

Car Price Prediction

Submitted by:

RAJ ARYAN BANKA

**ACKNOWLEDGMENT**

I would like to express my gratitude towards the team at DataTrained who helped me to understand the basic concepts required to analyse, interpret and model the data for the best possible result. The various machine learning concepts that they taught helped me in the realisation of this project.

I would also like to express my gratitude towards https://www.cardekho.com/ as the dataset for this project was obtained through their web page.

I took inspiration from various research papers and materials, they are mentioned as following:

- Paper by Sunil Ray on 7 Regression Techniques you should know! - this paper helped me to differentiate between the various machine learning regression models and which models would perform well with my dataset

- Paper by SmritiS on Regression Model Scoring Metrics – as this paper describes which metrics should be used to check the accuracy of a model’s prediction. This was very essential to justify the result.

I would also like to thank the team at Flip Robo since, they provided with a clear and concise definition for this project. They also taught me about various Data Mining techniques that came in handy while collecting data for this project.

**INTRODUCTION**

* 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.

This project has implication in the real world too since, there is an increasing demand for used cars in todays market. Since, the covid19 pandemic started we’ve seen a rise in the purchase of consumer goods. Used cars have are in high demand throughout the nation. Since, the production of new cars has slowed down more and more people are looking to buy cars that have been tried and tested and have survived the test for longevity of a vehicle.

* Conceptual Background of the Domain Problem

This projects uses a lot of Data Mining Concepts and needs a good understanding of Regression Models in Machine Learning. I’ve used selenium and chromium drivers to visit pages and search for the page elements. Therefore, a decent understanding of Selenium is required to understand the various methods and library used to obtain data.

The obtained data has many inconsistency. Some features contain outliers and physically impossible data which can be managed using feature engineering. Also, the data needs normalization for some features and contains multicolinearity betweeen a few features as well. Keeping that in mind we need to perform data transformation and data engineering.

Since, this is a regression problem. We need to use the appropriate metric for this algorithm. Therefore, I’ve use r2 score to check the fitting accuracy of the model and Mean Absolute Error (MAE) as the metric to calculate the error between actual values and predicted values. Some understanding of these metrics are required to understand the Model and Thereby, understand how to solve the Domain Problem using machine learning algorithms.

* Review of Literature

There have been a lot of machine learning projects conducted on car price prediction. However, the Price Prediction for used cars have been few to say the least. Along with the variables used to calculate car prices, there are some additional variables that needs to considered when we talk about Price Prediction for used cars. A few common examples of these variables are Driven Kilometers and Number of owners. These variables can help determine how cars with similar specifications can drastically variate in price. This paper takes some inspiration from the work done on Car Price Prediction.

As seen from the statistical market analysis. The demand for used cars and especially premium used cars has been rising globally. This project gives an in-depth understanding of which factors affect the prices of used cars the most and how consumers can get an estimated price for any used car using Machine Learning and Data Mining.

The dataset for this project has been taken from only a single website i.e, carsdekho.com due to time constraint as it can take a lot of time to collect data from multiple resources even with automation. However, I was able to collect data for approximately 16,000 used cars. This is a pretty decent data sample as this can help the model to achieve good accuracy with low error.

During the model selection phase I had to do an in-depth research on evaluation metrics. As it is common to choose r2 score to check the fit of the model and RMSE score to check for the average error. Since, RMSE has the benefit of penalizing large errors more so can be more appropriate in some cases, for example, if being off by 10 is more than twice as bad as being off by 5. This applies to our problem as being off by 1,00,000 is more than twice as bad as being off by 50,000. It is relative how close anyone wants their value to be. But, it is a general consensus that being off by 50,000 is more acceptable than being off 1,00,000.

To simplify this issue further I have normalized the target feature (“Price”) using Numpy library’s log function. This helps the model to fit the data with better accuracy and gives an relatively even lower RMSE value. This normalization does not affect the results of the final prediction as it can be converted back to the original values using Numpy’s exp() or exponential function.

* Motivation for the Problem Undertaken

- The main objective behind this project is analyse the used cars market and provide consumers with an accurate prediction for any used car that they wish to purchase.

- It also helps sellers to get a rough estimation of what markup prices they can expect for their cars.

- The foundings of these project can also be used by websites like cardekho.com, carwale.com etc to analyze the car specifications and features to provide an estimate price.

- The findings of this project can help to create automated systems that uses the power of Machine Learning to make predictions for millions of used cars globally.

- Car Enthusiasts can also benefit from this project as they are the people who would want to buy a used car which doesn’t have it’s price listed. This model gives a better understanding of what the prices could be.

- As we delve towards a more autonomous future this project might help in reducing the time taken in analysing by a human and use the power of Machine Learning to make the process more efficient and fast meanwhile, learning from the new data.

**Analytical Problem Framing**

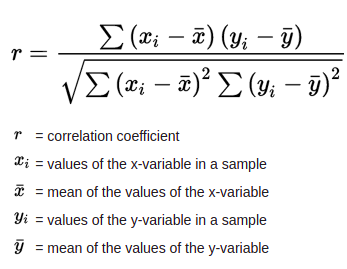
* Mathematical/ Analytical Modeling of the Problem

Describe the mathematical, statistical and analytics modelling done during this project along with the proper justification.

I used mathematical modelling function such as description and skew functions in the pandas library. The description function gives a statistical analysis for all the features, statistical analysis such as minimum value, maximum value, average, standard deviation, interquartile values and count. These values helps to analyse how the data is distributed. The skew function allows to measure the degree of skewness in a features distribution.

EDA or Exploratory Data Analysis is used to find the statistical representation of data. Using univariate analysis we find the distribution of various features. We use bivariate analysis to see how data is distributed for the different values in the target column. Using catplot function inside seaborn library for Bi-variate Analysis. Through univariate analysis we find that the data is highly skewed and needs to be transformed and normalized. Using corr function to find the correlation between the target variable and feature can help determine which features are useful and also which features have multicolinearity and negative correlation as they can prevent from getting better accuracy. The correlation function can be given as:-





* Data Sources and their formats

The dataset for this project has been mined using Web Scraping tools like Selenium. The used car data has been taken from [www.cardekho.com](http://www.cardekho.com/). CarDekho.com is India's leading car search venture that helps users buy cars that are right for them. Its website and app carry rich automotive content such as expert reviews, detailed specs and prices, comparisons as well as videos and pictures of all car brands and models available in India. The company has tie-ups with many auto manufacturers, more than 4000 car dealers and numerous financial institutions to facilitate the purchase of vehicles.

This dataset contains some common features found in car price prediction datasets such as Brand, Model, Manufacturing Year, Engine capacity, Transmission type, Fuel Type, Power, Torque and Price. It also contains some additional features such as distance driven in the car, number of owners the car had and location which can help the model to give better accuracy by correlating these features with the Target variable i.e, Price. For example, it is a known fact that the Price of a car is inversely proportional to the number of Owners it has. As the car is sold from one person to another the value of the car decreases.

Data Features in the dataset can be described as following:-

i) Brand – This is a categorical feature as there can be only a finite amount of brands of car.

ii) Model – Cars from the same brand can have many different models ranging between a vast Price gap. Therefore, model feature can help differentiate between various cars of the same brand. This is also a categorical feature as a car can have only a finite number of models.

iii) Manufacturing Yr – This feature tells the year in which the car was manufactured and as we know generally the price of a car decreases over time. Therefore, the older a car is the lower will be it’s Price. It is important to note that this theory is applicable to cars belonging to same brand and model.

iv) Engine – This feature is a continuous feature that tells the volume of a car’s engine. It is one of the most important specification of car. The engine volume is measured in CC.

v) Transmission – This is a categorical feature as a car can have only two type of transmission either Automatic or Manual. Generally two cars with same specification but different transmission differ slightly in Price as a car with Automatic Transmission can cost slightly more than a car with Manual Transmission.

vi) Driven\_kms – This is a continuous feature. It is inversely proportional to the price of a car. This feature tells us about how much distance has a car been driven i.e, how much a car has been used. As the value of a car decreases the more it is driven or used.

vii) Fuel Type – This a categorical feature. Fuel Type can vary from car to car but it can be divided into 5 categories. The most common fuel types are Petrol and Diesel.

viii) No. Owners – This too is a categorical feature. As number of owners are divided into roughly 5 categories as seen from the dataset.

ix) Mileage – This is a continuous feature. Now, when talking about mileage there are many variables that can be used to determine the mileage of a car. Engine, Power, Torque,Transmission etc can play an important factor in determining mileage. It can vary from Brand to Brand as well. As mileage is efficiency of a car to travel maximum distance by burning least amount of fuel. A car’s mileage is measured

in kmpl or kilometer per litre.

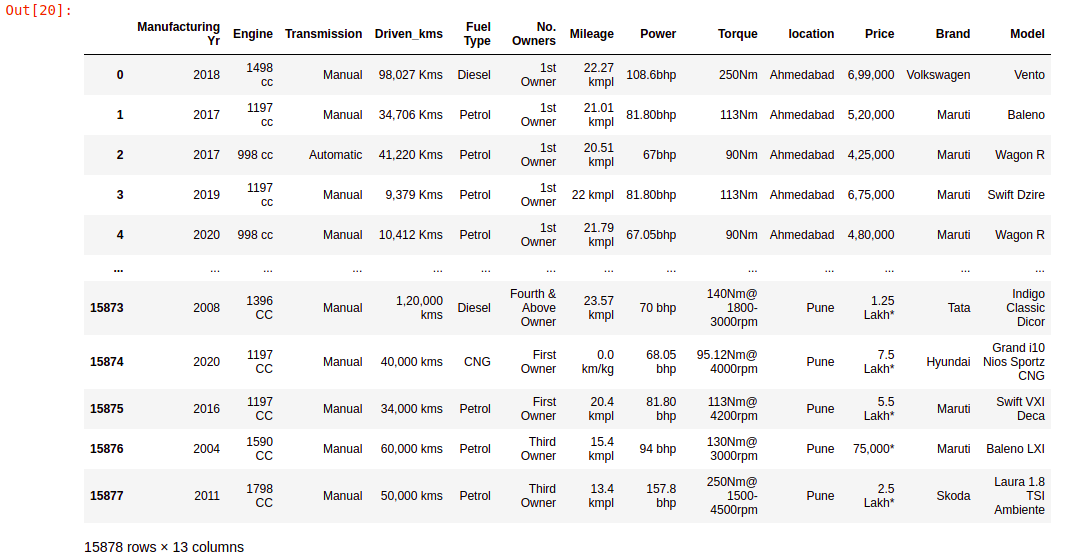
x) Power – This is a continuous feature. The power of a car is measured in Bhp or Brake Horse Power. This feature determines how many brake horse power a car generate. Usually the power is directly proportional to Price of a car.

xi) Torque – This is also a continuous feature. It is the force by which a car moves certain distance. Torque is measured in pound-feet(lb-ft) or newton-meters (Nm).

xii) Location – This is a categorical feature. The data has been taken from 12 cities in India. This feature can help the model to differentiate which cities have the same car listed at a higher price. Therefore, the model can adjust the price of a car accordingly.

xiii) Price – This is the target feature and it is continuous. The feature contains values between a large range therefore, it may need to be scaled down. This features tells us the price of a car in INR.

The dataset is represented in this format:-



* Data Preprocessing Done

What were the steps followed for the cleaning of the data? What were the assumptions done and what were the next actions steps over that?

The Data Preprocessing done in this project can be divided into two phases.

1. Data Collection Phase
2. Model building Phase

**Data Collection Phase:**

I used selenium library’s webdriver module to visit cardekho.com webpage. From there the data collection process can be divided into the following steps:-

* On the home page of the website we can see the used car navigation button. So the web element of that navigation button can be used to get the used car data.
* On clicking it we see a dialog box with multiple metropolitan cities to choose from we can find the web elements for all of these major cities.
* Then using the get\_attributes function we can find the urls of the pages for these locations.
* It is seen that there are many used cars listed on this page. However, all of the cars listed aren’t shown at first. Once, we scroll to the last list item in this page more item load and increase page length. Therefore, I used selenium’s **ActionChains** module to scroll down to the last element until the last element for current list items is same as the last element for previous list items.
* Once, the last list item has been reached I collected the urls of all the list items.
* Then, on visiting a few of these pages it is seen that the pages are divided into two different formats. Therefore, a nested try block with two conditions can be set for each features according to the page format.
* The most common features available for different cars are: Brand Name, Model Name, Car Price, Manufacturing Yr, Transmission type. However, few other specification are hidden and they can only be accessed by hitting the **view specification button.** So we need to find the web element for this button to collect data for other attributes such as Engine volume, Driven Kilometres, Number of Owners, Mileage, Power and Torque.
* Therefore, we will find the web elements for all of the above features and add values of these web elements to their respective lists.
* We repeat the above step for all the used cars pages.
* After collecting the data for all the used cars. We will create a dataset **df** with all the mentioned features in it using python’s pandas library.
* Another column is added for car’s location.
* Now all the values given in the page as “NaN” can be replaced with “null” value using python’s numpy library.

**Model Building Phase**

**Data Cleaning**

The dataset needs to be cleaned as all units of needs to be converted to standard form, they need to be converted to numerical data and the units needs to be removed from them. Also the data inside categorical column needs to be encoded and changed to numerical values.

On using pandas isnull function it is seen that there are three entries with null values present in them. We will drop these entries and reset the index of dataset. Now, since the Brand\_Model feature contains unnecessary data such as year or blank spaces we will check for them and keep the necessary information about Brand and Model only.

Then we need to separate the Brand Names from Models. It is seen that only the brands Land Rover and Aston Martin consists of two words. All other Brands in this dataset consist of only one word. The other half of the Brand\_Model element consists the Model data and we will separate it from the Brands.

The elements in Manufacturing Year element consists of year value in string which needs to be converted to integer values. Engine column contains volume of an engine followed by the unit of measurement cc or CC (Cubic Centimeter). We will remove the unit of measurement and convert the string numericals to integer numerical values. The same can be applied for columns Driven\_kms and Power. In these columns the units of measurements are kms (or Kms) and bhp ( or Bhp) respectively.

In columns Mileage and Torque, there are different units of measurement for these values. Particularly in column mileage the units of measurement are very different and cannot to converted to each other so, we need to calculate the price of driving a kilometer to get the mileage of a vehicle. However, it should be noted that the price of these commodities changes everyday. In column Torque the data is divided into two units of measurements ‘NM’ and ‘KGM’ written in multiple orientation in small and capital letters. KGM needs to be converted to Nm to keep a standard unit of measurement.

Column “No. Owners” contains two representation for no of owners of a car. For example, 1st Owner and First Owner. Therefore, we will standardize it for 1st Owner, 2nd Owner and 3rd Owner.

Columns Transmission contain some physically impossible data other than Manual or Automatic. Since, the transmission of a car can either be Manual of Automatic we will drop all other entries other than the one’s with these values. The same logic can be applied to column Fuel Type. As only Petrol, Diesel, LPG and CNG are the possible values.

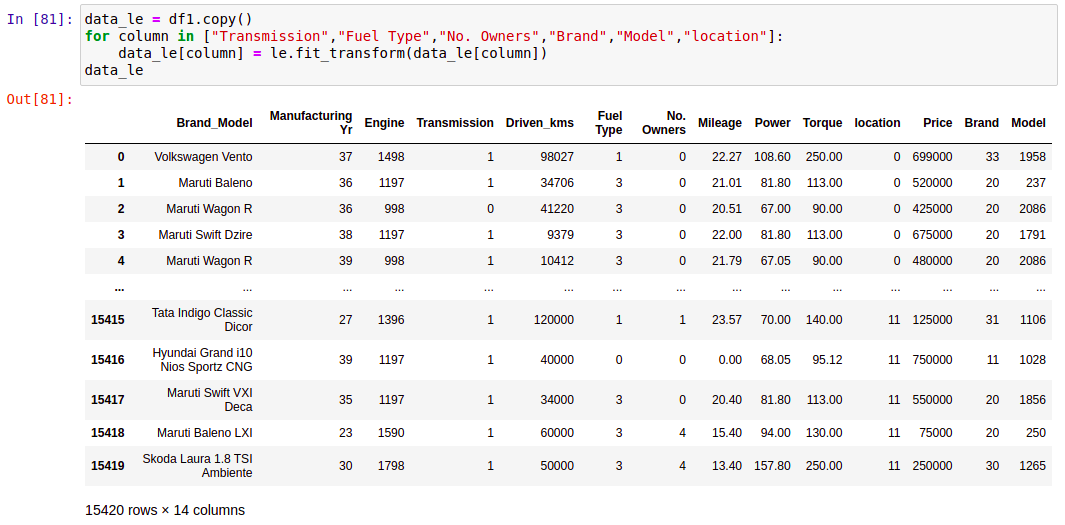
Finally all the columns with continuous data along with Price is converted to numerical using pandas to\_numeric function.

**Feature Engineering**

We can normalize the distribution in column Manufacturing Yr by substracting all the year values from the minimun year value. This will help the model fit the data better.

**Encoding**

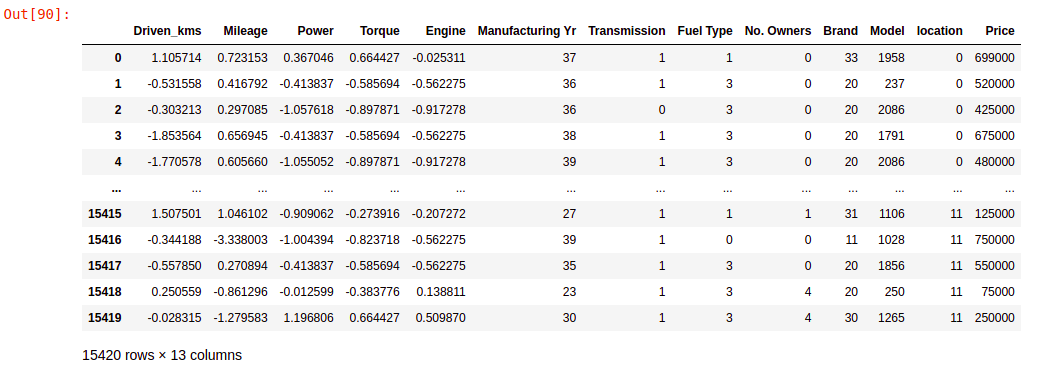
Columns Brand, Model, Transmission, Fuel Type, No. Owners, Location have categorical values. Therefore, they are converted to numerical values using LabelEncoder from sklearn.preprocessing library.



**Transforming**

All the continuous features differ vastly in ranges. We will transform all of these using sklearn.preprocessing library’s PowerTransformer function.

The features are converted to the following format.



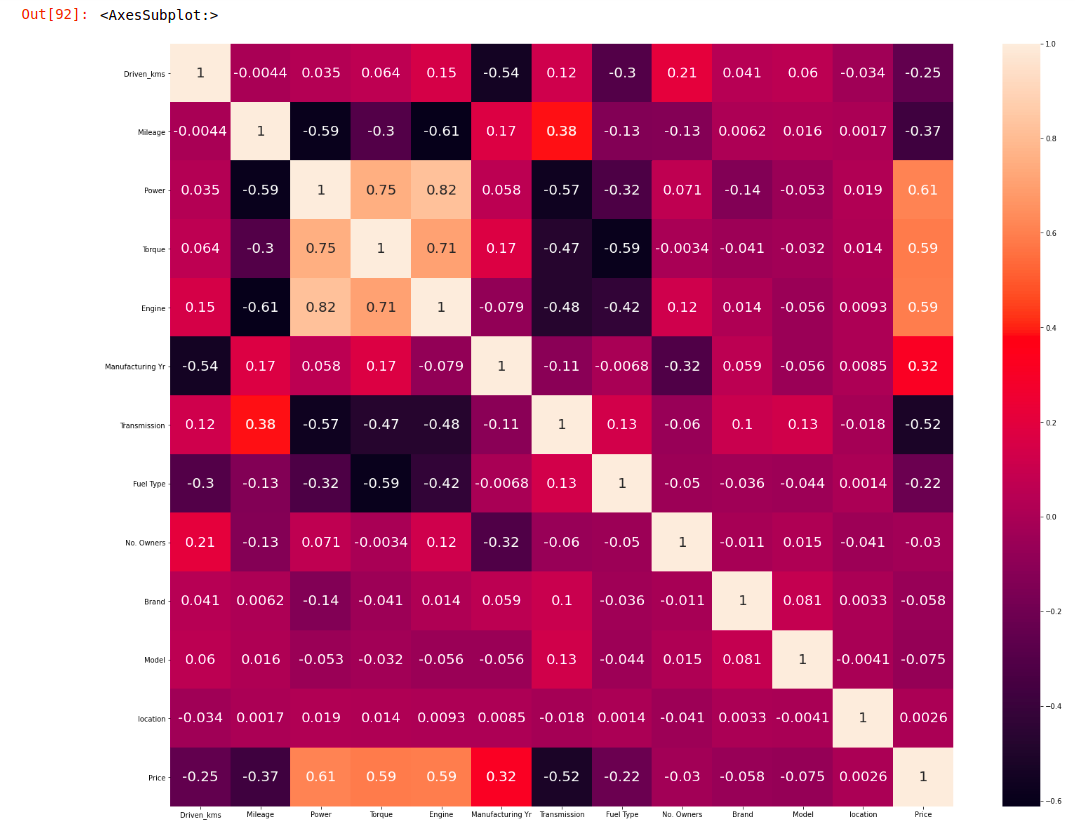
* Data Inputs- Logic- Output Relationships

After preprocessing the dataset, we have a dataset with categorical and continuous columns. From the heat map of correlation between features in

the dataset it is seen that there are correlation between independent

variables and there is correlation between independent variables and

dependent variable it can seen in the following diagram:



It is seen that columns Power, Torque and Engine have high correlation

between them. That means, that these features have multicolinearity

between them. Since, these columns have a high correlation with the

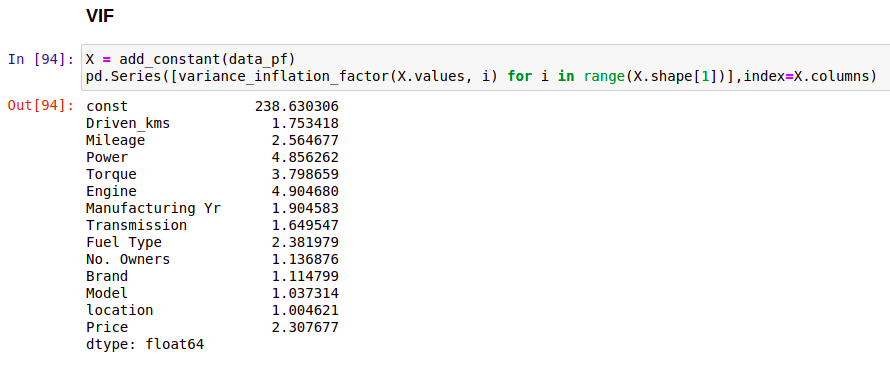
target variable we need to find whether these columns fall within an

acceptable range of multicolinearity. To measure this we use a statistical

technique called the **variance inflation factor (VIF).** We add a constant

feature to the dataset, to account for the difference in scale between all the

features. On applying VIF we get the following values for each feature.



It can be seen that the VIF score for columns with multicolinearity is

much closer to the acceptable range of VIF values. Therefore, they can be

left untreated for now. However, it should be noted that there are various

feature engineering methods that can be applied to these columns to get a

lower VIF score for these columns.

Columns Driven\_kms, Mileage, Transmission, Fuel Type have negative

correlation with the target variable. This means that these features are

inversely correlated with the target variable and can have a major

influence on the target variable. The other columns No. Owners, Brand,

Model, Location have small amounts of correlation with the target

feature. We should keep these columns as they might help in small

amounts of error correction. The range of Prices in our dataset is very

large therefore, even small amount of correlation can help reduce error.

* Hardware and Software Requirements and Tools Used

Listing down the hardware and software requirements along with the tools, libraries and packages used. Describe all the software tools used along with a detailed description of tasks done with those tools.

**Hardware**

A system with these configuration.

* Intel i5 processor,
* 8 GB RAM,
* 4 GB Graphics Card.

**Software**

* Linux OS (Any operating system that runs Anaconda can be used)
* Jupyter Notebook (Anaconda),
* Chrome Browser,
* Chromium Browser,
* LibreOffice.

The libraries and packages used in this project are:

* pandas
* numpy
* selenium
* time
* from selenium import webdriver
* from selenium.common.exceptions import NoSuchElementException
* import time
* from selenium.webdriver.common.action\_chains import ActionChains
* from selenium.webdriver.common.keys import Keys
* matplotlib.pyplot
* seaborn
* from sklearn.preprocessing import
  + LabelEncoder
  + PowerTransformer
* from scipy import stats
* from sklearn.model\_selection import
  + train\_test\_split
  + GridSearchCV
  + RandomizedSearchCV
  + Kfold
  + cross\_val\_score
* from sklearn.metrics import
  + mean\_squared\_error
  + r2\_score
  + accuracy\_score
  + mean\_absolute\_error
* from sklearn.linear\_model import
  + LinearRegression
  + Ridge
  + Lasso
* from sklearn.tree import DecisionTreeRegressor
* from sklearn.neighbors import KNeighborsRegressor
* from sklearn.ensemble import
  + RandomForestRegressor
  + AdaBoostRegressor
  + GradientBoostingRegressor
* from xgboost import XGBRegressor

**Model/s Development and Evaluation**

* Identification of possible problem-solving approaches (methods)

Exploratory Data Analysis (EDA) is implemented to give statiscal analysis of the dataset. It can be divided into two phases Univariate Analysis and Bivariate Analysis.

On implementing univariate analysis it is seen that columns Driven\_kms, Engine and Price have right skewness in their distribution. All other continuous columns have normal distribution. This skewness can reduce the accuracy of model prediction. The categorical columns are analysed by countplot. In column Manufacturing Yr we can see that the dataset has low sample size for old cars. As seen from the graph below we can say that the cars which are older than 2 years but not older than 11 years have the highest concentration in the dataset. Columns Transmission, Fuel Type, No. Owners have class imbalance in them therefore, these feature will have high variance.

Outliers can be seen in the continuous columns in this dataset. Removing outliers can solve the problem of skewness. We use seaborn’s boxplot function can be used to show the outliers in all the continuous columns.

It is seen that the target feature, Price has continuous data. Therefore, machine learning regression models can be used on the dataset to predict the values in the target feature.

The metrics used to find the model accuracy are **r2 score and RMSE value.** We use r2 or r-squared score to measure the goodness of fit of a model. It is important to know that the model is able to learn from the dataset. Higher value of r2 indicates that the model is able to learn better. On the other hand RMSE value measures the error in prediction. It is useful for dataset with a vast range of values. As RMSE score punishes the algorithm that gives large values of error. Even, if the average error is lower than other models. We cannot have a model where large amounts of error is given for some samples.

* Testing of Identified Approaches (Algorithms)

Linear Model such as Linear Regression, Lasso and Ridge regression can be used to solve the multicolinearity. Although, the dataset contains only a small amount of multicolinearity it could be beneficial to use these algorithms so that the dataset can fit better on the model.

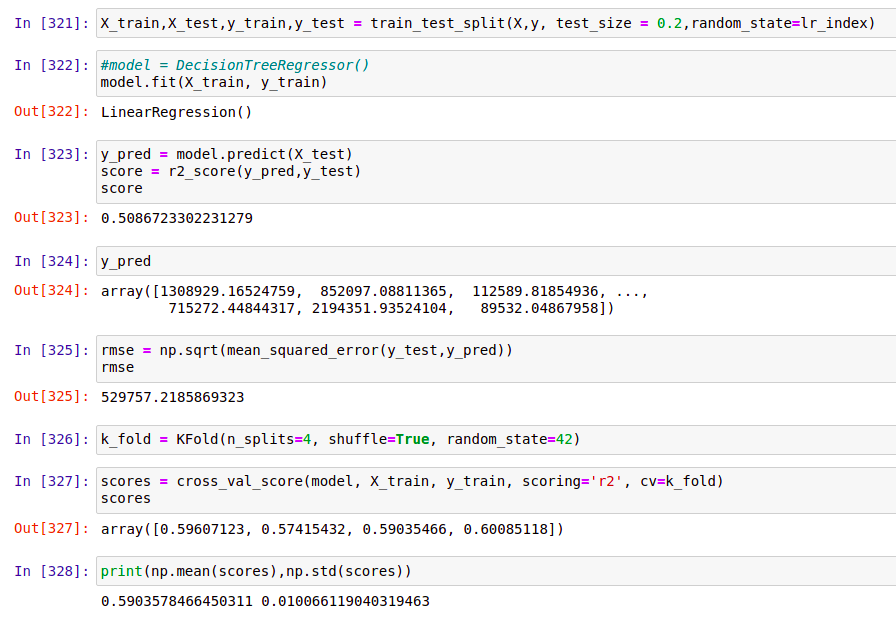
Supervised Learning Algorithms such as Decision Tree Regressor and KNN Regressor can be used as well. Decision Tree algorithms create relationships between features and splits the dataset into multiple subsamples. This further helps the to model to choose the values with least amount of error and increases the purity of the prediction. The KNN model works by finding the distances between a query and all the examples in the data. It predicts based on the specified number of neighbors (K) closest to the query.

Finally, ensemble techniques can be used as well. They are classfied into two categories Bagging and Boosting. Bagging is a technique for reducing prediction variance by producing additional data for training from a dataset by combining repetitions with combinations to create multi-sets of the original data. Boosting is an iterative strategy for adjusting an observation's weight based on the previous classification. For bagging I’ve used Random Forest Classifier and for boosting techniques I’ve used AdaBoost, XGBRegressor and GaussianNB. Random Forest algorithm consists of multiple decision trees and selects the best output from all the decision tree to give higher accuracy than decision tree algorithm.

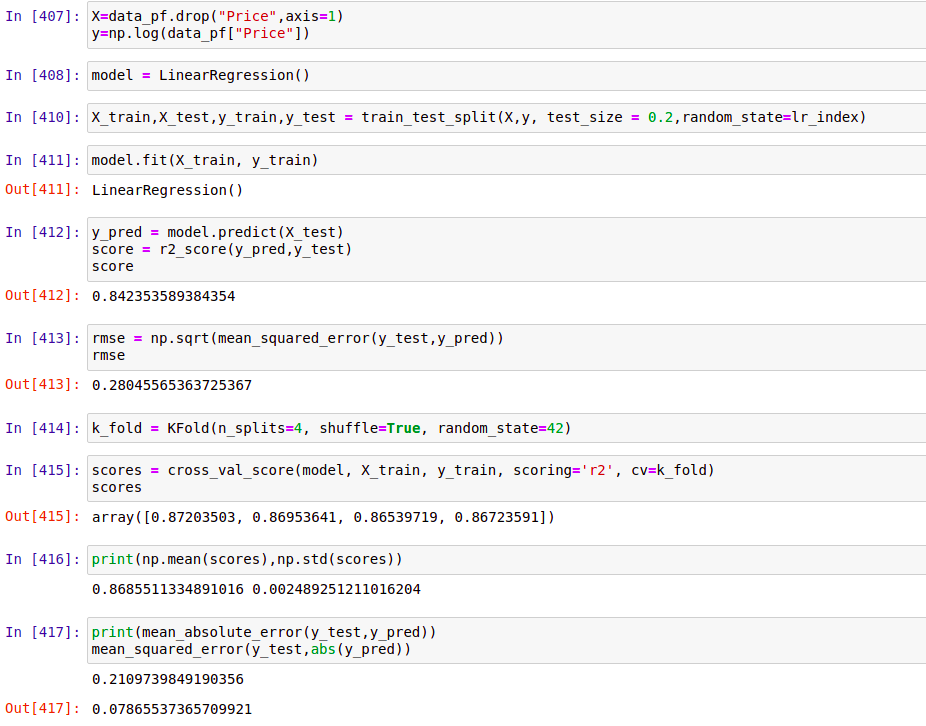
* Run and Evaluate selected models

**Linear Models**

1. Linear Regression:



We can see that on applying Linear Regression. We get a r2 score of 0.509 which is not good. The linear regression model is unable to fit the data well. This can occur due to the large range of values present in the target feature. We can improve the accuracy of this model by normalizing the target feature but it will also scale the rmse value. This can be seen below:



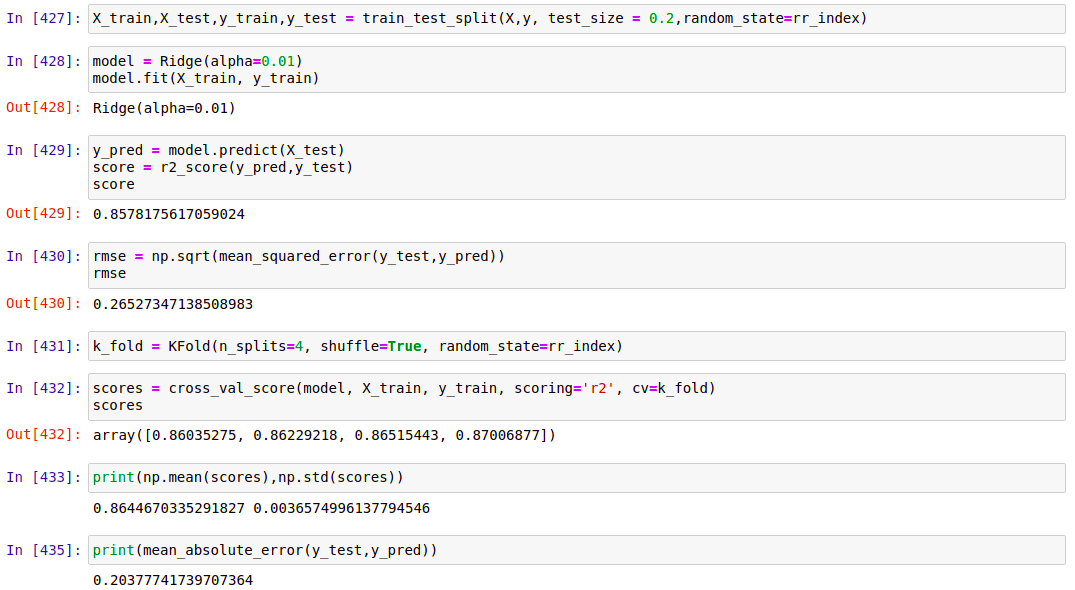
As seen from the above results on normalization the r2 score of the

model increases by a lot. This proves that normalization of target

feature can make better prediction. For linear regression model we

get r2 score of 0.842 and a rmse score of 0.28

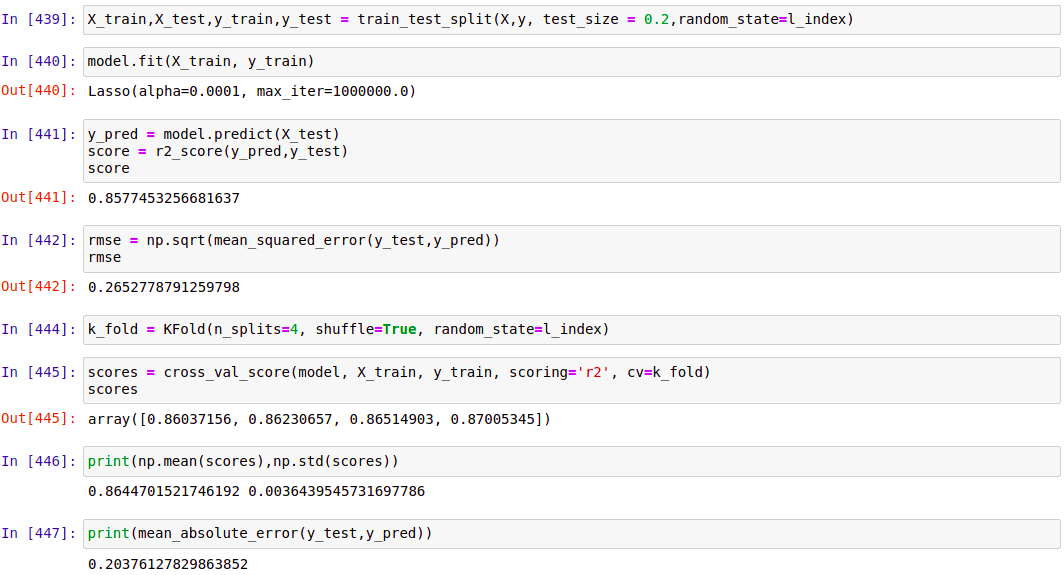
2. Ridge Regression:



We can see that on applying ridge regression on the dataset we get a r2 score of 0.8578 and a RMSE value of 0.265. This better than linear regression as this model is able to solve some amount of

multicolinearity and therefore is able to fit the dataset better and predict the values with better accuracy.

3. Lasso Regression:



We can see that on applying lasso regression on the dataset we get a r2 score of 0.8577 and a RMSE value of 0.265. This better than linear regression but similar to Ridge Regression. It can be said that

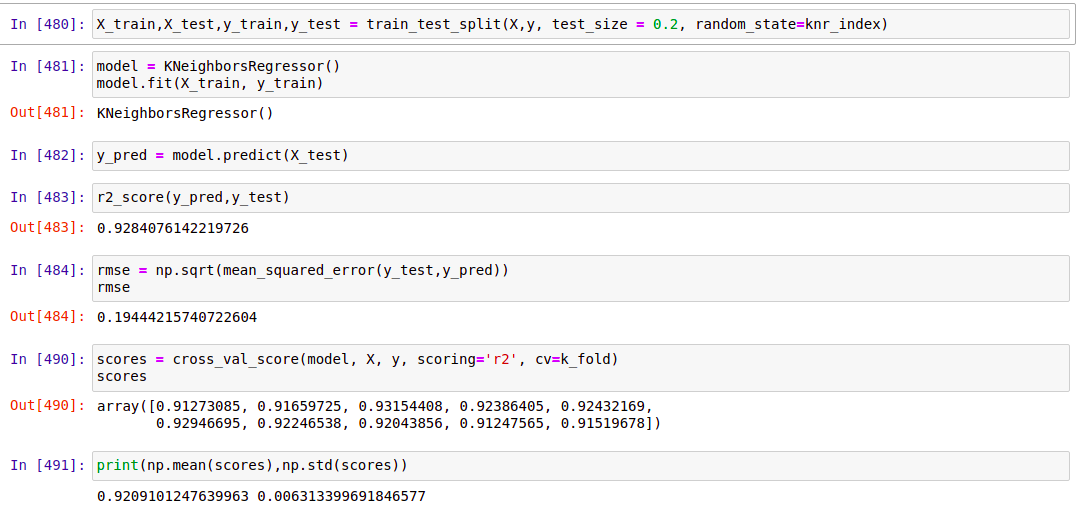
the model fits the dataset a little worse than Ridge Regression.

**Decison Tree Regression**

****

We can see that on applying Decision Tree regression on the dataset we get a r2 score of 0.958 and a RMSE value of 0.154. This is a better score than all the linear models. There is negligible amount of underfitting in this model as seen from the cross validation score. The mean cross validation score is 0.9436 with a standard deviation of 0.0057. We use cross validation metric to check for underfitting and overfitting of models.

**K-Nearest Neighbor Regression**

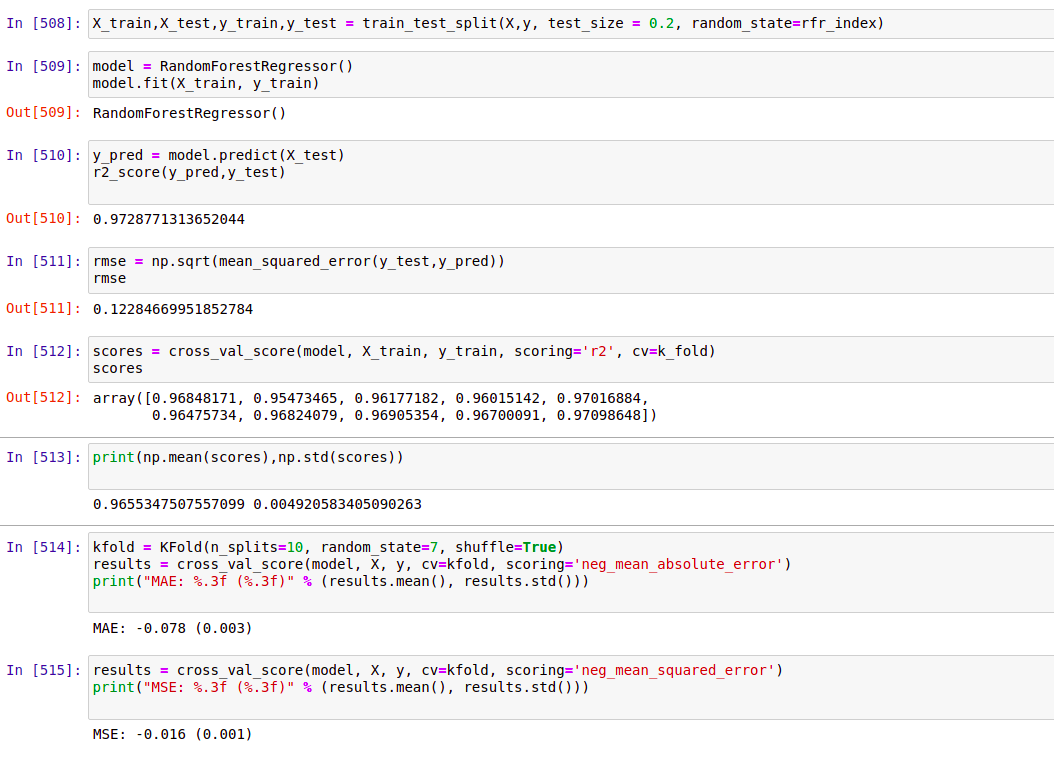


We can see that on applying KNN regression on the dataset we get a r2 score of 0.9284 and a RMSE value of 0.1944. This is a good score however it performs worse than Decision Tree model. The dataset fits well on the model. There is negligible amount of overfitting in this model as seen from the cross validation score. The mean cross validation score is 0.9209 with a standard deviation of 0.0063. Therefore we can say that the model performs well for all subsamples of dataset.

**Ensemble Techniques**

1. **Bagging**

**Random Forest Regressor**

****

We can see that on applying Random Forest regression on the dataset we get a r2 score of 0.9728 and a RMSE value of 0.1228. This is the best score yet. This is expected as Random Forest is modelled on top decision tree and it takes the best nodes with least error to make predictions. The dataset fits well on the model. There is negligible amount of overfitting in this model as seen from the r2 cross validation score. The mean cross validation score is 0.9655 with a standard deviation of 0.0049. Also the cross validation score for mean squared error is the lowest. Therefore we can say that the model performs well for all subsamples of dataset.

1. **Boosting**

Boosting algorithms used:

* AdaBoost Regressor
* XGBoost Regressor
* GradientBoosting Regressor

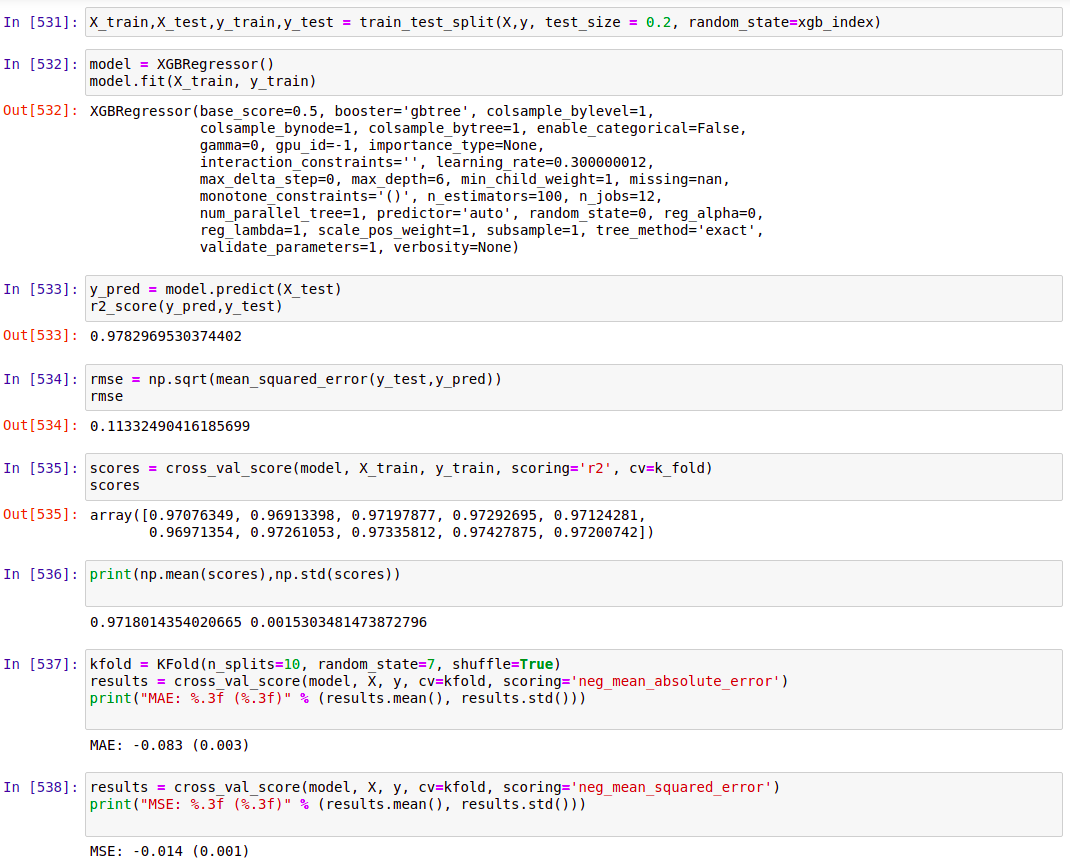
**AdaBoost Regressor**

****

We can see that on applying AdaBoost regressor on the dataset

we get a r2 score of 0.863 and a RMSE value of 0.269. This score is worse than Random Forest Regressor. There is negligible amount of overfitting in the model as seen from Cross Validation score . The mean cross validation score is 0.9329 with a standard deviation of 0.0106. Therefore we can say that although this model doesn’t fit the dataset very well we get the least amount of error from this model.

**XGBoost Regressor**

****

We can see that on applying XGBoost regression on the dataset we get a r2 score of 0.9783 and a RMSE value of 0.1133. This is the best score yet. The dataset fits well on the model. There is negligible amount of overfitting in this model as seen from the r2 cross validation score.The mean cross validation score is 0.955 with a standard deviation of 0.0112. Also the cross validation score for mean squared error is the lowest. Therefore we can say that although this model doesn’t fit the dataset very well when compared with Random Forest Regressor, we get the least amount of error from this model. The mean squared error cross validation score for this model is 0.014 with a standard deviation of 0.001 . It should be noted that our target is to achieve a model with least amount of error and so this model is the best at predicting values with minimum error.

**GradientBoosting Regressor**

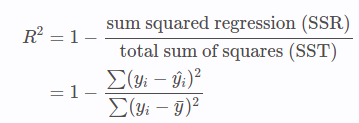
****

We can see that on applying GradientBoosting regression on the dataset we get a r2 score of 0.94032 and a RMSE value of 0.178. This is score is worse than that of XGBoost. The dataset fits well on the model. There is negligible amount of overfitting in this model as seen from the r2 cross validation score.The mean cross validation score is 0.9328 with a standard deviation of 0.0106. Also the cross validation score for mean squared error is quite high. Therefore we can say that although this model fits the dataset well when compared with AdaBoost Regressor, we get more error in this model. The mean squared error cross validation score for this model is 0.034 with a standard deviation of 0.002 .

* Key Metrics for success in solving problem under consideration

**R2 score :-**

It is important to check whether the dataset fits the model correctly. The coefficient of determination, or R2, is a measure that provides information about the goodness of fit of a model. In the context of regression it is a statistical measure of how well the regression line approximates the actual data. It is therefore important when a statistical model is used either to predict future outcomes or in the testing of hypotheses.

  
  
  
  
  
  
  
Where,

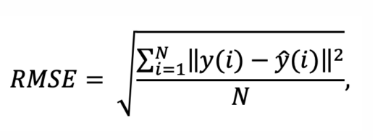
So, we can say that higher the value of R2, higher will be the goodness of fit of a model. And the better the goodness of fit for a model, the better it is at predicting the target values.

**RMSE score :-**

Root mean square error or root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions. It shows how far predictions fall from measured true values using Euclidean distance.

To compute RMSE, calculate the residual (difference between prediction and truth) for each data point, compute the norm of residual for each data point, compute the mean of residuals and take the square root of that mean. RMSE is commonly used in supervised learning applications, as RMSE uses and needs true measurements at each predicted data point.

Root mean square error can be expressed as



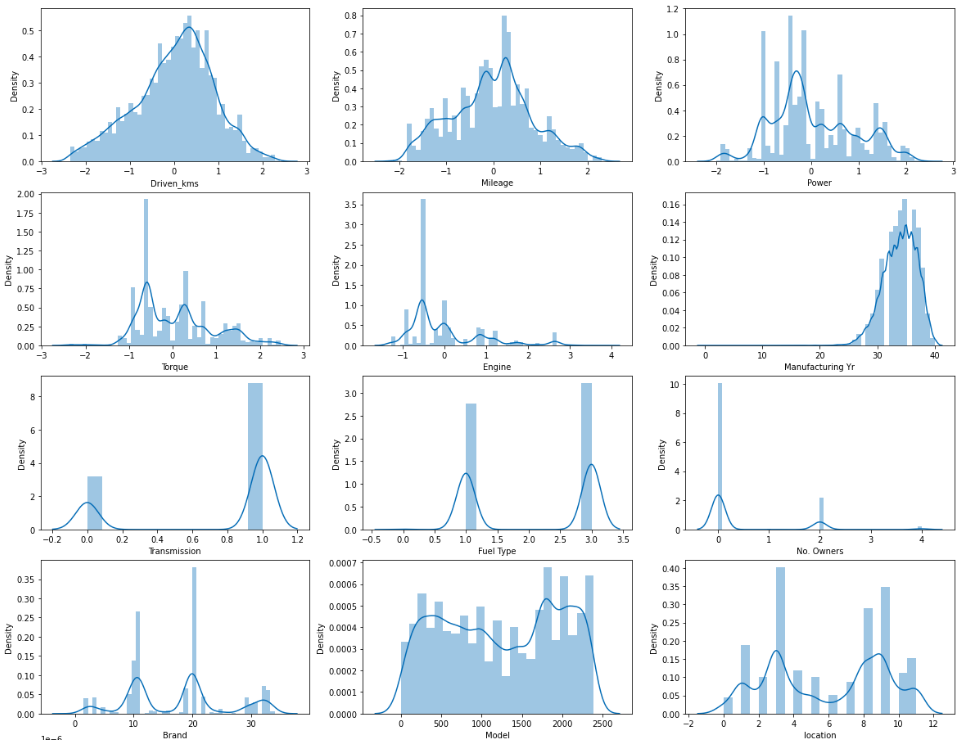
where N is the number of data points, is the i-th measurement, andis its corresponding prediction.

**Cross Validation**

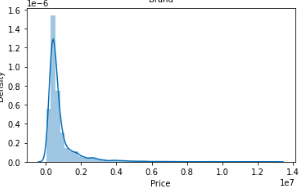
To check if a model fits the dataset with accuracy this measure is used. The models can have underfitting or overfitting dataset.   
Underfitting occurs when a statistical model or machine learning algorithm cannot capture the underlying trend of the data. Intuitively, underfitting occurs when the model or the algorithm does not fit the data well enough. Overfitting occurs when a [statistical model](https://chemicalstatistician.wordpress.com/2014/01/05/machine-learning-lesson-of-the-day-classification-and-regression/) or [machine learning](https://chemicalstatistician.wordpress.com/2014/01/04/machine-learning-lesson-of-the-day-supervised-and-unsupervised-learning/) algorithm captures the noise of the data. Intuitively, overfitting occurs when the model or the algorithm fits the data too well. If the model score is higher for the testing data than it is for the entire dataset then it can be said to be overfitting. Vice versa if the model score for testing data is lower than the score for entire dataset than than the model is said to be underfitting.

* Visualizations

**Distribution Graph**

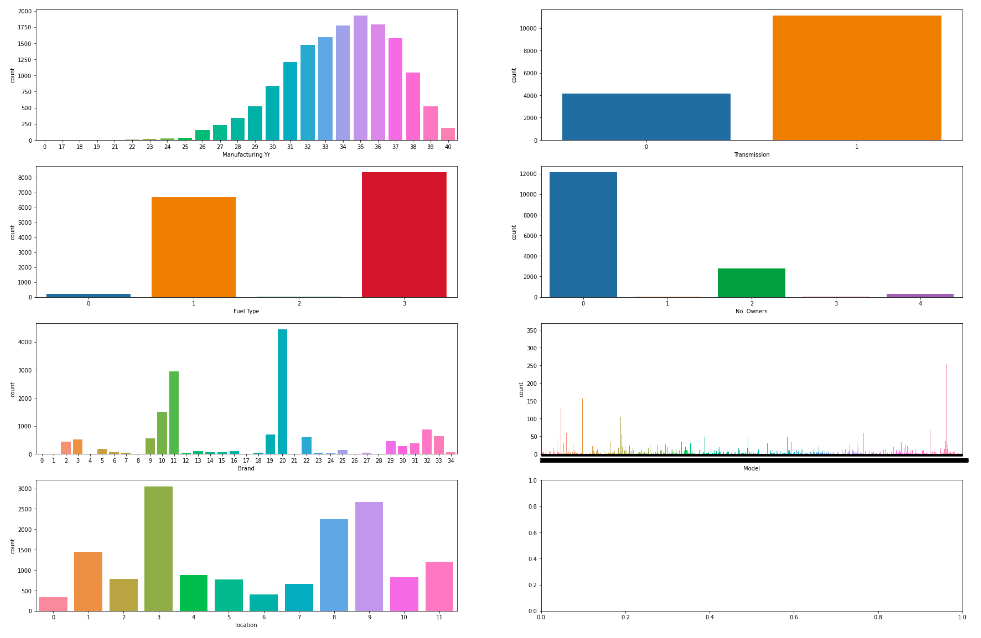
****

From the above graphs we can see the distribution for various features in the dataset. Continuous columns Driven\_kms and Mileage have normal distribution but columns Power, Torque and Engine have some amount of right skewness. This skewness is solved using power transformer.



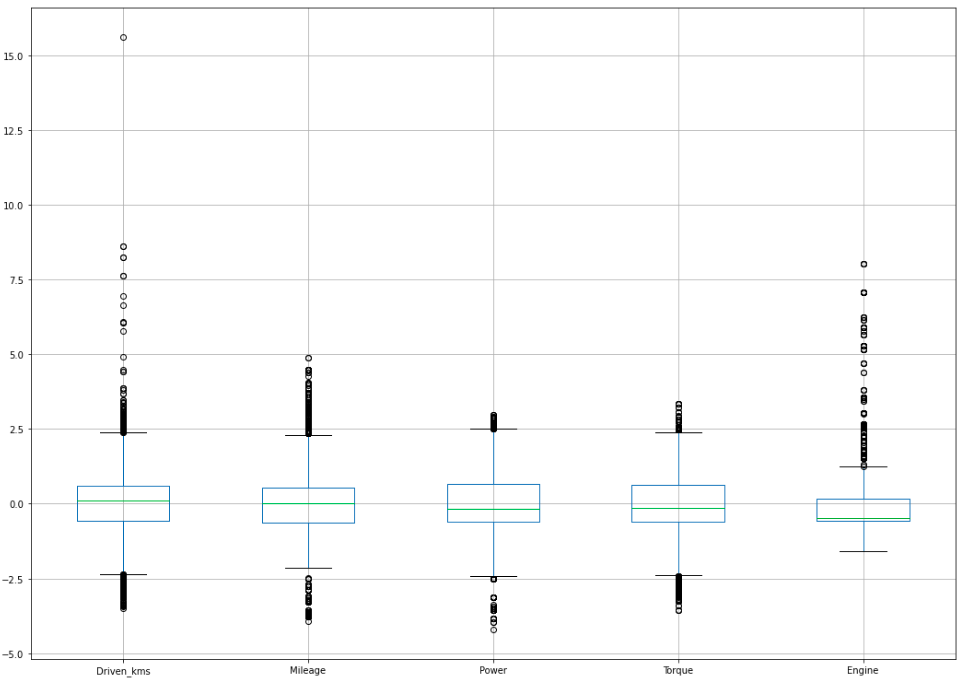
From the distribution of Price feature it can be said that the Price range for used cars varies a lot. This can be solved by normalizing the feature. As done above using np.log function

**Countplot for Categorical Columns**

****

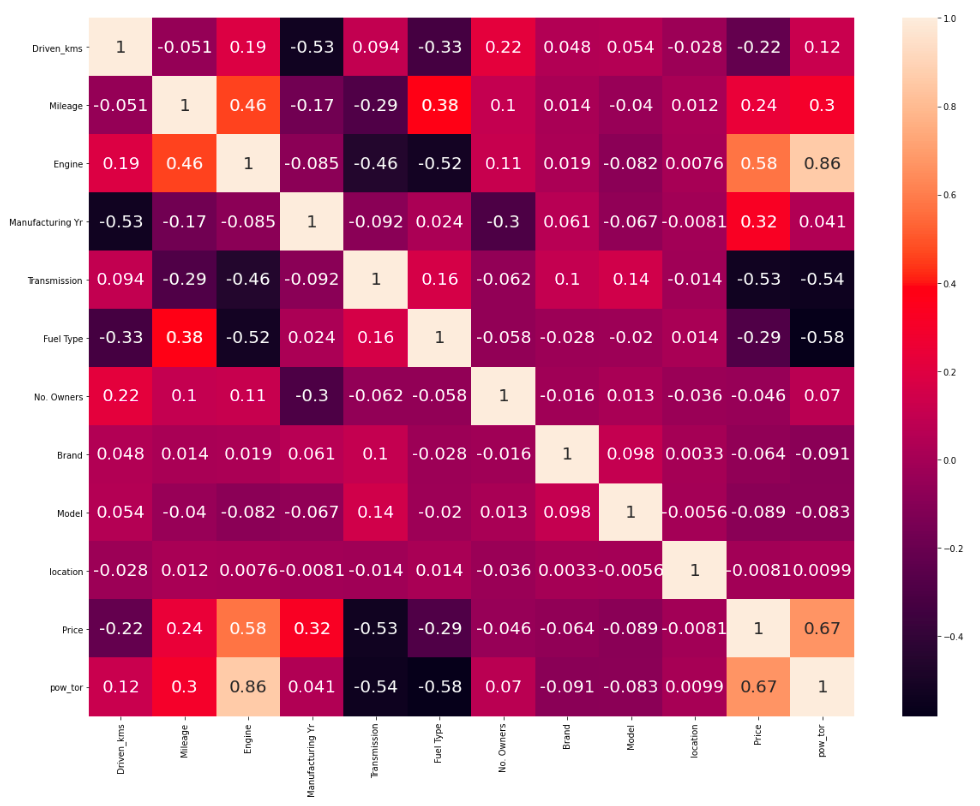
From the above graph it is seen that the categorical columns have highly imbalanced classes. In column Manufacturing Yr the data is highly distributed in classes 26 to 40. Therefore, the models will be better at predicting values for these classes as they have more samples in the dataset. In columns Transmission, Fuel Type, No. Owners, Brand and location there are multiple classes where large amounts of imbalance is present. The classes with more samples will have lower error during model prediction as the model is able to understand the dataset better for these classes. As we can see in the Brand column there are a large number of unique values. This might cause the model accuracy to reduce. However in models like Decision Tree and Ensemble Techniques this may help in predicting more accurate result.

**Boxplot Analysis**



A boxplot analysis of the dataset shows that features with continuous data have some outliers in them. We are allowed to remove maximum of 7-8% of the dataset entries. Using z-score will remove more than the allowed percentage. Therefore, we use Inter-Quartile Range or IQR to remove some of these outliers. It is seen that Engine column has a large number of outliers. So, except the engine column outliers from all other columns are removed using IQR. On doing so we get a data loss of approx. 7%.

**Heat Map after Transforming the dataset.**

From the above heatmap it can be seen that column Power and Transform have been merged to form one column pow\_tor which will remove some multicolinearity from the dataset. This column is highly correlated with the engine column. But, the VIF values for these columns are not alarmingly high and therefore this multicolinearity can be ignored.

* Interpretation of the Results
  1. Univariate Analysis

Using distribution graph we are able to see that there are few columns with skewed distribution. This can be interpreted as these columns have some outliers and so we need to get rid of these outliers. But, first we need to transform the columns to normalize the distribution that’s why I’ve used power transformer to normalize these features.

Using count plot for categorical features it is seen that the columns

have imbalanced class distribution. This can be interpreted as the

This might cause the model to make errors in prediction as classes with minimum distribution might not get interpreted by models well.

* 1. Boxplot

Using boxplot we can visualize the outliers in the dataset and eliminate these outliers. Continuous features with large number of outliers can affect the model’s learning. Also these outliers cannot be removed as they can cause reduce in the dataset more than the allowed limit. However, we can remove the outliers from those features with small number of outliers.

* 1. HeatMap

As seen from the heatmap we have high multicolinearity between three columns Power, Torque and Engine. We can solve this using Principal Component Analysis but from the VIF values of these features it is seen that the multicolinearity between them is not alarmingly high. However, to improve the accuracy of the model columns power, torque are merged together using feature engineering to lower the VIF scores in correlation with Engine feature. Columns Power and Torque are removed from the dataset.

* 1. Preprocessing

During preprocessing it is seen that a lot of the features have physically impossible values, features with numerical string values with units of measure and features with null values. Also some of these columns values with wrong unit of measures. All the features with numerical values are transformed into integer or float values. All the entries with unwanted values are cleaned.

Features with categorical values are encoded using label Encoder.

Features with continuous values are transformed and scaled to help the model learn better.

The target feature Price has a very large range of values. The models might not be able to fit these values well. Therefore, this feature is transformed using log transform.

* 1. Modelling
     1. Regression Models
        1. Linear Regression – this model performs the worst among all the regression model. However, this model gives a good r2 score but the RMSE score or error when scaled to the original values of the target is very high i.e, the model’s average error is very high.
        2. Lasso and Ridge Regression – these models are used to solve the multicolinearity problem. This can be seen from the r2 score of these model. They are able to perform a bit better than Linear Regression. However, their accuracy don’t make a significant difference in their scores compared to linear regression. Thus, proving that the dataset does not have high multicolinearity.
        3. Decision Tree Regression – Among the basic regression models this model is able to perform the best. It is able to fit the dataset well and keep the error score to minimum.
        4. K-NN Regressor – this model also fits the dataset well and is able to give a low error score. However, it does not perform better than Decision Tree Regressor.
     2. Ensemble Techniques
        1. Bagging using Random Forest – Random Forest is able to perform better than all the Regression Models and fits the model well while giving a low error.
        2. Boosting
           1. AdaBoost – AdaBoost model is not able to fit the dataset well as seen from the r2 score of the model. Therefore, it also gives large amount of mean error.
           2. XGBoost – XGBoost gives the highest r2 score and lowest RMSE score meaning the dataset fits the best in XGBoost Regressor model. It also means that this model gives the lowest amount of mean error i.e, the difference between actual and predicted values are lowest.
           3. GradientBoosting – this model is able to perform well but has a worse r2 score and higher RMSE score than XGBoost.

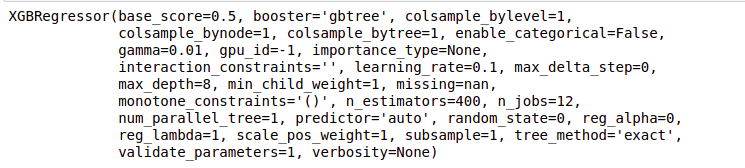
Therefore, out of all of the model XGBoost Regressor is selected for hyperparameter tuning to improve the model accuracy even more by tuning the default parameters in the model.

f Hyperparameter Tuning

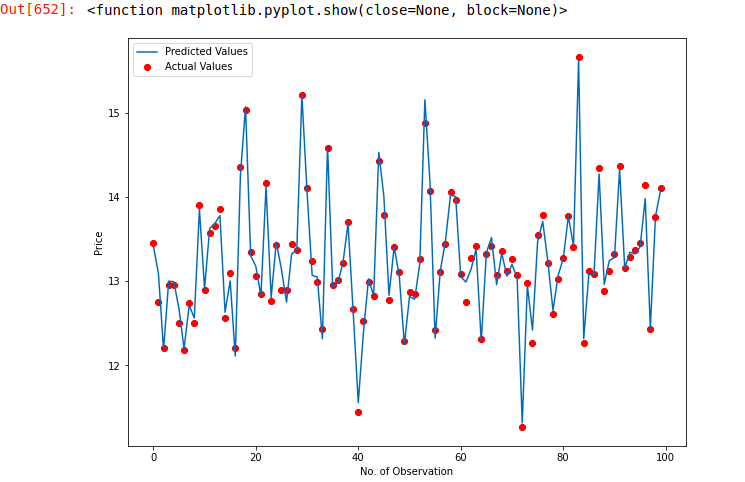
After analysing the results from all the models. It is seen that XGBoost Regressor has the best r2 score among all the models. Therefore, it can be said that this model fits the dataset best out of any other model. Also the RMSE score that we get for XGBoost Regressor is the best amoung all the models.

There are a few parameters that can be used to improve the model’s

accuracy. These parameters have some default values which can be changed to tune the model and achieve a better accuracy. These parameters can be found by using .get\_params() function. On further analysis of these parameters it is seen that parameters: max\_depth, learning\_rate, n\_estimators, gamma and num\_parallel\_tree can be implemented to increase the models learning accuracy. GridSearchCV is used to find the best combination of these parameters’ values which has the highest r2 score. From the result we get the following values of all the parameters.



On applying the dataset to this model we get a r2 score 0.9797 which is slightly better than the model without hyperparameter tuning. The model has negligible amount of underfitting. We get the best RMSE score out of any other model. The RMSE value is 0.107. The following graphs explains how well the dataset is able to predict the values.



In the above graph it is seen that the actual values are displayed using scatter plot and the actual values are displayed using line graph. There are a few points where the predicted values differ from the actual values. However, the difference between these values are not alarmingly high and therefore, we can say that this model is able to perform well.

When the values are predicted for unscaled target values and compared against the target values we get a RMSE score of 151528.49 which is relatively high. Also the mean\_absolute\_error (MSE) value for this model is 56014.255. This is relatively a good score considering the large range of values for the target feature.

**CONCLUSION**

* Key Findings and Conclusions of the Study

It is noted from the dataset that the target feature Price has continuous data ranging from 35,000 to 1,30,00,000. This is a large range of values. The dataset includes a mix of continuous and categorical features. Some features have colinearity which can be solved using feature enginerring.

Since, the target feature has a large range of values the values for normalized. Some data cleaning is also done for all the columns in the dataset. The dataset is split into two set training and testing. The training and testing sets are divided in the ratio of 4:1 respectively. The models are able to fit the dataset well and give prediction with high accuracy for the normalized target values.

* Learning Outcomes of the Study in respect of Data Science

From the visualizing the feature in dataset I was able to assess:

1. Which features require data cleaning.
2. Which features have a large range of values and thus require normalization.
3. Which features consists of categorical values.
4. How much outliers are present in a feature.
5. What are the feature that consists of multicolinearity between them.
6. How well does the dataset fits on a model.

It was a great learning opportunity to create a regression model for dataset with such a large range of target values. The various algorithms implemented improved my understanding of these models. For example various articles on similar regression models tells that models like Random Forest and Decision Tree does not require scaling of Target features and will give almost similar result when scaled or unscaled. However, linear regression models do require scaling. XGBoost Algorithm is able to handle large range of target values better than other Boosting models. Model XGBoost Regressor gives the highest degree of fit or R2 score and lowest error or RMSE score. Therefore, this model is selected for hypertuning. After hypertuning the parameters the resultant combination of parameter values will be used in the final model which will give the prediction with least average error.

While working on this project the biggest challenge was to find the most suitable metric to measure the error and accuracy of the model. Due to the nature of the scaled target features the metric used for measurement of the mean error can give a small error value in theory but when measured against scaled values the mean error can have a large value as seen in the final model study.

* Limitations of this work and Scope for Future Work

The most notable limitation of the solution provided is that the mean\_absolute\_error (MAE) for the final model gives a value of approx. 49,000 which is high considering the minimum target values is 35,000. This can improved using some more research. An ideal MAE score for this dataset should be below 10,000. Therefore, on further research the MAE score can be reduced, the degree of fit of the model can be improved and a model could be developed to make predictions with lower error values.

Steps to be followed to extend this study are:

1. Collecting more data feature to create more correlations between the independent features and the target feature.
2. Remove irrelevant data and data features.
3. Analyse the dataset more in depth.
4. Remove all multicolinearity.
5. Create a model that focuses on both smaller values and larger values. As the error value for small values would be smaller and that for larger values would be relatively large.