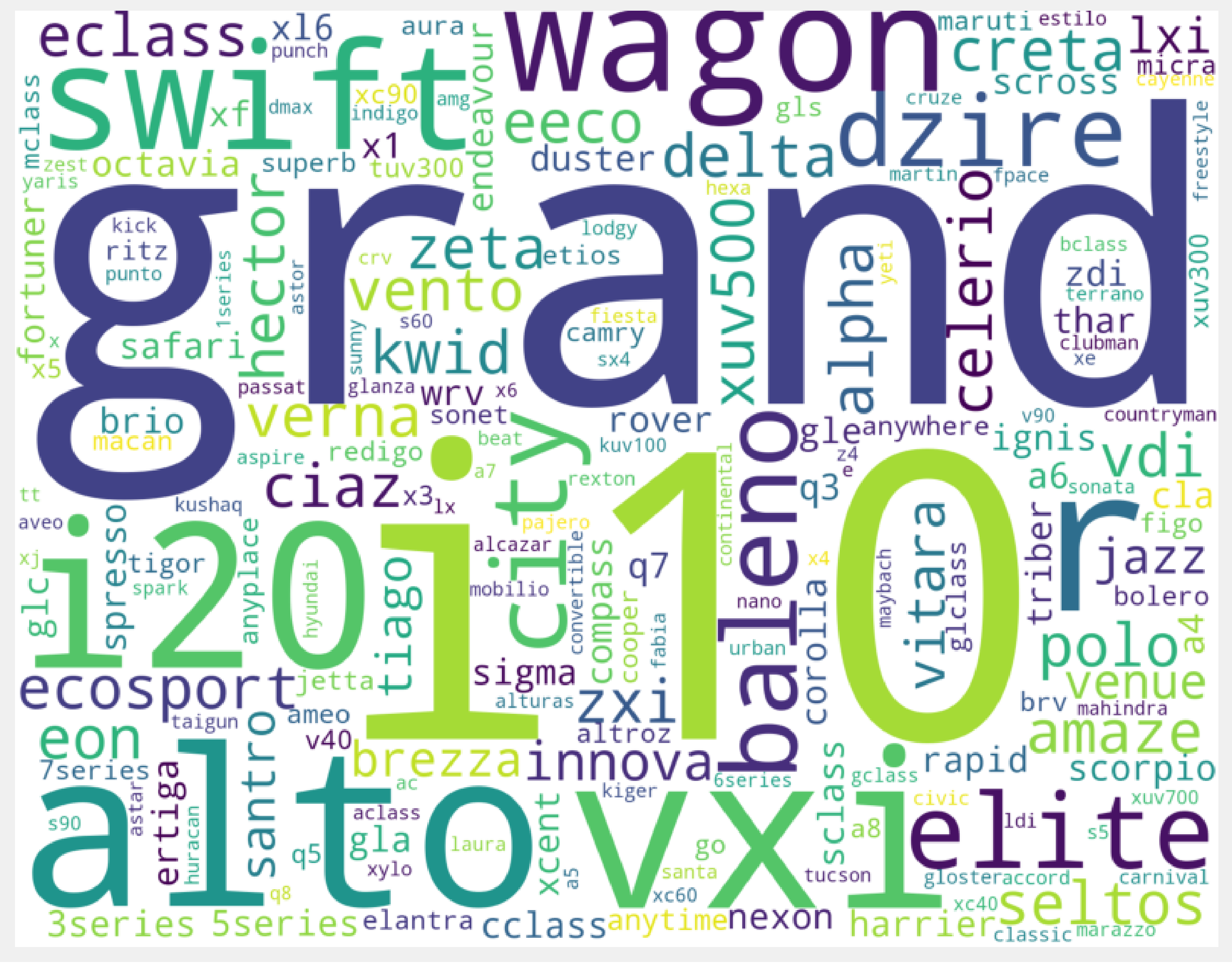
FLIP ROBO

USED CAR PRICE PREDICTION



Submitted by:

SANTOSH H. HULBUTTI

Table of Contents

ACKNOWLEDGMENT

This project is completed using knowledge/information available on internet.

Following are the websites & YouTube Channels, which were used to scape data & understand concepts related to ML, AI & Data Visualization.

Websites:

1. towardsdatascience.com
2. medium.com
3. analyticsvidya.com
4. DataTrained LMS Platform
5. Carwale.com
6. Official documentation of ScikitLearn, Matplot library, AutoViz, Sweet Viz, Pandas Library & Seaborn library.
7. Kaggle.com
8. UCI ML Repository
9. Stackoverflow.com
10. YouTube Channels:
    a. Krish Naik
    b. Sidhdhardan
    c. Keith Galli

I would like to thank FlipRobo Technologies, for giving an opportunity to work as an intern during this project period. And also like to thank mentor Ms. Gulshana Chaudhary for assigning the project.

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 FlipRobo Technologies' clients works with small traders, who sell used cars. With the change in market due to covid 19 impact, FlipRobo Technologies' 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.

Conceptual Background of the Domain Problem

Determining whether the listed price of a used car is a challenging task, due to the many factors that drive a used vehicle's price on the market. The focus of this project is developing machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases.is considerable risk of default involved, because the loan is being provided to low-income populations.

Accurate car price prediction involves expert knowledge, because price usually depends on many distinctive features and factors. Typically, most significant ones are brand and model, age, power and mileage. The fuel type used in the car, different features like colour, type of transmission, dimensions, & other features influence the car price. In this project, we applied different methods and techniques in order to achieve higher precision of the used car price prediction.

Regression Algorithms are used because they provide us with continuous value as an output and not a categorized value because of which it will be possible to predict the actual price a car rather than the price range of a car.

Review of Literature

Data was collected from 'Carwale.com'. This data was cleaned and analysed. It showed the impact that various factors had on the price of the car. Model is then created using the data by splitting into dependent and independent variable, followed by train test split to train the regression models.

The algorithm having the lease difference between r2 score of test-set and cross validation score will be used for hyperparameter tuning. The best parameters are used to tune the model. This model is given to the client in further using to visualise data for car price prediction. We have used machine learning model to predict the above.

We will look at all the features with following goals in mind:

- Relevance of the feature
- Distribution of the feature
- Cleaning the feature
- Visualization of the feature
- Visualization of the feature as per loan default status for data analysis

After having gone through all the features and cleaning the dataset, we will move on to machine learning regression modelling:

- Pre-processing the dataset for models
- Testing multiple algorithms with multiple evaluation metrics
- Select evaluation metric as per our specific business application
- Hyper-parameter tuning using GridSearchCV for the best model parameter
- And finally saving the best model

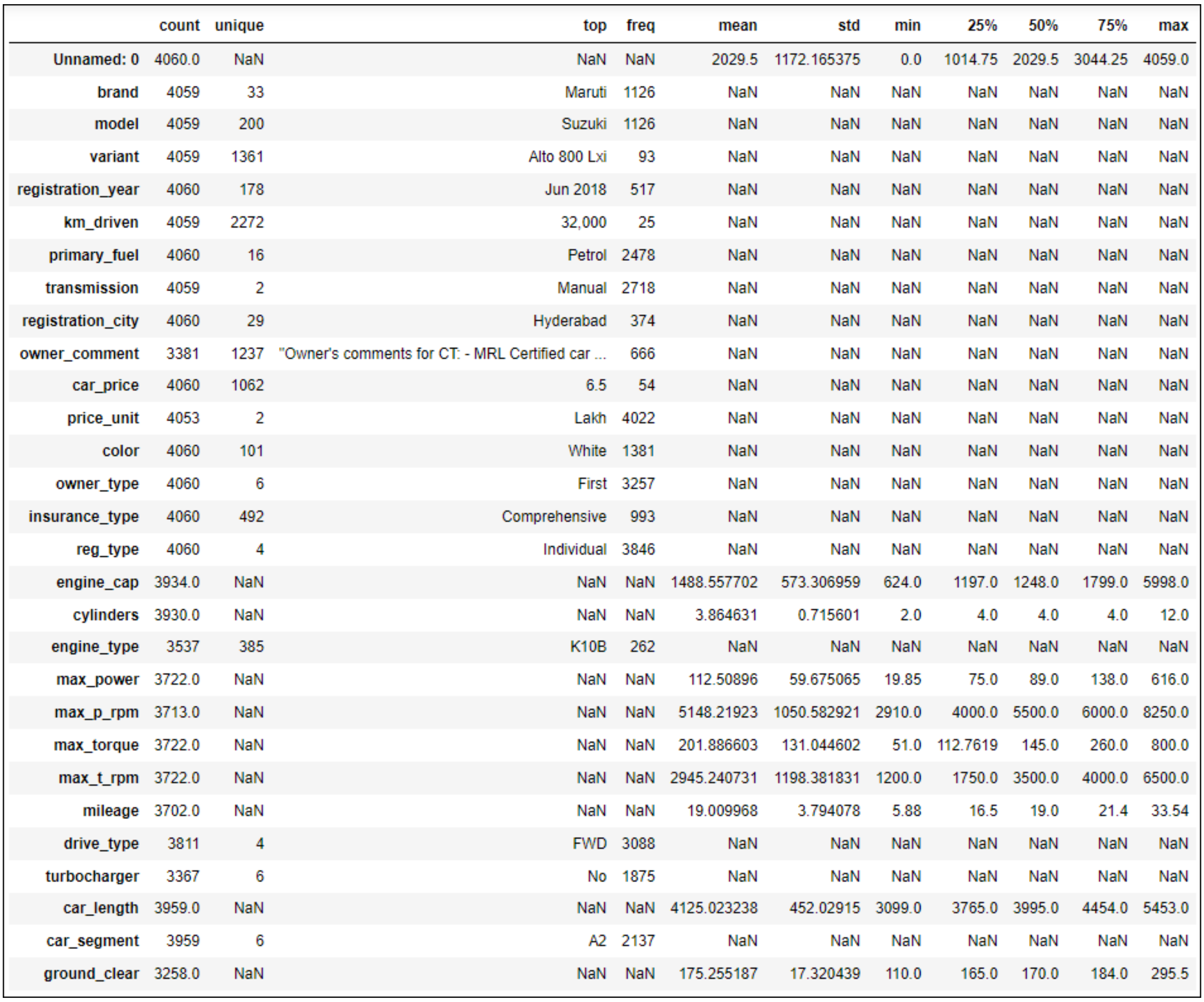
Motivation for the Problem Undertaken

Car has become a significant part of most of the households, specially where the public transport is not advanced. Hence Used car plays the pivotal role among cars as it expands the market of cars to a wider population. This is an opportunity to grab to make a profitable business. Providing best service/product at affordable price will attract more customer & thereby increasing the profit & business growth.

ANALYTICAL PROBLEM FRAMING

Mathematical/ Analytical Modeling of the Problem

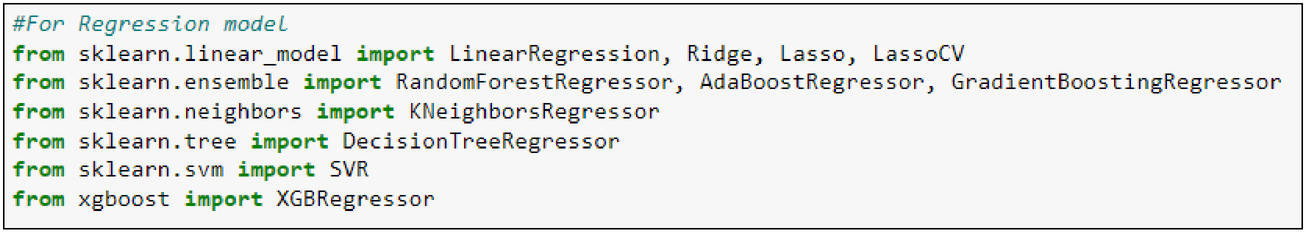
(Describe the mathematical, statistical and analytics modelling done during this project along with the proper justification.)



	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Unnamed: 0	4060.0	NaN	NaN	NaN	2029.5	1172.165375	0.0	1014.75	2029.5	3044.25	4059.0
brand	4059	33	Maruti	1126	NaN	NaN	NaN	NaN	NaN	NaN	NaN
model	4059	200	Suzuki	1126	NaN	NaN	NaN	NaN	NaN	NaN	NaN
variant	4059	1361	Alto 800 Lxi	93	NaN	NaN	NaN	NaN	NaN	NaN	NaN
registration_year	4060	178	Jun 2018	517	NaN	NaN	NaN	NaN	NaN	NaN	NaN
km_driven	4059	2272	32,000	25	NaN	NaN	NaN	NaN	NaN	NaN	NaN
primary_fuel	4060	16	Petrol	2478	NaN	NaN	NaN	NaN	NaN	NaN	NaN
transmission	4059	2	Manual	2718	NaN	NaN	NaN	NaN	NaN	NaN	NaN
registration_city	4060	29	Hyderabad	374	NaN	NaN	NaN	NaN	NaN	NaN	NaN
owner_comment	3381	1237	"Owner's comments for CT: - MRL Certified car ...	666	NaN	NaN	NaN	NaN	NaN	NaN	NaN
car_price	4060	1062	6.5	54	NaN	NaN	NaN	NaN	NaN	NaN	NaN
price_unit	4053	2	Lakh	4022	NaN	NaN	NaN	NaN	NaN	NaN	NaN
color	4060	101	White	1381	NaN	NaN	NaN	NaN	NaN	NaN	NaN
owner_type	4060	6	First	3257	NaN	NaN	NaN	NaN	NaN	NaN	NaN
insurance_type	4060	492	Comprehensive	993	NaN	NaN	NaN	NaN	NaN	NaN	NaN
reg_type	4060	4	Individual	3846	NaN	NaN	NaN	NaN	NaN	NaN	NaN
engine_cap	3934.0	NaN	NaN	NaN	1488.557702	573.306959	624.0	1197.0	1248.0	1799.0	5998.0
cylinders	3930.0	NaN	NaN	NaN	3.864631	0.715601	2.0	4.0	4.0	4.0	12.0
engine_type	3537	385	K10B	262	NaN	NaN	NaN	NaN	NaN	NaN	NaN
max_power	3722.0	NaN	NaN	NaN	112.50896	59.675065	19.85	75.0	89.0	138.0	616.0
max_p_rpm	3713.0	NaN	NaN	NaN	5148.21923	1050.582921	2910.0	4000.0	5500.0	6000.0	8250.0
max_torque	3722.0	NaN	NaN	NaN	201.886603	131.044602	51.0	112.7619	145.0	260.0	800.0
max_t_rpm	3722.0	NaN	NaN	NaN	2945.240731	1198.381831	1200.0	1750.0	3500.0	4000.0	6500.0
mileage	3702.0	NaN	NaN	NaN	19.009968	3.794078	5.88	16.5	19.0	21.4	33.54
drive_type	3811	4	FWD	3088	NaN	NaN	NaN	NaN	NaN	NaN	NaN
turbocharger	3367	6	No	1875	NaN	NaN	NaN	NaN	NaN	NaN	NaN
car_length	3959.0	NaN	NaN	NaN	4125.023238	452.02915	3099.0	3765.0	3995.0	4454.0	5453.0
car_segment	3959	6	A2	2137	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ground_clear	3258.0	NaN	NaN	NaN	175.255187	17.320439	110.0	165.0	170.0	184.0	295.5

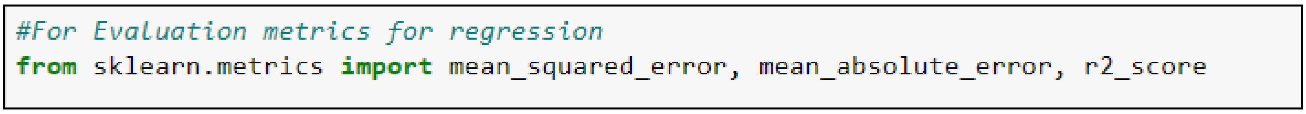
- There was total **4060** rows of observation.
- 4 features namely, **'Unnamed: 0', 'owner_comment', 'variant, 'price_unit',** were removed as they carry no value for predicting price of the car.
- Statistical techniques used:
    - Skewness check using **'.skew()'** method & removing using power transformation method,
    - Outliers' removal using **'Z-Score'** method (3 Std deviation method),
    - Correlation check using **'.corr()'** & heatmap method,
    - Minimizing Multi collinearity using **'Variance Inflation Factor(VIF)',**
    - Scaling input data using **'StandardScaler()'** method**,**
    - Graphical modelling done through seaborn, matplotlib.
- After Pre processing we used **'.describe()'** method to check description of the data.

- Machine Learning algorithms used:



```
#For Regression model
from sklearn.linear_model import LinearRegression, Ridge, Lasso, LassoCV
from sklearn.ensemble import RandomForestRegressor, AdaBoostRegressor, GradientBoostingRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from xgboost import XGBRegressor
```

- Model Evaluation metrics used:



```
#For Evaluation metrics for regression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

- The final model was tuned using Hyper parameter tuning & validated using Cross validation score.

Data Sources and their formats

- The data was scraped using selenium library & saved in **CSV** format.
- There were 28 attributes (**27 features and 1 target**).
- The target variable is continuous numeric data.
- Following are the relevant features in the data;

Brand *– brand name of the car*

Model *– model name of the car*

km_driven *– total kilometers driven by the car*

primary_fuel *– primary fuel feeded to the car*

transmission *– type of transmission/gear train mechanism*

registration_city *– name of the city where the car registration was done*

car_price *– price of the car in Lakhs of rupees (Target variable)*

color *– color of the car*

owner_type *– level of Ownership transfer*

insurance_type *– is car insured as on the day of listing it on the website*

reg_type *– type of registration of car (individual/commercial/taxi/corporate etc)*

engine_cap *– volumetric capacity of the engine*

cylinders *– no. of cylinders in engine*

max_power *– max power of the engine*

max_p_rpm *– rpm at max power*

max_torque *– max torque of the engine*

max_t_rpm *– rpm at the max torque*

mileage *– average kilometers drive per 1 liter of fuel*

drive_type *– rear wheel drive/ front wheel drive/ all-wheel drive*

turbocharger *– is the car has turbocharger*

car_segment *– car segment based on the car length*

ground_clear *– ground clearance*

registration_month *– month of the registration*

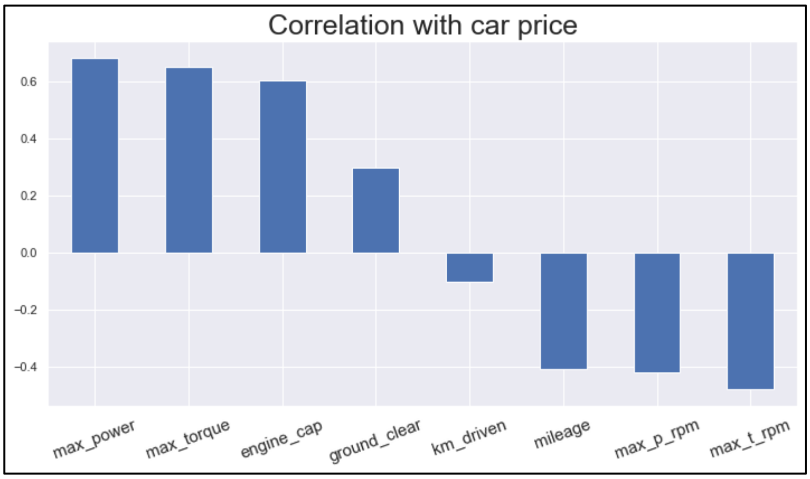
registration_year2 *– year of the registration*

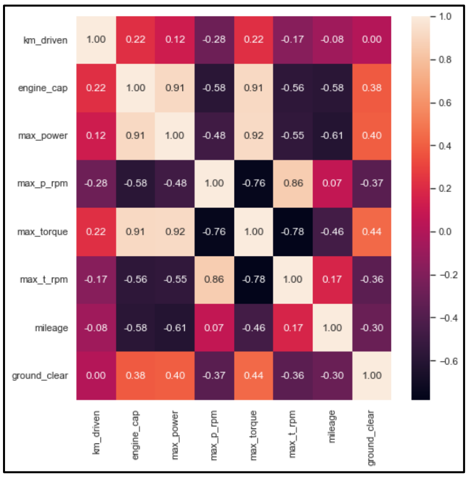
Data Pre-processing Done

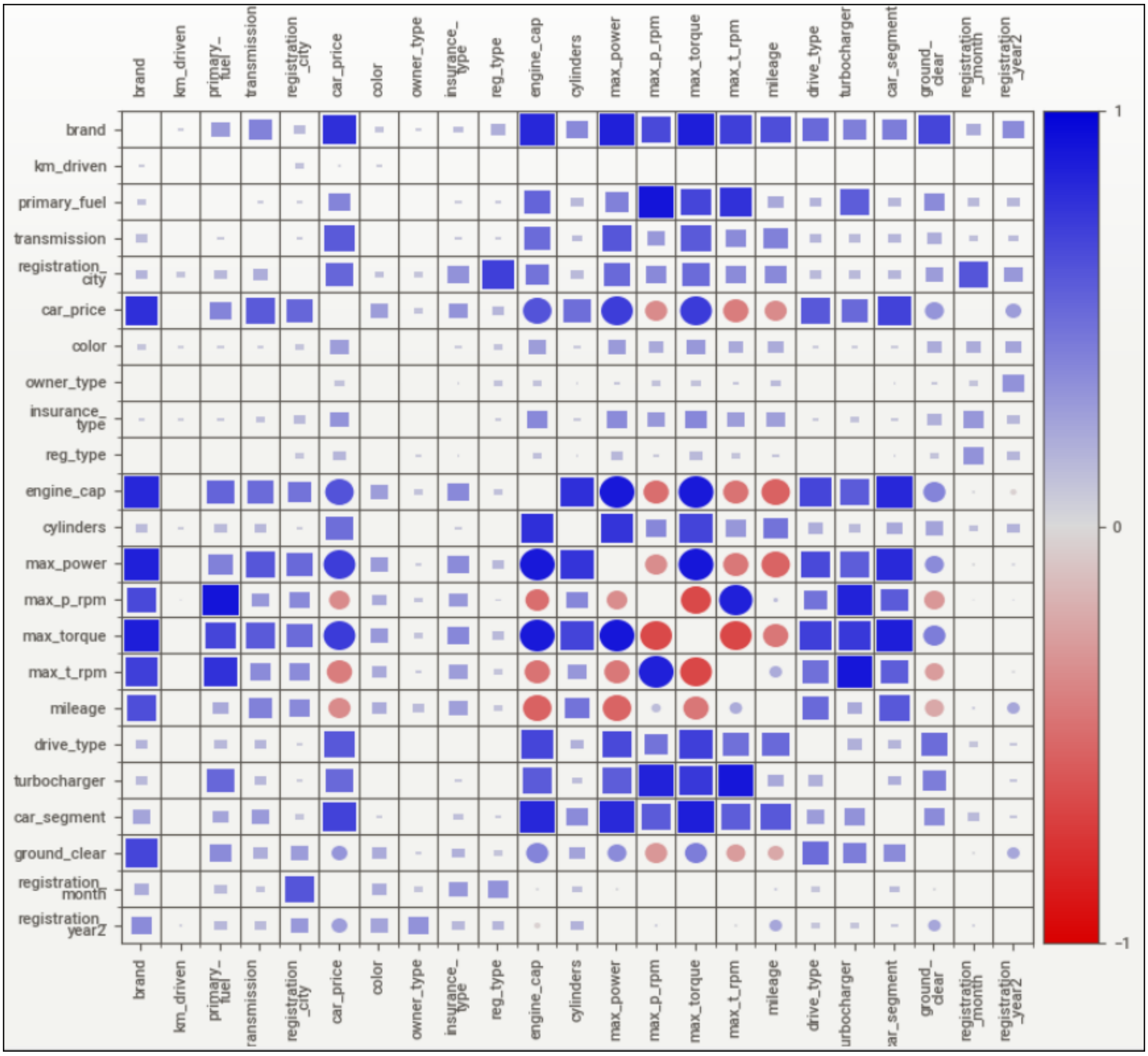
1. Data Imported using Pandas **‘.read_csv()’** method,
2. Dropping unnecessary columns/features
3. Dropping Duplicate entries,
4. Checking for data consistency & unusual data entries,
5. Checking for unique entries, null values,
6. Checking for datatype count,
7. Skewness is removed using Yeo-Johnson Power Transformer.
8. Outliers are removed using Z-score method. Data loss observed was 1.13%.
9. Some features are removed as based on Correlation using seaborn heatmap & Multicollinearity check using VIF value:
10. Standard scaling is applied on the entire train & test data.
11. We used train_test_split to split data for machine learning.

Data Inputs- Logic- Output Relationships

Following plot shows the relation between numerical features and target variable







■ Squares are categorical associations (uncertainty coefficient & correlation ratio) from 0 to 1. The uncertainty coefficient is asymmetrical, (i.e. ROW LABEL values indicate how much they PROVIDE INFORMATION to each LABEL at the TOP).

● Circles are the symmetrical numerical correlations (Pearson's) from -1 to 1. The trivial diagonal is intentionally left blank for clarity.

Hardware and Software Requirements and Tools Used

a. Software

i. ➔ Jupyter Notebook (Python 3.9)
ii. ➔ Microsoft Office

b. Hardware

i. ➔ Processor – AMD Ryzen 5
ii. ➔ RAM - 8 GB
iii. ➔ Graphic Memory - 4Gb, Nvidia GEFORCE RTX1650

c. Python Libraries

i. ➔ Pandas
ii. ➔ Numpy
iii. ➔ Selenium
iv. ➔ Matplotlib
v. ➔ Seaborn
vi. ➔ Scipy
vii. ➔ Sklearn
viii. ➔ AutoViz & SweetViz

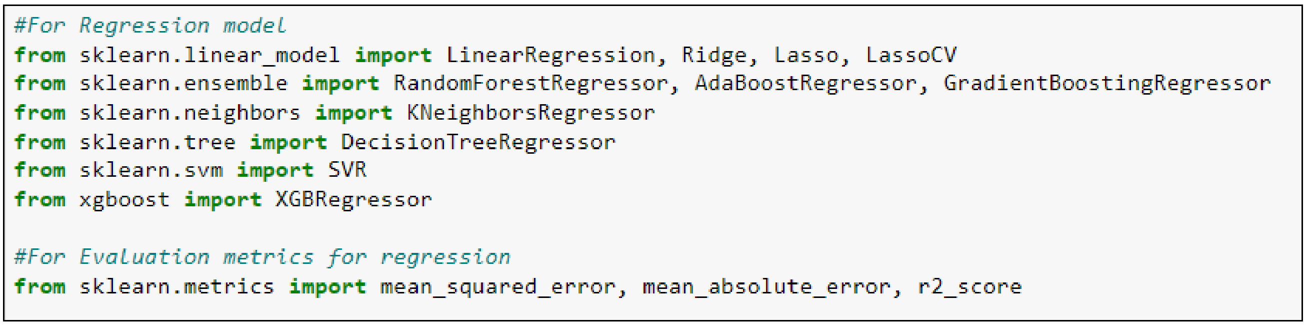
MODEL/S DEVELOPMENT AND EVALUATION

Identification of possible problem-solving approaches (methods)

The data set was was analysed both statistically and graphically. The statistical analysis showed that,

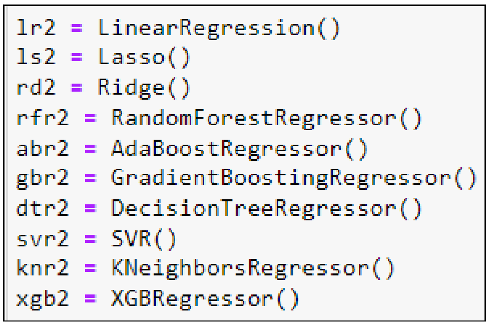
1. data has outliers, skewness, null values & zero values
2. independent variables were continuous & discrete numerical, nominal & categorical type data.
3. Data was cleaned missing values were treated using groupby function. Unrealistic data was removed.
4. Outliers were removed using z-score method, about 1.74% of data removed.
5. Skewness of numerical columns were transformed using yeo-Johnson method to have within allowed limits of +/-0.5.
6. Some features were dropped as the entries were did not have any meaning with respect to target variable.
7. Total data loss in pre-processing was 2.38%.

Testing of Identified Approaches (Algorithms)



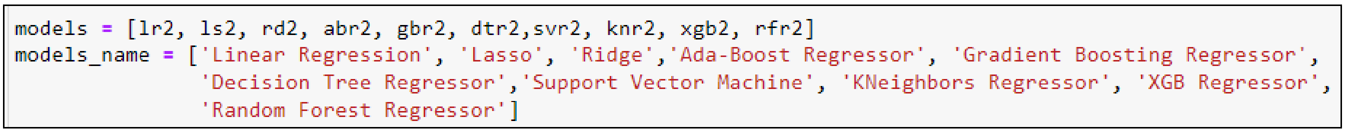
```
#For Regression model
from sklearn.linear_model import LinearRegression, Ridge, Lasso, LassoCV
from sklearn.ensemble import RandomForestRegressor, AdaBoostRegressor, GradientBoostingRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from xgboost import XGBRegressor

#For Evaluation metrics for regression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

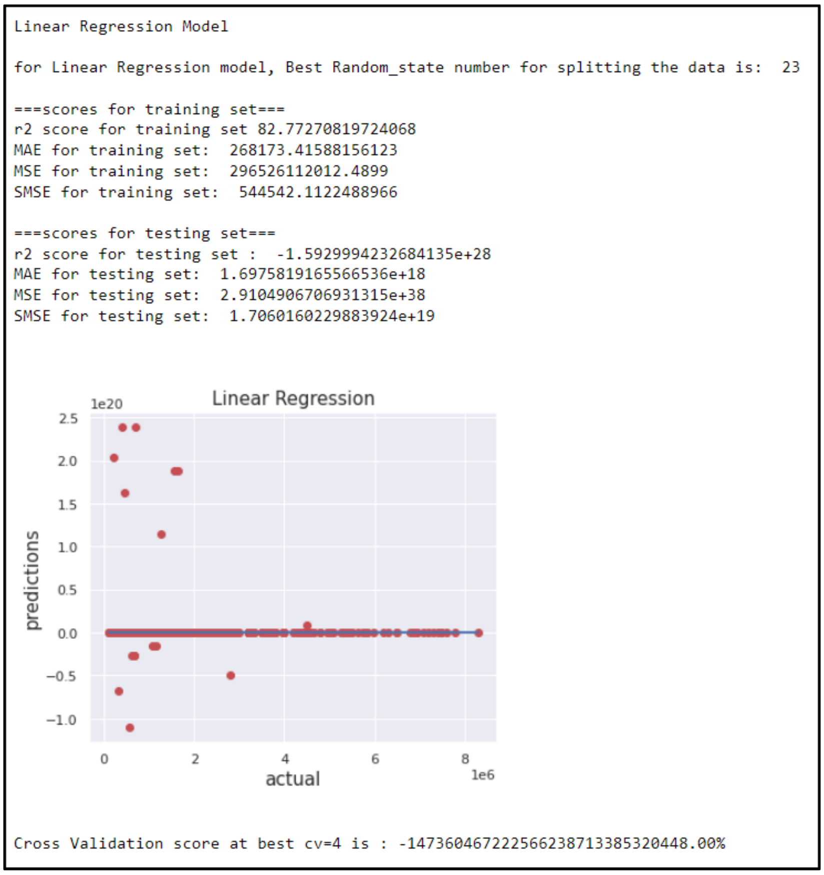


```
lr2 = LinearRegression()
ls2 = Lasso()
rd2 = Ridge()
rfr2 = RandomForestRegressor()
abr2 = AdaBoostRegressor()
gbr2 = GradientBoostingRegressor()
dtr2 = DecisionTreeRegressor()
svr2 = SVR()
knr2 = KNeighborsRegressor()
xgb2 = XGBRegressor()
```

Run and evaluate selected models



```
models = [lr2, ls2, rd2, abr2, gbr2, dtr2,svr2, knr2, xgb2, rfr2]
models_name = ['Linear Regression', 'Lasso', 'Ridge','Ada-Boost Regressor', 'Gradient Boosting Regressor',
               'Decision Tree Regressor','Support Vector Machine', 'KNeighbors Regressor', 'XGB Regressor',
               'Random Forest Regressor']
```



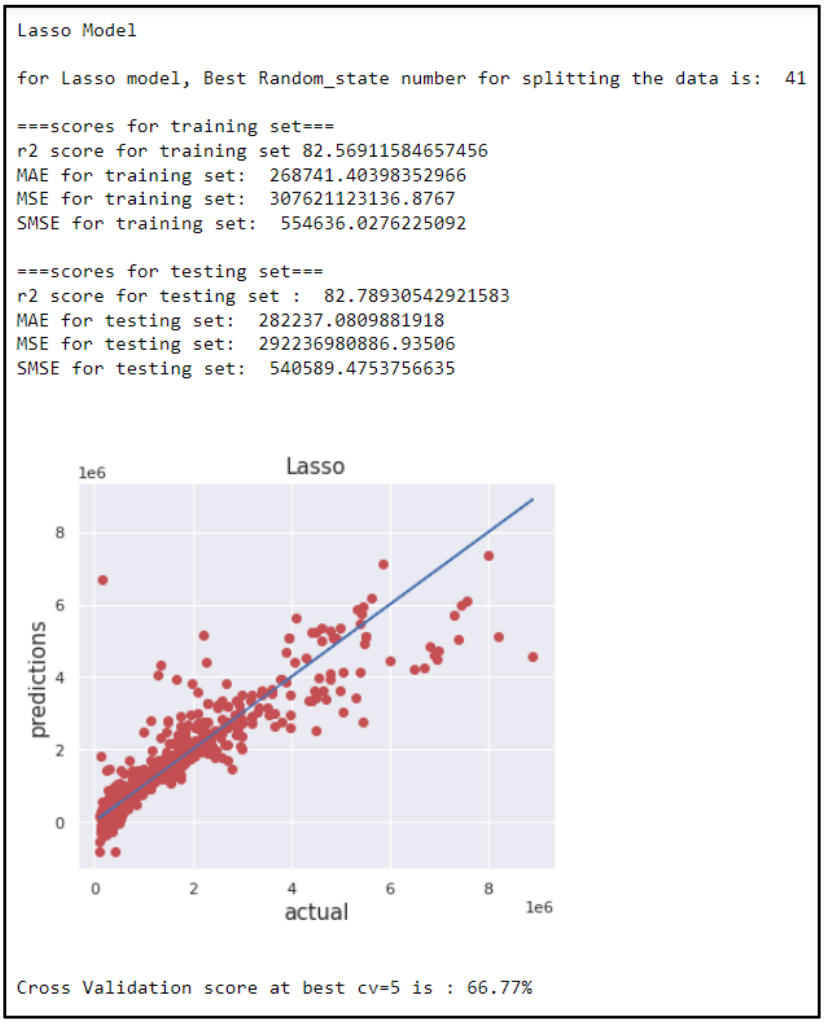
```
Linear Regression Model

for Linear Regression model, Best Random_state number for splitting the data is:  23

===scores for training set===
r2 score for training set 82.77270819724068
MAE for training set:  268173.41588156123
MSE for training set:  296526112012.4899
SMSE for training set:  544542.1122488966

===scores for testing set===
r2 score for testing set :  -1.5929994232684135e+28
MAE for testing set:  1.6975819165566536e+18
MSE for testing set:  2.9104906706931315e+38
SMSE for testing set:  1.7060160229883924e+19
```

```
Cross Validation score at best cv=4 is : -14736046722256623871338532044800%
```



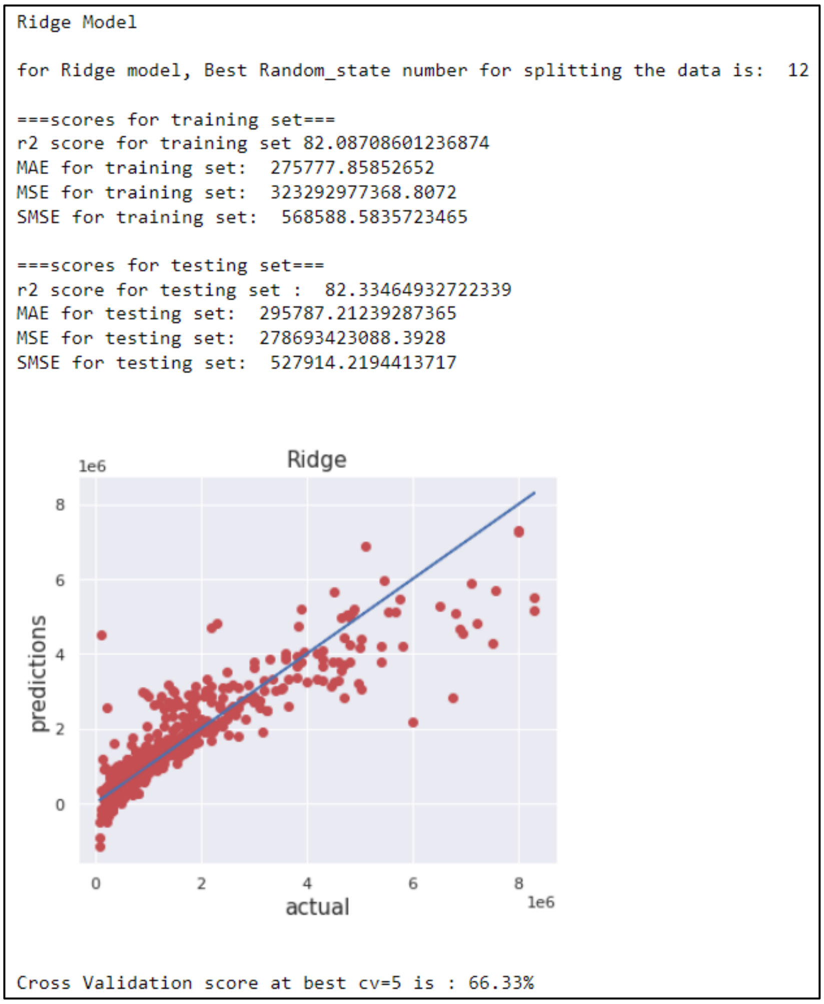
```
Lasso Model

for Lasso model, Best Random_state number for splitting the data is:  41

===scores for training set===
r2 score for training set 82.56911584657456
MAE for training set:  268741.40398352966
MSE for training set:  307621123136.8767
SMSE for training set:  554636.0276225092

===scores for testing set===
r2 score for testing set :  82.78930542921583
MAE for testing set:  282237.0809881918
MSE for testing set:  292236980886.93506
SMSE for testing set:  540589.4753756635
```

```
Cross Validation score at best cv=5 is : 66.77%
```



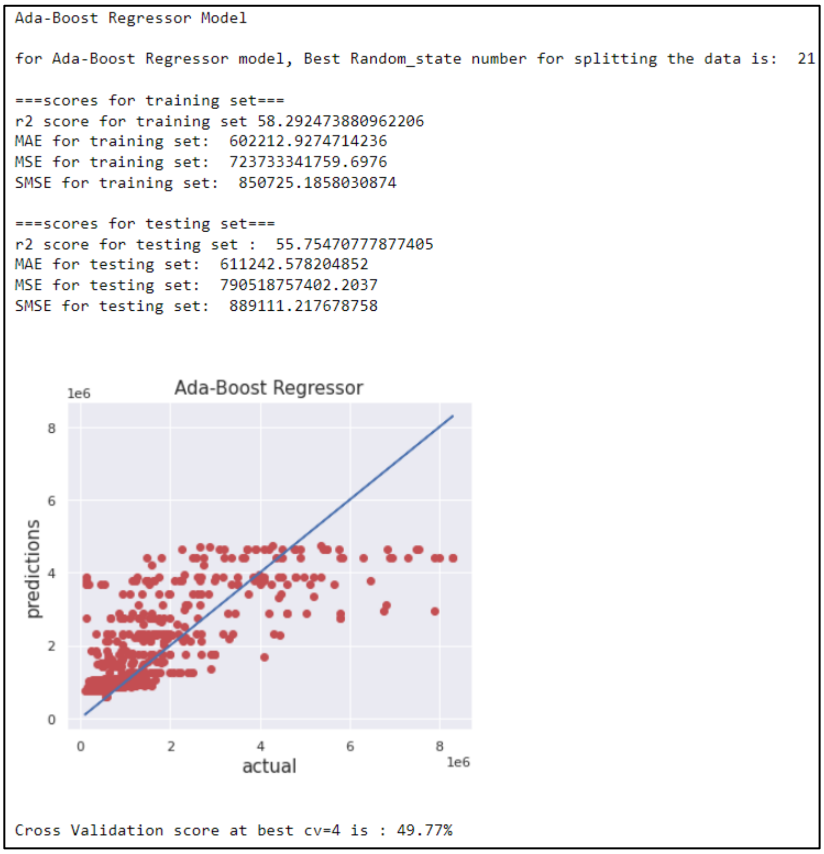
```
Ridge Model

for Ridge model, Best Random_state number for splitting the data is:  12

===scores for training set===
r2 score for training set 82.08708601236874
MAE for training set:  275777.85852652
MSE for training set:  323292977368.8072
SMSE for training set:  568588.5835723465

===scores for testing set===
r2 score for testing set :  82.33464932722339
MAE for testing set:  295787.21239287365
MSE for testing set:  278693423088.3928
SMSE for testing set:  527914.2194413717
```

```
Cross Validation score at best cv=5 is : 66.33%
```



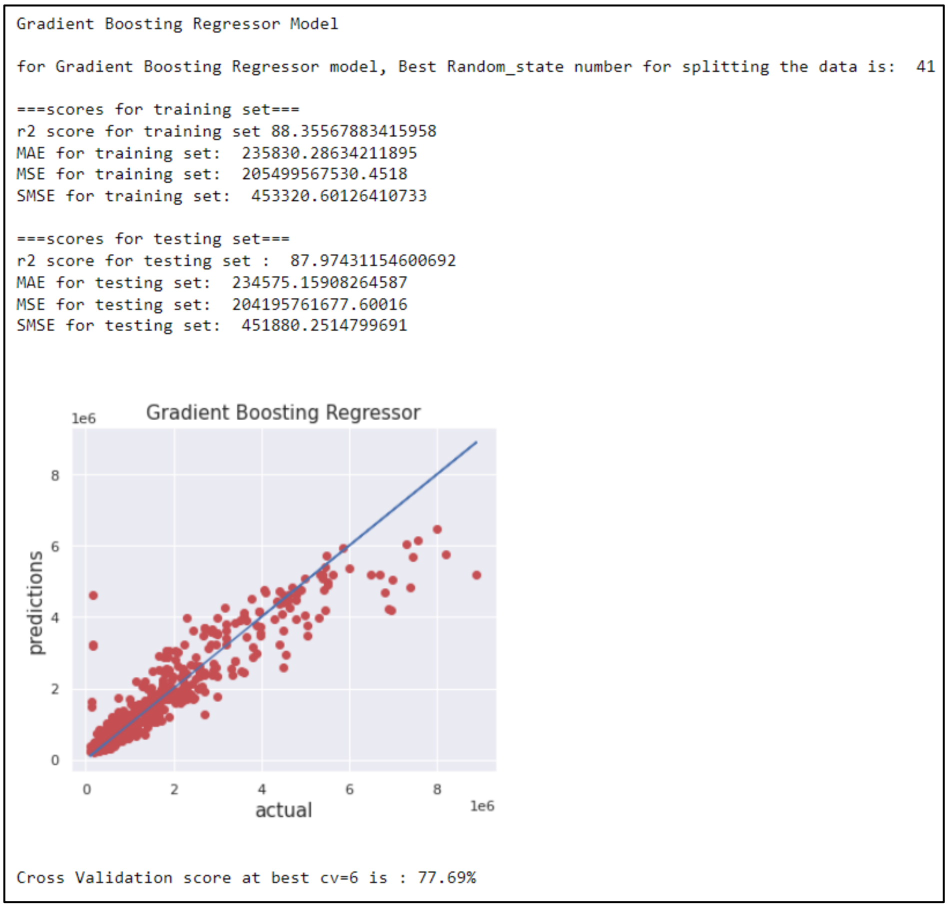
```
Ada-Boost Regressor Model

for Ada-Boost Regressor model, Best Random_state number for splitting the data is:  21

===scores for training set===
r2 score for training set 58.292473880962206
MAE for training set:  602212.9274714236
MSE for training set:  723733341759.6976
SMSE for training set:  850725.1858030874

===scores for testing set===
r2 score for testing set :  55.75470777877405
MAE for testing set:  611242.578204852
MSE for testing set:  790518757402.2037
SMSE for testing set:  889111.217678758
```

```
Cross Validation score at best cv=4 is : 49.77%
```



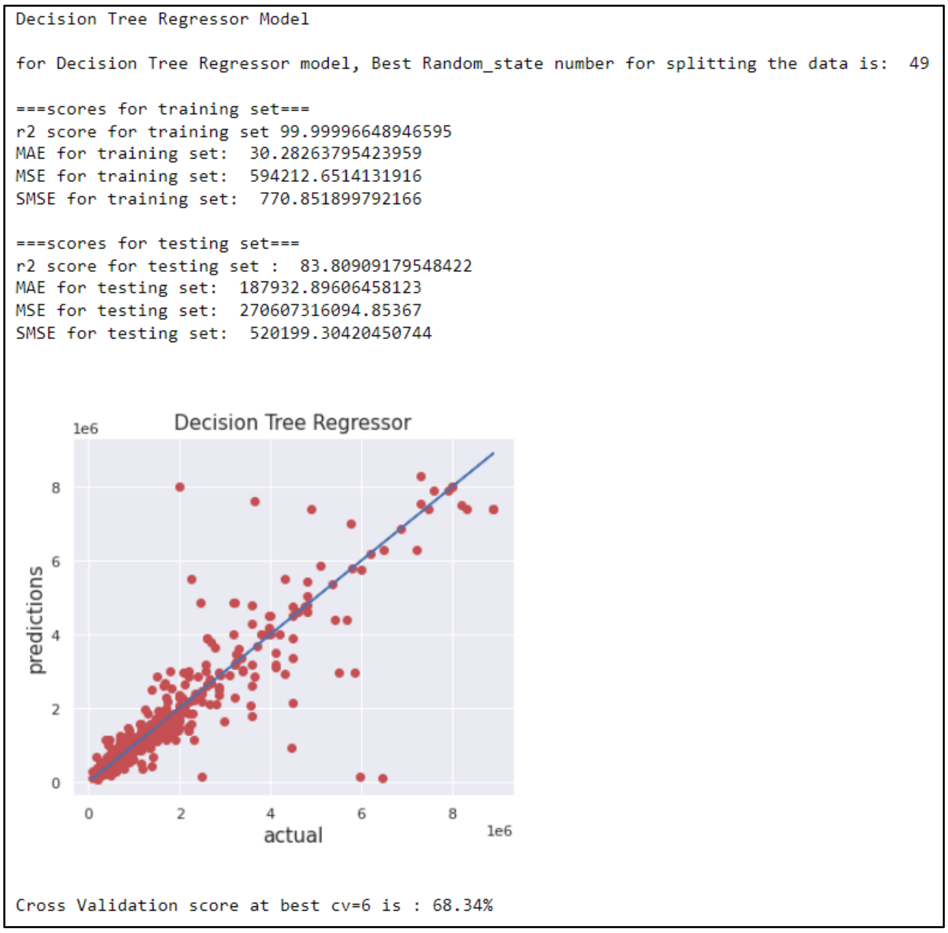
```
Gradient Boosting Regressor Model

for Gradient Boosting Regressor model, Best Random_state number for splitting the data is:  41

===scores for training set===
r2 score for training set 88.35567883415958
MAE for training set:  235830.28634211895
MSE for training set:  205499567530.4518
SMSE for training set:  453320.60126410733

===scores for testing set===
r2 score for testing set :  87.97431154600692
MAE for testing set:  234575.15908264587
MSE for testing set:  204195761677.60016
SMSE for testing set:  451880.2514799691
```

```
Cross Validation score at best cv=6 is : 77.69%
```



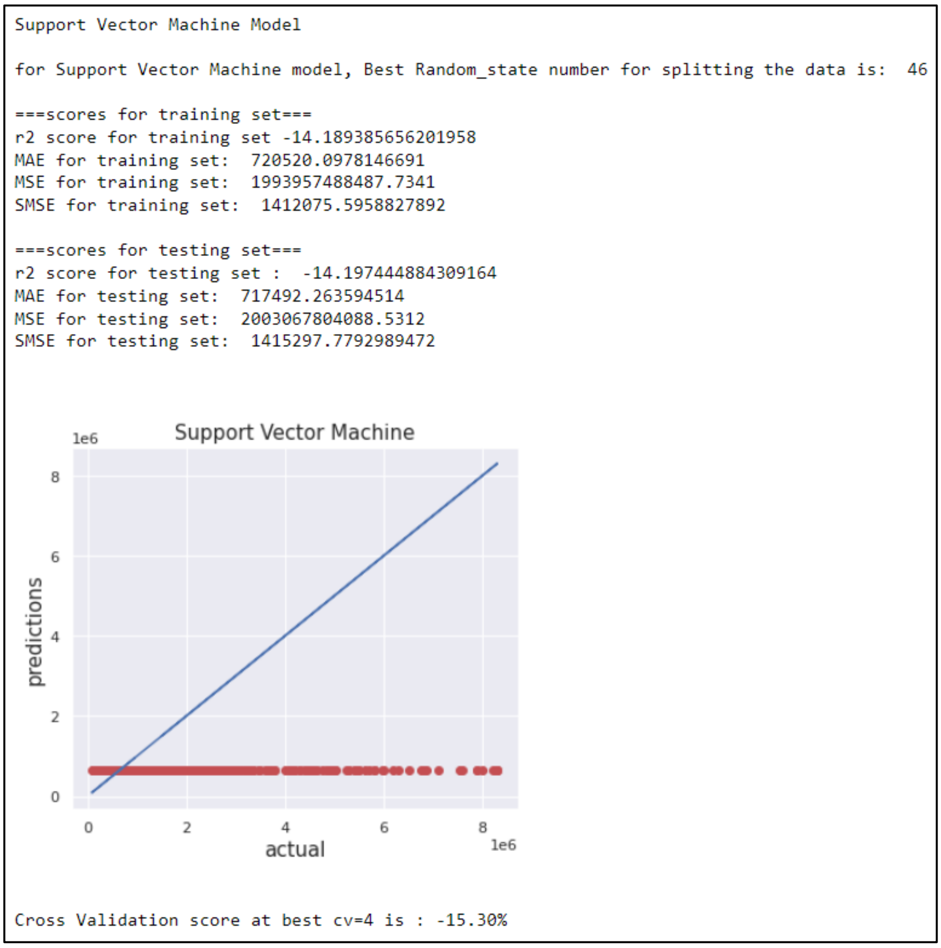
```
Decision Tree Regressor Model

for Decision Tree Regressor model, Best Random_state number for splitting the data is:  49

===scores for training set===
r2 score for training set 99.99996648946595
MAE for training set:  30.28263795423959
MSE for training set:  594212.6514131916
SMSE for training set:  770.851899792166

===scores for testing set===
r2 score for testing set :  83.80909179548422
MAE for testing set:  187932.89606458123
MSE for testing set:  270607316094.85367
SMSE for testing set:  520199.30420450744
```

```
Cross Validation score at best cv=6 is : 68.34%
```



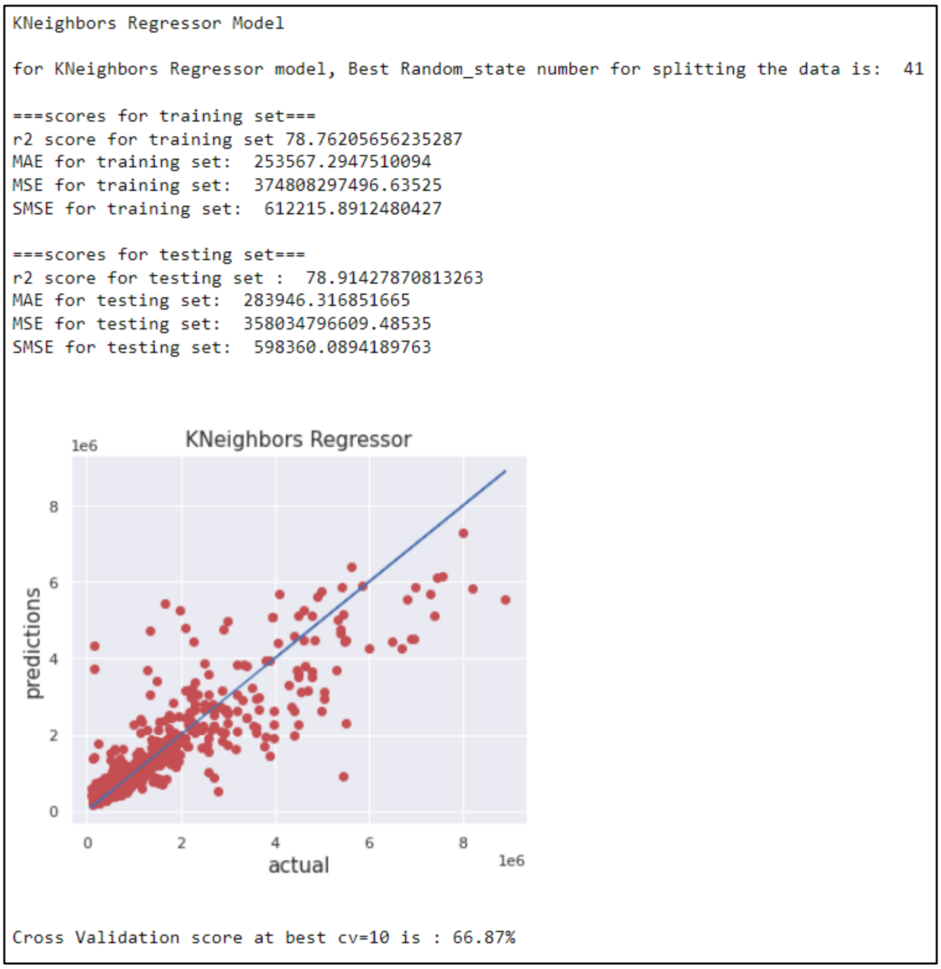
```
Support Vector Machine Model

for Support Vector Machine model, Best Random_state number for splitting the data is:  46

===scores for training set===
r2 score for training set -14.189385656201958
MAE for training set:  720520.0978146691
MSE for training set:  1993957488487.7341
SMSE for training set:  1412075.5958827892

===scores for testing set===
r2 score for testing set :  -14.197444884309164
MAE for testing set:  717492.263594514
MSE for testing set:  2003067804088.5312
SMSE for testing set:  1415297.7792989472
```

```
Cross Validation score at best cv=4 is : -15.30%
```



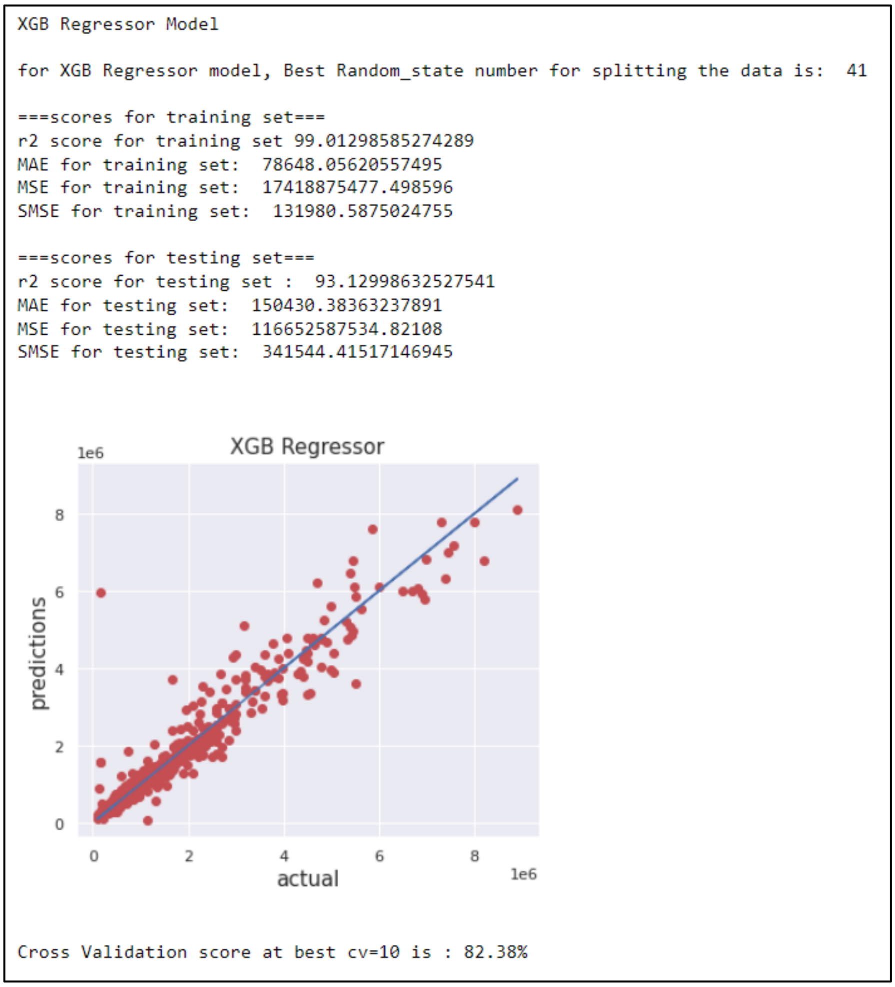
```
KNeighbors Regressor Model

for KNeighbors Regressor model, Best Random_state number for splitting the data is:  41

===scores for training set===
r2 score for training set 78.76205656235287
MAE for training set:  253567.2947510094
MSE for training set:  374808297496.63525
SMSE for training set:  612215.8912480427

===scores for testing set===
r2 score for testing set :  78.91427870813263
MAE for testing set:  283946.316851665
MSE for testing set:  358034796609.48535
SMSE for testing set:  598360.0894189763
```

```
Cross Validation score at best cv=10 is : 66.87%
```



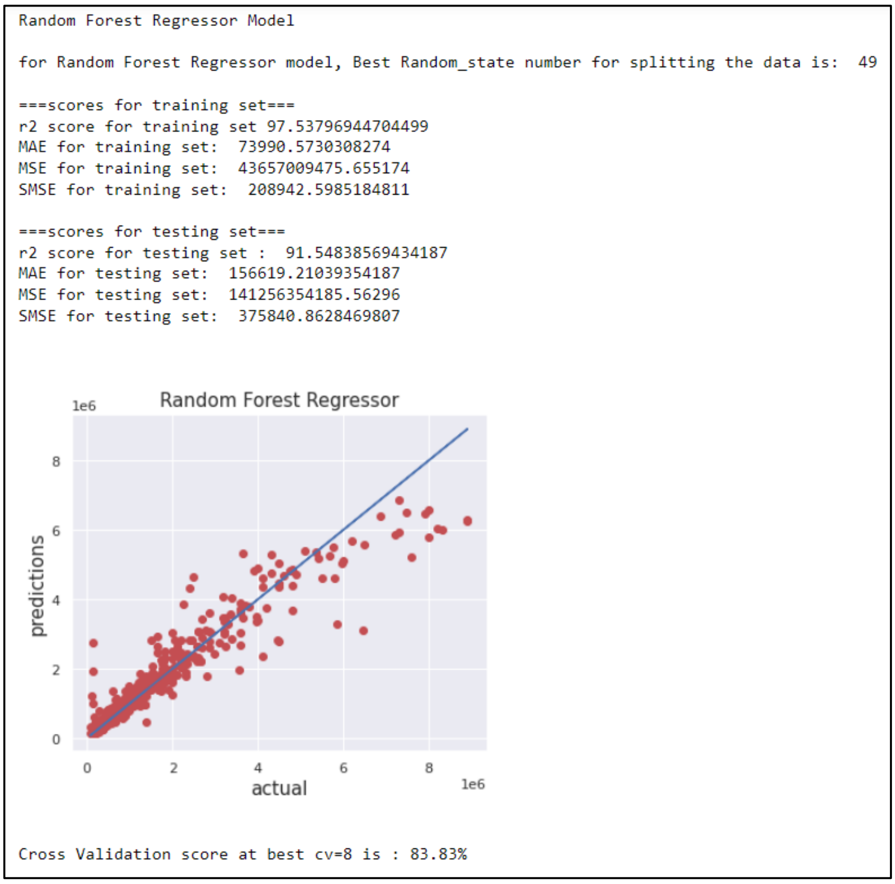
```
XGB Regressor Model

for XGB Regressor model, Best Random_state number for splitting the data is:  41

===scores for training set===
r2 score for training set 99.01298585274289
MAE for training set:  78648.05620557495
MSE for training set:  17418875477.498596
SMSE for training set:  131980.5875024755

===scores for testing set===
r2 score for testing set :  93.12998632527541
MAE for testing set:  150430.38363237891
MSE for testing set:  116652587534.82108
SMSE for testing set:  341544.41517146945
```

```
Cross Validation score at best cv=10 is : 82.38%
```



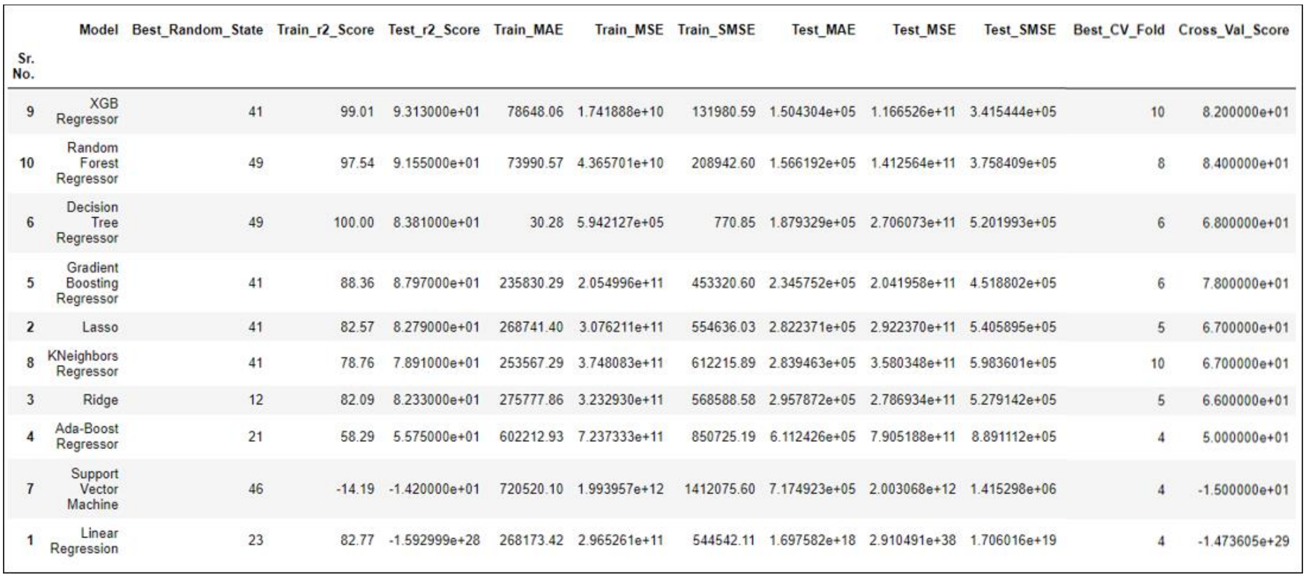
```
Random Forest Regressor Model

for Random Forest Regressor model, Best Random_state number for splitting the data is:  49

===scores for training set===
r2 score for training set 97.53796944704499
MAE for training set:  73990.5730308274
MSE for training set:  43657009475.655174
SMSE for training set:  208942.5985184811

===scores for testing set===
r2 score for testing set :  91.54838569434187
MAE for testing set:  156619.21039354187
MSE for testing set:  141256354185.56296
SMSE for testing set:  375840.8628469807
```

```
Cross Validation score at best cv=8 is : 83.83%
```



Sr. No.	Model	Best_Random_State	Train_r2_Score	Test_r2_Score	Train_MAE	Train_MSE	Train_SMSE	Test_MAE	Test_MSE	Test_SMSE	Best_CV_Fold	Cross_Val_Score
9	XGB Regressor	41	99.01	9.313000e+01	78648.06	1.741888e+10	131980.59	1.504304e+05	1.166526e+11	3.415444e+05	10	8.200000e+01
10	Random Forest Regressor	49	97.54	9.155000e+01	73990.57	4.365701e+10	208942.60	1.566192e+05	1.412564e+11	3.758409e+05	8	8.400000e+01
6	Decision Tree Regressor	49	100.00	8.381000e+01	30.28	5.942127e+05	770.85	1.879329e+05	2.706073e+11	5.201993e+05	6	6.800000e+01
5	Gradient Boosting Regressor	41	88.36	8.797000e+01	235830.29	2.054996e+11	453320.60	2.345752e+05	2.041958e+11	4.518802e+05	6	7.800000e+01
2	Lasso	41	82.57	8.279000e+01	268741.40	3.076211e+11	554636.03	2.822371e+05	2.922370e+11	5.405895e+05	5	6.700000e+01
8	KNeighbors Regressor	41	78.76	7.891000e+01	253567.29	3.748083e+11	612215.89	2.839463e+05	3.580348e+11	5.983601e+05	10	6.700000e+01
3	Ridge	12	82.09	8.233000e+01	275777.86	3.232930e+11	568588.58	2.957872e+05	2.786934e+11	5.279142e+05	5	6.600000e+01
4	Ada-Boost Regressor	21	58.29	5.575000e+01	602212.93	7.237333e+11	850725.19	6.112426e+05	7.905188e+11	8.891112e+05	4	5.000000e+01
7	Support Vector Machine	46	-14.19	-1.420000e+01	720520.10	1.993957e+12	1412075.60	7.174923e+05	2.003068e+12	1.415298e+06	4	-1.500000e+01
1	Linear Regression	23	82.77	-1.592999e+28	268173.42	2.965261e+11	544542.11	1.697582e+18	2.910491e+38	1.706016e+19	4	-1.473605e+29

We selected ***Random Forest Regressor*** for the following reasons:

- minimum MAE value on test set & highest cross val score.
- minimum difference between Cross val score & test score.
- train score & test score are almost similar.

Key Metrics for success in solving problem under consideration

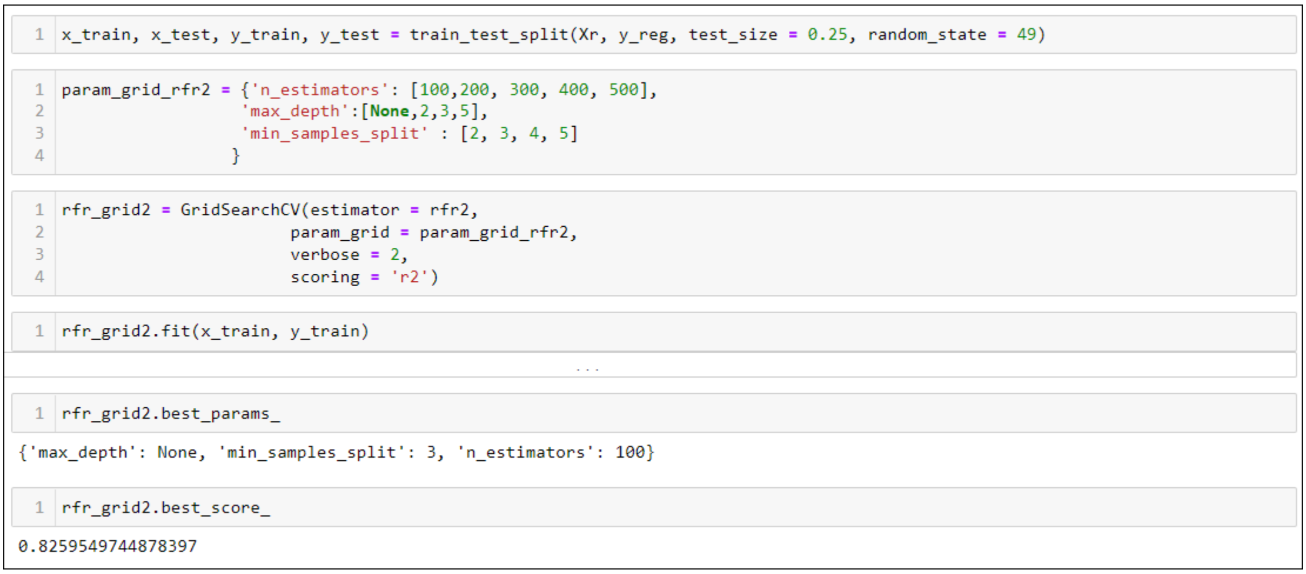
Following metrics used for evaluation:

1. Mean absolute error which gives magnitude of difference between the prediction of an observation and the true value of that observation.

2. Root mean square error is one of the most commonly used measures for evaluating the quality of predictions.

3. R2 score which tells us how accurate our model predict result, is going to important evaluation criteria along with Cross validation score.

Hyperparameter Tuning:



```
x_train, x_test, y_train, y_test = train_test_split(Xr, y_reg, test_size = 0.25, random_state = 49)
```

```
param_grid_rfr2 = {'n_estimators': [100,200, 300, 400, 500],
                  'max_depth':[None,2,3,5],
                  'min_samples_split' : [2, 3, 4, 5]
                 }
```

```
rfr_grid2 = GridSearchCV(estimator = rfr2,
                         param_grid = param_grid_rfr2,
                         verbose = 2,
                         scoring = 'r2')
```

```
rfr_grid2.fit(x_train, y_train)
```

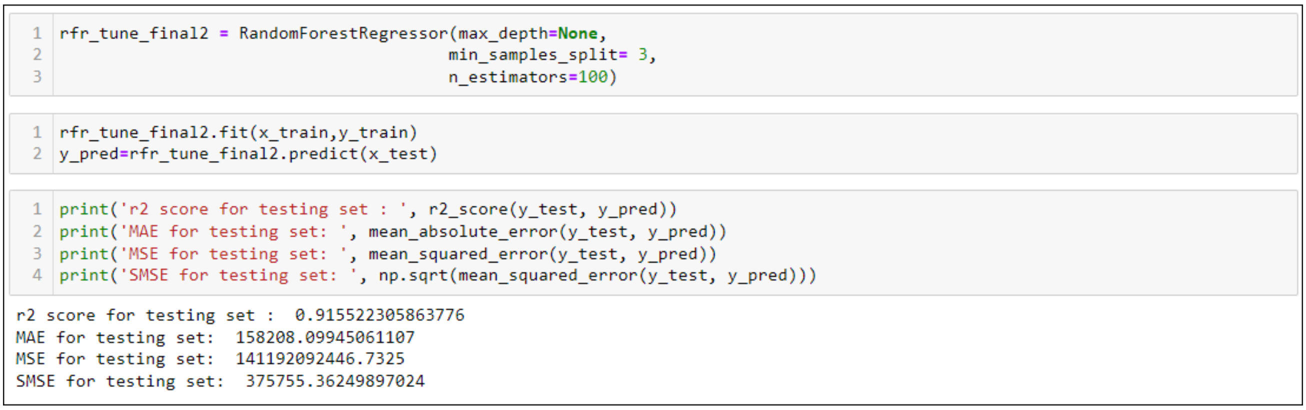
```
...
```

```
rfr_grid2.best_params_
```

```
{'max_depth': None, 'min_samples_split': 3, 'n_estimators': 100}
```

```
rfr_grid2.best_score_
```

```
0.8259549744878397
```

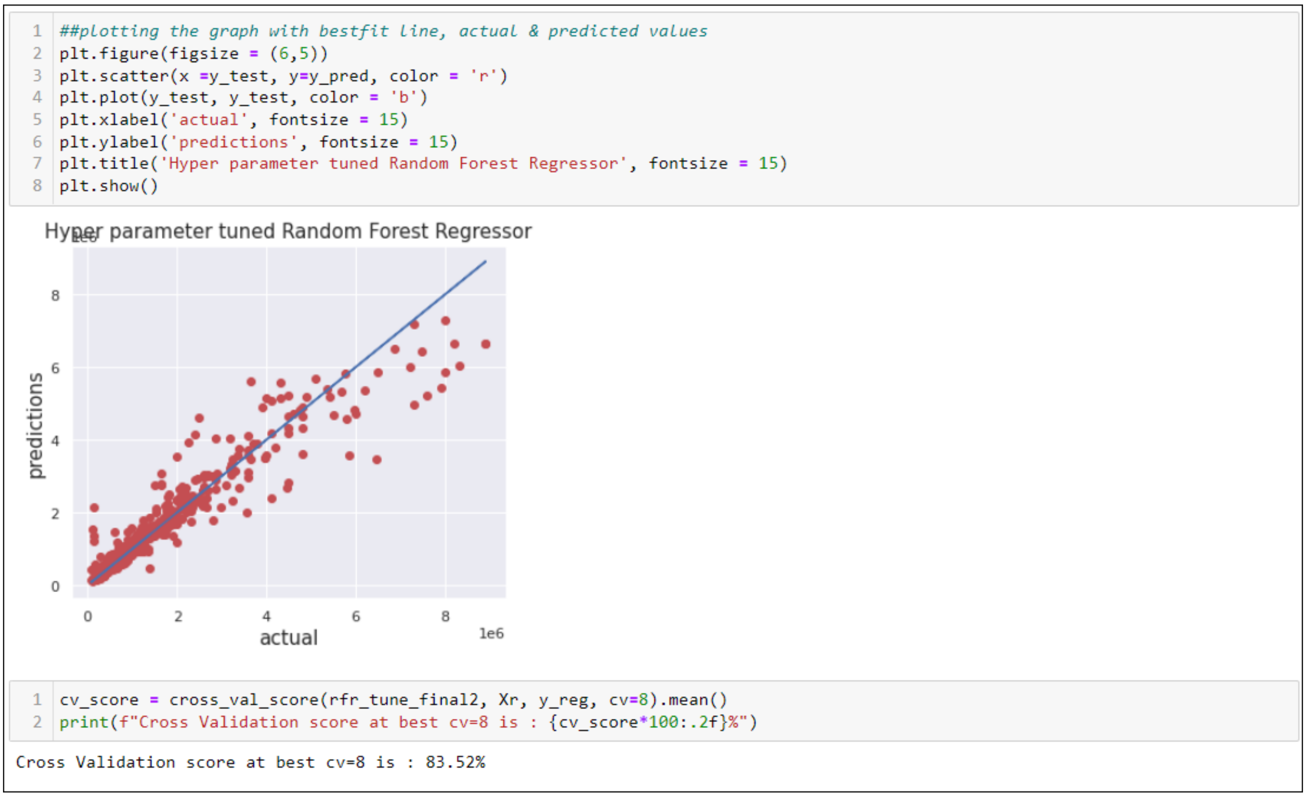


```
rfr_tune_final2 = RandomForestRegressor(max_depth=None,
                                        min_samples_split= 3,
                                        n_estimators=100)
```

```
rfr_tune_final2.fit(x_train,y_train)
y_pred=rfr_tune_final2.predict(x_test)
```

```
print('r2 score for testing set : ', r2_score(y_test, y_pred))
print('MAE for testing set:  ', mean_absolute_error(y_test, y_pred))
print('MSE for testing set:  ', mean_squared_error(y_test, y_pred))
print('SMSE for testing set:  ', np.sqrt(mean_squared_error(y_test, y_pred)))
```

```
r2 score for testing set :  0.915522305863776
MAE for testing set:  158208.09945061107
MSE for testing set:  141192092446.7325
SMSE for testing set:  375755.36249897024
```

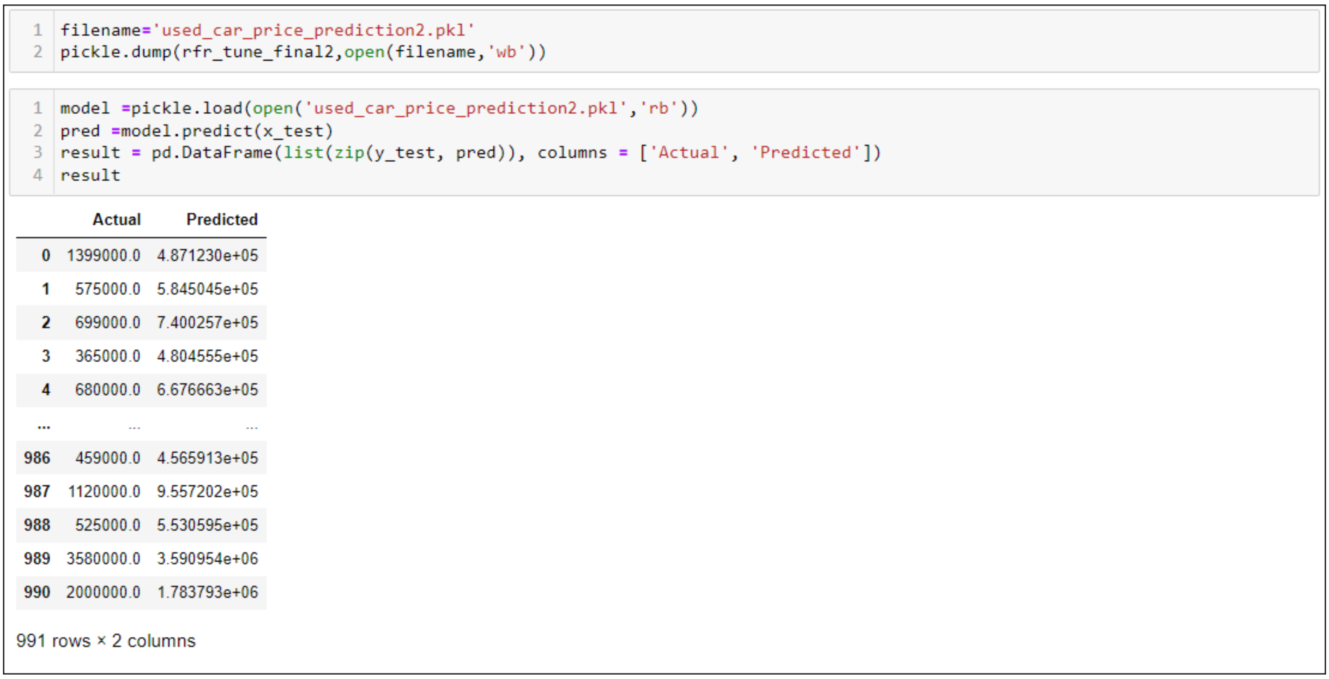


```
##plotting the graph with bestfit line, actual & predicted values
plt.figure(figsize = (6,5))
plt.scatter(x =y_test, y=y_pred, color = 'r')
plt.plot(y_test, y_test, color = 'b')
plt.xlabel('actual', fontsize = 15)
plt.ylabel('predictions', fontsize = 15)
plt.title('Hyper parameter tuned Random Forest Regressor', fontsize = 15)
plt.show()
```

```
cv_score = cross_val_score(rfr_tune_final2, Xr, y_reg, cv=8).mean()
print(f"Cross Validation score at best cv=8 is : {cv_score*100:.2f}%")
```

```
Cross Validation score at best cv=8 is : 83.52%
```

Saving & predictions of the model on Test data provided



```
filename='used_car_price_prediction2.pkl'
pickle.dump(rfr_tune_final2,open(filename,'wb'))
```

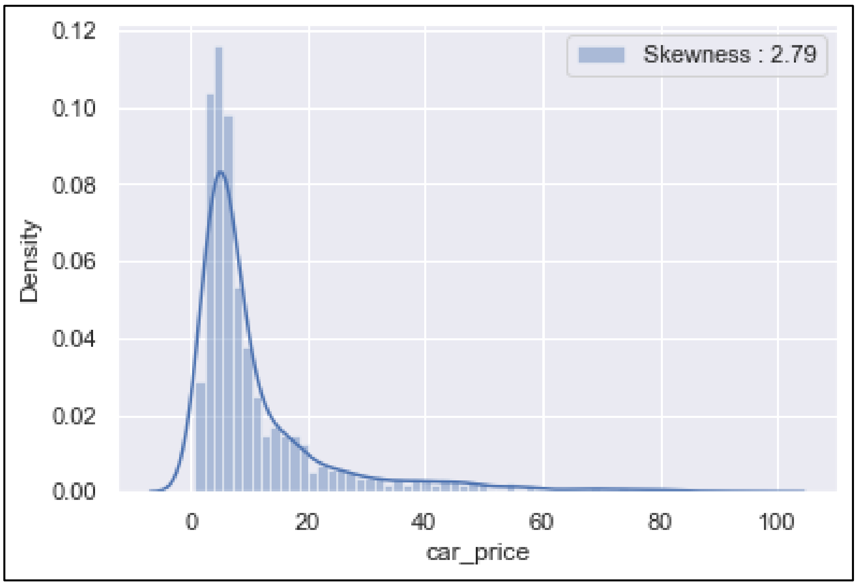
```
model =pickle.load(open('used_car_price_prediction2.pkl','rb'))
pred =model.predict(x_test)
result = pd.DataFrame(list(zip(y_test, pred)), columns = ['Actual', 'Predicted'])
result
```

	Actual	Predicted
0	1399000.0	4.871230e+05
1	575000.0	5.845045e+05
2	699000.0	7.400257e+05
3	365000.0	4.804555e+05
4	680000.0	6.676663e+05
...	...	...
986	459000.0	4.565913e+05
987	1120000.0	9.557202e+05
988	525000.0	5.530595e+05
989	3580000.0	3.590954e+06
990	2000000.0	1.783793e+06

991 rows × 2 columns

VISUALIZATIONS & EDA

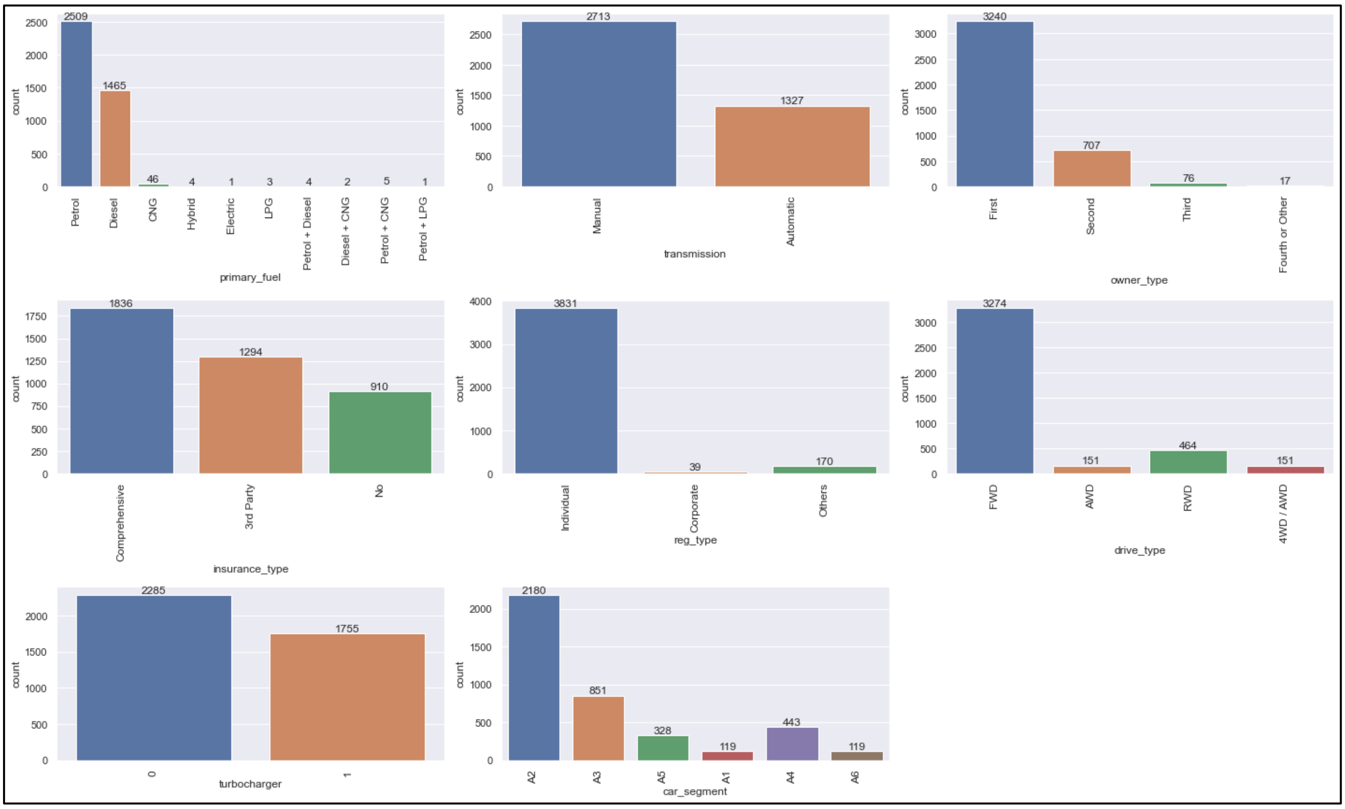
Target Variable:

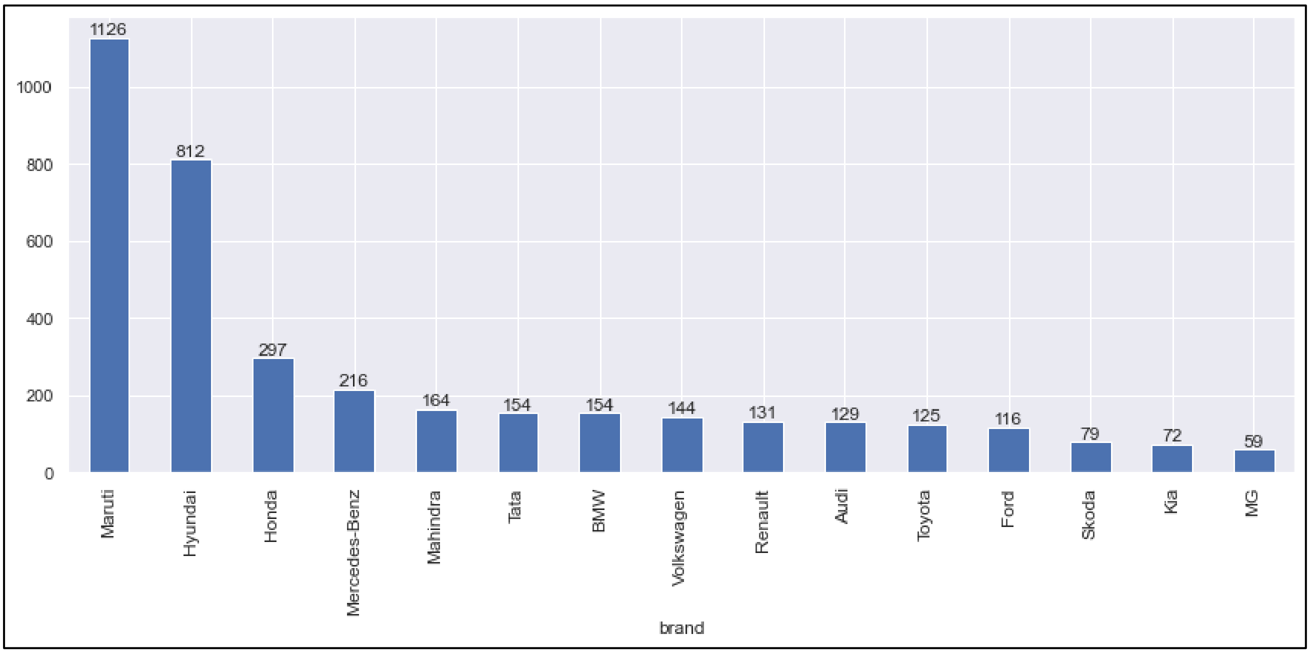


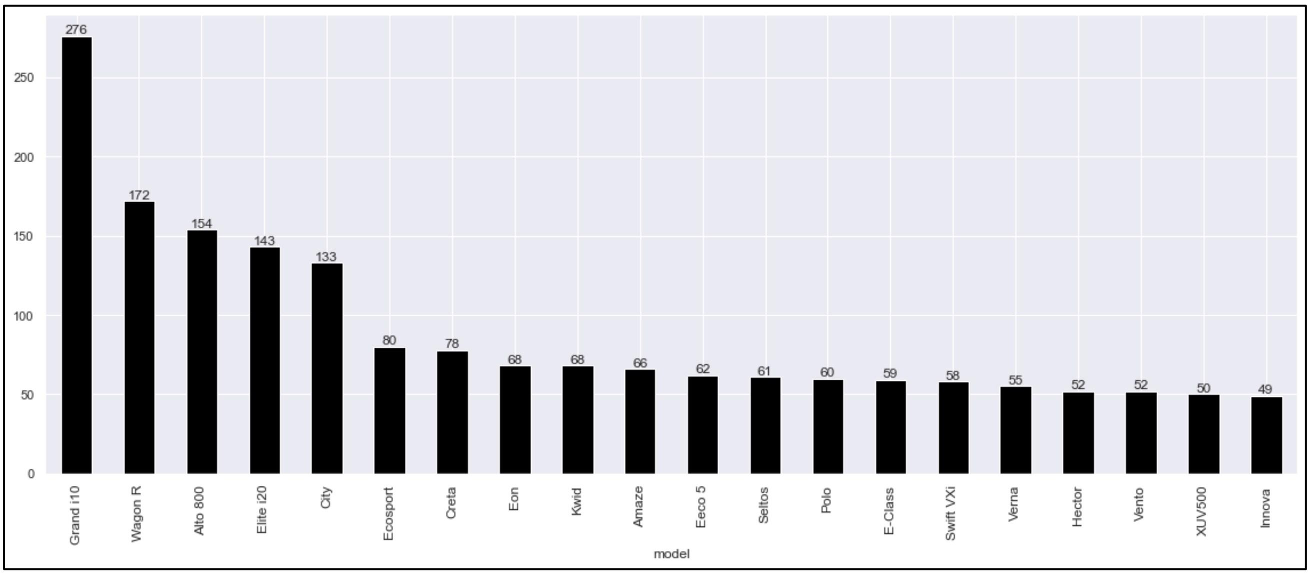
Observation:

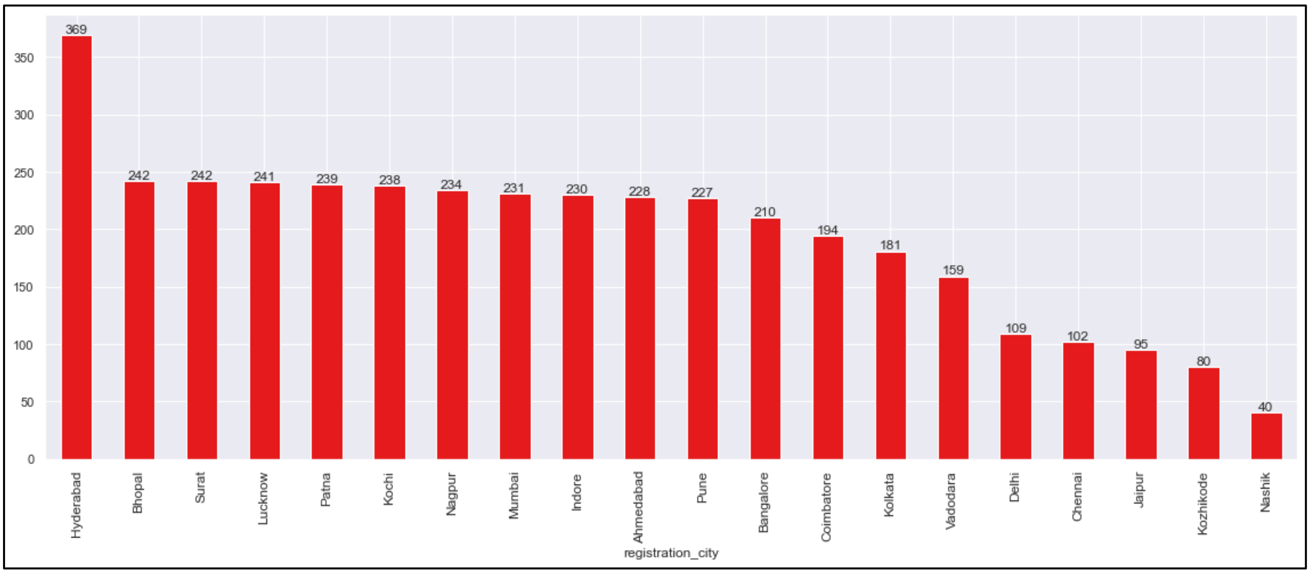
We see that input data was highly imbalanced and more than 89% of the observations were non-defaulters. The data was balanced before feeding it to ML models using SMOTE.

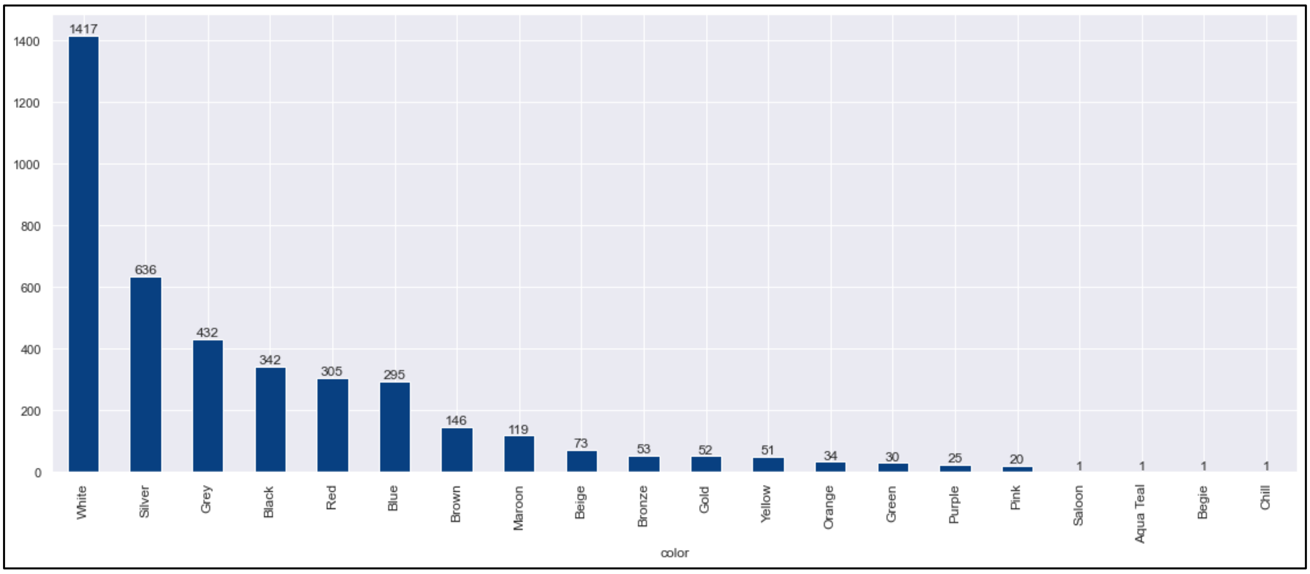
Independent Variables:

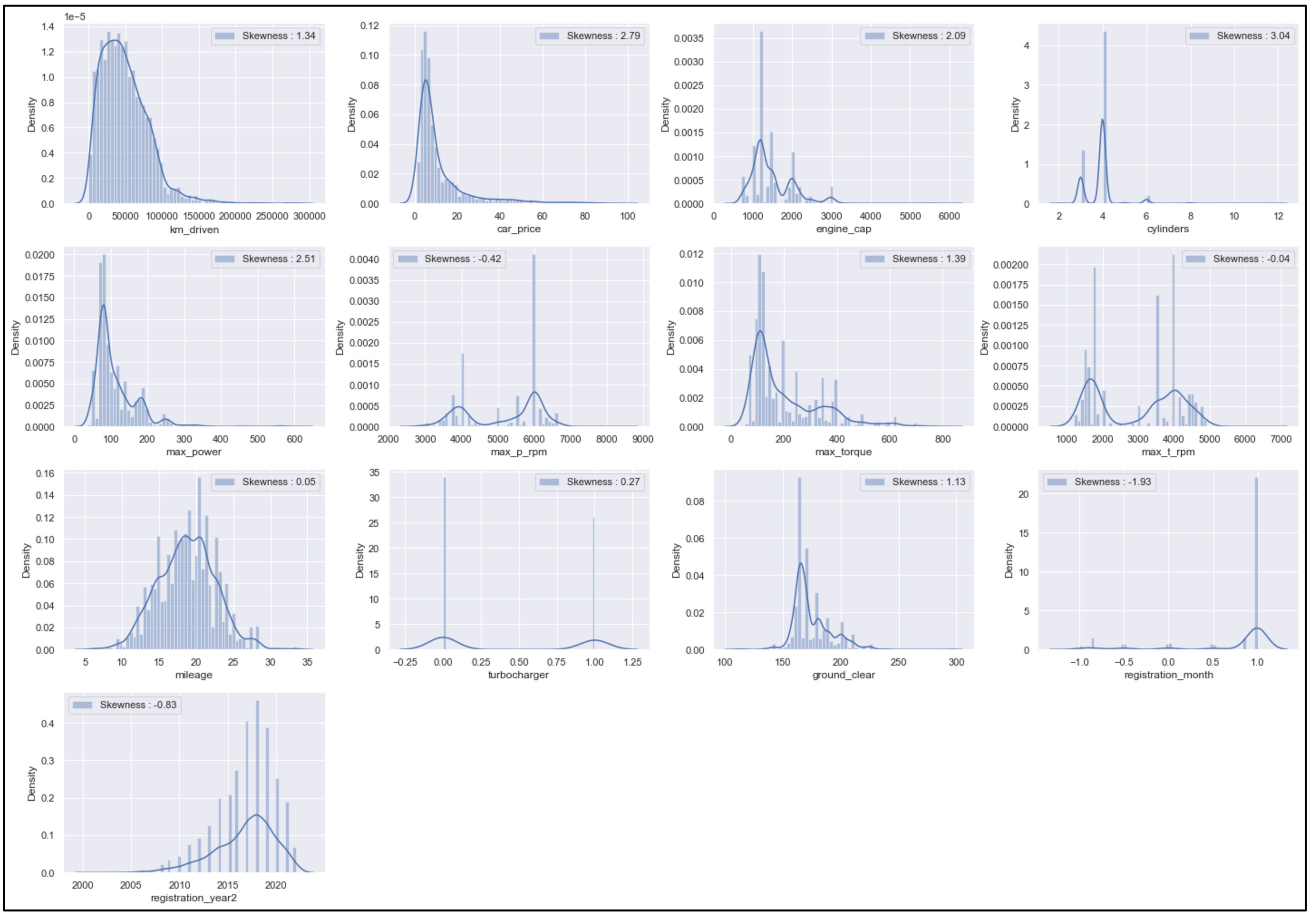



1126
812
297
216
164
154
154
144
131
129
125
116
79
72
59
1000
800
600
400
200
0
Maruti
Hyundai
Honda
Mercedes-Benz
Mahindra
Tata
BMW
Volkswagen
Renault
Audi
Toyota
Ford
Skoda
Kia
MG
brand

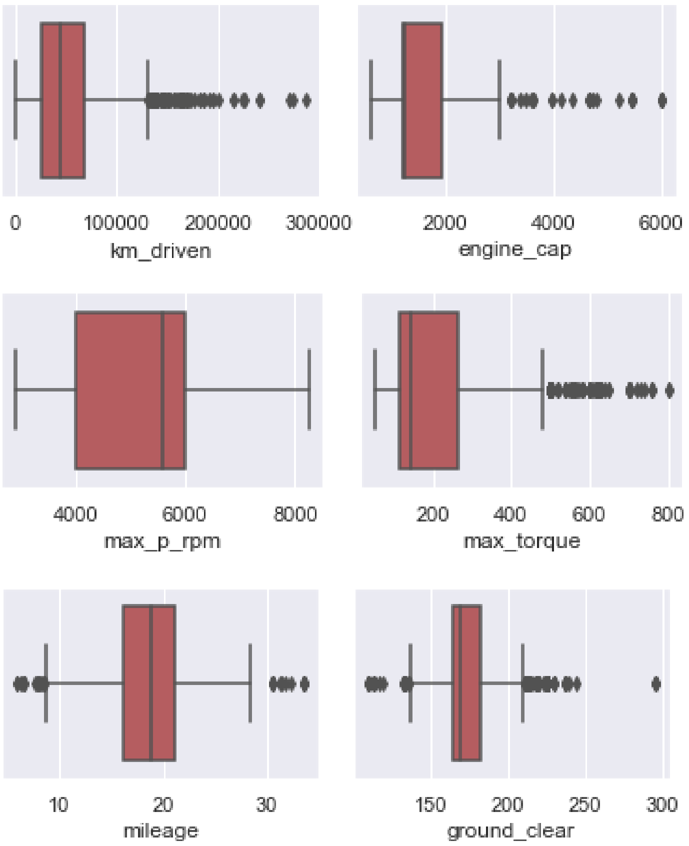

276
172
154
143
133
80
78
68
68
66
62
61
60
59
58
55
52
52
50
49
250
200
150
100
50
0
Grand i10
Wagon R
Alto 800
Elite i20
City
Ecosport
Creta
Eon
Kwid
Amaze
Eeco 5
Seltos
Polo
E-Class
Swift VXi
Verna
Hector
Vento
XUV500
Innova
model


369
242
242
241
239
238
234
231
230
228
227
210
194
181
159
109
102
95
80
40
350
300
250
200
150
100
50
0
Hyderabad
Bhopal
Surat
Lucknow
Patna
Kochi
Nagpur
Mumbai
Indore
Ahmedabad
Pune
Bangalore
Coimbatore
Kolkata
Vadodara
Delhi
Chennai
Jaipur
Kozhikode
Nashik
registration_city

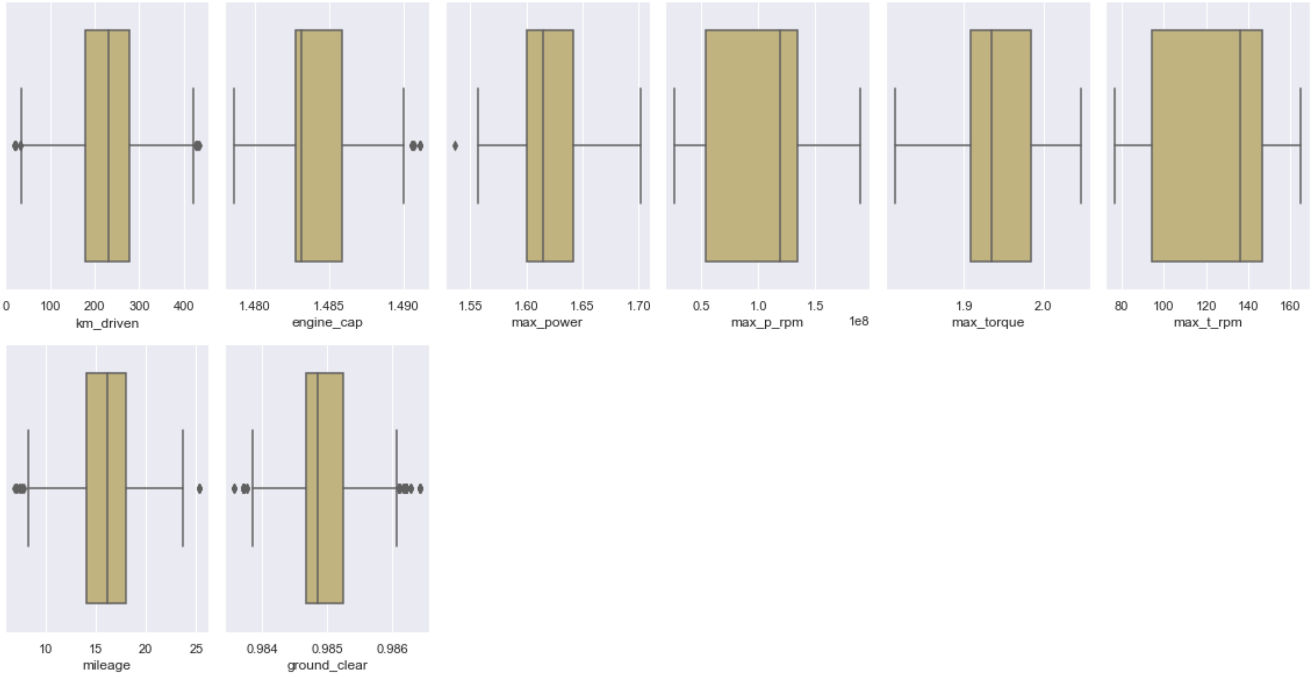

1400
1200
1000
800
600
400
200
0
1417
636
432
342
305
295
146
119
73
53
52
51
34
30
25
20
1
1
1
1
White
Silver
Grey
Black
Red
Blue
Brown
Maroon
Beige
Bronze
Gold
Yellow
Orange
Green
Purple
Pink
Saloon
Aqua Teal
Begie
Chill
color


1e−5
Skewness : 1.34
Density
km_driven
0 50000 100000 150000 200000 250000 300000
Skewness : 2.79
Density
car_price
0 20 40 60 80 100
Skewness : 2.09
Density
engine_cap
0 1000 2000 3000 4000 5000 6000
Skewness : 3.04
Density
cylinders
2 4 6 8 10 12
Skewness : 2.51
Density
max_power
0 100 200 300 400 500 600
Skewness : -0.42
Density
max_p_rpm
2000 3000 4000 5000 6000 7000 8000 9000
Skewness : 1.39
Density
max_torque
0 200 400 600 800
Skewness : -0.04
Density
max_t_rpm
1000 2000 3000 4000 5000 6000 7000
Skewness : 0.05
Density
mileage
5 10 15 20 25 30 35
Skewness : 0.27
Density
turbocharger
−0.25 0.00 0.25 0.50 0.75 1.00 1.25
Skewness : 1.13
Density
ground_clear
100 150 200 250 300
Skewness : -1.93
Density
registration_month
−1.0 −0.5 0.0 0.5 1.0
Skewness : -0.83
Density
registration_year2
2000 2005 2010 2015 2020

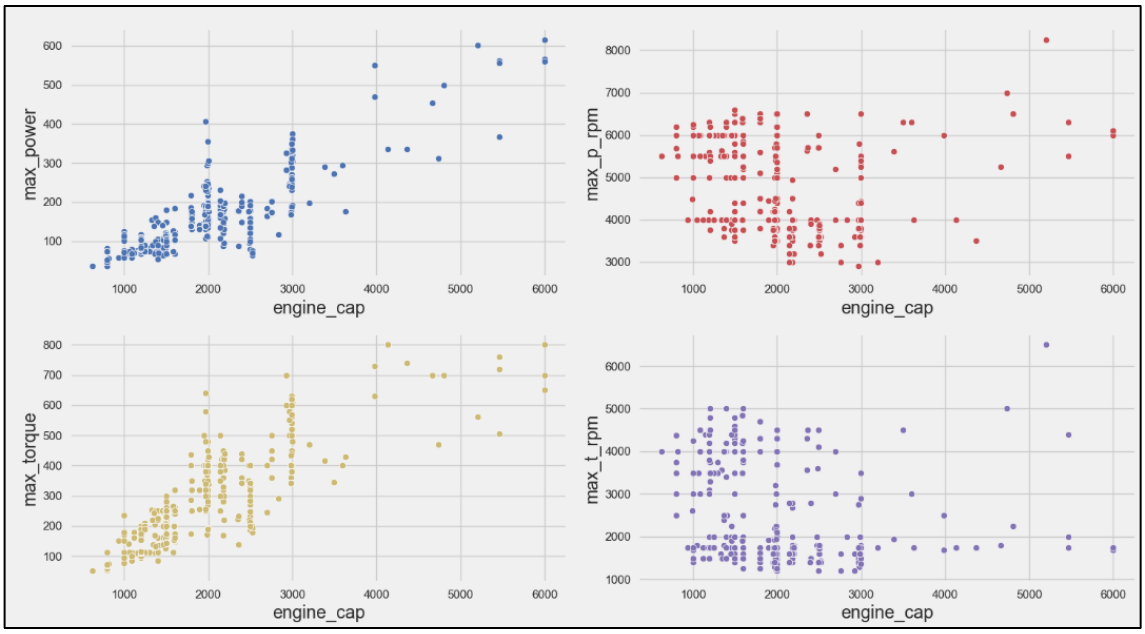
Outliers in Given data:

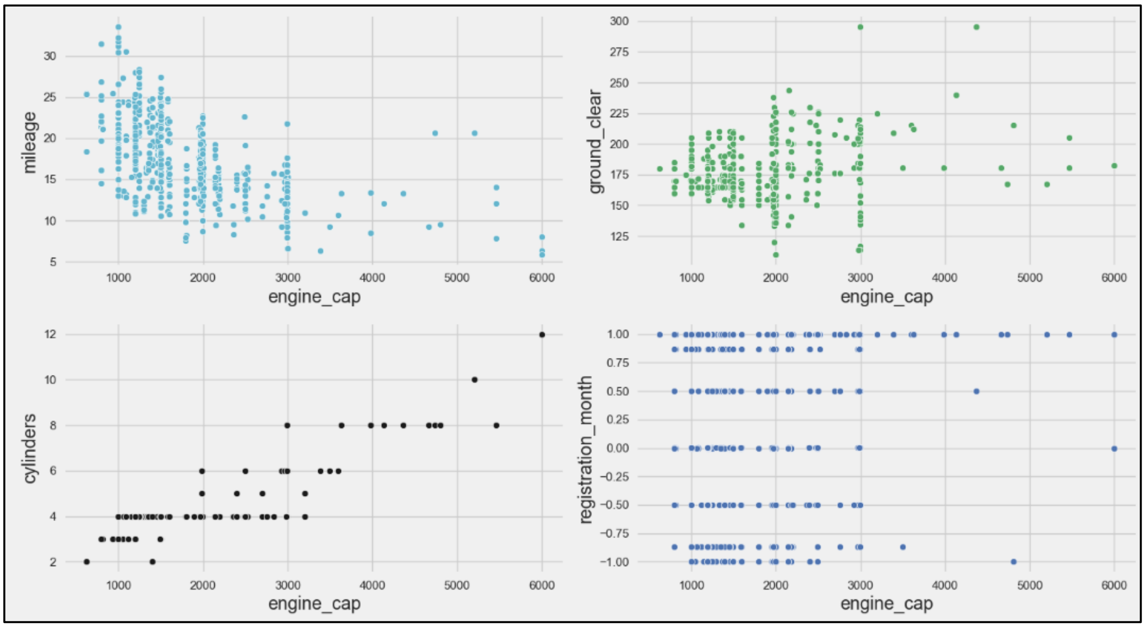


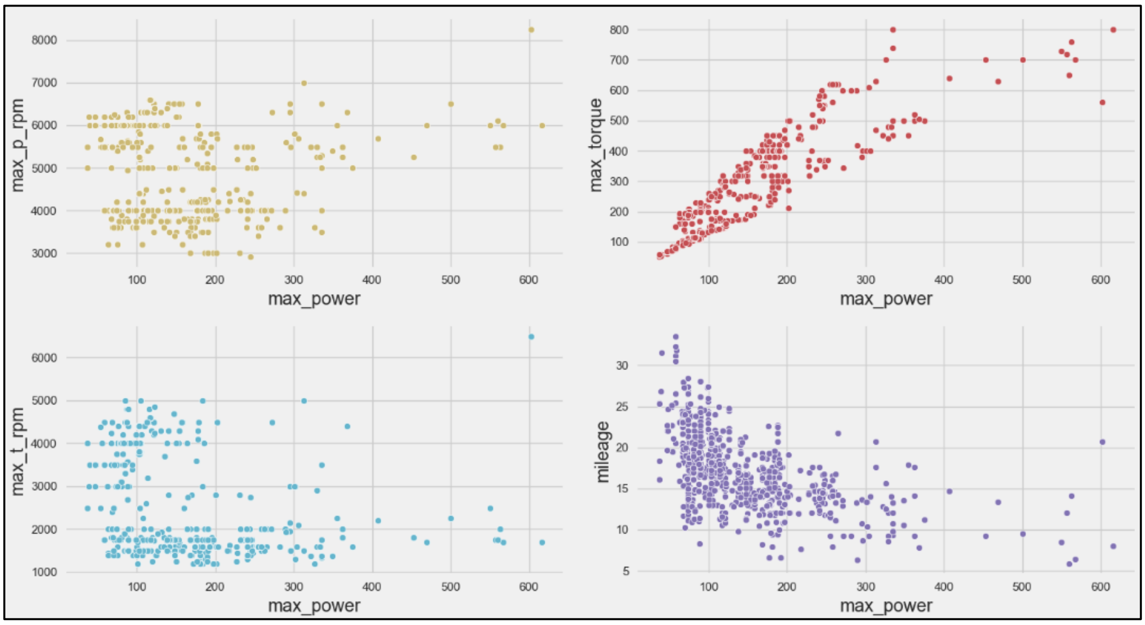
After removing outliers;

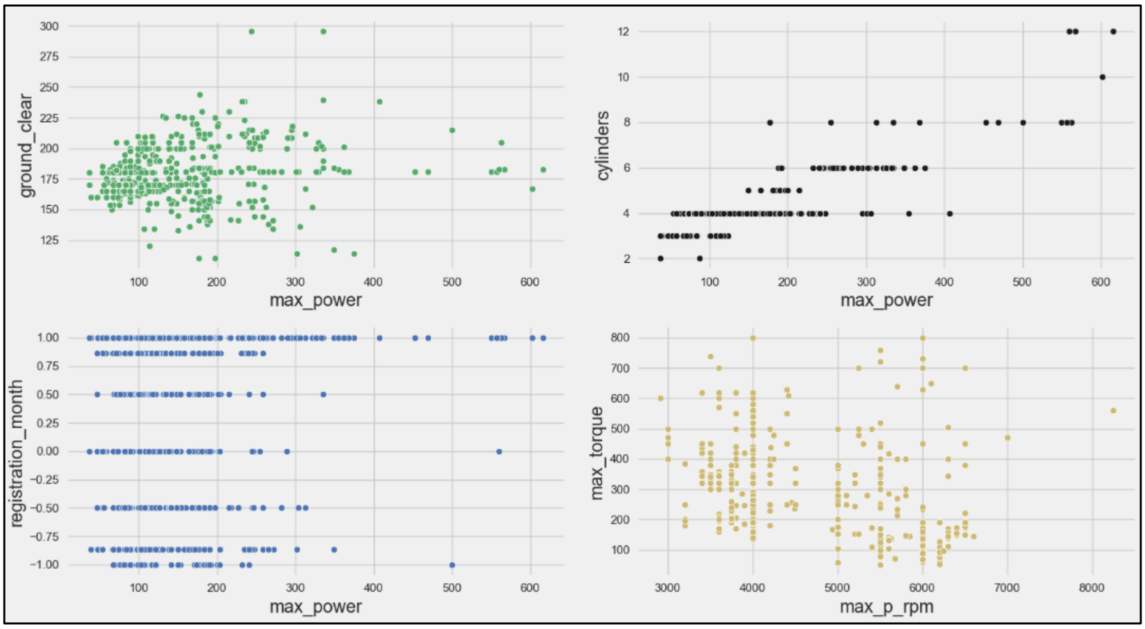


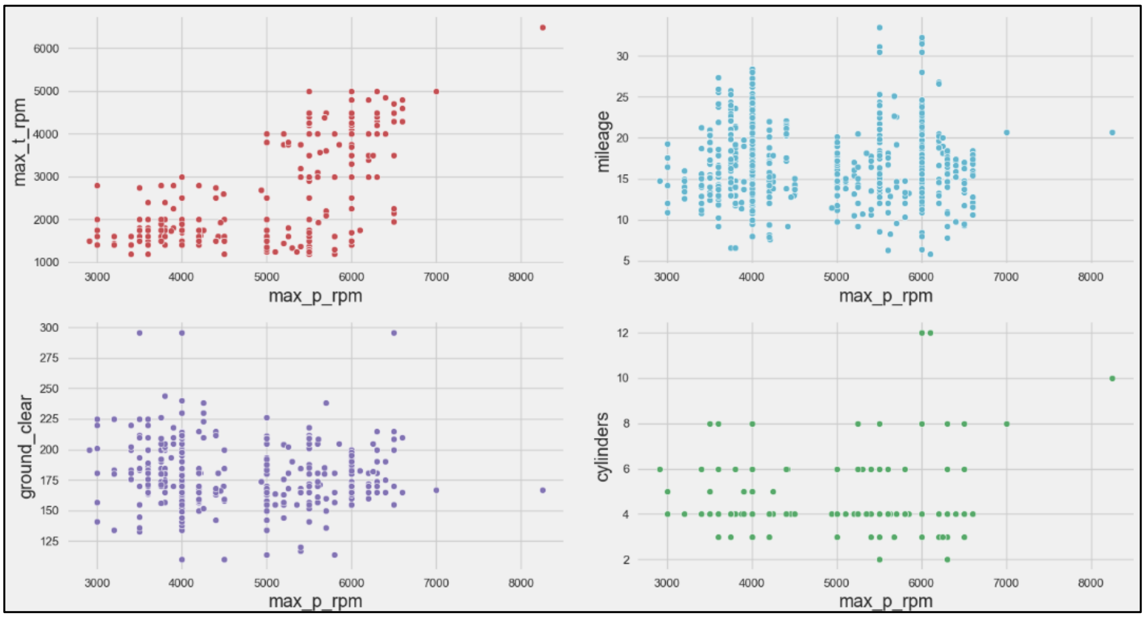
Bivariate Analysis:

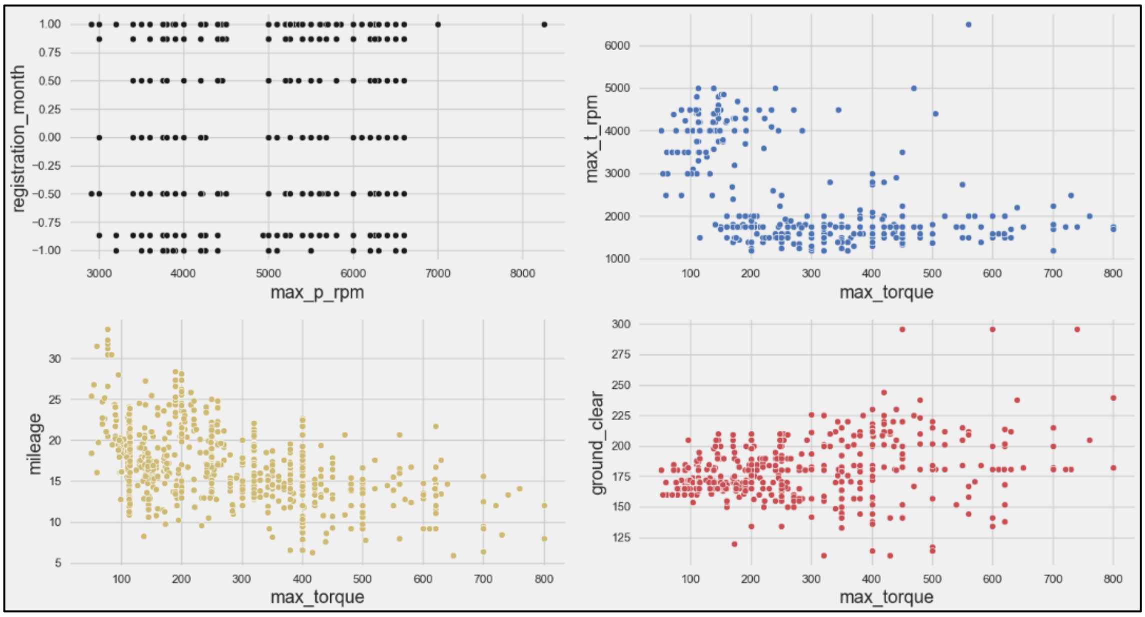


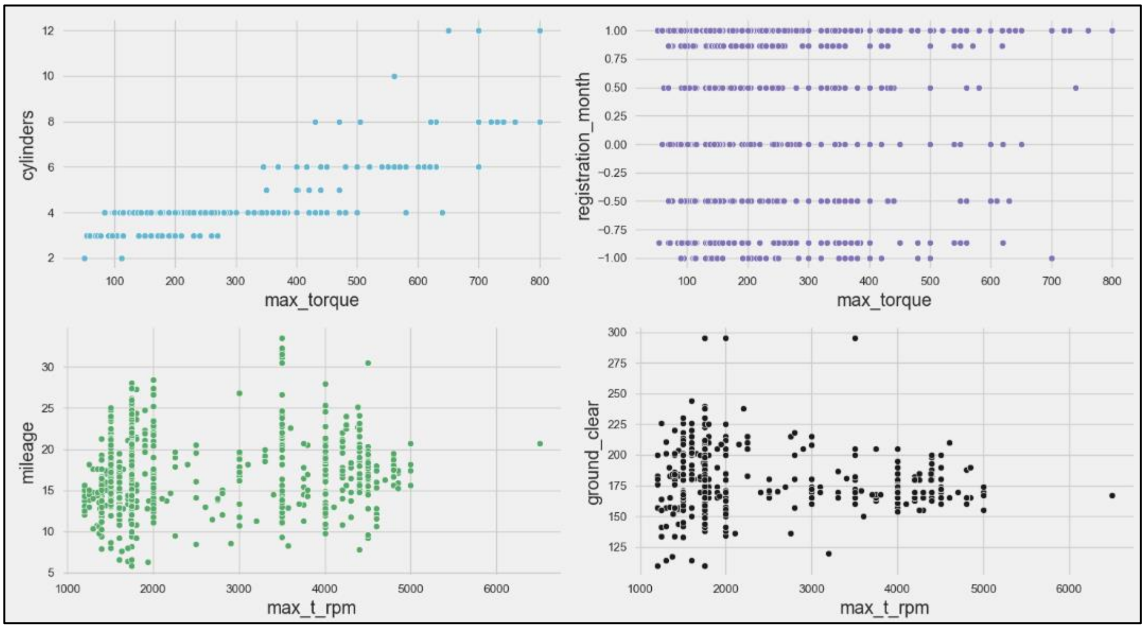


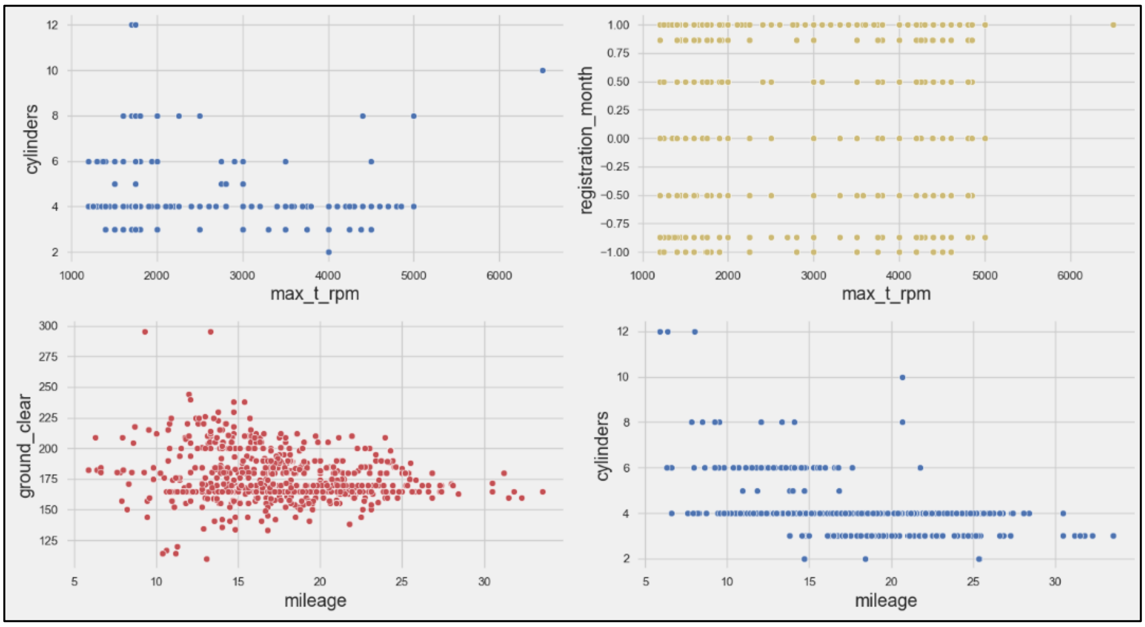


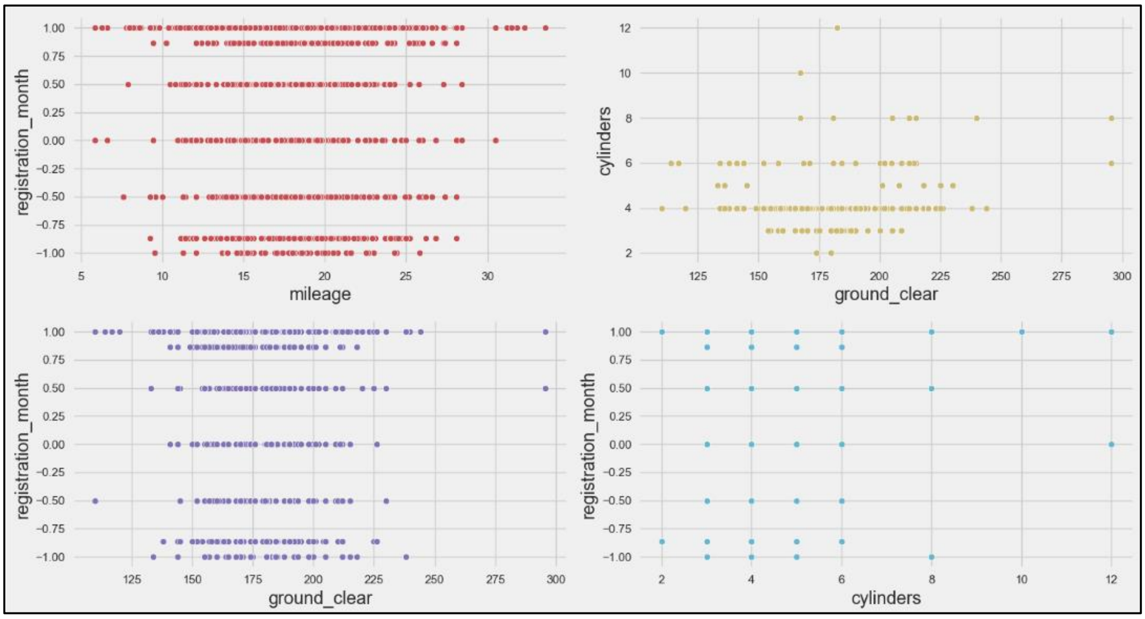

ground_clear
300
275
250
225
200
175
150
125
100
200
300
400
500
600
max_power
cylinders
12
10
8
6
4
2
100
200
300
400
500
600
max_power
registration_month
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
100
200
300
400
500
600
max_power
max_torque
800
700
600
500
400
300
200
100
3000
4000
5000
6000
7000
8000
max_p_rpm

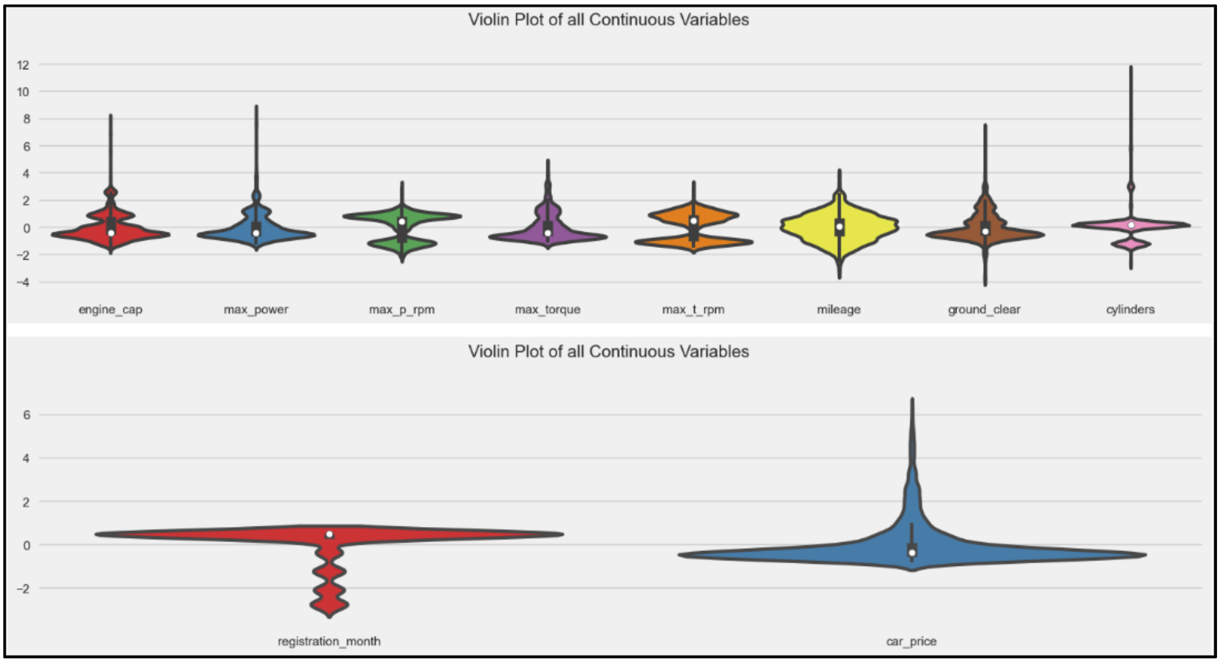

max_t_rpm
6000
5000
4000
3000
2000
1000
3000
4000
5000
6000
7000
8000
max_p_rpm
mileage
30
25
20
15
10
5
3000
4000
5000
6000
7000
8000
max_p_rpm
ground_clear
300
275
250
225
200
175
150
125
3000
4000
5000
6000
7000
8000
max_p_rpm
cylinders
12
10
8
6
4
2
3000
4000
5000
6000
7000
8000
max_p_rpm

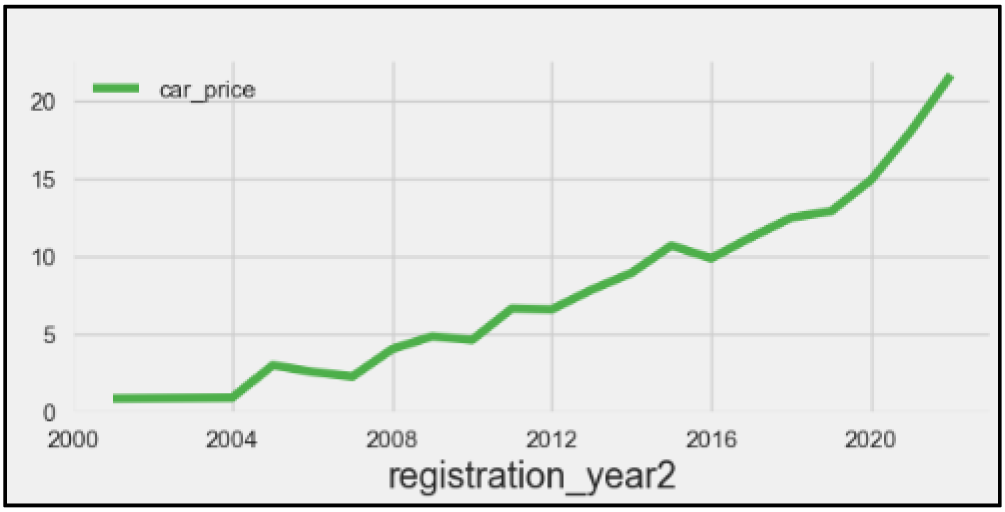

registration_month
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
3000
4000
5000
6000
7000
8000
max_p_rpm
max_t_rpm
6000
5000
4000
3000
2000
1000
100
200
300
400
500
600
700
800
max_torque
mileage
30
25
20
15
10
5
100
200
300
400
500
600
700
800
max_torque
ground_clear
300
275
250
225
200
175
150
125
100
200
300
400
500
600
700
800
max_torque

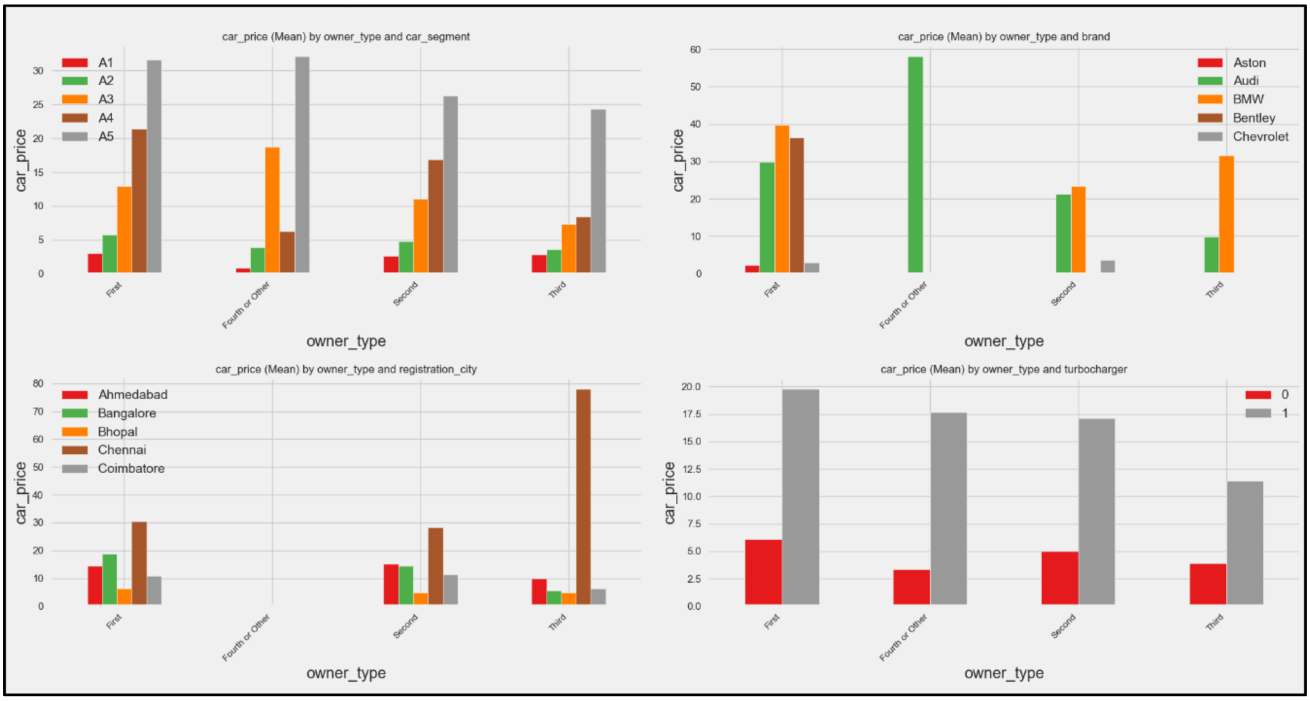

cylinders
2
4
6
8
10
12
max_torque
100
200
300
400
500
600
700
800
registration_month
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
max_torque
mileage
5
10
15
20
25
30
max_t_rpm
1000
2000
3000
4000
5000
6000
ground_clear
125
150
175
200
225
250
275
300
max_t_rpm

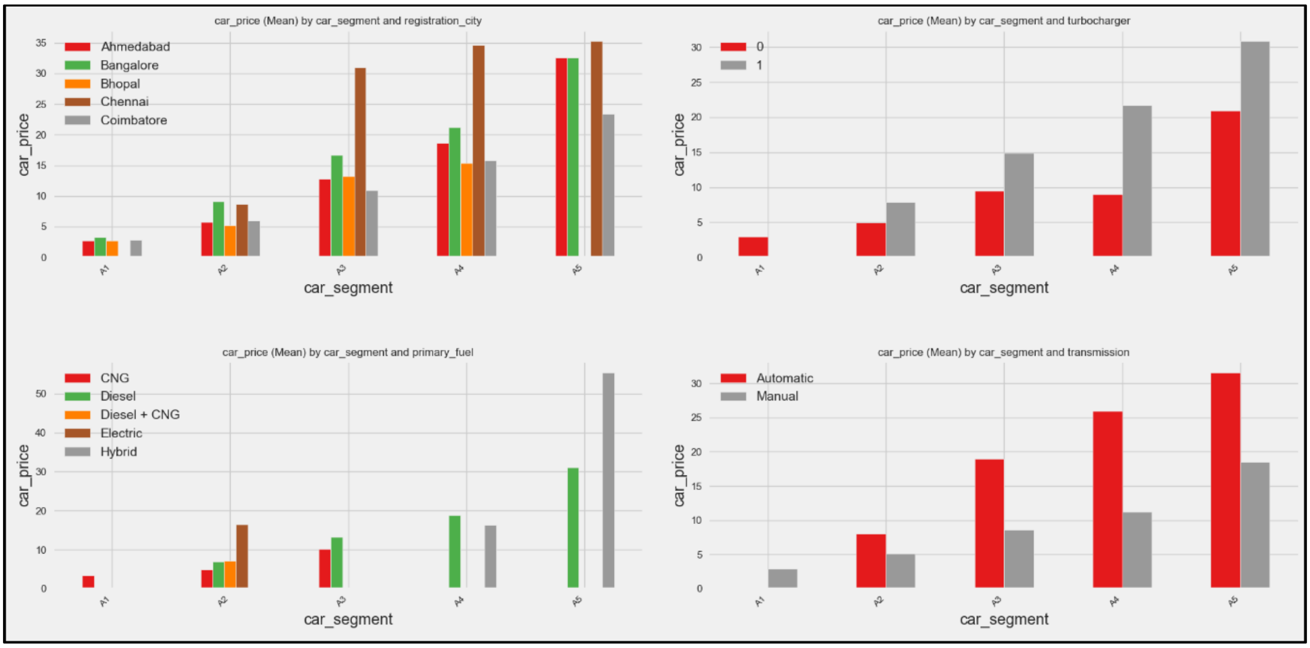

cylinders
2
4
6
8
10
12
max_t_rpm
1000
2000
3000
4000
5000
6000
registration_month
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
max_t_rpm
ground_clear
125
150
175
200
225
250
275
300
mileage
5
10
15
20
25
30
cylinders
mileage

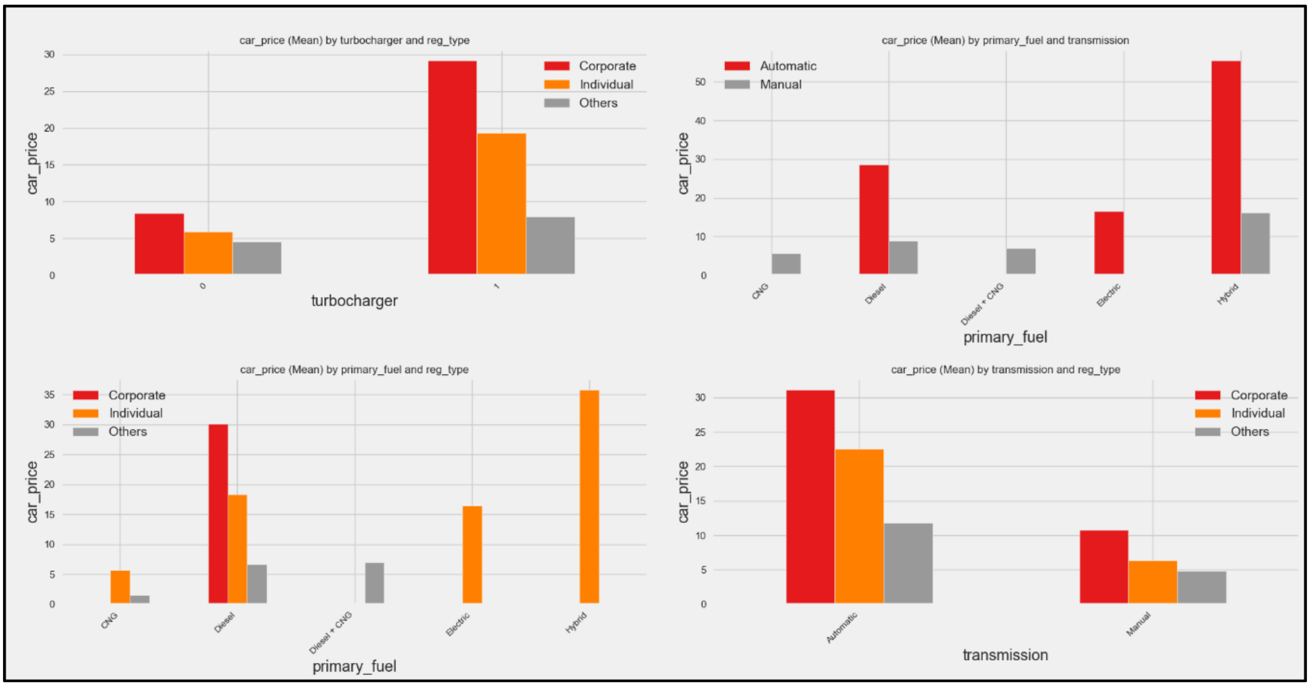

registration_month
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
mileage
5
10
15
20
25
30
cylinders
2
4
6
8
10
12
ground_clear
125
150
175
200
225
250
275
300
registration_month
ground_clear
registration_month
cylinders


Violin Plot of all Continuous Variables
12
10
8
6
4
2
0
−2
−4
engine_cap
max_power
max_p_rpm
max_torque
max_t_rpm
mileage
ground_clear
cylinders
Violin Plot of all Continuous Variables
6
4
2
0
−2
registration_month
car_price

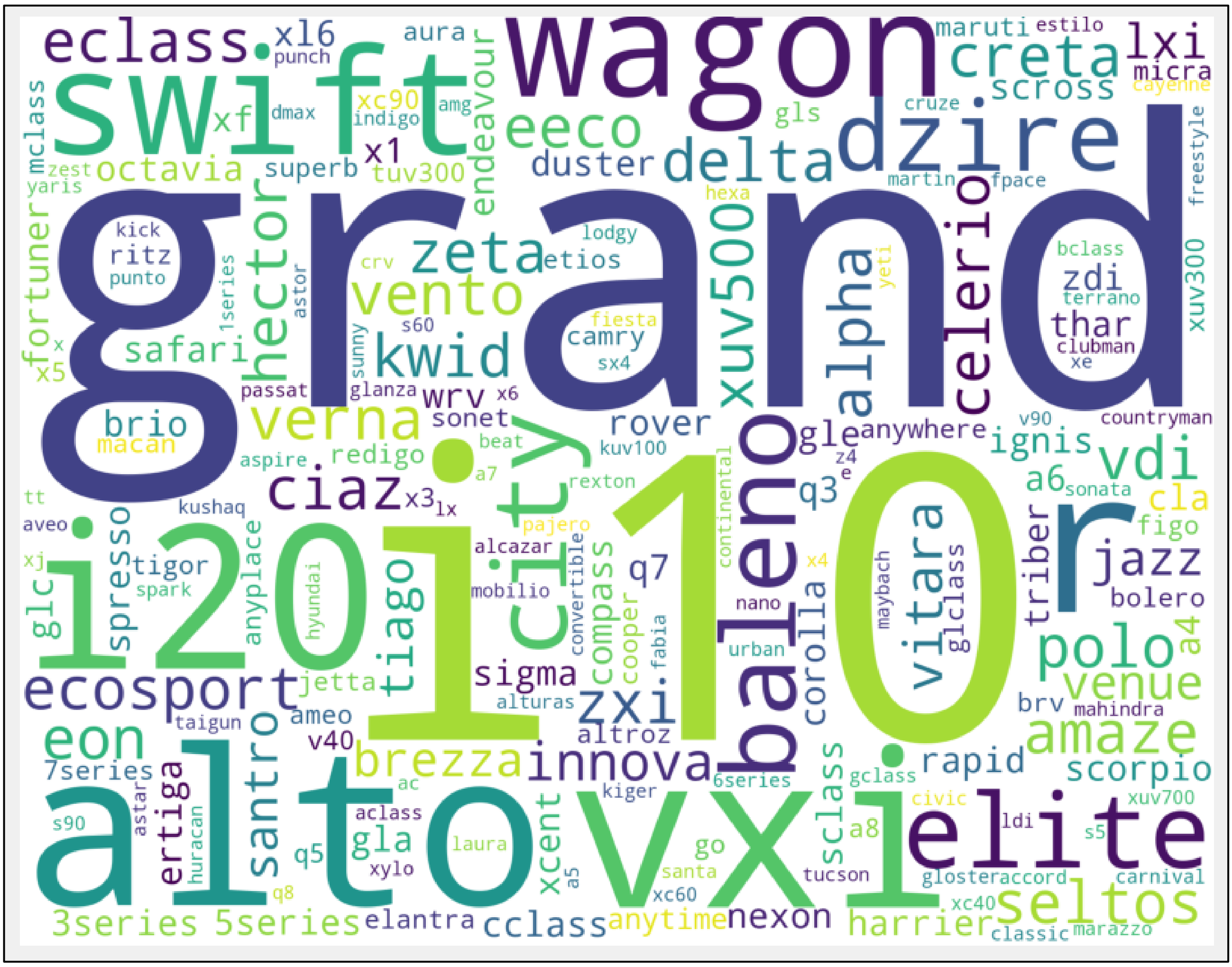

car_price
20
15
10
5
0
2000
2004
2008
2012
2016
2020
registration_year2


car_price (Mean) by owner_type and car_segment
A1
A2
A3
A4
A5
car_price
30
25
20
15
10
5
0
First
Fourth or Other
Second
Third
owner_type
car_price (Mean) by owner_type and brand
Aston
Audi
BMW
Bentley
Chevrolet
car_price
60
50
40
30
20
10
0
First
Fourth or Other
Second
Third
owner_type
car_price (Mean) by owner_type and registration_city
Ahmedabad
Bangalore
Bhopal
Chennai
Coimbatore
car_price
80
70
60
50
40
30
20
10
0
First
Fourth or Other
Second
Third
owner_type
car_price (Mean) by owner_type and turbocharger
0
1
car_price
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
First
Fourth or Other
Second
Third
owner_type


car_price (Mean) by car_segment and registration_city
Ahmedabad
Bangalore
Bhopal
Chennai
Coimbatore
car_price
0
5
10
15
20
25
30
35
A1
A2
A3
A4
A5
car_segment
car_price (Mean) by car_segment and turbocharger
0
1
car_price
0
5
10
15
20
25
30
A1
A2
A3
A4
A5
car_segment
car_price (Mean) by car_segment and primary_fuel
CNG
Diesel
Diesel + CNG
Electric
Hybrid
car_price
0
10
20
30
40
50
A1
A2
A3
A4
A5
car_segment
car_price (Mean) by car_segment and transmission
Automatic
Manual
car_price
0
5
10
15
20
25
30
A1
A2
A3
A4
A5
car_segment


car_price (Mean) by turbocharger and reg_type
Corporate
Individual
Others
car_price
0
5
10
15
20
25
30
0
1
turbocharger
car_price (Mean) by primary_fuel and transmission
Automatic
Manual
car_price
0
10
20
30
40
50
CNG
Diesel
Diesel + CNG
Electric
Hybrid
primary_fuel
car_price (Mean) by primary_fuel and reg_type
Corporate
Individual
Others
car_price
0
5
10
15
20
25
30
35
CNG
Diesel
Diesel + CNG
Electric
Hybrid
primary_fuel
car_price (Mean) by transmission and reg_type
Corporate
Individual
Others
car_price
0
5
10
15
20
25
30
Automatic
Manual
transmission

Word cloud of car models;



Observations from Univariate & Bivariate Analysis:

1. Petrol & diesel were the most commonly used primary fuel for cars.
2. Most of the cars were having manual transmission system as compared to Automatic transmission system.
3. Of all the cars listed on the website almost 70% of the cars were being sold for the first time from respective owners.
4. 25% of the cars were not insured.
5. More than 90% of the cars were having Individual type of registrations.
6. More than 65% of the cars were front wheel driven.
7. Almost half of the listed cars were having turbocharger device.
8. Most of the cars were of A2 Class.
9. Maruti & Hyundai were the most common brands of the cars listed for selling.
10. Grand i10 & WagonR were common type of models sold online on carwale.com.
11. Hyderabad city had the greatest number of used car sellers.
12. White was the most common color of the cars listed on carwale.com
13. Engine volumetric capacity is positively correlated to cylinders, max power & max torque.
14. max power & max torque are positively correlated.
15. Latest the registration year of the car higher the price of the car listed on carwale.com
16. A3, A4 & A5 class of cars are costlier.
17. Cars with automatic transmission & turbocharger devices are usually costlier as seen from the graphs.

CONCLUSION

Key Findings and Conclusions of the Study

- Cars from Maruti & Hyundai brands with white color, latest registration year, turbocharger device, automatic transmission is more likely to be sold on the carwale.com
- Random Forest Regressor was best performing model, with accuracy above 90%.

Learning Outcomes of the Study in respect of Data Science

- Identifying best possible techniques to scrape data from specific websites.
- Dealing with huge amount of data scraping
- Data cleaning & preprocessing is vey much easy if you know what data was scraped & how it was scraped.

Limitations of this work and Scope for Future Work

- Some important data was not listed by the seller. This makes data scraping consume more time & electricity.
- Some cars were used to sell immediately & were unlisted ASAP, this creates issues while data scraping.
- Sometimes page takes too much time to load & scraping fails.
- high computational setup is required for scraping large data.