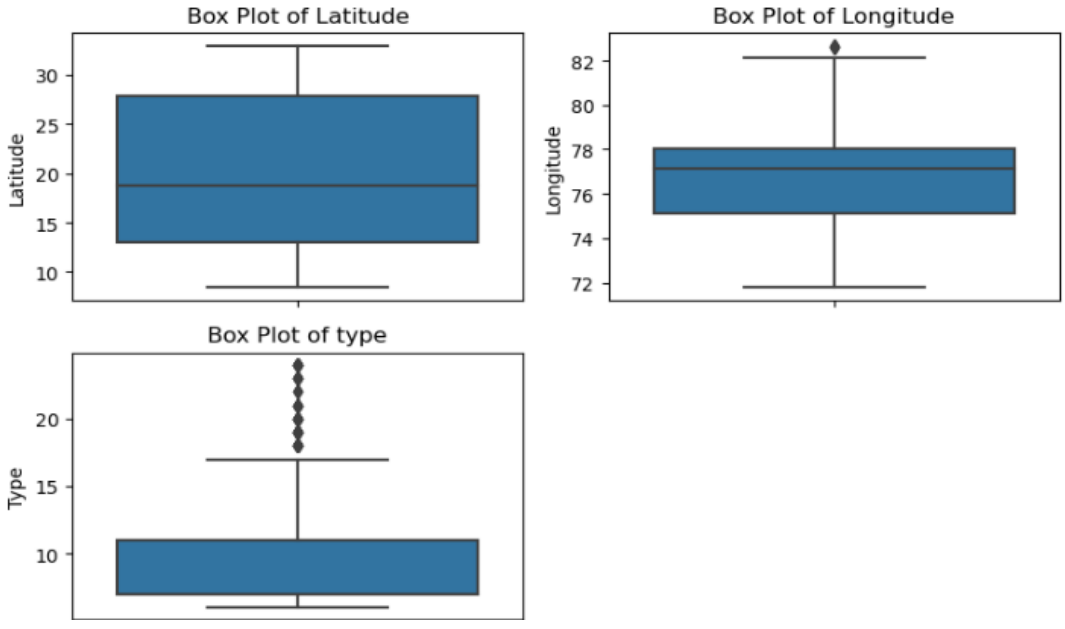




- **Outliers**



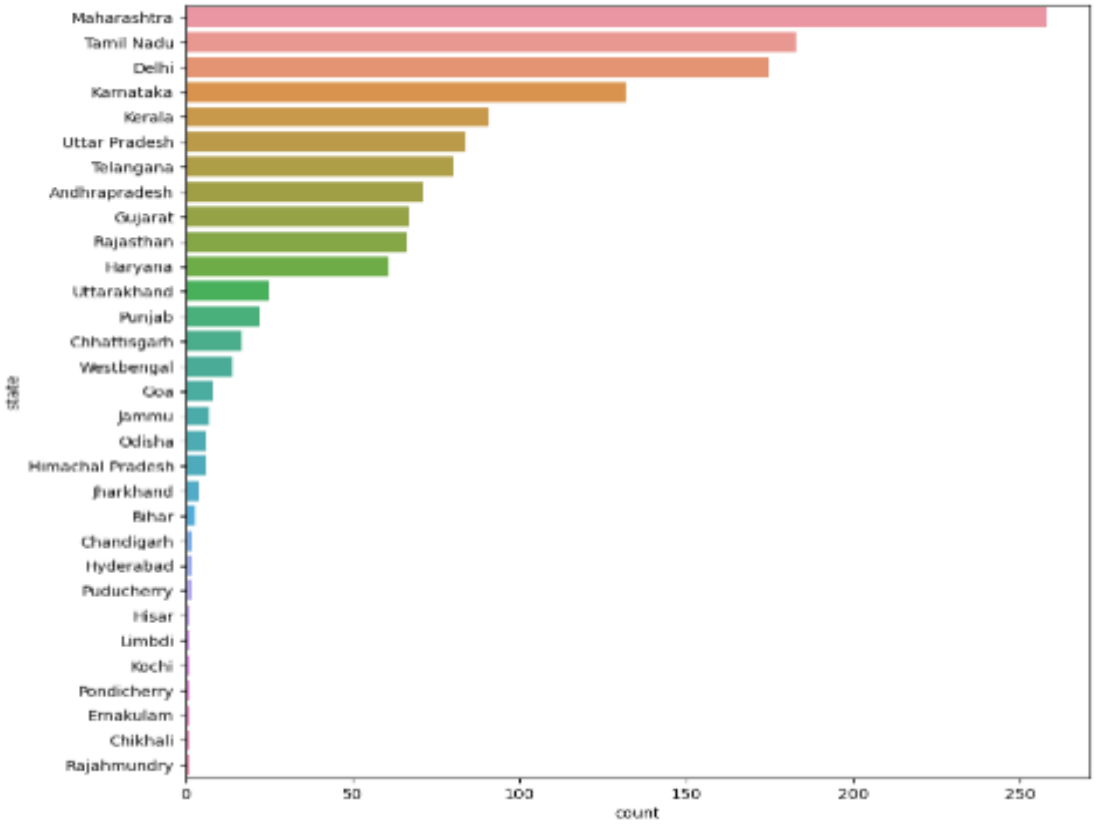
We could see the outliers mainly in lattitude and longitude.I am going to remove the outliers since to main the data integrity.



- **Standardizing state names**

In [20]:
```
import pandas as pd
from difflib import get_close_matches

def standardize_states(df):

    df['state'] = df['state'].str.lower().str.strip()

    state_dict = {}

    for state in df['state']:
        if state in state_dict:
            continue
        close_matches = get_close_matches(state, state_dict.keys(), n=1, cutoff=0.8)

        if close_matches:
            state_dict[state] = state_dict[close_matches[0]]
        else:
            state_dict[state] = state.title()

    df['state'] = df['state'].map(state_dict)

standardize_states(df)
```

In [21]:
```
df['state'] = df['state'].replace('Jammu And Kashmir','Jammu')
df['state'] = df['state'].replace('Delhi Ncr','Delhi')
df['state'] = df['state'].replace('Hyderabadu00A0','Hyderabad')
```

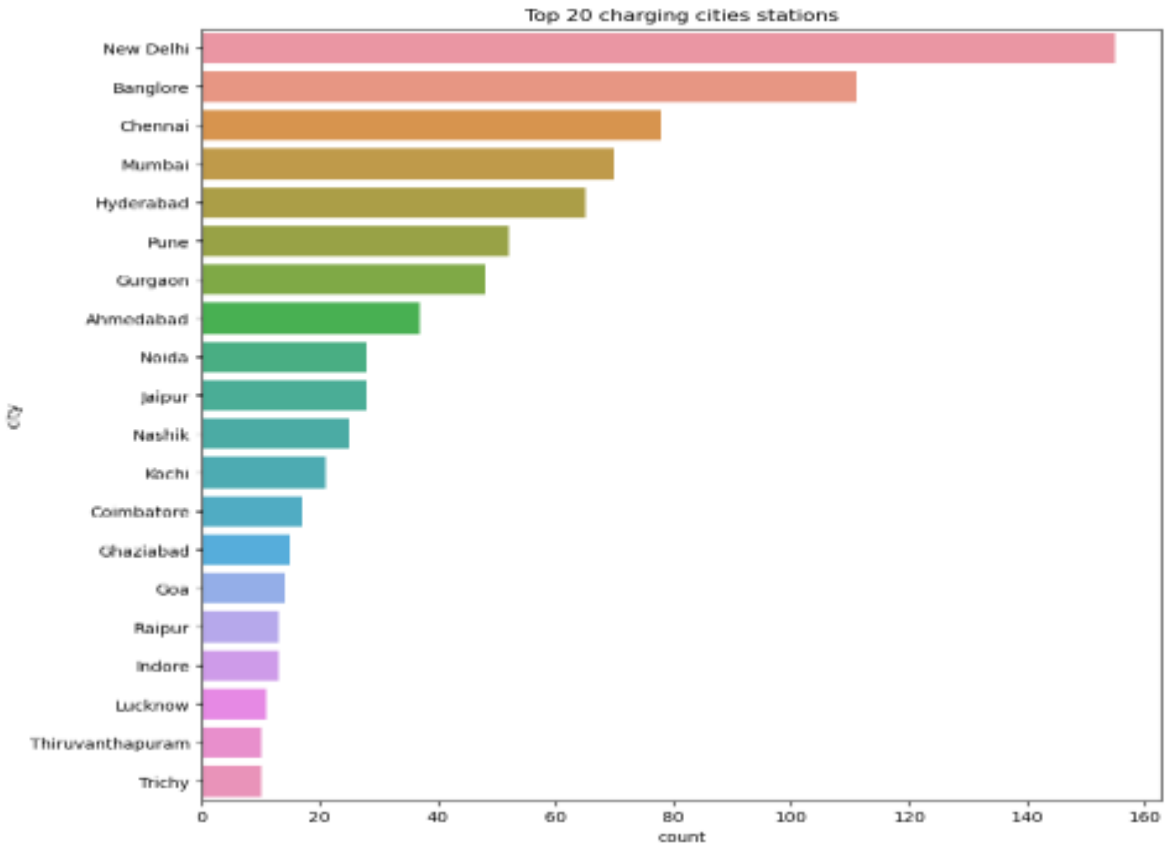


* Maharastra has the more than 250 charging stations
* Followed by Tamil nadu and Delhi has around 150 to 200 charging sations
* Union territories have the least charging station (less than 50 in number)
* Chikhali , Rajahmundry has very less charging station below 10.

- **Standardizing city names**

In [25]:
```
df['city']=df['city'].apply(lambda x : x.capitalize())
```

In [26]:
```
## Mapping dictionary
city_map={
    'Gurgaon':['Gurugram','gurugram','Gurgaon'],
    'Banglore':['Bengaluru','Bangalore','Banglore'],
    'New Delhi':['New Delhi','Delhi','delhi','NEW DELHI','New delhi'],
    'Trichy':['TRICHY','Trichy','Tiruchirappalli'],
    'Pondicherry':['Pondicherry','Puducherry'],
    'Hyderabad':['Hyderabad','HYDERBAD','Hyderbad']
}

# fuction to map cities
def map_cities(city):
    for key,values in city_map.items():
        if city in values:
            return key
    return city

df['city']=df['city'].apply(map_cities)
```

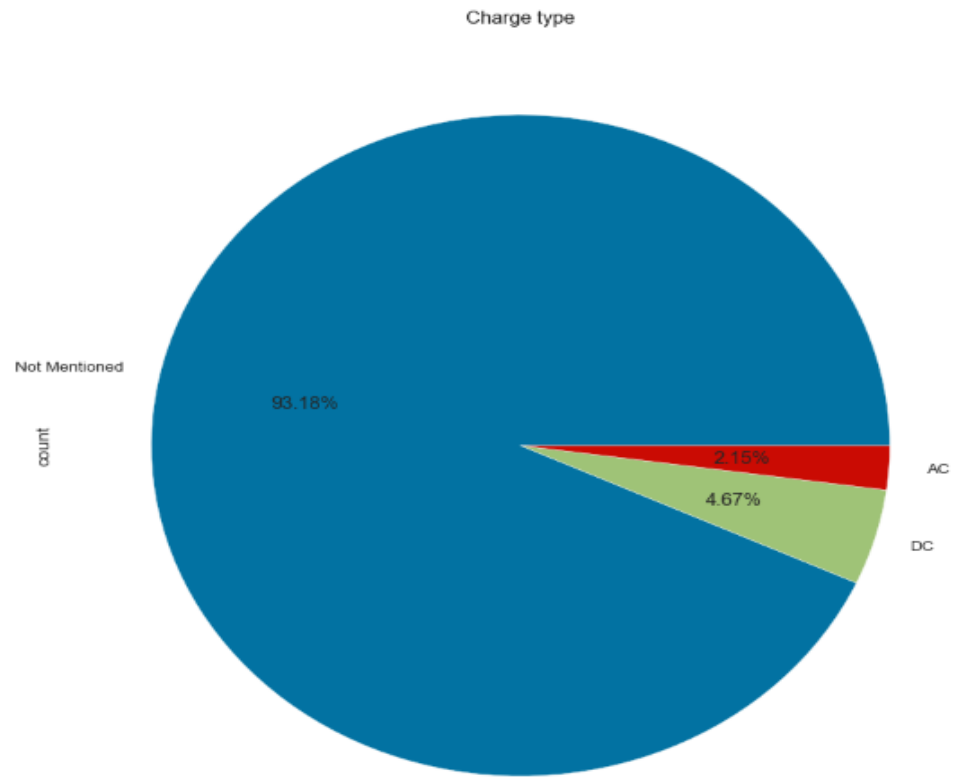


* New Delhi has 150 charging stations.
* Followed by Chennai and Bangalore has around 100 to 120 charging sations
* Other metropolitan cities in india like Surat have the least charging station (less than 10 in number)
* Cities like Villipuram , other parts of south Tamil nadu has less stations for charging .

- **Creating a new column as charge type [Feature extraction]**

```
In [29]: # Deriving the charge type
         def create_charging_type(name):
             if 'AC' in name:
                 return 'AC'
             elif 'DC' in name:
                 return 'DC'
             else:
                 return 'Not Mentioned'

         df['charge_type'] =df['name'].apply(create_charging_type)
```

```
In [30]: df['charge_type'].value_counts()

Out[30]: charge_type
         Not Mentioned    1298
         DC                 65
         AC                 30
         Name: count, dtype: int64
```



* Only 2 % of charging stations are AC station mentioned
* Only 4.20 % of charging stations are DC station mentioned
* About 93.79% where charge type has not mentioned

- **Checking Null values**

```
In [33]: df.isnull().sum()

Out[33]: name            0
         state           0
         city            0
         address        39
         lattitude       6
         longitude       6
         type            8
         charge_type     0
         dtype: int64
```

In [34]:
```
## Dropping the null value rows  in longitude ,latitude and type column
df= df.dropna(subset=['longitude','lattitude','type'])

## Fill null addresses based on state and city information
df['address'] = df.apply(lambda row: f"{row['city']}, {row['state']} Address" if pd.isnull(row['address']) else row['address'],
```

* Dropping the null value rows in longitude ,latitude and type column
* Fill null addresses based on state and city information

- **Checking the Duplicates**

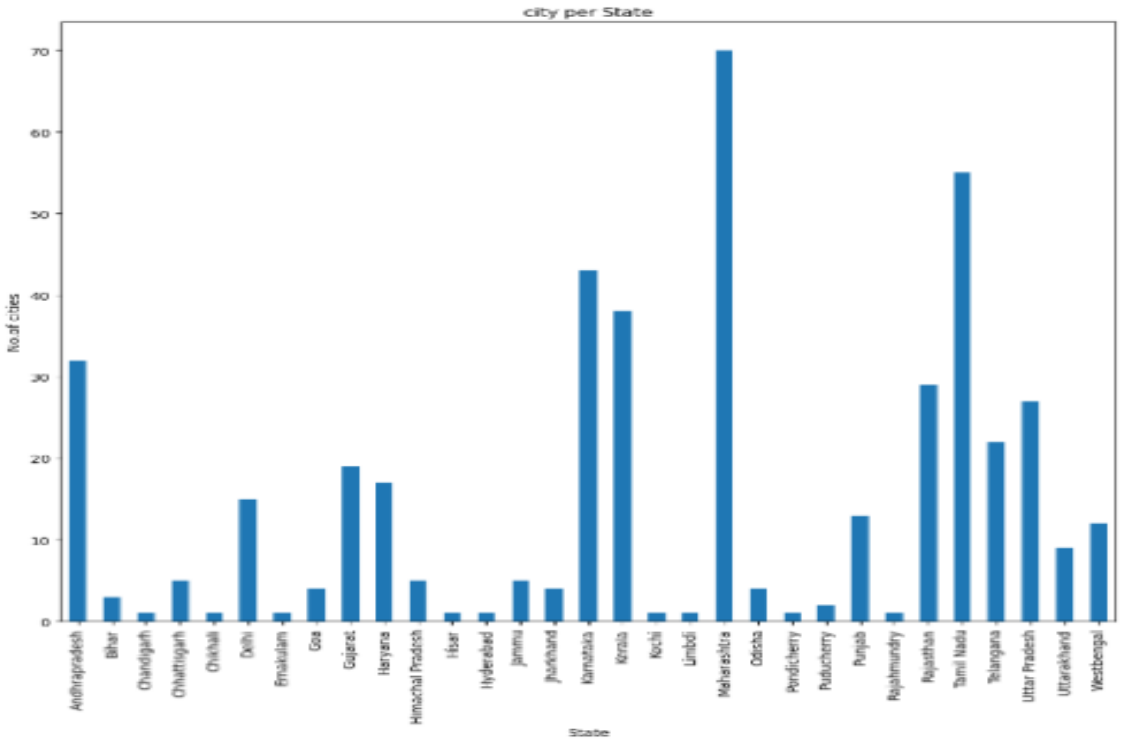
In [37]:
```
duplicate_rows = df.duplicated()

num_duplicates = duplicate_rows.sum()

duplicate_data = df[duplicate_rows]
print("Duplicate rows:")
print(duplicate_data)
```

```
Duplicate rows:
                                          name          state  \
555                       Oberoi Mall, Goregaon    Maharashtra
579   The Grand Legacy, Panchgani-Mahabaleshwar Road    Maharashtra
830                     EESL High court station     Tamil Nadu
832                   EESL Chennai Egmore metro     Tamil Nadu
837                        EESL Chelmsford Club          Delhi
```

In [38]:
```
## Removing the duplicate rows
df=df.drop_duplicates()
```

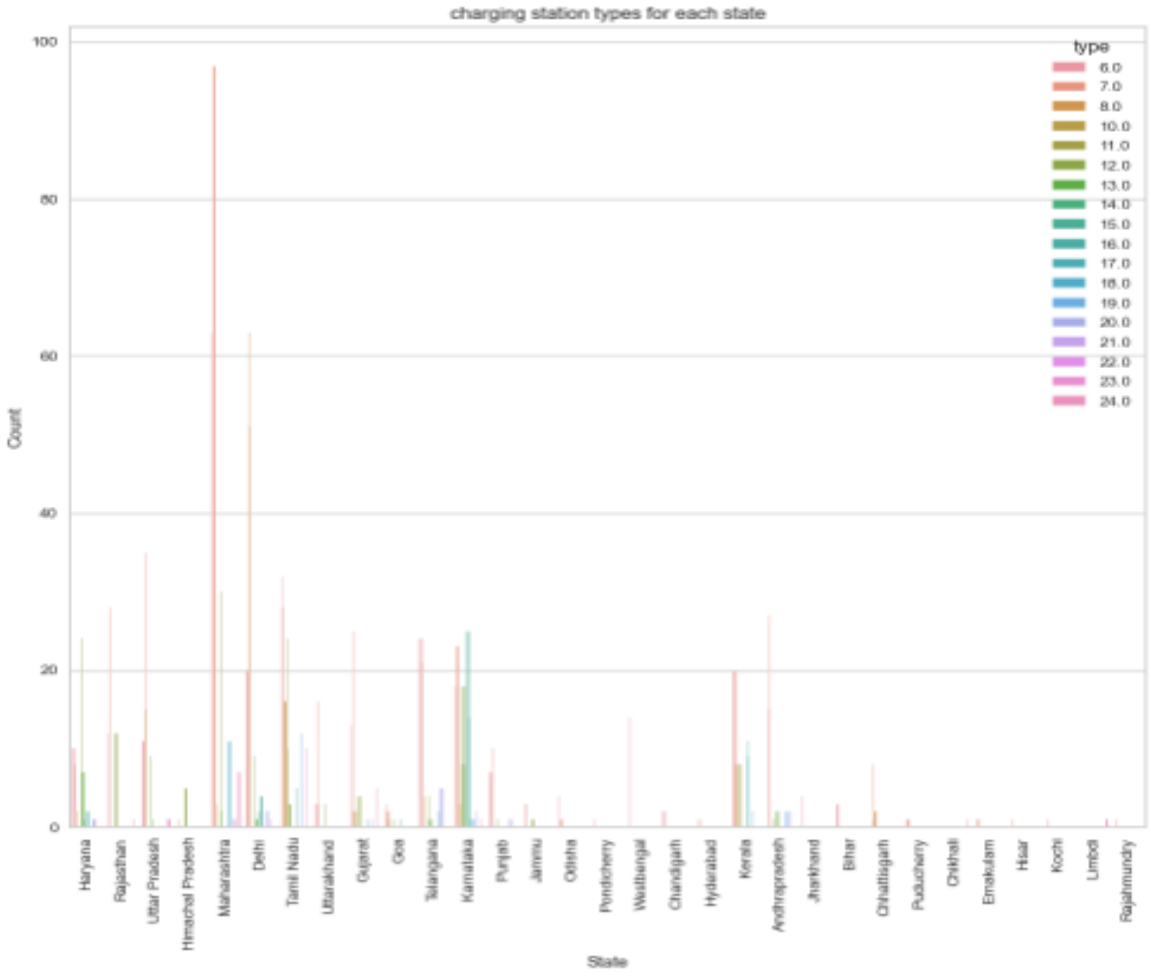
- **Bivariant Analysis**

State Vs City



* Analysing the State column , it's evident that Maharashtra has the highest count of EV charging stations , thus follwed by Delhi.
* Tamil nadu and Karnataka also have 2nd highest count of EV charging stations .
* Most of the charging stations are within the states of Maharashtra , Delhi , Tamil nadu , Karnataka and Kerala in India.
* In cities , Delhi exhibits the highest number of EV charging stations followed by Bangalore , Chennai , Mumbai and Hyderabad.

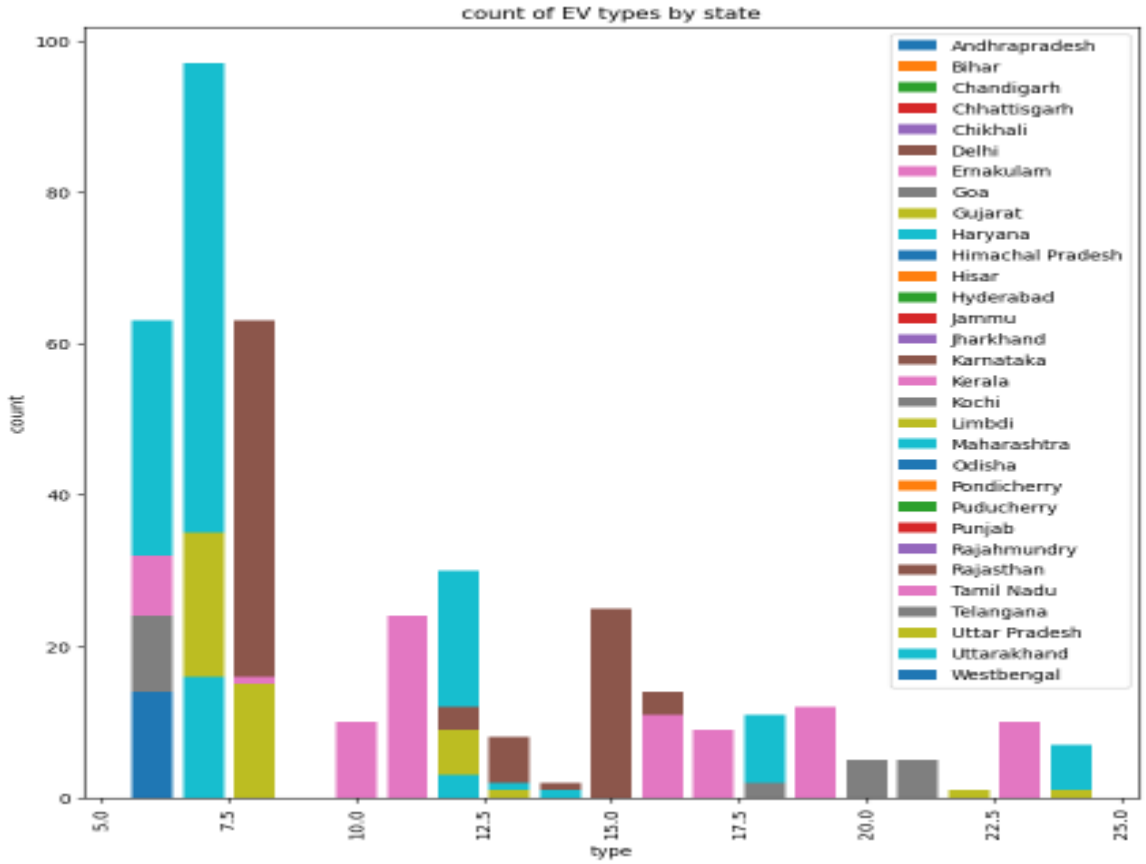
Type Vs State



* In India 7 kWh EV charging stations are highly present.
* After 7 kWh , there are 6,8, and 12 KWh EV stations are significantly seen.

Grouping data by state and type , counting each type in each group

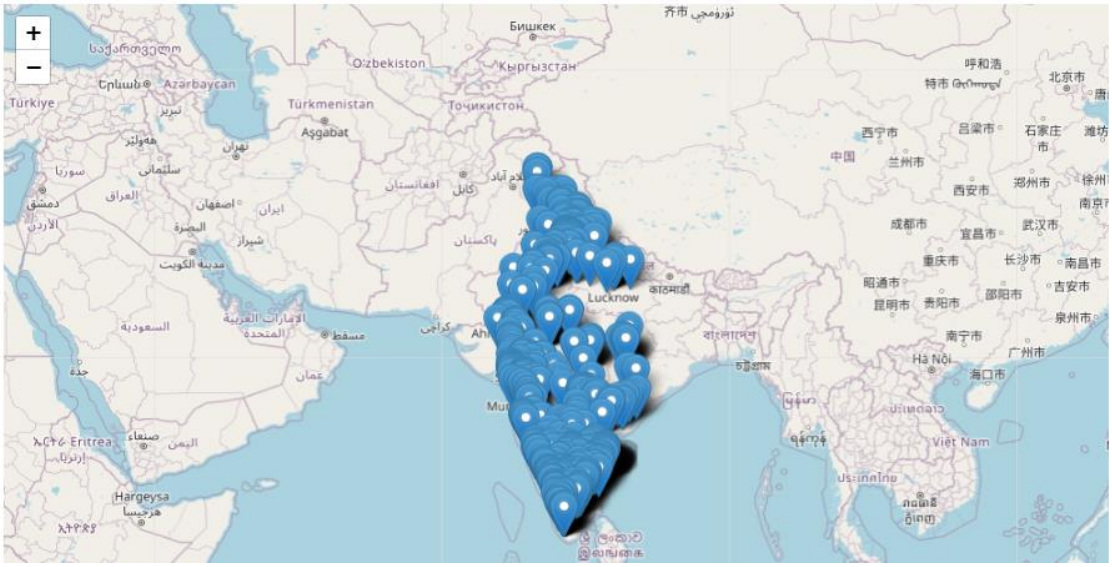
In [42]:
```
## grouping data by state and type , counting each type in each group
group_data = df.groupby(['state','type']).size().reset_index(name='count')

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

for state,group in group_data.groupby('state'):
    plt.bar(group['type'],group['count'],label=state)

plt.xlabel('type')
plt.ylabel('count')
plt.title('count of EV types by state')
plt.xticks(rotation=90)
plt.legend()
plt.show()
```



* In Maharashtra , the majority of charging stations type are 6 and 7 KWh charging capacity.
* In Gujarat , Tamil nadu , Kerala charging stations capacity is about 10, 11 and 19 KWh.
* Telangana and Haryana offeres 20,21 KWh of charging capacities.
* Karnataka and Rajasthan provide with 15 KWh of power capacity

- **Geographical Map of EV charging stations**

Out[43]:



* There is no significant EV charging states in the Eastern part of India.

- **Encoding**

```
In [44]: categorical_columns = ['state', 'city', 'charge_type']

         df_encoded = pd.get_dummies(df, columns=categorical_columns, dtype=int)
```

5.0 PCA

In Principal Component Analysis (PCA), loadings represent the weights or coefficients associated with Each original feature in a particular principal component (PC). They essentially tell how much each original feature contributes to the formation of that specific PC.

- **Scaling the data**

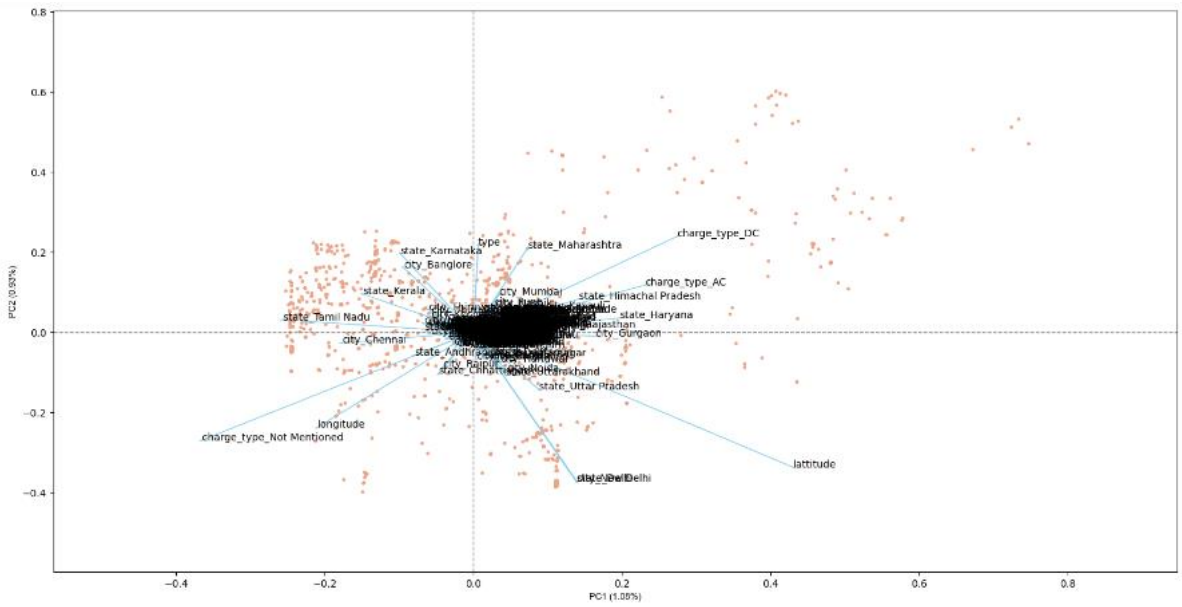
```
In [48]: from sklearn.preprocessing import StandardScaler

         df_scaled=pd.DataFrame(StandardScaler().fit_transform(df),columns=df.columns)
```

- **PCA Components**

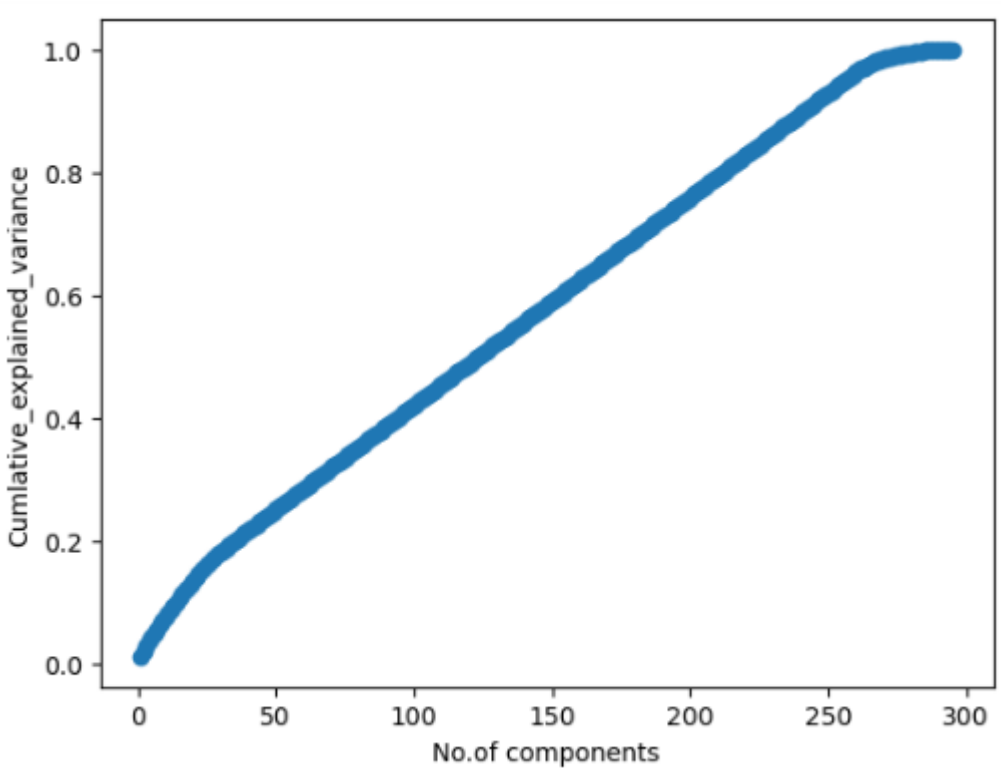
	lattitude	longitude	type	state_Andhrapradesh	state_Bihar	state_Chandigarh	state_Chhattisgarh	state_Chikhali	state_Delhi	state_Ernakulam	...	city_
PC1	0.43	-0.21	0.01	-0.08	-0.00	-0.03	-0.05	0.01	0.14	-0.02	...	
PC2	-0.34	-0.24	0.22	-0.06	-0.01	-0.01	-0.10	0.04	-0.37	0.01	...	
PC3	-0.03	0.47	0.17	0.04	-0.03	0.03	0.12	-0.02	-0.05	-0.01	...	
PC4	0.02	-0.04	0.28	-0.11	-0.00	-0.01	-0.13	0.36	0.27	-0.01	...	
PC5	0.07	0.01	-0.01	0.09	-0.02	-0.00	0.12	0.29	-0.34	-0.01	...	
...	...	...	...	...	...	...	...	...	...	...	...	
PC291	0.63	0.37	0.00	0.01	0.00	0.00	0.00	0.01	0.00	-0.00	...	
PC292	0.00	-0.00	0.00	0.16	0.04	0.03	0.08	0.12	0.27	0.02	...	
PC293	0.00	-0.00	-0.00	-0.01	-0.00	-0.00	-0.01	0.22	-0.02	-0.00	...	
PC294	0.00	-0.00	0.00	-0.11	-0.03	-0.02	-0.05	-0.16	-0.18	-0.02	...	
PC295	-0.00	-0.00	0.00	-0.08	-0.02	-0.02	-0.04	0.40	-0.13	-0.01	...	

Principal Component Summary:

	Standard deviation	Proportion of Variance	Cumulative variance Ratio
PC1	3.19	0.01	0.01
PC2	2.74	0.01	0.02
PC3	2.59	0.01	0.03
PC4	2.09	0.01	0.04
PC5	2.01	0.01	0.04
...	...	...	...
PC291	0.00	0.00	1.00
PC292	0.00	0.00	1.00
PC293	0.00	0.00	1.00
PC294	0.00	0.00	1.00
PC295	0.00	0.00	1.00



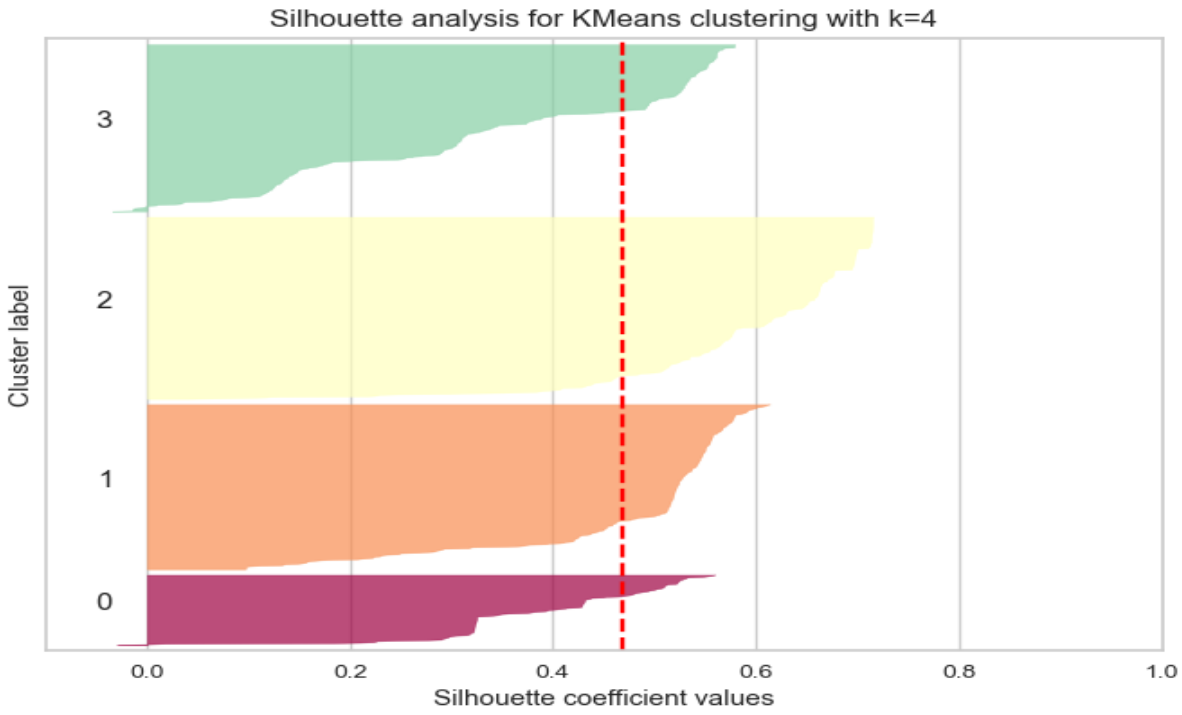
* Thus the plot shows that the state Maharashtra , city Delhi , charge_type DC has the highest importance and high variance.



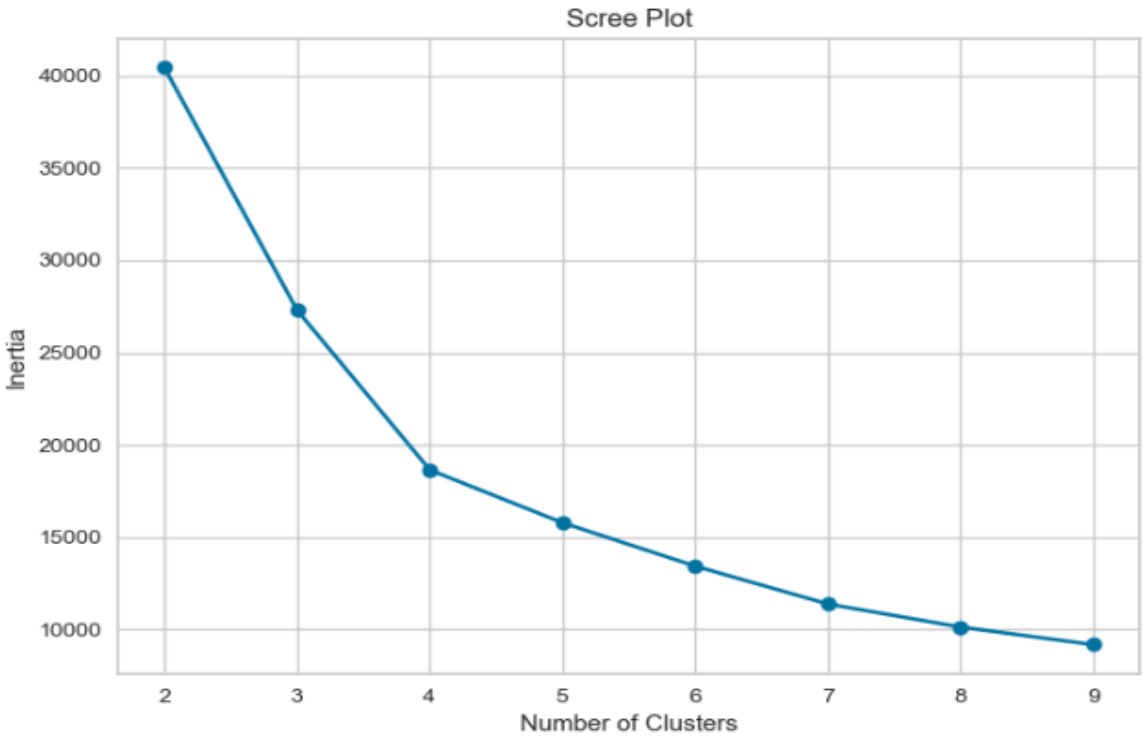
* It indicates that adding more principal components does not significantly increase the explained variance.
* There is no multicollinearity between the independent variables.
* Thus there is no need of Principle component analysis.
* The Scree Plot shows a straight line, it suggests that each additional principal component contributes roughly the same amount of variance to the overall dataset.

6.0 Kmeans Clustering

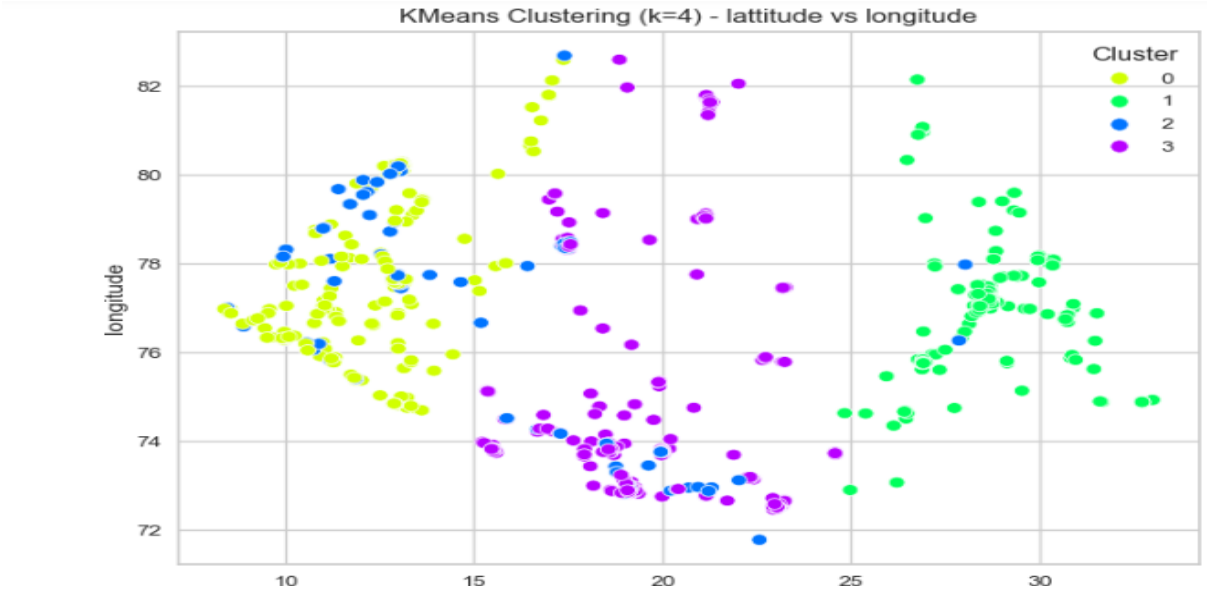
- **Silhouette analysis for KMeans clustering with k**



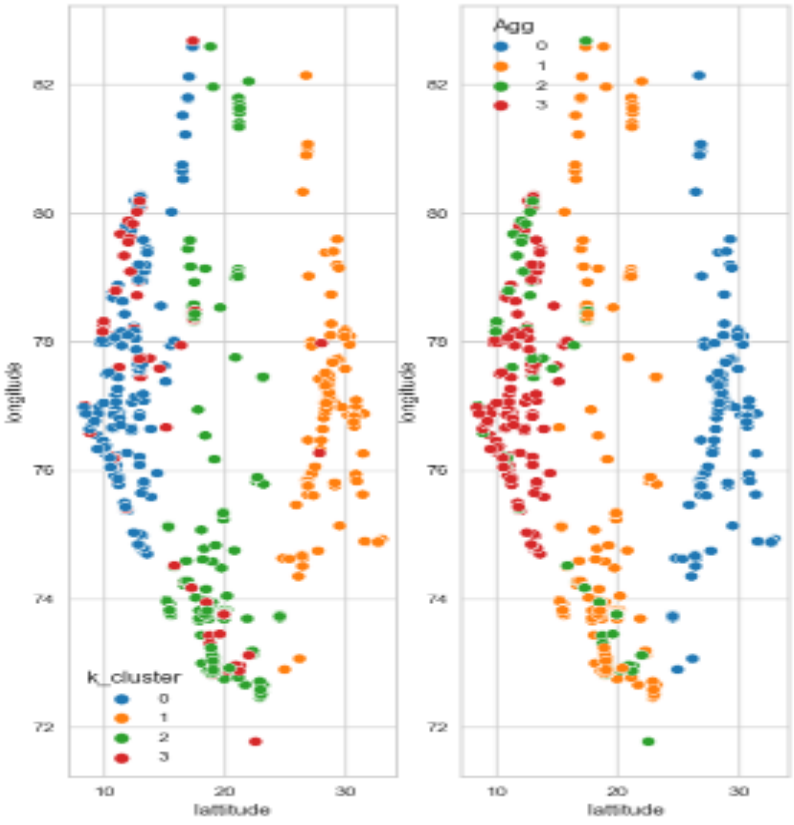
- **Inertia Offset**



* Based on the analysis of the provided graphs, it appears that there are four clusters with favorable inertia offset, indicating well-defined clusters.
* The scree plot reveals an upward trend in variance explained after the 4th cluster, suggesting that the optimal choice for this dataset may extend beyond three clusters.
* The silhouette_score shows k=2, a higher silhouette score indicates better-defined clusters because of the wide area, whereas the avg silhouette_coefficient is about 0.5 , thus the clusters are not overlaped.



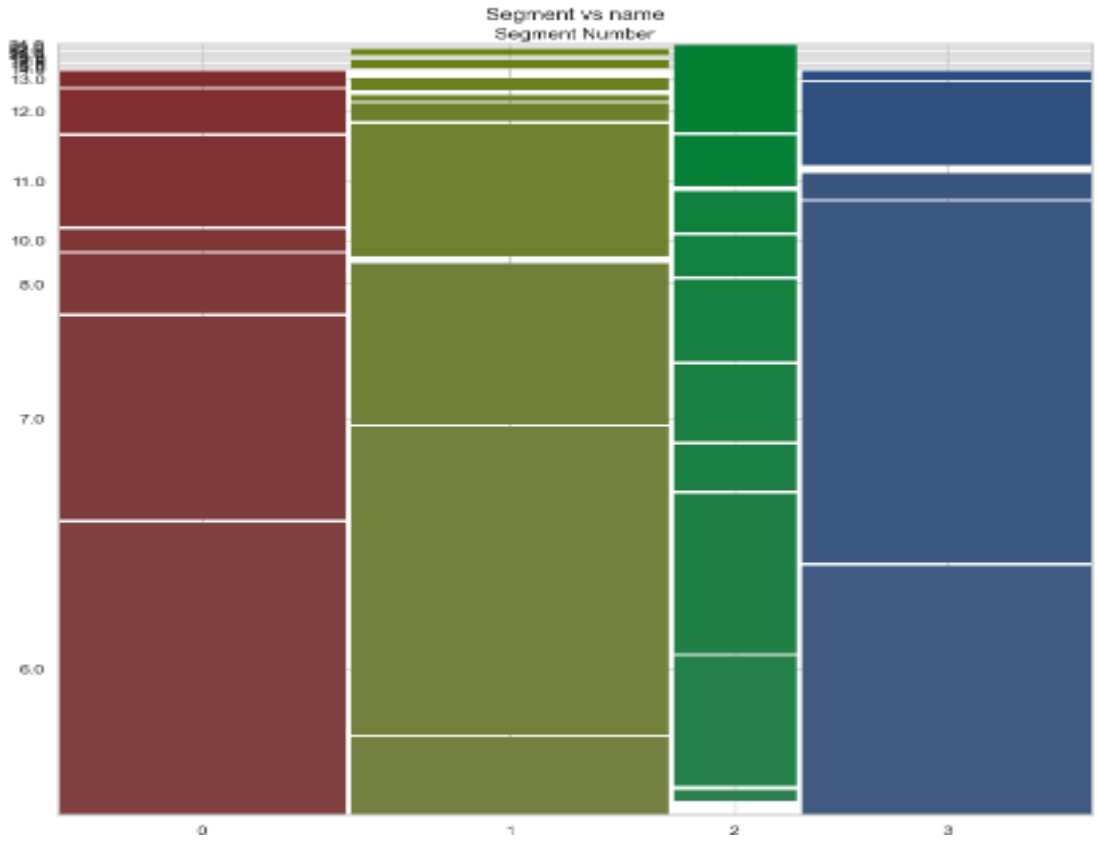
7.0 Agglomerative Clustering



8.0 Segment profiling

* Segment 0, 1,2, have about 300 clustered features which contribute to the profiling

- .**Mosaic Plot for Segment Membership and Type**



9.0 Conclusion

* Maharashtra leads in EV charging stations, followed closely by Delhi, Tamil Nadu, and Karnataka, collectively hosting half of India's stations.
* New Delhi tops city counts, followed by Bengaluru, Chennai, Mumbai, and Hyderabad.
* 7 kWh and 6 KWh stations are prevalent in cluster 0 and 3 , with significant numbers also for 8 to 12 kWh variants in all clusters
* Clustering analysis reveals well-defined clusters, with four showing favorable scores,the scree plot suggests the potential for more than four clusters.
* Setting up the EV charging stations in the Eastern part of India ,and increasing the number of stations in the union territories, will increase the selling rate.
* Most of the charging type is not mentioned in the dataset, for further analysis .
* The metroplitian cities have prominant number of stations,except Surat.
* Overall, this analysis highlights regional distribution patterns and charging station capacities in India.