Car Price Prediction from Auction Prices

Brett Huffman - CSCI 5300 - Main Project Phase I

Abstract

The objective of this project is to harness past auction data to build a car price prediction engine. Auction data was normalized, split into training and validation sets, and used to train and validate a Logical Regression Model Neural Network. The sole output of the system will be the estimated vehicle price.

Buyers, dealerships, lenders, and insurance companies would all be interested customers of such a service.

1 Problem To Be Solved

For a used car shopper, understanding the true value of a car has always been one of the greatest unknowns in the buying experience. The used car's price is composed of the auction price, the dealer's costs, and the dealer markup. Anything a buyer pays for a car above the auction price and dealer costs is pure profit for the dealership and expense for the buyer.

Additionally, if a dealer gets a vehicle from trade-in, they will rarely pay above auction price. This makes the auction price a good estimate of a dealer's base cost from trade-ins.

Therefore, the auction price is an important aspect of car buying and gives purpose in performing this project.

A second purpose for this project could be to check the accuracy of the Manheim Market Report, the commercial product which tracks vehicle resale prices. Most professionals in the business use MMR for determining true wholesale vehicle price ([Rep21]).

2 Data Set Source

The data set source for this project is a public Kaggle.com repository, Used Car Auction Prices ([Tun15]). The data was scraped from auction results in 2014 and 2015.

One deficiency of the data is that it is stuck in time and has not been kept up-to-date. However, should the project prove useful, the system could be re-trained with current data and be indispensable for users.

Another problem is that the data does not include all vehicles and models. It is limited to only models that were auctioned during the 2014-2015 period. However, because the data source is so large – nearly 559K rows – a good sampling of most modern vehicles is represented.

Finally, there are a number of missing fields in the data. The most blaring omissions are make, model and odometer readings. These problems were handled in various ways in the data preparation:

- Missing Make and Model rows were entirely eliminated from the data set as they would yield almost meaningless results and only offer unwanted "noise" to the results.
- Missing Odometer data was replaced with -1 values

3 Data Distribution

Extensive modifications needed to be accomplished on the data set due to bad data distribution. The output labels were heavily imbalanced due to several high-priced sales (including a $240,000 super car sale). This caused the output distribution to be heavily skewed to the lower end of sales prices. Almost 80% of the data was within the first 10% of the data.

To fix this problem, any rows with a selling price greater than $40k were removed. This yielded the output label distribution shown in Figure 1.

Another interesting modification that had to be made was to the rows with super-high odometer readings. There were several high values with the highest being 297,000 miles. As with the sales price, these odometer values were causing data to be skewed and anything above 200,000 miles were removed.

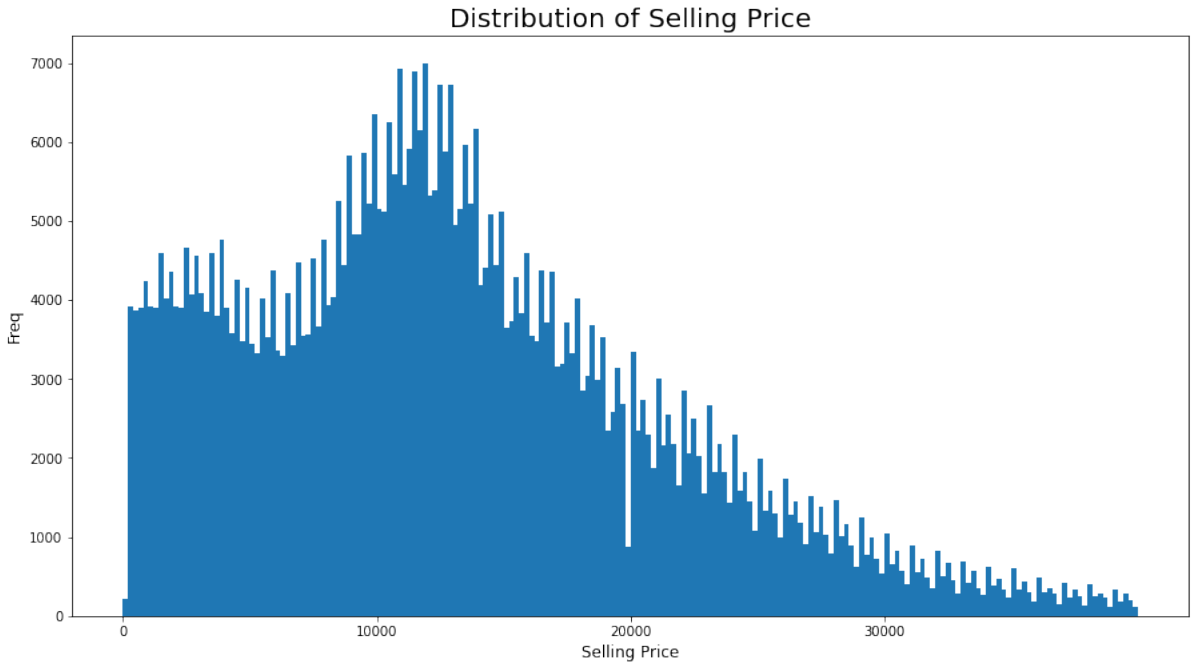

Figure 1: Final Output Variable (Selling Price) Distribution

This change yielded the distribution shown in Figure 2.

A keen eye will see that there is a large group of odometer readings right next to the 0 position. These rows had null odometer values and were purposefully set to -1 during data cleaning. For now, these rows are going to be left in the data. Should later phases of this project yield poor results due to the -1 values, they will be removed. However, they make up such a large portion of the data, they will be left in for now.

All other data elements seemed to be well distributed. For instance, car auctions by state were not terribly imbalanced (Figure 3).

Additional items such as brand are well distributed as can be seen in this Distribution by Brand (Figure 4).

An interesting add-on to this project might be to look up each of the VIN (Vehicle ID Numbers) and retrieve even more relevant information about each car. Each row of auction data contains the vehicles VIN. However, that step will only be performed should more specific information be needed.

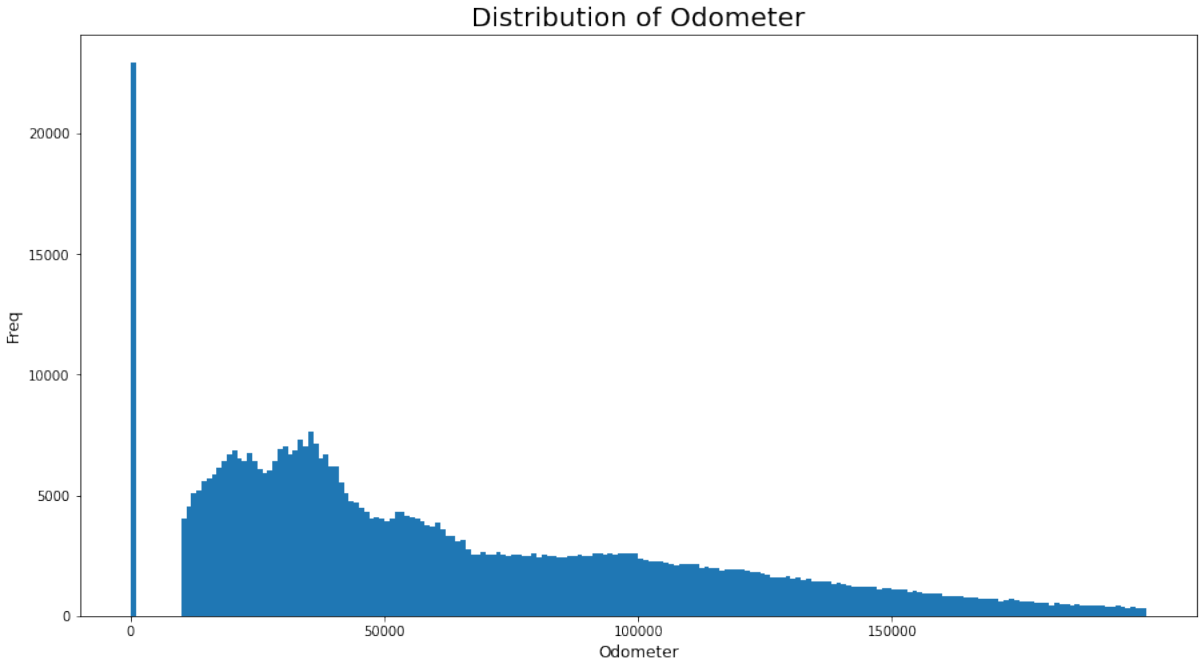

Figure 2: Final Odometer Distribution

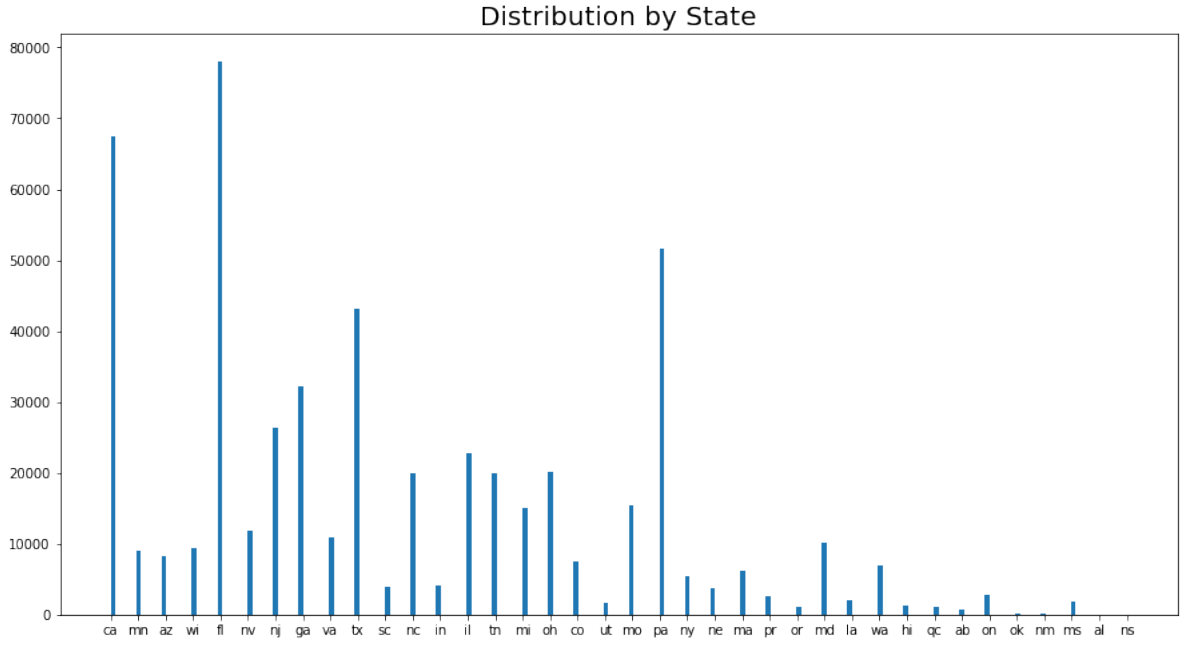

Figure 3: Distribution by Auction State

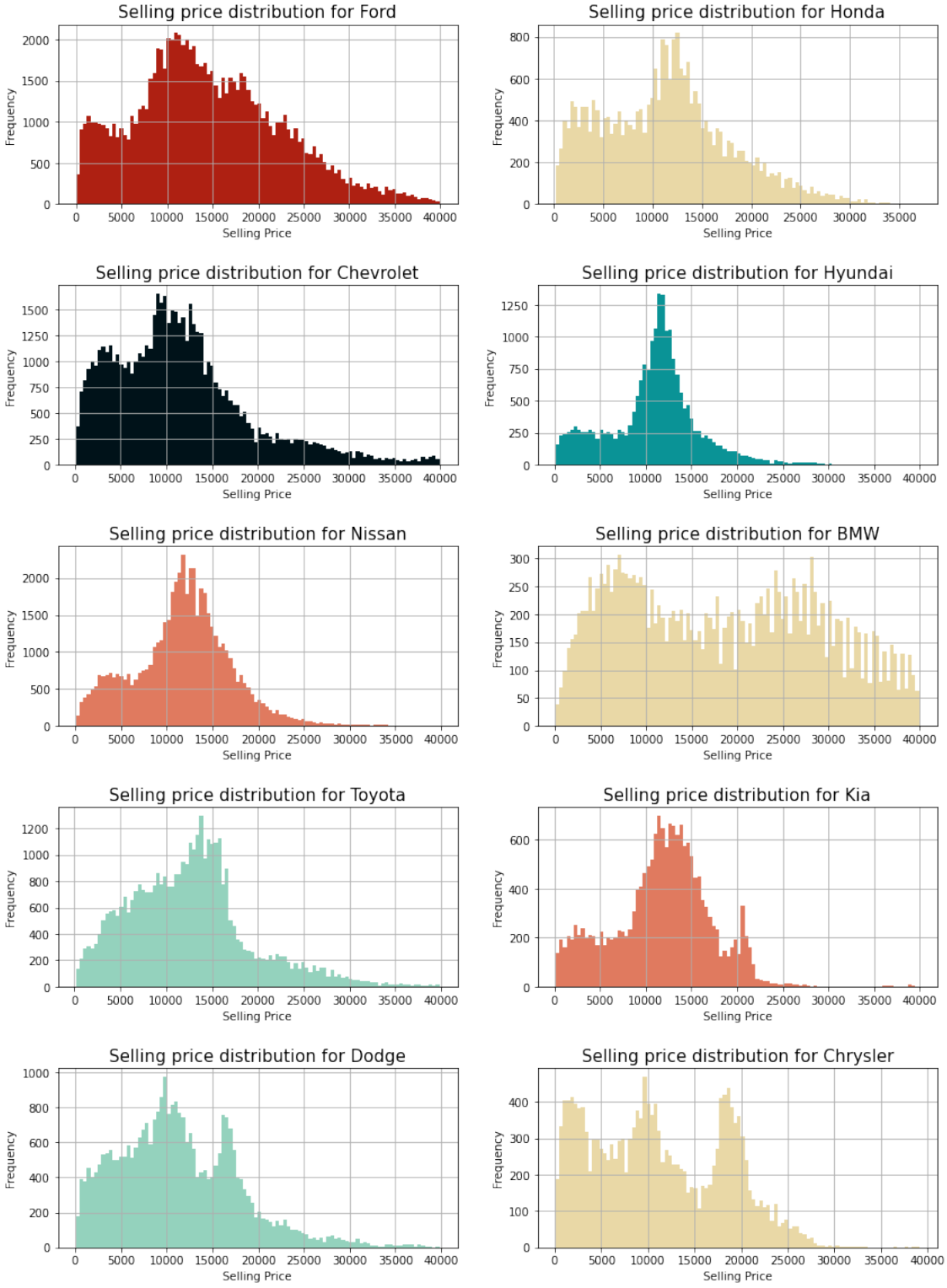

Figure 4: Distribution by Brand

	year	make	model	body	transmission	state	condition	odometer	color	interior	mmr	sellingprice
0	2015	Kia	Sorento	SUV	automatic	ca	5.0	16639	white	black	20500	21500
1	2015	Kia	Sorento	SUV	automatic	ca	5.0	-1	white	beige	20800	21500
2	2014	BMW	3 Series	Sedan	automatic	ca	4.5	-1	gray	black	31900	30000
3	2015	Volvo	S60	Sedan	automatic	ca	4.1	14282	white	black	27500	27750
5	2015	Nissan	Altima	Sedan	automatic	ca	1.0	-1	gray	black	15350	10900

Figure 5: Data sample before conversion of strings to numbers

	year	make	model	body	transmission	state	condition	odometer	color	interior	mmr	sellingprice
0	2015	0	0	0	0	0	5.0	1.798528e-06	0	0	0.000023	0.000066
1	2015	0	0	0	0	0	5.0	-1.080911e-10	0	1	0.000024	0.000066
2	2014	1	1	1	0	0	4.5	-1.080911e-10	1	0	0.000036	0.000093
3	2015	2	2	1	0	0	4.1	1.543758e-06	0	0	0.000031	0.000086
5	2015	3	3	1	0	0	1.0	-1.080911e-10	1	0	0.000017	0.000034

Figure 6: Data sample after conversion of string fields to indexes and normalization with normalized fields highlighted

4 Data Conversion and Normalization

At this point, the data looked as shown in Figure 5. Many columns including Make, Model, Body, etc were being represented as strings. They needed to be number values for proper inclusion into the neural network.

The Pandas library was used to convert each column into a number representation of each string. A custom function was built for each data element and a master array was kept for each data field converted. This will allow users to return values back to their original string representation and to duplicate the findings with other data.

Once the data was completely comprised of numeric data, Data Normalization was accomplished by using the mean average followed by the standard deviation. This was completed as described in the normalization training video, "[AI] How to normalize and un-normalize a tabular data for neural networks?" [Adh20].

The normalized data with the changed fields highlighted can be seen in figure 6.

References

[Adh20] Badri Adhikari. [ai] how to normalize and un-normalize a tabular data for neural networks? https://www.youtube.com/watch?v=Tu8Dl3zorgg, 2020.

[Rep21] Manheim Market Report. Manheim market report valuation. https://publish.manheim.com/en/services/valuation, 2021.

[Tun15] Bojan Tunguz. Used car auction prices. https://www.kaggle.com/tunguz/used-car-auction-prices, 2015. Data scraped and posted by Bojan Tunguz.