# Market Research using Machine Learning Techniques for a New Car Company Entering The Space

Author: Gregory Hovesen


Ryerson University

ID: 500769246
Section: D10
Supervisor: Dr. Kachana Padmanabhan
December 3rd, 2021

# Table of Contents



## Abstract


A new automotive company is deciding to enter the market with their new range of products. They are implementing data science tools to understand the current market. The goal of the analysis is to understand the leading factors that go into the price of a vehicle, which in turn, will help with the future price generation of their products to remain competitive, or exceed the current market. As you can assume, many factors go into the pricing of a vehicle. Parameters such as weight, capacity, engine type, and many other factors can be said to influence the final sales price of each unit. To accomplish this analysis, multiple linear regression will be used to create a model. This will give insight regarding each contributing factor. This analysis will be performed on previous market data, however, this can be replicated with a forever changing/expanding dataset to provide more accurate, to-date modeling. This analysis will be implemented through various softwares. These softwares include Numpy, Seaborn, Scikit, and Pandas, using the Python Language. These will be used for data-cleaning of the datasets,

exploratory analysis to understand the factors of the dataset, and perform regressionional analysis. After the completion of the analysis, matplotlib will be used to visualize the data in a more viewer friendly manner to present the analysis' findings. This in turn will help the company in making future decisions.

The main dataset used in this analysis can be found on Kaggle at:

> Pouyaaskari. (2021, January 15). *Automobile-dataset*. Kaggle. Retrieved September 27, 2021, from https://www.kaggle.com/pouyaaskari/automobile-dataset.

To conclude, this project will be able to determine a price prediction model for future products. What attributes have the strongest relationship with the dependent variable, price? And finally, will be able to determine whether or not the company's products are at a competitive rate compared to industry averages.

**Literature Review**

As a new car company trying to enter a very competitive market, loads of market research is required to have a successful entry. Relying on market data from various public datasets regarding vehicle attributes, we can determine the main factors that contribute to the pricing of a vehicle. In our datasets, attributes such as weight, dimensions, capacity and engine size, to name a few, are all contributing factors to our model. This literature review examines the data science methods required to follow through with creating a pricing model. Along with researching the methods being used, various articles that reveal more regarding the pricing of a vehicle will be examined to cross reference our results. As this form of market research has been completed before, this literature review will also highlight the similarities and differences between this project and the others.

According to D'Allegro (2021), mileage and condition are the key factors into pricing a used car. This information helps us determine whether or not information in our dataset coincides with real world instances of car pricing. As condition is a very subjective attribute, a more nominal representation of a vehicle's condition can be solely based on mileage as the vehicle's main asset is the engine. Higher mileage, lower selling price is the relation in question. Along with this, D'Allegro (2021) mentions that even though there are many tangible attributes regarding car pricing, some soft attributes may hold a higher weight regarding a minivan versus a sports car. For example, a manual transmission is more desirable in comparison to an automatic transmission for the sports car market than the minivan market. Finally, an attribute such as colour may have greater weight for one buyer compared to the other. This can be crucial to the

cars price as more common car colours will sell faster than others which will come with a higher price tag than those that sit.

The way we are able to determine all of the factors that go into the pricing of a vehicle, many data science techniques will be used. Foremost, a multiple linear regression model will be created based off of the data. With that being said, this process has been carried out many times regarding a price prediction model in any industry. An example for this would be in the paper by Y. E. Cakra and B. Distiawan Trisedya (2015) regarding the prediction of stock prices using linear regression models. As mentioned by Y. E. Cakra and B. Distiawan Trisedya (2015), stock pricing models have to be continuously updated to create accurate predictions, which is also the case for car pricing. Although car pricing is not up to the second like stock pricing, everytime a vehicle is sold, the market value changes, requiring us to reconsider the current model. Another valid pull from this article is the fact that this paper is using linear regression based on sentiment analysis. Should our dataset have intangible values such as colour, we will have to consider them in our model as it was stated by Y. E. Cakra and B. Distiawan Trisedya (2015).

As stated in the article by Dubin, R.A. (1998), regression analysis was used to predict housing prices. The statistical technique used for this model was the ordinary least squares. This paper is particularly useful for this project as, like other regression models, many attributes such as house type, location, and number of rooms, to name a few were used; however, the model mentions it is missing a large quantity of information. This being the correlation between the prices of neighbouring houses. This is directly related to our dataset regarding missing market information. Although this article is from 1998, the process of model creation remains the same.

In the article "Support vector regression analysis for price prediction in a car leasing application.", Listiani, M. (2009) mentions the fact that many factors go into the pricing of a vehicle. This meaning there is a large amount of independent variables making up the dependent value, car price. As there are so many variables, the only way we can create an accurate model is through machine learning. Listiani, M. (2009), states that “a good model is required to have a good performance on future or previously unseen data”, which is exactly what we are trying to accomplish throughout this project. Listiani, M. (2009) proceeds to mention two key faults with regression models. Under-fitting, and Over-fitting. These two problems have the same effect on the model as they both render the model less accurate. When a model is overfit, the model is too heavily reliant on the training data and is poor at accurately predicting new values. On the other hand, under-fitting models is the result of the model being too simple. This does not allow for all attributes to be accounted for in the model.

In the article “OLD CAR PRICE PREDICTION WITH MACHINE LEARNING”, Gajera, P., Gondaliya, A., & Kavathiya (2021) compare five different models of car price prediction. The result of this paper is that using the Random Forest Regression model was the most efficient model at 93.11% success rate. This is particularly impressive as the comparison is against KNN-Regression, Decision-Tree Regression, XGBoost Regression and finally Linear Regression. As linear regression directly relates to my project, I look forward to creating a more accurate model than Gajera, P., Gondaliya, A., & Kavathiya (2021), with their success rate of 76.46%. As mentioned by D’Allegro (2021), mileage has a huge role in pricing a vehicle, and this was also the case for the article in question.  As Gajera, P., Gondaliya, A., & Kavathiya (2021) used a dataset of over 92000 records, their model may be very intricate, which may help them with their accuracy. On the other hand,

Finally, to understand the implementation of regression models, Sci-Kit Learn is the Python package in question. As mentioned in the article “Scikit-learn: Machine learning in Python”, Sci-Kit is a Python package with many machine learning algorithms at its disposal and is directed at the non-specialist user for medium scale problems. Sci-kit takes pride in its interactive nature to make machine learning more accessible for the Python community. The backend processes of this package are Numpy and Scipy. As mentioned, the majority of the package is underwritten python, however there are some C++ libraries that are also used in its implementation. Sci-kit learn also has a website with all of its documentation and real examples to assist with your problems at hand. As this will be used in my analysis, this information is very pertinent and useful.

The datasets in question have similar independent variables such as transmission and engine type. These being categorical values along with other numerical attributes like mileage. We will use the required packages from Sci-Kit Learn, in conjunction with data cleaning techniques to improve our data. In conclusion, as this project has been implemented in the past, the aim of this project is to improve the accuracy of our model in comparison to the others. Along with other papers such as stock price prediction, we can use this new found information to compare and contrast the methods used by others previously.

Below is the link to the project Git Repository:

https://github.com/GHovCIND820/CIND820-Final-Project

**Data Description and Exploratory Data Analysis**

The data used for this project is the Automobile_data.csv dataset. The variable in question for this project is the price variable, which resides in the final column of the dataset. There are many techniques you can use to understand more about your data. Using the pandas package, we are able to quickly understand the shape, column names, Non-Null Count, and finally Dtype(int64, float64, object, etc). This is all possible by running the .info() method. The results are as follows;

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):

| # | Column | Non-Null Count | Dtype |
|---|---|---|---|
| 0 | symboling | 205 non-null | int64 |
| 1 | normalized-losses | 205 non-null | object |
| 2 | make | 205 non-null | object |
| 3 | fuel-type | 205 non-null | object |
| 4 | aspiration | 205 non-null | object |
| 5 | num-of-doors | 205 non-null | object |
| 6 | body-style | 205 non-null | object |
| 7 | drive-wheels | 205 non-null | object |
| 8 | engine-location | 205 non-null | object |
| 9 | wheel-base | 205 non-null | float64 |
| 10 | length | 205 non-null | float64 |
| 11 | width | 205 non-null | float64 |
| 12 | height | 205 non-null | float64 |
| 13 | curb-weight | 205 non-null | int64 |

```
 14  engine-type        205 non-null    object
 15  num-of-cylinders   205 non-null    object
 16  engine-size        205 non-null    int64
 17  fuel-system        205 non-null    object
 18  bore               205 non-null    object
 19  stroke             205 non-null    object
 20  compression-ratio  205 non-null    float64
 21  horsepower         205 non-null    object
 22  peak-rpm           205 non-null    object
 23  city-mpg           205 non-null    int64
 24  highway-mpg        205 non-null    int64
 25  price              205 non-null    object
dtypes: float64(5), int64(5), object(16)
memory usage: 41.8+ KB
```

As you can see, the dataset has 205 entries, with 26 attributes attached. This shows us that there are 16 object attributes, and 10 numeric attributes. When looking further at this information, I immediately noticed the price attribute was in fact an object rather than a numeric attribute which led me to believe there was quite a lot of data cleaning required. This dataset denoted missing values as "?", rather than NaN. This was apparent in the "normalized-losses" column. The way this was handled was by converting the dtype to numeric and finally replacing the "?" with the column mean. Methods such as .to_numeric(), .fillna() and .mean() were all used to complete this step of the cleaning. In the initial cleaning, I converted all the object attributes to numeric values by explicitly mapping the values. I first checked the unique values in the columns to understand how many "code" numbers there were to assign. For example, the "fuel-type" attribute had two unique values; gas or diesel. This was then converted and replace by gas:1 and

diesel:2. This process was replicated on the following columns; “aspiration”, “num-of-doors”, “body-style”, “drive-wheels”, “engine-location”, “engine-type”, “num-of-cylinders”, and “fuel-system”. Finally, the attributes “bore”, “stroke”, “horsepower”, “peak-rpm”, and “price” were all converted to numerical values. This process now converted the whole datatset to numeric attributes which allows us to begin the creation of the first Linear Regression Model. To complete the data cleaning, a new csv file was created named “Cleaned_Data.csv” using the pandas .to_csv() method.

**Initial Results**

The dataset contained many object attributes which had to be handled accordingly. A few attributes such as normalized-losses contained some missing values. This however was denoted from the source using a “?”. This was handled by converting the appropriate cells to NaN values and then <Na> values prior to being assigned the columns average. Other columns such as num-of-cylinders were denoted by the english spelling of the numbers, hence were changed from “three” to 3. To complete this data cleaning, the dataset was split between the object attributes and the integer/float attributes. After the cleaning, the dataset was concatenated and Cleaned_Car_Data.csv was created. Looking at the data through some visualizations and methods, I could find that the dataset’s average car price was just over 13,000 dollars. The upper and lower bounds were 45,400 and 5,000 dollars respectively. We can conclude that the majority of car prices are below the dataset’s mean by referring to the mean and median. We can conclude that the dataset has a positively asymmetric distribution. When deciphering the correlation matrix heatmap, we can determine some of the driving factors in the dependent variable that is price. Engine-Size, Curb-Weight and Horsepower are the three attributes with the greatest

positive correlation to the price. Whereas Highway-Mpg and City-Mpg have the greatest negative correlation. These findings can help with the optimization of the initial model created. This model took every numerical value into account. This included all the attributes that were converted as well. The initial model resulted in the following accuracy and statistics.

Mean Absolute Error: 2929.7148881358653

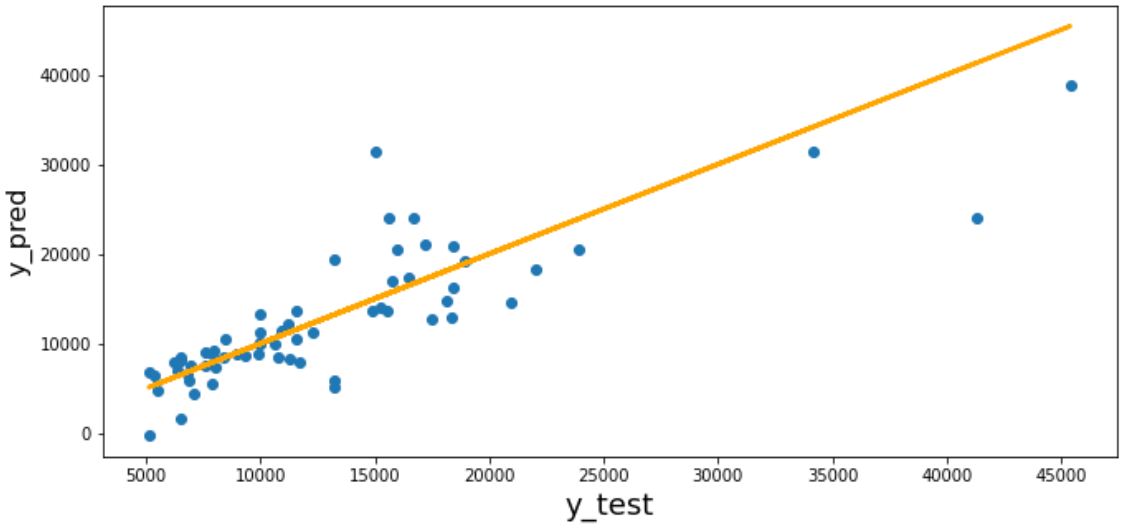
Mean Squared Error: 19681481.204782143

Root Mean Squared Error: 4436.381544094483

R Squared Score:  0.6724444782976456

Explained Variance Score:  0.6770069282135331

With this being said based on the mean price of 13207.13 and the Root Mean Squared Error of 4436.38, we can conclude that the model has an accuracy of predicting the cars price roughly two thirds of the time. This will have to be optimized for greater accuracy to render this model valid. This hopefully can be somewhat accomplished by the omittance of irrelevant attributes in the model to reduce clutter. Other manipulations to the dataset such as using dummy variables for the cars "make", may help improve accuracy. Below is a visualization of the initial model;



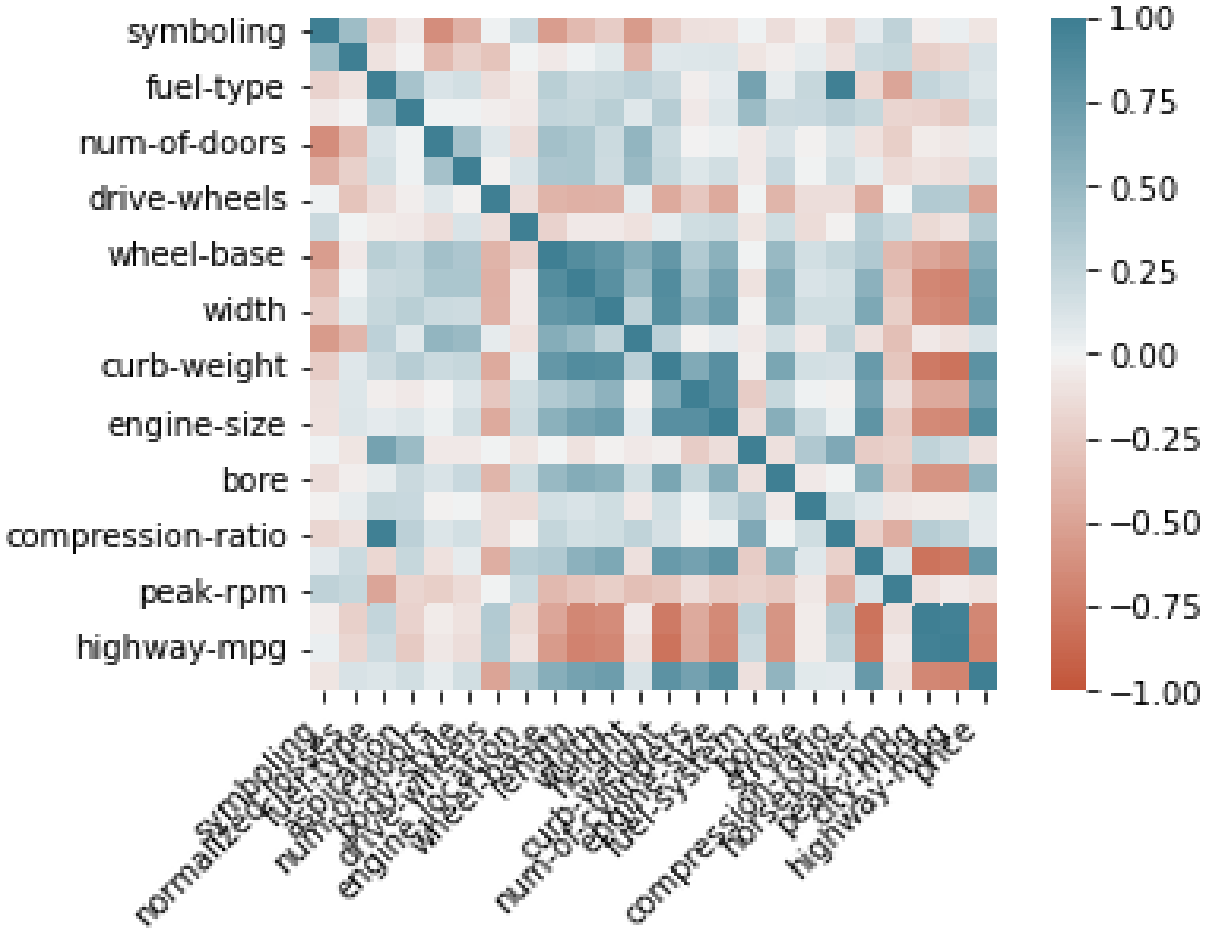
**Visualizations of Initial Results**

When creating a Multiple Linear Regression model, we wish to first understand information such as mean, standard deviation, upper and lower quartile of the dataset by using the .describe() function. Most importantly, the information regarding the target variable are as follows;

| | |
|---|---|
| count | 205.000000 |
| mean | 13207.129353 |
| std | 7868.768212 |
| min | 5118.000000 |
| 25% | 7788.000000 |
| 50% | 10595.000000 |
| 75% | 16500.000000 |
| max | 45400.000000 |

Name: price, dtype: float64

Next, to understand the correlation between attributes, a heatmap was created. To achieve this, the seaborn package was implemented by using the .heatmap() method. This map is below;




By referring to the scale on the right of the map, you can see the darker the square gets, the higher the correlation coefficient, whether it be positive or negative. To simplify this, we can solely output the correlation coefficients of the independent variables to the dependent price variable. This is done through using the .corr() method. This result is shown below;

symboling -0.082201

normalized-losses 0.133999

fuel-type 0.110207

aspiration 0.177285

num-of-doors 0.057180

body-style 0.178642

drive-wheels -0.490291

engine-location 0.331013

wheel-base 0.583168

length 0.682986

width 0.728699

height 0.134388

curb-weight 0.820825

num-of-cylinders 0.687770

engine-size 0.861752

fuel-system -0.115521

bore 0.532300

stroke 0.082095

compression-ratio 0.070990

horsepower 0.757917

peak-rpm -0.100854

city-mpg -0.667449

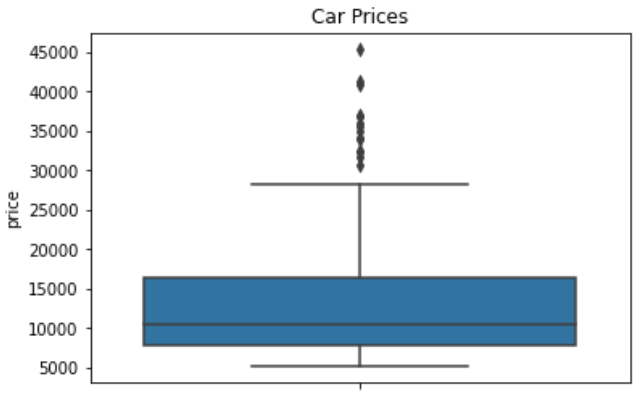
highway-mpg -0.690526

price 1.000000

Name: price, dtype: float64

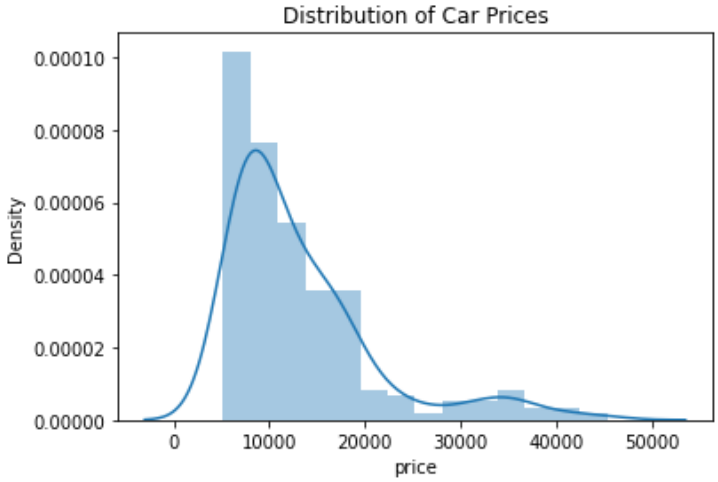
Moving forward, a box plot is created to visualize the car prices to better understand where the bulk of the data lies. This is done through seaborn's .boxplot() function. This result is shown below;

Next, visualizations of the attributes that have the highest correlation coefficients from the above output were plotted against the price variable.

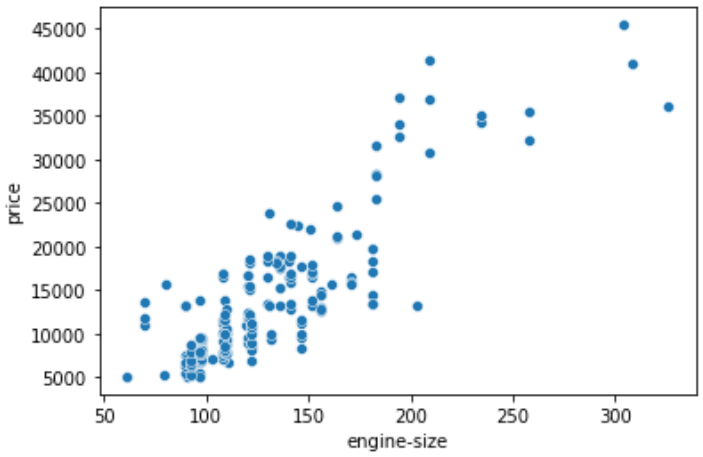



As mentioned previously, you can see the majority of the prices lie between $8000 and $17500.

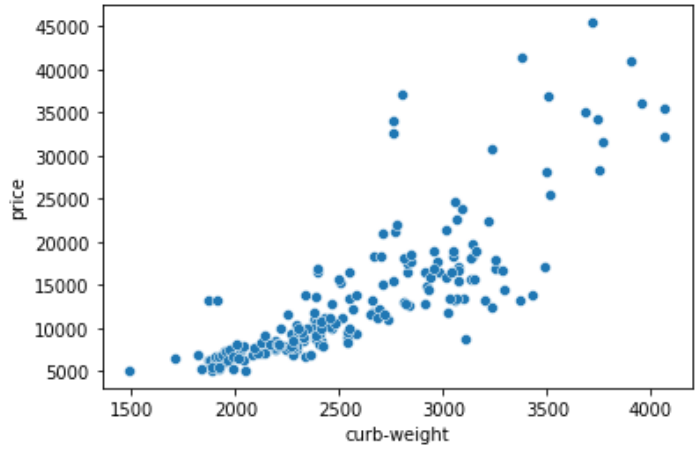
To fully understand the dependent variable, a distribution plot is created using the seaborn .distplot() function resulting the the following output;

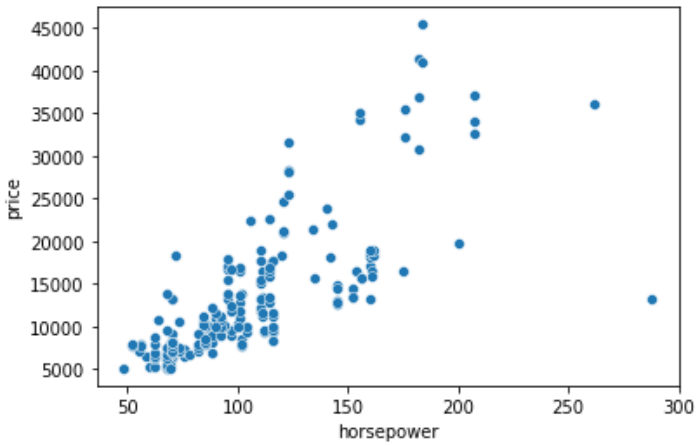


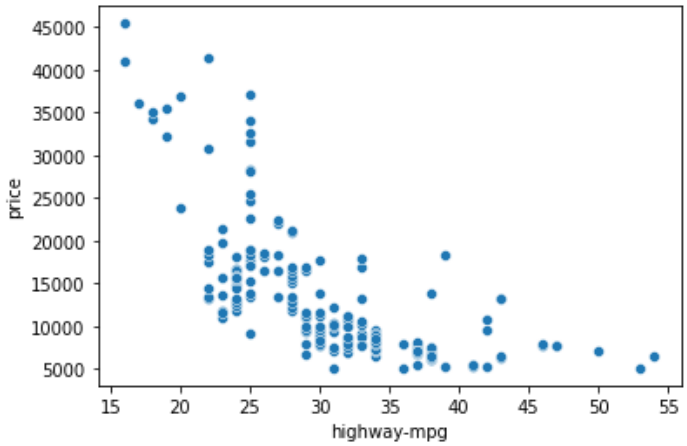

Finally, from referencing the correlation plot, we can graph the relationship between the highest correlated variables to the target attribute. The first plot will be done by the seaborn .scatterplot() function with the x variable being “engine-size” and the y variable, of course, being “price”. The output is below;

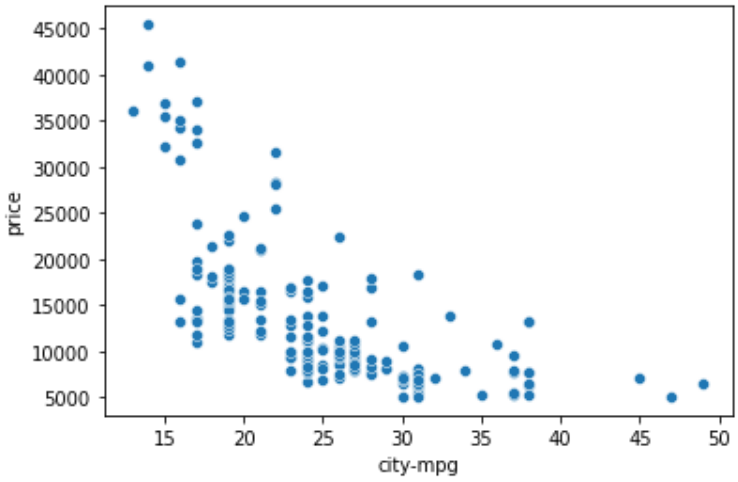



As the correlation coefficient of 0.86 suggests, the relationship between engine size and price is positive. This means that the larger the engine, the higher the price. This same process will be implemented on the following attributes; “curb-weight”, “horsepower”, “highway-mpg”,and “city-mpg”. The results are as follow;






As displayed in the plots above, the curb-weight and horsepower variables also have a positive correlation with the car price. Finally, the highway and city miles per gallon attributes have a negative correlation with car price.

**Model Improvements**

With the lack of normalization/or the lack of codes and/or dummy variables, this model may have become too explicit. In this section, we look at various tweaks that were made to create the most accurate model possible with these techniques and dataset.

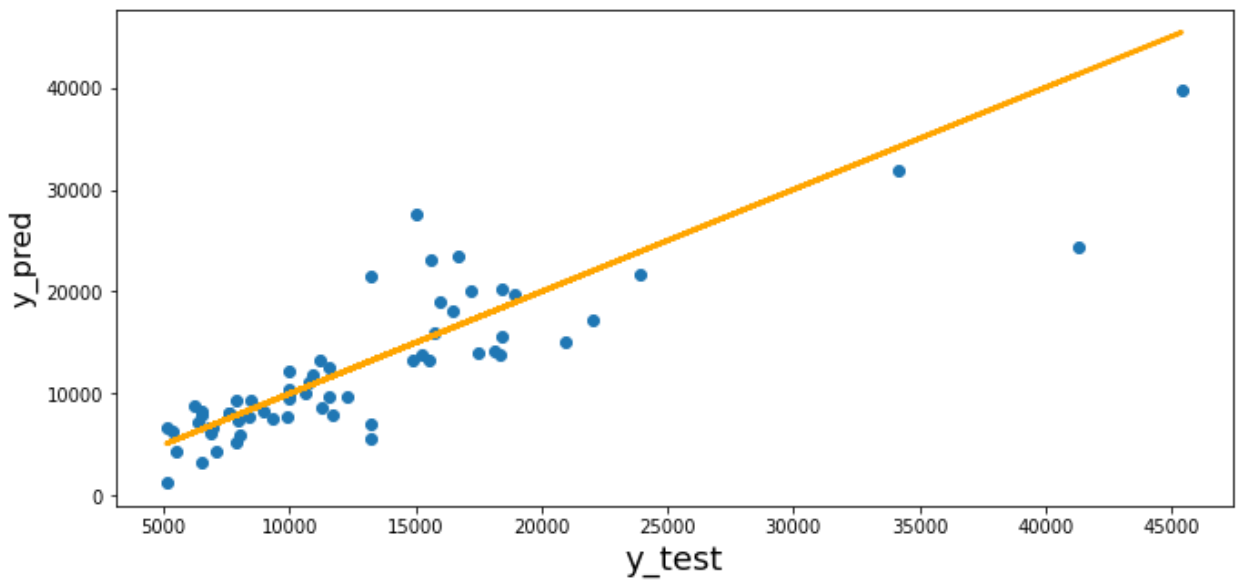
**Model 2;**

With the second attempt at creating a model, the dataset was reverted back to its original form and the data cleaning process was restarted, however different techniques were used. When handling numeric values, the same method was applied, however, rather than explicitly mapping the categorical variables, I used categorical codes instead. For example, the first implementation of this method was done with the "fuel-type" attribute. The first step is to convert the object variable into a categorical variable using the .astype() method. Then create a new column in the dataset, which in this case was named "fuel-type-code", which has the value of the cat.code of the "fuel-type" variable. This cat.code implementation was used on the rest of the object variables in the dataset. This second model resulted in the following accuracy and statistics.

Mean Absolute Error: 2761.4515413827135
Mean Squared Error: 16448590.227534316
Root Mean Squared Error: 4055.686159891359
R Squared Score:  0.7262489292757532
Explained Variance Score:  0.7333128779324034

With the change in approach to the data cleaning, the model has been successfully improved. This success is documented through the R Squared Score. The initial model resulted in an $R^2$ value of 0.672, whereas Model 2 resulted in an $R^2$ value of 0.726. This is a good step forward in improving the model however, a more accurate model is possible as shown next.

Below is a graphical representation of the accuracy of Model 2;




**Model 3;**

After many more iterations of data cleaning, the third model consists of a mixture of the previous two implementations along with some scaling of certain attributes. The processes used were the same as documented above, however, the "drive-wheels" attribute was handled using the dummy variable approach, and the rest of the object variables were transformed using cat.codes. The way the dataset was normalized was through sklearn's MnMaxScaler package. The most success was found when every attribute was transformed using this package. This third model resulted in the following accuracy and statistics.

Mean Absolute Error: 2658.4179500122423

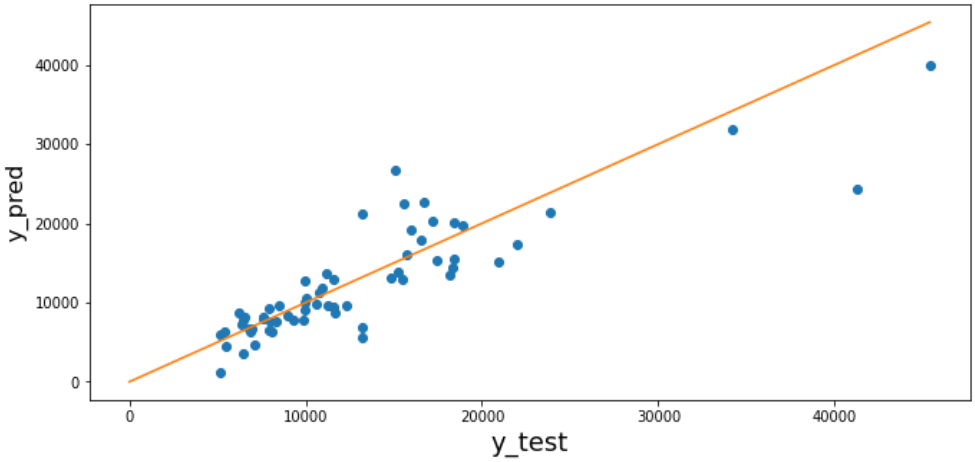
Mean Squared Error: 15362685.307905115

Root Mean Squared Error: 3919.526158594316

R Squared Score:  0.7443214589175706

Explained Variance Score:  0.7506434129780915

With this model having an accuracy of 74%, this is close to a desirable level of accuracy, however one final model was created. Below is a graphical representation of the model.




**Final Model;**

The final model of this project resulted in the following accuracy and statistics.

Mean Absolute Error: 0.2663366808764814

Mean Squared Error: 0.19654533129512172

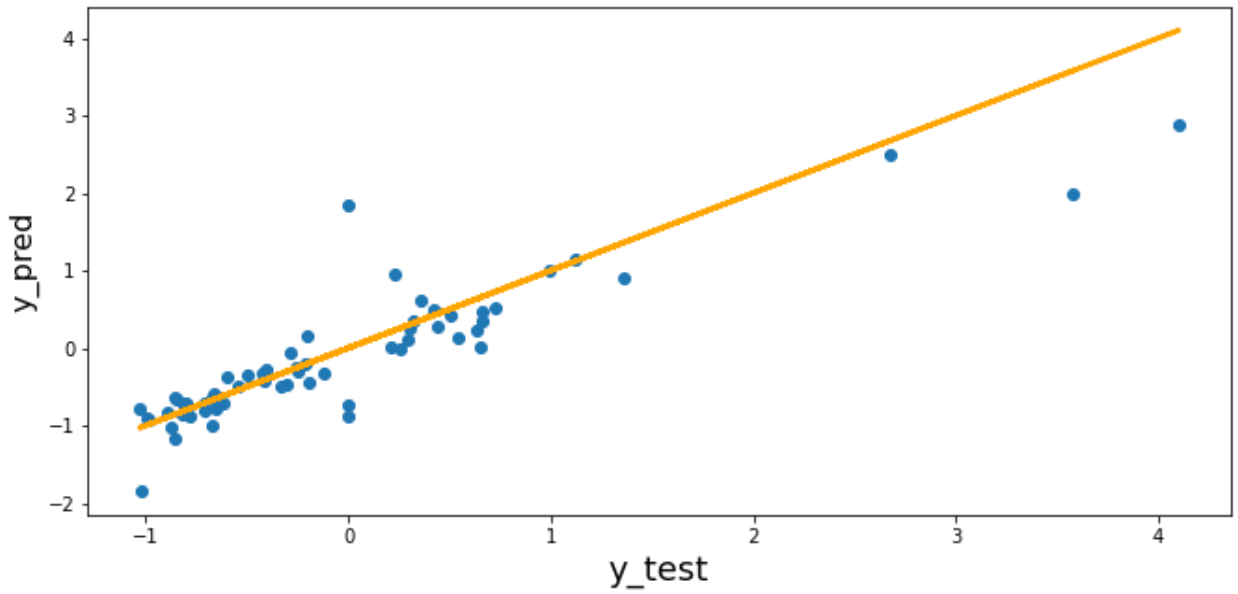
Root Mean Squared Error: 0.44333433353973584

R Squared Score: 0.7984514450107454

Explained Variance Score: 0.8067557546099543

The accuracy of almost 80% is the maximum that was obtainable during this research. This model went through many tests and trials from combining previous model ideology. This result was obtained by performing the same data cleaning processes to deal with the object data which could be directly converted to numeric values. The difference now was the creation of a dummy variable function to run on the categorical variables. This function used Pandas get_dummies() method and implemented on the following columns; “symboling”, “make”, “fuel-type”, “aspiration”, “body-style”, “drive-wheels”, “engine-location”, “engine-type” and “fuel-system”. This function also removed the original column in the dataset. After the creation

of these dummy variables, the dataset was scaled through SciKit's StandardScaler package. The columns that were scaled are all the numeric columns that did not get converted to dummies. After completing these steps, the final dataset was created through the .to_csv() method, and was named "Dummy_Auto.csv". The final model's scatter plot is displayed below;



**Conclusions**

This project's objective was to perform market research for a new automotive company wishing to enter the market. The foal was to create a Linear Regression Model to predict car prices based off of the dataset used. Along with this, we wanted to understand the main contributing factors to the price. This was accomplished through exploratory data analysis, data cleaning, and regression model creation. All of the work was completed through Google Colab's Jupyter Notebooks. The Python Language was used to implement Scikit Learn packages as well as Pandas, Numpy, Seaborn and Matplotlib. All of these packages were required to create Lineare Regression Models, operate and manipulate datasets, and visualize data.

This project analysis resulted in the creation of a Linear Regression model of 80% accuracy in predicting the test set of the dataset, with a 97% accuracy in the training set. This was obtained through multiple iterations of data cleaning, attribute engineering and regression model building. This result came after four models that resulted in the accuracy going up gradually. There were other models created, however, these others had worse accuracy, therefore were not included in the report. The final model incorporated all the attributes in the dataset as this was the most successful, in comparison to some that were created with attributes omitted.

When looking at the most important attributes contributing to vehicle price, we can conclude that the engine size has the greatest impact. This is a confirmation from our prior research conducted during the literature review, regarding some of the leading factors. Along with engine size, the curb weight of the vehicle is the second leading contributor. Horsepower and width of the vehicle are also highly correlated to the price. This all makes sense as the larger the vehicle, the larger engine with high horsepower is required to move the vehicle, which in

turn, results in a higher price. These are all positively correlated attributes. This leads to the point that highway miles per gallon, as well as city miles per gallon are very much correlated to the price, however, negatively. This also helps to circle back with the positively correlated attributes, as the large weight, and large engine size results in poor fuel mileage. We can conclude that the larger the vehicle, the larger the engine required, resulting in poor fuel mileage, which then ultimately results in higher vehicle pricing.

For future research regarding the topic, I believe that a larger dataset with the same attributes would be beneficial. This could help with all aspects of the project. As materials and labour continue to increase in price, this will ultimately result in higher vehicle prices. To handle this, the model should continuously be tweaked and re-created with new data when available. This will continue to render this methodology valid and accurate, however if not updated, may render this obsolete.

**References:**


D'Allegro, J. (2021, September 13). *Just what factors into the value of your used car?* Investopedia. Retrieved October 18, 2021, from https://www.investopedia.com/articles/investing/090314/just-what-factors-value-your-used-car.asp.

Dubin, R.A. Predicting House Prices Using Multiple Listings Data. *The Journal of Real Estate Finance and Economics* **17,** 35–59 (1998). https://doi-org.ezproxy.lib.ryerson.ca/10.1023/A:1007751112669

Gajera, P., Gondaliya, A., & Kavathiya, J. OLD CAR PRICE PREDICTION WITH MACHINE LEARNING.

Listiani, M. (2009). Support vector regression analysis for price prediction in a car leasing application. *Unpublished. https://www. ifis. uni-luebeck. de/~ moeller/publist-sts-pw-andm/source/papers/2009/list09. pdf*.

Nelli, F., & Safari, an O'Reilly Media Company. (2015). *Python data analytics: Data analysis and science using pandas, matplotlib, and the python programming language* (1st ed.). Apress.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.

Scikit-learn: Machine Learning in Python, Pedregosa *et al.*(2011), JMLR 12, pp. 2825-2830

Venkatasubbu, P., & Ganesh, M. (2019). Used Cars Price Prediction using Supervised Learning Techniques. *International Journal of Engineering and Advanced Technology (IJEAT)*, *9*(1S3).

Wright, D. B., & London, K. (2009). *Modern regression techniques using R*. SAGE Publications Ltd https://www-doi-org.ezproxy.lib.ryerson.ca/10.4135/9780857024497

Y. E. Cakra and B. Distiawan Trisedya, "Stock price prediction using linear regression based on sentiment analysis," *2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2015, pp. 147-154, doi: 10.1109/ICACSIS.2015.7415179.