CAR PRICE PREDICTION PROJECT

Submitted by:

Ayush Yadav

ACKNOWLEDGMENT

I would like to thank FlipRobo Technologies for giving me the opportunity to work on this project. I am very grateful to DataTrained team for providing me the knowledge which helped me a lot to work on this project.

Reference sources are:

1. Google

2. Stackoverflow

3. DataTrained Notes

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.

- ## Review of Literature

  The aim of this project is to build a model which can be used to predict the car price. This project is more about data exploration, finding the better insights from data and using the skills and techniques to build an efficient model which can predict the prices after the recession faced during Covid-19 crisis. Since we scrape a good amount of data that are related to the price of car, we can do better data exploration and derive some interesting features using the available columns.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  In this project we have worked to predict the price of the used car. 'Price' is our target column and it is continuous in nature, so we're dealing with regression type of problem.

  This project is based on 2 main phases-

  A. Data Collection : In this section, we need to scrape the data of used cars from websites (Olx, cardekho, Cars24 etc.). Web scraping is required for this. We have to fetch data for different locations. Generally, the features columns are Brand, model, variant, manufacturing year, driven kilometers, fuel, number of owners, location and at last target variable Price of the car. This data is to give a hint about important variables in used car model. We can make changes to it, we can add or you can remove some columns, it completely depends on the website from which we are fetching the data. We have tried to include all types of cars in our data for example- SUV, Sedans, Coupe, minivan, Hatchback.

  B. **Model Building:** After collecting the data, you need to build a machine learning model. Before model building do all data pre-processing steps. Try different models with different hyper parameters and select the best model. Follow the complete life cycle of data science. Include all the steps like.

  1. Data Cleaning
  2. Exploratory Data Analysis
  3. Data Pre-processing
  4. Model Building
  5. Model Evaluation

6. Selecting the best model

- Data Sources and their formats

Data sources for our model are online Car Re-sale websites like Car24. We scraped our data using the web scrapping tool Selenium. Let's have a look at our dataset.

	Brand	Model	History	Transmission_Type	Fuel_Type	Purchase_Year	Location	KM_Driven	Owner	Price
0	Maruti	Swift	Non-Accidental	MANUAL	Diesel	2015	Mumbai	30736.0	1st Owner	477099
1	Honda	Amaze	Non-Accidental	MANUAL	Petrol	2014	Mumbai	35265.0	1st Owner	426699
2	Maruti	Alto K10	Non-Accidental	MANUAL	Petrol	2016	Mumbai	29393.0	1st Owner	306799
3	Volkswagen	Ameo	Non-Accidental	MANUAL	Petrol	2016	Mumbai	11414.0	2nd Owner	476399
4	Maruti	Swift	Non-Accidental	MANUAL	Petrol	2017	Mumbai	21485.0	1st Owner	527899
...	...	...	...	...	...	...	...	...	...	...
2457	Maruti	Wagon R 1.0	Non-Accidental	MANUAL	Petrol	2011	Bangalore	85650.0	1st Owner	278299
2458	Hyundai	i10	Non-Accidental	AUTOMATIC	Petrol	2011	Bangalore	58231.0	2nd Owner	337699
2459	Maruti	Swift	Non-Accidental	MANUAL	Petrol	2018	Bangalore	65030.0	1st Owner	631899
2460	Maruti	Ritz	Non-Accidental	MANUAL	Petrol	2010	Bangalore	98931.0	2nd Owner	311099
2461	Hyundai	Santro Xing	Non-Accidental	MANUAL	Petrol + LPG	2011	Bangalore	67394.0	1st Owner	294199

2462 rows × 10 columns

Let's have a look at the data formats:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2381 entries, 0 to 2380
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Brand              2381 non-null   object
 1   Model              2381 non-null   object
 2   History            2122 non-null   object
 3   Transmission_Type  2381 non-null   object
 4   Fuel_Type          2135 non-null   object
 5   Purchase_Year      2381 non-null   int64
 6   Location           2381 non-null   object
 7   KM_Driven          2135 non-null   float64
 8   Owner              2379 non-null   object
 9   Price              2381 non-null   int64
dtypes: float64(1), int64(2), object(7)
memory usage: 186.1+ KB
```

- There are missing values (null values) in the dataset.
- Our dataset contains object, integer and float data type.

Now, let’s check the feature columns of the dataset.

```
Index(['Brand', 'Model', 'History', 'Transmission_Type', 'Fuel_Type',
       'Purchase_Year', 'Location', 'KM_Driven', 'Owner', 'Price'],
      dtype='object')
```

- The dataset contains following columns:
  - **Brand:** Shows the brand name of the car
  - **Model:** Shows the model of the car
  - **History:** Shows the accidental record history of the car.
  - **Transmission_Type:** Transmission type of the car (Manual or Automatic).
  - **Fuel_Type:** Fuel used in the car.
  - **Purchase_Year:** The year in which the car was first purchased.
  - **Location:** Shows location of the seller.
  - **KM_Driven:** Total Kms driven so far.
  - **Owner:** Number of owners.
  - **Price:** This is the target column, the price of the car.

- Data Preprocessing Done

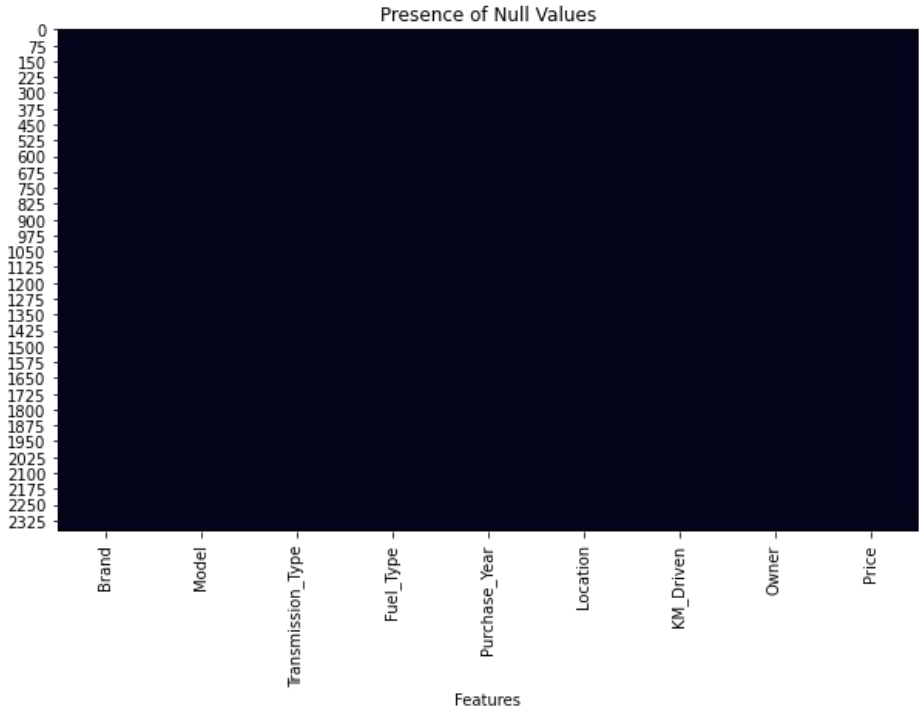
  <u>Checking for the null values in the dataset:</u>

  After loading the dataset, we checked for the presence of null values or missing values in the dataset. We found that there were null values in the dataset which we have handled and removed successfully.

```
df.isnull().sum()
```

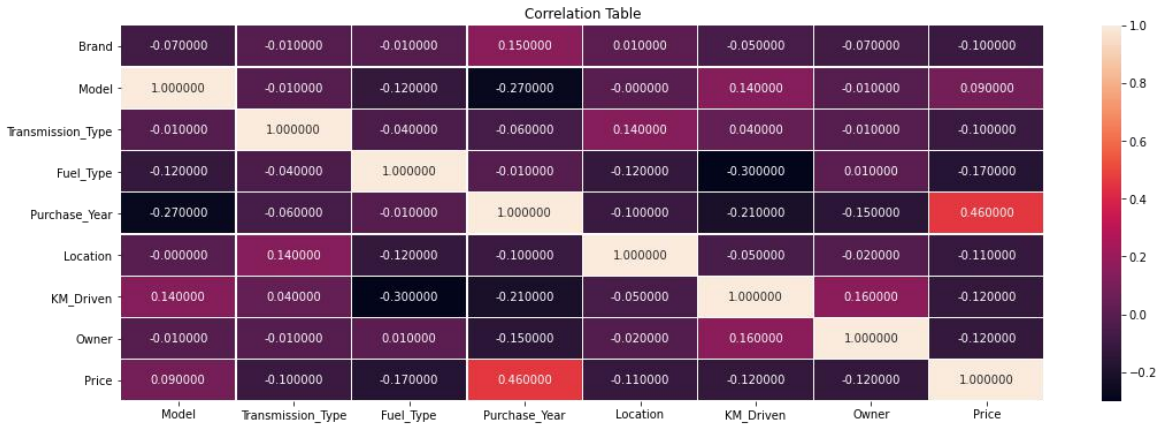
```
Brand                0
Model                0
History            259
Transmission_Type    0
Fuel_Type          246
Purchase_Year        0
Location             0
KM_Driven          246
Owner                2
Price                0
dtype: int64
```

  After handling the missing values

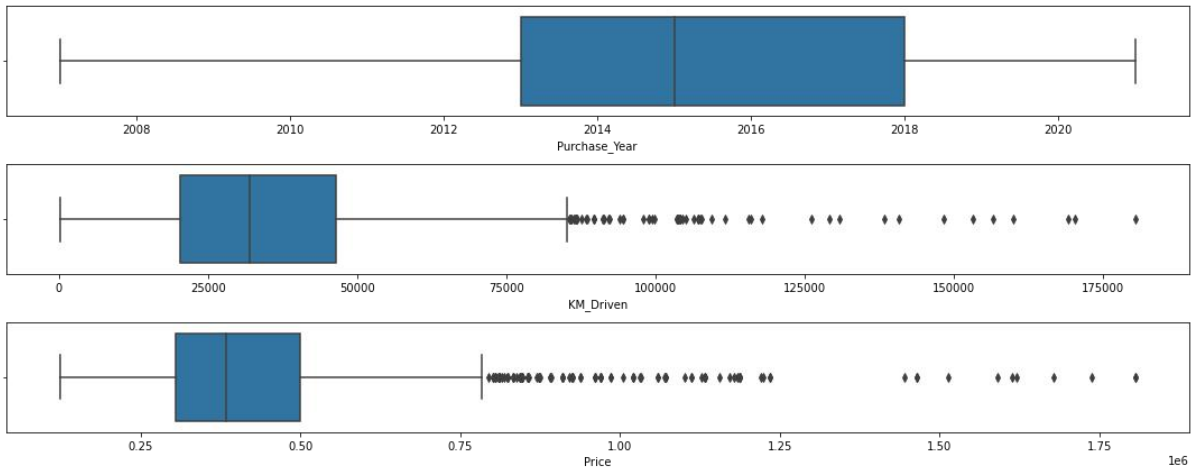


Checked for the statistical summary of the dataset:

Checked for the correlation of the features columns with the target column:



Checked for the outliers



The BoxPlot says that, there is outliers in KM driven and Price columns. We've used Z-Score Technique to treat the outliers of the dataset. We lost approx 11% data while removing the outliers.

We are using LabelEncoder to convert the categorical data into numerical
data.

```
from sklearn.preprocessing import LabelEncoder

LE = LabelEncoder()

variables = ['Brand', 'Model', 'Transmission_Type', 'Fuel_Type', 'Location', 'Owner']

for v in variables:
    df[v] = LE.fit_transform(df[v])
```

```
# Checking for the dataset after Labelencoding.

df.head()
```

	Brand	Model	Transmission_Type	Fuel_Type	Purchase_Year	Location	KM_Driven	Owner	Price
0	10	62	1	0	2015	6	30736.0	0	477099
1	4	7	1	1	2014	6	35265.0	0	426699
2	10	6	1	1	2016	6	29393.0	0	306799
3	20	8	1	1	2016	6	11414.0	1	476399
4	10	62	1	1	2017	6	21485.0	0	527899

Skewness in the dataset:

```
x.skew()
```

```
Brand              -0.050770
Model               0.314241
Transmission_Type   0.000000
Fuel_Type          -2.588587
Purchase_Year      -0.310240
Location           -0.728968
KM_Driven           0.589565
Owner               1.436801
dtype: float64
```

- Columns Fuel_Type, Location, KM_Driven, Owner contains skewness.

Treated skewness with power transform:

```
# Removing the skewness using the power transform

from sklearn.preprocessing import power_transform
x = power_transform(x)
x
```

```
array([[ 0.41726829,  0.98560993,  0.        , ...,  0.12489667,
        -0.04551072, -0.51311167],
       [-1.62968251, -1.09355798,  0.        , ...,  0.12489667,
         0.18285546, -0.51311167],
       [ 0.41726829, -1.18910328,  0.        , ...,  0.12489667,
        -0.11601592, -0.51311167],
       ...,
       [ 0.41726829,  1.16923036,  0.        , ..., -1.58416295,
         2.16951445, -0.51311167],
       [ 0.41726829,  0.98560993,  0.        , ..., -1.58416295,
         1.44393941, -0.51311167],
       [ 0.41726829,  0.82958686,  0.        , ..., -1.58416295,
         2.59652236,  1.94889352]])
```

StandardScaler:

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

x = scaler.fit_transform(x)
x
```

```
array([[ 0.41726829,  0.98560993,  0.        , ...,  0.12489667,
        -0.04551072, -0.51311167],
       [-1.62968251, -1.09355798,  0.        , ...,  0.12489667,
         0.18285546, -0.51311167],
       [ 0.41726829, -1.18910328,  0.        , ...,  0.12489667,
        -0.11601592, -0.51311167],
       ...,
       [ 0.41726829,  1.16923036,  0.        , ..., -1.58416295,
         2.16951445, -0.51311167],
       [ 0.41726829,  0.98560993,  0.        , ..., -1.58416295,
         1.44393941, -0.51311167],
       [ 0.41726829,  0.82958686,  0.        , ..., -1.58416295,
         2.59652236,  1.94889352]])
```

- Hardware and Software Requirements and Tools Used

  I have used i3 processor with 4GB RAM as hardware.

  Software:

  1. Jupyter Notebook (Anaconda 3)
  2. Python 3.9

  Libraries used:

  a. Pandas
  b. NumPy
  c. Matplotlib
  d. Seaborn
  e. Scipy
  f. Selenium

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  As we already mentioned above we have continuous target variable so based on the type of variable we are using Regression approach for our model and will use some algorithms like

  i. Linear Regression
  ii. Ridge Regression
  iii. Decision Tree Regression
  iv. Random Forest Regression

v. KNeighbors Regression

vi. AdaBoost Regression

vii. Gradient Boosting Regression

- Testing of Identified Approaches (Algorithms)

  After initiating the instances for our model we will create a function which will give us all the metrics score we are using in our model building by passing the model into it

```
Method = []       # algo used
R2_Score = []     # R2 score of algorithm
MAE = []          # mean absolute error
MSE = []          # mean squared error
RMSE = []         # root mean squared error
CVScore = []      # mean of cross val score
Std = []          # standard deviation in cross val


def r2score(model):
    print(30*" ",model,10*" ")
    Method.append(str(model))

    # Training score
    model.fit(x_train,y_train)
    print("\nTraining Score {} %".format(round(model.score(x_train,y_train)*100,3)))
    y_pred = model.predict(x_test)

    # R2 score value
    r2 = r2_score(y_test,y_pred)
    print("\nCoeff. of determination = %.2f"%r2)
    R2_Score.append(round(r2,2))

    # Mean absolute error
    mae=mean_absolute_error(y_test,y_pred)
    print("Mean absolute error = ",mae)
    MAE.append(mae)

    # Mean squared error
    mse=mean_squared_error(y_test,y_pred)
    print("Mean Squared error = ",mse)
    MSE.append(mse)

    # Root mean squared error
    print("Root means sq. error = ",np.sqrt(mean_squared_error(y_test,y_pred)))
    RMSE.append(np.sqrt(mean_squared_error(y_test,y_pred)))

    # cross validation
    cvs=cross_val_score(model,x,y,cv=6,scoring='r2')
    print("\nCross val score = ",round(cvs.mean()*100,3),"%")
    CVScore.append(round(cvs.mean()*100,3))
```

- Run and Evaluate selected models

Evaluation of Algorithms:

```
# Table view of result of each metrix from above algorithms
evaluations = pd.DataFrame({"Model":Method,"R2 Score":R2_Score,"MAE":MAE,"MSE":MSE,
                            "RMSE":RMSE,"CV Score":CVScore,"Stnd_dev":Std})
evaluations
```

	Model	R2 Score	MAE	MSE	RMSE	CV Score	Stnd_dev
0	LinearRegression()	0.46	85294.237321	1.300181e+10	114025.490475	25.313	0.321115
1	Ridge()	0.46	85300.951176	1.300352e+10	114032.988770	25.325	0.320863
2	DecisionTreeRegressor()	0.62	56685.916824	8.981685e+09	94771.754240	26.816	0.561469
3	RandomForestRegressor()	0.73	51848.865343	6.362946e+09	79768.073923	48.005	0.520202
4	KNeighborsRegressor()	0.50	76592.173913	1.194966e+10	109314.502295	25.766	0.298190
5	AdaBoostRegressor()	0.44	92709.929143	1.339072e+10	115718.276593	24.278	0.417614
6	GradientBoostingRegressor()	0.70	59010.401763	7.067059e+09	84065.802938	48.545	0.413240

- As per our above analysis we can say that Random Forest Regressor is performing best among all other models.

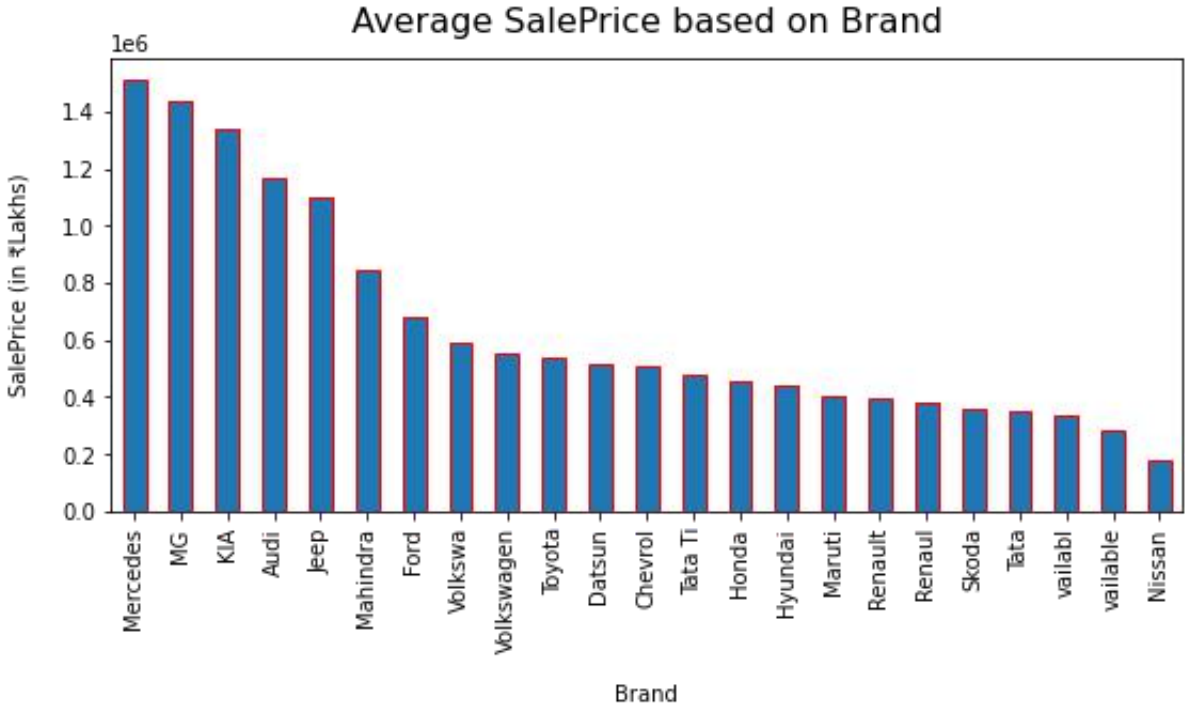
- **After testing various algorithms and hypertunnig the best two algorithms we find that both RandomForestRegressor and GradientBoostingRegressor are giving better results but we are finalising the GradientBoostingRegressor algorithm for our model as there is minimum difference in coefficient of determination and cross validation score and also less variance in result in the higher Price.**
- **The selected model gives coefficient of determination value =0.71 with minimum mean absolute error of 56384.947 and cross validation score 0.53 with a least standard deviation of 0.20.**

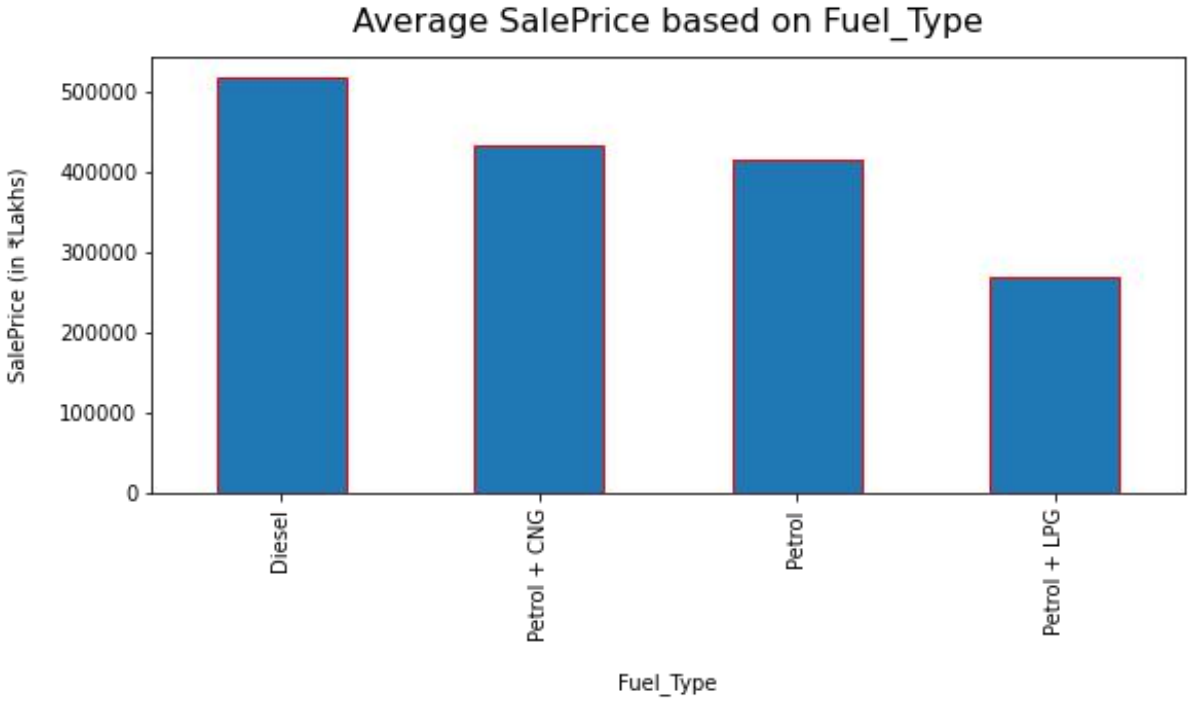
- Key Metrics for success in solving problem under consideration

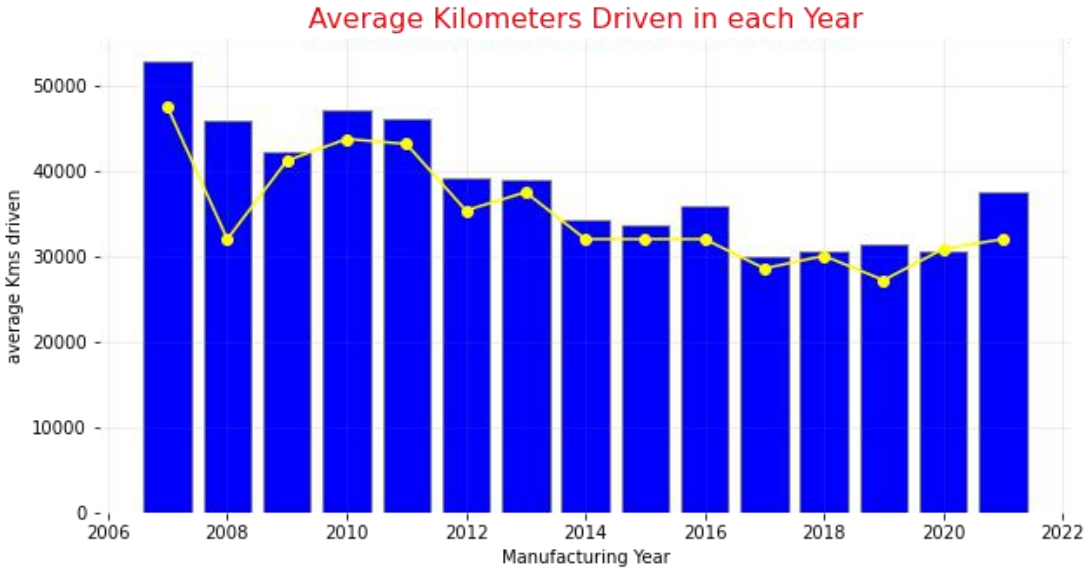
  In this project we have used the following methods to decide the best model:

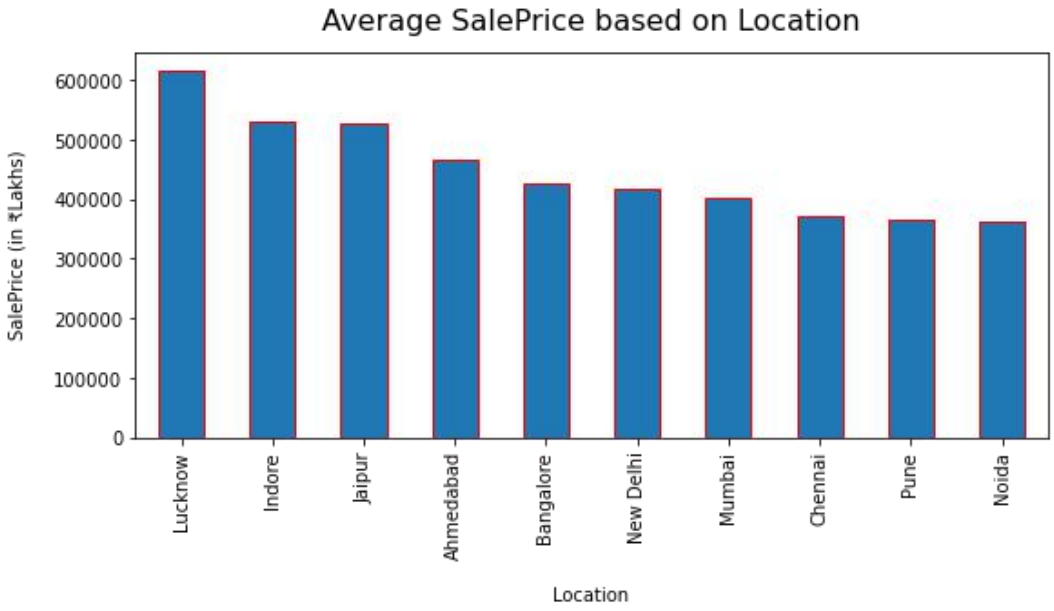
  - R2 Score
  - Mean Absolute Error
  - Root Mean Squared Error
  - Cross Validation Score
  - Standard deviation

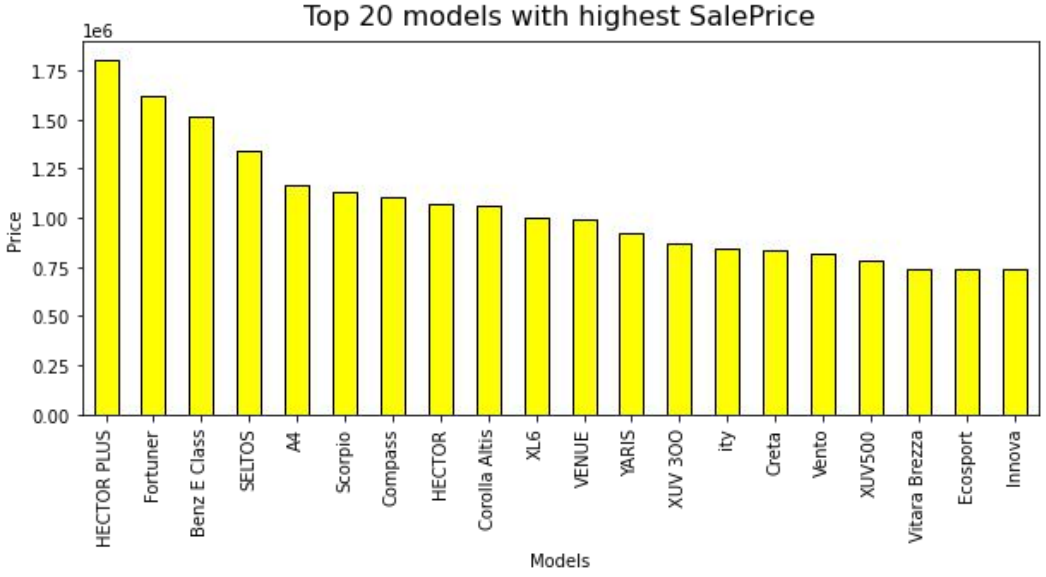
- Visualizations

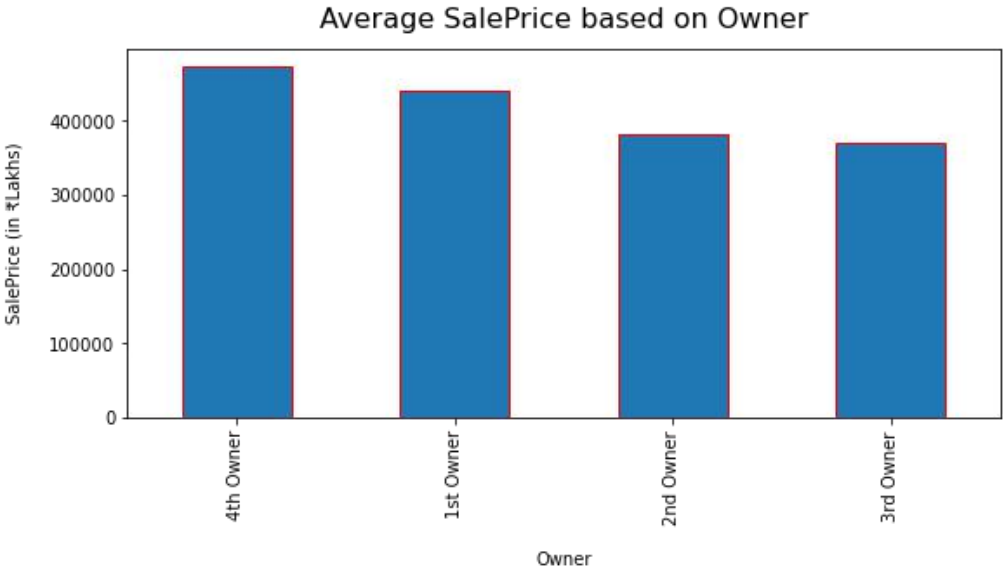


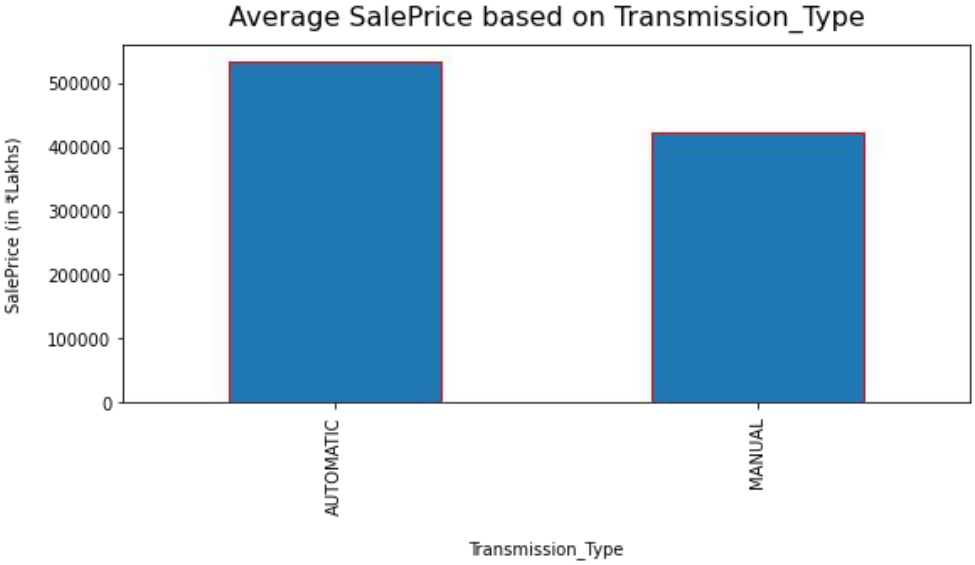



Average Kilometers Driven in each Year
average Kms driven
50000
40000
30000
20000
10000
0
2006
2008
2010
2012
2014
2016
2018
2020
2022
Manufacturing Year

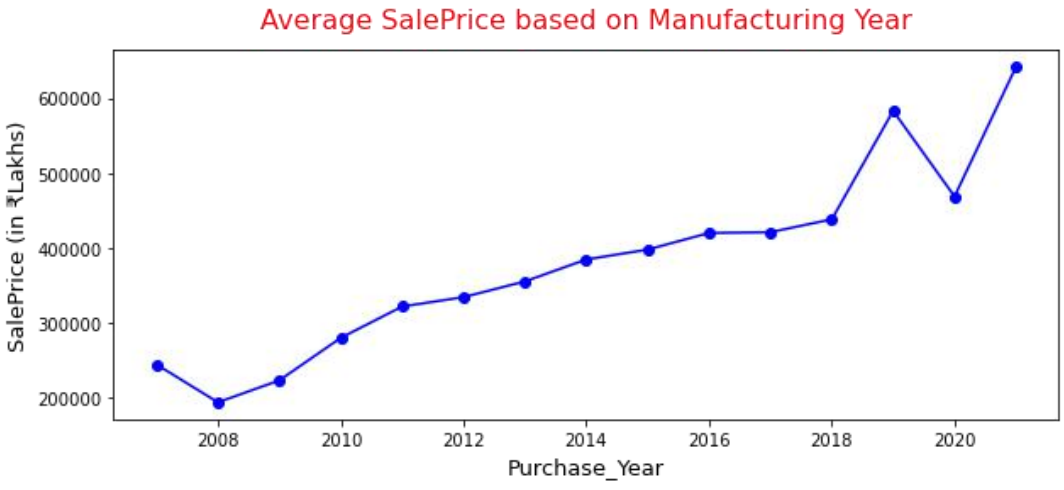

Average SalePrice based on Location
SalePrice (in ₹Lakhs)
600000
500000
400000
300000
200000
100000
0
Lucknow
Indore
Jaipur
Ahmedabad
Bangalore
New Delhi
Mumbai
Chennai
Pune
Noida
Location


Top 20 models with highest SalePrice
1e6
Price
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
HECTOR PLUS
Fortuner
Benz E Class
SELTOS
A4
Scorpio
Compass
HECTOR
Corolla Altis
XL6
VENUE
YARIS
XUV 300
ity
Creta
Vento
XUV500
Vitara Brezza
Ecosport
Innova
Models


Average SalePrice based on Owner
SalePrice (in ₹Lakhs)
400000
300000
200000
100000
0
4th Owner
1st Owner
2nd Owner
3rd Owner
Owner


Average SalePrice based on Transmission_Type
SalePrice (in ₹Lakhs)
500000
400000
300000
200000
100000
0
AUTOMATIC
MANUAL
Transmission_Type


Average SalePrice based on Purchase_Year
SalePrice (in ₹Lakhs)
600000
500000
400000
300000
200000
100000
0
2021
2019
2020
2018
2017
2016
2015
2014
2013
2012
2011
2010
2009
2008
2007
Purchase_Year



CONCLUSION

- Key Findings and Conclusions of the Study

After working on this project ,we got a lot of insights of how to collect data , pre-process it ,remove unwanted data and how to tackle null values to create a good model. In this project we find various factors responsible for the sale price of a car. Let's take each variable to describe our observations from this project

✓ <u>Brand:</u>

After analyzing the dataset we draw the outcome that in the Indian market the brands like 'Maruti', 'Hyundai', 'Honda' are covering major portion of the market. If we talk about the valuations of the brands then the premium category brands like Mercedes, Audi, KIA, JEEP are having more value.

✓ <u>Purchase Year:</u>

Most of the cars which are listed for the re-sale belongs to the year 2010 to 2018. There was less number of cars who was listed from the year 2020, this may be due to the effect of COVID-19 pandemic on the automobile sector.

✓ Most valuable Models:

Mercedes Benz, Audi, KIA are some of the most valuable models of the cars in the Indian market.

✓ Owner:

We can say that as the number of owners increases the price of the car decreases. This depends on the purchase year of the car also. If the car is recently purchased then the resale value is more.

✓ Fuel Type:

We can say that in the Indian market cars comes with different fuel type option such as Petrol, Diesel, CNG, Hybrid (combination of two fuel option in the same car). But, the dataset says that Petrol cars are more popular in the Indian market and the Diesel cars are costlier. This may be one of the reasons for the popularity of Petrol cars in the Indian market as the cost is less as compared to other combination.

✓ Transmission Type:

We have 3 types of transmission system available in the market: MANUAL, AUTOMATIC & HYBRID. Our dataset contains cars from two category only I.e. MANUAL and AUTOMATIC. We found that the cost of AUTOMATIC transmission type car is more and this may be the reason of less popularity of AUTOMATIC type car in the Indian market.

- Learning Outcomes of the Study in respect of Data Science

By working on this projects, we got to know more about the Indian car market and its resell values. We also come to know about the different features which decide the valuation of the car. This project also given us the exposure to the different types of techniques used in the Machine Learning world specially the Regression type of problems.

- Limitations of this work and Scope for Future Work

  This project was a challenging task for me and this also helped me to sharpen my knowledge. After working on the project, I find some difficulties like the website was not very much updated and there were some missing values and outliers in the different feature columns.

THANK YOU