

***A REPORT***

***ON***

***PREDICTING THE RESALE VALUE***

***OF USED CARS***

***PREPARED BY -***

***SHOURYA CHOUHAN***

***DINESH GIRI***

***ADITYA NAIR***

***SUBMITTED TO -***

***PROF. VENU GUNTUPALLI***

***TABLE OF CONTENTS*** -

1. Abstract
2. Introduction
3. Performance Metrics
4. Understanding the Dataset
5. Data Cleaning & Exploratory Data Analysis
6. Model implementation
7. Evaluation
8. Conclusion
9. References

**ABSTRACT**

In today’s world, if a car owner decides to sell off his existing car and replace the same with a new one, it is important for the owner to know the true worth of their existing car as that would help the owner get a fair deal for his existing car. Using the Databricks community edition as the software and machine learning techniques we aim to find the prices of used cars in the USA - keeping in mind the depreciation in cost a vehicle would undergo in due time.

As mentioned, we aim to create a machine learning model which helps us predict the prices of the used car using the techniques - Linear Regression, XGBoost and Random forest, before finally concluding on which of the 3 mentioned models would be best suited for predicting the prices of the used car.

**INTRODUCTION**

With the advent of internet and all the technology available at people’s disposal, people are known to make informed choices/decisions with regards to almost everything one can imagine - be it buying ordering food online, travelling on a vacation etc. Even with cars - its no different. Whenever a person wants to buy a new car and is looking at replacing it with an older one, he would like to get the best value out of his existing car. Car is an asset whose value keeps depreciating with time and hence it is imperative for the owner to know if he is getting the best value out of his existing car.

Using machine learning techniques in databricks, we aim at creating the best model for predicting the resale value of used cars in USA. The ultimate goal here is to ensure that human efforts and time spend on understanding the value of the used car is reduced and the machine does that for us with a single click.

**PERFORMANCE METRICS**

Since, we are dealing with a regression problem, hence we shall use a host of features like - Manufacturer, make & model, Year of manufacture, city etc. as input into our model.

We shall be evaluating 3 machine learning techniques in this paper -

1. Linear Regression
2. Gradient boosted tree regression
3. Random Forest

The evaluation of the machine learning models shall be done on the basis of -

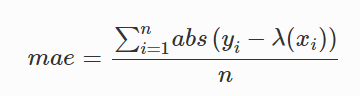
1. Mean Absolute Error (MAE) - a metric for model evaluation - which is used with regression models, MAE - the mean absolute error of a model is the mean of its absolute values for each of the prediction error i.e. individual prediction error over all the instances of the test set.

where -

y i - the real target value for the test instance *x* *i*

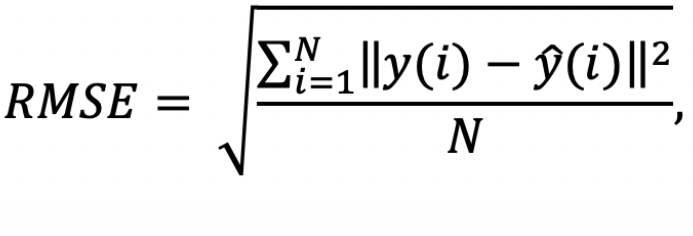
*λ*(*x* *i* ) - the target value that is predicted for the test instance *x* *i*

n - the total number of instances



1. Root mean square error(RMSE) Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results.

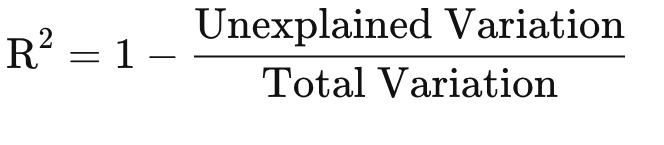
The formula is:



Where N is the number of data points, y(i) is the i-th measurement, and y ̂(i) is its corresponding prediction.

1. R2 score: R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse).

The formula is:



***UNDERSTANDING THE DATASET***

The dataset used for the purpose of this paper was downloaded from Kaggle -

Link - <https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data>

It is a single csv file - with no separate dataset for testing and training. The dataset consists of over 4.5L rows and has 26 columns/features making the dataset suitable for Big data analysis. As the title suggests, the target feature for our analysis is Price and the rest of the features shall be used in order to predict the resale value of the car. Some of the other features that shall aid in predicting the car price are -

1. Year - The year in which the car was purchased
2. Region - The area of USA where the car is available for selling
3. Manufactuer - The brand of the car manufacturer
4. Model - Which model is the car belonging to
5. Cylinders - helps understand the fuel capacity of the car
6. Fuel - Type of fuel transmission of the vehicle - petrol, diesel etc.
7. Odometer - No. of Kms driven by the vehicle.
8. Transmission - Whether the vehicle is an automatic or manual vehicle
9. Size - Tells us about the size of the car
10. Type - Whether the vehicle is a sedan, hatchback etc.
11. Paint\_color - Color of the vehicle
12. Description - A short description of the vehicle provided by the owner.

City, County, State, vehicle coordinates etc. are amongst the other features present in the dataset.

In this model creation at the start, we need to preprocess the dataset. So we used the Numpy and pandas libraries in python to do the preprocessing of the dataset. At the start, there were 26 columns in the dataset. But when creating the machine learning model we did not use all 26 columns. So we need to select the columns that need to create the machine learning model.

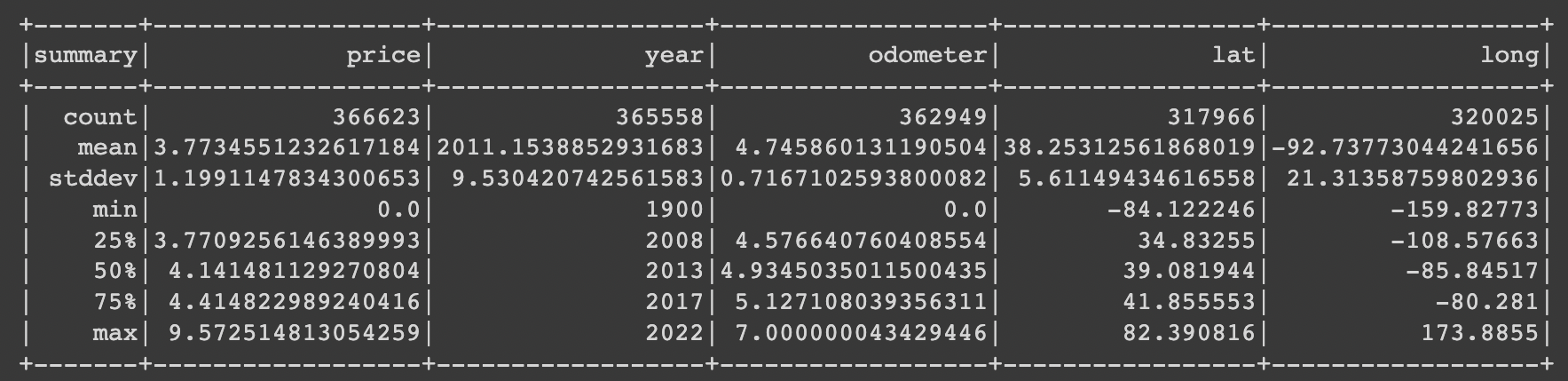
***DATA CLEANING***

Then we check for the null values in the dataset. Here we drop the rows that contain null values. Because in this case, we cannot add average, mean or mode to fill data. Because we know that this is a used car dataset. So after people buy cars, they tend to do changes or upgrades to the cars. So we think about that scenario and we planned not to add values to missing values. So we drop the rows that contain null values and continue the process. Then we look into the price column values.

* "url", "region\_url", "image\_url", "description" columns are unnecessary. let's drop them
* It seems pyspark interpreted "lat" and "long" columns as string
* We should clean these columns from alphabetic characters
* And we should check the numeric range of "lat" and "long" columns
* Some categorical columns have own null indicators like "other", set them null

# **Data Wrangling**

* "price" and "odometer" columns have right skewed distribution (mean is bigger than median)
* "price" and "odometer" columns have too big numbers and too much standard deviation, we can take the log of them
* "posting\_date" column can be split to "posting\_year", "posting\_month"



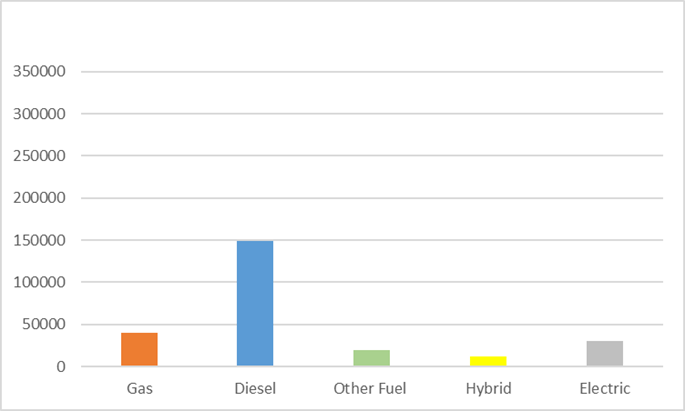
**Exploratory Data Analysis**

At the very beginning, we had ascertained that Price is the most important feature for us in this paper and hence, the exploratory data analysis that is done revolves around the ‘Price’ feature.

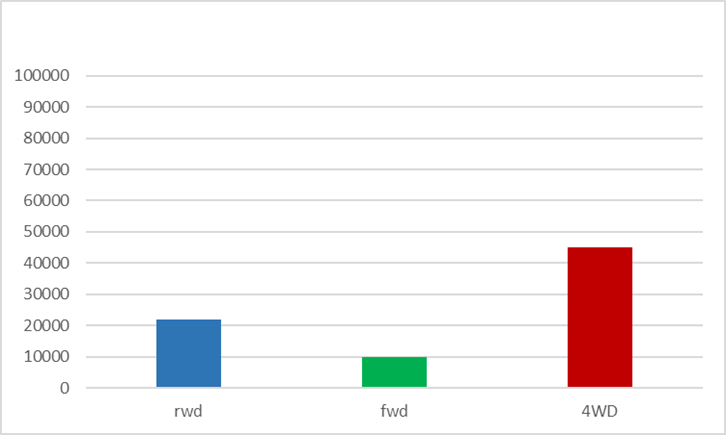
Firstly, we compare -

Price vs Type of fuel -

From the figure shown below, we understand and analyze that Diesel cars are the most expensive in the US market, whereas the other fuel types are seen to have a much lower price point. Therefore, we understand that The type of fuel used to run a vehicle has an affect on the price of the vehicle.

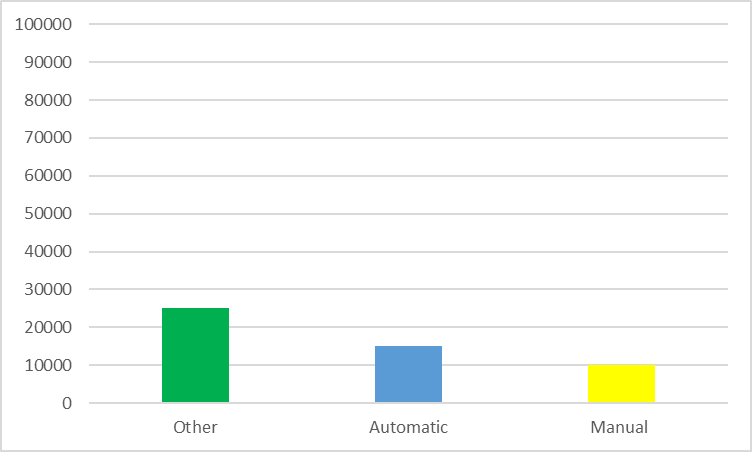


Price vs Type of Drive



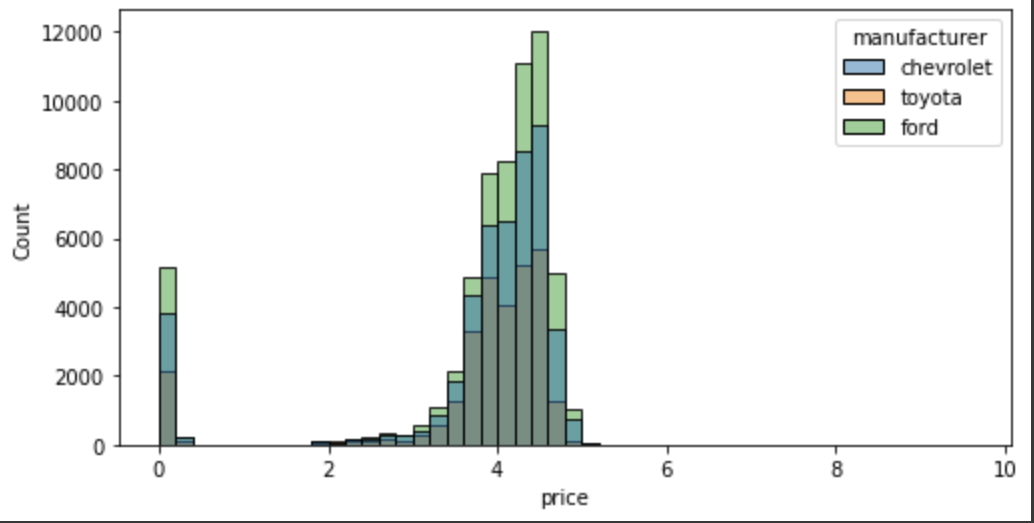
From the figure above, we can deduce that even the type of drive plays an important role in the prediction of the price. The vehicles with 4WD are the most expensive in the US market, followed by the reverse wheel drive (rwd) and then the fwd.

Price vs Type of Transmission

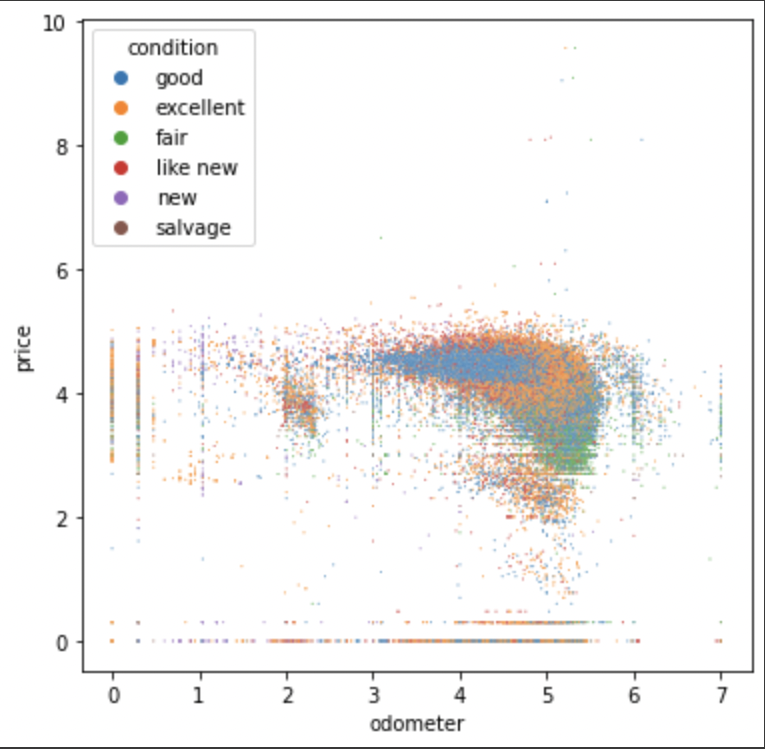


From the above plot, we find out that most of the Automatic and Manual vehicles are priced less than $20K, while the vehicles with other transmission type are seen to be more expensive.

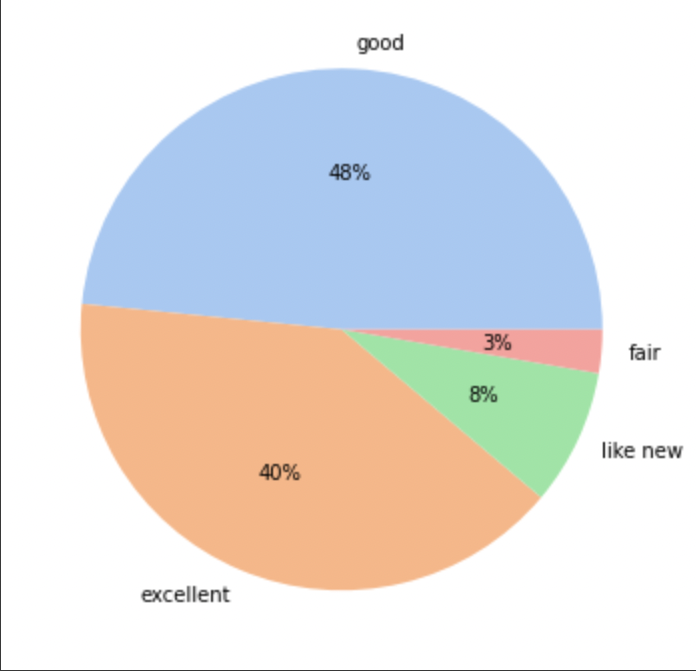
Histogram of cars according to their manufacturers to see the distribution according to their prices.



A scatter plot to show distribution of cars according to their condition. Where x='odometer', y='price', hue='condition'



Using a pie chart to get an idea about the distribution of cars in the dataset according to their condition.

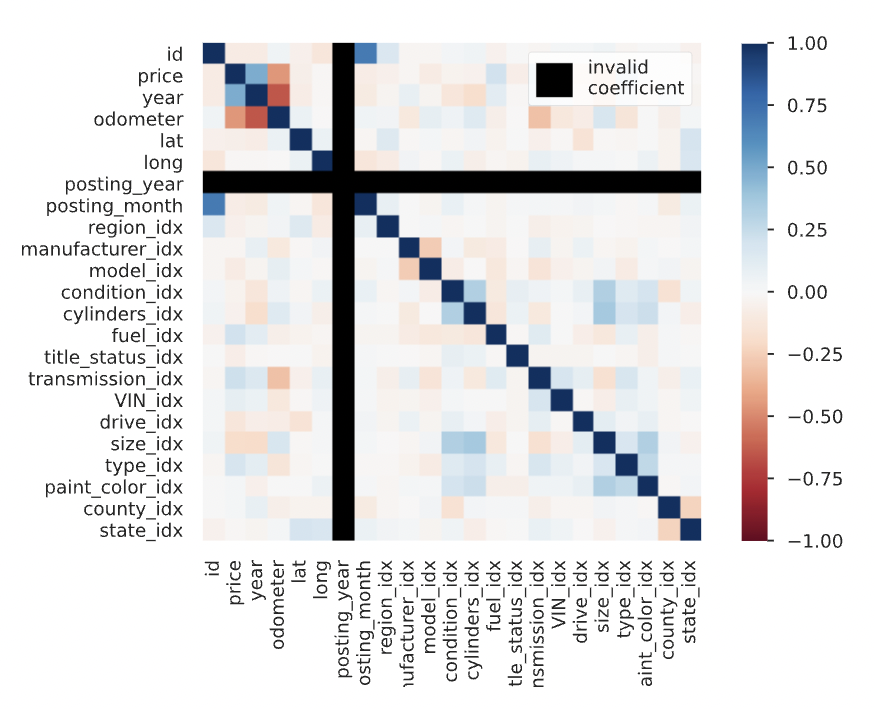


We also go on to plot the correlation matrix using a heat map in order to understand the correlation between each of the variables using the Phi\_k correlation analyzer library

From the heatmap generated below, we find that Odometer is seen to have a positive correlation of 0.7 with regards to price. Therefore we see that even in terms of feature Odometer, we see price plays an important role. Also, since Model is highly correlated with most of the features, we understand that it is not a very important feature for price.

### **Spearman's ρ**

The Spearman's rank correlation coefficient (*ρ*) is a measure of monotonic correlation between two variables, and is therefore better in catching nonlinear monotonic correlations than Pearson's *r*. It's value lies between -1 and +1, -1 indicating total negative monotonic correlation, 0 indicating no monotonic correlation and 1 indicating total positive monotonic correlation.



### Spearman's ρ

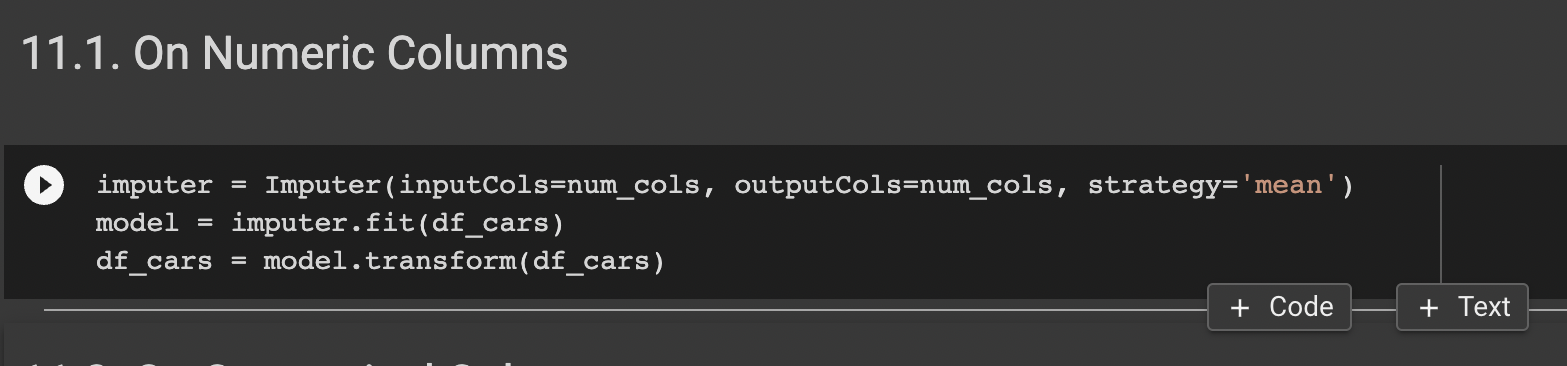
The Spearman's rank correlation coefficient (*ρ*) is a measure of monotonic correlation between two variables, and is therefore better in catching nonlinear monotonic correlations than Pearson's *r*. It's value lies between -1 and +1, -1 indicating total negative monotonic correlation, 0 indicating no monotonic correlation and 1 indicating total positive monotonic correlation.

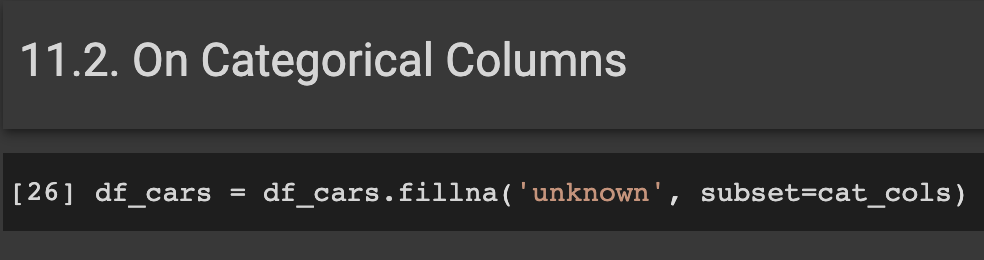
To calculate *ρ* for two variables *X* and *Y*, one divides the covariance of the rank variables of *X* and *Y* by the product of their standard deviations.

Now, we move onto data cleaning, wherein we find out basis the output below we decide to remove out all the features that are seen to have a null value percentage of above 50% and look at filling the other missing data in those features having lesser null values using domain knowledge and understanding.

**Imputation**

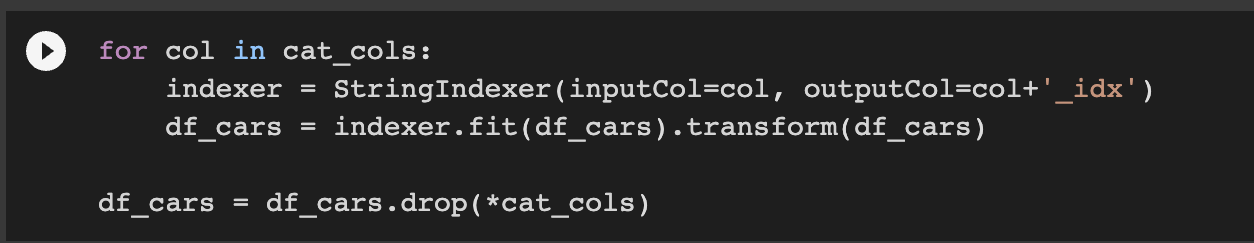
Imputation estimator for completing missing values, using the mean, median or mode of the columns in which the missing values are located. The input columns should be of numeric type.





**Categorical Column encoding**

* Encoding categorical data is a process of converting categorical data into integer format so that the data with converted categorical values can be provided to the different models.
* A label indexer that maps a string column of labels to an ML column of label indices. If the input column is numeric, we cast it to string and index the string values.



**Splitting data into train and test (Random Splitting)**

In this process, 80% of data was split for Train data and 20% was taken as test data.

**MODEL IMPLEMENTATION**

In this section, we will look at some of the applied machine learning models in the same order.

1. Linear Regression
2. Random Forest
3. GBT Regressor

**1. Linear Regression:**

Linear regression is the most commonly used method of predictive analysis. It uses linear relationships between a dependent variable (target) and one or more independent variables (predictors) to predict the future of the target.

**Results of Linear Regression**:

| Evaluation metrics | Score |
| --- | --- |
| Mean absolute error | 0.7306 |
| Root mean square error | 1.1811 |
| R2 | 0.0357 |

**2. Random Forest:**

Random Forest is a bagging algorithm where m no. of decision trees with low bias and high variance are trained on samples of data so that every model learns a different aspect of data and aggregation of their results leads to a decrease in overall overfitting.

**Results on Random Forest:**

| Evaluation metrics | Score |
| --- | --- |
| Mean absolute error | 0.6578 |
| Root mean square error | 1.1038 |
| R2 | 0.1578 |

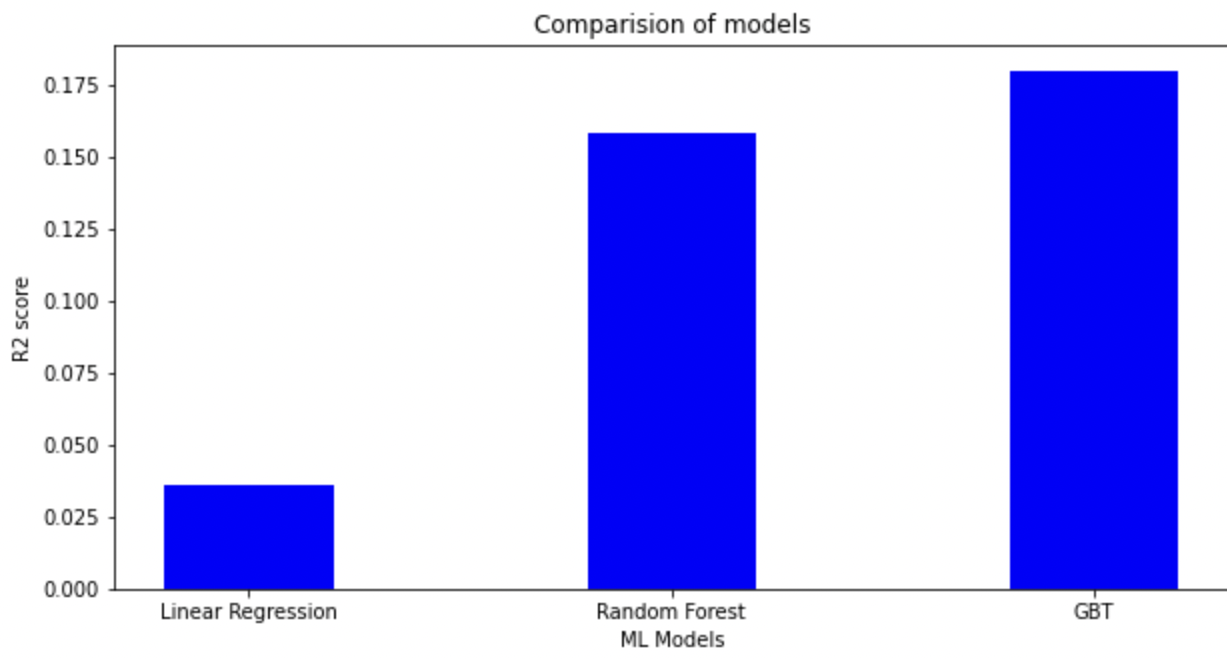
**3. Gradient boosted trees:**

This estimator builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function.

**Results on GBT:**

| Evaluation metrics | Score |
| --- | --- |
| Mean absolute error | 0.6534 |
| Root mean square error | 1.0894 |
| R2 | 0.1796 |

**Comparison of Models:**



From the above plot, we can observe that GBTis performing best with the biggest r2 score of 0.1796 so we will save this model for future use.

**Summary :**

Our goal was to build a predictive model which can predict the price of the used car given 25 varieties of features and 458213 rows.

Initially, data exploration was done to get the insights from data, and then data cleaning was performed to remove the noise from the data. Then, missing values were computed using the iterative imputer method and some insights of the data we got while performing EDA.

At Last, after applying ML models we concluded that GBT performed best on our dataset with a MAE of 0.6534 and r2 of 0.1796

**Runtime of different models in Google collab**

●Random forest : 6mins using cpu

2 mins using gpu

●Linear regression: 3 mins

●GBT : 5 mins

***REFERENCES***

* <https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data>
* <https://python.plainenglish.io/price-prediction-machine-learning-model-for-used-cars-using-pyspark-ba7cefc9d4dc>
* <https://link.springer.com/referenceworkentry/10.1007/1-4020-0612-8_580>
* <https://link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8_525>
* <https://scikit-learn.org/stable/modules/model_evaluation.html>
* <https://towardsdatascience.com/first-time-machine-learning-model-with-pyspark-3684cf406f54>
* <https://www.tutorialspoint.com/google_colab/google_colab_using_free_gpu.htm>