FLIP ROBO

CAR PRICE PREDICTION PROJECT

SUBMITTED BY:

Kundan

ACKNOWLEDGEMENT

The car price prediction project is been completed through the aid of many factors, we have searched over the web for appropriate websites in order to collect the latest data, In that process we landed up in finding website car trade.com where we got almost 5000 data which belongs to almost 180 different models of car along different regions of India, next in the row we have Droom.com where we had collected almost 500 cars information, followed by car dheko.com where we collected almost same 500 cars information.

In the process of making an informative, productive project we traversed through many websites out of which towardsdatascience, geeksforgeeks, stackoverflow are significant one, We also got required aid by flirobo and datatrained company.

INTRODUCTION

- **Business Problem Framing**

  With the covid 19 impact in the market, we have seen lot of changes in the car market. Now some cars are in demand hence making them costly and some are not in demand hence cheaper. One of our clients works with small traders, who sell used cars. With the change in market due to covid 19 impact, our client is facing problems with their previous car price valuation machine learning models. So, they are looking for new machine learning models from new data. We have to make car price valuation model through new data which is going to predict the price of used cars.

- **Conceptual Background of Domain Problem**

  First of all we should be knowing how to scrap the required data from various websites through the techniques of webscrapping, In order to do that we should extract all the urls of each car using relevant xpaths , we should be knowing how to build a linear regression model through collected data.

- **Literature Review**

  To get the enough amount of relevant data we have to crawl through google to get perfect website, In that process we ended up landing in car trade.com,droom India and Quickr.

  We have listed out 8 such websites from which we can get the adequate information, They are as follows Quickr,Indian Outdoor.com, Ola India, Car dhekho, Olx, Cars24, Droom India, Carwale,Car Trade India.

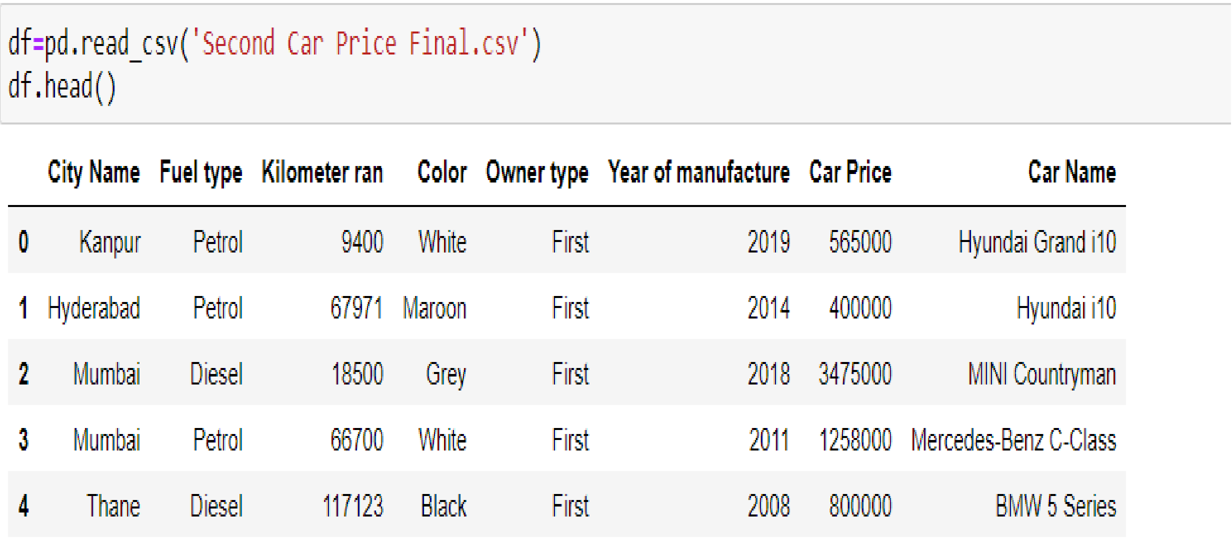
- **Motivation for the Problem Undertaken**

  The motivation for this project is to provide our client with new model to excel in their business and also to upgrade myself by solving the real world problem through virtual environment.

Analytical Problem Framing

- ## Mathematical and Analytical understanding of the problem

  The data we collected was in in terms of rows and columns, there are 8 columns and 5993 rows out of which 3 columns where of integer type and remaining are object type, so in order to make the machine understands this we have to convert the categorical value to numerical values we accomplish it through label encoding or one hot encoding. We did scaling of the dataset in order to have values of variables within certain limits so that machine can perform better, for this purpose we used different types of scaling namely standard scaling, min max scaling and Robust scaler, later we reduce the skewness using various techniques.

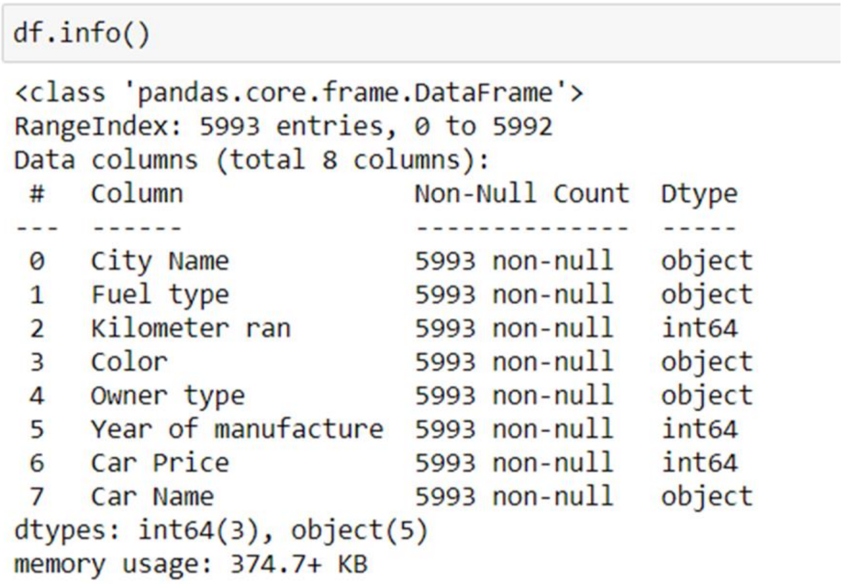


```
df=pd.read_csv('Second Car Price Final.csv')
df.head()
```

	City Name	Fuel type	Kilometer ran	Color	Owner type	Year of manufacture	Car Price	Car Name
0	Kanpur	Petrol	9400	White	First	2019	565000	Hyundai Grand i10
1	Hyderabad	Petrol	67971	Maroon	First	2014	400000	Hyundai i10
2	Mumbai	Diesel	18500	Grey	First	2018	3475000	MINI Countryman
3	Mumbai	Petrol	66700	White	First	2011	1258000	Mercedes-Benz C-Class
4	Thane	Diesel	117123	Black	First	2008	800000	BMW 5 Series

- ## Data Source and Their Format

  The major portion of the data we collected from car trade.com , there are 8 major parameters and 5993 rows out which 3 are of integers type column and remainings are of categorical type columns.



```
df.info()
```

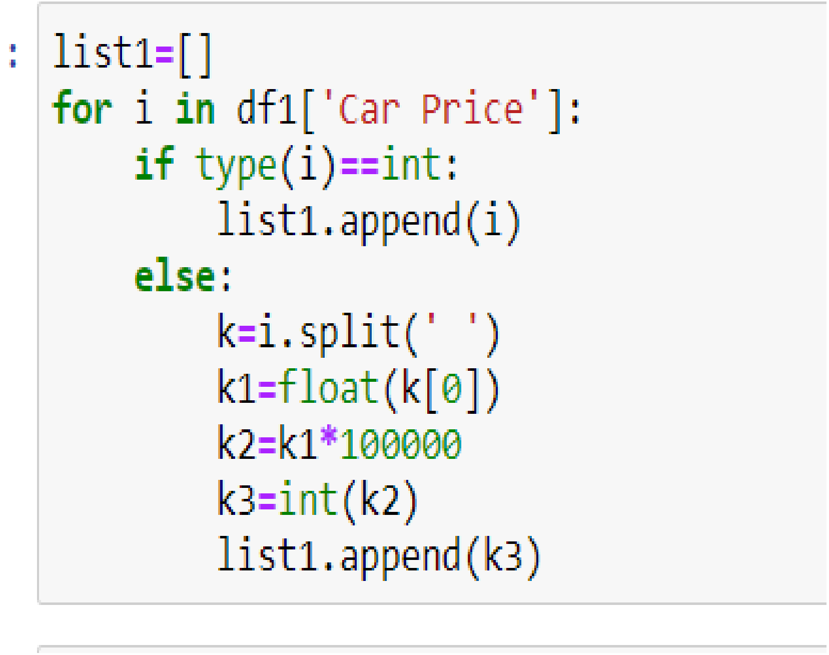
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5993 entries, 0 to 5992
Data columns (total 8 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   City Name            5993 non-null   object
 1   Fuel type            5993 non-null   object
 2   Kilometer ran        5993 non-null   int64 
 3   Color                5993 non-null   object
 4   Owner type           5993 non-null   object
 5   Year of manufacture  5993 non-null   int64 
 6   Car Price            5993 non-null   int64 
 7   Car Name             5993 non-null   object
dtypes: int64(3), object(5)
memory usage: 374.7+ KB
```

- **Data Processing**

There were enough amount of anomalies in the data, we looked column by column for anomalies and we cleaned each column separately , through the aid of pandas, loops and regular expressions we achieved it.

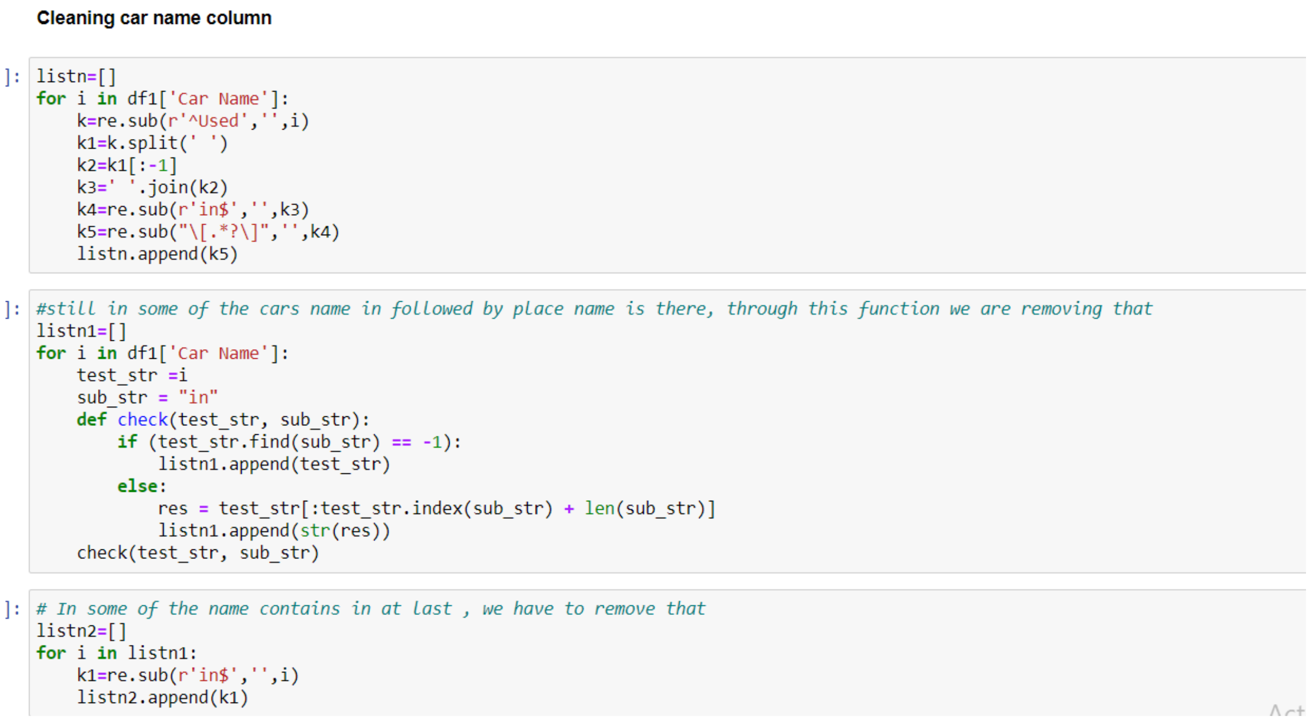
1.We imported the collected data to a separate notebook which was meant for data preprocessing .

2.The car price column contains the data which were of strings type and we adopted regular expressions to clean it out.



```
:  list1=[]
   for i in df1['Car Price']:
       if type(i)==int:
           list1.append(i)
       else:
           k=i.split(' ')
           k1=float(k[0])
           k2=k1*100000
           k3=int(k2)
           list1.append(k3)
```

3.The Name of Car column contained a string out of which we have to extract only name of the car, it was a challenging one and we accomplished it through regular expressions.



Cleaning car name column

```
]: listn=[]
   for i in df1['Car Name']:
       k=re.sub(r'^Used','',i)
       k1=k.split(' ')
       k2=k1[:-1]
       k3=' '.join(k2)
       k4=re.sub(r'in$','',k3)
       k5=re.sub("\[.*?\]",'',k4)
       listn.append(k5)
```

```
]: #still in some of the cars name in followed by place name is there, through this function we are removing that
   listn1=[]
   for i in df1['Car Name']:
       test_str =i
       sub_str = "in"
       def check(test_str, sub_str):
           if (test_str.find(sub_str) == -1):
               listn1.append(test_str)
           else:
               res = test_str[:test_str.index(sub_str) + len(sub_str)]
               listn1.append(str(res))
       check(test_str, sub_str)
```

```
]: # In some of the name contains in at last , we have to remove that
   listn2=[]
   for i in listn1:
       k1=re.sub(r'in$','',i)
       listn2.append(k1)
```

- ## Data Input-Logic-Output Relationships

  Data Input : It was a precise dataset, which has been scaled having low skewness and very minimal outliers.

  Logic : The logic here is linear regression algorithm which predict the response variable, the linear regression algorithm we used in this case are

  1.Linear Regression
  2,Lasso Regression
  3.Ridge Regression
  4.ElasticNet
  5.Ransac Regressor
  6.Support vector Regressor
  7.Random Forest Regressor

  Data output : We got a model predicting the price of second hand car

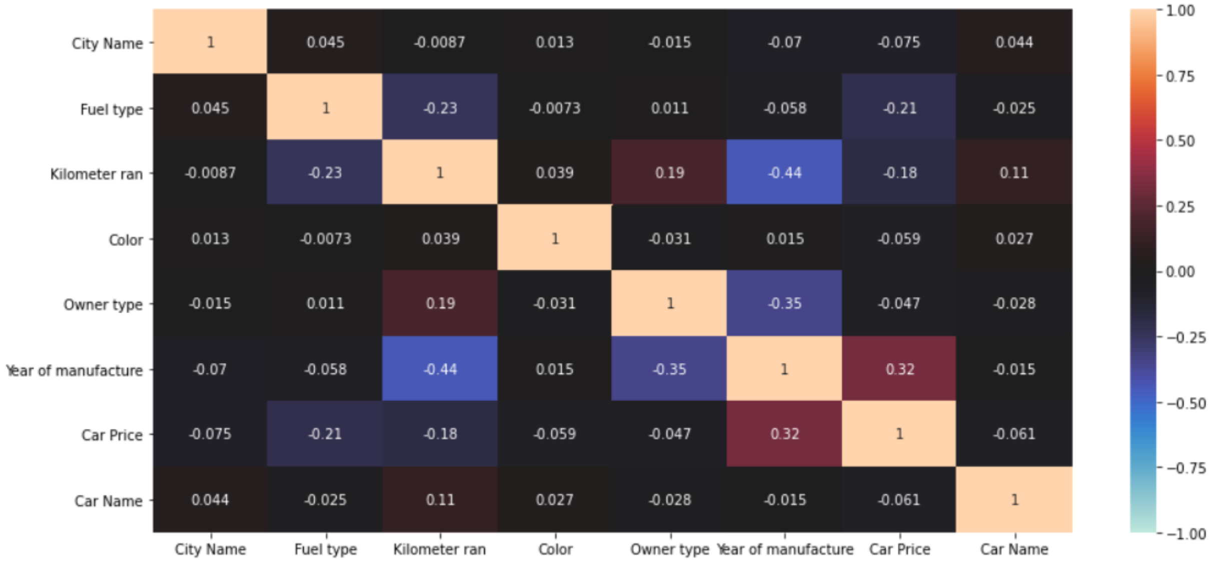
- ## Hardware and software tools used

  Hardware : We used the hardware of 8GB RAM,1Tb ROM and i5 processor.

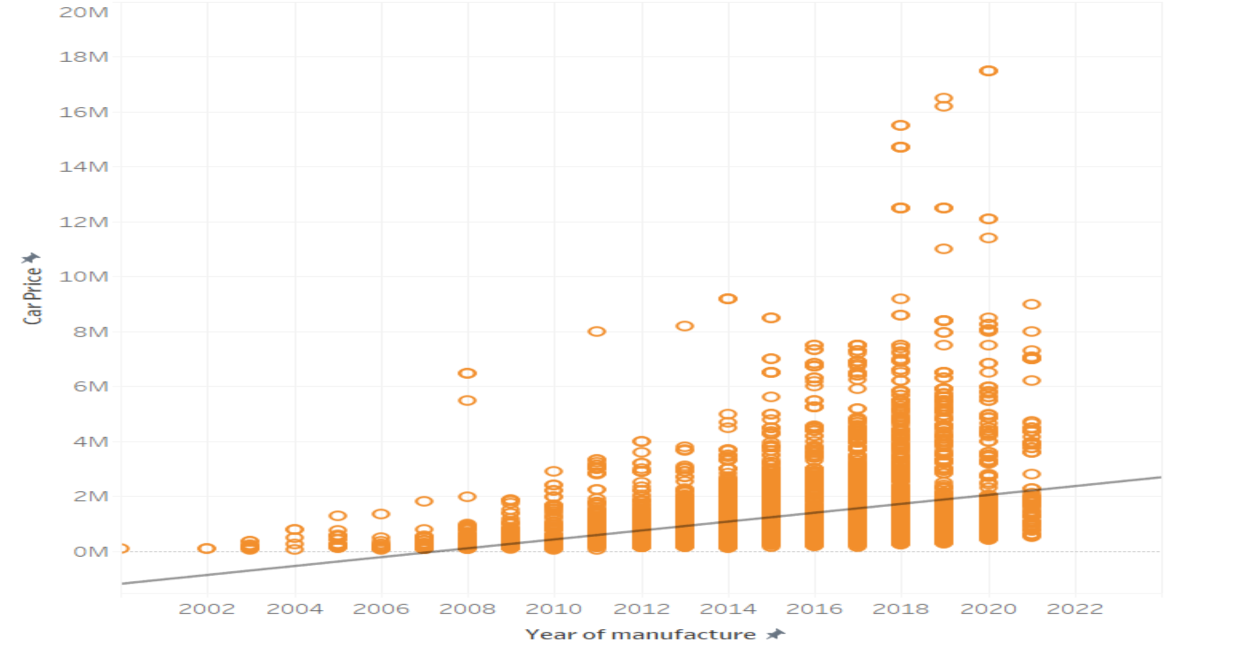
  Software : For better visualization we used Tableau Public, Jupyter notebook from anaconda navigator for coding and webscrapping and Microsoft word and Power Point Presentation for creating the report and making the presentation respectively.

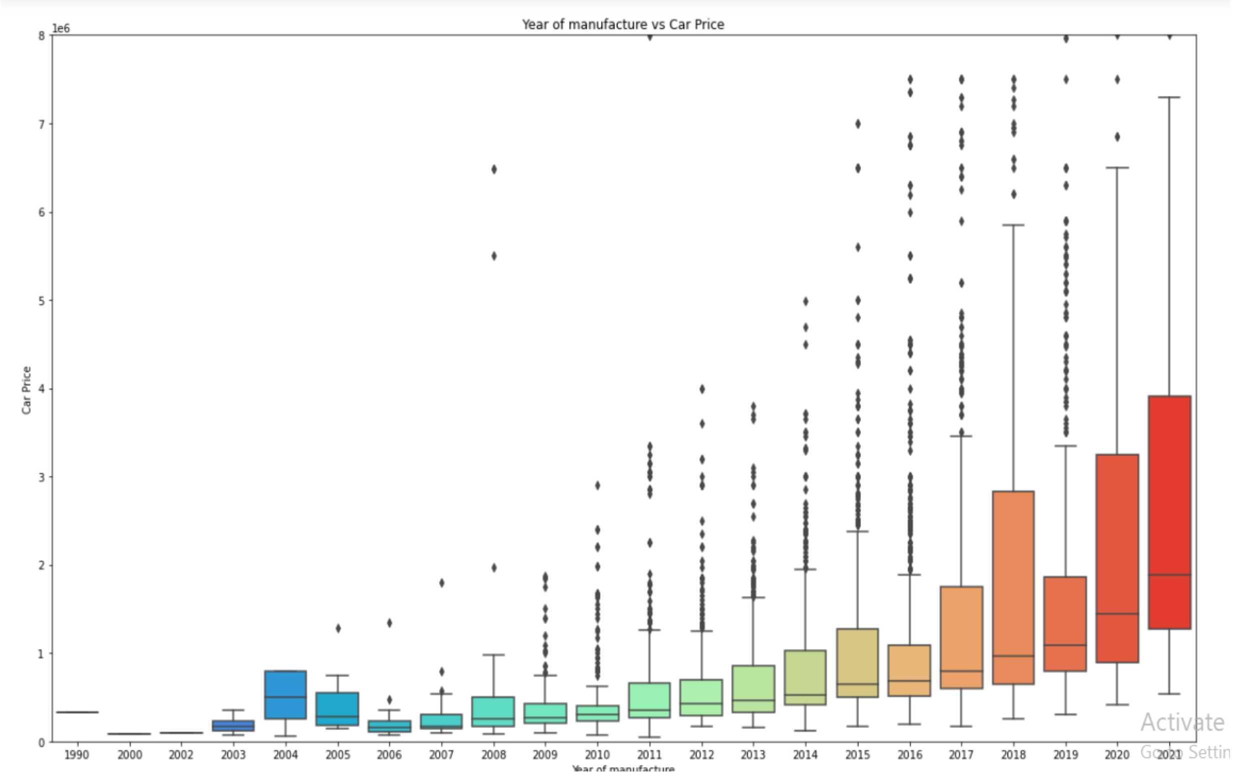
Exploratory Data Analysis

The following are the interpretation we got from data analysis, For analyzing the data we took the help of data visualization libraries like seaborn, Matplotlib, plotly,Tableau software, First we drew the correlation and heatmaps from which we came to know which are the significant variables which decides the second hand car price.



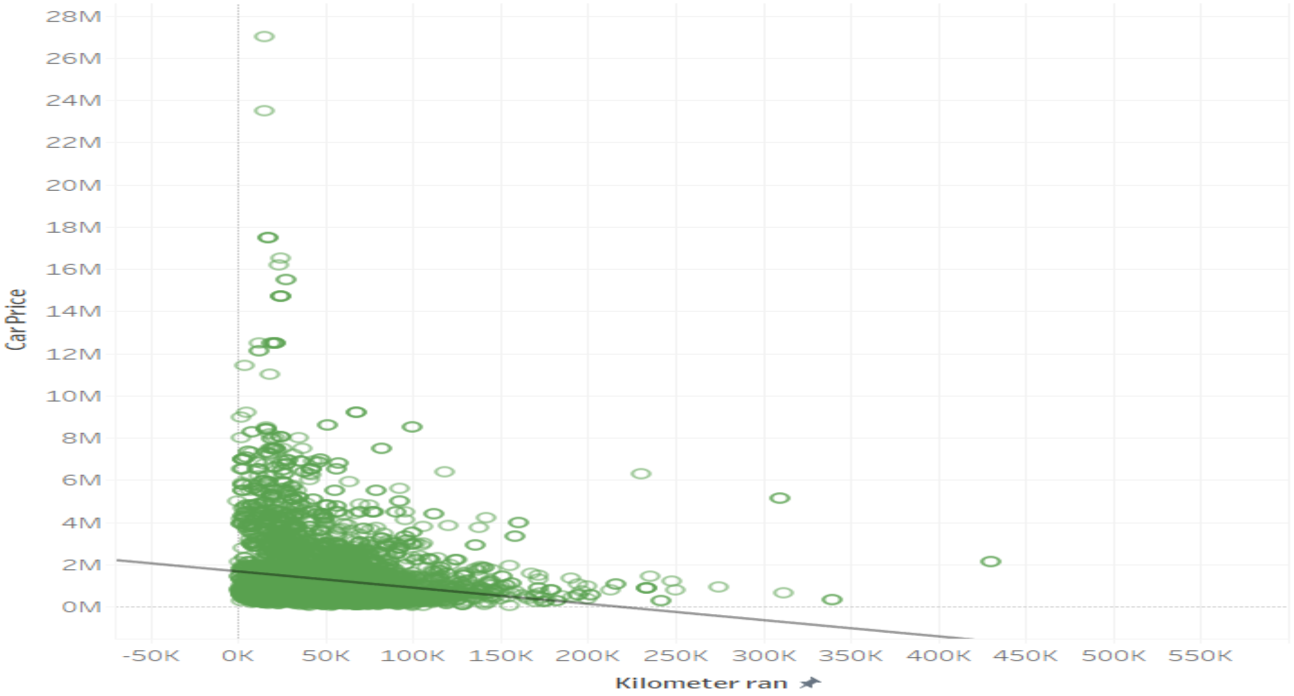
1.Car Price vs Year of manufacture





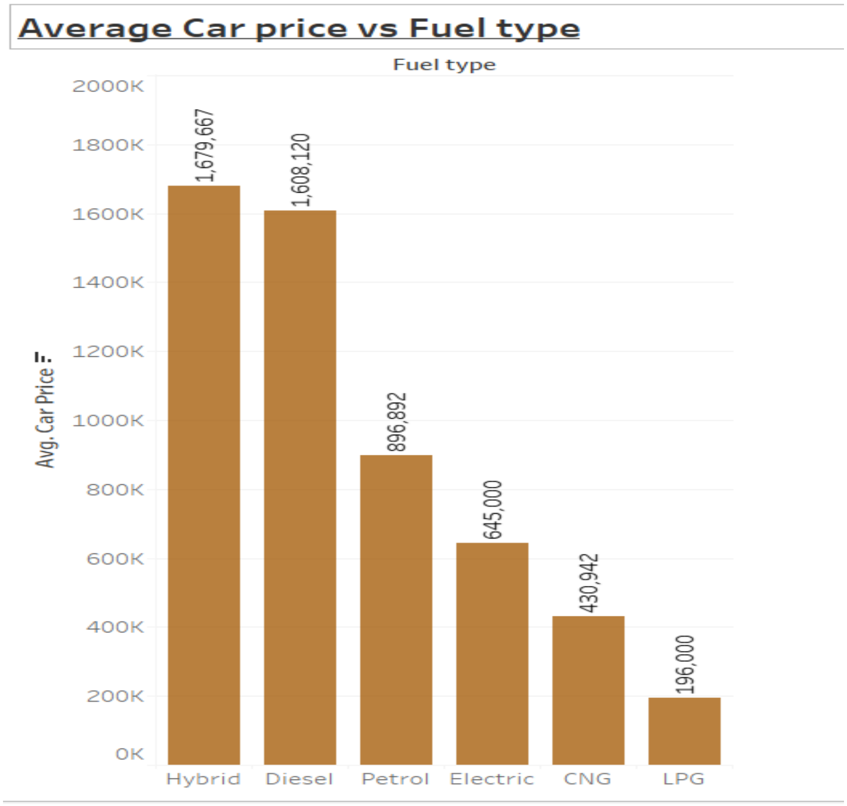
From the above graphs we can infer that as the year of the car increases the price will decreases.

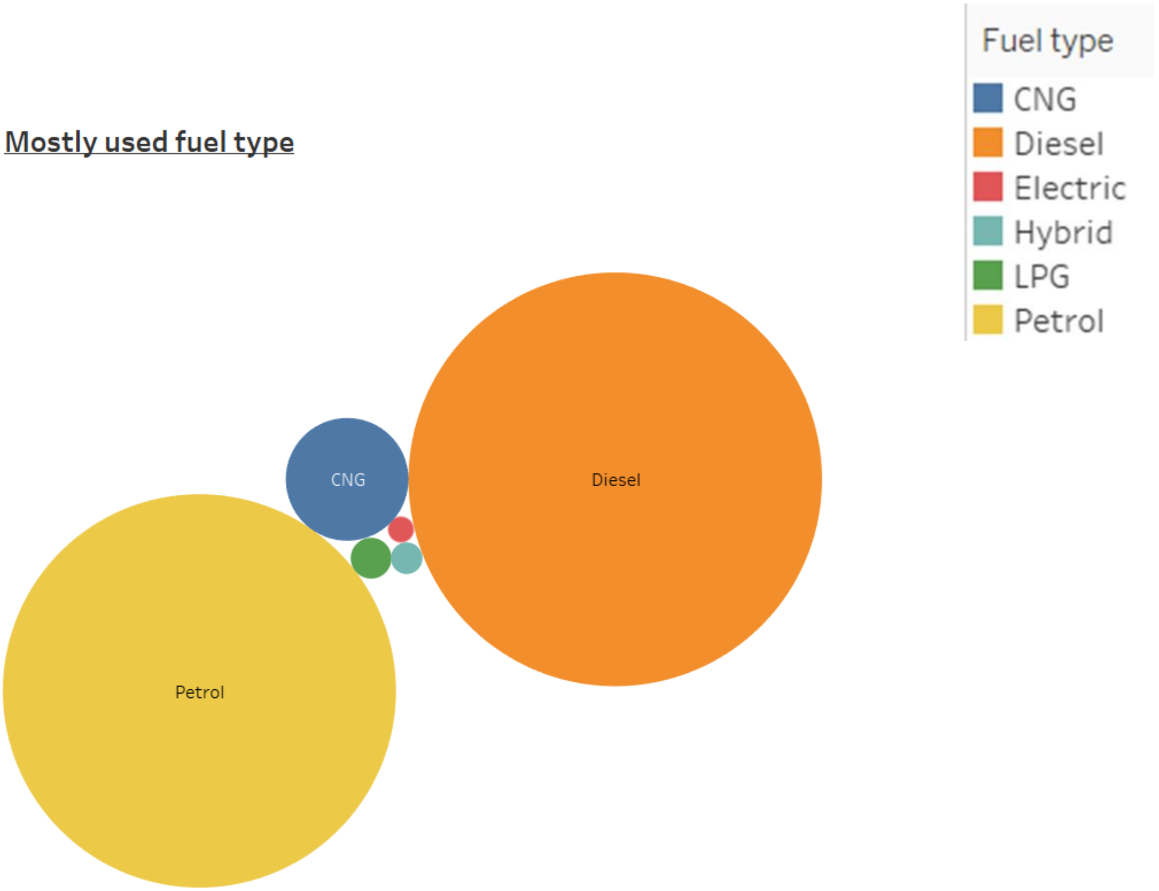
2.Car Price vs Kilometer Ran



From the above graph we can infer that as kilometer ran increases the price of the car also going to decreases.

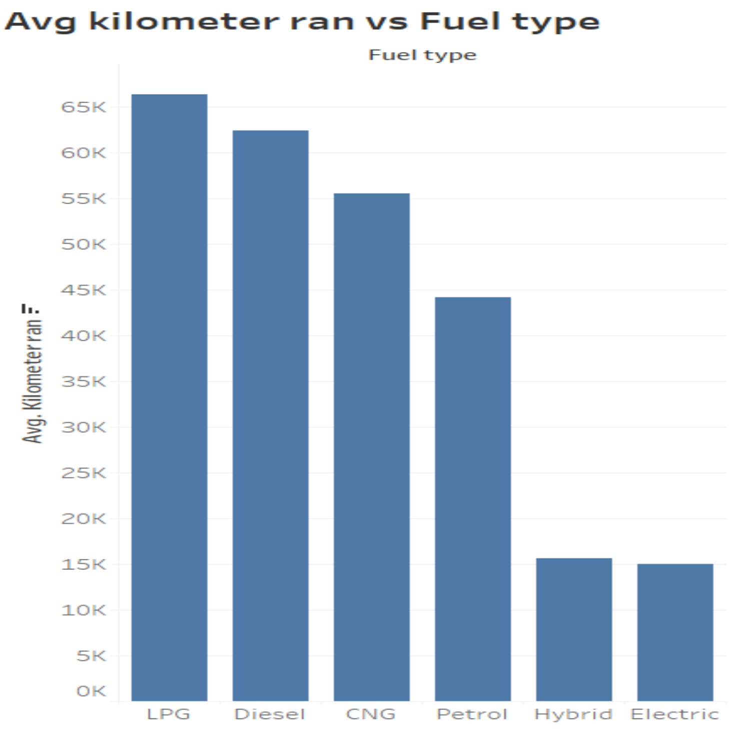
3.Car Price vs Fuel type

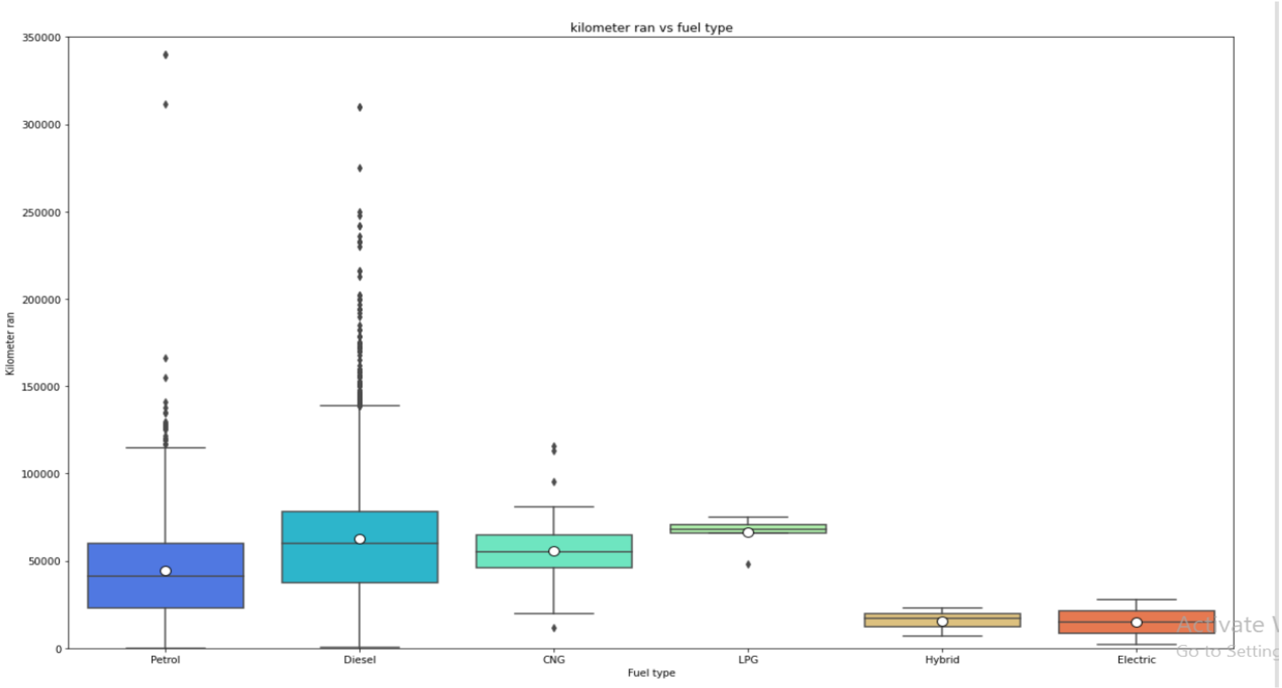




1. From the above graphs we can say that mostly used fuel type is diesel followed by petrol and CNG
2. The average price of hybrid fuel type cars are more followed by diesel and petrol.

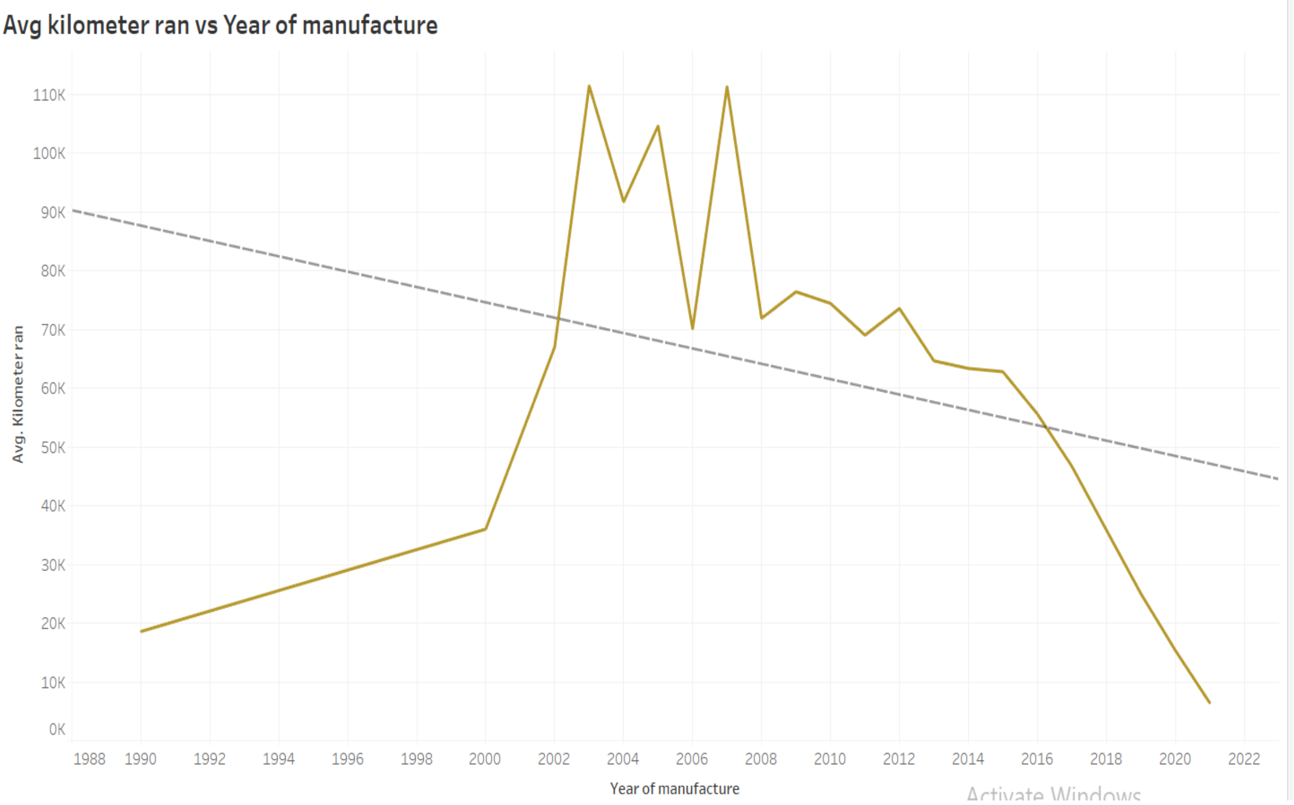
4.Kilometer ran vs fuel type

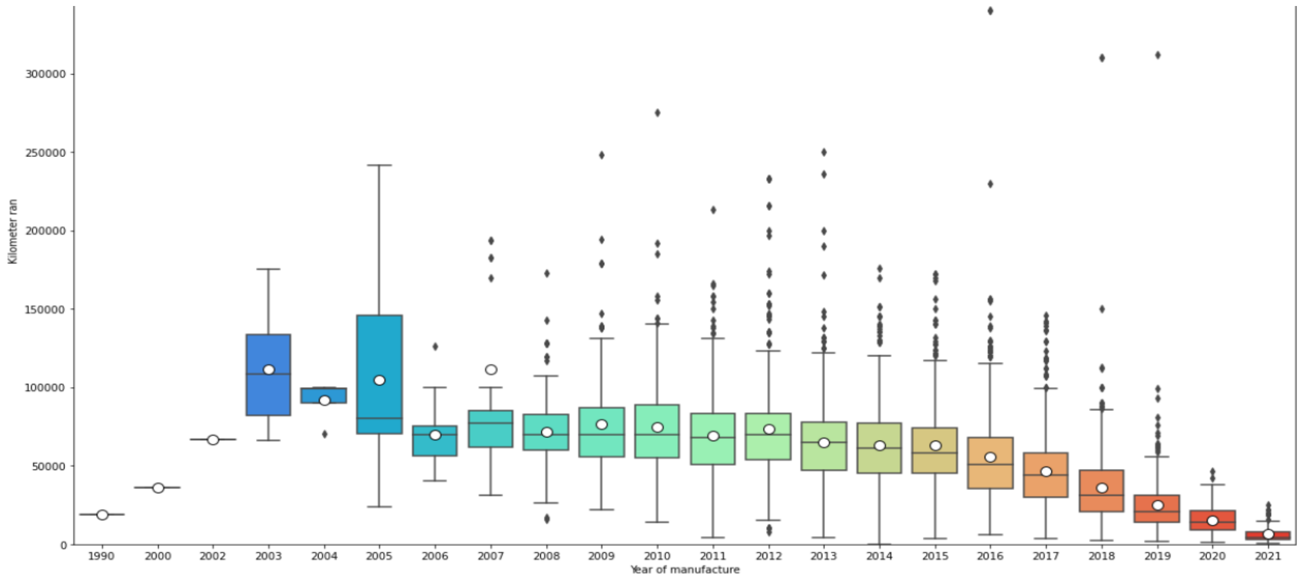




Let me interpret this result like this, suppose let's say one of our client bought all fuel type of vehicle on same day and each day he/she ran each vehicle same distance on nth day if anyone ask just tell me if you want to sell which one you will sell first and last, then this graphs tells that client will sell electric vehicle first followed by hybrid, petrol, CNG, Diesel and LPG.

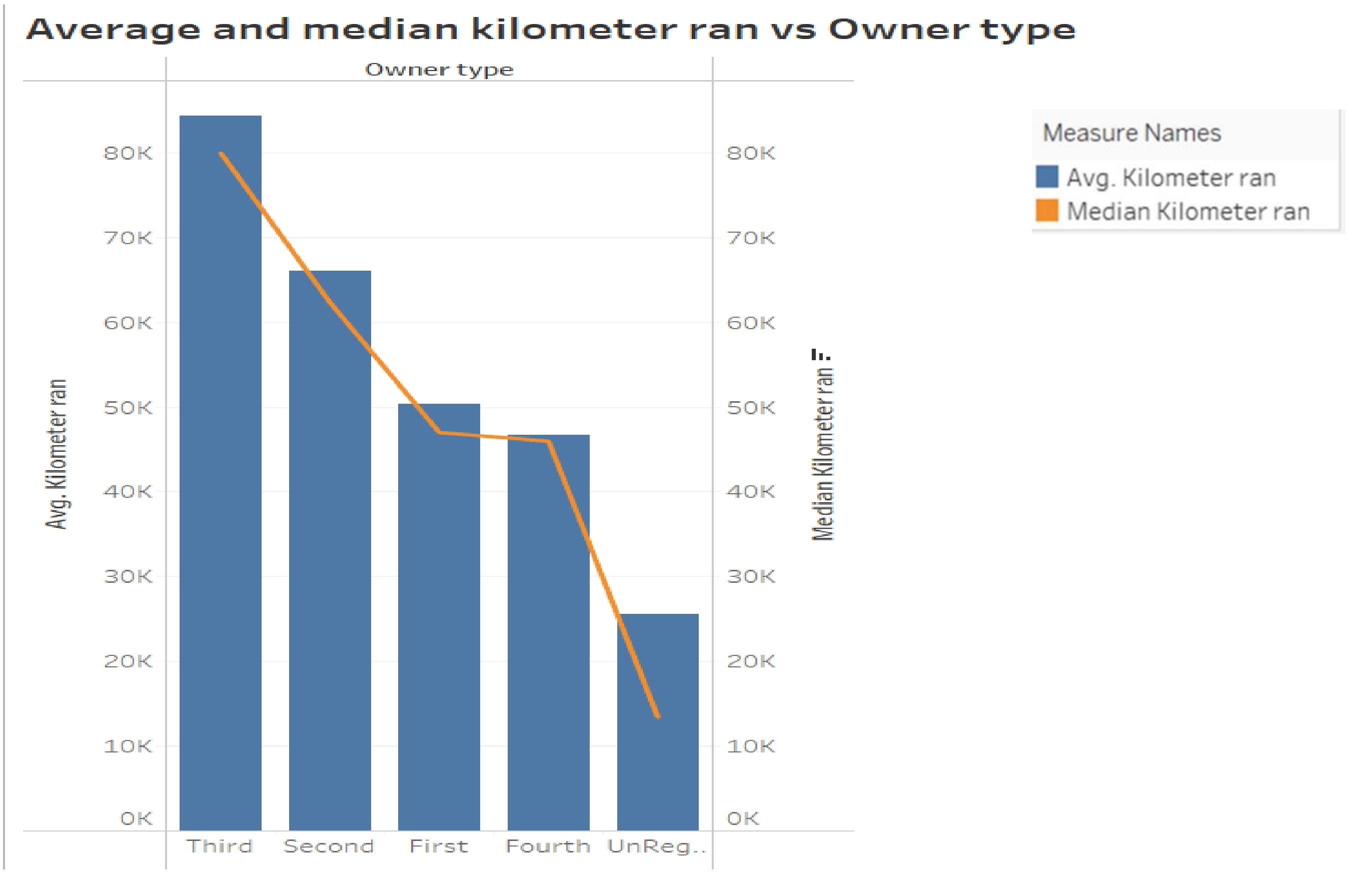
5.Year of manufacture vs Kilometer ran





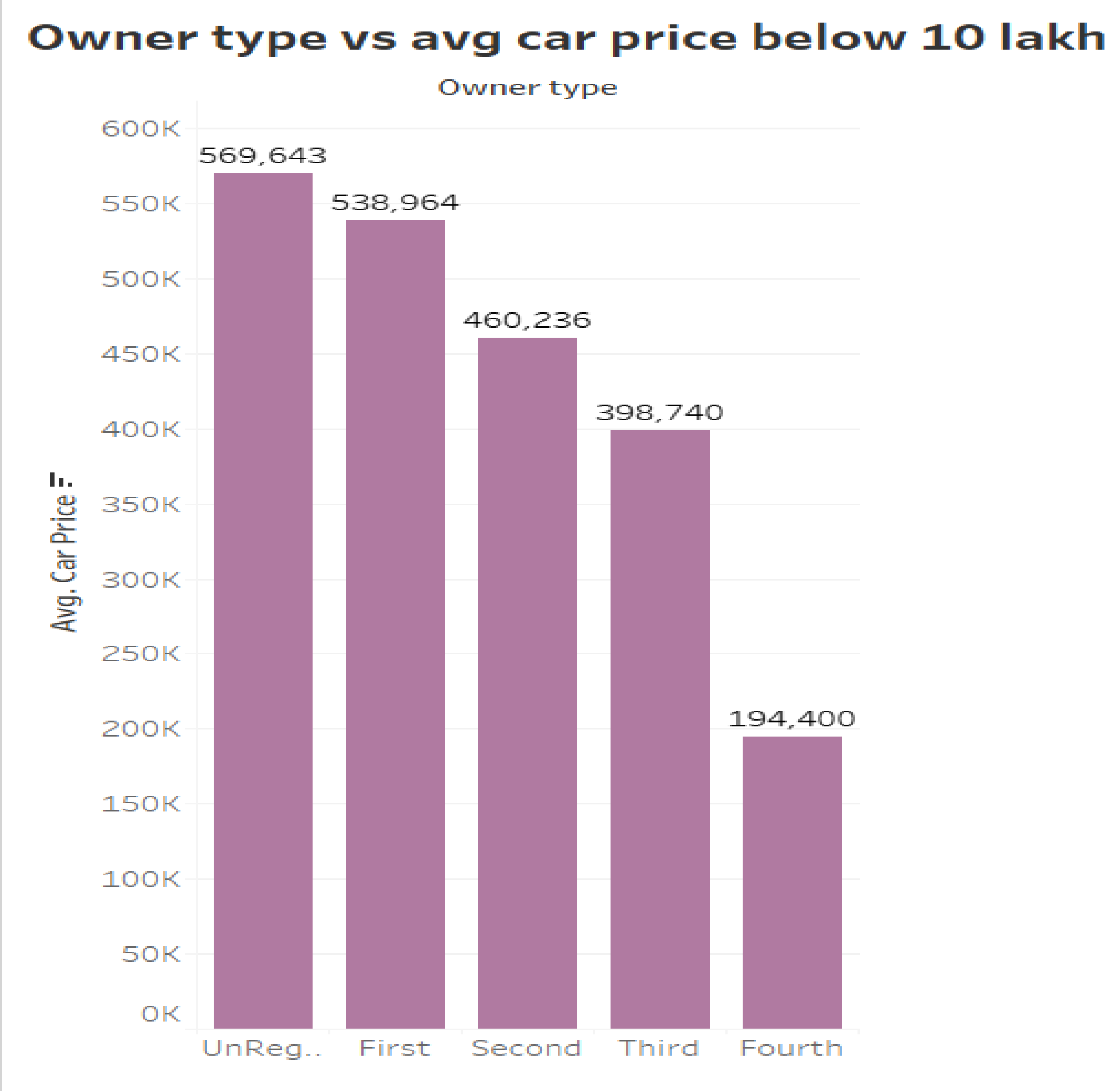
From the above graph we can infer that vehicle from 2002 to 2015 ran maximum kilometers and vehicle which are manufactured after 2016 ran significantly less.

6.Kilometer ran vs Owner type



1.It displays a general fact that as owner type increases the kilometer driven would be increases but if the owner type is 4, that is fourth owner then the kilometer ran is less than compared to 1st owner type.

7.Average car price of cars less than 10 lakh vs owner type

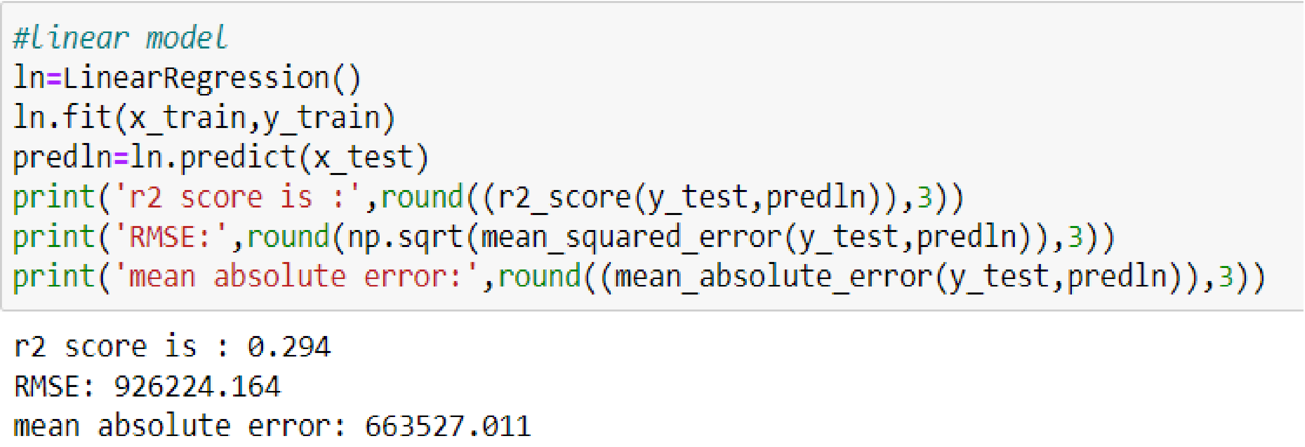


The above graphs indicates that as far as possible buy unregistered, first and second owner car, because if you buy after that the price of the car will decrease quite significantly.

Model Building

We have used several linear regression models to evaluate and finalize the best models, The major models we have used as follows.

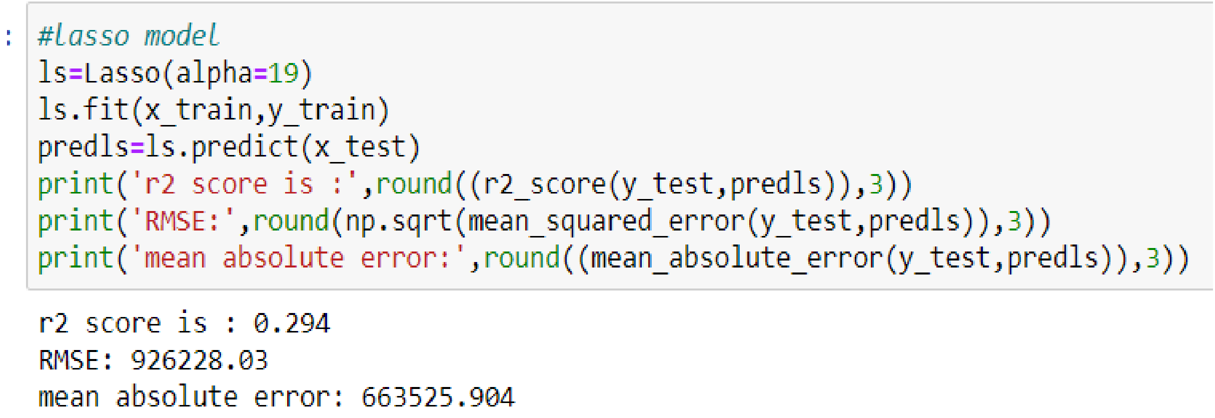
1.Linear Regression



```python
#linear model
ln=LinearRegression()
ln.fit(x_train,y_train)
predln=ln.predict(x_test)
print('r2 score is :',round((r2_score(y_test,predln)),3))
print('RMSE:',round(np.sqrt(mean_squared_error(y_test,predln)),3))
print('mean absolute error:',round((mean_absolute_error(y_test,predln)),3))
```

```
r2 score is : 0.294
RMSE: 926224.164
mean absolute error: 663527.011
```

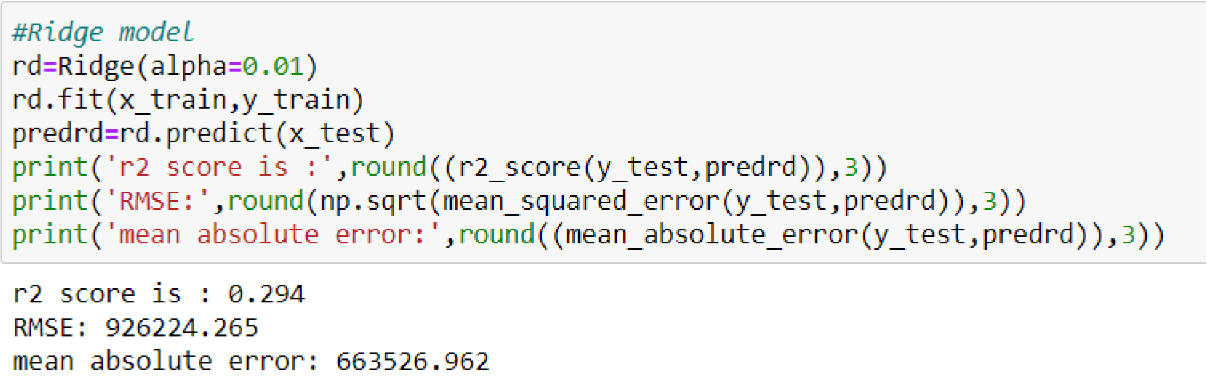
2.Lasso Regression model



```python
#lasso model
ls=Lasso(alpha=19)
ls.fit(x_train,y_train)
predls=ls.predict(x_test)
print('r2 score is :',round((r2_score(y_test,predls)),3))
print('RMSE:',round(np.sqrt(mean_squared_error(y_test,predls)),3))
print('mean absolute error:',round((mean_absolute_error(y_test,predls)),3))
```

```
r2 score is : 0.294
RMSE: 926228.03
mean absolute error: 663525.904
```

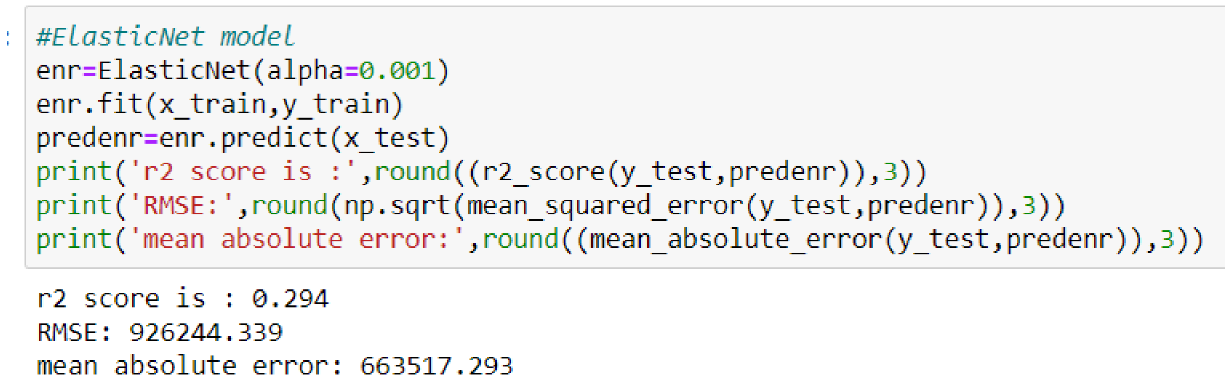
3.Ridge Regression



```python
#Ridge model
rd=Ridge(alpha=0.01)
rd.fit(x_train,y_train)
predrd=rd.predict(x_test)
print('r2 score is :',round((r2_score(y_test,predrd)),3))
print('RMSE:',round(np.sqrt(mean_squared_error(y_test,predrd)),3))
print('mean absolute error:',round((mean_absolute_error(y_test,predrd)),3))
```

```
r2 score is : 0.294
RMSE: 926224.265
mean absolute error: 663526.962
```

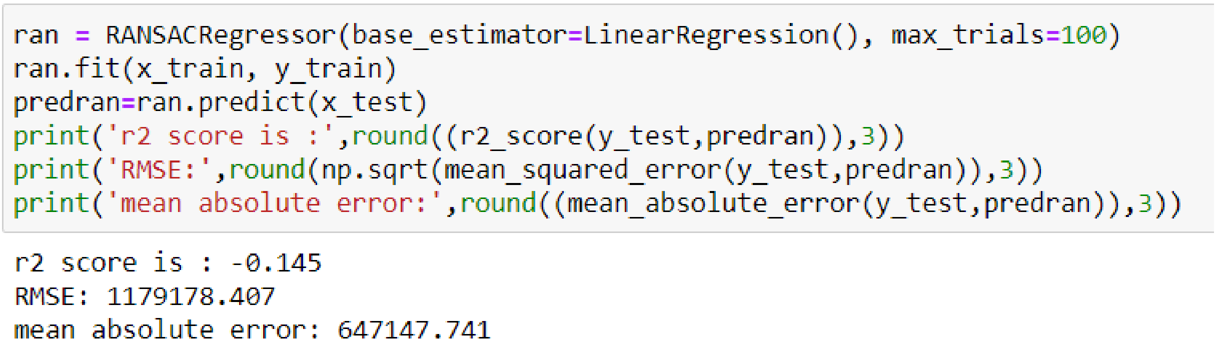
4.Elasticnet Regression



```
#ElasticNet model
enr=ElasticNet(alpha=0.001)
enr.fit(x_train,y_train)
predenr=enr.predict(x_test)
print('r2 score is :',round((r2_score(y_test,predenr)),3))
print('RMSE:',round(np.sqrt(mean_squared_error(y_test,predenr)),3))
print('mean absolute error:',round((mean_absolute_error(y_test,predenr)),3))
```

```
r2 score is : 0.294
RMSE: 926244.339
mean absolute error: 663517.293
```

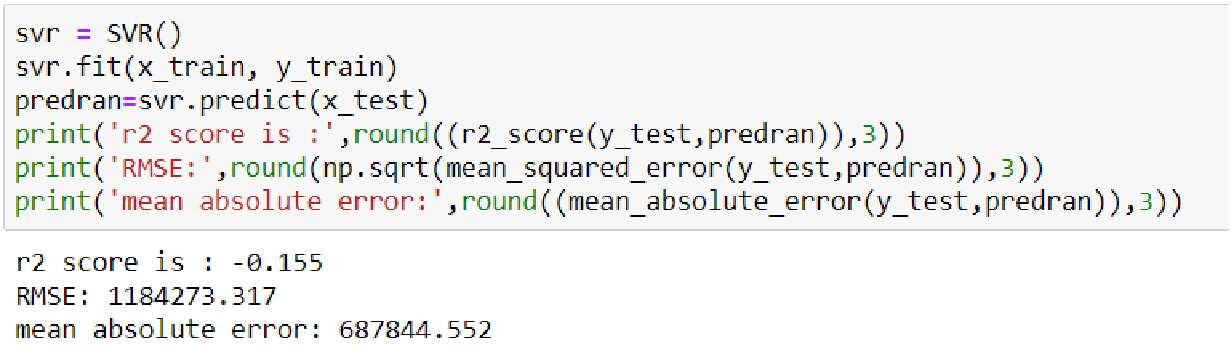
5.Ransac Regressor



```
ran = RANSACRegressor(base_estimator=LinearRegression(), max_trials=100)
ran.fit(x_train, y_train)
predran=ran.predict(x_test)
print('r2 score is :',round((r2_score(y_test,predran)),3))
print('RMSE:',round(np.sqrt(mean_squared_error(y_test,predran)),3))
print('mean absolute error:',round((mean_absolute_error(y_test,predran)),3))
```

```
r2 score is : -0.145
RMSE: 1179178.407
mean absolute error: 647147.741
```

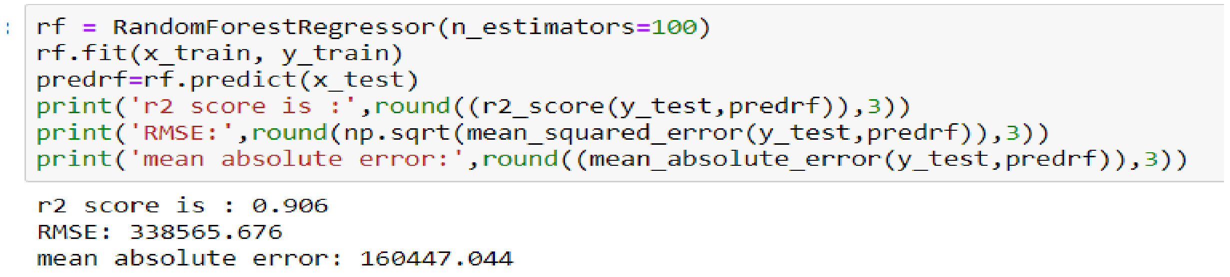
6.Support Vector Regressor



```
svr = SVR()
svr.fit(x_train, y_train)
predran=svr.predict(x_test)
print('r2 score is :',round((r2_score(y_test,predran)),3))
print('RMSE:',round(np.sqrt(mean_squared_error(y_test,predran)),3))
print('mean absolute error:',round((mean_absolute_error(y_test,predran)),3))
```

```
r2 score is : -0.155
RMSE: 1184273.317
mean absolute error: 687844.552
```

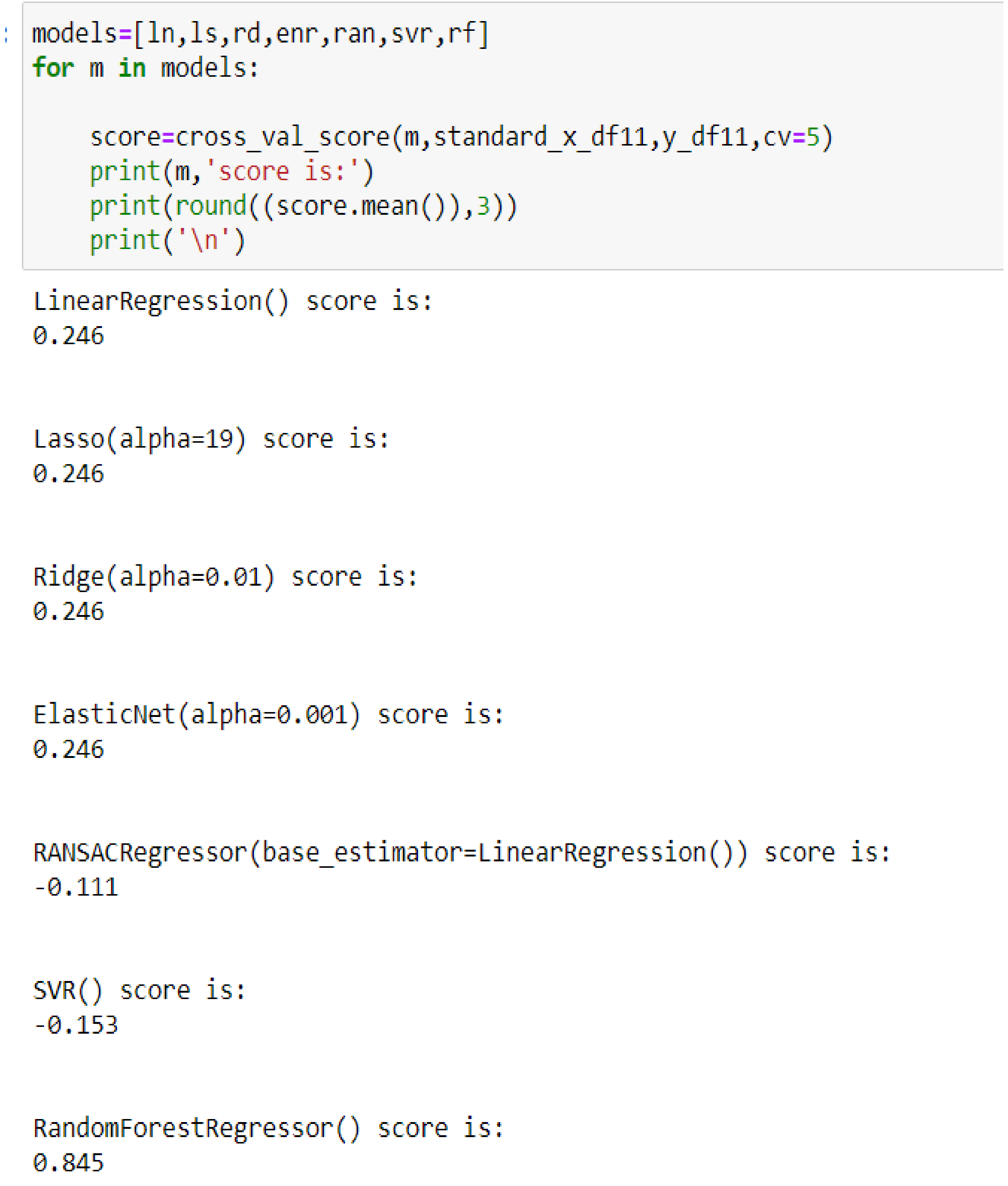
7.Random Forest Regressor



```
rf = RandomForestRegressor(n_estimators=100)
rf.fit(x_train, y_train)
predrf=rf.predict(x_test)
print('r2 score is :',round((r2_score(y_test,predrf)),3))
print('RMSE:',round(np.sqrt(mean_squared_error(y_test,predrf)),3))
print('mean absolute error:',round((mean_absolute_error(y_test,predrf)),3))
```

```
r2 score is : 0.906
RMSE: 338565.676
mean absolute error: 160447.044
```

Cross Validation Score

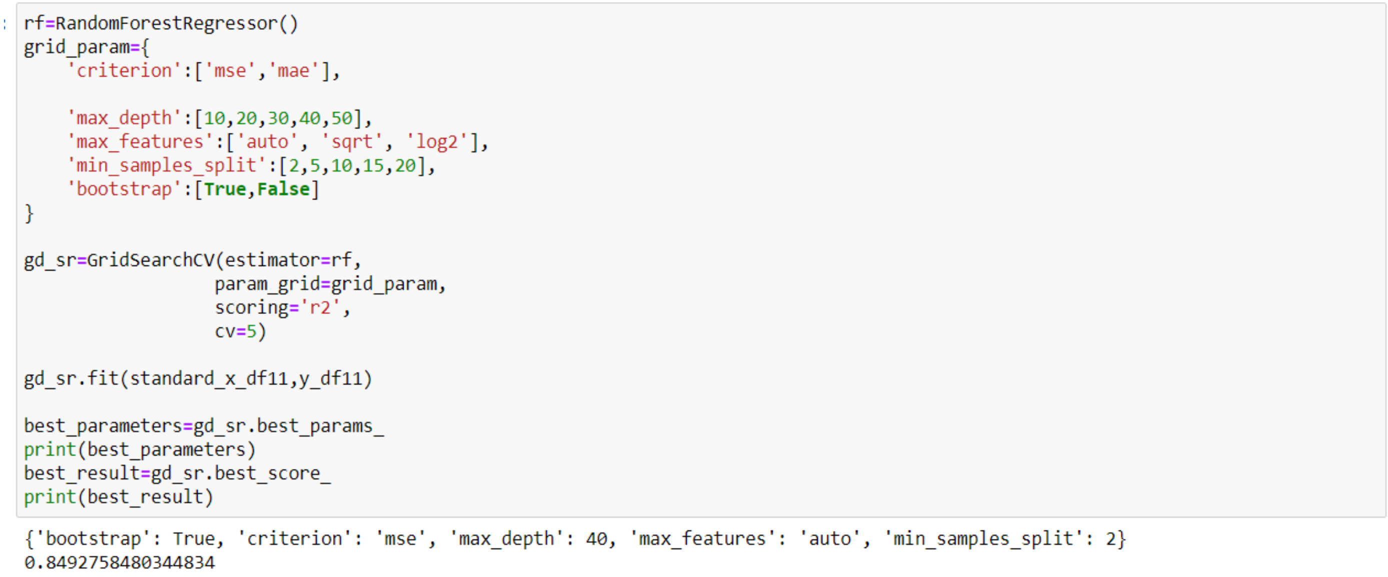


```
models=[ln,ls,rd,enr,ran,svr,rf]
for m in models:

    score=cross_val_score(m,standard_x_df11,y_df11,cv=5)
    print(m,'score is:')
    print(round((score.mean()),3))
    print('\n')
```

```
LinearRegression() score is:
0.246


Lasso(alpha=19) score is:
0.246


Ridge(alpha=0.01) score is:
0.246


ElasticNet(alpha=0.001) score is:
0.246


RANSACRegressor(base_estimator=LinearRegression()) score is:
-0.111


SVR() score is:
-0.153


RandomForestRegressor() score is:
0.845
```

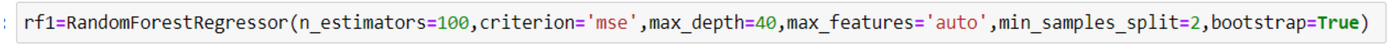
The difference between accuracy and cross validation is less for random forest regressor, so it is the best model

Hyper Parameter Tuning



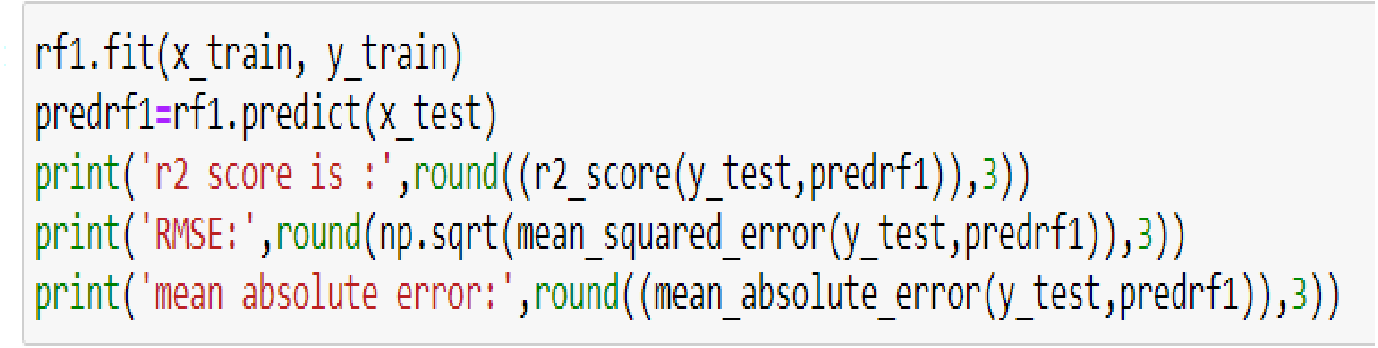
```
rf=RandomForestRegressor()
grid_param={
    'criterion':['mse','mae'],

    'max_depth':[10,20,30,40,50],
    'max_features':['auto', 'sqrt', 'log2'],
    'min_samples_split':[2,5,10,15,20],
    'bootstrap':[True,False]
}

gd_sr=GridSearchCV(estimator=rf,
                  param_grid=grid_param,
                  scoring='r2',
                  cv=5)

gd_sr.fit(standard_x_df11,y_df11)

best_parameters=gd_sr.best_params_
print(best_parameters)
best_result=gd_sr.best_score_
print(best_result)
```

```
{'bootstrap': True, 'criterion': 'mse', 'max_depth': 40, 'max_features': 'auto', 'min_samples_split': 2}
0.8492758480344834
```

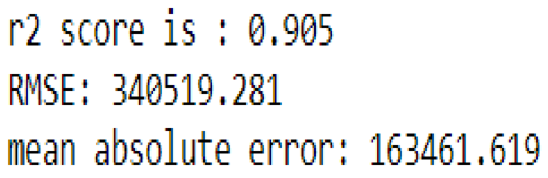


```
rf1=RandomForestRegressor(n_estimators=100,criterion='mse',max_depth=40,max_features='auto',min_samples_split=2,bootstrap=True)
```

Final Accuracy



```
rf1.fit(x_train, y_train)
predrf1=rf1.predict(x_test)
print('r2 score is :',round((r2_score(y_test,predrf1)),3))
print('RMSE:',round(np.sqrt(mean_squared_error(y_test,predrf1)),3))
print('mean absolute error:',round((mean_absolute_error(y_test,predrf1)),3))
```



```
r2 score is : 0.905
RMSE: 340519.281
mean absolute error: 163461.619
```

Conclusions

- ## Key findings and the conclusions of the study

  Based on the above key findings we would like to give some prescription for our client

  1. As far as possible try to buy car before 2016 so that we can sell it comparatively higher price.
  2. Try to buy vehicle which ran less than 1 lakh kilometer
  3. While choosing the fuel type give the first preference to Diesel, followed by Petrol, CNG, Hybrid
  4. Try to avoid vehicle from 2002 to 2015, because it will get sold for less price, If its vintage then you can go for it.
  5. Try to buy vehicle after 2016 eventually you will get higher price while selling.
  6. Try to buy unregistered , first owner or second owner don't go above that if its not so good in other areas.

- ## Learning outcomes of the study in respect to data science

  Learnt many things from this project, following are the significant one

  1. How to deal with xpaths and getting the urls.
  2. Before we step into the project we should have a foresight of its complete picture and completed version.
  3. Learnt that whenever we are checking whether an variable is of object or integer type we shouldn't include int in double inverted comma because its an keyword.
  4. Overall experience got enriched from this project.

!! Thank you !!