Shine Bright like an Affordable Diamond
Multiple Linear Regression using R and Python

Bailey Korfhage, bkorfhage2@bellarmine.edu
Matthew Carrico, mcarrico2@bellarmine.edu

ABSTRACT

The task of this project was to analyze a data set using R studio and Python and complete a multiple linear regression (MLR) model. The data set we chose required a variety of continuous and discrete variables that could potentially predict another variable. We used R studio and Python to explore the data, create the MLR model, and check its accuracy.

I. INTRODUCTION

Team members obtained the data set from Kaggle.com, where the user obtained it from Australian Diamond Importers Prices. The data set contained values including carat, cut, color, clarity, depth, table, price, length, and width. Using MLR, we chose to try and predict the price from the other factors given in the data set.

II. BACKGROUND

A. Data Set Description

This data set came from the Australian Diamond Importers Prices which is updated to Kaggle.com annually. This document explores a dataset containing prices and attributes for approximately 54,000 round-cut diamonds. As an age group who has hopes of obtaining potential engagement rings, team members thought it would be interesting to investigate the variation of prices and the attributes.

B. Machine Learning Model

The multiple linear regression model is an approach for data synthesis using mathematical figures. It tries to use multiple variables at the same time to predict another variable. The model uses the squares of the residuals to evaluate the equation of the linear regression and the R-squared value, which informs the user about the predictability of the equation. The closer to 100%, the better the model can correctly predict the actual data.

III. EXPLORATORY ANALYSIS

This data set contains 53,940 samples with 9 columns with both continuous and discrete data types. None of the values of the dataset contained missing values. Due to this, plotting of the variables were not necessary to investigate.

Table 1: Data Types

Variable Name	*Data Type*
carat	continuous
cut	discrete
color	discrete
clarity	discrete
depth	continuous
table	continuous
price	continuous
length	continuous
width	continuous

IV. METHODS

A. Data Preparation

The original data set had 10 columns. The length, width, and depth columns were originally labeled as x, y, and z. According to Kaggle.com, these columns in context were length, width, and depth of the diamond in millimeters. Also, the dataset had an extra depth column, so the team members decided to delete one (the one that was not the original 'z' column).

B. Experimental Design

Note: The validation set for R studio was made manually by taking 5 entries from the data set and making a new file.

Table X: Experiment Parameters

Experiment Number	Parameters
1-R	80/20 split for train and test – 5 values for validate
2-R	90/10 split for train and test – 5 values for validate
3-R	60/40 split for train and test – 5 values for validate
1-P	80/10/10 split for train, test, validate
2-P	70/15/15 split for train, test, validate
3-P	90/5/5 split for train, test, validate

C. Tools Used

The following tools were used for this analysis:

- R libraries: tidyverse, catools
    - Tidyverse allowed team members to import the dataset into R studio for EDA and MLR analysis.
    - Catools allowed the dataset to be split into training and testing sets so that the data could be trained for the MLR model and then tested to investigate strength.
- Python libraries: pandas, sklearn, numpy
    - Pandas allowed team members to import the data set and to convert the categorical variables into a numeric pattern.
    - Sklearn was the major tool used for MLR. It split the data into the training, testing, and validation set. It also set up the model for linear regression and helped with built in calculations for MSE and R-squared.
    - Numpy was used for stacking the validation results.

V. RESULTS

A. Mean square Error and R-Square calculation

Explain the MSE and R square and discuss the results using relevant formulas.

Mean Square Error (MSE) is a calculation that tells you how close the regression line is to a set of points. The formula is calculated by taking the difference between the actual value and the predicted value, squaring it, and taking the average of all those squares. Because this dataset is so large and the values are spread so wide, the MSE is very big. The R-squared value determines the variance in the dependent variable, but overall is another determination of fit. For the dataset, the R-squared value was very high, 92%, indicating that the regression was a good model for the dataset.

B. Discussion of Results

All models provided the same R-squared value of 92% for both R studio and Python. However, the MSE different depending on the train and test split. The best MSE was with the split of 90% for training the model, 5% for testing the data, and 5% for validation. I think that this was the best model because the variation in diamond prices are so wide, it allowed the training to absorb the dataset more as a whole.

C. Problems Encountered

The dataset did not have a separate set for validating the model. So, team members were required to take entries from the original set and add it to a new file. This may not have been the most efficient way to create a random validation set, but it proved to be effective for the use in R studio..

D. Limitations of Implementation

The R-squared value indicates that this model is good for the dataset as it is higher than other investigations. However, the MSE indicates that there is much variation in the differences between actual and prediction, which is not preferable. Team members believe that the validation set comparison is most likely the best measure, as that is the whole point for creating the model is to apply it. The validation set comparison is very close, so we believe this is a good model.

E. Improvements/Future Work

We think that the table variable in the dataset is not as relevant as the others. Exclusion of this might be an improvement on the model. All other variables are commonly heard in reference to diamond valuation. We believe this dataset might have also been too large and varied, which created the large MSE.

VI. CONCLUSION

Overall, the model and the effort behind it proved to be very successful. Despite the large MSE, the R-squared and the application on the validation set proved that the model was fitting. The dataset was not prepared as the team members wanted, but they were able to collaborate to clean it up for use. We used R studio and Python to explore, split, and train the data for the model. Each program had three different experiments with different splits for training and testing. These experiments varied in results, but all had the same R-squared, indicating consistently good fit.