MS&E 226: Project Part 3
Inference on Regression Model of Price on Used-Cars

Oseas Ayerdi,
Paul Magon de La Villehuchet

Abstract

The automotive industry has rebounced from the ashes and is now thriving; new-car sales are now back to pre-crisis levels and have steady year-on-year growth-projections. This positive outlook on the new-car market is also visible in used-car markets, where sales are increasing. The existence of online platforms connecting sellers and buyers, like eBay Motors Germany, offers an easy and convenient medium to sell and buy a car. With all this information readily available online, predicting the price of a car becomes possible and can prove valuable to both sides (sellers and buyers).

Introduction

We present the prediction part for our two predictive models; in addition, we also provide inference of our model for our continuous response variable, $price$. Our first model (regression) aims at predicting *price*, (i.e. the asking price for a new advert for a used-car on eBay). The second model (classification) predicts a binary response variable, *boolean* (i.e. whether or not the advert will be available in the eBay platform one month after it is created). In Part 1 of the project, we presented the cleaning steps we would take to arrive to the final dataset which is used to train, validate, and test models to predict *price* and *boolean*. The resulting, clean dataset is $129,835$ observations with 11 columns, these being: *price*, *brand*, *kilometer*, *age*, *powerPS*, *gearbox*, *vehicleType*, *fuelType*, *seller* (the type of seller: private, commercial), *notRepairedDamage* (whether the car has an unrepaired damage), *boolean*. In Part 2, we presented the development of the two models and the steps followed to build the best models possibles. The focus of this part is to go one step further and test whether models are good at prediction and whether we can infer properties from the models.

Prediction

Regression

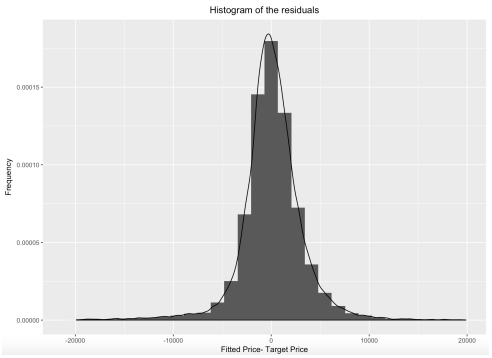
We tested our model on the test data and got an even better $RMSE$!

Department of Management Science and Engineering,
Stanford University, 2017

Set	R^2	$RMSE$
Training	0.7126	4,876
Validation	NA	4,979
Testing	NA	4,897

As we can see, the model is really working well and both validation and training errors provided good estimates of the test error. We believe there are two main reasons explaining such results. First, the model we used was much more simple than other models, hence less prone to over-fitting. Secondly, the amount of data is substantial (around 125,000 observations) and helps build a better model. Finally, we might have been a bit lucky as well.

We represented the distribution of the residuals. This helped us understand the precision of prediction: 85% of predictions are within a $6,000$ € intervals ($\pm 3,000$).



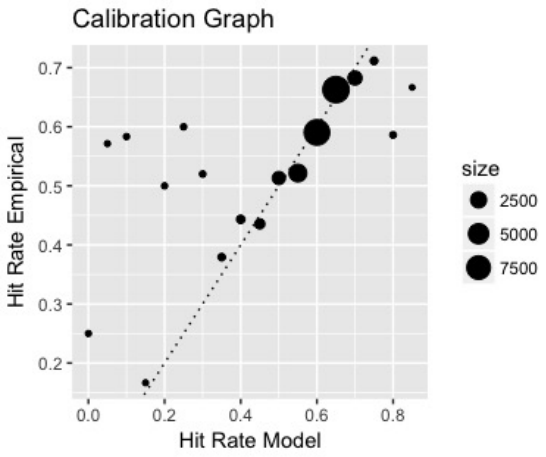
Thus, we are really satisfied with this model!

Classification

There was no superior model for classification in our last report. As such, we will focus on the logistic regression with all the covariates included (i.e. our baseline).

Set	AUC	$Mean - Loss$
Training	0.5177	0.3982
Validation	0.5161	0.3932
Testing	0.5154	0.3897

The resulting calibration graph on the test data is presented below:



The calibration plot shows good results for the instances that have a 40% chance of being being labeled 1 (i.e. that the advert is removed from the platform within a month of being posted). However, as the *AUC* and *Mean-Loss* values show, this is far from perfect, and we expected as much given the results for this model in Part 2. As expected, the model generalizes well but it is not really helpful since it performs barely above our "dumb" baseline, using majority vote.

Inference

Obviously for this part, we are using our regression model!

Coefficients

The model we are using is based on the following covariates: *age*, *kilometer*, *powerPS*, *vehicleType*, *brand*. We also added some interaction terms: *brand:vehicleType*, *age:vehicleType*, *I(age2̂):vehicleType*.
Our dataset actually contains 40 different brands and 8 types of vehicles which makes around 400 different coefficients (due to the interaction terms). To analyze the coefficients, we are going to focus only on "universal" covariates: *Intercept*, *age*, *kilometer*, *powerPS*, *vehicleType*. For *vehicleType*, we are just going to consider *sedans* and *coupes*. For *brand*: *Audi* and *BMW*, as these two are really frequent in the dataset.

Statistical significance
The following table gives the value of the chosen coefficients including the standard deviation and the $p-value$ for the $T-statistic$ on the training set. To assess statistical significance, we are using the $p-value$ method. Given a coefficient and a standard deviation, **R** computes the $p-value$ and this gives us an accurate measure of the false-positive risk we are assuming with the non-null coefficient. Typically, we stay with the usual value for size, $\alpha = 0.05$.

Covariate	$\hat{\beta}$	$\hat{\sigma}_{\hat{\beta}}$	$p-value$
Intercept	22,000	530	0
age	-2,200	43	0
kilometer	-0.0032	0.00055	0
powerPS	57	0.35	0
sedan	-2,500	640	0.00001
coupe	11,000	890	0
Audi	1,174	2,200	0.05
BMW	6,200	1,200	0

From this table, we can see that "universal" covariates are all statistically significant. We are not surprised by the value of the coefficients in general: the sign and the intensity seem to be reasonable and confirm our intuitions about the drivers of a car price. However, some interesting observations can be drawn from the table: first of all the value of the coefficient *kilometer* is really low. The fitted value of the price will only decrease by 30 €for every 10,000 km driven *ceteris paribus*, which is counter-intuitive. One of the reason for that is that the mileage of the car is correlated with the age of the car: $\hat{\rho}_{age,km} = 0.45$ and *age* is included first as a covariate and also in the interaction terms. We can also see from the table that the *Audi* brand is not statistically significant. This problem is recurrent among categorical variables: because we have many brands, some with very few models (there are only 3 Ferrari in the dataset), some brands may not be statistically significant. A more precise approach would be to choose brands one by one.

Coefficients in the testing set
We now fit our model on the testing set and compare both the value of the coefficients and their significance. The following table summarizes the results.

Covariate	$\hat{\beta}$	$\hat{\sigma}_{\hat{\beta}}$	$p-value$
Intercept	24,000	980	0
age	-2,500	90	0
kilometer	-0.0033	0.0010	0
powerPS	58	0.6	0
sedan	-4,800	1100	0
coupe	9,300	1,600	0
Audi	3,700	4,900	0.4
BMW	-600	3,500	0.8

Fitting the data on the testing set gives us two major insights about the dataset: first, "universal" covariates (including *vehicleType*) do not vary much with the dataset. Order of magnitudes are the same and $p-value$ remains 0.
This is not a surprising result. What is more surprising is the difference for brands: *Audi* and *BMW* were statistically significant but they are no longer (with $\alpha = 0.05$). Moreover, the value of the coefficient has deeply changed. We believe that this is due to the repartition between testing and training. Indeed, brand is not enough to predict the price, what matters is brand coupled with *vehicleType*, *age*.

Significance among datasets

Our simple model uses less covariates than the baseline with all covariates. However, we chose the covariates in our new model based on intuition (human bias about what predicts the price of car) and statistical significance. So it is not a surprise to see that the significant covariates are the same on both models.

Bootstrap

Bootstrap estimation

We sample $B = 100$ times with replacement n indices from the training dataset, where n is the size of said dataset. We then use these indices to construct a new dataset. We train the chosen prediction model with this new dataset and save the value of the model's resulting coefficient for the six covariates discussed up to this point. We use this simulation to generate the bootstrap distribution, using histograms.

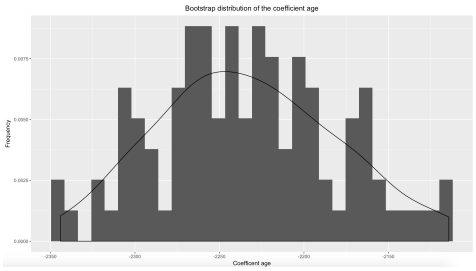
Below is the table comparing the resulting estimations of $\hat{\beta}_{bootstrap}$ with $\hat{\beta}_{regression}$:

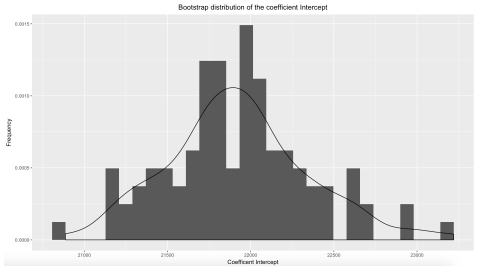
Covariate	$\hat{\beta}_{reg}$	$\hat{\beta}_{boot}$	$\hat{\sigma}_{\hat{\beta}_{reg}}$	$\hat{\sigma}_{\hat{\beta}_{boot}}$
Intercept	22,000	22,000	530	430
age	-2,200	-2,200	43	52
kilometer	-0.0032	-0.0032	0.00055	0.0008
powerPS	57	57	0.35	1.02
sedan	-2,500	-2,500	640	430
coupe	11,000	11,200	890	1,600
Audi	1,200	1,200	2,200	1,200
BMW	6,200	5,900	1,200	1,100

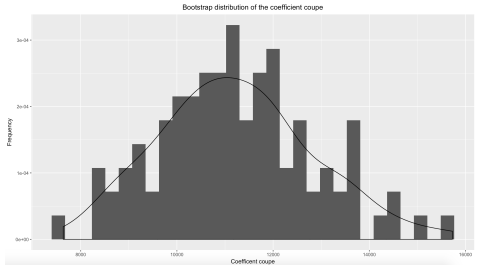
The bootstrap estimation of coefficients and standard deviations is consistent with the estimation given by **R**.

Bootstrap distribution

To compute confidence interval, we have two methods. Using normal method, we are going to have the same intervals as the ones given by **R** since estimations are roughly the same. However, these methods make a strong assumption: that the distribution is linear. One advantage of bootstrap is the possibility to confirm the shape of the distribution. The following graphs represent the bootstrap distribution of some of the coefficients.







We can see from the distribution that the bootstrap distribution of the coefficients is not always perfectly Gaussian. For some coefficient, like *powerPS*, the distribution is asymmetric and is more weighted below the mean. In the next table, we compute the confidence interval for the three methods: **R**, bootstrap-normal, bootstrap-quantile.

X_i	$CI_{\mathbf{R}}$	$CI_{Boot-Norm}$	$CI_{Boot-Quant}$
1	[20,930; 23,020]	[21,160; 22,840]	[21,410; 23,000]
2	[-2,330; -2,160]	[-2,300; -2,100]	[-2,400; -2,160]
3	[-0.0034; -0.0031]	[-0.0047; -0.0016]	[-0.0040; -0.0010]
4	[56; 58]	[55; 59]	[55; 59]
5	[-3,800; -1,290]	[-3,340; -1,650]	[-3,630; -1,620]
6	[9,180; 12,670]	[8,610; 14,340]	[7,400; 14,100]
7	[-3,200; 5,550]	[-1,150; 3,550]	[-1,820; 3,180]
8	[3,800; 8,500]	[5,680; 6,115]	[4,200; 8,700]

As expected, we can see from the table that the confidence interval are not exactly the same. In general, all three methods give the same results. Bootstrap is sometimes more accurate than **R** in that it gives smaller confidence interval. Difference between normal and quantile method is not obvious.

Limits of the model

Outliers in the data

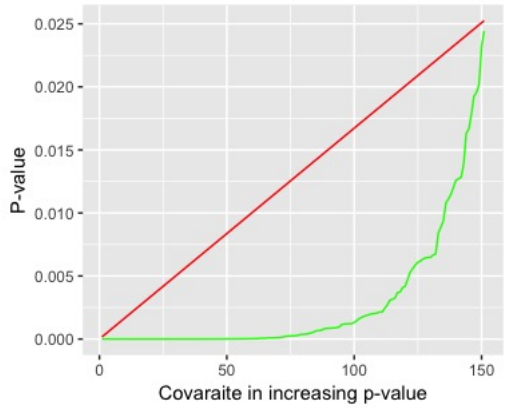
There are several limitations to our model. Perhaps one of the most obvious ones after reviewing deltas between predicted values and actual values in the dataset are classical and one-of-a-kind vehicles. Regarding the former, our model would look at the age of the car, along with the rest of the covariates, but it would not find in any of them any indication for the vintage or 'uniqueness' factor. Same goes for one-of-a-kind vehicles. That being said, these happen quite seldom, and it was not the purpose of our model. We were focused mostly on minimizing

RMSE. Had we included a covariate to measure 'uniqueness' (which is a non-trivial problem to begin with), we perhaps could have improved our accuracy with those instances, but at the expense of lowering our accuracy with the bulk of the vehicles. This, we believe, would have had a net effect of increasing the *RMSE* of our chosen model. Nonetheless, done correctly, it would have been an improvement, and this is not present in our current approach.

Collinearity
Among the covariates, only *age* and *kilometer* are correlated. But because *age* is used as a covariate and in interactions terms, the effect of *kilometer* is negligible. Hence, we have no reasons to be worried about collinearity with our dataset.

Omitted variable bias
Some variables are missing in the dataset. All the covariates from the dataset are objective metrics describing the car (age, kilometer, power, brand, etc). But we know that the price of a car, especially in the used car market and even more online, can be also a function of more qualitative features such as the color of the car, the condition of the car, the number of pictures in the ad, the quality of these pictures, the length of the description. Hence, having those would be interesting, especially to add to quantitative features.

Multiple-hypothesis testing
With regards to multiple hypothesis testing, we decided to go with the $Benjamini-Hochberg$ procedure, rather than $Bonferroni$, as the latter is seldom used in industry. Carrying out this procedure in our model based on the training dataset, results in us rejecting the first 151 null-hypothesis (which is around half the number of covariates). The graph below focuses on the $p-values$ (green line) that were below the $\frac{\alpha j}{p}$ line (red line). Although around half the covariates are considered null with the $Benjamini-Hochberg$ procedure, all our "universal" covariates are statistically significant different than 0. This remains true even with the Bonferroni procedure. Therefore, we are confident in our model.



Correlation or Causation?

With our model, we developed several significant relationships between the price of a car and different parameters of the car. We are willing to interpret this relations as causal and not just correlations. Indeed, from a human point of view, what are the questions someone considers when buying a used car? What mileage, what age, what type of car, what brand? Hence, from a human point a view, this factors appears as causes for the price of the car. From a statistical point of view (which is what matters), our relations as robust to every test we tried. Therefore, we are willing to consider them as causal relationships.

Discussion

Uses

The pricing-regression model could be used in multiple ways. One of these uses would be aiding the budget/forecasting process at eBay. For example, it could be used downstream of another model that predicts expected type of adverts to be posted on eBay; those predicted values would then be placed in the pricing model we have created to estimate the asking price, which in turn, could be used to estimate the potential revenue to eBay from a successful transaction (as they take a portion of the total transaction as commission). More interestingly, the model could be used to provide additional services to eBay users. For example, a signal could be given to the user before he/she posts the advert indicating that the price is too high/too low compared to what is predicted by the model. Likewise for users wanting to purchase a vehicle from an advert, there could be a label next to the price that would indicate how much higher/lower the advertised price is compared to what the model estimates, giving an indication if a good deal has been found or not.

Updating

Because our model is a simple linear regression with relatively few features (vs. the data available to train it), we would recommend that the model be updated on a regular basis. Certainly, it wouldn't make sense to keep it static, with new cars coming in every year, in the very least, this should be done on an annual basis, in order to account for these new models coming from the manufacturer's product line. However, as already discussed, due to the fact that is relatively easy to update the model, this should be done as frequently as possible, which in our opinion would be once per day.

Interpretation

As discussed in section one of this project, we took several filtering decisions during the cleaning step of the data. These should be taken into consideration when executing decisions based on the predicted values of our regression. To refresh the reader's memory, the filtering decisions were: i) years of cars out of the range of 1951 and 2017 were excluded, ii) cars with prices lower than 4 million and $1,000$ euros were also removed from the dataset, and iii) adverts with $powerPS$ higher than $1,000$ and lower than 20 were also

removed. All these steps need to be taken into account when using the model. For instance, it would be absurd to use the model to predict the price of a car that is outside the scope of the training data.

Data Collection

We obtained our dataset from Kaggle, which in turn, was scrapped from the eBay Germany website; insofar as the collection-process is concerned, there is nothing that we would've done differently. Some additional information that we would've liked to have to improve our model includes number, n, of previous owners; it is likely that a similar car in make and year, with the same kilometers, could be viewed differently in the market if the number of owners was different; in particular, we expect that the price would be inversely proportional to this number. An additional data point we would like to see in future analysis of used-car adverts is a categorical covariate, $upgrade - type$ to indicate if there were any upgrades done to the car beyond the standard ones from the original manufacturer; for example, some car enthusiasts may add the so called 'batman-layer' to their cars. Though this may be in the description of the vehicle in the advert, it would be easier to incorporate that data in the model by having a section in the advert that indicates what type of additional work was done on the car akin to the yes/no answer about un-repaired damages, which is present in the original dataset. We would expect cars having this type of work done on them to receive a higher asking price than those without.

Parting Remarks

Overall, as already expressed in Part 2 and earlier in this report, we are very happy with the results of the chosen regression model; not only do we achieve a very reasonable *RMSE* value, the results are also interpretable, and make sense! What is even better, is that we get an order of magnitude view on the impacts of age, distance-driven, and make (amongst others) on the expected price of a car, and we find this very interesting to both review and share. More complicated models would most likely achieve a better value in our main KPI for prediction, but we doubt the interpretation would be at the level of this model's. Additionally, it is a simple thing to train and store.