ev-market-segmentation-t1-r

July 22, 2024

BEHAVIORAL AND PSYCHOGRAPHIC ANALYSIS

Behavioral Segmentation is a form of customer segmentation that is based on patterns of behavior displayed by customers as they interact with a company/brand or make a purchasing decision. It allows businesses to divide customers into groups according to their knowledge of, attitude towards, use of, or response to a product, service or brand.

Psychographic segmentation approach involves an understanding of a consumer's lifestyle, interests, and opinions. We have combined the two types of analysis because a consumer's lifestyle, interests and opinions are mirrored in their purchasing behavior.

The dataset we have used is a survey of people who own particular brands of fuel-based vehicles and it contains some basic information such as their age, salary, loan status, marital status, number of dependents, education, occupation and the make of their car and its price.

[63]:

```
import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from sklearn.preprocessing import StandardScaler
import nltk



df = pd.read_csv(r'/Indian automoble buying behavour study 1.0.csv')
df.head()
df.isnull().sum()
print(df.head)
print(df.columns)
```

```
<bound method NDFrame.head of     Age Profession Marrital Status      Education
No of Dependents  \
0    27   Salaried          Single  Post Graduate                 0
1    35   Salaried         Married  Post Graduate                 2
2    45   Business         Married       Graduate                 4
3    41   Business         Married  Post Graduate                 3
4    31   Salaried         Married  Post Graduate                 2
..   …        …              …            …                …
```

```
94   27   Business          Single       Graduate                 0
95   50   Salaried         Married  Post Graduate                 3
96   51   Business         Married       Graduate                 2
97   51   Salaried         Married  Post Graduate                 2
98   51   Salaried         Married  Post Graduate                 2

   Personal loan House Loan Wife Working   Salary  Wife Salary  Total Salary  \
0            Yes         No           No   800000            0        800000
1            Yes        Yes          Yes  1400000       600000       2000000
2            Yes        Yes           No  1800000            0       1800000
3             No         No          Yes  1600000       600000       2200000
4            Yes         No          Yes  1800000       800000       2600000
..           …          …            …       …            …            …
94            No         No           No  2400000            0       2400000
95            No         No          Yes  3800000      1300000       5100000
96           Yes        Yes           No  2200000            0       2200000
97            No         No          Yes  2700000      1300000       4000000
98           Yes        Yes           No  2200000            0       2200000

      Make    Price
0      i20   800000
1     Ciaz  1000000
2   Duster  1200000
3     City  1200000
4      SUV  1600000
..     …       …
94     SUV  1600000
95     SUV  1600000
96    Ciaz  1100000
97  Creata  1500000
98    Ciaz  1100000

[99 rows x 13 columns]>
Index(['Age', 'Profession', 'Marrital Status', 'Education', 'No of Dependents',
       'Personal loan', 'House Loan', 'Wife Working', 'Salary', 'Wife Salary',
       'Total Salary', 'Make', 'Price'],
      dtype='object')
```

[64]:
```python
df.head ()
df.drop(['Make'], axis=1, inplace=True)
df.head()
```

[64]:
```
   Age Profession Marrital Status      Education  No of Dependents  \
0   27   Salaried          Single  Post Graduate                 0
1   35   Salaried         Married  Post Graduate                 2
2   45   Business         Married       Graduate                 4
3   41   Business         Married  Post Graduate                 3
```

```
4   31   Salaried         Married  Post Graduate                  2

  Personal loan House Loan Wife Working   Salary  Wife Salary  Total Salary  \
0           Yes         No           No   800000            0        800000
1           Yes        Yes          Yes  1400000       600000       2000000
2           Yes        Yes           No  1800000            0       1800000
3            No         No          Yes  1600000       600000       2200000
4           Yes         No          Yes  1800000       800000       2600000

     Price
0   800000
1  1000000
2  1200000
3  1200000
4  1600000
```

[65]:

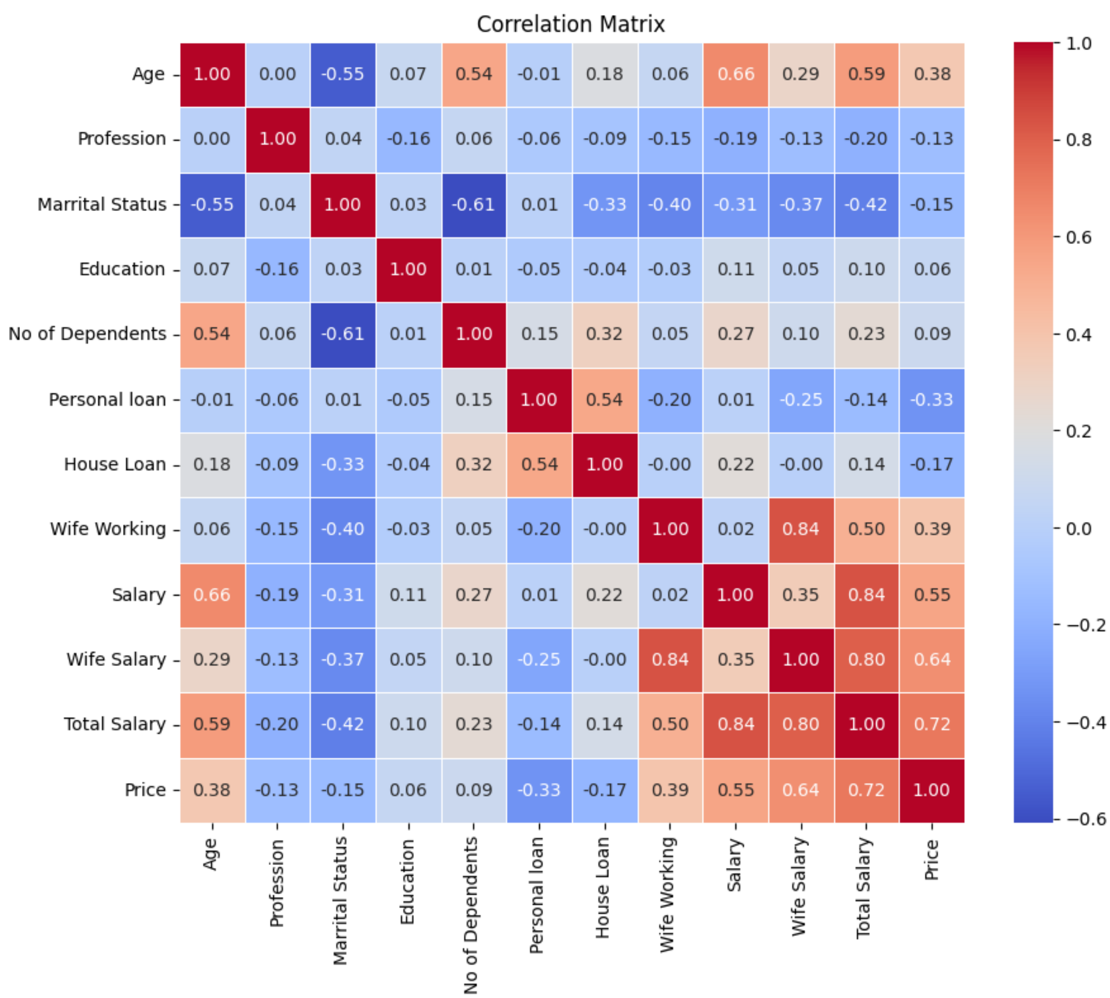
```python
mappings = {
    'Profession': {'Salaried': 0, 'Business': 1},
    'Marrital Status': {'Single': 1, 'Married': 0},
    'Education': {'Post Graduate': 1, 'Graduate': 0},
    'Personal loan': {'Yes': 1, 'No': 0},
    'House Loan': {'Yes': 1, 'No': 0},
    'Wife Working': {'Yes': 1, 'No': 0}
}
data = df.replace(mappings)
data.head()
```

[65]:

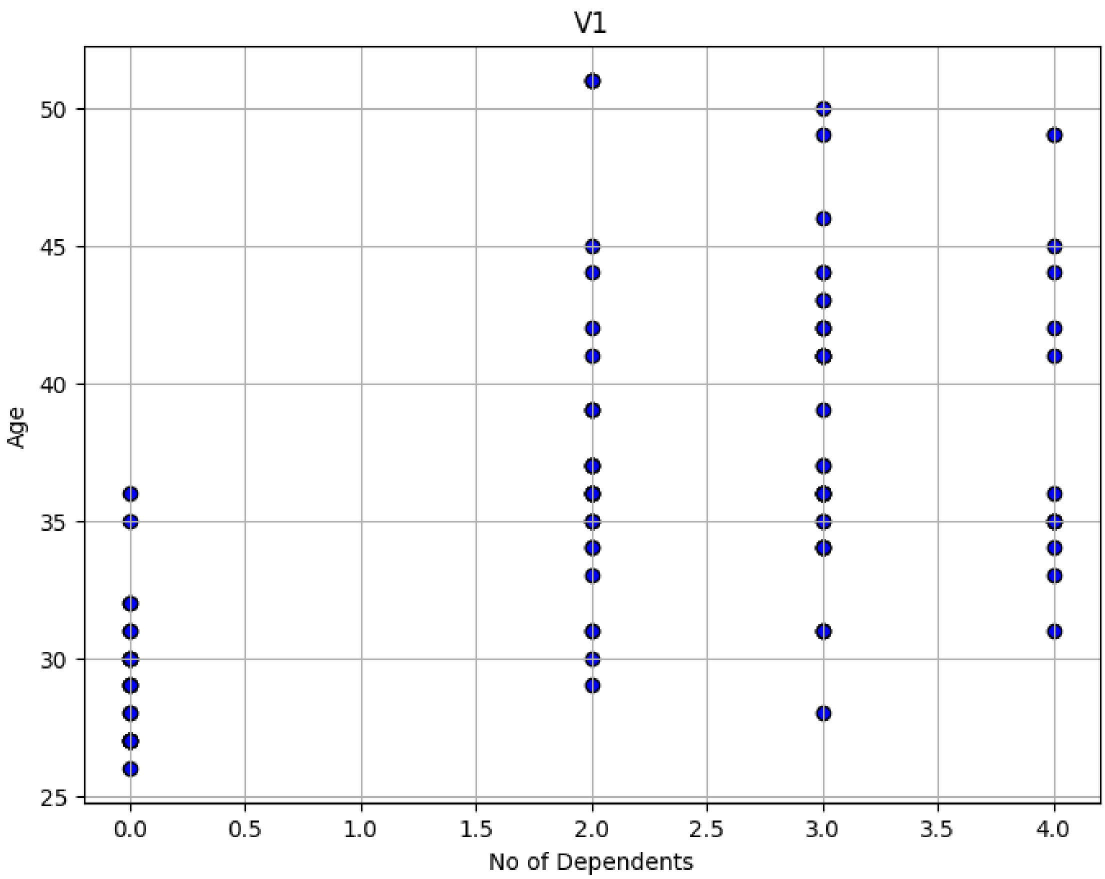
```
   Age  Profession  Marrital Status  Education  No of Dependents  \
0   27           0                1          1                 0
1   35           0                0          1                 2
2   45           1                0          0                 4
3   41           1                0          1                 3
4   31           0                0          1                 2

   Personal loan  House Loan  Wife Working   Salary  Wife Salary  Total Salary  \
0              1           0             0   800000            0        800000
1              1           1             1  1400000       600000       2000000
2              1           1             0  1800000            0       1800000
3              0           0             1  1600000       600000       2200000
4              1           0             1  1800000       800000       2600000

     Price
0   800000
1  1000000
2  1200000
3  1200000
4  1600000
```

[66]:

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


data_numeric = data.apply(pd.to_numeric, errors='coerce')


corr_matrix = data_numeric.corr()

plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)

plt.title('Correlation Matrix')

plt.show()
```

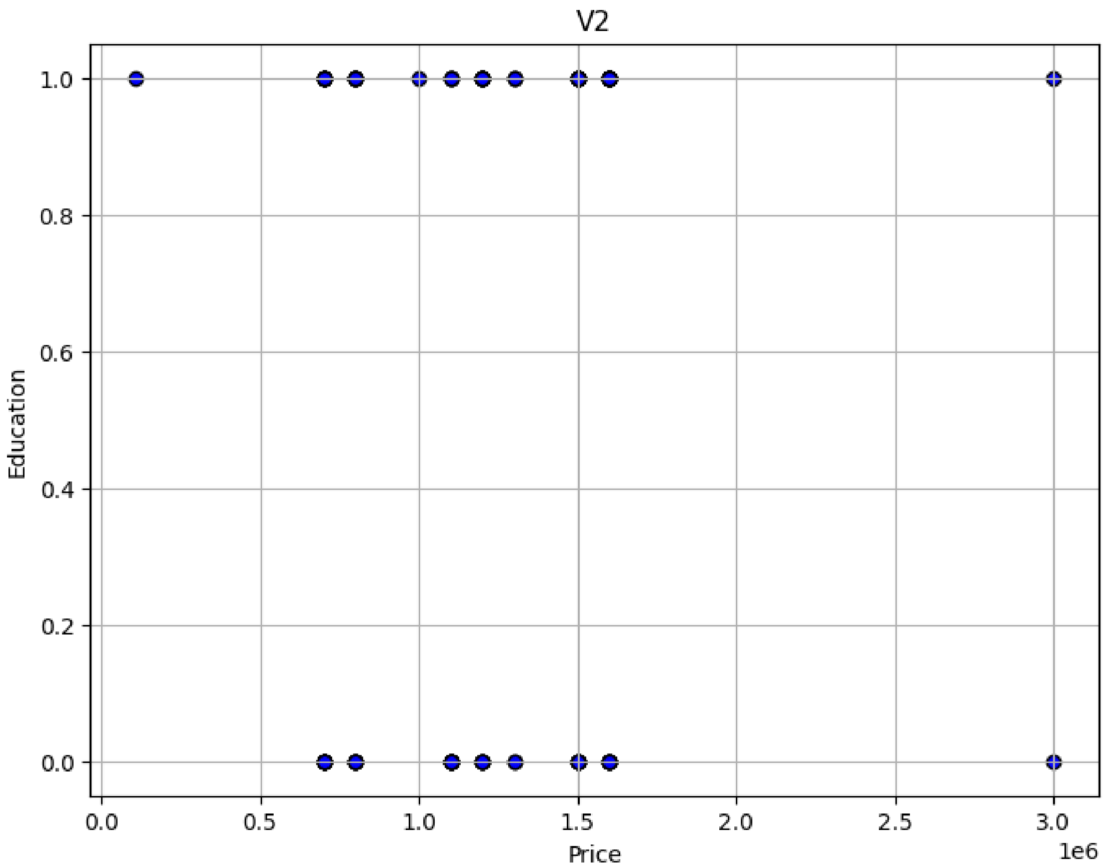


[67]:
```
plt.figure(figsize=(8, 6))
plt.scatter(data['No of Dependents'], data['Age'], color='blue', edgecolor='k')


plt.title('V1')
plt.xlabel('No of Dependents')
plt.ylabel('Age')


plt.grid(True)
plt.show()
```



[68]:
```
plt.figure(figsize=(8, 6))
plt.scatter(data['Price'], data['Education'], color='blue', edgecolor='k')


plt.title('V2')
plt.xlabel('Price')
plt.ylabel('Education')
```

```
plt.grid(True)
plt.show()
```



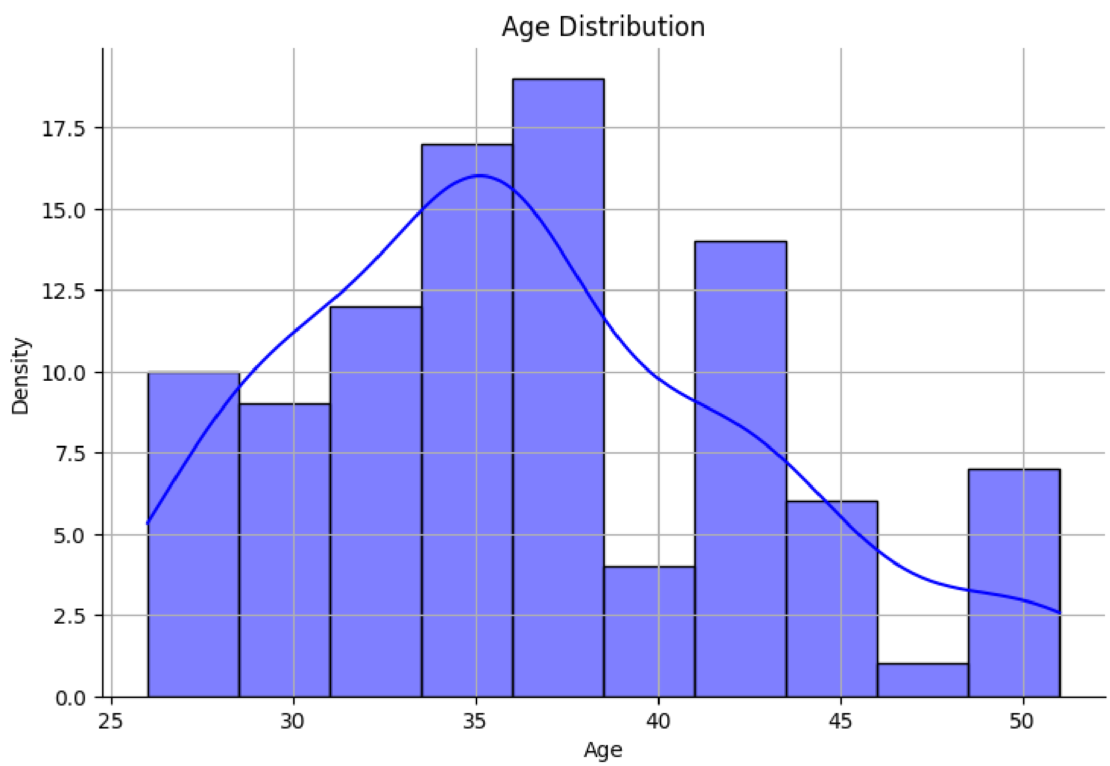
Demographic Analysis

[69]:

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



plt.figure(figsize=(10, 6))
sns.displot(data['Age'], kde=True, bins=10, color='blue', aspect=1.5)


plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Density')
```

```
plt.grid(True)


plt.show()
```

<Figure size 1000x600 with 0 Axes>



Pepole Between age group of 20 to 25 constitute Most of the market .

Geographic Analysis

[70]:
```
df = pd.read_csv(r'/EVStats.csv')
df.head ()
df.isnull().sum()
```

[70]:
```
Sl. No                                                                 0
State                                                                  0
Two Wheelers (Category L1 & L2 as per Central Motor Vehicles Rules    0
Two Wheelers (Category L2 (CMVR))                                      0
Two Wheelers (Max power not exceeding 250 Watts)                       0
Three Wheelers (Category L5 slow speed as per CMVR)                    0
Three Wheelers (Category L5 as per CMVR)                               0
Passenger Cars (Category M1 as per CMVR)                               0
```

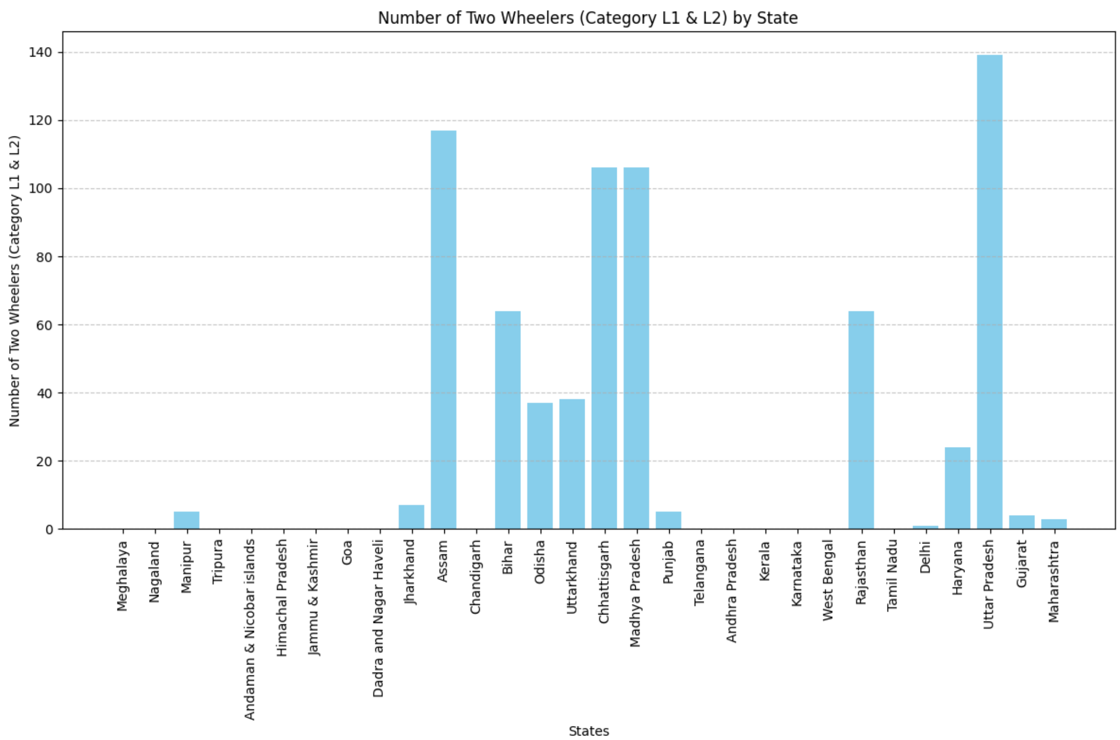
```
Buses                                                                 0
Total in state                                                        0
dtype: int64
```

[71]:
```
df.describe()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 10 columns):
 #   Column                                                               Non-
Null Count  Dtype
---  ------
--------------  -----
 0   Sl. No                                                               30 non-
null     int64
 1   State                                                                30 non-
null     object
 2   Two Wheelers (Category L1 & L2 as per Central Motor Vehicles Rules  30 non-
null     int64
 3   Two Wheelers (Category L2 (CMVR))                                    30 non-
null     int64
 4   Two Wheelers (Max power not exceeding 250 Watts)                     30 non-
null     int64
 5   Three Wheelers (Category L5 slow speed as per CMVR)                  30 non-
null     int64
 6   Three Wheelers (Category L5 as per CMVR)                             30 non-
null     int64
 7   Passenger Cars (Category M1 as per CMVR)                             30 non-
null     int64
 8   Buses                                                                30 non-
null     int64
 9   Total in state                                                       30 non-
null     int64
dtypes: int64(9), object(1)
memory usage: 2.5+ KB
```

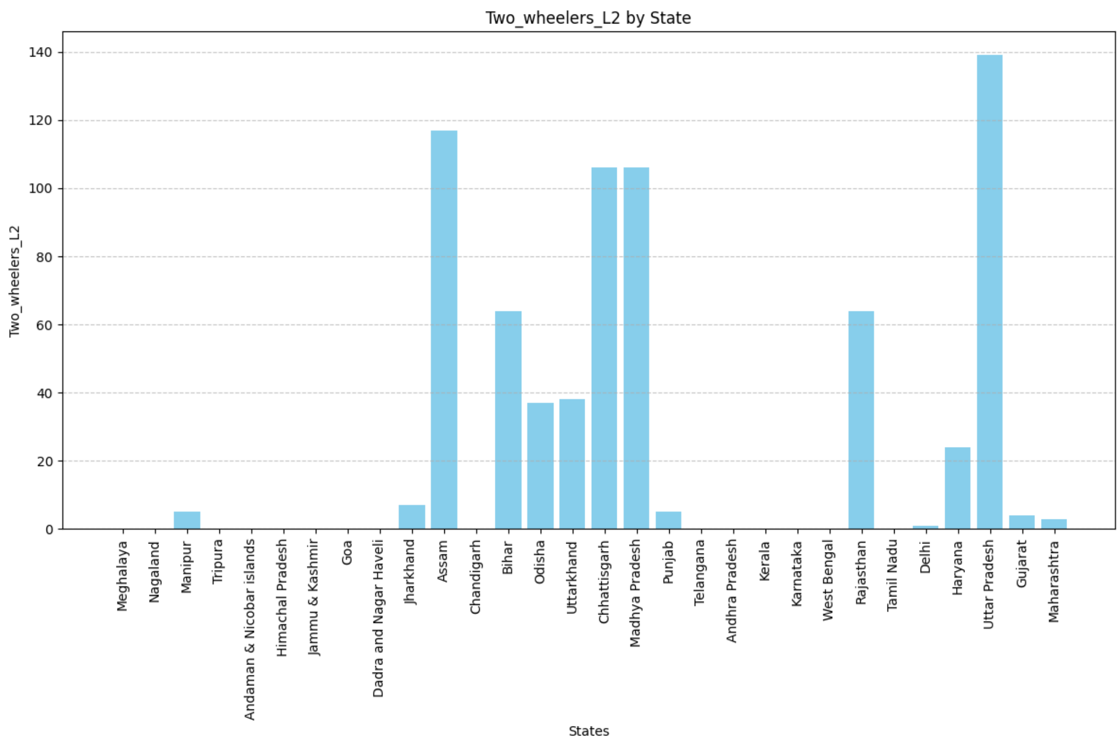
[72]:
```
import pandas as pd
import matplotlib.pyplot as plt


df = pd.read_csv(r'/EVStats.csv')

states = df['State']
Three_wheelers_L5 = df['Three Wheelers (Category L5 as per CMVR)']
```

```
plt.figure(figsize=(12, 8))
plt.bar(states, Three_wheelers_L5, color='skyblue')


plt.xlabel('States')
plt.ylabel('Number of Two Wheelers (Category L1 & L2)')
plt.title('Number of Two Wheelers (Category L1 & L2) by State')
plt.xticks(rotation=90)


plt.grid(axis='y', linestyle='--', alpha=0.7)


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



[73]:
```
states = df['State']
Two_wheelers_L2 = df['Two Wheelers (Category L2 (CMVR))']


plt.figure(figsize=(12, 8))
plt.bar(states, Three_wheelers_L5, color='skyblue')
```

```
plt.xlabel('States')
plt.ylabel('Two_wheelers_L2')
plt.title('Two_wheelers_L2 by State')
plt.xticks(rotation=90)


plt.grid(axis='y', linestyle='--', alpha=0.7)


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



Depending on the type of Electric Vehicle the startup comes with, it can target that particular state. What is important to consider is that for most of these electric vehicles that market would be a fairly developed city in that state, because consumers should be willing to purchase the electric vehicle and factors like cost versus average consumer income and the resources to charge the EV (e.g. Charging Stations) and being able to maintain it are important.

Target Segment

The younger population is more likely to purchase products with new technology, especially Electric Vehicles as they are aware of the environmental benefits and would like to bring that change, but our report showed that younger population buys less expensive vehicles and so Electric Vehicles not being affordable can be a downside. It is then suggested to target a segment which is still eager to try new technologies but financially well enough to be able to afford Electric Vehicles. These people are likely to be in an age-group of 30 to 40 years.

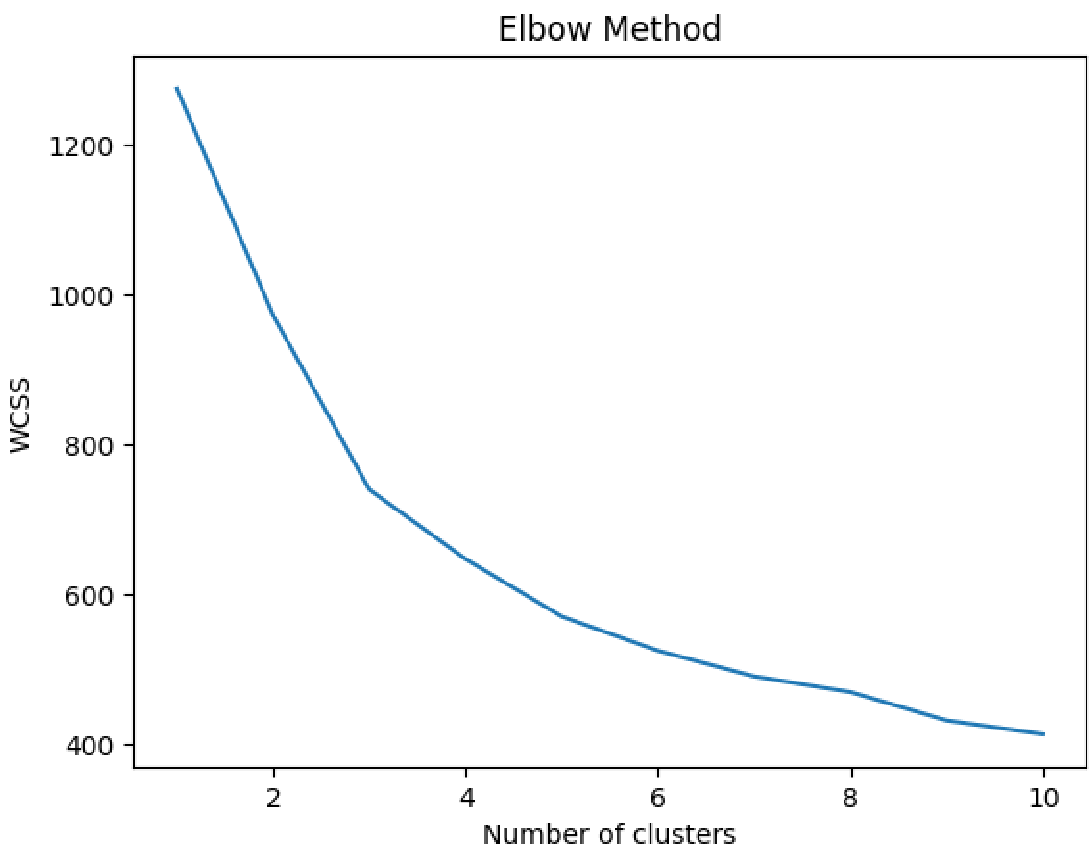
People from urban cities with available infrastructure and education about technology and its benefits will tend to purchase electric vehicles more. People who are married and who have eependents are more likely to go ahead and purchase a vehicle and so they could be targeted. Average salary of people who buy vehicles is around 30 lakh and the most purchases for automobiles lies in the range 10-20 lakh and lesser for two-wheelers. These aspects need to be kept in mind too.. Marketing Mix

Kmean clustering Algorithm

[79]:
```python
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score

from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
    kmeans.fit(scaled_data)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

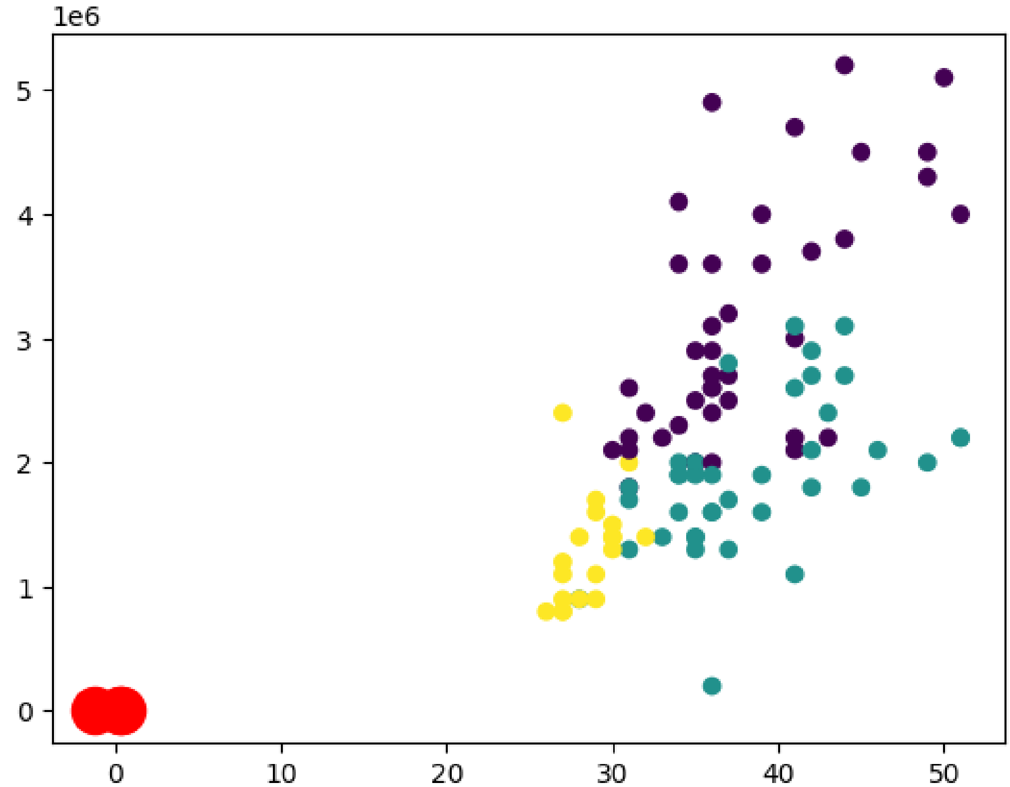
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```



[84]:
```python
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt


optimal_k = 3

scaler = StandardScaler()

data_numeric = data_numeric.dropna()
scaled_data = scaler.fit_transform(data_numeric)


kmeans = KMeans(n_clusters=3, random_state=42)


kmeans.fit(scaled_data)


data_numeric['cluster'] = kmeans.labels_

# Visualize the clusters
plt.scatter(data_numeric['Age'], data_numeric['Total Salary'], c=kmeans.labels_)
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
 ↪s=300, c='red')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```



4ps of Marketing Mix

PRODUCT

The type of product would obviously depend on the EV Startup, but throughout our analysis we figured that for India it is best to enter the market with two-wheelers because the most automobile market-share is of two-wheelers. Most people would purchase a two-wheeler because it is cost effective, and the current infrastructure would support that. Another type of product EV Startup can look into is public transport vehicles, because the current government policies are supportive for revamping public transport to electric-based engines.

PLACE

Infrastructure is another important aspect that has to be kept in mind while creating any product and launching it. Major urban cities of the country should be targeted as these are the places where infrastructure would support. Another reason for targeting urban cities is that here it is more likely to have an educated population willing to buy Electric Vehicles because they are aware of the environmental benefits.

PRICE

Affordability is a major issue with the growth of Electric Vehicles. It is important to keep in mind that in order to appeal to the consumers, the company's product has to be cost effective to both purchase and maintain. The product's price should ideally range.

PROMOTION

Promotion is product dependent. The best possible promotion is to educate people of the benefits of EV/HEV/PHEV over fuel-based vehicles. If the Startup comes up with an affordable product that should definitely be promoted.