PS1: Used Car Price Prediction

Yi Lin

February 7, 2021

1. Introduction

What drives the prices of used cars? From the buyer's perspective, to determine if a used car has a reasonable posted price or not would be hard when you look through the listings online. Several factors might impact the actual worth of a vehicle, including but not limited to year, manufacturer, condition, odometer, etc. From the view of a seller, it might be challenging to set the price of a used car properly. In this dataset, the purpose is to adopt several machine learning algorithms, including Linear Regression, Ridge Regression, LASSO, and Elastic Net to build up models and find an optimal predictive model for predicting used car prices.

2. Data Cleaning

2.1 Drop useless column

Notice that the feature like 'paint_color' is not typical related to used car price. So, we drop the entire feature of 'paint_color' from the dataset.

2.2 Pruning Outliers

1. Price. Outliers may cause much difference in both standard deviation values and mean, which directly/ indirectly impact the model accuracy. To remove such outliers of price, I pruned the dataset using IQR, leaving 10% values of both sides of the distribution.

2. Year. A count plot of the dataset crossing categorical variable 'Year' showed that used cars varied by different years. Vehicles with too many years were not good indicators of a 'used' car for predicting price. I pruned the dataset by removing the vehicles with years below the year of 1980.

3. Odometer. Based on the OneDrive, many operators would consider that car would be overhaul start at around 750,000 miles. Therefore, I dropped all mileage above 750,000 miles. Besides, a scatterplot of odometer and price indicated that outliers, so I dropped those outliers with max and min values. Therefore, I removed outliers from three numerical columns.

2.3 Pruning Missing Values

Notice that there is only missing in column 'odometer'. So, I pruned the dataset by dropping these null from 'odometer'. There is not missing values from the dataset.

3. Data Preprocessing

In the dataset, several categorical features cannot be applied to machine learning algorithms. And such categorical data should be converted to numbers.

3.1 Manually map the categorical string to integers

Categorical variables such as 'condition' and 'cylinder' have the ordinal number of mapping those categorical string into integer intuitionally.

3.2 One-Hot Vectorization

Unlike 'condition' and 'cylinder,' I converted these categorical data (type, manufacturer, condition, fuel, and F4) into One-Hot vectors. After converted these categorical variables, I broke the one-hot encoded vector as separated features. For instance, 'type' with [sedan, pickup, SUV, truck], and one-hot vector is [1,0,0,0]. I break the 'type' feature into 4 additional columns - [sedan], [pickup], [SUV], [truck]. And so, only [sedan] column would be 1, and the other three columns would be 0. Finally, I created a new data frame called 'df_ready_to_train,' which combines numerical variables (price, year, odometer, F1, F2, F3) and one-hot encoded vectors as separated features. Note that I also need to reset the data frame index as well.

4. Data Explorartory Analysis

To investigate and analyze all features' general trends concerning the target feature (price), I adopt several visualization tools such as scatterplot, catplot, and barplot. And I notice that the price decreases as the age of car increases. The price decreases while the odometer of the car increases. The price increases when the condition of the car looks like new or excellent. Generally, the price of Subaru is slightly higher than that of Ford. Regardless of the manufacturer, truck and pickup prices are higher than that of SUV and sedan. Holding all other factors, SUV and sedan with 4 cylinders have higher prices than those with 6 or 8 cylinders; pickup and truck with 6 cylinders have higher than those with 4 or 8 cylinders. Compare to the other three types of cars, sedan has a relatively lower price in different cylinders.

In addition, to analyze the impact of the statical significance of F1-F4 attributes, I apply OLS to see the P-value of each attribute. Assume alpha is 0.05. Notice that only a = 0.325, b = 0.647, c = 0.599 which are larger than 0.05, implies a, b, c are statistically insignificant. And the rest of the attributes like F1, F2, F3 are statistically significant. Also, by using VIF score, I found that VIF scores of F1, F2, F3 are less than 5, but VIF scores of a, b, c are more than 5. This indicates that F1, F2, F3 have less multicollinearity, but a, b, c have high multicollinearity. Besides, when I use visualization tools like the Plot Correlation matrix, I notice that F3 doesn't have a high correlation with the price (core = 0.0415).

When I apply the Plot Correlation matrix, I found that the features like year, condition, odometer, F2, sedan, truck have relatively high positive and negative correlation with price. Besides the correlation with price, I interestly found that the correlation between F2 and year have a strong positive relation (corr = 0.79). Since 'year' has somewhat powerful relationship with price, and 'F2' has strong relationship with 'year', it might implies that special modification F2 has positive and powerful impact related to year and price.

In addition, when I apply Backward Elimination for Feature Selection, I notice that the p-value of F2 = 0.741 is greater than 0.05, which also implied that F2 is statistically insignificant. Therefore, I may remove F2 modifications before selling a car.

5. Model Selection & Evalution

I utilized several simple methods with 60-20-20 on testing, training, and validation. I first split the training & testing set into 80-20 and then split the training & validation set into 60-20. The models, including Linear Regression, Ridge Regression, Lasso, Elastic Net, were imported from sklearn. Since the training set is 60% of the dataset, I put this set into the fitting and put both testing and validation sets to calculate the predictive y values. And then use predictive y values to calculate both RMSE and R^2. I use validation set to determine the best model, and then use testing set to estimate the model accuracy.

Table 1: Model Results

ML Algorithms	Testing Data RMSE	Validation Data RMSE	Testing Data R^2	Validation Data R^2
Linear Regression	4286.16	4384.25	0.69106	0.66192
Ridge	4286.02	4383.83	0.69108	0.66198
LASSO	4309.56	4384.93	0.68767	0.66181
Elastic Net	4815.01	4818.17	0.61012	0.59169

RMSE is the measure of how spread out the residuals are, which means that how concentrated the data is around the line of best fit. So the smaller values of RMSE, the better fit. Notice that the RMSE of the Ridge Regression model is the smallest one which is 4383.83. Therefore, I choose Ridge Regression Model as the model to test its performance. By apply the testing set, I calculate that the RMSE and R^2 of the Ridge Regression Model are 4286.02 and 69.11%, respectively. Here, R^2 of the linear regression model is 69.11%, which indicates that there was 69.01% of data fit the regression model.

6. Conclusion

In this project, I conducted a study on regression model performance. To determine what drives the prices of used cars. The dataset came from DSCI552 class and the data preparation was processed by applying python. The final dataset contains 7394 rows and 19 attributes. I tested data by applying into 4 models, including Linear regression, Ridge, LASSO, and Elastic Net on the cleaned-dataset. And each model was analyzed by applying the same testing data. To compare and contrast the model, I used Root Mean Square Error (RMSE) as the criterion. The ridge regression provided the highest performance with RMSE = 4286.02 and followed by linear regression with 4286.16 errors, Lasso with 4309.56 errors, and Elastic Net with 4815.00 errors. Therefore, I would recommend ridge regression to build up the price evaluation model. And the R^2 of ridge regression is 69.01%, which shows that its performance has 69.01% of data fit the regression model. The future work could be applied to Neural Network to improve the prediction performance.

References

[1] Brownlee, J. (2019, August 14). How to one hot encode sequence data in python. Retrieved February 08, 2021, from https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/

[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. doi: 10.1109/ICBIR.2018.8391177

[3] Aarshay JainAarshay graduated from MS in Data Science at Columbia University in 2017 and is currently an ML Engineer at Spotify New York. He works at an intersection or applied research and engineering while designing ML solutions to move product metrics. (2020, October 18).

A complete tutorial on ridge and lasso regression in python. Retrieved February 08, 2021, from https://www.analyticsvidhya.com/blog/2016/01/ridge-lasso-regression-python-complete-tutorial/

[4] Anerisavani. (2020, November 03). EDA and price prediction of used vehicles. Retrieved February 08, 2021, from https://www.kaggle.com/anerisavani/eda-and-price-prediction-of-used-vehicles

[5] Msagmj. (2020, June 01). Data cleaning + eda + used cars prediction(86%). Retrieved February 08, 2021, from https://www.kaggle.com/msagmj/data-cleaning-eda-used-cars-prediction-86#Handling-outliers

[6] Shin, T. (2020, May 06). A machine learning project - predicting used car prices. Retrieved February 08, 2021, from https://towardsdatascience.com/a-machine-learning-project-predicting-used-car-prices-efbc4d2a4998#c0d7