Project Report

Estimating used car selling prices

Applying Linear Regression on Craigslist Dataset

Introduction and related work

Nearly everybody sooner or later needs to buy or sell a car. And every time people want to get a fair deal. There are actually two ways to find out a car price. First - to manually check prices on different car marketplaces like CarGurus, Autotrader, Craigslist, and then come up with a fair price based on those listings. People use this way a lot, but it has some disadvantages. For example, they can check just a small number of listings, and by not understanding the market well enough make a poor pricing decision that is biased towards very high or low price.

Another way to find out a car value is to use services that can predict a trade-in value. Examples of such services would be Shift, Carfax and many others. They have access to huge car databases plus well trained machine learning models that can predict a price just based on a car's VIN number. However, there is no information which models and algorithms they are using. One major disadvantage of those services is that they make money on trade-ins, so the predicted price will always be lower than a real market value, although it can give people at least some idea.

This project is located on the intersection between those two ways of pricing, so the goal is to eliminate all the disadvantages mentioned above and create a machine learning model that will predict a fair car selling price.

Data

The data used for this project is a big collection of scraped car listings from the Craigslist website (all the United States). Total number of listings is around 500K, number of columns is 25. Scrape month: January 2020. All the data columns are well explained in the presentation slides.

Preprocessing

The first thing to do was to select only useful columns from all the 25 available in the dataset. Columns that would become features in a machine learning model.

The following columns were selected:

- manufacturer
- model
- year
- odometer
- price
- state
- drive
- title_status
- condition
- paint_color
- type
- transmission

After selecting the right columns, the next challenge was to clean all of them. Here is an explanation of the most important things that were done:

There were too many different car models - 35K. It happened because when people create a listing on Craigslist website, they can put there anything they want, so there was no unified naming. However, the first word in each model name was perfectly describing each car. For example “camry new generation” needs to be just “camry”. That’s exactly what was needed - splitting a model string by empty space and then taking only the first element. Result: 35K models to 6K models. Manufacturer and make columns were combined together afterwards.

To make the list of car models even smaller, the decision was to choose only top 100 cars that were the most frequent in the dataset. Although, in the next section about Linear Regression that number dropped to top 50 for the last (and the most precise) model.

Other cleaning included states, condition, type, title_status, and transmission columns and its well explained in the jupyter file.

There was a need to limit odometer, year and price ranges, as some of them were too high or too low. No data scaling was done, as it is not needed for a linear regression model and will not change the final results.

At the end of preprocessing there was a mix between categorical and numerical columns. Dummy variables of 0s and 1s were created for categorical ones. All remaining N/As were dropped.

Linear Regression Modeling and Evaluation

This part is about multiple tries of training the model and evaluating the results, adding or removing different features, testing multiple numerical ranges in prices, years, odometer readings, which finally led to a well performing model.

The first regression model was based only on 4 features: Price, odometer, year and model. Those features are the core ones and the most important in this project. Other categorical columns improved accuracy just slightly.

Evaluation for each regression model was be based on 3 measures: Mean Absolute Error(MAE), Root Mean Squared Error(RMSE), R2 Score.

The first run was definitely not the best, with MAE: $3136, RMSE: $4458, and R2: 0.68

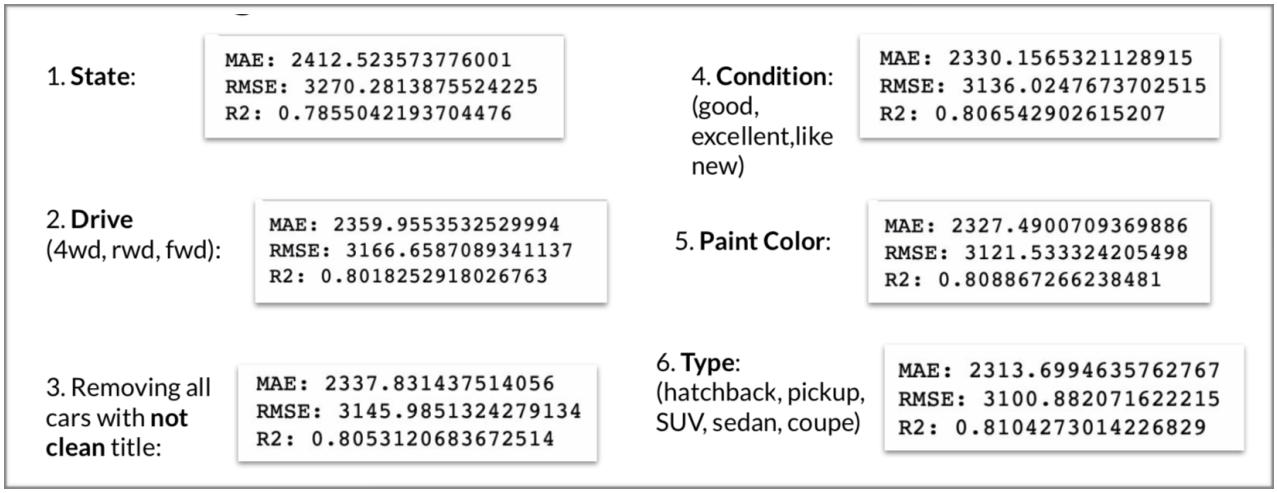
The main reason for this was too big price range, which was from $1K to $70K. There was too many big errors in predicting high-end cars, so the price range was adjusted to $2K-$30K.

The second run was better: MAE: $2461, RMSE: $3337, R2: 0.77.

That was still not enough and the goal was to somehow drop the mean error below $1K and try to eliminate as many huge errors that were more than $5K.

The next step was to start adding categorical features and checking how they change the model.

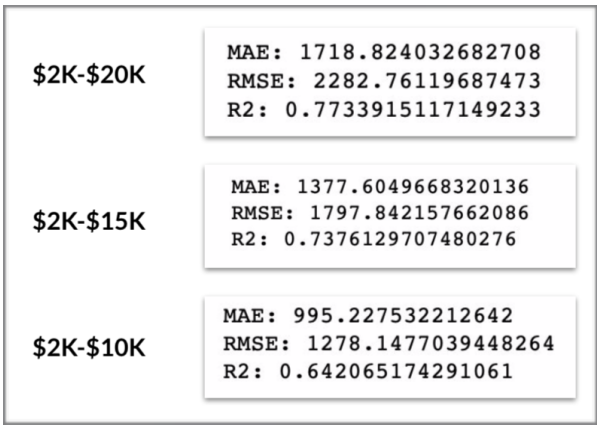
The following picture is taken from the presentation slides and perfectly shows those attempts:



All those features were added in the same order as numbers, and they were added cumulatively. Also, on this step categorical column with a car transmission type was dropped because it actually made the results worse.

The model was slightly better, but still not ready for a real world predictions, as the error was too big.

The idea was to continue adjusting price, as in the past it worked well. And the results were great.



It turned out that the model worked best on cars that were in the price range $2K-$10K. However, for the last model, there was a need to use top 50 models instead of top 100, because by limiting the price range number of cars in the dataset also dropped. So when selecting top 100, the last ones in the list had very few data points.

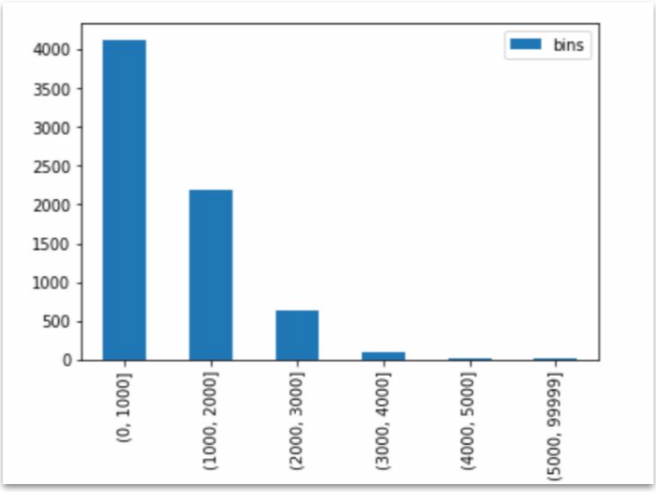
Two more Linear Regression were also tested: Lasso and Ridge, although they had nearly identical results to the simple Linear

Regression.

Results

It was possible to achieve the desired MAE under $1K and to minimize big errors.

bins	
(0, 1000]	4128
(1000, 2000]	2189
(2000, 3000]	626
(3000, 4000]	100
(4000, 5000]	22
(5000, 99999]	15



From these results we can see that:
- 89.4% prediction errors were not higher than 2K.
- 58.3% of predictions were 0-1K close to the real prices.

It is probably still not the best model and has a room for improvements, but it's important to remember that linear regression will never give a very precise prediction, so this result can be considered as pretty precise one.

Can it be used in real life? Yes! But not as a main source of truth. It still has errors, although if you are lucky, the model will give you very precise results.
This project can be just a good starting point to dig deeper into the car price prediction.
It has a big potential and many companies using machine learning for it. They just have a lot more data and better ML models.