Used Car Price Prediction - Comparing 8 Models

Pranav Patil <pranav.patil@rutgers.edu>

22:544:650 Data Mining - Spring 2021 - Professor Hui Xiong

Abstract

Determining the price of a used car is a challenging task, as many factors play an important role in deciding a car's price. The goal of this project is to develop 8 different machine learning models to accurately predict the price of a used car and compare the results of the models. The features of a car are the primary basis of price-prediction. Various learning methods are implemented and evaluated to find which factors contribute the most towards determining the price of a used car and which model gives importance to which features. The results show that XGBoost is the best model for predicting the price of a used car.

Introduction and Background

Predicting the price of a used car is important as well as interesting. The sales of used-cars have increased significantly in the last decade, whereas the sales of new cars have registered a decrease during the same period. More than 500,000 listings were made in 2020 within the United States on Craigslist, which is the world's largest collection of used vehicles for sale. Predicting the true value of a used car is not as simple as it may seem. The value of a used car depends on a number of factors- the age of the car (when it was purchased), its make (and model), the origin (whether it is imported or local), its odometer reading (the number of miles it has been driven) and its horsepower. But, in recent times, a large number of factors in addition to the ones that are aforementioned play an important role in determining the price of a used car. For example, since fuel prices are constantly on the rise, fuel economy also plays a critical role. Other factors such as the type of fuel, the number of cylinders, the size of the car, the paint color, the condition of the car, and the transmission type (automatic or manual) are often overlooked while determining a used vehicle's price. But these factors are as important as the other factors, and should be considered while predicting the price of a used car.

Methodology and Implementation

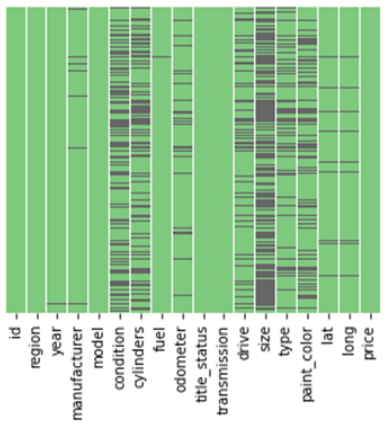
In this section, I will discuss the methodology and the implementation of different steps from the initial to the final stage. The several stages in this project are as follows: -

- Stage 1 - **Data Collection**: -
  The dataset that I have used is Used Cars Dataset (Vehicles listings from Craigslist.org). This dataset contains all the information that Craigslist provides on car sales within the United States in 2020. The dataset used in this project is available on Kaggle [1]. The dataset has 458213 rows and 26 columns (variables).

- Stage 2 - **Data Cleaning**: -

Data Cleaning consists of the following sub-steps: Removing irrelevant features, identifying null values, filling null values, and removing outliers.
The initial step was to remove the features (variables) that did not have any significance towards analysis and price prediction. I removed features such as 'URL', 'VIN', 'description', etc. from the dataset. The next step was to identify and deal with the null values. Most of the features had a large number of null values. To fill the null values, I used the ExtraTreeRegressor as it had the lowest MSE (Mean Squared Error) value as compared to other imputer methods.

```
id                0
region            0
year           1050
manufacturer  18220
model          4846
condition    192940
cylinders    171140
fuel           3237
odometer      55303
title_status   2577
transmission   2442
drive        134188
size         321348
type         112738
paint_color  140843
lat            7448
long           7448
price             0
dtype: int64
```



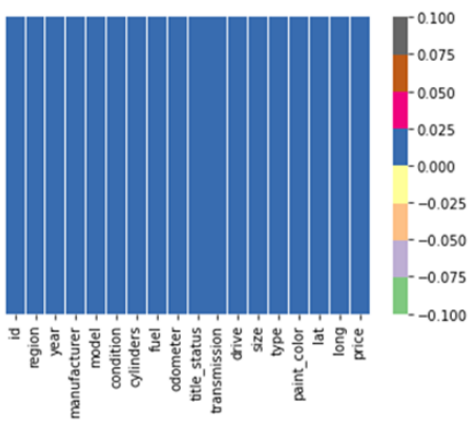


Fig 1 (a). Null Values Fig 1 (b). Original dataset Fig 1 (c). After applying ExtraTreeRegressor

After filling the null values, I checked if there was any correlation between the variables. This dataset had real-life data that had an extremely high range of values and different types of variables, which might be the reason why I found that there was no correlation between the variables in which I was most interested. The last step in data cleaning is removing outliers. I used the Interquartile Range (IQR) method to do so. So, the original dataset had a shape of (458213, 26) and the cleaned dataset had a shape of (395980, 18). Thus, I removed 62233 rows and 8 columns, and the dataset was now ready for preprocessing.

- Stage 3 - **Data Preprocessing and Transformation**: -
  To apply the ML models on the categorical variables, I first needed to transform them into numerical variables. I used the sklearn LabelEncoder [2] library for this purpose. The dataset is also not normally distributed; all features have different ranges. Thus, I transformed the data into a normalized form. For normalization, I used the sklearn MinMaxScaler [3] library. After transformation was complete, I divided the dataset into training and testing sets. 90% of the data was split for the training and 10% of the data was used as test data.

- Stage 4 - **Implementation of Different Models**: -
  I utilized different machine learning algorithms to predict the price of a used car (target variable). The models that I applied are: Linear Regression, Ridge Regression, Lasso Regression, K-Neighbors Regressor, Random Forest Regressor, Bagging Regressor, Adaboost Regressor, and XGBoost (eXtreme Gradient Boosting). Out of these, there were many ensemble learning methods [4] and there were a few instance-based learning methods. For the purpose of implementing the models in Python, I imported the required modules and packages for every model from the Scikit-learn (sklearn) machine learning library. For every model, a general approach was to initially fit the model, implement the model using the imported

packages and modules, calculate the accuracy of the model using the Mean Squared Logarithmic Error (MSLE) and R^2 score, plot the feature importance graph, and visualize the true and predicted value of the price of the car. Although every model had a specific approach, the general approach was as aforementioned. For evaluating the 8 models and comparing their accuracies, I have used the MSLE, Root Mean Squared Logarithmic Error (RMSLE), and the R^2 score.

Results

1. Linear Regression

Linear Regression is one of the simplest models and it consumes relatively less time for training the data. It is a linear approach for modelling the relationship between a dependent variable and one or more independent variables. Linear Regression model gave most importance to the following 5 features of a car: year, cylinder, transmission, fuel, and odometer.

```
MSLE : 0.0027500644158833104
Root MSLE : 0.052441056586259874
R2 Score : 0.628718069669277 or 62.8718%
```

2. Ridge Regression

Ridge Regression is a technique for analyzing multiple regression data that suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. I found the best value of alpha using the yellowbrick AlphaSelection library. Using this alpha value, I implemented RidgeRegressor from sklearn library. Ridge Regression model gave most importance to the following 5 features of a car: year, cylinder, transmission, fuel, and odometer. These results were quite similar to that of Linear Regression.

```
MSLE : 0.0027500613589579166
Root MSLE : 0.052441027439953124
R2 Score : 0.6287184584417796 or 62.8718%
```

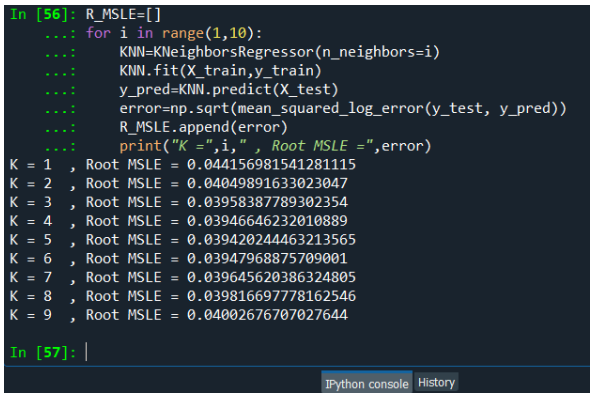
3. Lasso Regression

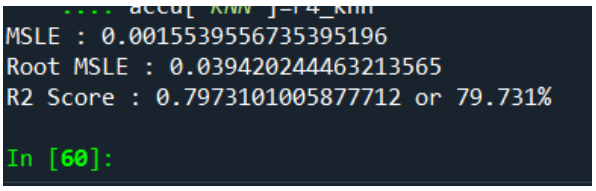
Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point as mean. The lasso procedure encourages simple, sparse models (i.e., models with fewer parameters). For this dataset, the results of Lasso Regression were not better than any other models.

```
MSLE : 0.0027500019269870385
Root MSLE : 0.05244046078160487
R2 Score : 0.6287295858178608 or 62.873%
```

4. K-Neighbors Regressor

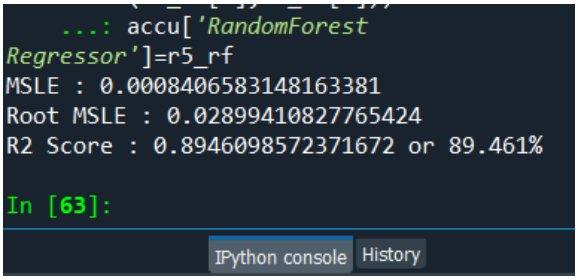
This is a very popular algorithm in the machine learning world where the target is predicted by local interpolation of the targets associated with the nearest neighbors in the training set. k-NN is a type of instance-based learning where the function is only approximated locally and all computation is deferred until function evaluation. The best value of k was found out to be k = 5. So, I trained the dataset using n_neighbors = 5. The results were better than other models applied so far as the accuracy was high with low error: -





5. Random Forest

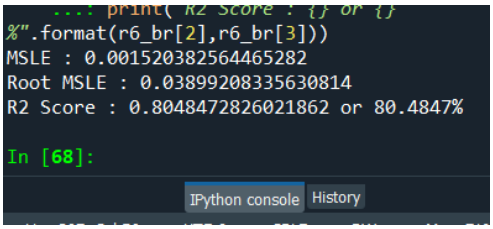
The random forest [5] is a classification algorithm consisting of many decision trees. It uses bagging and feature randomness when building each tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree. In this model, 180 decisions were created with max_features 0.5. Random forest gave the most importance to the 'year' feature followed by the 'odometer' feature. The accuracy of this model was the highest until this point with 89.46%.



6. Bagging Regressor

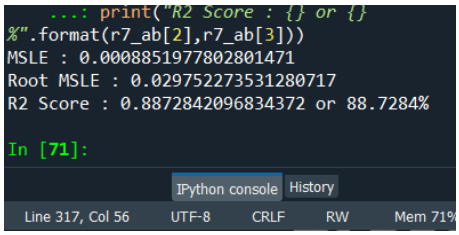
A Bagging regressor [6] is an ensemble meta-estimator that fits base regressors each on random subsets of the original dataset and then aggregates their predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it. I used the

DecisionTreeRegressor as the estimator with max_depth=20. It created 50 decision trees and the results were as follows: -



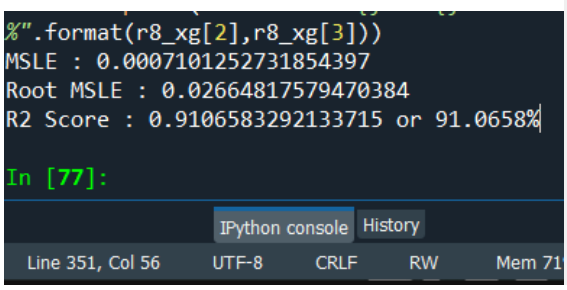
7. AdaBoost Regressor

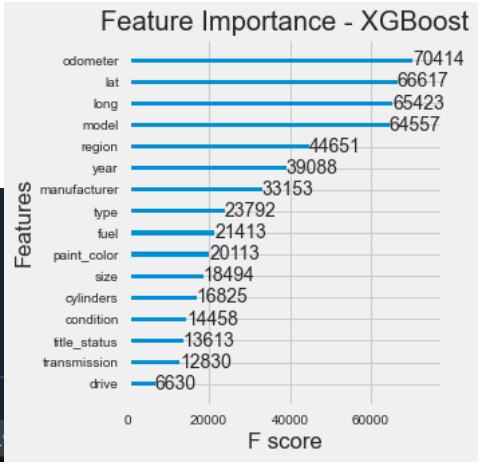
An AdaBoost regressor [7] is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction. As such, subsequent regressors focus more on difficult cases. This model gave the most importance to the following features: year, odometer, model, size, and others in a decreasing order.



8. XGBoost (eXtreme Gradient Boosting)

XGBoost is an ensemble learning method. XGBoost [8] is an implementation of gradient boosted decision trees designed for speed and performance. It is considered as one of the most powerful algorithms because of its high scalability, which drives fast learning through parallel and distributed computing and offers efficient memory usage. In this model, 200 decision trees are created with 24 max depth and with a 0.4 learning rate. XGBoost showed the best performance out of all models. The feature importance graph and accuracy results are as follows: -



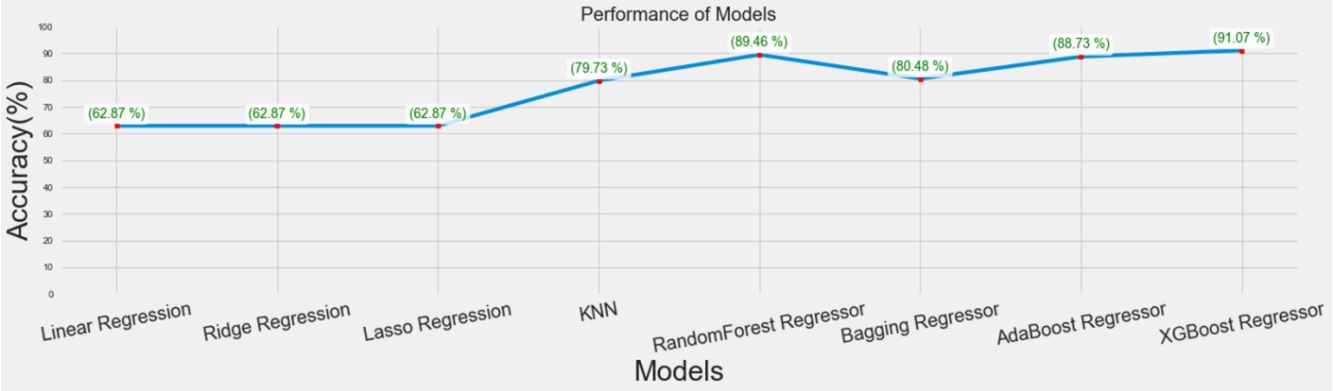


Comparing the Accuracy of All Models

Using the MSLE, and the R^2 score, the results of all models can be quantified as follows: -

	A	B	C	D	E	F	G	H	I	J
1		Linear Reg	Ridge Regr	Lasso Regr	KNN	RandomFc	Bagging Re	AdaBoost	XGBoost Regressor	
2	MSLE	0.00275	0.00275	0.00275	0.001554	0.000841	0.00152	0.000885	0.00071	
3	Root MSLE	0.052441	0.052441	0.05244	0.03942	0.028994	0.038992	0.029752	0.026648	
4	R2 Score	0.628718	0.628718	0.62873	0.79731	0.89461	0.804847	0.887284	0.910658	
5	Accuracy(%)	62.8718	62.8718	62.873	79.731	89.461	80.4847	88.7284	91.0658	
6										

Model Performance: -



Conclusion

We can conclude that the XGBoost Regressor outperformed all other models with an accuracy of 91.07%. As compared to Linear Regression, all other decision-tree based methods performed better in terms of accuracy. Random Forest and AdaBoost Regressor were almost as accurate as XGBoost with a slightly less accuracy.

By utilizing different Machine Learning models, I had aimed to get a better accuracy percentage and a lower error rate. My purpose was to predict the price of used cars from a dataset of 458213 records and 26 variables. I achieved the goal by using different packages and modules from the Scikit-learn Machine Learning library to predict the price of a used car.

References

1. https://www.kaggle.com/austinreese/craigslist-carstrucks-data

2. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

3. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

4. https://en.wikipedia.org/wiki/Ensemble_learning

5. https://en.wikipedia.org/wiki/Random_forest

6. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingRegressor.html

7. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html

8. https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/

9. 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.

10. https://scikit-learn.org/stable/modules/classes.html (Scikit-learn: Machine Learning in Python)