Project 4 Regression Analysis

Inesh Chakrabarti, Lawrence Liu, Nathan Wei

Introduction

For the first part of this project we will do regression analysis. The dataset we chose to use is one of diamond characteristics. We will conduct regressions to predict the price of a diamond given some features.

Dataset

Let us begin by understanding the dataset. The dataset consists of information about 53940 round-cut diamonds with ten features:

Feature	Description
`carat`	weight of the diamond (0.2–5.01)
`cut`	quality of the cut (Fair, Good, Very Good, Premium, Ideal)
`color`	diamond colour, from J (worst) to D (best)
`clarity`	a measurement of how clear the diamond is (I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best))
`x`	length in mm (0–10.74)
`y`	width in mm (0–58.9)
`z`	depth in mm (0–31.8)
`depth`	total depth percentage
`table`	width of top of diamond relative to widest point (43-95)
`price`	price in US dollars ($326-$18,823)

Question 1.1

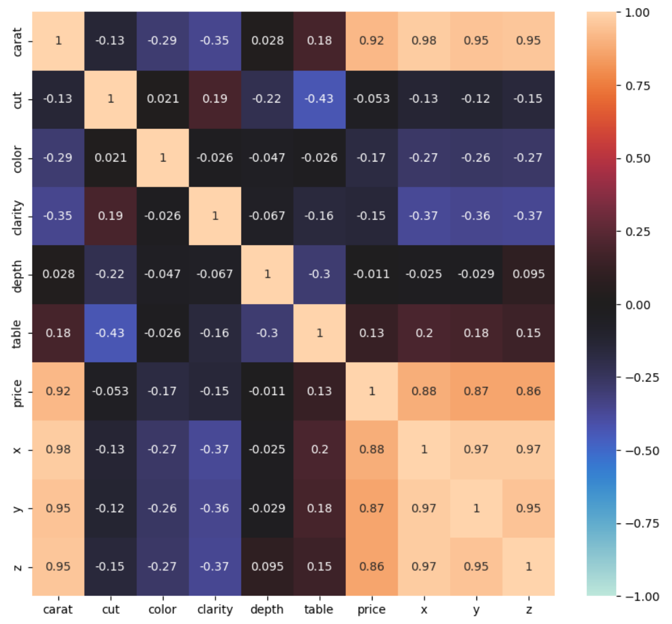


Figure 1: Feature Pearson Correlation Heatmap

We will be using the nine features to predict the `price`. We can begin by computing the Pearson correlation matrix heatmap for these features in the dataset in Figure 1. Note that a pearson correlation coefficient r is defined as such:

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \tag{1}$$

We have assigned quantitative values to the qualitative labels `cut`,`color`,`clarity` as ascending natural numbers based on ideality.

Feature	Correlation
`carat`	0.9215914337868304
`cut`	-0.05349263851362828
`color`	-0.1725093772499559
`clarity`	-0.14680175361025616
`depth`	-0.010647725608533299
`table`	0.12713358133531918
`x`	0.8844357793744166
`y`	0.865421694764742
`z`	0.861250266123968

(a) Price

Feature	Correlation
`carat`	0.7694571626172851
`cut`	0.00542011950342582
`color`	-0.011980043670033661
`clarity`	0.04512538515850012
`depth`	-0.03572374489729493
`table`	0.08458507638109278
`x`	0.7873455524189906
`y`	0.7717301198408058
`z`	0.7655421629234554

(b) Price per Carart

Table 1: Pearson Correlation Coefficients

The values for correlation for `price` is given in Table 1(a). We see that, unsurprisingly, there is a massive collection of high correlation squares at the bottom right. These indicate high Pearson correlation coefficient between `price` and `x`, `y`, `z`. Similarily, there is also a high correlation with `carat`. All of these suggest that the size of the diamond itself is the most significant predictor as to its price.

However, unexpectadly, we had a negative perason coeffciient for the quality of the `cut`, `color`, and `clarity`. This was odd, so we similarily calculated the Pearson correlation values for Price per Carat instead, given in Table 1(b). We observed that the coefficient for `cut` and `clarity` became slightly positive, while color became close to zero. These seemed more in line with our expectations,and confirmed that the rarity of high carat and high clarity in a diamond at the same time was making the corresponding values in Table 1(a) negative.

Question 1.2

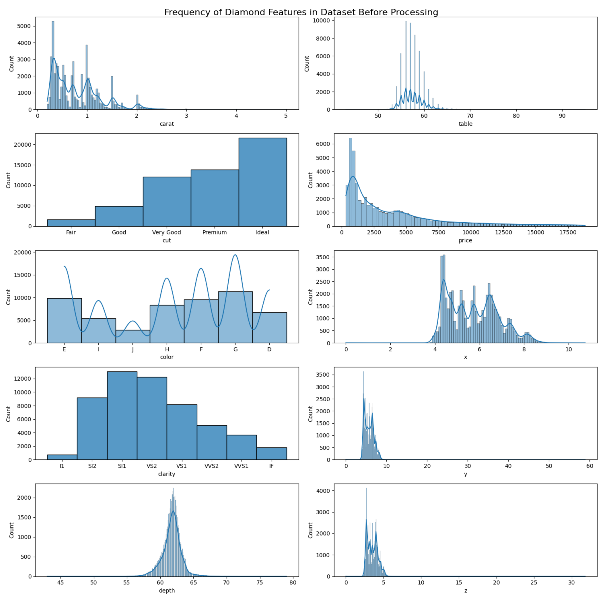
Feature	Skewness
`carat`	1.116645920812613
`depth`	-0.08229402630189467
`table`	0.7968958486695427
`x`	0.3786763426463927
`y`	2.4341667164885554
`z`	1.5224225590685583

(a) Before Processing

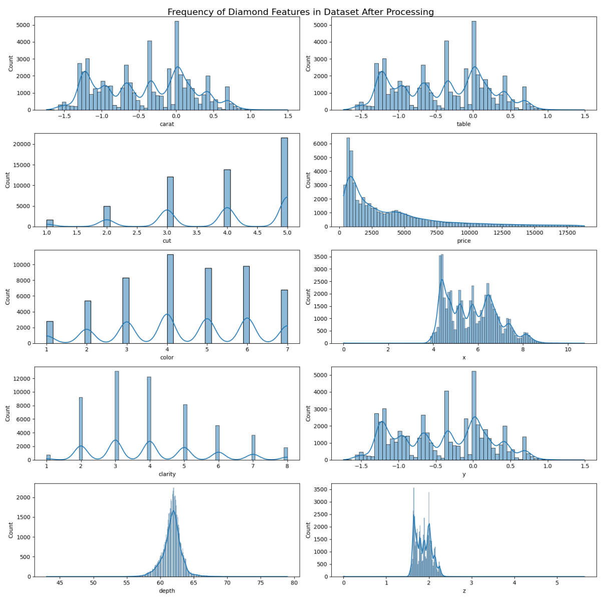
Feature	Method	Skewness
`carat`	Box Cox	0.020450070764268666
`depth`	No Change	-0.08229402630189467
`table`	Box Cox	0.020450070764268666
`x`	No Change	0.3786763426463927
`y`	Box Cox	0.020450070764268666
`z`	Square Root	0.0139742634447758

(b) After Processing

Table 2: Skewness of Features



(a) Before Processing



(b) After Processing

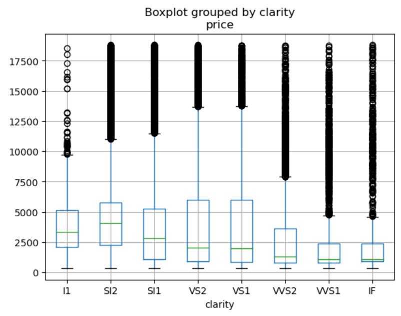
Figure 2: Histogram and KDE for Features

Now, we examine the frequency distributions within each feature. We find that some of our numerical features are somewhat skewed—apparent from Table 2(a) and Figure 2(a). We target a skewness less than 0.5, as this would imply that the distribution is somewhat symetric. As such, we try three different methods: square-root, logrithm, and box-cox transformation. The first two are somewhat self-explanatory; box-cox uses a non-zero value of λ and conducts the following transformation:

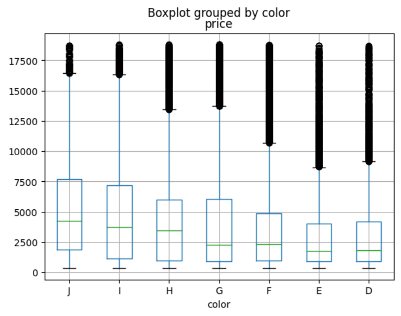
$$y'_{\lambda} = \frac{y^{\lambda} - 1}{\lambda \cdot \bar{g}_y^{\lambda - 1}} \tag{2}$$

where $\bar{g}_y$ is defined as the geometric mean of y.
We next try all of these methods, along with no transformatino, and take the minimum to try to minimize skewness in our data. The results are shown in Table 2(b) and Figure 2(b).

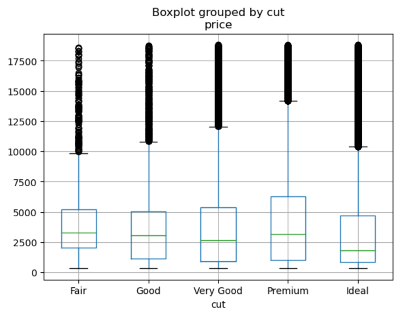
Question 1.3



(a) Clarity



(b) Color



(c) Cut

Figure 3: Categorical feature vs price boxplots

We see that idek what the fuck we see bro??

Question 2.1

Now, before we train our regression models, we begin by splitting our data into training and testing sets. As such, we must now standardize the feature columns. The function used to do this, `scaledTrainTest()`, as well as `scaledTrainTestSplit()` can be found in `utils.py`.

Question 2.2

Next, we begin with feature selection. We note that some of the features may not be useful or may cause overfitting in our models as they do not carry useful useful information about the variable that we are trying to predict. To tell whether this is the case, we use two metrics: Mutual Information (MI) and F score.

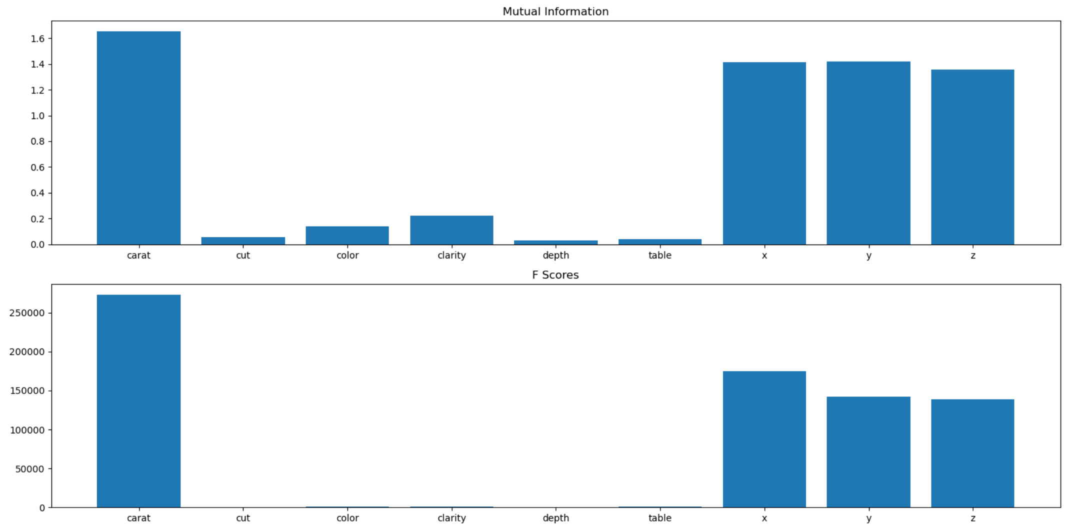


Figure 4: Bar graph of F score and Mutual Information for features

Feature	Mutual Information	F Score
`carat`	1.64589286853222835	273144
`cut`	0.058351547792578895	139
`color`	0.13968295427702282	1465
`clarity`	0.21672877051000627	1079
`depth`	0.030486803506200033	5.074
`table`	0.03818752327438668	802
`x`	1.4058664238563194	174973
`y`	1.4172107199595354	142130
`z`	1.3569258463570115	138947

Table 3: Mutual Information and F Score Values for features

It is clear by observation from Figure 3 and Table 2 that the lowest mutual information is present in `depth` and `table`. It is also important to note that the mutual information in `cut` is also very small. This makes sense as we saw earlier that the Pearson correlation coefficient for `cut` with respect to price per carat was almost zero. Similarily, the Pearson correlation coefficients for depth and table were very close to zero.

From this point, we will be testing the regression models without these three features, and with these three features and comparing the performance to determine the general performance.

Regression Models

Question 3

For this entire section we perform 10-fold cross validation and then measure the average RSME error for the training and validation sets. For the random forest model we also measure "Out-of-Bag Error" and R^2 score.
"Out-of-Bag Error" refers to

Linear Regression

Question 4.1

We begin with a simple regression, least square linear regression. Begin by noting the optimization problem:

$$\underset{\theta}{\operatorname{argmin}} \frac{1}{2}||\mathbf{Y} - \theta^T \hat{\mathbf{X}}||^2 \tag{3}$$

Taking the derivative with respect to θ and setting it equal to zero gives us

$$\theta = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} \mathbf{X}^T\mathbf{Y} \tag{4}$$

Now, we can add regularization terms, beginning with the simple L1 regularization, also known in this case as a Lasso regression. This would make our new optimization problem:

$$\underset{\theta}{\operatorname{argmin}} \frac{1}{2}||\mathbf{Y} - \theta^T \hat{\mathbf{X}}||^2 + \alpha||\theta|| \tag{5}$$

Note that in this case, our learned values for θ would become somewhat sparse as L1 regularization linearly penalizes any non-zero parameters. As such, the gradient for the regularization term does not change until the parameter is zero, which leads to sparsity. However, in this case, as there aren't many terms, we would potenntially only have a few zero'd parameters in our predictive polynomial.

With L2 regularization, also known as a Ridge regression, we would have:

$$\underset{\theta}{\operatorname{argmin}} \frac{1}{2}||\mathbf{Y} - \theta^T \hat{\mathbf{X}}||^2 + \lambda||\theta||^2 \tag{6}$$

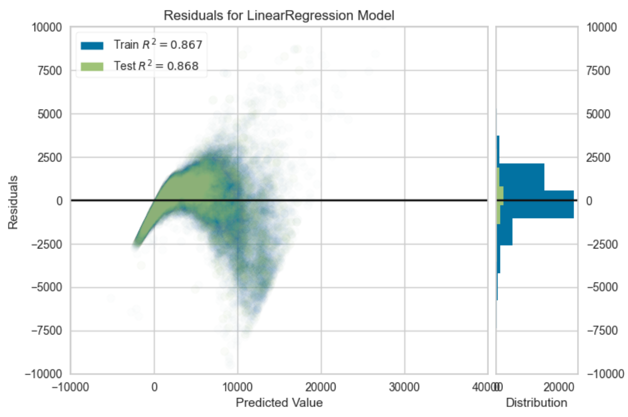
Note in this case, we wouldn't have parameters that are zero; rather, we would have extremely small parameters. This is because the gradient for L2 regularization decreases rapidly as the parameter itself becomes smaller and smaller due to the parabolic nature of the term.

Question 4.2

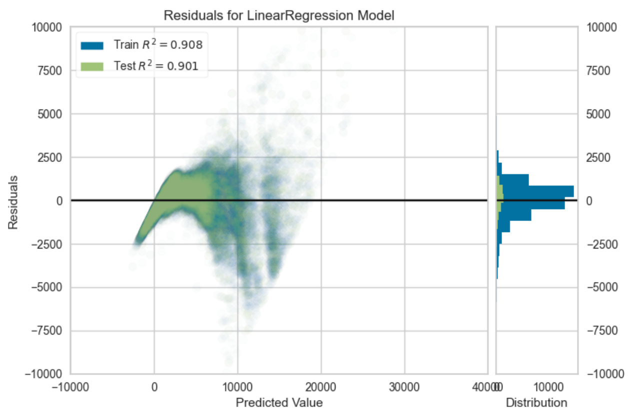
Model	Description	Train RSME	Test RSME
1	Features processed	1453.1935502136905	1459.5109959451215
2	Features unprocessed	1216.495767837737	1217.7987618616985
3	Features processed using $log($`price`$)$	958.0855791176566	957.5523367504156
4	Features unprocessed using $log($`price`$)$	1215.167267164955	1292.934288205552

Table 4: Ordinary Least Squares Regression Model

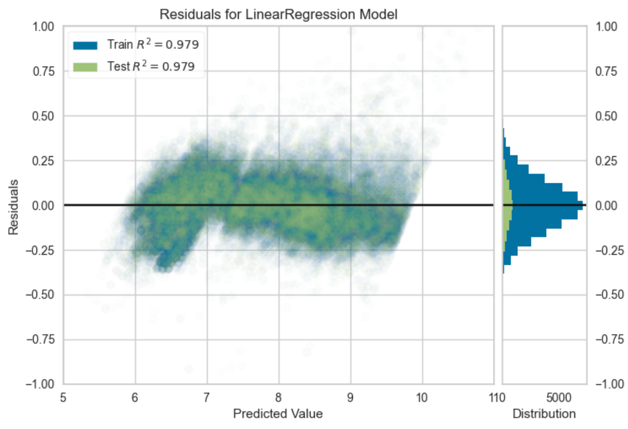
Beginning with ordinary least squares regression, I first varied the number of features available to the model based on mutual information and quickly found that keeping all the features led to the best regression model. This makes sense due to the nation of a linear regression—if a feature isn't very relevant, its associated parameter will be very small—and so for Ridge and Lasso, we also kept all of the features.



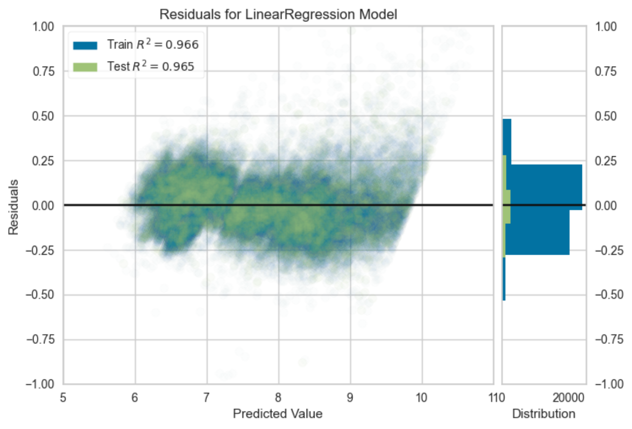
(a) Model 1 Residuals



(b) Model 2 Residuals



(c) Model 3 *log* Residual



(d) Model 4 *log* Residuals

Figure 5: Residual Plots

We then tested the linear regression model with processed data (deskewed as in Question 1.2) and unprocessed data. We noted that the model without processed features seemed to perform much better as shown in Table 4 and visualized in Figure 5. Note that Model 2's residual histogram is distributed far more sysmtetrically than Model 1, implying that the linear regression fit better in this case. However, we see in both Figures 5(a) and 5(b) that Models 1 and 2 predict many of the prices to be negative.

As such, we decided to test the model by taking the natural logrithm of `price` before fitting the regression. Note in Table 4 that with unprocessed Features, this model performed about the same, but with processed features the RSME decreased very significantly. We also see that our new *log* residual plots seem to follow a linear trend, which makes sense as this implies that the error is increasing as the price of the diamond itself increases. We also note that unlike earlier, Model 3 has normally distributed residuals while Model 4 does not. This is interesting, because this is the opposite of Models 1 and 2; that is, this time, the model with deskewed data has the more normal distribution of residuals.

Next, we test the Lasso Model by iterating through alpha values 10^k such that $k \in \mathbb{Z}$ and $-4 \geq k \leq 1$. We also try both the $log(\texttt{price})$ model, and try both with and without deskewing.

Next we test the Ridge model with all the same parameters are the Lasso model, except we also test whether or not the

Question 4.3

Question 4.4

Polynomial Regression

Neural Network

Random Forest

LightGBM, CatBoost, and Bayesian Optimization

Project 4 Twitter Data

Inesh Chakrabarti, Lawrence Liu, Nathan Wei