Performance Comparison of Different Machine Learning Algorithms in Predicting Used Car Prices

Roy David Alexis John Bosco[1] and Meenakshikumar Somasundaram[2]

[1] Department of Electrical and Computer Engineering, University of Waterloo, Canada rdalexis@uwaterloo.ca
[2] Department of Electrical and Computer Engineering, University of Waterloo, Canada m2somasu@uwaterloo.ca

Abstract. Owning a car is a big dream especially for people in the middle-class bracket. Some people may be able to afford a new car, whereas others prefer buying a used car that costs less compared to a new car. So, it is important to determine if the used car is worthy of the investment. The aim of this project is to determine the worth/value of a used car by predicting its price using different machine learning algorithms by taking several factors such as make, model, year, mileage, etc., into account. The predictions from these algorithms are then evaluated and compared against each other to find out which algorithm has delivered more accurate results. From our experiments with seven models, it is observed that the LightGBM model with an R squared value of 0.926 performs the best.

Keywords: car · prediction · price.

1 Introduction

With the increasing demand for private cars all around the world, there is a significant rise in the demand for used cars which creates a business opportunity for both the buyer and the seller. In several countries, people prefer buying used cars to new cars because of their affordability. Car sellers sometimes take advantage of this scenario by listing unrealistic prices for the used cars owing to the demand. Therefore, arises a need for a used car prediction system to determine the car value by taking a variety of features into consideration. Prediction of used car prices is not an easy problem because of the impact of several factors on car prices. Some of the important factors include its year of manufacture, manufacturer, model, type of engine it uses, distance traveled (mileage) and the type of fuel it uses. Other features such as a blind-spot monitor, sound system, air conditioner and GPS navigator may influence the price as well. Some of these features and their impact on price is discussed in detail in Section 3.

In this paper, the performances of different machine learning models such as Linear Regression, K-Nearest Neighbor, Decision Trees, Random Forest, Bagging

Regressor, XGBoost and LightGBM are compared based on certain evaluation metrics. The main objective of this paper is to find the best model in predicting the price of used cars.

The structure of this paper is as follows. In Section 2, other related works that predict the price of used cars are discussed. Section 3 talks about the dataset, its pre-processing and the implementation of different machine learning models. Section 4 consists of a comparative study of the learning models in predicting the prices for used cars based on two critical evaluation metrics. In Section 5, future improvements are discussed followed by the conclusion.

2 Literature Review

Several experiments have been done previously on used car prediction.

Pudaruth [1] worked on predicting the prices of used cars in Mauritius by comparing four machine learning techniques such as multiple linear regression, naïve Bayes, decision trees, and k-nearest neighbors. The author has created a used car data set from newspaper ads for over a month period. After cleaning, the dataset had less than one hundred records with just three car manufacturers. Their experiment concluded that kNN performs better than other machine learning techniques. The price prediction accuracy is not reliable because their dataset is limited, and the mean error has a huge variation with respect to different car manufacturers.

N. Monburinon, P. Chertchom, T. Kaewkiriya, S. Rungpheung, S. Buya and P. Boonpou [2] compared the performance of regression based-supervised machine learning models. A Kaggle Dataset based on a German e-commerce website was used for evaluation and prediction of car prices. From the dataset, they removed unnecessary features of the car which do not impact the price and applied label encoding to clean. They have also created a correlation matrix to find out the features and found that power, kilometer driven, year and gearbox significantly impacts the price of used cars. Mean absolute error was chosen as criteria to find out the best model and they observed that the gradient boost regression (0.28) performs better than multiple linear regression (0.55) and random forest (0.35).

Enis Gegic et al [3] applied three machine learning algorithms such as Artificial Neural Networks, Support Vector Machine and Random Forest on the dataset which was collected from the web portal autopijaca.ba using a web scraper. After cleaning incomplete information, less than eight hundred samples were chosen. The continuous features of cars such as year, mileage, price, etc., were converted to various categorical values. They built machine learning classifier models such as Artificial Neural Network, SVM and Random Forest and evaluated them. Their observation was that the prediction accuracy was less and they built ensemble model of all the previously mentioned classifiers. Firstly, Random Forest was applied to categorize samples into three categories and further applied SVM and ANN. The prediction accuracy with ensemble classifier was found to be more than doubled than single classifier. They took

the classification approach but for continuous datasets such as price prediction, regression is more preferable.

Pal, Nabarun et al [4] used the Random Forest method to predict the prices of used cars. They have taken used car dataset from Kaggle which was created from eBay-Kleinanzeigen which had over three hundred thousand records. They cleaned the dataset by removing outliers, removing car listings posted by dealerships, removing unsold cars, etc., Finally, they selected limited features of used cars which are considered important such as price, kilometers driven, make, vehicle type and applied Random Forest model with five hundred decision trees for price prediction. But they have not evaluated the performance of other machine learning techniques over their Random Forest model and also selected very limited features. Hence, it is not guaranteed that their model is the best in predicting the prices of used cars.

3 Implementation

The dataset chosen for this project is obtained from Craigslist which contained records of the largest collection of used cars and is available on Kaggle [6]. There are a variety of features present in the dataset which include price, year, manufacturer, model, condition, cylinders, fuel type, odometer, transmission, vehicle identification number, drivetrain type, size and color.

Following is the basic statistical information derived from the given dataset without data cleaning.

- Number of entries in the dataset is close to 4,36000.
- In the column *price*, the minimum value is 0 which implies that there are free cars available. These entries should be removed from the dataset. Similarly, the maximum value is 3.6 billion which is clearly an outlier.
- The maximum value of *odometer* column is 10 million miles which is again an outlier.
- The column *county* is totally empty.
- There are a lot of missing values among features.
- Many columns such as id, url, image_url and so on dont have any influence on the price of a used car at all.

3.1 Data Cleaning

Before starting our analysis, the first important step performed on the dataset was data cleaning. Data cleaning (or data cleansing) is the process of identifying incorrect, inaccurate, incomplete or irrelevant parts of the data and then replacing, modifying or deleting them.

3.1.1 Removal of outliers This section talks about the removal of outliers from the dataset. Following is the box plot for the *price* column which shows the distribution of price in the dataset (see Fig. 1). The price range between first

quartile to third quartile of our dataset was 4900-17989, but the maximum price was 3.64 billion, which was extremely high for a used car. Interquartile range was used to remove the outliers. From the statistical information mentioned above, we observed that there were free cars available in the dataset. So, a minimum threshold value for price had to be set. The threshold was set to 600 for the current dataset. Similar to the *price* column, there are outliers present in the

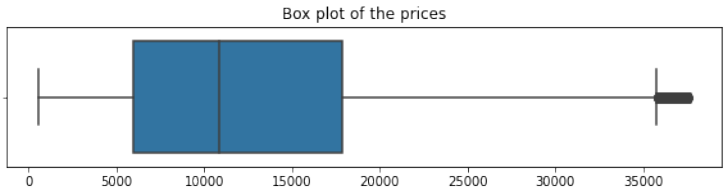

Fig. 1. Box plot displaying outliers for the price column

odometer column as well (see Fig. 2). The maximum value for *odometer* column is 10 million miles which is clearly an outlier. As opposed to the *price* column this column can accept lower values as there can be relatively new, used cars with very less mileage for sales.

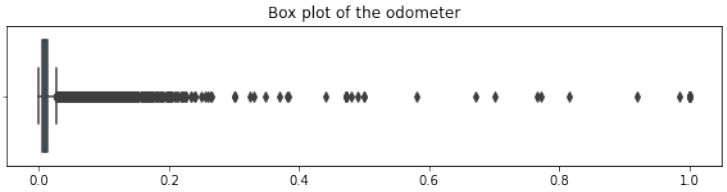

Fig. 2. Box plot displaying outliers for the odometer column

3.1.2 Removal of unrelated features In this section, the features that did not have any influence on the price of a car were removed. Those unrelated features include id, url, image_url, region, region_url, title_status, VIN (Vehicle Insurance Number), description and county. It can be seen from the dataset that most of the cars were being sold in the US and the used car prices did not vary much geographically across the countries or across the states in the US. As a result, features such as state, lat and long were removed.

3.1.3 Handling of missing values In this section, the focus was on the values that are missing with respect to each feature in the dataset. Percentage of missing values in each feature was calculated. Following were the measures taken in order to handle missing values in the dataset.

- Column *size* had to be removed because there were too much of missing values(65%) in the dataset.
- Columns such as year, model, fuel, transmission, drive, type and paint_color contained very few missing values. So, those corresponding rows had to be removed.

– For columns such as condition(36%) and cylinders(31%), the missing values were replaced with *null* values.

3.1.4 Removal of insensible data samples In this section, the samples that did not make any sense were removed. Generally, mileage is inversely proportional to price. The price of higher mileage(odometer value) cars tend to be low, whereas lower mileage cars tend to be expensive. However, the scatter plot(see Fig. 3) shows that there are some cars with very less mileages(which are almost like new cars) sold for extremely low prices. This scenario is impractical in the real world.

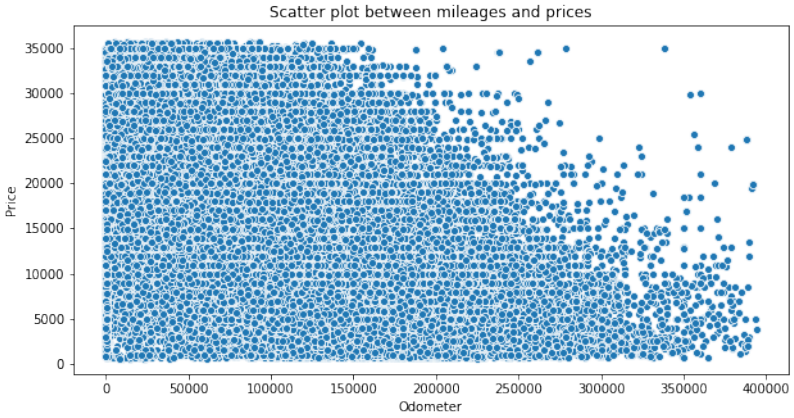

Fig. 3. Relationship between prices and mileages

So, only those samples where the sum of the values in the columns *price* and *odometer* is greater than 5000 were considered. Secondly, very old cars had not been taken into consideration. i.e., the samples older than 1960 were removed, the reason being that those samples were not sufficient in number and including them brings uncertainty to the prediction.

3.1.5 Removal of insufficient records In this section, the records that were less in number with respect to a feature had been removed. For example, when the samples were grouped by column *model*, there were some models with less than 50 records. These samples had to be removed as they were not sufficient for our prediction and also it will narrow down the capability of our model.

3.2 Data Visualization

After the data cleaning, the dataset was analysed. This involved a detailed study on the dataset and finding the relationship between the features. Any influence of other features on the column *price* had been identified. The distributions of both *price* and *odometer* look better after data cleaning(see Fig. 4).

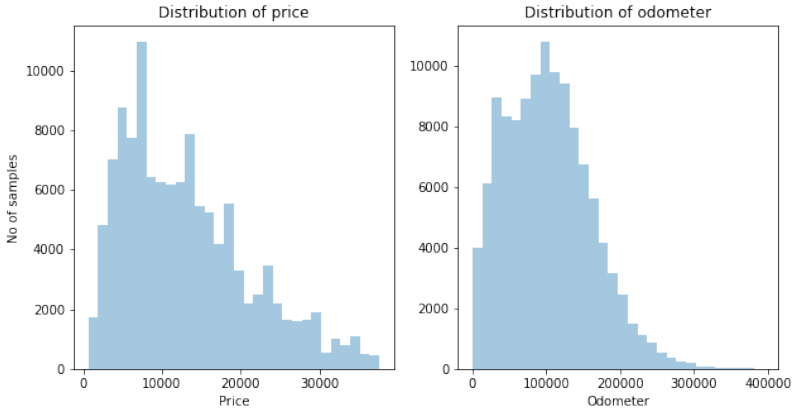

Fig. 4. Influence of color on price

There are 12 colors in the dataset including the customized colors. Of these, there are 11 different common colors such as orange, black, red, silver, grey, blue, white, green, brown, yellow and purple.

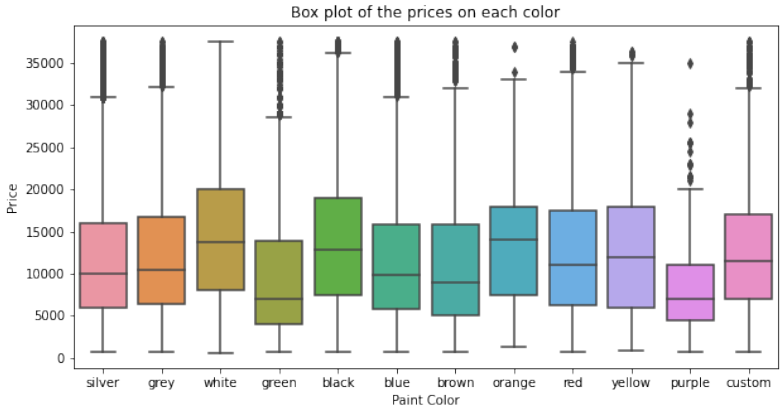

Fig. 5. Influence of color on price

It can be seen from the plot(see Fig. 5) that the top 4 selling car colors are white, black, orange and yellow. In contrast, the least selling car colors are green and purple. The statement above may not be completely correct because of the relatively less number of samples for orange, yellow and purple.

Following are the important points captured during our data analysis with other features(see Fig. 6).

– The prices for pickup cars, trucks and buses are high because they cost more when they are purchased as new vehicles. The prices for sedan, hatchback and mini-van are more stable and of similar range.

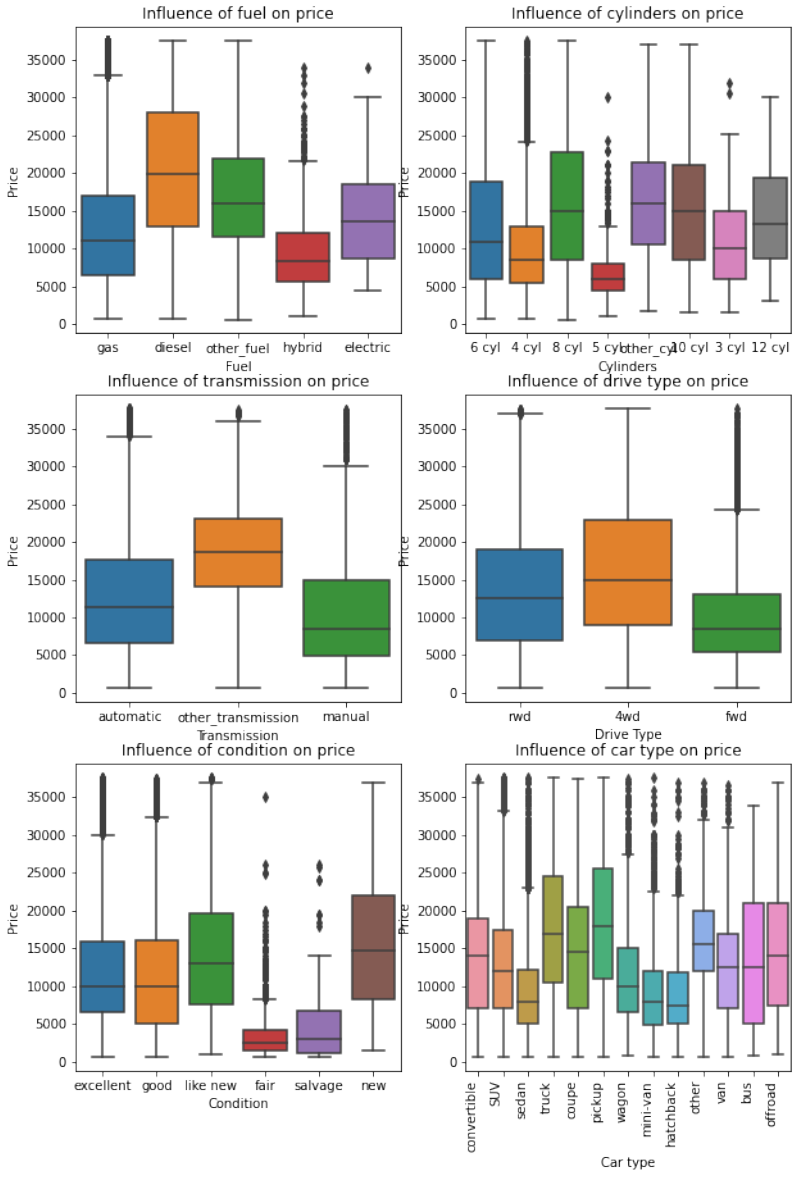

Fig. 6. Influence of type, condition, cylinders, fuel, transmission and drive on price

- New and like-new cars tend to be more expensive, while cars with fair and salvage conditions tend to be much cheaper.
- Cars with 8, 10, 12 or other cylinders tend to be more expensive, while 4 and 5-cylinder cars are cheaper.
- Diesel cars are more expensive than gas and hybrid cars.
- Cars with "other" transmission (possibly CVT) tend to be the most expensive. Cars with automatic transmission tend to be more expensive than those with manual transmission.
- Cars equipped with all-wheel drive tend to be more expensive than those with front-wheel drive.

After the analysis, the data was fed to the models. Before feeding the data, there were two preparation tasks to done. One is to transform the values in all categorical features(*model*,*condition*, *cylinders*, *fuel*, *transmission*, *drive*, *type*, *paint_color* to numeric values using One hot Encoding scheme. The other one is to normalize the numerical features(*year*, *odometer*) using StandardScaler method.

3.3 Regression Models and Tuning

Used car price prediction is a regression problem in which the price of a car is a dependent variable and the characteristics of the car(make, model, registration year, fuel type, transmission type, . . .) are independent variables. The input is denoted by $X = \{X^1, X^2, ..., X^N\}$ and the output is denoted by Y . The regression model represents the dependency relationship between Y and X

$$Y = f(X; \theta) \tag{1}$$

where f is a function modelling the dependency of the output on the input and parameters θ.

The dataset was then split into training, validation and test sets. Since the end goal is to find the model which predicts the most accurate price, various regression models including both simple and ensemble models were used for the prediction. The models were trained with hyperparameter optimization using k-fold cross validation and then exposed to the test dataset. To evaluate the performance of a model, root mean square error (RMSE) and r2-score were considered. The RMSE provides the error rate which is computed by taking the square root of the mean of difference of predicted values and actual values. So the lower the RMSE, the better the model.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y})^2} \tag{2}$$

The r2-score is the coefficient of determination which shows how well the predicted values match with the actual values. The r2-score is in the range of 0 to

1 and higher scores shows that the model is performing more accurately.

$$R^2 = 1 - \frac{\sum(y_i - \widehat{y})^2}{\sum(y_i - \overline{y})^2} \tag{3}$$

Then r2-score and root mean squared error (RMSE) were calculated and compared to find the best model for this dataset.

The regression models provided by sklearn[7], XGBoost and LightGBM[8] were used to predict the price of used cars. There are a plethora of simple and ensemble regression models available from sklearn library. The ensemble models are popular because the prediction is more accurate due to the ability to combine results from multiple models. The following subsection covers the overview of various models that have been used in this experiment and the tuning of their hyperparameters for the current dataset.

3.3.1 Linear Regression Linear Regression is a supervised machine learning algorithm and is the most commonly used model for predictive analysis. This is taken as the baseline model for this experiment. The linear regression model fits a linear line through the data points such that the residual sum of squares between the data points and the predictions are minimum. The model is represented by

$$y = Ax + B \tag{4}$$

Here A is the slope of the line and B is the intercept. The model has both slope and intercept as tuning parameters and the model was trained with the dataset with intercept enabled.

3.3.2 K-Nearest Neighbors K-Nearest Neighbors is a machine learning technique which can be used for both regression and classification. Basically, the knn model adopts similarity in features to predict any new observations. Particularly, in regression use case, the model does the prediction based on nearest neighboring datapoints by taking average of their values. To find the closest neighboring datapoints, distance is calculated from the new datapoint to every other datapoint in the training set and the neighbors with shortest distances are chosen for prediction. The model uses Euclidean and Manhattan distance for continuous variables and Hamming distance for categorical variables. To tune the model, cross validation was done with various neighbor values ranging from 1 to 15 and the value 4 was chosen as the best parameter. Apart from the neighbors, uniform weights were chosen for all of the closest neighbors.

3.3.3 Decision Trees Decision trees are supervised machine technique which predicts the target variable through learning decision rules from the given set of features present in the dataset. While building the tree from the root node, a series of decisions are made based on a set of information present in the data. In other words, the data is split based on information gain which is the amount

of information present in the data. The decision trees can be applied to both regression and classification problems. In this experiment, the new observations are determined by taking average of the existing observations present in that area. In regression trees, the mean square error is generally used to split the node into two or multiple sub-nodes. There are various tuning parameters provided by the model such as maximum depth of the tree which determines maximum number of sub-nodes from the tree, minimum samples per leaf which restricts the minimum number of data samples present in leaf node, etc., This model is tuned to keep the leaves as pure and samples per leaf is kept as 1 so that the prediction performance is improved.

3.3.4 Random Forest Random Forest is an ensemble learning technique that creates forest by generating and merging multiple decision trees randomly. Also, the trees learn on random sample and at every node splitting is performed by selecting random set of features. Hence, the issue of overfitting in a single decision tree is avoided here. There are various tuning parameters provided such as maximum depth of the tree, minimum samples per leaf, bootstrap, etc., After experiments, the maximum depth of the tree was configured to 30 and bootstrap was set to true to choose subsets of original data.

3.3.5 Bagging Regressor Bagging Regressor is an ensemble model that takes predictions from each base regressor which was fed with random sample of data, aggregating the prediction results and come up with final prediction. It uses a base estimator and fits it on subsets of data. For our experiment, a decision tree regressor was used as the base estimator and bootstrap was enabled so that the samples are drawn with replacement.

3.3.6 XGBoost XGBoost is a popular ensemble learning technique that implements gradient boosted decision trees. The boosting technique used here builds decision trees sequentially such that the upcoming trees minimizes errors from the previous tree. It also adopts gradient descent algorithm to minimize loss function when selecting new models. The models supports Gradient Boosting, Stochastic Gradient Boosting and Regularized Gradient Boosting. There are various tuning parameters available, among which the maximum tree depth was configured to 20 and booster was set to gbtree after various experiments.

3.3.7 LightGBM Light Gradient Boost Machines (LGBM) is an advanced gradient boosting technique which is based on trees. It differs from other tree based boosting framework by implementing leaf wise growth rather than level wise as in traditional decision trees. It is very much distributed and optimized for higher efficiency and accuracy. There are a variety of tuning parameters available such as boosting method, metric, number of leaves, maximum depth of tree, bagging fraction, feature fraction, etc., Cross validation was done to figure out the tuning parameters, and the number of leaves was set as 80, minimum

data in leaf was set as 4, metric as l2 root, bagging fraction as 0.78, feature fraction as 0.68, etc.,

4 Results and Analysis

Both simple and ensemble machine learning models were trained and tested. The training time, testing time, RMSE and r2-score were captured and compared to identify the best performing model for the current dataset. The linear regression model took 7 seconds to train and predicted the test data within fraction of a second. The r2-score obtained was 0.78 and RMSE was 3726.7. The performance was pretty low and cannot be accepted. After a rigorous cross validation with various n_neighbors parameter, k-nn model produced best r2-score of 0.871 and RMSE of 2892.6 with n_neighbors set to 4. The model took nearly 14 seconds to train but a humongous 674 seconds to complete the prediction. Though the performance of k-nn was better than linear regression, it is still considered less efficient. The decision tree provides more parameters to tune but there is a higher chance of overfitting. So, the hyperparameters were carefully chosen to test our model. An r2-score of 0.857 and RMSE of 3041 was observed which is less compared to k-nn. The decision tree model completed training in 5.6 seconds and testing in less than a second. Out of three simple models chosen, k-nn achieved the best performance. But, this was still less than 90% accurate.

The actual and predicted used car prices by the simple models are illustrated in Fig. 7. From the figure, it is observed that the decision tree predictions were scattered more than KNN. There were also more errors when the car price is very low or very high.

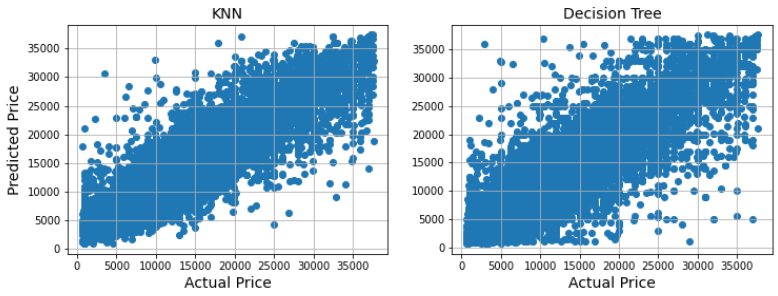

Fig. 7. Performance of KNN and Decision Tree

In ensemble models, the random forest provided r2-score of 0.907 and RMSE of 2449.8 which is better than all of the simple models seen so far. Although random forest took a huge 318 second period to train with the dataset because of generation of multiple decision trees, it took just a second to predict the test set. The next ensemble model bagging regressor which adopts bagging technique took 37 seconds to train and 0.76 seconds to predict the test data. The model gave an r2-score of 0.905 and RMSE of 2477. The model's performance was slightly lesser than random forest.

The performance of random forest and bagging regressor is illustrated in Fig. 8. From the figure, it is clear that the random forest and bagging models' performances are better than simple models. But there are errors throughout the price range and margin of error is still high.

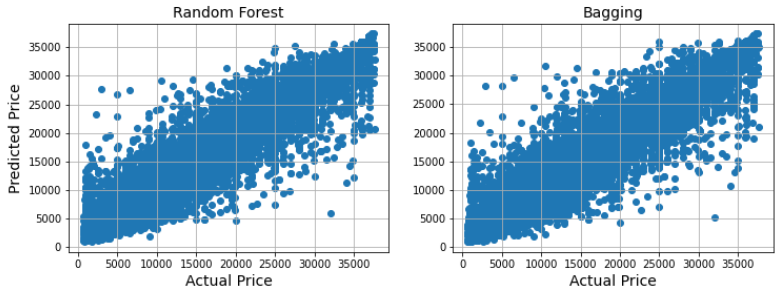

Fig. 8. Performance of Random Forest and Bagging

XGBoost is an award winning machine learning technique which adopts boosting methodology. It produced an r2-score of 0.912 and RMSE of 2389. The training time took a huge 534 seconds and testing time was 0.009 seconds. The next gradient boosting model LGBM took 80 seconds to train and 14 seconds to predict the test dataset. The model produced an r2-score of 0.926 and RMSE of 2193.

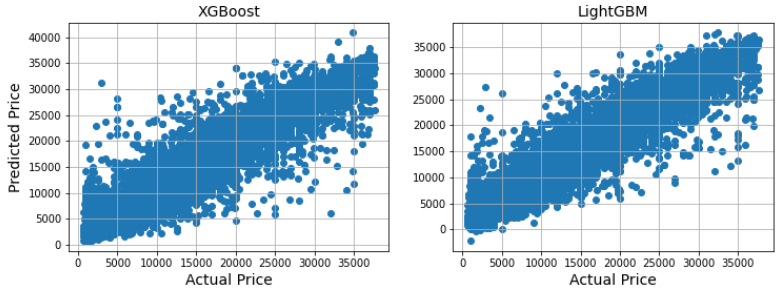

Fig. 9. Performance of XGBoost and LightGBM

Fig. 9 illustrates performance of XGBoost and LGBM models. The margin of error between the actual and predicted is better in LGBM than the previous model and LGBM performs better than XGBoost. In XGBoost, there were more errors in the lower and higher price ranges when compared to LGBM. The summary of the various models' performance is described in the Table 1.

Table 1. Performance of Simple and Ensemble Models

Model Name	r2-score	RMSE	Train Time(s)	Test Time(s)
Linear Reg	0.786336	3726.7	6.672550	0.045270
KNN	0.871280	2892.561805	13.457407	673.752
Decision Tree	0.857677	3041.564852	5.565440	0.073758
Random Forest	0.907666	2449.856974	308.150335	1.216029
Bagging	0.905569	2477.515453	37.237024	0.763299
XGBoost	0.912183	2389.174617	533.795024	0.009059
LightGBM	0.926037	2192.640160	80.046942	13.958435

5 Conclusion

The used car price prediction with craigslist dataset was quite challenging as there are several attributes that impact the price. Also, the dataset had a lot of features which needed to be analyzed carefully and preprocessed before being exposed to the models. Three simple and four ensemble models were taken for prediction. Several iterations of training were performed to tune the hyperparameters of the models and then the models' performances were evaluated. It is inferred that the ensemble models perform better than simple models by producing better accuracy and being less prone to overfitting. Overall, LGBM produced the best r2-score of 0.926 and RMSE of 2193. XGBoost was the next best model whose performance was closer to LGBM. Although it had the highest training time among all the models, the testing time was the lowest of all at 0.009 seconds.

6 Future Work

The machine learning world is continuously evolving with machines learning fine-grain information from the dataset. The convolution neural networks (CNN) could be explored further with our model as the baseline. A CNN could be built to use the images of the cars to infer useful information about the condition of the car, manufacturer, make, model etc., and combine that information with the ensemble models to improve prediction accuracy and performance. Also, a recommendation engine can be built for used cars by finding out cars which are similar to the user's interest.

References

1. Pudaruth, Sameerchand, " 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.

2. N. Monburinon, P. Chertchom, T. Kaewkiriya, S. Rungpheung, S. Buya and P. Boonpou, "Prediction of prices for used car by using regression models," 2018 5th International Conference on Business and Industrial Research (ICBIR), Bangkok, 2018, pp. 115-119, https://doi.org/10.1109/ICBIR.2018.8391177
3. Enis Gegic et al. "Car Price Prediction Using Machine Learning Techniques." TEM Journal 8.1 (2019): 113–118. Web.
4. Pal, Nabarun et al. "How Much Is My Car Worth? A Methodology for Predicting Used Cars Prices Using Random Forest", Future of Information and Communications Conference (FICC) 2018.
5. Data cleaning, https://www.kaggle.com/austinreese/craigslist-carstrucks-data/kernels.
6. Craigslist Dataset from Kaggle, https://www.kaggle.com/austinreese/craigslist-carstrucks-data.
7. Scikit-learn, https://scikit-learn.org.
8. LightGBM Python Package, https://lightgbm.readthedocs.io/en/latest/Python-Intro.html.