MARKET SEGMENTATION
ON
ELECTRIC VEHICLES

TEAM: DHANDHUKIYA

Date: 18/7/2022

Members:

- Dhruv Kumar Dhandhukiya
- Anish Desai
- Ansh Ashok Shriwas
- Adnan Habib
- Aniket Singh

GitHub link: [Adnan232/Electric-Vehicle-Segmentation-Analysis (github.com)]

PROBLEM STATEMENT:

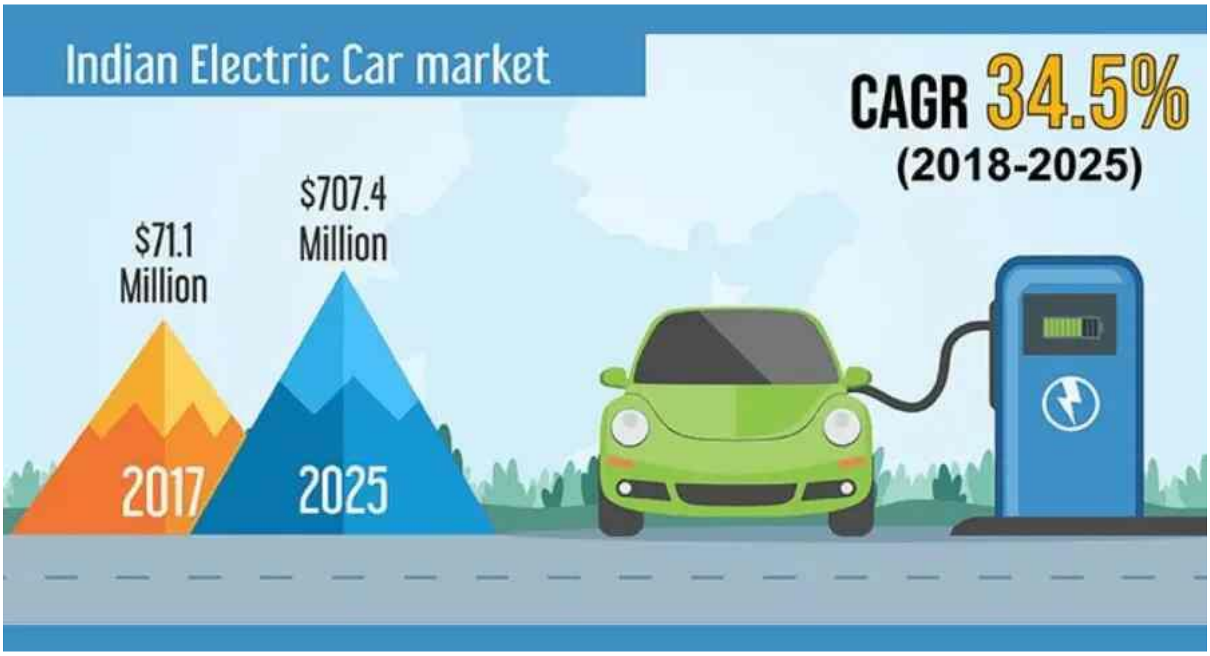
Using market segmentation, analyse the electric vehicles in India and come up with a feasible strategy to enter the market and targeting the segments that are most likely to buy the Electric Vehicles.

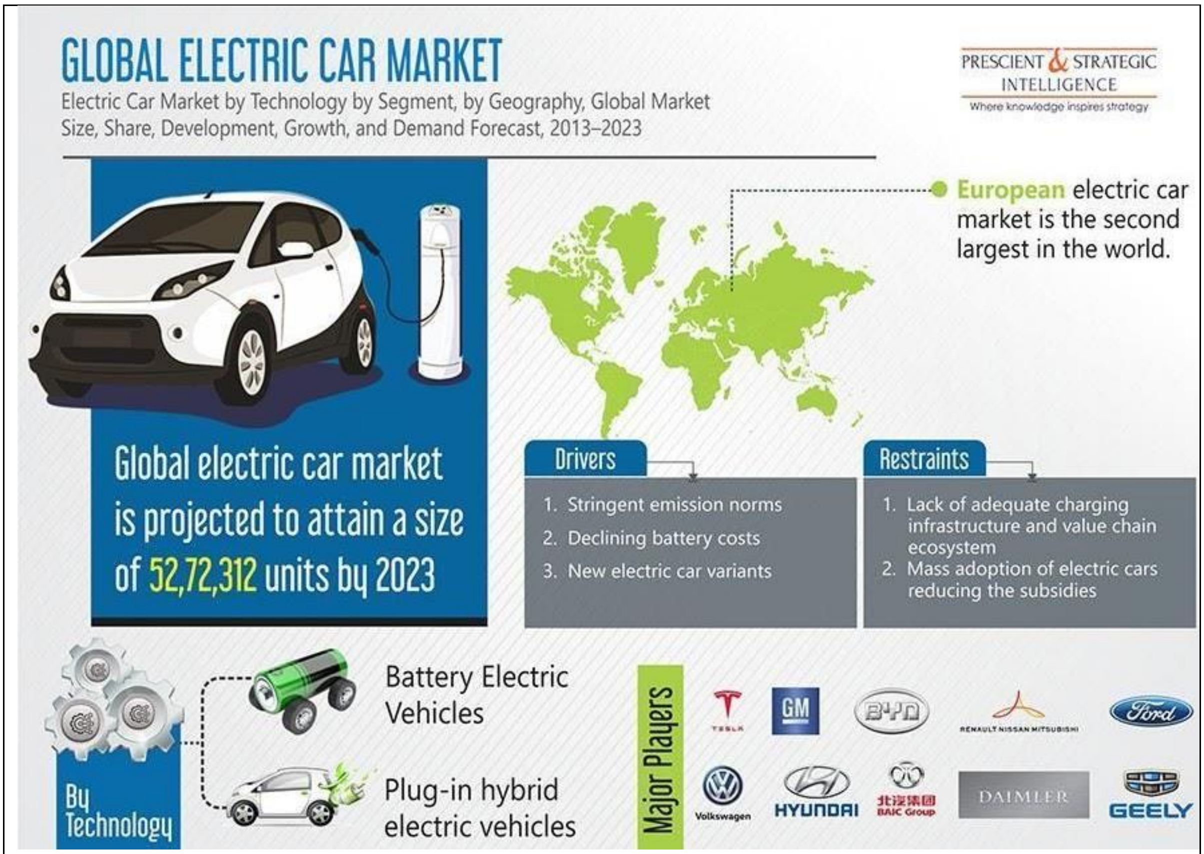
Overview:

What is Electric Vehicle?

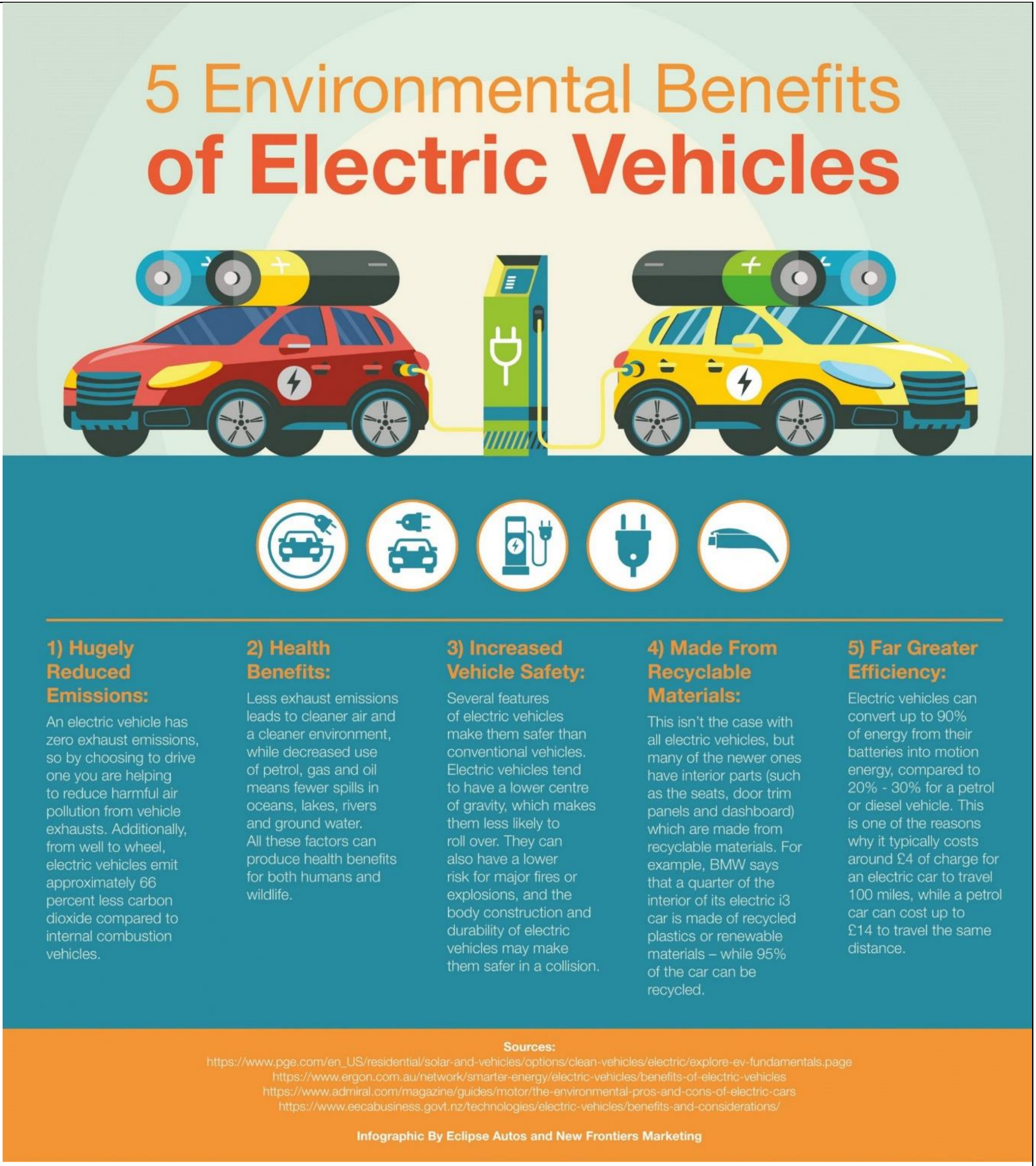
The electric vehicle is a vehicle that runs on electricity alone. Such a vehicle does not contain an internal combustion engine like the other conventional vehicles. Instead, it employs an electric motor to run the wheels. These vehicles are becoming very popular nowadays. They are considered to be a promising solution for the future transportation. The most common example is Tesla.

Business Opportunity:





Benefits of Electric Vehicles:



Data Sources:

Data was taken from the Kaggle website

[Indian Consumers Cars purchasing behaviour | Kaggle]

Market Segmentation:

1. Behavioral Segmentation:

Behavioral segmentation is a form of marketing segmentation that divides people into different groups who have a specific behavioral pattern in common. Users may share the same lifecycle stage, previously purchase particular products, or have similar reactions to your messages.

Benefits of Behavioral Segmentation

- Improves targeting accuracy
- Helps provide better-personalized experience
- Sifts engaged users from uninterested
- Saves money
- Makes it easier to track success
- Helps build loyalty to your brand

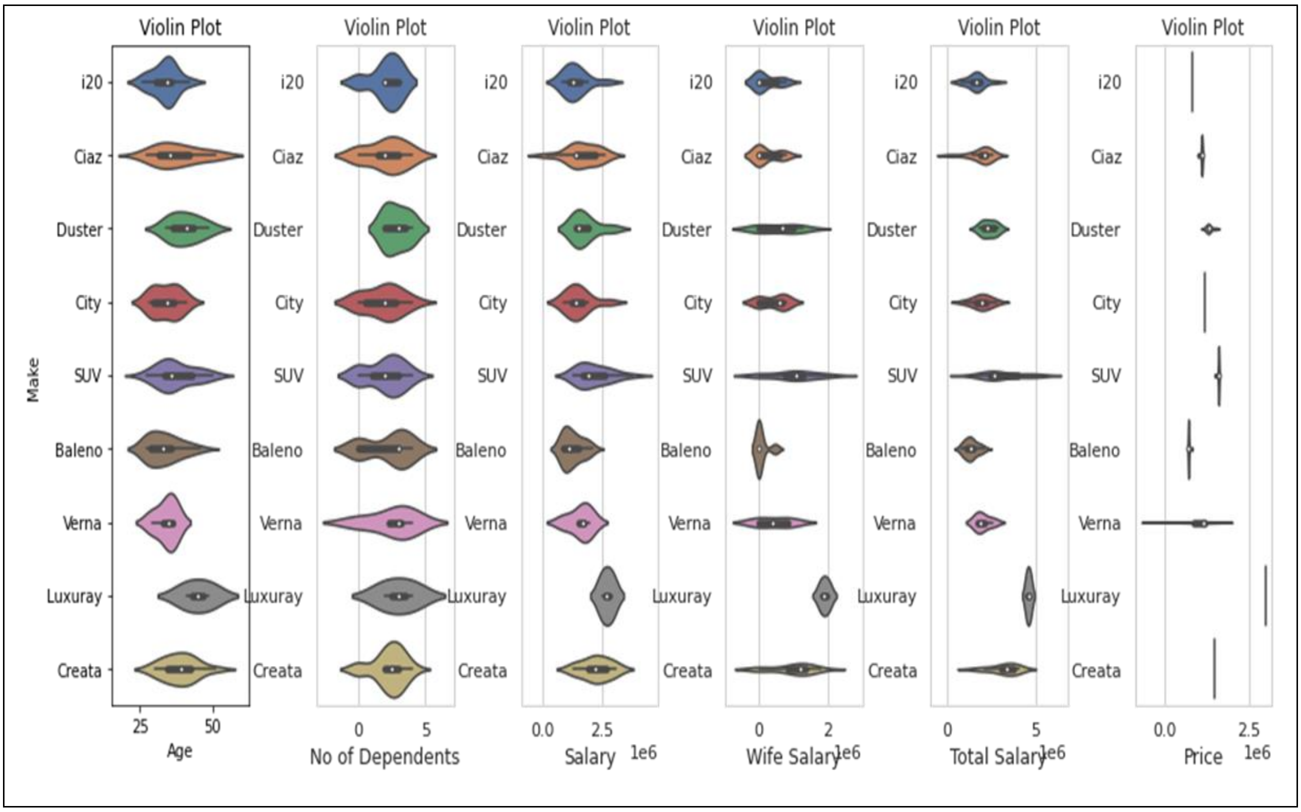
2. Psychographic Segmentation:

Psychographic segmentation's emphasis on characteristics like personality and values differs from demographic segmentation, which uses a specific trait (like gender, age, income, etc.) to categorize potential audiences.

Market researchers use psychographic characteristics to help develop and position their products and marketing messages for different target groups

Marketers use both demographics and psychographics in their market research to create their marketing strategy. So we will combine these both categories as well.

The violin plot below gives us some insight on the relation between the segmentation and descriptive variables in our data.



Observations:

Age: Younger consumers purchase less expensive vehicles. This can be explained simply as they have lesser dependents, lesser income and are single, and so they don’t have both the option and the need to buy more expensive vehicles.

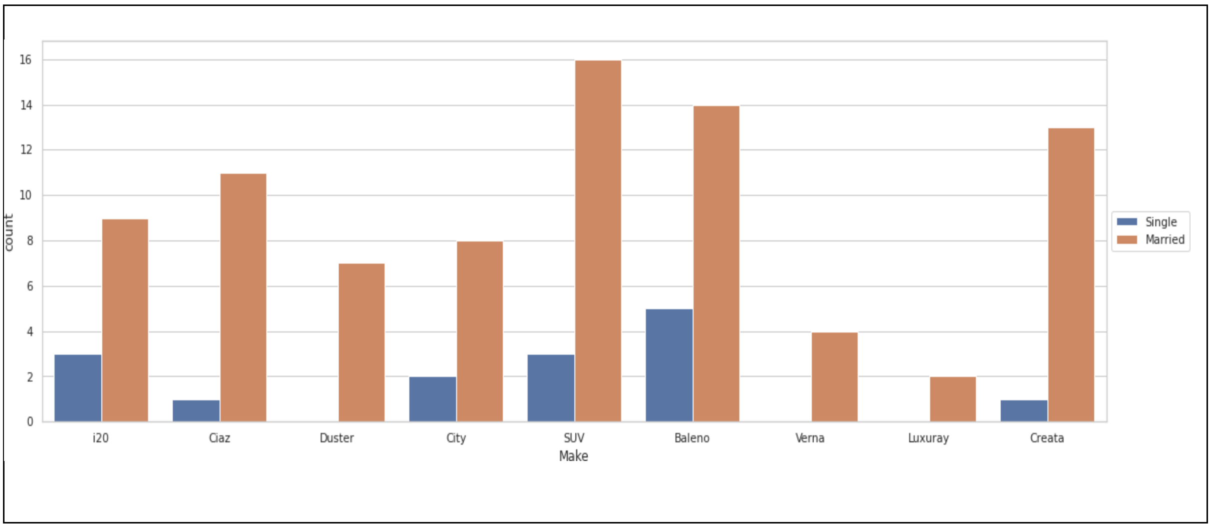
Number of Dependents: Greater number of dependents makes the consumer buy a vehicle with more seats and so they tend to prefer SUVs.

Salary: If you overlap the normalised salary plots with price plot, you would observe the median of salary violin plot matches that of the price of the vehicle indicating a very direct relationship, which makes sense asmost people would buy vehicles they can afford.

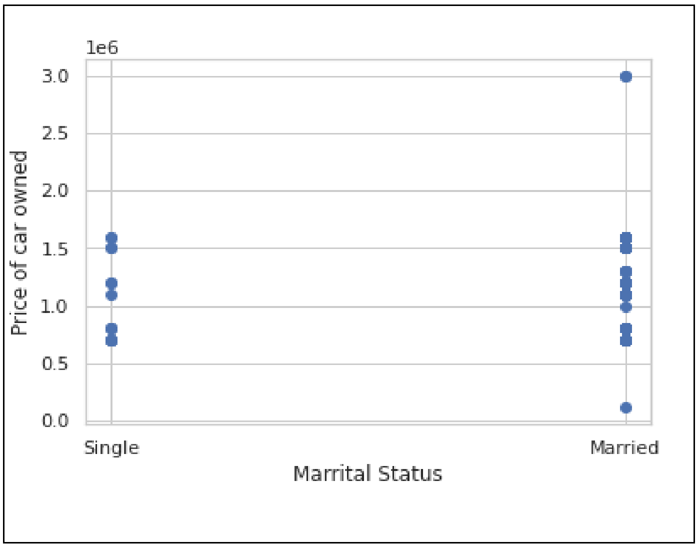
Dependency of make and price of vehicles on other descriptor variables

1) Marital Status:

Make of vehicle they tend to purchase

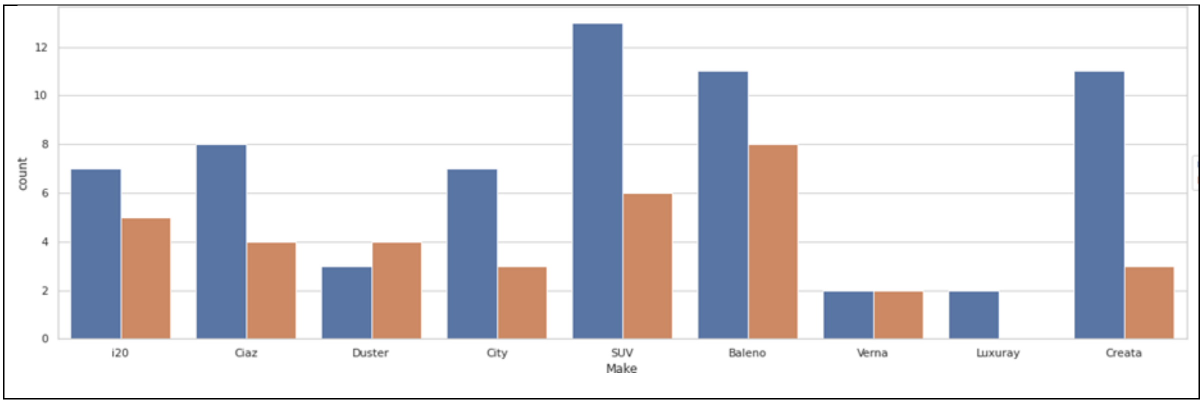


Price of vehicle they owned:

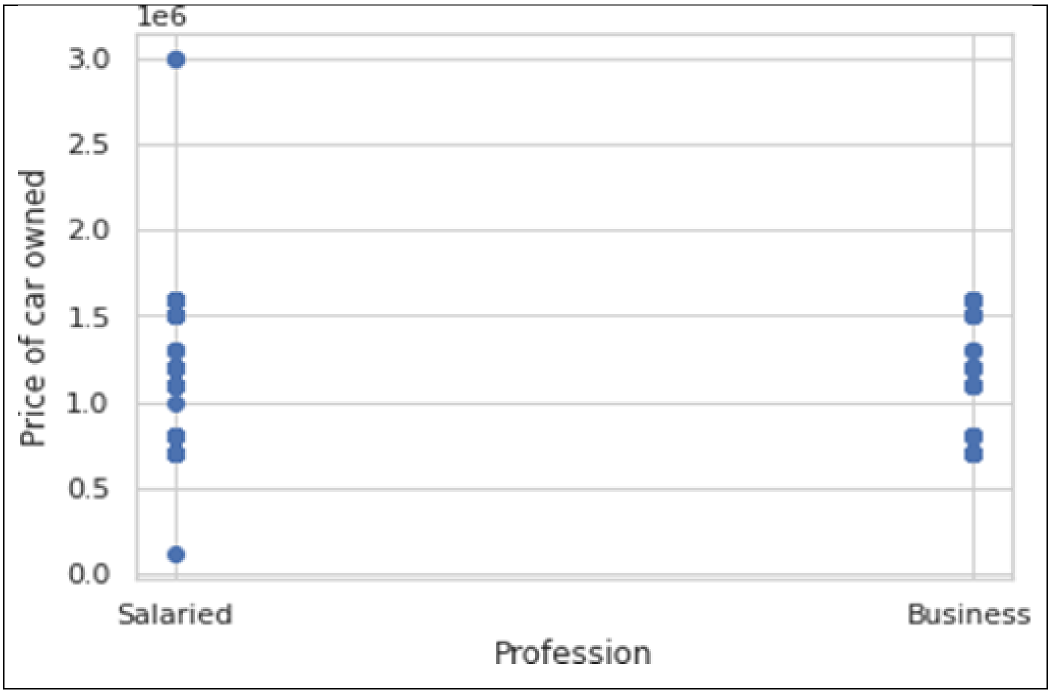


2) Profession:

Make of vehicle they tend to purchase:

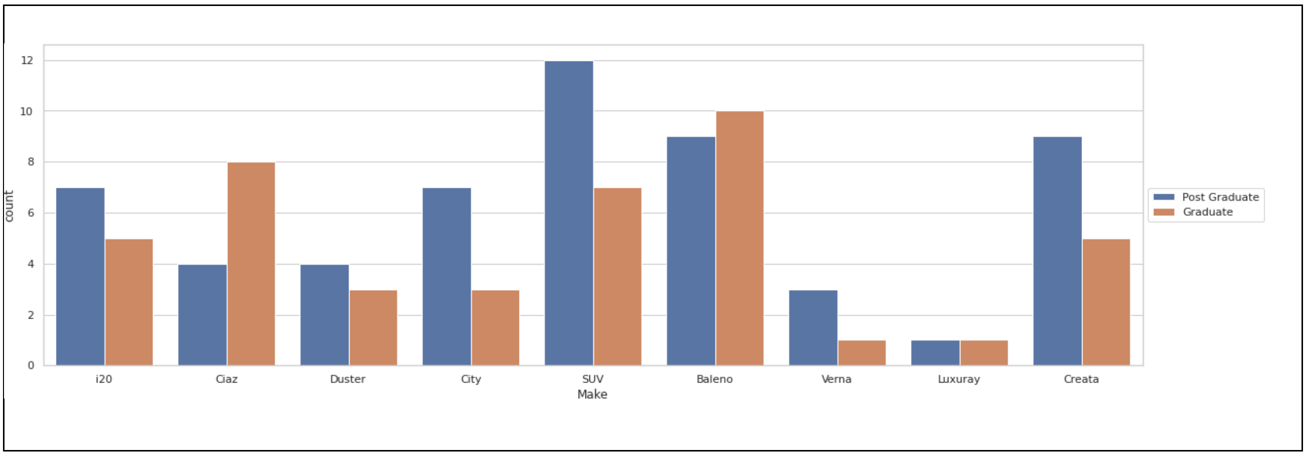


Price of vehicle owned:

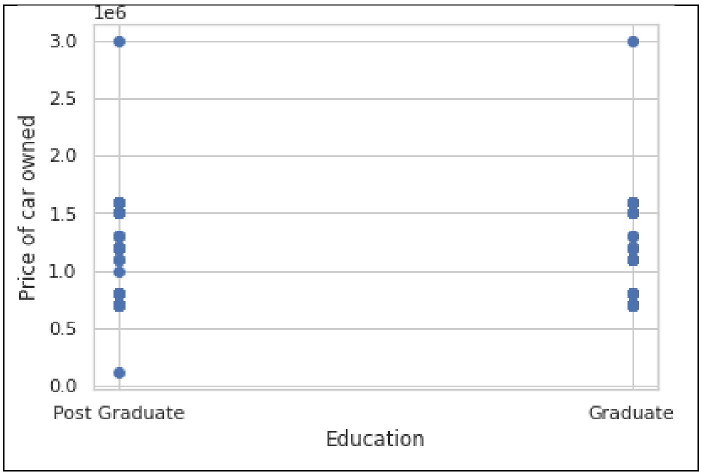


3) Education:

Make of vehicle they tend to purchase:

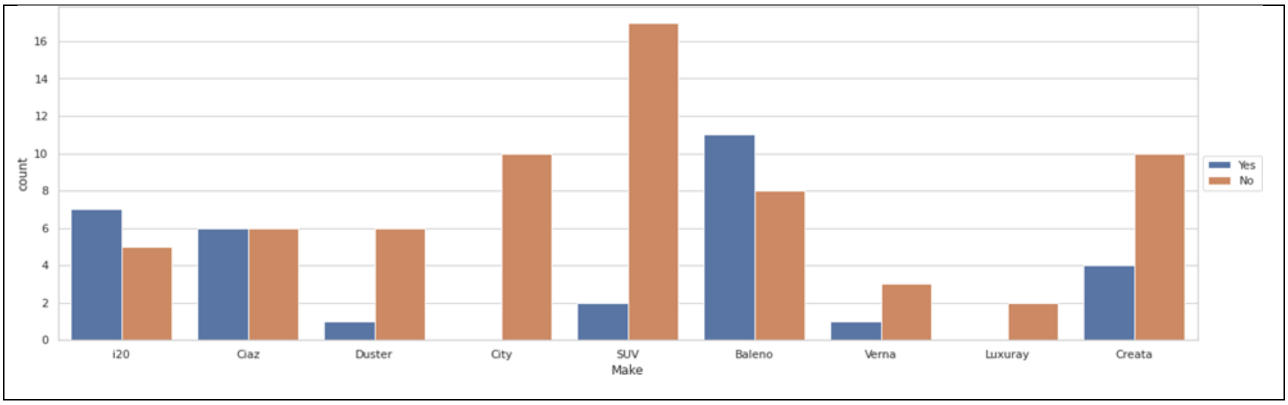


Price of vehicle owned:

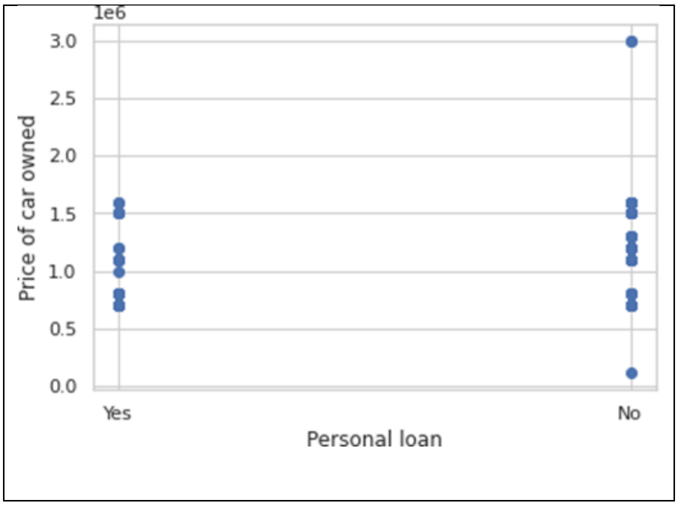


4) Personal Loan:

Make of vehicle they owned:



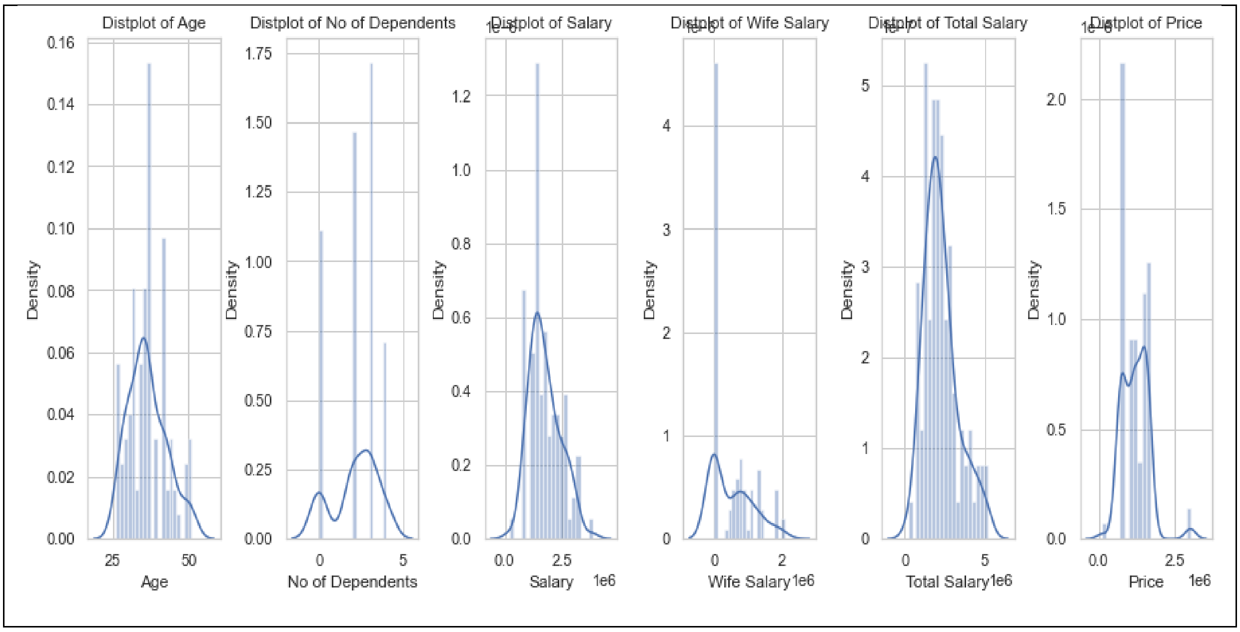
Price of Vehicle owned:



Demographic Segmentation:

Demographic segmentation is a market segmentation technique where an organization's target market is segmented based on demographic variables such as age, gender, education, income, etc. It helps organizations understand who their customers are so that their needs can be addressed more effectively. 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.

We have used the same dataset we used for behavioral and psychographic analysis and the following plots help us understand the socio-demographic structure of the market:



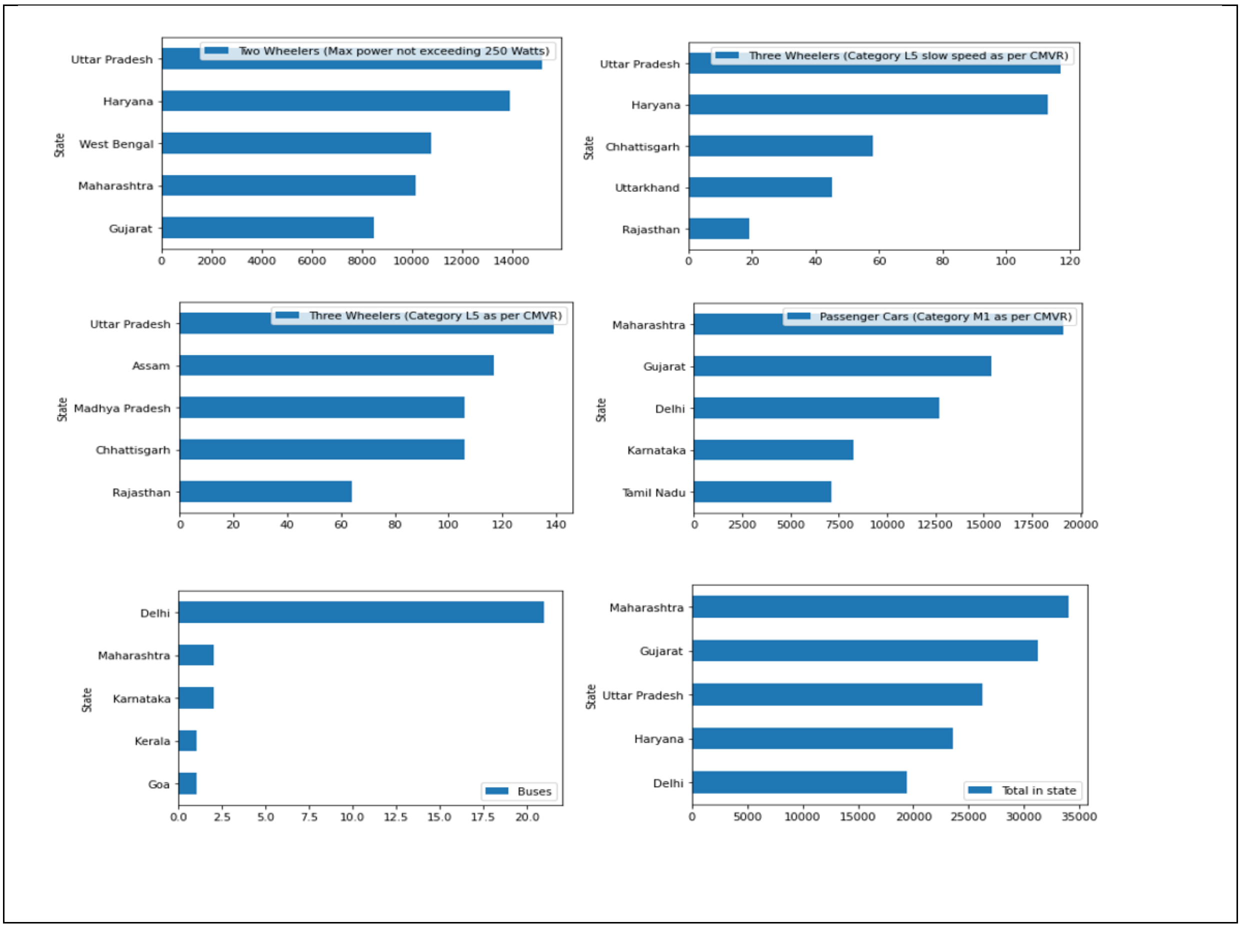
Geographic Segmentation:

Geographic segmentation is the process of dividing people into groups based on location, such as city, country, state, region, and even continent. It can help you tailor your approach during seasons customers may need your product.

For example, a fisherman in Alaska may only buy more equipment leading up to the salmon season. Whereas a fisherman in Orange Beach, Alabama, might purchase equipment all year round.

In contrast to other types of segmentation — demographic, psychographic, and behavioral — location-based segmentation analysis is easier to see results from. It doesn’t take a lot of research to identify someone’s location and the characteristics of a certain area, versus figuring out potential customers purchasing behaviors and psychographics. Here we have made divisions in terms of states and union territories in India.

For geographic analysis we used state-wise sales of different types of Electric Vehicles dataset which would help us understand our target region. Based on the type of electric vehicle, states with higher numbers of electric vehicles can be targeted as people in these states are more likely to purchase them. Given below are bar charts showing the top 5 states in sales of a particular EV type:



Depending on the type of Electric Vehicle the start-up 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

Model fitting:

K-Means Clustering Algorithm

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

What is K-Means Algorithm?

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim oif this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K centre points or centroids by an iterative process.
- Assigns each data point to its closest k-centre. Those data points which are near to the particular k-centre, create a cluster.

Data pre-processing:

The libraries used are:

- Numpy – For Computations
- Pandas – Manipulating the datasets
- Scikit-learn – For ML-based applications

```
In [1]: #Importing required libraries
        import numpy as np
        import pandas as pd
```

```
In [8]: #Label Encoding Categorical Variables
        from sklearn.preprocessing import LabelEncoder
```

With the help of this dataset, we can implement Behavioural, Psychographic and Demographic Segmentation of Indian Automobile Market. This helps us with understanding the various attributes leading to the consumer buying behaviour.

```
In [2]: #Loading the dataset
        df = pd.read_csv("Indian automoble buying behavour study 1.0.csv")
```

After uploading the ‘Indian automobile buying behaviour study 1.0’ dataset, we inspect the dataset. It has 99 rows and 13 columns.

```
In [4]: #Size of the dataset
        df.shape

Out[4]: (99, 13)
```

The various features include Age, Profession, Education, Number of dependents of the person buying the Vehicle and if the buyer is married or not, spouse is earning or not to ascertain the net income of the family. It further includes the Maker and the Price of the Vehicle of the buyer.

```
In [3]: #Inspecting the dataset
        df.head()

Out[3]:
```

	Age	Profession	Marrital Status	Education	No of Dependents	Personal loan	House Loan	Wife Working	Salary	Wife Salary	Total Salary	Make	Price
0	27	Salaried	Single	Post Graduate	0	Yes	No	No	800000	0	800000	i20	800000
1	35	Salaried	Married	Post Graduate	2	Yes	Yes	Yes	1400000	600000	2000000	Ciaz	1000000
2	45	Business	Married	Graduate	4	Yes	Yes	No	1800000	0	1800000	Duster	1200000
3	41	Business	Married	Post Graduate	3	No	No	Yes	1600000	600000	2200000	City	1200000
4	31	Salaried	Married	Post Graduate	2	Yes	No	Yes	1800000	800000	2600000	SUV	1600000

We then use describe() function to gain in-depth insight about the mean, median and other statistical figures about the attributes of the dataset. This gives us an idea about the buyers’ essential ranges.

In [5]:
```
#Statistics for Numerical Variables
df.describe()
```

Out[5]:

	Age	No of Dependents	Salary	Wife Salary	Total Salary	Price
count	99.000000	99.000000	9.900000e+01	9.900000e+01	9.900000e+01	9.900000e+01
mean	36.313131	2.181818	1.736364e+06	5.343434e+05	2.270707e+06	1.194040e+06
std	6.246054	1.335265	6.736217e+05	6.054450e+05	1.050777e+06	4.376955e+05
min	26.000000	0.000000	2.000000e+05	0.000000e+00	2.000000e+05	1.100000e+05
25%	31.000000	2.000000	1.300000e+06	0.000000e+00	1.550000e+06	8.000000e+05
50%	36.000000	2.000000	1.600000e+06	5.000000e+05	2.100000e+06	1.200000e+06
75%	41.000000	3.000000	2.200000e+06	9.000000e+05	2.700000e+06	1.500000e+06
max	51.000000	4.000000	3.800000e+06	2.100000e+06	5.200000e+06	3.000000e+06

There aren’t any missing values or irrelevant attributes and thus needs no data handling.

In most of the algorithms, categorical values cannot be handled. Thus, there is a need to convert these categorical values to numerical levels. For this, we make use of the Label Encoder function.

In [8]:
```
#Label Encoding Categorical Variables
from sklearn.preprocessing import LabelEncoder
labelen = LabelEncoder()
df['Profession'] = labelen.fit_transform(df['Profession'])
df['Marrital Status'] = labelen.fit_transform(df['Marrital Status'])
df['Education'] = labelen.fit_transform(df['Education'])
df['Personal loan'] = labelen.fit_transform(df['Personal loan'])
df['House Loan'] = labelen.fit_transform(df['House Loan'])
df['Wife Working'] = labelen.fit_transform(df['Wife Working'])
```

In [9]:
```
#Encoded dataset
df.head()
```

Out[9]:

	Age	Profession	Marrital Status	Education	No of Dependents	Personal loan	House Loan	Wife Working	Salary	Wife Salary	Total Salary	Make	Price
0	27	1	1	1	0	1	0	0	800000	0	800000	i20	800000
1	35	1	0	1	2	1	1	1	1400000	600000	2000000	Ciaz	1000000
2	45	0	0	0	4	1	1	0	1800000	0	1800000	Duster	1200000
3	41	0	0	1	3	0	0	1	1600000	600000	2200000	City	1200000
4	31	1	0	1	2	1	0	1	1800000	800000	2600000	SUV	1600000

ML algorithms work better when feature values are on relatively on a similar scale and close to normalized distribution. Using StandardScaler(), we scale the entire numerical portion of the dataset for an enhanced productive result.

In [10]:

```
#Scaling the dataset
from sklearn.preprocessing import StandardScaler
scaled_df=df
scaled_df[["Age","Salary","Wife Salary","Total Salary","Price"]]=StandardScaler().fit_transform(df[["Age","Salary","Wife Salary",
scaled_df
```

Out[10]:

	Age	Profession	Marrital Status	Education	No of Dependents	Personal loan	House Loan	Wife Working	Salary	Wife Salary	Total Salary	Make	Price
0	-1.498630	1	1	1	0	1	0	0	-1.397118	-0.887055	-1.406760	i20	-0.904843
1	-0.211304	1	0	1	2	1	1	1	-0.501877	0.108995	-0.258937	Ciaz	-0.445579
2	1.397855	0	0	0	4	1	1	0	0.094950	-0.887055	-0.450240	Duster	0.013685
3	0.754191	0	0	1	3	0	0	1	-0.203464	0.108995	-0.067633	City	0.013685
4	-0.854967	1	0	1	2	1	0	1	0.094950	0.441012	0.314975	SUV	0.932213
...	...	...	...	...	...	...	...	...	...	...	...	...	...
94	-1.498630	0	1	0	0	0	0	0	0.990190	-0.887055	0.123671	SUV	0.932213
95	2.202434	1	0	1	3	0	0	1	3.079085	1.271054	2.706274	SUV	0.932213
96	2.363350	0	0	0	2	1	1	0	0.691777	-0.887055	-0.067633	Ciaz	-0.215947
97	2.363350	1	0	1	2	0	0	1	1.437811	1.271054	1.654102	Creata	0.702581
98	2.363350	1	0	1	2	1	1	0	0.691777	-0.887055	-0.067633	Ciaz	-0.215947

99 rows × 13 columns

Libraries used for the algorithm:

K - Means Algorithm

In [37]:

```
# Importing Important Libraries
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

In [38]:

```
X_scaled = StandardScaler().fit_transform(obj_df)
X_scaled = pd.DataFrame(X_scaled,columns=['Age', 'Profession', 'Marrital Status', 'Education', 'No of Dependents',
                                          'Personal loan', 'House Loan', 'Wife Working', 'Salary', 'Wife Salary',
                                          'Total Salary','Price'])

x = X_scaled.to_numpy()
X_scaled
```

Out[38]:

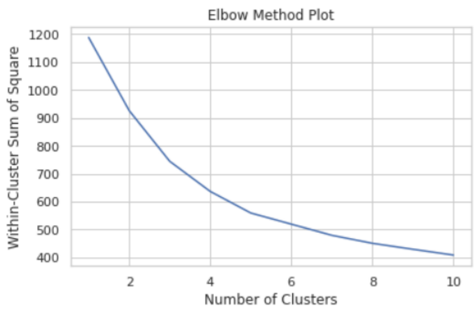
	Age	Profession	Marrital Status	Education	No of Dependents	Personal loan	House Loan	Wife Working	Salary	Wife Salary	Total Salary	Price
0	-1.498630	-0.739510	-2.366432	0.876275	-1.642313	1.446980	-0.772512	-1.051847	-1.397118	-0.887055	-1.406760	-0.904843
1	-0.211304	-0.739510	0.422577	0.876275	-0.136859	1.446980	1.294479	0.950708	-0.501877	0.108995	-0.258937	-0.445579
2	1.397855	1.352247	0.422577	-1.141195	1.368594	1.446980	1.294479	-1.051847	0.094950	-0.887055	-0.450240	0.013685
3	0.754191	1.352247	0.422577	0.876275	0.615867	-0.691095	-0.772512	0.950708	-0.203464	0.108995	-0.067633	0.013685
4	-0.854967	-0.739510	0.422577	0.876275	-0.136859	1.446980	-0.772512	0.950708	0.094950	0.441012	0.314975	0.932213
...	...	...	...	...	...	...	...	...	...	...	...	...
94	-1.498630	1.352247	-2.366432	-1.141195	-1.642313	-0.691095	-0.772512	-1.051847	0.990190	-0.887055	0.123671	0.932213
95	2.202434	-0.739510	0.422577	0.876275	0.615867	-0.691095	-0.772512	0.950708	3.079085	1.271054	2.706274	0.932213
96	2.363350	1.352247	0.422577	-1.141195	-0.136859	1.446980	1.294479	-1.051847	0.691777	-0.887055	-0.067633	-0.215947
97	2.363350	-0.739510	0.422577	0.876275	-0.136859	-0.691095	-0.772512	0.950708	1.437811	1.271054	1.654102	0.702581
98	2.363350	-0.739510	0.422577	0.876275	-0.136859	1.446980	1.294479	-1.051847	0.691777	-0.887055	-0.067633	-0.215947

99 rows × 12 columns

```
In [39]: wcss = []

         for i in range(1, 11):
             kmeans = KMeans(n_clusters = i, init = 'k-means++',
                             max_iter = 300, n_init = 10,random_state = 0)
             kmeans.fit(X_scaled)
             wcss.append(kmeans.inertia_)
```

```
In [40]: plt.plot(range(1, 11), wcss)
         plt.title('Elbow Method Plot')
         plt.xlabel('Number of Clusters')
         plt.ylabel('Within-Cluster Sum of Square') # Within cluster sum of squares
         plt.tight_layout()
         plt.show()
```



Either take K=3 or K= 5

1) when k= 3

k = 3

```
In [43]: kmeans = KMeans(n_clusters = 3, init = 'k-means++',
                         max_iter = 300, n_init = 10,random_state = 42)
         kmeans.fit(X_scaled)

C:\Users\dhruv\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OM
P_NUM_THREADS=1.
  warnings.warn(

Out[43]: KMeans
         KMeans(n_clusters=3, random_state=42)
```

```
In [44]: y = kmeans.predict(X_scaled)
         y_df = pd.DataFrame(y,columns=['Class'])
```

```
In [45]: final_data = pd.concat([df,y_df],axis=1)
         final_data
```

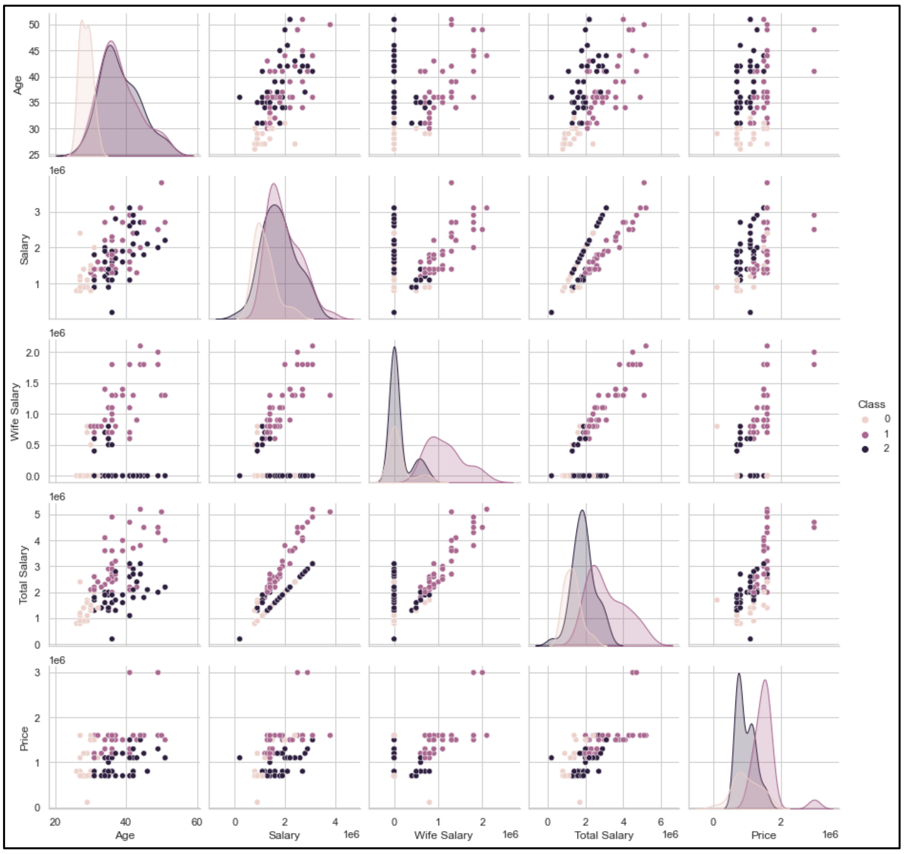
Out[45]:

	Age	Profession	Marrital Status	Education	No of Dependents	Personal loan	House Loan	Wife Working	Salary	Wife Salary	Total Salary	Make	Price	Class
0	27	Salaried	Single	Post Graduate	0	Yes	No	No	800000	0	800000	i20	800000	0
1	35	Salaried	Married	Post Graduate	2	Yes	Yes	Yes	1400000	600000	2000000	Ciaz	1000000	2
2	45	Business	Married	Graduate	4	Yes	Yes	No	1800000	0	1800000	Duster	1200000	2
3	41	Business	Married	Post Graduate	3	No	No	Yes	1600000	600000	2200000	City	1200000	1

Class 0: total salary equals to husband salary

Class 1: total salary is greater than husband salary

Class 2: total salary is nearly equal and greater than husband salary



2) when k= 5

k = 5

In [48]:
```
kmeans1 = KMeans(n_clusters = 5, init = 'k-means++',
                 max_iter = 300, n_init = 10,random_state = 42)
kmeans1.fit(X_scaled)
```

```
C:\Users\dhruv\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1332: UserWarning: KMeans is known to have a memory leak
on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OM
P_NUM_THREADS=1.
  warnings.warn(
```

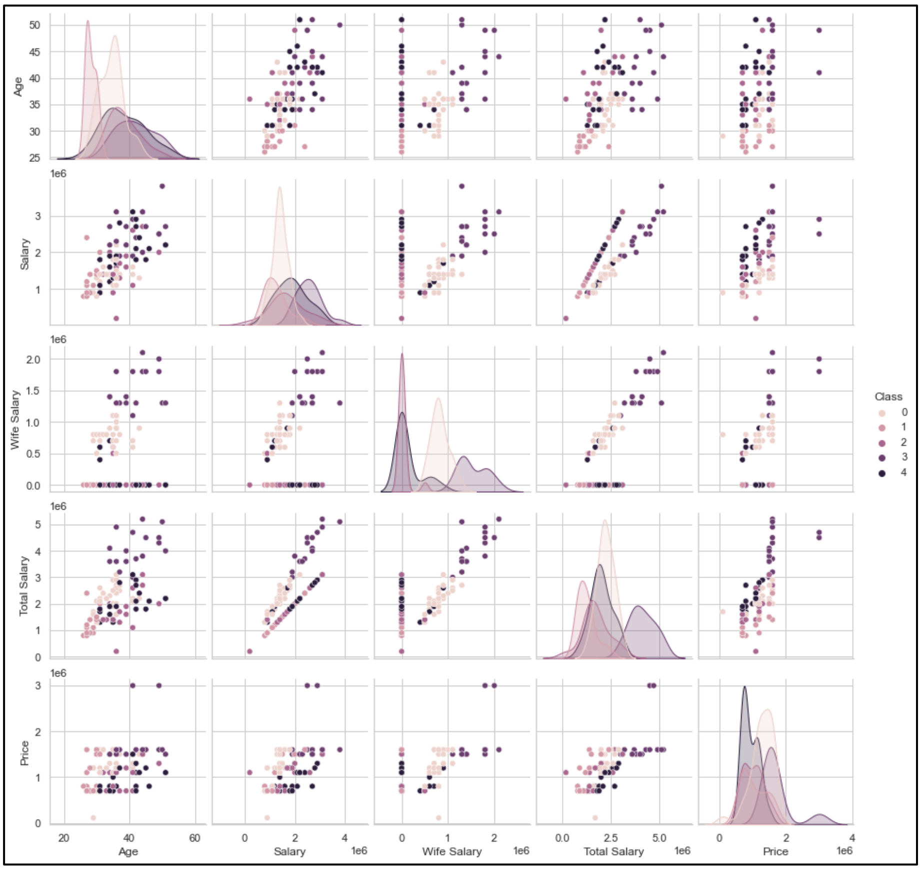
Out[48]:
```
KMeans
KMeans(n_clusters=5, random_state=42)
```

In [49]:
```
y1 = kmeans1.predict(X_scaled)
y1_df = pd.DataFrame(y1,columns=['Class'])
```

In [50]:
```
final_data1 = pd.concat([df,y1_df],axis=1)
final_data1
```

Out[50]:

	Age	Profession	Marrital Status	Education	No of Dependents	Personal loan	House Loan	Wife Working	Salary	Wife Salary	Total Salary	Make	Price	Class
0	27	Salaried	Single	Post Graduate	0	Yes	No	No	800000	0	800000	i20	800000	1
1	35	Salaried	Married	Post Graduate	2	Yes	Yes	Yes	1400000	600000	2000000	Ciaz	1000000	4
2	45	Business	Married	Graduate	4	Yes	Yes	No	1800000	0	1800000	Duster	1200000	4



To reduce the homogeneity, we chose k= 3

<u>PROFILING AND DESCRIBING POTENTIAL SEGMENTS (Done by Adnan Habib)</u>

When profiling the segments, we attempt to examine the qualities of the instances within each segment and identify the traits that the data points inside the clusters have in common. To estimate the range of values in each segment, for numerical attributes, we find the mean of that particular attribute as well as its maximum and minimum values.

Additionally, for categorical attributes, we identify the mode of classes in each segment's attribute and generalise the segment's characteristic as the mode. And if we discover that two or more classes of an attribute are present in the segment with nearly same frequencies, we take into account all of the classes with nearly identical frequencies to the mode class.

1. SEGMENT 0

Number of instances in segment 0 is 18

Numerical Value Summary:

Segment 0	Range of Values	Average value
AccelSec	5.1 - 10	8.0
Top Speed (KmH)	140 – 200	164
Range (Km)	180 – 440	325
Efficiency (WhKm)	153 – 232	179.8
Fast Charge (KmH)	210 - 560	396.5
Price Euro	25500 - 65000	40049.6
Seats	5-seater	5

Categorical Value Summary:

Segment 0	Mode Class(s)
Power Train	FWD, RWD
Body Style	Hatch Back, SUV
Segment	C

We can see the range of values for each numerical attribute from the summary above. Additionally, we can see that segment 0 only has a 5-seater seat. And segment 0 EVs, which are FWD, RWD, SUV, and hatchback-style vehicles, belong to segment C. These are referred to as medium automobiles in segment C. The interior room and outward compactness of C-segment vehicles are well-balanced.

In segment 0, we can observe that the majority of the vehicles are classified as medium or family vehicles and belong to sector C. These vehicles are recommended for those with a reasonable budget because they are not significantly more expensive than section 3 vehicles in terms of pricing in Euros, which ranges from 25500 to 65,000. And five-seater vehicles are common.

2. SEGMENT 1

Number of instances in segment 1 is 25

Numerical Value Summary:

Segment 1	Range of Values	Average value
AccelSec	7.3 – 22.4	9.92
Top Speed (KmH)	123 - 167	147.68
Range (Km)	160 - 400	278.15
Efficiency (WhKm)	154 - 273	178.6
Fast Charge (KmH)	190 - 435	308.7
Price Euro	29146 - 70631	36443
Seats	4 – 7-seater	5.15

Categorical Value Summary:

Segment 1	Mode Class(s)
Power Train	FWD
Body Style	SUV
Segment	B

We can see the range of values for each numerical attribute from the summary above. And since the average for segments 1 is 5.15, we can deduce that there are more seats in segments 4 and 5 than there are in the remaining segments. Additionally, segment B-class EVs in segment 1 have a FWD powertrain and an SUV body style.

Sedans and hatchbacks in the same price range lack the street presence of segment B SUVs. They aren't overly tough or even overly powerful, though.

3. SEGMENT 2

Number of instances in segment 2 is 34

Numerical Value Summary:

Segment 2	Range of Values	Average value
AccelSec	2.8 – 7.5	5.64
Top Speed (KmH)	160 - 250	195.44
Range (Km)	280 - 750	393.96
Efficiency (WhKm)	171 - 270	217.48
Fast Charge (KmH)	340 - 930	534.14

Price Euro	45000- 102990	65418.96
Seats	5 - 7-seater	5.37

Categorical Value Summary:

Segment 2	Mode Class(s)
Power Train	AWD
Body Style	SUV
Segment	D, E

We can see the range of values for each numerical attribute from the summary above. We can observe that segment 2 seats are 5, 6, and 7 seats, and since the average is 5.37, we can deduce that there are more 5 seaters than there are other seats in segment 2. The segment D and E EVs in segment 2 have an AWD powertrain and an SUV body style. Segment D: Mid-sized family vehicles with an abundance of luxury features that come at the expense of the drivetrain and engine, making them more difficult to operate.

Segment E: Executive luxury cars are denoted by the letter E. Compared to mid-size cars, they are substantially longer. In India, e-segment cars contain some enormous and huge automobiles as well as some sumptuous rides with a lengthy wheelbase. These are well-known among business people since they begin with the letter E and radiate luxury and quality.

4. SEGMENT 3

Number of instances in segment 3 is 15

Numerical Value Summary:

Segment 3	Range of Values	Average value
AccelSec	6.5 – 12.7	9.86
Top Speed (KmH)	130 - 160	142.33
Range (Km)	95 - 440	207.33
Efficiency (WhKm)	156 - 181	168.2
Fast Charge (KmH)	170 - 590	327.2
Price Euro	20129 - 45000	30655.6
Seats	2 - 4-seater	3.73

Categorical Value Summary:

Segment 3	Mode Class(s)
Power Train	RWD
Body Style	Hatch Back
Segment	A, B

We can see the range of values for each numerical attribute from the summary above. And since the average for segment 3 is 3.73, we can deduce that there are more seats in segment 3 with four seats than there are in segment 1 with two or three seats. And the segment 3 electric vehicles (EVs) that belong to segments A and B have a RWD powertrain and a hatchback body style.

Segment A: Because A-segment vehicles are more focused on affordability than on size or amenities, they are compact. These have compact, effective engines.

Segment B: If your driving needs are everyday commuting and sporadic lengthy trips, B-segment automobiles or small hatchbacks are suitable for you. They provide additional steadiness on the road due to their slightly larger size compared to A-segment vehicles.

5. SEGMENT 4

Number of instances in segment 4 is 11

Numerical Value Summary:

Segment 4	Range of Values	Average value
AccelSec	2.1 - 10	4.03
Top Speed (KmH)	150 - 410	247.5
Range (Km)	310 - 970	474
Efficiency (WhKm)	104 - 223	184.1
Fast Charge (KmH)	540 - 940	736.9
Price Euro	46380 - 215000	109193
Seats	4- and 5-seater	4.56

Categorical Value Summary:

Segment 4	Mode Class(s)
Power Train	AWD
Body Style	Sedan
Segment	F

We can see the range of values for each numerical attribute from the summary above. We can see that segment 4 seats are 4- and 5-seaters, and since the average is 4.56, we may deduce that segment 4 has slightly more 5-seaters than 4-seaters. Additionally, the segment 4 EVs are AWD-powered sedans that are part of the section F.

F-class vehicles are renowned for having incredible designs, superior handling, superior comfort, and higher performance.

TARGET SEGMENT (Done by Adnan Habib)

The target segment which we choose from the 5 segments is segment 2. This segment consists of 34 instances. The summary of the segment 2 is:

Number of instances in segment 2 is 34

Numerical Value Summary:

Segment 2	Range of Values	Average value
AccelSec	2.8 – 7.5	5.64
Top Speed (KmH)	160 - 250	195.44
Range (Km)	280 - 750	393.96
Efficiency (WhKm)	171 - 270	217.48
Fast Charge (KmH)	340 - 930	534.14
Price Euro	45000- 102990	65418.96
Seats	5 - 7-seater	5.37

Categorical Value Summary:

Segment 2	Mode Class(s)
Power Train	AWD
Body Style	SUV
Segment	D, E

These vehicles have an acceleration range of 2.8 to 7.5 m/s^2. In segment 2, these vehicles' top speeds range from 160 to 250 km/h. Additionally, the efficiency of vehicles in this sector ranges from 171 to 270 WhKm. Additionally, the price range in this market category is between 45,000 and 102,990 euros. The most popular chairs in this group are the five-seaters.

An all-wheel drive (AWD) vehicle is one having a powertrain capable of supplying power to all of its wheels, whether continuously or on demand. The engine train of these segment cars is AWD. The body type for the vehicles in this sector is SUV (A sport utility vehicle or SUV is a car classification that combines elements of road-going passenger cars with features from off-road vehicles, such as raised ground clearance and four-wheel drive). Cars in this area are under segments D and E, therefore they range from luxury vehicles to mid-sized family vehicles.

Segment 2 is the best market segment to enter; therefore, if we produce cars with top speeds between 160 and 250 km/h and acceleration between 2.5 and 7.8 m/s^2, we will be more successful in the market. Additionally, vehicles with all-wheel drive (AWD) and SUV body types appear to boost sales. Five-seaters appear to be the most popular seats on the market, according to the data. Therefore, it appears that we can be more successful in the market with these car attributes.

<u>Marketing Mix with regard to EV:</u>

PRICE

Affordability is the number one issue for any vehicle, more so in the case of EVs. The more cost efficient a product is the more it's sale. We can see from the above analysis that the product's price should ideally range between 10 to 20 lakh, as most people would make a purchase in this range.

PRODUCT

Product totally depends on the start-up, it's design, it's mechanics. Having said that, in general, if an EV start-up has to get successful in India, it's key would be to get into 2-wheeler EV business.

Another type of product EV Start up can look into is public transport vehicles, because the current government policies are supportive for revamping public transport to electric-based engines

PLACE

Major cities of the country (Especially metropolitan cities) 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 aware population willing to buy Electric Vehicles.

For different types of vehicles, the list of top states which will promise a good market have been given in our geographical analysis.

PROMOTION

In EV business, awareness is the key, with its edge over fuel using vehicles, more and more people should be made aware of its advantages.

ELECTRIC VEHICLE – CODE IMPLEMENTATION

GitHub Link: Adnan232/Electric-Vehicle-Segmentation-Analysis (github.com)