Electric Vehicle Market

Market Segmentation

Abstract

Market segmentation becomes a vital strategy for emerging markets to investigate and execute for widespread adoption of emerging mobility technologies like electric vehicles (EVs). As a low-emission and low-operating-cost vehicle, EV adoption is anticipated to increase drastically in the near future. As a result, it will stimulate a significant amount of future academic study interest. By utilising an integrated research framework of "perceived benefits-attitude-intention," the primary goal of this study is to explore and identify several sets of possible buyer categories for EVs based on psychographic, behavioural, and socioeconomic characterisation.

Data Collection

The data has been collected manually, and the sources used for this process are listed below:

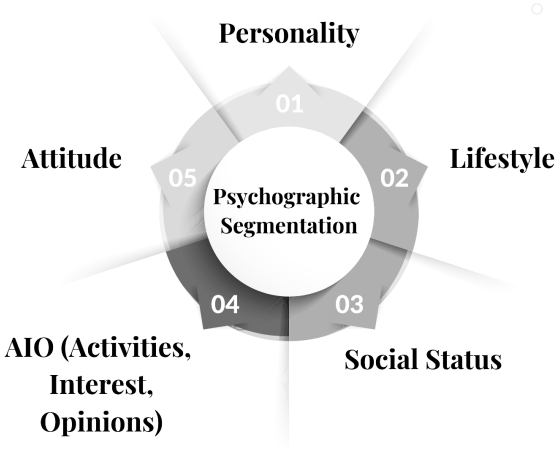
- https://www.kaggle.com/
- https://data.world/

Market Segmentation

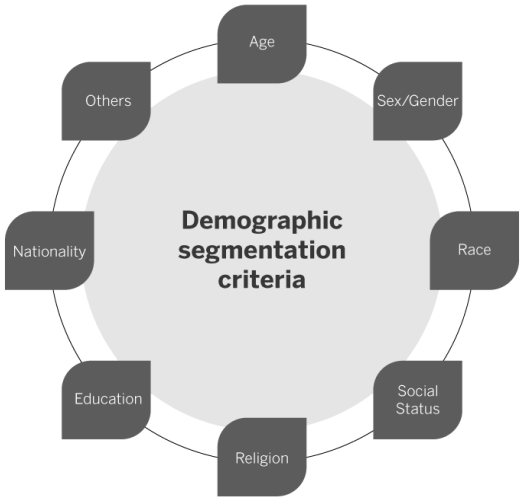
Target Market: The target market of Electric Vehicle Market Segmentation can be categorised into Geographic, SocioDemographic, Behavioral, and Psychographic Segmentation.

Behavioural Segmentation: searches directly for similarities in behaviour or reported behaviour. Example: prior experience with the product, the amount spent on the purchase, etc.

Psychographic Segmentation: Psychographic information identifies a person's preferences, way of life, values, hobbies, and personality in order to identify their pain spots and potential ways to entice them to make a purchase through a conversion funnel.



Demographic Segmentation: A market segmentation technique based on factors like age, gender, income, etc. is known as demographic segmentation. Organisations can perform better by better understanding consumer behaviour thanks to segmentation.
Age, sex, gender, religion, and level of education are all important demographic factors in the study. Businesses need to keep up with this always-shifting market whether they are launching a new product, making modifications, or putting in place new services.



Segmenting for Electric Vehicle Market

Increased demand for fuel-efficient, high-performance, and low-emission vehicles, strict government pollution restrictions, falling prices for electric vehicle batteries, and rising gasoline prices all contribute to the expansion of the electric vehicle market. Additionally, it is anticipated that constraints such as a lack of infrastructure for charging, high production costs, range anxiety, and serviceability will restrain the growth of the EV industry. Additionally, the leading players in the electric vehicle market should benefit greatly from aspects including technological growth, proactive government initiatives, and the development of self-driving electric car technology. On the basis of type, vehicle type, vehicle class, top speed, vehicle drive type, and region, the global market for electric vehicles is divided into segments. Battery electric vehicles (BEV), plug-in hybrid electric vehicles (PHEV), and fuel cell electric vehicles (FCEV) are the three categories by kind. Two-wheelers, passenger cars, and commercial vehicles are divided based on the kind of vehicle. Mid-priced and luxury class vehicles are divided according to vehicle class.

Implementation

Packages/Tools used:

- **Pandas:** To load datasets
- **Numpy:** For various mathematical calculations in an array
- **Scikit-learn:** An open source data analysis library, and the gold standard for Machine Learning (ML)
- **Plotly, Matplotlib, Seaborn:** Visualisation libraries for EDA
- **Statsmodel:** For statistical analysis

Data Preprocessing

Data preprocessing, which is a crucial phase in the data mining process, can be defined as the altering or dropping of data before to usage in order to ensure or increase performance.

Data Cleaning

Data cleaning is an essential step in the data preprocessing phase, which involves identifying and correcting or removing errors, inconsistencies, and missing values in the dataset. Here are some common data cleaning tasks:

1. Handling Missing Values
2. Handling Outliers
3. Standardizing or Normalizing Data
4. Correcting Data Types

Our data was free of missing values, duplicates, outliers or typos.

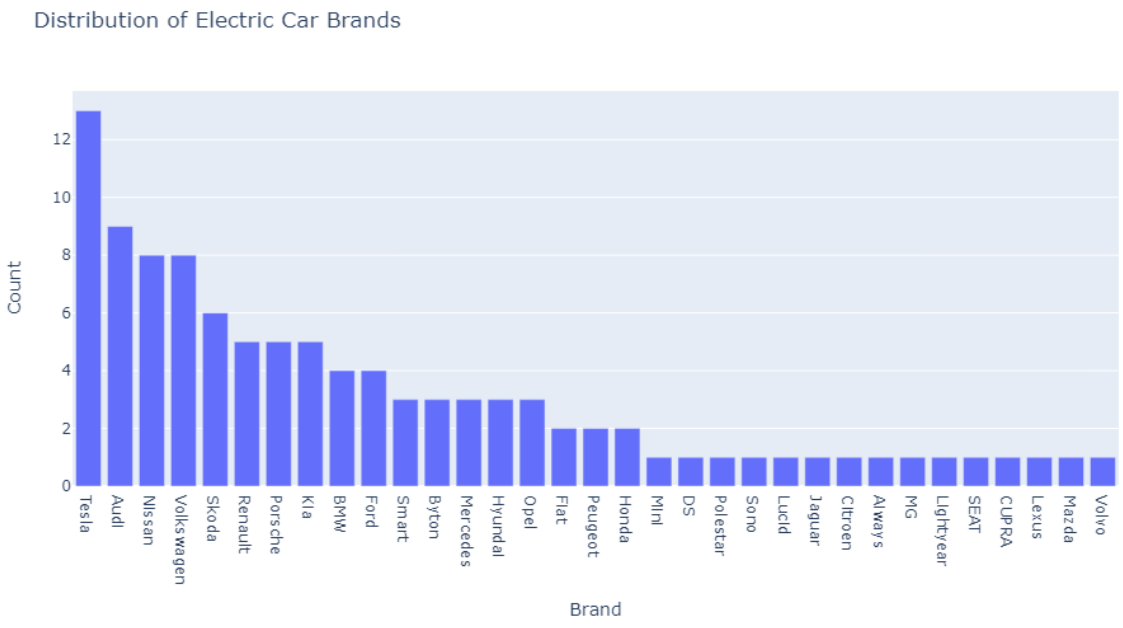
Brand	Model	AccelSec	TopSpeed_KmH	Range_Km	Efficiency_WhKm	FastCharge_KmH	RapidCharge	PowerTrain	PlugType	BodyStyle	Segment	Seats
Tesla	Model 3 Long Range Dual Motor	4.6	233	450	161	940	Yes	AWD	Type 2 CCS	Sedan	D	5
Volkswagen	ID.3 Pure	10.0	160	270	167	250	No	RWD	Type 2 CCS	Hatchback	C	5
Polestar	2	4.7	210	400	181	620	Yes	AWD	Type 2 CCS	Liftback	D	5
BMW	iX3	6.8	180	360	206	560	Yes	RWD	Type 2 CCS	SUV	D	5
Honda	e	9.5	145	170	168	190	Yes	RWD	Type 2 CCS	Hatchback	B	4

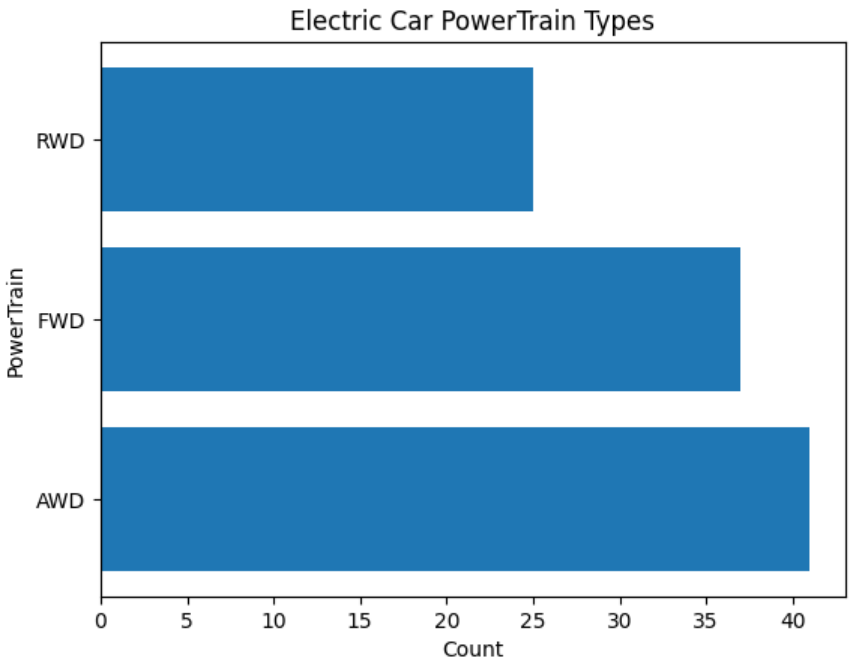
Sample of our dataset

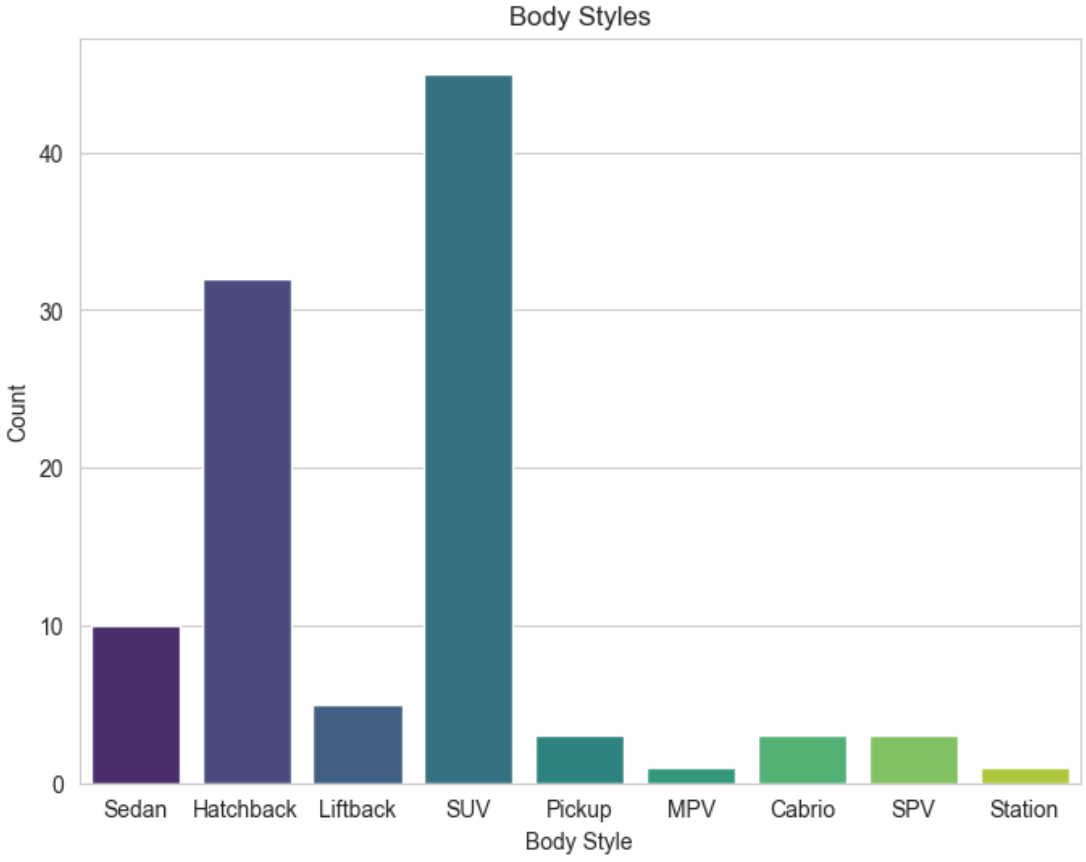
Exploratory Data Analysis

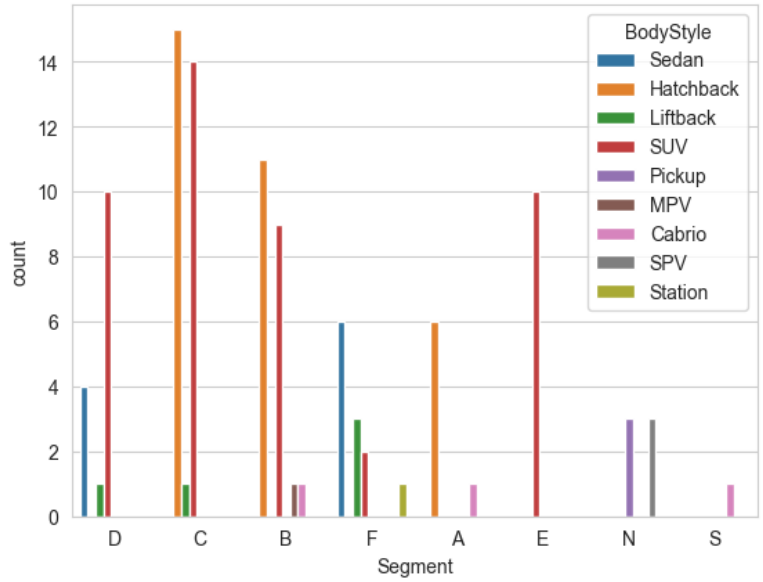
Exploratory Data Analysis (EDA) is a crucial step in the data analysis process. It involves examining and summarizing the main characteristics, patterns, and relationships in the dataset using various statistical and visualization techniques. EDA helps in understanding the data, uncovering insights, identifying outliers, and formulating hypotheses for further analysis.

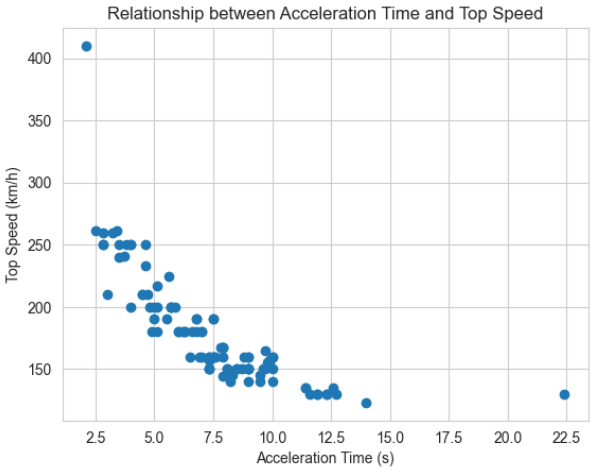
Some of the EDA's are shown below.

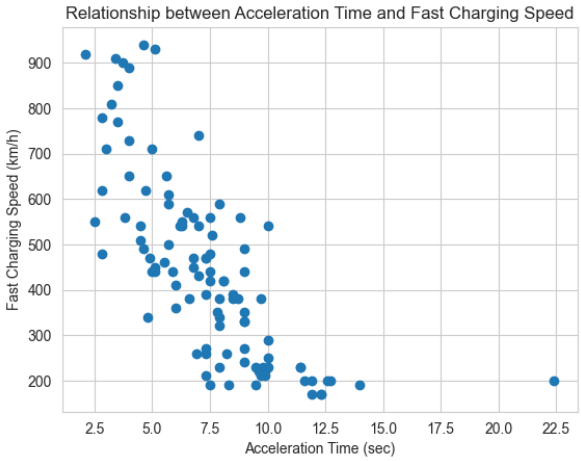


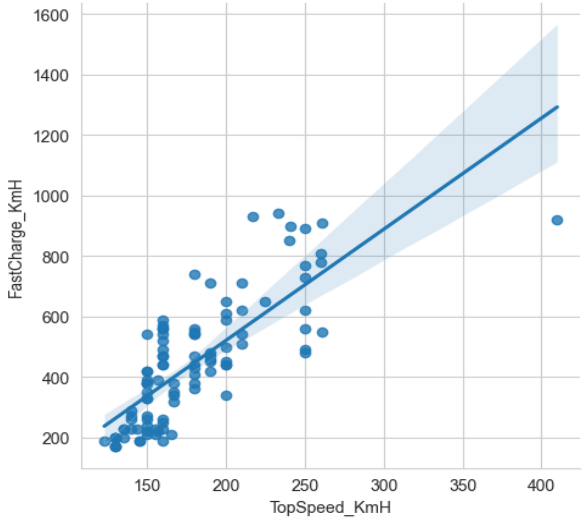

Electric Car PowerTrain Types
RWD
FWD
AWD
PowerTrain
0
5
10
15
20
25
30
35
40
Count

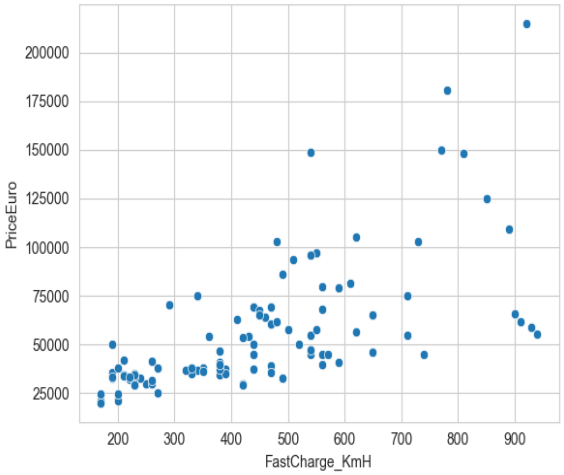

Body Styles
40
30
20
10
0
Count
Sedan
Hatchback
Liftback
SUV
Pickup
MPV
Cabrio
SPV
Station
Body Style

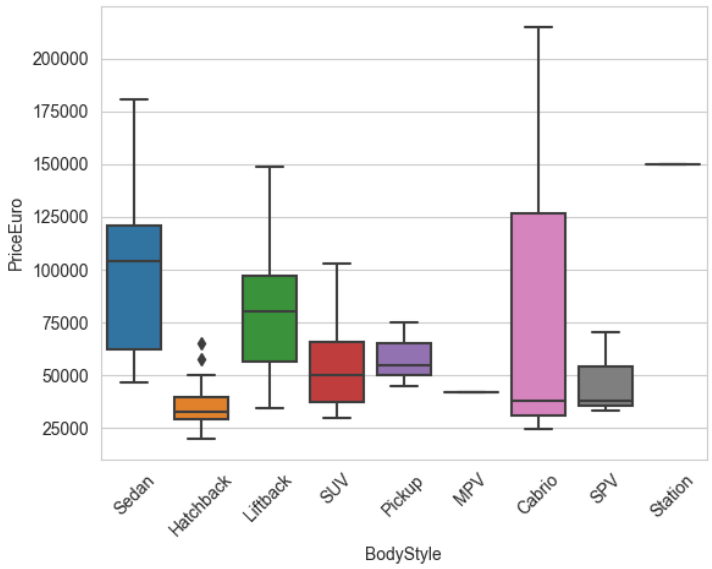

BodyStyle
Sedan
Hatchback
Liftback
SUV
Pickup
MPV
Cabrio
SPV
Station
count
Segment
D
C
B
F
A
E
N
S

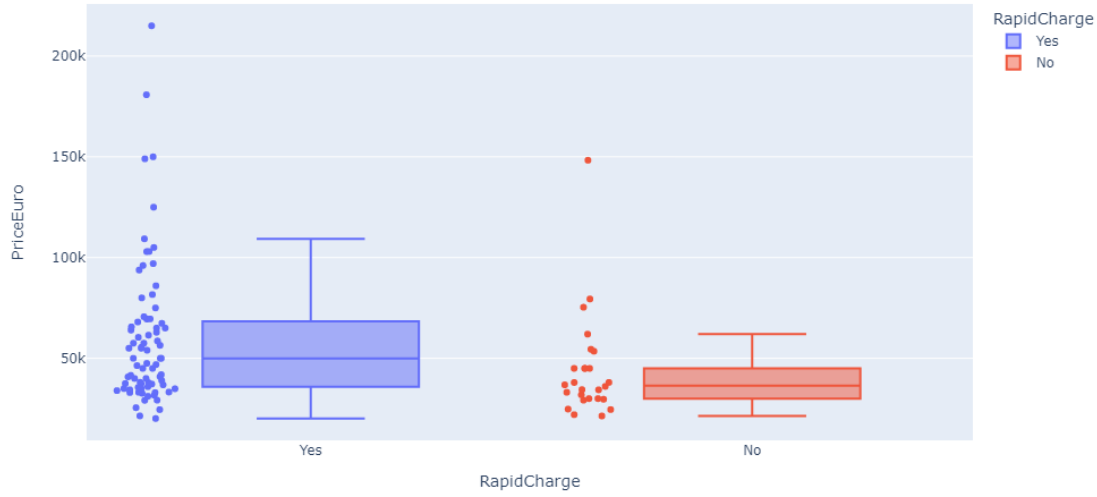

Relationship between Acceleration Time and Top Speed
Top Speed (km/h)
Acceleration Time (s)

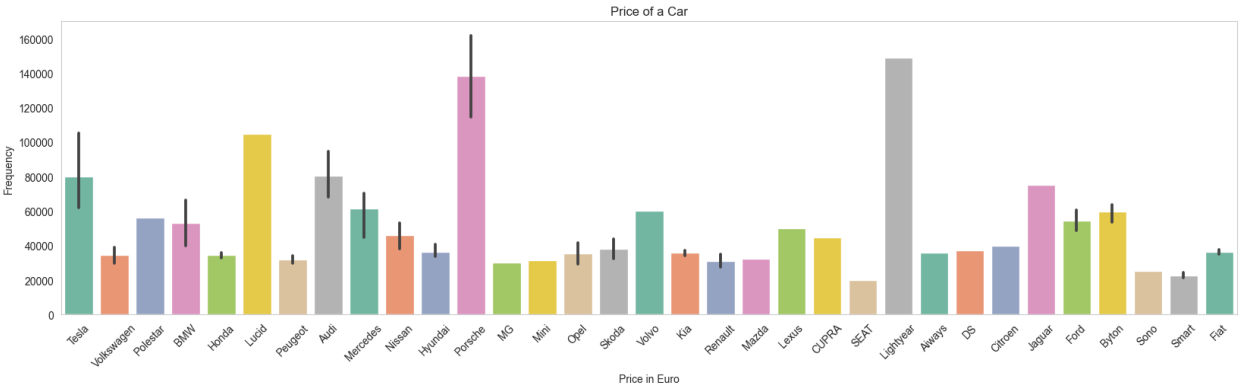

Relationship between Acceleration Time and Fast Charging Speed
Fast Charging Speed (km/h)
Acceleration Time (sec)


FastCharge_KmH
TopSpeed_KmH

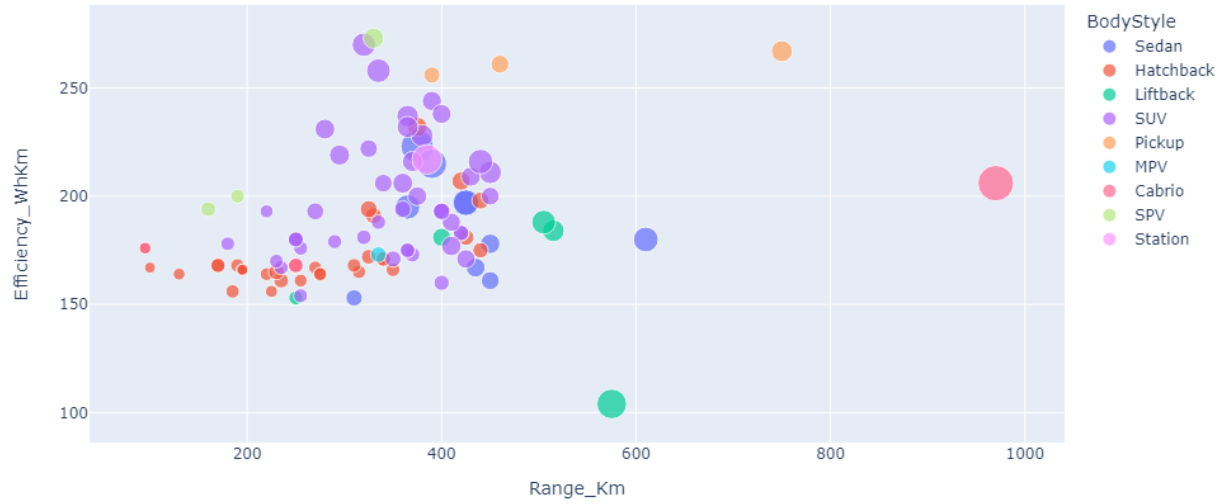

PriceEuro
FastCharge_KmH


PriceEuro
200000
175000
150000
125000
100000
75000
50000
25000
Sedan
Hatchback
Liftback
SUV
Pickup
MPV
Cabrio
SPV
Station
BodyStyle

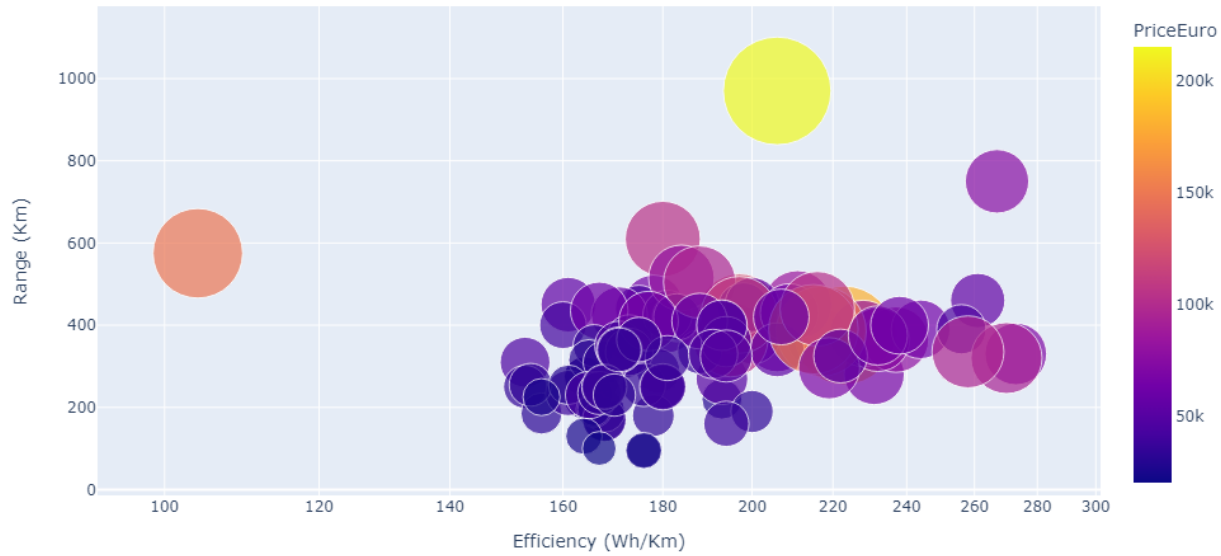

RapidCharge
Yes
No
PriceEuro
200k
150k
100k
50k
Yes
No
RapidCharge


Price of a Car
Frequency
160000
140000
120000
100000
80000
60000
40000
20000
0
Tesla
Volkswagen
Polestar
BMW
Honda
Lucid
Peugeot
Audi
Mercedes
Nissan
Hyundai
Porsche
MG
Mini
Opel
Skoda
Volvo
Kia
Renault
Mazda
Lexus
CUPRA
SEAT
Lightyear
Aiways
DS
Citroen
Jaguar
Ford
Byton
Sono
Smart
Fiat
Price in Euro

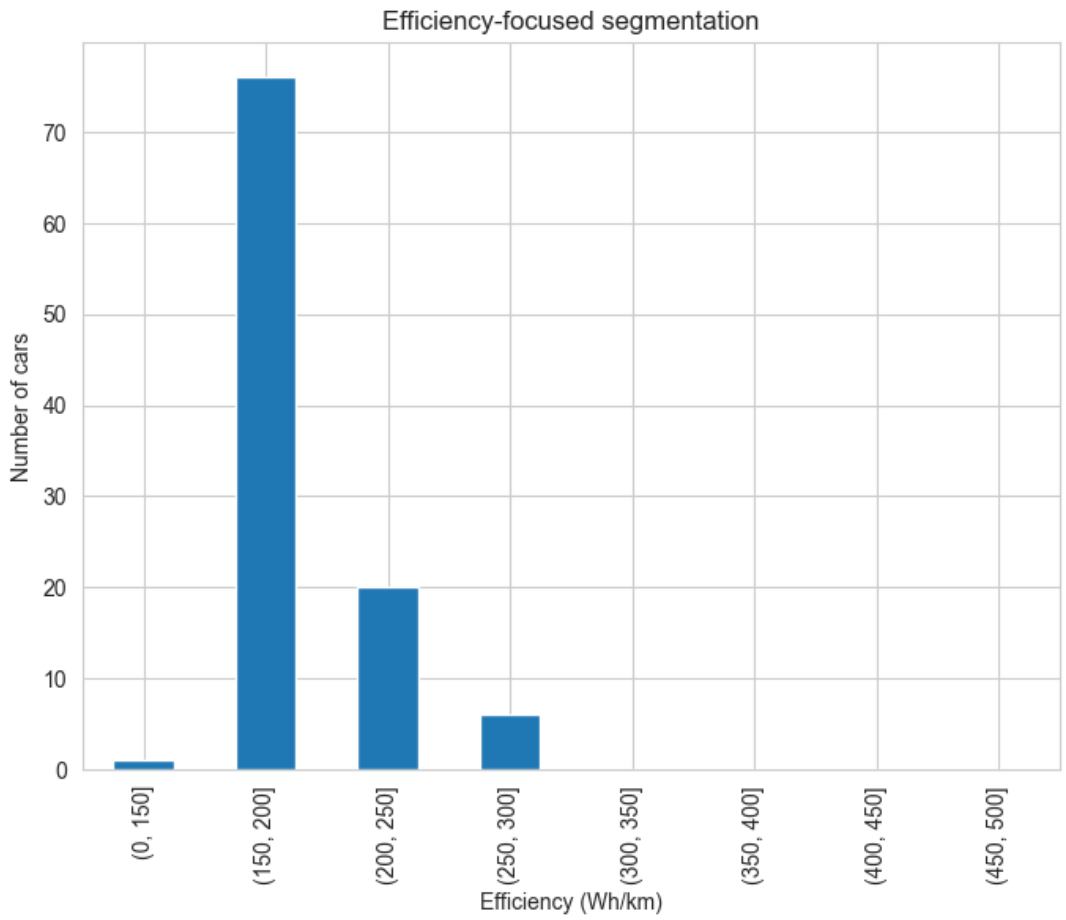
Range vs Efficiency by Body Style and Price

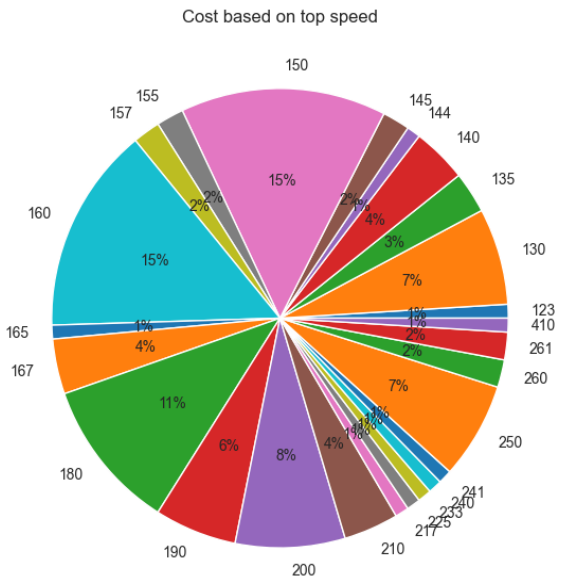


Electric Cars: Range vs Efficiency by Price Group



Vehicle Segmentation





Linear Regression

Linear regression is a popular statistical modeling technique used to understand the relationship between a dependent variable (target variable) and one or more independent variables (predictor variables). It assumes a linear relationship between the variables, where the dependent variable can be modeled as a linear combination of the independent variables.

```
x=df1[['AccelSec','Range_Km','TopSpeed_KmH','Efficiency_WhKm', 'RapidCharge','PowerTrain']]
y=df1['PriceEuro']
```

Here x is the independent variable and y the dependent variable or the target variable.

```
# Train the logistic regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

```
LinearRegression()
```

```
# Make predictions on the test data
y_pred = model.predict(X_test)
```

```
r2=(r2_score(y_test,y_pred))
print(r2*100)
```

```
78.7707238023266
```

Our model gave as an r2 value of 78.77, which is pretty good value.

Linear regression using OLS method

OLS Regression Results

Dep. Variable:	PriceEuro	**R-squared:**	0.721
Model:	OLS	**Adj. R-squared:**	0.704
Method:	Least Squares	**F-statistic:**	41.36
Date:	Mon, 15 May 2023	**Prob (F-statistic):**	1.57e-24
Time:	13:37:13	**Log-Likelihood:**	-1155.0
No. Observations:	103	**AIC:**	2324.
Df Residuals:	96	**BIC:**	2342.
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-9.263e+04	2.53e+04	-3.659	0.000	-1.43e+05	-4.24e+04
AccelSec	1753.0004	1048.759	1.672	0.098	-328.769	3834.770
Range_Km	36.3000	22.629	1.604	0.112	-8.618	81.218
TopSpeed_KmH	581.7484	80.158	7.257	0.000	422.636	740.861
Efficiency_WhKm	117.6685	70.307	1.674	0.097	-21.890	257.227
RapidCharge	1465.5687	4496.958	0.326	0.745	-7460.822	1.04e+04
PowerTrain	-5235.8309	2956.235	-1.771	0.080	-1.11e+04	632.248

Omnibus:	84.867	Durbin-Watson:	2.060
Prob(Omnibus):	0.000	Jarque-Bera (JB):	741.645
Skew:	2.644	Prob(JB):	8.99e-162
Kurtosis:	15.036	Cond. No.	6.15e+03

Decision Tree

Decision Tree is a popular algorithm for regression problems that involves building a tree-like model to predict the target variable.

```
from sklearn.tree import DecisionTreeRegressor
```

```
# Create a decision tree regressor
regressor = DecisionTreeRegressor(random_state=42)
```

```
# Train the regressor on the training set
regressor.fit(X_train, y_train)
```

```
DecisionTreeRegressor(random_state=42)
```

```
# Make predictions on the testing set
y_pred = regressor.predict(X_test)
```

```
from sklearn.metrics import mean_squared_error
# Evaluate the performance of the regressor
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

```
print('Mean Squared Error:', mse)
print('R-squared Value:', r2)
```

```
Mean Squared Error: 543589596.032258
R-squared Value: 0.6733264779314232
```

Comparing both linear regression and decision tree, lm gave us good r2 value.

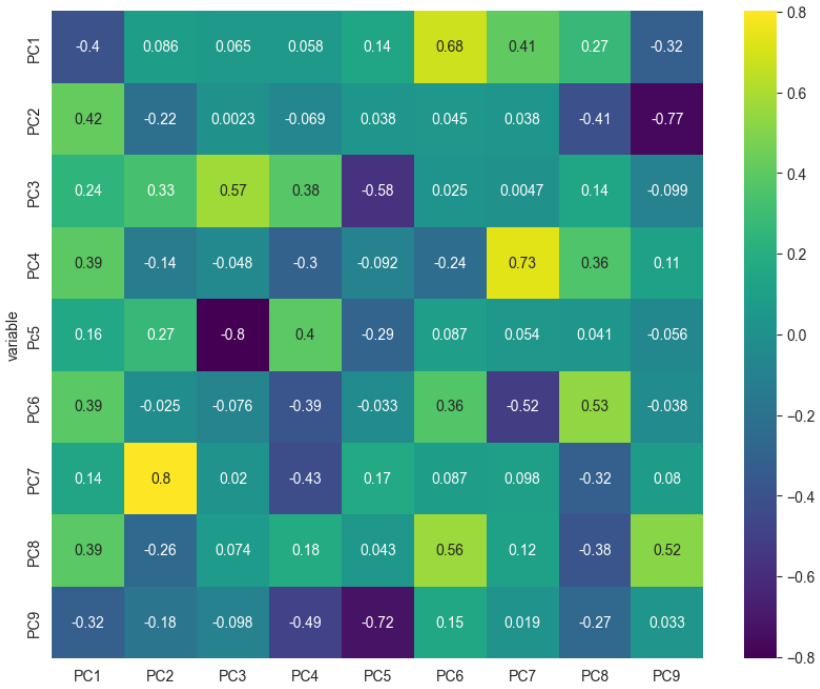
Principle Component Analysis

```
features = ['AccelSec','TopSpeed_KmH','Efficiency_WhKm','FastCharge_KmH', 'RapidCharge',
            'Range_Km', 'Seats', 'PriceEuro','PowerTrain']
# Separating out the features
x = df1.loc[:, features].values
x = StandardScaler().fit_transform(x)
```

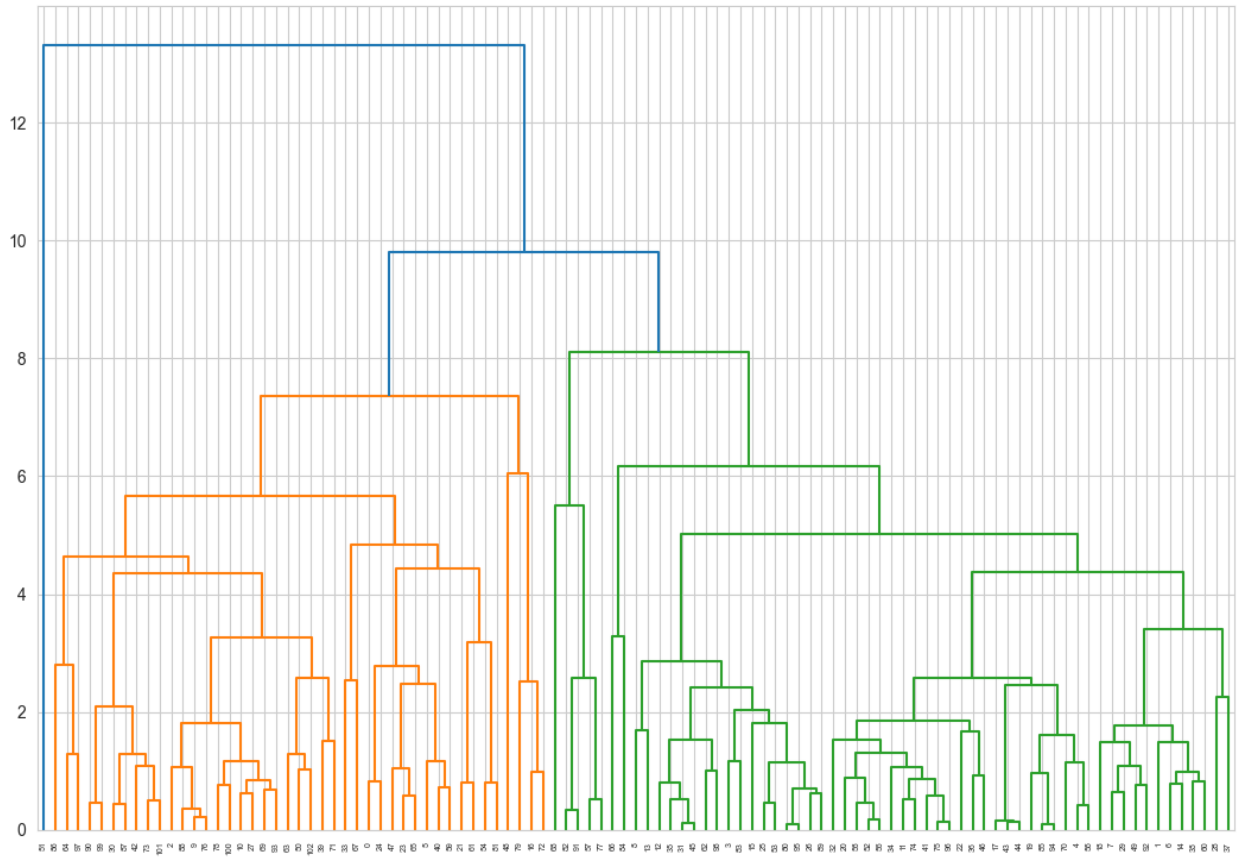
```
pca = PCA(n_components=9)
t = pca.fit_transform(x)
data2 = pd.DataFrame(t, columns=['PC1', 'PC2','PC3','PC4','Pc5','PC6', 'PC7', 'PC8','PC9'])
data2
```

The features taken for PCA were:
'AccelSec','TopSpeed_KmH','Efficiency_WhKm','FastCharge_KmH', 'RapidCharge',
'Range_Km', 'Seats', 'PriceEuro','PowerTrain'

	PC1	PC2	PC3	PC4	Pc5	PC6	PC7	PC8	PC9
0	2.429225	-0.554599	-1.147772	-0.882791	0.839988	-0.959297	0.998880	0.711148	-0.396662
1	-2.322483	-0.345449	0.896473	-1.305529	0.079598	0.235116	-0.213678	-0.544135	-0.181867
2	1.587851	0.008899	-0.650523	0.041024	0.593537	-0.698248	0.058718	0.248837	-0.202775
3	0.291018	-0.000150	-0.307702	-0.514196	-1.608861	0.291624	0.364999	-0.235543	0.261663
4	-2.602679	-0.626489	-0.888088	0.585294	-0.802108	0.027387	-0.084955	-0.507790	-0.049904
...	...	...	...	...	...	...	...	...	...
98	-0.297170	0.446713	-0.463601	0.102542	-0.346005	-0.100457	0.031080	0.202253	0.145390
99	2.335018	0.630747	0.985883	1.560112	-0.817327	-0.121906	0.164115	-0.255651	0.141023
100	0.780642	0.426821	-0.298636	0.708598	0.481728	-0.540071	-0.139753	-0.048733	-0.367509
101	1.540920	0.698754	0.422384	1.094921	-0.298113	-0.307992	-0.363230	0.127251	-0.190397
102	0.915051	0.261495	2.410642	0.188002	0.340820	0.015609	-0.171875	0.567633	-0.200822

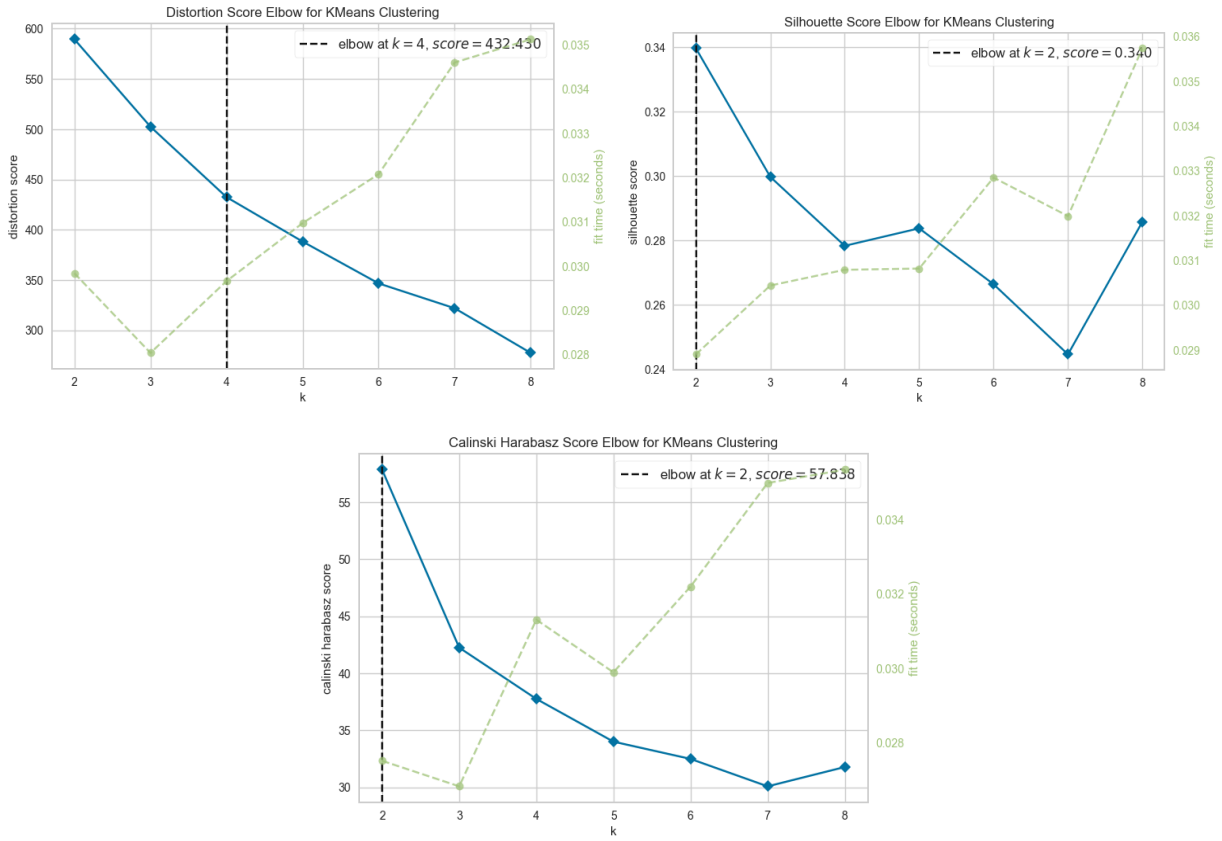


Dendrogram: Hierarchical classification method



A dendrogram is a hierarchical representation of data that is commonly used in clustering analysis. It visually represents the relationships between different data points or clusters in a hierarchical manner.

Elbow for K-Means clustering



The elbow method is a heuristic technique used to determine the optimal number of clusters in K-means clustering. It involves plotting the within-cluster sum of squares (WCSS) against the number of clusters, and identifying the point where the decrease in WCSS begins to level off.

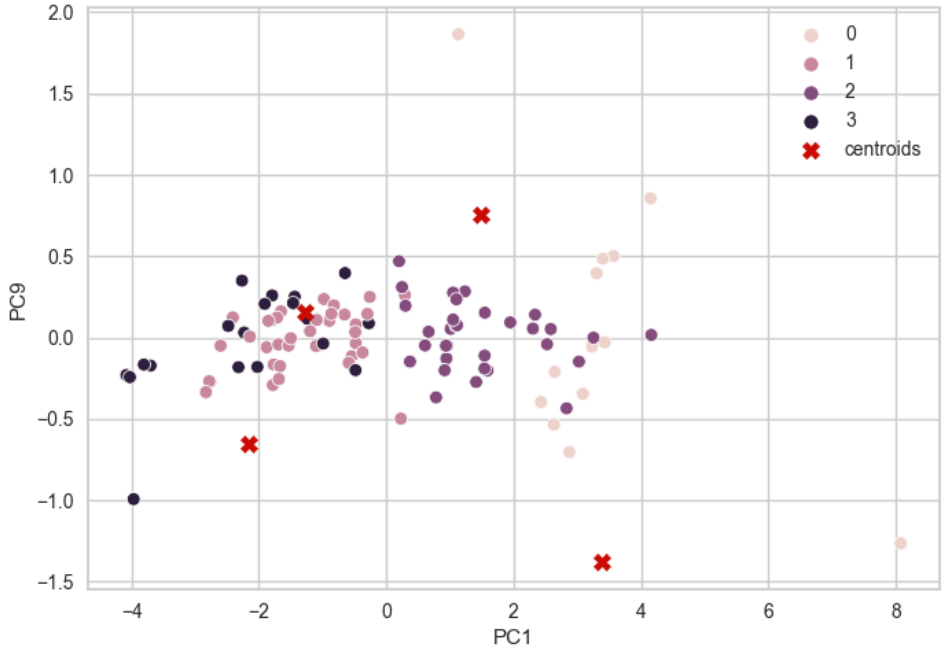
K-Means clustering

K-means clustering is a popular unsupervised machine learning algorithm used for clustering or grouping similar data points together. It aims to partition a dataset into K distinct clusters, where each data point belongs to the cluster with the nearest mean value.

1. Data Clustering: K-means clustering is primarily used for data clustering, where it can automatically group similar data points together based on their feature similarities. It is

useful for exploratory data analysis, customer segmentation, image recognition, document clustering, and more.

2. Algorithmic Approach: K-means clustering follows an iterative algorithmic approach. It starts by randomly initializing K cluster centroids and iteratively assigns data points to their nearest centroids, updates the centroids based on the assigned data points, and repeats this process until convergence.
3. Distance Metric: K-means clustering typically uses Euclidean distance as a distance metric to calculate the similarity between data points and centroids. However, other distance metrics can also be used depending on the nature of the data.
4. Choosing the Number of Clusters: One of the challenges in K-means clustering is determining the optimal number of clusters (K). This can be addressed using techniques like the elbow method, silhouette analysis, or domain knowledge.
5. Feature Scaling: It is recommended to scale the features before applying K-means clustering to ensure that all features contribute equally to the clustering process. This is particularly important when features have different scales or units.
6. Limitations: K-means clustering has some limitations. It assumes that clusters have a spherical shape and an equal number of points, which may not always hold true in real-world scenarios. It can also be sensitive to the initial choice of centroids, and outliers can significantly affect the clustering results.
7. Extensions and Variants: Several extensions and variants of K-means clustering exist, such as hierarchical K-means clustering, fuzzy K-means clustering, and K-means++ initialization. These variations address some of the limitations and offer different ways to handle specific data characteristics.



Regression with PCA

```
X=data2[['PC1', 'PC2','PC3','PC4','Pc5','PC6', 'PC7','PC8','PC9']]
y=df1['PriceEuro']
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

```
# Create a linear regression model object
model_2 = LinearRegression()
```

```
model_2.fit(x_train, y_train)
```

```
LinearRegression()
```

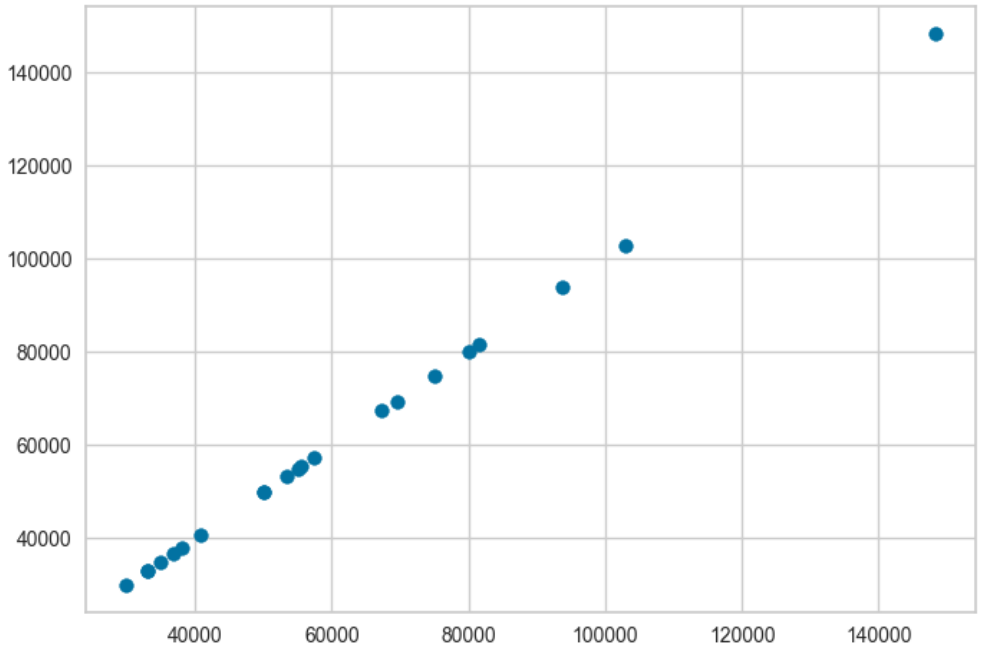
```
# Predict the output variable using the test set
y_pred = model_2.predict(x_test)
```

```
# Calculate R-squared value
r2 = r2_score(y_test, y_pred)
print("R-squared:", r2)
```

```
R-squared: 1.0
```

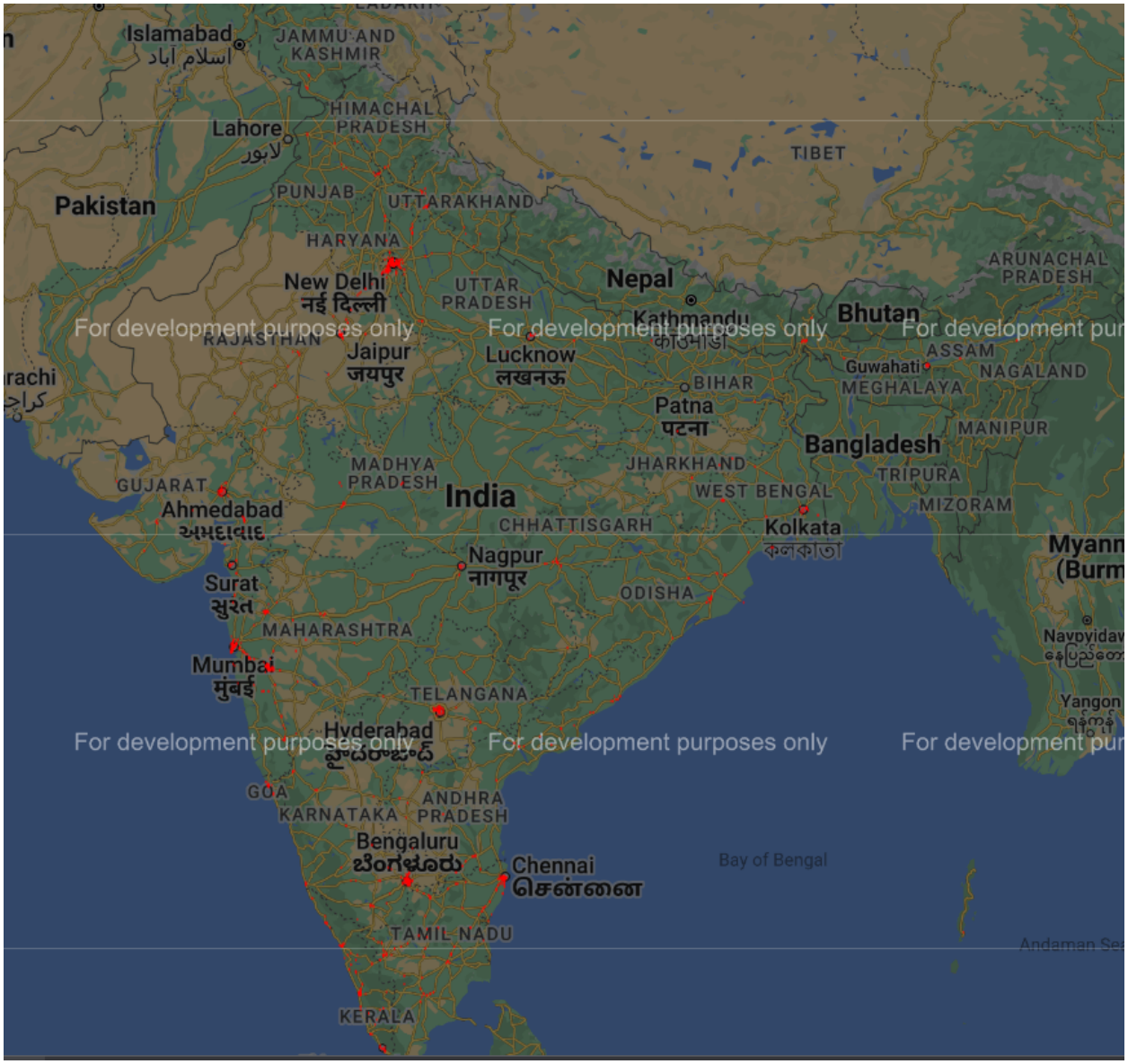
```
predictions=model_2.predict(x_test)
predictions
```

```
array([ 67358.,  55000.,  50000., 102945.,  81639.,  79990.,  93800.,
        36837.,  69484.,  55480.,  30000.,  38105.,  53500.,  34900.,
        57500.,  32997., 148301.,  75000.,  40795.,  33000.,  50000.])
```



EV Charging stations in India

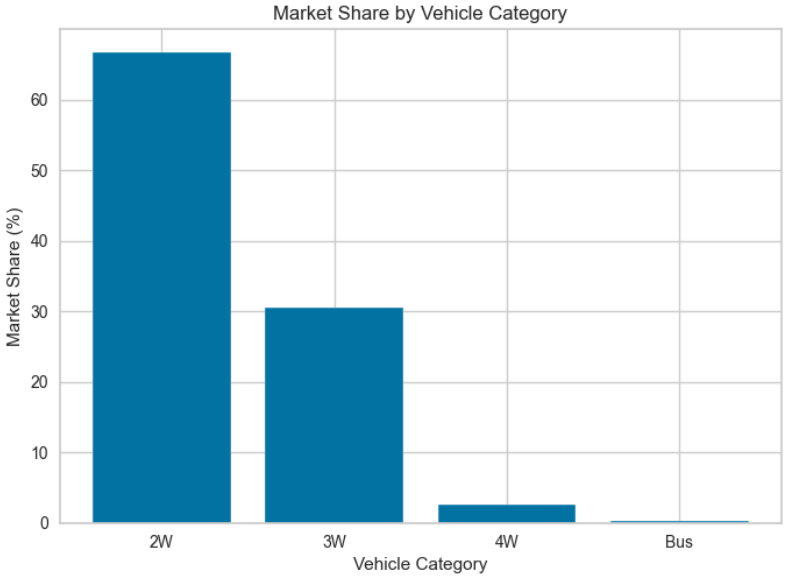
A graph has been plotted showing the EV charging stations in India. The locations were plotted using gmplot.

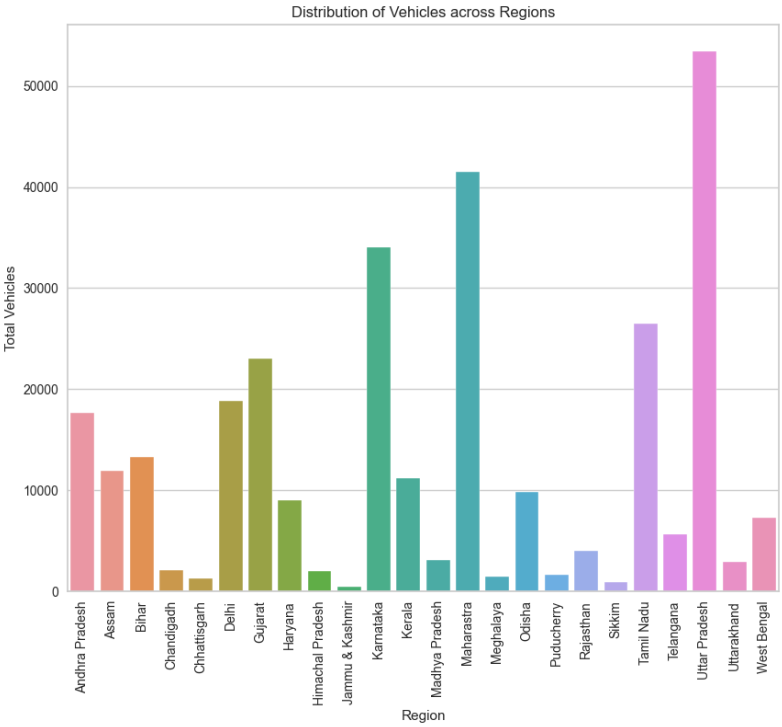


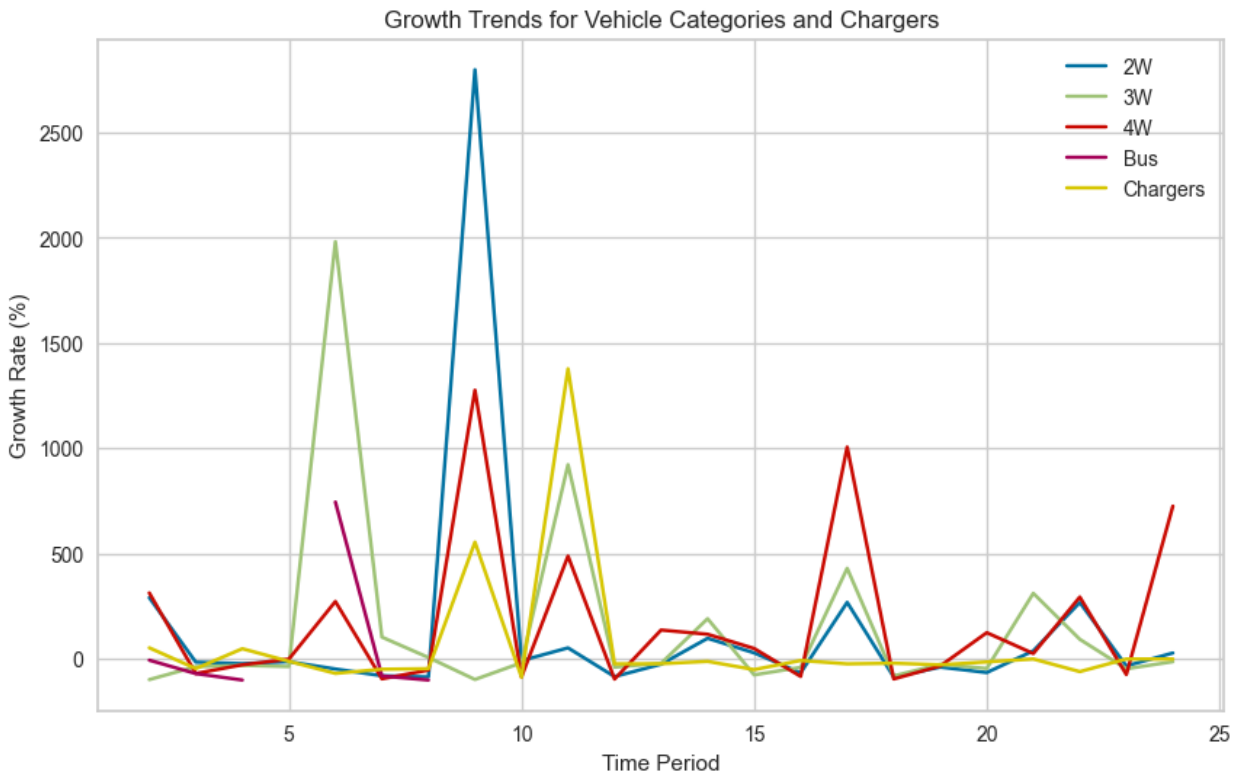
Gmplot is an interactive graph on which EV stations has been plotted. The red dots represent the location of the stations.

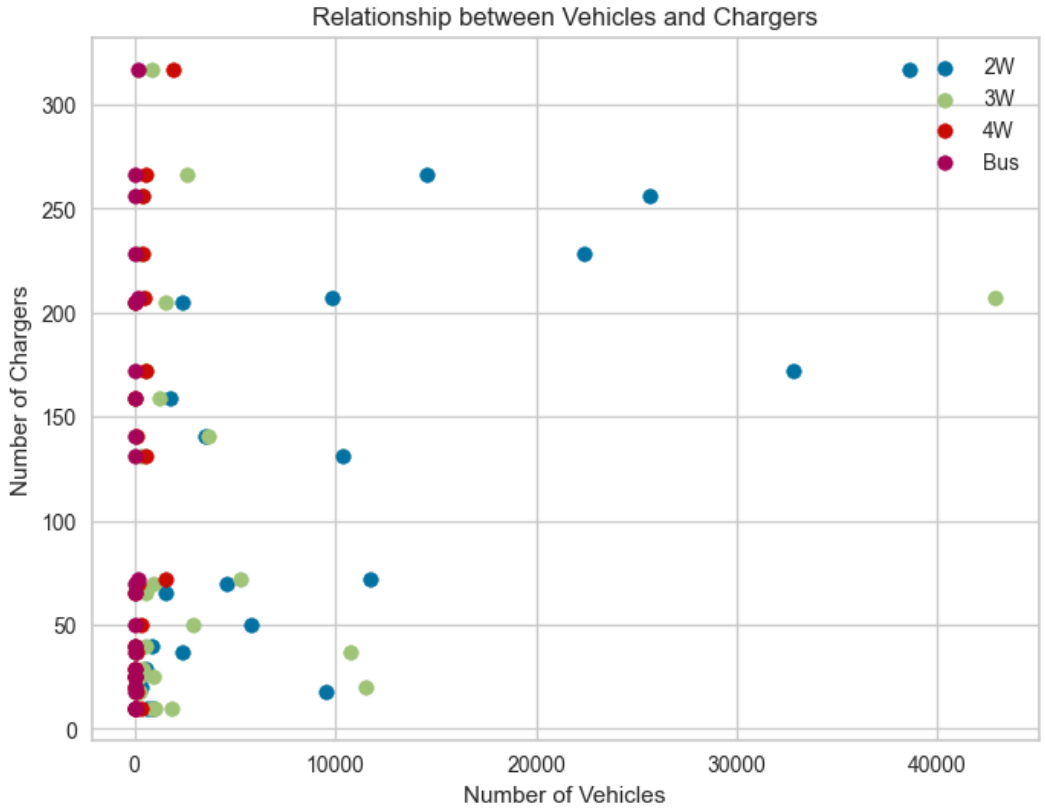
Moreover, an analysis of 2W,3W,4W,and E-buses market share has been analysed. EV in 23 states has been taken into account.

	Region	2W	3W	4W	Bus	Chargers
0	Uttar Pradesh	9852	42881	458	197	207
1	Maharastra	38558	893	1895	186	317
2	Karnataka	32844	568	589	57	172
3	Tamil Nadu	25642	396	426	0	256
4	Gujarat	22359	254	423	22	228






Growth Trends for Vehicle Categories and Chargers
2W
3W
4W
Bus
Chargers
Growth Rate (%)
2500
2000
1500
1000
500
0
5
10
15
20
25
Time Period


Relationship between Vehicles and Chargers
2W
3W
4W
Bus
Number of Chargers
300
250
200
150
100
50
0
0
10000
20000
30000
40000
Number of Vehicles