Predicting Pre-owned Car Prices

Project Luther

Anjali Sebastian - October 14, 2018

Motivation

I recently made two car purchase decisions within the last 6 months. Although I ended up purchasing a new car both times, I looked at the pre-owned car market while making my decision. I spent a lot more time looking at pre-owned cars than new cars even though my final decision ended up being a new car . One of the key reasons behind this, is the high variance and uncertainty in purchasing a pre-owned car. The large number of parameters that are provided, often confuse more than assist in the decision making process. I kept thinking to myself if only there was a simpler way to quickly narrow down on good deals. The key motivation behind this project is to predict pre-owned car prices and use the model to identify the best deals - in other words develop a pricing guide that actually works.

Design

I used data from www.carmax.com which I scraped and cleaned. As part of my cleaning I looked for outliers and removed these outliers. I encoded all categorical features before I started to develop the model. Before beginning the modeling process I carried out data exploration which involved looking at the correlation between features and identifying possible feature engineering that could improve my model. While exploring my data I noticed that some of my parameters were correlated with each other. For eg. “engine size” and “number of cylinders” are highly correlated (0.90+). Keeping both these features is just redundant so I used “engine size” as it is more continuous. I transformed my target aka “price“ with a log transformation as it was right skewed. The right skew occurs because the dataset has very few data points at the high end luxury level (price 35,000 USD +). This also indicates that our error after modeling for this region will be much higher.

Before I started modeling I split my dataset into train set and a test set. I used only the train set to arrive at the best model by comparing various regression algorithms. I used StandardScaler() to scale all my features to the same level since the parameters have widely different ranges (eg. miles in thousands vs horsepower in hundreds). I used the LASSO and

RIDGE algorithms with PolynomialFeatures for degree 1, 2 and 3 to see which fitted the best. I also did cross validation to ensure that I wasn't overfitting. My best model was a RIDGE with Polynomial Degree = 2 and StandardScaler(). The fit for the train set with this model was as follows:
Mean Squared Error = 0.01636
R^2 = 0.88949

After model selection I tested the model using the test set.

Data

I used data from www.carmax.com which I scraped and cleaned. As part of my cleaning process I looked for outliers and removed these outliers. I did not try to impute missing values as I was able to scrape 2000+ data points and the missing values tended to be for certain parameters. After performing cleaning and dropping data points with missing values my dataset was around 1700. I also started out with a very large number of parameter 30+ that I narrowed down by feature engineering and more specifically looking at p-values and correlations.

Data Features used for the Model

Continous	Categorical
Y = Price	Brand - encoded as Luxury (0-1)
Year	Exterior Color - encoded as light, dark, and prime
Mileage	Engine Type - encoded as normal, turbo, alternate
Drive Type	
Engine Size	
Horsepower	
Compression Ratio	
Curb Weight	
Seating Capacity	
City EPA MPG	
Highway EPA MPG	

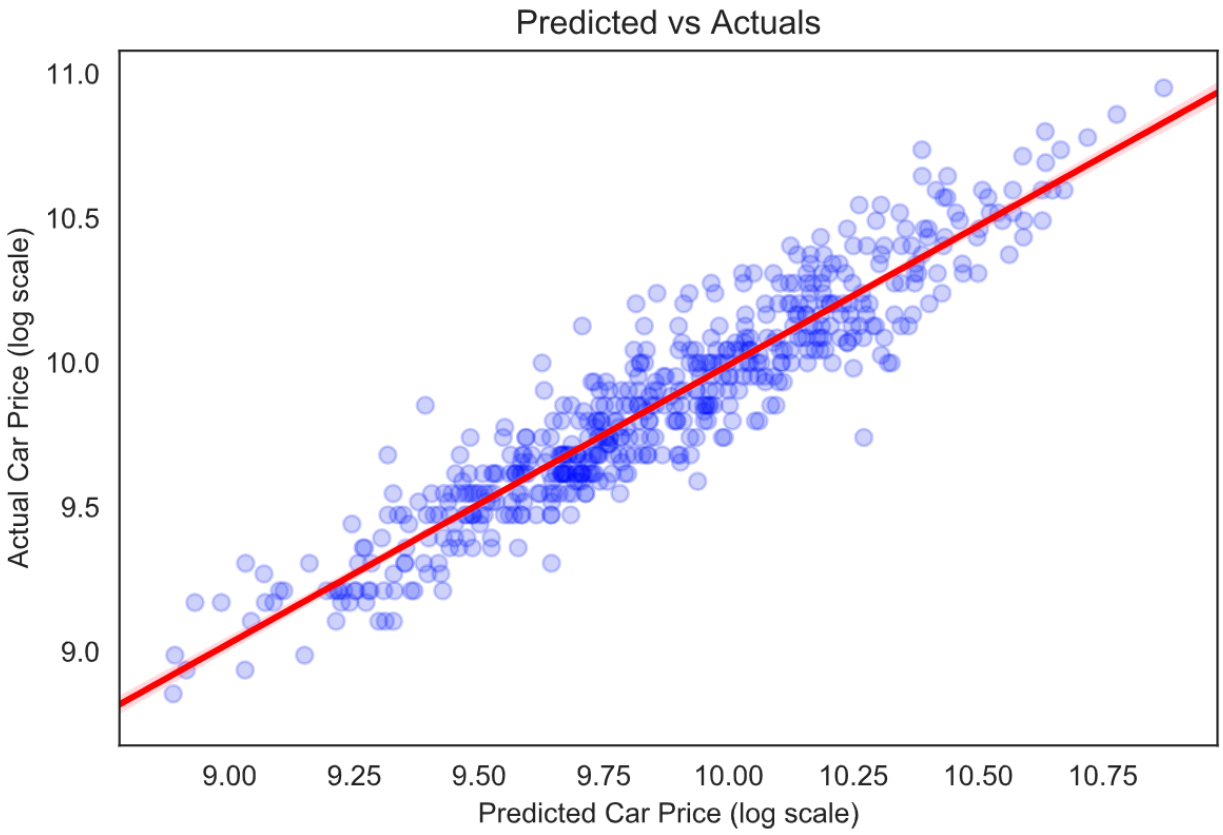
Tools

IDE	Jupiter Notebooks	
Data handling and cleaning	Numpy	Pandas
Data Visualization	Matplotlib	Seaborn
Web scraping	Selenium	
Modelling	Statsmodel	Sklearn
Documentation	Powerpoint	Typora
Version Management	Github	

Results

On the test set the model performed well and the Mean Squared Error and R^2 where similar to the train set.

R^2 Test = 0.876
MSE Test = 0.017
Error % = e^(sqrt(MSE)) ~ 1.1%

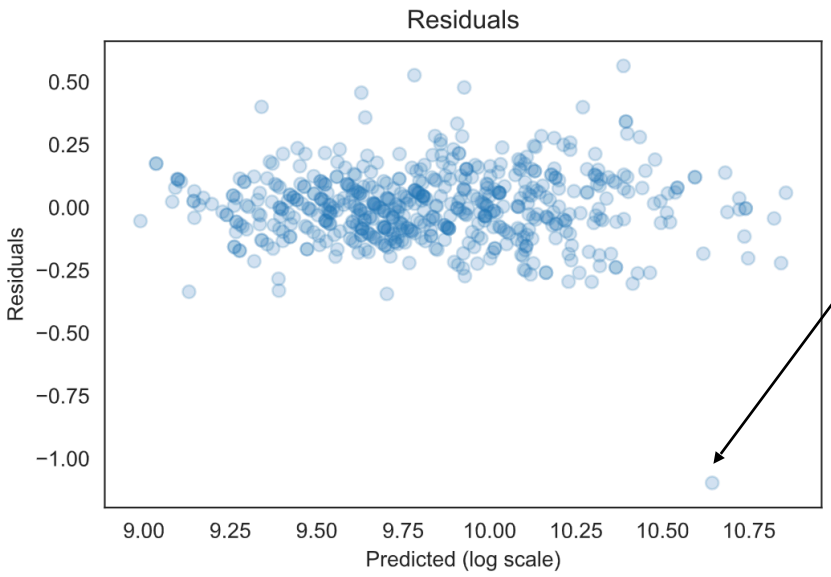


Real World Use

The model can be used to find the best deals by looking at the residuals of the test data. Test data with the highly negative residuals are very good deals. The following is a list of cars where the residual value is lower than -0.25.

Stock_n	Year	Brand	Model	Price	Mileage
16327825	2014	Dodge	Ram 1500 Express	21998.0	30000
16004348	2012	Ford	Taurus SEL	10998.0	75000
16238775	2014	Ford	Mustang	16998.0	22000
16004903	2016	Jeep	Cherokee Sport	17998.0	35000
16424417	2013	Volvo	XC60 T6	21998.0	45000
16497145	2016	Infiniti	Q50 Premium	21599.0	43000
16237691	2015	Dodge	Ram 1500 Tradesman	16998.0	39000
16250755	2016	Chevrolet	Silverado 1500 Work Truck	22599.0	2000
15777010	2017	Honda	Accord EX	14599.0	13000
16497595	2015	Nissan	Quest S	17998.0	27000

The model can also be used by companies like carmax to check if they have any erroneous or inaccurately priced cars on the website. When first plotting the residuals there was one very significant outlier. This outlier had a residual that was less than -1.0 while almost all other residuals where between 0.50 to -0.50.



GMC Yukon SLT, 2017

This outlier corresponds to an erroneous entry on the website. A GMC Yukon SLT, 2017 which was priced at only 13,000 USD when it should be around 40,000 USD. Which is what the model predicted.

What I would do differently

I would have liked to have added another data source in addition to carmax either through scraping or from a pre-existing dataset. I was particularly interested in modeling the depreciation of a car and for that I needed the original price which I could have got from Kelly Blue Book or a similar site. I was unable to do this as there was a shortage of time.

I would spend more time on feature engineering instead of tweaking the model. I definitely think that more exploration of the data would have given better results. I typically like to explore data using Tableau before I look at in in python which I was unable to do for this project. I did several Linear Regression Models which were not as good the LASSO and RIDGE models and were not as useful.

I felt that I should have focused more on the real world use, and on the presentation.

Future Work

- Use original price as a feature to better predict prices.
- Predict car depreciation of a new car and key features that either increase or decrease depreciation.
- Add more data points
- Build models for specific a brand and year