Car Price Prediction

Submitted by:

Aniruddha Sawant

ACKNOWLEDGMENT

Aniruddha Sawant ("I") acknowledge that I have collected date from 'cars24.com' and used the data and files provided by Flip Robo ("Company") namely 'Problem Statement' to build the predictive model and have used 'sample documentation' for guidance and to write report.

INTRODUCTION

- **Background of the Domain Problem: -**
  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.

- **Business Problem: -**
  There are lot of changes in the automobile market due to COVID 19. Because of this small car traders are suffering the most, hence we are building a car price prediction model to predict the price of used cars. This model will help the car traders to make better decision and decide their car's price.

- **Review of Literature: -**

  There are total 29 features or variables present in the data that would be used to predict the price of a car. But before using those features to predict the outcome we have collected the data, cleaned the data, transform the data into structured format and run many EDAs to make findings. This process ends up in selecting those features which are important to predict the price and build the model.

  Have used total 7 machine learning models to train the data however have chosen LinearRegression Model since its accuracy of train data and test data is high and close to each other i.e., 67.35% and 67.33% respectively.

- **Motivation for the Problem Undertaken: -**
  We are required to model the price of car with the available independent variables. This model will then be used by car trader to understand how exactly the prices vary with the variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be a good way for the management to understand the pricing dynamics of a new market.

Analytical Problem Framing

- **Mathematical/ Analytical Modelling of the Problem**

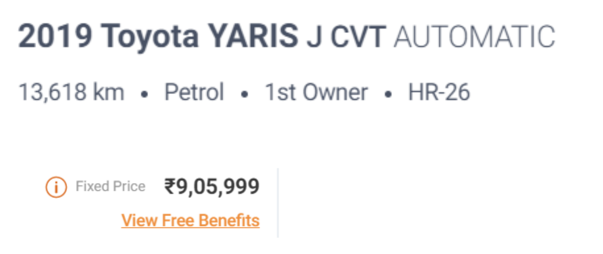
1. Identifying sources of the data
2. Data collection
3. Data structuring
4. Analysing the data
5. Cleaning and processing the data
6. Selecting most important features
7. Writing down findings and observations
8. Using different models to train the data
9. Selecting best fitted model for predictions
10. Predicting outcome for test data

- **Data Sources and their formats**

We have used cars24.com website to collect the data to build the model. We have used selenium to crawl the data. We have created the data set named CarData.csv by crawling the data.

(Note: - Have crawled more than 6,000 of data but there are duplicate data present in the database and it took so many hours to crawl the data hence have not used any different website to crawl again due to system limitations.)

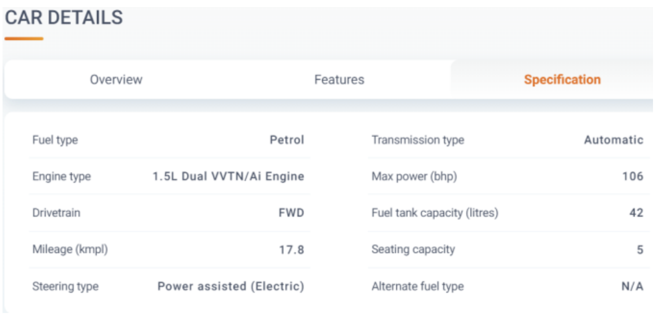
1. Data source sample: - ([Sample link]())


2019 Toyota YARIS J CVT AUTOMATIC

13,618 km • Petrol • 1st Owner • HR-26

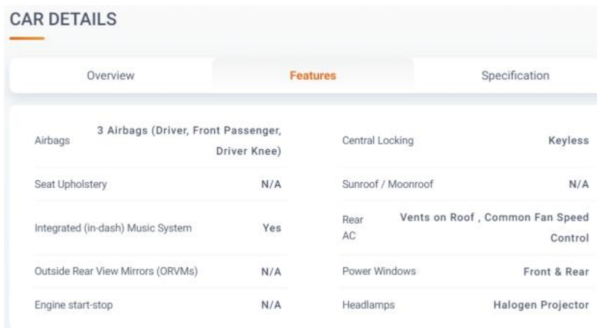
Fixed Price ₹9,05,999

View Free Benefits


CAR DETAILS

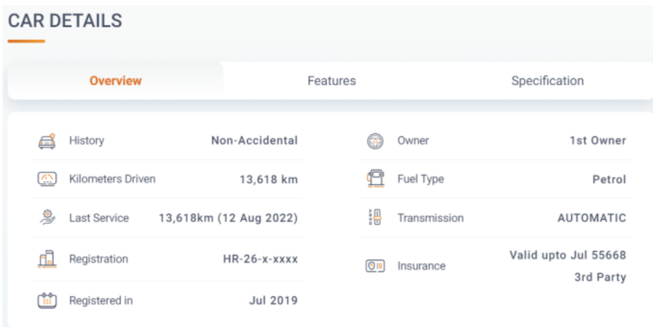
Overview | Features | **Specification**

Fuel type	Petrol	Transmission type	Automatic
Engine type	1.5L Dual VVTN/Ai Engine	Max power (bhp)	106
Drivetrain	FWD	Fuel tank capacity (litres)	42
Mileage (kmpl)	17.8	Seating capacity	5
Steering type	Power assisted (Electric)	Alternate fuel type	N/A


CAR DETAILS

Overview | **Features** | Specification

Airbags	3 Airbags (Driver, Front Passenger, Driver Knee)	Central Locking	Keyless
Seat Upholstery	N/A	Sunroof / Moonroof	N/A
Integrated (in-dash) Music System	Yes	Rear AC	Vents on Roof , Common Fan Speed Control
Outside Rear View Mirrors (ORVMs)	N/A	Power Windows	Front & Rear
Engine start-stop	N/A	Headlamps	Halogen Projector


CAR DETAILS

Overview | Features | Specification

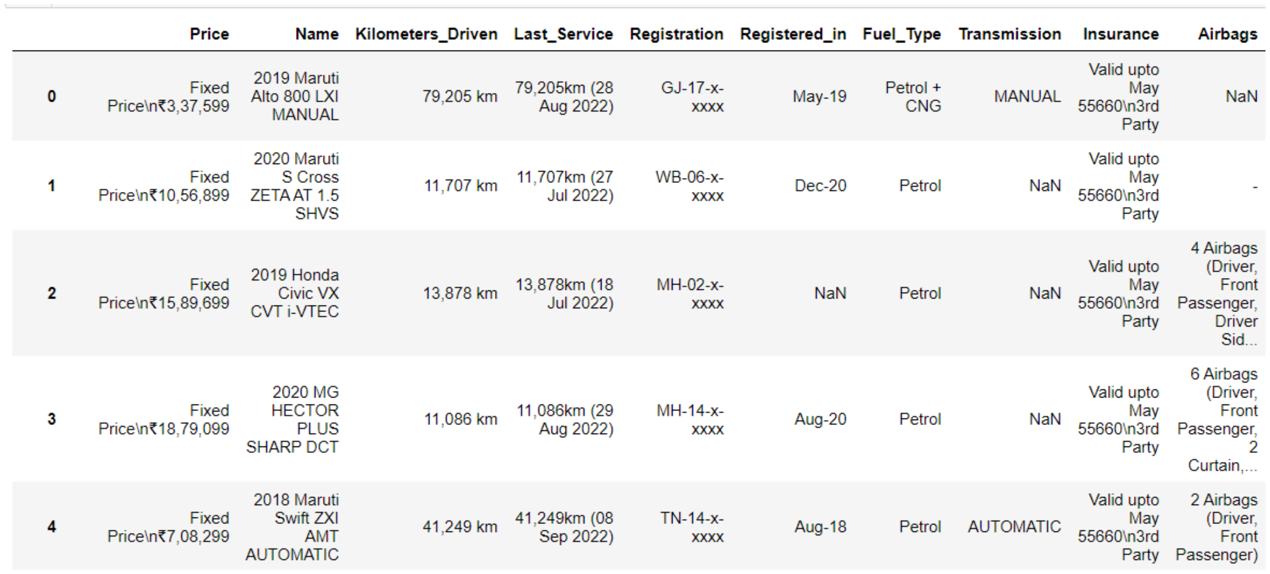
History	Non-Accidental	Owner	1st Owner
Kilometers Driven	13,618 km	Fuel Type	Petrol
Last Service	13,618km (12 Aug 2022)	Transmission	AUTOMATIC
Registration	HR-26-x-xxxx	Insurance	Valid upto Jul 55668 3rd Party
Registered in	Jul 2019		

2. Dataset sample: -

   Below dataset has 29 features and 1 label i.e. Price. Dataset was being cleaned using Python and excel.

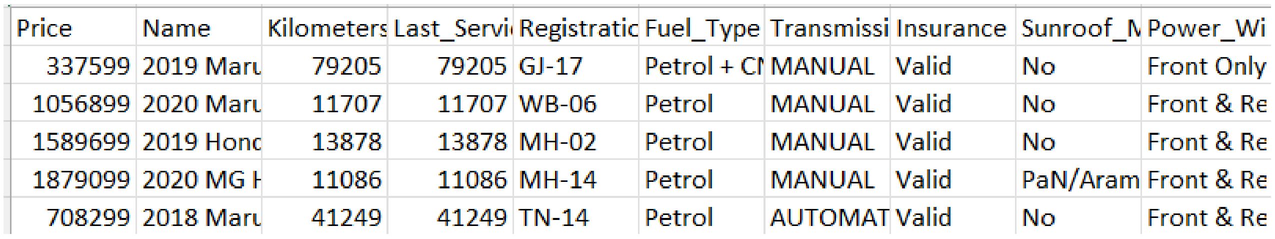
   Samples: -

   Crawled data



	Price	Name	Kilometers_Driven	Last_Service	Registration	Registered_in	Fuel_Type	Transmission	Insurance	Airbags
0	Fixed Price\n₹3,37,599	2019 Maruti Alto 800 LXI MANUAL	79,205 km	79,205km (28 Aug 2022)	GJ-17-x-xxxx	May-19	Petrol + CNG	MANUAL	Valid upto May 55660\n3rd Party	NaN
1	Fixed Price\n₹10,56,899	2020 Maruti S Cross ZETA AT 1.5 SHVS	11,707 km	11,707km (27 Jul 2022)	WB-06-x-xxxx	Dec-20	Petrol	NaN	Valid upto May 55660\n3rd Party	-
2	Fixed Price\n₹15,89,699	2019 Honda Civic VX CVT i-VTEC	13,878 km	13,878km (18 Jul 2022)	MH-02-x-xxxx	NaN	Petrol	NaN	Valid upto May 55660\n3rd Party	4 Airbags (Driver, Front Passenger, Driver Sid...
3	Fixed Price\n₹18,79,099	2020 MG HECTOR PLUS SHARP DCT	11,086 km	11,086km (29 Aug 2022)	MH-14-x-xxxx	Aug-20	Petrol	NaN	Valid upto May 55660\n3rd Party	6 Airbags (Driver, Front Passenger, 2 Curtain,...
4	Fixed Price\n₹7,08,299	2018 Maruti Swift ZXI AMT AUTOMATIC	41,249 km	41,249km (08 Sep 2022)	TN-14-x-xxxx	Aug-18	Petrol	AUTOMATIC	Valid upto May 55660\n3rd Party	2 Airbags (Driver, Front Passenger)

After cleaning using Excel: -



Price	Name	Kilometers	Last_Servi	Registratic	Fuel_Type	Transmissi	Insurance	Sunroof_N	Power_Wi
337599	2019 Maru	79205	79205	GJ-17	Petrol + CI	MANUAL	Valid	No	Front Only
1056899	2020 Maru	11707	11707	WB-06	Petrol	MANUAL	Valid	No	Front & Re
1589699	2019 Honc	13878	13878	MH-02	Petrol	MANUAL	Valid	No	Front & Re
1879099	2020 MG H	11086	11086	MH-14	Petrol	MANUAL	Valid	PaN/Aram	Front & Re
708299	2018 Maru	41249	41249	TN-14	Petrol	AUTOMAT	Valid	No	Front & Re

- **Data Pre-processing**

1. Clean and organized the data.
2. Checked the data type of each column.
3. Changed the Object data type to Integer.
4. Checked whether the data has any Null Values and fill those Null values using Mean and Mode method.
5. Checked whether the data is categorical data or continuous data.
6. There are many categorical columns which has same output with different variable so standardized the data.
   E.g: - In ‘Central_Locking’ column
   'remote':'Remote'
   'yes':'Yes'
   'YES':'Yes’
7. Encoded remaining Object data type columns to integer using encoder technique.
8. Checked the co-relation of features with label i.e Price.
9. Checked the Multicollinearity between features.
10. Checked the VIF score of features.
11. Checked the Distribution of data.
12. Identified and removed outliers those are not allowed above and below the specific limit.

- **Data Inputs**

1. Name – It has the name of the vehicle.
2. Kilometers_Driven – It mentions Kilometers driven by the vehicle.
3. Last_Service – Last service kilometers and date.
4. Registration – Registration number of the vehicle.
5. Registered_in – Registration date.
6. Fuel_Type – Type of fuel.
7. Transmission – Whether the vehicle is Manual or Automatic.
8. Insurance – Insurance date.
9. Airbags – Whether the car have airbags or not.
10. Seat_Upholstery – Material used in car seat.
11. Integrated_Music – Music system present in the car.
12. Rear_View_Mirrors – Whether the car has rear view mirrors or not.
13. Engine_start_stop – Which type of engine the vehicle has.
14. Central_Locking – Which type of locking system the car has.
15. Sunroof_Moonroof – Whether the vehicle has roof or not.
16. Rear_AC – Description on the rear AC in the car.
17. Power_Windows - Description on the Power Windows in the car.
18. Headlamps – Type of headlamp the car has.
19. Engine_type – Type of engine.
20. Drivetrain – What type of drivetrain the vehicle has.
21. Mileage – Mileage of the company.
22. Steering_type – What type of steering type the company has.
23. Transmission_type – Type of transmission the vehicle has.
24. Max_power – Max power of the car.
25. Fuel_tank_capacity – Capacity of the fuel tank in the car.
26. Seating_capacity – Seating capacity of car.
27. Alternate_fuel_type – Whether the car allows alternate fuel type.
28. History – History of the vehicle.
29. Price – Price of the vehicle and this is the output column.

- **Hardware and Software Requirements and Tools Used**

1. Libraries and packages used
    - import numpy as np – For Numpy work
    - import pandas as pd – To work on DataFrame
    - import seaborn as sns – Plotting Graphs
    - import matplotlib.pyplot as plt - Plotting Graphs
    - import pickle – To save the Model
    - from sklearn.preprocessing import StandardScaler (To scale the train data), OrdinalEncoder(To encode object data to Integer), PowerTransformer (To remove skewness from dataset)
    - enc = OrdinalEncoder() = Assigned OrdinalEncoder to variable
    - from statsmodels.stats.outliers_influence import variance_inflation_factor – To calculate VIF score
    - from sklearn.model_selection import train_test_split – To split the data into train and test.
    - from sklearn import metrics, from sklearn.linear_model import Ridge, Lasso, RidgeCV, LassoCV – To regularize the model.
    - from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score, accuracy_score – To calculate and analyse model metrics.
    - import warnings, warnings.filterwarnings('ignore') – To ignore unwanted Warnings
2. Machine Learning models used
    - from sklearn.linear_model import LinearRegression
    - from sklearn.tree import DecisionTreeRegressor
    - from sklearn.ensemble import AdaBoostRegressor
    - from sklearn.ensemble import GradientBoostingRegressor
    - from sklearn.ensemble import RandomForestRegressor
    - from sklearn.neighbors import KNeighborsRegressor
3. Hardware used – 11th Gen Intel(R) Core (TM) i3-1115G4 @ 3.00GHz   3.00 GHz with 8.00 GB RAM and Windows 11
4. Software used – Anaconda and Jupyter Notebook to build the model and Excel to clean and structured the data.

Model/s Development and Evaluation

- **Identification of possible problem**
  1. The data was not structured and organized hence cleaned the data using various data cleaning and pre-processing techniques.
  2. There are many outliers present in the data hence removed outliers.
  3. There was a skewness in the data hence have removed the skewness from the data.
  4. Scaled the data using Standard Scalar to make the data standardized to build a model.

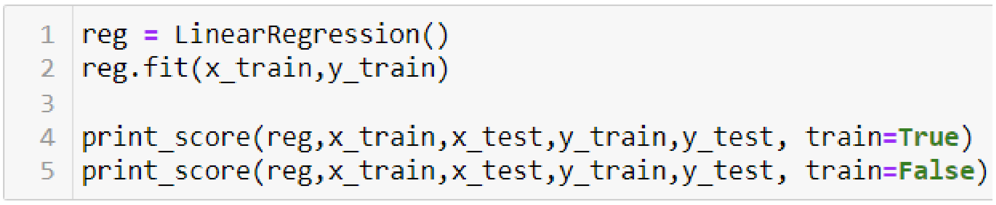
- **Testing of Identified Approaches**

  These are the algorithms which have been used to train and test data.

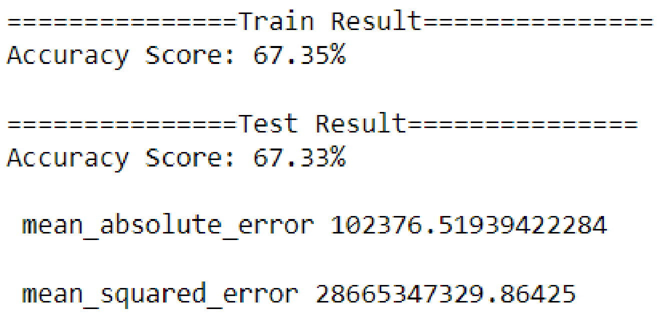
  1. LinearRegression
  2. DecisionTreeRegressor
  3. AdaBoostRegressor
  4. GradientBoostingRegressor
  5. RandomForestRegressor
  6. KNeighborsRegressor
  7. Support Vector Regression

- **Run and evaluate selected models**

1. **LinearRegression**: -

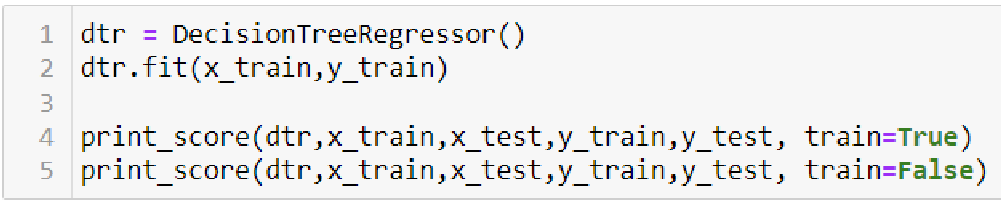


```
reg = LinearRegression()
reg.fit(x_train,y_train)

print_score(reg,x_train,x_test,y_train,y_test, train=True)
print_score(reg,x_train,x_test,y_train,y_test, train=False)
```

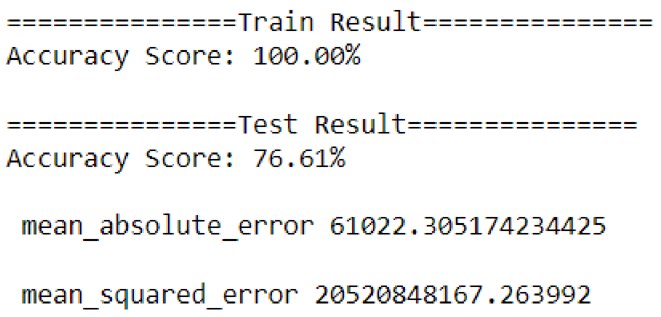


```
===============Train Result===============
Accuracy Score: 67.35%

===============Test Result===============
Accuracy Score: 67.33%

 mean_absolute_error 102376.51939422284

 mean_squared_error 28665347329.86425
```

2. **DecisionTreeRegressor**: -



```
dtr = DecisionTreeRegressor()
dtr.fit(x_train,y_train)

print_score(dtr,x_train,x_test,y_train,y_test, train=True)
print_score(dtr,x_train,x_test,y_train,y_test, train=False)
```



```
===============Train Result===============
Accuracy Score: 100.00%

===============Test Result===============
Accuracy Score: 76.61%

 mean_absolute_error 61022.305174234425

 mean_squared_error 20520848167.263992
```

3. **AdaBoostRegressor**: -

```
ada = AdaBoostRegressor()
ada.fit(x_train,y_train)

print_score(ada,x_train,x_test,y_train,y_test, train=True)
print_score(ada,x_train,x_test,y_train,y_test, train=False)
```

```
===============Train Result===============
Accuracy Score: 57.60%

===============Test Result===============
Accuracy Score: 55.07%

 mean_absolute_error 153793.32955713593

 mean_squared_error 39415070278.03112
```

4. **GradientBoostingRegressor**: -

```
gbdt = GradientBoostingRegressor()
gbdt.fit(x_train,y_train)

print_score(gbdt,x_train,x_test,y_train,y_test, train=True)
print_score(gbdt,x_train,x_test,y_train,y_test, train=False)
```

```
===============Train Result===============
Accuracy Score: 85.11%

===============Test Result===============
Accuracy Score: 79.90%

 mean_absolute_error 75192.5135144641

 mean_squared_error 17635591320.572407
```

5. **RandomForestRegressor**: -

```
rfr = RandomForestRegressor()
rfr.fit(x_train,y_train)

print_score(rfr,x_train,x_test,y_train,y_test, train=True)
print_score(rfr,x_train,x_test,y_train,y_test, train=False)
```

```
===============Train Result===============
Accuracy Score: 97.84%

===============Test Result===============
Accuracy Score: 86.79%

 mean_absolute_error 54218.23786166842

 mean_squared_error 11587753426.816917
```

6. **KNeighborsRegressor**: -

```
knr = KNeighborsRegressor()
knr.fit(x_train,y_train)

print_score(knr,x_train,x_test,y_train,y_test, train=True)
print_score(knr,x_train,x_test,y_train,y_test, train=False)
```

```
===============Train Result===============
Accuracy Score: 82.39%

===============Test Result===============
Accuracy Score: 71.56%

 mean_absolute_error 89830.3626187962

 mean_squared_error 24947042897.9007
```

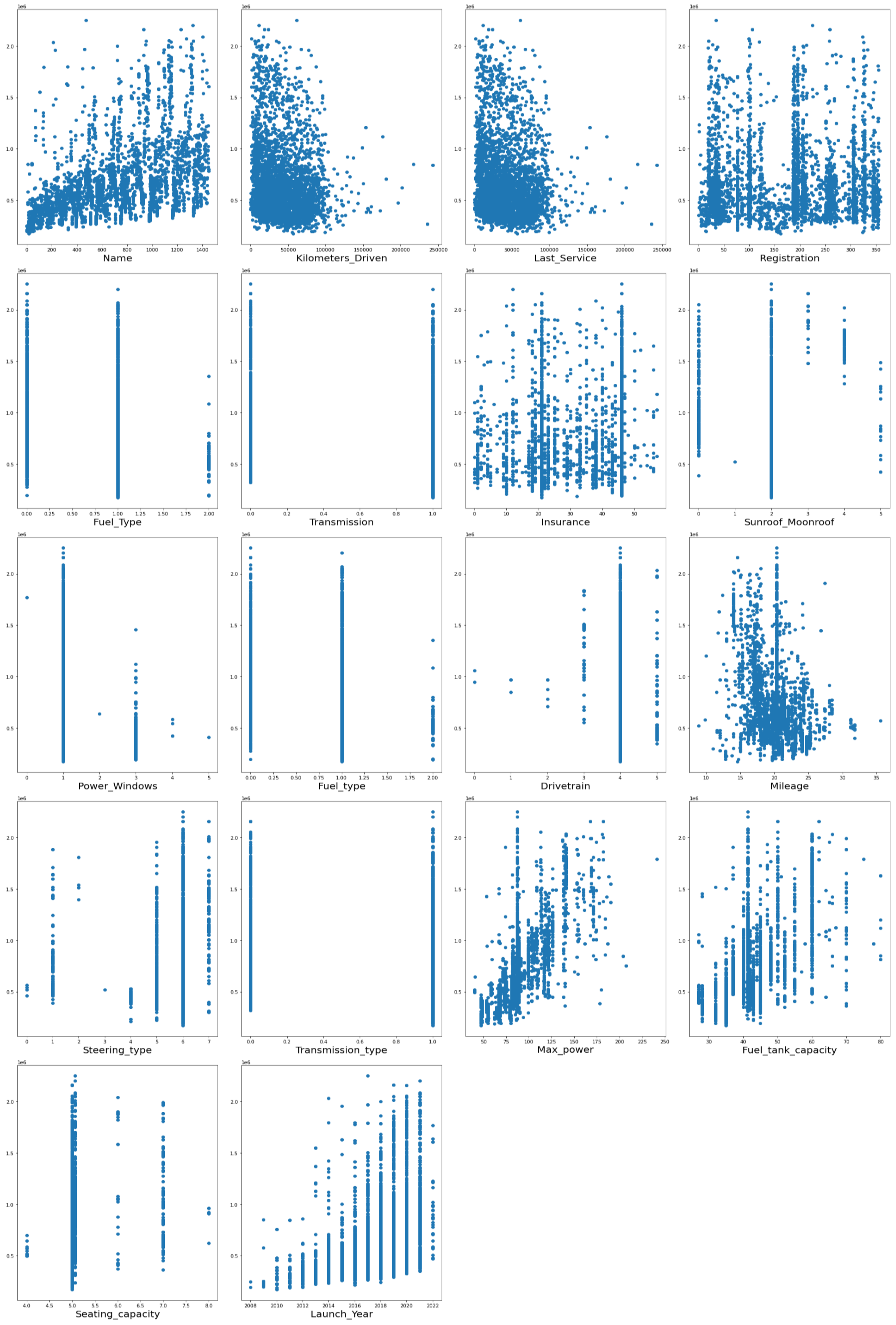
7. **Support Vector Regression: -**

```
svr = SVR()
svr.fit(x_train,y_train)

print_score(svr,x_train,x_test,y_train,y_test, train=True)
print_score(svr,x_train,x_test,y_train,y_test, train=False)
```

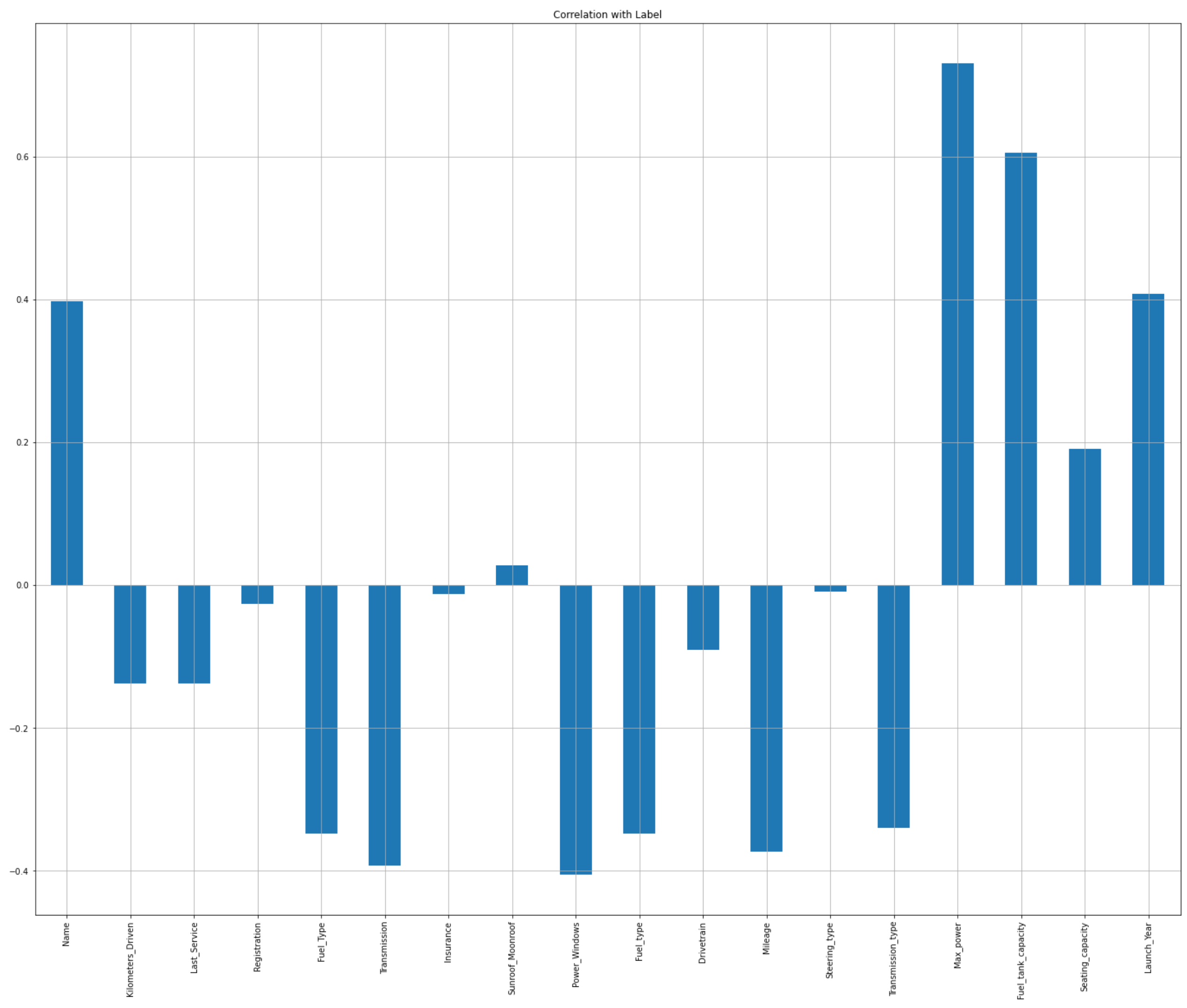
```
===============Train Result===============
Accuracy Score: -5.74%

===============Test Result===============
Accuracy Score: -4.99%

 mean_absolute_error 205857.07725353728

 mean_squared_error 92104971953.68492
```

- **Visualizations**

1. Scatter Plot: -



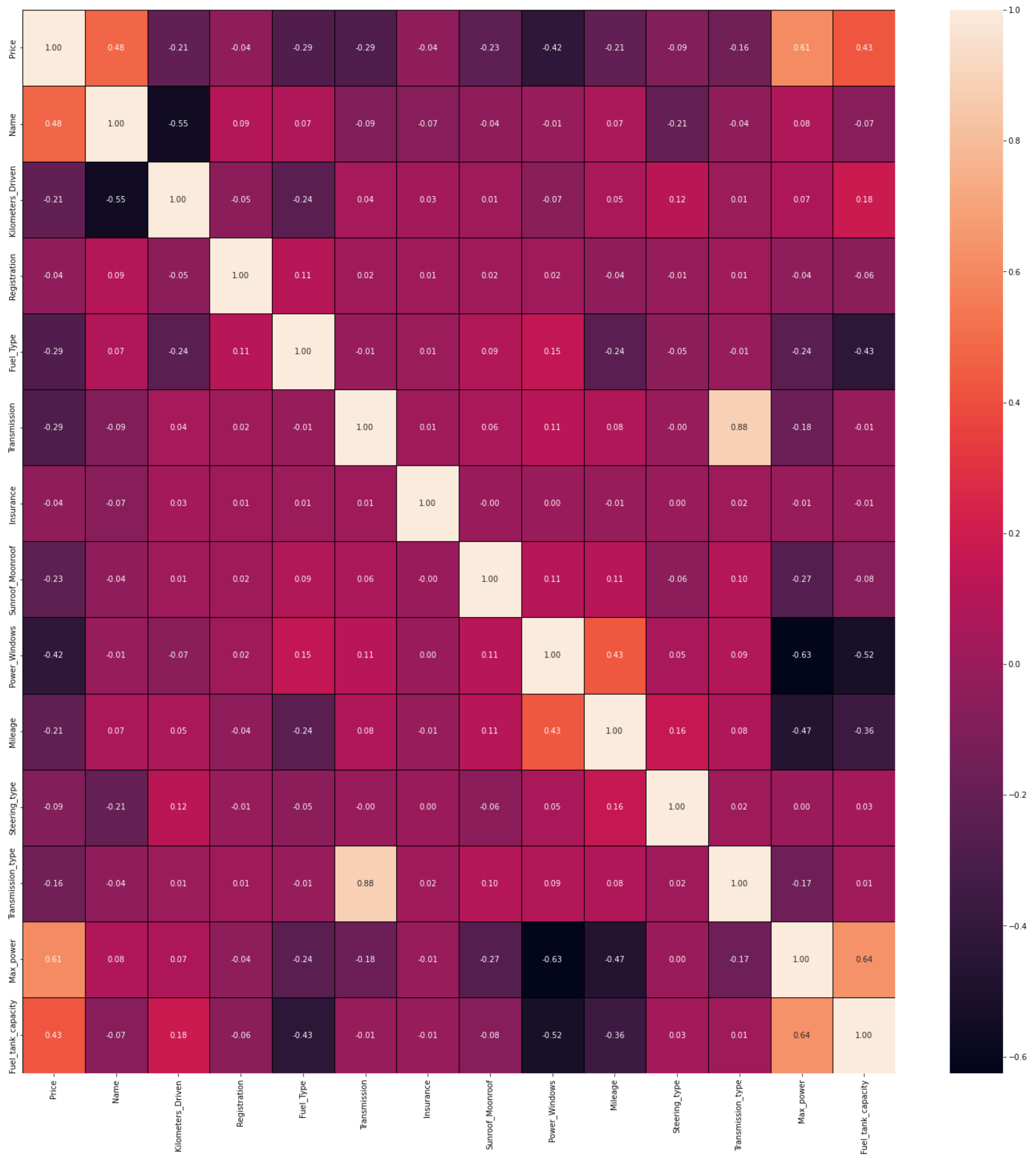
2. BarPlot: -



Observations:

- Name, Fuel_Type, Transmission, Power_Windows, Mileage, Max_power, Fuel_tank_capacity and Launch_Year has high co-relation with Label.
- Registration, Insurance, Sunroof_Moonroof and Steering_type columns has Low/No co-relation with Label.

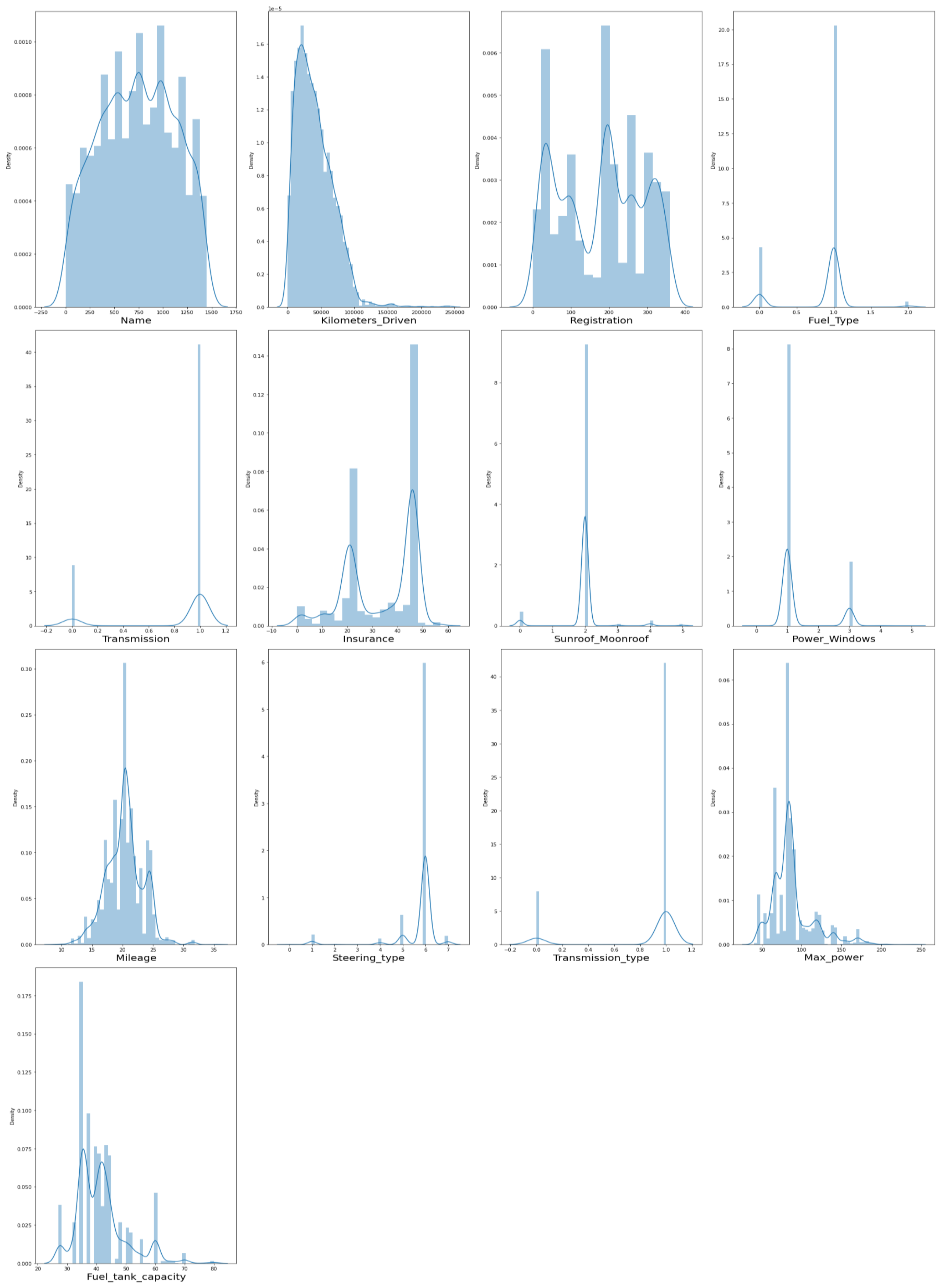
3. HeatMap: -



Observations:

- Transmission_type and Transmission has Multicollinearity problem.

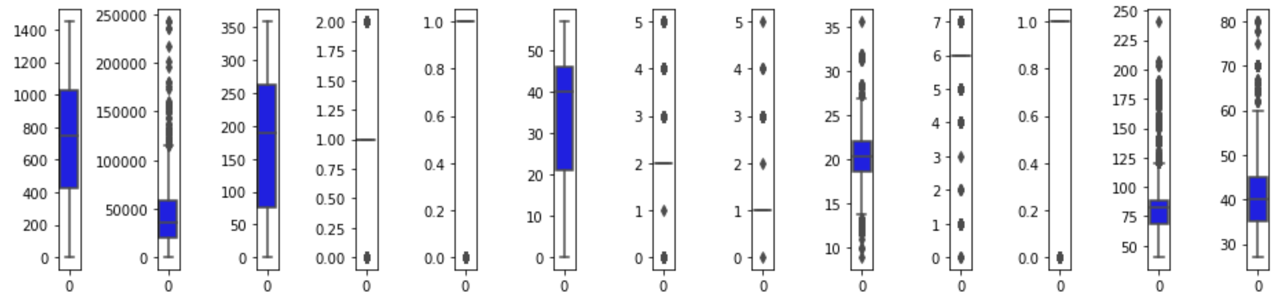
4. Distribution Plot: -



Observations:

- Not considering skewness of categorical data columns.
- Kilometers_Driven, Max_power and Fuel_tank_capacity has skewness.

5. Box Plot: -



Observations:

- Kilometers_Driven, Mileage, Max_power and Fuel_tank_capacity Column has outliers.

- **Interpretation of the Results**

  Below is the list of highly influencing features or variables to predict the sales price of the house.

  1. Launch_Year = It is very important that the car launch year should be latest which leads to high price of the car.
  2. Fuel_tank_capacity = High fuel tank capacity leads to high price of car.
  3. Max_power = Max power plays an important role in decides the price of the car.
  4. Power_Windows = Types of power windows also impacts the price of the car.

CONCLUSION

- **Key Findings and Conclusions of the Study**
  5. Selecting LinearRegression model since the Accuracy score i.e., 67.35% and test scores i.e., 67.33 % are greater and close to each other.
  6. mean_absolute_error is also low for LinearRegression model.
  7. LinearRegression model is not overfitted since r2_score of LassoCV and RidgeCV is same and close to test score i.e., 67.32%.

- **Learning Outcomes of the Study in respect of Data Science**
  1. Data Cleaning helps to convert unorganized and unstructured data into structured data which will be used to make findings.
  2. Data visualization helps understand and analyse the data.
  3. Model building helps to predict outcomes, in this case LinearRegression model fits perfect for this dataset.
  4. While doing pre-processing the high VIF problem arises as many highly co-related features have high VIF score.

- **Limitations of this work and Scope for Future Work**
  1. There are 30 features present in the dataset however due to pre-processing and visualization we have cutdown some features, hence this could become a disadvantage in the future as we update the dataset and there is a possibility that we may lose some important information.
  2. It is necessary to keep an eye on new and updated data to further train the model and make decision as per new data.