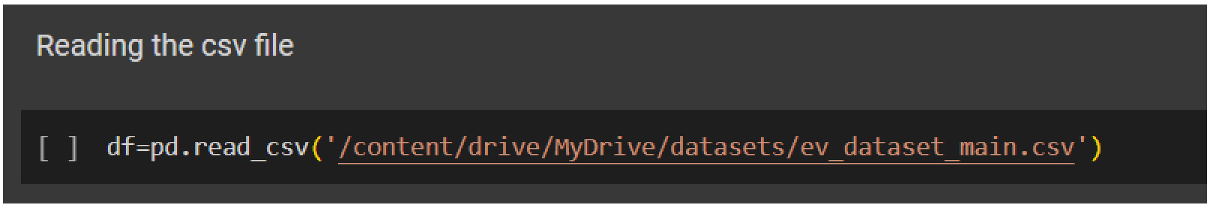
Project 2.1 Report

Vedant Poal

- **Loading data**

  We utilized the NumPy library to access data files stored on a Google Drive, which we seamlessly integrated into our Colab file.



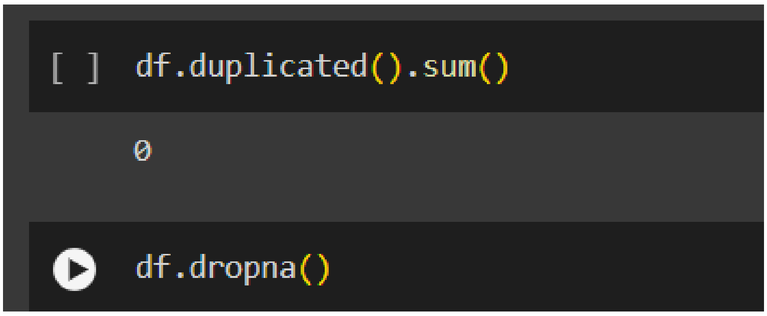
Reading the csv file

```
df=pd.read_csv('/content/drive/MyDrive/datasets/ev_dataset_main.csv')
```

Subsequently, we employed the info() and describe() functions on the dataset to glean insights into its structure, characteristics, and data types.

- **Preprocessing**

  Invariably, datasets require refinement as they are not always flawless. In this scenario, we addressed duplicate entries and eliminated rows or fields containing null or missing values to enhance the dataset's quality and reliability.

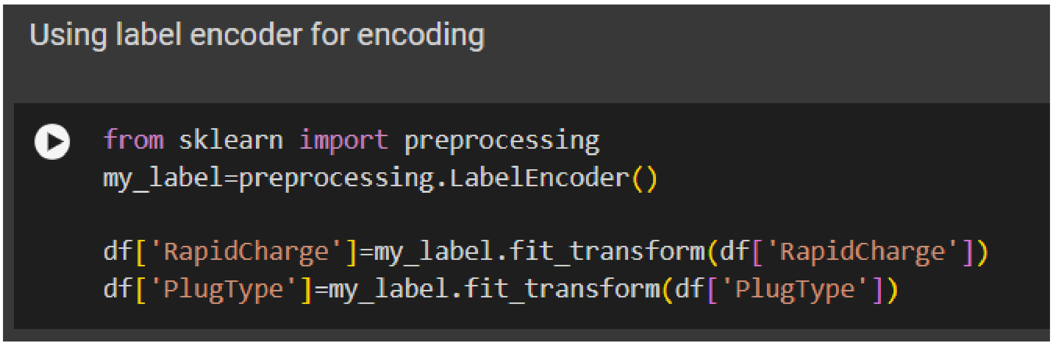


```
df.duplicated().sum()
```

```
0
```

```
df.dropna()
```

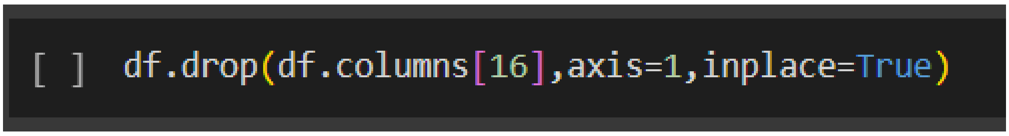
In the model plotting process, string values pose a limitation. To circumvent this, we applied label encoding to the "rapidcharge" and "plugtype" fields, converting them into numerical representations for compatibility.



Using label encoder for encoding

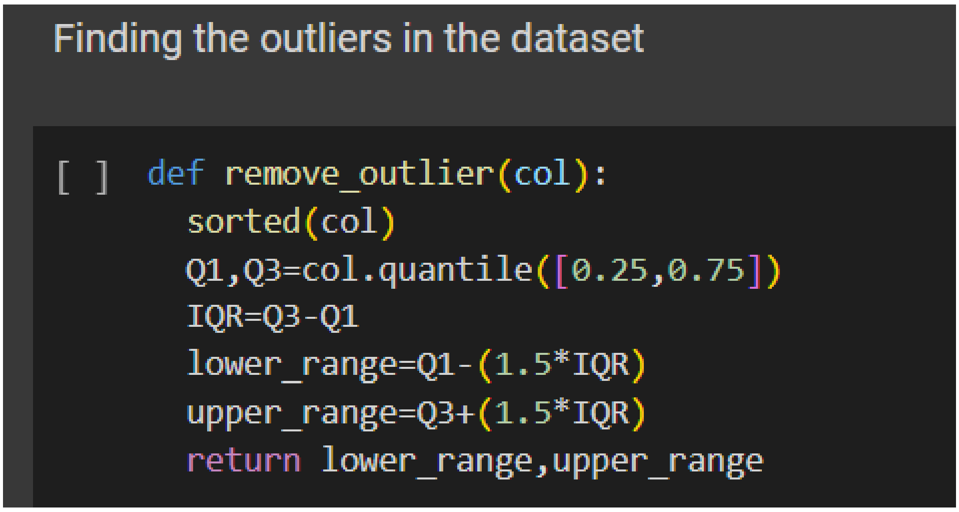
```
from sklearn import preprocessing
my_label=preprocessing.LabelEncoder()

df['RapidCharge']=my_label.fit_transform(df['RapidCharge'])
df['PlugType']=my_label.fit_transform(df['PlugType'])
```

We also eliminated column number 16 from the dataset, as it contained arbitrary values and lacked any meaningful significance.



```
df.drop(df.columns[16],axis=1,inplace=True)
```

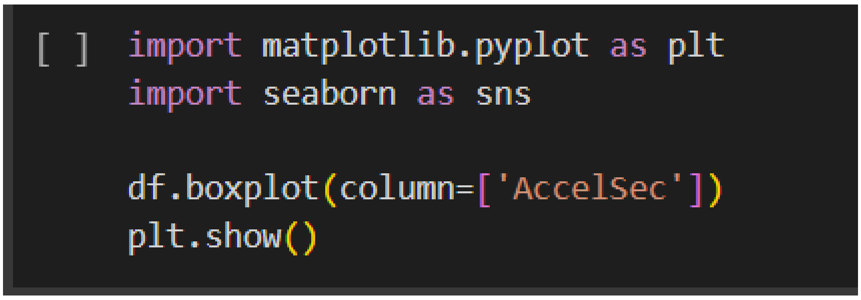
Next, as part of preprocessing, we excluded outliers or noisy data points from the dataset. These values have the potential to influence the model's predictions negatively, making it imperative to eliminate them for accurate results.



Finding the outliers in the dataset

```
def remove_outlier(col):
  sorted(col)
  Q1,Q3=col.quantile([0.25,0.75])
  IQR=Q3-Q1
  lower_range=Q1-(1.5*IQR)
  upper_range=Q3+(1.5*IQR)
  return lower_range,upper_range
```

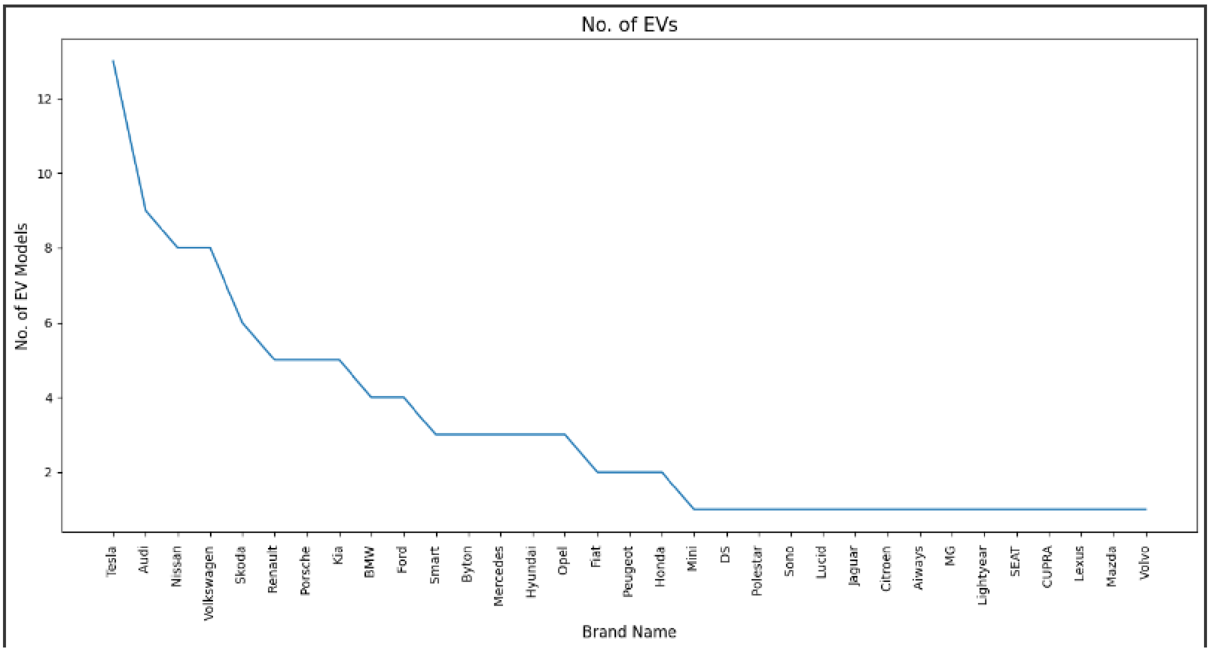
We developed a custom function to detect and remove outliers from the dataset. Utilizing a boxplot, we identified outliers and subsequently eliminated them using the defined function.

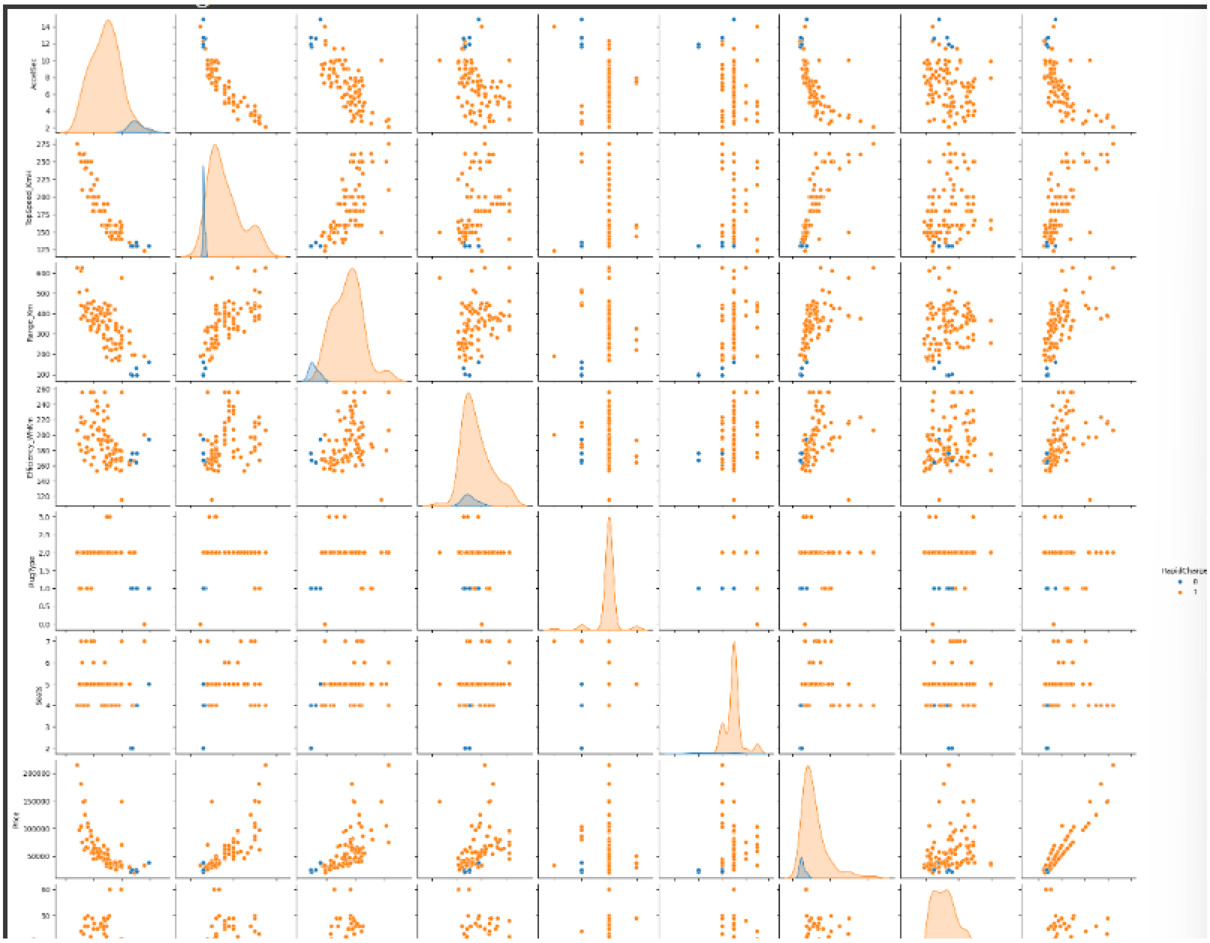


```
import matplotlib.pyplot as plt
import seaborn as sns

df.boxplot(column=['AccelSec'])
plt.show()
```

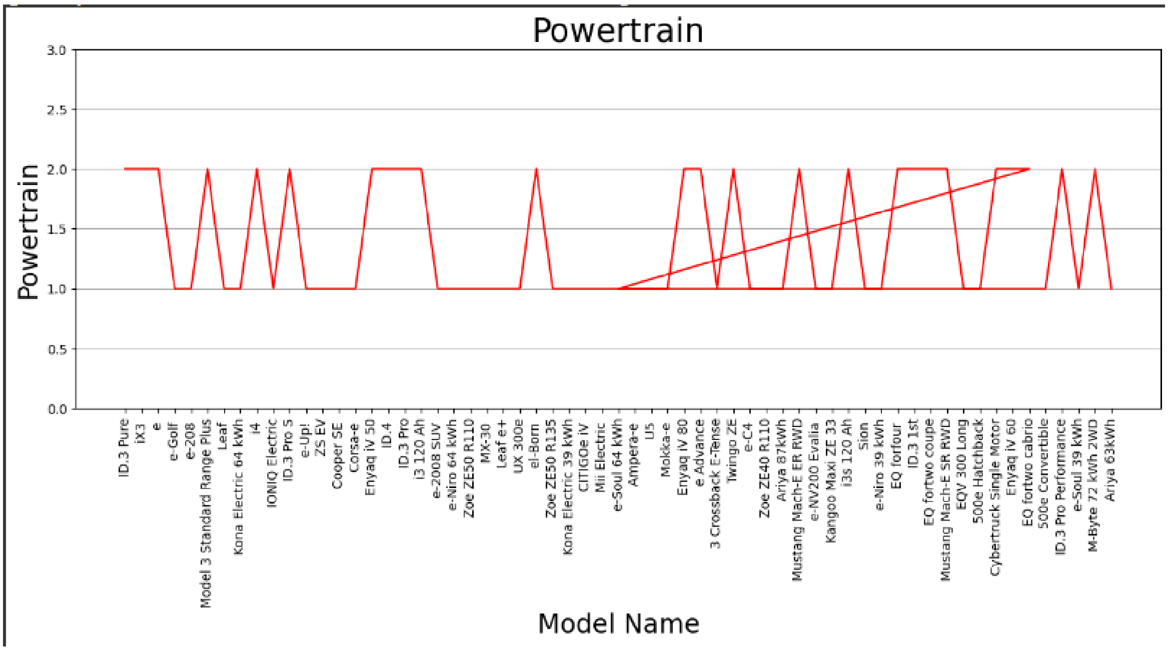
- **Visualization of data**

  Data visualization holds significant importance, providing a visual representation of the dataset's characteristics. By visually inspecting the data, we gain valuable insights into its structure and potential challenges, aiding in devising effective strategies to address the underlying problems.

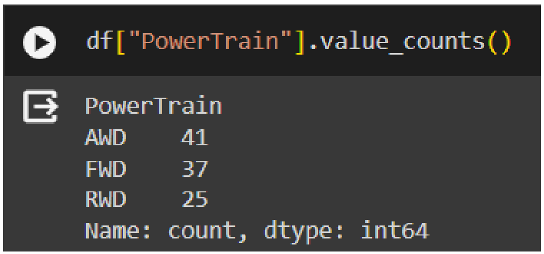

No. of EVs
No. of EV Models
2
4
6
8
10
12
Tesla
Audi
Nissan
Volkswagen
Skoda
Renault
Porsche
Kia
BMW
Ford
Smart
Byton
Mercedes
Hyundai
Opel
Fiat
Peugeot
Honda
Mini
DS
Polestar
Sono
Lucid
Jaguar
Citroen
Aiways
MG
Lightyear
SEAT
CUPRA
Lexus
Mazda
Volvo
Brand Name


AccelSec
TopSpeed_KmH
Range_Km
Efficiency_WhKm
PlugType
Seats
Price
RapidCharge
0
1

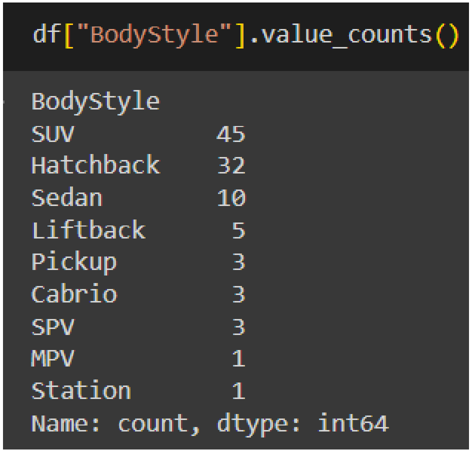
- **What type of EV the company will produce?**
  In answering the question about the type of electric vehicle the company produces, we focused on two attributes: powertrain and body style. These attributes indicate whether the car is RWD, AWD, or FWD, as well as whether it is a sedan or SUV, facilitating a comprehensive understanding of the company's EV lineup.



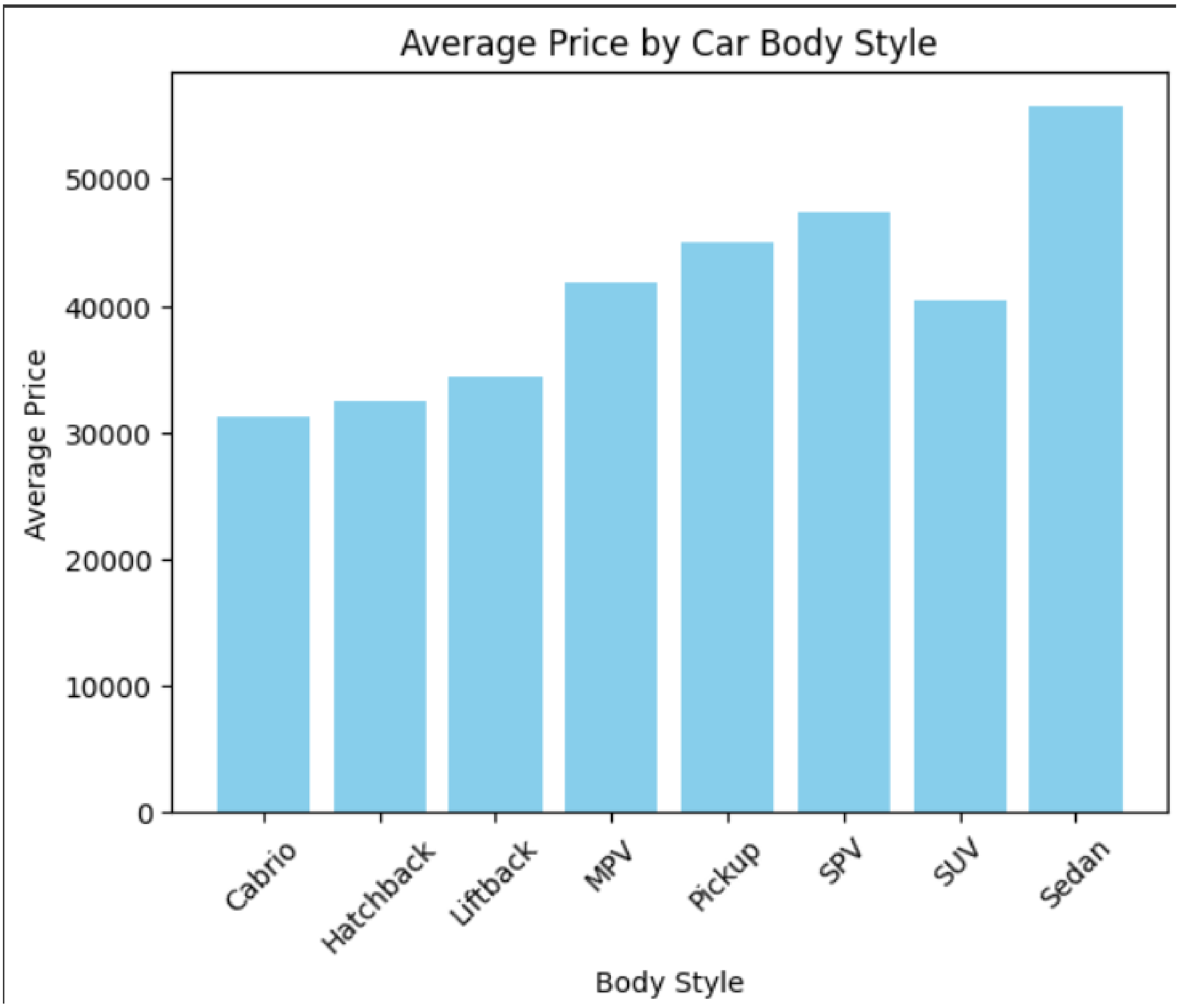
Employing filtration techniques, we constructed a dataset by prioritizing attributes, starting with powertrain and subsequently incorporating body style. Through comparisons of prices within this filtered dataset, we provided insights to answer the question. This methodological approach enabled a systematic evaluation of electric vehicle types, facilitating a well-informed response based on pricing considerations.



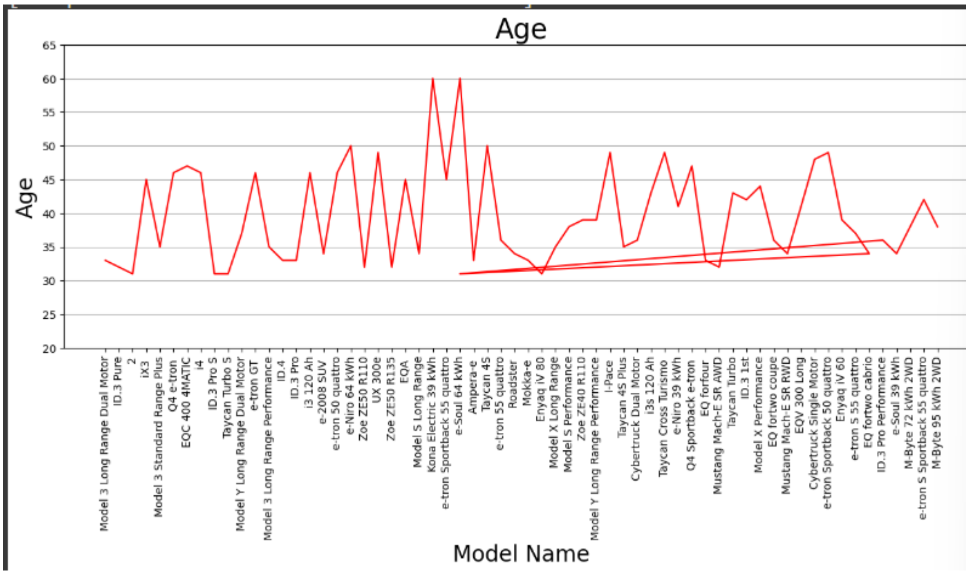
```
df["PowerTrain"].value_counts()

PowerTrain
AWD     41
FWD     37
RWD     25
Name: count, dtype: int64
```



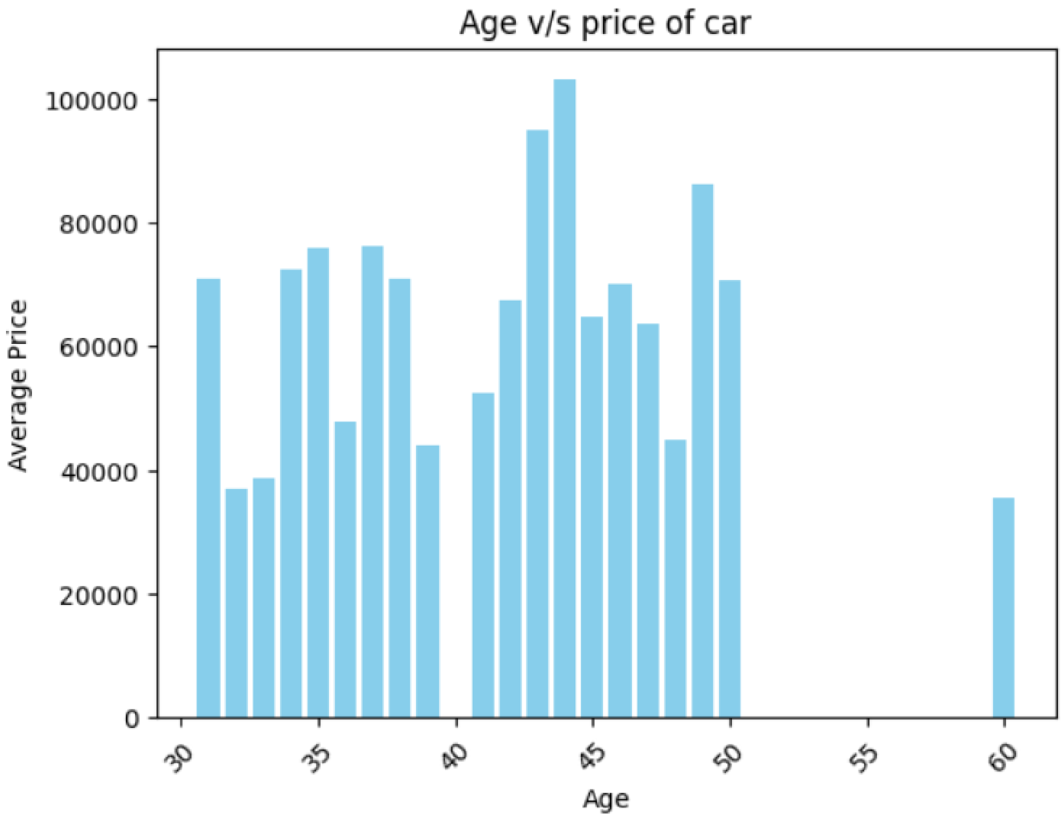
```
df["BodyStyle"].value_counts()

BodyStyle
SUV          45
Hatchback    32
Sedan        10
Liftback      5
Pickup        3
Cabrio        3
SPV           3
MPV           1
Station       1
Name: count, dtype: int64
```

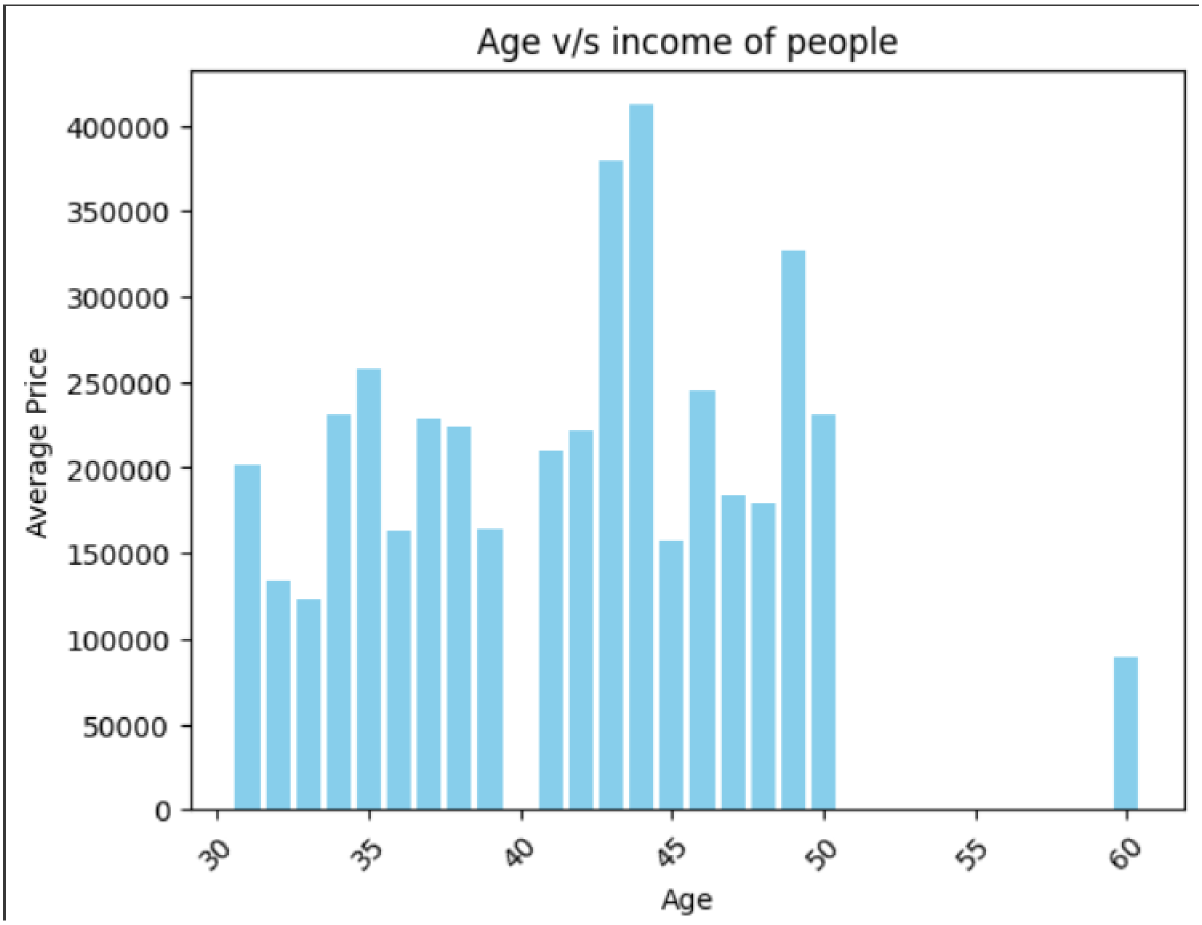


- **Who is target customer?**
  To determine the target customer, we analysed attributes such as age, income, and car price. Employing filtration techniques, we identified the ideal customer profile. By focusing on these key factors, we obtained insights to effectively pinpoint the target demographic for the product.



Age v/s price of the car


Age v/s price of car
Average Price
Age


Age v/s income of people
Average Price
Age