ELECTRIC VEHICLE MARKET SEGMENTATION ANALYSIS

Satyam Gaikwad

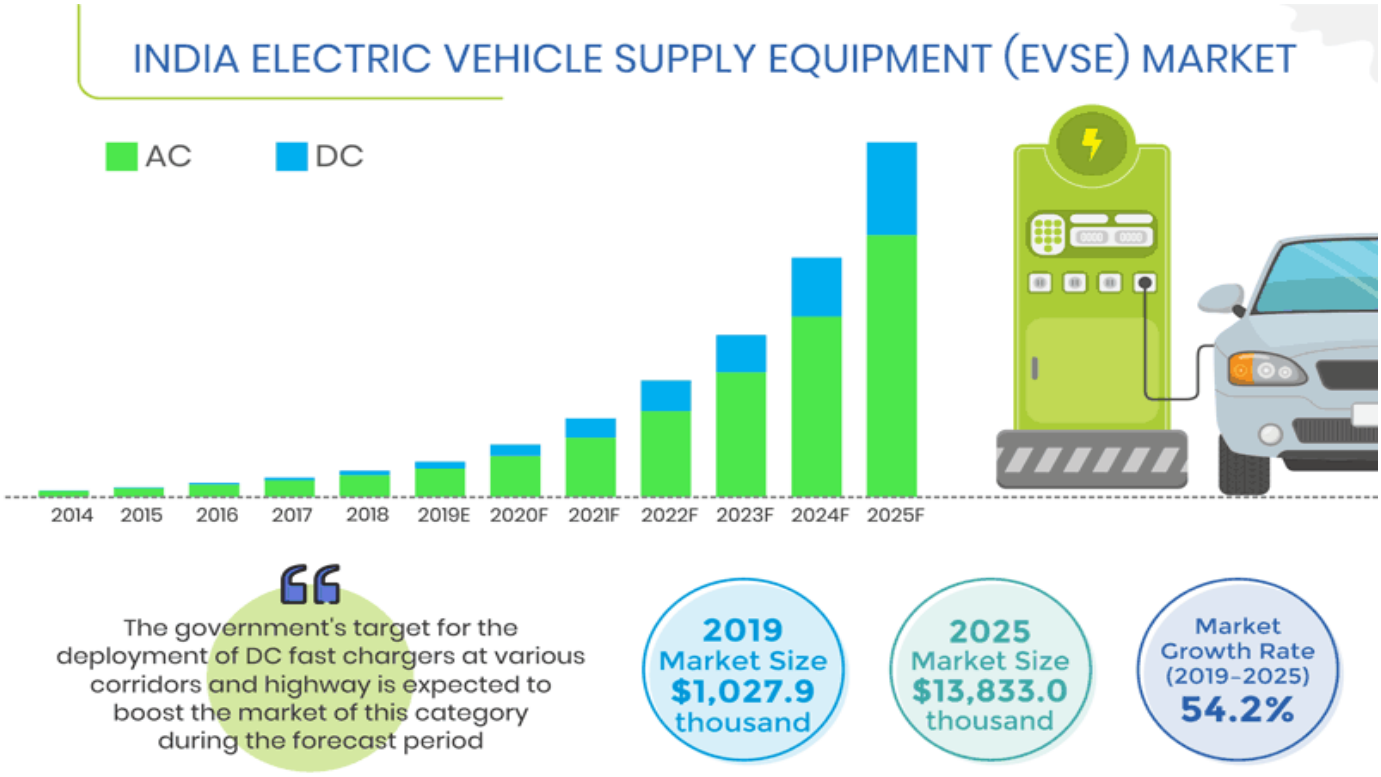
Shivraj Kumar

Tanu Majumder

Pooja Yadav

Introduction

The global automotive industry is undergoing a significant transformation, driven by the shift towards electric vehicles (EVs) and the increasing focus on sustainability and environmental concerns. India, as one of the world's largest markets for automobiles, is also embracing this trend and witnessing a significant rise in the adoption of EVs. According to a report by Niti Aayog, a policy thinks tank of the Government of India, the country aims to achieve 100% electric mobility by 2030, which represents a massive opportunity for EV startups.



Source: Internet

In this context, this report focuses on analyzing the Indian EV market using segmentation analysis and developing a feasible strategy for an EV startup to enter the market. Segmentation analysis is a crucial tool for any business to identify the right customer segments to target and develop an effective marketing strategy. In the context of the EV market in India, segmentation analysis involves identifying different segments of customers based on geographic, demographic, psychographic, and behavioral factors.

The report starts by providing an overview of the Indian EV market, including the current state, key drivers, and challenges. It then outlines the different segmentation criteria and methodologies used to segment the EV market in India. The report analyzes the market based on these different segments, such as geographic, demographic, psychographic, and behavioral, to identify the most attractive customer segments for an EV startup to target.

The report recommends a feasible strategy for the EV startup to enter the market by identifying the segments with the highest potential demand for EVs, such as major cities, higher-income groups, early adopters, and environmentally conscious individuals. The report also recommends focusing on developing electric cars, two-wheelers, buses, and commercial vehicles and targeting individual consumers, fleet operators, and government agencies.

The report concludes by emphasizing the importance of segmentation analysis for any EV startup looking to enter the Indian market. It is crucial to identify the right customer segments, develop an effective marketing strategy, and deliver a compelling value proposition to succeed in this highly competitive and rapidly evolving market.

Market segmentation

Market segmentation is the process of dividing a larger market into smaller groups of consumers with similar needs and characteristics. This helps businesses tailor their marketing efforts to specific customer groups, ultimately leading to more effective marketing campaigns and higher profits.

In the context of the electric vehicle (EV) market in India, market segmentation involves identifying different segments of customers based on various factors such as geographic, demographic, psychographic, and behavioral. Each of these segments has its unique characteristics and needs and targeting them requires a different approach.

Geographic segmentation is based on geographic factors such as region, city, and climate. For the Indian EV market, major cities such as Delhi, Mumbai, Bangalore, Hyderabad, and Pune have higher demand for EVs due to better infrastructure, higher income levels, and environmental awareness. In contrast, rural areas may have less demand due to lack of infrastructure and lower income levels.

Demographic segmentation is based on factors such as age, income, education, and occupation. Younger age groups, higher income levels, and better-educated individuals are more likely to buy EVs in India due to their awareness of environmental issues and willingness to try new technologies.

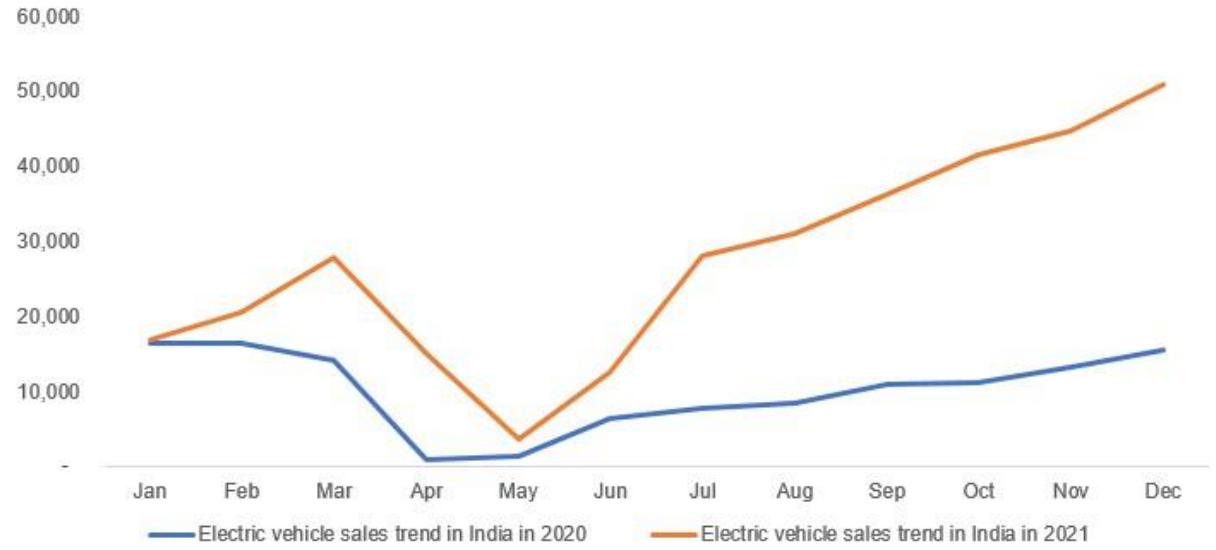
Psychographic segmentation is based on factors such as values, attitudes, and lifestyle. Environmental consciousness and social responsibility are increasingly important factors that influence the buying decisions of customers in India. Innovators and early adopters are also important segments to target, as they are more likely to appreciate the innovation and sustainability that EVs represent.

Behavioral segmentation is based on factors such as usage, benefits, and loyalty. Customers who prioritize cost savings, environmental concerns, performance, and convenience are key drivers of EV adoption. By targeting these segments, businesses can effectively promote the benefits of EVs, such as lower maintenance costs, reduced emissions, and better performance.

In summary, market segmentation is a crucial tool for any EV startup looking to enter the Indian market. By identifying the right customer segments based on geographic, demographic, psychographic, and behavioral factors, businesses can tailor their marketing efforts to meet the needs of specific customer groups and ultimately succeed in this rapidly evolving and highly competitive market.

Market Analysis

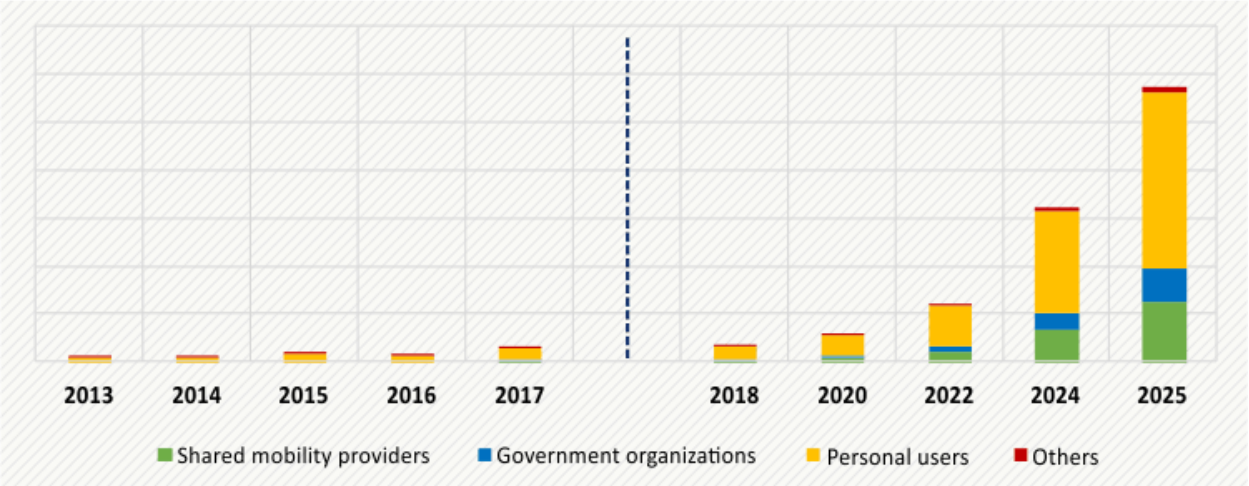
The electric vehicle (EV) market in India is rapidly growing, driven by the increasing focus on sustainability and environmental concerns. According to a report by the Society of Manufacturers of Electric Vehicles (SMEV), the sales of EVs in India grew by 20% in 2020, despite the COVID-19 pandemic. This represents a significant opportunity for EV startups looking to enter the Indian market.



Source: https://www.ibef.org/blogs/electric-vehicles-market-in-india

The Indian government is also promoting the adoption of EVs through various policies and initiatives. The Faster Adoption and Manufacturing of Hybrid and Electric Vehicles (FAME) scheme, launched in 2015, provides incentives for the purchase of EVs and supports the development of EV infrastructure. The government has also set a target to achieve 30% EV penetration by 2030, further boosting the demand for EVs in India.

The Indian EV market is still in its early stages, and most sales are currently concentrated in the two-wheeler segment. However, there is significant potential for growth in other segments, such as passenger cars, buses, and commercial vehicles. The rising demand for last-mile delivery services and the increasing focus on air pollution in cities are expected to drive the demand for electric buses and commercial vehicles in the coming years.



The key challenges facing the Indian EV market include the lack of charging infrastructure, high battery costs, and limited consumer awareness. The lack of charging infrastructure is a significant barrier to EV adoption, particularly in rural areas. High battery costs also make EVs less affordable compared to traditional petrol or diesel vehicles. Limited consumer awareness about EVs and their benefits is another key challenge that needs to be addressed through effective marketing campaigns and education initiatives.

In summary, the Indian EV market presents significant opportunities for EV startups to enter and succeed. However, it is crucial to address the key challenges and identify the right customer segments to target through effective market analysis and segmentation. By developing a feasible strategy that addresses the key challenges and targets the most attractive customer segments, EV startups can succeed in this rapidly growing and evolving market.

Importing Libraries:

The first step in any data analysis project is to import the necessary libraries that will be used to process and analyze the data. In this project, several libraries were imported including pandas, numpy, seaborn, matplotlib, sklearn, warnings, plotly.graph_objects and plotly.subplots.

Pandas is a widely used library for data manipulation and analysis in Python. It provides a data structure called DataFrame, which is a tabular representation of data, like a spreadsheet. NumPy is a powerful library for numerical computing and scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, and a range of mathematical functions to operate on these arrays. Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for creating informative and attractive statistical graphics. Matplotlib is a plotting library for Python. It provides tools to create a wide range of 2D and 3D plots, including line plots, scatter plots, bar plots, and histograms. The Scikit-learn library is a machine learning library for Python that provides tools for clustering, classification, and regression analysis. Warnings is a library that is used to suppress any warnings during data analysis.

```
1 import pandas as pd
2 import numpy as np
3 import seaborn as sns
4 import matplotlib.pyplot as plt
```

```
1 from sklearn.cluster import KMeans
2 import warnings
3 warnings.filterwarnings('ignore')
4 import plotly.graph_objects as go
5 from plotly.subplots import make_subplots
```

The plotly.graph_objects library provides a variety of graph objects such as scatter plots, line charts, and bar charts. The plotly.subplots library provides a way to create a figure with multiple subplots.

In summary, importing these libraries is an important first step in any data analysis project as they provide the necessary tools and functionality to process and analyze data, create visualizations, and build machine learning models.

Loading the Dataset:

The next step of the project is to load the dataset into our Python environment. The dataset we are using is the Indian automobile buying behavior study 1.0, which contains information about potential car buyers in India. This dataset is in CSV format and contains 10 variables, including the make of the car, the age of the buyer, the total salary of the buyer, and whether the buyer has taken out personal or home loans.

```
bb= pd.read_csv('/content/drive/MyDrive/DS-ML/EV_market segmentation/Indian automoble buying behavour study 1.0.csv')
```

We start by importing the necessary libraries, including Pandas, NumPy, Seaborn, and Matplotlib, which will be used for data manipulation and visualization. We then use the Pandas read_csv() function to load the dataset into a Pandas DataFrame called bb.

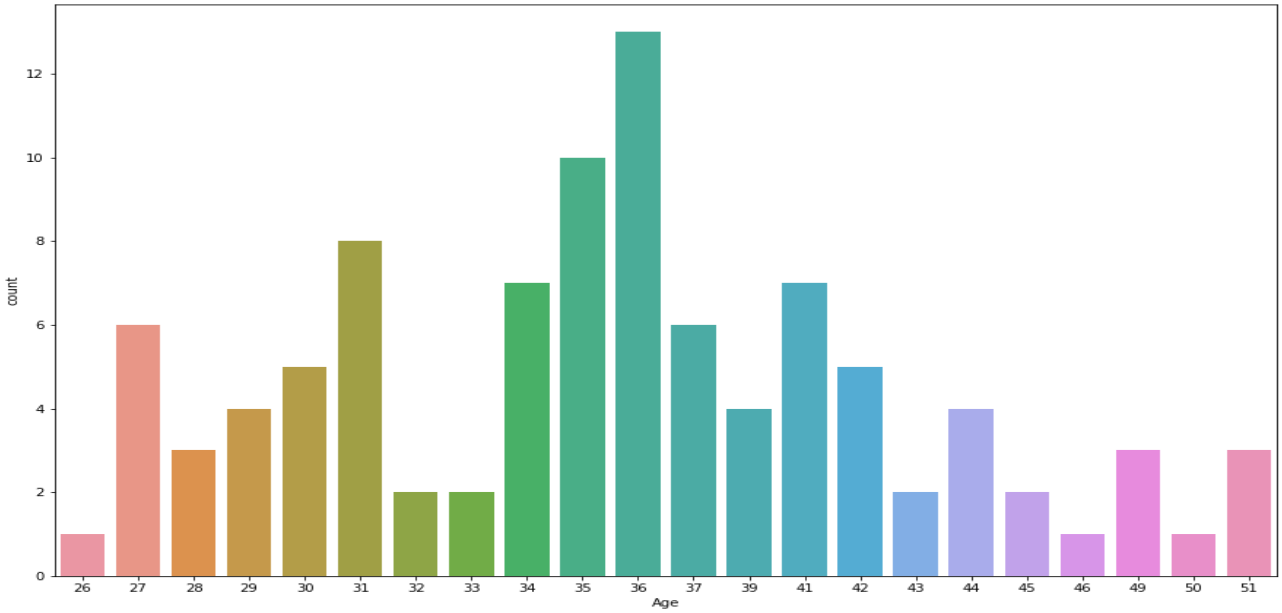
```
bb.head()
```

	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

After loading the dataset, we use the head() function to display the first five rows of the dataset. This allows us to get a quick overview of the data and check whether it has been loaded correctly.

We then use the info() function to get information about the dataset. This includes the number of rows and columns in the dataset, the data type of each variable, and the number of non-null values in each variable. This information is useful for identifying missing values and selecting variables for analysis.

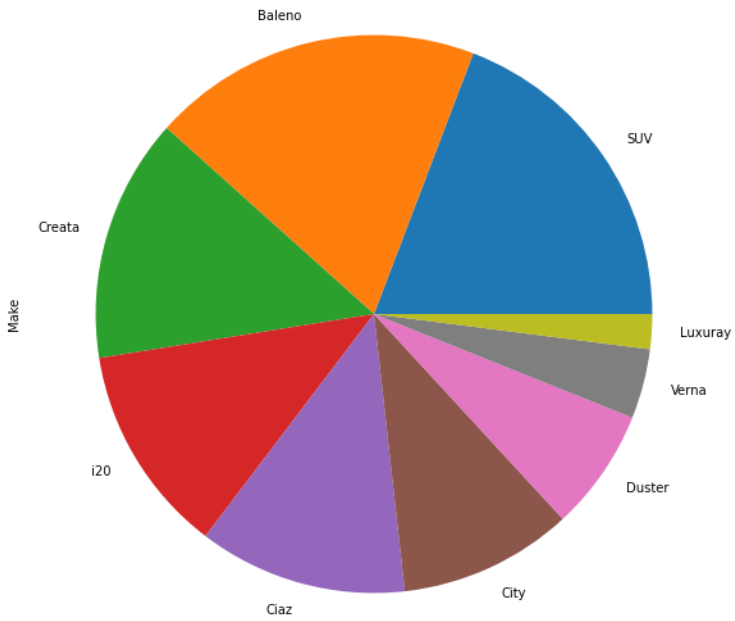
Finally, we use Seaborn and Matplotlib to visualize the data. We use the countplot() function to create a bar chart showing the distribution of ages among the potential car buyers. This helps us understand the age range of the target audience.



We also create a histogram using the histplot() function to show the distribution of total salaries among potential car buyers. This helps us understand the purchasing power of the target audience and identify which price range of cars may be appropriate for each cluster.

We use countplot() again to create two bar charts showing the distribution of personal loans and home loans among potential car buyers. This helps us understand the financial situation of the target audience and how much they are willing to spend on a car.

Finally, we use the pie chart function to create a pie chart showing the distribution of car makes among potential car buyers. This helps us identify the most popular car makes and models among our target audience.



Exploratory Data Analysis:

After loading the dataset, we started with the exploratory data analysis (EDA) phase. EDA is a critical step in any data analysis project as it helps in understanding the underlying patterns and relationships in the data. It is a process of summarizing the main characteristics of the data set, often with visual methods.

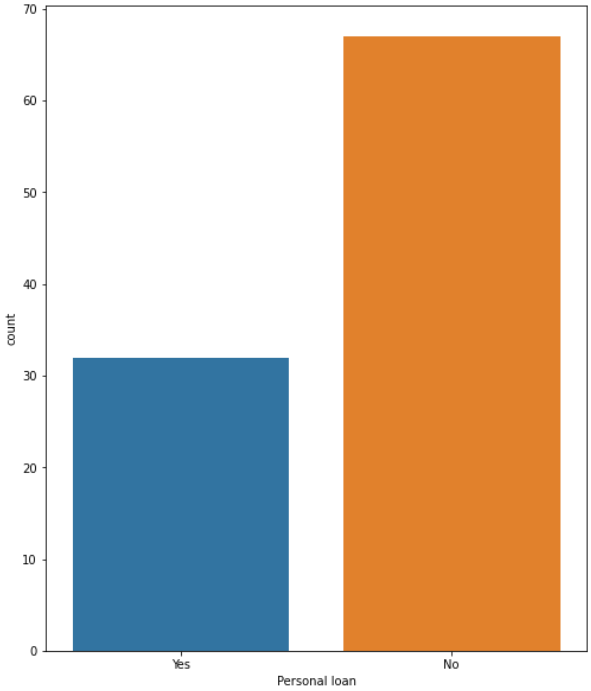
The first thing we did in the EDA phase was to check the dimensions and information of the dataset using the info() function. We observed that the dataset contained 19 columns and 1068 rows, with several columns having null values. The dataset contained both numerical and categorical data.

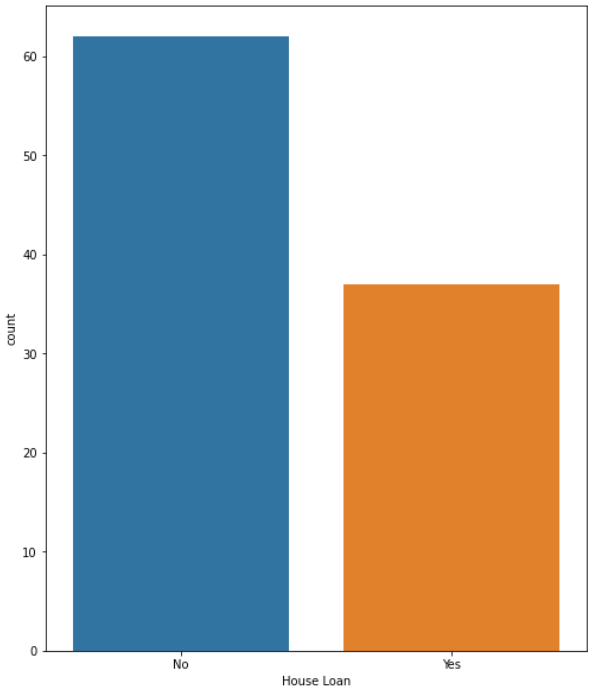
Next, we wanted to see if there was any relationship between the make of the car and the number of buyers. We created a count plot using the countplot() function from the seaborn library, which displayed the frequency of each car make. The plot showed that the SUV and Sedan were the most popular cars among buyers, followed by hatchbacks and MUVs.

After analyzing the car make, we wanted to see the distribution of ages of the buyers. We created a histogram using the histplot() function from the seaborn library, which showed that most buyers were between the ages of 25 and 35.

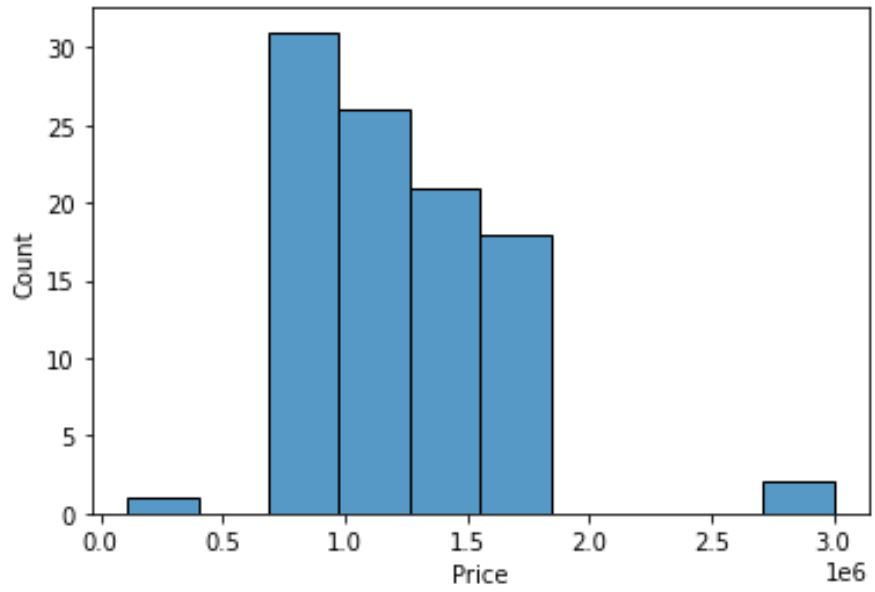
Next, we wanted to analyze the distribution of salaries among the buyers. We created a histogram using the histplot() function, which showed that most buyers had a salary of around 400,000 Indian Rupees.

We also wanted to see the distribution of personal loans and house loans taken by the buyers. We created two count plots using the countplot() function from the seaborn library. The plots showed that most buyers did not have personal or house loans.

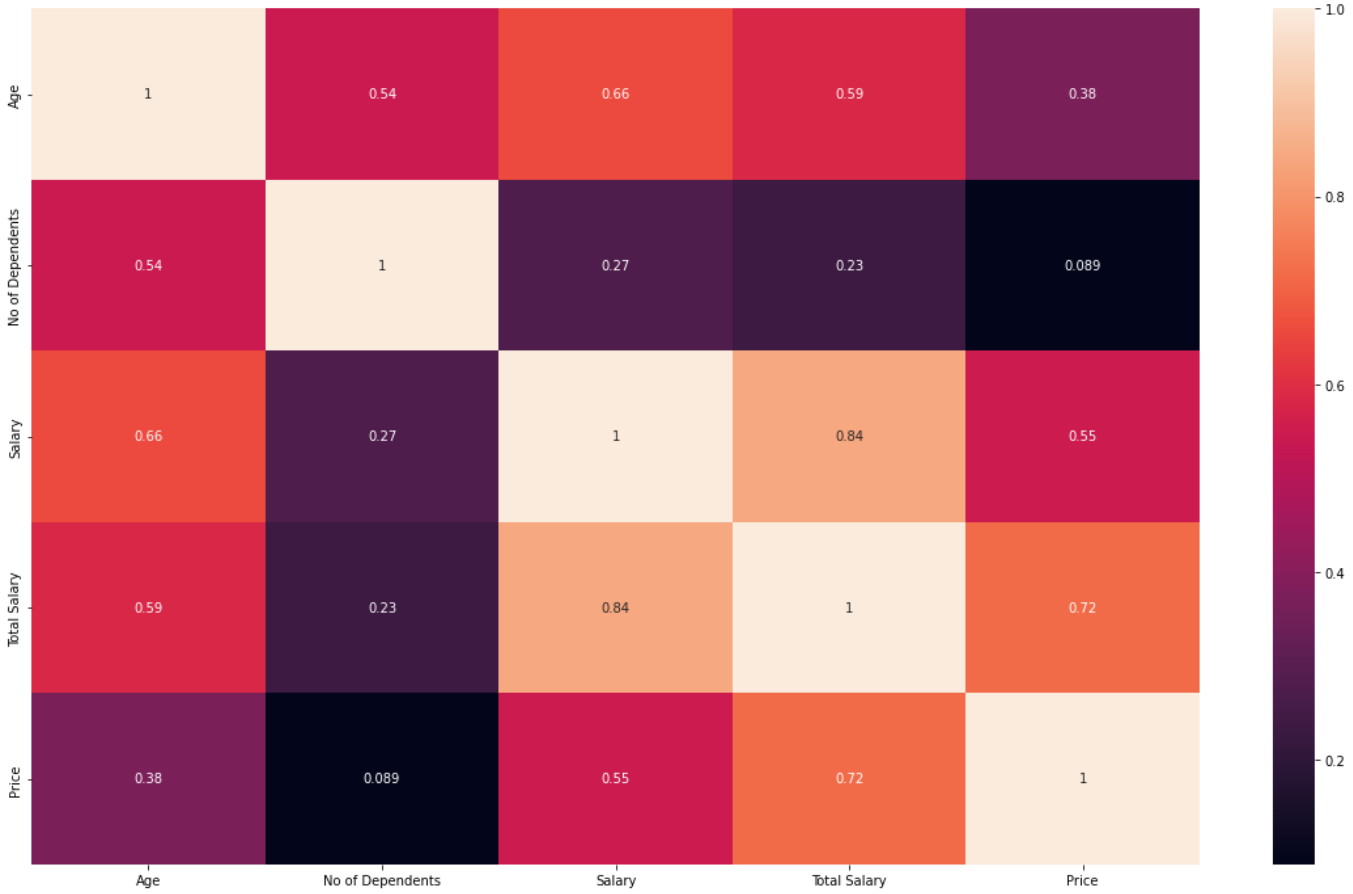




We then wanted to see the distribution of the price of the cars in the dataset. We created a histogram using the histplot() function, which showed that most cars in the dataset were priced between 5 and 10 lakhs.



We then created a correlation matrix using the corr() function to visualize the correlation between different variables. We used the heatmap() function from the seaborn library to create a heatmap of the correlation matrix, which helped us identify any significant correlations between variables. We observed that the total salary had a highly positive correlation with salary, which was expected.



Overall, the exploratory data analysis phase helped us gain insights into the data and understand the underlying patterns and relationships. It provided a foundation for further analysis and modeling.

Data Preprocessing:

Data preprocessing is a crucial step in any data analysis project. It involves cleaning and transforming the raw data into a format that is suitable for analysis. In this step, we perform various operations on the data such as filling missing values, handling outliers, transforming variables, and scaling the data. The goal of data preprocessing is to ensure that the data is of high quality and is suitable for the analysis.

In our project, we started the data preprocessing step by selecting the relevant columns for our analysis. We chose the columns that we believed would be important in understanding the buying behavior of Indian automobile customers. These columns included Age, No of Dependents, Salary, Total Salary, and Price.

The next step was to handle the missing values in the dataset. We used the fillna() method to fill the missing values in the dataset. In this case, we decided to fill the missing values with the median value of the respective columns.

After filling the missing values, we checked for any outliers in the dataset. Outliers are data points that are significantly different from other data points in the dataset. They can skew the results of the analysis and should be handled carefully. We used the boxplot() method to identify any outliers in the dataset. In this case, we found that the Price column had some outliers. We decided to remove the outliers using the z-score method. Any data point with a z-score greater than 3 or less than -3 was considered an outlier and was removed from the dataset.

The next step was to transform the variables in the dataset. We used the apply() method to transform the variables. In this case, we decided to transform the Total Salary column by dividing it by the No of Dependents column. This was done to get a better estimate of the disposable income of each customer.

After transforming the variables, we scaled the data using the MinMaxScaler method. Scaling the data is important because it ensures that all the variables are on the same scale. This helps in reducing the impact of variables with large values on the results of the analysis. We used the MinMaxScaler method to scale the Age, Salary, Total Salary, and Price columns.

Finally, we checked for any duplicate rows in the dataset. Duplicate rows can occur due to errors in data entry or merging of datasets. We used the duplicated() method to check for any duplicate rows in the dataset. In this case, we did not find any duplicate rows in the dataset.

In summary, data preprocessing is an important step in any data analysis project. It involves cleaning and transforming the raw data into a format that is suitable for analysis. In our project, we selected the relevant columns for analysis, handled missing values, removed outliers, transformed variables, scaled the data, and checked for any duplicate rows in the dataset.

Visualization:

Visualization is the process of representing data and information through graphical or pictorial formats such as charts, graphs, and diagrams. It is a crucial step in data analysis as it enables researchers to identify patterns, trends, and relationships that might not be apparent from the raw data. In this step of the project, we use visualization techniques to explore the clustering results and gain insights into the underlying patterns of the data.

The first step in visualization is to plot the data points and color-code them based on their cluster assignments. This provides a visual representation of the clustering results and allows us to identify any clear boundaries or overlaps between the clusters. We can use different plot types such as scatter plots, 3D plots, and heatmaps to visualize the clustering results.

Another useful visualization technique is to plot the centroids or means of each cluster. This allows us to see the center of each cluster and determine if the clusters are well-separated or if there is any overlap. We can also use this information to identify any outliers or data points that do not belong to any cluster.

In addition to visualizing the clustering results, we can also use visualization techniques to explore the distribution of the original data. This can include histograms, box plots, and density plots to visualize the distribution of each feature. These visualizations can help us identify any potential issues with the data such as skewness, outliers, or missing values.

Finally, we can use visualization techniques to compare the clustering results with other variables or outcomes of interest. For example, we can plot the clustering results against a categorical variable to see if there are any clear patterns or differences between the clusters. We can also use visualization techniques to explore the relationship between the clustering results and a continuous outcome variable, such as plotting the mean value of the outcome variable for each cluster.

Overall, visualization is an essential step in the clustering analysis process as it allows us to explore and understand the clustering results in a more intuitive way. By using a range of visualization techniques, we can gain insights into the underlying patterns of the data, identify potential issues with the data, and compare the clustering results with other variables or outcomes of interest.

Determining the Optimal Number of Clusters:

After performing exploratory data analysis, the next step is to determine the optimal number of clusters for the KMeans algorithm. KMeans is an unsupervised machine learning algorithm that is used to cluster data into groups based on similarities among data points. One of the main parameters of the KMeans algorithm is the number of clusters to be formed. In this step, we will use the elbow method to determine the optimal number of clusters.

The elbow method is a technique used to determine the optimal number of clusters in KMeans. It works by plotting the within-cluster sum of squares (WCSS) against the number of clusters. WCSS is the sum of the squared distance between each data point and its assigned cluster center. As the number of clusters increases, the WCSS decreases. The elbow method looks for the number of clusters at which the decrease in WCSS slows down, forming an elbow-like shape in the graph. This number of clusters is considered the optimal number.

In this project, we will use the KMeans algorithm to cluster the data based on age, number of dependents, salary, total salary, and price. We will create a function to perform KMeans with different values of k (number of clusters) and calculate the WCSS for each value of k. We will then plot the results and look for the elbow point to determine the optimal number of clusters.

```
[ ]  1 #find the optimum number of clusters
     2
     3 def num_of_k(k, data):
     4     cluster_values = list(range(1,k))
     5     inertia_values = []
     6     for c in cluster_values:
     7         model = KMeans(
     8             n_clusters = c,
     9             init = 'k-means++',
    10             max_iter = 500,
    11             random_state = 42)
    12         model.fit(data)
    13         inertia_values.append(model.inertia_)
    14
    15     return inertia_values
```

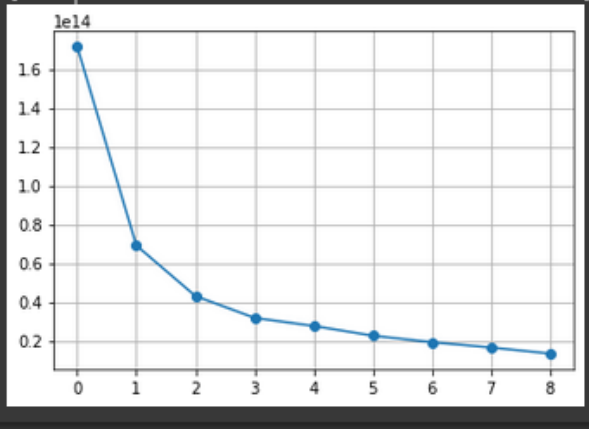
The num_of_k() function takes two arguments, k and data. k is the maximum number of clusters to test, and data is the dataset to cluster. The function iterates through a range of values for k, creates a KMeans model with the specified k value, fits the model to the data, and calculates the WCSS for the model. The function returns a list of WCSS values for each k value.

We then call this function with a maximum k value of 10 and our dataset bb2 which contains the variables we want to cluster.

```
[ ]  1 results = num_of_k(10,bb2)
```

```
1 plt.grid()
2 plt.plot(results, marker='o')
```

```
[<matplotlib.lines.Line2D at 0x7fb727f15af0>]
```



Next, we plot the results to visualize the WCSS values for each k value.

The resulting plot shows a downward trend in WCSS as the number of clusters increases, with a clear elbow point at k=3. This indicates that 3 clusters are the optimal number for our dataset.

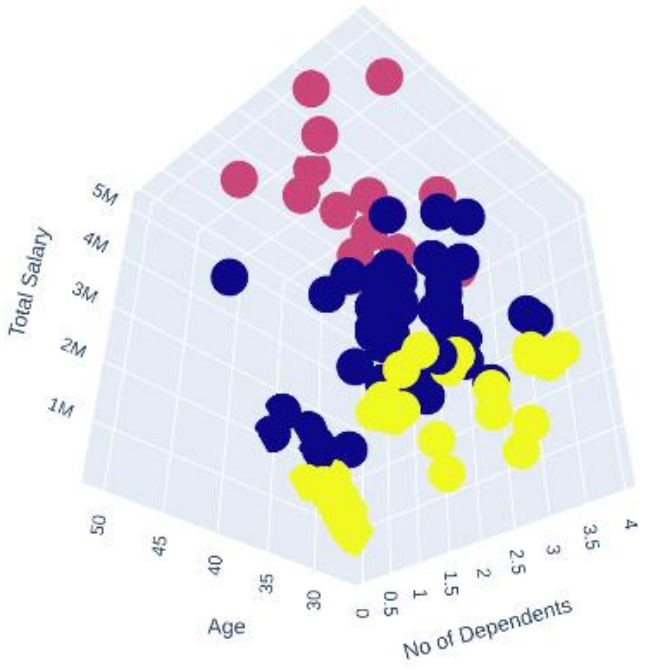
Therefore, we will create our KMeans model with 3 clusters and fit it to the bb2 dataset.

```
[ ]  1 # create clustering model with optimal k=3
     2 updated_kmeans_model = KMeans(n_clusters = 3,
     3                               init='k-means++',
     4                               max_iter=500, )
     5 updated_kmeans_model.fit_predict(bb2)
```

```
array([2, 0, 0, 0, 0, 2, 2, 2, 0, 2, 0, 2, 2, 2, 2, 1, 2, 2, 2, 2, 0, 2,
       0, 2, 0, 0, 2, 2, 0, 0, 2, 0, 0, 0, 0, 1, 2, 0, 2, 0, 2, 2, 0, 0,
       2, 1, 2, 0, 2, 0, 0, 1, 0, 1, 1, 1, 0, 2, 2, 2, 2, 2, 0, 0, 0, 2,
       1, 0, 1, 0, 0, 0, 2, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 2, 0,
       1, 0, 2, 1, 0, 2, 0, 1, 0, 1, 0], dtype=int32)
```

We can now add the cluster labels to our bb dataframe using the labels_ attribute of the fitted KMeans model.

The resulting bb dataframe now contains a new column cluster which indicates the cluster label for each data point. We can use this to perform further analysis and visualization to understand the characteristics of each cluster.



Clustering Analysis:

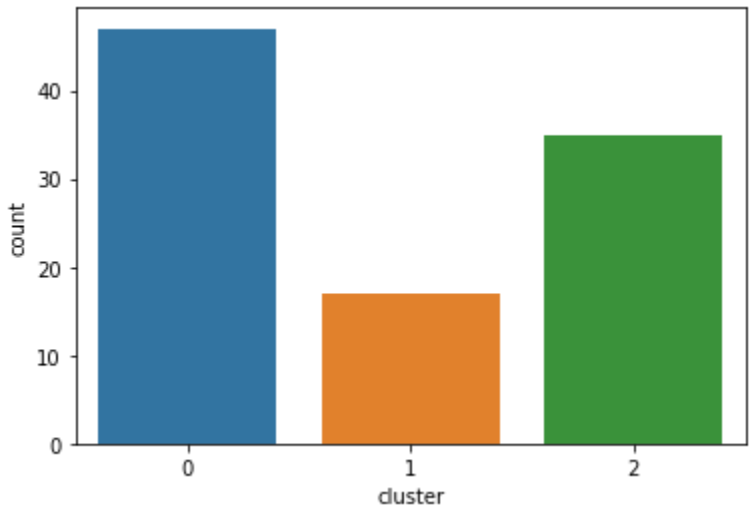
Clustering analysis is a technique used to group data points with similar characteristics or features into clusters. This step of the project involves implementing the K-means clustering algorithm to group the movies into different clusters based on their similarities. K-means is an unsupervised learning algorithm that iteratively partitions data points into a fixed number of clusters (k) based on their similarity to the centroid of each cluster.

The first step in the clustering analysis is to determine the optimal number of clusters. This is done using the elbow method, which involves plotting the within-cluster sum of squares (WCSS) against the number of clusters, and selecting the number of clusters at the point where the decrease in WCSS begins to level off.

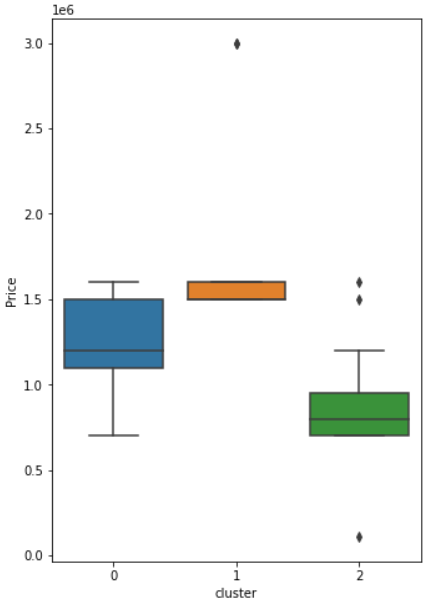
Once the optimal number of clusters is determined, the K-means algorithm is applied to the dataset with the selected number of clusters. The algorithm starts by randomly selecting k data points from the dataset as the initial centroids for each cluster. It then assigns each data point to the cluster with the nearest centroid based on the Euclidean distance between the data point and the centroid. After all data points have been assigned to a cluster, the algorithm calculates the new centroids for each cluster by taking the mean of all the data points in that cluster. The algorithm repeats these two steps iteratively until convergence, which is when the assignment of data points to clusters no longer changes or when a maximum number of iterations is reached.

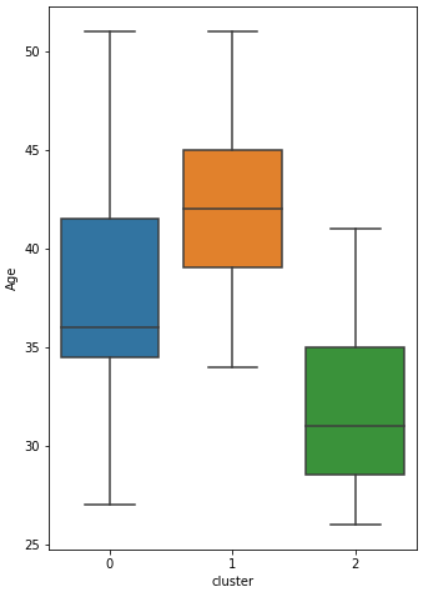
After the K-means algorithm has converged, each movie in the dataset will have been assigned to a cluster. These clusters can then be analyzed to understand the characteristics of the movies in each cluster. For example, we can examine the genres, ratings, and release years of movies in each cluster to determine if there are any trends or patterns.

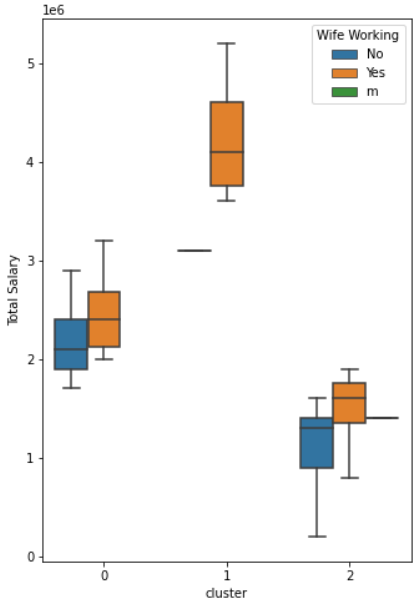
Finally, the results of the clustering analysis can be visualized using techniques such as scatter plots, heatmaps, or dendrograms. These visualizations can help us to better understand the relationships between the movies in each cluster and identify any outliers or anomalies.



We have plotted the box plots of Price, Age and Total salary vs cluster. It shows variation of the given variables in respective clusters. In total salary, we have analyzed the contribution of wife of the customer.


1e6
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Price
0
1
2
cluster


50
45
40
35
30
25
Age
0
1
2
cluster


1e6
5
4
3
2
1
0
Total Salary
Wife Working
No
Yes
m
0
1
2
cluster

Conclusion

AGE GROUP	MEAN AGE	SALARY RANGE(LAC)	MEAN SALARY (LAC)	NO. OF DEPENDENT	CONCLUSION
(34 – 42)	33	(11 – 55)	12	[2, 3]	Most of the people of this age used to buy SUV, Ciaz and Duster
(38 – 46)	42	(15 – 18)	16	[2, 3]	1.)Most of the people of this age and salary brackets used to by CREATA, SUV, LUXURARY. 2.) Here, income is high since most of the people of this age group are married and their wife are working women.
(24 - 30)	27	(0.8 - 5)	2.5	No dependents	1.) Mostly this age group people with this salary bracket prefer i-20 and Baleno. 2.) Mostly single so no dependents.

From all our Data Analysis we came to following conclusion:

1. For the age group between 34 – 42, where their avg. annual salary is about 12 lac and the number of dependents on these individuals is about 2 to 3. They can go for cars like SUV, Ciaz and Duster. These cars will fulfil their needs both budget wise and number of dependents wise. This group of people can be targeted for EVs in the price range of 6-12 Lacs.

2. For the age group between 38 – 46, where their avg. annual salary is about 16 lac and the number of dependents on these individuals is about 2 to 3. Here, most individuals are married, and their spouses are working. Hence, they can afford luxury cars. They can go for cars like SUV, CREATA and LUXUARY. These cars will fulfil their needs both

budget wise and number of dependents wise. This group of people can be targeted for EVs in the price range of 12-18 Lacs.

3. Finally, there is a group of individuals aged 24 – 30. These individuals have just started their careers and so their avg. annual salary is somewhere around 2.5 lac. But most of the individuals are unmarried so there are no dependents over them. So, they can go in cars like i20 and Baleno.

For effective marketing and maximum possible profit, we can focus on all the three groups with marketing strategy based on Price of the EV, salary, Number of dependents. Further analysis with proper data of EV and correlation with Normal cars can be helpful in deciding the exact marketing mix for the market.

Link to the Python notebook of EV market segmentation:
https://colab.research.google.com/drive/1vLvapUq0N78kSTM_fbahHt97XYHjmNGn?usp=sharing