USED CAR PREDICTION MODEL

Submitted By:

Junaid Shaikh

ACKNOWLEDGMENT

I would like to thank Flip Robo Technologies for providing me with the opportunity to work on this project from which I have learned a lot.
Most of the concepts used to predict the price of used cars are learned from Data Trained Institute and below documentations.

Some of the reference sources are as follows:

- Medium.com
- StackOverflow

Contents

INTRODUCTION

BUSINESS PROBLEM FRAMING

In this project, we have to make used car price valuation model using new machine learning models from new data. Because with the change in market due to covid 19 impact, our client is facing problems with their previous car price valuation machine learning models. 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.

Conceptual Background OF The Domain Problem

Firstly, we will prepare our own dataset using web scraping.

After that we will check whether the project is a regression type or a classification type.

We will also check whether our dataset is balanced or imbalanced. If it is an imbalanced one, we will apply sampling techniques to balance the dataset.

Then we will do model building and check its accuracy.

Our main motto is to build a model with good accuracy and for that we will also go for hyperparameter tuning.

Review OF Literature

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.

Motivation OF The Problem Undertaken

There are websites that offers an estimate value of a car. They may have a good prediction model. However, having a second model may help them to give a better prediction to their users. Therefore, the model developed in this study may help online web services that tells a used car's market value.

Analytical Problem Framing

Mathematical/ Analytical Modeling OF The Problem

In this project we have performed various mathematical and statistical analysis such as we checked description or statistical summary of the data using describe, checked correlation using corr and visualized it using heatmap.
The given dataset has 9807 rows and 18 rows. Price as the target column containing continuous value. Hence it is a regression problem and regression algorithms will be used while building the model. There are null values in the dataset. It was observed that more than 90% similar values in some columns, if kept, then it will create high skewness in the model hence decided to drop those columns. To get better insight on the features different visualization tools have been used like distribution plot, bar plot and count plot. Outliers and skewness were detected in the dataset which were then reduced using percentile method and yeo-Johnson method respectively. I have used all the regression algorithms while building model then hyper-tunned the best model and saved the best model. At last, I have predicted the label using saved model.

Data Sources & Data Formats

The data was collected from cardekho.com & carwale.com websites in excel format. The data was scrapped using selenium. After scrapping required features the dataset is saved as excel file.

Also, my dataset was having 9807 rows and 18 columns including target variable. In this particular dataset I have object type of data which has been changed as per our analysis about the dataset.

Features Information:

- Year : what year is the car of.
- Fuel : Type of fuel used for car engine
- kms : Car running in kms till the date
- owners: previous number of owners
- Engine: Engine CC
- transmission : Type of gear transmission used in car
- Milage : Overall milage of car in Km/ltr
- Seating: Availability of number of seats in the car
- power : Maximum power of engine used in car in bhp
- power_rpm : Maximum power rpm of engine used in car in bhp
- torque: max torque of engine used in car
- torque_rpm: max torque rpm of engine used in car
- gear: number of gears in a car
- drive: whether it is FWD,RWD or AWD type vehicle
- values_cylinder: number of cylinder of engine in a a car.
- No_cylinder: per cylinder number of cylinder
- price : Price of the car

After scrapping, the data was provided in csv (comma separated values) format.

The given dataset has 9807 rows and 18 rows. There are no null values in the dataset.

Dataset was imported using Panda's library and then transformed into data-frame.

Dataset

```
data
```

	year	fuel	kms	owners	transmission	mileage	engine	power	power_rpm	seats	gear	drive	steering	torque	torque_rpm	values_cylinder
0	1995	4	35000	4	1	0.00	2112	0.00	0	6	0	0	Power	0	0	4
1	2016	4	120000	1	1	28.09	1248	88.50	0	5	5	2	Power	200	1750	4
2	2014	5	60000	2	1	16.51	1586	103.20	0	5	5	2	Power	145	4100	4
3	2017	5	40000	1	1	21.21	1197	81.80	0	5	5	2	Power	113	4200	4
4	2018	5	46000	1	1	23.10	998	67.04	0	5	5	2	Power	90	3500	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9802	2017	4	40000	1	2	18.56	1995	190.00	4000	5	8	3	Power	400	1750	4
9803	2015	4	14000	1	2	16.66	1968	177.00	3750	5	0	2	Power	380	1750	4
9804	2019	5	15000	1	2	0.00	1991	194.00	5500	5	9	1	Power	320	1650	4
9805	2020	5	15000	1	2	17.00	1353	138.00	6000	5	7	2	Power	242	1500	4
9806	2017	4	24009	1	2	16.38	1999	177.00	4000	5	8	3	Power	430	1750	4

9807 rows × 18 columns

Data Pre-processing

Current dataset is raw data. By proper Data Transformation methods, a lot of valuable insights can be gained.

Then statistical analysis was done by checking shape, value counts, info etc.….

Then while looking into the value counts, I found some columns with more than 90% data having same values, this creates skewness in the model and there are chances of getting model bias, so I have dropped those columns with more than 90% same values.

While checking for null values I found no null values in the dataset.

I have also dropped steering column as I found it has more than 90% same value.

Next as a part of feature extraction I converted the year column to how old the car is w.r.t this year. Thinking that this data will help us more.

In this project we have performed various mathematical and statistical analysis such as description or statistical summary of the data using describe, checked correlation using corr and visualized it using heatmap. Then we have used Z-Score to plot outliers and remove them.

```
data.describe()
```

	year	fuel	kms	owners	transmission	mileage	engine	power	power_rpm	seats	gear
count	9807.000000	9807.000000	9807.000000	9807.000000	9807.000000	9807.000000	9807.000000	9807.000000	9807.000000	9807.000000	9807.000000
mean	2015.275517	4.524421	52637.781279	1.187213	1.273478	17.971169	1509.431325	107.292730	2372.082084	5.183950	5.236056
std	3.098123	0.527628	34571.329637	0.436023	0.495805	6.099944	638.297041	60.316947	2526.569381	1.009245	1.495200
min	1989.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2013.000000	4.000000	29000.000000	1.000000	1.000000	15.400000	1197.000000	74.000000	0.000000	5.000000	5.000000
50%	2016.000000	5.000000	48914.000000	1.000000	1.000000	18.700000	1248.000000	88.730000	4.000000	5.000000	5.000000
75%	2018.000000	5.000000	70091.000000	1.000000	2.000000	21.790000	1598.000000	122.000000	4200.000000	5.000000	6.000000
max	2021.000000	5.000000	910000.000000	4.000000	2.000000	33.540000	5998.000000	626.000000	8250.000000	10.000000	10.000000

Data Inputs-Logic-Output Relationship

Since I had numerical and non-numerical columns, I have plotted dist. plot to see the distribution of each column data.

I have used box plot for each pair of categorical features that shows the relation between target and independent features.

Hardware and Software Requirements and Tools Used

Hardware required:

- Processor — core i5 and above
- RAM — 8 GB or above
- SSD — 250GB or above

Software/s required: Anaconda

LIBRARIES:

The tools, libraries, and packages we used for accomplishing this project are pandas, NumPy, matplotlib, seaborn, SciPy, sklearns's, mlxtend, xgboost, joblib.

Through panda's library we loaded our csv file 'Data file' into data frame and performed data manipulation and analysis.

With the help of NumPy we worked with arrays.

With the help of matplotlib and seaborn we did plot various graphs and figures and done data visualization.

Train_test_split is a function in Sklearns's model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearns's Train_test_split will make random partitions for the two subsets.

With sklearns's StandardScaler package we scaled all the feature variables onto single scale. As these columns are different in scale, they are standardized to have common scale while building machine learning model. This is useful when you want to compare data that correspond to different units.

With sklearns's package we imported many regression models, we could obtain cross_val_score, which is an accuracy metric used to evaluate model, we could obtain best parameters of a model using GridsearchCV or RandomizedSearchCV, we could reduce skewness using power transform library of sklearns's.

Model/s Development and Evaluation

Identification of possible problem-solving approaches

For skewness removal I have used power transform method, for scaling down I have used standard scaling.

```
In [77]: #removing skewness by power transform
         from sklearn.preprocessing import power_transform
         x=power_transform(x,method='yeo-johnson')
```

Scaling

```
In [78]: #applying standard scaling method on X parameters
         from sklearn.preprocessing import StandardScaler
         scalar=StandardScaler()
         x=scalar.fit_transform(x)
```

Apart from this multicollinearity refers to the collinearity between the features. Multicollinearity occurs when our model includes multiple factors that are correlated with each other's other than with label. It makes more difficult for the model predict and affects the accuracy. They are treated using VIF method. This algorithm reduces the no. of columns by removing highly correlated feature columns.

Testing of Identified Approaches (Algorithms)

Since price was my target variable and it was a numerical column, so this problem was regression problem. And I have used all regression algorithms to build my model. By looking into the difference of accuracy score and cross validation score I found LGBM as a best model with least difference. Also, to get the best model we must run through multiple models and to avoid the confusion of overfitting we have go through cross validation.

Run & evaluate selected models

Below are all the models through which we need to iterate through to find the best model

```
models=[GradientBoostingRegressor(),NuSVR(),LinearRegression(),Ridge(),RidgeCV(),BayesianRidge(),SGDRegressor(),SVR(),
        AdaBoostRegressor(),LinearSVR(),KNeighborsRegressor(),RandomForestRegressor(),BaggingRegressor(),
       DecisionTreeRegressor(),LGBMRegressor(), XGBRFRegressor(),XGBRegressor(),LogisticRegression(),ExtraTreesRegressor()]
```

Below is the code to iterate all the models, train them and display their acc

```
for i in models:
    x_train,x_test,y_train,y_test=train_test_split(x,y,random_state = 42,test_size=0.20)
    scores=cross_val_score(i,x_train,y_train,cv=5,scoring='r2',n_jobs=-1)
    score=np.mean(scores)
    i.fit(x_train,y_train)
    y_pred=i.predict(x_test)
    if r2_score(y_test,y_pred)>score:
        diff=r2_score(y_test,y_pred)-score
    else:
        diff=score-r2_score(y_test,y_pred)
    print('*'*10)
    print(i)
    print('score',score)
    print('r2',r2_score(y_test,y_pred))
    print('diff',diff)
    print('mae',mean_absolute_error(y_test, y_pred))
    print('rmse',np.sqrt(mean_squared_error(y_test, y_pred)))
```

Key Metrics for success in solving problem under consideration

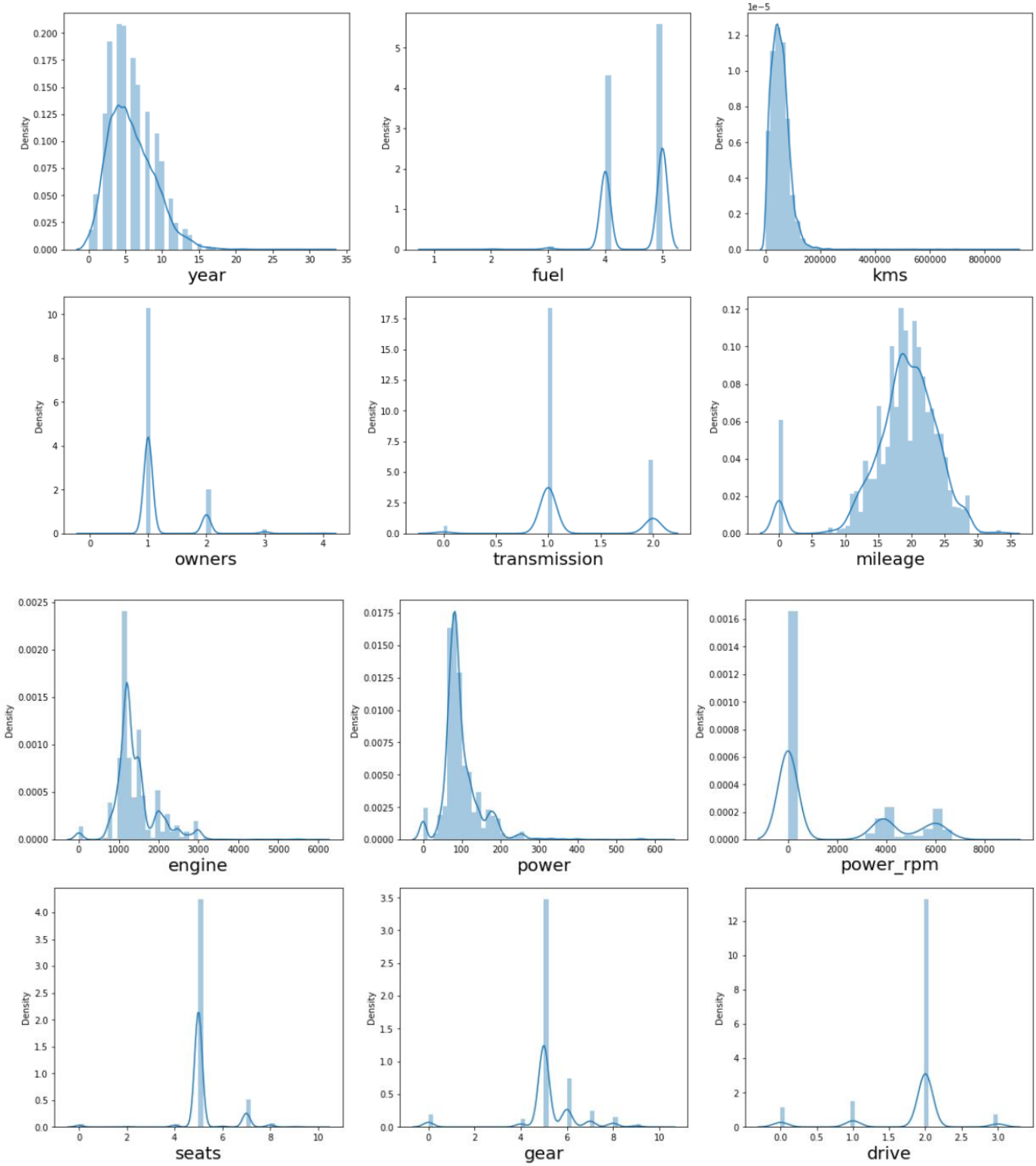
Following metrics were used to evaluate our model:

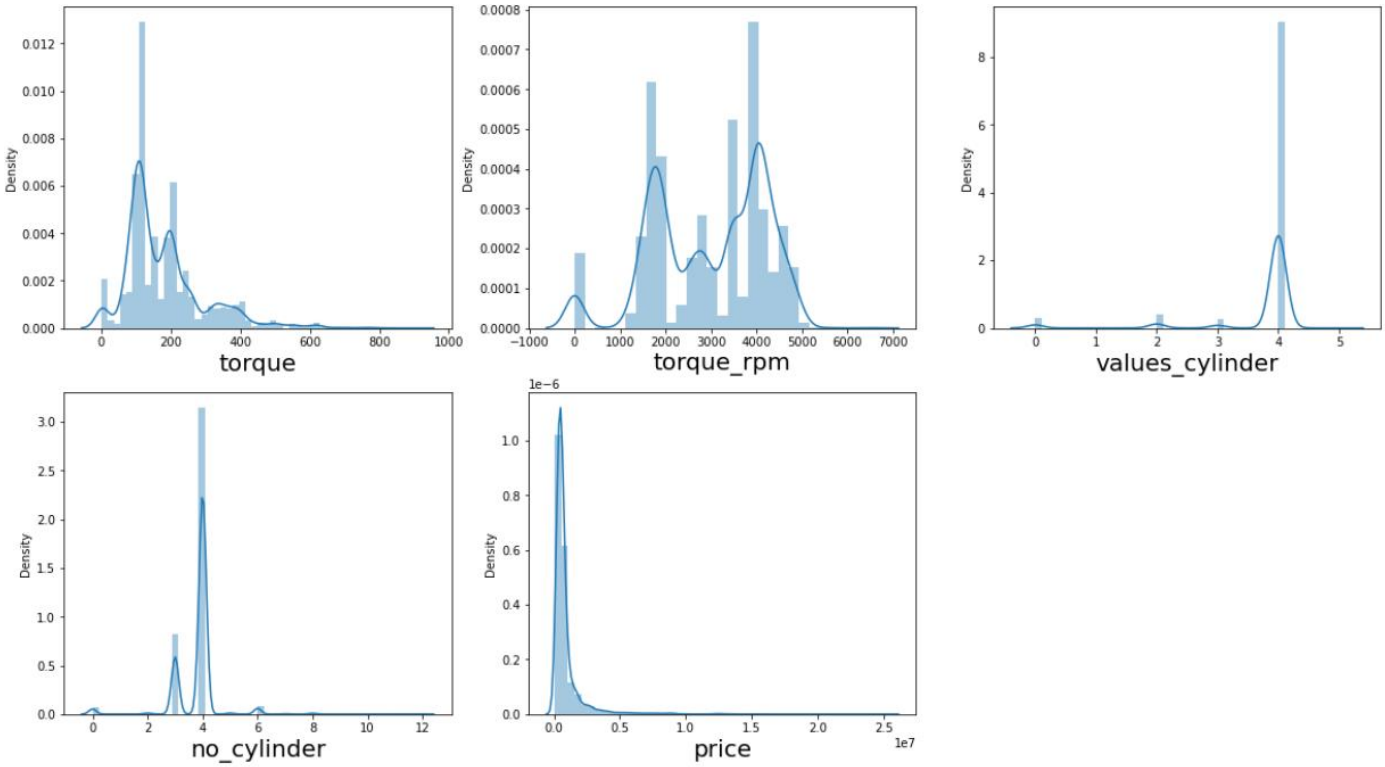
--- Cross Val Score
--- R2 Score
--- STD error
--- mean absolute error
--- square root of mean absolute error

Visualization

Univariate Analysis:

Distribution plot of all columns

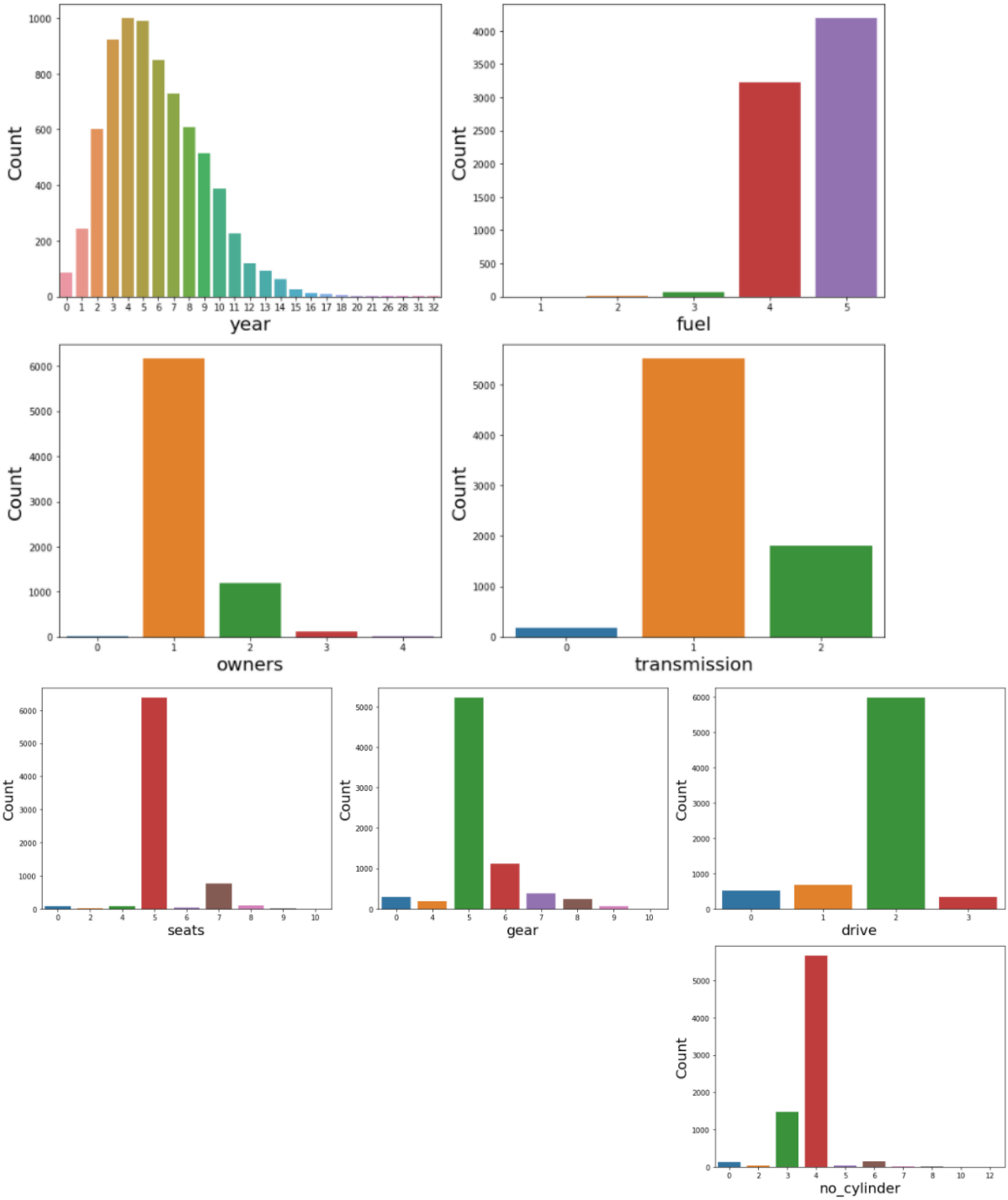




Observations:

- kms column has normal distributed value points but it is highly skewed and contains outliers
- mileage column has bimodal distributed value points with less skewness and less outliers
- engine & power column has normal distributed value points but contain outliers
- power_rpm column has bimodal distributed value points
- torque & torque_rpm column has bimodal distributed value points with outliers

Count plot of all categorical column

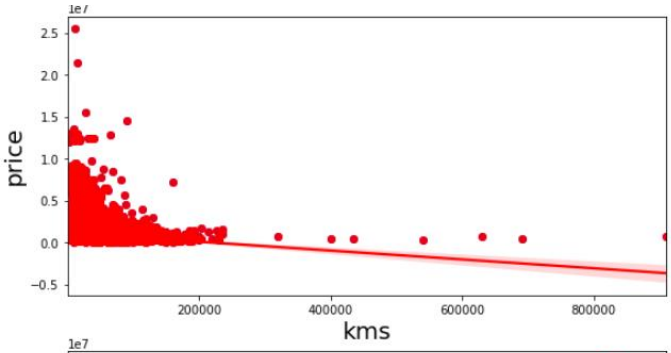


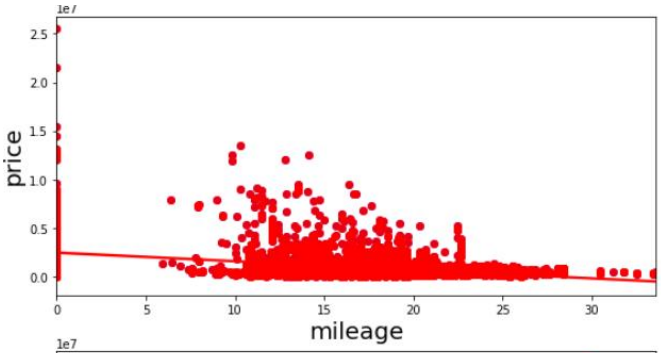
Observation:

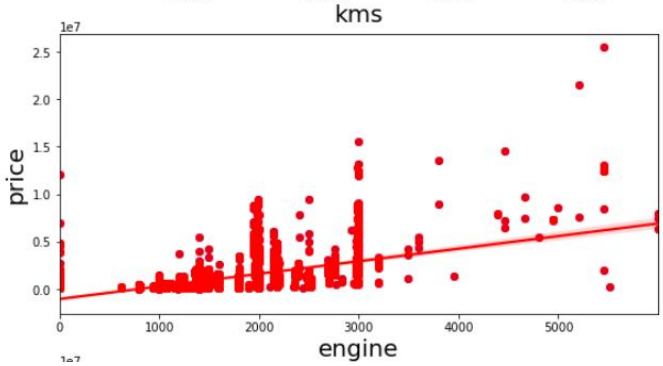
From the above graph we can observe the value_counts of datapoints of resp column

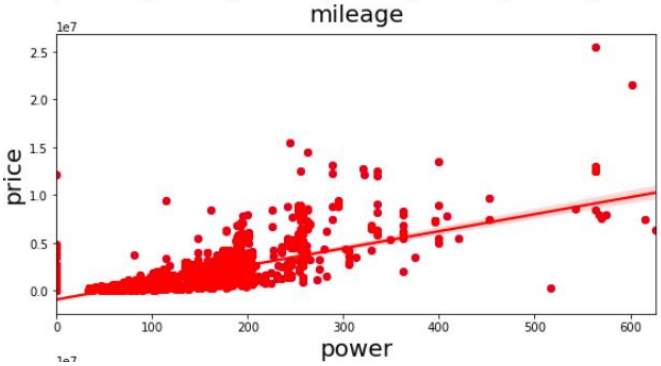
we can make a note about the data i.e., dataset contains data mostly of which year, transmision, range of gears of a car.....

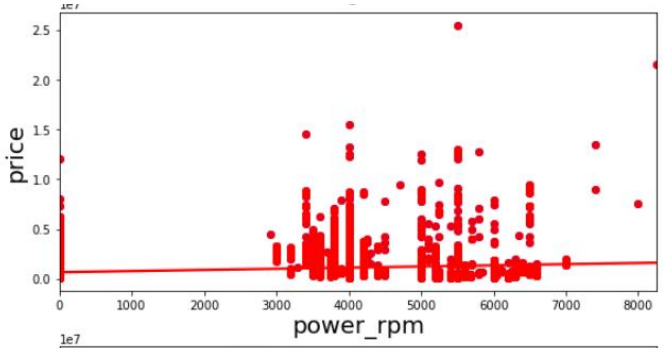
Bivariate analysis for numerical columns

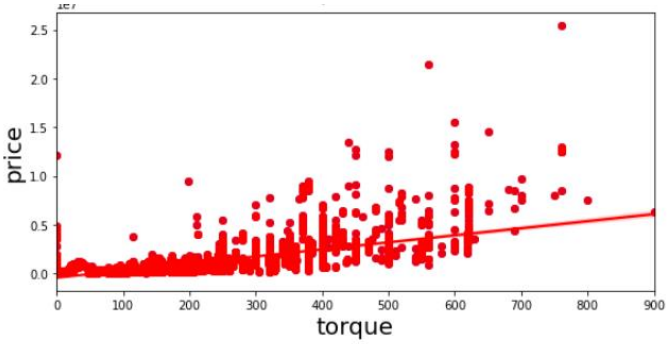


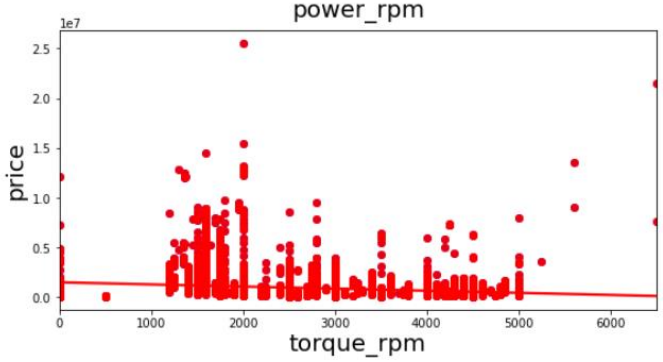










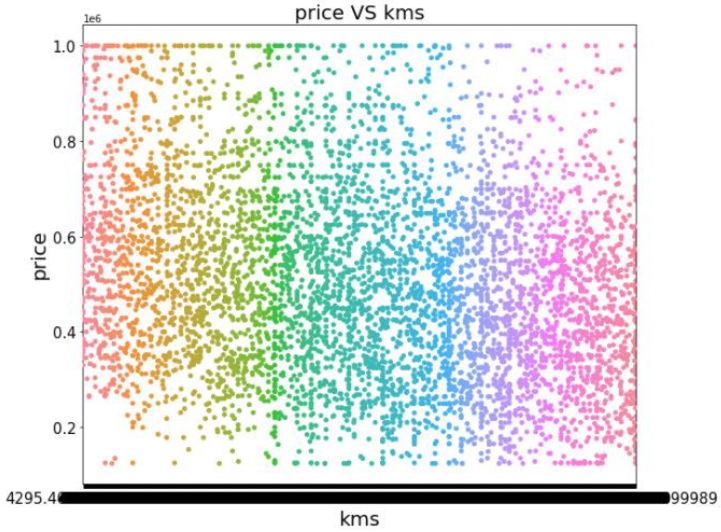


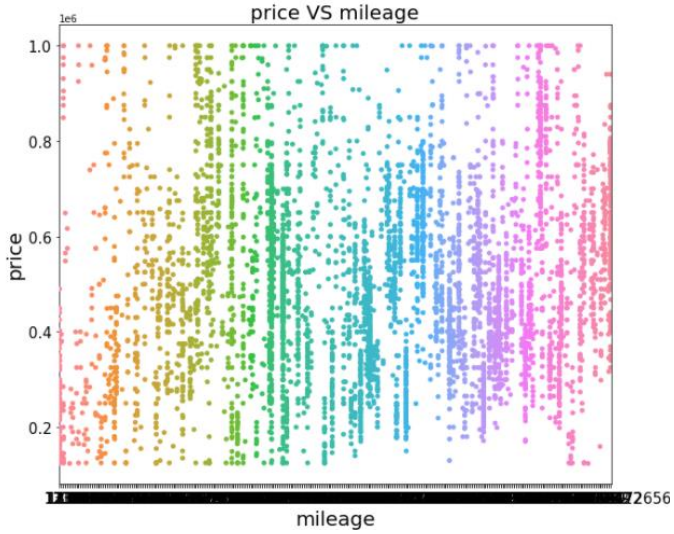
Observation:

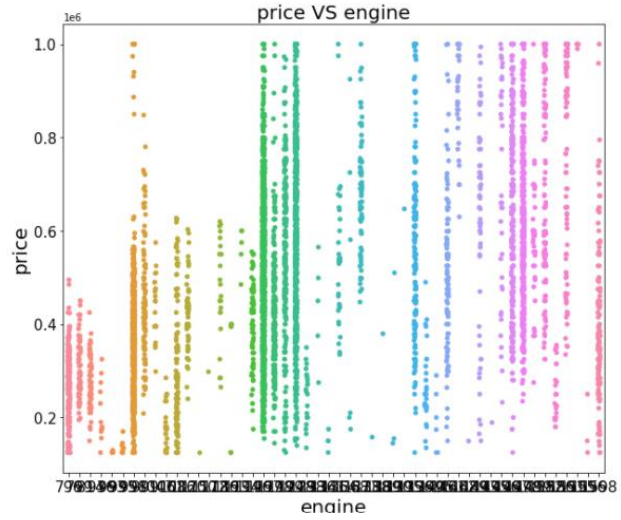
Columns having positive correlation with price are: torque, power, engine & power_rpm

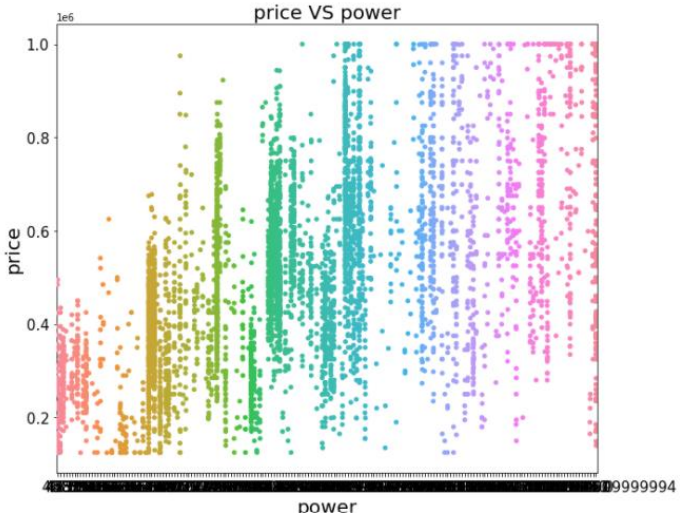
Columns having negative correlation with price are: kms, mileage & torque_rpm

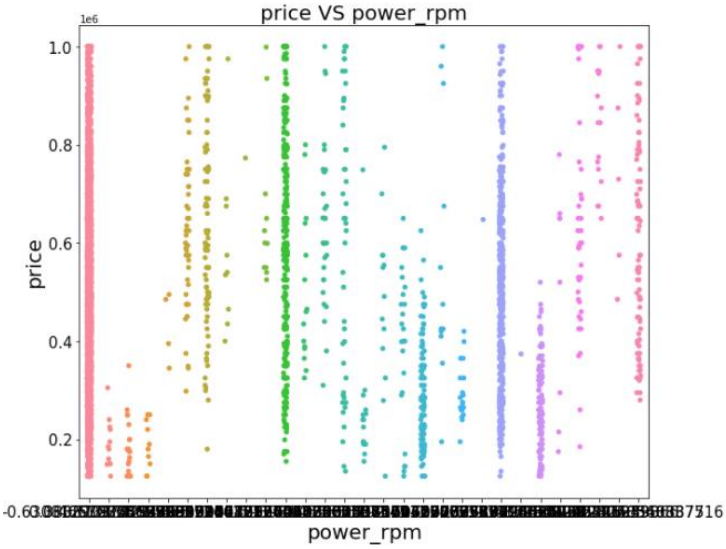
Strip Plot of each column values w.r.t label column

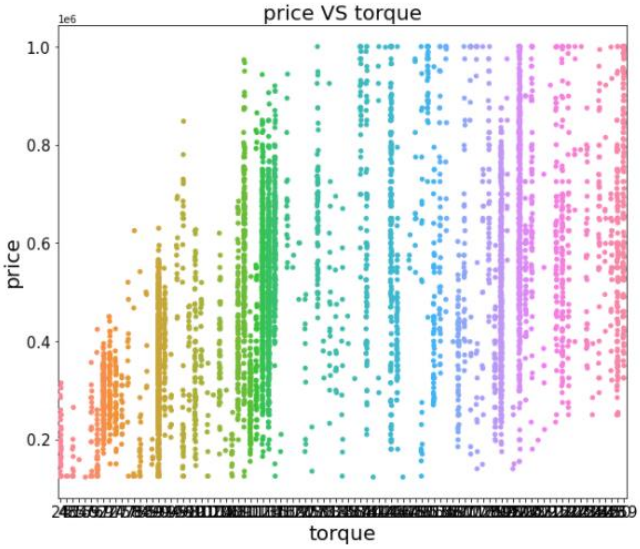


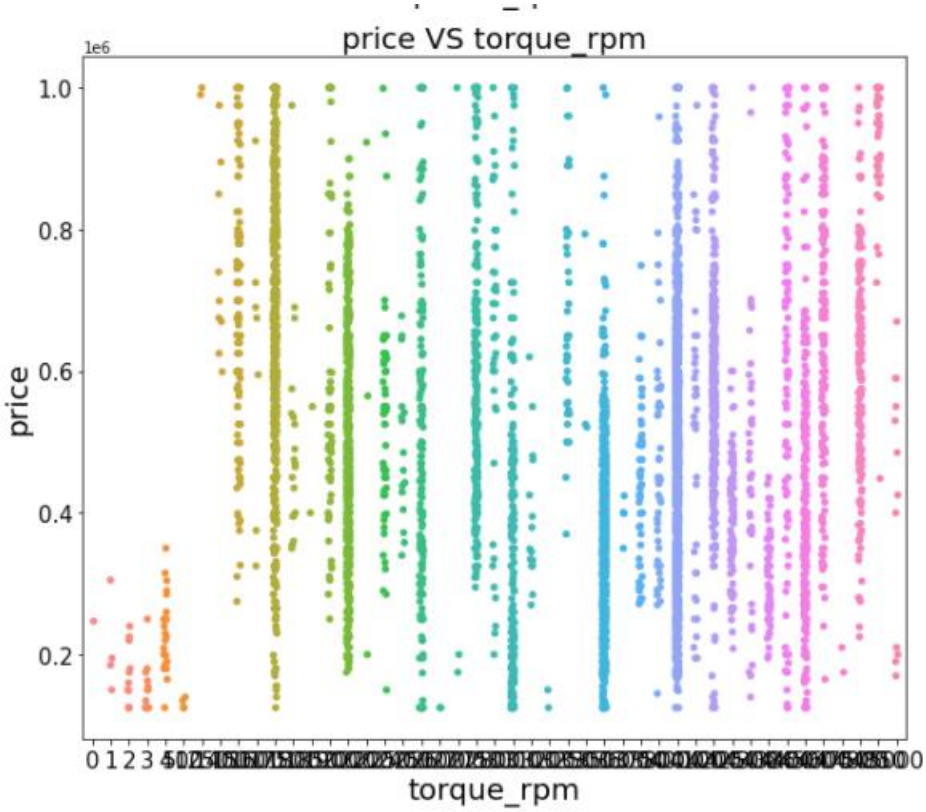








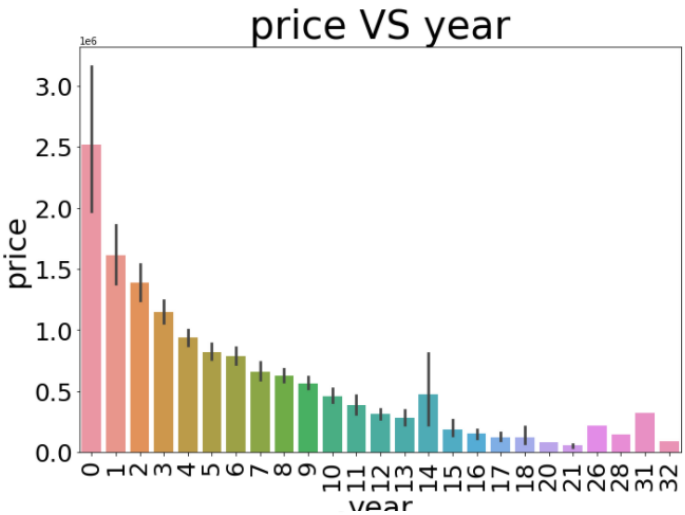


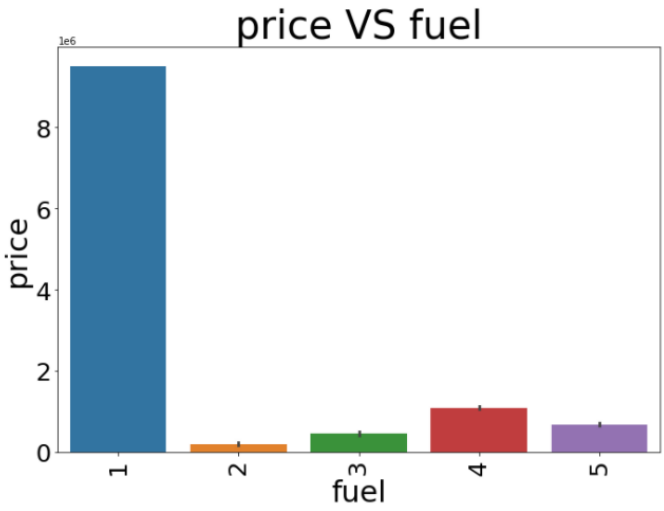


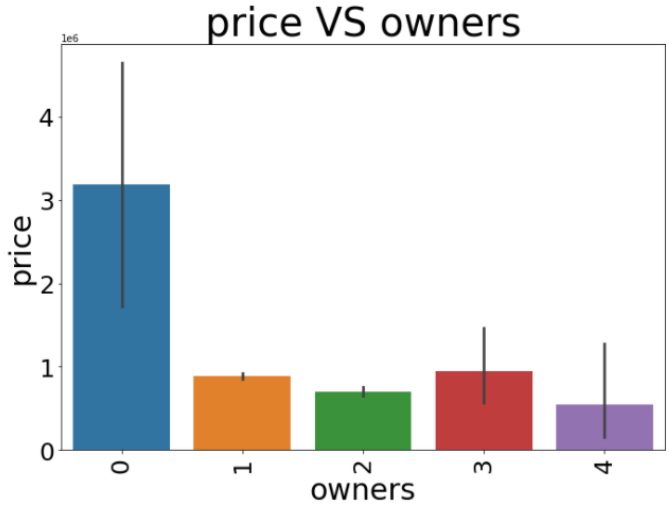
Observation:

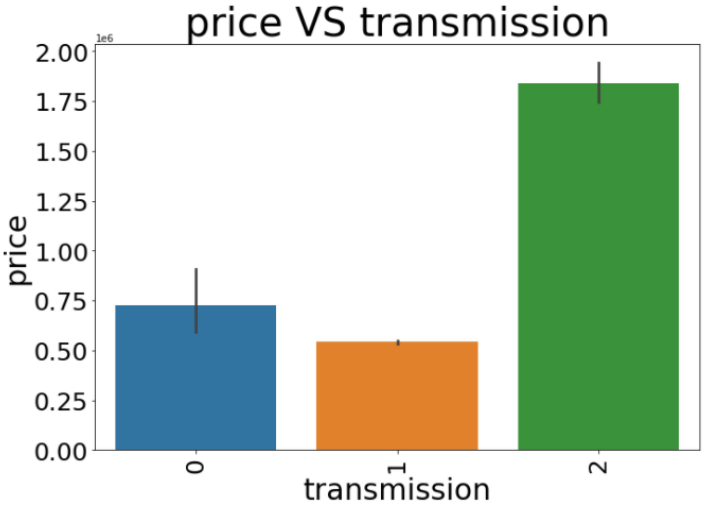
for higher value of torque and higher value of engine, price rate of car is higher

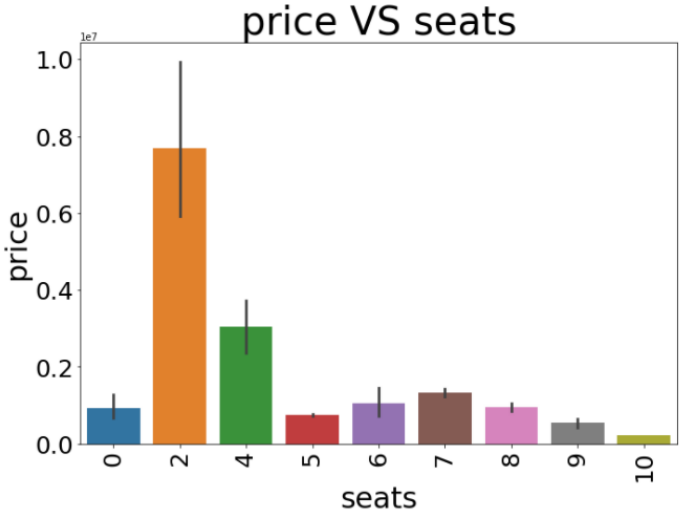
Bar graph for categorical column

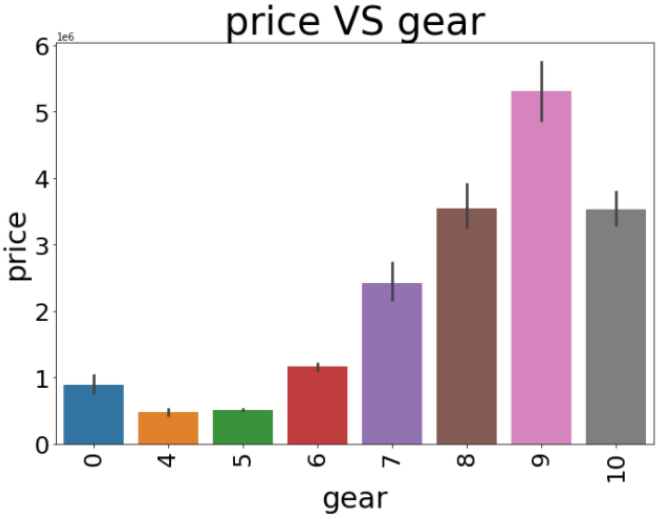


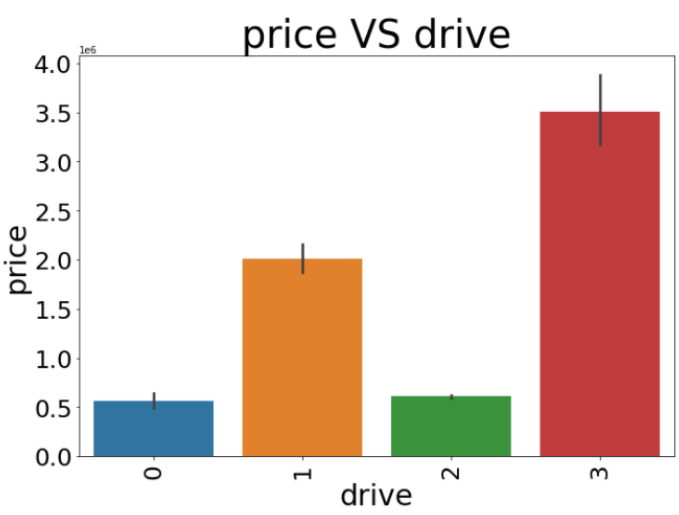


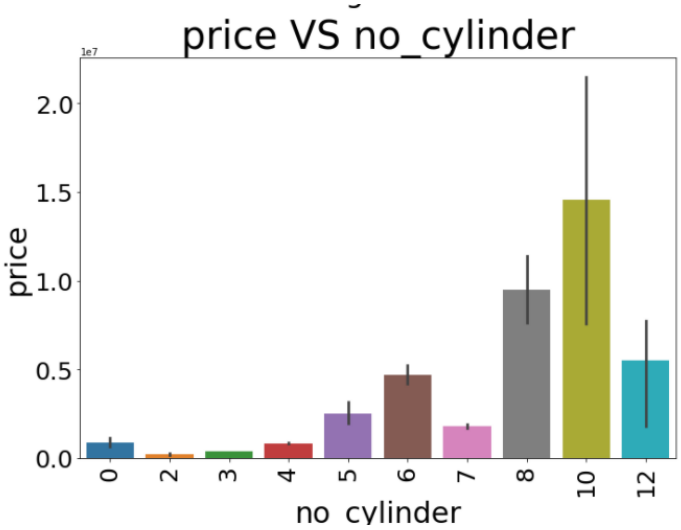










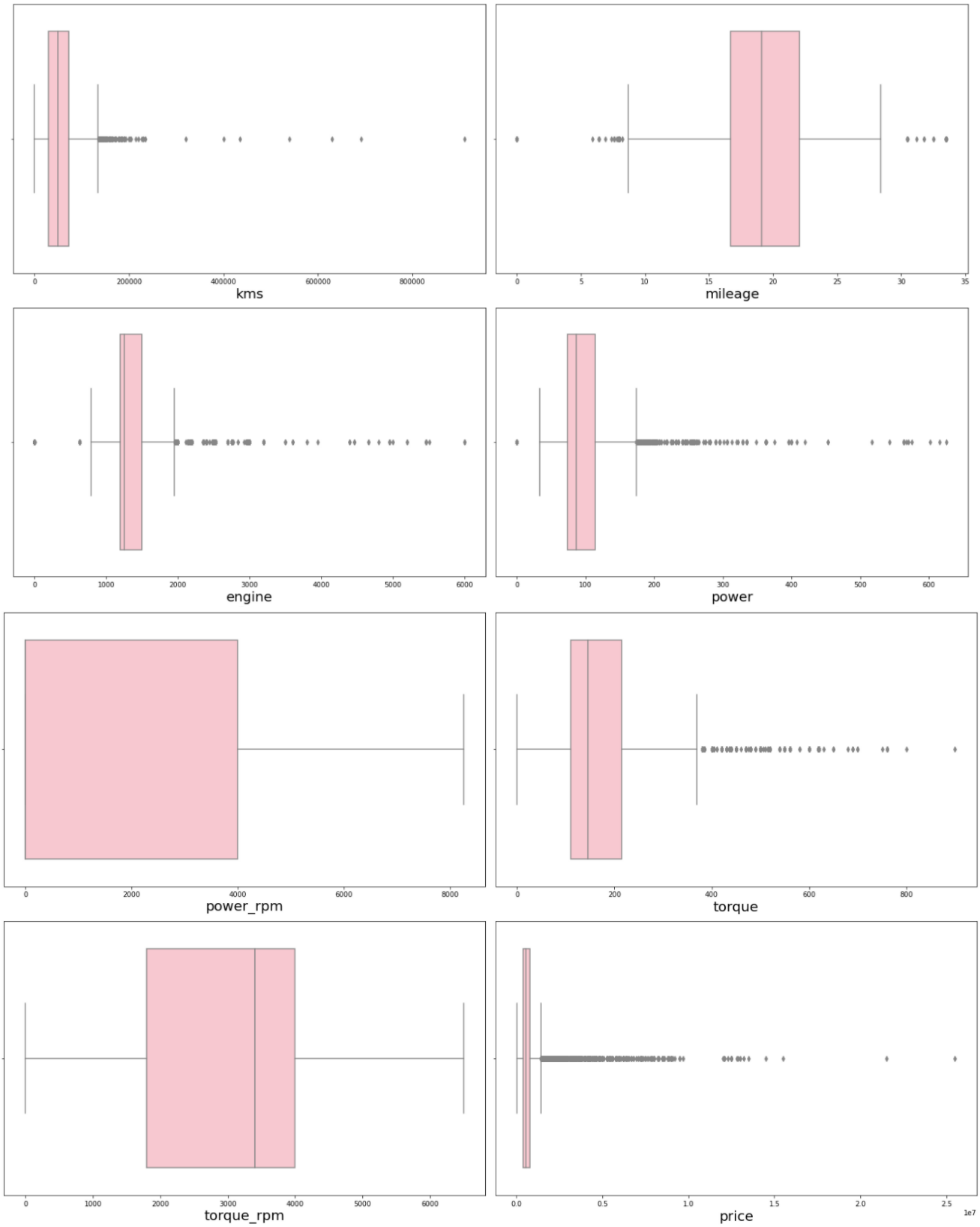


Observation:

- newer models of car have higher sell price
- cars having hybrid fuel type have higher sell price
- unregistered cars have higher sell price
- automatic transmission car have higher sell price
- 2 seater cars have highest sell price
- cars with more number of gear have higher sell price
- cars having all whell drive have higher sell price
- higher the number of cylinder of an engine in a car has higher sell price

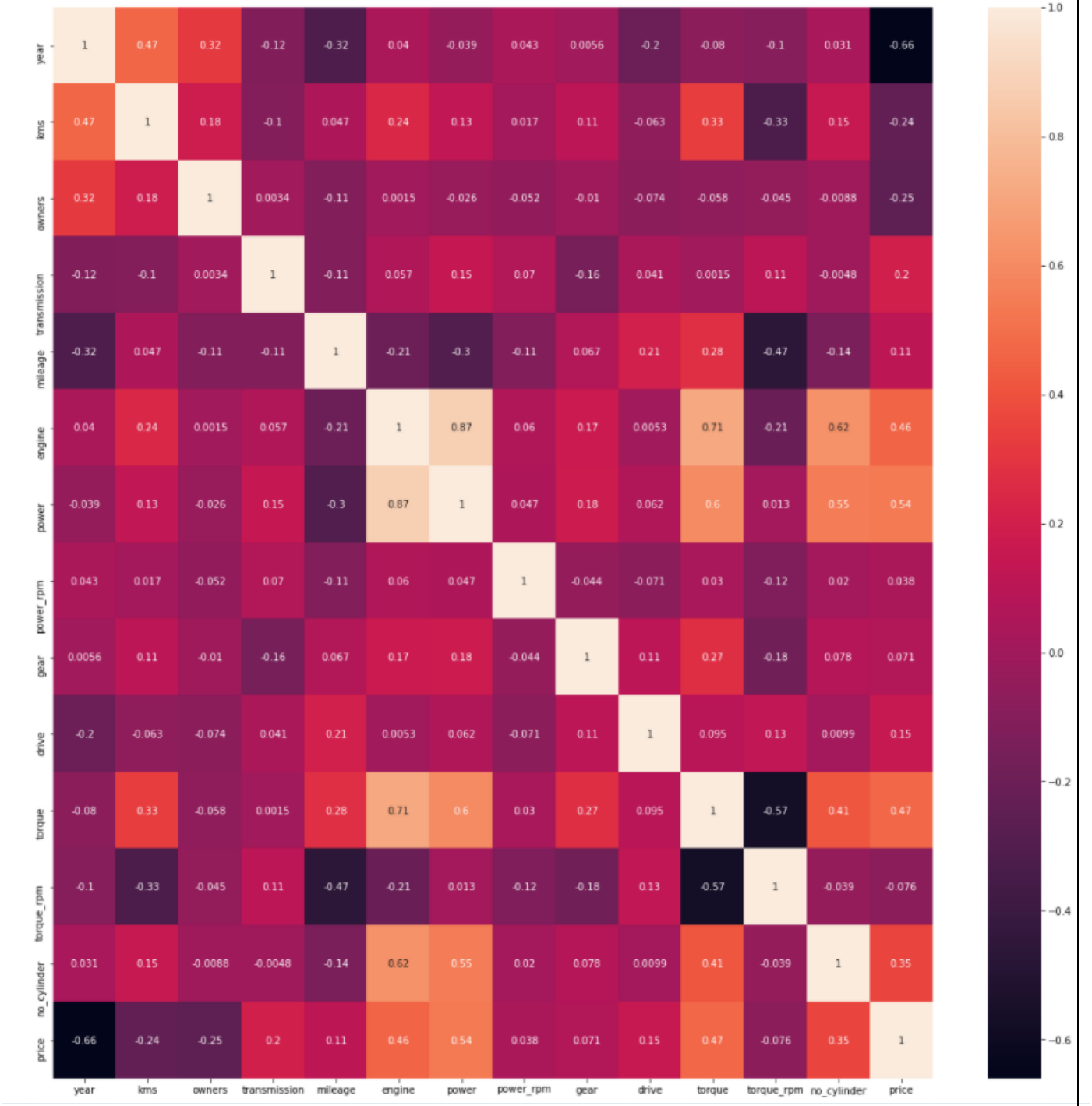
Check For Outliers

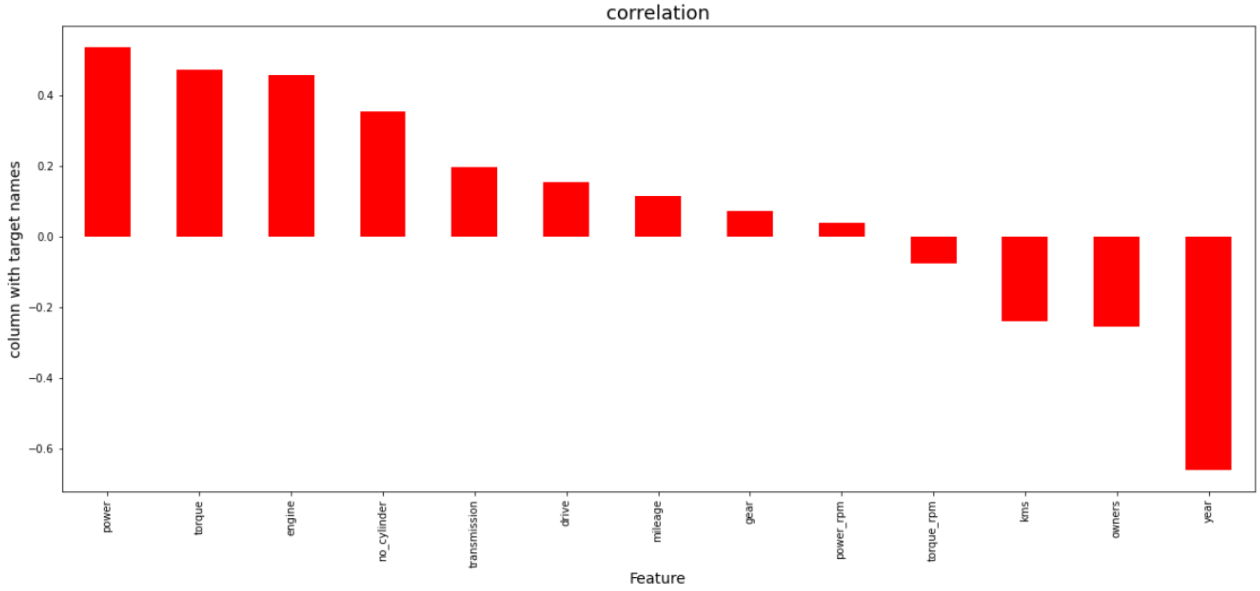
Visualize Boxplot of every column having datapoints of continuous nature



Outliers are present in column kms, mileage, engine, power, torque & price, removed them using z-score & percentile method

Check For Correlation





Observation:

We can analyze that column gear, power_rpm & torque_rpm have least correlation, but we will not drop them for now and analyze further

VIF to detect multicollinearity and remove them

	vif	Features
0	8.668412	year
1	1.573318	kms
2	9.924319	owners
3	9.629034	transmission
4	2.414966	mileage
5	6.855390	engine
6	5.961532	power
7	1.078978	power_rpm
8	32.520693	gear
9	24.194473	drive
10	4.768760	torque
11	2.710884	torque_rpm
12	47.498484	no_cylinder

Observation:

Drop column no_cylinder, drive, gear

Find the best feature column:

	Specs	Score
2	kms	21984814.0
8	power_rpm	4657543.0
13	torque_rpm	406294.0
6	engine	71001.0
12	torque	43585.0
7	power	9295.0
0	year	5049.0
5	mileage	723.0
3	owners	205.0
4	transmission	200.0
10	gear	166.0
11	drive	116.0
14	no_cylinder	108.0
1	fuel	73.0
9	seats	47.0

Drop column seats & fuel as they have least significance regarding target variable.

Model Building

Perform Feature Scaling

Before we start model building, we need to perform feature scaling on all columns, to avoid biasing of data.

Also check for skewness in data and remove it.

```
kms           0.321992
mileage       0.069566
engine       -0.135959
power         0.380536
power_rpm     1.141089
torque        0.505762
torque_rpm   -0.650511
price         0.444821
dtype: float64
```

```
kms           0.321992
mileage       0.069566
engine       -0.135959
power         0.380536
power_rpm     0.966809
torque        0.505762
torque_rpm   -0.650511
price         0.444821
dtype: float64
```

Model Building

As we know, this is a regression problem we need to build a model using regression algorithm models.

First, we need to write a function which can find us best random state for train test split.

Then we shall iterate through all the models supporting regression algorithms to find the best models.

From above we get to know that the top 3 models are:

- LGMBRegressor
- XGBRegressor
- RandomForest Regressor

Fine tune all these models and find their best parameters to use.

Next, find the best random state for train test split.

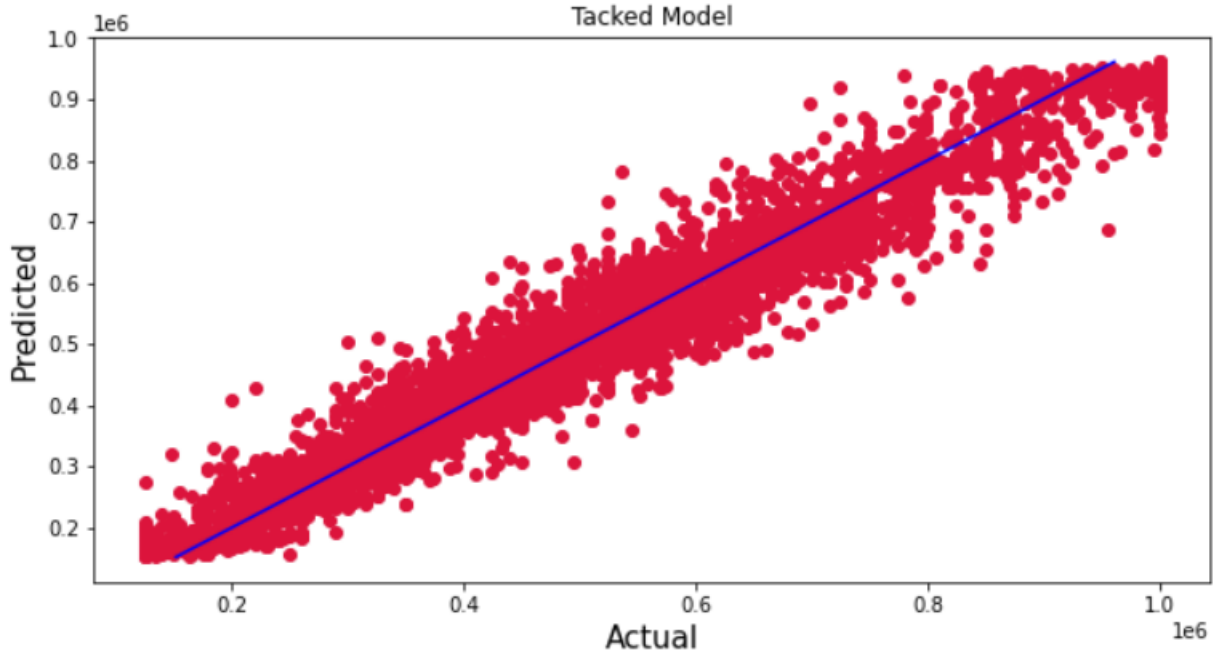
we obtain test accuracy of more than 88%.

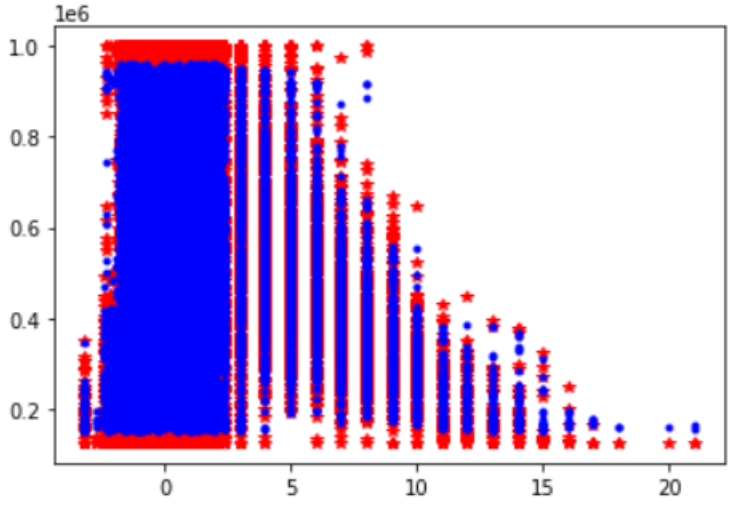
CV score of this model is more than 96%.

To analyze our model, we shall find the difference between actual and predicted value.

Visualize model accuracy using predicted values

```
plt.figure(figsize=(10,5))
plt.scatter(y, y_pred, c='crimson')
p1 = max(max(y_pred), max(y_pred))
p2 = min(min(y_pred), min(y_pred))
plt.plot([p1, p2], [p1, p2], 'b-')
plt.xlabel('Actual', fontsize=15)
plt.ylabel('Predicted', fontsize=15)
plt.title("Tacked Model")
plt.show()
```





the above graph shows the closeness of actual and predicted values

Interpretation of the Results

The dataset was very challenging to handle it had 18 features.

Firstly, the datasets were having any null values.

But there was huge number of similar entries in columns, so we must be careful while going through the statistical analysis of the datasets.

And proper plotting for proper type of features will help us to get better insight on the data. I found maximum numerical columns in the dataset, so I have chosen bar plot to see the relation between target and features.

I notice a huge amount of outliers and skewness in the data so we have choose proper methods to deal with the outliers and skewness. If we ignore these outliers and skewness, we may end up with a bad model which has less accuracy.

Then scaling dataset has a good impact like it will help the model not to get biased. Since we have not removed outliers and skewness completely from the dataset, so we must choose Normalization.

We must use multiple models while building model using dataset as to get the best model out of it.

I Stacked all the top models and concluded it as the best model with 96% accuracy score. Also, I have improved the accuracy of the best model by running hyper parameter tunning.

At last, I have predicted whether the price of used car using saved model. It was good!! that I was able to get the predictions near to actual values.

Conclusion

Key Features and conclusion of the study

In this project we have tried to predict the price of used car. The best accuracy score was achieved by fine-tuned the stacked model built by combining all three top models.

LEARNING OUTCOMES OF THE STUDY IN RESPECT OF DATA SCIENCE

Through different powerful tools of visualization, we were able to analyze and interpret different hidden insights about the data.

Through data cleaning we were able to remove unnecessary columns and outliers from our dataset due to which our model would have suffered from overfitting or underfitting.

The data was improper scaled, so we scaled it to a single scale using sklearns's package StandardScaler.

The columns were skewed due to presence of outliers which we handled through percentile technique.

Model was then built having accuracy more than 96% using train dataset.

LIMITATIONS OF THIS WORK AND SCOPE FOR FUTURE WORK

Due to the presence of lot of outliers, we are unsure whether the model is going to perform well to a completely new dataset.

The scope for future work is to collect as many data as we can so that the model can be built more efficiently.