Car Price Prediction

Submitted by:

S.Srividya

Acknowledgement

I would like to express my gratitude to my guide Shubham Yadav (SME, Flip Robo) for his constant guidance, continuous encouragement and unconditional help towards the development of the project. It was he who helped me whenever I got stuck somewhere in between. The project would have not been completed without his support and confidence he showed towards me.

Lastly, I would l like to thank all those who helped me directly or indirectly toward the successful completion of the project.

Introduction

Business Problem Framing

Machine Learning is a field of technology developing with immense abilities and applications in automating tasks, where neither human intervention is needed nor explicit programming.

The power of ML is such great that we can see its applications trending almost everywhere in our day-to-day lives. ML has solved many problems that existed earlier and have made businesses in the world progress to a great extent.

Today, we'll go through one such practical problem and build a solution(model) on our own using ML.

We are about to deploy an ML model for car selling price prediction and analysis. This kind of system becomes handy for many people.

Imagine a situation where you have an old car and want to sell it. You may of course approach an agent for this and find the market price, but later may have to pay pocket money for his service in selling your car. But what if you can know your car selling price without the intervention of an agent. Or if you are an agent, definitely this will make your work easier. Yes, this system has already learned about previous selling prices over years of various cars.

So, to be clear, this model will provide you will the approximate selling price for your car based on the fuel type, years of service, showroom price, the number of previous owners, kilometres driven, if dealer/individual, and finally if the transmission type is manual/automatic. And that's a brownie point.

Conceptual Background of the Domain Problem

The goal of this statistical analysis is to help us understand the relationship between car features and how these variables are used to predict car price.

Review of Literature

From the dataset I get to know that it is a Regression problem and Sale Price of car varies on its properties. And there are so many features which help to find it.

Motivation for the Problem Undertaken

I am doing this for practice, to get more hands-on data exploration, Feature extraction and Model building.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

I have used Log transformation for transforming the continuous numerical variable containing non-zero elements only as during analysis I found that these variables were not normally distributed, so transformed them using log normal transformation so that the features will be close to normal distributed. I have done some testing separately to check the importance of categorical variables with respect to the Sale Price of the Car. Use of Mean, Median to replace the Missing Values in features. Use of Correlation matrix to check the importance and correlation of numerical variables with respect to target variable Sale price and Feature scaling using Min Max scaler as we have positive data points.

Data Sources and their formats

Data I collected from OLX and car Dekho using web scrapping.
There are more than 5000 observations and 10+ features including the target feature Price in dataset.

Data Pre-processing Done

I have handled the missing values in data set. Based on the Data description I have imputed the missing data. Most of the features having nan values which were described as absence of feature in data description, so I have replaced them with 'not available' for each feature having nan value. Mostly NaN values are replaced with mode of the column.

Data Inputs- Logic- Output Relationships

I have found out that with continuous numerical variable there is a linear Relationship with the Sale Price. And for categorical variable, I have used Boxplot for each categorical feature that shows the relation with the median sale price for all the sub categories in each categorical variable. For continuous numerical variables I have used scatter plot to show the relationship between continuous numerical variable and target variable.

Hardware and Software Requirements and Tools Used

The system requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view. System requirements are all of the requirements at the system level that describe the functions which the system as a whole should fulfil to satisfy the stakeholder needs and requirements, and is expressed in an appropriate combination of textual statements, views, and non-functional requirements; the latter expressing the levels of safety, security, reliability, etc., that will be necessary.

Hardware requirements: -

1. Processor — core i5 and above
2. RAM — 8 GB or above
3. SSD — 250GB or above

Software requirements: -

Anaconda

Libraries: -

From sklearn.preprocessing import StandardScaler

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.

from sklearn.preprocessing import Label Encoder

Label Encoder and One Hot Encoder. These two encoders are parts of the SciKit Learn library in Python, and they are used to convert categorical data, or text data, into numbers, which our predictive models can better understand.

from sklearn.model_selection import train_test_split,cross_val_score

Train_test_split is a function in Sklearn 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, Sklearn train_test_split will make random partitions for the two subsets.

The algorithm is trained and tested K times, each time a new set is used as testing set while remaining sets are used for training. Finally, the result of the K-Fold Cross-Validation is the average of the results obtained on each set.

from sklearn.neighbors import KNeighborsRegressor

K Nearest Regressor (KNN) is a very simple, easy to understand, versatile and one of the topmost machine learning algorithms. KNN used in the variety of applications such as finance, healthcare, political science, handwriting detection, image recognition and video recognition

from sklearn.linear_model import LinearRegression

The library sklearn can be used to perform linear regression in a few lines as shown using the LinearRegression class. It also supports multiple features. It requires the input values to be in a specific format hence they have been reshaped before training using the fit method.

from sklearn.tree import DecisionTreeRegressor

Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as Neural Network. Its training time is faster compared to the neural network algorithm. The time complexity of decision trees is a function of the number of records and number of attributes in the given data. The decision tree is a distribution-free or non-parametric method, which does not depend upon probability distribution assumptions. Decision trees can handle high dimensional data with good accuracy

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

For feature transformation I have used Log normal transformation to make the continuous non zero variables close to normal distributed. Use of Annona test to check the importance of categorical features. Use of Pearson's correlation coefficient to check the correlation between dependent and independent features. Use of Min Max scaler to scale down the features and one label encoding to encode categorical features in numeric.

Testing of Identified Approaches (Algorithms)

Listing down all the algorithms used for the training and testing.

- KNN = KNeighborsRegressor ()
- LR = LinearRegression ()
- SVR = SVR ()
- DT = DecisionTreeRegressor ()
- RF = RandomForestRegressor ()

I applied all these algorithms in the dataset.

Run and Evaluate selected models

```
********************************************************************************
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
accuracy score of -> LinearRegression()
R2 Score:                  0.3979956390065177
Mean Absolute Error:       348179.8504065433
Mean Squared error:        292679573364.76843
Root Mean Squared Error:   540998.6814815433
[ 0.28239608  0.33746338  0.33606765  0.31048683  0.2418123  -0.14478279
 -1.21195601 -0.80759542]
cross validation score:    -0.08201349924323054
Difference between R2 score and cross validatio score is -  0.31598213976328715
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
********************************************************************************
********************************************************************************
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
accuracy score of -> RandomForestRegressor()
R2 Score:                  0.942427658793392
Mean Absolute Error:       70772.20371593907
Mean Squared error:        27990242851.65163
Root Mean Squared Error:   167302.84770933108
[0.81426078 0.97184866 0.93190511 0.82971152 0.22977537 0.09891211
 0.54632582 0.34290107]
cross validation score:    0.5957050552245671
Difference between R2 score and cross validatio score is -  0.3467226035688249
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
********************************************************************************
********************************************************************************
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
accuracy score of -> DecisionTreeRegressor()
R2 Score:                  0.8025280134218236
Mean Absolute Error:       96162.70219558361
Mean Squared error:        96005976913.21326
Root Mean Squared Error:   309848.31274869526
[ 0.73227367  0.96457272  0.95520164  0.73384649 -0.1876508   0.08595641
  0.22987608  0.04940252]
cross validation score:    0.4454348405627029
Difference between R2 score and cross validatio score is -  0.3570931728591207
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
********************************************************************************
********************************************************************************
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
accuracy score of -> KNeighborsRegressor()
R2 Score:                  0.8026027253754061
Mean Absolute Error:       154408.3837577287
Mean Squared error:        95969653816.3777
Root Mean Squared Error:   309789.6928827325
[ 0.73776722  0.87953519  0.83798815  0.67058096  0.05325959 -0.01707755
 -0.6462828  -0.40734599]
cross validation score:    0.26355309605388344
Difference between R2 score and cross validatio score is -  0.5390496293215227
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
********************************************************************************
```

```
********************************************************************************
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
accuracy score of -> GradientBoostingRegressor()
R2 Score:                  0.8624312091230009
Mean Absolute Error:       162959.35438397055
Mean Squared error:        66882530478.25193
Root Mean Squared Error:   258616.57038606773
[0.74921071 0.82597116 0.7765431  0.75897622 0.59228481 0.35898361
 0.28989605 0.37541969]
cross validation score:    0.5909106711834917
Difference between R2 score and cross validatio score is -  0.27152053793950914
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
********************************************************************************
********************************************************************************
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
accuracy score of -> Ridge()
R2 Score:                  0.39688808772876827
Mean Absolute Error:       348169.6531177393
Mean Squared error:        293218037297.01697
Root Mean Squared Error:   541496.1101402455
[ 0.27720092  0.33307342  0.3323356   0.30833176  0.23881617 -0.10196335
 -1.16429719 -0.76512607]
cross validation score:    -0.0677035922869613
Difference between R2 score and cross validatio score is -  0.32918449544180695
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
********************************************************************************
********************************************************************************
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
accuracy score of -> SVR()
R2 Score:                  -0.09127767561770539
Mean Absolute Error:       427796.6110119287
Mean Squared error:        530552110943.5022
Root Mean Squared Error:   728390.0815795766
[-0.38090044 -0.4127388  -0.32210507 -0.25509455 -0.01735427 -0.02341763
 -0.33261669 -0.15301844]
cross validation score:    -0.23715573682443783
Difference between R2 score and cross validatio score is -  -0.3284334124421432
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
********************************************************************************
```

Hyper Parameter Tuning

```
:
from sklearn.model_selection import GridSearchCV

parameters = {"max_depth":range(21,25),
              "criterion":['mse'],
              "max_features": ['auto', 'sqrt'],
              "min_samples_leaf":range(1,5)}

clf = GridSearchCV(RandomForestRegressor(), parameters, refit = True, verbose = 3)

clf.fit(x_train,y_train) #fitting train and test data
```

```
GridSearchCV(estimator=RandomForestRegressor(),
             param_grid={'criterion': ['mse'], 'max_depth': range(21, 25),
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_leaf': range(1, 5)},
             verbose=3)
```

```
clf.best_params_  #Best parameters
```

```
{'criterion': 'mse',
 'max_depth': 21,
 'max_features': 'sqrt',
 'min_samples_leaf': 1}
```

```
clf_pred=clf.best_estimator_.predict(x_test)
```

```
r2_score(y_test, clf_pred)
```

```
0.9441445648442222
```

Our model learnt approx 94.41%

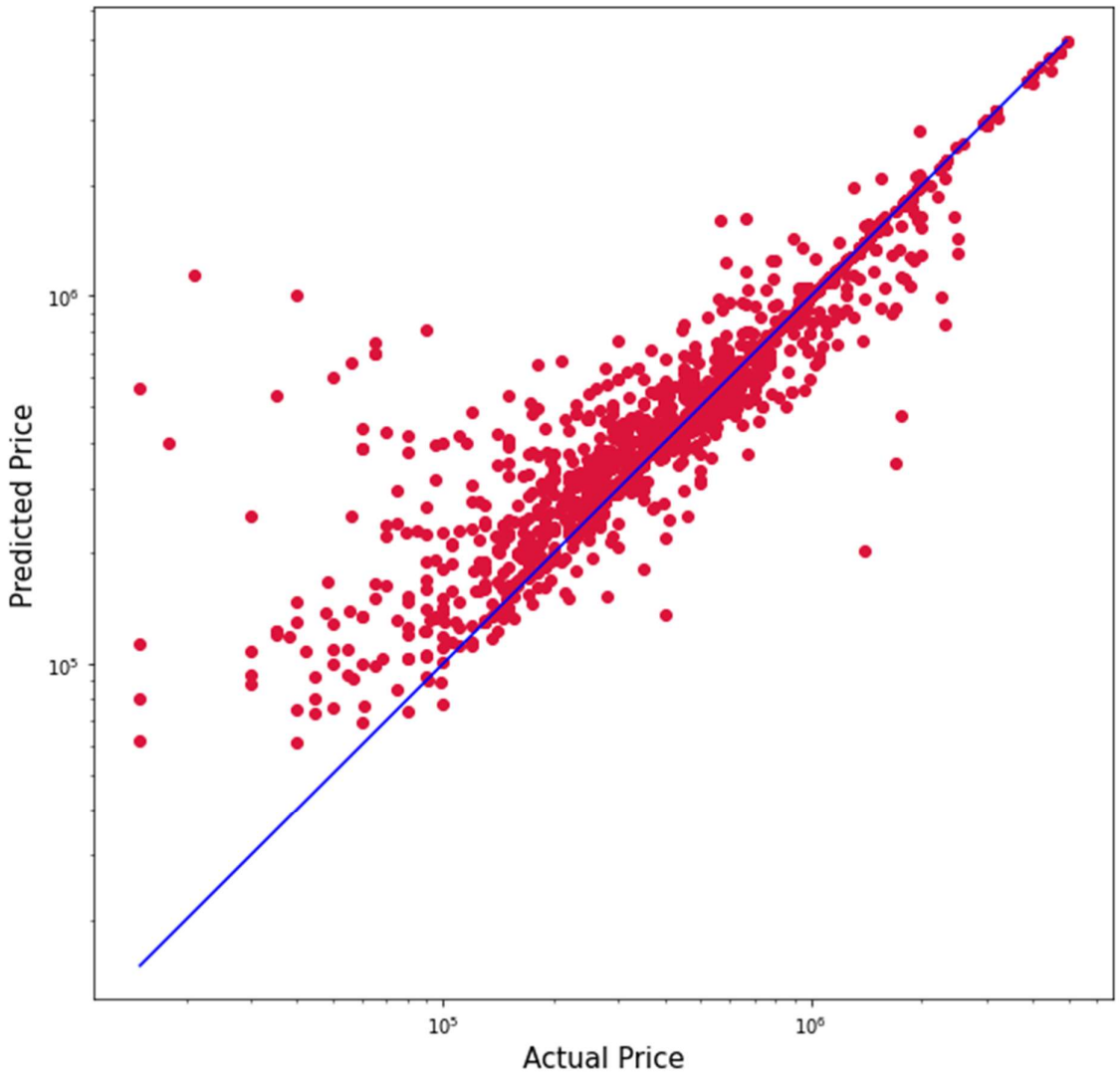
Saving the model

```
import joblib
joblib.dump(clf.best_estimator_,"Carz.obj")
RF_from_joblib=joblib.load('Carz.obj')
Predicted = RF_from_joblib.predict(x_test)
Predicted
```

```
array([ 949088.57142857,  221606.0001    , 4200000.        , ...,
        370000.        ,  300000.        ,  150445.64112564])
```

Plotting Actual vs Predicted Results

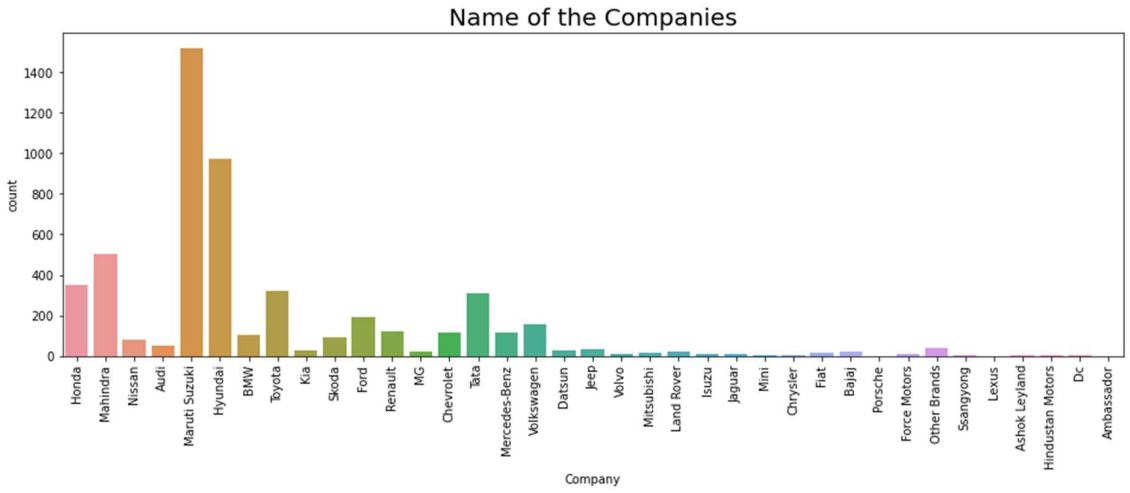
```
plt.figure(figsize=(10,10))
plt.scatter(y_test, Predicted, c='crimson')
plt.yscale('log')
plt.xscale('log')
p1 = max(max(Predicted), max(y_test))
p2 = min(min(Predicted), min(y_test))
plt.plot([p1, p2], [p1, p2], 'b-')
plt.xlabel('Actual Price', fontsize=15)
plt.ylabel('Predicted Price', fontsize=15)
plt.axis('equal')
plt.show()
```

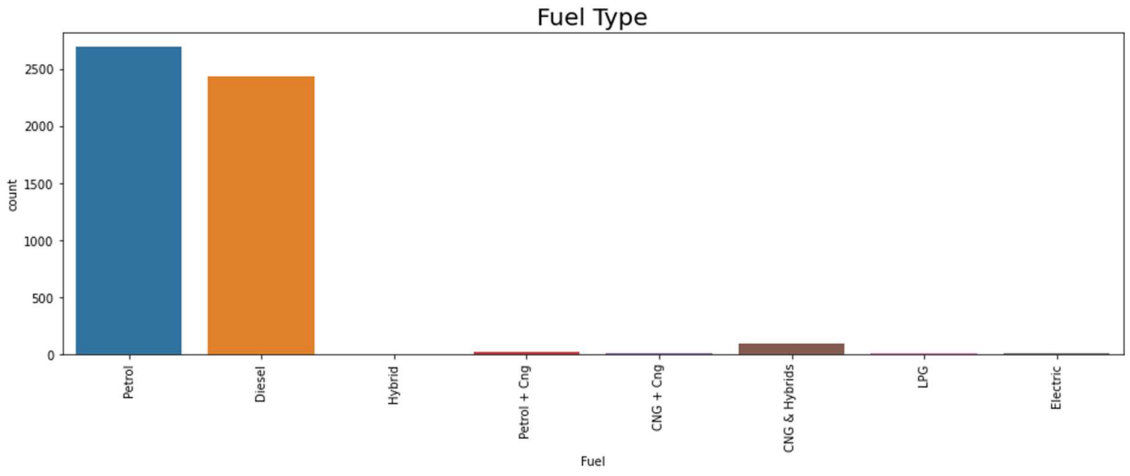

Predicted Price
Actual Price
10^6
10^5
10^5
10^6

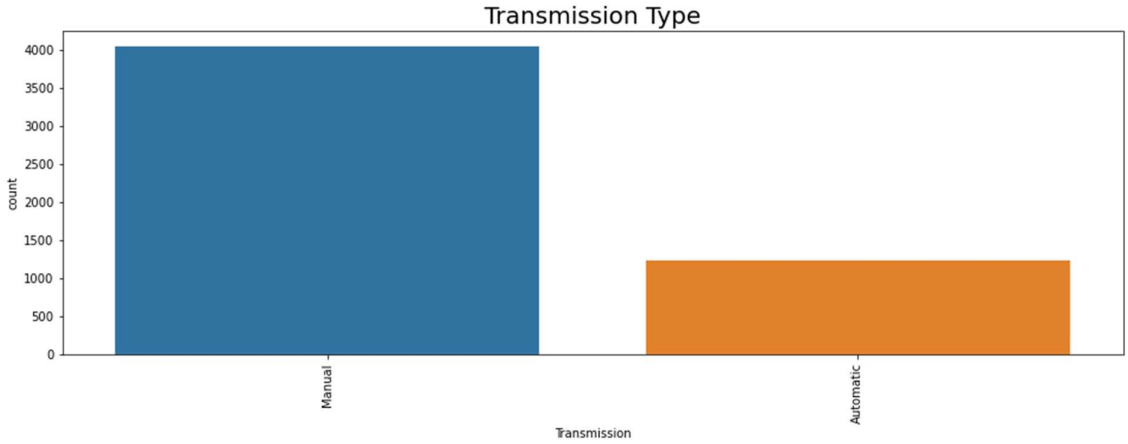
Key Metrics for success in solving problem under consideration

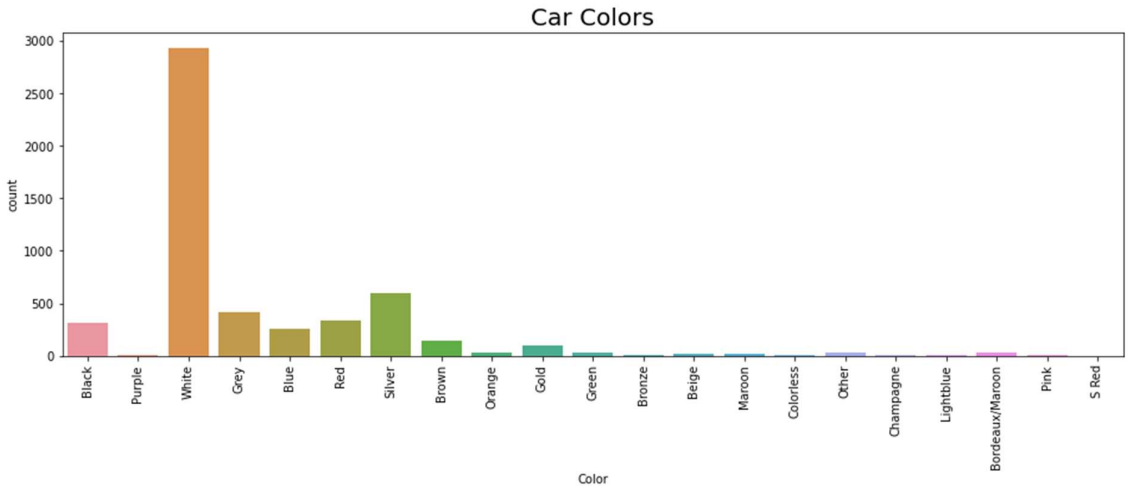
As this is a regression problem, we are required to predict the continuous feature (Sale Price) I have used R2 score, mean absolute error, mean squared error and root mean squared error.

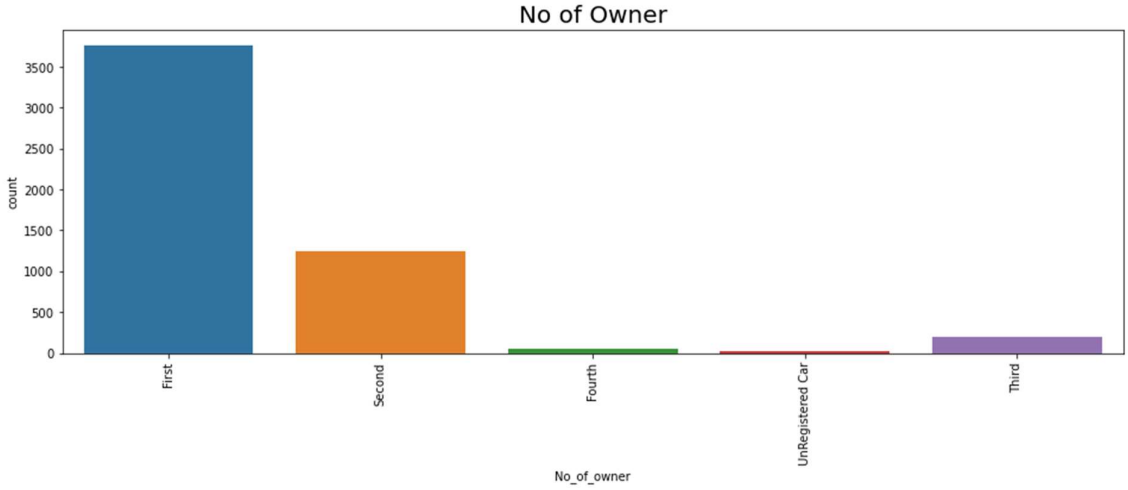
Visualizations

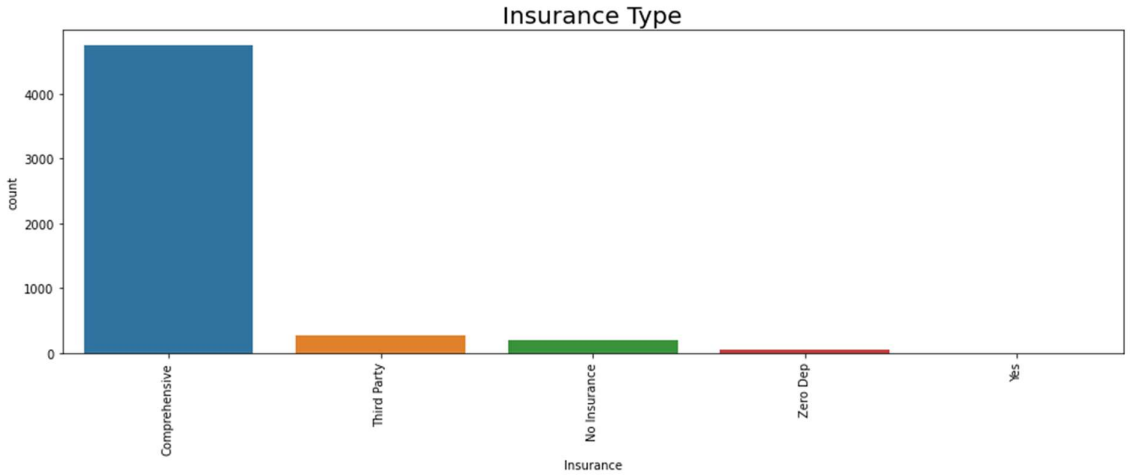


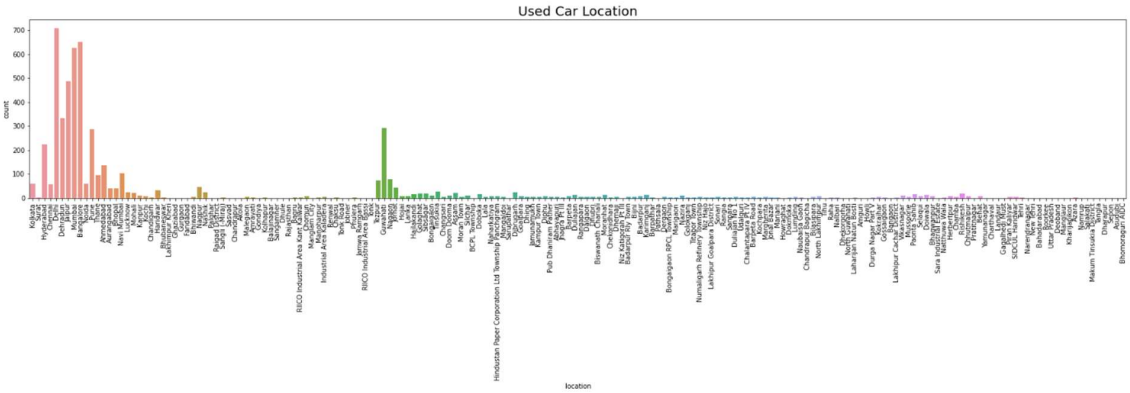


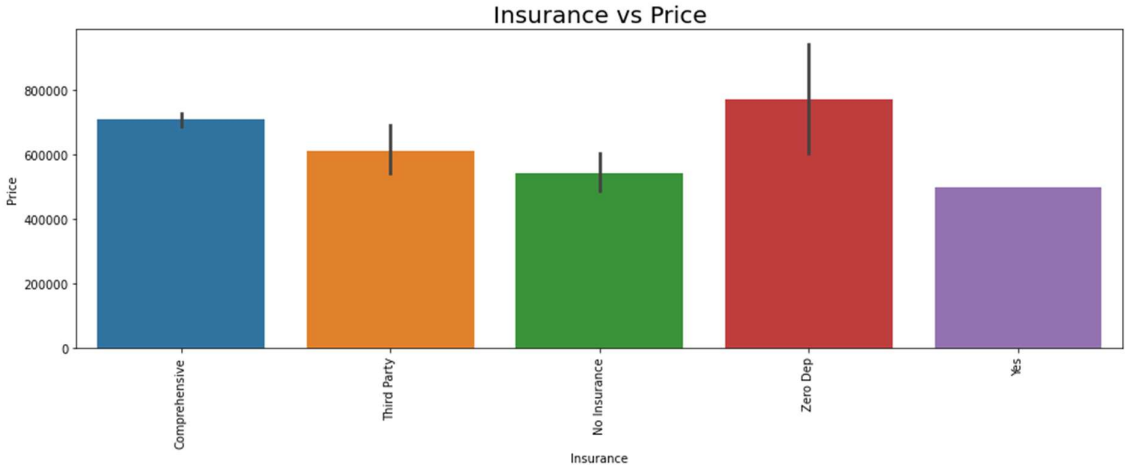

Transmission Type
count
4000
3500
3000
2500
2000
1500
1000
500
0
Manual
Automatic
Transmission

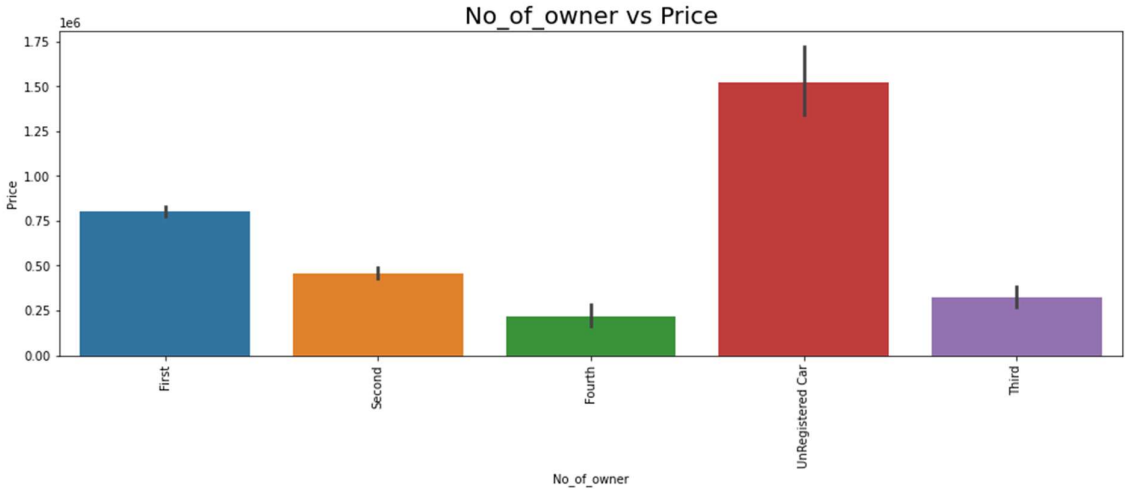

Car Colors
count
3000
2500
2000
1500
1000
500
0
Black
Purple
White
Grey
Blue
Red
Silver
Brown
Orange
Gold
Green
Bronze
Beige
Maroon
Colorless
Other
Champagne
Lightblue
Bordeaux/Maroon
Pink
S Red
Color

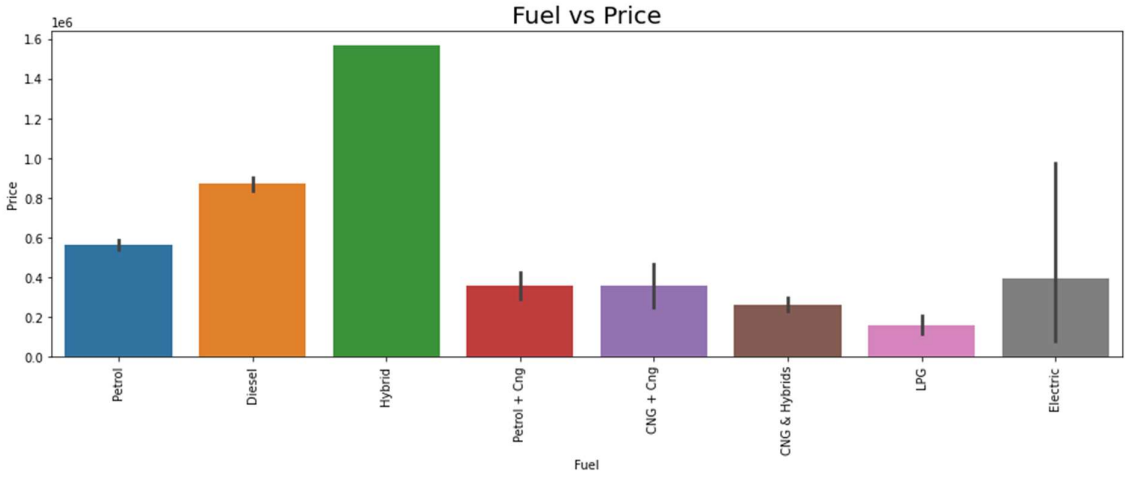

No of Owner
count
3500
3000
2500
2000
1500
1000
500
0
First
Second
Fourth
UnRegistered Car
Third
No_of_owner

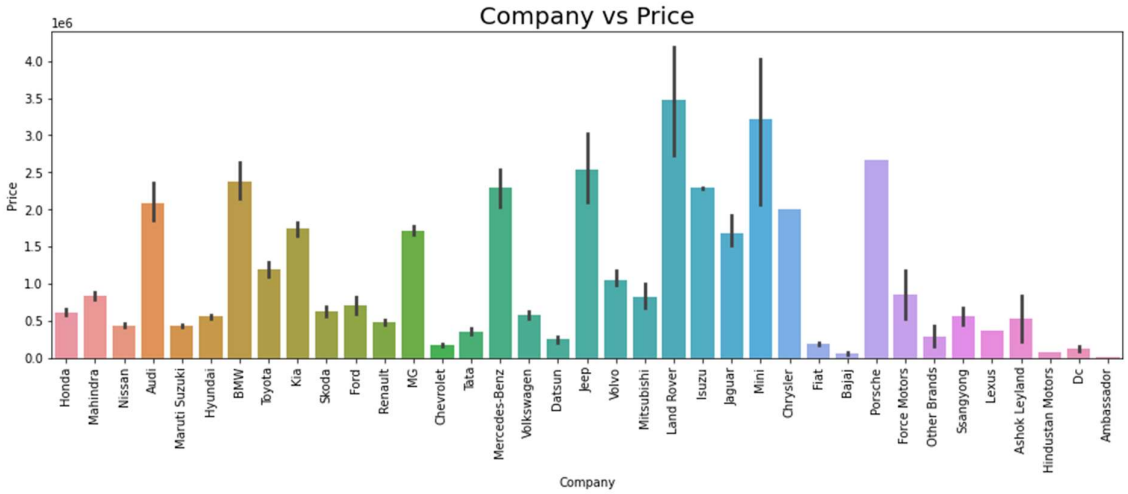

Insurance Type
count
4000
3000
2000
1000
0
Comprehensive
Third Party
No Insurance
Zero Dep
Yes
Insurance

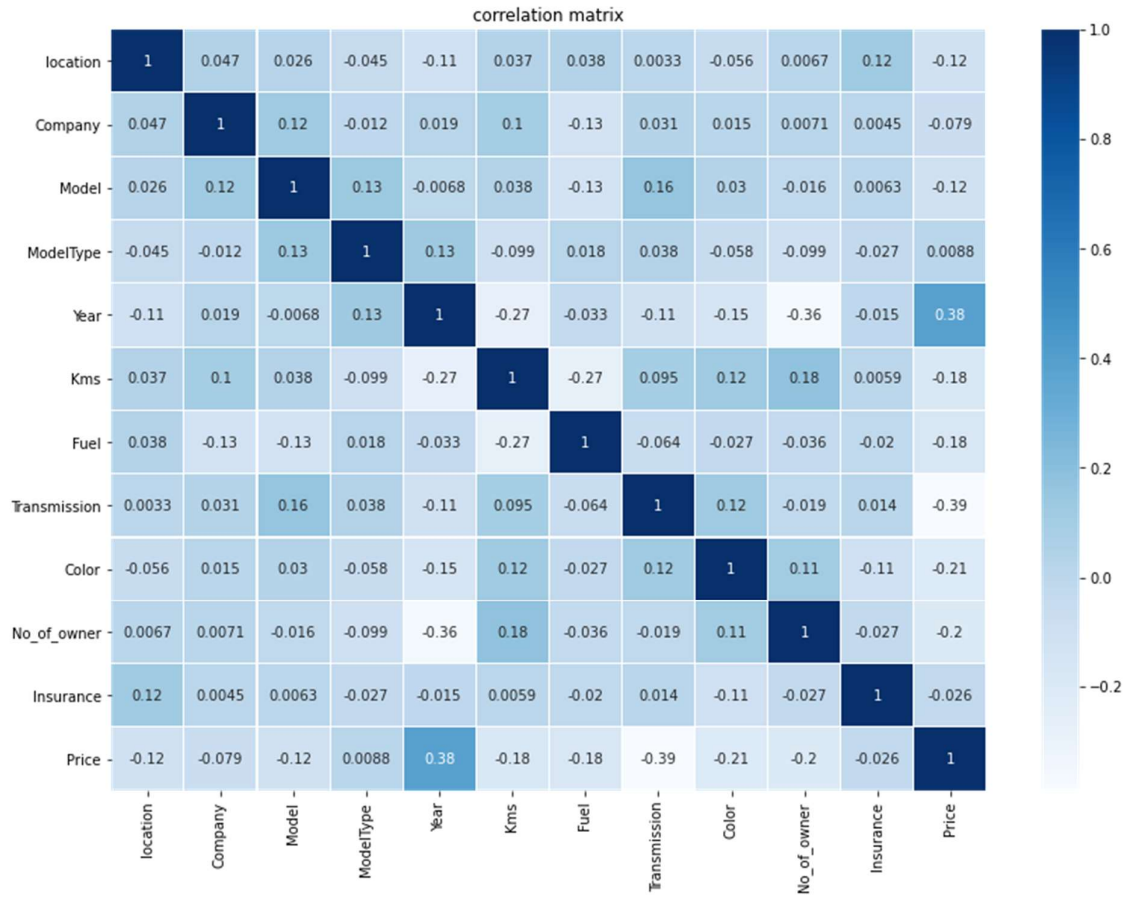

Used Car Location
count
location


Insurance vs Price
Price
800000
600000
400000
200000
0
Comprehensive
Third Party
No Insurance
Zero Dep
Yes
Insurance


No_of_owner vs Price
1e6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Price
First
Second
Fourth
UnRegistered Car
Third
No_of_owner


Fuel vs Price
1e6
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Price
Petrol
Diesel
Hybrid
Petrol + Cng
CNG + Cng
CNG & Hybrids
LPG
Electric
Fuel


Company vs Price
1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Price
Honda
Mahindra
Nissan
Audi
Maruti Suzuki
Hyundai
BMW
Toyota
Kia
Skoda
Ford
Renault
MG
Chevrolet
Tata
Mercedes-Benz
Volkswagen
Datsun
Jeep
Volvo
Mitsubishi
Land Rover
Isuzu
Jaguar
Mini
Chrysler
Fiat
Bajaj
Porsche
Force Motors
Other Brands
Ssangyong
Lexus
Ashok Leyland
Hindustan Motors
Dc
Ambassador
Company



Observation:

- Maruti Suzuki cars are more in dataset.
- People use Petrol and Diesel car more.
- People use Manual Car.
- People choose White colour car over other colour.
- Most of the car owner is first.

- People take compressive type of Insurance.
- We have Delhi, Dehradun, Mumbai, Bangalore and Jaipur location car most.
- If the Insurance type is Zero Dep, Price of the car will be high.
- If the car is unregistered, Price will be high.
- If the Fuel type is Hybrid, Price will be high.
- Prices are high for Land Rover.
- Prices are very high in Faridabad.
- Price is highly co related with car purchased year.

Interpretation of the Results

Most of the features were having missing values. From Random Forest Regressor R2 score was 94.41, means 95% of the variance of the dependent variable being studied is explained by the variance of the independent variable.

Higher the R2 score means the model is well fit for the data. However, if R2 score is very high, it might be a case of overfitting. Other metrics Mean Absolute Error, Mean Squared Error and Root Mean Squared Error, with gradient boosting these scores are less then compared to other models. If these errors are less that means the model shows less errors.

Conclusion

Key Findings and Conclusions of the Study

From this dataset I get to know that each feature plays a very import role to understand the data. Data format plays a very important role in the visualization and Appling the models and algorithms.

Learning Outcomes of the Study in respect of Data Science

The power of visualization is helpful for the understanding of data into the graphical representation its help me to understand that what data is trying to say, Data cleaning is one of the most important steps to remove missing value or null value fill it by mean, median or by mode or by 0.
Various algorithms I used in this dataset and to get out best result and save that model. The best algorithm is Random Forest Regressor.

Limitations of this work and Scope for Future Work

Limitations of this project is we have less number of features. If we get interior column, where we will get feature like, A/C, air bag etc. More the number of features, more accuracy we'll get.

In future, if someone do the proper and detail study of this dataset's each column than the accuracy will be so high.