Predicting Used-car Price in UK

James Li

Project Statement:

My friend in UK is considering buying a used-car. This project aims to provide insights on what is a reasonable price range for a particular car. There are many factors that can influence car price: brand, model, mileage traveled, etc. Understanding the relationship between these factors and the listing car price is a major part of this project.

Data:

Found from Kaggle:

https://www.kaggle.com/adityadesai13/used-car-dataset-ford-and-mercedes

According to the owner of the data, they are scraped from online resources.

This page contains one data sheet for 9 car brands. On each sheet, it contains data for the price of the car, as well as the model, registration year, gear box type, mileage traveled, fuel type, road tax, mpg, and engine size. Each sheet is separated into two parts: 75% of them are used for fitting the models and 25% are used for cross-validation.

Exploratory data analysis:

What are the factors?

The factors we are modelling here are: car model, registration year, gear box type, mileage traveled, fuel type, road tax, mpg, and engine size.

What are the brands?

The brands we are looking at are: Audi, BMW, Ford, Hyundai, Mercedes-Benz, Skoda, Toyota, Vauxhall, and Volkswagen.

What are the fuel types?

Petrol, Diesel, Hybrid, Electric, or Other.

What are the types of gear box?

Manual, Automatic, Semi-Auto, or Other.

Any unusualness?

One data point says the car is from 2060. Since we are still in 2021, this data point is thus removed from the data set.

My friend doesn't know much about cars – so do I. We don't know which brand or which model has better quality or design, but we do know that the shorter miles the car has traveled, the newer the car is.

Data Analysis

Start from fitting a simple MLR with the formula:

price~b_0+$b_1$mileage+$b_2$tax+$b_3$mpg+b_{m1}modelA4+b_{m2}modelA5+…+b_{y1}year2019+b_{y2}year2018+…+b_{t1}transmissionManual+b_{t2}transmissionAuto+…b_{b1}brandAudi+b_{b2}brandBmw+…

Note that we are treating year, model, transmission (gear box type), and brand as dummy variables.

Examining the diagnostics as well as the summary of the fit, a few issues emerged:

1. Non-Gaussian: there is a pattern in the scale-location plot: the variance of the residuals may not be constant; the qq-plot also has a heavy tail, raising even more concerns about whether Gaussian assumption can applied.

2. We have too many regressors. This is mainly due to the fact that there are a large number of car models.

Issue 1: **Non-Gaussian:** Non-gaussian has many limitations. Because the main purpose of this study is to construct prediction intervals (price range) for a particular car, the Gaussian assumption is necessary to construct such intervals. To combat non-Gaussianity, log transformation is applied (Many thanks to Dr. Kowal who provided this suggestion in my proposal!).

Note: when doing leverage analysis in R, R gives the warning that some points have leverage 1. This is because these data points are the only data of in its group. Those points themselves decide what the fitted value is, and thus the leverage of those points is 1. As suggested by R, *these points are not shown in the leverage plot.*

After log transformation, both the qq-plot and the scale-location plot are more reasonable for applying Gaussian assumption. (The square root transformation was also applied, but its qq plot has relatively heavier tails and the scale-location plot has a more clear pattern compare to log-transformation).

Issue 2: Too Many Regressors. There are 194 car models across all brands. Using the current model, each brand will have its own parameter, resulting in a large number of regressors in total. Such model is complex and is vulnerable to overfitting and lack of predictability. Because the main purpose of this project is about prediction, attempts are made to reduce the model.

Two main tools used are the partial F test and lasso regression. Two reduced models are first used to test if they are significantly different from the full model. An aggressive approach – completely ignore all model information- is firstly tried but results in a large increase in RSS (p-value $< 2.2*10^{-16}$). A less aggressive approach is to remove all brand information, and after running the partial F test, I found that this reduced model is no different from the full model. This is because each brand has its own set of model names, and there is no intersection of model names between two brands. Because model names are already incorporated in the model, adding brand names are no longer necessary since they depend on the model names. The previous full model is consequently fitted again without using the brand names.

A less aggressive approach is to remove some model categories. The difficulty here is about variable selection. My friend and I are both unfamiliar with cars, and we are not sure which models are similar and which are alike. Thus, lasso regression is introduced to help with variable selection.

Lasso Regression: Using the *glmnet* package, lasso regression is applied to reduce the number of regressors. Surprisingly, lasso regression removes all regressors related to car models, and more regressors such as some gear box types are also removed by lasso. While our model gets significantly less complex, it's worth noting that the RSS is 7 times the full model. The robustness of these models will be tested in the cross-validation section.

Encoding model as a factor: In the last two approach we are operating under the dummy variable approach: each car model has its own parameter. Another way to deal with too many regressors is to encode model as a factor.

Again, the diagnostic plots are examined. No clear pattern is found in the residual plot nor the scale-location plot, and the qq-plot has a very light tail. Using the partial F test, we found that the RSS is no better than the aggressively reduced model (completely ignoring all model information). Moreover, encoding model as a factor needs additional justification. This encoding assumes equal distancing between each car model, and the order of which models are encoded as numbers are important. Again, due to a lack of knowledge about car models, these justifications can't be performed. Thus, this model is discarded.

The models applied so far assume linearity. A potential way to increase the flexibility is to fit splines. Most of the variables in our data are discrete: model, year, transmission, fuel type, and these variables can not be used to fit splines. Tax, mpg (miles per gallon), and mileage are all continuous variables and are thus candidate for fitting splines. Thus, the following model is proposed:

$\log(\text{price}) \sim b_0 + f_1(\text{mileage}) + f_2(\text{tax}) + f_3(\text{mpg}) + b_{m1}\text{modelA4} + b_{m2}\text{modelA5} + \ldots + b_{y1}\text{year2019} + b_{y2}\text{year2018} + \ldots + b_{t1}\text{transmissionManual} + b_{t2}\text{transmissionAuto} + \ldots$

By plotting the three smooth functions, one can see that only f_3 is significantly nonlinear. Thus, there is no need to add smooth terms to mileage and tax, and the model is simplified into:

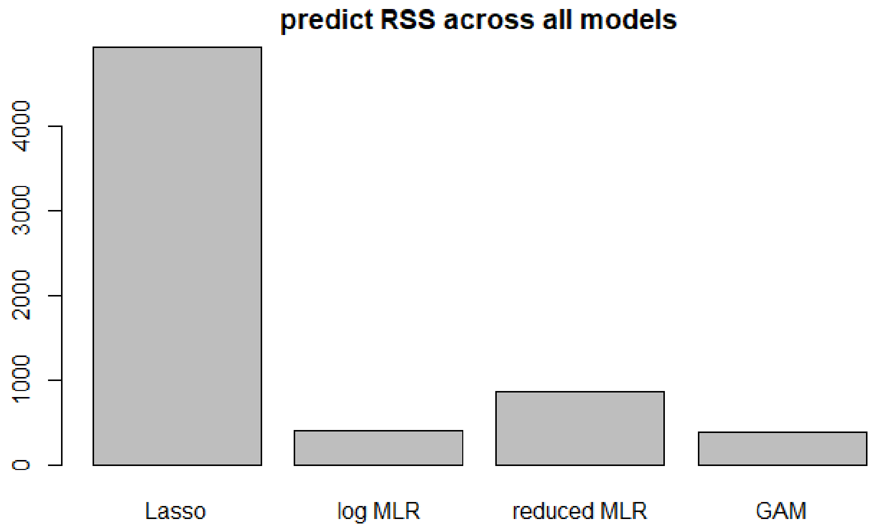
$\log(\text{price}) \sim b_0 + b_1\text{mileage} + b_2\text{tax} + f_3(\text{mpg}) + b_{m1}\text{modelA4} + b_{m2}\text{modelA5} + \ldots + b_{y1}\text{year2019} + b_{y2}\text{year2018} + \ldots + b_{t1}\text{transmissionManual} + b_{t2}\text{transmissionAuto} + \ldots$

The RSS decreases slightly due to more flexibility.

Discussion

Cross-Validation

The raw data is separated into two sets: training set (75%) and test set (25%). All the models discussed above are fitted using the training set and the test set are completely new data for all these models. Point estimates are predicted using each model and RSS is calculated to compare performance.



The simplest MLR model (with log transformation) and the GAM model both give good result in terms of prediction RSS.

The reduced model, with 45 regressors, has higher RSS compare to the last two models, but the lasso model, with only 5 regressors, has an extremely large RSS. This is as expected since the model also has very large RSS on the training data. However, one advantage of using lasso model is its ability to make predictions for data that has never been seen by the model. Consider the example below:

Training data:

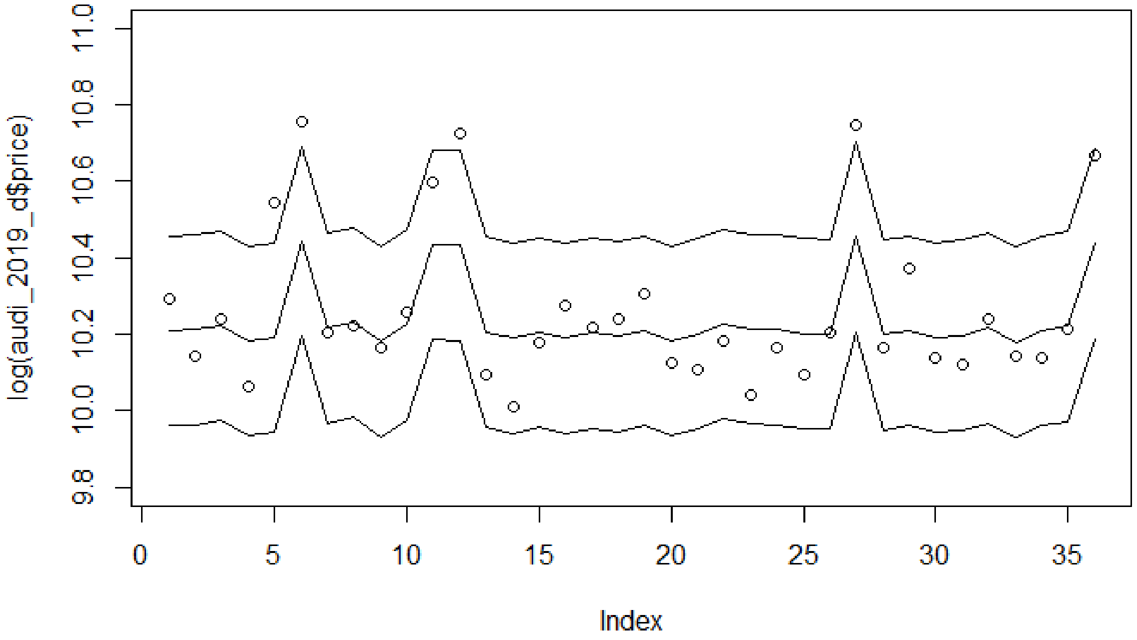
model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize	brand
Galaxy	2016	16170	Manual	45973	Diesel	125	56.5	2	ford
A4	2019	32000	Semi-Auto	3076	Diesel	145	44.1	2	audi
Up	2017	6950	Manual	29000	Petrol	145	68.9	1	vw
EcoSport	2017	10800	Manual	17140	Petrol	125	52.3	1	ford
B-MAX	2016	9498	Manual	14853	Diesel	0	74.4	1.5	ford

Now if we want to predict the price for an Audi A4 made in **2020**, all models except the lasso model is able to generate a point estimate. This is because the lasso model in this project selects year2019 as the only dummy variable among all the years. If year2019 is encoded 1, then all other years can be treated as 0.

Prediction intervals are also generated to compare results. Because the gam model and the lasso model are not suited for such intervals, their results are not shown.

Both models did a surprisingly great job in capturing the real price within the prediction interval.

However, the price range of these two models appears to be too large



Use the Audi A4 diesel 2019 as an example, one can see that the predicted price range is very large and doesn't offer very useful information. To provide some comparison, the predicted price range on Kelly Bluebook is usually smaller than $10,000 (around £7,000).

After considering all these tradeoffs, I personally think the simplest log model stands out. Although it has many regressors, all of them are still linear and thus can be computed easily using matrix operations. It also gives better point estimates (smaller RSS) and narrower price range, meaning it can give more reliable and insightful information about a particular used-car in UK.

There a few issues left unresolved:

1. Can we construct intervals for gam models using a similar approach as we did in the bootstrap? The gam model appears to have smaller RSS and I'm interested to see if its prediction intervals are better (or narrower).

2. Can we add interaction terms in the model? The size of the data is a bit too large to be analyzed on my laptop after adding an interaction term, as I constantly ran into memory limit. Computers with larger RAMs(such as the rice CLEAR machines) might be used to resolve this issue.

Code and Data

Please see the attached rmd file. You can also access it using the github link

https://github.com/LiYueqian-James/Predicting-Used-Car-Price-in-UK

Reference:

Data:

https://www.kaggle.com/adityadesai13/used-car-dataset-ford-and-mercedes

Resources:

Dr. Kowal’s notes, code, and labs.