INSTITUTE FOR ADVANCE COMPUTING AND SOFTWARE DEVELOPMENT AKURDI, PUNE.

Documentation On,

"Auction Car Price Prediction"

e-DBDA MAY 2021

<u>*Submitted By:*</u>

Group No: 3

Ganesh Raju Akhade. **(1303)**

Kamlesh Dattatray Vidhate **(1321)**

Mr. Prashant Karhale
Centre Coordinator

Mr. Akshay Tilekar
Project Guide

A

PROJECT REPORT ON

“Auction Car Price Prediction”

Submitted by

Group No: 3

GANESH RAJU AKHADE **(1303)**

KAMLESH DATTATRAY VIDHATE **(1321)**

Centre Coordinator

Mr. PRASHANT KARHALE

Under the Guidance of

Mr. AKSHAY TILEKAR

In the fulfillment of e – Diploma in Big Data Analytics course from

Institute for Advance Computing and Software Development Akurdi, Pune

in the academic year

May 2021 – Sept.2021

e-DBDA

May 2021

Institute for Advance Computing and Software Development

Akurdi, Pune. 411044

ACKNOWLEDGEMENT

It gives us immense pleasure to present our report for project on **"AUCTION CAR PRICE PREDICTION."** The able guidance of all teaching staff of this department made the study possible. They have been a constant source of encouragement throughout this project. We would like to express our grateful thanks to **Mr. Prashant Karhale Sir, Mr. Akshay Tilekar Sir** who guided us properly for this project. We would also like to express our sincere thanks to Institute for Advance Computing and Software Development Akurdi, Pune for giving us an opportunity to explore the subject and use our knowledge by conducting this project.

Group No: 1

Ganesh Raju Akhade (1303)

Kamlesh Dattatray Vidhate (1321)

e- DBDA, May 2021

Institute for Advance Computing and Software Development, Akurdi

PROJECT OVERVIEW

A Auction car price prediction has been a high interest research area, as it requires noticeable effort and knowledge of the field expert. Considerable number of distinct attributes are examined for the reliable and accurate prediction. To build a model for predicting the price of used cars in Bosnia and Herzegovina, we applied three machine learning techniques (Random Forest Regressor, Linear Regression, Gradient Boosting Regressor). However, the mentioned techniques were applied to work as an ensemble. The data used for the prediction was collected from the web portal autopijaca.ba using web scraper that was written in PHP programming language. Respective performances of different algorithms were then compared to find one that best suits the available data set. The final prediction model was accuracy of 97.71% .

CONTENTS

Chapter - 1

INTRODUCTION

Deciding whether a used car is worth the posted price when you see listings online can be difficult. Several factors, including mileage, make, model, year, etc. can influence the actual worth of a car. From the perspective of a seller, it is also a dilemma to price a used car appropriately [2-3]. Based on existing data, the aim is to use machine learning algorithms to develop models for predicting used car prices.

Accurate car price prediction involves expert knowledge, because price usually depends on many distinctive features and factors. Typically, most significant ones are brand and model, age, horsepower and mileage. The fuel type used in the car as well as fuel consumption per mile highly affect price of a car due to a frequent changes in the price of a fuel. Different features like exterior color, door number, type of transmission, dimensions, safety, air condition, interior, whether it has navigation or not will also influence the car price. In this paper, we applied different methods and techniques in order to achieve higher precision of the used car price prediction.

GLOSSARY

Training data: The data that are used to train regression models.

Validation data: The data that are used to test the performance of the regression model during the training process. Validation data are used to fine tune parameters in the classification model and the data should not be part of the training data.

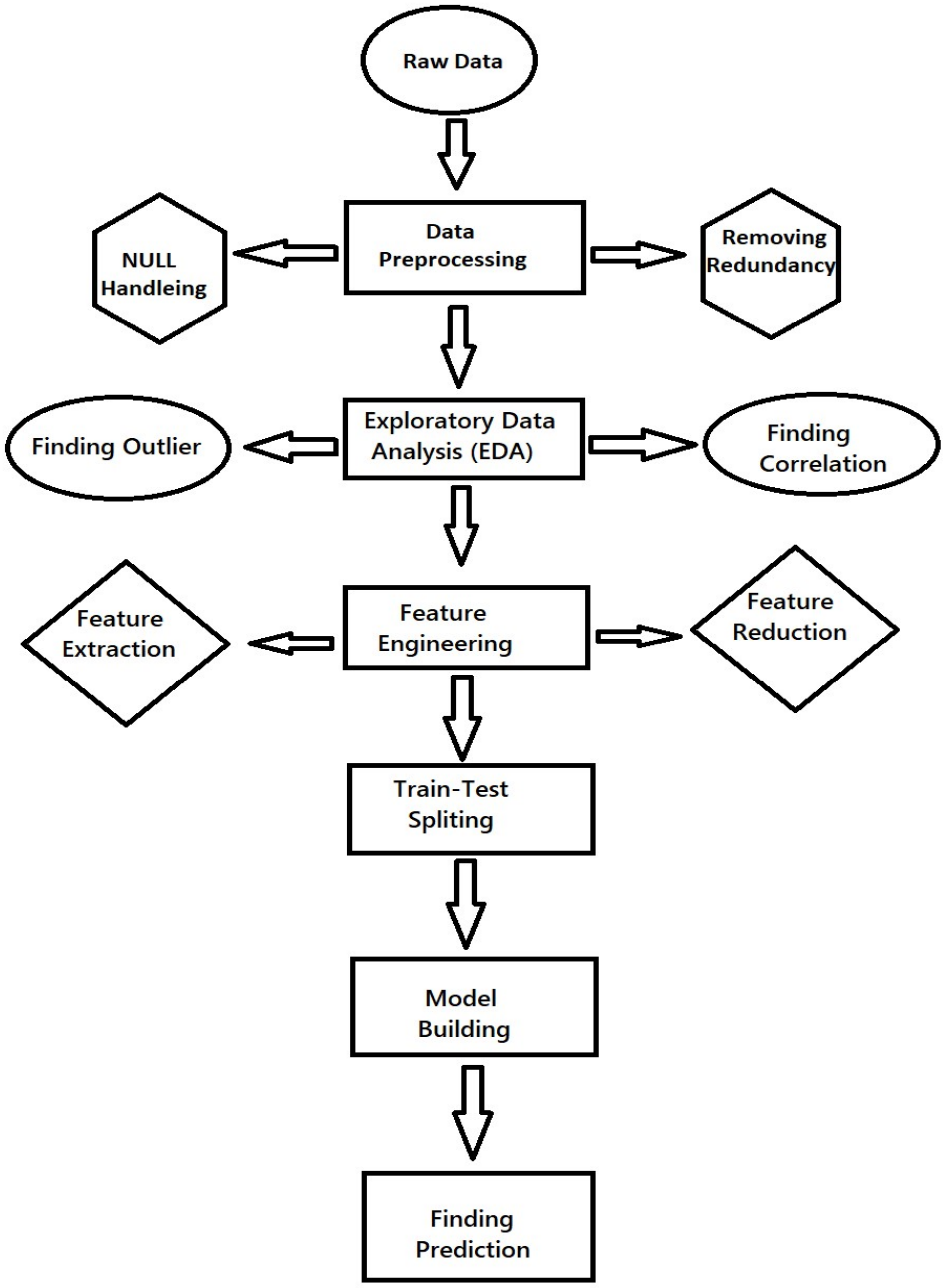
Testing data: The data that are used to test the performance of the trained regression model (after training process). Testing data are used to evaluate the final performance of the regression model. Testing data should not be part of training data or validation data. No fine tune should be made based on the result of testing data.

Overfitting: The regression model performs consistently better on training data than on validation data and testing data.

OBJECTIVE

- Develop a computational model for Auction Car Price Prediction.
- Analyse the computation model to better understand the important features for fall and rise of prices. Based on this understanding, validate, and further improve the performance of the model on price prediction.
- Based on the result of the computation model, develop a model design for used car price prediction.

BLOCK DIAGRAM



Chapter – 2

WORKING METHODOLOGY

We utilized several classic and state-of-the-art methods, including ensemble learning techniques, with a 90% - 10% split for the training and test data. To reduce the time required for training, we used 500 thousand examples from our dataset. Linear Regression, Random Forest and Gradient Boost were our baseline methods. For most of the model implementations, the open-source Scikit-Learn package [7] was used.

Compared to Linear Regression, most Decision-Tree based methods did not perform comparably well. This can be attributed to the apparent linearity of the dataset. We believe that it can also be attributed to the difficulty in tuning the hyperparameters for most gradient boost methods. The exception to this is the Random Forest method which marginally outperforms Linear Regression. However Random Forests tend to overfit the dataset due to the tendency of growing longer trees. This was worked upon by restricting the depth of trees to different values and it was observed that beyond limiting depth to 36 resulted in negligible improvement in prediction performance but progressively increased overfitting. As expected, light GBM performed marginally better than Random Forest Regressor and Linear Regression but had a significantly faster training time. Building up from the relatively good performance of Linear Regression.

Function Description are as follows:

mean : Mean value.

std : Standard deviation.

mad : Median absolute value.

max: Largest values in array.

min: Smallest value in array.

sma: Signal magnitude area.

Energy: Average sum of the squares.

iqr : Interquartile range.

entropy: Signal Entropy.

arCoeff : Autorregresion coefficients

correlation : Correlation coefficient.

maxFreqInd : Largest frequency component.

meanFreq : Frequency signal weighted average.

Car Price Prediction Process:

Data is collected from a local web portal for selling and buying cars autopijaca.ba [9], during winter season, as time interval itself has high impact on the price of the cars in Bosnia and Herzegovina.

Since manual data collection is time consuming task, especially when there are numerous records to process, a “web scraper” as a part of this research is created to get this job done automatically and reduce the time for data gathering. Web scraping is well known technique to extract information from websites and save data into local file or database. Manual data extraction is time consuming and therefore web scrapers are used to do this job in a fraction of time. Web scrapers are programmed for specific websites and can mimic regular users from website’s point of view.

After raw data has been collected and stored to local database, data pre-processing step was applied. Many of the attributes were sparse and they do not contain useful information for prediction. Hence, it is decided to remove them from the dataset. The attributes “state”, “city”, and “damaged” were completely removed.

Data cleaning is one of the processes that increases prediction performance, yet insufficient for the cases of complex data sets as the one in this research. Applying single machine algorithm on the data set accuracy was less than 50%. Therefore, the ensemble of multiple machine learning algorithms has been proposed and this combination of ML methods gains accuracy of 92.38%. This is significant improvement compared to single machine learning method approach. However, the drawback of the proposed system is that it consumes much more computational resources than single machine learning algorithm.

Features:

The following attributes were captured for each car: brand, model, car condition, fuel, year of manufacturing, power in kilowatts, transmission type, millage, colour, city, state, number of doors, four wheel drive (yes/no), damaged (yes/no), navigation (yes/no), leather seats (yes/no), alarm (yes/no), aluminium rims (yes/no), digital air condition (yes/no), parking sensors (yes/no), xenon lights (yes/no), remote unlock (yes/no), electric rear mirrors (yes/no), seat heat (yes/no), panorama roof (yes/no), cruise control (yes/no), abs (yes/no), esp (yes/no), asr (yes/no) and price expressed in BAM (Bosnian Mark).

Time Domain:

This section presents details about the most commonly used time-domain-based characteristics in the context of smartphone inertial sensors. These features can be divided into two types of functions: statistical functions and non-statistical functions. The statistical functions involve calculations such as minimum, maximum, average, standard deviation, among other formulas. The non-statistical functions involve several calculations such as areas, and calculation of Bins Distribution, among others. The set of time domain features as found in the literature. All of them are applied to the x, y, and z axes of the inertial sensors

Time ➔ min, max, amplitude, amplitude peak, sum, absolute sum, Euclidian norm, mean, absolute mean, mean square, mean absolute deviation, sum square error, variance, standard deviation, Pearson coefficient, zero crossing rate, correlation, cross-correlation, auto-correlation, skewness, kurtosis, area, absolute area, signal magnitude mean, absolute signal magnitude mean, magnitude difference function.

Among the features mentioned above, some special features can generate other new features through a process of chaining mathematical functions. For example, the signal magnitude feature can be combined with other features, such as mean and variance, and generate new features from this combination. The same happens with the features based on the vertical and horizontal components of the signals. In addition, the signals generated by these special features present sizes equal to the size of the original signals, while the other normal features generate compressed signals with sizes equal to the number of time windows defined in the segmentation step. In other words, these features work as data fusion techniques, since the coordinates x, y and z are transformed into only one axe. For these reasons, we classify the features with this type of behavior as low-level features, where the extraction process is performed in the raw data.

Frequency Domain:

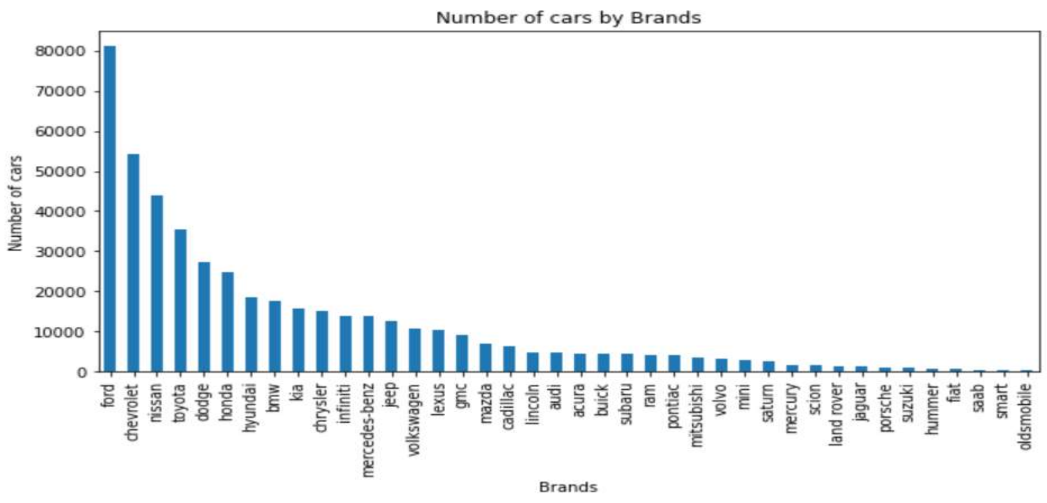
This section presents details about the most frequently used frequency domain features in the context of smartphone inertial sensors. These features present an alternative to signal analysis based on the frequency spectrum of the values of a certain time window. The features are calculated based on the low-level fast Fourier transform (FFT) or Wavelet features. Following shows the set of frequency features found in the literature. All of them are applied to the x, y, and z axes of the inertial sensors.

Frequency ➔ Energy, energy normalized, power, centroid, entropy, DC component, peak, coefficient sum.

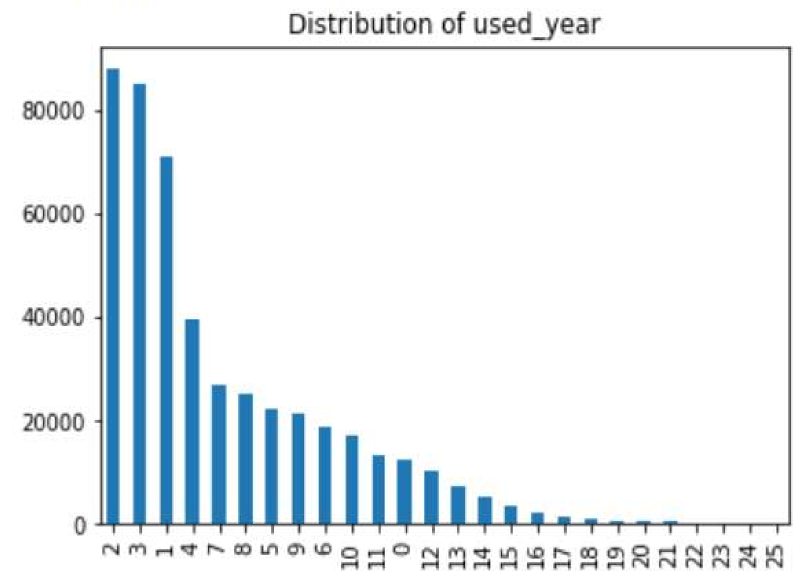
The frequency domain features described above depend strictly on the low-level features. Both transformations consist of a mathematical tool that transitions between variables over time for frequency variables, that is, the signal is decomposed into a set of real and imaginary values that represent components of waves called frequencies.

DATA VISUALIZATION: -

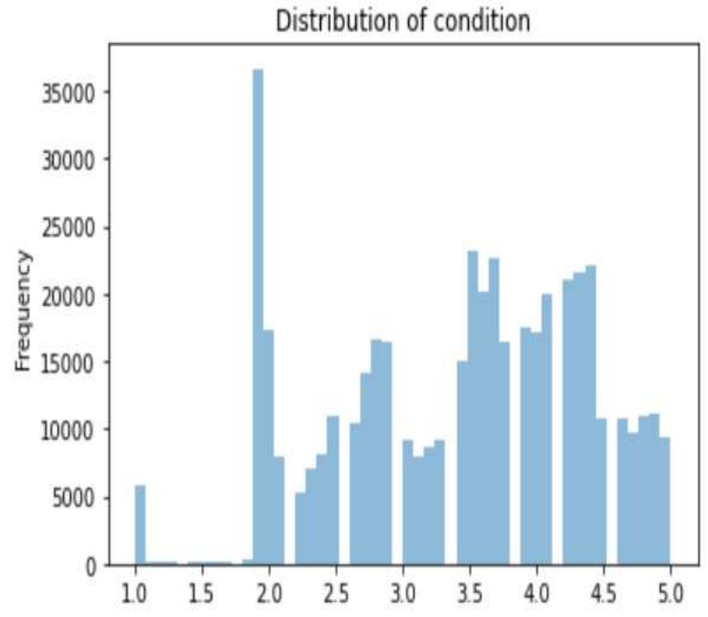
We performed EDA on the Car Auction dataset to understand the factors influencing the Auction of used cars. It was a large dataset containing about 558k rows. we have deleted the null values and irrelevant columns to get better understandings of the data. In this analysis, we plotted some graphs to show the distribution of selling price according to different factors like Odometer, Year of the model and the Brands of the vehicle. From this analysis, we could understand that, the factors like Odometer (the distance travelled by the vehicle), Brands of vehicle and Year of the model are mainly influencing the auction of used cars. Cars with lesser odometer are getting good selling price in the auction.



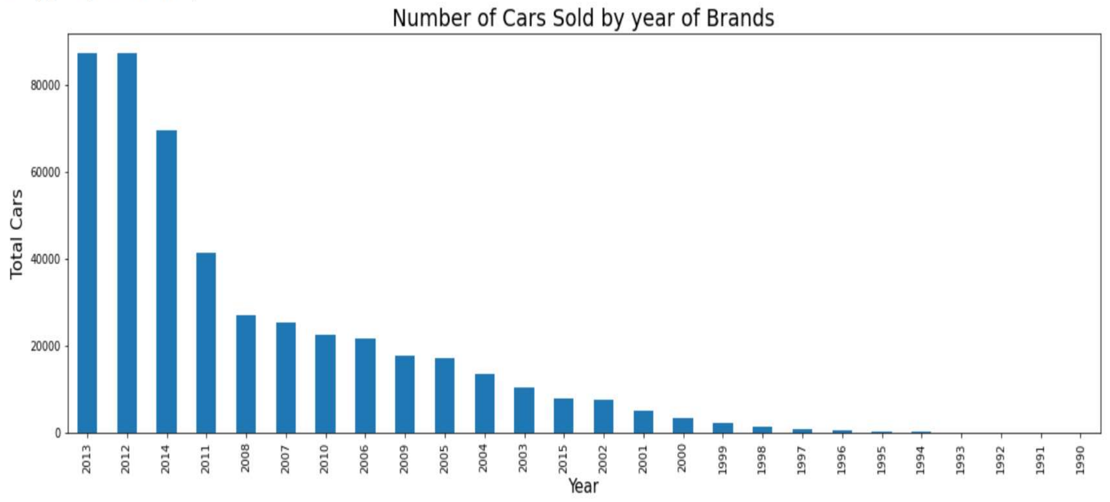
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3799739
```



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f37b6022450>
```



```
Text(0, 0.5, 'Total Cars')
```

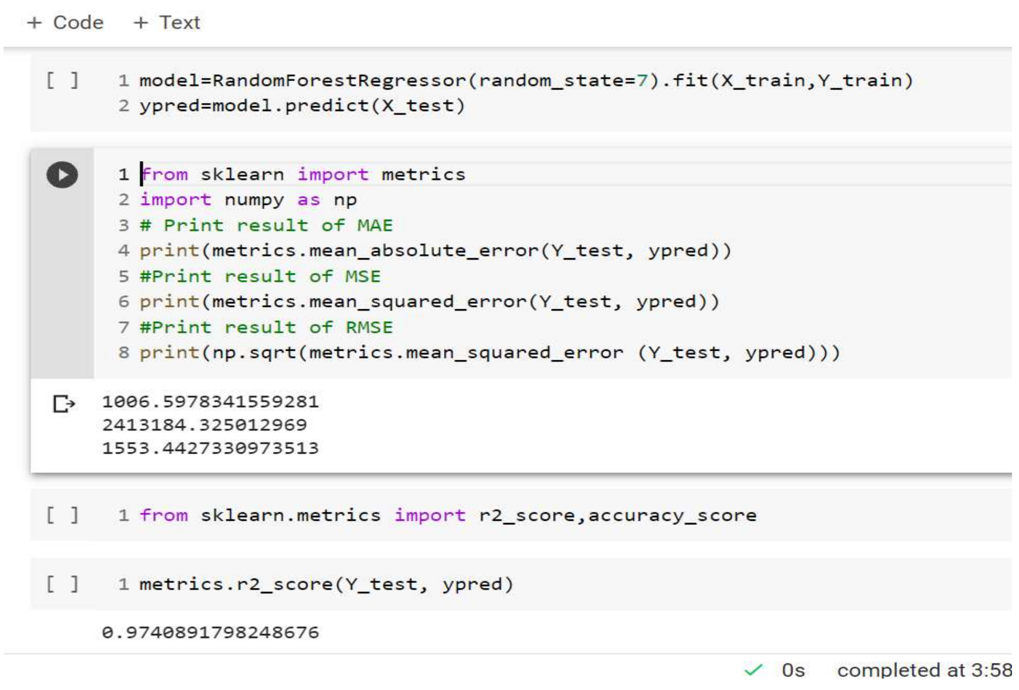


Based on above visualization, we clearly infer that the demand of used car increasing as we go further.

EXPERIMENTAL RESULTS: -

Random Forest Regressor:

Random Forest is an ensemble learning based regression model. It uses a model called decision tree, specifically as the name suggests, multiple decision trees to generate the ensemble model which collectively produces a prediction. The benefit of this model is that the trees are produced in parallel and are relatively uncorrelated, thus producing good results as each tree is not prone to individual errors of other trees. This uncorrelated behaviour is partly ensured by the use of Bootstrap Aggregation or bagging providing the randomness required to produce robust and uncorrelated trees. This model was hence chosen to account for the large number of features in the dataset and compare a bagging technique with the following gradient boosting methods.



```
model=RandomForestRegressor(random_state=7).fit(X_train,Y_train)
ypred=model.predict(X_test)
```

```
from sklearn import metrics
import numpy as np
# Print result of MAE
print(metrics.mean_absolute_error(Y_test, ypred))
#Print result of MSE
print(metrics.mean_squared_error(Y_test, ypred))
#Print result of RMSE
print(np.sqrt(metrics.mean_squared_error (Y_test, ypred)))
```

```
1006.5978341559281
2413184.325012969
1553.4427330973513
```

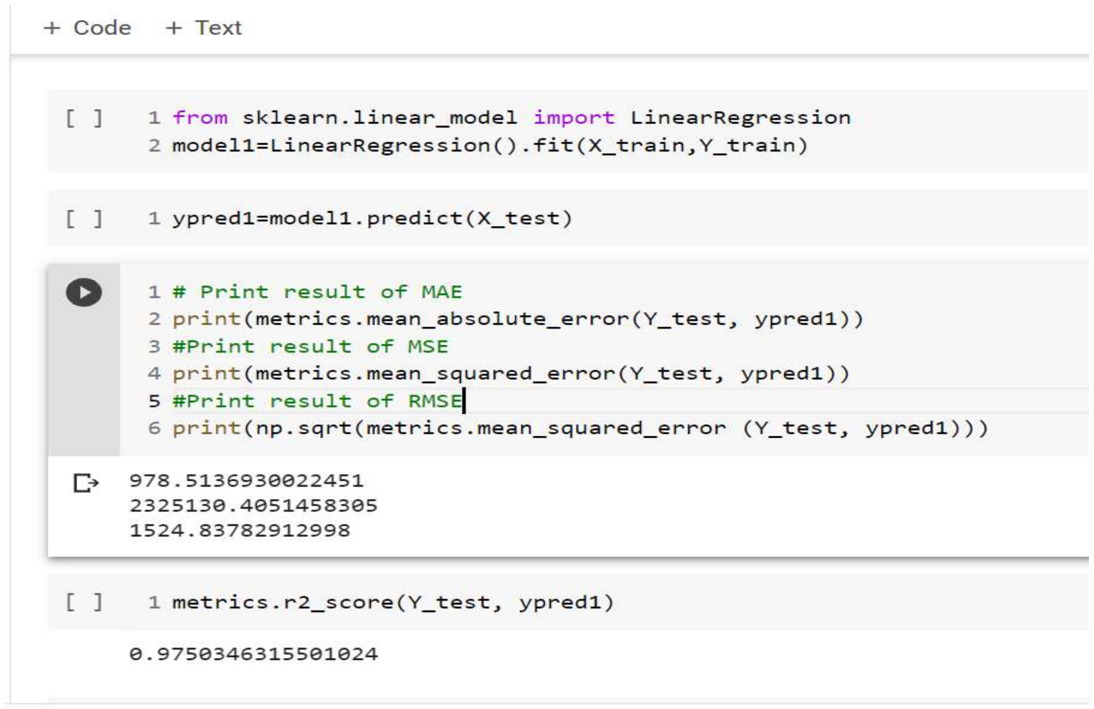
```
from sklearn.metrics import r2_score,accuracy_score
```

```
metrics.r2_score(Y_test, ypred)
```

```
0.9740891798248676
```

Linear Regression:

Linear Regression was chosen as the first model due to its simplicity and comparatively small training time. The features, without any feature mapping, were used directly as the feature vectors. No regularization was used since the results clearly showed low variance.



```
from sklearn.linear_model import LinearRegression
model1=LinearRegression().fit(X_train,Y_train)
```

```
ypred1=model1.predict(X_test)
```

```
# Print result of MAE
print(metrics.mean_absolute_error(Y_test, ypred1))
#Print result of MSE
print(metrics.mean_squared_error(Y_test, ypred1))
#Print result of RMSE
print(np.sqrt(metrics.mean_squared_error (Y_test, ypred1)))
```

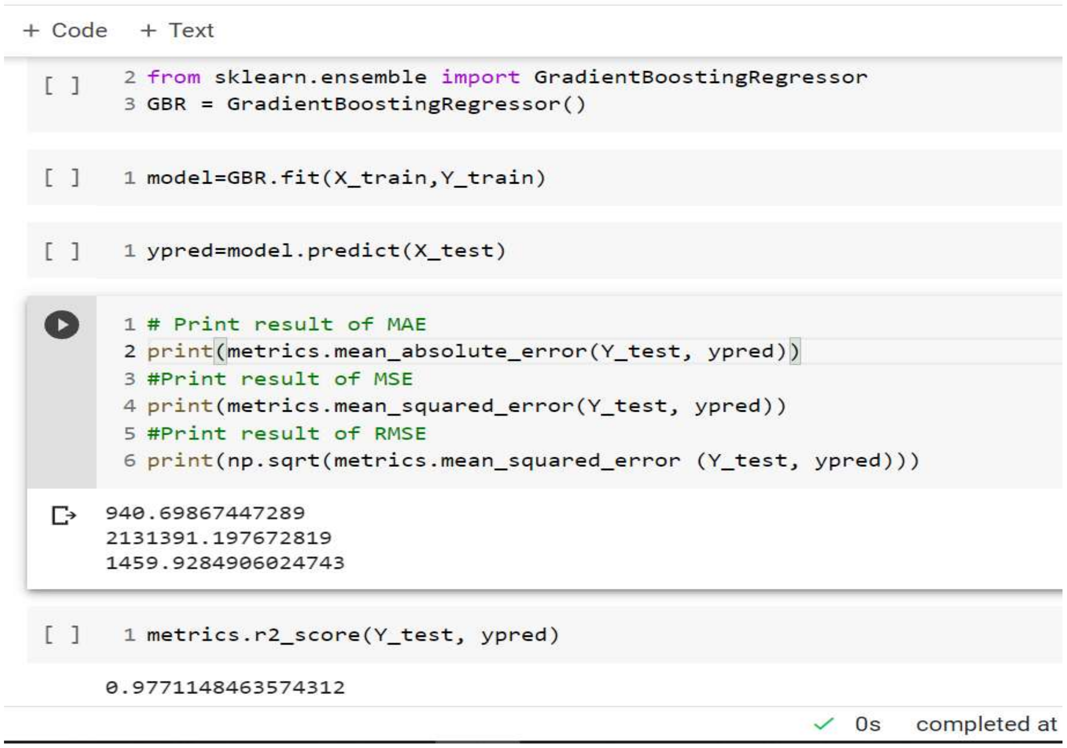
```
978.5136930022451
2325130.4051458305
1524.83782912998
```

```
metrics.r2_score(Y_test, ypred1)
```

```
0.9750346315501024
```

Gradient Boosting Regressor:

Gradient Boosting is another decision tree based method that is generally described as “a method of transforming weak learners into strong learners”. This means that like a typical boosting method, observations are assigned different weights and based on certain metrics, the weights of difficult to predict observations are increased and then fed into another tree to be trained. In this case the metric is the gradient of the loss function. This model was chosen to account for non-linear relationships between the features and predicted price, by splitting the data into 100 regions.



```
from sklearn.ensemble import GradientBoostingRegressor
GBR = GradientBoostingRegressor()
```

```
model=GBR.fit(X_train,Y_train)
```

```
ypred=model.predict(X_test)
```

```
# Print result of MAE
print(metrics.mean_absolute_error(Y_test, ypred))
#Print result of MSE
print(metrics.mean_squared_error(Y_test, ypred))
#Print result of RMSE
print(np.sqrt(metrics.mean_squared_error (Y_test, ypred)))
```

```
940.69867447289
2131391.197672819
1459.9284906024743
```

```
metrics.r2_score(Y_test, ypred)
```

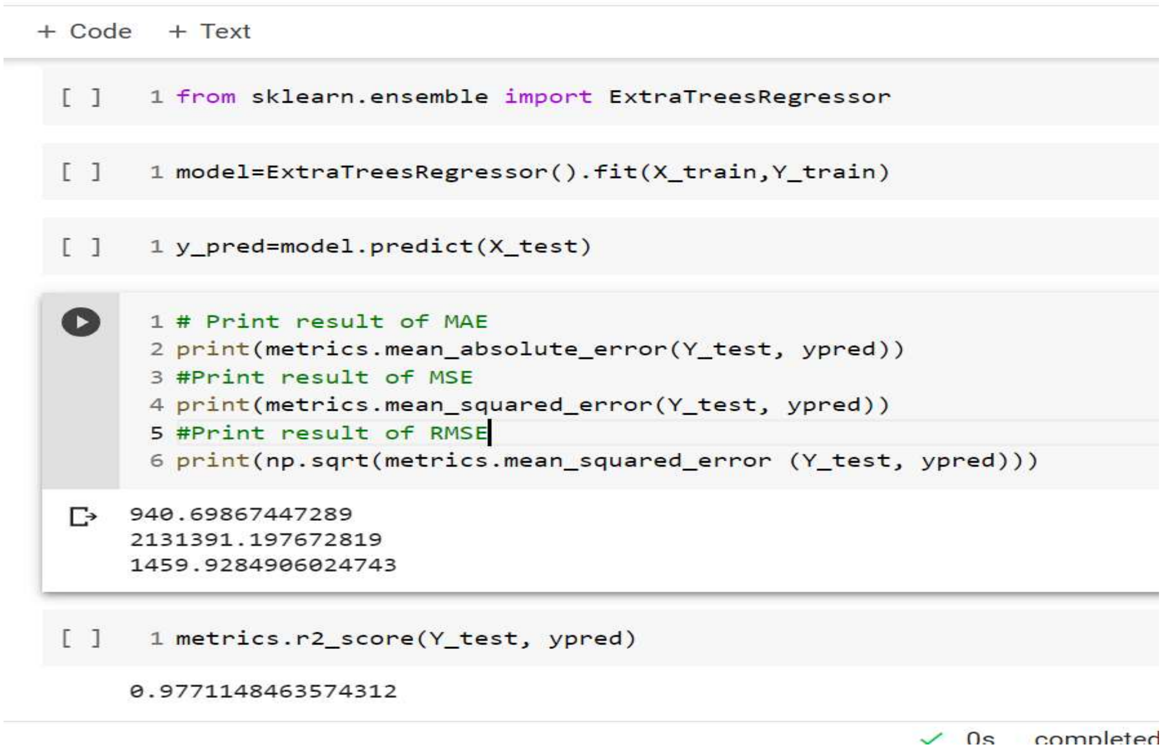
```
0.9771148463574312
```

Extra Tree Regressor:

Extra Trees for short, is an ensemble machine learning algorithm. Specifically, it is an ensemble of decision trees and is related to other ensembles of decision trees algorithms such as bootstrap aggregation (bagging) and random forest.

The Extra Trees algorithm works by creating a large number of unpruned decision trees from the training dataset. Predictions are made by averaging the prediction of the decision trees in the case of regression or using majority voting in the case of classification.

The Extra-Trees algorithm builds an ensemble of unpruned decision or regression trees according to the classical top-down procedure. Its two main differences with other tree-based ensemble methods are that it splits nodes by choosing cut-points fully at random and that it uses the whole learning sample (rather than a bootstrap replica) to grow the trees.



```
from sklearn.ensemble import ExtraTreesRegressor
```

```
model=ExtraTreesRegressor().fit(X_train,Y_train)
```

```
y_pred=model.predict(X_test)
```

```
# Print result of MAE
print(metrics.mean_absolute_error(Y_test, ypred))
#Print result of MSE
print(metrics.mean_squared_error(Y_test, ypred))
#Print result of RMSE
print(np.sqrt(metrics.mean_squared_error (Y_test, ypred)))
```

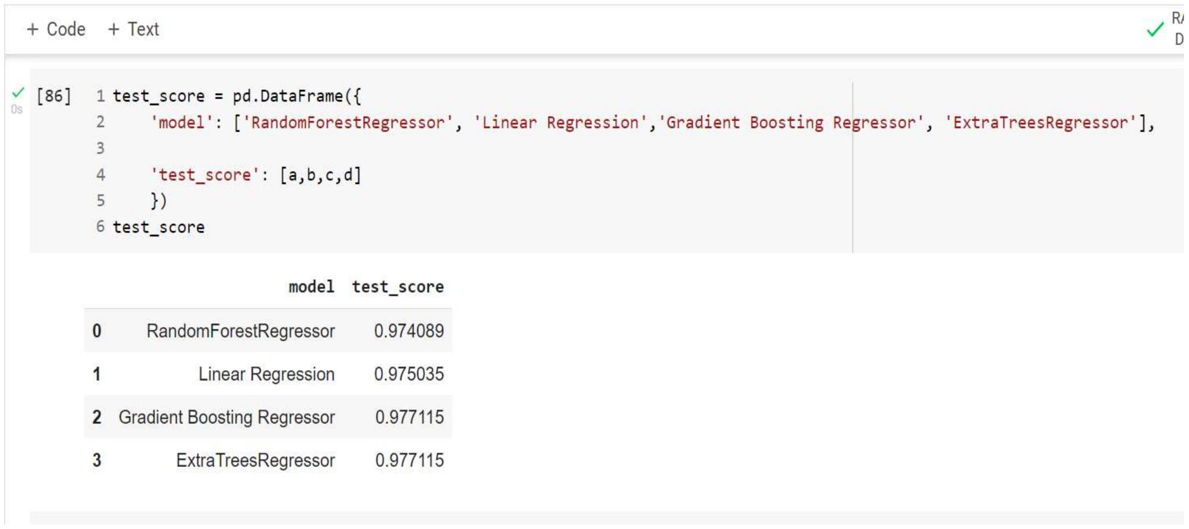
```
940.69867447289
2131391.197672819
1459.9284906024743
```

```
metrics.r2_score(Y_test, ypred)
```

```
0.9771148463574312
```

Model Analysis:

Test Accuracy:

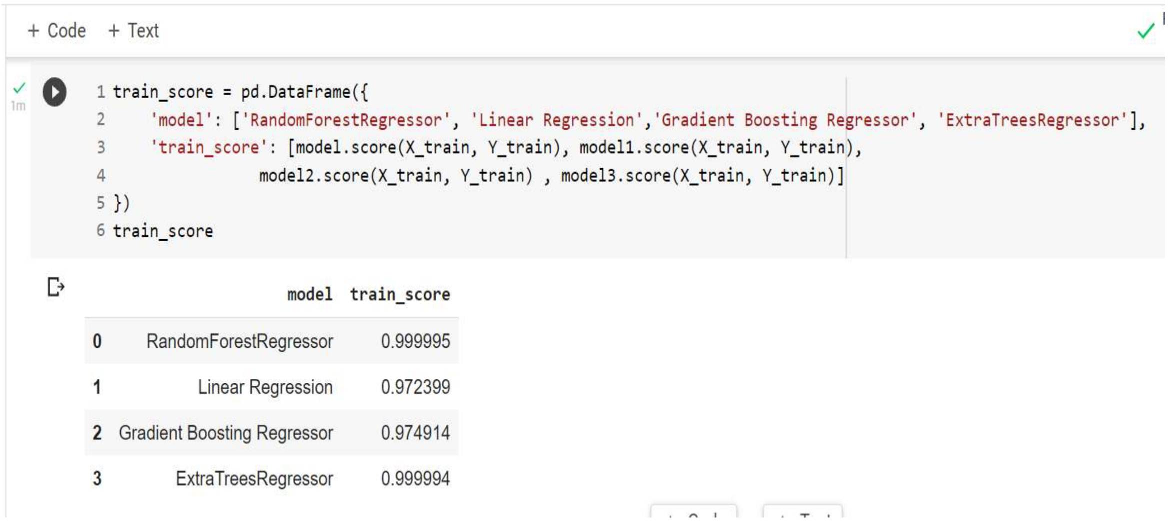


```
test_score = pd.DataFrame({
    'model': ['RandomForestRegressor', 'Linear Regression','Gradient Boosting Regressor', 'ExtraTreesRegressor'],

    'test_score': [a,b,c,d]
    })
test_score
```

	model	test_score
0	RandomForestRegressor	0.974089
1	Linear Regression	0.975035
2	Gradient Boosting Regressor	0.977115
3	ExtraTreesRegressor	0.977115

Train Accuracy:



```
train_score = pd.DataFrame({
    'model': ['RandomForestRegressor', 'Linear Regression','Gradient Boosting Regressor', 'ExtraTreesRegressor'],
    'train_score': [model.score(X_train, Y_train), model1.score(X_train, Y_train),
                    model2.score(X_train, Y_train) , model3.score(X_train, Y_train)]
})
train_score
```

	model	train_score
0	RandomForestRegressor	0.999995
1	Linear Regression	0.972399
2	Gradient Boosting Regressor	0.974914
3	ExtraTreesRegressor	0.999994

As per experimental analysis and model building, we have successfully got the accuracy of train and test data in Extra Tree Regressor Model is 0.99994 and 0.9771 respectively . This shows best results for Auction Car Price Prediction.

Chapter - 3

SIGNIFICANCE

Deciding whether a used car is worth the posted price when you see listings online can be difficult. Several factors, including mileage, make, model, year, etc. can influence the actual worth of a car. From the perspective of a seller, it is also a dilemma to price a used car appropriately [2-3]. Based on existing data, the aim is to use machine learning algorithms to develop models for predicting used car prices.

Chapter - 4

FUTURE SCOPE

For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates and train on clusters of data rather than the whole dataset. To correct for overfitting in Random Forest, different selections of features and number of trees will be tested to check for change in performance.

APPLICATIONS

- To predict car prices based on given features.
- To choose car as per his range.

Chapter - 6

CONCLUSION

Car price prediction can be a challenging task due to the high number of attributes that should be considered for the accurate prediction. The major step in the prediction process is collection and preprocessing of the data. In this research, PHP scripts were built to normalize, standardize and clean data to avoid unnecessary noise for machine learning algorithms.

Data cleaning is one of the processes that increases prediction performance, yet insufficient for the cases of complex data sets as the one in this research. Applying single machine algorithm on the data set accuracy was less than 50%. Therefore, the ensemble of multiple machine learning algorithms has been proposed and this combination of ML methods gains accuracy of 97.71%.

This is significant improvement compared to single machine learning method approach. However, the drawback of the proposed system is that it consumes much more computational resources than single machine learning algorithm. Although, this system has achieved astonishing performance in car price prediction problem our aim for the future research is to test this system to work successfully with various data sets.

BIBLIOGRAPHY

[1] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J.L. Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In Proceedings of the International Workshop of Ambient Assited Living, 2012.

[2] C. Cortes and V. Vapnik. Support-vector networks. Machine learning, 20(3):273–297, 1995.