Taylor Imhof

Bellevue University | DSC 680

Project One: Milestone #2 (Draft White Paper)

11/12/2022

BUSINESS PROBLEM

Prospective used car buyers are getting gouged when purchasing a used car from shady dealerships and auto shops. There should be a better way for the customers to bargain for a better price and have a clear understanding of the prices they *should* be paying.

BACKGROUND/HISTORY

Buying a used car can be such a pain. Not only are there so many different auto shops and dealerships to choose from, many of them are also looking to optimize their profits so they might be asking for much more than the vehicles are valued at. Whether buying for a child or simply to save money, more and more people each year are deciding to drive used vehicles instead of buying new. As such, it is important to equip buyers with more tools so that they can head to the dealership with a better understanding of what they are getting themselves into. As a driver of a used vehicle myself, I purchased my current car through USAA's guaranteed price service where the bank had done their own research to know how much the vehicle is worth and had an agreement with select vendors to sell at a certain price (i.e., no added fees or price gouging). In a similar effort, I felt that I would be able to put my new machine learning skills to the test and create a price prediction model that prospective buys could use in order to understand the types of vehicles they could be able to afford given certain desired characteristics about the car itself.

DATA EXPLANATION (DATA PREP/DATA DICTIONARY)

The dataset that I used for this project can be found at the University of California Irvine's Machine Learning Repository. I found an automobile data set that I felt measured a lot of interesting features that could have predictive power on a vehicle's selling price.

Link to dataset: UCI Machine Learning Repository Automobile Data Set

There are 205 records each with 26 features (25 explanatory and 1 dependent/target variable). For this business problem, the target variable that I attempted to predict was the <price> or the selling price of the vehicle. As stated previously, there were quite a few features contained in this dataset. Some of the more common characteristics associated with vehicles such as make, fuel type, weight, engine, horsepower, and city/highway miles-per-gallon (mpg) were included. There were also some more interesting features such as engine type, physical dimensions, stroke, bore, and compression ratio.

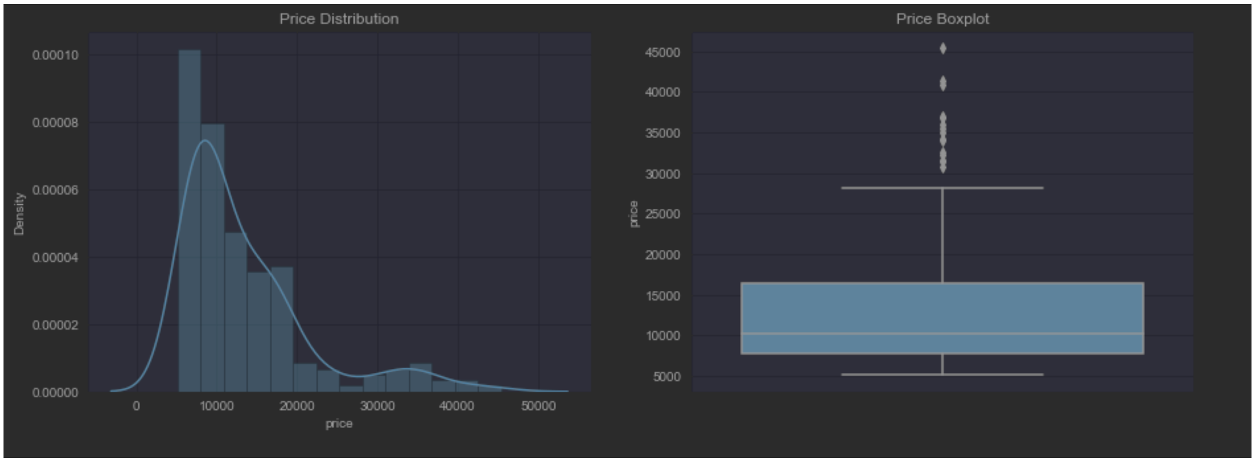
It is also worth mentioning that this data was captured and donated to the repository in 1987.

METHODS

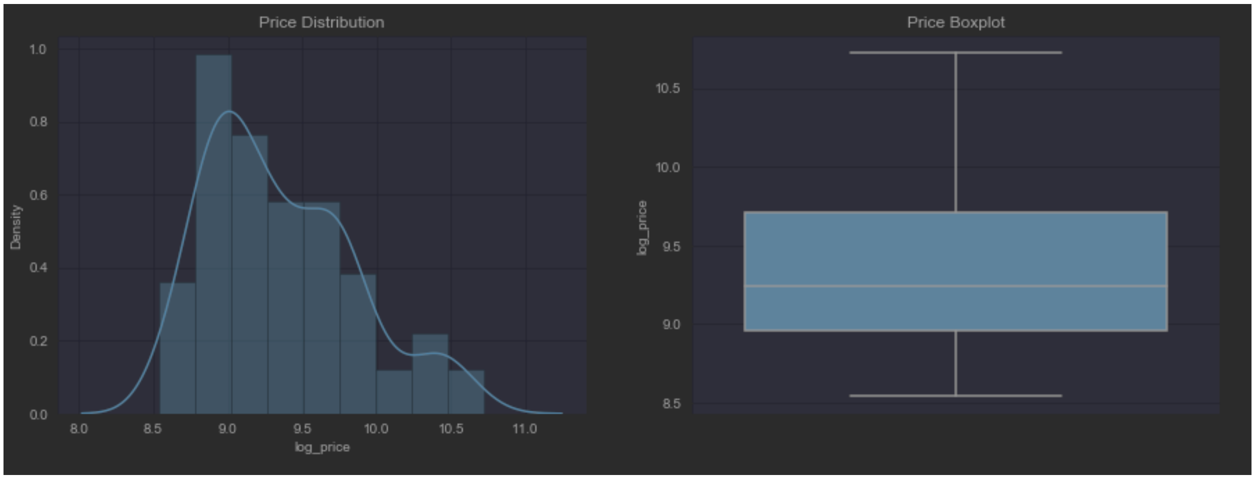
Since this business problem entails predicting a continuous price target variable, I knew that I was at the very least going to be training a linear regression model, as this type of model is the most commonly used to solve this type of problem. Before model training, however, there were some data cleansing steps that needed to be undertaken. I utilized Python's Pandas library in order to read the data from a comma-separated-values (csv) format into a dataframe so that I could work more easily with the data. Once the data was read in, I analyzed some of the summary statistics of the features. I also examined how the data types of the features were encoded and discovered there were quite a few categorical variables that needed to be considered before using them to train a machine learning model. In order to convert these variables into values that would be better understood by a machine, I encoded them using the <LabelEncoder> object from the Sci-kit Learn Python package.

One of the most glaring data cleaning objectives I needed to accomplish was extracting the manufacturer's name from the supplied <CarName> column. To do this, I made use of a simple lambda function that splits each value by the spaces and captures only the first element of the long names. There were also some typos in some of the manufacturer names, so I corrected those as well. I also dropped two other features that were not going to be useful in predicting the price, such as the supplied <car_ID> and <symbolling> features. Also, it was fortunate that this dataset contained no missing or duplicated values.

After cleaning the data, I wanted to create a few visualizations so that I could get a better understanding of the underlying data trends. First, I wanted to see how the target price column was distributed in the dataset. To accomplish this, I leveraged the Seaborn and Matplotlib Python packages. I created a distribution plot as well as a box plot.

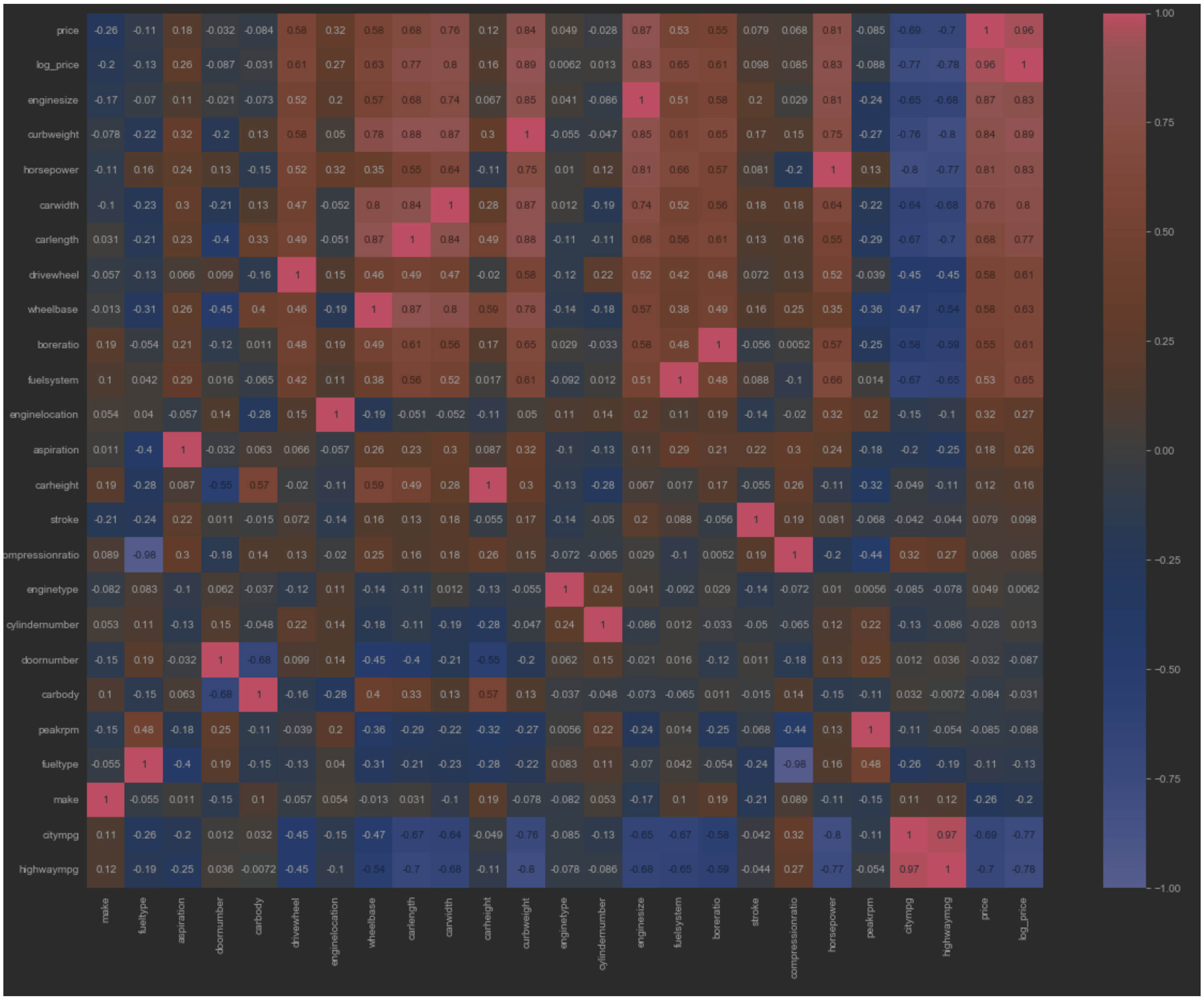


As can be seen in the distribution plot, it did appear that the price column exhibited a slight positive skew, and a mild bimodal distribution. After reviewing the boxplot, it did highlight that there were quite a few potential outliers. However, I did not decide to remove them due to the small nature of the dataset. Since the price column was positively skewed, I wanted to see what a logarithmic transformation on the column would look like. To do this, I created another <log_price> column within the dataset and provided it the logarithmic value of each price using NumPy's logarithm function.



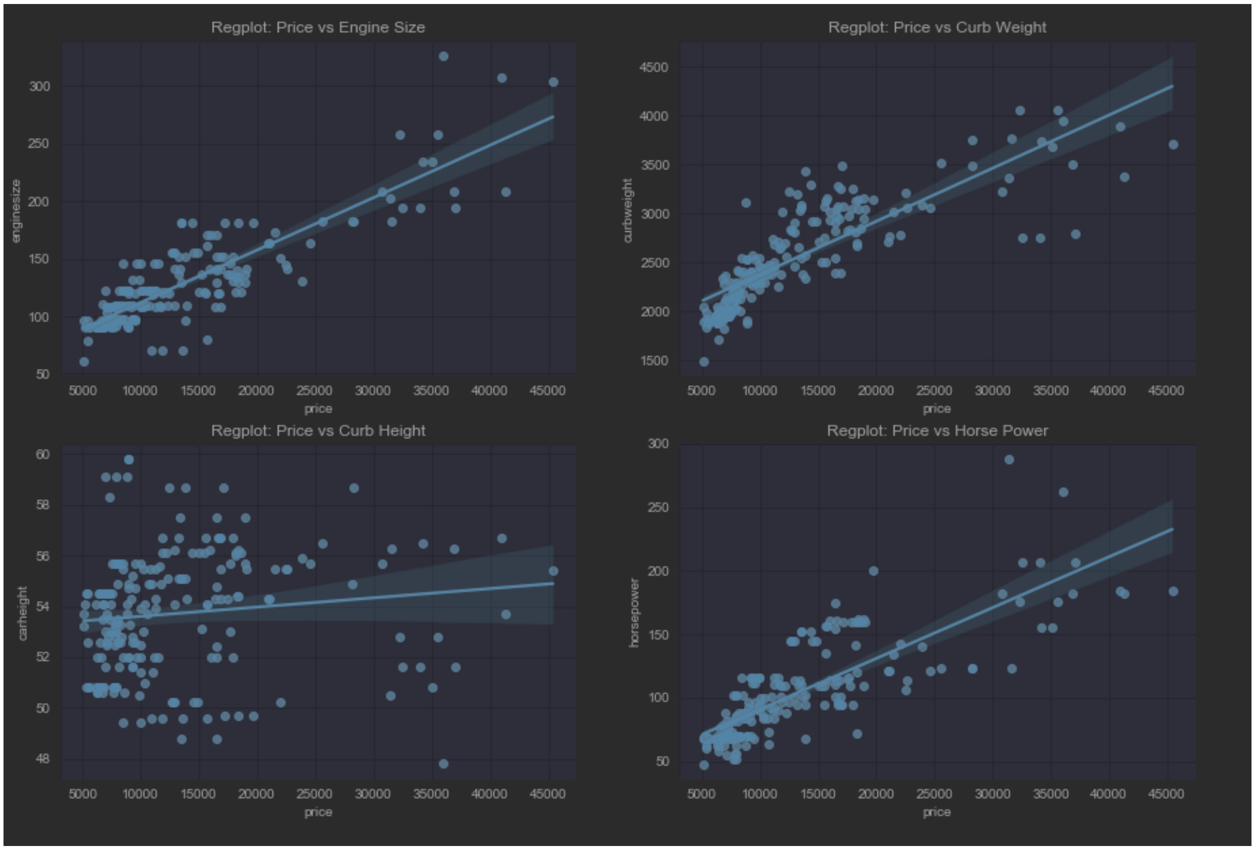
As seen above, the logarithmic transformation did seem to make the distribution more "normal," while the boxplot indicated the transformation reigned in all the potential outliers. For this reason, I decided I would keep this transformed target column in my back pocket to use during model training to see if it had any impact of predictive capabilities.

In previous projects, I have also found a benefit in reviewing a correlation matrix between all the features and target variable. In order to create one, I utilized Seaborn's handy <heatmap> function, passing in the Pearson's r coefficient calculated using Pandas' built-in <df.corr()> function.



I also sorted the values on the price target column so I could more easily see how each feature related to it. Unsurprisingly, I found that engine size, weight, and horsepower were positively correlated with the selling price. Interestingly, I found that both features measuring efficiency (city/highway mpg) were negatively correlated with the price target. I had thought that more expensive vehicles had higher efficiencies, though I concluded that this was a result of the data being captured in the 80’s as opposed to today where people are more concerned with their emission footprint.

I wanted to visualize how some of these stronger correlations looked like when plotted against the target price column. To do so, I utilized the <regplot> function from Seaborn.



Regplot: Price vs Engine Size

Regplot: Price vs Curb Weight

Regplot: Price vs Curb Height

Regplot: Price vs Horse Power

Outside of the curb height feature, the three others appeared to have a strong, linear relationship with the target making them great candidates for training a linear regression model.

ANALYSIS

Now that I had a much better understanding of the data, it was time to begin training my machine learning models. Prior to running the data through the model, I split it using the handy <train_test_split> function provided by Sci-kit Learn. I used a test size of 20% and set a random state so that the results would be re-producible in the future.

The linear model object that I used for this project was provided by Sci-kit Learn. After fitting the training data to the model, I used two metrics to analyze the performance. The mean-squared error, which calculates the average error between the predicted and measured values, was utilized as it would indicate how much the price estimation was off on average. I also analyzed the model's coefficient of determination, or r-squared (r2) value, which calculates the proportion of variance that can be explained by the dependent variable (i.e., price) to the total variance of the data.

The linear model resulted in an MSE of approximately 3580 and an r2 value of 0.84. While these metrics aren't poor by any means, if you were buying a used car and your prediction was off by more than even a thousand bucks, it would not have been very helpful in the first place. As such, I felt that I could improve upon the predictive power by training a few other machine learning models. In addition to the simple linear model, I opted to train a decision tree, k-Neighbors, and random forest regressor.

TRAINING RESULTS	MSE	R-squared
Decision Tree	2506	0.92
k-Neighbors (n=1)	3384	0.85
Random Forest	1860	0.96

As can be seen from the table above, the random forest regressor seemed to produce the best results in terms of the selected metrics. While $1860 is still quite a bit of error in the context of buying a used vehicle, I felt much better about this predictive model.

CONCLUSIONS

After training four predictive models and measuring their respective performance metrics, I found that the random forest regressor model performed the best. As such, I would have selected this trained model for any sort of third-party application that I would ship to customers so that they could use it before going to buy a vehicle from a dealership or auto shop.

ASSUMPTIONS

I feel that one of the largest assumptions that I made during this project was the fact that the prices for the vehicles measured in 1987 would be relevant in today's economy. A hundred bucks back then equates to about $260 with inflation (InflationTool, n.d.). Perhaps it would have been beneficial to scale the price points of the vehicle using this adjustment. However, it would also be important to understand how the prices of vehicles have trended over time, and that could be the topic of an entirely different time-series data science project.

LIMITATIONS/ CHALLENGES

The greatest limitation of this project is the limited number of observations in the dataset. Having only a little over 200 rows probably does not generalize very well to the global state of vehicle sales prices. Outside of not having a good amount of data to train the model, it is also difficult to validate the data by holding out only 20% (approximately 40 records) of the dataset.

FUTURE USES/ADDITIONAL APPLICATIONS

I feel that the models that were trained on this data could certainly be used as a basis for future similar projects that attempt to predict the sales price of vehicles. The models would certainly benefit from additional data points, especially from within the last decade or so. Also, there is the potential for transfer learning. Once the model has been adequately trained using more vehicle data, it could perhaps be tweaked and pivoted to predict the prices of other means of transportation such as motorcycles or perhaps even airplanes.

RECOMMENDATIONS

As stated in previous sections, I would recommend that more data points be collected and cleaned to run through the machine learning models. The small data set certainly limited the potential predictive power, and it would be interesting to see how much the performance metrics could be improved upon. There are also likely better models that could be pulled in to be trained using this data. I chose the four that I used during this project as they are the most commonly used methods to work with this type of data. It might also be worth it to reach out to a subject matter expert in the auto industry field to get a better understanding of the features. They could perhaps shine light on which features are more important than others, whereas I simply used the correlation matrix to calculate numbers for me.

IMPLEMENTATION PLAN

Once I felt comfortable with the performance of the predictive model, I would most likely stitch together a simple GUI application that would allow customers to input values for the vehicle characteristics that are measured in this dataset for the vehicle that they want to buy. The GUI app could be written using Python, though it might be useful to use a language better suited for this task such as Java or C#. I could also create a web-application that allows users to perform similar predictions from a website as opposed to a desktop application.

ETHICAL ASSESSMENT

I do not feel that there were too many ethical considerations that needed to be made during this project. The data contained no personally identifiable information (PII), and there was not a lot of underlying bias that needed to be dealt with. I would say that since the predictive model still had a decent amount of error, it could cause customers to go to dealerships with incorrect assumptions on how much they should pay for a certain vehicle which could certainly lead to a dissatisfied customer base.

AUDIENCE QUESTIONS

1. Where were the vehicles in the dataset located?
2. Why didn’t you handle the outliers?
3. Why did you use Seaborn as opposed to other packages like Plotly?
4. Why did you select the coefficient of determination for a performance metric?
5. Why did you use all the features in the dataset?
6. You didn’t train the models using the logarithmic target variable you created... Why is that?
7. If you were going to add more data, how would you go about it?
8. Are you happy with the final findings of this project?
9. Could this predictive model be used in other countries?
10. Do you plan on doing similar projects in the future?

REFERENCES

InflationTool. (n.d.). Value of 1987 US Dollars today. Retrieved from https://www.inflationtool.com/us-dollar/1987-to-present-value

Matplotlib. (n.d.) Matplotlib 3.6.2 documentation. Retrieved from https://matplotlib.org/stable/index.html

NumPy. (2002). NumPy documentation. Retrieved from https://numpy.org/doc/

Pandas. (2022). Pandas documentation. Retrieved from https://pandas.pydata.org/docs/

Seaborn. (2022). Seaborn: statistical data visualization. Retrieved from https://seaborn.pydata.org/

UCI ML Repository. (n.d.). University of California Irvine’s Machine Learning Repository. Retrieved from https://archive.ics.uci.edu/ml/index.php