

**Car Price Prediction Project**

Submitted by:

Priyanka Yadav

**ACKNOWLEDGMENT**

I would like to thank FlipRobo Technologies to give this opportunity to work on real time dataset. I would also like to thank my mentor Swati Mahaseth for helping me throughout this project.

**INTRODUCTION**

* Business Problem Framing

The prediction or modelling the price of car with the available independent variables. This model can be used by the management to understand how exactly the prices vary with the variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be a good way for the management to understand the pricing dynamics of a new market.

* Conceptual Background of the Domain Problem

Here by using best machine learning practices we can predict that that what factors affect the price of a car and how company can predict the prices of these car by looking at it’s condition and sell it on such a price that they can make huge profits.

* Review of Literature

Car becomes one of the necessary need of each and every person around the globe. It is a very large market and there are various companies working in the domain. Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in car sales and purchases. Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques used for achieving the business goals for car companies.

By the means of this reports, I will be summarizing and explaining:

* Which variables are important to predict the price of variable?
* How do these variables describe the price of the car?
* Motivation for the Problem Undertaken

The motivation that driven me towards the timeline of this project was the sheer size and complexity of this dataset. I have tried my best to follow best practices of machine learning throughout the project and follow the steps that doesn’t only predict the best results but also tried to keep the process simple to understand.

**Analytical Problem Framing**

* Mathematical/ Analytical Modeling of the Problem

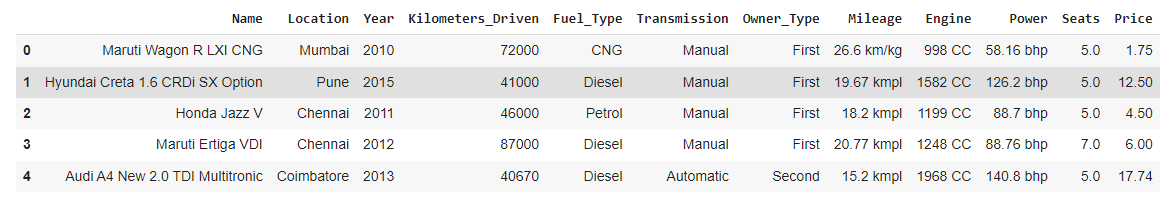
Mathematical Tools Used: Mean, Average, IQR, Standard deviation and Median for gaining insights of the dataset.

Analytical Tools Used: Correlation and Skewness for finding the relationships of dependent and independent variable and checking the distribution of data.

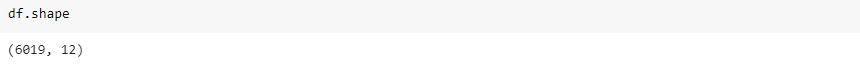
Packages Used: scikit-learn, pandas, seaborn, numpy and sklearn etc

* Data Sources and their formats

Data was extracted from the Car dekho website. Using web scrapping we extracted the dataset from the cardekho.com The data contained 12 columns and 6019 entries which are shown below:



Figure



Figure

* Data Preprocessing Done
* For keeping only the useful data we used feature engineering. We removed the data if unnecessary by qualifying them on certain conditions like their uniqueness, their correlation with target variable and the no of outliers present in that particular variable.
* We then tried the change the data into the form they belong to either by changing their type or by bifurcating the complex variables.
* We also tried to treat the outliers and skewness present in the data.
* We also scaled the data so that the data is standardized.
* Data Inputs- Logic- Output Relationships

After data pre-processing was completed, we separated the dependent and independent variable.

Other than **CarPrice** column all the columns were taken as input variables whereas the **CarPrice** column predicting the price of the house was taken as the target variable.

* State the set of assumptions (if any) related to the problem under consideration

It was pretty clear that machine learning will be used to predict the results for the dataset.

* Hardware and Software Requirements and Tools Used

Hardware Requirements: Laptop or PC with 4GB RAM or plus having i3 or above processor.

Software Requirements: Python and Jupyter Notebook.

Jupyter Tools/ Packages:pandas, numpy, seaborn, matplotlib.pyplot, import warnings, sklearn.preprocessing(Label Encoder & Standard Scaler), sklearn.model\_selection, sklearn.linear\_model etc.

**Model/s Development and Evaluation**

* Identification of possible problem-solving approaches (methods)
* For that, we used various statistical tools like median, correlation, data skewness, standard deviation for data analysis.
* We also used scikit-learn and other numerous analytical tools for achieving the target.
* Testing of Identified Approaches (Algorithms)

In this analysis, we have used multiple regressor algorithms andensemble techniques to boost the predictive scores for better results. With the help of this approach, we will be able to successfully predicting the price of a particular house based upon its features.

ALGORITHM USED:

* Linear Regression
* Decision Tree Regressor
* K-Neighbour Regressor
* Lasso Regressor
* Ridge Regressor
* Elastic

ENSEMBLE TECHNIQUE USED:

* Gradient Boosting Regressor
* AdaBoost Regressor
* Random ForestRegressor

We ran these algorithms in order to find best fitting model and best accuracy and other different parameters. The working of these algorithms are defined below:

* **Linear Regression**

**Linear Regression** is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence,the name is Linear Regression.  
In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

* **Decision Tree Regressor**

Decision tree analysis is a predictive modelling tool that can be applied across many areas. Decision trees can be constructed by an algorithmic approach that can split the dataset in different ways based on different conditions. Decisions trees are the most powerful algorithms that falls under the category of supervised algorithms.

The branches/edges represent the result of the node and the nodes have either:

1. Conditions [Decision Nodes]
2. Result [End Nodes]

The branches/edges represent the truth/falsity of the statement and takes makes a decision based on that in the example below which shows a decision tree that evaluates the smallest of three numbers:

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

* **K-Neighbour Regressor**

K nearest neighbours is a simple algorithm that stores all available cases and predict the numerical target based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970’s as a non-parametric technique.

A simple implementation of KNN regression is to calculate the average of the numerical target of the K nearest neighbours.  Another approach uses an inverse distance weighted average of the K nearest neighbours. KNN regression uses the same distance functions as KNN classification.

* **LassoRegressor**

**Lasso regression** is a type of [linear regression](https://www.statisticshowto.com/probability-and-statistics/regression-analysis/find-a-linear-regression-equation/)that uses [shrinkage](https://www.statisticshowto.com/shrinkage-estimator/). Shrinkage is where data values are shrunk towards a central point, like the [mean](https://www.statisticshowto.com/mean/). The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of [multi-collinearity](https://www.statisticshowto.com/multicollinearity/) or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

The acronym “LASSO” stands for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator.

* **Ridge Regressor**

Ridge Regression is a technique for analysing multiple regression data that suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. It is hoped that the net effect will be to give estimates that are more reliable.

* **ElasticRegressor**

Elastic net linear regression uses the penalties from both the lasso and ridge techniques to regularize regression models. The technique combines both the [lasso](https://corporatefinanceinstitute.com/resources/knowledge/other/lasso/) and ridge regression methods by learning from their shortcomings to improve on the regularization of statistical models.

* **Gradient Boosting Regressor**

Gradient boosting is a type of machine learning boosting. It relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. The key idea is to set the target outcomes for this next model in order to minimize the error.

The name gradient boosting arises because target outcomes for each case are set based on the gradient of the error with respect to the prediction. Each new model takes a step in the direction that minimizes prediction error, in the space of possible predictions for each training case.

* **AdaBoost Regressor**

[AdaBoost](https://en.wikipedia.org/wiki/AdaBoost), short for “Adaptive Boosting,” is a boosting ensemble machine learning algorithm, and was one of the first successful boosting approaches.

AdaBoost combines the predictions from short one-level decision trees, called decision stumps, although other algorithms can also be used. Decision stump algorithms are used as the AdaBoost algorithm seeks to use many weak models and correct their predictions by adding additional weak models.

The training algorithm involves starting with one decision tree, finding those examples in the training dataset that were misclassified, and adding more weight to those examples. Another tree is trained on the same data, although now weighted by the misclassification errors. This process is repeated until a desired number of trees are added.

* **Random Forest Regressor**

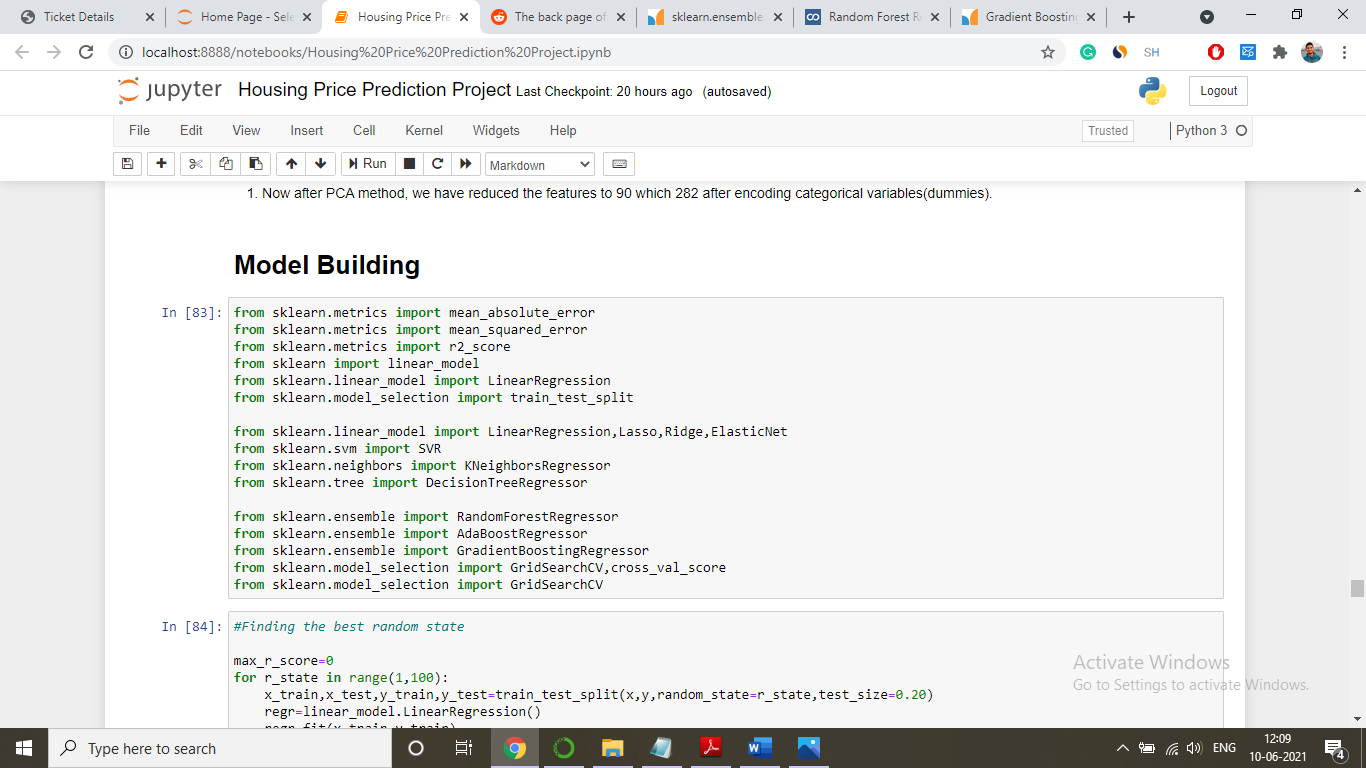
**Random Forest Regression** is a supervised learning algorithm that uses **ensemble learning** method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees.

* **Run and Evaluate selected models**

We have tried to fit different algorithms on our dataset and their description and snapshots are pasted below:

* + Applying different algorithms (Code)



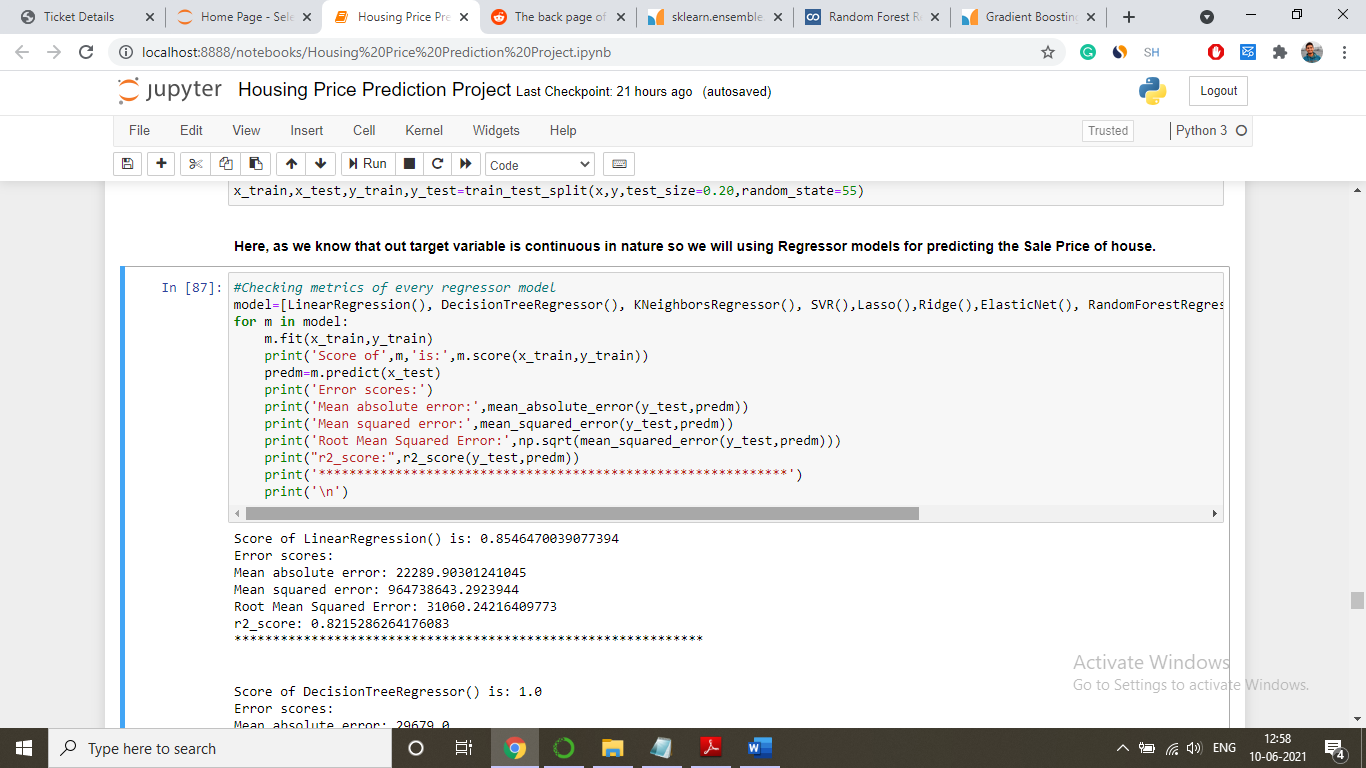


Figure 3: Using Different Models and Ensemble Techniques

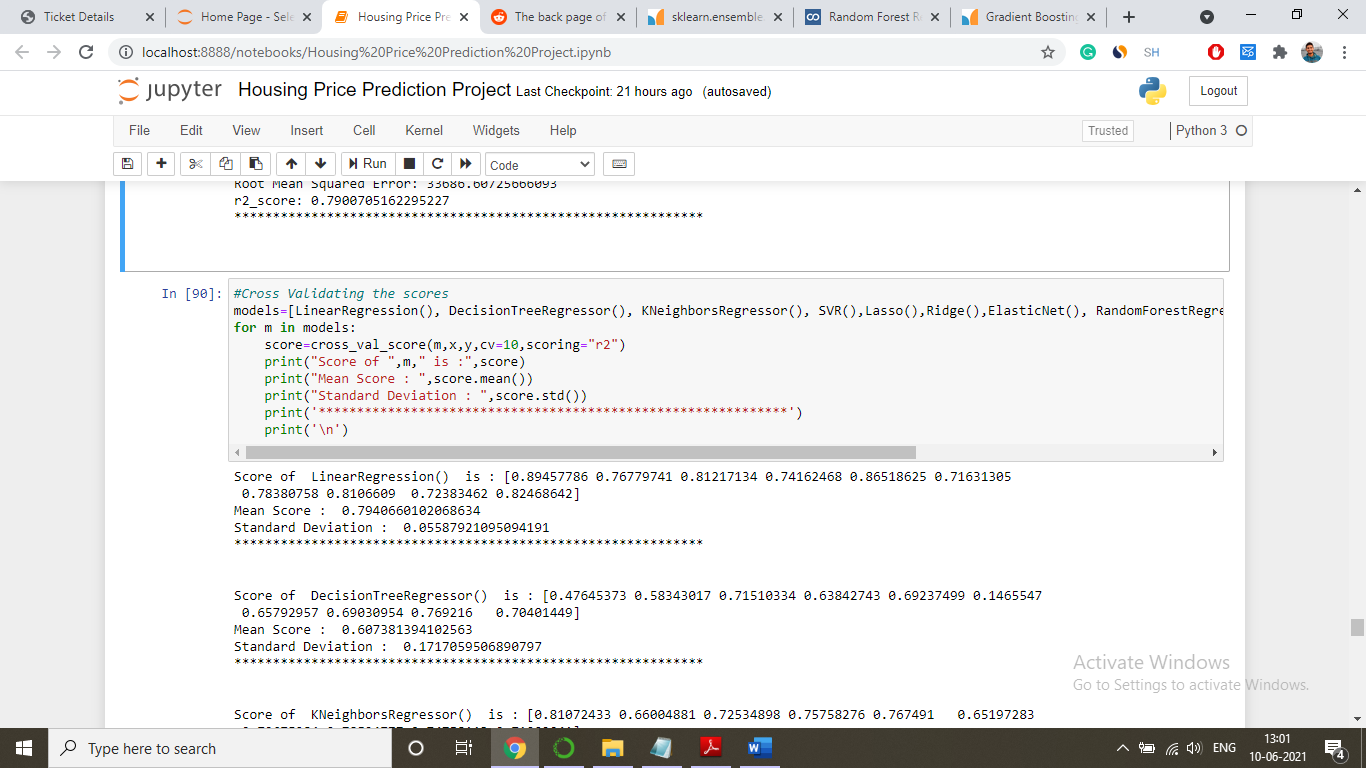


Figure 4: Cross Validating all the models

* + Output (Scores)

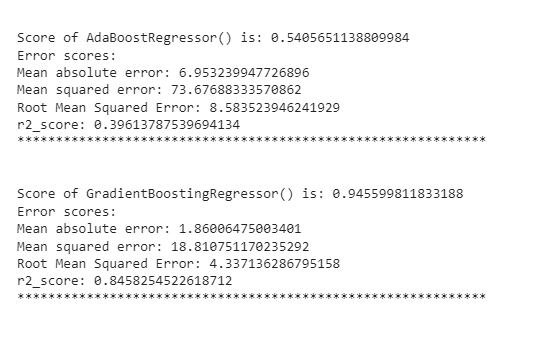
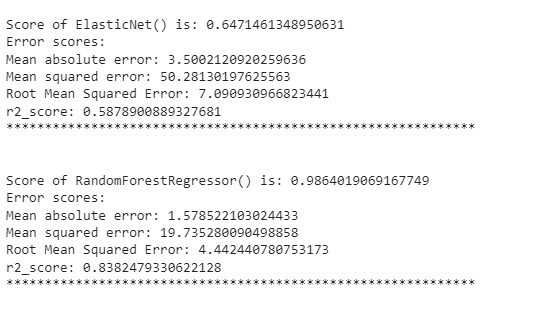
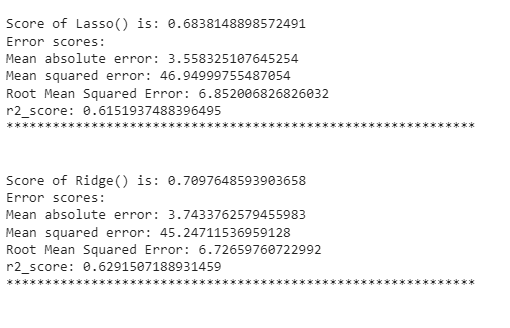
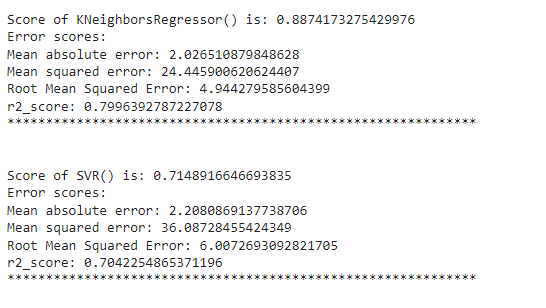
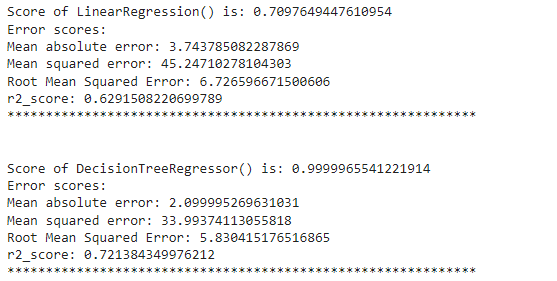


Figure 6:Scores of different models

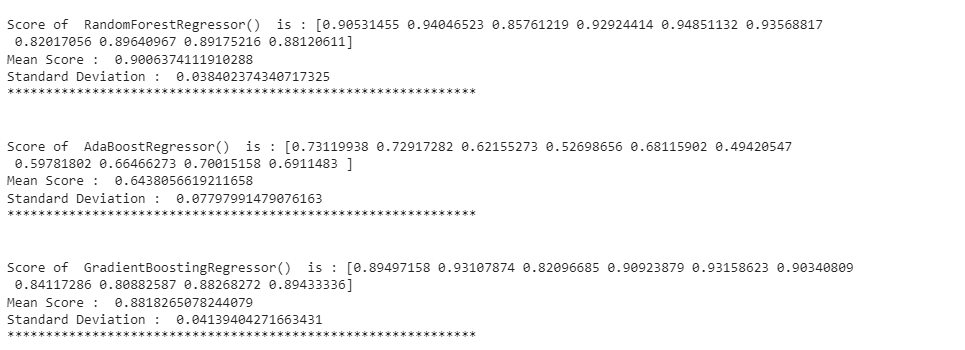
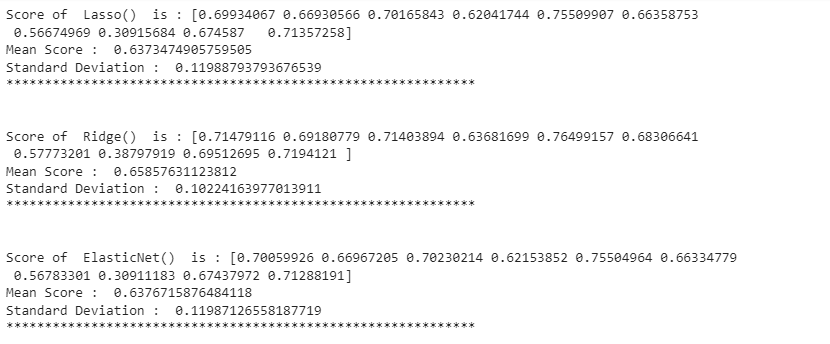
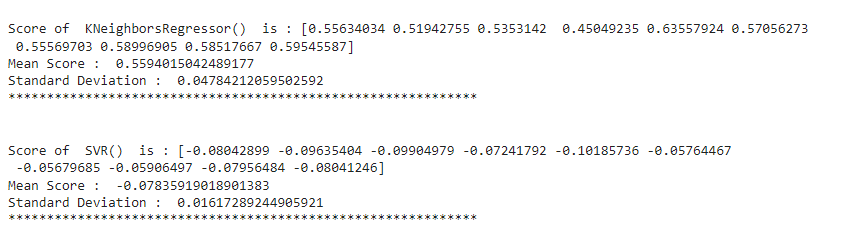
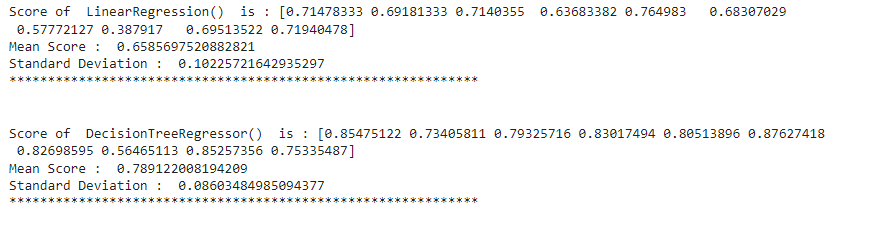


Figure 7: Cross Validation Scores of different models

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

When building and optimizing your regression model, measuring how accurately it predicts your expected outcome is crucial. However, this metric alone is never the entire story, as it can still offer misleading results. That's where these additional performance evaluations come into play to help tease out more meaning from your model.

The various metrics used to evaluate our model used in this analysis are:

* **r2 Score**

R Square measures how much variability in dependent variable can be explained by the model. It is the square of the Correlation Coefficient(R) and that is why it is called R Square.

R Square is calculated by the sum of squared of prediction error divided by the total sum of the square which replaces the calculated prediction with mean. R Square value is between 0 to 1 and a bigger value indicates a better fit between prediction and actual value.

R Square is a good measure to determine how well the model fits the dependent variables**. However, it does not take into consideration of overfitting problem.**

* **Mean Square Error**

MSE is calculated by the sum of square of prediction error which is real output minus predicted output and then divide by the number of data points. It gives you an absolute number on how much your predicted results deviate from the actual number. You cannot interpret many insights from one single result but it gives you a real number to compare against other model results and help you select the best regression model.

* **Root Mean Square Error**

Root Mean Square Error(RMSE) is the square root of MSE. It is used more commonly than MSE because firstly sometimes MSE value can be too big to compare easily. Secondly, MSE is calculated by the square of error, and thus square root brings it back to the same level of prediction error and makes it easier for interpretation.

* **Mean Absolute Error**

Mean Absolute Error(MAE) is similar to Mean Square Error(MSE). However, instead of the sum of square of error in MSE, MAE is taking the sum of the absolute value of error.

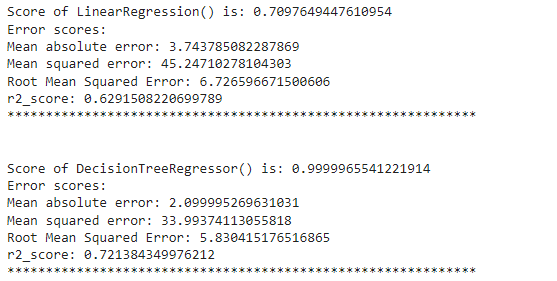


Figure 8: Different metrics of models

* **Visualizations**

In this analysis, we have used numerous graphs for getting insights of the data that was given to us so that we can interpret the data to get desired results.

* **Count Plots**

For finding count of different categorical features in the dataset.

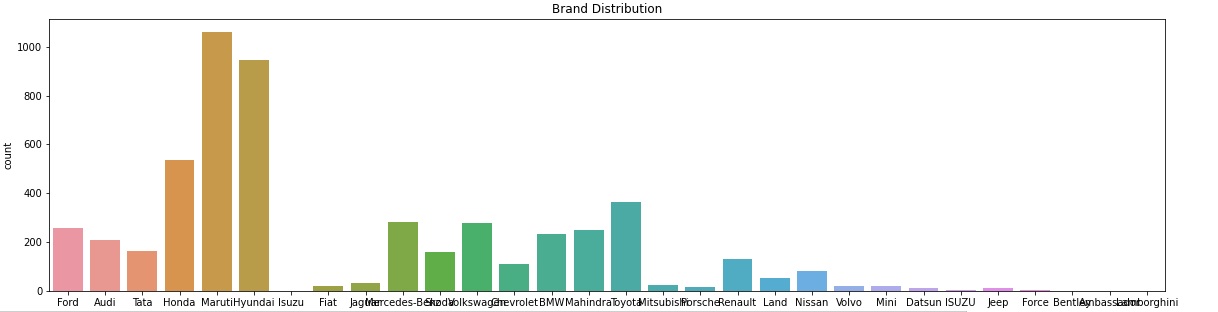
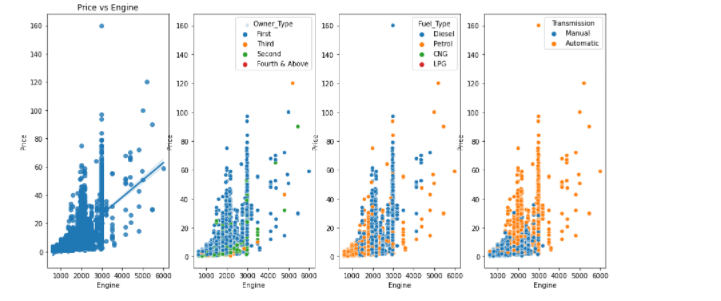
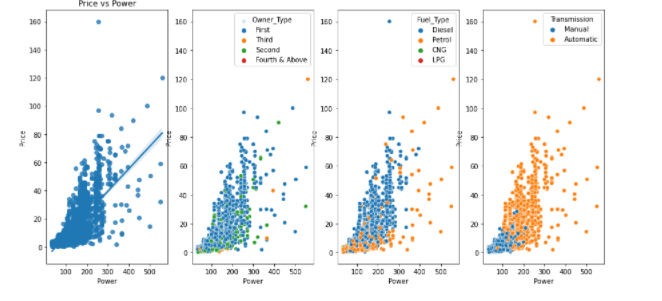


Figure 10: Countplot for different brands

* **Scatter Plot**

For finding relation between independent features and target variable (Price) of dataset.



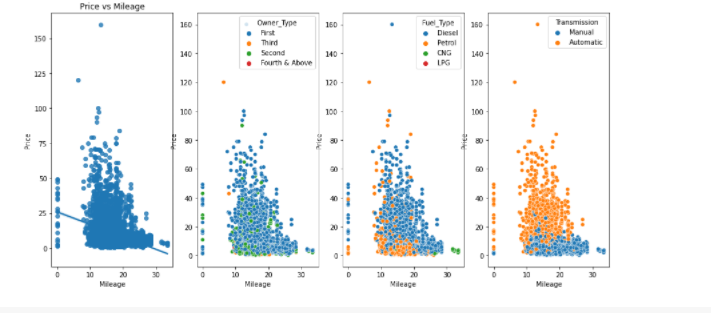


Figure 11: Barplot of Features(ExterQual, CentAir, GarageType and GarageQual) vs SalePrice

* **Swarm Plots**

For finding relation between location ,owner and price.

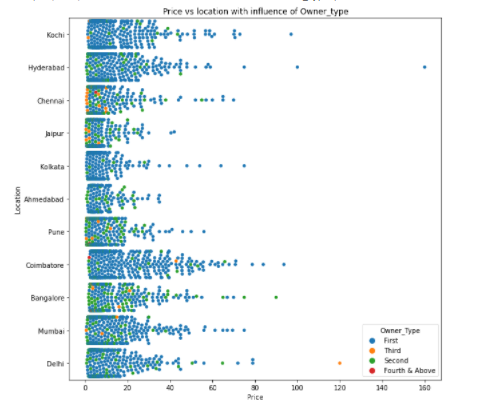
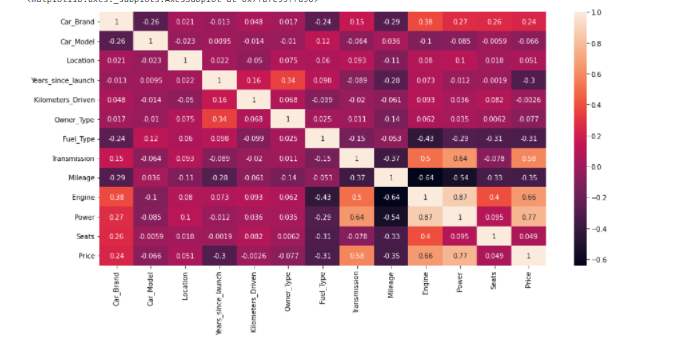


Figure 12: Scatter plot of features(OveralQual, YearBuilt and YearRemodAdd) vs Sales Price

* **Heatmaps**

Used in this analysis, for finding correlation between the independent variables and target variable.



* **Box Plots**

Used for plotting outliers present in different variables of the dataset.

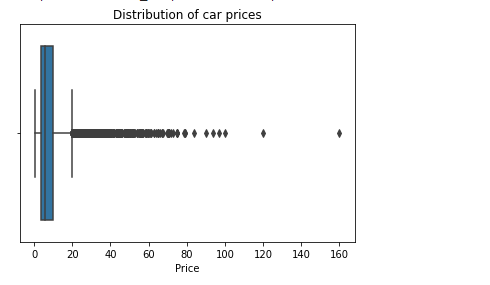


Figure 13: Finding outliers in prices

* **Interpretation of the Results**

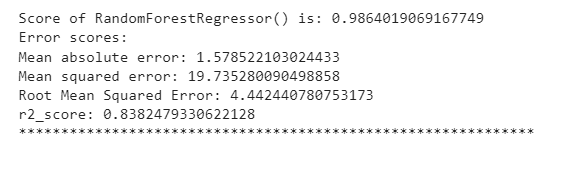
****

Figure 14: Scores for our best fitted model

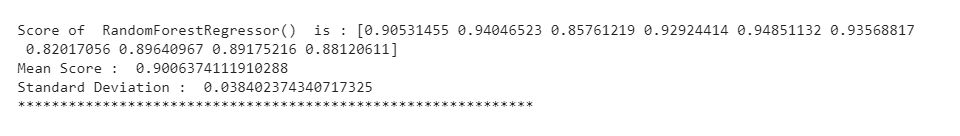
****

Figure 15: Cross Validating the scores for Random Forest Regressor Model

We got our best scores using Random ForestRegressor. By comparing it with results and metrics of other algorithm we came to a conclusion that it will be the best fit to our dataset.

**CONCLUSION**

* **Key Findings and Conclusions of the Study**
* After using different models for classification, we concluded that Random Forest Regressor was best suited to train the dataset for Car Price Prediction Project.
* The datasets hadvarious missing values .
* Some data had to be omitted from the datasets in order to remove multi-collinearity, and others reasons like irrelevancy to the dataset.
* The count of outliers in the data was quite high.
* Variables were not normally distributed.
* **Learning Outcomes of the Study in respect of Data Science**
* The visualization helped us in various ways like giving insights to the datasets that how it is arranged, correlated, how balanced it is and how much outliers it is containing.
* As the datasets was huge it created problems when we tried to manipulate it or cleanse it. As when we tried to cleanse it we had to take care that data doesn’t lose its relevancy and structure. So, we tried to remove much of the unnecessary data from the datasets that was in our reach during the given time.
* And finally, we came to know that the best algorithm used to train the machine for this the dataset is Random Forest Regressor as all the values along the metrics were highest(score) and was with least error(RMSE).
* **Limitations of this work and Scope for Future Work**
* Limitations that I encountered during this project was limited time and lack of proper processor because it would have sped up the whole project.
* As the time will pass by there will be new more efficient ML algorithms that will come to existence that can further improve the prediction but for the time being and in the near future this prediction based approach can help a lot of real estate companies in order to improve their efficiency and increase their profits.