INFO6105 - Data Science Engineering Methods and Tools
Used Car Price Prediction

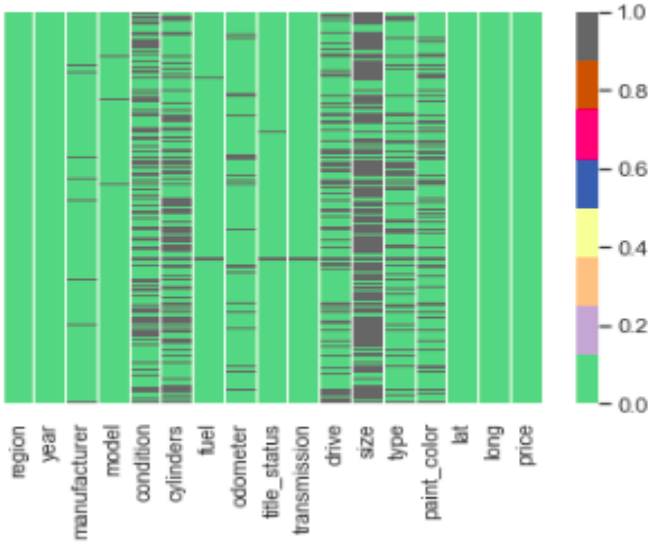
Problem statement : The prices of new cars in the industry is fixed by the manufacturer with some additional costs incurred by the Government in the form of taxes. So, customers buying a new car can be assured of the money they invest to be worthy. But recently, used cars sales are on a global increase. The car's value depends on many factor including year of registration, manufacturer, model, mileage, horsepower, origin and several other specific informations such as type of fuel and braking system, condition of bodywork and interiors, interior materials (plastics of leather), safety index, type of change (manual, assisted, automatic, semi-automatic), number of doors. For used cars the other important features are the number of previous owners, if it was previously owned by a private individual or by a company and the prestige of the manufacturer.Therefore, all the above features should be considered to determine the price of the used car accurately. There is a need for a used car price prediction system to effectively determine the worthiness of the car using a variety of features.

Goals : There are two three goals of this Data Science Project. First, to estimate the price of used cars by taking into account a set of numerous features, based on the large dataset of the historical data collected for a huge amount of cars. Second, to get a better understanding of the most relevant features that help determine the price of a used vehicle.

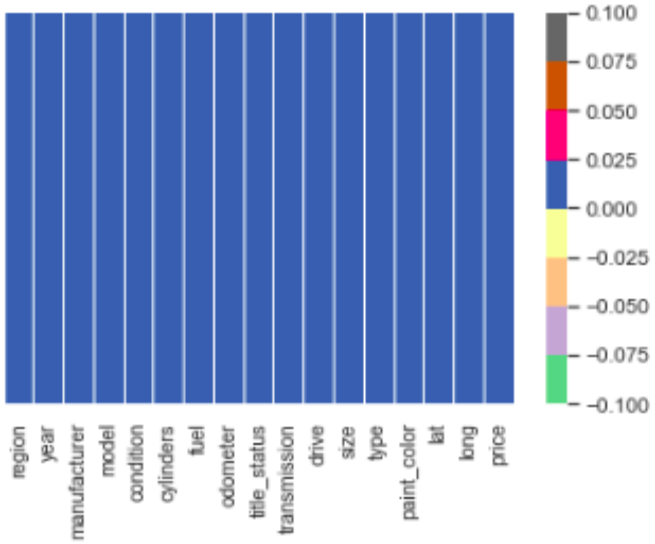
Data : https://www.kaggle.com/austinreese/craigslist-carstrucks-data
The data originally is 1.34GB, which is 458213 rows and 26 columns, for the scope of this project we reduce it to 100000 rows. The column 'condition' describes the condition of the car. There is an odometer column describing the distance travelled by the car. There is a price column which has to be calculated using the model created.

Data Pre-Processing : We use heatmap to visualize the null values. The heatmap before and after preprocessing of data in this project is as below:

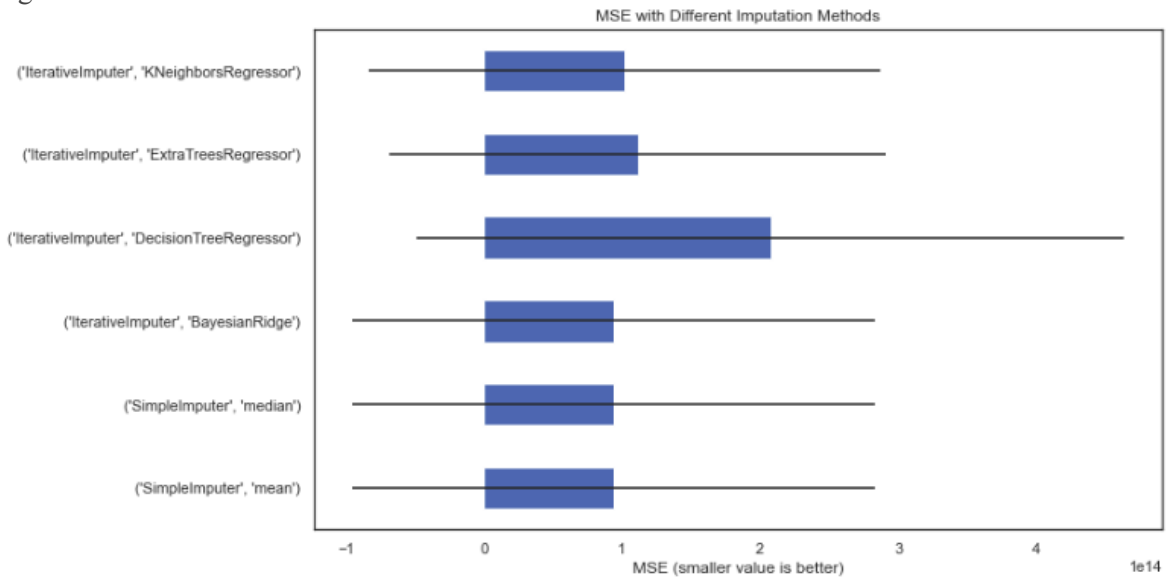
```
Out[437]: <AxesSubplot:>
```



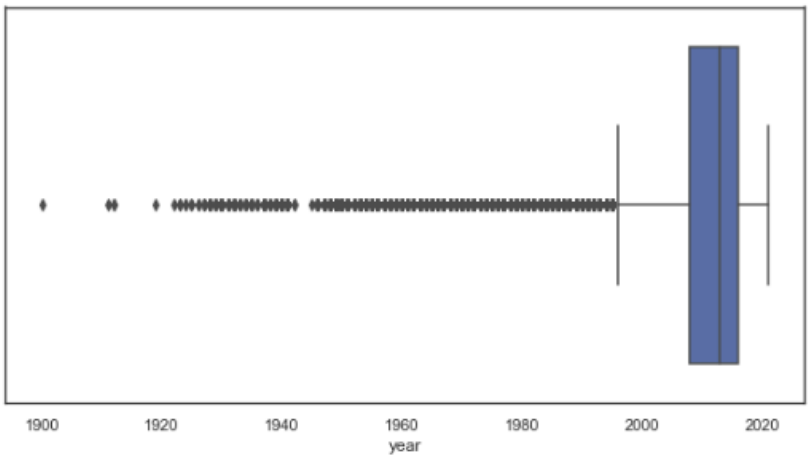
```
Out[450]: <AxesSubplot:>
```

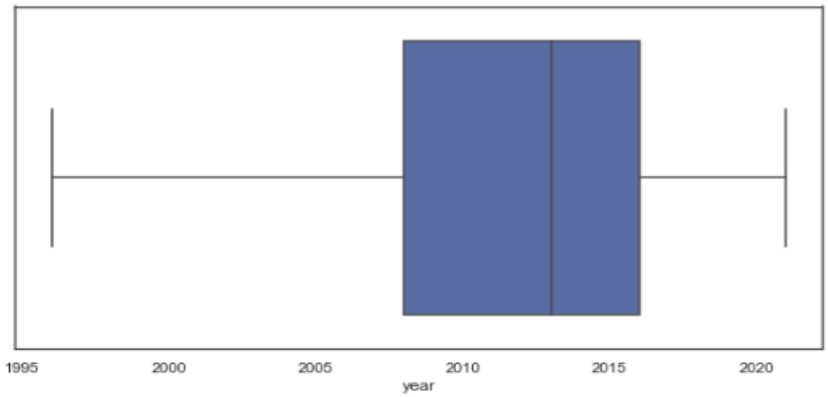


The data is then pre-processed by re-indexing the data frame to put the price column in the end as its value is to be computed using various models with different accuracy to find the best model with the highest accuracy. The missing value imputation is done using 6 different imputation methods: mean, median, bayesian ridge, decision tree regressor, extra trees regressor, KNeighborsregressor . Plotting the graph to find the MSE with different imputation methods and then selecting the imputation method having the smaller MSE value.



Then we detect the Outliers by plotting the histogram of the outliers. We use two methods to remove the outliers : Z-score and IQR score. Now, the box plot and histogram is plotted free from the outliers and we standardize all the features.
The image of data boxplot of data in year data column with outlier and without outliers are as below for example, all the other columns with outliers are also cleaned.





After comparing the results of both the methods as per the below table, we select the IQR-Score as it removes more outliers than Z-Score.

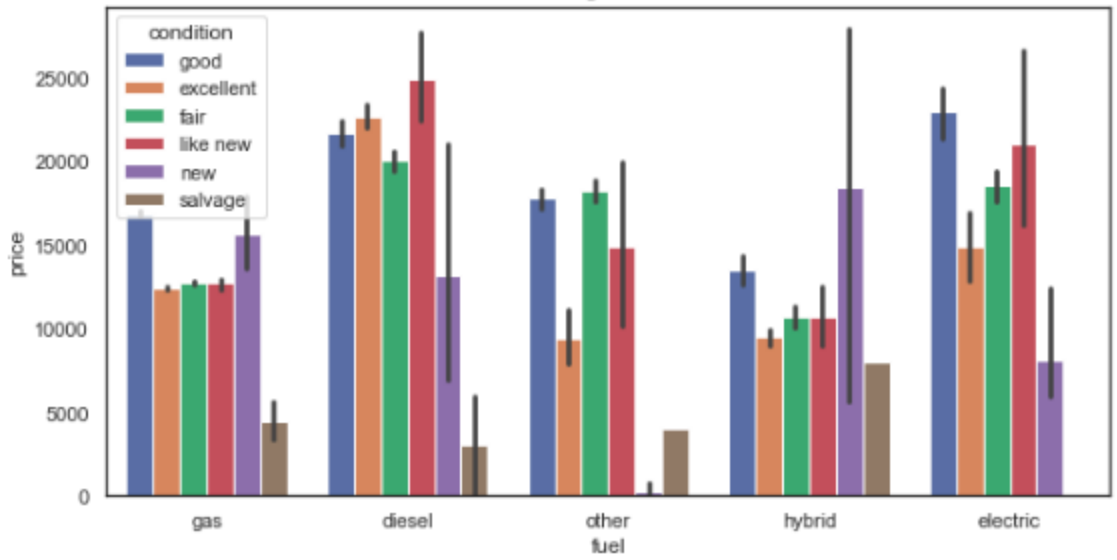
Comparision of both outliers removing methods

Methods	Model Performane
Original DataFrame (Before removing outliers)	(100000, 18)
Reduced DataFrame using **Z-Score** (After removing outlier)	(94026, 18)
Reduced DataFrame using **IQR-Score** (After removing outlier)	(88660, 18)

Data Visualization :

Data visualization is done to graphically represent the data for a better understanding, Here, we are using bar graphs for the visualization. One example of visualization in this project is a bar graph with fuel on x axis and price on y axis and the color of the bars depicting the condition of the cars. The snippet of that graph is as below:

Figure 2



Model Implementation :
The data is split as 85% training data and 15% test data and later on the training data is split into 17% validation data. Then, we find the MSE, RMSE, R2 Score, Accuracy(%).

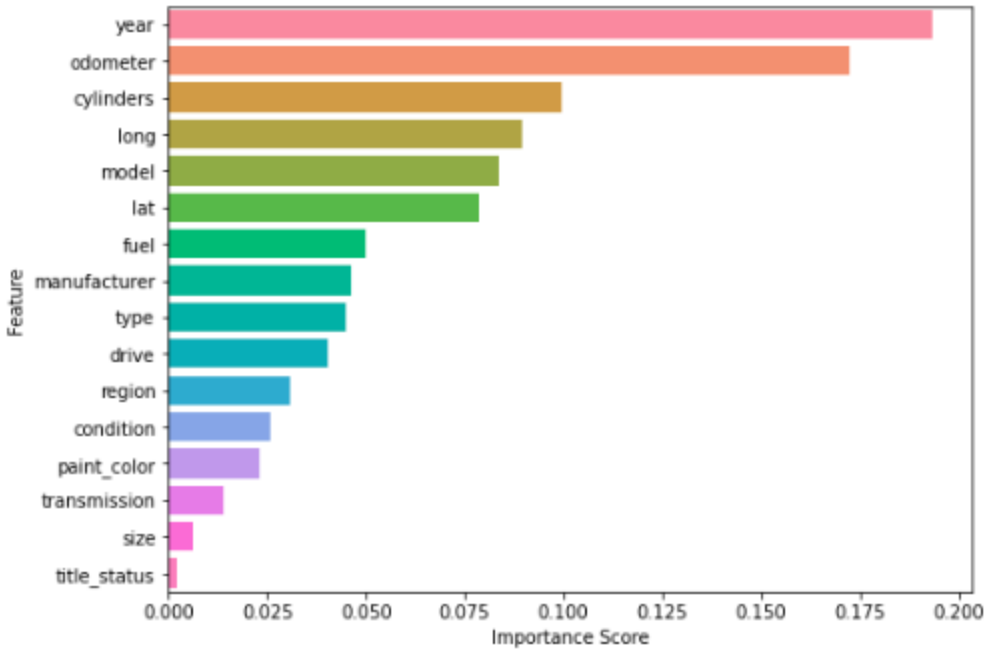
MSE - The mean squared error of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors - that is, the average squared difference between the estimated values and what is estimated. MSE is a risk function, corresponding to the expected value of the squared error loss. The fact that MSE is almost always strictly positive (and not zero) is because of randomness or because the estimator does not account for information that could produce a more accurate estimate.

RMSE - 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

R2 Score - It is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. 0% indicates that the model explains none of the variability of the response data around its mean.

We perform feature engineering to remove the unwanted features and improve the performance of the machine learning models that we will be using.
Feature engineering is the process of using domain knowledge to extract features from raw data. These features can be used to improve the performance of machine learning algorithms. Feature engineering can be considered as applied machine learning itself.
The below graph depicts that size and title_status are the not important features for consideration and can then be dropped to increase model accuracy.

```
Out[83]: <AxesSubplot:xlabel='Importance Score', ylabel='Feature'>
```



After feature engineering, we proceed to each of the models below and find out the MSE, RMSE and R2 Score for all the models and on the basis of those scores we decide the most suitable model with the highest scores.

1. Linear Regression

Linear regression is useful to find relationships between two continuous variables. One of the continuous variables is called a predictor or independent variable and the other continuous variable is response or dependent variable. It looks for statistical relationships but not deterministic relationships. Relationship between two variables is said to be deterministic if one variable can be accurately expressed by the other. The core idea is to obtain a line that best fits the data. The best fit line is the one for which total prediction error (all data points) are as small as possible. Error is the distance between the point to the regression line.

The result of our dataset on applying Linear Regression is as below :

MSE : 0.6328

RMSE : 0.7955

R2 Score : 0.3692 or 36.9177%

2. K-Neighbors Regressor

KNN regression is a non-parametric method that, in an intuitive manner, approximates the association between independent variables and the continuous outcome by averaging the observations in the same neighbourhood. The size of the neighbourhood needs to be set by us, or can be chosen using cross-validation to select the size that minimises the mean-squared error.

The result of our dataset on applying K-Neighbors Regressor is as below:

MSE : 0.2893

RMSE : 0.5378

R2 Score : 0.7116 or 71.1612%

3. Random Forest Regressor

Random forest is a type of supervised learning algorithm that uses ensemble methods (bagging) to solve both regression and classification problems. Random Forest is an ensemble machine learning technique capable of performing both regression and classification tasks using multiple decision trees and a statistical technique called bagging.The algorithm operates by constructing a multitude of decision trees at training time and outputting the mean/mode of prediction of the individual trees. Bagging along with boosting are two of the most popular ensemble techniques which aim to tackle high variance and high bias.

The result of our dataset on applying Random Forest Regressor is as below:

MSE : 0.1823

RMSE : 0.4269

R2 Score : 0.8183 or 81.8283%

4. Bagging Regressor

A Bagging regressor is an ensemble meta-estimator that fits base regressors each on random subsets of the original dataset and then aggregate their individual predictions either by voting or by averaging to form a final prediction. Bagging Regressor trains each regressor model on a random subset of the original training set and aggregates the predictions. Then, the aggregation averages over the iterations because the target variable is numeric. In the following recipe, we are going to showcase the implementation of a bagging regressor with bootstrap samples.

The result of our dataset on applying Bagging Regressor is as below:

MSE : 0.3137

RMSE : 0.5601

R2 Score : 0.6873 or 68.7282%

5. Adaboost Regressor

An AdaBoost regressor is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction. As such, subsequent regressors focus more on difficult cases. A decision tree is boosted using the AdaBoost. Adaboost helps you combine multiple “weak classifiers” into a single “strong classifier”.

We use GridSearchCV for hyper parameter tuning.

The result of our dataset on applying AdaBoost Regressor is as below:

MSE : 0.1631

RMSE : 0.4038

R2 Score : 0.8374 or 83.7438%

6. XGBoost

XGBOOST is known as "Extreme Gradient Boosting" as it uses a combination of software and hardware optimization to yield great results in the shortest amount of time. It is a decision-tree based ensemble Machine Learning algorithm that uses a gradient boosting framework.

That XGBoost is a library for developing fast and high performance gradient boosting tree models. That XGBoost is achieving the best performance on a range of difficult machine learning tasks.

XGBoost minimizes a regularized objective function that combines a convex loss function based on the difference between the predicted and target outputs and a penalty term for model complexity

The result of our dataset on applying XGBoost is as below:

MSE : 0.127

RMSE : 0.3563

R2 Score : 0.8734 or 87.3422%

7. DNN

Deep Neural Networks have an input layer, an output layer and few hidden layers between them. These networks not only have the ability to handle unstructured data, unlabeled data, but also non-linearity as well. They have a hierarchical organization of neurons similar to the human brain. The neurons pass the signal to other neurons based on the input received. If the signal value is greater than the threshold value, the output will be passed else ignored. As you can see the data is passed to the input layer and they yield output to the next layer and so on until it reaches the output layer where it provides the prediction yes or no based on probability. A layer consists of many neurons, and each neuron has a function called Activation Function. They form a gateway of passing the signal to the next connected neuron. The weight has the influence of the input to the output of the next neuron and finally, the last output layer. The weights initially assigned are random, but as the network gets trained iteratively, the weights are optimized to make sure that the network makes a correct prediction.

For performing DNN, we Normalize the data using MinMax Scaler and write custom keras function for getting MSE, RMSE and R-Square values.

The result of our dataset on applying DNN is as below:

MSE : 0.0223

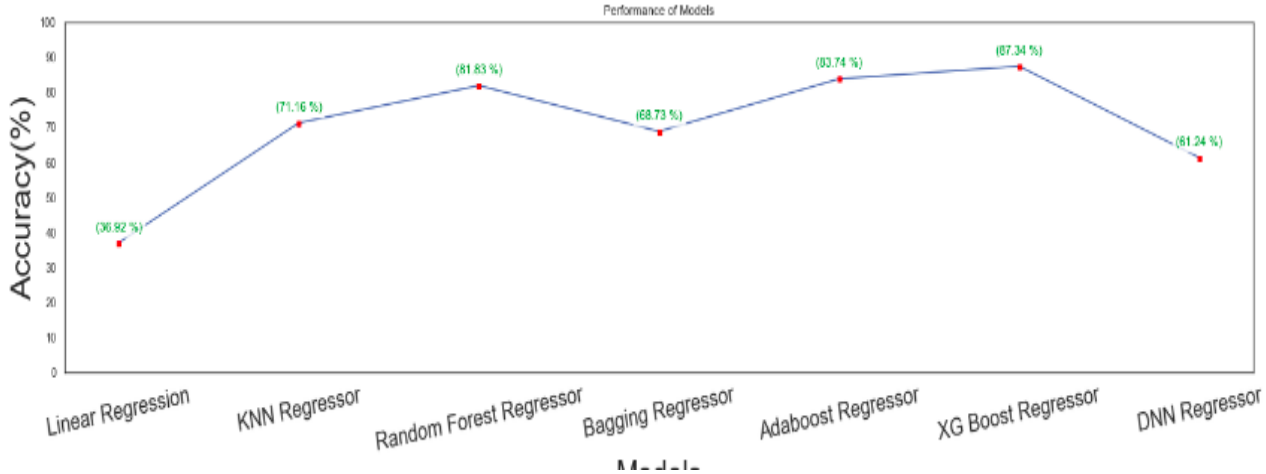
RMSE : 0.1495

R2 Score : 0.6124 or 61.2436%

Comparison of performance of different models:

Below is the tabular comparison and graphical representation of the results:

	Linear Regression	KNN Regressor	Random Forest Regressor	Bagging Regressor	Adaboost Regressor	XG Boost Regressor	DNN Regressor
MSE	0.6328	0.2893	0.1823	0.3137	0.1631	0.1270	0.0223
RMSE	0.7955	0.5378	0.4269	0.5601	0.4038	0.3563	0.1495
R2 Score	0.3692	0.7116	0.8183	0.6873	0.8374	0.8734	0.6124
Accuracy(%)	36.9177	71.1612	81.8283	68.7282	83.7438	87.3422	61.2436



Conclusion:

This is a big data set so the challenge of cleaning this data is also very crucial in order to increase the accuracy rate of the model. So, Initially we perform data cleaning by removing all the null values and outliers from the various columns in the dataset and then we implement different ML models on the cleaned dataset to predict the price of cars.

Data visualization is used to examine the relation between the features.

Performing different ML models, we aim to get a better result or less error with max accuracy.

The purpose here is to predict the price of 458213 data entries and 26 predictors.

From the result table, we conclude that XGBoost is the best model for the prediction of the used car prices as XGBoost is a regression model giving the best R2 Score values.