PROJECT REPORT ON

CAR PRICE PREDICTION

Submitted By

Aamina Ruvaida

ACKNOWLEDGMENT

I sincerely thank to “Flip Robo Technologies” team who let me to work on Used car price prediction project. Also, I gratefully thanked to Data Trained team who helped and guided me in the right direction. The car price prediction is challenging and interesting task to perform. While working on project I understand the global car market when research on the car Dekho website.

Reference:

CarDekho: New Cars, Car Prices, Buy & Sell Used Cars in India

Benefits of used cars and why they are so expensive? (theodysseyonline.com)

- ## 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.

- **Business goal:** The main aim of this project is to predict the price of used car based on various features. Machine Learning is a field of technology developing with immense abilities and applications in automating tasks. So, we will deploy an ML model for car selling price prediction and analysis. This kind of system becomes handy for many people. This model will provide the approximate selling price for the car based on different features like fuel type, transmission, price, weight, running in kms, engine displacement, milage etc and this model will help the client to understand the price of used cars.

- # Conceptual Background of the Domain Problem

Car Price Prediction is really an interesting machine learning problem as there are many factors that influence the price of a car in the second-hand market. In many developed countries, it is common to lease a car rather than buying it outright. A lease is a binding contract between a buyer and a seller (or a third party – usually a bank, insurance firm or other financial institutions) in which the buyer must pay fixed instalments for a pre-defined number of months/years to the seller/financer. After the lease period is over, the buyer has the possibility to buy the car at its residual value, i.e., its expected resale value. Thus, it is of commercial interest to seller/financers to be able to predict the salvage value (residual value) of cars with accuracy. If the residual value is under-estimated by the seller/financer at the beginning, the instalments will be higher for the clients who will certainly then opt for another seller/financer. If the residual value is over-estimated, the instalments will be lower for the clients but then the seller/financer may have much difficulty at selling these high-priced used cars at this over-estimated residual value. Thus, we can see that estimating the price of used cars is of very high commercial importance as well.

- **Review of Literature**

The second-hand car market has continued to expand even as the reduction in the market of new cars. According to the recent report on India's pre-owned car market by Indian Blue Book, nearly 4 million used cars were purchased and sold in 2018-19. The second-hand car market has created the business for both buyers and sellers. Most of the people prefer to buy the used cars because of the affordable price and they can resell that again after some years of usage which may get some profit. The price of used cars depends on many factors like fuel type, colour, model, mileage, transmission, engine, number of seats etc., The used cars price in the market will keep on changing. Thus, the evaluation model to predict the price of the used cars is required.

- **Motivation for the Problem Undertaken**

There are lots of website which shows the trending second hand cars. With the clear picture and features describe in the website. It is interesting to work on car price prediction, it will motivate us know the current trend of the global car market. It makes us to compare with car model, it motivates us to give the trader a better prediction by analysis various factor. So, that car trader can go ahead and predict the car price. The profile will be high and they can compete the market. The model developed in this study may help online web services that tells a used car's market values.

Analytical Problem Framing

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

First, I have scrapped the data in the cardekho website. Then saved the data in excel. Imported the necessary libraries and the scrapped data in jupyter notebook. In this problem, we have to predict the used car price. So, our target variables are car price, it is continuous. It is clearly shows that it is a Regression problem. I have checked the null value, data description, data info. I have converted the categorical data to numerical data using the Label Encoder. I have checked the Outliers and removed the outliers using the zscore method and checked the skewness and correlation of data. Then I have visualized the data using the distribution plot, stripplot, violin plot, scatter plot, box plot, kde plot to understand what data is trying to say. Then I scaled the data, checked the variation

inflation factor. There is no multi-collinearity exist in the data. I have used various machine learning model to make car price prediction. The I have done the cross-validation score of each model. Then done the hyperparameter tuning to increase the accuracy of the final model. Finally saved the model using the pickle and predict the car price.

- **Data Sources and their formats**

The data was collected from cardekho website. The data was scrapped using the Selenium. After scrapped the data, it is saved in excel format. The dataset contains integer, float and object.

- **Data Pre-processing Done**

At first, I have scrapped the data from cardekho website. Then imported the data. I have checked its unique value, data info, data description.

Checked the null values, there is no null values present in the data. Removed the unnamed: 0 column from data. Extracted the manufacturing year from the car model. From that I have extracted the car age. Checked the Outliers, in few columns' outliers are present. Removed the Outliers using the zscore method. Checked the skewed, variation inflation factor.

In data both categorical and numerical data present. Converted the categorical data to numerical data using the Label Encoder method.

- **Data Inputs- Logic- Output Relationships**

Using the Visualization Distribution plot, we can see the skewness present in data, then we can remove the skewness using the appropriate method.

Using the stripplot, we can understand the relationship between the target and Features variables.

I have used various distribution plot, after analysis we came to know that there is a linear relationship between the feature and target variable.

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

The **hardware** used in the project is

- Processor: core i3
- RAM : 12 GB
- SSD — 250GB or above

Software used in the project is

Anaconda Jupyter Notebook

Libraries used in the project is

```
# Importing Libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
```

```
# Printing all the columns and Row names

pd.set_option('display.max_columns',None)
pd.set_option('display.max_rows',None)
```

- **import pandas as pd**: **pandas** is a popular Python-based data analysis toolkit which can be imported using `import pandas as pd`. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a numpy matrix array. This makes pandas a trusted ally in data science and machine learning.
- **import numpy as np**: NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

- **import seaborn as sns:** Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.
- **Import matplotlib.pyplot as plt:** matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
from sklearn.linear_model import SGDRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import GradientBoostingRegressor
from xgboost import XGBRegressor
from sklearn.ensemble import BaggingRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.metrics import r2_score
```

The above is the machine learning model used in the project.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

We have scrapped the data from the website, it is not a clean dataset. I have done the data cleaning, data pre-processing. Removed the outliers present in the dataset using the zscore method. Removed the skewness using the yeo-johnson method. Converted the categorical data to numerical data using the Label Encoder. Checked the correlation of data and variation inflation factor. There is multicollinearity exist in the data. Removed the unnecessary column from the dataset. Then we have finally built the machine learning model to predict the car price prediction,

- ## Testing of Identified Approaches (Algorithms)

Since car price was my target and it was a continuous column with improper format which has to be changed to continuous float datatype column, so this particular problem was Regression problem. And I have used all Regression algorithms to build my model. By looking into the difference of r2 score and cross validation score I found DecisionTreeRegressor as a best model with least difference. Also, to get the best model we have to run through multiple models and to avoid the confusion of overfitting we have go through cross validation. Below is the list of Regression algorithms I have used in my project.

- Random Forest Regressor
- Decision Tree Regressor
- KNeighbors Regressor
- Gradient Boosting Regressor
- XGB Regressor
- SGD Regressor
- Bagging Regressor

- ## Key Metrics for success in solving problem under consideration

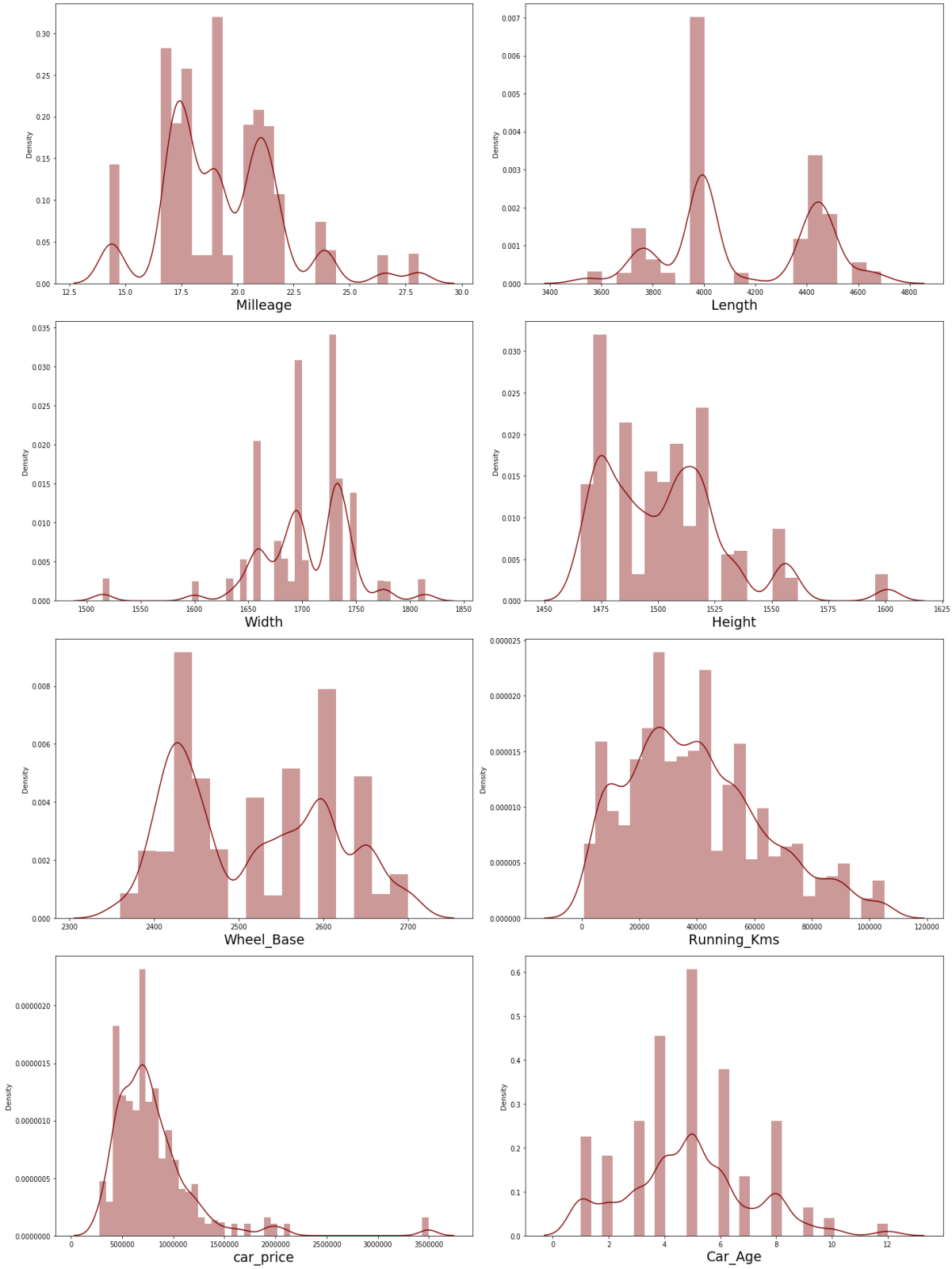
I have used mean squared error to measure the amount of error in statistical model. It assesses the average squared difference between the actual and predicted value.

I have used the mean absolute error to measure the error between the paired observation.

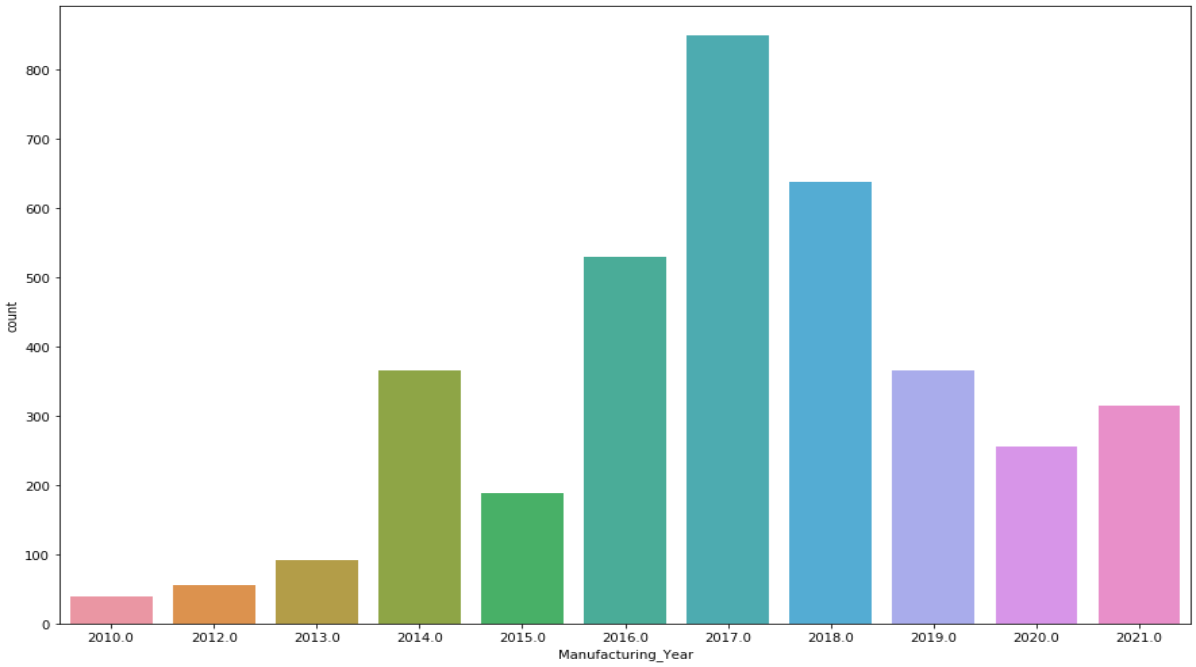
Used the r2 score for each model to tell us the accuracy of each model. So, that we can finalize which model is performing good.

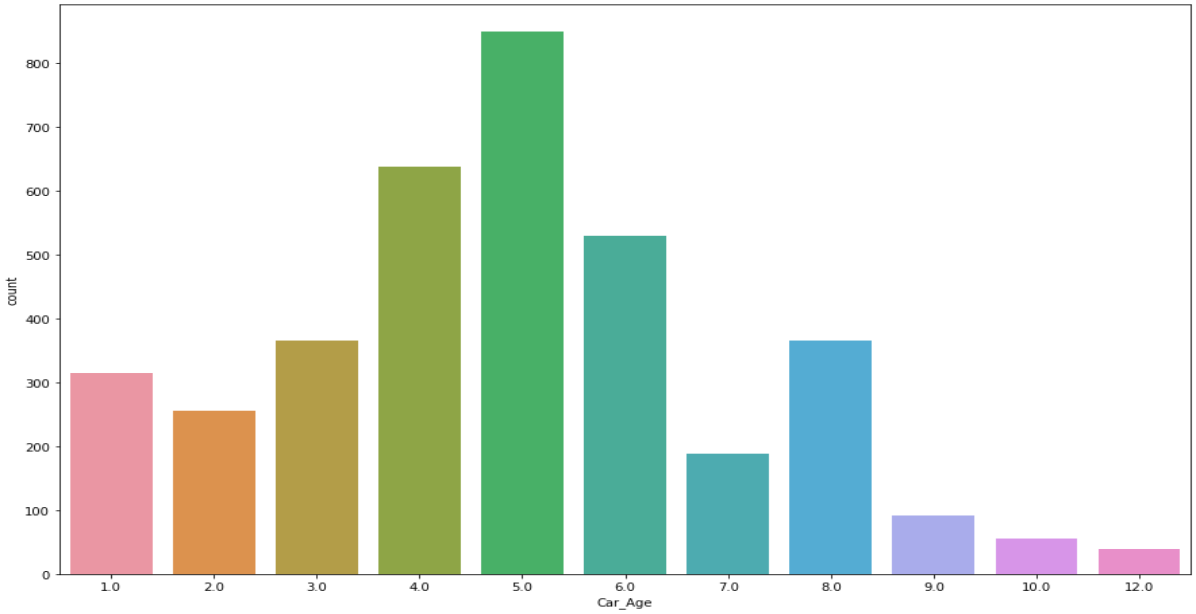
- ## Visualizations

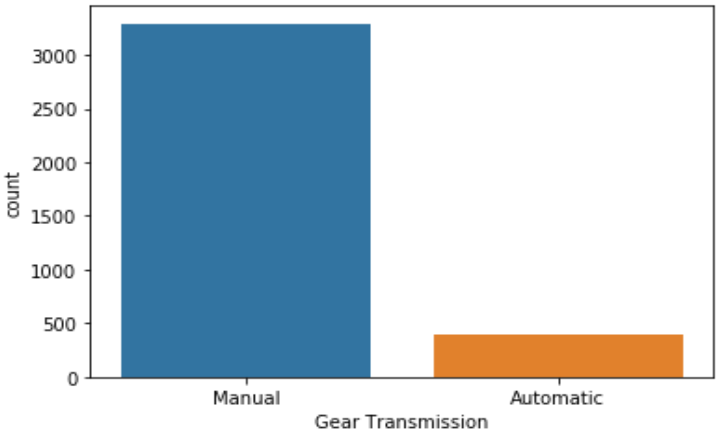
I have used various visualization plot like distribution plot, regression plot, stripplot, box plot, violin plot, scatterplot, kde plot, bar plot.

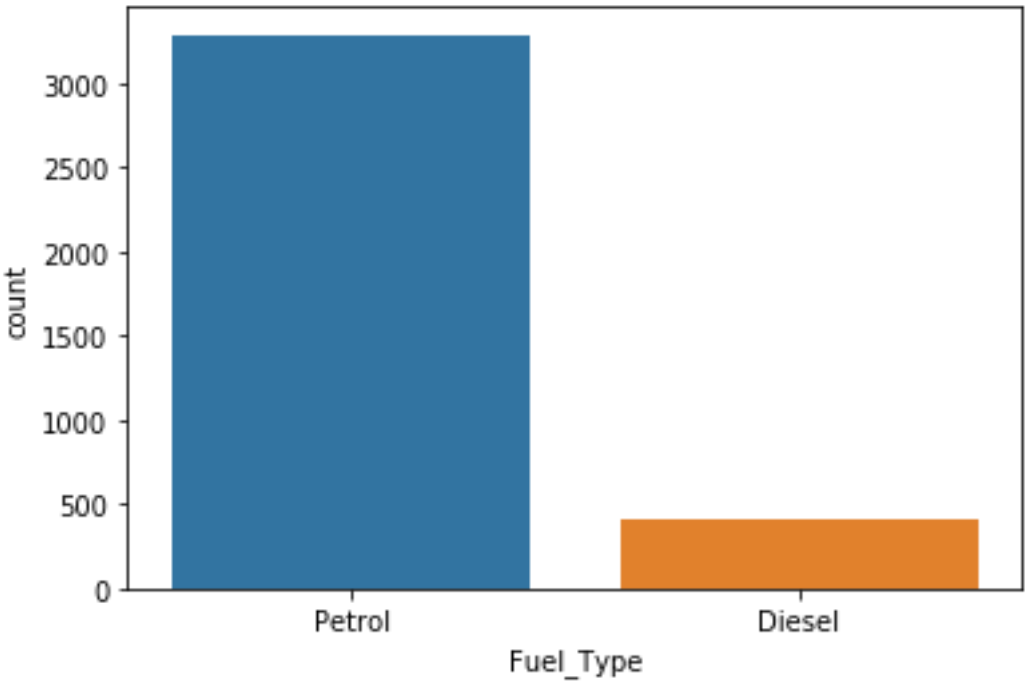


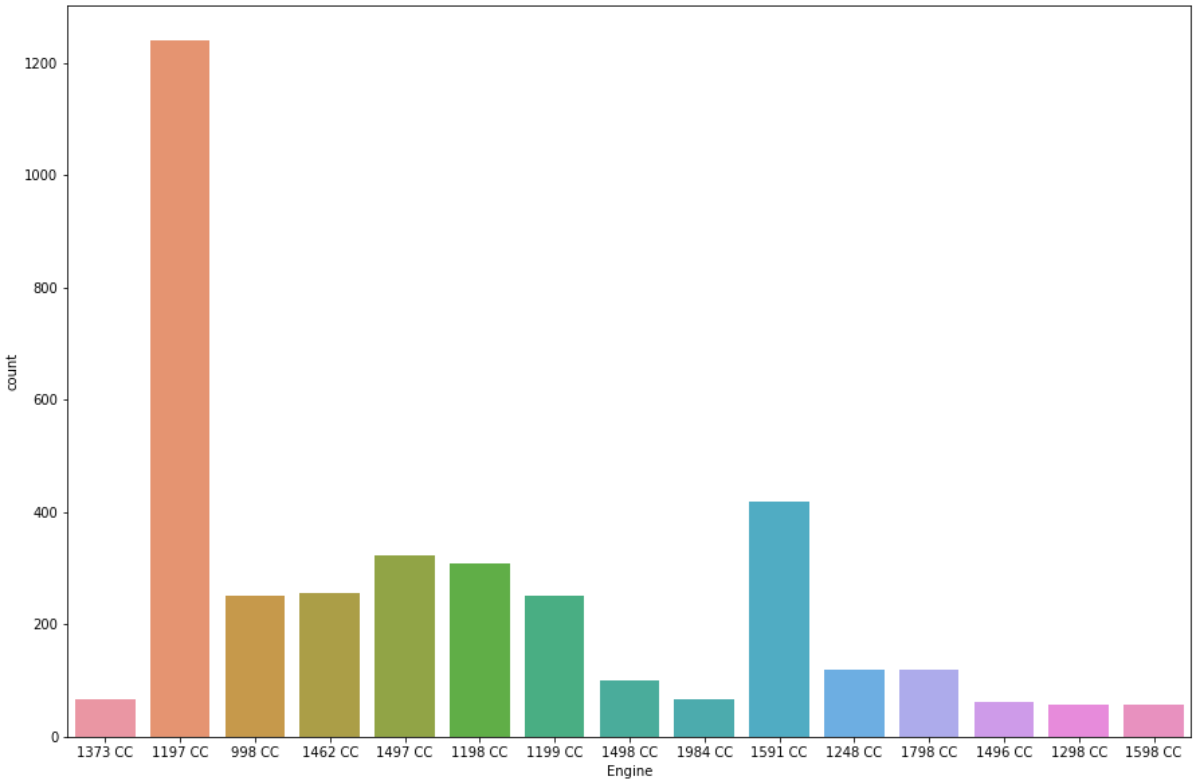
We see that in most of the column skewness is present. We will remove the skewness using the yeo-johnson method.

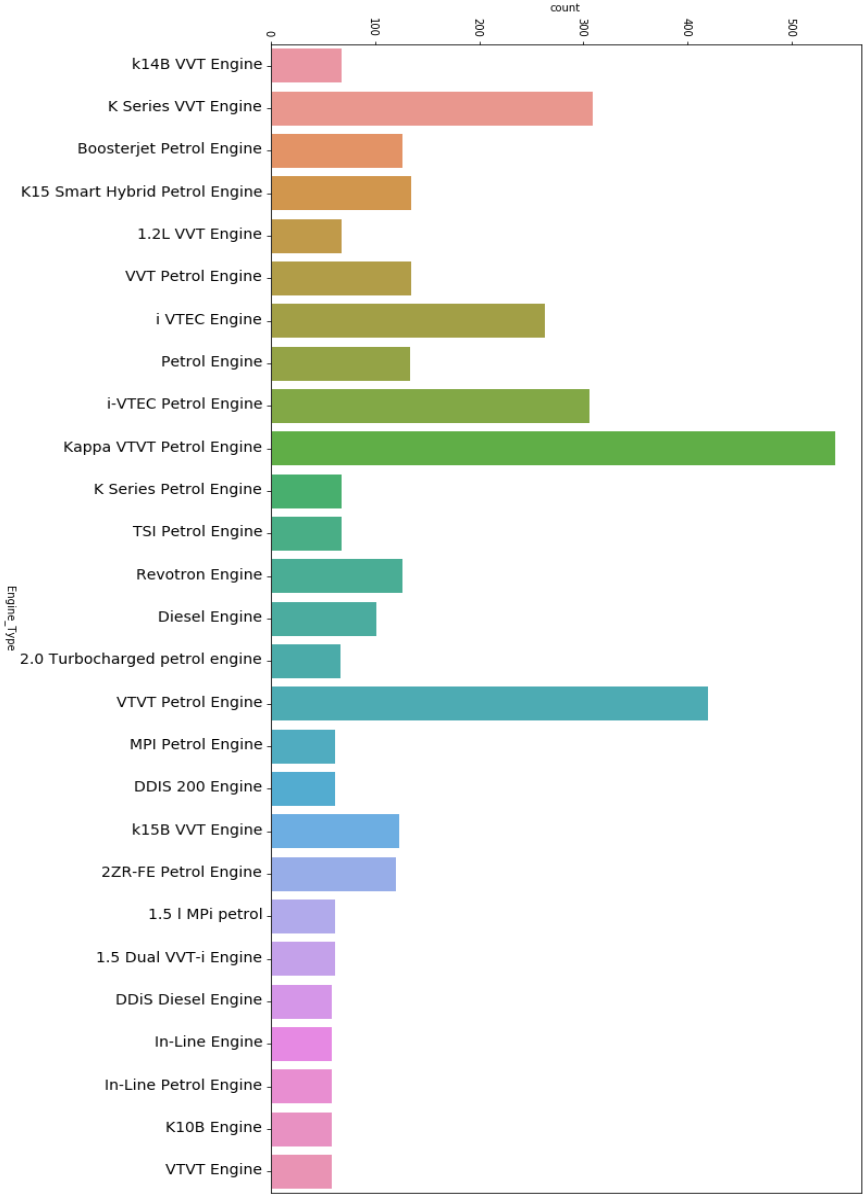

count
0
100
200
300
400
500
600
700
800
2010.0
2012.0
2013.0
2014.0
2015.0
2016.0
2017.0
2018.0
2019.0
2020.0
2021.0
Manufacturing_Year

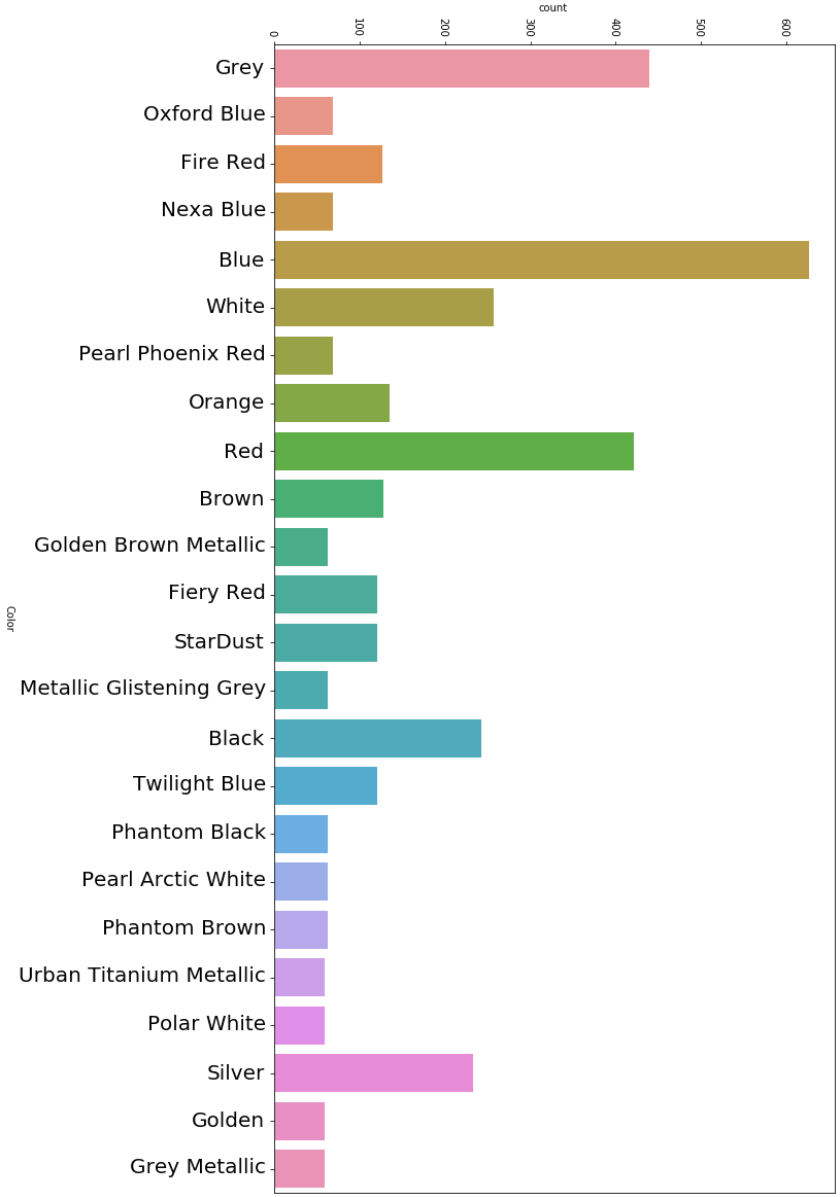

count
0
100
200
300
400
500
600
700
800
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
12.0
Car_Age

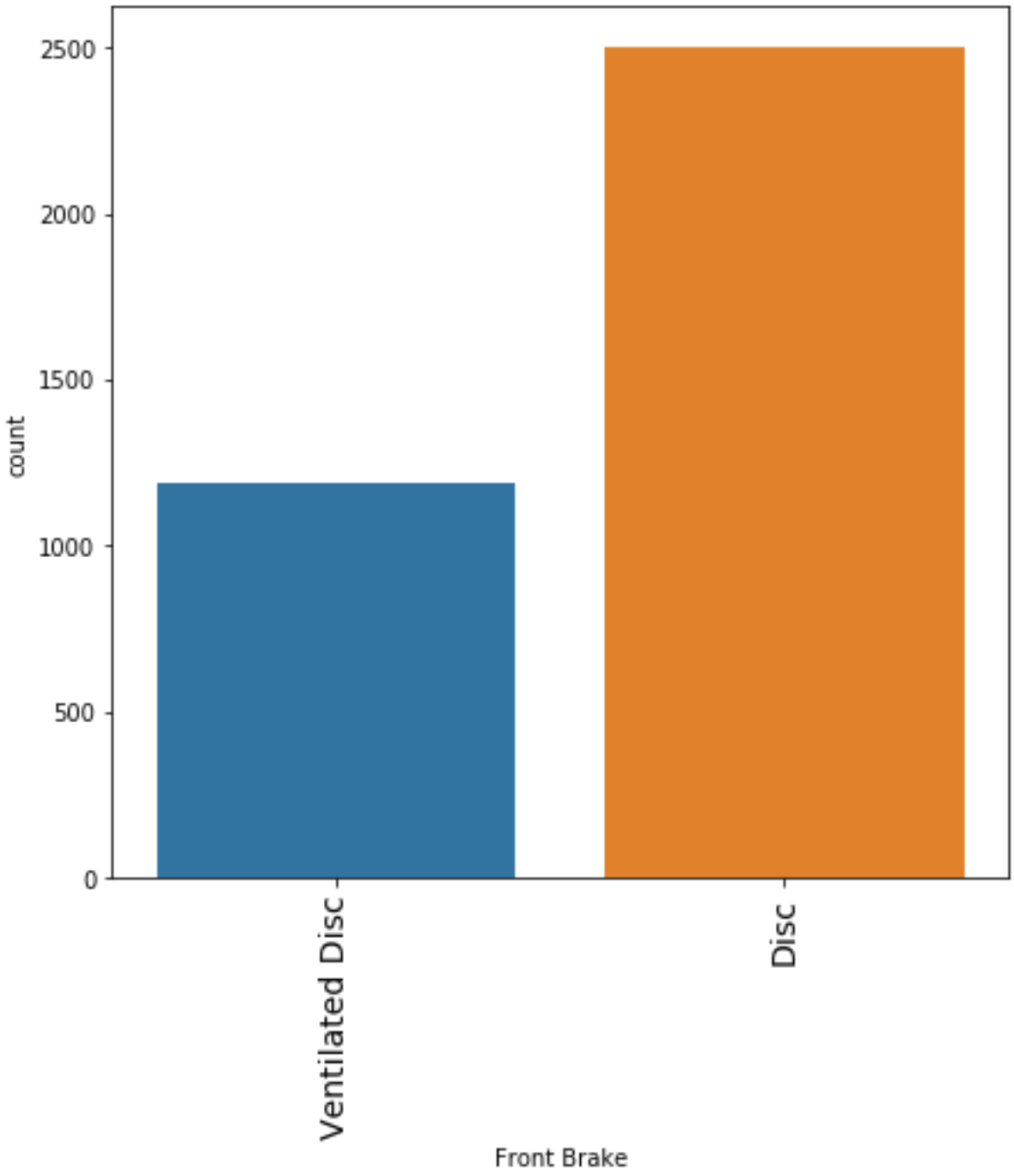

count
0
500
1000
1500
2000
2500
3000
Manual
Automatic
Gear Transmission


3000
2500
2000
1500
1000
500
0
count
Petrol
Diesel
Fuel_Type


1200
1000
800
600
400
200
0
count
1373 CC
1197 CC
998 CC
1462 CC
1497 CC
1198 CC
1199 CC
1498 CC
1984 CC
1591 CC
1248 CC
1798 CC
1496 CC
1298 CC
1598 CC
Engine

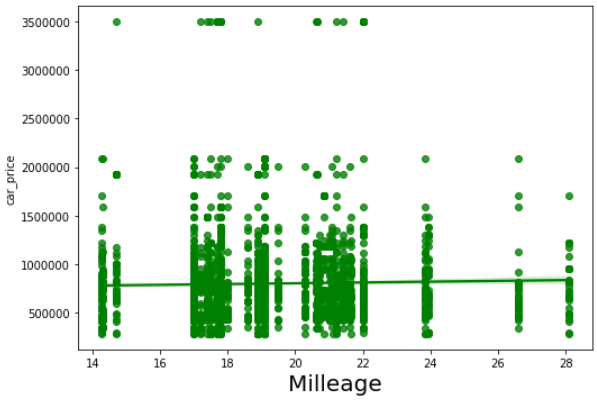

count
0
100
200
300
400
500
Engine_Type
k14B VVT Engine
K Series VVT Engine
Boosterjet Petrol Engine
K15 Smart Hybrid Petrol Engine
1.2L VVT Engine
VVT Petrol Engine
i VTEC Engine
Petrol Engine
i-VTEC Petrol Engine
Kappa VTVT Petrol Engine
K Series Petrol Engine
TSI Petrol Engine
Revotron Engine
Diesel Engine
2.0 Turbocharged petrol engine
VTVT Petrol Engine
MPI Petrol Engine
DDIS 200 Engine
k15B VVT Engine
2ZR-FE Petrol Engine
1.5 l MPi petrol
1.5 Dual VVT-i Engine
DDiS Diesel Engine
In-Line Engine
In-Line Petrol Engine
K10B Engine
VTVT Engine

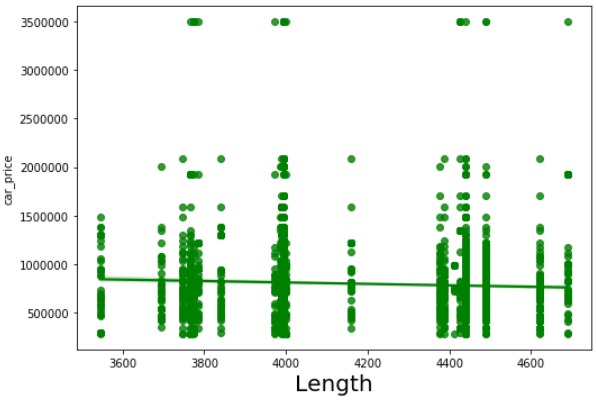

count
0
100
200
300
400
500
600
Color
Grey
Oxford Blue
Fire Red
Nexa Blue
Blue
White
Pearl Phoenix Red
Orange
Red
Brown
Golden Brown Metallic
Fiery Red
StarDust
Metallic Glistening Grey
Black
Twilight Blue
Phantom Black
Pearl Arctic White
Phantom Brown
Urban Titanium Metallic
Polar White
Silver
Golden
Grey Metallic

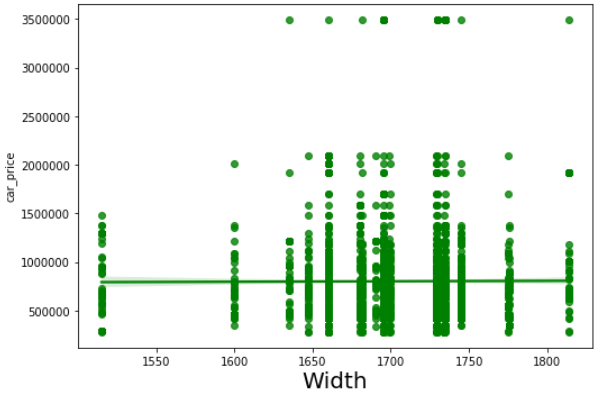


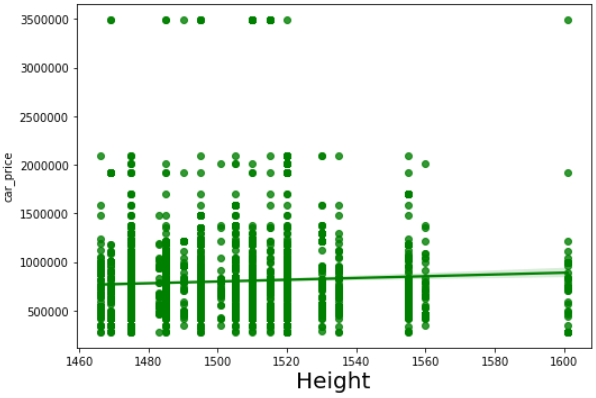
OBSERVATION:

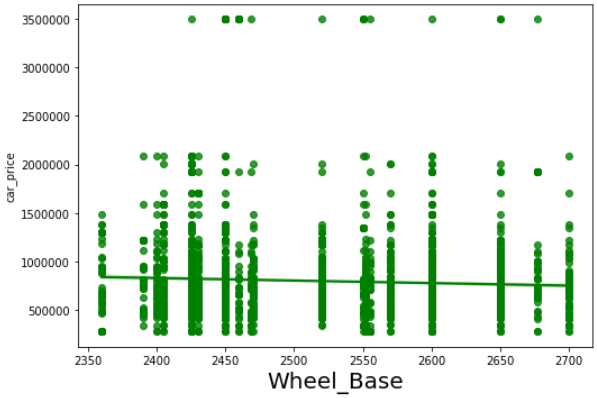
Majority of people using the Manual gear rather than Automatic. In fuel type majority of people using petrol. Blue color car is most prefavorably buying than other colors. In Front brake most of the car is having Disc front brake. In Engine 1199 CC Engine count is high.

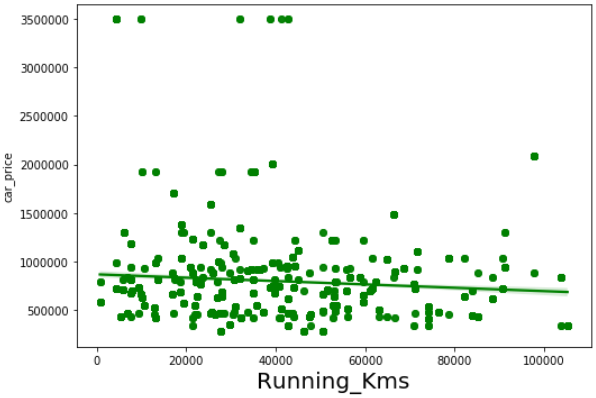


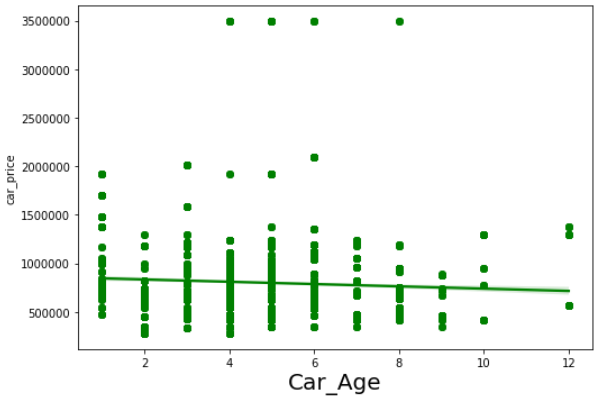






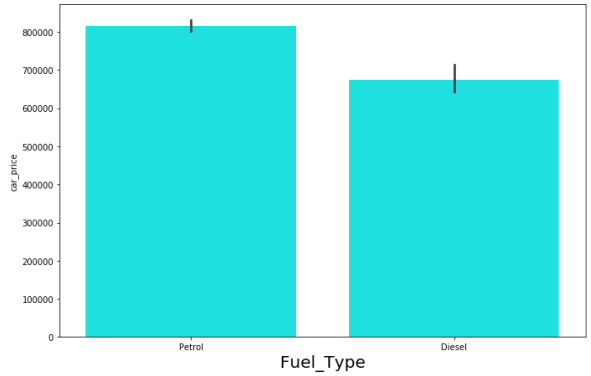




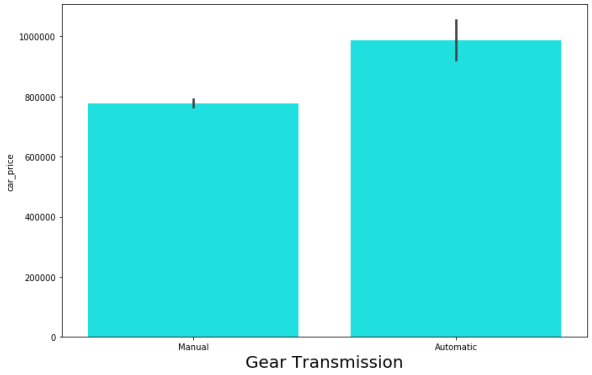


OBSERVATION:

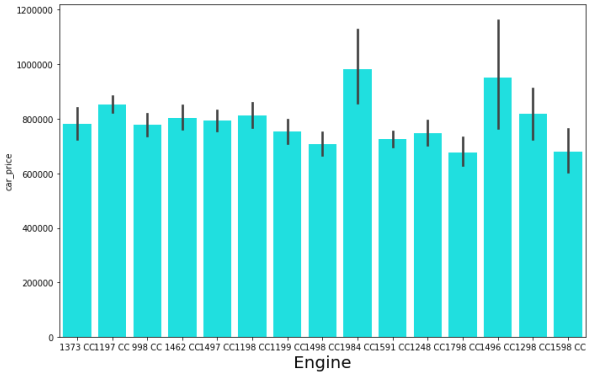
- ✓ Maximum cars are petrol driven and also diesel driven.
- ✓ Maximum cars are with Manual gear transmission.
- ✓ Disc front brake cars are more in number followed by Ventilated Disc.



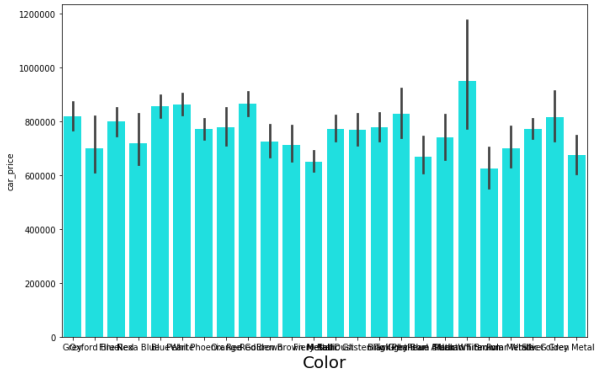
Fuel_Type



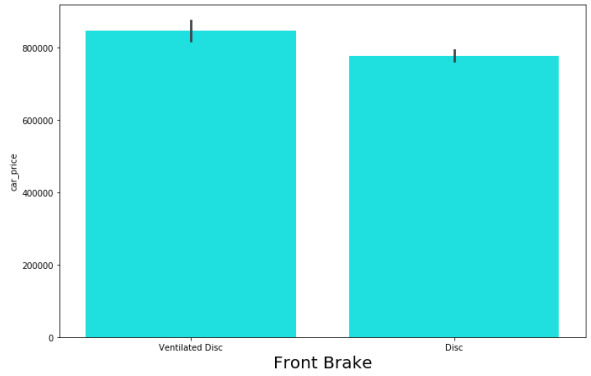
Gear Transmission



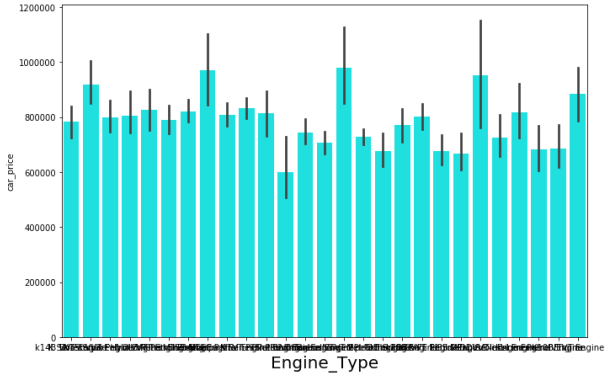
Engine



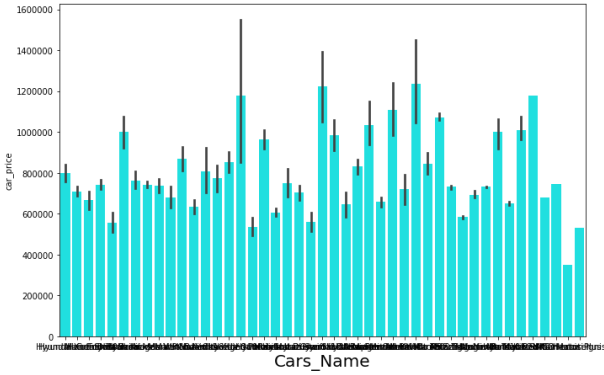
Color



Front Brake



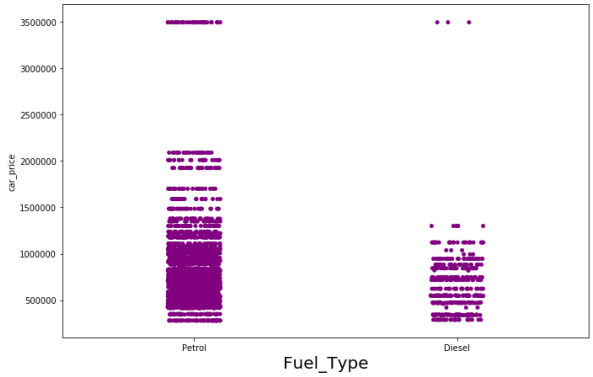
Engine_Type

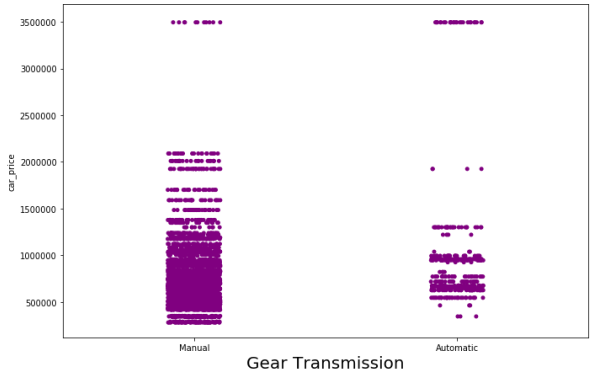


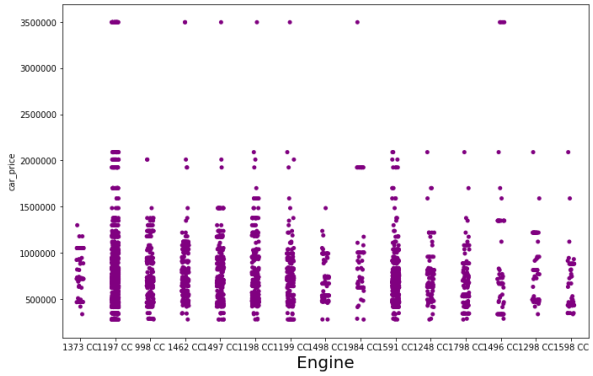
Cars_Name

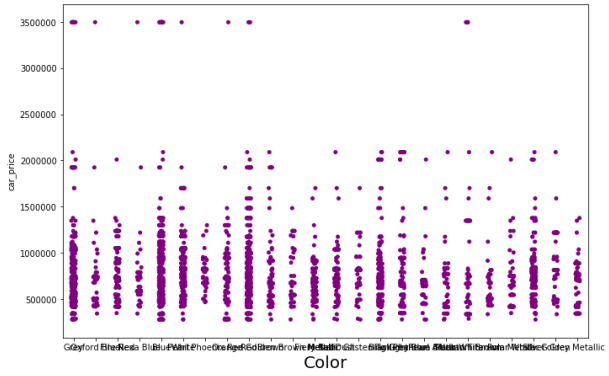
OBSERVATION:

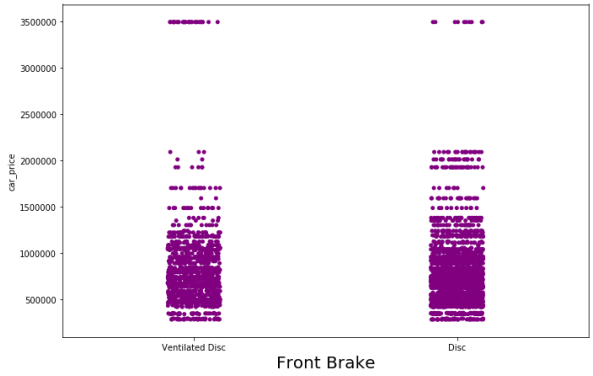
- ✓ For Diesel and Electric cars the price is high compared to Petrol, LPG and CNG.
- ✓ Cars with automatic gear are costlier than manual gear cars.
- ✓ Cars with Dual cast break discs front break are costlier compared to other cars.

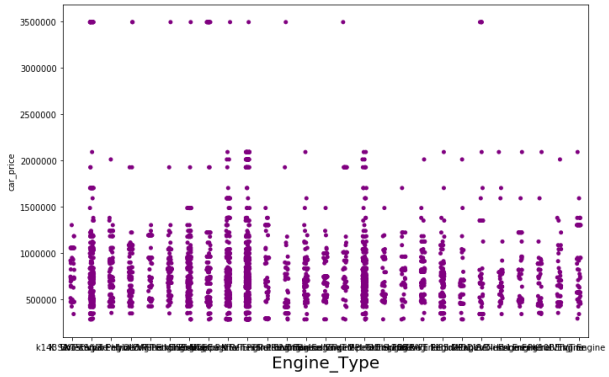


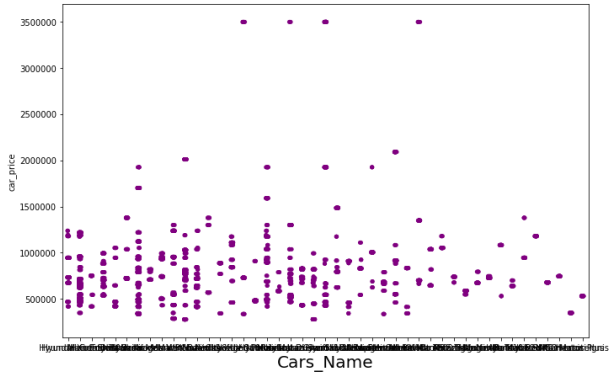








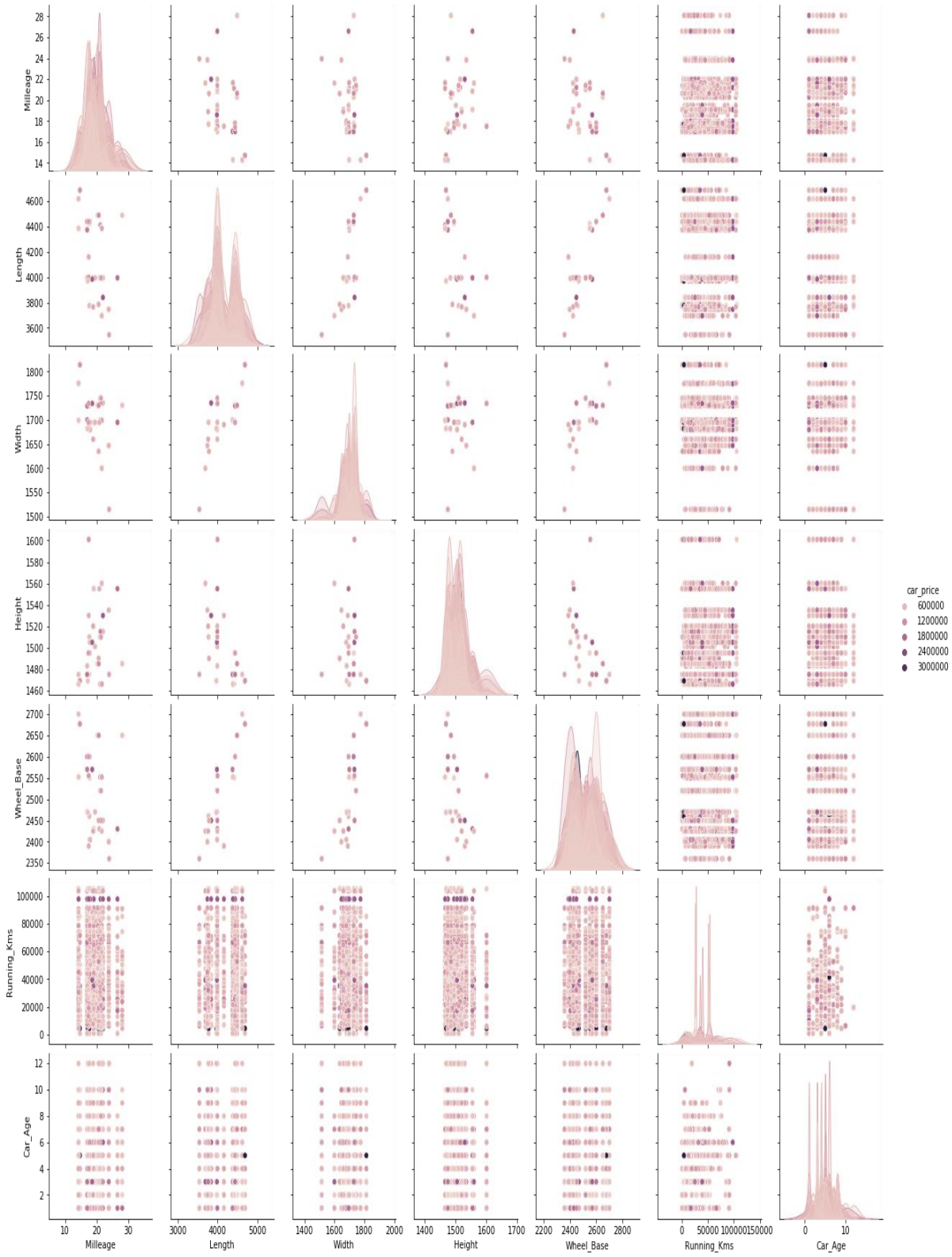




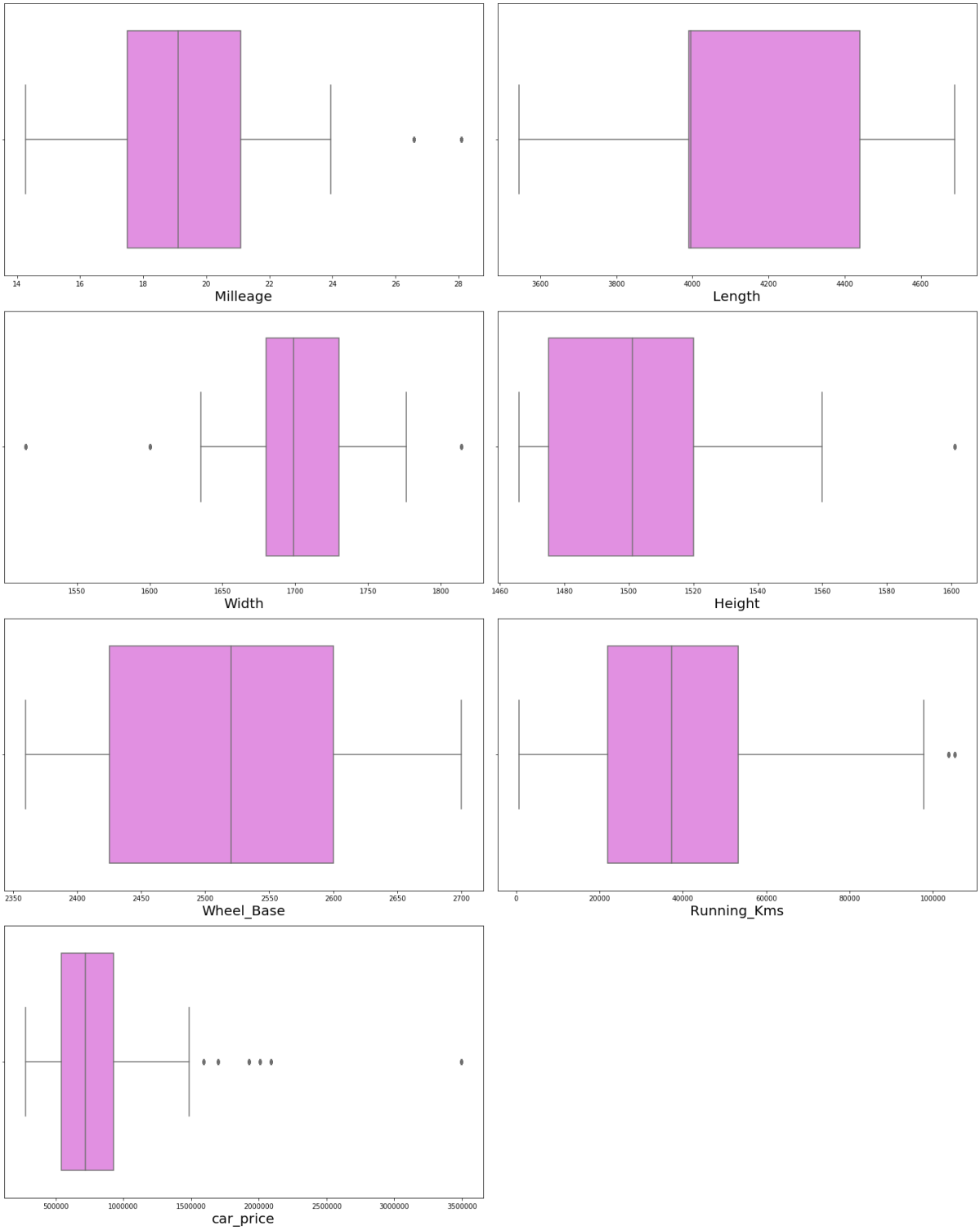
OBSERVATION:

- ✓ As length is increasing car price is also increasing.
- ✓ Weight also has linear relationship with car price.
- ✓ As top_speed is increasing car price is also increasing.
- ✓ Cars with 5 and 7 seats are having highest price.
- ✓ As the age of the car increases the car price decreases.

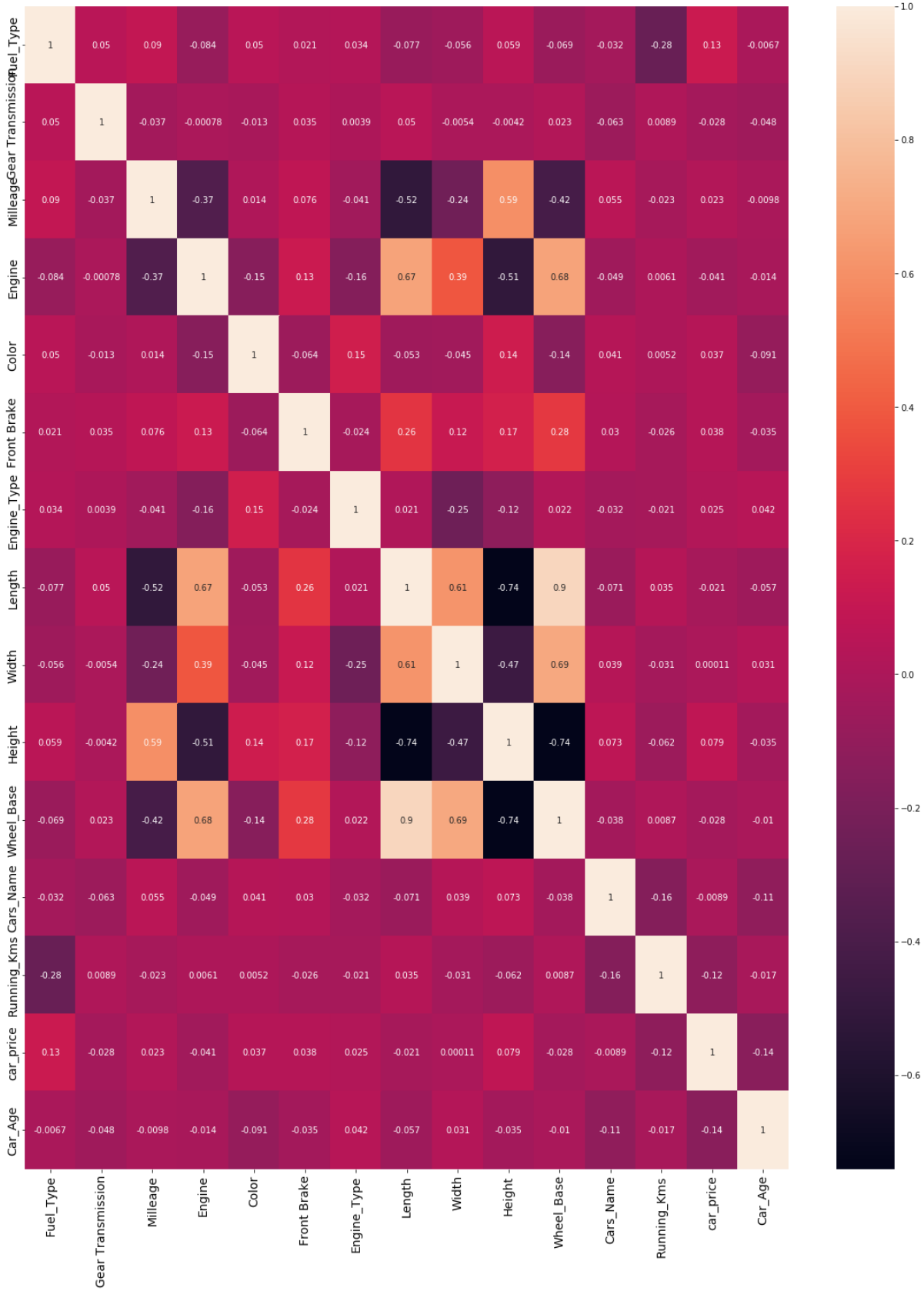
.

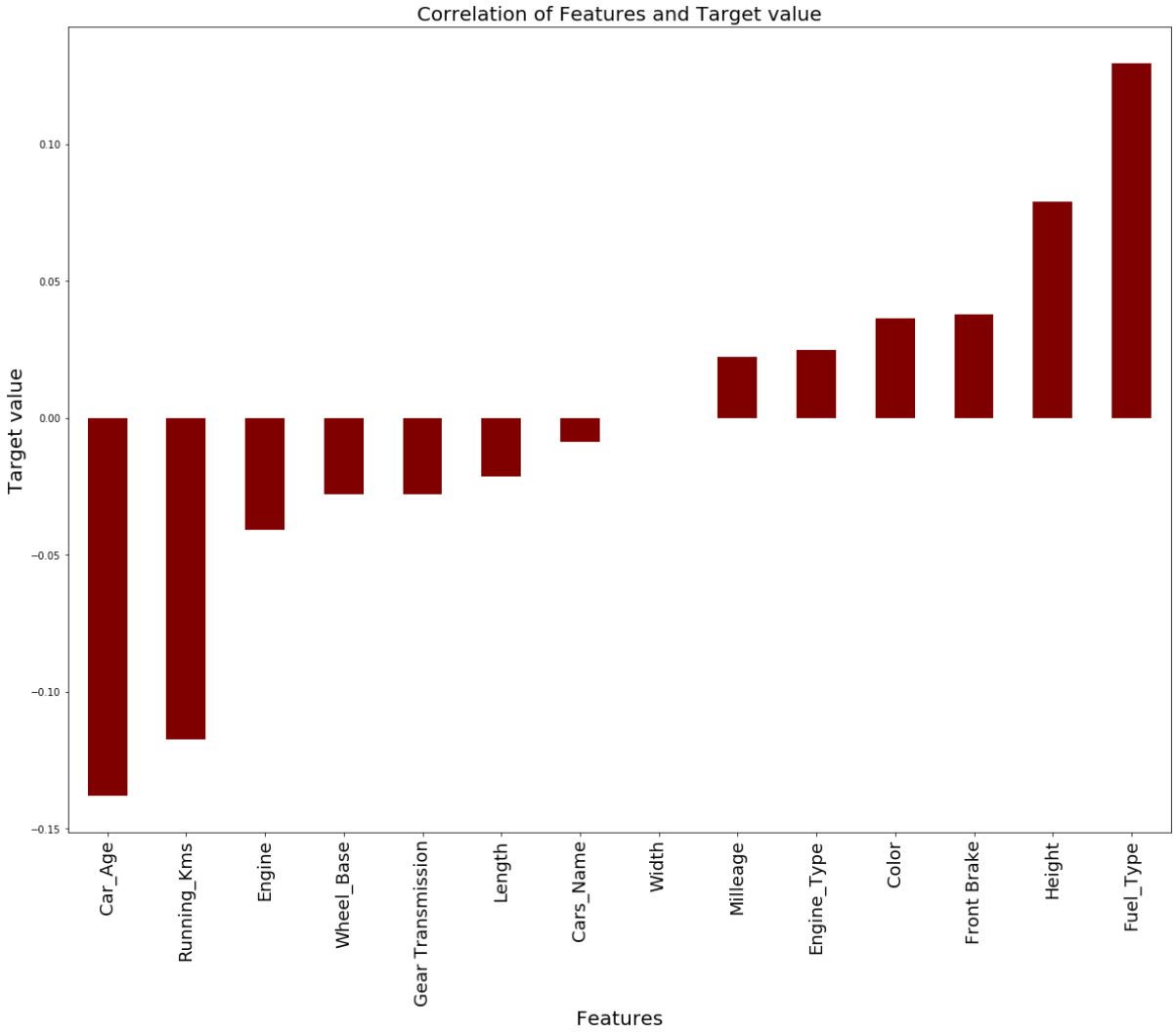


The above is the pair plot, it shows the relations with features and target variables.



In few columns, there is a outliers is present. We have removed the outliers using the zscore method.


Fuel_Type 1 0.05 0.09 -0.084 0.05 0.021 0.034 -0.077 -0.056 0.059 -0.069 -0.032 -0.28 0.13 -0.0067
Gear Transmission 0.05 1 -0.037 -0.00078 -0.013 0.035 0.0039 0.05 -0.0054 -0.0042 0.023 -0.063 0.0089 -0.028 -0.048
Milleage 0.09 -0.037 1 -0.37 0.014 0.076 -0.041 -0.52 -0.24 0.59 -0.42 0.055 -0.023 0.023 -0.0098
Engine -0.084 -0.00078 -0.37 1 -0.15 0.13 -0.16 0.67 0.39 -0.51 0.68 -0.049 0.0061 -0.041 -0.014
Color 0.05 -0.013 0.014 -0.15 1 -0.064 0.15 -0.053 -0.045 0.14 -0.14 0.041 0.0052 0.037 -0.091
Front Brake 0.021 0.035 0.076 0.13 -0.064 1 -0.024 0.26 0.12 0.17 0.28 0.03 -0.026 0.038 -0.035
Engine_Type 0.034 0.0039 -0.041 -0.16 0.15 -0.024 1 0.021 -0.25 -0.12 0.022 -0.032 -0.021 0.025 0.042
Length -0.077 0.05 -0.52 0.67 -0.053 0.26 0.021 1 0.61 -0.74 0.9 -0.071 0.035 -0.021 -0.057
Width -0.056 -0.0054 -0.24 0.39 -0.045 0.12 -0.25 0.61 1 -0.47 0.69 0.039 -0.031 0.00011 0.031
Height 0.059 -0.0042 0.59 -0.51 0.14 0.17 -0.12 -0.74 -0.47 1 -0.74 0.073 -0.062 0.079 -0.035
Wheel_Base -0.069 0.023 -0.42 0.68 -0.14 0.28 0.022 0.9 0.69 -0.74 1 -0.038 0.0087 -0.028 -0.01
Cars_Name -0.032 -0.063 0.055 -0.049 0.041 0.03 -0.032 -0.071 0.039 0.073 -0.038 1 -0.16 -0.0089 -0.11
Running_Kms -0.28 0.0089 -0.023 0.0061 0.0052 -0.026 -0.021 0.035 -0.031 -0.062 0.0087 -0.16 1 -0.12 -0.017
car_price 0.13 -0.028 0.023 -0.041 0.037 0.038 0.025 -0.021 0.00011 0.079 -0.028 -0.0089 -0.12 1 -0.14
Car_Age -0.0067 -0.048 -0.0098 -0.014 -0.091 -0.035 0.042 -0.057 0.031 -0.035 -0.01 -0.11 -0.017 -0.14 1
Fuel_Type
Gear Transmission
Milleage
Engine
Color
Front Brake
Engine_Type
Length
Width
Height
Wheel_Base
Cars_Name
Running_Kms
car_price
Car_Age
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6



This is correlation of Features and target variable. Car Age, Running kms, Engine, Wheel Base, Gear Transmission are negatively correlated.

- **Run and Evaluate selected models**

Model Building:

Random Forest Regressor:

```
rfr=RandomForestRegressor()
rfr.fit(x_train,y_train)
rfr_pred=rfr.predict(x_test)
print("Predicted value:\n",rfr_pred)
```

```
Predicted value:
 [1241380.90909091 1434564.64701965  634000.         ...  468000.
  680020.          635777.85714286]
```

```
print("R2 Score is:",r2_score(y_test,rfr_pred)*100)
print("Mean Squared Error value is:",mean_squared_error(y_test,rfr_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,rfr_pred))
```

```
R2 Score is: 92.5830831675461
Mean Squared Error value is: 7466378775.91398
Mean Absolute Error value is: 39589.22778808122
```

In Random Forest Regressor, gives a r2 score 92%.

Decision Tree Regressor:

```
dtr=DecisionTreeRegressor()
dtr.fit(x_train,y_train)
dtr_pred=dtr.predict(x_test)
print("Predicted value:\n",dtr_pred)
```

```
Predicted value:
 [1378000.         1557777.77777778  634000.         ...  468000.
  655000.          571000.         ]
```

```
print("R2 Score is:",r2_score(y_test,dtr_pred)*100)
print("Mean Squared Error value is:",mean_squared_error(y_test,dtr_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,dtr_pred))
```

```
R2 Score is: 86.62796697375714
Mean Squared Error value is: 13461208455.391056
Mean Absolute Error value is: 32209.619974663583
```

In Decision Tree Regressor, it gives a r2 score 87%

KNeighbors Regressor:

```
knn=KNeighborsRegressor()
knn.fit(x_train,y_train)
knn_pred=knn.predict(x_test)
print("Predicted value\n",knn_pred)
```

```
Predicted value
 [604400. 786000. 634000. ... 468000. 715000. 800200.]
```

```
print("R2 Score is:",r2_score(y_test,knn_pred)*100)
print("Mean Squared Error value is:",mean_squared_error(y_test,knn_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,knn_pred))
```

```
R2 Score is: 43.472681385053
Mean Squared Error value is: 56904288062.0155
Mean Absolute Error value is: 131605.03875968992
```

In KNeighbors Regressor gives a r2 score 43.4%

Gradient Boosting Regressor:

```
gbr=GradientBoostingRegressor()
gbr.fit(x_train,y_train)
gbr_pred=gbr.predict(x_test)
print("Predicted value:\n",gbr_pred)
```

```
Predicted value:
 [1209754.46115539 1050863.79471292  712577.81160378 ...  743921.97439198
  685178.15106721  707931.18386441]
```

```
print("R2 Score is:",r2_score(y_test,gbr_pred)*100)
print("Mean Squared Error value is:",mean_squared_error(y_test,gbr_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,gbr_pred))
```

```
R2 Score is: 65.19474785919441
Mean Squared Error value is: 35037361446.11287
Mean Absolute Error value is: 132600.1667176239
```

In Gradient Boosting Regressor it gives a r2 score 65%

XGB Regressor:

```
xgb=XGBRegressor()
xgb.fit(x_train,y_train)
```

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, enable_categorical=False,
             gamma=0, gpu_id=-1, importance_type=None,
             interaction_constraints='', learning_rate=0.300000012,
             max_delta_step=0, max_depth=6, min_child_weight=1, missing=nan,
             monotone_constraints='()', n_estimators=100, n_jobs=4,
             num_parallel_tree=1, predictor='auto', random_state=0, reg_alpha=0,
             reg_lambda=1, scale_pos_weight=1, subsample=1, tree_method='exact',
             validate_parameters=1, verbosity=None)
```

```
xgb_pred=xgb.predict(x_test)
print("Predicted value:\n",xgb_pred)
```

```
Predicted value:
 [1242898.6  1261912.2   633448.56 ...  470941.44  652090.6   559514.5 ]
```

```
print("R2 Score is:",r2_score(y_test,xgb_pred)*100)
print("Mean Squared Error value is:",mean_squared_error(y_test,xgb_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,xgb_pred))
```

```
R2 Score is: 91.16161215194505
Mean Squared Error value is: 8897329299.050596
Mean Absolute Error value is: 45306.255041787794
```

In XGB Regressor it gives a r2 score 91%

Bagging Regressor

```
br=BaggingRegressor()
br.fit(x_train,y_train)
```

```
BaggingRegressor()
```

```
br_pred=br.predict(x_test)
print("Predicted value:\n",br_pred)
```

```
Predicted value:
 [1351800.         1412066.66666667  634000.         ...  468000.
  693300.          596000.          ]
```

```
print("R2 Score is:",r2_score(y_test,br_pred)*100)
print("Mean Squared Error value is:",mean_squared_error(y_test,br_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,br_pred))
```

```
R2 Score is: 91.24131766443082
Mean Squared Error value is: 8817092246.35194
Mean Absolute Error value is: 41657.43073759788
```

In Bagging Regressor it gives a r2 score 91%

Cross-Validation score:

```
print("Cross Validation Score for Decision Tree Regressor:",cross_val_score(dtr,x,y,cv=10).mean()*100)
```

```
Cross Validation Score for Decision Tree Regressor: 65.57247022322025
```

```
print("Cross Validation Score for Random Forest Regressor:",cross_val_score(rfr,x,y,cv=10).mean()*100)
```

```
Cross Validation Score for Random Forest Regressor: 71.97776658117385
```

```
print("Cross Validation Score for Grandient Boosting Regressor:",cross_val_score(gbr,x,y,cv=10).mean()*100)
```

```
Cross Validation Score for Grandient Boosting Regressor: 59.082203760525765
```

```
print("Cross Validation Score for XGB Regressor Regressor:",cross_val_score(xgb,x,y,cv=10).mean()*100)
```

```
Cross Validation Score for XGB Regressor Regressor: 73.24844444137449
```

```
print("Cross Validation Score for Ada Boost Regressor:",cross_val_score(ada,x,y,cv=10).mean()*100)
```

```
Cross Validation Score for Ada Boost Regressor: 17.4140808942973
```

```
print("Cross Validation Score for KNeighbors Regressor:",cross_val_score(knn,x,y,cv=10).mean()*100)
```

```
Cross Validation Score for KNeighbors Regressor: 30.013527260167454
```

```
print("Cross Validation Score for Bagging Regressor:",cross_val_score(br,x,y,cv=10).mean()*100)
```

```
Cross Validation Score for Bagging Regressor: 71.58257467216062
```

After analysis of various model and cross validation score, Random Forest Regressor gives a good score. So, we will consider Random Forest Regressor model as final model.

Hyper Parameter Tuning:

```
from sklearn.model_selection import GridSearchCV
```

```
params={'n_estimators':[50,60],
        'max_features':['auto','log2'],
        'max_depth':range(3,10,3),
         'criterion':['squared_error','absolute_error'],
         'min_samples_split':range(3,10,3)}
```

```
grid_search=GridSearchCV(estimator=rfr,param_grid=params,cv=5,verbose=3)
```

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

```
Fitting 5 folds for each of 72 candidates, totalling 360 fits
[CV 1/5] END criterion=squared_error, max_depth=3, max_features=auto, min_samples_split=3, n_estimators=50;, score=0.279 tota
l time=   0.1s
[CV 2/5] END criterion=squared_error, max_depth=3, max_features=auto, min_samples_split=3, n_estimators=50;, score=0.221 tota
l time=   0.1s
[CV 3/5] END criterion=squared_error, max_depth=3, max_features=auto, min_samples_split=3, n_estimators=50;, score=0.191 tota
l time=   0.1s
[CV 4/5] END criterion=squared_error, max_depth=3, max_features=auto, min_samples_split=3, n_estimators=50;, score=0.236 tota
l time=   0.1s
[CV 5/5] END criterion=squared_error, max_depth=3, max_features=auto, min_samples_split=3, n_estimators=50;, score=0.158 tota
l time=   0.1s
[CV 1/5] END criterion=squared_error, max_depth=3, max_features=auto, min_samples_split=3, n_estimators=60;, score=0.269 tota
l time=   0.2s
[CV 2/5] END criterion=squared_error, max_depth=3, max_features=auto, min_samples_split=3, n_estimators=60;, score=0.224 tota
l time=   0.2s
[CV 3/5] END criterion=squared_error, max_depth=3, max_features=auto, min_samples_split=3, n_estimators=60;, score=0.202 tota
l time=   0.1s
[CV 4/5] END criterion=squared_error, max_depth=3, max_features=auto, min_samples_split=3, n_estimators=60;, score=0.258 tota
l time=   0.1s
```

```
grid_search.best_params_
```

```
{'criterion': 'squared_error',
 'max_depth': 9,
 'max_features': 'auto',
 'min_samples_split': 6,
 'n_estimators': 60}
```

```
grid_search.best_estimator_
```

```
RandomForestRegressor(max_depth=9, min_samples_split=6, n_estimators=60)
```

```
final_model=RandomForestRegressor(criterion='squared_error',max_depth=9,max_features='auto',min_samples_split=6,n_estimators=60)
final_model.fit(x_train,y_train)
```

```
RandomForestRegressor(max_depth=9, min_samples_split=6, n_estimators=60)
```

```
final_model_pred=final_model.predict(x_test)
print("Predicted value:\n",final_model_pred)
```

```
Predicted value:
 [1157989.36257627 1317746.46419943  649855.9745699  ...  603333.74735824
  704272.00324607  693658.14892224]
```

```
print("R2 Score is:",r2_score(y_test,final_model_pred)*100)
print("Mean Squared Error value is:",mean_squared_error(y_test,final_model_pred))
print("Mean Absolute Error value is:",mean_absolute_error(y_test,final_model_pred))
```

```
R2 Score is: 79.45825248471111
Mean Squared Error value is: 20678736344.62656
Mean Absolute Error value is: 96992.86959407124
```

Saving the Model:

```
import pickle
```

```
filename='car_price_predictions.pickle'
```

```
pickle.dump(final_model,open(filename,'wb'))
```

```
loaded_model=pickle.load(open(filename,'rb'))
```

```
loaded_model_pred=loaded_model.predict(x_test)
print("Predicted value:\n",loaded_model_pred)
```

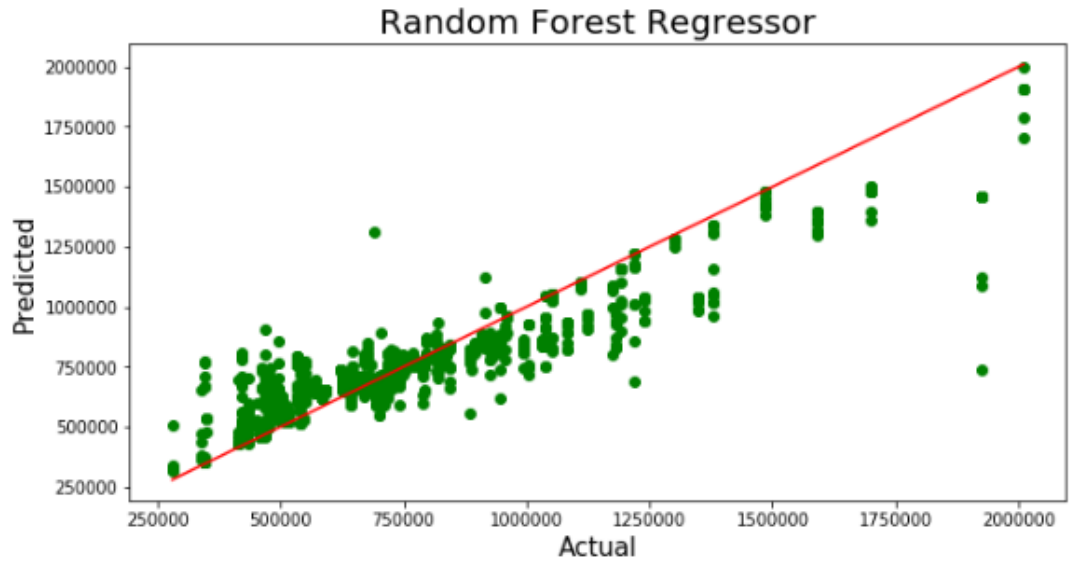
```
Predicted value:
 [1157989.36257627 1317746.46419943  649855.9745699  ...  603333.74735824
  704272.00324607  693658.14892224]
```

```
df=pd.DataFrame([loaded_model.predict(x_test)[:],y_test[:]],index=["Actual", "Predicted"])
df
```

	0	1	2	3	4	5	6	7	8	9
Actual	1.157989e+06	1.317746e+06	649855.97457	644321.53855	455106.135765	611326.735952	699026.994948	804856.801141	792895.675158	825120.219776
Predicted	1.378000e+06	1.590000e+06	634000.00000	669000.00000	430000.000000	546000.000000	710000.000000	718000.000000	816000.000000	726000.000000

```
df.to_csv("Predicted_car_price.csv")
```

```
plt.figure(figsize=(10,5))
plt.scatter(y_test, loaded_model_pred, c='green')
p1 = max(max(loaded_model_pred), max(y_test))
p2 = min(min(loaded_model_pred), min(y_test))
plt.plot([p1, p2], [p1, p2], 'r-')
plt.xlabel('Actual', fontsize=15)
plt.ylabel('Predicted', fontsize=15)
plt.title("Random Forest Regressor",fontsize=20)
plt.show()
```



We have finally saved the model using pickle and plot the actual and predicted value. It shows the linear relationship between the actual and predicted value.

- **Interpretation of the Results**
  - We have first scrapped the data from cardekho website.
  - We have removed the unnecessary column from the data set. Checked the null values, in our data set there is no null value present. When have extracted the car age by extracting the manufacturing year from the car model.
- And there was huge number of unnecessary entries in all the features so I have used feature extraction to get the required format of variables.
- And proper plotting for proper type of features will help us to get better insight on the data. I found both numerical columns and categorical columns in the dataset so I have chosen reg plot, strip plot and bar plot to see the relation between target and features.
- I notice a huge number of outliers and skewness in the data so we have chosen proper methods to deal with the outliers and skewness. If we ignore this outlier 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 removed outliers and skewness from the dataset so we have to choose Standardisation.
- We have to use multiple models while building model using dataset as to get the best model out of it.
- And we have to use multiple metrics like mse, mae, rmse and r2_score which will help us to decide the best model.
- I found DecisionTreeRegressor as the best model. Also, I have improved the accuracy of the best model by running hyper parameter tunning.
- At last, I have predicted the used car price using saved model. It was good!! that I was able to get the predictions near to actual values.

CONCLUSION

Key Findings and Conclusions of the Study

In this project, I have used various machine learning model to predict the car price. I have done the step-by-step analysis, data cleaning and data pre-processing. Then checked the correlation of data, variation inflation factor. Then I have use the r2 score to check the model accuracy. I have used the metrics mean squared error and mean absolute error to measure the error between the observed and the predicted value. I have done the cross-

validation score and analysis each model, their score. Then finalize the model, we have done the hyper parameter tuning to increase the accuracy. Then finally we have predicted the car price from our final model. It is very good that our actual car price and predicted value is almost same.

- **Learning Outcomes of the Study in respect of Data Science**

  The car price prediction data set is interesting and challenging to work upon. New analytical techniques of machine learning can be used in used car price research. The power of visualization has helped us in understanding the data by graphical representation it has made me to understand what data is trying to say.

  Data cleaning is important steps in project to avoid multi-collinearity issue. This predicted value helps us to understand the global car market and price of each car based on analysis of various features. This helps both buyer and seller of car for their profitable business.

- **Limitations of this work and Scope for Future Work**

  - Scrapping a data from website is crucial task, to scrape the data more than 5000 cars, it took many hours to collect data of each car.
  - In the data contains the Outliers and Skewness, by removing the outliers and skewness. We lost the data, even though we have got good accuracy after removing the outliers.

Thank you