Predicting 2016 Used Car Prices Using Supervised Machine Learning Techniques

Jason Rappazzo

11/28/2022

Contents

Abstract

This paper analyzes used car prices in the United States from 2016. The main goal is to come up with a machine learning model that can accurately predict the price of a used car given various features. Through this process we will be able to see which features of a used car are the most important in determining price. As far as the machine learning models, this paper will utilize supervised learning techniques such as random forest, decision tree, linear multiple regression, lasso regression, and ridge regression models. Each model will have it's respective performance analyzed and compared across other models. The Lasso Regression model performed the based based off of the root mean squared error value. With that being said, none of the models performed extraordinarily well. Their error was either very high or they over fit the training data. With hyper parameter tuning, more data points, and a possible decrease in the number of features, this can be improved upon.

Keywords: Used Car Price Prediction Model; Supervised Machine Learning; Random Forest; Regression.

1 Introduction

The used car market has been very active over the past decade. With the recent microchip shortage, used cars are an option to many people who cannot afford to buy a brand new car. They are also an option for people who do not want to wait for their car to be manufactured which may takes many months to complete.

The objective of this study is to compare various machine learning methods on the price prediction of the used car. With an accurate model, we are able to make the used car market more efficient by increasing the buyers knowledge. Lessening the likelihood that buyers will pay more than they have to and sellers are offering a fair price.

1.1 Related Works

Pudaruth [1] explores the used car market in Mauritius in 2014 by using historical data collected in newspapers. Four models were being analyzed (multiple linear regression, K-Nearest Neighbor, Naive Bayes, and Decision Trees). It was proved that all four types of models produced similar results and that more sophisticated models would need to be used in the future. It was noted that some of the most important features in a car to help predict the price is age, make, model, country, millage, and horsepower. Purdaruth used a very small amount of data comprised of only 97 different used cars. This makes utilizing machine learning techniques difficult. It was found that Naive Bayes was the most useful technique.

Gegic et al. [2] used three machine learning techniques to predict the price of cars in Bosnia. Artificial Neural Networks, Support Vector Machine, and Random Forests were utilized in this study. Data was scraped from a website and contained 797 cars. It was found that their models alone did not produce very high accuracy, but when all three were used together as an ensemble in JavaScript, they were able to achieve 87.38 percent accuracy. It was noted that one of the largest tasks in predicting the price of a car is collecting, cleaning, and standardizing the data.

Pandey et al. [3] Attempted to predict the price of used cars in the Indian automobile market using a fairly normally distributed data set from Kaggle. Random Forest and Extra Tree Regression models were created and achieved high accuracy. They used hyper parameter tuning, RandomizedSearchCV, to achieve highly accurate models in a short period of time. Cross-Validation was used to assess the possibility of over fitting in the models. Finally, the models were deployed in an app using Heroku. Similarly, Yadav et al. in their paper [4] used a dataset from Kaggle and attempted to predict the prices of used cars in India where the used car market is very large. During their pre-processing phase, they ran into outlier problems. To combat this, they cut off values after 3 standard deviations away from the mean. They also dropped all categorical variables and only used numeric variables. Their paper concluded that clustering with linear regression and Random Forest yields the best outcome.

Pal et al. [5] created a Random Forest model that achieved 83.63 percent accuracy on the test data using a Kaggle data set scraped from eBay on over 370,000 different care across 40 different brands. The data set had 20 unique features however, they went through a 9-step process to narrow down the features. This process brought it down to only 9 features. They created visual data analysis on their data. They tried to create a linear regression model however, it suffered from over fitting. The authors concluded that mileage, brand, and vehicleType were their most relevant features.

Zhang et al. [6] was able to achieve 83 percent accuracy on their model predicting the price of a used car. Many models were tested and compared amongst one another. Such models include logistic regression, SVM, Decision Tree, Extra Trees, AdaBoost, Random Forest. Out of these models, Random Forest proved to be the best model. The authors used 5 different features for their models (brand, powerPS, kilometer,sellingTime, VehicleAge). Like many others, they used a Kaggle data set for their data gathering.

Noor et al. [7] created a simple yet powerful model. Their multiple linear regression was able to achieve 98 percent precision. Initially they had 2000 car records that were recorded in around 2 months. After the pre-processing phase, this number got dropped to 1699. The authors used three features to build their final model (Model Year, Model, and Engine Type). Though they used one model for this particular article, the authors stated that other machine learning techniques may work well with the data that they were able to collect.

2 Modeling the Used Car Market

Using a data set from Kaggle comprising of the target variable, price, and the features, the goal of this study is to create a model which can most accurately predict what the price of a used car will be.

$$F : UsedCar\,Features \rightarrow UsedCar\,Price \tag{1}$$

With this, we can take new used car features and be able to predict what the price of the used car should be.

2.1 *Linear Regression*

Linear Regression, also known as Ordinary Least Squares (OLS) is the most simple method for regression analysis. The model can be written as:

$$\hat{y} = \hat{\beta_0} + \hat{\beta_i} \times x + \epsilon \tag{2}$$

Where $\hat{y}$ = predicted price of a used car.

Where β_0 = the intercept term.

Where $\beta_i\hat{}$ =the beta estimates for each feature.

Where ϵ = the error term.

As stated in [8], the model minimizes the mean squared error. The mean squared error is written as:

$$MSE(\beta) = \frac{1}{n}\sum_{i=1}^{n}\hat{\epsilon_i}^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2. \tag{3}$$

Where $\hat{y_i}$ = predicted price of a used car and y_i = actual price of the used car.

2.2 *Ridge Regression*

Ridge Regression is a regularized version of linear regression. This is achieved by adding equation 4 to the loss function.

$$\sum_{i=1}^{n}\beta_i^2 \tag{4}$$

The cost function is as follows:

$$J(\beta) = MSE(\beta) + \frac{1}{n}\alpha\sum_{i=1}^{n}\beta_i^2 \tag{5}$$

2.3 *Lasso Regression*

Like Ridge Regression, Lasso Regression adds a regularization term to the cost function, but used the l1 norm of the weight vector instead of half the square of the l2 norm. The cost function is as follows:

$$J(\beta) = MSE(\beta) + \alpha\sum_{i=1}^{n}|\beta_i| \tag{6}$$

2.4 *Decision Tree*

As explained in [8], Decisions Trees are, to be put simply, hierarchical if-else statements that can perform both regression and classification tasks. In this case, the Decision Tree model will only be used for the regression task of predicting the price of the used car. Figure 1 is an image that the authors used to most simply describe a very basic decision tree as a classification task. The model that I will be creating will work similarly, but each branch will output a predicted used car price as a numeric value. One of the problems with Decision Tree models is they tend to over fit the training data. It will be interesting to see how this model performs over the large data set that it will be trained and tested on.

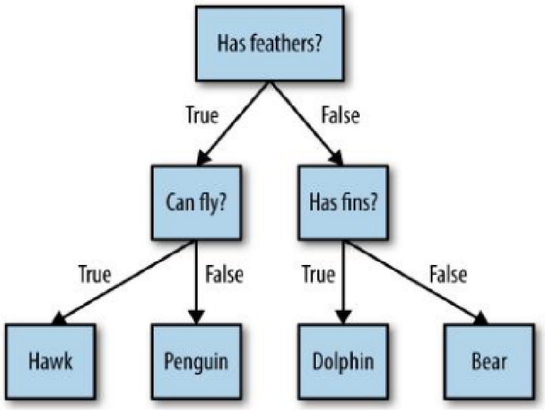

Figure 1: Basic Classification Decision Tree

2.5 *Random Forest Model*

Random Forest Models are one way to combat the over fitting problem that Decision Tree Models may have. They are comprised of many slightly different decision trees. The decision trees that make up the random forest are generated randomly giving it the name Random Forest. Depending on the number of trees in the forest, this could be a highly accurate model, but could also have the tendency to over fit the data.

3 Data

The data that I will be creating my models on is from the website Kaggle which was scraped from Craigslist. This data set contains data on used cars for sale in the United States. It contains over 470,000 observations with 23 features. For this study I will only be working with cars from 2016 and reducing the number of features. After some data wrangling, the final size of the data set that I will be using contains 8,396 observations and 8 columns.

The column names are as follows: 'price', 'year', 'manufacturer', 'condition', 'cylinders', 'fuel', 'odometer', 'transmission', 'type', "paint color", 'state'.

3.1 Features

Year: Year of the car being sold. In this study it will only be 2016

Price: The listing price of the used vehicle in USD.

Mean	**Standard Deviation**	**25th Percentile**	**75th Percentile**
23,018	**13,058**	**13,000**	**32,983**

Table 1: **Price Summary Statistics**

Manufacturer: The manufacturers name of the car. There are 35 different car manufacturers in this data set. The top 4 manufacturers are as follows:

Ford	**Chevrolet**	**Toyota**	**GMC**
1,647	**1,497**	**637**	**584**

Table 2: **Most Frequent Manufacturers**

Condition: The condition of the car being sold

Condition	**Frequency**
New	**28**
Like New	**1,026**
Excellent	**3,578**
Good	**3,741**
Fair	**14**
Salvage	**9**

Table 3: **Frequency of Each Car Condition**

Cylinders: The number of cylinders the car has.

Cylinders	Frequency
3 Cylinders	**5**
4 Cylinders	**2,644**
5 Cylinders	**28**
6 Cylinders	**3,456**
8 Cylinders	**2,184**
10 Cylinders	**65**
12 Cylinders	**3**
Other	**14**

Table 4: **Frequency of Each Car Cylinder**

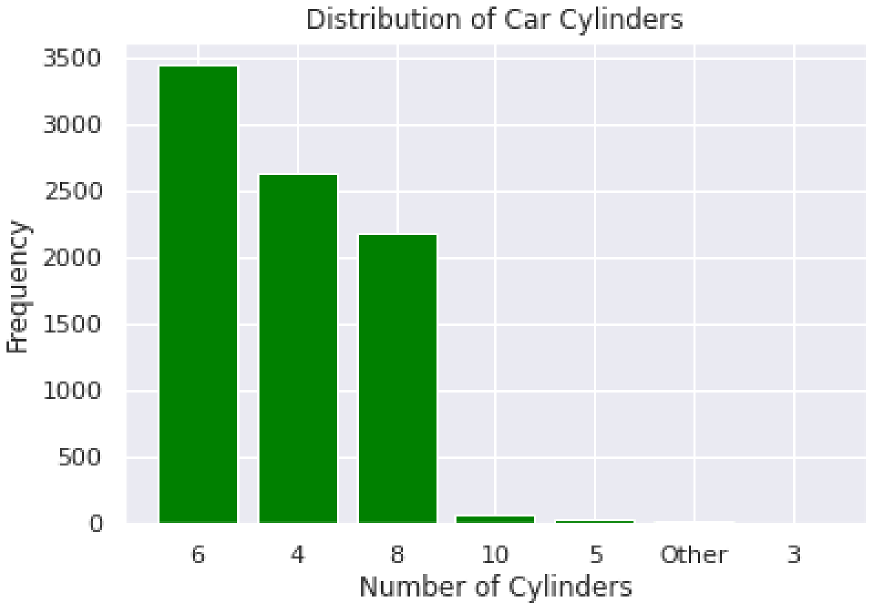

Figure 2: Cylinder Graph

Fuel: The type of fuel the car uses.

Fuel Type	Frequency
Gas	**7,715**
Diesel	**430**
Other	**168**
Hybrid	**71**
Electric	**12**

Table 5: **Frequency of Each Fuel Type**

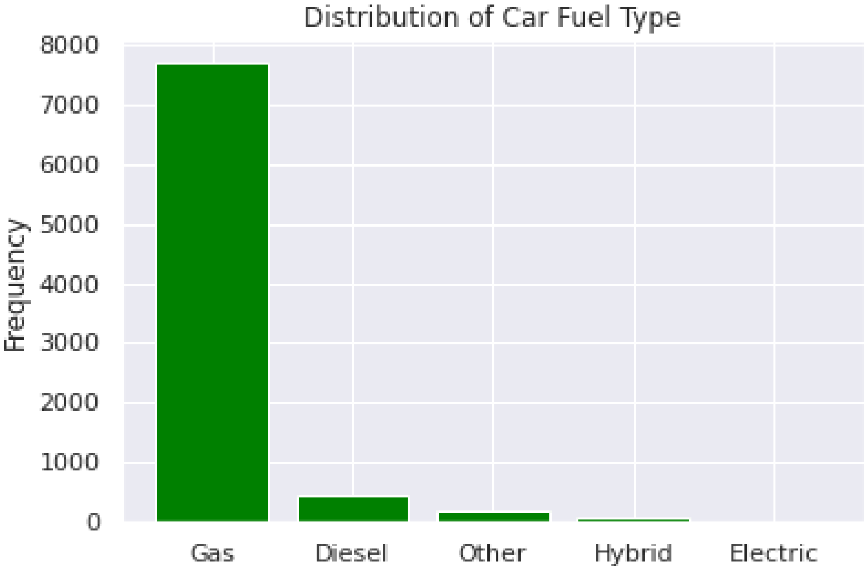

Figure 3: Fuel Graph

Odometer: The number of miles driven on the car.

Mean	Standard Deviation	25th Percentile	75th Percentile
64,355	**37,366**	**33,000**	**93,000**

Table 6: **Odometer Summary Statistics**

Transmission: Car Transmission type.

Automatic	Other	Manual
5,881	**2,296**	**219**

Table 7: **Car Transmission Frequency**

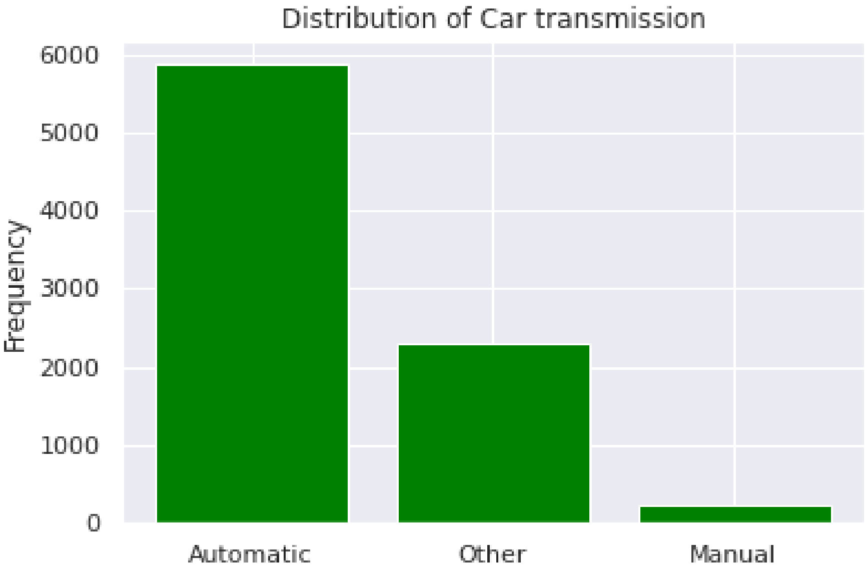

Figure 4: Transmission Graph

Type: What kind of car it is for example: Sedan, SUV, Truck, etc.

Paint Color: The color of the car.

State: What state the car is from.

3.2 Preparing For Machine Learning

To prepare the data for machine learning techniques, I first started by looking for outliers in the numeric data that were present. These two included price and odometer. After plotting these two separately, it was apparent that there were upper bound outliers for each. I followed a 90 percent winsorization to take care of these outliers. The 90 percent winsorization cut off the top 5 percent of the data to remove the outliers. Next, I split up the data into testing and training sets. I used a 95 percent test train split. I used StandardScaler from the sklearn package to standardize the scales for the numeric data. Next, I used OneHotEncoding for all of the categorical variables. After all of these steps, I was ready to build my machine learning models.

4 Results

To compare the four models performance against one another, we will be looking at the root mean squared error (RMSE). To find the best performing model, we want the smallest RMSE that does not over fit our training data. A 10-fold cross validation technique was performed for each model to ensure the model is not over fitting. The **linear regression** model had a RMSE around 7,684 and after performing a 10-fold cross validation, concluded that the model did not over fit the training data. The average of the cross validation scores was equal to 7,807 with a standard deviation of 332. The **ridge regression** with an alpha value equal to 10 performed very similarly as the linear regression with a RMSE equal to 7,701 with an average cross validation score of 7,805 with a standard deviation of 337. The **lasso regression** model also performed very similarly. With an alpha value equal to 0.001, the RMSE was 7,683 and the average cross validation score was 7,807 with a standard deviation of 332. All three linear regression models did not over fit the training data, however they had large RMSE which is not ideal. The table below shows the RMSE on top and the average 10-fold cross validation RMSE below.

Linear Regression	**Ridge Regression**	**Lasso Regression**
7,684	**7,701**	**7,683**
7,807	**7,805**	**7,807**

Table 8: **RMSE and Average 10-Fold Cross Validation RMSE**

The **decision tree** had a much lower RMSE than the linear regression models with a value of 1,129, but the average cross validation score was 6,992 meaning that the decision tree model over fit the training data. Similarly, the **random forest** model had a RMSE of 2,221 but an average cross validation score of 5,658 meaning it also over fit the training data.

Decision Tree	**Random Forest**
1,129	**2,221**
6,992	**5,658**

Table 9: **RMSE and Average 10-Fold Cross Validation RMSE**

Based off of the minimum root mean squared error, the lasso regression model performed the best. This model is able to predict the price of a 2016 used car give or take around 7,683 dollars.

5 Conclusion

The way the models have been set up, the Lasso Regression model has performed the based based off of the root mean squared error value. With that being said, none of the models performed very well. Their error was either very high or they over fit the training data. To combat this, hyper parameter tuning is very necessary. In further studies, this can be worked on. Another way to create better models is to have more data points. It is difficult to create a well performing machine learning models without thousands of data points. This may be why both the decision tree and random forest models both over fit the training data.

References

[1] Sameerchand Pudaruth. Predicting the price of used cars using machine learning techniques. *Int. J. Inf. Comput. Technol*, 4(7):753–764, 2014.

[2] Enis Gegic, Becir Isakovic, Dino Keco, Zerina Masetic, and Jasmin Kevric. Car price prediction using machine learning techniques. *TEM Journal*, 8(1):113, 2019.

[3] Abhishek Pandey, Vanshika Rastogi, and Sanika Singh. Car's selling price prediction using random forest machine learning algorithm. In *5th International Conference on Next Generation Computing Technologies (NGCT-2019)*, 2020.

[4] Anu Yadav, Ela Kumar, and Piyush Kumar Yadav. Object detection and used car price predicting analysis system (ucpas) using machine learning technique. *Linguistics and Culture Review*, 5(S2):1131–1147, 2021.

[5] Nabarun Pal, Priya Arora, Puneet Kohli, Dhanasekar Sundararaman, and Sai Sumanth Palakurthy. How much is my car worth? a methodology for predicting used cars' prices using random forest. In *Future of Information and Communication Conference*, pages 413–422. Springer, 2018.

[6] Xinyuan Zhang, Zhiye Zhang, and Changtong Qiu. Model of predicting the price range of used car. 2017.

[7] Kanwal Noor and Sadaqat Jan. Vehicle price prediction system using machine learning techniques. *International Journal of Computer Applications*, 167(9):27–31, 2017.

[8] Andreas C Müller and Sarah Guido. *Introduction to machine learning with Python: a guide for data scientists.* " O'Reilly Media, Inc.", 2016.