Recent Trends in Intensive Computing
M. Rajesh et al. (Eds.)

doi:10.3233/APC210194

Price Prediction for Pre-Owned Cars Using Ensemble Machine Learning Techniques

Chetna Longani[a,1], Sai Prasad Potharaju [a], Sandhya Deore [a]
[a]*Dept of Computer Engineering, Sanjivani College of Engineering, Kopargaon, Maharashtra, India*

Abstract. The Pre-owned cars or so-called used cars have capacious markets across the globe. Before acquiring a used car, the buyer should be able to decide whether the price affixed for the car is genuine. Several facets including mileage, year, model, make, run and many more are needed to be considered before getting a hold of any pre-owned car. Both the seller and the buyer should have a fair deal. This paper presents a system that has been implemented to predict a fair price for any pre-owned car. The system works well to anticipate the price of used cars for the Mumbai region. Ensemble techniques in machine learning namely Random Forest Algorithm, eXtreme Gradient Boost are deployed to develop models that can predict an appropriate price for the used cars. The techniques are compared so as to determine an optimal one. Both the methods provided comparable performance wherein eXtreme Boost outperformed the random forest algorithm. Root Mean Squared Error of random forest recorded 3.44 whereas eXtreme Boost displayed 0.53.

Keywords. Ensemble, eXtreme Gradient Boost, Random Forest, Regression

1. Introduction

The prices of new cars are tuned by different facets as in manufacturer's cost, dealer's margin, transportation charges, GST levied on the car. More upon, the worth of a car turns down every year by 20%. Majority of the population cannot afford to buy a car from the showroom, rather prefer buying a pre-owned car. The global market for used cars valued USD 1332.2 billion in the year 2019. Market is looking forward to a stretch with a CAGR (Compound annual growth rate) of 5.5% during the next decade [3]. The used car market in India valued USD 24.24 billion during the same year. Like the global market, Indian used car market is also forecasted to register a CAGR of 15% by next five years [1]. Rapid spread of Covid-19 across the globe too has resulted in a downturn on public transport, which further has increased in demand for the pre-owned cars. A number of online sites have assembled the automotive industry at one place, so that the end user can buy or sell with a click. These sites use different algorithms for generating the price for the used cars, hence may place incompatible results to the users. More upon, these systems provide the sell and purchase mostly in urban areas. A unified algorithm must be entailed for regulating the price. This paper aims at designing a system that uses ensemble machine learning techniques to develop models

[1] Chetna Longani, Dept of Computer Engineering, Sanjivani College of Engineering, Kopargaon, Maharashtra, India
Email:chetnadang16@gmail.com.

That can predict prices for used cars. The model is trained for data of Mumbai region. The dataset used for the project [4] has 2454 records.

Prediction is nothing but estimation made out of observations. It uses observable phenomena so as to make future projections for the scene. The historical data forms our observations; this historical data is so large in volume that it is tedious to make conclusions just by looking at the data. Manual interpretation of the data is a soulless task. To reduce this pain and to get future predictions at a fingertip, different machine learning algorithms have to be deployed.

Machine learning when used for prediction, an algorithm is employed to train historical dataset and output a model. This model is fed with unseen data, and the prospect of a particular outcome is forecasted. Ensemble methods of machine learning blend different base models, thereby combining decisions from multiple models and providing an optimal model for prediction. Ensemble techniques use a mass of models into an account, further averages these models and delivers a proper model that can be used for prediction. In the proposed system, a random forest algorithm is deployed for predicting the prices of used cars. Further, a model is built using the eXtreme gradient boosting algorithm and prices of used cars are predicted. Performances of these two models are compared.

Various aspects that affect the price of pre-owned as in year of purchase, run of the car in terms of kilometres, its showroom price, its mileage, engine capacity, seating capacity, power capacity of car battery are considered while building the model. Other factors as in whether the seller is the owner or a dealer, is the owner first or second owner of the car, is the gear manual or automatic, the fuel used is petrol or diesel are also accounted for building the model. Fair number of perceptible attributes is employed. The motto of the developed system will be achieved if both buyer and seller will accomplish the deal at a fair price. One can buy or sell his car in a short time if he finds a fair price for his car in no time.

The organization of the paper is as follows. A review of relevant work is provided in the next section. The methodology is described in section III. Section IV includes the comparison of both the ensemble machine learning techniques to predict the car prices. Finally, the paper ends with a conclusion and setting a trend for future work.

2. Literature Survey

In the literature, few researchers applied various machine learning techniques to predict the car cost as per the given requirements. In the research article [5], the author investigated the application to predict the cost of cars which are used in Mauritius city. Author used techniques like multiple linear k-nearest neighbours, naïve bayes and decision trees, regression analysis to make the predictions of car cost. He has used multiple linear regression analysis to find out correlations between different features. In KNN, the author has taken only three attributes as marks, year and cylinder volume to classify new samples. Using this, the author wants to confirm that cars with higher values for cylinder volume have higher normalized values than cars with lower values. The main weakness he found was that continuous values with output classes are not that much controlled by decision trees and naïve bayes. Hence, the author classified the price attribute into the range of price classes. The prediction is done on two car types where samples are taken from old newspapers.

The researchers applied three machine learning technologies namely, Artificial Intelligence (AI), Support Vector Machine (SVM) and Random Forest (RF) separately [6]. Authors collected data through different web portals. They used RF for classification and regression. ANN was used for adjusting the weights between neurons. In SVM, the model is trained in such a way that the input data is classified into two categories. With these three single machine learning algorithms, authors found that these are not reliable methods for prediction of car prices. Then authors used the ensemble method that combines three ML algorithms to classify the price of a car as cheap, moderate and expensive.

Wu et al. [7] conducted a study on car price prediction. For this they used adaptive neurofuzzy inference systems (ANFIs). Three attributes were considered for forecasting the price of car as marks on the car, engine style and manufacturing year. They also did the study of ANN with back propagation and compared it with ANFIs. Through their work, they concluded that ANFIs have more accurate probability to forecast car prices than ANN with BP. The model is trained to generate relationships between any input and the outputs of complex systems, and also advises on where to sell the car. K-nearest neighbor machine learning algorithm is used for forecasting.

In the research thesis [8], the author focused on depreciation rate variation between cars with hybrid engines and those with traditional engines. Multiple variable regressions were used to analyze each independent variable on car resale values. Six segments selected by author from car and drive.com for collecting data. Through Correlation matrix age, miles, make and mpg(miles per gas) factors were used for prediction.

Author Nabarun Pal et al. in paper [9], used random forest, a supervised learning method, to forecast the prices of used cars. By selecting the most correlated feature, the model can predict the price of cars accurately. Researchers have tried both linear and random forest regression methods. From their study, they found that random forest is better than linear regression. A grid search algorithm was used to find the optimum number of trees. From that they concluded that with 500 decision trees forest accuracy is good. In case of a regression problem, they took the maximum number of features from the input data set. In case of classification, they took Square root of some features. Hence the Problem they converted into a regression problem as random forest is used for cost prediction.

Ning Sun et al. in paper[10], to analyze the price for each type of vehicle they used the optimized back propagation neural network algorithm. In order to improve the convergence speed of the network topology and also to improve accuracy of the prediction model, a back propagation neural network algorithm is used. The LB-MCM method is used for selecting the number of hidden neurons. Using this method the speed of network structure gets improved and the neuron selection efficiency also gets improved. The deviation and weights of the network gets updated and trained by a back propagation algorithm in such a way that the output vector gets closed to the expected vector. When the number of iterations reaches the highest thresholds and misclassification rate is less than the given threshold then training gets terminated.

Extreme Gradient Boosting or XGBoost [11] is one of the most popular machine learning models in current times. ForeXGBoost is the technique for prediction which uses the sliding window to extract historical sales and production of data features. In this method, leverage parallel computing is used to reduce the training overhead. XGBoost has such features so that it can significantly improve the regularization, parallel processing and in the other data prediction and classification tasks, such as web text classification, malware classification, sales prediction, customer behaviour prediction, rate forecast and product classification.

3. Methodology

Data collection is the most prime step for any project. We have designed the system for used cars in the Mumbai region, for which the data of used cars is collected using [4] as on 15-March-2021. Total 4057 records were scrapped using the Beautiful Soup (BS4) package. The data fetched composed of null records as well. After the null records are curtailed from the data, 2454 records are retained. The features captured for each car include Year (year of purchase), Seller Type (Dealer/Individual), Driven, Nonowners (First/second/third or above owner), Fuel_Type(Petrol/Diesel/CNG), Gear Type(Manual/Auto), Used_years_2021, Kmpl(Kilometres per litre), Engg Capacity (Cubic capacity), Max Power(Brake Horse Power), Seat_Capacity(4/5/8 seater), Onroad_Price(in lakhs), Sell_Price, Model(Brand). Sell Price is the dependent variable.

We used ensemble machine learning techniques to implement the system. Ensemble techniques build multiple models, and then blend them. Thereby produces upgraded results than a single model would. We can train an ensemble and further use it to make predictions. Hence, an ensemble is a supervised learning algorithm. Using different ensemble methods, we can combine various models, thereby, moving on the path of achieving better accuracy. Suppose that you have designed an android application, before making it public you wish to know its ratings. What you can do is either ask your friends or family or colleagues to rate your app. This process would give you limited feedback. How about cumulating the reviews from fifty or more people who could be your family, friends or even strangers? Now the response that you will seek will be more assorted as the people in the closure possess different skills. This task of accumulating feedback will give you honest and accurate ratings. We can here conclude that a group of miscellaneous people make better decisions as compared to an individual. And the same is true if rather than using a single model, we use a group of diverse models. To achieve diversification in machine learning, we have ensemble techniques. Bagging, boosting, stacking and voting are famous ensemble methods. Bagging configures one ensemble model by deploying Bootstrapping as well as Aggregation, where bagging replaces observations from original datasets and creates multiple subsets holding observations. Further, a base model is fabricated for each of these subsets, which is run in parallel for each of these subsets, independent of each other. At the end, the results from these models are aggregated to give a final prediction. Unlike bagging, the boosting process works sequentially on models. Here, different models are erected; each of the subsequent models corrects errors of the previous model. The weighted mean of all the models produces the final outcome, thereby combining weak learners to form a strong learner. Each of these models contributes to boost the performance of the ensemble. Stacking is an ensemble technique that uses predictions outputted from multiple models to construct a new model, which is further used to make predictions on the test set. Random Forest is a bagging algorithm whereas XGBoost is a boosting algorithm; these algorithms are used to implement the proposed system.

Random Forest algorithm is a popular supervised machine learning algorithm that relies on the concept of ensemble learning and can be deployed for both classification and regression problems in machine learning. It operates by fabricating a number of decision trees during the training phase, further outputs the class. The class outputted is the mode of classes if the problem under consideration belongs to classification while in case of regression outputs the mean prediction of individual trees. Being based on bagging ensemble learning, it deploys multiple uncorrelated decision trees on various subsets of a given dataset. Each of these decision trees outputs a certain prediction. Based on the average of predictions, final output is predicted. The proposed system is a regression-based task, where we have to predict output labels that should be continuous numeric values. Hence, the average of previously observed labels will give us a final prediction. The greater number of trees in the forest makes a robust forest that leads to accurate and stable prediction.

3.1. Algorithm for Random Forest

i. Select random samples from a given dataset and build multiple subsets.
ii. Build a decision tree associated for every subset.
iii. Every decision tree will output a prediction.
iv. Take the average of these predicted values.
v. The average value will be the final prediction.

Figure 1 depicts how random forest produces the predicted value of an unseen data point.

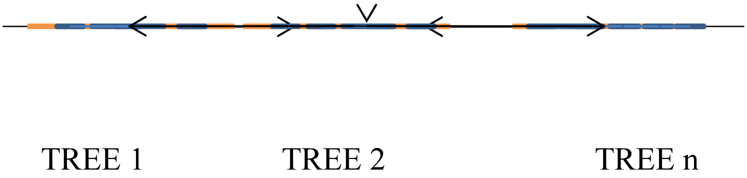


Figure 1. Test sample prediction by random forest

Being a decision tree-based ensemble machine learning algorithm, XGBoost aggregates output of several models and is one of the sturdiest techniques for fabricating predictive models. It uses a gradient boosting framework where errors are minimized by summing up weak learners applying gradient descent optimization algorithms. Boosting, being an ensemble technique, generates new models by correcting the errors made by preceding models. The process of sequentially adding models continues until no new model can be added further. Gradient Descent technique is used by the gradient boosting algorithm (GBM) to reduce the loss and to add a new model. Every decision tree takes a different subset of features; hence individual trees are different from each other. More upon, every successive tree takes the error of the preceding tree into account. In order to rectify the final yield of the model, value yielded from the new tree is added to the output of the existing sequence of trees. Process of XGboost is shown in Figure 2.

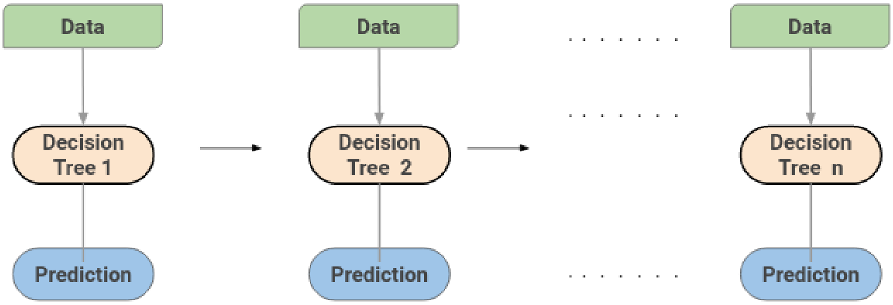
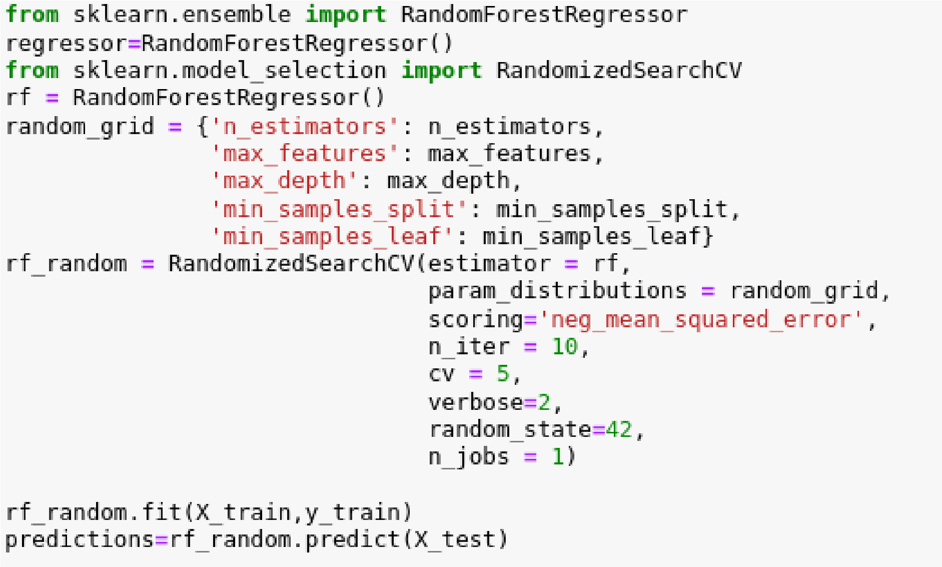


Figure 2. XGBoost prediction process

Extreme Boosting uses GBM at its core. XGBoost works on similar lines as in boosting weak learners using the gradient descent algorithm. XGBoost is used to address both classification and regression problems. XGBoost provides improvement over the base GBM framework by adding system optimization and enhancement in algorithms. The task of building a sequential tree is addressed using parallelized implementation that results in improved algorithmic performance. Using depth first approach for tree pruning, there is significant improvement in computational performance. It provides auto tree pruning so that decision trees do not grow after a certain limit. Cache awareness, out-of-core computing are offered in XGBoost algorithm which results in hardware optimization. It supports distributed computing for training very large models using a cluster of machines. The algorithmic enhancements such as Regularization, sparsity awareness, cross validation is supported by the algorithm. The algorithm also handles missing values on its own.

4. Implementation and Results Analysis

The system is implemented in python using anaconda as IDE. Sklearn tool is used. A python package sklearn. ensemble. Figure 3 Random Forest Regressor is used to implement the random forest algorithm. estimators i.e., the number of trees in the forest is set to 400, min_samples_split which is the minimum number of samples required to split an internal node is set to 5. The parameter min_samples_leaf that explores the minimum number of samples in newly created leaves is set to 5, max_features is the number of features to be considered while best split is set to auto. The parameter max_depth, the maximum depth of the tree is set to 15. Further RandomSearchCV is used to implement the fit and score method. neg_mean_squared_error is passed as a scoring method that computes mean squared error.

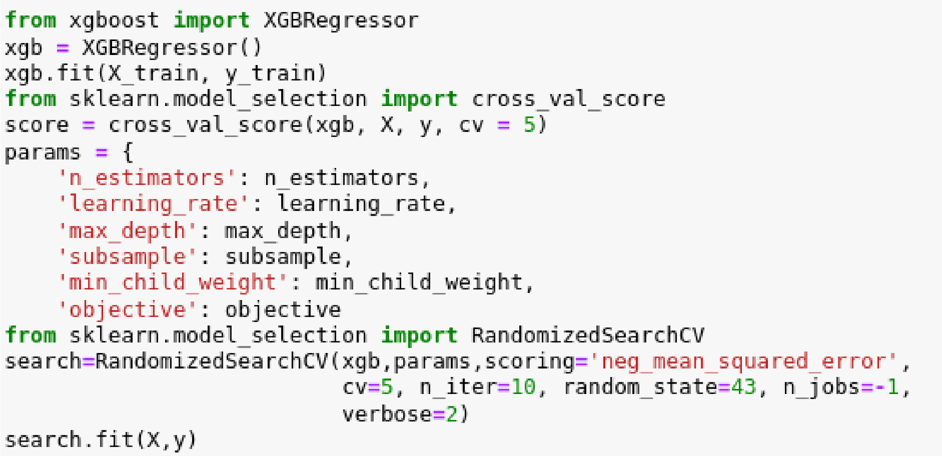


```
from sklearn.ensemble import RandomForestRegressor
regressor=RandomForestRegressor()
from sklearn.model_selection import RandomizedSearchCV
rf = RandomForestRegressor()
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf}
rf_random = RandomizedSearchCV(estimator = rf,
                              param_distributions = random_grid,
                              scoring='neg_mean_squared_error',
                              n_iter = 10,
                              cv = 5,
                              verbose=2,
                              random_state=42,
                              n_jobs = 1)

rf_random.fit(X_train,y_train)
predictions=rf_random.predict(X_test)
```

Figure 3. Using RandomForestRegressor

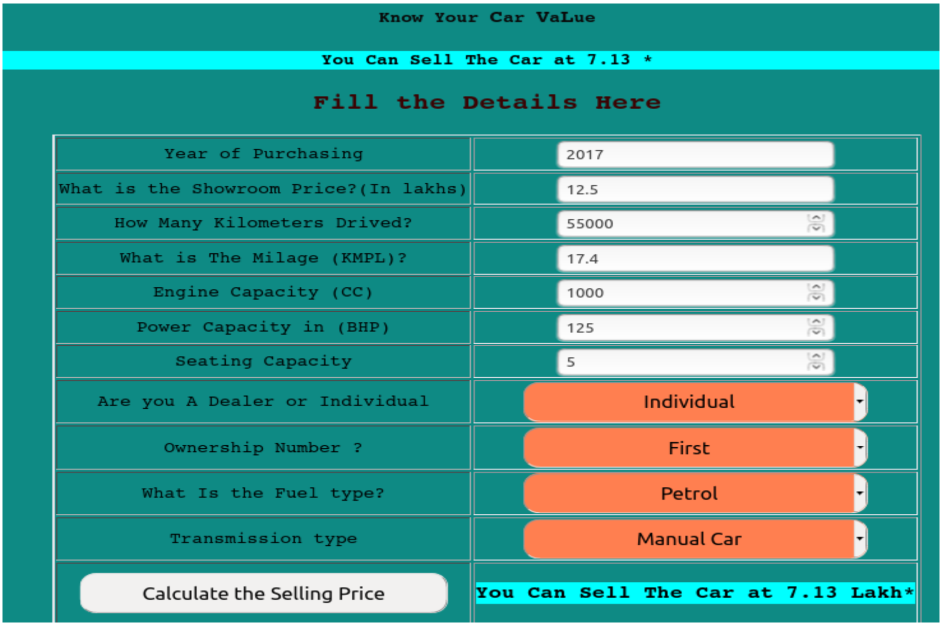
The used car price prediction system is also implemented using a scalable, portable, accurate machine library namely XGBoost library. It is an open-source library that provides effective implementation of Gradient Boost algorithm. The parameters used are n_estimators which is the number of trees to be built with a value of 400, learning_rate which is step size shrinkage used to prevent overfitting is set at 0.05, subsample which is the percentage of samples used per tree is set as 0.7. The parameter max_depth regulates how deep each tree should grow during any boosting round is set as 20, min_child_weight is 6. The parameter objective is used to specify the loss function to be used.



```
from xgboost import XGBRegressor
xgb = XGBRegressor()
xgb.fit(X_train, y_train)
from sklearn.model_selection import cross_val_score
score = cross_val_score(xgb, X, y, cv = 5)
params = {
    'n_estimators': n_estimators,
    'learning_rate': learning_rate,
    'max_depth': max_depth,
    'subsample': subsample,
    'min_child_weight': min_child_weight,
    'objective': objective
from sklearn.model_selection import RandomizedSearchCV
search=RandomizedSearchCV(xgb,params,scoring='neg_mean_squared_error',
                          cv=5, n_iter=10, random_state=43, n_jobs=-1,
                          verbose=2)
search.fit(X,y)
```

Figure 4. Using XGBRegressor

For this system a user interface is created using flask web framework. The interface will appear like the below figure 5. It is hosted on AWS for accessing remotely and it can be accessed through http://34.237.253.179:5001/



Know Your Car VaLue

You Can Sell The Car at 7.13 *

Fill the Details Here

Year of Purchasing	2017
What is the Showroom Price?(In lakhs)	12.5
How Many Kilometers Drived?	55000
What is The Milage (KMPL)?	17.4
Engine Capacity (CC)	1000
Power Capacity in (BHP)	125
Seating Capacity	5
Are you A Dealer or Individual	Individual
Ownership Number ?	First
What Is the Fuel type?	Petrol
Transmission type	Manual Car
Calculate the Selling Price	You Can Sell The Car at 7.13 Lakh*

Figure 5. Web User Interface for prediction

5. Results

Metrics considered to test the strength of the algorithm are Mean Absolute Error, Mean Squared Error, Root Mean Squared Error. These three metrics are used to evaluate both the regression algorithms. The three metrics have lower values for Xgboost, hence it has higher accuracy over the random forest algorithm. Table 1 shows the result analysis of both the algorithm.

Table 1. Result Analysis

Algorithm	**MAE**	**MSE**	**RMSE**
Random Forest	1.13	11.89	3.44
XGBoost	**0.17**	**0.28**	**0.53**

Graphical representation of the result analysis is given in Figure 4. Lowest values are considered as the best result.

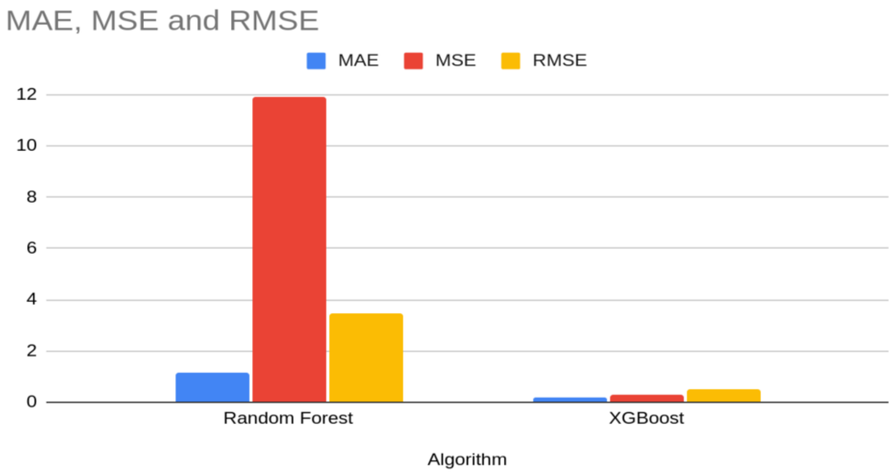


Figure 6. Graphical Representation of Result Analysis

6. Conclusion

The proposed system works well to predict a fair price for the pre-owned cars. The system skilfully projects prices for used cars in Mumbai region. The user of the system either the seller or the buyer, will get the honest price for the used car. Two popular ensemble machine learning algorithms namely Random Forest and XGBoost are deployed in order to implement a regression system for predicting used car prices. Both the techniques are comparable and offer high accuracy. Random Forest prevents overfitting by making use of more trees. With an ability to handle missing values, prevent overfitting, XGBoost is a widely used algorithm. As displayed by the results, XGBoost outperforms the Random Forest Algorithm. XGBoost is fast to execute and gives appreciable accuracy. The system proposed here is implemented for the Mumbai region only. However, it can be extended to other regions too, if the data available is in the suitable format.

References

[1] A report by Motor Intelligence on India Used Car Market - Growth, Trends, COVID-19 Impact, and Forecasts (2021 - 2026). https://www.mordorintelligence.com/industry-reports/india-used-car-market

[2] Sameerchand Pudaruth, Predicting the Price of Used Cars using Machine Learning Techniques, International Journal of Information & Computation Technology,ISSN 0974-2239 Volume 4, Number 7 (2014), pp. 753-764

[3] Enis Gegic et al., Car Price Prediction using Machine Learning Techniques, TEM Journal. Volume 8, Issue 1, Pages 113-118, ISSN 2217-8309, DOI: 10.18421/TEM81-16, February 2019.

[4] Jian-Da Wu,Chuang-Chin Hsu,Hui-Chu Chen, An expert system of price forecasting for used cars using adaptive neuro-fuzzy inference, Expert Systems with Applications 36 (2009) 7809–7817, 2008 Elsevier Ltd.

[5] Richardson M., 2009. Determinants of Used Car Resale Value, Thesis(BSc).,the Colorado College.

[6] Ning Sun, Hongxi Bai, Yuxia Geng, Huizhu Shi, Price Evaluation Model in Second-hand Car System based on BP Neural Network Theory, IEEE SNPD 2017, June 26-28, 2017.

[7] Zhenchang Xia, Shan Xue, Libing Wu, Jiaxin Sun, Yanjiao Chen & Rui Zhang, ForeXGBoost: passenger car sales prediction based on XGBoost, Distributed and Parallel Databases , volume 38, pages713–738(2020)SpringerLink,25 May 2020.