Project 4

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

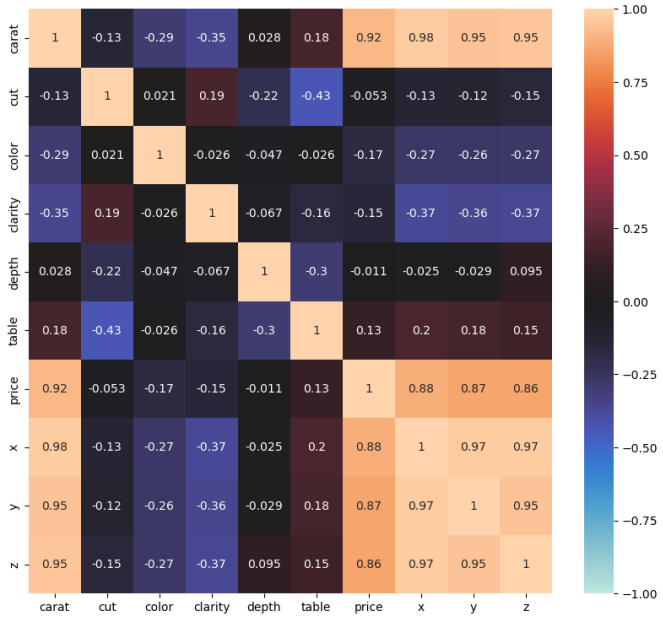


Figure 1: Feature Pearson Correlation Heatmap

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. Begin by noting that we 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`, textttz. 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

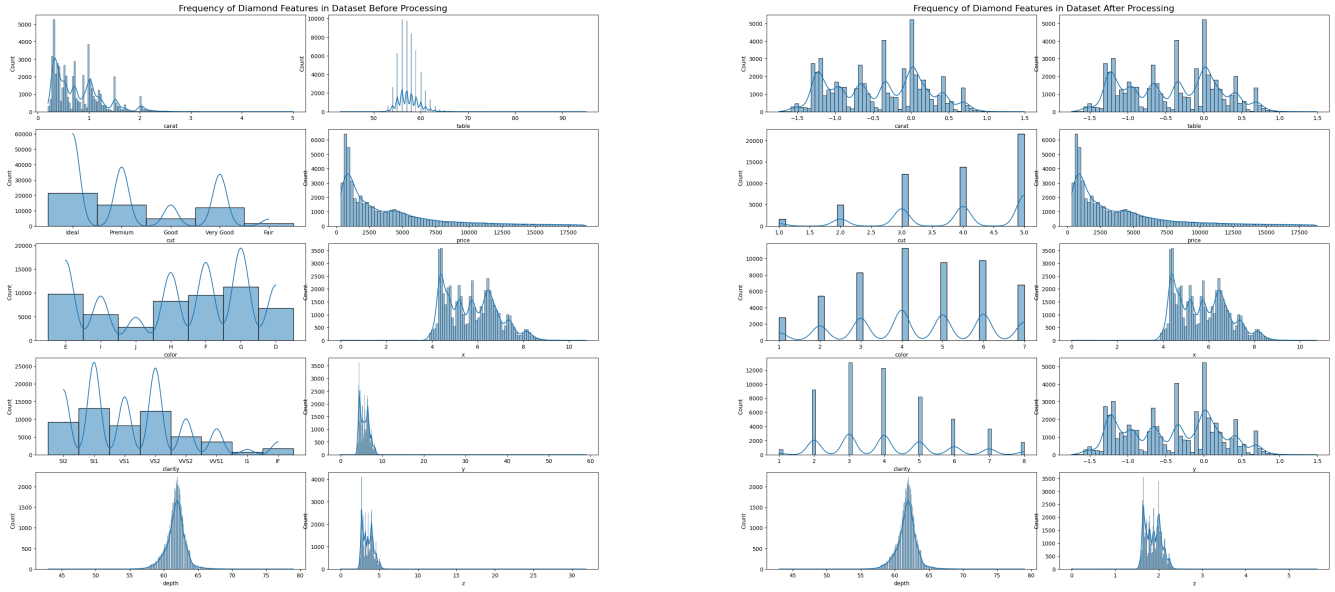
Feature	Skewness
carat	1.116645920812613
depth	-0.08229402630189467
table	0.7968958486695427
x	0.3786763426463927
y	2.4341667164885554
z	1.5224225590685583

(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

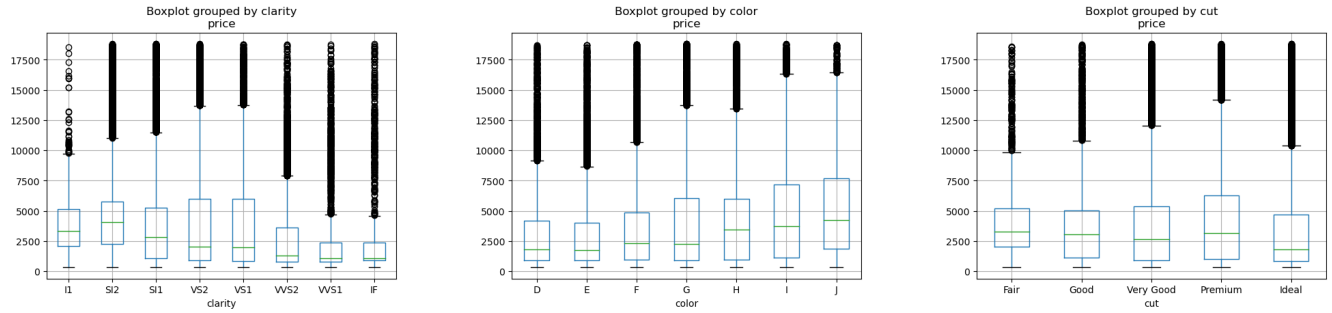
Figure 2: Histogram and KDE for Features

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:

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

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.

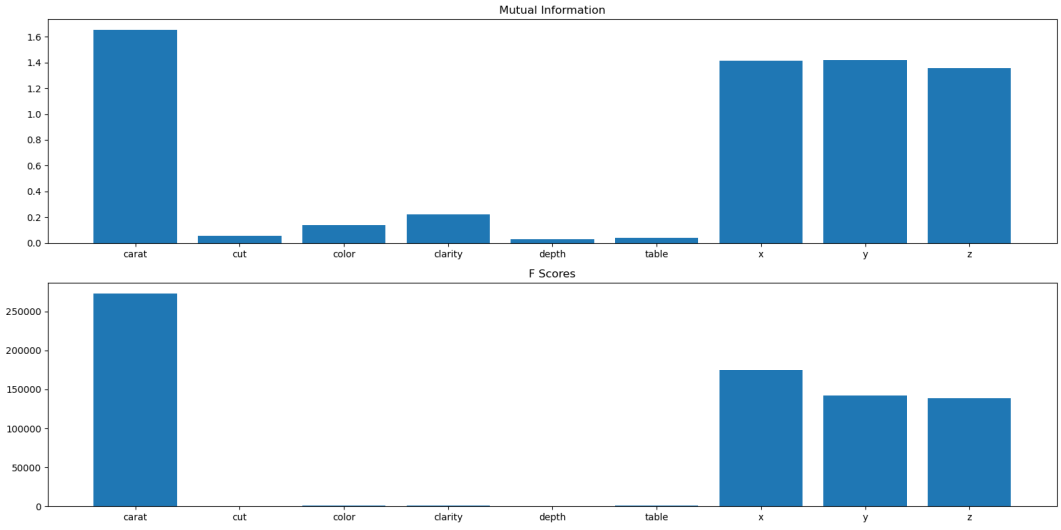


Figure 4: Bar graph of F score and Mutual Information for features

Feature	Mutual Information	F Score
`carat`	1.65	273144
`cut`	0.0578	139
`color`	0.138	1465
`clarity`	0.221	1079
`depth`	0.0307	5.074
`table`	0.0377	802
`x`	1.41	174973
`y`	1.42	142130
`z`	1.36	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.

DONT FORGET TO DO THIS!!!

Regression Models

Linear Regression

Question 4.1

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

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

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

$$\theta = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y} \tag{3}$$

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:

$$\underset{\theta}{\operatorname{argmin}} \frac{1}{2}||\mathbf{Y} - \theta^T \hat{\mathbf{X}}||^2 + \alpha||\theta|| \tag{4}$$

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:

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

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.