Project Report

Goa Business School, Goa University

Data Analytics

2020-21

Group 5

Car Analytics

Car Analytics for Indian cars Price prediction

And Give subjections for features in selected price range.

Group Members

Roll No.	Name of Students	Job Rolls
1918	Himanshul Keshari	Scrapping data, web application in Flask, sweetbes report, Plots
1928	Yogeshwar Manerikar	Sorting data , Handles collab , Power Bi,Model bulding
1952	Prateek Satardekar	Planning, schedule ,distribution of work , project lead ,plots
1955	Mrudul Shirodkar	Data cleaning, Selecting the column for model building, Plots

ACKNOWLEDGEMENT

We would like to thanks the krushna naik for her professional input, feedback, and support, as well explaining the need required to make our product a successful one. We would also like to thank Professor Baskar for their continued guidance and feedback throughout the course of the project.

In this Mount Everest journey every member is holding hand in hand to serve the mountains , I appreciate all my team members for there efforts .

Project links :- (full folder)

https://drive.google.com/drive/u/0/folders/1lLzxJKla1ZtbXDPQyNNc9m9sQRTt_mx7

Notebooks:-

https://drive.google.com/drive/u/0/folders/1IW8o77C5O2g0GNqvcdhsearpFEBY8i5R

Sweetviz Report Link:-

https://drive.google.com/drive/u/0/folders/1lLzxJKla1ZtbXDPQyNNc9m9sQRTt_mx7

Table of Contain

Why we choose this topic?

The automotive industry continues to face a dynamic set of challenges. For customers making a informed decision is difficult without having all the data organized .so data analytics is what will provide this tool for customers to make the right Decision .For those with
the right ambition it represents an exciting time with opportunities to differentiate and
stand out from the crowd. One area that has the opportunity to deliver significant competitive advantage is analytics.

What are the goals of this project?

The goal of the project is to make sure customer can get all the data available for making a informed decision before purchasing a automobile**.**To succeed will need to fully understand customer needs and behaviors in order to develop . a single view of the customer and thereby build compelling, differentiated offers throughout the sale and ownership cycle which are relevant in today's digital environment. For example, attract new target customer segments and provide a transformational retailing experience for shopping anytime, anywhere. Finally, customer retention needs to be placed high on the agenda and with that comes the increasing importance of understanding and influencing customer experience across the lifecycle and journey, not just during the purchase phase. So on the basis of costumer selection of there requirement we suggest them car and price

Proposal of topic

Analytics and information management present a significant opportunity for automakers to use quantitative techniques to support the planning of interventions across the customer lifecycle including but not limited to

Goals

• Understanding the potential value of different customer segments;

• Using that knowledge to strategically target new customers whilst maintaining the loyalty of existing customers; and

• Give the suggestion to customers .

- Predict the price depends upon features
- Give one UI for Customer to interact with Model

Data Life Cycle

1. Planning

- Business decision:-
    1. If any one want to buy a car then what factors are important in car that is subjects
    2. And based on features it predict price
- Type of data in project :-
    1. Discreet
    2. Categorical
    3. Nominal
    4. Ordinal
- Responsibility's :-
    1. Data capturing
    2. Data cleaning
    3. Unstructured Data into stretchered
    4. Store it for future use
- Data Capturing:-
  We are done web scraping from the different sites but that information is not enough. Then we approach keegel but it gives us US car price . we made one excel sheets and put all data in that. Size of excel file is (1220 *120) .This phase is very time consuming .

- Analyze the data:-
  From the structured data we took up all things for EDA methods all Car features are important to plot the graph And for building the model ,reused the columns.

Data Preparation

1. Get the data from Local System :-
   The data can be taken in csv file format .we can also save the file in drive and give the access to collab

```
2. #load data
3. from google.colab import files
4. uploaded = files.upload()
```

1. Head of the data set is uploaded (2)

```
df= pd.read csv('cars ds final1.csv')
df.head(2)
```

2. **See the shape of data set (Rows* column):-**
   For getting the idea of how many columns and rows ,and what we can remove from it

```
df.shape
```

3. Check the NULL values in Data set:-
   If null values are there it gives the error in predictions .and not getting exact result which we want .
   Missing value count in ascending order .

```
df.isnull()
#Summary of missing values
df.isnull().sum().sort_values(ascending=False)
```

4. Check the data type of all data
   If we want to change the data type into float64 we can do it .

```
df.info()
```

5. Fill nan values (0):-
   We fill the 0 in NAN space because we having null in 5 rows and then we removed those because 0 value is not compatible in Car Price case

```
new_df=df.fillna(0)
```

OR

```
df=df.dropna()
```

6. There are no NULL values in the dataset, hence it is clean.

```
df.isnull().sum()*100/df.shape[0]
```

7. **Describe the data for understand the data mean,max,std and outliers range**

```
df.describe()
```

8. How many heading or columns we have and then remove the columns

```
print(df.columns.values.tolist())
```

9. Drop unnecessary columns .
   https://colab.research.google.com/drive/1zxmUsLZxC6HdX2gcPzpZxGkHa9YnZ7AJ#scrollTo=WGs0nrc_5YTc&line=4&uniqifier=1

```
df.drop('Drivetrain',axis=1,inplace=True)
```

   Write the column name which is not important .

10. Heat map
    Find the correlation in the variables and depend upon that you take the decision

```
plt.figure(figsize = (15, 15))
sns.heatmap(df.corr(), annot=True, annot_kws={'size': 15})
```

11. Remove the rows
    It contains Zero values

```
#remove the null rows 1272 and all
df.drop([1272,1273,1274,1275],axis=0,inplace=False)
```

EDA

12. Sweetviz library

Fast and insightful Exploratory Data Analysis using Sweetviz 1.1,

- **What are the features, their types and distributions?**
- **Are there any correlations or relationships between them?**
- **How much missing data or duplicates are there, and what are some common values?**

These new features in Sweetviz 1.1 improve insights and usability, and are part of the ongoing development on this open-source project.

Sweetviz summarizes data in a clean, compact form that answers most of the initial questions I have when looking at a new dataset. It provides instant insights and saves me a ton of time, and I hope you find it as useful as I do

1. We neeed to split the data it can be 30-70 ,80-20 but we are selected the 70 30

```
2. df['split'] = np.random.randn(df.shape[0], 1)
3. msk = np.random.rand(len(df)) <= 0.7
4. train = df[msk]
5. test = df[~msk]
```

2. import the library's

```
!pip install sweetviz
import sweetviz
```

3. create the report

```
my_report = sweetviz.compare([train, "Train"], [test, "Test"])
```

4. give the report name and save

```
my_report.show_html('Report.html')
```

5. Report Link:-

https://drive.google.com/drive/u/0/folders/1lLzxJKla1ZtbXDPQyNNc9m9sQRTt_mx7

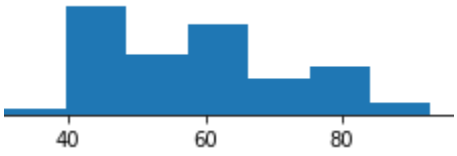
13. **ALL Plots on Processed data**

Save the clean data in new csv

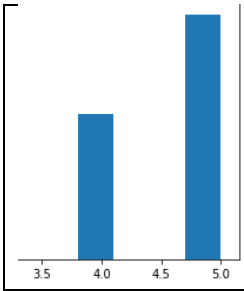
```
#for Dashboard Makeing for Better under standing
df.to_csv("car_procesed.csv")
```

This section for the coustomer who want to by the best car :-

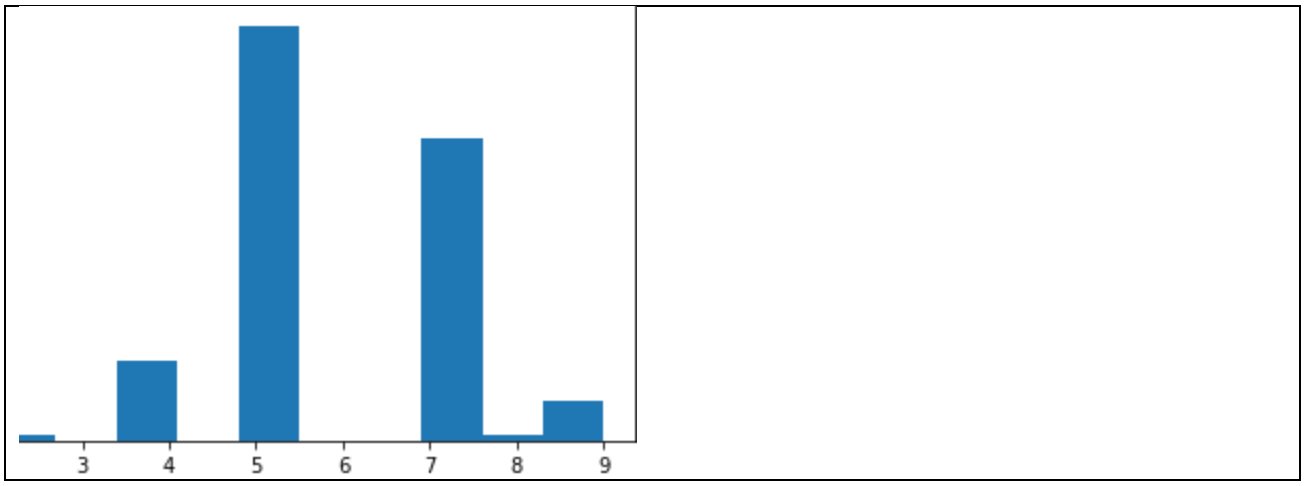
14. The fuel tank capacity Average from all cars is **40 Litters**



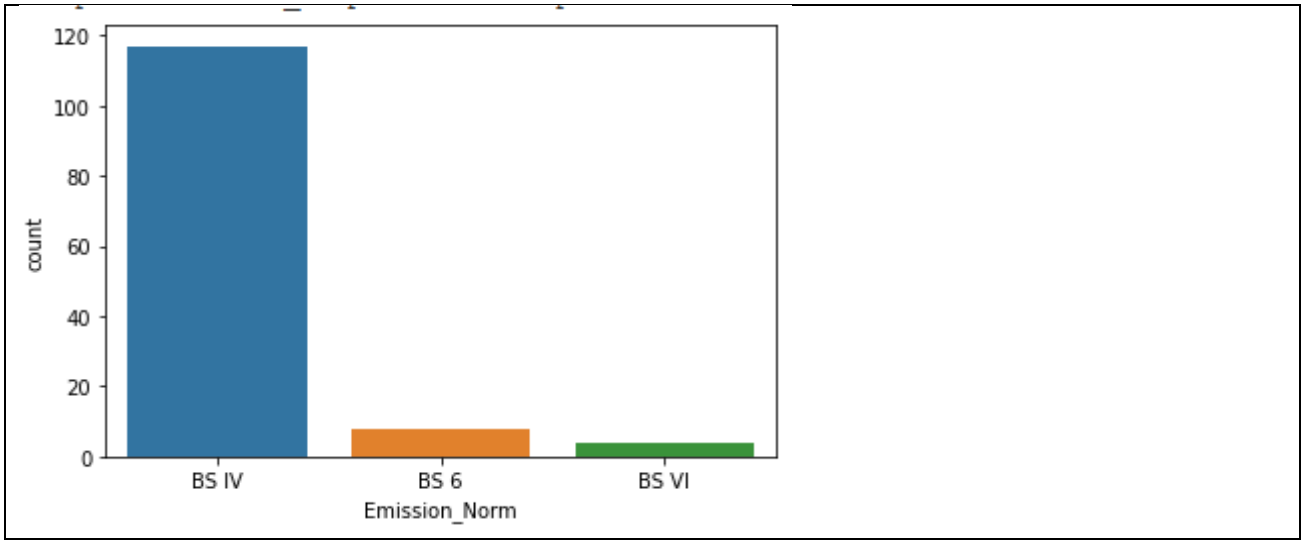
15. Average doors :- are **4 doors +1 back door**



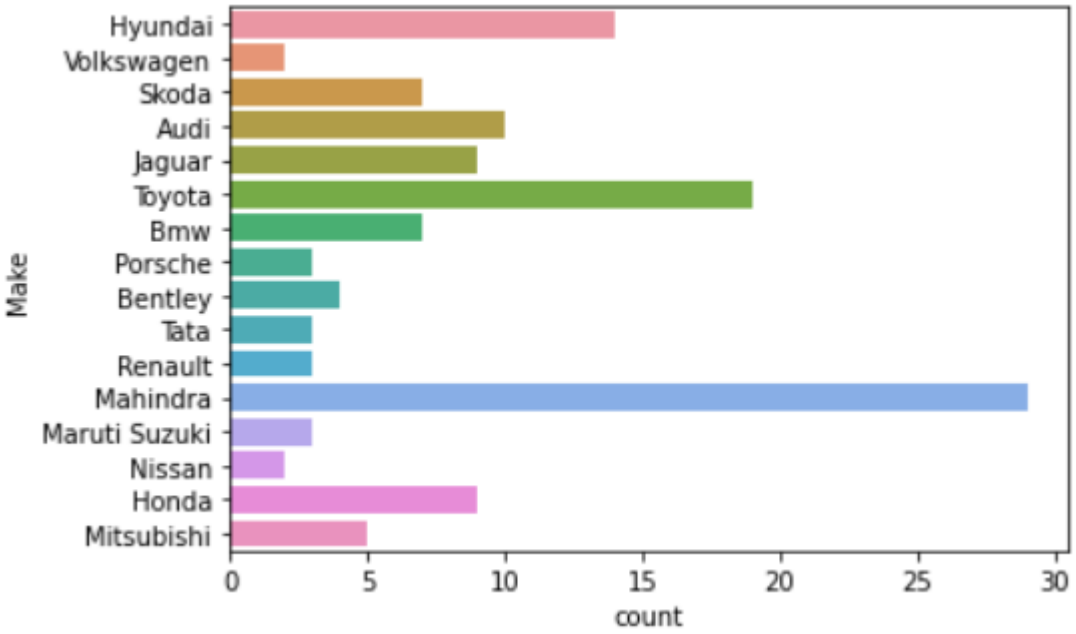
16. Average setting capacity **5**



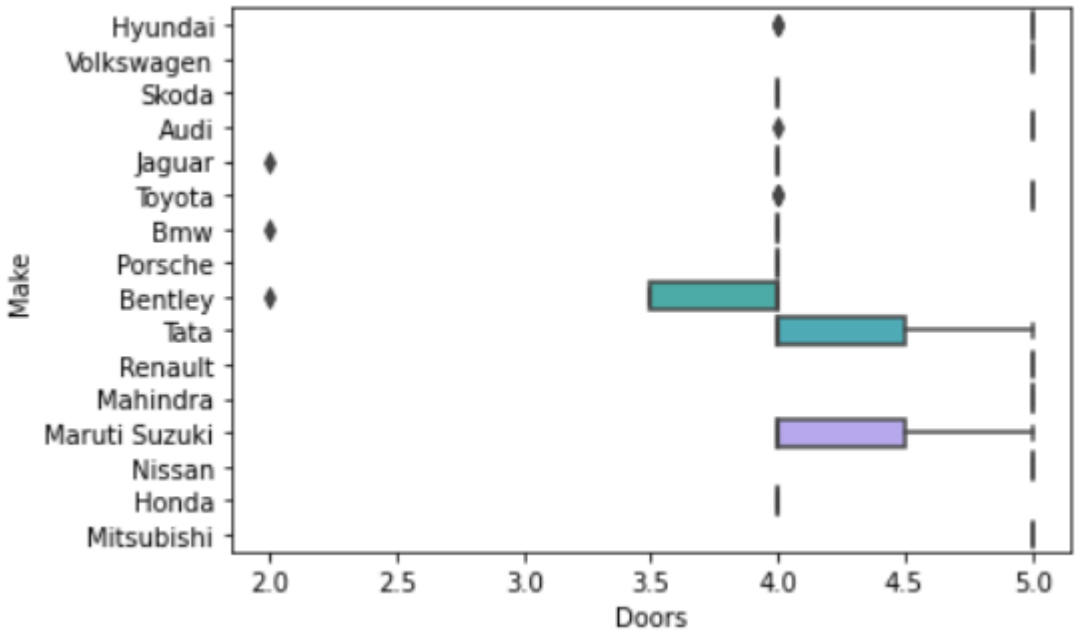
17. Emission_Norm is BS IV is getting in normal cars



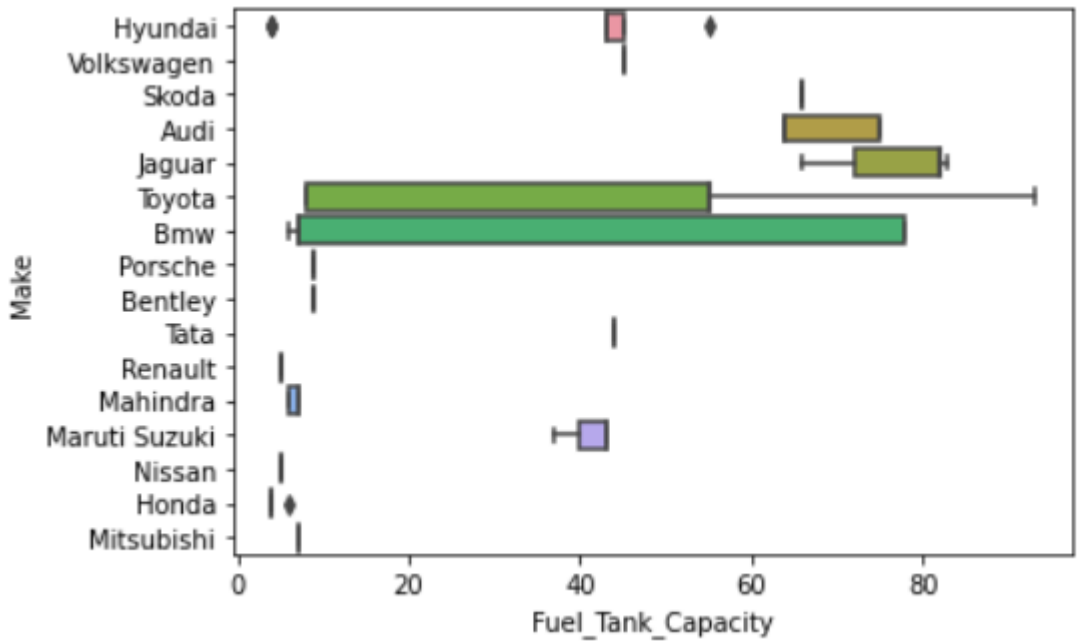
18. How many cares made by company



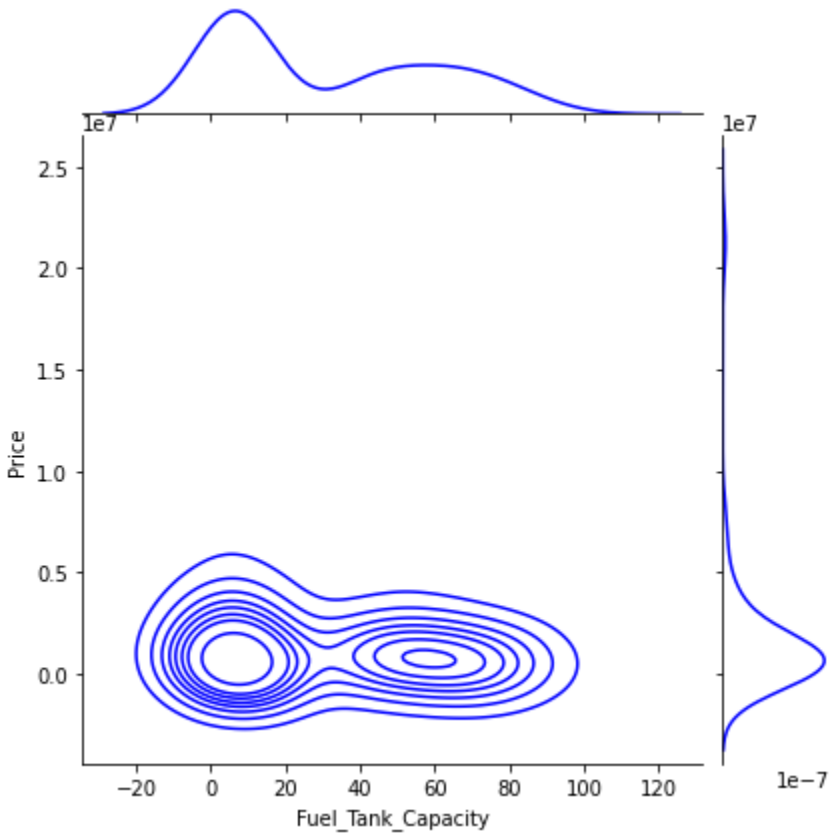
19. Outliers for doors for expansive car 2 doors cars



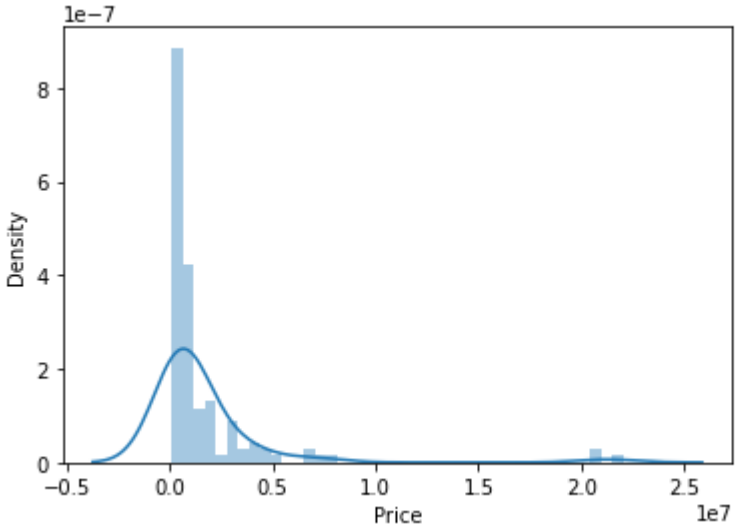
20. Outliers for Fuel tank capacity for expansive cars



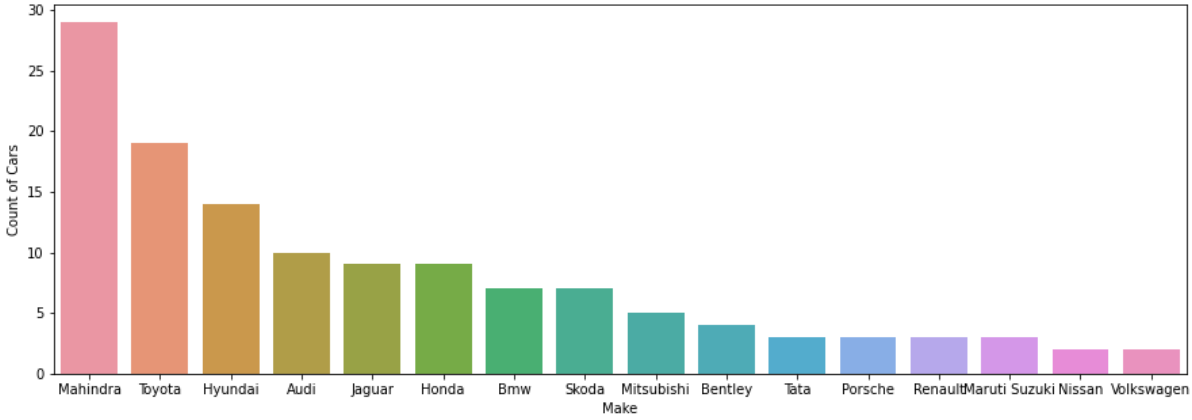
21. For checking maximum cluster points where it lies



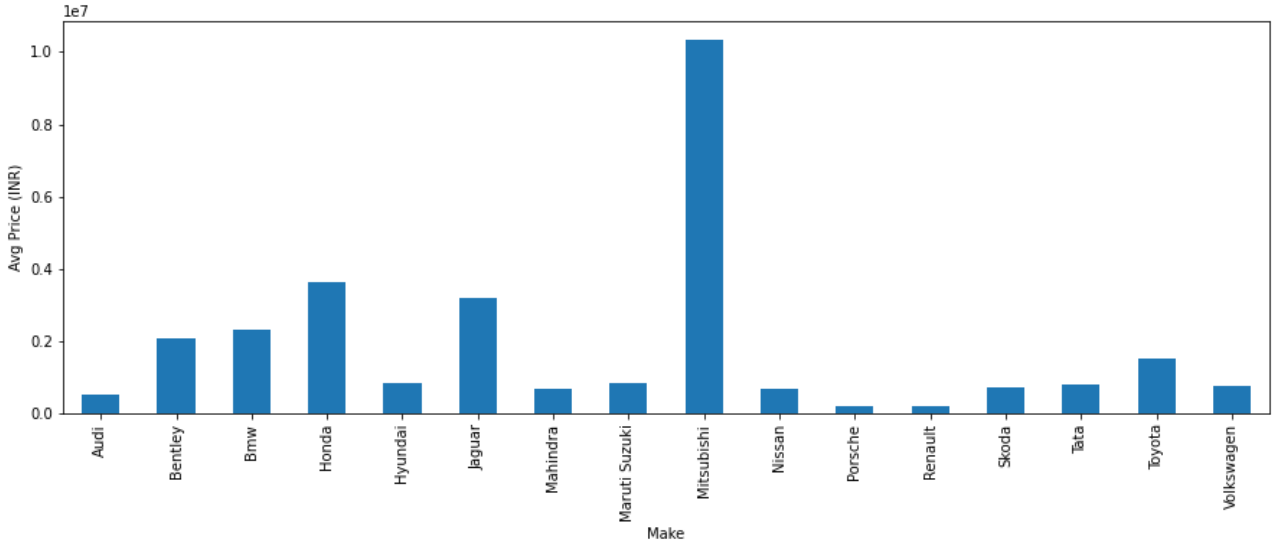
22. Checking the density of price 75% cars in range 0-25 lakhs INR



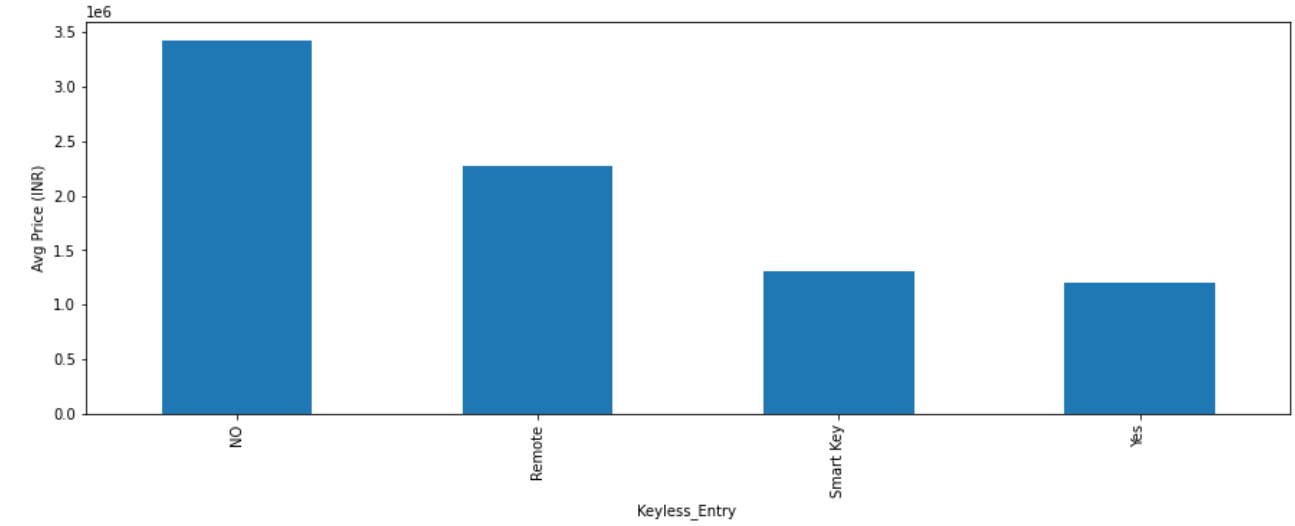
23. Count of cars Per company



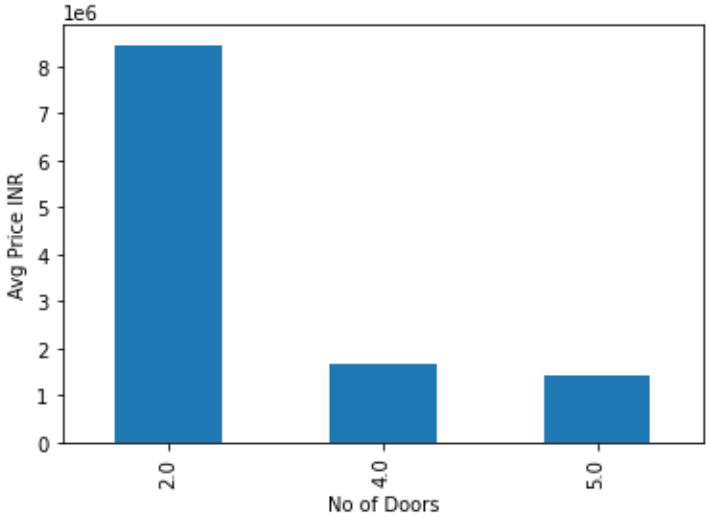
24. **Let's see average car price of each company.**



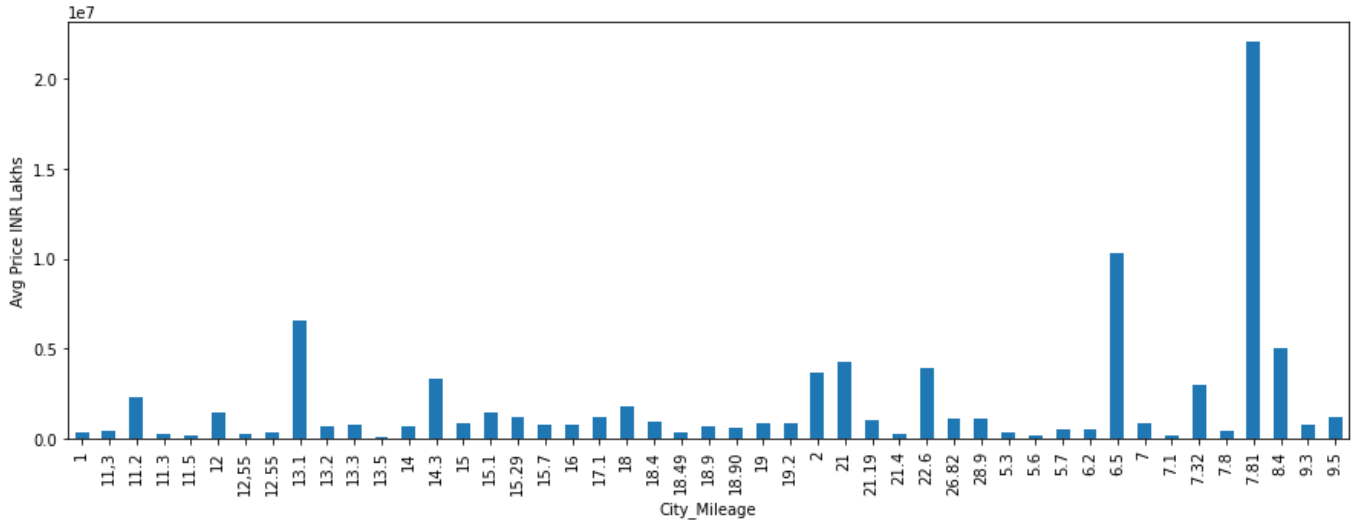
25. **Let's see average car price of each company.**



26. **within the 25 lakh you get 4 or 5 doors car**



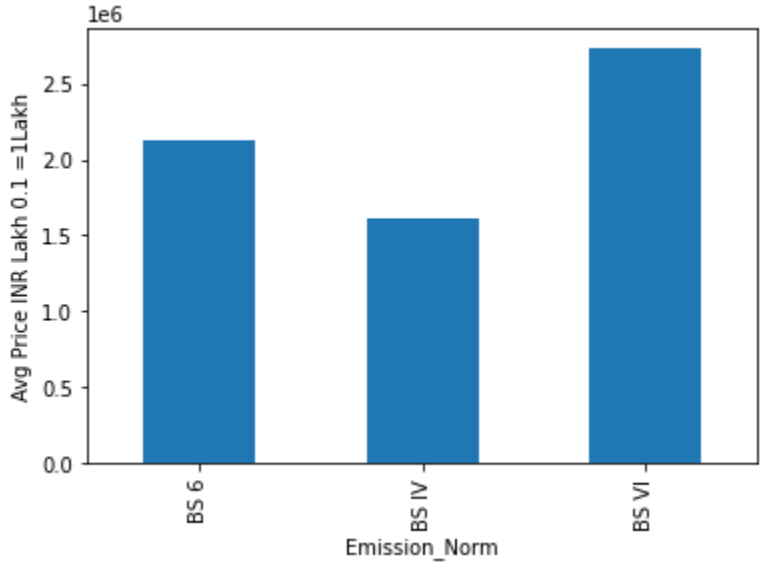
27. Car Price and Care Mileage



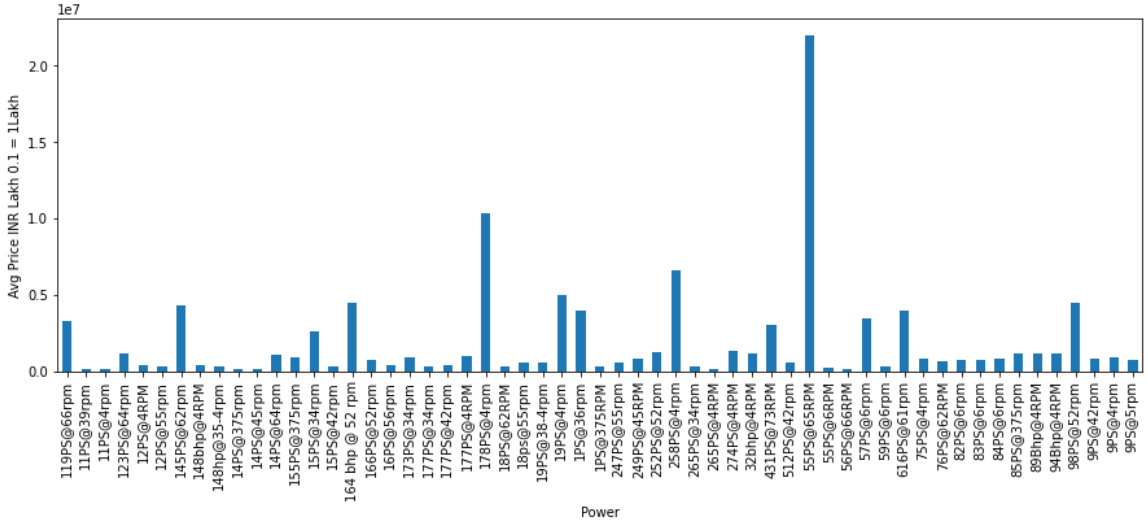
29. <15lakh = BS 4

<20lakh =Bs 5

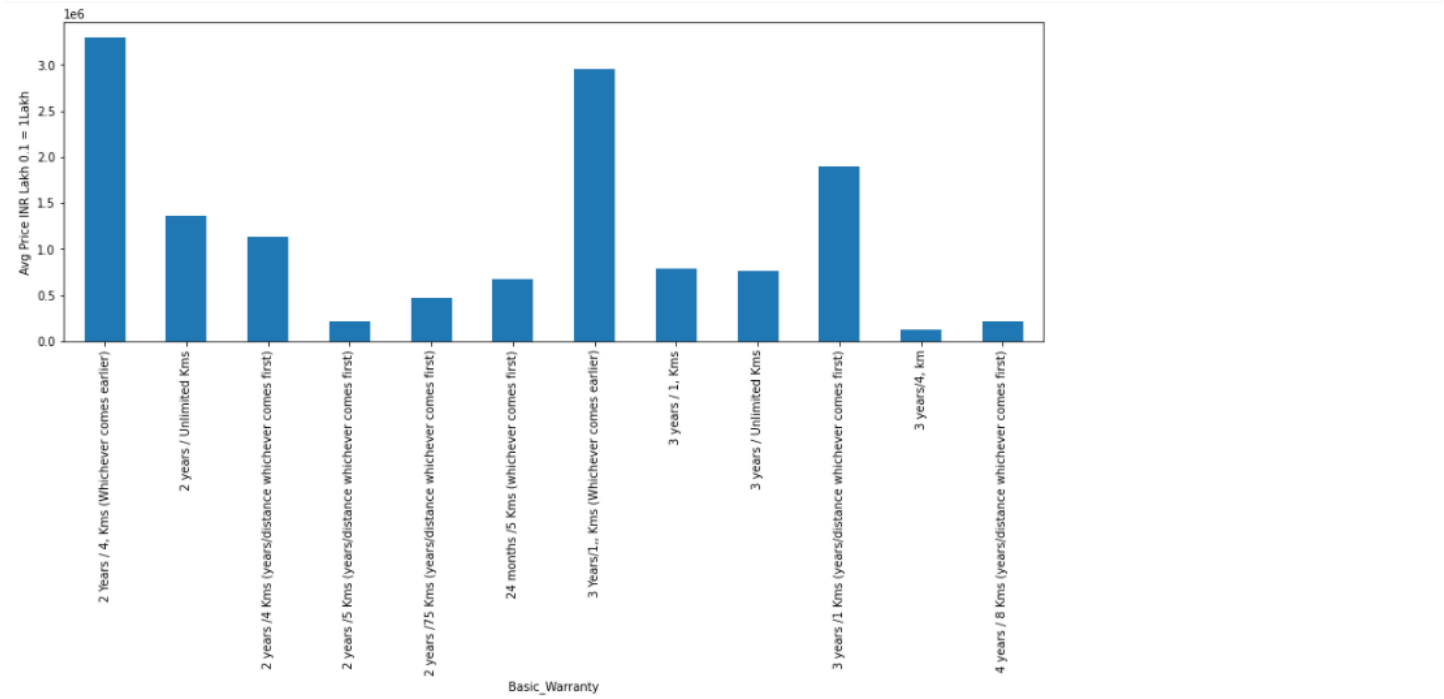
Above 20lakhs = BS 6



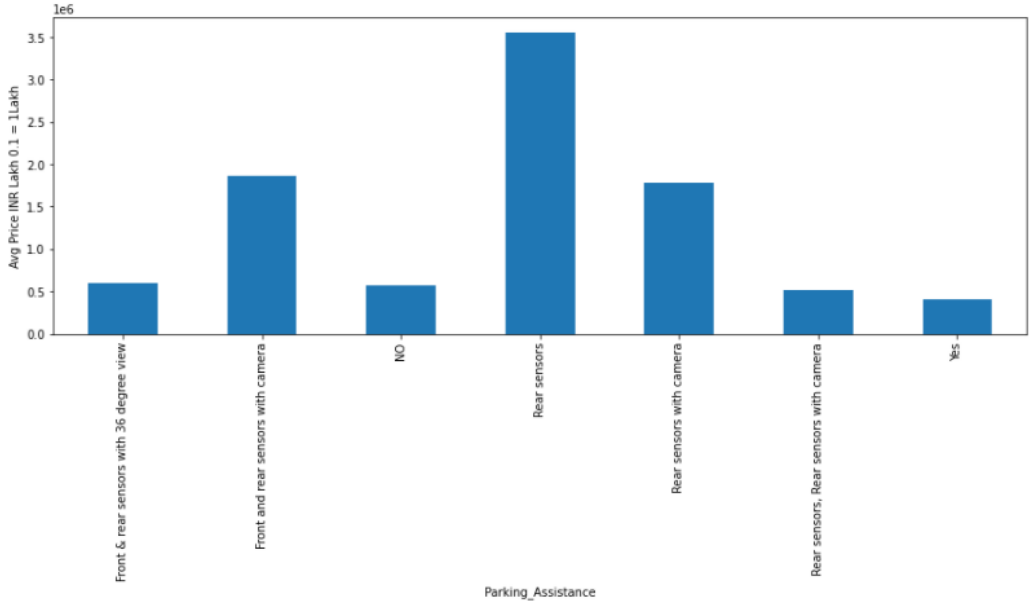
30. Power in price range



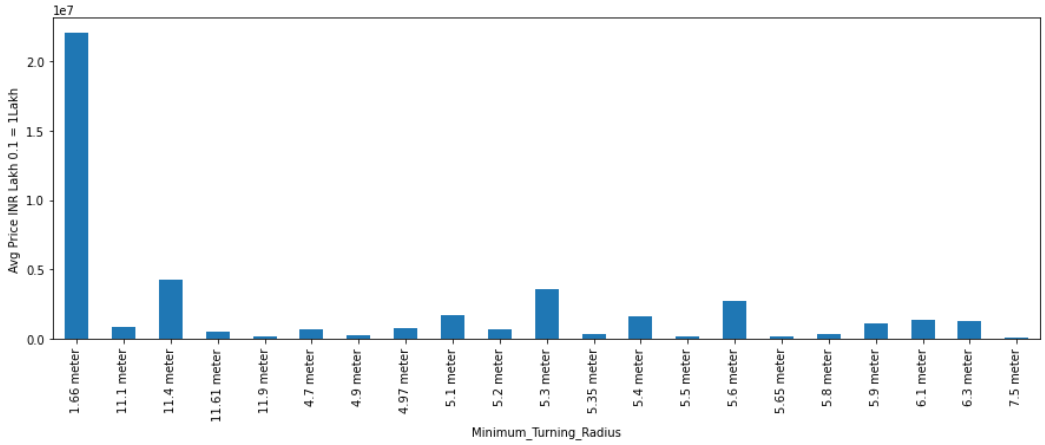
31. warranty of car in price range



32. parking assistance feature in range of price

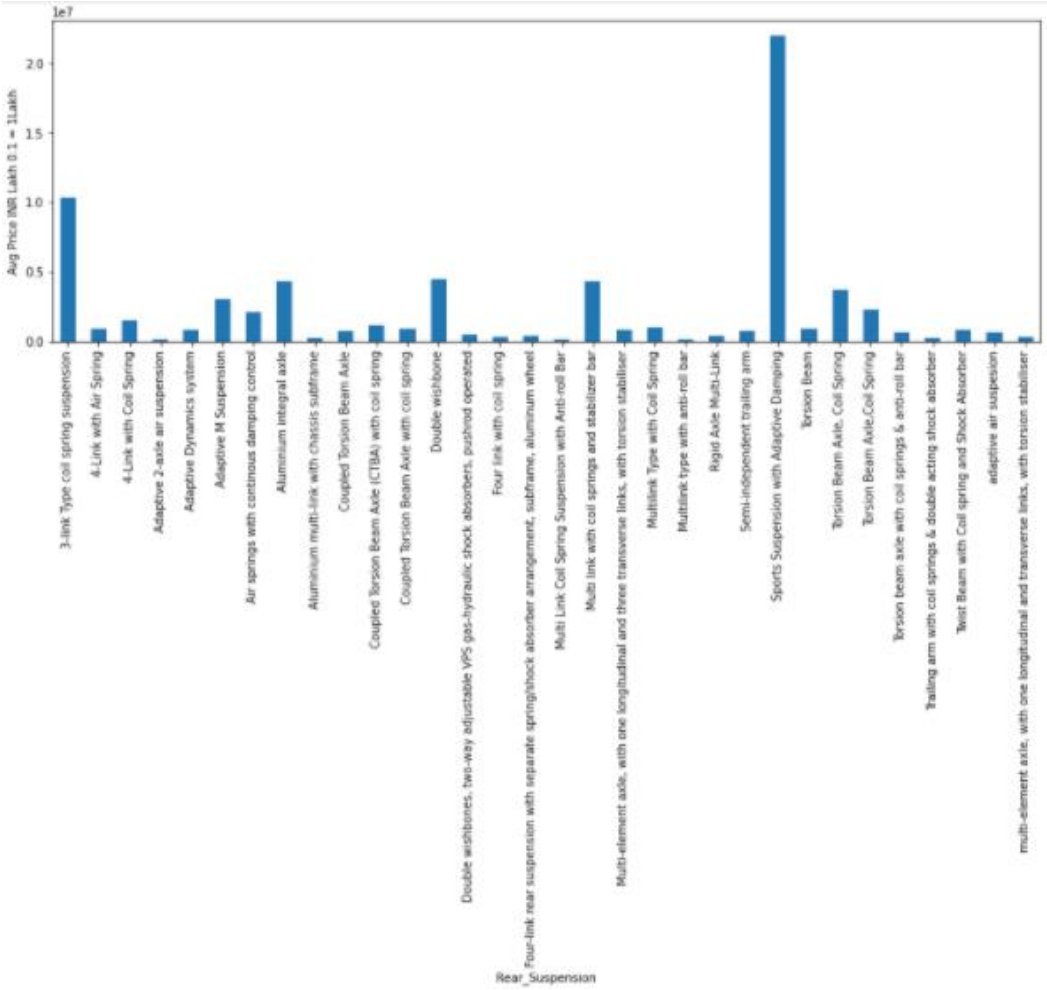


33. minimum turning radius in price of range

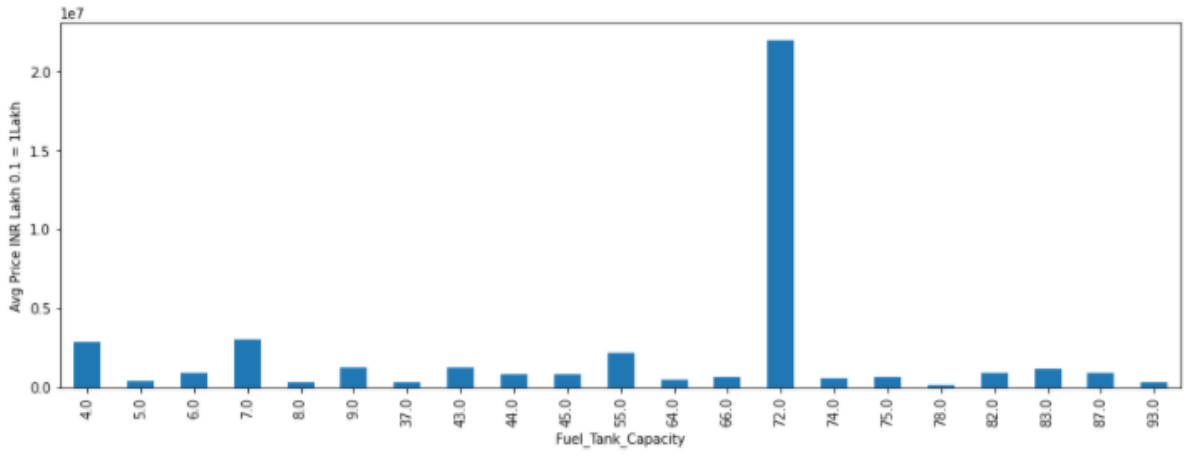


34.

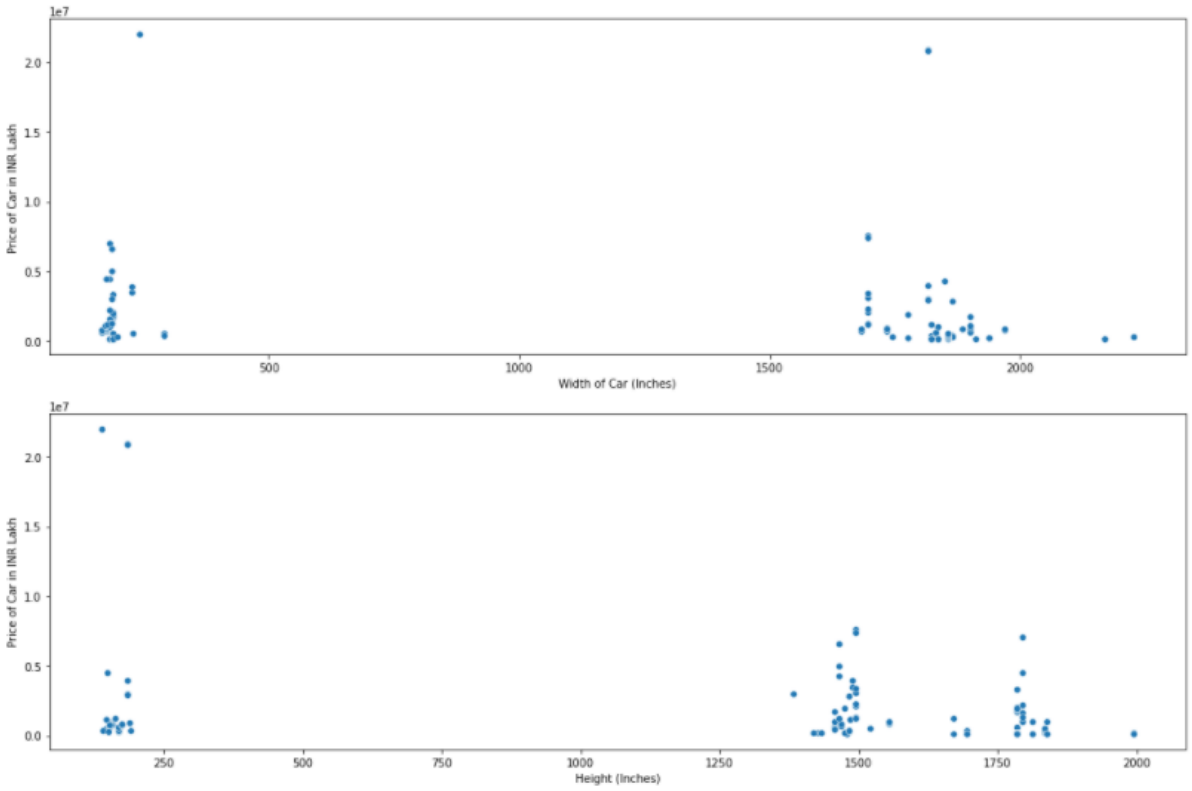
Rear_Suspension



35. **Fuel_Tank_Capacity**

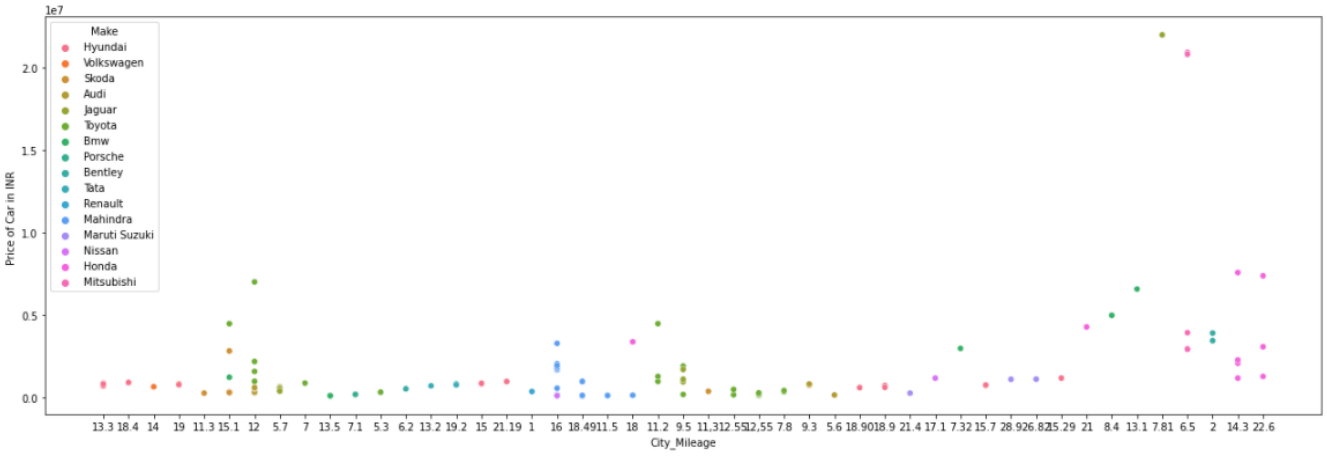


36. Height and width scatter plot



37.

It is expected that luxury brands don't care about mileage. Let's find out how price varies with brand category and mileage.



Model

Linear regression

We use this model for car price prediction base on feature

```
reg=linear_model.LinearRegression()
reg.fit(df[['City_Mileage','Height','Fuel_Tank_Capacity','Width','Seating_Capacity','Power']],df.Price)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

This are the columns :-

```
'City_Mileage','Height','Fuel_Tank_Capacity','Width','Seating_Capacity','Power
'
```

*price= *m1* City_Mileage+m2* Height+.....+m6*Power* +*b* *

1. ***b is Intercept***
2. ***m is Coefficient***

```
#this are the value of m  coefficients
reg.coef_
```

```
array([-1.88662989e+05,  1.18835512e+03, -7.79603574e+02,  3.59015818e+02,
       -2.18863132e+06,  2.01974572e+04])
```

```
#this is intercept
reg.intercept_
```

```
14570200.608719071
```

```
#'City_Mileage','Height','Fuel_Tank_Capacity','Width','Seating_Capacity','Po
wer'
reg.predict([[20,160,30,140,4,55]])
```

```
array([3370286.62036439])
```

Send this model for Production

```
[ ]  pickle.dump(reg,open('CarPricePrediction.pkl','wb'))
```

Split the data in to Train and test 20 80

Before we make the model, we need to split the data into train dataset and test dataset. We will use the train dataset to train the linear regression model. The test dataset will be used as a comparasion and see if the model get overfit and can not predict new data that hasn't been seen during training phase. We will 80% of the data as the training data and the rest of it as the testing data.

```
from sklearn.model_selection import train_test_split
x_train, x_test ,y_train ,y_test=train_test_split(x,y, test_size=0.2)
```

```
x=mdf.drop(columns='Price')
y=mdf['Price']
```

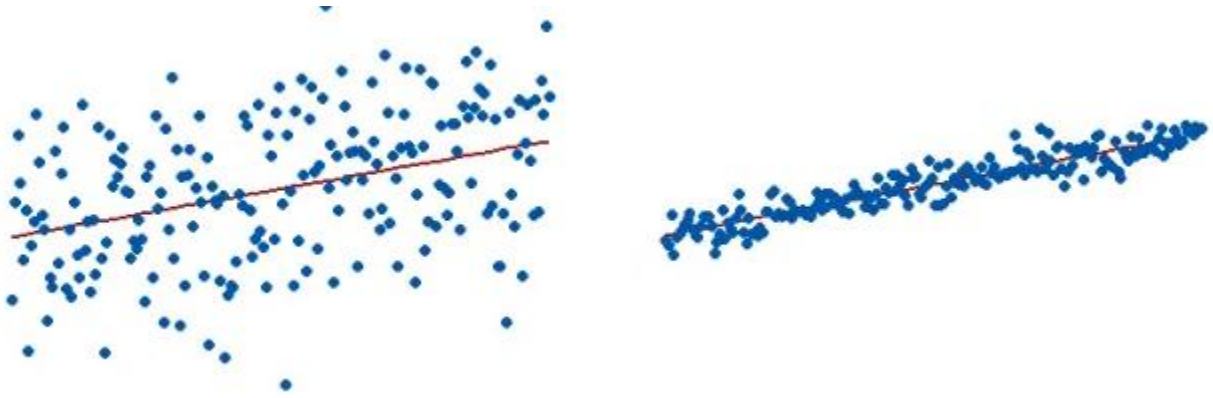
Price is save in Y

```
[ ]  from sklearn.linear_model import LinearRegression
     from sklearn.metrics import r2_score
     from sklearn.preprocessing import OneHotEncoder
     from sklearn.compose import make_column_transformer
     from sklearn.pipeline import make_pipeline
```

We use one hot encoder for the R2 soures

For improve the R2 score we run the model 10 times and store that results for best fit .

R2 = variance /total variance



```
#it gives 0.6 r2 scrore  i want max score = 0.6
for i in range(10):
  x_train, x_test ,y_train ,y_test=train_test_split(x,y, test_size=0.2)
  lr=LinearRegression()
  pipe=make_pipeline(column_trans,lr)
  pipe.fit(x_train,y_train)
  y_pred=pipe.predict(x_test)
  print(r2_score(y_test,y_pred),i)
```

```
0.37562704204581066 0
-0.3219701803977284 1
0.08218131299399956 2
0.6150371293408451 3
0.21981042230290904 4
0.3243474225688615 5
0.4786418500945152 6
-0.3697549839365415 7
0.4845678844371277 8
-0.49846448606386806 9
```

We cat'n get max score so we run it 100 times

```
scores=[]
for i in range(100):
  x_train, x_test ,y_train ,y_test=train_test_split(x,y, test_size=0.2)
  lr=LinearRegression()
  pipe=make_pipeline(column_trans,lr)
  pipe.fit(x_train,y_train)
  y_pred=pipe.predict(x_test)
  scores.append(r2_score(y_test,y_pred))
```

```
#the max score in stores
np.argmax(scores)
```

```
79
```

```
scores[79]
```

```
0.6195946175865271
```

It is going for Production

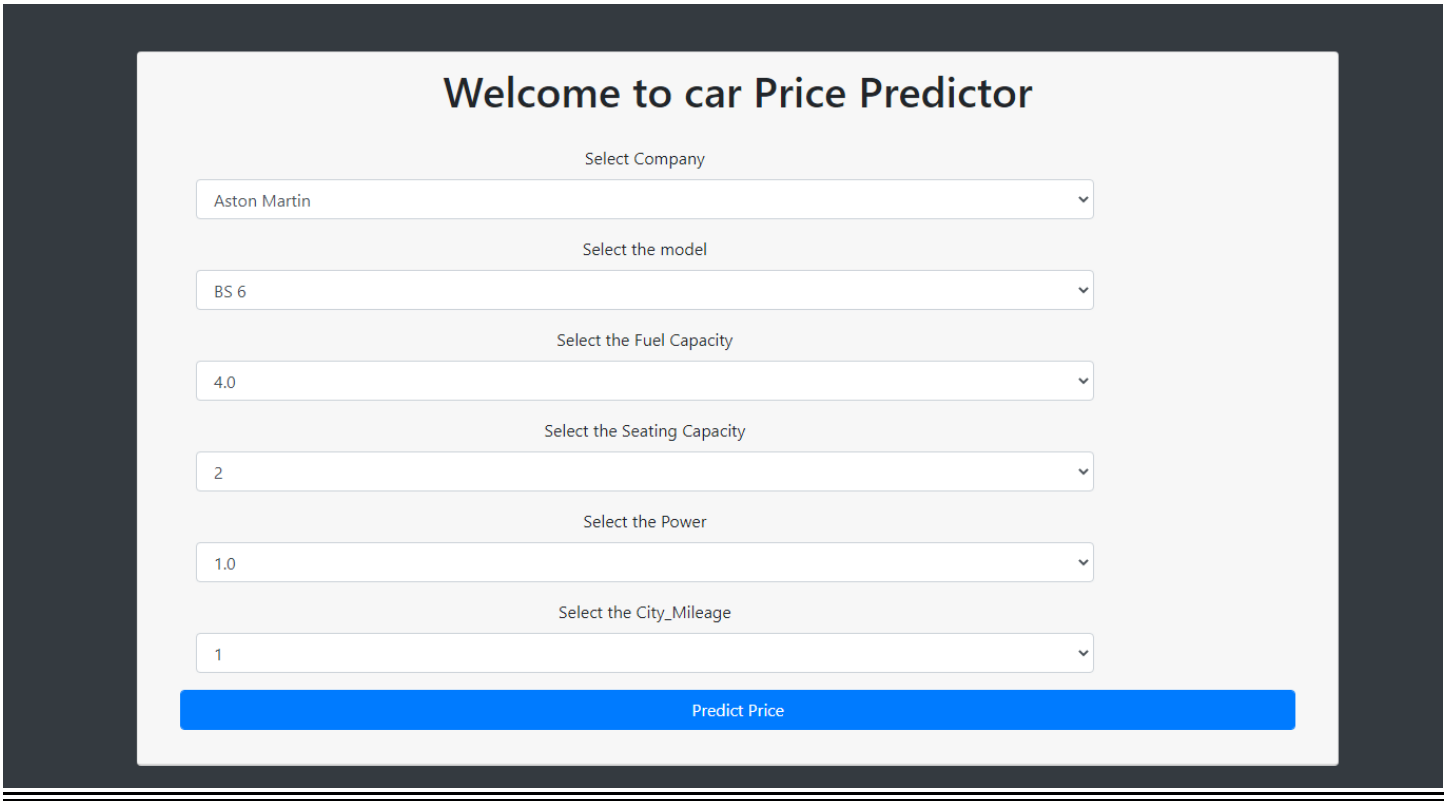
```
pickle.dump(pipe,open('LinearRegressionModelforCarPrice.pkl','wb'))
```

Conclusion

Variables that are useful to describe the variances in car prices `'City_Mileage','Height','Fuel_Tank_Capacity','Width','Seating_Capacity','Power'` are Our final model has satisfied the classical assumptions. The R-squared of the model is high, with 60.72% of the variables can explain the variances in the car price. The accuracy of the model in predicting the car price suggesting that our model may overfit the traning dataset.

We have already learn how to build a linear regression model and what need to be concerned when building the model.

Web Application for Prediction



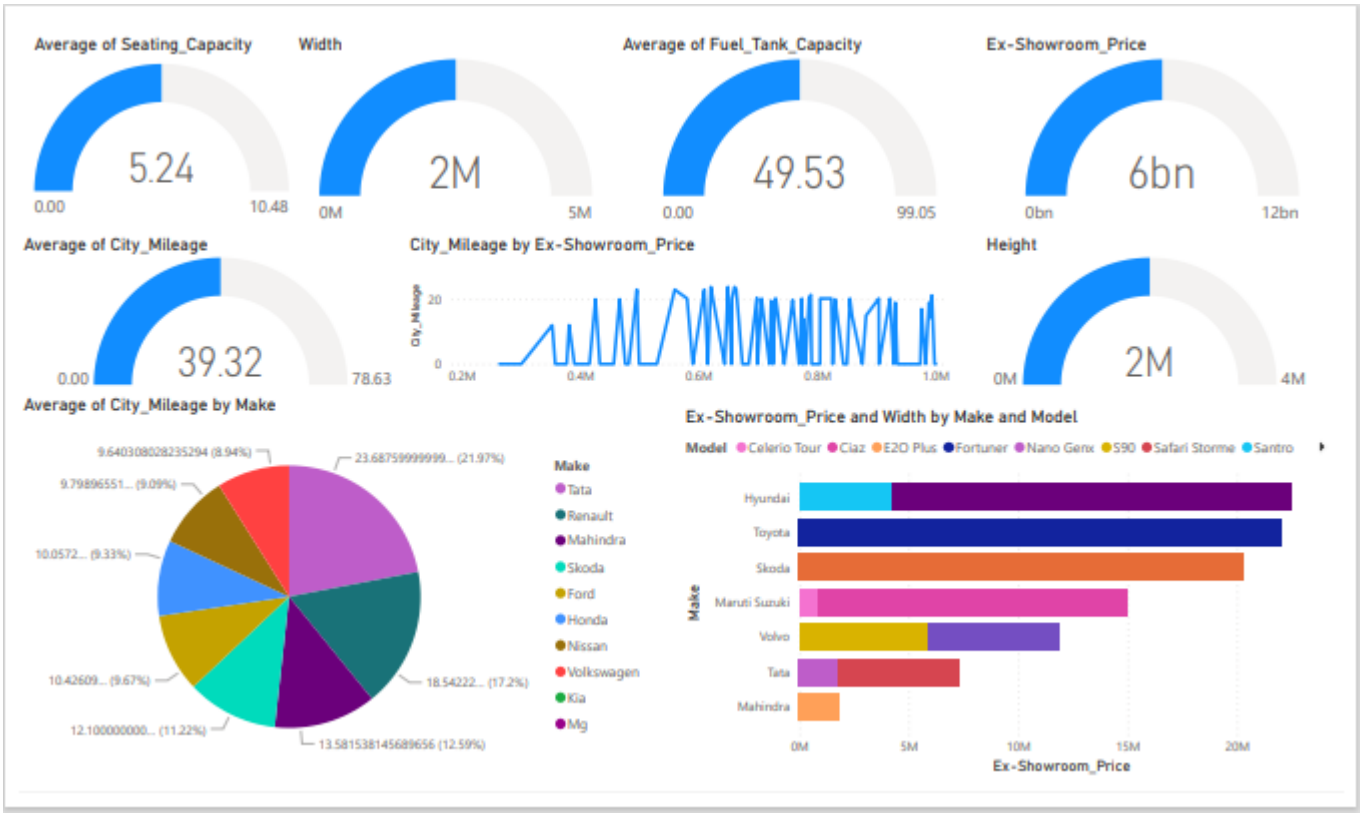
Reading .CSV Files
Using panda libraries in Django Framework

```python
from django.shortcuts import render
import pandas as pd

# Create your views here.
car = pd.read_csv("H:/car_prediction/car_predict/templates/cleand_carData.csv")


def index(request):
    companies = sorted(car['Make'].unique())
    models = sorted(car['Emission_Norm'].unique())
    Fuel_Tank_Capacity = sorted(car['Fuel_Tank_Capacity'].unique())
    Seating_Capacity = sorted(car['Seating_Capacity'].unique())
    Power = sorted(car['Power'].unique())
    City_Mileage = sorted(car['City_Mileage'].unique())
    value = {'companies': companies,
             'models': models,
             'Fuel_Tank_Capacity': Fuel_Tank_Capacity,
             'Seating_Capacity': Seating_Capacity,
             'Power': Power,
             'City_Mileage': City_Mileage,


             }
    print(companies,models,Fuel_Tank_Capacity,Seating_Capacity,Power,City_Mileage)
```

Output of .CSV Files
Using panda libraries in Django Framework

```
ystem check identified no issues (0 silenced).

ou have 19 unapplied migration(s). Your project may not work properly until you apply the m
grations for app(s): admin, auth, car_predict, contenttypes, sessions.
un 'python manage.py migrate' to apply them.
pril 23, 2021 - 08:05:48
jango version 3.1.6, using settings 'car_prediction.settings'
tarting development server at http://127.0.0.1:8000/
uit the server with CTRL-BREAK.
'Aston Martin', 'Audi', 'Bentley', 'Bmw', 'Datsun', 'Dc', 'Ferrari', 'Fiat', 'Force', 'Ford
, 'Honda', 'Hyundai', 'Isuzu', 'Jaguar', 'Jeep', 'Lamborghini', 'Land Rover Rover'
 'Lexus', 'Mahindra', 'Maruti Suzuki', 'Maserati', 'Mini', 'Mitsubishi', 'Nissan', 'Porsche
, 'Premier', 'Renault', 'Skoda', 'Tata', 'Toyota', 'Volkswagen', 'Volvo'] ['BS 6',
'BS III', 'BS IV', 'BS VI'] [4.0, 5.0, 6.0, 6.9, 7.0, 8.0, 9.0, 9.5, 15.0, 24.0, 28.0, 32.0
 35.0, 36.0, 37.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 48.0, 51.0, 52.0, 54.0, 55.
, 56.0, 57.0, 61.0, 62.0, 63.0, 64.0, 65.0, 66.0, 67.0, 68.0, 71.0, 72.0, 73.0, 74.0, 75.0,
76.0, 78.0, 82.0, 83.0, 85.0, 86.0, 87.0, 88.0, 92.0, 93.0, 93.5] [2, 4, 5, 6, 7,
, 9] [1.0, 2.0, 3.0, 7.0, 8.0, 9.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 18.0, 19.0, 22.0, 2
.0, 25.0, 32.0, 34.7, 35.0, 38.0, 41.0, 43.0, 45.0, 46.0, 47.0, 48.0, 54.0, 55.0,
6.0, 57.0, 58.0, 59.0, 60.0, 61.0, 62.0, 63.0, 63.9, 64.0, 65.0, 67.0, 68.0, 69.0, 71.0, 72
0, 73.0, 74.0, 75.0, 76.0, 76.6, 77.0, 78.0, 79.0, 81.0, 82.0, 83.0, 84.0, 84.3, 8
.0, 86.0, 88.4, 89.0, 93.0, 94.0, 95.0, 96.0, 98.0, 99.0, 100.0, 114.0, 119.0, 121.0, 123.0
 126.0, 128.0, 141.0, 143.0, 145.0, 148.0, 152.0, 155.0, 156.0, 164.0, 166.0, 173.
, 174.0, 177.0, 178.0, 185.0, 192.0, 225.0, 231.0, 233.0, 235.0, 245.0, 247.0, 248.0, 249.0
```

Dashboard


Average of Seating_Capacity
5.24
0.00
10.48
Width
2M
0M
5M
Average of Fuel_Tank_Capacity
49.53
0.00
99.05
Ex-Showroom_Price
6bn
0bn
12bn
Average of City_Mileage
39.32
0.00
78.63
City_Mileage by Ex-Showroom_Price
City_Mileage
20
0
0.2M
0.4M
0.6M
0.8M
1.0M
Height
2M
0M
4M
Average of City_Mileage by Make
9.640308028235294 (8.94%)
23.68759999999... (21.97%)
9.79896551... (9.09%)
10.0572... (9.33%)
10.42609... (9.67%)
18.54222... (17.2%)
12.100000000... (11.22%)
13.581538145689656 (12.59%)
Make
Tata
Renault
Mahindra
Skoda
Ford
Honda
Nissan
Volkswagen
Kia
Mg
Ex-Showroom_Price and Width by Make and Model
Model
Celerio Tour
Ciaz
E2O Plus
Fortuner
Nano Genx
S90
Safari Storme
Santro
Hyundai
Toyota
Skoda
Maruti Suzuki
Volvo
Tata
Mahindra
Make
0M
5M
10M
15M
20M
Ex-Showroom_Price

Reference

1. Aora, Ridhi. Answering question in “What are the ways to deal with auto-correlation problems in Multiple Regression Analysis?”. Accessed September 23 2019, via https://www.researchgate.net/post/What_are_the_ways_to_deal_with_auto-correlation_problems_in_Multiple_Regression_Analysis.
2. James, Gareth, Witten, Daniela, Hastie, Trevor, and Tibshirani, Robert. 2013. *An Introduction to Statistical Learning: with Applications in R* . New York: Springer.
3. Gujarati, Damodar and Dawn C. Porter. 2009. *Basic Econometrics* . New York: McGraw-Hill.
4. A Tour of Machine Learning Algorithms

5. https://statisticsbyjim.com/regression/interpret-r-squared-regression/