Car Price Prediction
Multiple Linear Regression using R and Python

Liezel Dela Cruz, ldelacruz@bellarmine.edu
Eli C. Dalton, edalton2@bellarmine.edu

ABSTRACT

In this project we obtained a dataset based on car prices. We then performed multiple linear regression and exploratory data analysis using Python and R. The regression analysis that was performed created a model that could be used to predict results, given specific input criteria. The value we want to predict is the car price, given specific criteria: ID, levy, manufacturer, model, production year, category, leather interior, fuel type, engine volume, mileage, cylinders, gear box type, drive wheels, doors, wheel, color, and airbags. The validity of this model will be measured by calculating the mean squared error and the r-square value. It will also be assessed through two experiments. Overall, the model created using python gave a much larger mean square error than the model created using R.

I. INTRODUCTION

There is a large variety of car manufacturers and car models that exist in the world. Factors such as manufacturing materials, fuel type, and production year are some elements that can affect car price. This dataset, extracted from Kaggle, contains information about the price, model, and car features with respect to each car ID. We selected this dataset to come up with a prediction model for these variables, in attempt to draw conclusions about the pricing of cars worldwide. For this project, we will be attempting to create a multiple linear regression model using python and R, which will vary depending on what the inputs are.

II. BACKGROUND

A. Data Set Description

The dataset was extracted from Kaggle and was downloaded as a csv file that contains 19237 rows and 18 columns. We chose to work with this dataset because we wanted to use one that predicted continuous variables like price. An important feature of the data is that it was collected by a car contractor who was interested to know about whether you could predict the price of cars given certain information.

B. Machine Learning Model

Multiple regression is like linear regression, but with more than one independent value, meaning that we try to predict a value based on two or more variables. In this case, we are using 17 variables to predict 1 variable, the price of a car. The model requires that the dataset is split into two - the independent and dependent variable. The model works by calculating the coefficients of each variable, as well as the y intercept. As such, when numerical values are plugged into each variable within the formula, we can predict the dependent variable (the y value - price).

III. EXPLORATORY ANALYSIS

This dataset contains 19237 row samples with 18 columns of various data types and some missing values. A complete listing is shown below in **Table 1**.

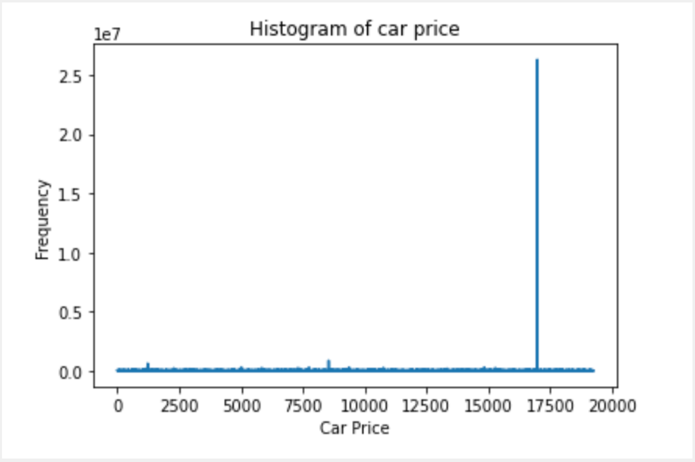
Table 1: Data Types

Variable Name	*Data Type*

ID	int64
Price	int64
Levy	object
Manufacturer	object
Model	object
Prod. Year	int64
Category	object
Leather interior	object
Fuel type	object
Engine Volume	object
Mileage	object
Cylinders	float64
Gear Box Type	object
Drive Wheels	object
Doors	object
Wheel	object
Color	object
Airbags	int64

It is also important to note that there are some missing variables within this dataset. Namely within the levy column, where null values are denoted as ’ - ‘. In addition, Figure 1 displays the distribution of car prices in the dataset. Here, we can see that the graph is unimodal and left-skewed, with the majority of cars being priced around 17,000.

Figure 1: Histogram of Car Prices



IV. METHODS

Now, we will discuss how we prepared the data for our model and how we performed multiple experiments using different parameters for the model.

A. Data Preparation

To prepare the data for analysis in RStudio, a few changes had to be made. A couple of columns were dropped because their data was not consistent which was causing issues in trying to run the analysis. The milage column had both text and the milage number within the column, so the text was removed in order for it to work properly. The biggest change was that I had to shorten the dataset because it kept crashing my system when I tried to perform certain operations.

To prepare the dataset using python, the levy column was dropped in order to provide a smoother operation and improve accuracy. Since it had a lot of missing values and the column did not serve such a huge significance for the analysis, it was removed. The ID column was also removed because it was deemed unnecessary for the prediction model.

B. Experimental Design

Table X: Experiment Parameters

Experiment Number	Parameters
	Experiments in Python:
1	All features with 80/10/10 split for train, validate, and test
2	All features with 70/15/15 split for train, validate, and test
	Experiments in R:
1	All features with 80/20 split for train and test
2	All features with 70/30 split for train and test

C. *Tools Used*

The following tools were used for this analysis: R running in R studio environment. Within R studio different built in tools were used, specifically, Tidyverse, catools, and ggplot were primarily used for different aspects of the analysis. Tidyverse is an R programming package that helps to transform and better present the data; it is particularly useful in the area of data science. Catools is another R package that has some built in statistical functions that are very functional for this type of analysis. Ggplot is yet another R package that as its name suggests is very useful for plotting data. It has many built in plot functions making plotting the data in different ways very simple.

Python running in Anaconda environment was also used for analyzing this data, particularly Pandas, matplotlib, seaborn, sciketlearn were employed to perform most of the operations. Pandas is a freely available Python package that is heavily used for data analysis and machine learning. Matplotlib is a python library that is used for data visualization in Python as its built in functions make the process much easier. Seaborn is another Python data visualization library that is actually based on matplotlib, but creates a higher-level interface leading to more finished looking statistical graphs. Finally, sciketlearn is a Python data analysis library that is considered one of the best to use for machine learning.

V. RESULTS

A. *Mean square Error and R-Square calculation*

The mean squared error (MSE), is a statistical measure of the amount of error in a statistical model. It uses the average squared difference between the observed and predicted values to assess the level of error. The closer this measure is to zero the better. The R-Square calculation is a goodness-of-fit measure for linear regression models. This measure indicates the percentage of the variance in the dependent variable that the independent variables explain collectively, so the higher the better for this percentage. More explicitly, an r-square value less than 50% is not a good model and a value less than 30% is a poor model. Also, it should be noted that due to the size and number of levels within each variable, it is not practical nor is it useful information to try and list out the entire regression formula. The MSE and R-Square measures will be able to clearly indicate the effectiveness of the models.

Experiments in Python:

1. In this first experiment the MSE was 678875402849463694799142912.00. This is a really large value, possibly due to the fact that we are dealing with a very large dataset. The R-Square was -13508650448385436.00.
2. In the second experiment, the MSE was 348711870764657908314537984.00. The R-Square was 18673828497784.21.

Experiments in R:

1. In this first experiment the MSE was 131,416,298.81. While this is understandably going to be higher because we were working with so much data, it is still very hight which is not great. The R-Square value for this experiment was 69.69%. This indicates that the model is okay. It is nearing that 50% threshold so I would hesitate to call it good.
2. In this second experiment the MSE was 133,973,343.86. As expected this increased compared to the first experiment. This was expected because less data was being used to train the model. However, the R-Square value for this experiment was 70.68% slightly better than in R experiment #1.

B. *Discussion of Results*

Experiments in Python: In the Python experiments, the second experiment was better since the MSE was the lowest out of the two. However, both models do not seem to be good at predicting car prices due to its really high MSE and the R-Square value. The negative R-Square value in the first experiment indicates that the model is predicting worse than the mean of the target values. Hence, the first experiment was not good.

Experiments in R: In the R experiments the results mostly followed what was expected. While both experiments produced okay models, this first experiment produced the better of the two. This made sense because more of the data was used to train the model in the first experiment so you would think it would be more accurate. However, while the MSE was lower in the first experiment the R-Square value was actually slightly better in the second. This was surprising, but the difference was so small that it is likely attributable to the randomness of the data selected for training and testing. Overall, the first model was best due to more data being used to train it.

C. Problems Encountered

Issues that were faced whilst obtaining the data was finding a dataset that predicted continuous variables like price. Initially, we were going to use a classification dataset which would not have been suitable for a project that is based on regression analysis. Whilst preparing the data, it was difficult to find missing values on Python due to some empty cells being filled with '-'. However, this was fixed by just removing the entire column since it did not serve as much purpose as the other columns.

Moreover, as referenced in an earlier section a notable problem occurred when trying to perform the analysis in R. The system kept crashing. Different attempts were made to remedy the problem, but it was ultimately determined that this issue was arising due to the size of the dataset and a couple of columns that contained inconsistent data. So for the R analysis the dataset had to be shortened in length, and the abnormal columns were dropped.

D. Limitations of Implementation

The biggest limitation of our model are the variables that were used in the dataset. Arguably there could be other variables that have a greater impact on the price of a car than any of the variables in our dataset. Therefore, models using other such variables could be more accurate in their modeling and prediction.

E. Improvements/Future Work

To improve our model in future work, there are several things that could be done. More experiments could definitely be performed to see if a better model could be reached. A particular future course of action that would be very interesting to see would be add and/or remove variables. This could allow the model to become narrowed down to the most essential variables, possibly leading to a better model that could be very accurate at predicting the price. It would also be interesting to find a different dataset that seeks to predict the same thing, and then see it is a better model than the one we used.

VI. CONCLUSION

In this project we sought to create a model to predict the price of a car, given several input variables.

Our results from the R experiments indicated that the more data used to train a model the more accurate it will be. The results from the python experiments displayed the same. While okay models were created in both R experiments, they could use some more work to become better and more accurate models.

REFERENCES

https://stackoverflow.com/
https://statisticsbyjim.com/