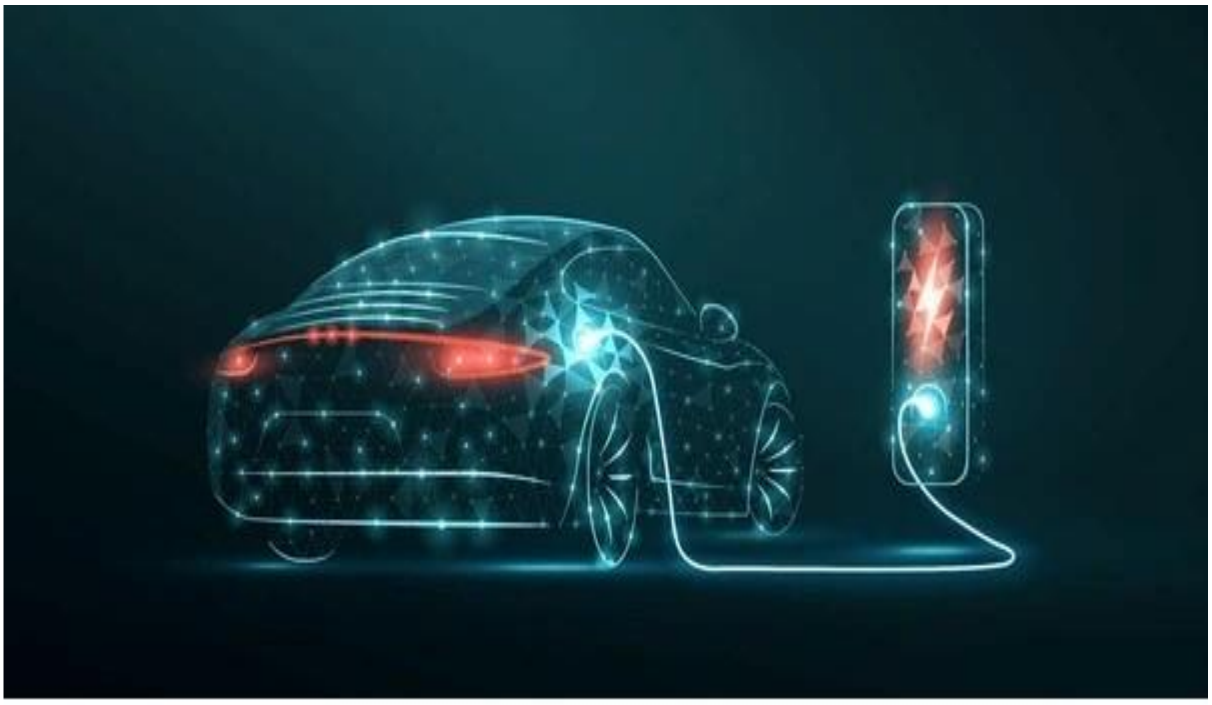
Dheeraj Nair

Electric Vehicle Market in India

Market Segment

https://github.com/Dheeraj1002/Fynn-labs-code-conversion



shutterstock.com · 2367123101

We will not stop until every car on the road is electric

-Elon Musk

Conceptual -

Market segmentation becomes a crucial tool for evolving transportation technology such as electric vehicles (EVs) in emerging markets to explore and implement for extensive adoption. EVs adoption is expected to grow phenomenally in near future as low emission and low operating cost vehicle, and thus, it drives a considerable amount of forthcoming academic research curiosity.

Market Analysis Using Geographic Datasets: Vehicles and Charging Stations

This analysis aims to understand market trends through geographic datasets that encompass the number of vehicles per state and the distribution of charging stations. By examining these data points, we can identify patterns in vehicle ownership, assess the availability and accessibility of charging infrastructure, and evaluate regional disparities in electric vehicle adoption.

Key Objectives:

1. **Vehicle Distribution Analysis**:
    - Assess the number of vehicles registered in each state to understand regional demand and preferences for different vehicle types, including electric and hybrid models.
    - Assess the total number of vehicles registered in each state, distinguishing between electric and non-electric vehicles. This will help in understanding regional demand and preferences for different vehicle types.
2. **Charging Station Mapping**:
    - Analize the location and density of charging stations across states to evaluate the adequacy of infrastructure supporting electric vehicle usage.
3. **Correlation Studies**:
    - Investigate the relationship between the number of vehicles and the availability of charging stations, identifying areas that may require infrastructure investment or incentives to promote electric vehicle adoption.

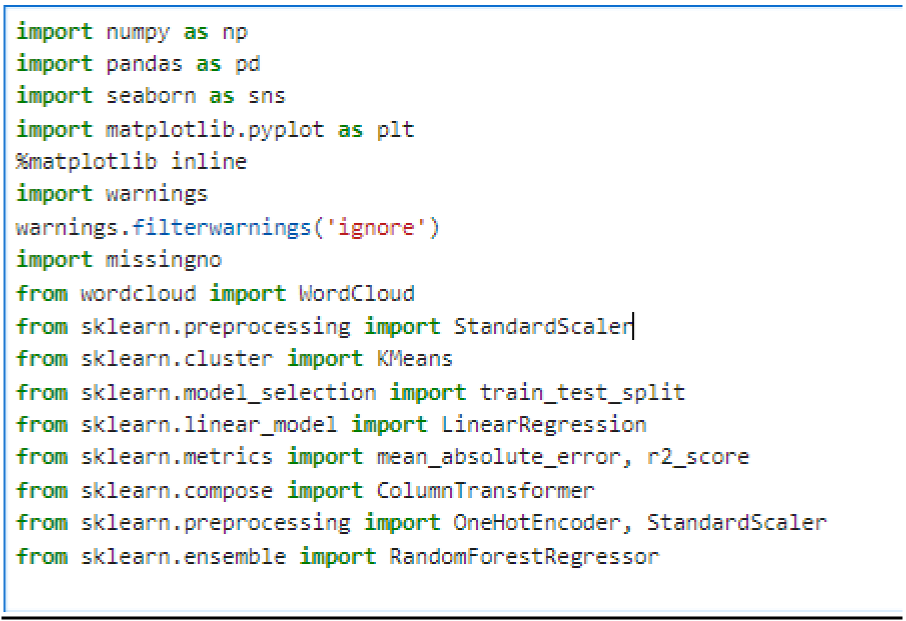
Data Collection

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

- https://www.kaggle.com/datasets
- https://data.gov.in/

Implementation

Packages/Tools used

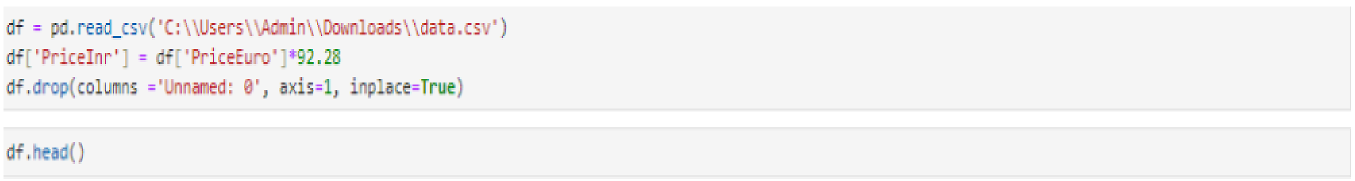


```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
import missingno
from wordcloud import WordCloud
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, r2_score
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.ensemble import RandomForestRegressor
```

- **NumPy**: A library for numerical computations in Python, providing support for arrays, matrices, and a collection of mathematical functions.
- **Pandas**: A data manipulation and analysis library, offering data structures like Data Frames for handling structured data.
- **Seaborn**: A statistical data visualization library built on top of Matplotlib, providing a high-level interface for drawing attractive statistical graphics.
- **Matplotlib**: A plotting library for Python, used for creating static, animated, and interactive visualizations in Python.
- **Warnings**: A module to manage warnings in Python. It's used to suppress warnings that may clutter the output.
- **Filter Warnings**: This line tells Python to ignore all warnings, which can be useful in notebooks or scripts where warnings are not relevant.
- **Missingno**: A library for visualizing missing data in Python, providing an easy way to visualize and understand data completeness.
- **WordCloud**: A library to generate word clouds from text data, visually representing word frequency.
- **StandardScaler**: A preprocessing tool from Scikit-learn that standardizes features by removing the mean and scaling to unit variance.

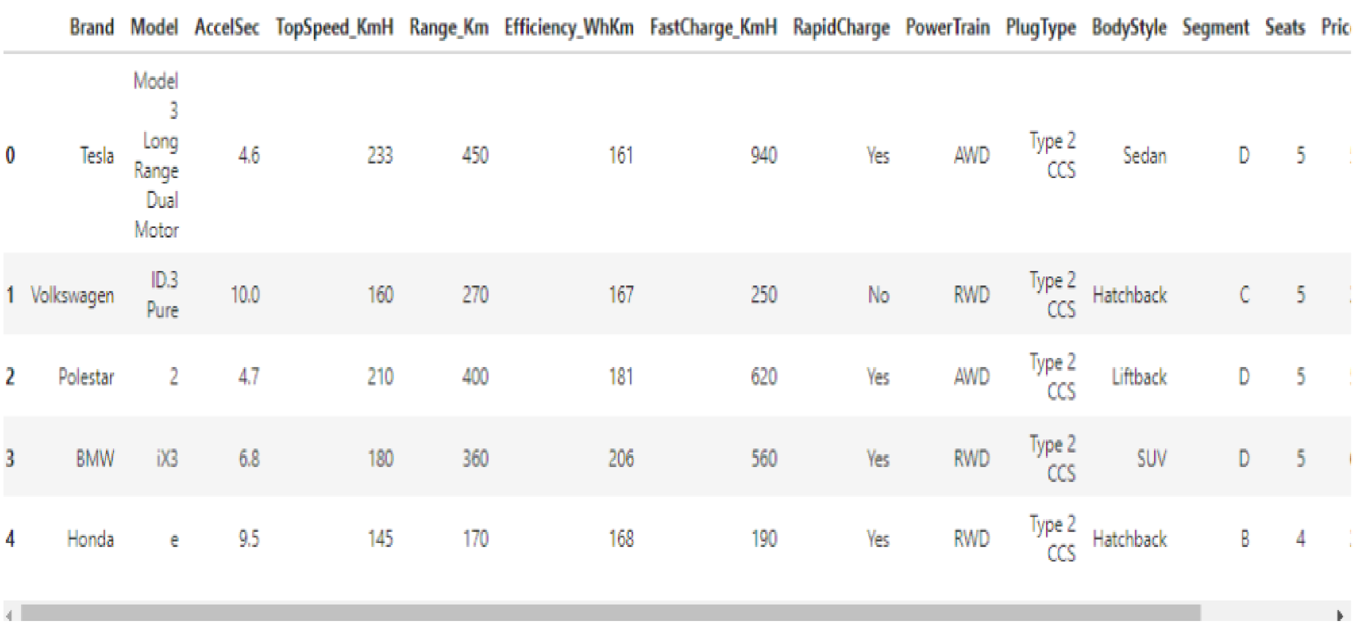
- **KMeans**: An unsupervised machine learning algorithm in Scikit-learn for clustering data into k distinct groups based on feature similarity.
- **train_test_split**: A utility function from Scikit-learn to split arrays or matrices into random train and test subsets.
- **LinearRegression**: A class in Scikit-learn for performing linear regression, modeling the relationship between dependent and independent variables.
- **mean_absolute_error**: A function to calculate the mean absolute error of a regression model.
- **r2_score**: A function to compute the coefficient of determination (R^2) regression score function
- **ColumnTransformer**: A class that allows you to apply different preprocessing steps to different columns of your dataset.
- **OneHotEncoder**: A preprocessing technique for converting categorical variables into a format that can be provided to ML algorithms.
- **RandomForestRegressor**: An ensemble learning method in Scikit-learn for regression that fits a number of decision tree regressors on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control overfitting.

Data-Preprocessing



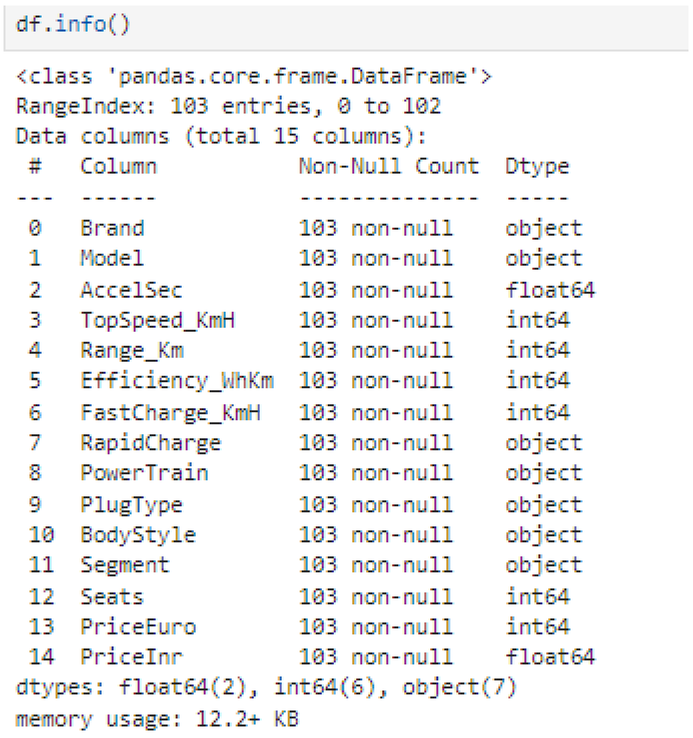
```
df = pd.read_csv('C:\\Users\\Admin\\Downloads\\data.csv')
df['PriceInr'] = df['PriceEuro']*92.28
df.drop(columns ='Unnamed: 0', axis=1, inplace=True)
```

```
df.head()
```



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

Added price Inr column which shows the Inr price of the vehicle



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103 entries, 0 to 102
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Brand            103 non-null    object 
 1   Model            103 non-null    object 
 2   AccelSec         103 non-null    float64
 3   TopSpeed_KmH     103 non-null    int64  
 4   Range_Km         103 non-null    int64  
 5   Efficiency_WhKm  103 non-null    int64  
 6   FastCharge_KmH   103 non-null    int64  
 7   RapidCharge      103 non-null    object 
 8   PowerTrain       103 non-null    object 
 9   PlugType         103 non-null    object 
 10  BodyStyle        103 non-null    object 
 11  Segment          103 non-null    object 
 12  Seats            103 non-null    int64  
 13  PriceEuro        103 non-null    int64  
 14  PriceInr         103 non-null    float64
dtypes: float64(2), int64(6), object(7)
memory usage: 12.2+ KB
```

>> Columns details

Brand, Model: Identifiers for the car manufacturer and the specific model.

AccelSec: Acceleration time from 0 to 100 km/h, a performance metric.

TopSpeed_KmH: The maximum speed the vehicle can achieve.

Range_Km: Distance the vehicle can cover on a full charge.

Efficiency_WhKm: Energy efficiency, measured in watt-hours per kilometer.

FastCharge_KmH: Charging speed in terms of kilometers per hour.

RapidCharge: A binary feature indicating whether rapid charging is supported.

PowerTrain, PlugType: Technical specifications related to the car's drivetrain and charging type.

BodyStyle, Segment: The car's design and market segment (e.g., sedan, SUV).

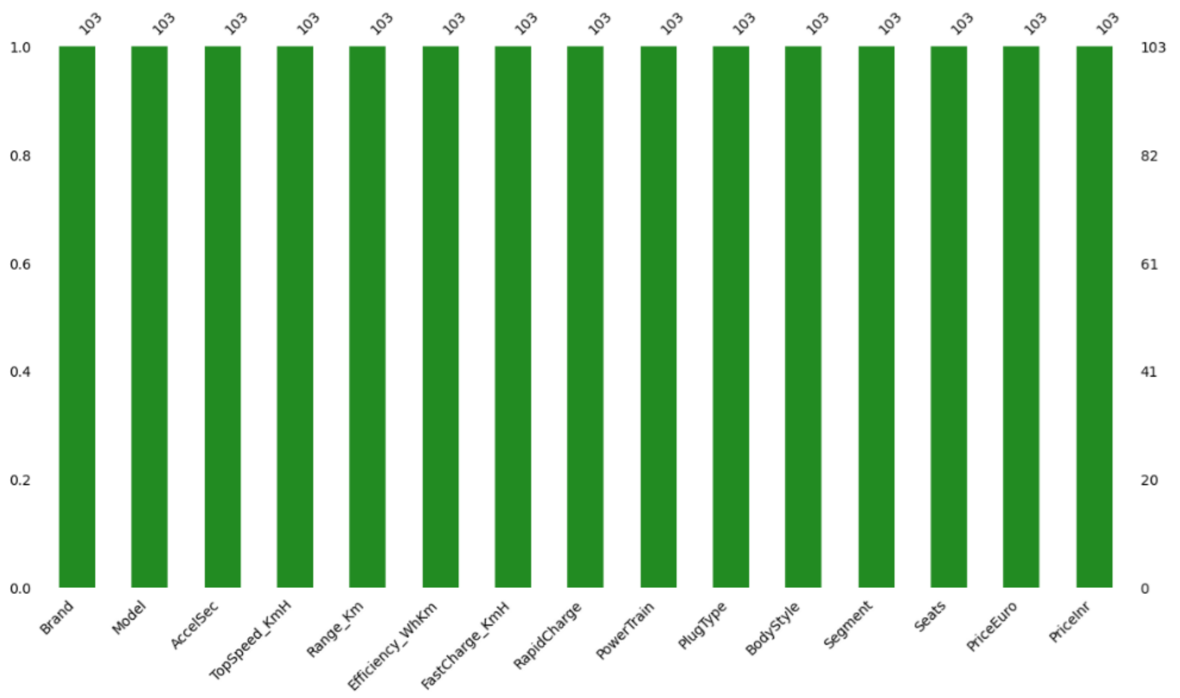
Seats: Number of seats in the vehicle.

PriceEuro: Price of the vehicle in Euros.

PriceInr : Price of the vehicle in Rupees¶

Missing Data

```
missingno.bar(df, color="forestgreen", sort="ascending", figsize=(14,7), fontsize=10);
```



There are no missing Data

Descriptive Analysis

```
df['RapidCharge'].replace(['No','Yes'],value=[0, 1],inplace=True)
```

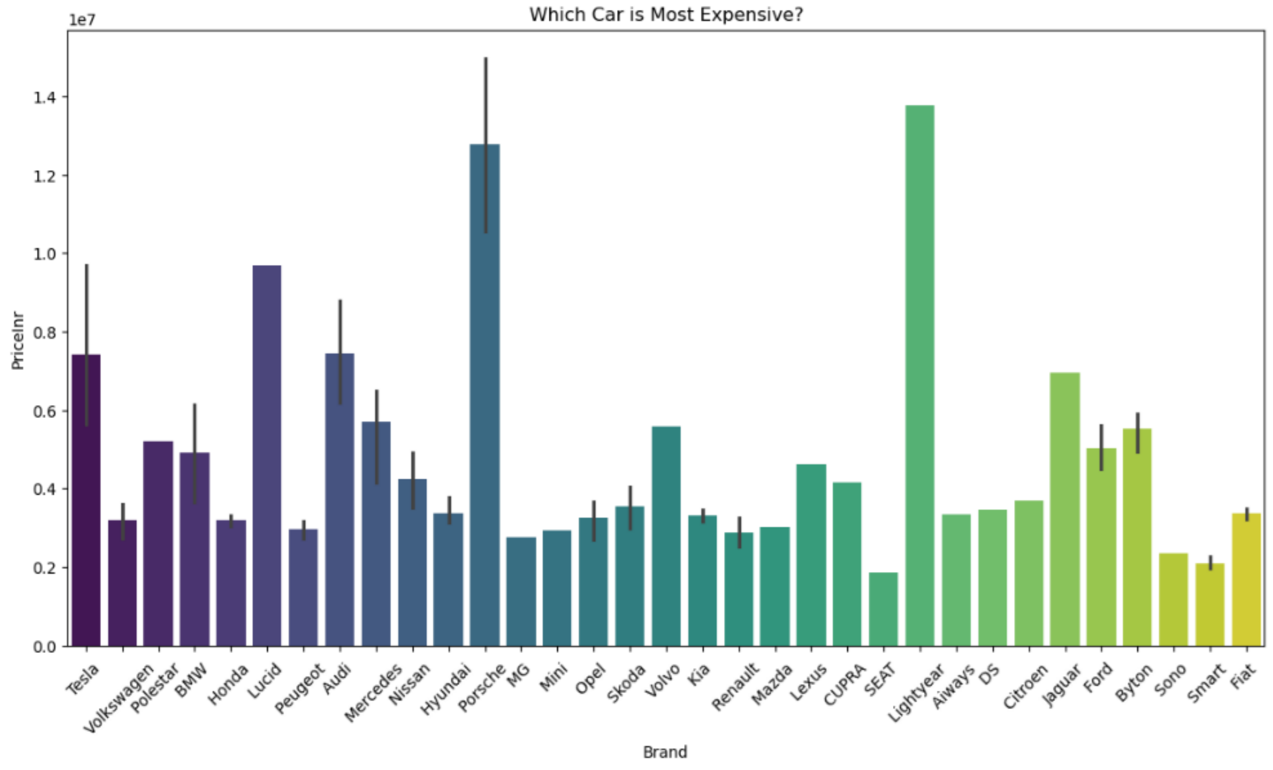
```
df.describe()
```

	AccelSec	TopSpeed_KmH	Range_Km	Efficiency_WhKm	FastCharge_KmH	RapidCharge	Seats	PriceEuro	PriceInr
count	103.000000	103.000000	103.000000	103.000000	103.000000	103.000000	103.000000	103.000000	1.030000e+02
mean	7.396117	179.194175	338.786408	189.165049	444.271845	0.747573	4.883495	55811.563107	5.150291e+06
std	3.017430	43.573030	126.014444	29.566839	203.949253	0.436529	0.795834	34134.665280	3.149947e+06
min	2.100000	123.000000	95.000000	104.000000	170.000000	0.000000	2.000000	20129.000000	1.857504e+06
25%	5.100000	150.000000	250.000000	168.000000	260.000000	0.500000	5.000000	34429.500000	3.177154e+06
50%	7.300000	160.000000	340.000000	180.000000	440.000000	1.000000	5.000000	45000.000000	4.152600e+06
75%	9.000000	200.000000	400.000000	203.000000	555.000000	1.000000	5.000000	65000.000000	5.998200e+06
max	22.400000	410.000000	970.000000	273.000000	940.000000	1.000000	7.000000	215000.000000	1.984020e+07

Descriptive data shows the mean, Standard deviation etc

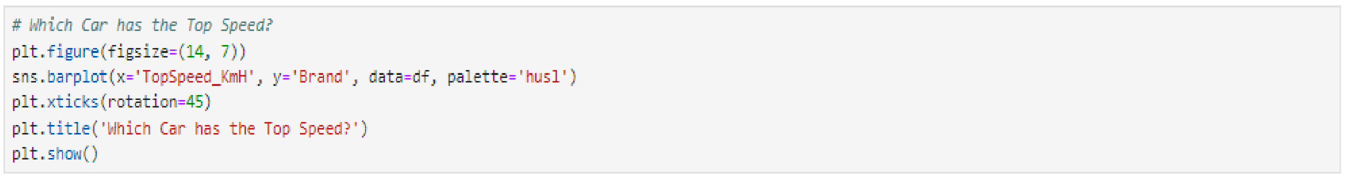
Most Expensive car

```
# Which Car is most Expensive?
plt.figure(figsize=(14, 7))
sns.barplot(x='Brand', y='PriceInr', data=df, palette='viridis')
plt.xticks(rotation=45)
plt.title('Which Car is Most Expensive?')
plt.show()
```

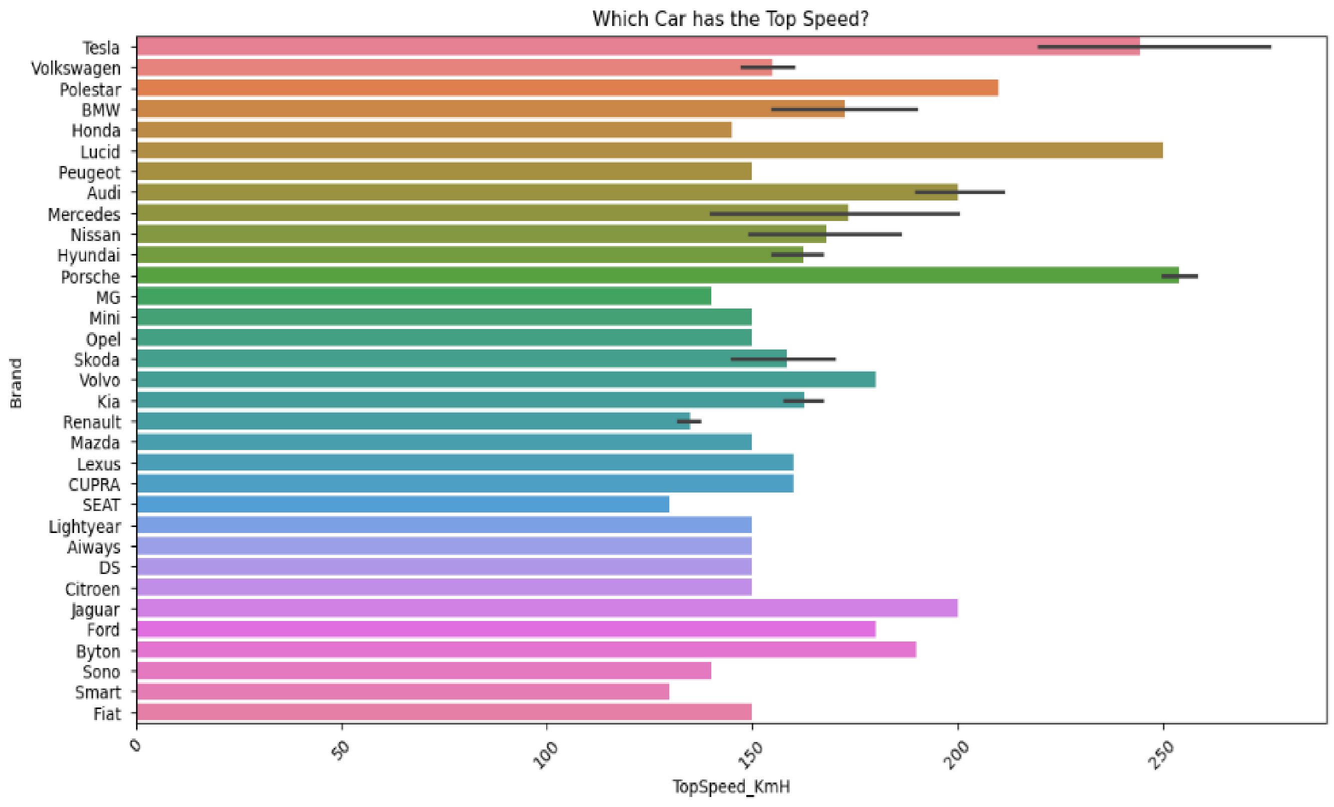


Porsche is the most expensive and seat in the cheapest.

Which car has the top speed

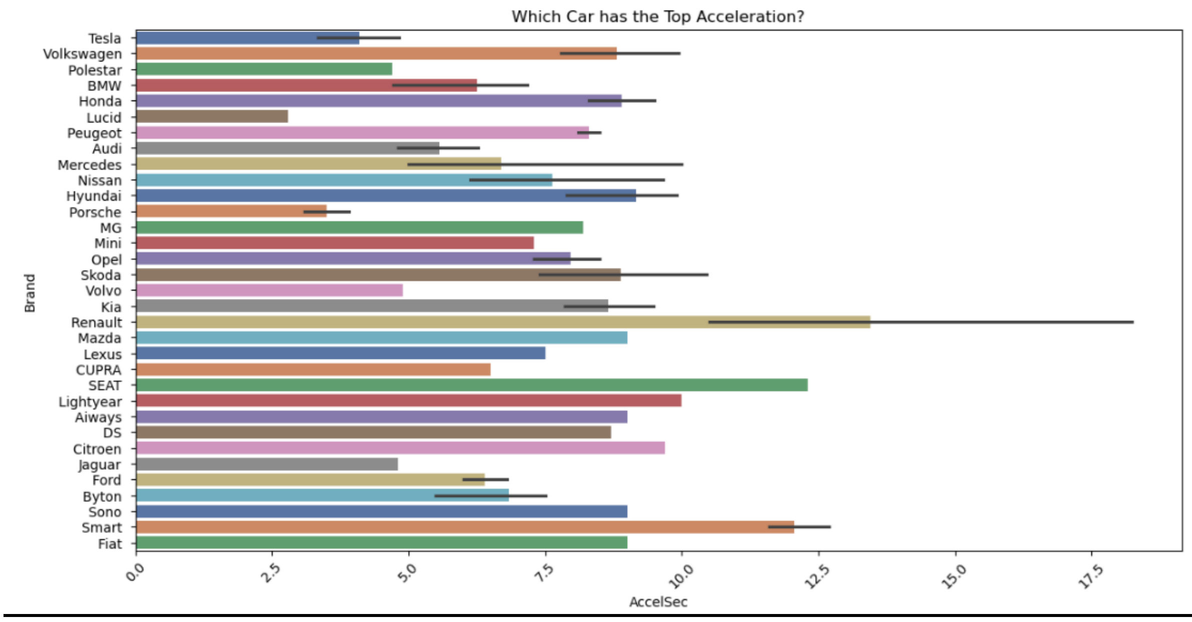


```
# Which Car has the Top Speed?
plt.figure(figsize=(14, 7))
sns.barplot(x='TopSpeed_KmH', y='Brand', data=df, palette='husl')
plt.xticks(rotation=45)
plt.title('Which Car has the Top Speed?')
plt.show()
```

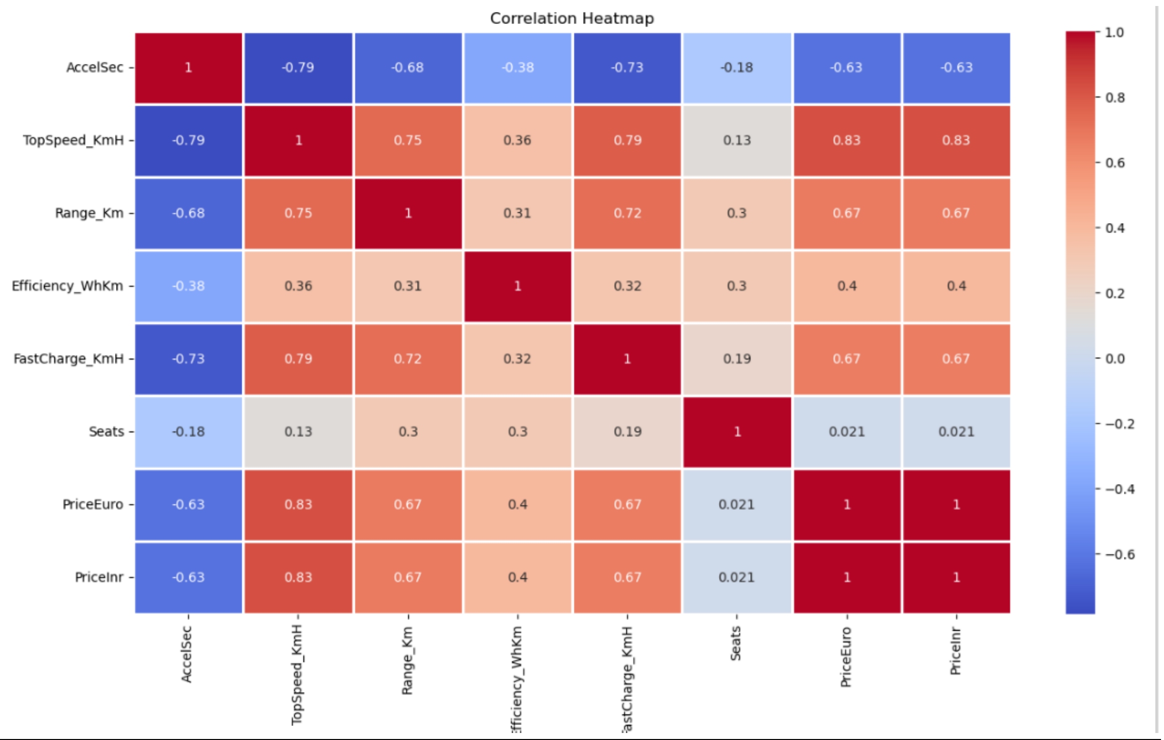


Tesla being the top seat in the low side in speed.

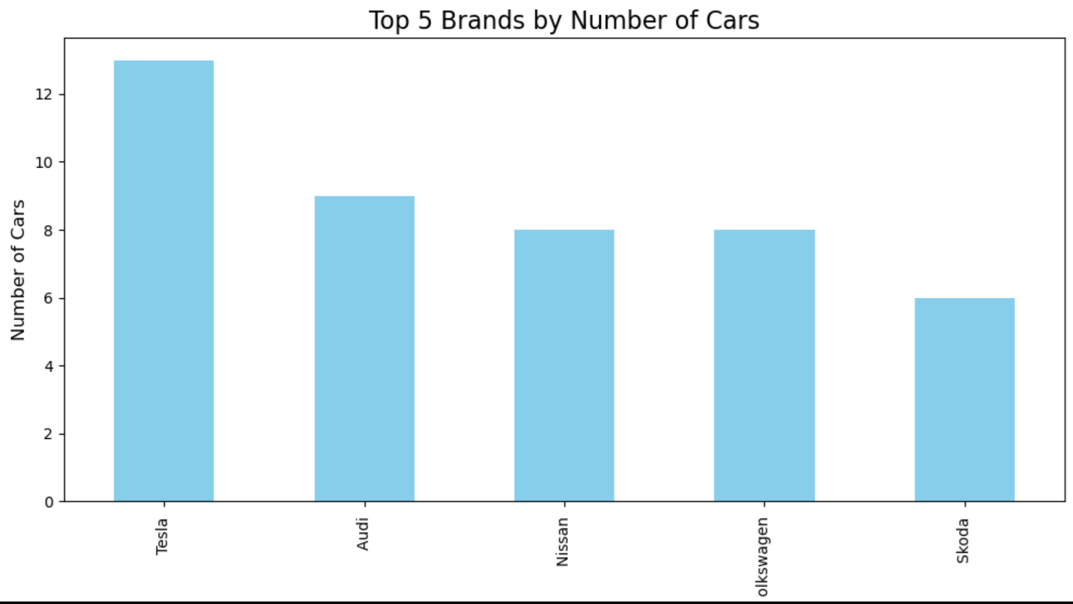
Which Car has the Top Acceleration? # Acceleration time from 0 to 100 km/h



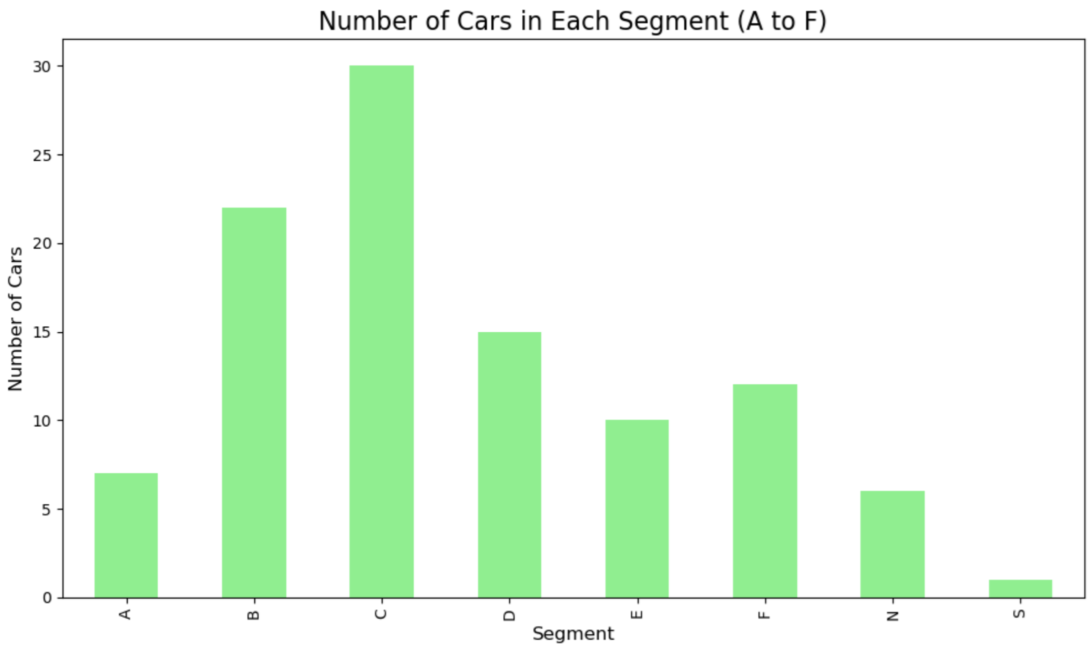
Correlation of the data using Heatmap



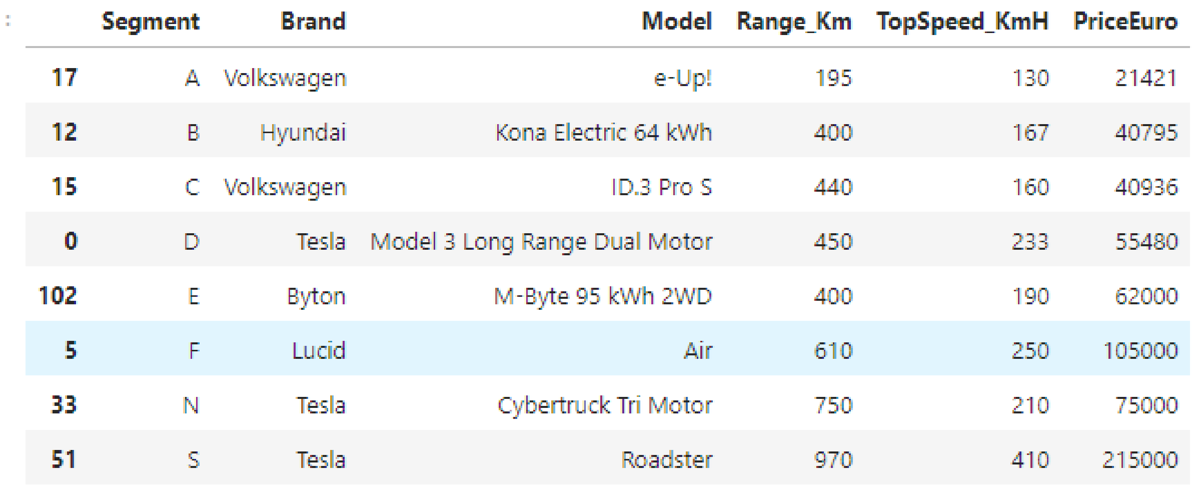
# Plot the top 5 brands in a bar chart



Segment of the cars

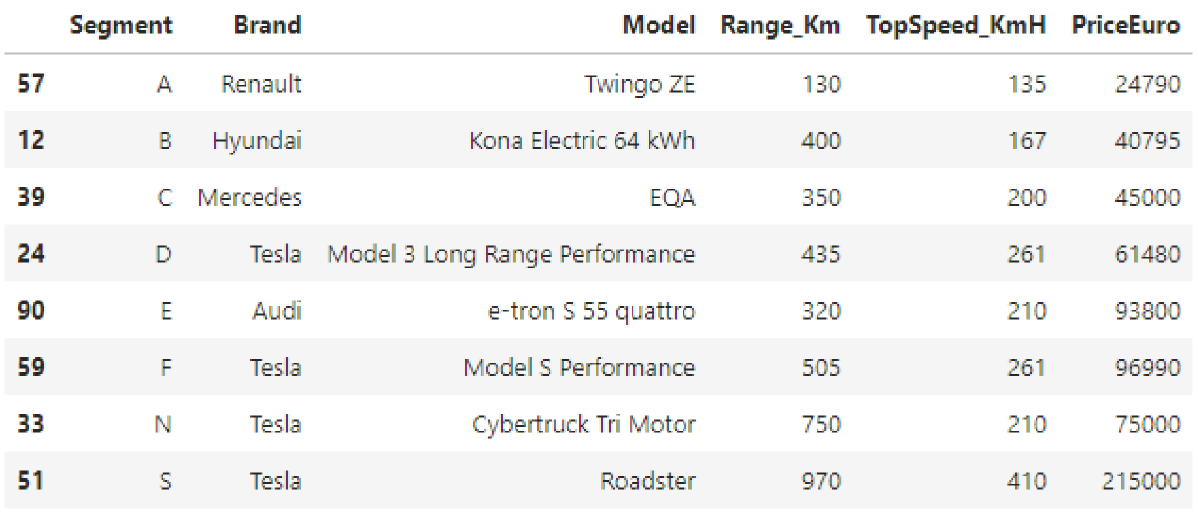


# Finding the best car in each segment through Range_Km



	Segment	Brand	Model	Range_Km	TopSpeed_KmH	PriceEuro
17	A	Volkswagen	e-Up!	195	130	21421
12	B	Hyundai	Kona Electric 64 kWh	400	167	40795
15	C	Volkswagen	ID.3 Pro S	440	160	40936
0	D	Tesla	Model 3 Long Range Dual Motor	450	233	55480
102	E	Byton	M-Byte 95 kWh 2WD	400	190	62000
5	F	Lucid	Air	610	250	105000
33	N	Tesla	Cybertruck Tri Motor	750	210	75000
51	S	Tesla	Roadster	970	410	215000

# Finding the best car in each segment through TopSpeed_KmH



	Segment	Brand	Model	Range_Km	TopSpeed_KmH	PriceEuro
57	A	Renault	Twingo ZE	130	135	24790
12	B	Hyundai	Kona Electric 64 kWh	400	167	40795
39	C	Mercedes	EQA	350	200	45000
24	D	Tesla	Model 3 Long Range Performance	435	261	61480
90	E	Audi	e-tron S 55 quattro	320	210	93800
59	F	Tesla	Model S Performance	505	261	96990
33	N	Tesla	Cybertruck Tri Motor	750	210	75000
51	S	Tesla	Roadster	970	410	215000

Here is a summary of the market segment analysis:

1. **Segment A (Low-end vehicles):**
   - Average Price: €22,693
   - Average Range: 143 km
   - Average Acceleration: 12.2 seconds (0-100 km/h)
   - Average Top Speed: 131 km/h
   - Efficiency: 169 Wh/km
   - Seats: 3.4
2. **Segment B (Entry-level compact vehicles):**
   - Average Price: €34,799
   - Average Range: 266 km
   - Average Acceleration: 8.7 seconds
   - Top Speed: 151 km/h
   - Efficiency: 169 Wh/km
   - Seats: 4.7
3. **Segment C (Mid-range compact vehicles):**
   - Average Price: €41,199
   - Average Range: 329 km
   - Acceleration: 7.8 seconds

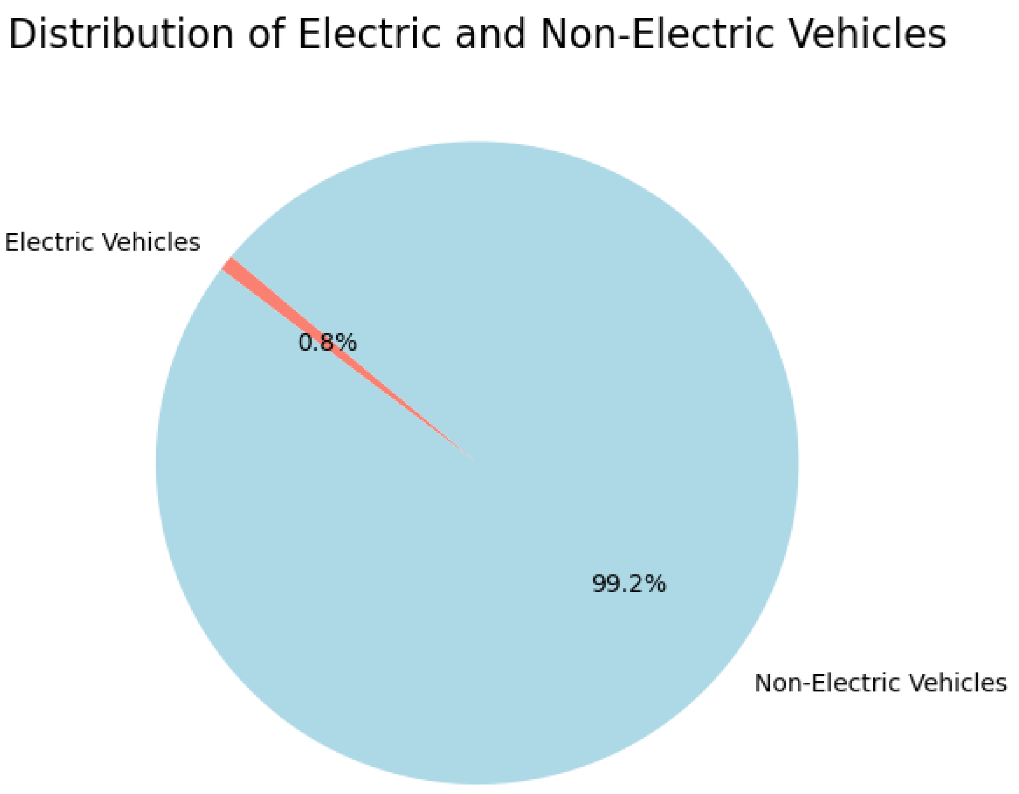
- Top Speed: 164 km/h
- Efficiency: 181 Wh/km
- Seats: 4.9

4. **<u>Segment D (Upper-mid-range vehicles):</u>**
    - Average Price: €58,488
    - Average Range: 400 km
    - Acceleration: 5.4 seconds
    - Top Speed: 202 km/h
    - Efficiency: 187 Wh/km
    - Seats: 5.3
5. **<u>Segment E (High-end vehicles):</u>**
    - Average Price: €74,269
    - Average Range: 346 km
    - Acceleration: 5.9 seconds
    - Top Speed: 197 km/h
    - Efficiency: 238 Wh/km
    - Seats: 5
6. **<u>Segment F (Luxury vehicles):</u>**
    - Average Price: €119,691
    - Average Range: 455 km
    - Acceleration: 4.0 seconds
    - Top Speed: 243 km/h
    - Efficiency: 194 Wh/km
    - Seats: 4.8
7. **<u>Segment S (Supercars):</u>**
    - Average Price: €215,000
    - Range: 970 km
    - Acceleration: 2.1 seconds
    - **<u>Top Speed: 410 km/h</u>**

- Efficiency: 206 Wh/km
- Seats: 4

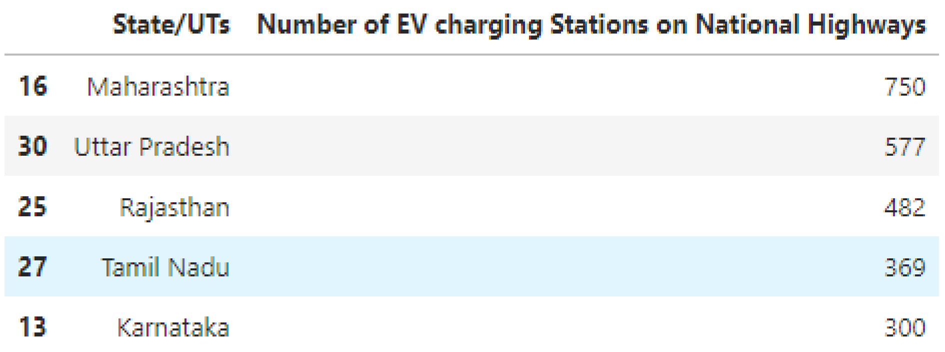
Insights

- Supercars in Segment S have extreme performance and high range, while low-end vehicles in Segment A are affordable but offer modest performance.
- Segment F (luxury cars) strikes a balance between range, acceleration, and top speed, at a premium price.
- Segment D vehicles provide a good middle ground in terms of price, range, and performance for the average consumer.



Only 0.8 % of the Indian vehicles are Electric Vehicles

Top 5 Charging station across India



	State/UTs	Number of EV charging Stations on National Highways
16	Maharashtra	750
30	Uttar Pradesh	577
25	Rajasthan	482
27	Tamil Nadu	369
13	Karnataka	300

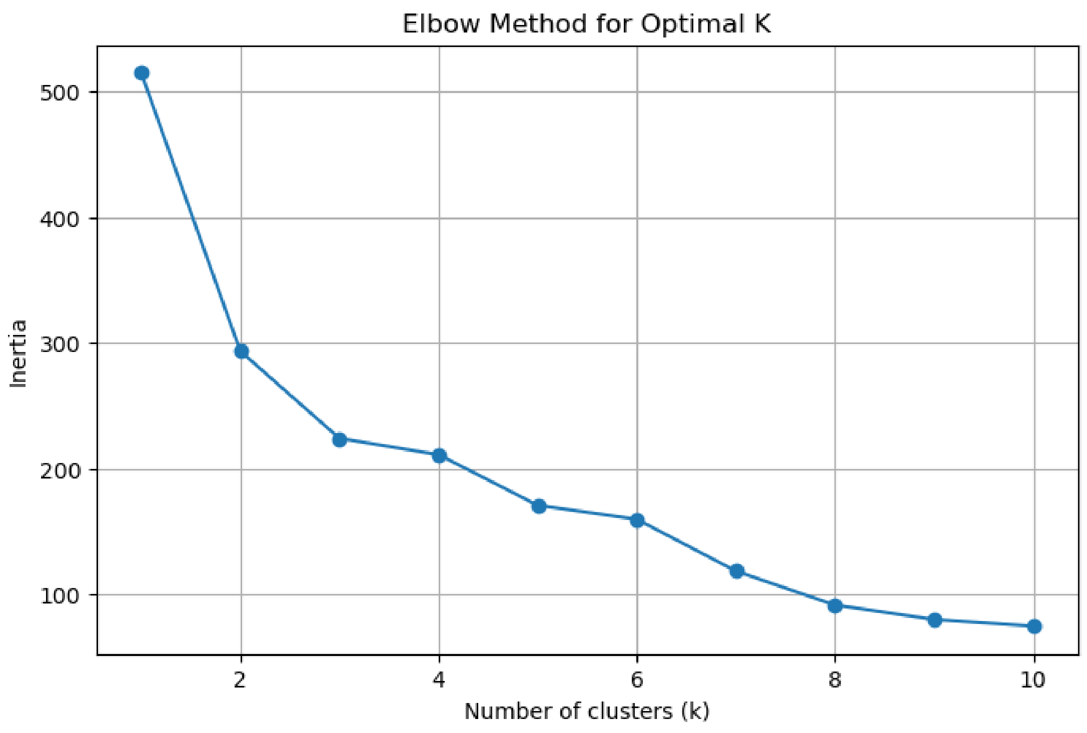
To perform **clustering** and **regression** tasks based on the dataset, we will break the analysis into two sections:

1. Clustering Analysis

We can use clustering to identify different groups of vehicles based on attributes like **Price**, **Range**, **Efficiency**, **Top Speed**, and **Acceleration**. Clustering will help uncover patterns in the vehicle market, revealing natural groupings that might correspond to budget cars, performance vehicles, or luxury cars.

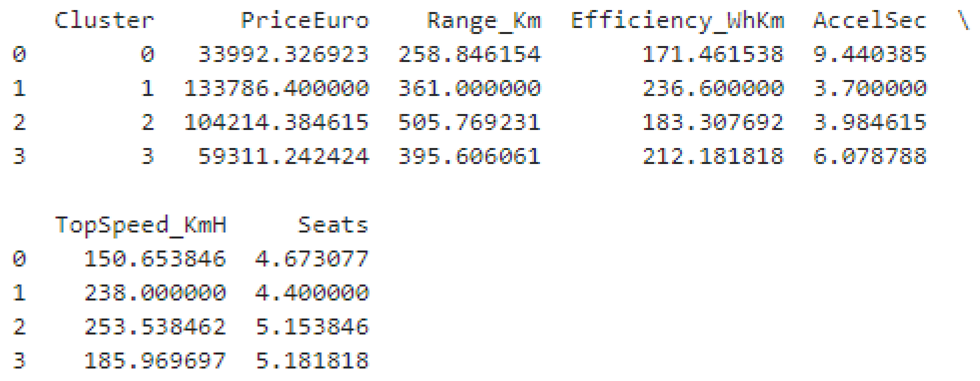
2. Regression Analysis

In a regression task, we will try to predict the vehicle's **Price** based on key features such as **Range**, **Acceleration**, **Top Speed**, and other relevant attributes. This will allow us to understand how various factors influence the price of a vehicle.



The **Elbow Method** plot shows the optimal number of clusters where the inertia (sum of squared distances to the nearest cluster center) begins to flatten, typically indicating the best balance between cluster quantity and quality.

Cluster Analysis



```
   Cluster      PriceEuro    Range_Km  Efficiency_WhKm  AccelSec  \
0        0   33992.326923  258.846154       171.461538  9.440385
1        1  133786.400000  361.000000       236.600000  3.700000
2        2  104214.384615  505.769231       183.307692  3.984615
3        3   59311.242424  395.606061       212.181818  6.078788

   TopSpeed_KmH     Seats
0    150.653846  4.673077
1    238.000000  4.400000
2    253.538462  5.153846
3    185.969697  5.181818
```

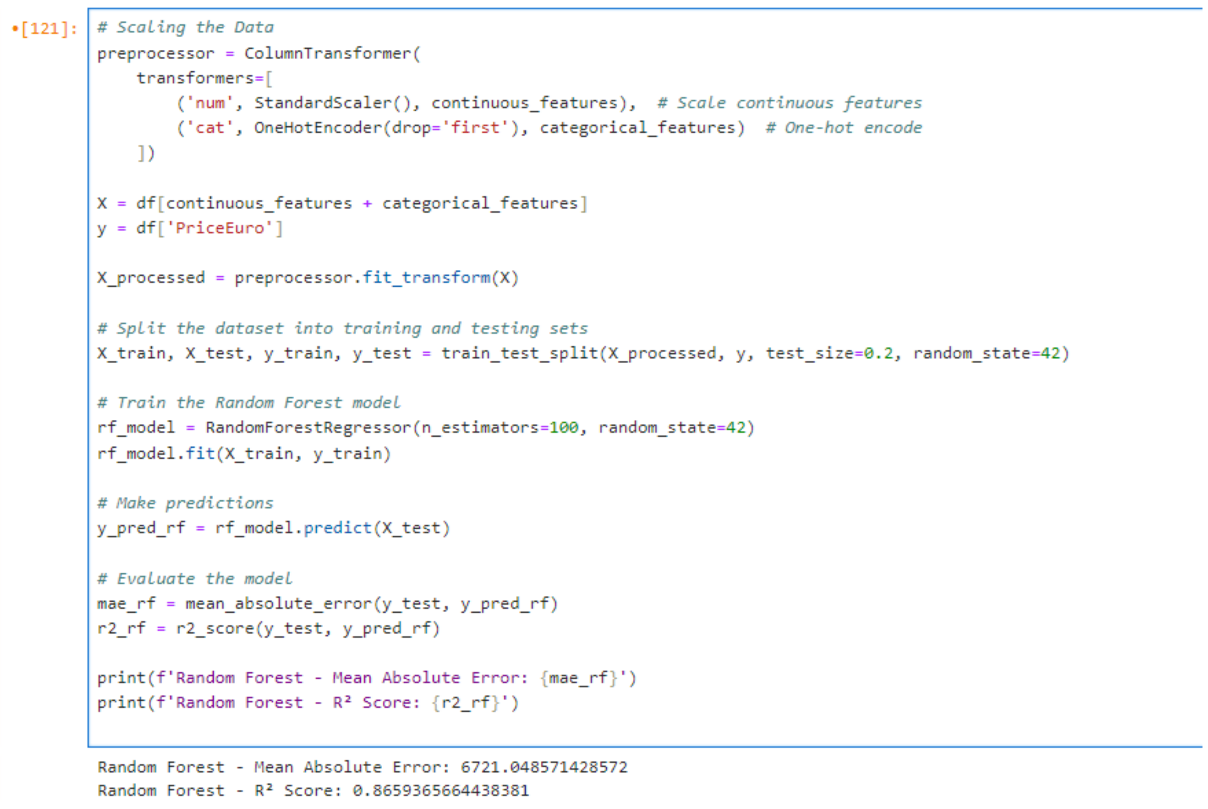
- Cluster 0
  This cluster represents vehicles with a moderate price and range. The efficiency is relatively high, and the acceleration is slower compared to other clusters. This may indicate vehicles targeted at budget-conscious consumers who prioritize efficiency and moderate range over performance.
- Cluster 1
  This cluster features high-end vehicles with a high price and good range. The acceleration is quick, indicating a focus on performance. These vehicles may cater to luxury consumers who value speed and range over cost.
- Cluster 2
  This cluster showcases vehicles that balance performance, range, and price. With a substantial range and good acceleration, these vehicles are likely aimed at consumers looking for a mix of comfort, practicality, and performance.
- Cluster 3
  Vehicles in this cluster have a moderate price and good range, with decent acceleration. They might appeal to consumers looking for a reliable and comfortable vehicle without venturing into the luxury market.

Random Forest

Definition: Random Forest is an ensemble machine learning algorithm that utilizes multiple decision trees to improve predictive accuracy and control overfitting. It is widely used for both classification and regression tasks.

How It Works:

- **Ensemble Learning: Random Forest builds multiple decision trees during training and merges their outputs. Each tree is trained on a random subset of the data and features, which introduces diversity and improves generalization.**
- **Bootstrap Aggregating (Bagging): Each tree is trained on a random sample of the dataset (with replacement), helping to reduce variance.**
- **Feature Randomness: For each split in the decision trees, a random subset of features is considered, which enhances the model's robustness and reduces overfitting.**



```
•[121]: # Scaling the Data
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), continuous_features),  # Scale continuous features
        ('cat', OneHotEncoder(drop='first'), categorical_features)  # One-hot encode
    ])

X = df[continuous_features + categorical_features]
y = df['PriceEuro']

X_processed = preprocessor.fit_transform(X)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_processed, y, test_size=0.2, random_state=42)

# Train the Random Forest model
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Make predictions
y_pred_rf = rf_model.predict(X_test)

# Evaluate the model
mae_rf = mean_absolute_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)

print(f'Random Forest - Mean Absolute Error: {mae_rf}')
print(f'Random Forest - R² Score: {r2_rf}')
```

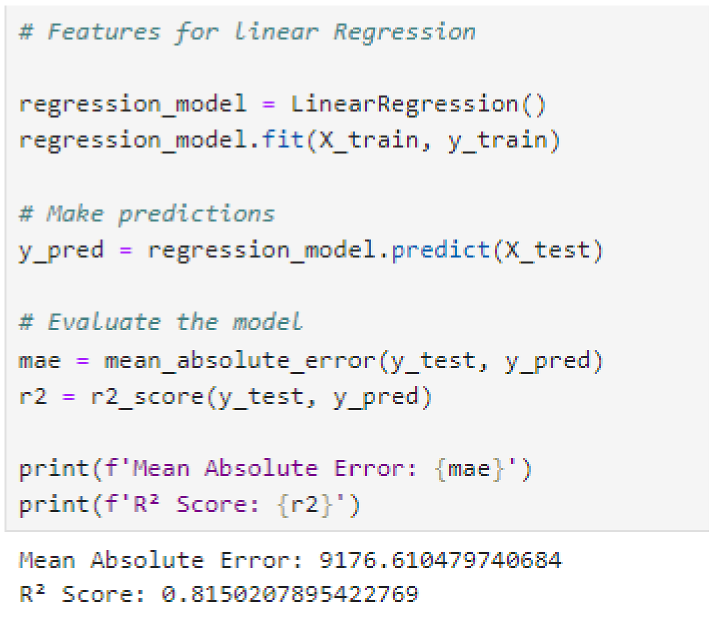
```
Random Forest - Mean Absolute Error: 6721.048571428572
Random Forest - R² Score: 0.8659365664438381
```

Linear Regression

Definition: Linear Regression is a statistical method used for modeling the relationship between a dependent variable and one or more independent variables. The relationship is assumed to be linear.

How It Works:

- **Modeling**: Linear Regression fits a straight line (or hyperplane in higher dimensions) to the data that minimizes the sum of the squared differences (errors) between the observed values and the values predicted by the model.
- **Equation**: The general form of the linear regression equation is: y=b0+b1x1+b2x2+...+bnxn+ϵy = b_0 + b_1x_1 + b_2x_2 + ... + b_nx_n + \epsilony=b0+b1x1+b2x2+...+bnxn+ϵ where yyy is the dependent variable, b0b_0b0 is the intercept, b1,b2,...,bnb_1, b_2, ..., b_nb1,b2,...,bn are the coefficients for the independent variables x1,x2,...,xnx_1, x_2, ..., x_nx1,x2,...,xn, and ϵ\epsilonϵ is the error term.



```
# Features for Linear Regression

regression_model = LinearRegression()
regression_model.fit(X_train, y_train)

# Make predictions
y_pred = regression_model.predict(X_test)

# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Absolute Error: {mae}')
print(f'R² Score: {r2}')
```

```
Mean Absolute Error: 9176.610479740684
R² Score: 0.8150207895422769
```

Conclustion

Random Forest demonstrated superior predictive performance, evidenced by a Mean Absolute Error (MAE) of approximately 6271.04 and an R^2 score of about 0.86. This indicates that the Random Forest model captured a substantial portion of the variance in the data, making it effective for predicting the number of EV charging stations based on various features. The model's ensemble nature allowed it to account for non-linear relationships and interactions between variables, which is particularly beneficial given the complexities of real-world data.

Linear Regression, while straightforward, produced less accurate predictions in this context. Its reliance on the assumption of linear relationships may have limited its ability to effectively model the underlying data patterns, especially in a dataset with potentially complex interactions among predictors.

<u>Marketing mix</u>

• Price: refers to the value that is put for a product. It depends on segment targeted, ability of the companies to pay, ability of customers to pay supply - demand and a host of other direct and indirect factors.

• Product: refers to the product actually being sold – In this case, the service. The product must deliver a minimum level of performance; otherwise even the best work on the other elements of the marketing mix won't do any good.

• Place: refers to the point of sale. In every industry, catching the eye of the consumer and making it easy for her to buy it is the main aim of a good distribution or 'place' strategy. Retailers pay a premium for the right location. In fact, the mantra of a successful retail business is 'location, location, location'.

• Promotion: this refers to all the activities undertaken to make the product or service known to the user and trade. This can include advertising, word of mouth, press reports, incentives, commissions and awards to the trade. It can also include consumer schemes, direct marketing,contest and prizes

All the elements of the marketing mix influence each other. They make up the business plan for a company and handle it right, and can give it great success. The marketing mix needs a lot of understanding, market research and consultation with several people, from users to trade to manufacturing and several others

Thank you