Understanding Influences on the Sales Price and Clarity of Diamonds

TEAM DIAMONDS IN THE ROUGH

Alec Meyer | Marcella Anderson | John King | Rachel Grober

1 Introduction

1.1 Questions of Interest:

1. Quantitative: How well can we predict the price of a diamond via a model we build?
2. Qualitative: How well can we predict the clarity of a diamond via the other predictors present in the dataset?

Our data was scraped from the Australian Diamond Importers on February 24, 2022. The dataset has 219,704 observations that include features such as shape, size, color, cut, clarity, symmetry, polish, depth, girdle, culet, price, and more.

Four C's (Clarity, Cut, Color, Carat) is a commonly used phrase among diamond enthusiasts and represents the four most important aspects of a diamond. With that being said, we chose to narrow down our predictors to only those that represent a part of the four C's. Thus, we removed all predictors except size, color, cut, depth_percent, table_percent, meas_lengt, meas_width, and meas_depth. This helped us achieve less complicated, more useful prediction models by limiting the number of predictors to those which professionals view as relevant to a diamond.

Furthermore, while cleaning the model we noticed multicollinearity between several of the predictors. This multicollinearity could have caused reduced accuracy and increased standard errors in the coefficients of our predictors if we chose to complete inference in the future. We sought to understand and minimize this problem. When we looked at the Variance Inflation Factor (VIF) we found there was a significant relationship between the width and length (**Figure 1**). When we graph the variables we can clearly see their linear relationship to each other (**Figure 2**). In order to solve this we multiplied the two variables together to create a new variable. We then noticed this variable had a high linear correlation with size. We expected this as the value for size in diamonds is a product of width, length, and several other dimensional variables. We can see in (**Figure 3**)that there was significant correlation between size and the width x length variable. We removed the size variable from our model and checked our VIF values again. As shown in (**Figure 4**) we no longer had values which would cause concern.

We recognize that we could have removed all four of these size related predictors and just kept size; however, we noticed that pure diamond size was not as responsible for price/clarity as the actual values of depth percent, depth, width, and length were. For example, two diamonds, A and B, with size n, could have lengths x and y, respectively. However, diamond A could have a higher sales price than diamond B.

Our objective for the project's qualitative analysis will be to predict the clarity of a diamond based on its other predictors. Of the "4C's," clarity was chosen as the predictor of choice for the qualitative prediction because, as can be seen in **figure 5**, the variable seems reasonably distributed amongst more than one or two categories. In the box plot in **figure 6**, there is not a significant difference in the average prices of the different clarities. We also see in this chart that the values where clarity is " " do not contain data that lies outside of the normal ranges of the rest of the data. Finally, **figure 7** showcases the order of value perceived with each of the clarity categories for a diamond.

2 Analysis of Question One

How well can we predict the price of a diamond via a model we build?

2.1 Methodology

To be able to complete both of our methodologies (multiple linear regression and regularized regression) utilizing the most amount of information possible, we began the analysis of our first question by cleaning our data through compressing the present factors. The variables that had factors compressed are as follows:

- culet_size (compressed to 'above', 'below', and 'none')
- cut (compressed to 'above', 'below', and 'none')
- polish (compressed to 'above' and 'below')
- symmetry (compressed to 'none' and 'conditioned')
- fluor_color (compressed to 'none' and 'fluor_colored)
- fluor_intensity (compressed to 'above', 'below', and 'none')
- girdle_max (compressed to 'above', 'below', and 'none')
- girdle_min (compressed to 'above', 'below', and 'none')
- eye_clean (compressed to 'listed' and 'none')
- fancy_color_dominant_color (compressed to 'listed' and 'none')
- fancy_color_secondary_color (compressed to 'listed' and 'none')
- fancy_color_overtone (compressed to 'listed' and 'none')

Since we chose to compress these variables we reduced our candidate pool from 136 to 36.

2.1.1 Multiple Linear Regression

When performing supervised learning, we recognized that one of the largest challenges in building a model comes from finding a method for which both variance and squared bias are low. This led us to focusing on building a multiple linear regression model that will reduce the bias-variance tradeoff. In addition, we chose to utilize multiple linear regression because we know that in many cases, it can outperform more complicated models when it comes to prediction. Due to our earlier cleaning of our data to fulfill the four assumptions needed to utilize multiple linear regression, we feel confident that this method will be effective in helping us understand the relationship between the predictors and our response variable.

2.1.2 Regularized Regression

In further efforts to shrink our model we performed lasso and ridge regressions. This was done to increase the bias of our model and decrease the variance. By finding an optimal lambda value with which we can shrink our model's coefficients towards zero, we are introducing bias to the model. We looked to find the model whose variance decreases enough to offset this bias without increasing so much it raises the test MSE.

We chose to use these methods for our project because we had a large number of predictors. This large number is partly due to a significant portion of our variables being factors. With a lot of factors the models we were creating included each option from the factors as their own predictor. This large number of predictors increased the variance of our model significantly. Through the bias-variance trade off we know if we increase the bias we decrease the variance. Since lasso and ridge regressions introduce bias we can use them to help select a model which performs better and has lower variance.

In an effort to find the model which best fit our data, we calculated the optimal lambda for ridge and lasso with the lowest test MSE and with a test MSE one standard deviation away. We then looked at the test MSE of these four numbers and found the minimum value.

2.2 Results

2.2.1 Multiple Linear Regression

We began splitting our data into a training and testing set using a 70/30 split to help us develop a prediction model using multiple linear regression. The results from the multiple linear regression are displayed in **figure 14** of the appendix. Within this figure, you can see that all of the predictors are extremely significant to our model except five of the variables due to their p-values.. Among these five predictors, only three variables proved to be not useful to our model, given a threshold of 0.05. Therefore we removed the following variables: *cutlet_size_bin_none*, *cutlet_condition_bin_none*, and *fancy_color_secondary_color_none*.

We would like to note that if preferred, it could be useful to look further into the variable *cutlet_size_bin_none*, as it had a relatively small p-value, despite not being small enough to satisfy the threshold. However, we chose to leave this variable out to have a smaller, less complex model. Given the results of the multiple linear regression, our model would have 34 predictors and look as follows:

$$\begin{aligned}\widehat{Y} = -512.87 &- X_{clarityBinBelow} * 410.06 + X_{size} * 7095.61 - X_{colorD} * 53.83 - X_{colorE} * 167.45 - \\ &X_{colorF} * 209.38 - X_{colorG} * 257.02 - X_{colorH} * 354.69 - X_{colorI} * 623.26 - X_{colorJ} * 974.09 \\ &- X_{colorK} * 1{,}358.32 - X_{colorL} * 1{,}829.96 - X_{colorM} * 2{,}236.60 - X_{cutbinbelow} * 1{,}658.50 - \\ &X_{cut\ binnone} * 403.22 - X_{depthPercent} * -6.19 - X_{tablePercent} * 1.31 - X_{measLength} * 7.63 + \\ &X_{measWidth} * 115.89 + X_{measdepth} * 4.39 + X_{fancyColorOvertoneBinNone} * 765.64 + \\ &X_{fancyColorSecondaryColorBinNone} * -463.01 - X_{girdleMinBinNone} * 252.03 - X_{girdleMaxBinBelow} \\ &* 24.90 + X_{gidleMaxBinNone} * 178.68 + X_{fluorIntensityBinBelow} * 107.14 + X_{fluorIntensityBinNone} \\ &* 178.84 - X_{FluorColorBinNone} * 44.19 - X_{LabHRD} * 144.59 - X_{LabIGI} * 548.78 + \\ &X_{eyeCleanBinNone} * 64.23 + X_{PolishBinBelow} * 291.92 + X_{SymmetryBinNone} * 208.87\end{aligned}$$

We then analyzed our model on both our training set and our testing set. Given our model was built on our training set, we wanted to ensure we were not overfitting since our model had seen this data before. As a result, we obtained an MSE for our model on both the training set and the testing set. Our training MSE was 677,659.2. Our testing MSE was 683,603.8. Since there was not a significant difference in the training and testing errors, we felt confident that we did not overfit our model.

2.2.2 Regularized Regression

We ran ridge and lasso regressions on the diamond data to attempt to remove some of the variance from our model. Using cross validation we found the optimal lambda for both methods as well as the lambda for the model whose CV error is within 1 standard deviation of the lowest model. Using the same testing and training sets as above, we found calculated the test MSE for our model when using these values:

Lambda	Lambda	Test MSE
Best Ridge CV Error	0.1	683,617.3
Best Lasso CV Error	0.1	683,652.7
1 Standard Deviation Ridge	43.28761	688,944.1
1 Standard Deviation Lasso	14.17474	693,078.9

As you can see above the ridge model with lambda 0.1 has the best test MSE. This model has the following equation:

$$\widehat{Y} = -515.76 + X_{clarityBinBelow}*-410.06 + X_{size}*7098.66 + X_{colorD}*-8.84 + X_{colorE}*-122.56 + X_{colorF}*-164.71 + X_{colorG}*-212.54 + X_{colorH}*-310.44 + X_{colorI}* -570.10 + X_{colorJ}*-930.15 + X_{colorK}*-1314.46 + X_{colorL}*-1786.27 + X_{colorM}*-2192.95 + X_{cutBinBelow}*-1646.15 X_{cutBinNone}*-400.98 + X_{depthPercent}*-6.16 + X_{tablePercent}*-1.33 + X_{measLength}*-8.30 + X_{measDepth}*4.76 + X_{measWidth}*115.81 + X_{fancyColorOvertoneBinNone}*743.67 + X_{fancyColorSecondaryColorBinNone}*-475.96 + X_{girdleMinBinBelow}*-3.01 + X_{girdleMinBinNone}*-198.21 + X_{girdleMaxBinBelow}*25.47 + X_{gidleMaxBinNone}*119.38 + X_{culetSizeBinNone}*-149.53 + X_{culetSizeBinSmall}*-76.13 + X_{culetConditionBinNone}*328.80 + X_{fluorIntensityBinBelow}*107.96 + X_{fluorIntensityBinNone}*178.84 + X_{FluorColorBinNone}*-44.85 + X_{LabHRD}*-143.94 + X_{LabIGI}*-158.90 + X_{eyeCleanBinNone}*64.50 + X_{PolishBinBelow}*295.40 + X_{SymmetryBinNone}*242.27$$

2.3 Discussion

2.3.1 Multiple Linear Regression

When analyzing the effectiveness of our multiple linear regression model, we looked to the adjusted R^2. In the output included in **figure 14**, you will notice our adjusted R^2 had a value of 0.8604. This means that 86.04% of the variation in the total sales price of diamonds can be explained by our predictors. We felt this was a relatively high value for our dataset and could be justified by the idea that there were many significant predictors when utilizing a threshold of 0.05 for analyzing their p-values. In addition, we noticed that we obtained a much more useful model when we compressed our factored variables versus when we did not.

2.3.2 Regularized Regression

Modeling our data with a regularized regression, we found our best option was a ridge regression with a lambda of 0.1. This lambda is very small and therefore does not cause much reduction in the coefficients of our model. We therefore infer there is not much reduction in variance in our model.

2.4 Conclusion

In order to predict the price of diamonds with the data in our dataset we recommend using the multiple linear regression model discussed in 2.2.1. We chose this over regularized regression for two main reasons.

To begin, the multiple linear regression model had a lower test MSE, showing it performed better on data it had not seen. In addition, reduction in variance is the goal of regularized regression and our model did not benefit from this reduction. This suggests regularized regression is not an effective method at increasing the accuracy of our model.

Our multiple linear regression model has a relatively small disparity between the training MSE and the testing MSE, as well as a high adjusted R^2. This gives us confidence in our model to predict the price of diamonds valued under 10k using all but 3 of the variables in our dataset.

3 Analysis of Question Two

How well can we predict the clarity of a diamond via its other predictors?

3.1 Methodology

The goal of this question is to provide a tool for gemologists to help classify the clarity of a diamond. The clarity of a diamond is generally separated into 11 categories:

I3 I2 I1 SI2 SI2 SI1 VS2 VS1 VVS2 VVS1 IF

These categories were not evenly distributed so we felt it would make sense to group these factors into a single binary predictor of "below" and "above". The big question is where should this split be? We decided to split this scale between VS2 and VS1 to keep the responses in each group as close to a 50/50 split as possible. Meaning that ~50% of the clarities were marked as *below* and the remaining clarities were marked as *above*. This split allowed us to maximize our prediction accuracy as the split meant we wouldn't have to exclusively worry about false negatives or false positives. The overall split was around 52% below VS1 and 48% above VS2. This newly created predictor is called *clarityBin*. The only adjustment to our data was removing the *clarity* predictor as *clarityBin* replaces it which means our classification tests were modeled against 27 predictors.

3.1.1 K-Nearest Neighbors

K-nearest neighbors is a classifier that decides the state of a resultant response based on the relative "closeness" between it and the nearby other responses as a function of the predictors. This is a very simple method of classification, but it also is fairly robust as it does not assume any underlying distribution and does not require mathematical solvability.

One potential challenge with this classification method is that a specific number of neighbors must be chosen to examine beforehand. In this case, we chose k by using a cross validation method wherein we randomly select a training set 1/2 the size of the dataset and then split the training set into 10 folds so that each fold can be used as the test set. The error from each iteration is then averaged across all 10 iterations. The k-value that performs the best is then selected.

The last challenge of using K-nearest neighbors in this application is its inability to work with categorical predictors and NA values. To overcome this issue we had to very carefully remove predictors which were overwhelmingly NA and were deemed unimportant in the 4Cs setting. The final model used for K-nearest neighbors will be:

clarityBin, size, color, cut, total_sales_price, depth_percent, table_percent, meas_length, meas_width, meas_depth

3.1.2 Random Forest

Random forest classification is a type of classification where a subset of predictors are used to create a simple decision tree, also known as a weak learner. If there are many weak learners, they can be averaged into an ensemble method, which is a culmination of weak learners. These weak learner decision trees on their own are very simple models that suffer from high variance. However, when high variance models are averaged, the overall variance is reduced.

The reason we felt that a random forest classifier would be best for our dataset is because it requires no distributional assumptions and works well with non-continuous predictors. The one downside to random forest however is its interpretability. This means that our final model will be very difficult to intuitively understand. However, this does not seem to be much of an issue for our specific research, because we are striving for a rudimentary classification tool.

One final benefit to Random Forest is that it can work very well with missing data. When working with real world data, there is a high chance that there will be missing values and Random Forest doesn't require any further manipulation of the set which could interfere with prediction. This being said, the Random Forest implemented on our dataset will contain all predictors (besides *date*, and *diamond_id*).

Random forest requires only two parameters besides the response and predictors:

m - number of predictors in the randomly selected model
n - number of trees to generate

3.2 Results

3.2.1 K-Nearest Neighbors

Running K-nearest neighbors on the dataset specified in section Methodology - K-Nearest Neighbors resulted in a final misclassification rate of 23.69%. This result was obtained by first using cross validation to obtain our best value of k. Based on **Figure 10**, a k of 3 is the most optimal value. The confusion matrix using this k value is shown in **Figure 11**.

3.2.2 Random Forest

The Random Forest model used 27 predictors so the *mtry* value was set to 5 ($\sqrt{27} \approx 5$). The *ntree* value, which is the number of separate trees created, was set to a standard value of 500. A 50/50 split was done on the dataset to separate it into a training set and a testing set. The results for this model are included in **Figure 12** and **Figure 13** in the appendix. The overall misclassification rate was 24.19%

3.3 Discussion

3.3.1 Random Forest

As mentioned previously, the response variable, *clarityBin,* is split into two observations, *above* and *below*. Based on the Random Forest model, the misclassification rate was 24.11%. This means that 24.11% of the time our classifier correctly predicts if a diamond's clarity is above or below the predetermined cutoff. This misclassification rate is fairly high especially for the precision at which diamond clarity is determined.

The main reason for such a high misclassification rate is because there isn't much separation between VS1 and VS2 relative to all of the other predictors. This means that our model was commonly predicting VS1s as VS2s and vice versa. Random forest was rerun three times on the same dataset where only VS1 was removed, only VS2 was removed and both VS1 and VS2 were removed. The results of this test are shown below:

VS1 Removed - Misclassification Rate: 12.62%
VS2 Removed - Misclassification Rate: 16.86%

Both Removed - Misclassification Rate: 8.23%

Based on these results it is clear to see the overlap of VS1 and VS2 which causes a higher misclassification rate. However, it is still difficult to determine if even an 8% misclassification rate is viable in the field of gemology.

Random Forest gives us a graph of variable importance which has *total_sales_price* as the most important variable. Below that are seven other variables, which also appear to be fairly important. After those seven this is a drop of importance. The only notable variable from this chart is *cut*. *Cut* is one of the 4C's and has much less importance than the other 2C's (*color* and *carat*) when predicting clarity.

3.3.2 K-Nearest Neighbors

KNN was used because we did not need to make any assumptions about our data. Once we found an optimal K, we were left with a KNN classifier which successfully predicted if a diamond was below or above our designated clarity with an accuracy of around 24%. As stated above in methodology for question two, our new clarity variable was 48/52 split. A 24% misclassification rate is usable, but not very beneficial for this type of split.

3.3.3 Random Forest vs K-Nearest Neighbors

Random Forest and K-nearest neighbors had very similar misclassification rates of right around 24%, however, they used very different datasets. Random Forest was able to take into account all predictors while K-nearest neighbors had to be modified to work with non-NA and factored categorical predictors. Based on these differences it is difficult to compare the two approaches.

3.4 Conclusion

Diamonds are a very high value item which require trained professionals to determine all of their four C's (clarity, cut, color, carat). With a final misclassification rate of around 24% we have determined that this would not be sufficient for reliably predicting the clarity of a diamond. Our model could be used as a tool to help guide a gemologist in the right direction, but would not be sufficient for complete prediction.

4 Project Summary

Our dataset required extensive cleaning to be useful in building predictive models. In order to predict price this cleaning included condensing many of the discrete variables into larger groups. In order to predict clarity this cleaning included encoding clarity values to *above* and *below* a specified point. For K-nearest neighbors specifically the data needed to be standardized and encoding leaving only numerical predictors.

Once the data was cleaned and organized we were able to build our models. Using predictive analysis we are confident multiple linear regression was an effective way to predict our response, total sales price.

Though using regularized regression methods did not increase the accuracy of our model, they were useful in showing us that decreasing the variance of the model through this method was not necessary to achieve an optimal outcome. When it came to predicting clarity, two fairly different approaches were taken. K-nearest neighbors didn't take into account all of the predictors while Random Forest was able to use all of them. Surprisingly the prediction results between the two methods were very similar. Unfortunately, the results were determined to be inconclusive if used by a gemologist.

Throughout the duration of this project, our team was challenged by the obstacles that are presented when utilizing a real-world dataset. Despite this, our team collaborated with one another to brainstorm and implement useful solutions to these problems and were able to expand our learning within the data science field. We now feel more confident in our own abilities to navigate similar challenges in the real-world and are excited to use what we have learned to make a difference in our future careers.

Appendix

	GVIF
size	3.404221
color	1.260832
clarity	1.073828
cut	2.149765
depth_percent	1.858515
table_percent	1.927184
meas_width	5.179787
meas_length	5.090476
meas_depth	1.211680

Figure 1

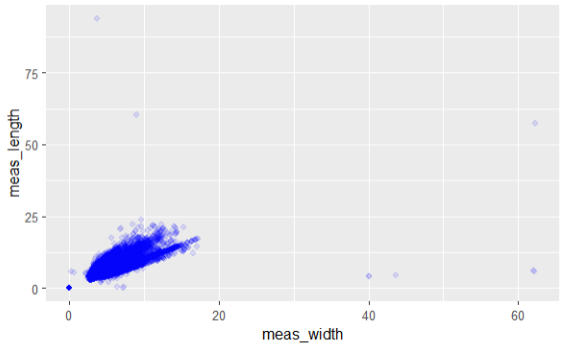

Figure 2

	GVIF
color	1.211301
clarity	1.057981
cut	1.264669
depth_percent	1.853099
table_percent	1.907853
widthlength	1.245830
meas_depth	1.150778

Figure 3

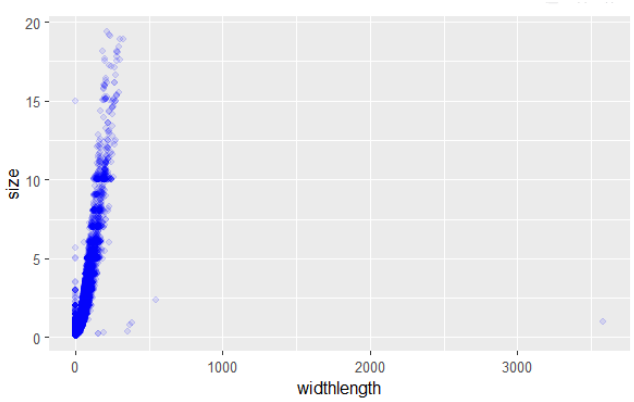

Figure 4

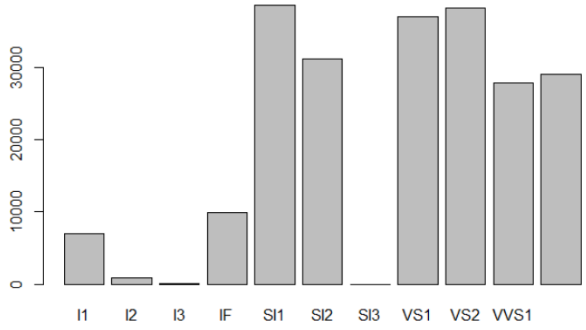

Figure 5

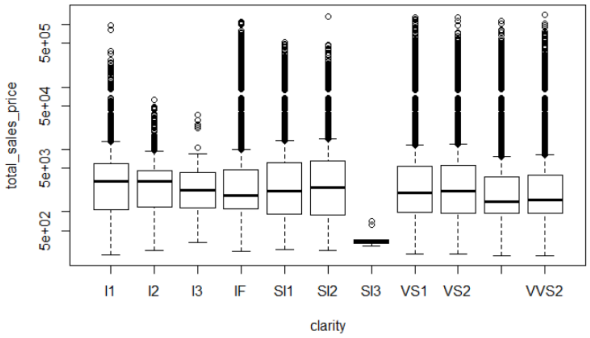

Figure 6

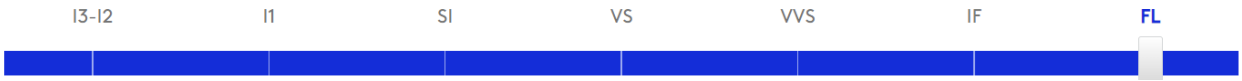

Figure 7

Figure 8

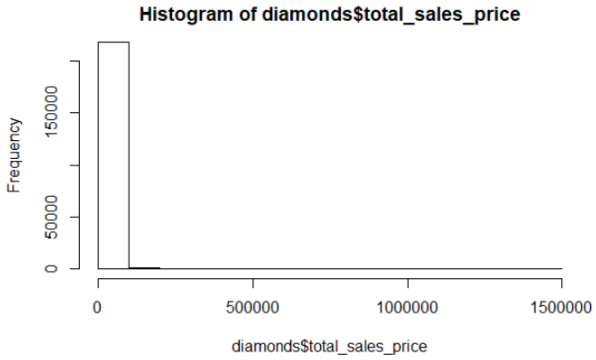

Figure 9

```
(Intercept)        colorD        colorE        colorF        colorG        colorH        colorI        colorJ        colorK        colorL
-11196.849146  -1773.429626  -1841.776357  -2354.721394  -2955.220383  -3438.689737  -4578.242674  -5584.480210  -6474.241257  -7905.762023
      colorM     clarityI2     clarityI3     clarityIF    claritySI1    claritySI2    claritySI3    clarityVS1    clarityVS2   clarityVVS1
 -9310.380428  -5537.588984  -5921.088787   5193.477575     32.729775  -1518.296918  13689.444338   2302.401087   1725.793662   2696.831036
  clarityVVS2  cutExcellent       cutFair       cutGood       cutIdeal       cutNone  cutVery Good depth_percent table_percent   widthlength
  2789.043220   -118.501339 -10493.200806  -8140.416659      0.000000      0.000000   -466.462009    -10.514837      4.592192    612.951501
   meas_depth
   548.296843
```

Lasso model

	Above	Below
Above	30273	9699
Below	9150	30426

Figure 10

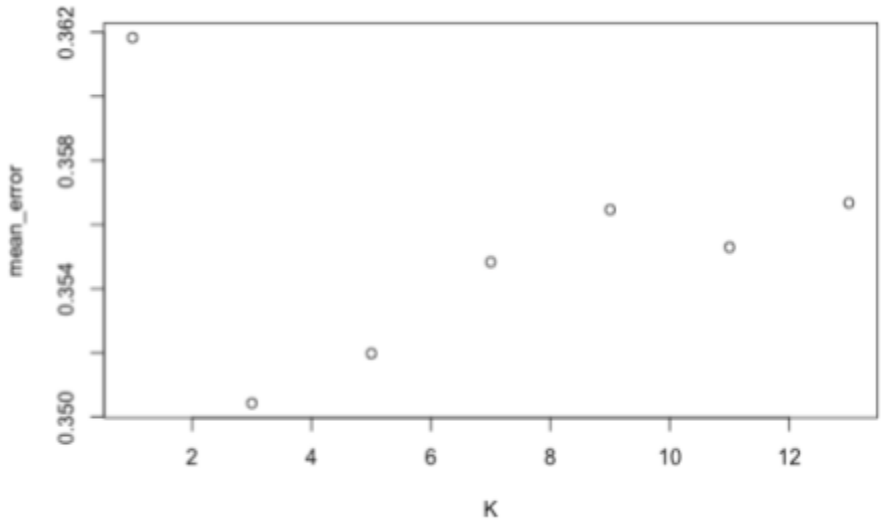

Figure 11

	Above	Below
Above	39236	13213
Below	13366	44037

Figure 12

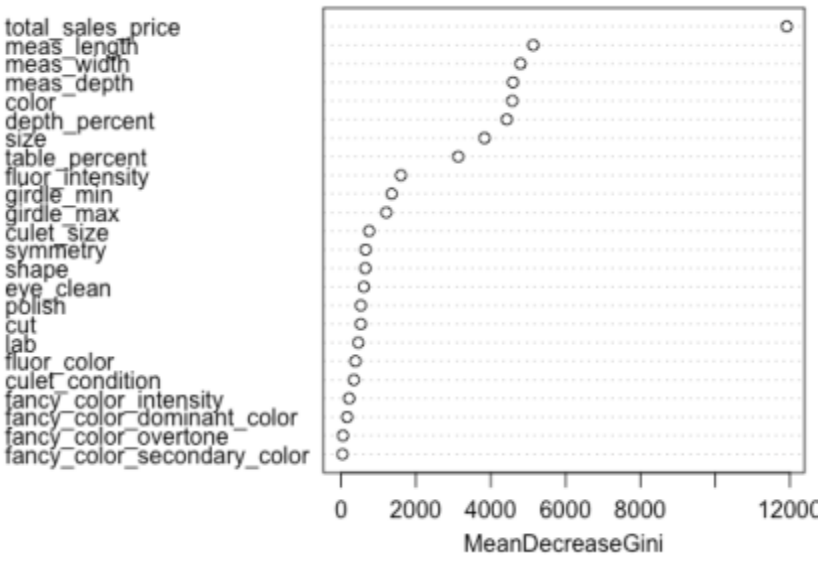

Figure 13

```
Coefficients:
                                      Estimate Std. Error t value Pr(>|t|)
(Intercept)                         -512.8663    97.1740  -5.278 1.31e-07 ***
clarityBinbelow                     -410.0601     4.6311 -88.545  < 2e-16 ***
size                                7095.6087    14.5616 487.283  < 2e-16 ***
colorD                               -53.8294    23.2675  -2.314 0.020696 *
colorE                              -167.4487    23.2579  -7.200 6.07e-13 ***
colorF                              -209.3789    23.2842  -8.992  < 2e-16 ***
colorG                              -257.0230    23.3462 -11.009  < 2e-16 ***
colorH                              -354.6902    23.4568 -15.121  < 2e-16 ***
colorI                              -623.2622    23.6032 -26.406  < 2e-16 ***
colorJ                              -974.0891    23.9112 -40.738  < 2e-16 ***
colorK                             -1358.3197    24.4766 -55.495  < 2e-16 ***
colorL                             -1829.9592    26.4223 -69.258  < 2e-16 ***
colorM                             -2236.6025    29.6961 -75.316  < 2e-16 ***
cut_binbelow                       -1658.4981   206.2829  -8.040 9.06e-16 ***
cut_binNone                         -403.2188     8.4166 -47.908  < 2e-16 ***
depth_percent                         -6.1902     0.3183 -19.449  < 2e-16 ***
table_percent                         -1.3089     0.3137  -4.172 3.02e-05 ***
meas_length                           -7.6319     3.3397  -2.285 0.022301 *
meas_width                           115.8902     4.7675  24.308  < 2e-16 ***
meas_depth                             4.3877     1.2810   3.425 0.000614 ***
fancy_color_overtone_binNone         765.6390    30.3830  25.200  < 2e-16 ***
fancy_color_secondary_color_binNone -463.0099    42.5621 -10.878  < 2e-16 ***
girdle_min_binbelow                   -3.2788     6.8353  -0.480 0.631452
girdle_min_binNone                  -252.0261    40.0825  -6.288 3.23e-10 ***
girdle_max_binbelow                  -24.9003     8.3702  -2.975 0.002932 **
girdle_max_binNone                   172.8624    39.8614   4.337 1.45e-05 ***
culet_size_binNone                  -145.8491    80.3626  -1.815 0.069543 .
culet_size_binsmall                  -72.4668    84.5639  -0.857 0.391476
culet_condition_binNone             -326.0670    10.5443 -30.924  < 2e-16 ***
fluor_intensity_binbelow             107.1490     9.1321  11.733  < 2e-16 ***
fluor_intensity_binNone              178.6814     8.0369  22.233  < 2e-16 ***
fluor_color_binNone                  -44.1893    11.6976  -3.778 0.000158 ***
labHRD                              -144.5929    22.6288  -6.390 1.67e-10 ***
labIGI                              -548.7820    10.4138 -52.697  < 2e-16 ***
eye_clean_binNone                     64.2322     5.5085  11.661  < 2e-16 ***
polish_binbelow                      291.9187    35.7472   8.166 3.21e-16 ***
symmetry_binbelow                    208.8747    28.0127   7.456 8.94e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 823.3 on 132121 degrees of freedom
Multiple R-squared:  0.8605,    Adjusted R-squared:  0.8604
F-statistic: 2.263e+04 on 36 and 132121 DF,  p-value: < 2.2e-16
```

Figure 14

Contributions

Team Member Contributions:

Marcella Anderson: Marcy and Rachel teamed up for the entirety of analysis for question one. The two equally divided up determining code, writing in the final report, and making presentation slides.

John King: Researched KNN and implemented it into our project. Researched manual model selection by researching diamond aspects.

Alec Meyer: All of Random Forest analysis, most of the KNN analysis, half of Multicollinearity analysis, part of model selection. Question two discussion and results, project summary.

Rachel Grober: Rachel and Marcy teamed up for the entirety of analysis for question one. The two equally divided up determining code, writing in the final report, and making presentation slides.

Team Member	Contribution Percentage
Alec	25%
John	25%
Marcy	25%
Rachel	25%