Online Vehicle Booking Market

Raghav Soni

- **Problem Statement:**

  The given task is to perform the segment analysis of vehicle booking market in India and create a feasible strategy to enter into the Indian market.

- **Data Sources:**

  In this analysis, the dataset used is of an online taxi booking service company named OLA.

  Reference Link: https://www.kaggle.com/datasets/nimish23/ola-trips?resource=download

- **Data Preprocessing:**

  - **Importing the Libraries**: Here the libraries used are NumPy, pandas, matplotlib and seaborn. NumPy is used for array operations. Pandas is used to load and manipulate the dataset. Matplotlib and Seaborn is used for data visualization.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```
✓ 0.0s

  - **Loading the dataset:**

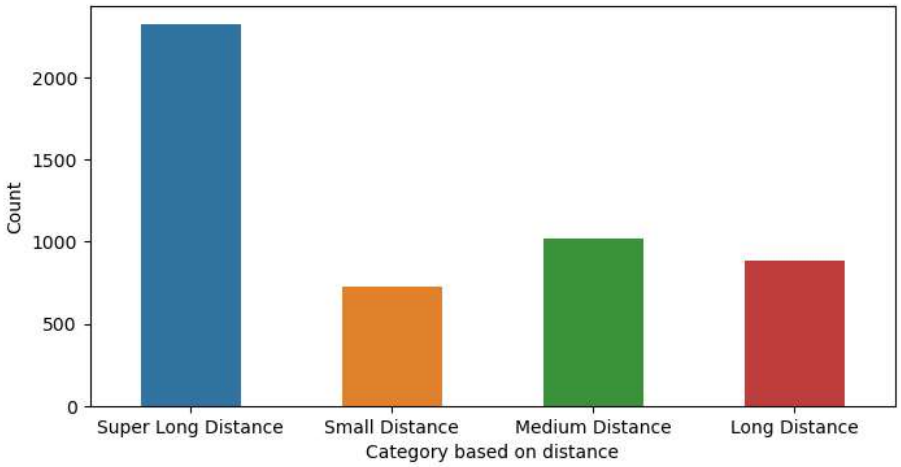
```
df = pd.read_csv('OLA_Dataset.csv')
df.head(3)
```
✓ 0.3s

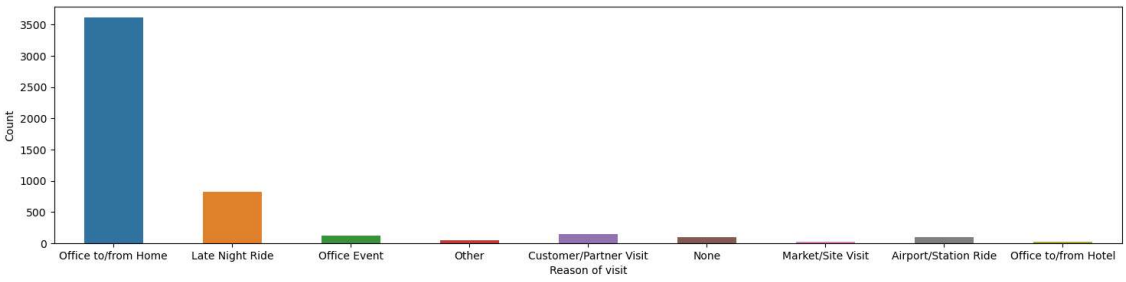
	booking id	booking_date_time	gender	month	day_of_week	time_of_day	\
0	1890061540	43249.91944	Male	May	Tue	0.919444	
1	1542148932	43153.92500	Female	February	Thu	0.925000	
2	1672692603	43194.88264	Female	April	Wed	0.882639	

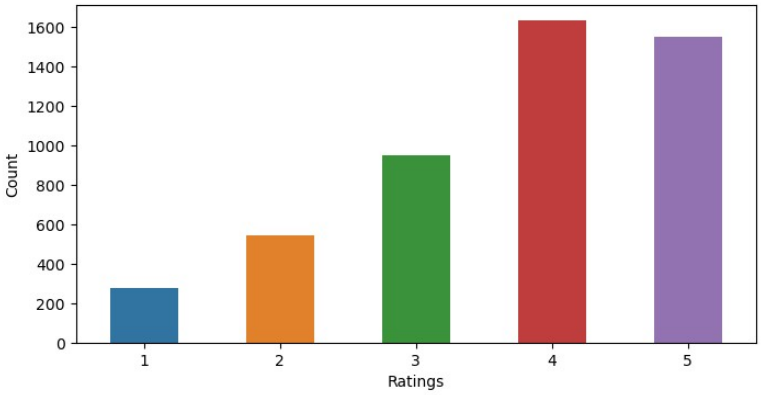
	distance_travelled	time_taken	reason	toll	category	\
0	17	58.0	Office to/from Home	0	Mini	
1	18	43.0	Late Night Ride	0	Mini	
2	2	5.0	Office to/from Home	0	Prime	

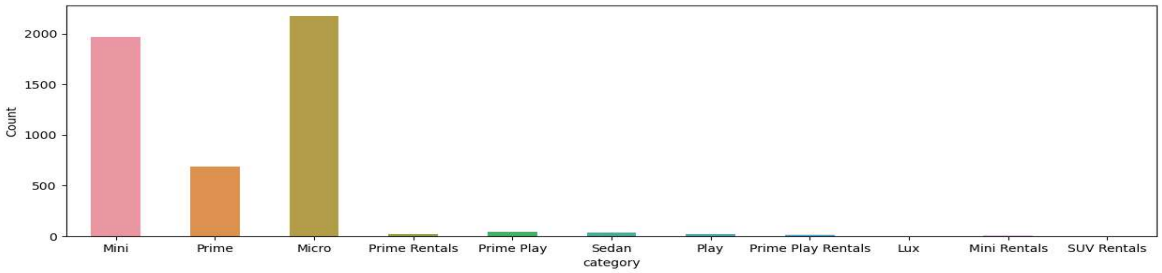
	commission_base_cost	driver_base_cost	total_tax	total_trip_cost	ratings
0	57.73	230.91	21.94	311.00	3
1	52.04	208.16	19.76	279.96	5
2	19.70	78.81	7.49	106.00	5

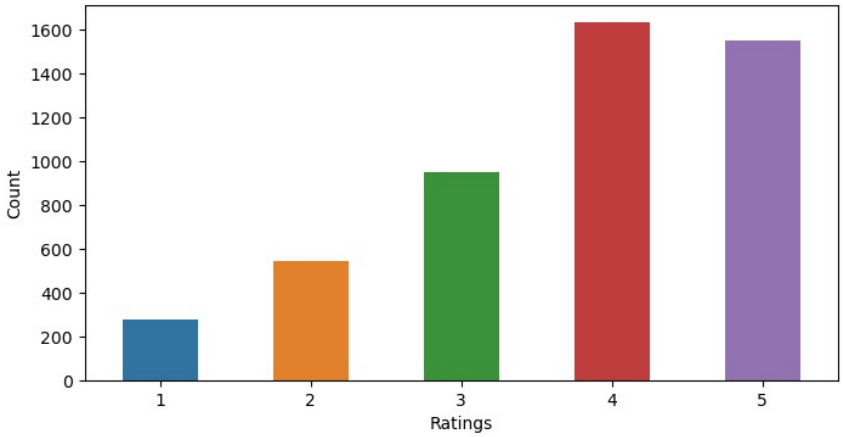
- **Handling Null values.**
- **Performing EDA and data visualization**
  - **Univariate analysis**

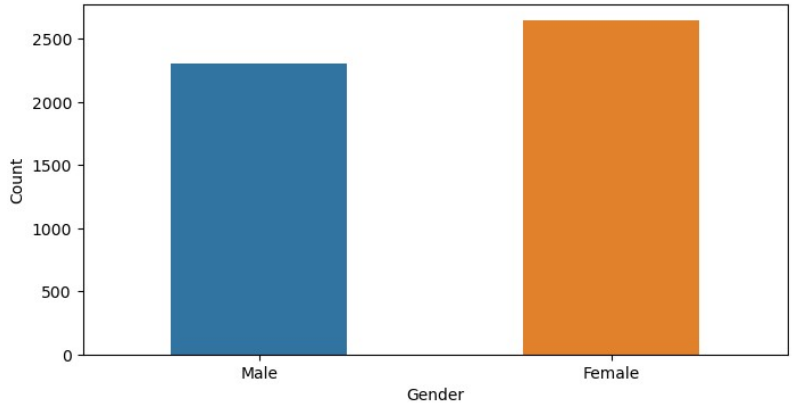


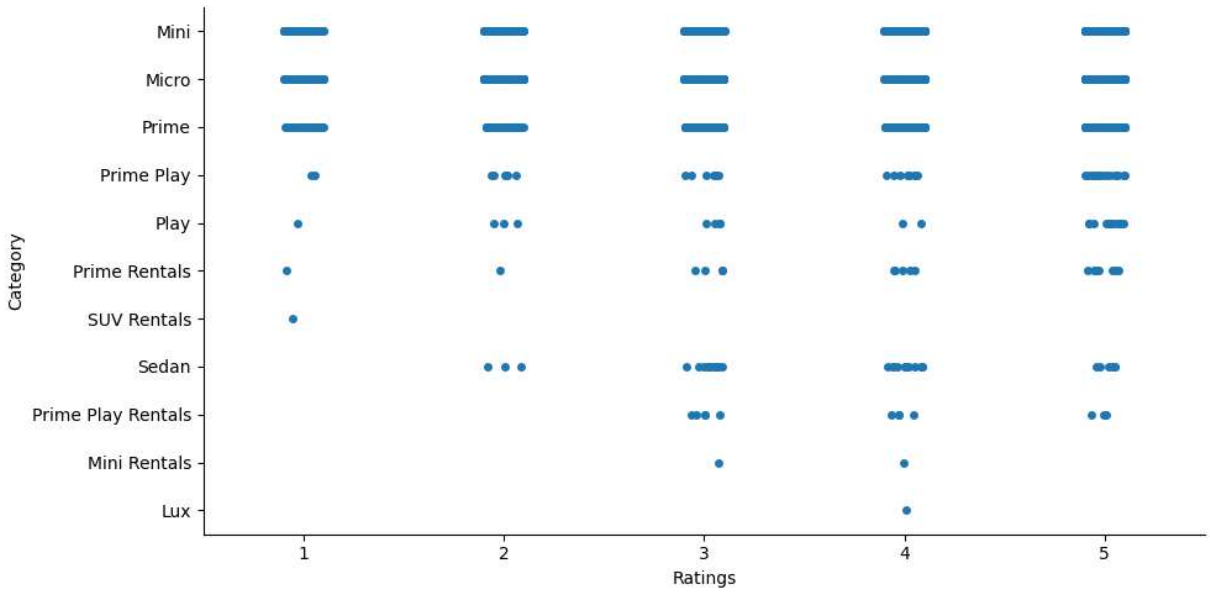


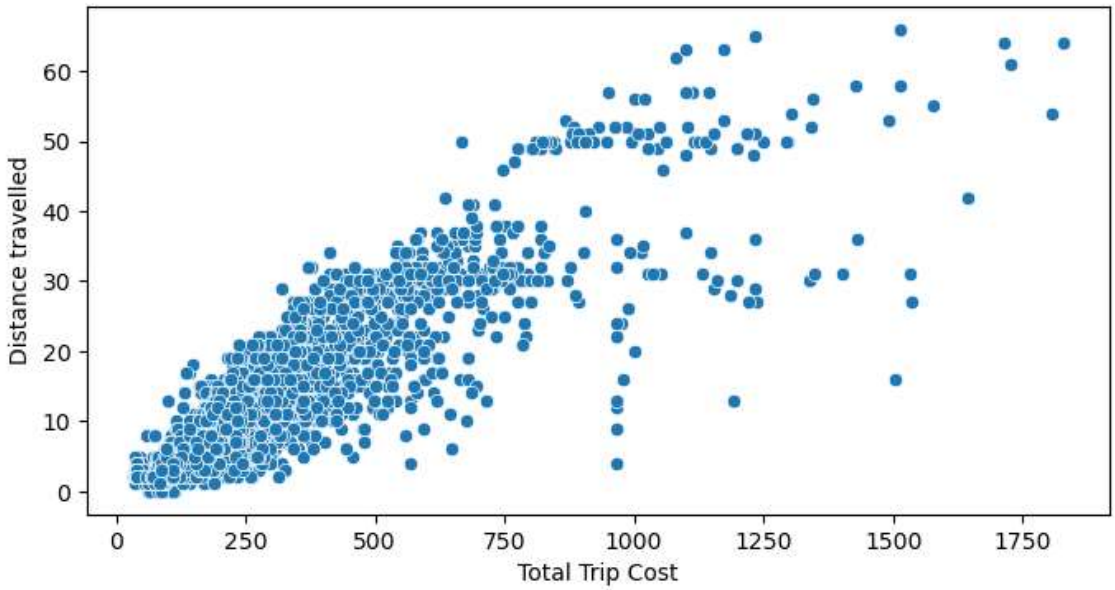


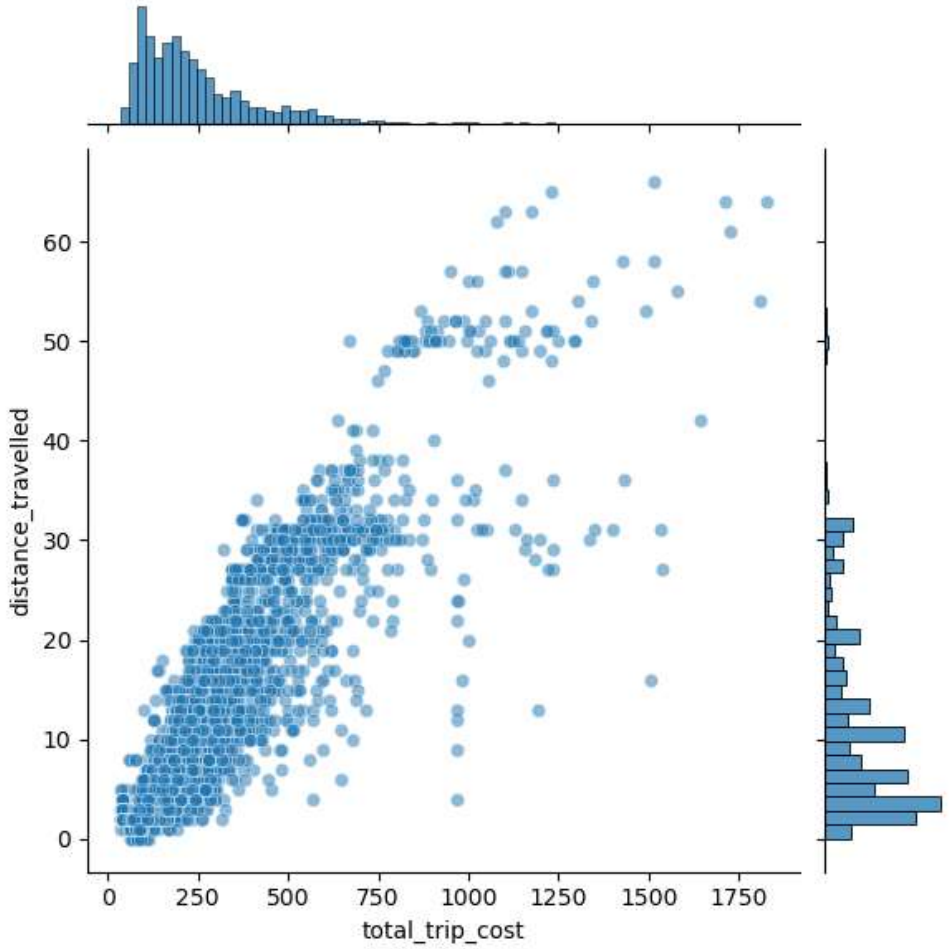



1600
1400
1200
1000
800
600
400
200
0
Count
1
2
3
4
5
Ratings

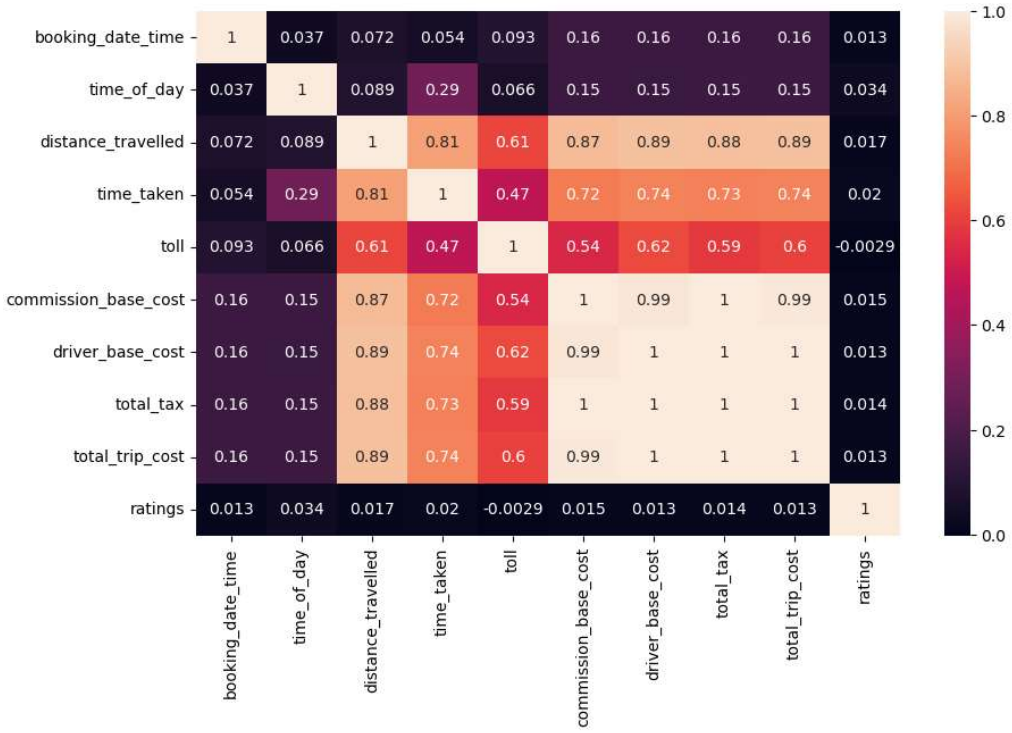

2500
2000
1500
1000
500
0
Count
Male
Female
Gender


Mini
Micro
Prime
Prime Play
Play
Prime Rentals
SUV Rentals
Sedan
Prime Play Rentals
Mini Rentals
Lux
Category
1
2
3
4
5
Ratings


Distance travelled
Total Trip Cost
0
10
20
30
40
50
60
0
250
500
750
1000
1250
1500
1750


distance_travelled
total_trip_cost
0
10
20
30
40
50
60
0
250
500
750
1000
1250
1500
1750

- **Multivariate analysis**



- **Label encoding:**
  Using label encoding for categorical data. Here the row named ‘category’ contains the category of vehicle used by the customer as a taxi is used for label encoding.

• Segment Extraction:

- **K Means clustering**

- **For Psychographic, Demographic, Behavioral analysis**

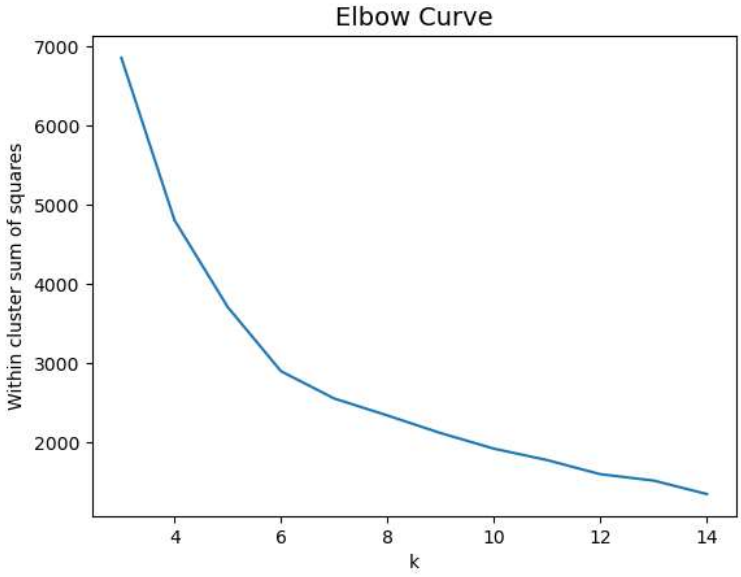
```
# KMeans Clustering

from sklearn.cluster import KMeans
# for determing K (Elbow curve)
k_vals = np.arange(3,15)
wss = []
for i in k_vals:
    kmodel = KMeans(n_clusters = i)
    kmodel.fit(scdata)
    wss.append([i,kmodel.inertia_])

print(wss)
```

✓ 2.5s

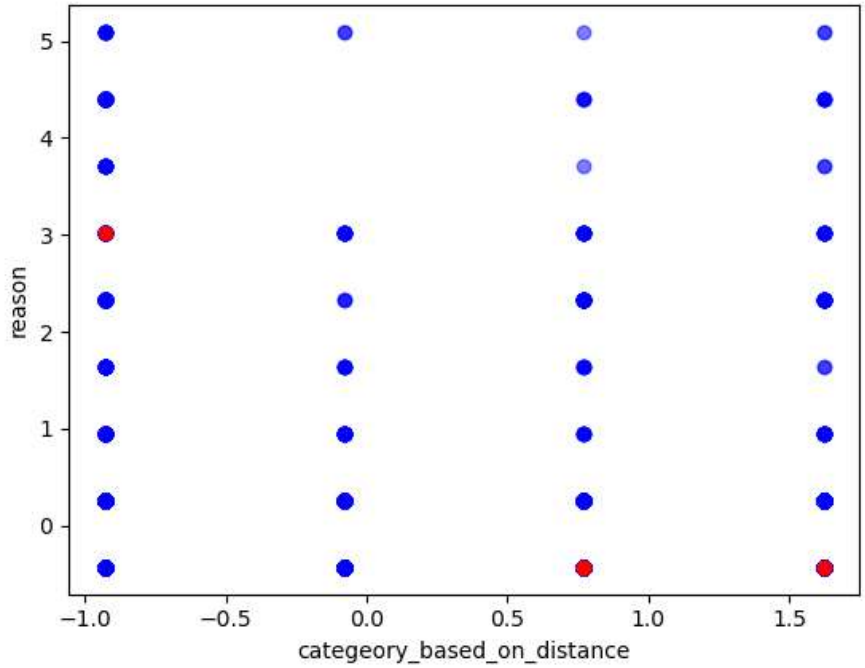
Output: [[3, 6852.686205029736], [4, 4800.282692885492], [5, 3703.387962981871], [6, 2893.156933482198], [7, 2549.0109042370336], [8, 2334.372907505163], [9, 2111.88907881666], [10, 1915.6622784270146], [11, 1770.9027529432246], [12, 1592.5462541823217], [13, 1512.2938345381808], [14, 1342.253060643583]]

```
wss = pd.DataFrame(wss, columns = ['k', 'WSS'])

sns.lineplot(x = 'k', y = 'WSS', data = wss)
plt.ylabel('Within cluster sum of squares')
plt.title('Elbow Curve', fontsize = 14,)
plt.show()
```

✓ 0.3s



```
K=4
Centroids = (scdata.sample(n=K))
plt.scatter(scdata["categeory_based_on_distance"],scdata["reason"],c='blue', alpha=0.5)
plt.scatter(Centroids["categeory_based_on_distance"],Centroids["reason"],c='red')
plt.xlabel('categeory_based_on_distance')
plt.ylabel('reason')
plt.show()
```
✓ 0.1s

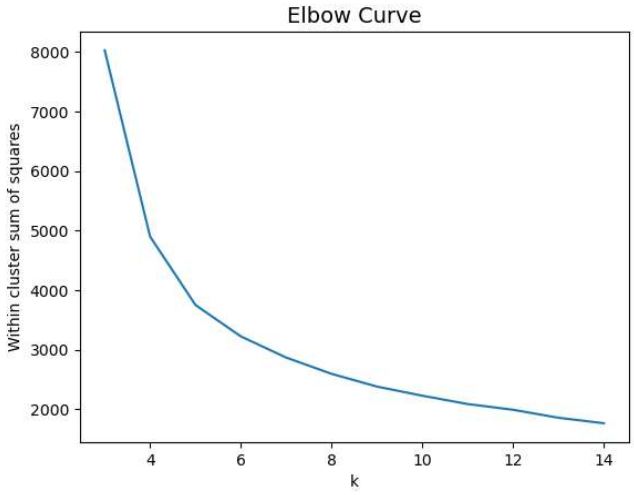


- **For Financial Analysis**

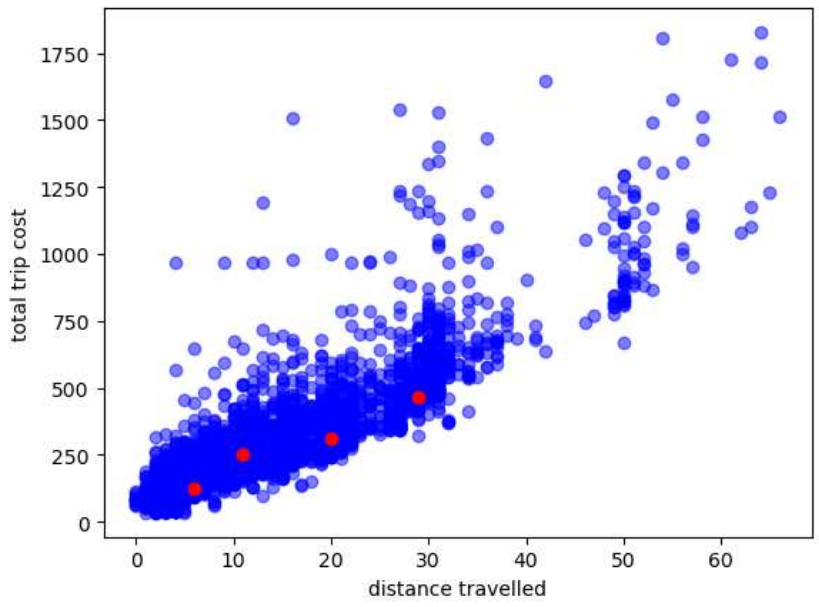
```
# K Means Clustering

# for determing K (Elbow curve)
k_vals = np.arange(3,15)
wss = []
for i in k_vals:
    kmodel = KMeans(n_clusters = i)
    kmodel.fit(datasc)
    wss.append([i,kmodel.inertia_])

print(wss)
```
✓ 3.6s

Output: [[3, 8027.108140242174], [4, 4901.163162706449], [5, 3752.9239010055385], [6, 3225.592332986855], [7, 2870.020817196908], [8, 2597.060173729244], [9, 2383.081214045975], [10, 2229.894690371897], [11, 2089.922937485169], [12, 1993.8105321579935], [13, 1859.884384124361], [14, 1767.8277511063811]]

```
wss = pd.DataFrame(wss, columns = ['k', 'WSS'])

sns.lineplot(x = 'k', y = 'WSS', data = wss)
plt.ylabel('Within cluster sum of squares')
plt.title('Elbow Curve', fontsize = 14)
plt.show()
```
✓ 0.2s




```
K=4
Centroids = (df.sample(n=K))
plt.scatter(df["distance_travelled"],df["total_trip_cost"],c='blue',alpha = 0.5)
plt.scatter(Centroids["distance_travelled"],Centroids["total_trip_cost"],c='red')
plt.xlabel('distance travelled')
plt.ylabel('total trip cost')
plt.show()
```
✓ 0.2s



- **Hierarchical Clustering**
  - **For Psychographic, Demographic, Behavioral analysis**

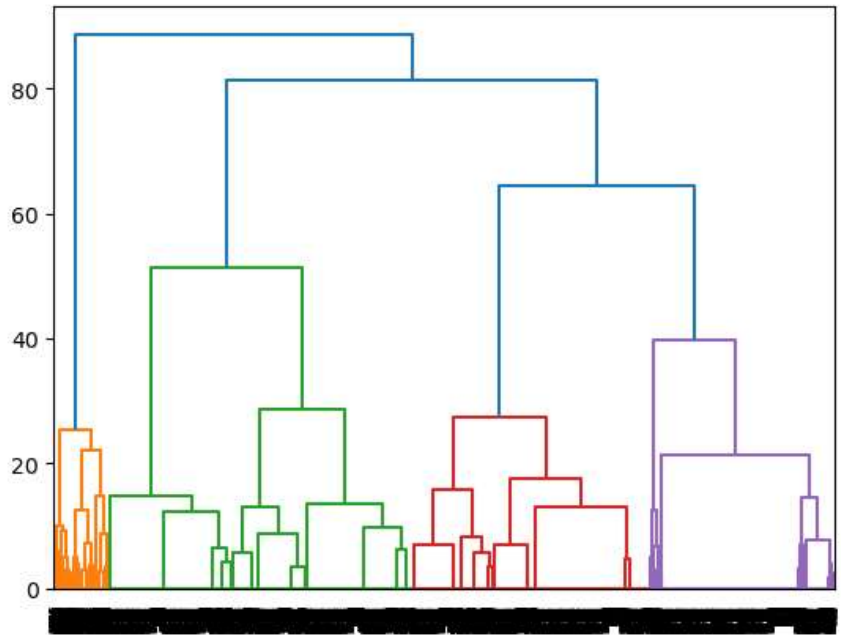
```
# Hierarchical Clustering
from sklearn.cluster import AgglomerativeClustering
agc = AgglomerativeClustering(n_clusters=4)
model = agc.fit(scdata)
model.labels_
```

✓ 0.9s

```
array([1, 1, 1, ..., 0, 1, 0], dtype=int64)
```

```
from scipy.cluster.hierarchy import dendrogram, linkage

linkage_data = linkage(scdata, method='ward', metric='euclidean')
dendrogram(linkage_data)

plt.show()
```

✓ 2m 14.2s



- **For Financial analysis**

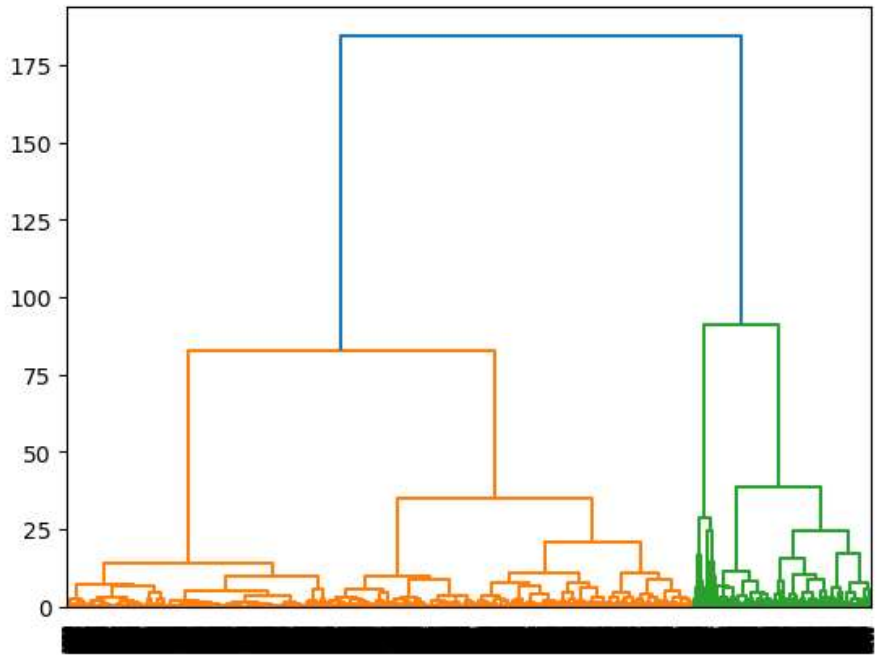
```
# Hierarchical Clustering
from sklearn.cluster import AgglomerativeClustering
agc = AgglomerativeClustering(n_clusters=4)
model = agc.fit(datasc)
model.labels_
```

✓ 0.9s

```
array([1, 1, 3, ..., 1, 1, 3], dtype=int64)
```

```
from scipy.cluster.hierarchy import dendrogram, linkage

linkage_data = linkage(datasc, method='ward', metric='euclidean')
dendrogram(linkage_data)

plt.show()
```

✓ 1m 51.4s



- **Profiling and describing potential segments**

    - The aim of the profiling step is to get to know the market segments resulting from the extraction step. At the profiling stage, we inspect a number of alternative market segmentation solutions. This is particularly important if no natural segments exist in the data, and either a reproducible or a constructive market segmentation approach has to be taken.
    - Data-driven segmentation solutions are usually presented to users (clients, managers) in one of two ways:
        - (1) as high-level summaries simplifying segment characteristics to a point where they are misleadingly trivial, or
        - (2) as large tables that provide, for each segment, exact percentages for each segmentation variable

- **Selection of target segment**

    - Market segmentation is a strategic marketing tool. The selection of one or more target segments is a long-term decision significantly affecting the future performance of an organization.
    - Based on common segments in the online vehicle booking market, here are a few examples:
        - Business Travelers: Professionals who frequently travel for work and prioritize reliable, convenient transportation solutions.
        - Tourists and Leisure Travelers: Individuals or families seeking flexible, hassle-free transportation options during holidays or sightseeing trips.
        - Airport Transfers: Passengers needing transportation to and from airports, often requiring punctuality and reliability.
        - Corporate Clients: Companies needing transportation services for employees, clients, or events, which may involve bulk bookings and customized service agreements.
        - Special Events and Occasions: Customers requiring transportation for weddings, parties, conferences, etc., and willing to pay for premium, customized services.

- **Customizing the Marketing Mix**

    - Market segmentation does not stand independently as a marketing strategy. Rather, it goes hand in hand with the other areas of strategic marketing, most importantly: positioning and competition.
    - To best ensure maximizing on the benefits of a market segmentation strategy, it is important to customize the marketing mix to the target segment. The selection of one or more specific target segments may require the design of new, or the modification or re-branding of existing products (Product),

changes to prices or discount structures (Price), the selection of suitable distribution channels (Place), and the development of new communication messages and promotion strategies that are attractive to the target segment (Promotion).

- **Potential customer base in the early market**

    - As per the above analysis and visualizations potential customer base can be acquired is of commuters and they prefer micro vehicle for commuting as it has low price as compared to the other services such as sedan, SUVs.

- **The Most optimal market segment**

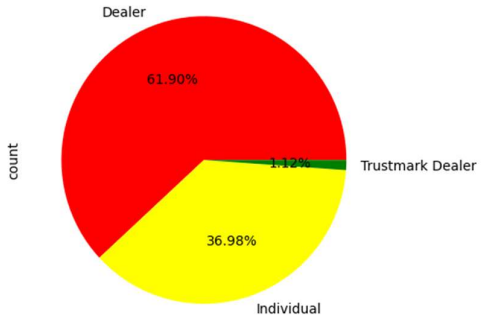
    - As per the above analysis and visualization the most optimal market segment is of commuters. As seen above in the graphs there are people who uses taxi service for their daily route is more compared to the casual movers who goes for outings at evening. There are a greater number of female customers as compared to male customers who uses taxi services. Customers prefer micro or mini car services as compared to sedans and SUVs.
    - Key Considerations for Each Segment.
    - Young Professionals in Urban Areas
        - Marketing Strategies: Leverage digital marketing, social media campaigns, and partnerships with co-working spaces and tech companies.
        - Service Features: Offer premium services, loyalty programs, and seamless app integration with other urban services (e.g., public transport, food delivery).
    - Students and Young Adults
        - Marketing Strategies: Campus promotions, influencer partnerships, and student discounts.
        - Service Features: Cost-effective options, group booking discounts, and integration with university events and activities.
    - Commuters in Suburban Areas
        - Marketing Strategies: Targeted online ads, local partnerships with businesses and schools, and community engagement.
        - Service Features: Reliable scheduling, subscription models, and family-friendly options (e.g., child seats, carpooling).
    - Tourists and Business Travelers
        - Marketing Strategies: Collaborations with hotels, airlines, and travel agencies; presence on travel websites and apps.
        - Service Features: Easy booking for airport transfers, multilingual support, and customizable travel packages.

- **Conclusion:** By focusing on these segments, the online vehicle booking service can effectively capture a diverse and profitable customer base, ensuring a strong market presence and growth potential. Each segment requires tailored marketing strategies and service features to meet the unique needs and preferences of its members.

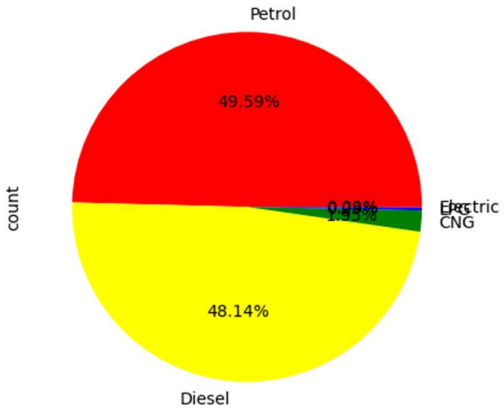
- **Link to GitHub repository**

  - https://github.com/SYRS-21/Vehicle-booking-Market-Analysis.git

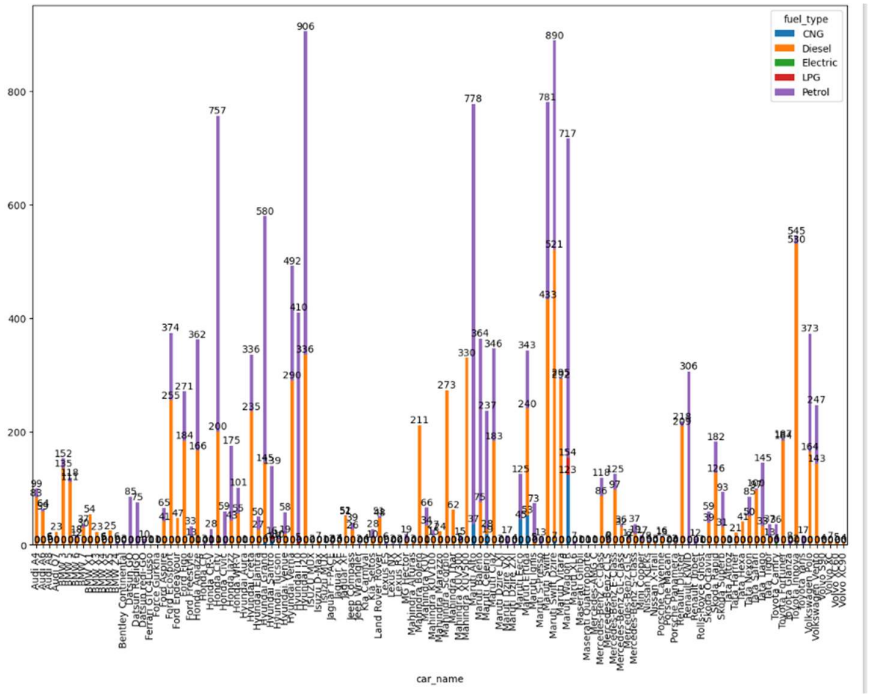
Car-Dekho data-set.



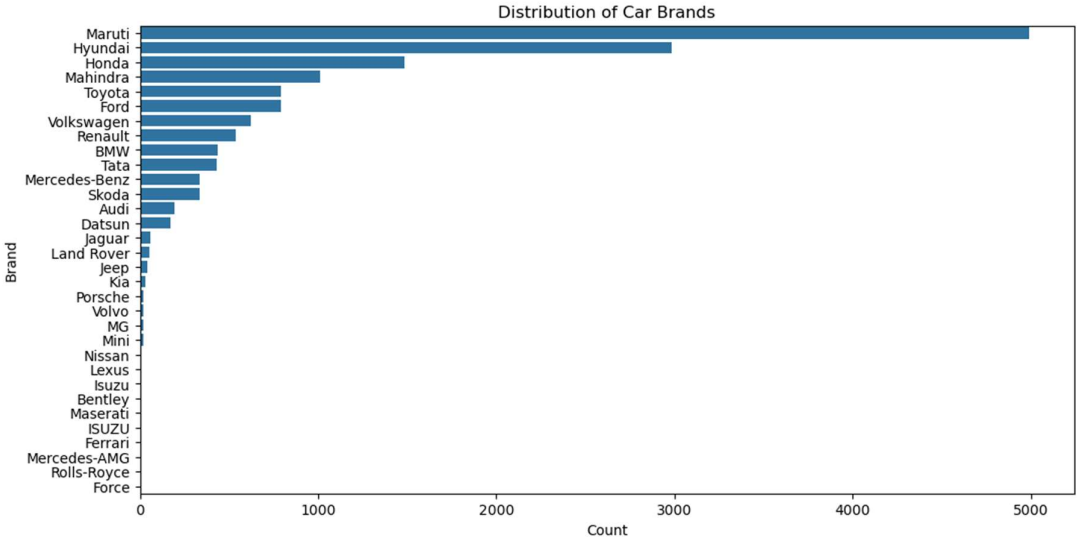
Based on the pie chart analysis of the Online Vehicle Booking Market, the data reveals that dealers dominate the market with a substantial share of 61.90%. This indicates that the majority of vehicle bookings are handled through dealerships, which suggests a strong preference or trust among consumers for professional dealer services. Individual sellers account for 36.98% of the market, reflecting a significant segment where buyers are engaging directly with car owners, possibly driven by the appeal of potentially lower prices or direct negotiations. Trustmark dealers, although representing a small fraction at 1.12%, indicate a niche market segment that is likely focused on certified pre-owned vehicles, emphasizing quality and reliability assurances. This distribution highlights the competitive landscape and consumer preferences within the online vehicle booking market, suggesting that while traditional dealers hold the majority, there is a notable opportunity for growth in both individual and Trustmark dealer segments.



In conclusion, the pie chart suggests that the Online Vehicle Booking Market is predominantly driven by traditional fuel types (petrol and diesel), with a minor yet noteworthy presence of CNG and electric vehicles. This distribution underscores the importance of focusing on petrol and diesel vehicles for immediate market strategies while also recognizing the growing potential of alternative fuel vehicles.

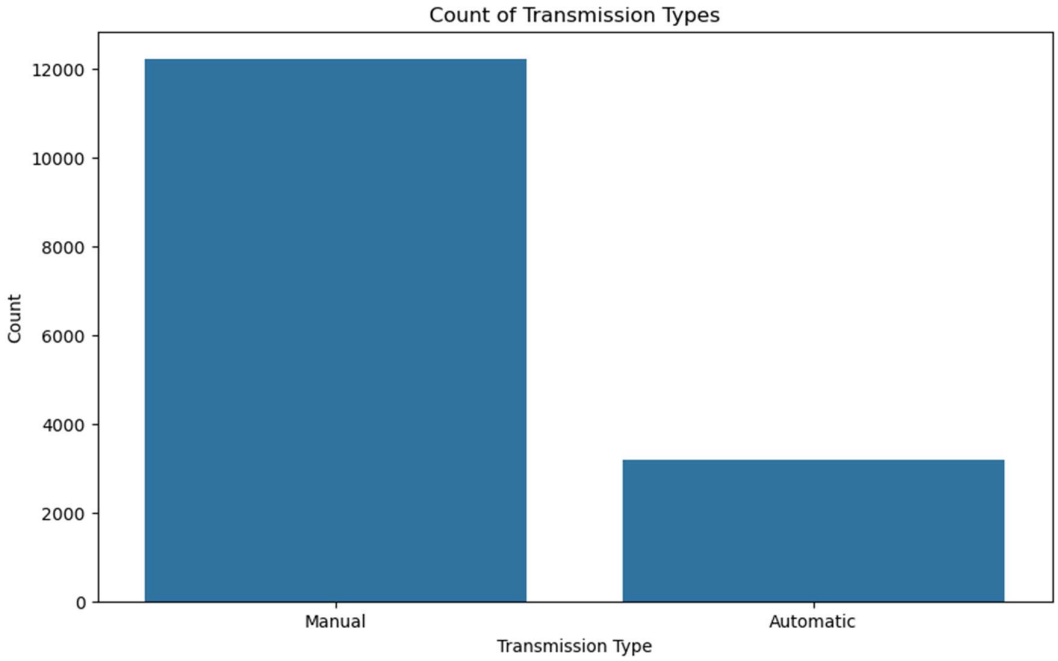


Based on the bar graph analysis of car names and fuel types in the Online Vehicle Booking Market, Hyundai i20 and Maruti Swift Dzire emerge as the most popular vehicles, with 906 and 890 bookings respectively. These two models significantly outperform other cars, which average around 500 bookings. This indicates a strong consumer preference for these specific models, likely due to factors such as brand reputation, fuel efficiency, affordability, and overall value. The dominance of Hyundai i20 and Maruti Swift Dzire suggests that they are highly trusted and sought after in the market, making them key models for dealers and individual sellers to focus on. The data highlights the importance of these top-performing vehicles in driving market trends and consumer choices within the online vehicle booking industry.

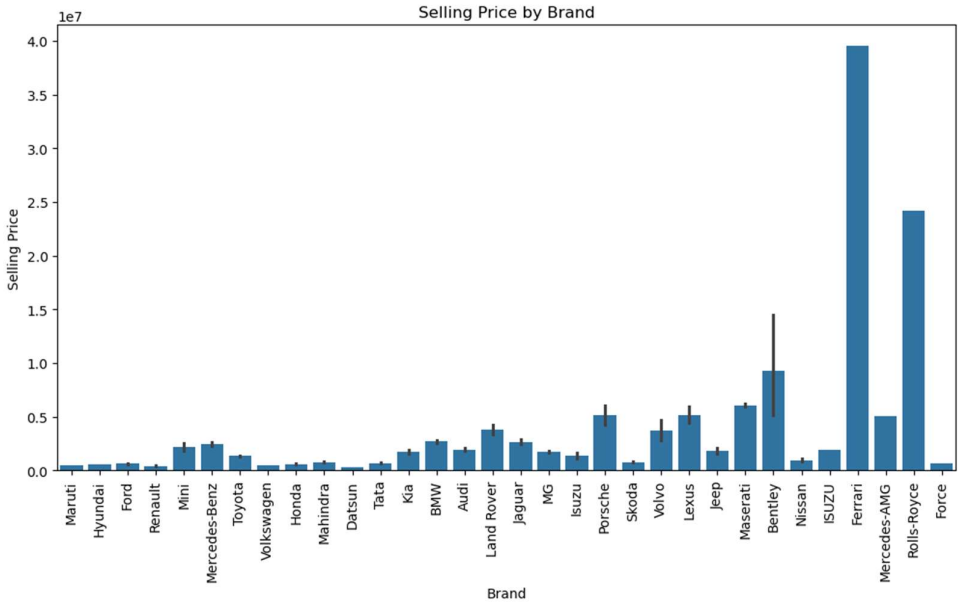


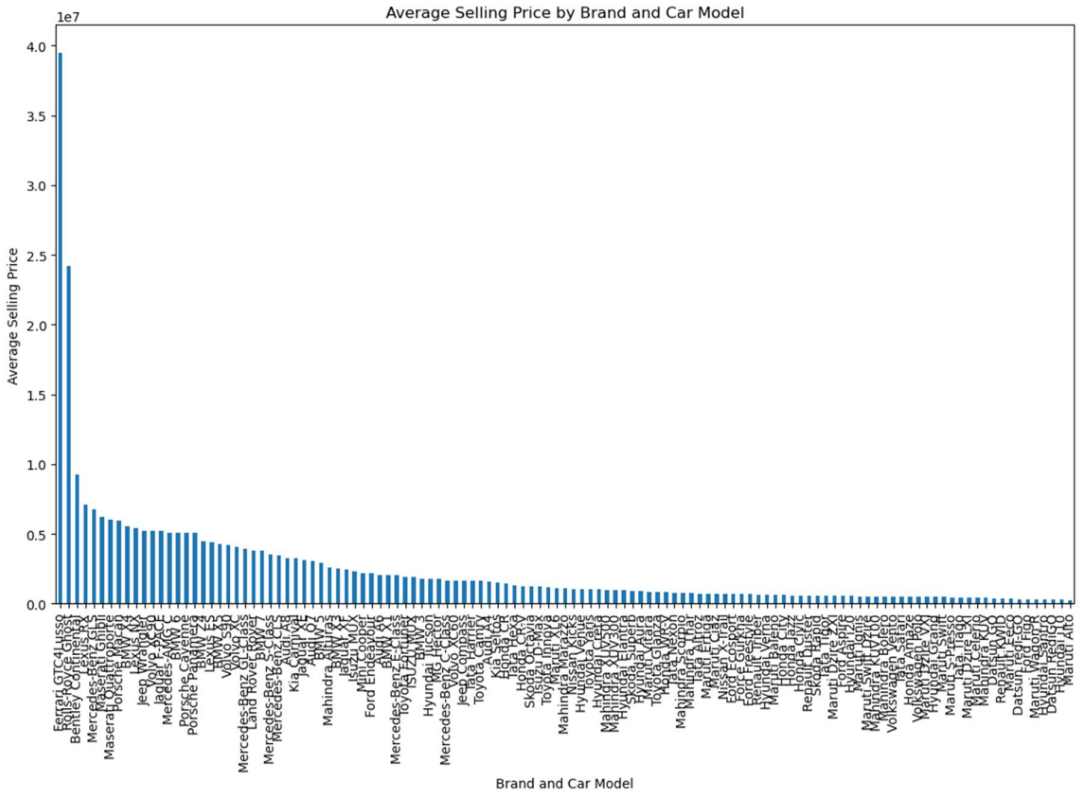
Based on the chart showing the distribution of car brands in the Online Vehicle Booking Market, Maruti stands out with the highest count, exceeding 5000 bookings. This indicates that Maruti is the most preferred brand among consumers, likely due to its strong market presence,

reputation for reliability, affordability, and extensive service network. The overwhelming preference for Maruti suggests that it plays a crucial role in the online vehicle booking market, making it a dominant force that shapes consumer behaviour and market dynamics. For stakeholders, this insight emphasizes the importance of focusing on Maruti models to meet consumer demand and capitalize on its popularity. Other brands, while still significant, trail behind, highlighting the competitive advantage Maruti holds in this sector.

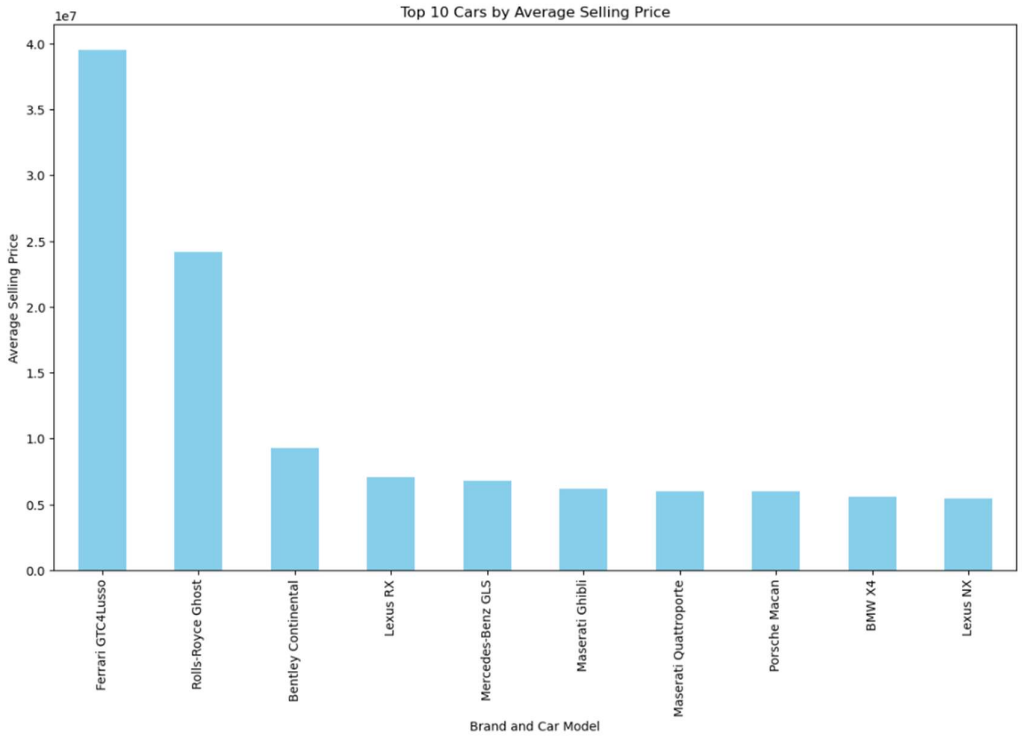


The market is currently skewed towards manual transmission vehicles, but the notable presence of automatic transmission cars highlights an opportunity for growth in this segment. Dealers and sellers should continue to focus on offering a wide range of manual transmission vehicles while also expanding their inventory of automatic cars to cater to evolving consumer preferences and capture a larger share of the market.





The graphs suggests that while the overall market is diverse, there is a lucrative segment for luxury car brands like Ferrari and Rolls-Royce. For sellers and dealers, this underscores the importance of targeting high-net-worth individuals and positioning these premium brands effectively to maximize profitability. Additionally, it indicates that the online vehicle booking platform can cater to a wide range of consumer segments, from budget-conscious buyers to luxury car enthusiasts.



Here are the top 10 cars based on average selling prizes.

From the analysis of the Online Vehicle Booking Market, several key conclusions can be drawn. The market exhibits a diverse range of consumer preferences and trends, with clear indications of strong demand for specific brands, such as Maruti, and transmission types, predominantly manual. Maruti emerges as the most popular brand, reflecting its widespread consumer trust and affordability. Manual transmission cars dominate the market, emphasizing their popularity due to lower costs and higher fuel efficiency. Additionally, luxury brands like Ferrari and Rolls-Royce command significantly higher selling prices, highlighting a niche market for affluent buyers seeking premium vehicles. Overall, the dataset underscores the importance of understanding and catering to diverse consumer segments, from budget-conscious buyers favouring practicality to high-end consumers seeking luxury and exclusivity in their vehicle purchases. This comprehensive understanding is essential for stakeholders in the online vehicle booking market to effectively target their offerings and maximize market penetration and profitability.

- **Link to GitHub repository**

https://github.com/BhavyaParekh/Online_Vehicle_Booking_Market_Analysis

EV Market Analysis
Makam Sujal Kumar

- **Problem Statement:**

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

- **Data Sources**: https://www.kaggle.com/datasets/nimish23/olatrips?resource=download

- **Data Preprocessing:**

Importing the Libraries :

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings(action='ignore')

# display all columns of the dataframe
pd.options.display.max_columns = None

# display all rows of the dataframe
pd.options.display.max_rows = None

# to display the float values upto 6 decimal places
pd.options.display.float_format = '{:.3f}'.format

# import train-test split
from sklearn.model_selection import train_test_split


#from termcolor import colored
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix
from sklearn.metrics import confusion_matrix
from matplotlib.colors import ListedColormap
```

Loading the Dataset :

```
[189] data = pd.read_csv('/content/All-time Table-Bangalore-Wards.csv', sep =',')

[190] data.head()
```

	Ward	Searches	Searches which got estimate	Searches for Quotes	Searches which got Quotes	Bookings	Completed Trips	Search-to-estimate Rate	Estimate-to-search for quotes Rate	Quote Acceptance Rate	Quote-to-booking Rate	Cancelled Bookings	Booking Cancellation Rate	Conversion Rate	Drivers' Earnings	Average Distance per Trip (km)	Average Fare per Trip
	Other Wards	30,71,626	27,12,391	18,90,877	9,75,708	9,54,414	5,89,585	88.3%	69.7%	51.6%	97.8%	3,64,809	38.2%	19.2%	₹12,00,07,713	10.370	₹175.04
	Shantala Nagar	24,01,161	23,84,599	20,67,853	8,56,568	8,41,607	4,74,271	99.3%	86.7%	41.4%	98.3%	3,67,010	43.6%	19.8%	₹7,11,77,412	7.480	₹138.74
	oddakanahalli	17,27,250	17,11,471	14,56,511	5,96,480	5,87,089	3,50,792	99.1%	85.1%	40.9%	98.4%	2,36,139	40.2%	20.3%	₹5,92,11,778	9.020	₹159.54
	Agara	12,28,026	12,14,797	10,43,176	5,20,539	5,08,562	3,36,728	99.1%	85.9%	49.9%	97.7%	1,71,714	33.8%	27.5%	₹4,75,68,017	7.630	₹137.67
	Koramangala	11,69,452	11,58,300	9,94,218	5,28,008	5,15,062	3,34,904	99.1%	85.8%	53.1%	97.5%	1,80,001	34.9%	28.6%	₹4,45,96,808	8.970	₹128.74

Exploratory Data Analysis :

Understanding the features

```
data.describe()
```

	Average Distance per Trip (km)
count	245.000
mean	7.412
std	1.135
min	5.010
25%	6.580
50%	7.220
75%	8.190
max	11.420

```
[192] data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 245 entries, 0 to 244
Data columns (total 18 columns):
 #   Column                            Non-Null Count  Dtype  
---  ------                            --------------  -----  
 0   Ward                              245 non-null    object 
 1   Searches                          245 non-null    object 
 2   Searches which got estimate       245 non-null    object 
 3   Searches for Quotes               245 non-null    object 
 4   Searches which got Quotes         245 non-null    object 
 5   Bookings                          245 non-null    object 
 6   Completed Trips                   245 non-null    object 
 7   Search-to-estimate Rate           245 non-null    object 
 8   Estimate-to-search for quotes Rate 245 non-null   object 
 9   Quote Acceptance Rate             245 non-null    object 
 10  Quote-to-booking Rate             245 non-null    object 
 11  Cancelled Bookings                245 non-null    object 
 12  Booking Cancellation Rate         245 non-null    object 
 13  Conversion Rate                   245 non-null    object 
 14  Drivers' Earnings                 245 non-null    object 
 15  Average Distance per Trip (km)    245 non-null    float64
 16  Average Fare per Trip             245 non-null    object 
 17  Distance Travelled (km)           245 non-null    object 
dtypes: float64(1), object(17)
memory usage: 34.6+ KB
```

Dropping the Duplicates :

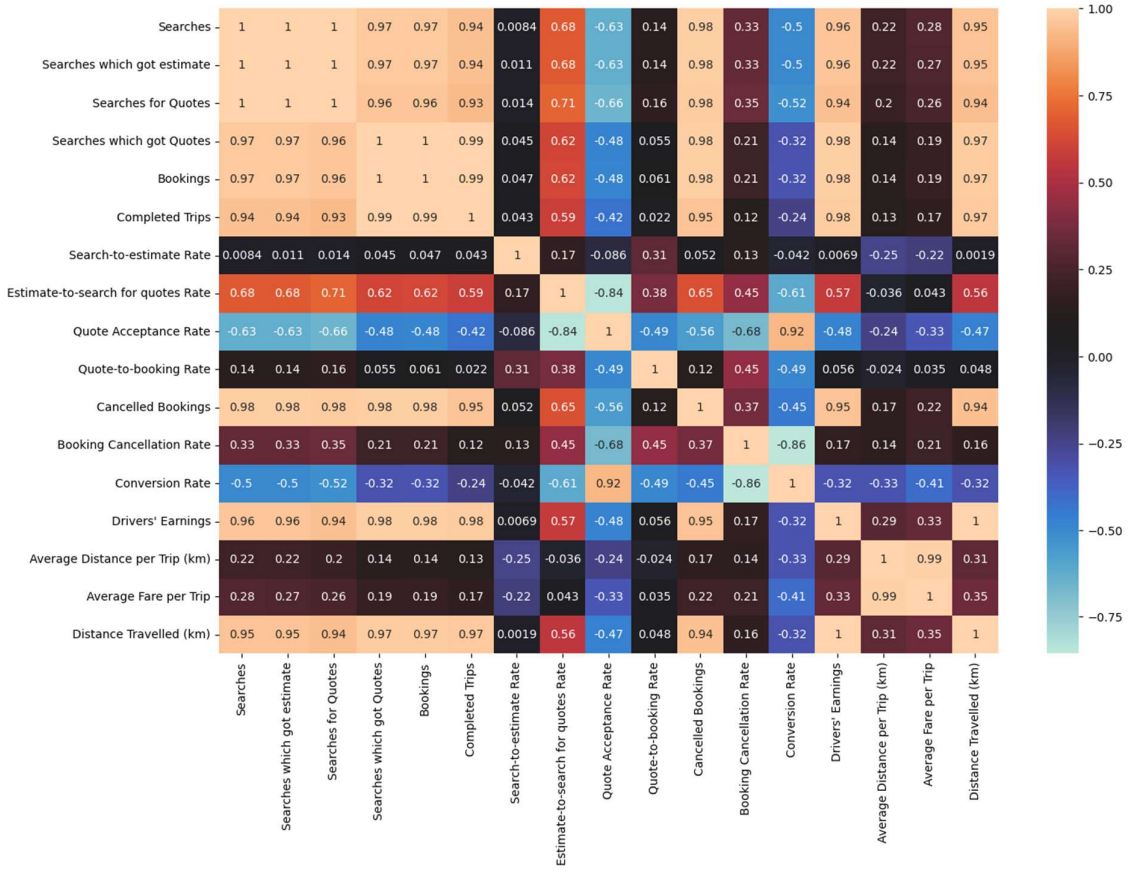
Droping the Duplicates

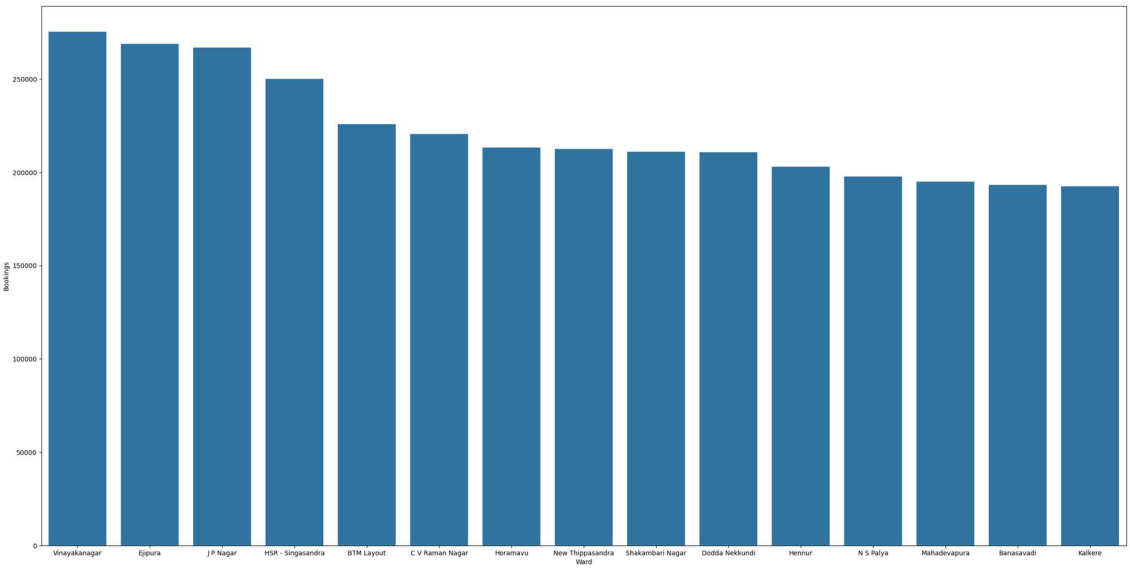
```
[194] data.duplicated().sum()
```

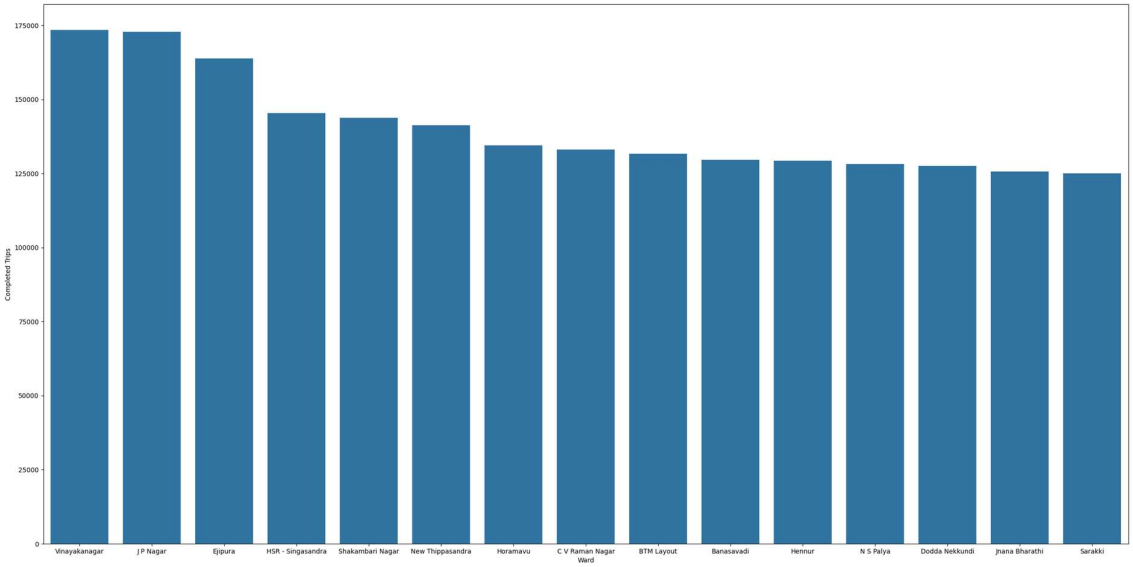
```
0
```

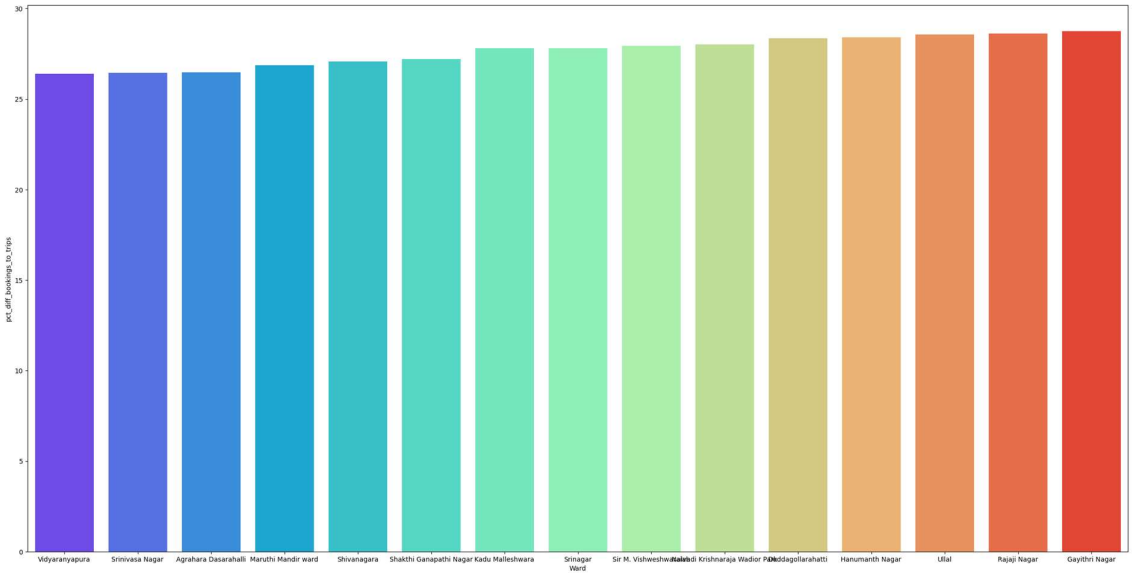
```
[195] data.drop_duplicates(inplace = True)
```

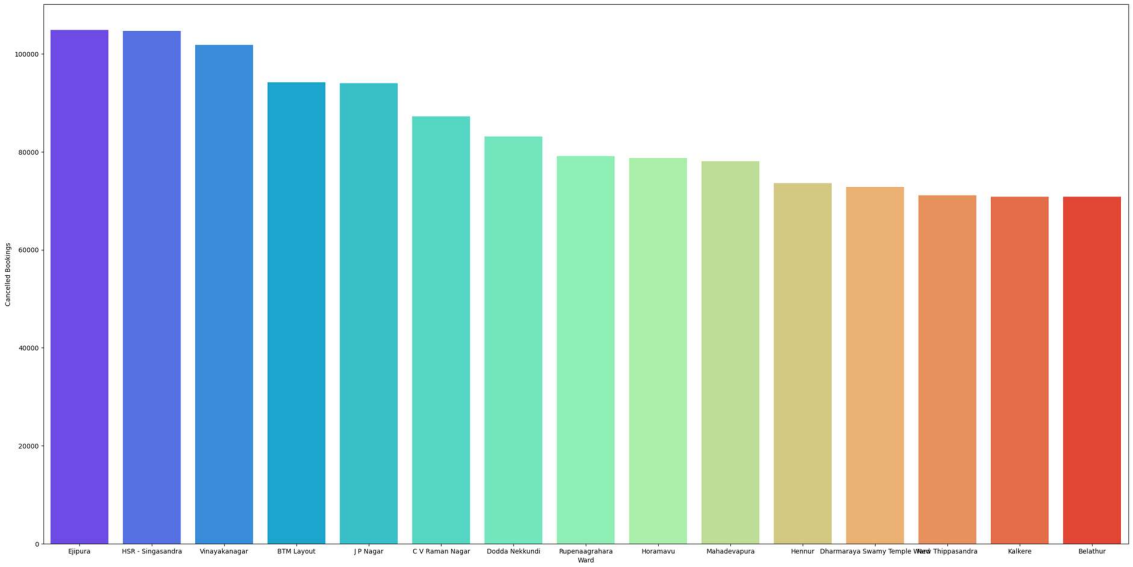
Heatmap :

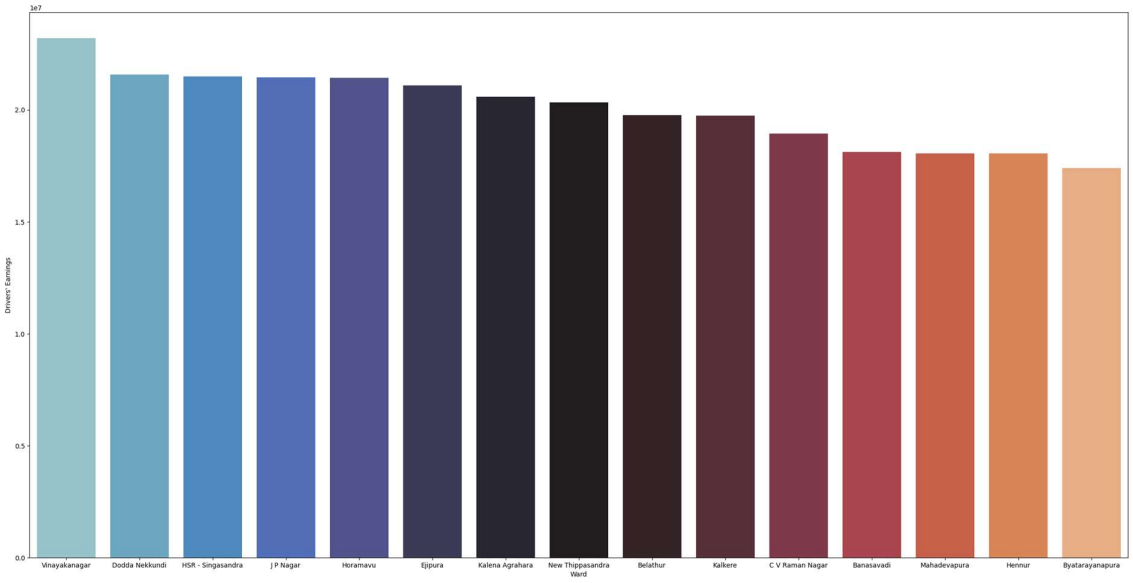

Searches 1 1 1 0.97 0.97 0.94 0.0084 0.68 -0.63 0.14 0.98 0.33 -0.5 0.96 0.22 0.28 0.95
Searches which got estimate 1 1 1 0.97 0.97 0.94 0.011 0.68 -0.63 0.14 0.98 0.33 -0.5 0.96 0.22 0.27 0.95
Searches for Quotes 1 1 1 0.96 0.96 0.93 0.014 0.71 -0.66 0.16 0.98 0.35 -0.52 0.94 0.2 0.26 0.94
Searches which got Quotes 0.97 0.97 0.96 1 1 0.99 0.045 0.62 -0.48 0.055 0.98 0.21 -0.32 0.98 0.14 0.19 0.97
Bookings 0.97 0.97 0.96 1 1 0.99 0.047 0.62 -0.48 0.061 0.98 0.21 -0.32 0.98 0.14 0.19 0.97
Completed Trips 0.94 0.94 0.93 0.99 0.99 1 0.043 0.59 -0.42 0.022 0.95 0.12 -0.24 0.98 0.13 0.17 0.97
Search-to-estimate Rate 0.0084 0.011 0.014 0.045 0.047 0.043 1 0.17 -0.086 0.31 0.052 0.13 -0.042 0.0069 -0.25 -0.22 0.0019
Estimate-to-search for quotes Rate 0.68 0.68 0.71 0.62 0.62 0.59 0.17 1 -0.84 0.38 0.65 0.45 -0.61 0.57 -0.036 0.043 0.56
Quote Acceptance Rate -0.63 -0.63 -0.66 -0.48 -0.48 -0.42 -0.086 -0.84 1 -0.49 -0.56 -0.68 0.92 -0.48 -0.24 -0.33 -0.47
Quote-to-booking Rate 0.14 0.14 0.16 0.055 0.061 0.022 0.31 0.38 -0.49 1 0.12 0.45 -0.49 0.056 -0.024 0.035 0.048
Cancelled Bookings 0.98 0.98 0.98 0.98 0.98 0.95 0.052 0.65 -0.56 0.12 1 0.37 -0.45 0.95 0.17 0.22 0.94
Booking Cancellation Rate 0.33 0.33 0.35 0.21 0.21 0.12 0.13 0.45 -0.68 0.45 0.37 1 -0.86 0.17 0.14 0.21 0.16
Conversion Rate -0.5 -0.5 -0.52 -0.32 -0.32 -0.24 -0.042 -0.61 0.92 -0.49 -0.45 -0.86 1 -0.32 -0.33 -0.41 -0.32
Drivers' Earnings 0.96 0.96 0.94 0.98 0.98 0.98 0.0069 0.57 -0.48 0.056 0.95 0.17 -0.32 1 0.29 0.33 1
Average Distance per Trip (km) 0.22 0.22 0.2 0.14 0.14 0.13 -0.25 -0.036 -0.24 -0.024 0.17 0.14 -0.33 0.29 1 0.99 0.31
Average Fare per Trip 0.28 0.27 0.26 0.19 0.19 0.17 -0.22 0.043 -0.33 0.035 0.22 0.21 -0.41 0.33 0.99 1 0.35
Distance Travelled (km) 0.95 0.95 0.94 0.97 0.97 0.97 0.0019 0.56 -0.47 0.048 0.94 0.16 -0.32 1 0.31 0.35 1
Searches
Searches which got estimate
Searches for Quotes
Searches which got Quotes
Bookings
Completed Trips
Search-to-estimate Rate
Estimate-to-search for quotes Rate
Quote Acceptance Rate
Quote-to-booking Rate
Cancelled Bookings
Booking Cancellation Rate
Conversion Rate
Drivers' Earnings
Average Distance per Trip (km)
Average Fare per Trip
Distance Travelled (km)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75


Bookings
250000
200000
150000
100000
50000
0
Vinayakanagar
Ejipura
J P Nagar
HSR - Singasandra
BTM Layout
C V Raman Nagar
Horamavu
New Thippasandra
Shakambari Nagar
Dodda Nekkundi
Hennur
N S Palya
Mahadevapura
Banasavadi
Kalkere
Ward

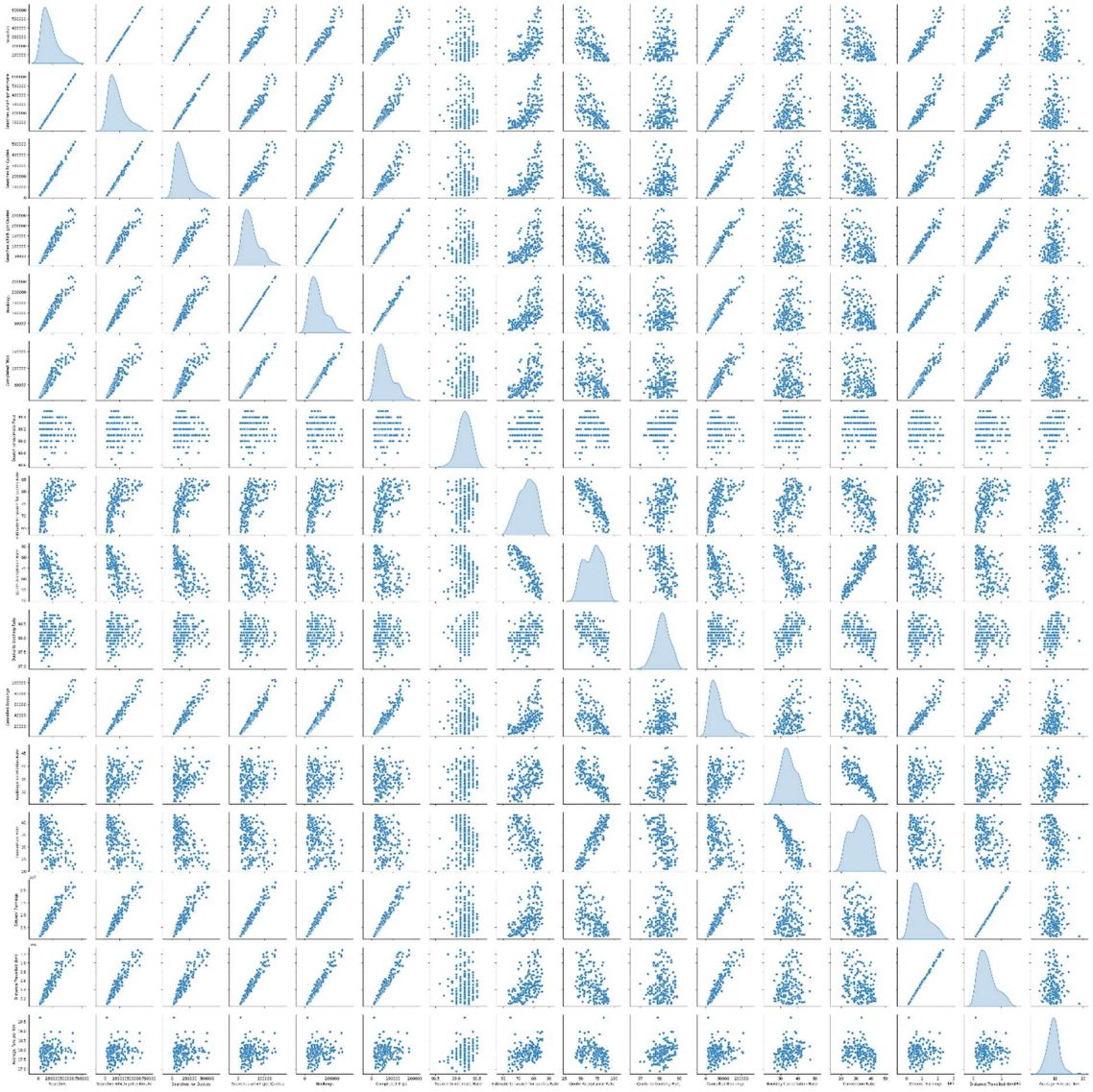

Completed Trips
175000
150000
125000
100000
75000
50000
25000
0
Vinayakanagar
J P Nagar
Ejipura
HSR - Singasandra
Shakambari Nagar
New Thippasandra
Horamavu
C V Raman Nagar
BTM Layout
Banasavadi
Hennur
N S Palya
Dodda Nekkundi
Jnana Bharathi
Sarakki
Ward


pct_diff_bookings_to_trips
30
25
20
15
10
5
0
Vidyaranyapura
Srinivasa Nagar
Agrahara Dasarahalli
Maruthi Mandir ward
Shivanagara
Shakthi Ganapathi Nagar
Kadu Malleshwara
Srinagar
Sir M. Vishweshwaraiah
Ward


Cancelled Bookings
100000
80000
60000
40000
20000
0
Ejipura
HSR - Singasandra
Vinayakanagar
BTM Layout
J P Nagar
C V Raman Nagar
Dodda Nekkundi
Rupenagrahara
Horamavu
Mahadevapura
Hennur
Dharmaraya Swamy Temple Ward
Thippasandra
Kalkere
Belathur
Ward


1e7
Drivers' Earnings
2.0
1.5
1.0
0.5
0.0
Vinayakanagar
Dodda Nekkundi
HSR - Singasandra
J P Nagar
Horamavu
Ejipura
Kalena Agrahara
New Thippasandra
Belathur
Kalkere
C V Raman Nagar
Banasavadi
Mahadevapura
Hennur
Byatarayanapura
Ward

Pair Plot :



Multi Collinearity Test :

Multi collinearity test

```
[254] np.linalg.det(data.corr()).astype(float)
```

```
2.2210653643380272e-35
```

Corellation Heatmap after :

Scaling the data :

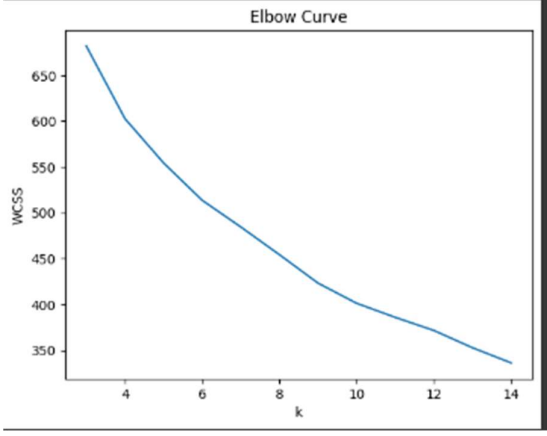
```
[269] # Scaling down the dataset

      from sklearn.preprocessing import StandardScaler
      scaler = StandardScaler()

      scaled_data = scaler.fit_transform(data)
```

• Segment Extraction:
K-Means Clustering :

```
273] # KMeans Clustering

     from sklearn.cluster import KMeans
     # for determing K (Elbow curve)
     k_vals = np.arange(3,15)
     wcss = []
     for i in k_vals:
       kmeans = KMeans(n_clusters = i)
       kmeans.fit(scaled_data)
       wcss.append([i,kmeans.inertia_])

     print(wcss)
```

Elbow Method :

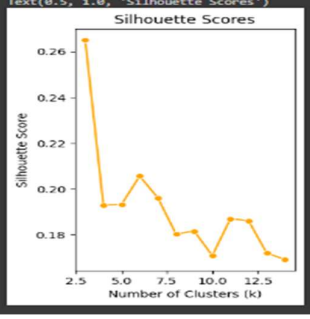
```
wcss = pd.DataFrame(wcss, columns = ['k', 'WCSS'])
sns.lineplot(x = 'k', y = 'WCSS', data = wcss)
plt.ylabel('K')
plt.ylabel('WCSS')
plt.title('Elbow Curve')
plt.show()
```



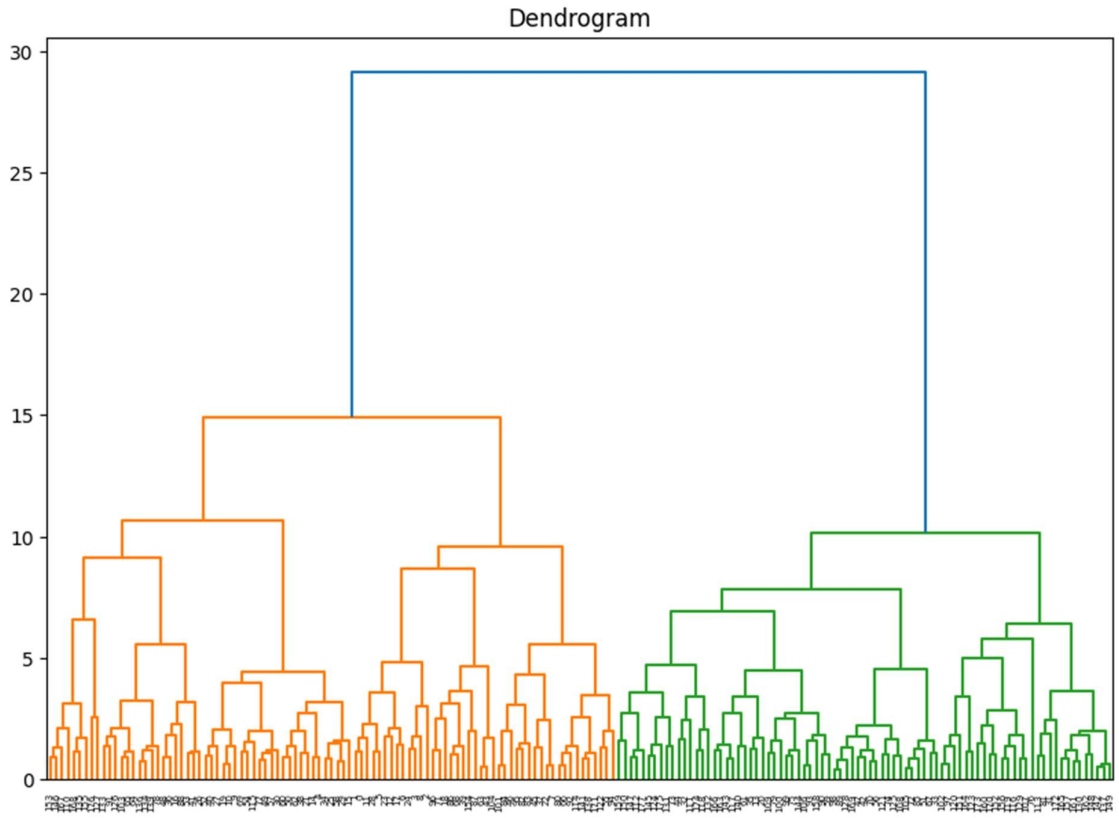
Silhouette Score :

```
[277] from sklearn.metrics import silhouette_score

      k_vals = np.arange(3, 15)
      wcss = []
      silhouette_scores = []

      for i in k_vals:
          kmeans = KMeans(n_clusters=i, random_state=0)
          kmeans.fit(scaled_data)

          # Calculate WCSS (Within-Cluster Sum of Square)
          wcss.append([i, kmeans.inertia_])

          # Calculate silhouette score
          score = silhouette_score(scaled_data, kmeans.labels_)
          silhouette_scores.append([i, score])

      # Convert to DataFrame for plotting
      wcss_df = pd.DataFrame(wcss, columns=['k', 'WCSS'])
      silhouette_df = pd.DataFrame(silhouette_scores, columns=['k', 'Silhouette Score'])
```

```
[278] # Plot Silhouette Scores
      plt.subplot(1, 2, 2)
      sns.lineplot(x='k', y='Silhouette Score', data=silhouette_df, marker='o', color='orange')
      plt.xlabel('Number of Clusters (k)')
      plt.ylabel('Silhouette Score')
      plt.title('Silhouette Scores')

      Text(0.5, 1.0, 'Silhouette Scores')
```

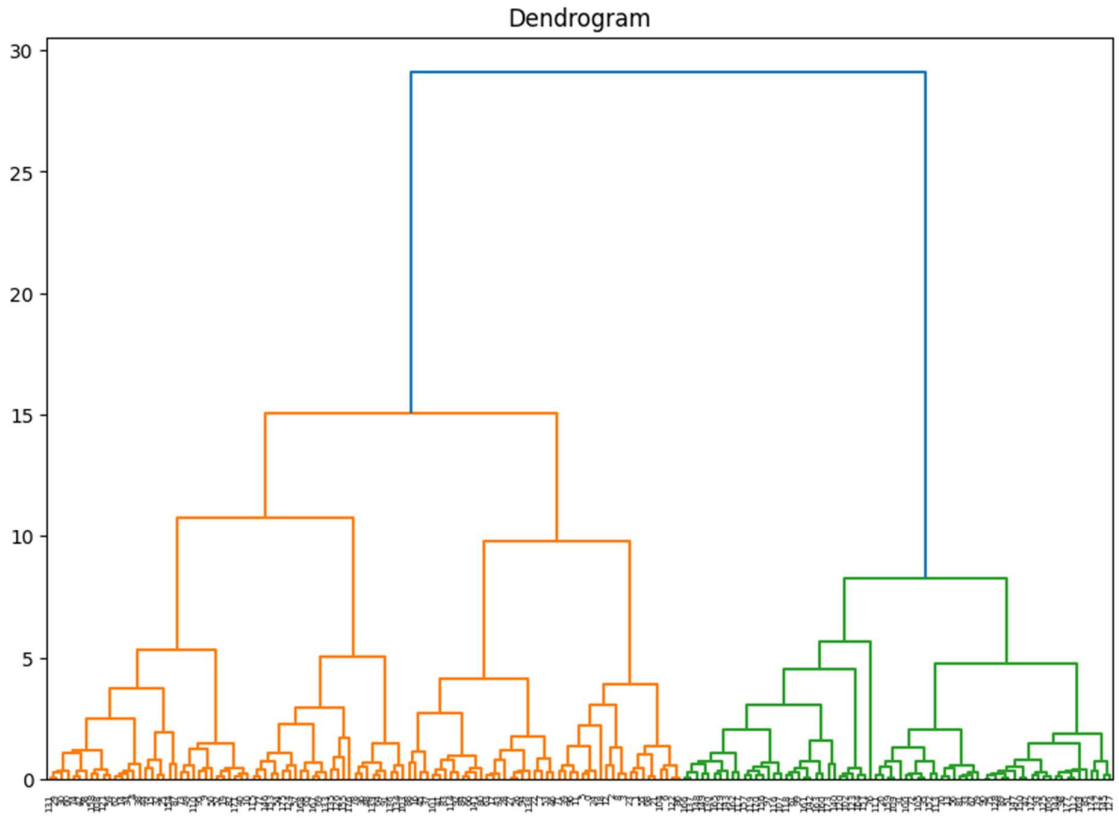


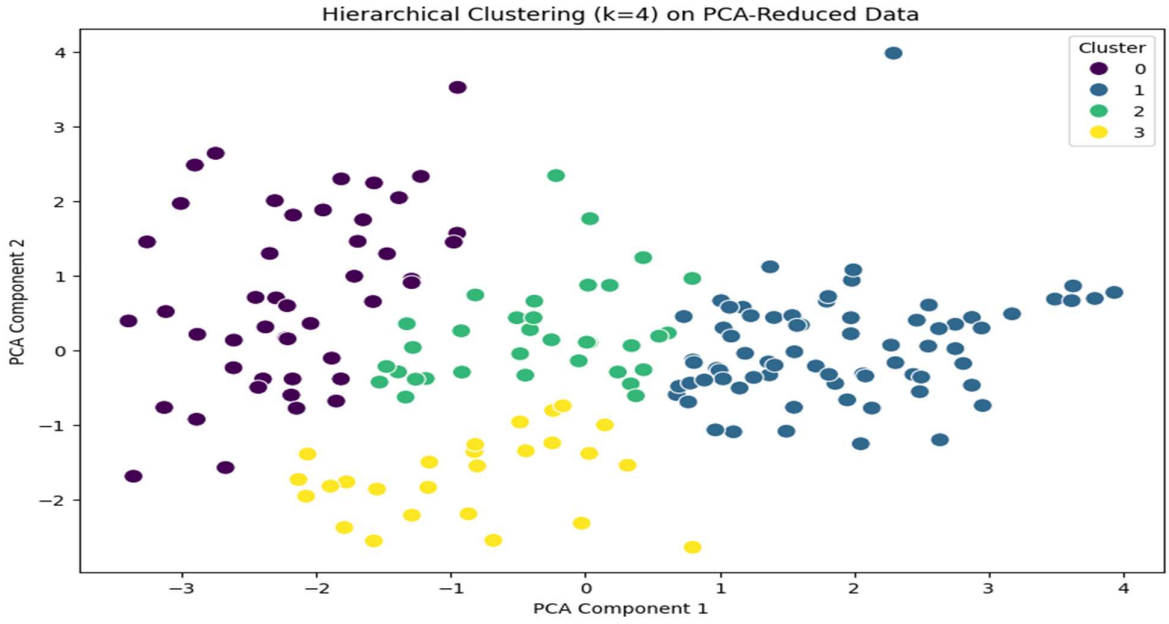
Hierarchical Clustering :
Dendrogram :



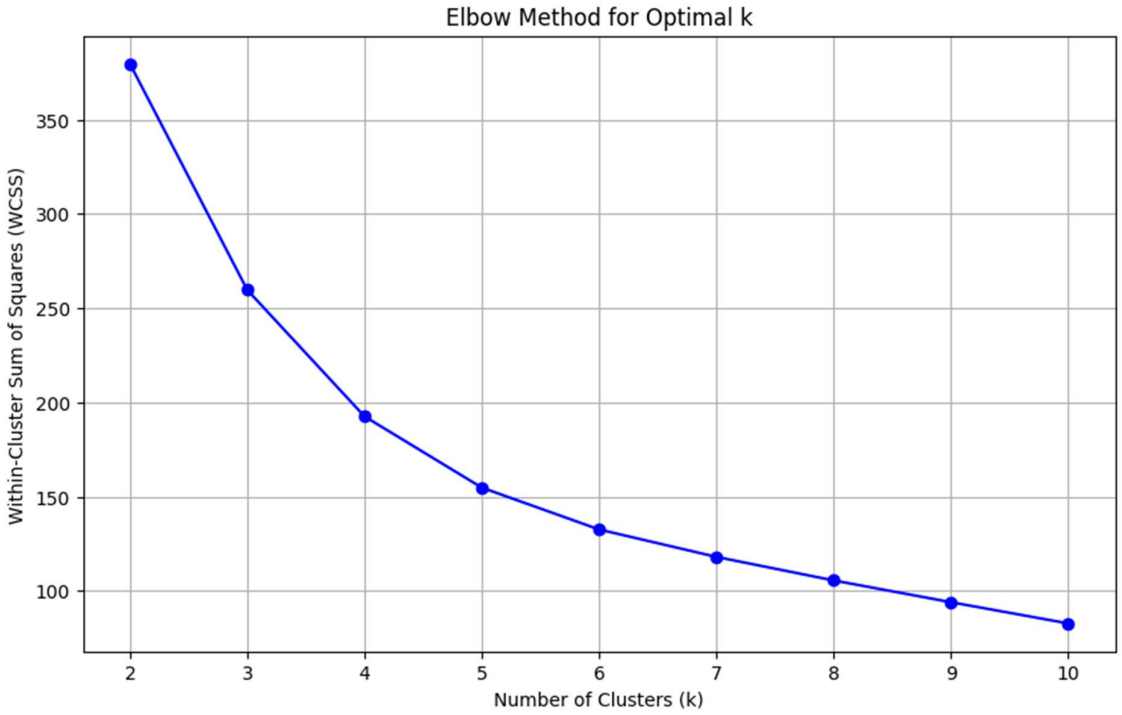
Performing PCA :

```
from sklearn.decomposition import PCA

# Perform PCA to reduce the dimensions to 2 for visualization
pca = PCA(n_components=2)
pca_data = pca.fit_transform(scaled_data)

# Convert PCA results to DataFrame for easier plotting
pca_df = pd.DataFrame(pca_data, columns=['PCA1', 'PCA2'])

# Dendrogram for hierarchical clustering
linked = linkage(pca_data, method='ward')

plt.figure(figsize=(10, 7))
dendrogram(linked, orientation='top', distance_sort='descending', show_leaf_counts=True)
plt.title('Dendrogram')
plt.show()

# Hierarchical Clustering with k=4
k = 4
hc = AgglomerativeClustering(n_clusters=k, affinity='euclidean', linkage='ward')
hc_labels = hc.fit_predict(pca_data)

# Calculate silhouette score for k=4
silhouette_avg = silhouette_score(pca_data, hc_labels)
print(f'Silhouette Score for k={k}: {silhouette_avg}')

# Add cluster labels to PCA DataFrame
pca_df['Cluster'] = hc_labels

# Plotting the clusters
plt.figure(figsize=(10, 7))
sns.scatterplot(x='PCA1', y='PCA2', hue='Cluster', palette='viridis', data=pca_df, s=100)
plt.title(f'Hierarchical Clustering (k={k}) on PCA-Reduced Data')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.show()
```





Elbow Method after PCA :



Area-Wise Analysis and Segment Profiling

1. **Data Exploration and Segment Identification**
    - **Exploratory Data Analysis (EDA)**: Conduct comprehensive EDA to understand the geographical distribution of customers using your online vehicle booking service.
    - **Segment Identification**: Use clustering techniques (e.g., KMeans clustering) to identify distinct segments based on geographical patterns, customer behavior, and preferences.

2. **Profiling Market Segments**
    - **Segment Characteristics**: Summarize each identified segment based on key demographic and behavioral attributes.
    - **Segment Needs and Preferences**: Highlight specific needs and preferences of each segment related to vehicle preferences, pricing sensitivity, service expectations, and geographical preferences.

3. **Selection of Target Segments**
    - **Strategic Considerations**: Discuss the strategic implications of selecting target segments, such as long-term impact on business performance and alignment with organizational goals.

- **Examples of Target Segments**:
  - **Urban Commuters**: Focus on reliable, cost-effective transportation solutions for daily commuting needs.
  - **Tourists and Leisure Travelers**: Provide flexible and convenient options for airport transfers and sightseeing trips.
  - **Corporate Clients**: Offer customized services for business events and employee transportation.
  - **Special Events**: Cater to weddings, parties, and conferences with premium and personalized services.

4. **Customizing the Marketing Mix**
   - **Tailored Strategies**: Develop marketing strategies tailored to each target segment, encompassing the 4Ps (Product, Price, Place, Promotion).
   - **Examples**:
     - **Product**: Adapt vehicle fleet and service offerings to meet segment-specific preferences.
     - **Price**: Introduce pricing strategies that appeal to each segment's budget constraints and willingness to pay.
     - **Place**: Optimize distribution channels and service availability based on geographical demand.
     - **Promotion**: Design promotional campaigns that resonate with the communication styles and media consumption habits of each segment.

5. **Optimal Market Segment**
   - **Data-Driven Insights**: Use insights from area-wise analysis and clustering to identify the most optimal market segment.
   - **Criteria for Optimal Segment**: Consider factors such as market size, growth potential, profitability, and alignment with organizational capabilities.
   - **Example**: Based on analysis, identify segments like urban commuters or business travelers who exhibit higher frequency and potential for repeat bookings.

6. **Key Considerations for Each Segment**
   - **Young Professionals in Urban Areas**: Focus on digital marketing and seamless service integration.
   - **Students and Young Adults**: Offer cost-effective options and leverage partnerships with educational institutions.
   - **Suburban Commuters**: Emphasize reliability and community engagement through local partnerships.
   - **Tourists and Business Travelers**: Enhance convenience through collaborations with travel-related businesses and multilingual support.

7. **Conclusion**
    - **Strategic Direction**: Emphasize the importance of segment-specific strategies in capturing diverse customer bases and sustaining growth.
    - **Future Growth Potential**: Highlight opportunities for expansion and market penetration based on identified segments and their unique requirements.

Github Link : https://github.com/Sujal9079/ev_market

EV Market Segmentation
Kota Anusha

Importing Libraries :

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

Loading the Dataset :

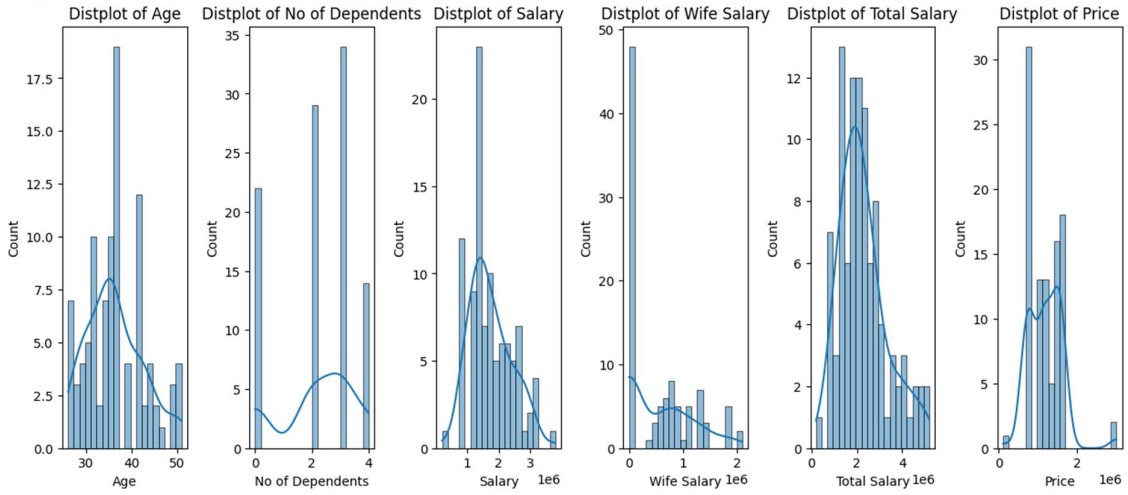
```
print("Loading data...")
data = pd.read_csv("/content/Indian automoble buying behavour study 1.0.csv", encoding= "ISO-8859-1")
print("loaded data")
```

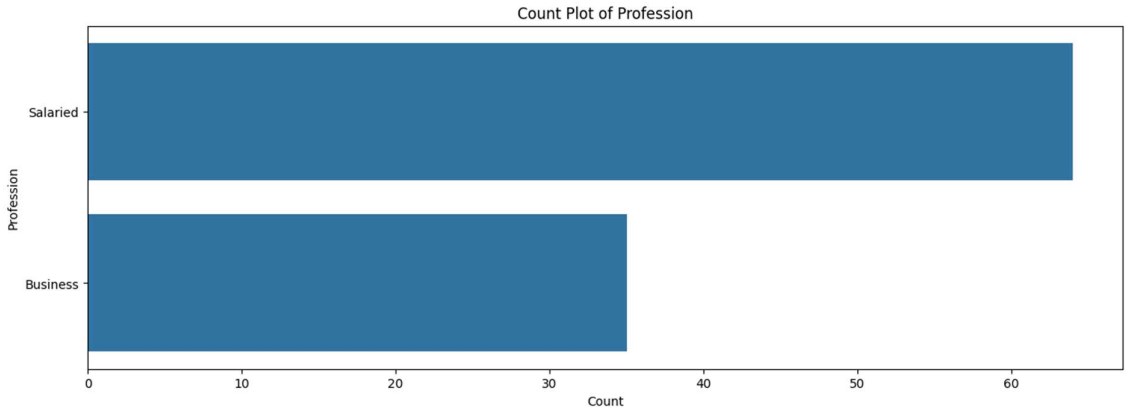
```
Loading data...
loaded data
```

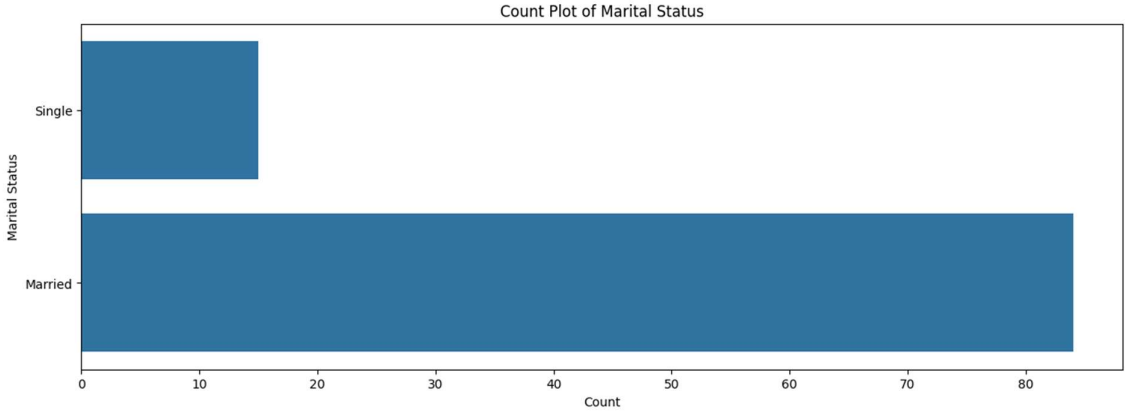
```
data.head()
```

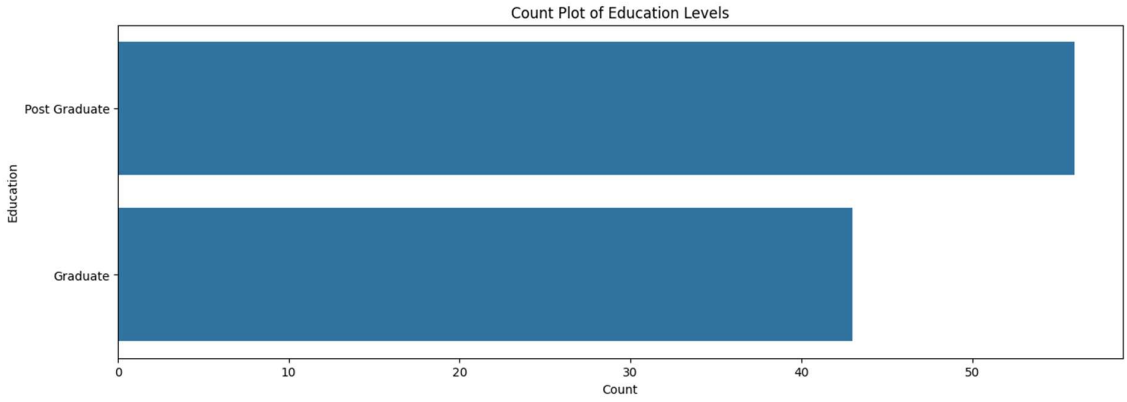
	Age	Profession	Marrital Status	Education	No of Dependents	Personal loan	House Loan	Wife Working	Salary	Wife Salary	Total Salary	Make	Price
0	27	Salaried	Single	Post Graduate	0	Yes	No	No	800000	0	800000	i20	800000
1	35	Salaried	Married	Post Graduate	2	Yes	Yes	Yes	1400000	600000	2000000	Ciaz	1000000
2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

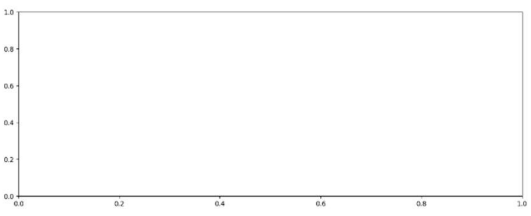
Exploratory Data Analysis :

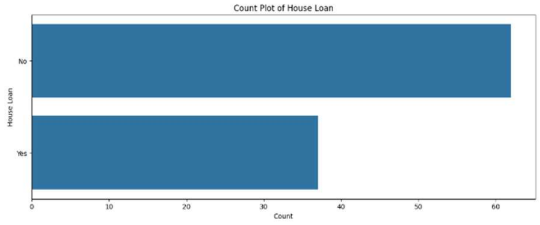


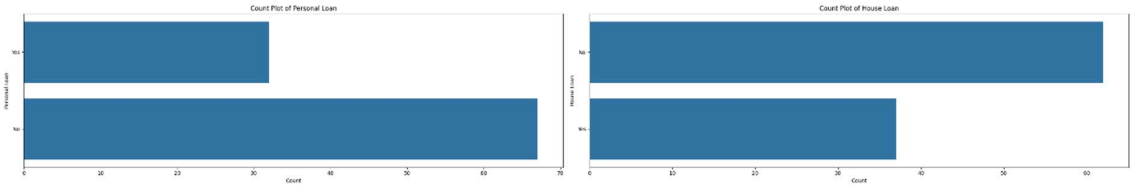

Count Plot of Profession
Salaried
Business
Profession
0
10
20
30
40
50
60
Count

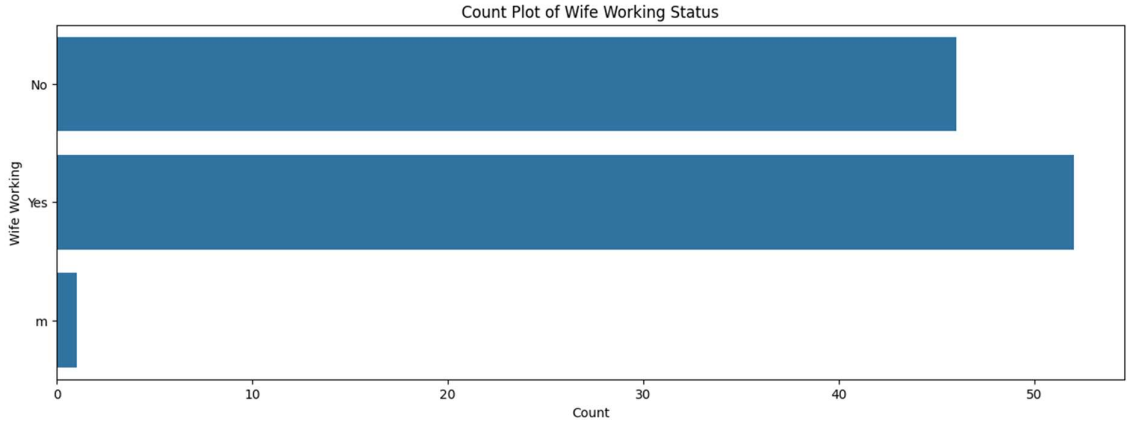

Count Plot of Marital Status
Single
Married
Marital Status
0
10
20
30
40
50
60
70
80
Count

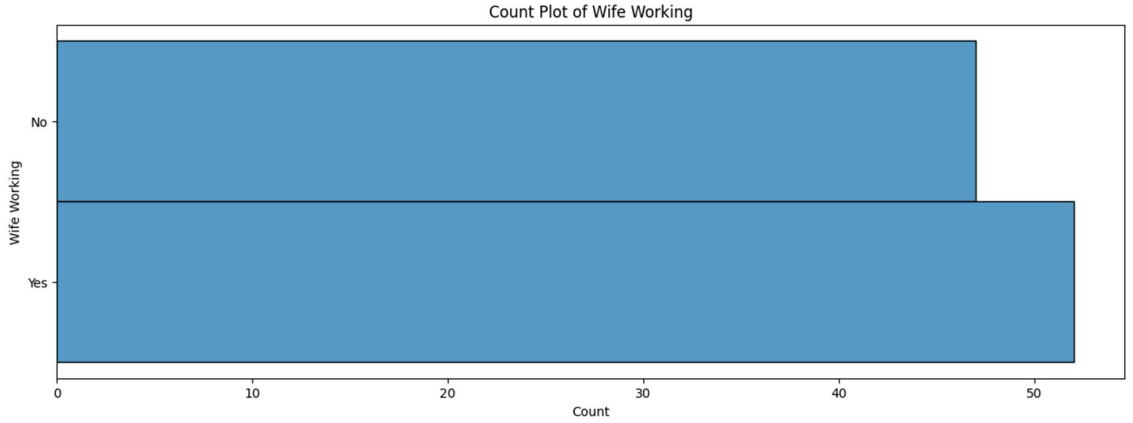

Count Plot of Education Levels
Post Graduate
Graduate
Education
0
10
20
30
40
50
Count

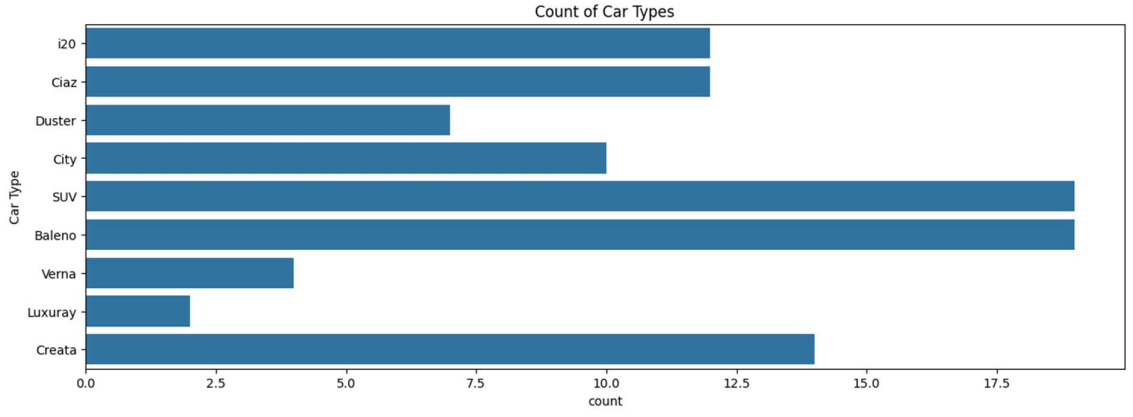

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Count Plot of House Loan
No
Yes
House Loan
0
10
20
30
40
50
60
Count

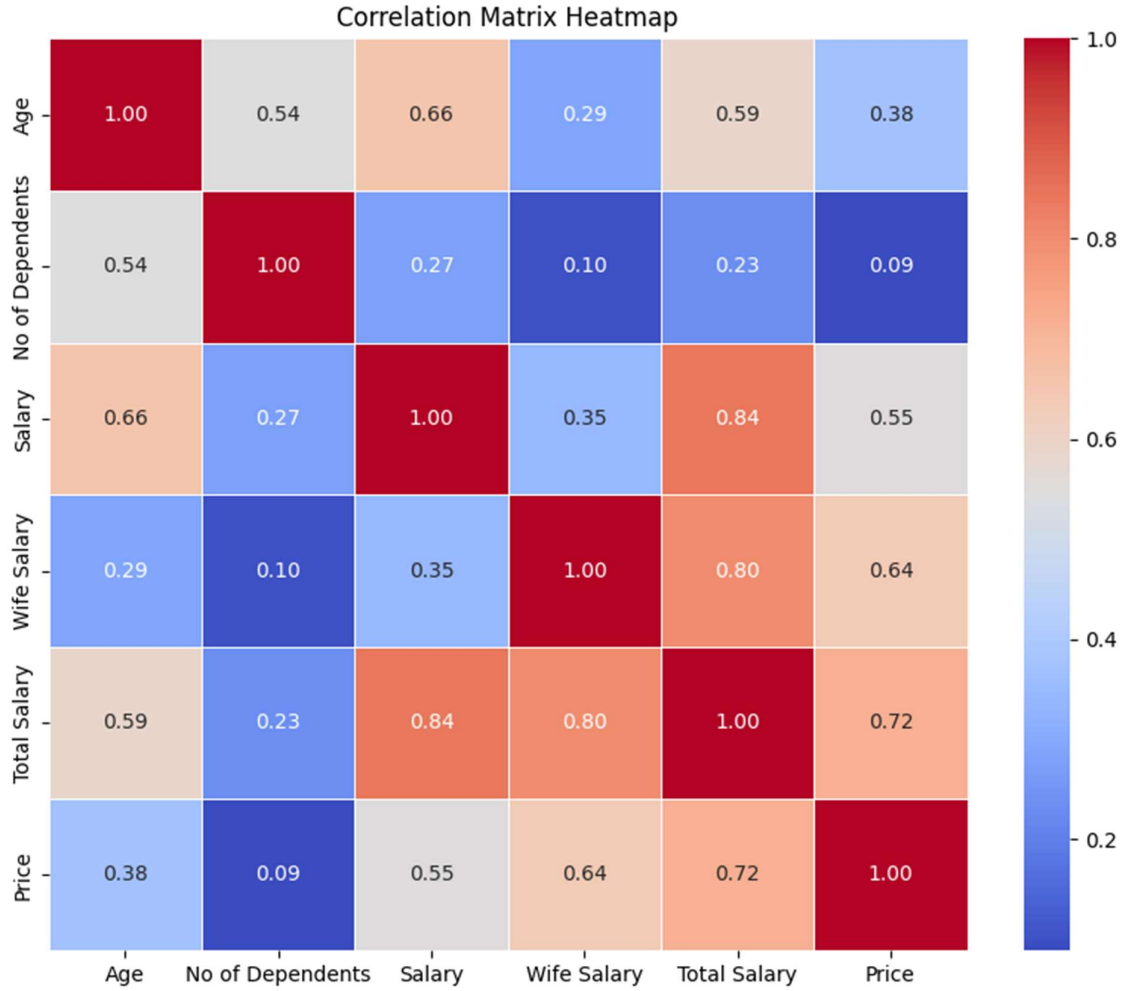

Count Plot of Personal Loan
Yes
No
Personal Loan
0
10
20
30
40
50
60
70
Count
Count Plot of House Loan
No
Yes
House Loan
0
10
20
30
40
50
60
Count

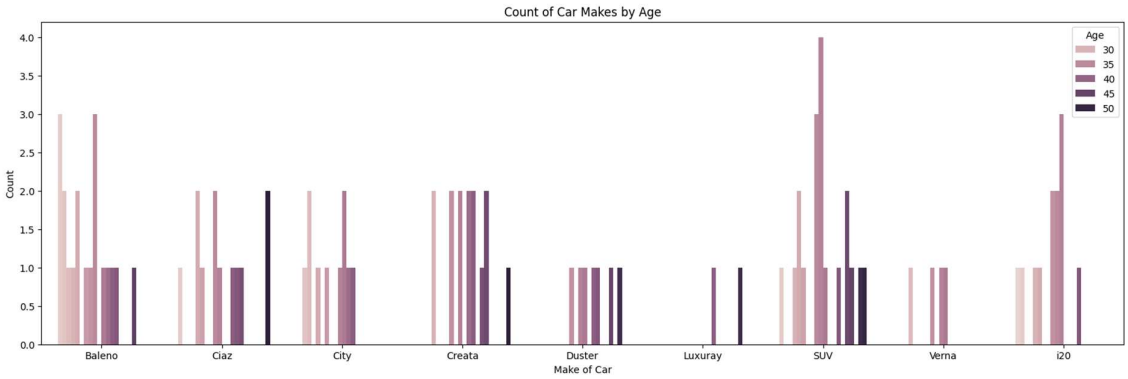

Count Plot of Wife Working Status
No
Yes
m
Wife Working
0
10
20
30
40
50
Count


Count Plot of Wife Working
No
Yes
Wife Working
0
10
20
30
40
50
Count

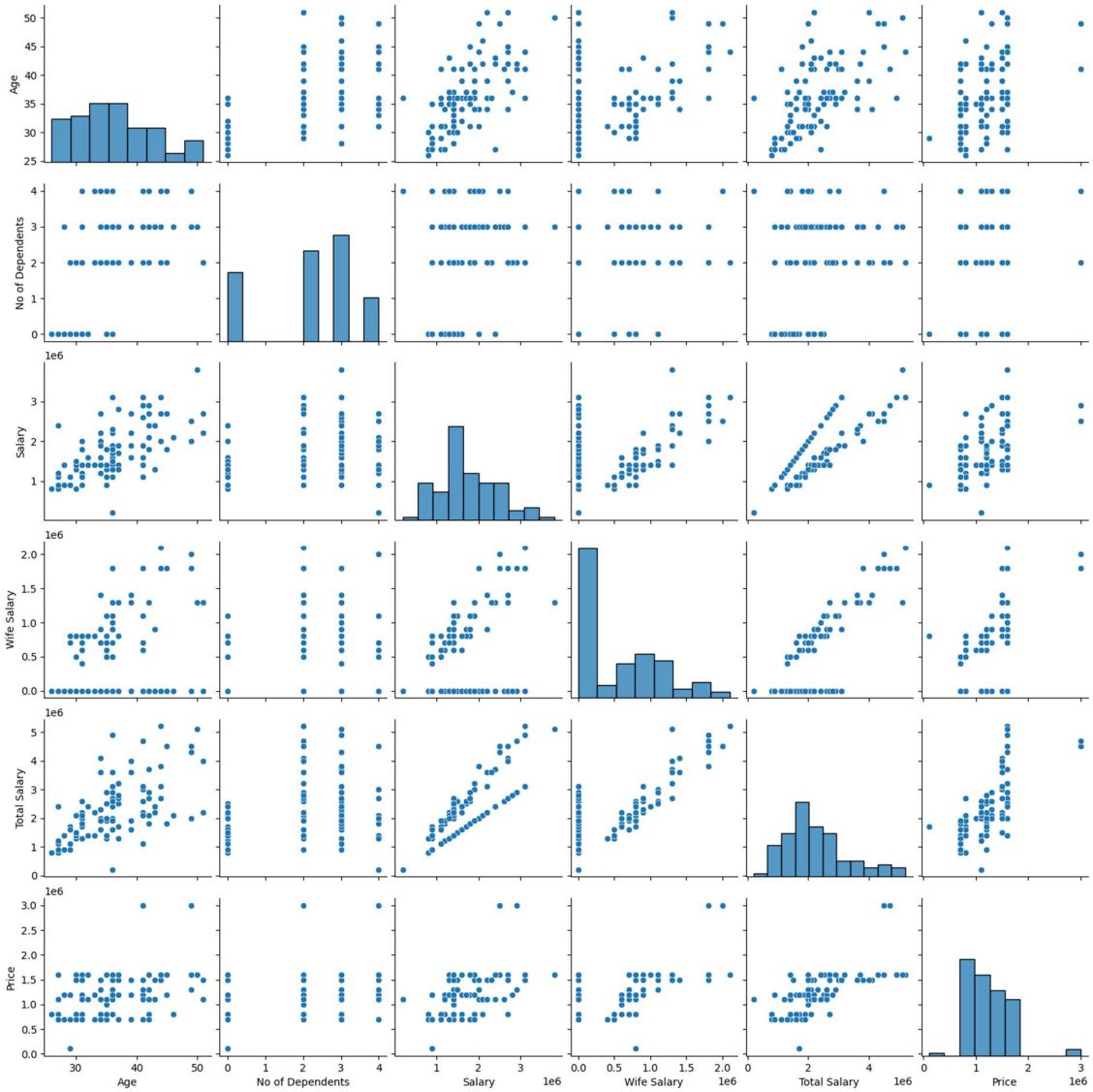

Count of Car Types
i20
Ciaz
Duster
City
SUV
Baleno
Verna
Luxuray
Creata
Car Type
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
count

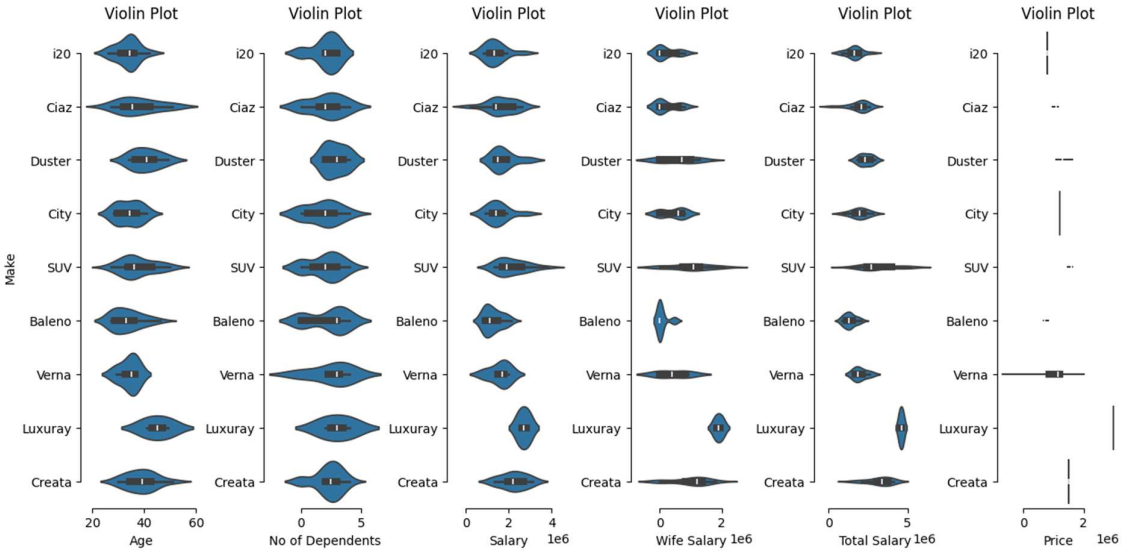
Correlation Matrix :

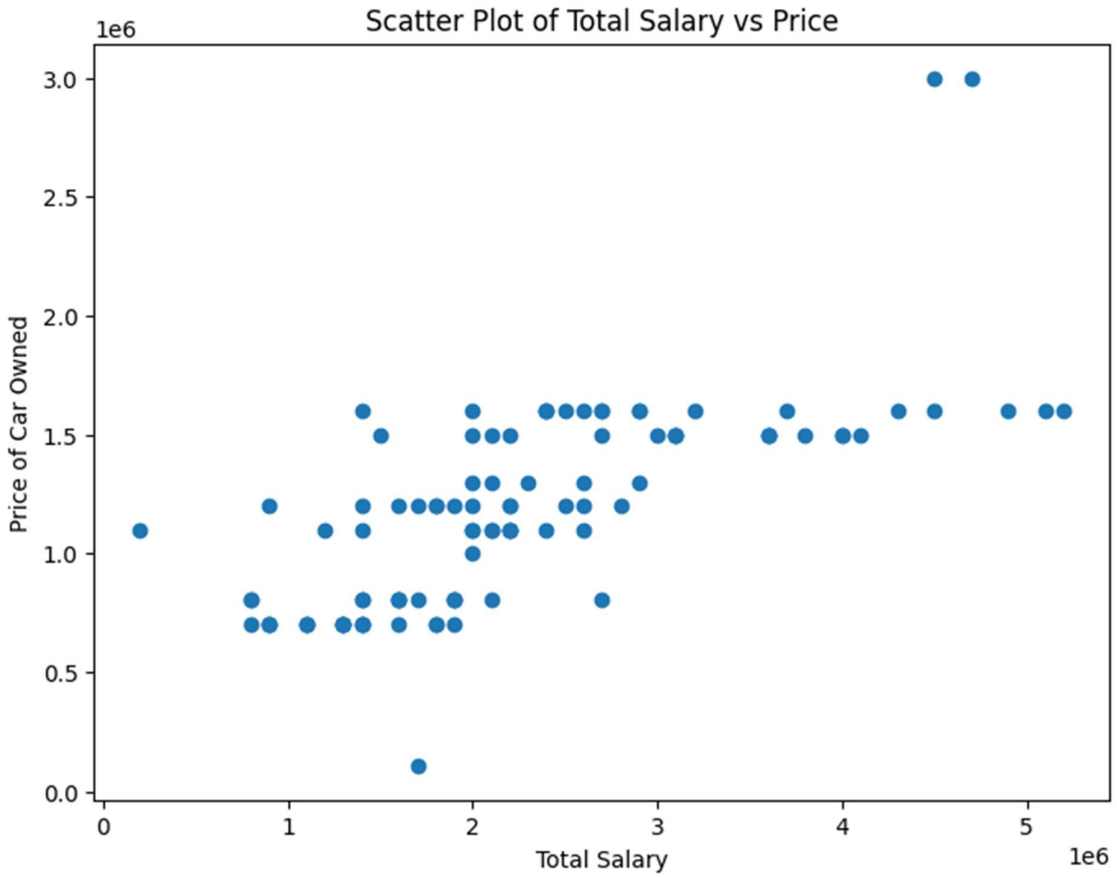


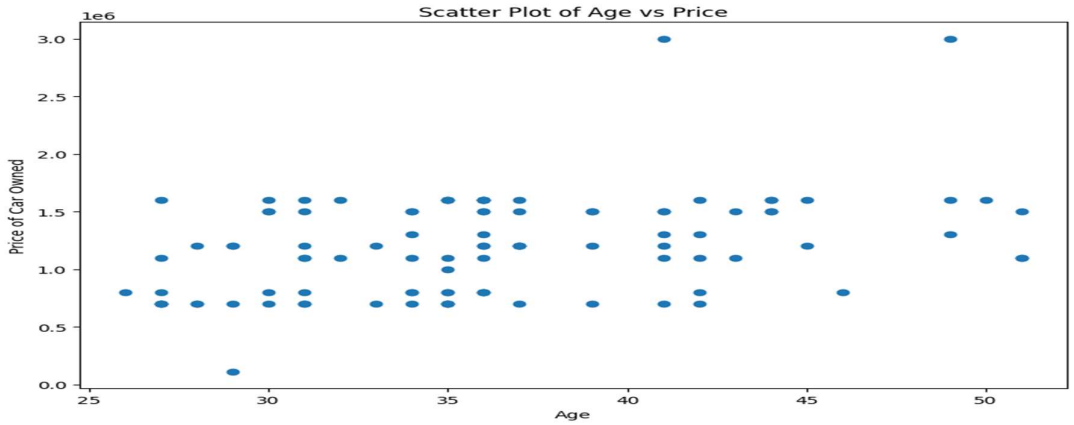

Count of Car Makes by Age
Age
30
35
40
45
50
Count
Baleno
Ciaz
City
Creata
Duster
Luxuray
SUV
Verna
i20
Make of Car

Pair Plot :


Age
No of Dependents
Salary
Wife Salary
Total Salary
Price
1e6


Violin Plot
Violin Plot
Violin Plot
Violin Plot
Violin Plot
Violin Plot
Make
i20
Ciaz
Duster
City
SUV
Baleno
Verna
Luxuray
Creata
Age
No of Dependents
Salary
1e6
Wife Salary
1e6
Total Salary
1e6
Price
1e6


Scatter Plot of Total Salary vs Price
1e6
Price of Car Owned
Total Salary
1e6



Label Encoding :

```
from sklearn.preprocessing import LabelEncoder

cols = ['Profession', 'Marrital Status', 'Education', 'Personal loan','House Loan','Wife Working']
#
# Encode labels of multiple columns at once
#
data[cols] = data[cols].apply(LabelEncoder().fit_transform)
#
# Print head
#
data.head()
```

	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

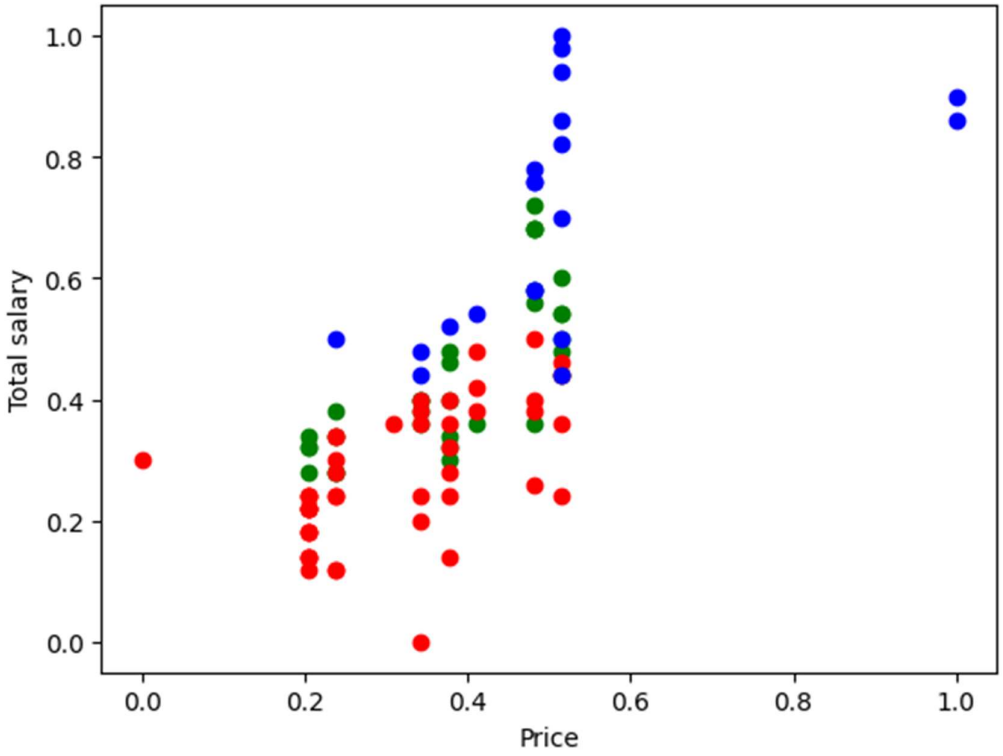
Scaling the Data :

```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

# Example: Assuming 'Make' is a categorical feature
# You should encode categorical variables appropriately before clustering
# For simplicity, let's assume 'Make' is removed or encoded before this step.

# Instantiate KMeans clustering with desired parameters
clustering_kmeans = KMeans(n_clusters=2)

# Fit the model and predict clusters
d['clusters'] = clustering_kmeans.fit_predict(d.drop('Make', axis=1))  # Assuming 'Make' is categorical and dropped or encoded

# Optionally, you can scale the data if necessary
# scaler = StandardScaler()
# d_scaled = scaler.fit_transform(d.drop('clusters', axis=1))  # Scale numerical features if needed

# Example usage with scaled data:
# clustering_kmeans.fit(d_scaled)
# d['clusters'] = clustering_kmeans.labels_

# Display or process the clustered data
print(d.head())
```

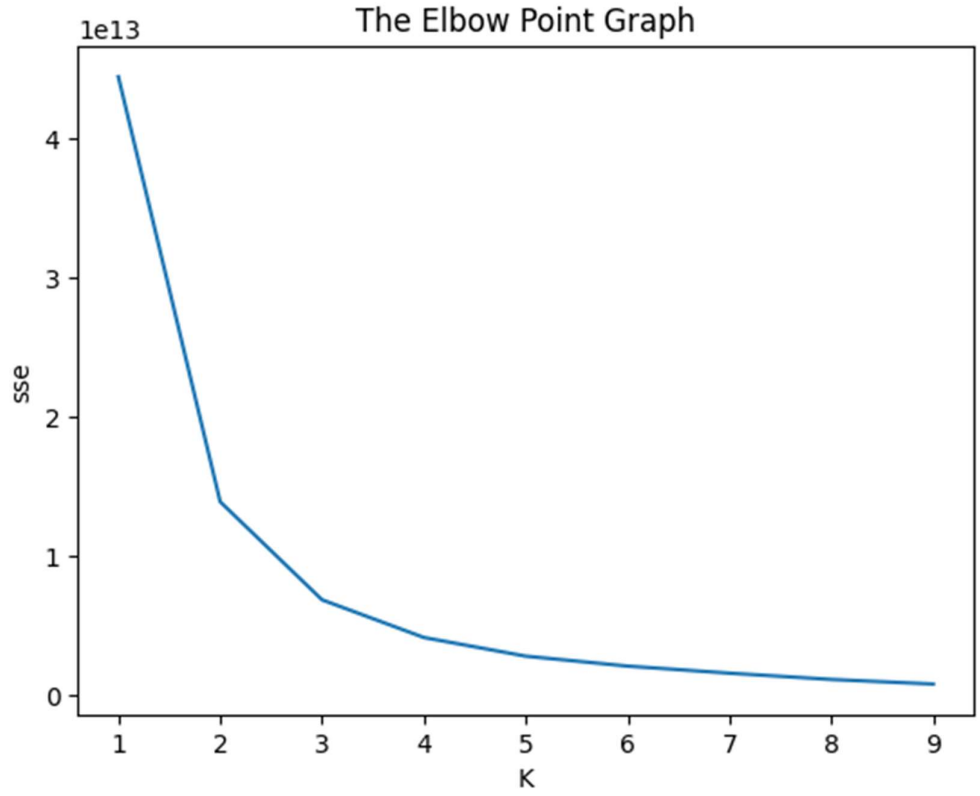
```
   Age  Profession  Marrital Status  Education  No of Dependents  \
0   27           1                1          1                 0
1   35           1                0          1                 2
2   45           0                0          0                 4
3   41           0               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

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

   Total Salary    Make    Price  clusters
0        800000     120   800000         0
1       2000000    Ciaz  1000000         0
2       1800000  Duster  1200000         0
3       2200000    City  1200000         0
4       2600000     SIV  1600000         0
```

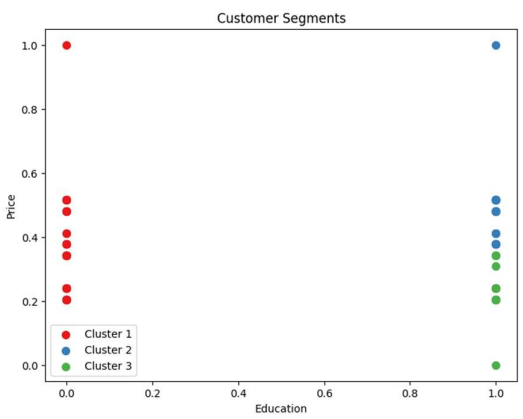
K- Means Clustering :

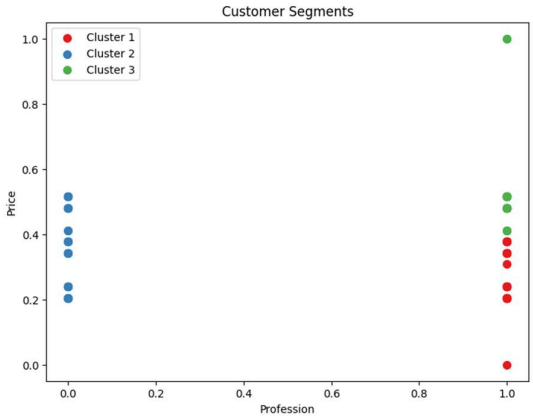
```
from sklearn.cluster import KMeans

# Instantiate KMeans clustering with desired parameters
clustering_kmeans = KMeans(n_clusters=2)

# Fit the model and predict clusters on numeric data
clusters = clustering_kmeans.fit_predict(numeric_data)

# Create a new DataFrame with cluster labels
d_with_clusters = d.copy()
d_with_clusters['clusters'] = clusters

# Optionally, print the updated DataFrame with cluster labels
print(d_with_clusters)
```

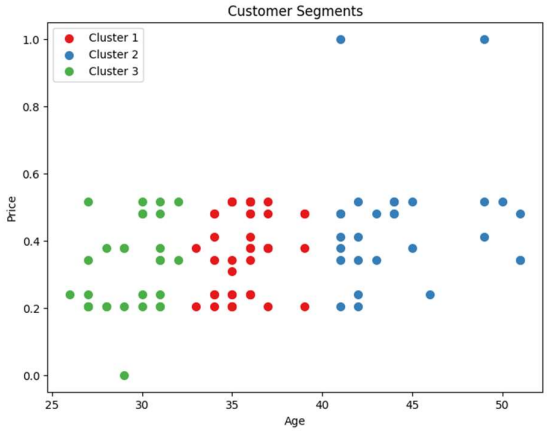


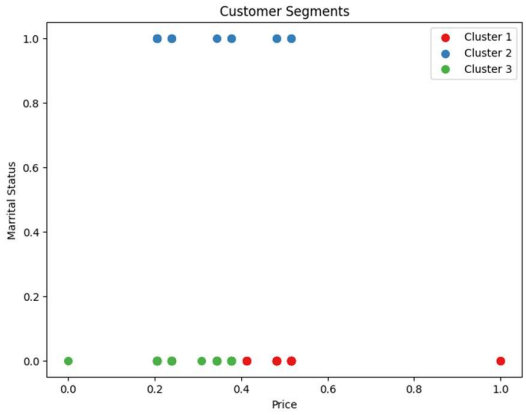
Elbow Method :

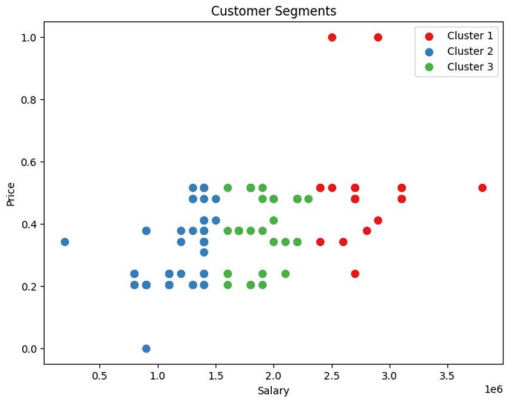


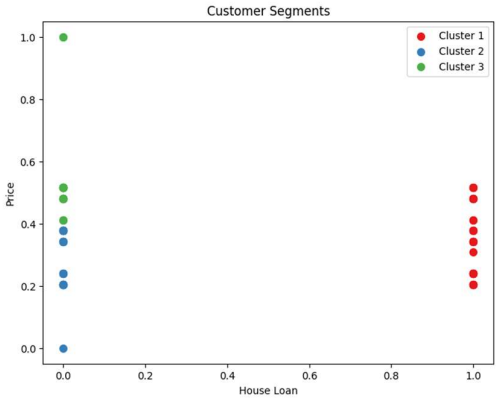

Customer Segments
Price
Education
Cluster 1
Cluster 2
Cluster 3


Customer Segments
Price
Profession
Cluster 1
Cluster 2
Cluster 3

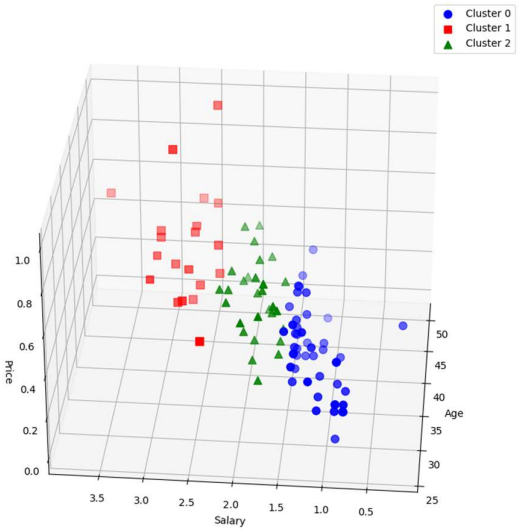

Customer Segments
Price
Age
Cluster 1
Cluster 2
Cluster 3

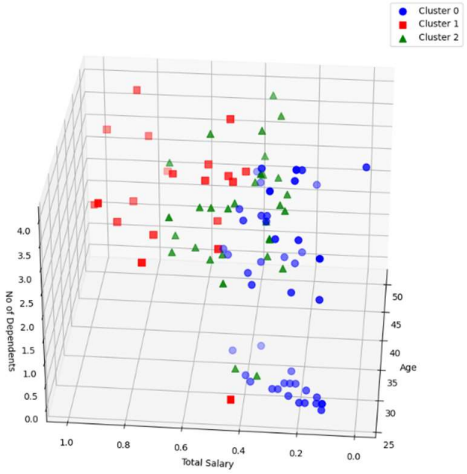

Customer Segments
Marrital Status
Price
Cluster 1
Cluster 2
Cluster 3


Customer Segments
Price
Salary
1e6
Cluster 1
Cluster 2
Cluster 3


Customer Segments
Price
House Loan
Cluster 1
Cluster 2
Cluster 3


Customer Segments
Price
Wife Salary
Cluster 1
Cluster 2
Cluster 3





Github Link : https://github.com/kotaAnusha1/market