Cornell University

ORIE 4741

Final Report

Predicting the Price of Used Cars

Authors: Yihao Zhu, Xinci Liu, Lirou Luo, Liming Zhao

Semester: Fall, 2020
Date: 12/13/2020

CONTENTS

1 Introduction

The current pandemic is altering the used-car industry: transactions are shifted away from the dealership lots and largely digitized. It becomes essential for prospect buyers to know how to best utilize the information provided online to judge the value of their prospect investments. Buyers should understand how each feature is factored into the pricing of a used car, and whether their targeting vehicles possess features that would enable them to carry values into the foreseeing future. These days, there are a plenty of car-dealing website that provide details on different types of car models that were listed, yet most data features seem complex and uncorrelated to the general public.

In this project, our goal is to analyze the data we scrapped from https://www.truecar.com/used-cars-for-sale/listings to identify the primary factors influencing the listing price of used cars in the U.S., with the hope of using this information to benefit customers. This website provides general information on about 1 million used cars that were listed, such as brand, model, mileage, price, year of manufacturing, location, ownership and accident records. We will use these features to predict on the listing price of used cars, and these predictions will be validated with the actual prices listed.

2 Data Set Description

The latest available data set on used cars we could find was on cars listed online in 2017. Since cars are expected to depreciate overtime, we would like to have data on cars listed this year in order to make valid price predictions. Therefore we decided to use the data extraction tool *Bazhuayu* to scrap information on used cars currently listed on the website *Truecar*. We originally obtained 9,122 lines of records and 11 columns on car features. To tidy up the data, we used python to split the columns such as '$additional\_information$' that contains more than one feature, and also applied feature transformation such as one-hot encoding for features with set values (see Table 1). Since we are only using this overall data set identify popular models, we dealt with missing data by simply removing rows with '/' entries. More careful missing data imputation will be performed on individual model data sets later in this section.

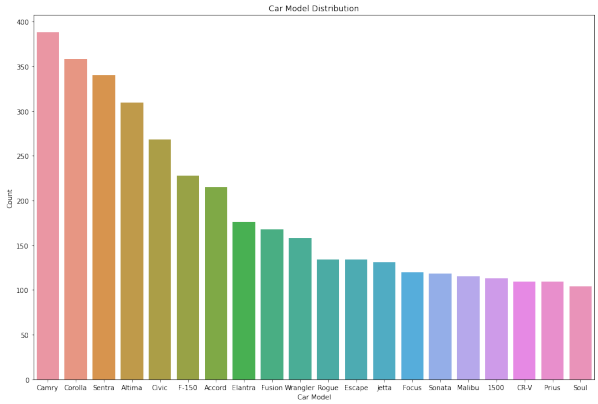

Figure 1: Car Model Type Distribution

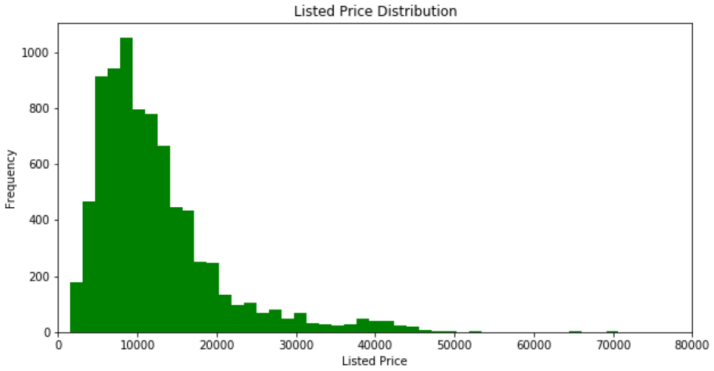

Figure 2: Listed Price Distribution

We used bar graph or histogram to plot each non-Boolean feature against count. As seen in Fig. 1, we are able to identify the most common model type of used cars listed on Truecar, which includes *Toyota*'s 'Camry' and 'Corolla' models, *Nissan*'s 'Sentra' and 'Altima', *Honda*'s 'Civic' etc.. Since there are 368 unique car model types in our data set, we decided that it would be more appropriate to select and only look at the top five most common car model types. We also looked at the price distribution (Fig. 2), and found that most cars are listed for prices ranging from \$1,500 to \$ 80,000, apart from an outlier of price \$158,778. The ipynb file in our repository also contains data visualizations for the distributions of mileage, interior/exterior color and location. With these observations, we decided to scrap another set of data for each of the five most popular car types from the *Truecar*

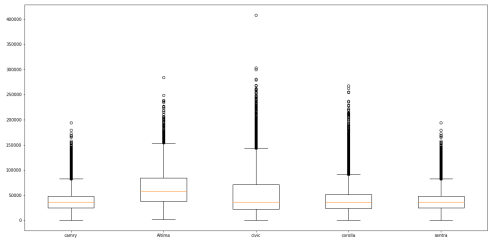

Figure 3: Mileage Distribution of Samples from Top 5 Most Popular Used Car Models

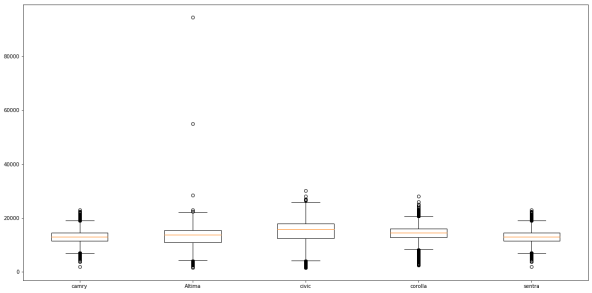

Figure 4: Listed Price Distribution of Samples from Top 5 Most Popular Used Car Models

website, this time setting conditions on the desired car model types, at the same time making sure that the samples cover a broad enough price range to reflect the population. In these five new individual data sets, we also included another feature called 'Trim Level', which indicates the version of a vehicle model and the set combination of features equipped. Possible entries for this column include 'L', 'LE', 'SE', 'XLE' and 'XSE'.

The box plots in Figure 3 and 4 display the price and mileage distribution of car samples from our five chosen car models, since previous preliminary analysis using best subset selection indicates that 'mileage' is an important price determinant. The upper and lower quantiles for the listed prices of Camry, Altima, Civic, Corolla and Sentra fall between $10,986 and $17,899, with the quantile box of Civic positioning more towards the higher price level and with a higher median of $15,794. We can see that several cars from the Altima data set have abnormal listed prices as high as $94,500, we removed these outliers to avoid distorting our statistical analysis. We expect this price distribution to be useful later when determining whether the average prediction error is significant or acceptable.

Table 1: Feature Summary from the Original Data Set

feature	**type**	**encoding**	**feature**	**type**	**encoding**
make[a]	categorical	/	state[b]	categorical	one-hot encoding
model[a]	categorical	/	color[b]	categorical	one-hot encoding
year[b]	numerical	/	value[c]	categorical	/
mileage[b]	numerical	/	value description	string	/
trim level[b]	categorical	one-hot encoding	URL	string	/
number of accidents[b]	numerical	/	picture	graphical	/
number of owners[b]	numerical	/	discount[d]	numerical	/
usage[b]	categorical	one-hot encoding			

[a]: For the prediction of a specific car model, *make* and *model* are not useful features.
[b]: The feature is included in the data processing.
[c]: The website evaluated the list price of the cars, e.g., excellent price and fair price, which should not be included in the prediction.
[d]: Discount varies dealer-by-dealer. We only intend to predict the listed price excluding the discount.

3 Data Set Processing & Model Fitting

3.1 Missing Data Imputation

After we applied the same data cleaning procedure described above to our newly scrapped data for the five most popular car models, we found that a small portion of the listed price could not be recognized by the scrapper,

resulting in missing data in the price column of the Camry data set. Therefore, we first tried to feed the data set into GLRM to impute the missing price. Here, we used the data set of *Camry* to illustrate the processing. The loss functions were selected based on the types of features: $HuberLoss$ was chosen to describe *price*, *year* and *mileage*; $BvSLoss$ was chosen to describe two ordinal variables *accidents* and *pastowners*; $WeightedHingeLoss$ and $LogisticLoss$ were employed to quantify boolean variables *usage* and *trim level*. The non-negative regularizer was selected, because all entries should be non-negative to have valid meaning. In regards to the rank, all trials from rank 1 to full rank were performed and we found rank $k = 8$ gave rise to the lowest MAE of the price in the observable space.

Subsequently, we generated a new data set, called A_{impute}, by retaining all data in the observable space and only filling the missing data with imputed values. The new data set was fed into models and obtained MAEs of \$2,335 on the test set.Another practical treatment to the missing data is to delete all samples with missing features. The rationale is that only a small proportion of samples has missing data. The new data set excluding samples with missing features was named as A_{delete}, fed into models and obtained MAEs of \$1,660 on the test set.Comparing the MAE of A_{impute} and A_{delete}, we found that A_{impute} resulted in a higher MAE. It may be explained by the selection of loss function and regularizer can be further optimized. Considering the fact that we have abundant samples, we decided to delete samples with missing 'owner' or 'listed price' entries for the sake of accuracy.

3.2 Feature Selection & Model Fitting

For feature selection, we decided to use lasso since lasso regression tends to set the coefficients of unimportant features as zeroes. The whole feature selection and model fitting process was done using R. To start with, we applied lasso regression to the Camry data set only in order to examine the feasibility of a model with reduced features. The cv.glmnet() function provided by the glmnet package is a great tool for fitting a lasso model over the entire data set to obtain features with non-zero coefficients. Essentially, the cv.glmnet() function performs a cross validation fit of a lasso model on the entire data set, and during the fitting process it searches for the possible optimal lambda values for the lasso regression.

After running the cv.glmnet() function, it returned two possible candidates for the best lambda value for lasso regression: "lambda.min" and "lambda.1se". Here, "lambda.min" refers to the lambda value which minimizes the test error from cross validation, so technically it results in the best lasso model. On the other hand, "lambda.1se" refers to the lambda value which produces a more parsimonious model than the "best model" while the model error is still within one standard error of the "best model". To examine the respective model complexity, we tried using both "lambda.min" and "lambda.1se" for our lasso regression on the entire data set. "lambda.min" gave a model with 75 features with non-zero coefficients, and "lambda.1se" resulted in a model with only 34 features with non-zero coefficients. Considering that there are 85 features and around 13,000 rows in the original Camry car data, ideally we want a simpler model to avoid overfitting, and hence we picked "lambda.1se" as the best lambda value for lasso. Accordingly, we have selected 34 features which will be included in our model for Camry car data. These 34 features are shown in Figure 5:

```
> feature_names
 [1] "year"              "mileage"           "L"                 "LE"
 [5] "XLE"               "XSE"               "Accident"          "Past.Owners"
 [9] "is_Personal_Use"   "AL"                "AR"                "AZ"
[13] "FL"                "HI"                "IA"                "LA"
[17] "MA"                "MD"                "MS"                "NC"
[21] "NJ"                "OH"                "SC"                "TN"
[25] "TX"                "VA"                "WA"                "WY"
[29] "Black.exterior"    "Blue.exterior"     "Gold.exterior"     "Gray.exterior"
[33] "Gray.interior"     "Red.interior"      "Unknown.interior"
```

Figure 5: Selected Features for Camry Data Set

The next step is to first generate a 10-fold cross validation split of the data set. Then, using 10-fold cross validation we fit a full model using all 85 features without any regularization and then compare its result to models with those 34 selected features. For models with selected features, we thought it would be a better practice to fit

three different models: one with no regularization terms, another one with L1 regularization, and the other one with L2 regularization. Contrary to the criterion of finding the best lambda discussed above, here when fitting model with L1 or L2 regularization (*i.e.* ridge or lasso regression) we actually want a model with less test error because we are not performing feature selection anymore. Therefore, it makes more sense to pick lambda values to be equal to "lambda.min" as opposed to "lambda.1se" for both ridge and lasso regression. Another important note is that we are using mean absolute percentage error (MAPE) as the metric for computing errors across all data sets. The reason behind picking MAPE rather than some other kind of error measurement is that we think car price prediction only makes sense when compared to the real target price. Essentially, MAPE measures what percentage is a prediction off from its real value and takes the average of all prediction points. The results we obtained from fitting four models on Camry data set are presented in Figure 6:

full_zeroreg_model <dbl>	zeroreg_model_with_selected_features <dbl>	ridge_model_with_selected_features <dbl>	lasso_model_with_selected_features <dbl>
9.607457	9.652270	9.611820	9.647839
9.698081	9.681745	9.637616	9.676796

Figure 6: Model Fitting Error on Camry Data Set

In Figure 6, the first row is the average training MAPE's from cross validation for each type of model; the second row includes the average test MAPE's from cross validation for each type of model. Comparing the full model with no regularization to the model with no regularization and selected features, we can tell that they both have similar training and test MAPE's. However, the difference between training and test MAPE on full model is much greater than that on model with selected features. This shows that it is very likely that the full model is overfitting the data set yet the other model is not, which further proofs that the model with features selected by lasso method has reduced chance of overfitting while still retains most of the accuracy of full model. Therefore, lasso is a feasible and effective way of performing feature selection. This warrants the use of lasso for feature selection on other data sets as well.

By looking at the second, third, and last column in Figure 6 we concluded that ridge and lasso regression models both made some insignificant improvement on the model without any regularization term, with ridge regression being slightly better than lasso regression in this case. The minor improvement could be due to regularization at work or it could also be attributed to the uncertainty involved in cross validation and model fitting. Currently we do not have a large data set on which we can thoroughly examine the difference among zeroreg, ridge, and lasso. However, it is best for us to always fit all three models on the rest of the data sets for consistency and integrity.

The results from feature selection and model fitting on Corolla, Sentra, Altima, and Civic car data sets are presented below:

```
> feature_names
 [1] "year"              "mileage"           "SE"                "XLE"
 [5] "XSE"               "Accident"          "is_Personal_Use"   "AL"
 [9] "AR"                "AZ"                "CA"                "FL"
[13] "GA"                "HI"                "IA"                "LA"
[17] "MA"                "MD"                "MS"                "NJ"
[21] "NM"                "NY"                "OH"                "OK"
[25] "SC"                "TN"                "TX"                "WA"
[29] "Brown.exterior"    "Unknown.interior"
```

Figure 7: Selected Features for Corolla Data Set

full_zeroreg_model <dbl>	zeroreg_model_with_selected_features <dbl>	ridge_model_with_selected_features <dbl>	lasso_model_with_selected_features <dbl>
7.974168	8.007282	8.011665	8.005076
8.046472	8.039904	8.040226	8.037660

Figure 8: Model Fitting Error on Corolla Data Set

```
> feature_names
 [1] "year"              "mileage"           "S"                 "SR"
 [5] "SL"                "Accident"          "Past.Owners"       "is_Personal_Use"
 [9] "AL"                "AR"                "AZ"                "CA"
[13] "FL"                "HI"                "IA"                "LA"
[17] "MS"                "MT"                "NC"                "NJ"
[21] "NM"                "NV"                "NY"                "OH"
[25] "SC"                "TN"                "TX"                "WV"
[29] "WY"                "Black.exterior"    "Red.exterior"      "Beige.interior"
```

Figure 9: Selected Features for Sentra Data Set

full_zeroreg_model <dbl>	zeroreg_model_with_selected_features <dbl>	ridge_model_with_selected_features <dbl>	lasso_model_with_selected_features <dbl>
8.618339	8.655209	8.677001	8.655625
8.692174	8.690087	8.708406	8.690424

Figure 10: Model Fitting Error on Sentra Data Set

```
> feature_names
 [1] "year"              "mileage"           "S"                 "SR"
 [5] "SL"                "Accident"          "Past.Owners"       "is_Personal_Use"
 [9] "AL"                "AR"                "AZ"                "CA"
[13] "FL"                "HI"                "IA"                "LA"
[17] "MO"                "MS"                "MT"                "NC"
[21] "NE"                "NJ"                "NY"                "SC"
[25] "TN"                "TX"                "UT"                "Green.exterior"
[29] "Unknown.exterior"  "Gray.interior"     "Unknown.interior"
```

Figure 11: Selected Features for Altima Data Set

full_zeroreg_model <dbl>	zeroreg_model_with_selected_features <dbl>	ridge_model_with_selected_features <dbl>	lasso_model_with_selected_features <dbl>
9.412487	9.505305	9.572284	9.506564
9.615978	9.598013	9.656051	9.598960

Figure 12: Model Fitting Error on Altima Data Set

```
> feature_names
 [1] "year"              "mileage"           "LX"                "Sport"
 [5] "EX.L"              "Touring"           "Accident"          "Past.Owners"
 [9] "is_Personal_Use"   "AL"                "AR"                "DE"
[13] "FL"                "GA"                "HI"                "IA"
[17] "ID"                "LA"                "MA"                "MD"
[21] "MN"                "MO"                "MS"                "MT"
[25] "NC"                "NE"                "NH"                "NJ"
[29] "NM"                "NY"                "OH"                "OR"
[33] "SC"                "TN"                "UT"                "WA"
[37] "WV"                "Black.exterior"    "Brown.exterior"    "Gold.exterior"
[41] "Green.exterior"    "White.exterior"    "Beige.interior"    "Gray.interior"
[45] "Unknown.interior"
```

Figure 13: Selected Features for Civic Data Set

full_zeroreg_model <dbl>	zeroreg_model_with_selected_features <dbl>	ridge_model_with_selected_features <dbl>	lasso_model_with_selected_features <dbl>
8.873327	8.921079	8.923296	8.918340
8.958055	8.972501	8.967764	8.969552

Figure 14: Model Fitting Error on Civic Data Set

4 Results Interpretation

4.1 Error Discussion

Compared with the listed prices, the mean absolute percentage errors (MAPE) for the predicted prices of the five car models are all below 10%. The rationale of the MAPE error can be justified by exploring and comparing the predicted prices from other sources, *e.g.*, *Kelley Blue Book* (KBB). In this regard, we are able to specify the features, which are included in our model, in the prediction system of KBB and see its price range. Take 2019 Camry SE as an example, shown as Figure 15, the predicted price was $19,101 accompanying with a fair price

range from $17,172 to $ 21,029. If converting to mean absolute percentage error, the predicted price will be $19,101 ± 10.1%. This proved the rationality of our MAPE errors.

Figure 15: Predicted price range of 2019 Camry SE given by KBB.

4.2 Application

An interesting application of the model was conducted to predict the price of a car owned by one of our friends from CA. The model of his car is 2007 Sentra with beige interior. The car has been driven over 97,701 miles, and it has 5 past owners and 0 accident record. He bought this car two years ago for $5,500, and intends to sale this car. The suggested price given by our model is $4,240 ± 10%. This a reasonable price, implying the car depreciating $630 per year.

4.3 Weapon of Math Destruction & Fairness

One potential Weapon of Math Destruction resulting from our car price predicting model is that the model may change the behavior of car dealers and car buyers after they know the results, thereby leading to changes in prices of different used cars. Car dealers can interpret the results to see which cars have a low rate of devaluation and are more popular among consumers. Therefore, car dealers will be more willing to purchase these models and increase prices appropriately. Based on this conjecture, the value-preserved car will be more value-preserving, even be over-priced. On the contrary, some cars that are predicted to be not value-preserved may be less favored by car dealers and consumers. Therefore, the predicting model has a reinforcement impact on the behaviors of the consumers and dealers.

In regards to fairness, the features included in our model do not contain protected attributes, *e.g.*, race, color, national origin, sex, religion, etc., as well as other covariates that may associate with the protected attribute, *e.g.*, zip code and race. Therefore, our model will not lead to demographic discrimination.

5 Conclusion

Our project showed promising price prediction results by building models each with features tailored to one of the five different car models: 'Camry', 'Civic', 'Sentra', 'Altima' and 'Corolla'. We found that for different car models, the set of significant price determinants can vary greatly. Generally speaking, year of manufacturing, mileage, the number of past accidents and past owners all play important roles in fluctuating the listed price of a used car. In addition, some states can have a noticeable impact on used car's price, because in some states, such as Florida, used cars are generally cheaper than those in other states. Cars with unique exterior and interior color also tend to be listed at a higher price range. Our work suggests that since cars with different models tend to have different price determining factors, it is necessary to train machine learning models with different features for different car model type.

Customers can use our model to figure out the expected price of the used cars that they are interested in. With such knowledge, they can take the initiative in price negotiation. Car dealers can also use our model to quote price for customers. In the future, we would like to extract car price data sets from more different resources and for more different car models, so that we can establish a more comprehensive used car price evaluation system.

References

[1] "Used Cars Listings" *Truecar*, 2 December 2020, www.truecar.com/used-cars-for-sale/listings
[2] "What's My Car Worth?" *Kelley Blue Book*, 12 December 2020, https://www.kbb.com/used-cars/
[3] O'Neil, Cathy. Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy. Penguin Books, 2017.