Used Car Price Prediction

Submitted by:

Shashwat Shukla

ACKNOWLEDGMENT

In used car price prediction project, I collected data from various online website but most of the data I fetched from cars24 because this website has more used cars and data is accurate without any null missing value rest of website like – olx and Car trade, this website has used car listing but not more. I took help from google in some steps. I scrapped data from these websites with the help of Selenium web driver.

INTRODUCTION

- **Business Problem Framing**

  As the market of used car is increasing gradually, there are lot of big car market players in the markets as well as small and local vendor. If we can google the used car, we will find thousand of website and thousands of physical stores. Indian market is going very flexible with used car and even small town is the best market for used car as they will buy those used car which is banned in big metro cities. So, when the market is too big, we need to set some parameter which will help in prediction of used cars. This will also help to seller as well as buyer.

- **Conceptual Background of the Domain Problem**

  This project is directly linked with sales, marketing and automobile domain. The problem we are helping to predict the used car price because all the automobile companies have their policies to sell new cars but no company or vendor has any model or any policies to setup the price of used cars, which make difference between in pricing of used cars.

- **Review of Literature**

  For this used car prediction model, I did some research on google to check how they predict the car price but did not get any solid proof as every website has some difference in pricing. As I go deep with this model the difference on price is remain same but we can reduce this difference with our model.

- **Motivation for the Problem Undertaken**
  The motivation behind this project to build a regression model.

Analytical Problem Framing

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

  As I performed the data scrapping with Selenium web driver. There are most of the feature has categorical values like – year of manufacturing, model of car and fuel type and two feature has continuous data which Run and price. There are no null values in the dataset. So, I do not need to perform and filling missing values or perform any mathematical step.

- **Data Sources and their formats**

  All the data is sourced from used selling websites. Some data in categorical format and some data in continuous format. As I scrapped the data, So I take care of null values and data set format.

- **Data Preprocessing Done**

I Scrapped the data from websites. So, I took care of many data cleaning step thing like – Null values, data type and miss type but I also performed some step to make data clean to build the model. As there some column which has values in character format, I used label encoder to change character into the integer as model only understand the numbers.

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

  In the used car price prediction dataset, the target has continuous values and most of the feature has categorical values. The input values are categorical and output is continuous. The output has very strong relation with input columns. Year of manufacturing has very strong relation with Output (Price). So, I can say that feature has good relation with output.

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

  For this used car price prediction model, I Used my personal laptop which has below configuration –

  RAM – 2 GB
  Operating system – Windows 10
  Processor – i5
  Hard Disk – 500 GB

  This configuration is not enough for big dataset but for this model which has small dataset it worked perfectly.

Model/s Development and Evaluation

- **Identification of possible problem-solving approaches (methods)**

  For the prediction of used car price, there are only 6 feature and each feature has direct relation with target and this prediction directly link with real life data, this made easy to understand feature and goal of the dataset. So, I used simple approach to predict the used car price.

- **Testing of Identified Approaches (Algorithms)**

For used car price prediction I used several metrics and model which is mentioned below –

StandardScaler
train_test_split
cross_val_score
LinearRegression
GradientBoostingRegressor
KNeighborsRegressor
BayesianRidge
SVR
RandomForestRegressor

- **Run and evaluate selected models**

  I used 6 model to predict the price of used car the code and results mentioned below –

LinearRegression

```
### Importing the LinearRegression and checking the r2 score, cross validation and errors
lr= LinearRegression()
lr.fit(x_train,y_train)
pred_lr=lr.predict(x_test)
r2_lr =r2_score(y_test,pred_lr)
scr = cross_val_score(lr, x,y, cv=5)
Corss_lr = scr.mean()
mae_lr = mean_absolute_error(y_test,pred_lr)
mse_lr = mean_squared_error(y_test,pred_lr)
rmse_lr = np.sqrt(mse_lr)
print("\t\t\t LinearRegression Model Accuracy and r2 score, cross validation and errors",  '\n\nR2 score for LinearRegression mod
```

```
                    LinearRegression Model Accuracy and r2 score, cross validation and errors

R2 score for LinearRegression model 37.48124145150287
 Cross Validation score  27.728660349389756
Mean_absolute_error 134934.64462032236
Mean_squared_error 36139425227.23567
Root mean_squared_error 190103.72228663927
```

GradientBoostingRegressor

```
### Importing the GradientBoostingRegressor and checking the r2 score, cross validation and errors
gbr= GradientBoostingRegressor()
gbr.fit(x_train,y_train)
pred_gbr=gbr.predict(x_test)
r2_gbr =r2_score(y_test,pred_gbr)
scr = cross_val_score(gbr, x,y, cv=5)
Corss_gbr = scr.mean()
mae_gbr = mean_absolute_error(y_test,pred_gbr)
mse_gbr= mean_squared_error(y_test,pred_gbr)
rmse_gbr = np.sqrt(mse_gbr)
print("\t\t\t GradientBoostingRegressor Model Accuracy and r2 score, cross validation and errors",  '\n\nR2 score for GradientBoo
```

GradientBoostingRegressor Model Accuracy and r2 score, cross validation and errors

```
R2 score for GradientBoostingRegressor model 79.89301343916908
 Cross Validation score  72.92328469603412
Mean_absolute_error 74818.80522634079
Mean_squared_error 11622990510.864029
Root mean_squared_error 107809.97407876523
```

BayesianRidge

```
### Importing the BayesianRidge and checking the score and r2 score
br= BayesianRidge()
br.fit(x_train,y_train)
pred_br=br.predict(x_test)
r2_br =r2_score(y_test,pred_br)
scr = cross_val_score(br, x,y, cv=5)
Corss_br = scr.mean()
mae_br = mean_absolute_error(y_test,pred_br)
mse_br= mean_squared_error(y_test,pred_br)
rmse_br= np.sqrt(mse_br)

print("\t\t\t BayesianRidge Model Accuracy and r2 score, cross validation and errors",  '\n\nR2 score for BayesianRidge model', r
```

BayesianRidge Model Accuracy and r2 score, cross validation and errors

```
R2 score for BayesianRidge model 37.474190532216944
 Cross Validation score  -7.979611392234354
Mean_absolute_error 134928.17729524203
Mean_squared_error 36143501062.65264
Root mean_squared_error 190114.44201494174
```

SVR

```
### Importing the SupportVectorRegressor and checking the score and r2 score
svr= SVR()
svr.fit(x_train,y_train)
pred_svr=svr.predict(x_test)
r2_svr=r2_score(y_test,pred_svr)
scr = cross_val_score(svr, x_scaled,y, cv=5)
Corss_svr = scr.mean()
mae_svr = mean_absolute_error(y_test,pred_svr)
mse_svr = mean_squared_error(y_test,pred_svr)
rmse_svr = np.sqrt(mse_svr)
print("\t\t\t SupportVectorRegressor Model Accuracy and r2 score, cross validation and errors",  '\n\nR2 score for SupportVectorF
```

SupportVectorRegressor Model Accuracy and r2 score, cross validation and errors

```
R2 score for SupportVectorRegressor model -4.962463187934318
 Cross Validation score - -10.465287834869134
Mean_absolute_error 183397.73289326878
Mean_squared_error 60674318846.30752
Root mean_squared_error 246321.57608765725
```

RandomForestRegressor

```
### Importing the RandomForestRegressor and checking the score and r2 score
RFR= RandomForestRegressor()
RFR.fit(x_train,y_train)
pred_RFR=RFR.predict(x_test)
r2_RFR =r2_score(y_test,pred_RFR)
scr = cross_val_score(RFR, x_scaled,y, cv=5)
Corss_RFR = scr.mean()
mae_RFR = mean_absolute_error(y_test,pred_RFR)
mse_RFR = mean_squared_error(y_test,pred_RFR)
rmse_RFR = np.sqrt(mse_RFR)

print("\t\t\t RandomForestRegressor Model Accuracy and r2 score, cross validation and errors",  '\n\nR2 score for RandomForestReg
```

RandomForestRegressor Model Accuracy and r2 score, cross validation and errors

```
R2 score for RandomForestRegressor model 92.12563084752732
 Cross Validation score  85.93525148101723
Mean_absolute_error 39135.64719512196
Mean_squared_error 4551836629.588327
Root mean_squared_error 67467.30044687077
```

KNeighborsRegressor

```
### Importing the KNeighborsRegressor and checking the score and r2 score
KNN= KNeighborsRegressor()
KNN.fit(x_train,y_train)
pred_KNN=KNN.predict(x_test)
r2_KNN =r2_score(y_test,pred_KNN)
scr = cross_val_score(KNN, x_scaled,y, cv=5)
Corss_KNN = scr.mean()
mae_KNN = mean_absolute_error(y_test,pred_KNN)
mse_KNN = mean_squared_error(y_test,pred_KNN)
rmse_KNN = np.sqrt(mse_KNN)

print("\t\t\t KNeighborsRegressor Model Accuracy and r2 score, cross validation and errors",  '\n\nR2 score for KNeighborsRegress
```

KNeighborsRegressor Model Accuracy and r2 score, cross validation and errors

```
R2 score for KNeighborsRegressor model 50.14973919980306
 Cross Validation score  38.323603177487634
Mean_absolute_error 113781.9512195122
Mean_squared_error 28816307530.313587
Root mean_squared_error 169753.6672072612
```

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

  I used different metrics to compare the metrics. These metrics are below mentioned –

  cross_val_score
  MAE-Mean_absolute_error
  MSE-Mean_squared_error
  RMSE-Root mean_squared_error

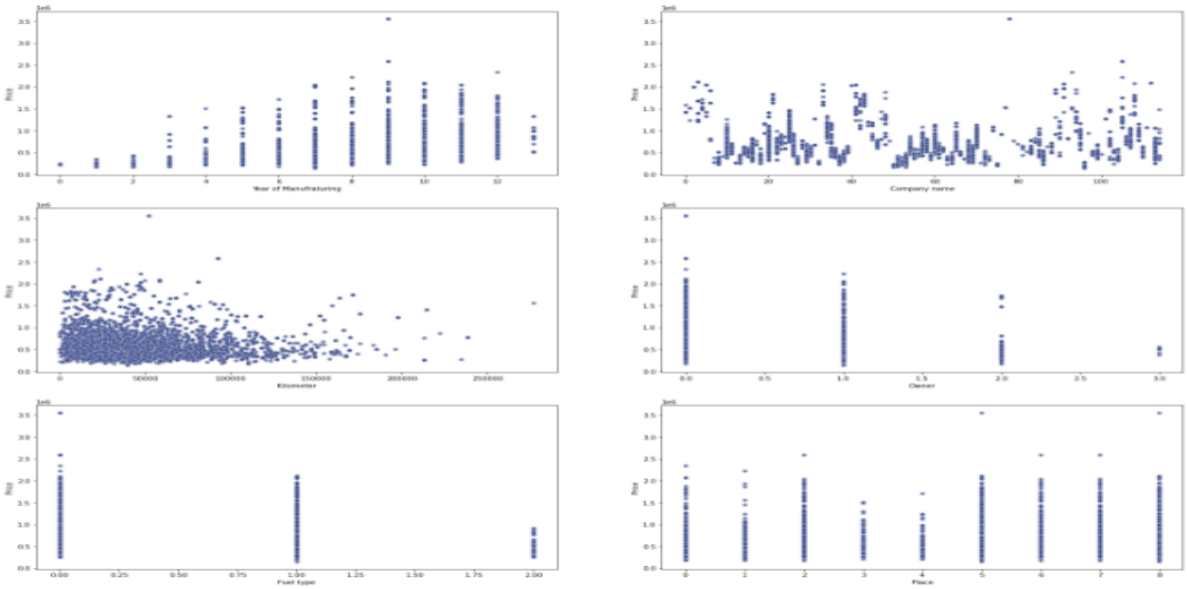
  The above metrics I used to compare the different models and these metrics I used in all model accuracy score.

- **Visualizations**

  In used car price prediction dataset, I used several plotting which is mentioned below –

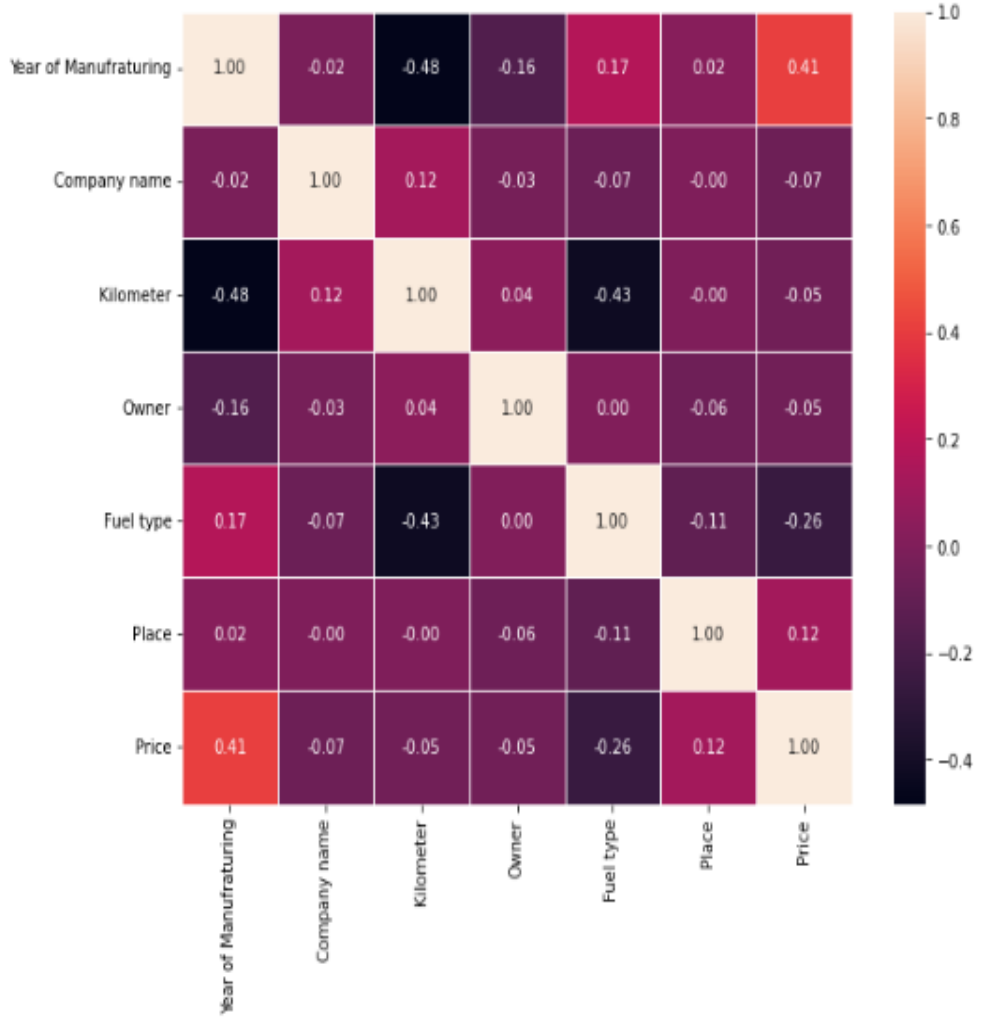
  Scatter Plot

  Scatter plot is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data. I used this scatter will all feature and targets.



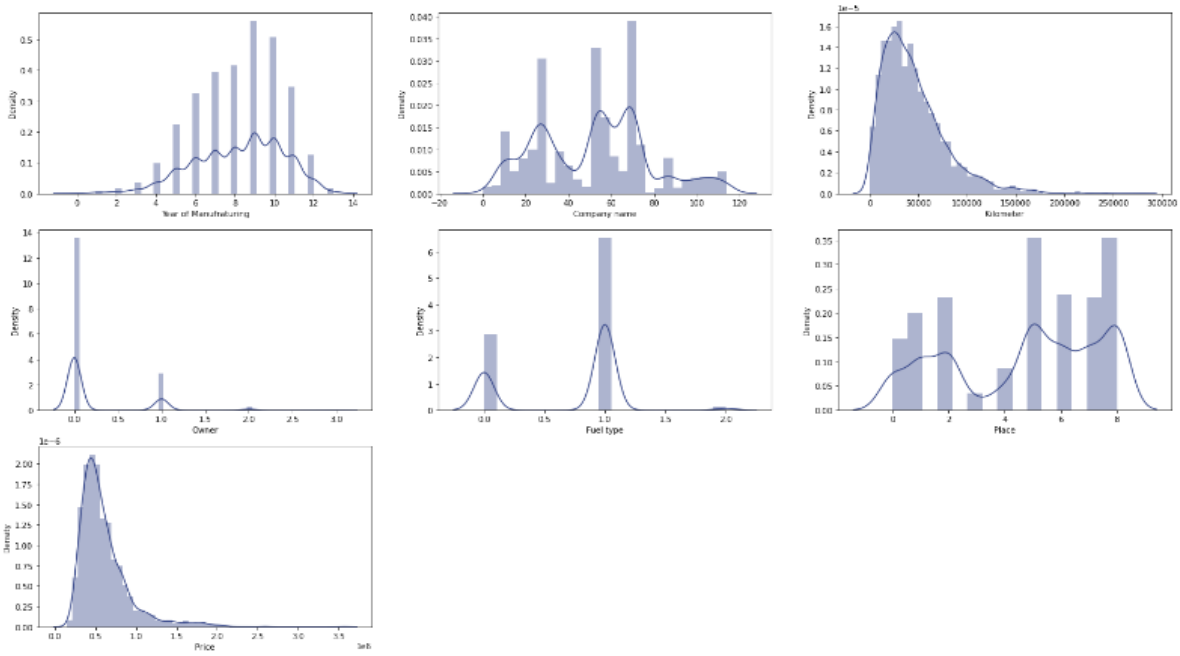
Heat Map

Multicollinearity can lead to wider confidence intervals that produce less reliable probabilities in terms of the effect of independent variables in a model. I used Heatmap to check the multicollinearity.



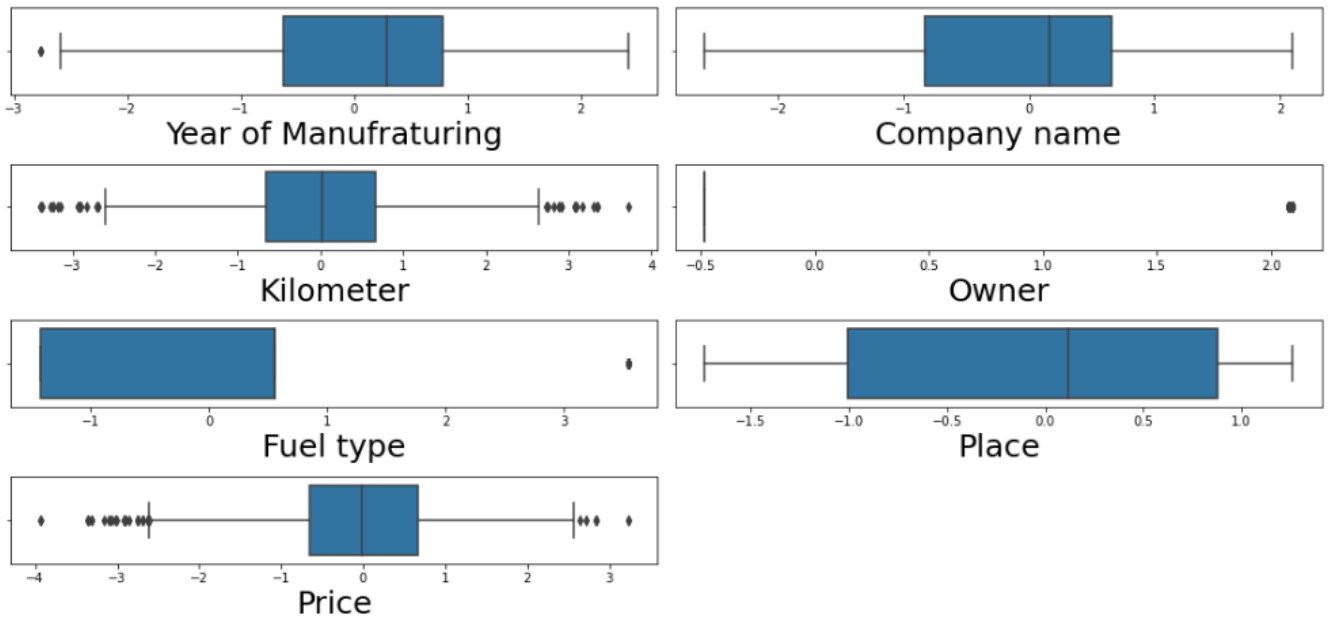
Density Plot

A density plot is a representation of the distribution of a numeric variable. It uses a kernel density estimate to show the probability density function of the variable. It is a smoothed version of the histogram and is used in the same concept.



Box Plot

Box plot or boxplot is a type of chart often used in explanatory data analysis. Box plots visually show the distribution of numerical data and skewness through displaying the data quartiles (or percentiles) and averages.



- **Interpretation of the Results**

  Scatter plot –

  - Year of manufacturing is putting good effect on used car price. Car price is less for old cars and high for recent years car.
  - All companies have sales depending on popularity. So, Maruti Suzuki has good sales compared to the other.
  - Price has directly relation with kilometre. The car which has low run has strong relation with price. The range between o-1.5k kilometre has more impact on price compared to these cars which has more kilometre range 1.5k.
  - Car price are higher for 1st and 2nd owner cars.
  - Price are higher for Petrol+ Cng and petrol.
  - Place has positive impact on price but I can say price is not putting more impact on price approx. all places have equal price.

  HeatMap-

  No feature has Multicolinearity with each other.

  - Feature with maximum relation with target - Year of Manufacturing (41%)
  - Feature with minimum relation with target - Owner and kilometre (-0.05%)

  Density Plot –

  - Most of the features are categorical values and price and kilometre column has some right skewness.

  Box Plot –

  - Only 2 columns have outliers which is Kilometers and price rest of feature have categorical values and has only one outlier which is acceptable.

CONCLUSION

- **Key Findings and Conclusions of the Study**

  On working with this dataset, I found that there is no exact feature which can put direct impact on target (Price) but I can say that year of manufacturing or Year of buying has very good impact on target (Price) and rest considered after that one. So, I can say used car price varies with year of manufacturing or year of buying.

  There are no specific criteria for the prediction of price of used car all companies have their own way to fix the price for used car but the similar thing is that every company consider the Manufacturing year.

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

  The clean data is the played very important role in building the model. The challenge I faced, when I scraped the data there were some columns which has numeric value but the datatype is object and when I tried to change the datatype it shows error that did not change as the values has mixed type of values which is integer and float. The process to deal with this problem is to remove comma (,) and specials character like ($ and KM). So, I again scrapped the data and remove this and found the changed datatype.

  In this data set I did not face any complex challenge because I scrapped the data and I took care of all the mistake which will be create any complexity in building the model.

  The accuracy and supporting metrics results mentioned below –

  I performed 6 model prediction Model Accuracy and r2 score, cross validation and errors are mentioned below -

  - Accuracy score and cross validation score for LinearRegression model 37.48 and cross validation score is 27.72
  - Accuracy scores cross validation score for GradientBoostingRegressor model 79.89 and cross validation score is 72.92
  - Accuracy scores cross validation score for **RandomForestRegressor model 91.90 and cross validation score is 86.24**
  - Accuracy scores cross validation score for BayesianRidge model 37.47 and cross validation score is -7.97
  - Accuracy scores cross validation score for KNeighborsRegressor model 50.14 and cross validation score is 38.32

- Accuracy scores cross validation score for SupportVectorRegressor model -4.96 and cross validation score is -10.46

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

  I think there is no limitation for project as this is very gradually increasing market and very big companied also participated in this field and the fact is there is no particular criteria to predict the pricing of the used car. Every company are also facing challenge how to predict which will create uniformity for pricing to the seller as well as buyer.

  I think this market is going to be very big soon. As the buying capacity and interest toward having cars is increasing gradually. The future scope is very vast.

  I only think that we need to work on that with large dataset that will help cover approx. all the model and cars and make model will be more accurate for every car every model. This is the only limitation I think with this project because scrapping the data from various website took good time and cleaning data too.