Predicting the Price of Round Cut Diamonds

STAT 684

Shaun Cass

Introduction

For centuries diamonds have been sought after as symbols of wealth and prosperity. They have been adorned on the crowns of royals and in the past century used as a symbol of a couple's eternal love. In the eyes of the public they are a valuable commodity with the worldwide retail market for diamond jewelry having an estimated worth of $60 billion dollars in 2010 (Pisani 2012). Round cut diamonds (also called round brilliant diamonds) are the most popular shape of diamond and are often used in jewelry such as engagement rings. This style of diamond has been around since the 1700s and the techniques used to make this style of diamond have been modified over the years to produce more brilliant and outstanding diamonds (*GIA 4Cs* 2012).

These precious stones are bought and sold by a variety of individuals. There are diamond cutters, jewelry manufacturers, and even just regular people for which there exists a need to understand a round cut diamond's worth. There are websites that one can use that allow them to find the minimum, average, and maximum price of a round cut diamond based on a few basic characteristics (The Diamond Pro 2021). However, many may still need to know if it is possible to more accurately predict the price of a round cut diamond.

This report aims to showcase a method that predicts round cut diamonds with a certain degree of acceptable error based on just a relatively small amount of basic round cut diamond features. The features used in this report are the same sort of features included in a diamond grading report from respected organizations like the Gemological Institute of America ("Sample Natural Diamond Reports." 2021). It is the hope of the author that the model built for this report will aid both professionals and regular individuals in predicting the price of common round cut diamonds whose true value ranges from the low hundreds of dollars to a little less than $19,000.

Methods

About the Data

The Diamonds data set was originally procured by requesting the data set from Dr. Crawford at Texas A&M. Upon further inspection, it was realized that this data set was a common data set used for learning how to make exploratory graphics and perform rudimentary inference. This data set comes standard in the ggplot2 package and includes a description of the data that we have reproduced in table 1.

The Diamonds data set contained the the characteristics of 53,940 round cut diamonds. The variables associated with each diamond are as follows: price (in US Dollars), carat (weight), Cut (Fair, Good, Very Good, Premium, Ideal), Color (D (best) to J (worst)), Clarity (I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best), x (length in mm), y (width in mm), z (depth in mm), depth (total depth percentage), and table (total table percentage). Cut, color, and clarity were ordered categorical variables as described in table 1 while the rest were numerical variables. Based on the data description and the use of the Diamonds data set in other reports (Zheng 2020, chapter 5), the data was believed to contain independent observations and the methods used in this report treated each observation as independent of one another.

Table 1: Description of the Diamonds data set from the ggplot2 package (H. Wickham 2016).

Variable	Description
price	Price in US dollars ($326–$18,823)
carat	Weight of the diamond (0.2–5.01)
cut	Quality of the cut (Fair, Good, Very Good, Premium, Ideal)
color	Diamond color, from D (best) to J (worst)
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 = z / mean(x, y) = 2 * z / (x + y) (43–79)
table	Width of top of diamond relative to widest point (43–95)

There exists two primary graders of diamonds: the Gemological Institute of America (GIA) and the American Gem Society (AGS). Both use the 4Cs to characterize a faceted diamond: color, clarity, cut, and carat weight. Both have similar grading scales for color and clarity, and only slightly differ on the grading for cut. The GIA is used as the main reference for the remainder of this report.

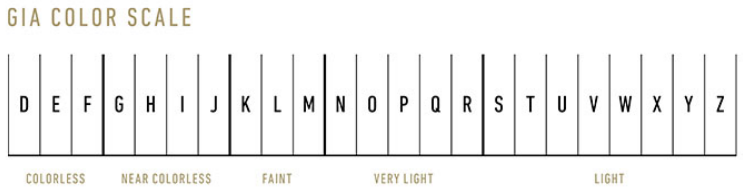

Figure 1: GIA color scale (GIA 4Cs 2017).

The GIA scale for the color of a diamond ranges from the top grade "D" (colorless) to the worst grade "Z" (light yellow) (Figure 1). The diamonds in the data set used only ranged from "D" to "J" meaning that that there may be diamonds with a worse color than the ones in this data set such that the predictive model created would not yield accurate predictions.

Figure 2: GIA clarity scale (GIA 4Cs 2017).

The clarity scale that the GIA uses to grade a faceted diamond ranges from the best grade Flawless to the worst grade I3 (figure 2). Inclusions are imperfections usually caused by the heat and pressure that become apparent with the use of a microscope. An example would be a mineral crystal within the diamond itself or even a break in the gemstone (*GIA 4Cs* 2013). The Diamonds data set only contained diamonds with a clarity that ranged from Internally Flawless (IF) to the first level of Included (I1) so again there may be diamonds with a clarity of Flawless, I2, or I3 for which the predictive model created would not yield accurate predictions.

Figure 3: GIA cut scale (GIA 4Cs 2017).

The GIA scale for the cut of a diamond depends on how the diamond interacts with light and ranges from the best grade Excellent to the worst grade Poor (Figure 3). The cut of the diamonds in the data set ranged from the best grade Ideal to the worst grade Fair. Both scales slightly differ, but each contains five levels. It is unsure if each level directly translates to one another. It may be that the diamond is graded using a variation of the American Gem Society scale as seen in figure 4; however, the scale in the diamonds data set still differs slightly. Further investigation may be needed before using the predictive model on diamonds using either the GIA or AGS cut scale.

AGS	0	1	2	3	4	5	6	7	8	9	10
	AGS Ideal	AGS Excellent	AGS Very Good	AGS Good		AGS Fair			AGS Poor		
INDUSTRY	Excellent		Very Good	Good		Fair			Poor		

Figure 4: AGS cut scale (American Gem Society 2021).

Figure 5 illustrates the anatomy of a typical cut diamond. The top horizontal facet of a diamond is called the table and the table size as used in the data set is the percentage of the length of this facet compared to the average the average girdle diameter. The total depth as used in the data set is the percentage of the overall depth relative to the average girdle diameter (*GIA 4Cs* 2014). Since these were round cut diamonds, x (length) and y (width) are two different measurements of the diameter and their average is used when calculating the the table percentage and the total depth percentage.

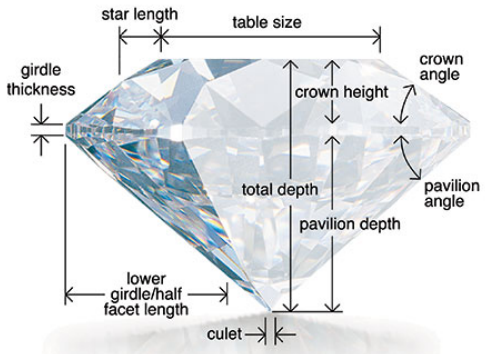

Figure 5: GIA anatomy of a diamond (GIA 4Cs 2017).

The data was analyzed and a predictive model was built using the statistical programming language R within Rstudio along with the following packages: doParallel, GGally, rpart.plot, tidymodels, tidyverse, and vip.

Table 2: Abnormally large values of y in the Diamonds data set.

id	carat	cut	color	clarity	depth	table	price	x	y	z
24068	2.00	Premium	H	SI2	58.9	57	12210	8.09	58.9	8.06
49190	0.51	Ideal	E	VS1	61.8	55	2075	5.15	31.8	5.12

Missing Values and Outliers

The data initially had no missing values, but upon further inspection it was found that there were 20 rows where either the x, y, or z measurement had a value of zero. These zero values were replaced with missing values (NA in R) as it would not be possible for a round cut diamond to have zero length, width, or depth. 12 of these rows only contained one variable with a missing value: the z variable. Because the total depth percentage, x, and y were not missing and the total depth percentage by definition was a calculation using x, y, and z, it was considered appropriate to replace these 12 missing z values using the following formula derived from the total depth percentage calculation:

$$z = \frac{depth\% * (x + y)}{2 * 100}$$

The calculated z value was rounded to the second decimal place to match the format of the other z values in the data set.

Additionally, two observations were found where y (width in mm) was abnormally larger than x (length in mm). Since round cut diamonds are human or machine cut diamonds where x and y are supposed to be relatively close to each other, it was highly unlikely that y would be six to seven times greater than x as seen in table 2. At first it was thought that the values could have been accidentally multiplied by ten; however, the depth (z) of the diamond tends to be smaller than the average of x and y, and round cut diamonds having greater than 100% depth percentage appeared to be extremely unlikely as a round cut diamond is considered poor once its depth percentage is greater than 70.9% (Figure 6).

Total Depth	
Possible Grade	Parameter Range
Excellent to Poor	57.5% to 63.0%
Very Good to Poor	56.0% to 64.5%
Good to Poor	53.0% to 66.5%
Fair to Poor	51.9% to 70.9%
Poor	<51.9% to >70.9%

Figure 6: GIA total depth percentage grading for round cut diamonds (Blodgett et al. 2009)

Likewise, using GIA's $Facetware^{(R)}$ which compares proportions of round cut diamonds to 38.5 million proportion sets, the highest total depth percentage one is able to achieve using a given set of parameters is 83.8% with a typical error of 0.2% - 0.3% ("About GIA Facetware®." 2021). Examining table 2 further, it was also apparent that the z value was extremely close in value to x. Based on this information it was decided that the z values in table 2 were actually supposed to be the y values. These two observations were corrected by replacing the y values with the given z values and calculating new z values using the same formula as before.

Two other observations were found where the depth (z) was greater than the average of x and y (table 3). These observations were also corrected so using the same z formula as before.

Table 3: Diamonds where depth (z) is greater than the average diameter.

id	carat	cut	color	clarity	depth	table	price	x	y	z
48411	0.51	Very Good	E	VS1	61.8	54.7	1970	5.12	5.15	31.80
49906	0.50	Very Good	G	VVS1	63.7	58.0	2180	5.01	5.04	5.06

To find additional discrepancies in the data, the total depth percentage was calculated for all other observations using the formula in table 1. There were obvious differences between the calculated and recorded total depth percentages as seen in figure 7. It was found that there were 120 observations with a 1% error or more between the recorded and calculated total depth percentage. Going further, there were 18 observations with a 10% error or more between the recorded and calculated total depth percentage. The x and y features of these 18 observations were noticeably similar with x ranging from 0.75 t0 1.62 times the size of y. The z values were again suspected based on their recorded values to be the reason for the noticeable difference between the recorded and calculated total depth percentages. The z values were thus replaced for these 18 observations using the same formula as before. To avoid possibly biasing the data, any remaining observations that had a difference between the recorded and calculated total depth percentage were left as is.

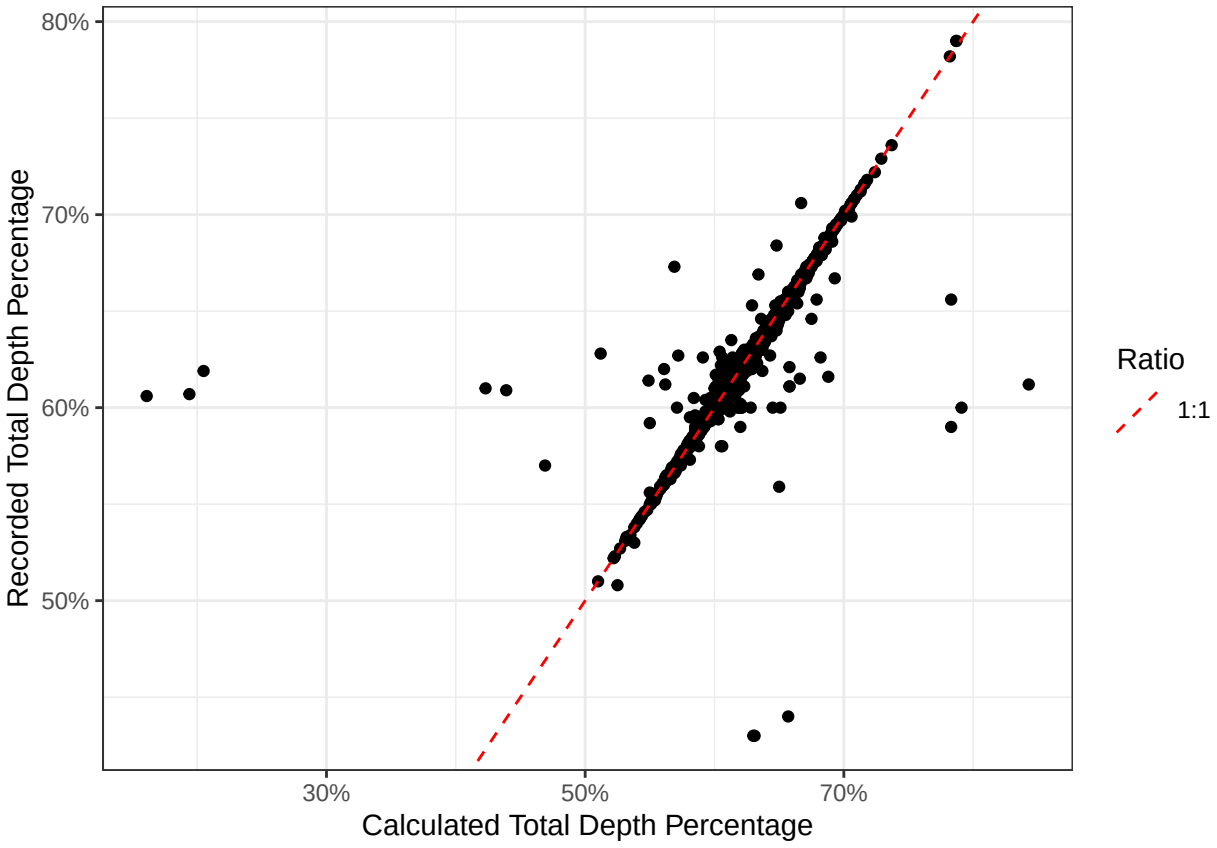

Figure 7: Comparison of the recorded and calculated total depth percentage.

After cleaning the data, only eight observations in total remained that contained missing values. Seven of the observations had missing values in x, y, and, z, while the last observation had only missing values in x and z. It was decided to keep these observations instead of deleting them in order to retain as much data as possible. A bagged tree imputation method with 25 trees was implemented programmatically using the

step_impute_bag() function in the recipes package that is included in the tidymodels package. This method was chosen as a tree can be constructed in the presence of other missing data, trees generally have good accuracy, and they do not extrapolate outside the bounds of the training data. Bagged tree imputation methods also provide reasonable values with a lower computational cost than random forest imputation methods (Kuhn and Johnson 2020, chapter 8.5). The method was implemented in such a way to avoid data leakage e.g. the pre-processing was done within each fold of the 10-fold cross validation and used appropriately for the training and test set.

Examining the Cleaned Data

The response variable for the Diamonds data set was decided to be the price in US dollars for each diamond.

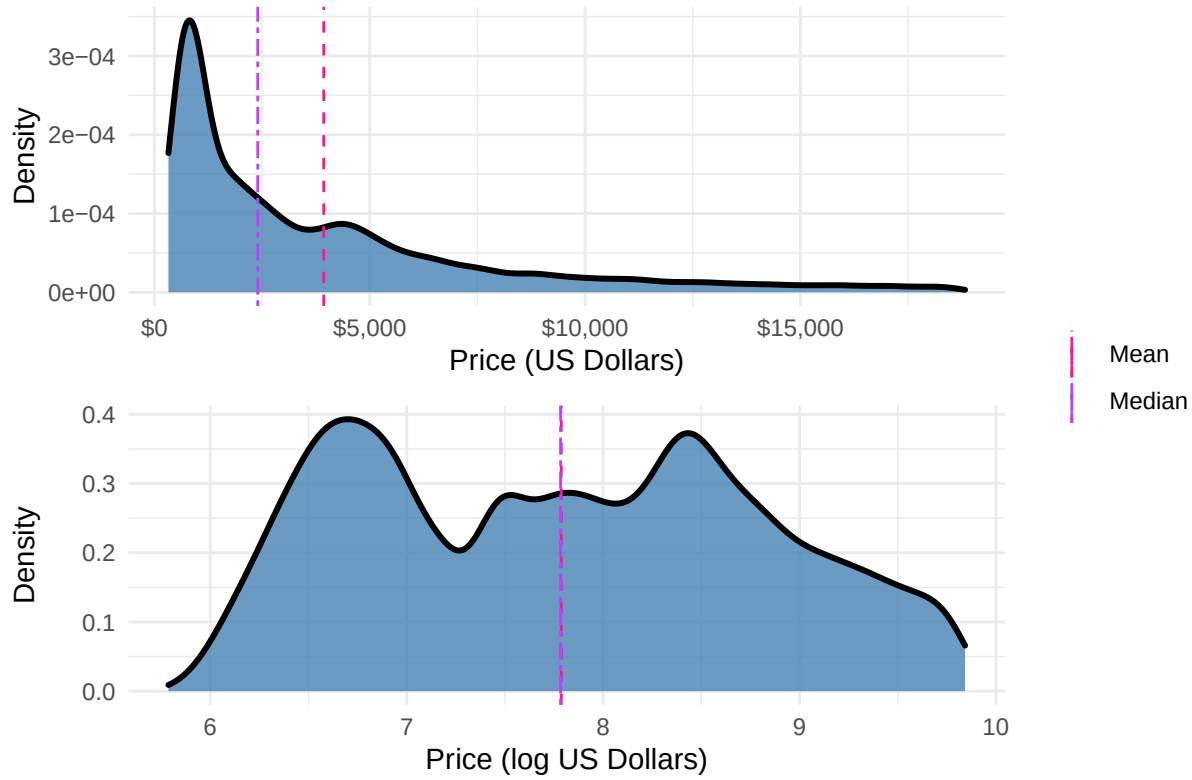

Figure 7.1: Distribution of price and its natural log transformation.

Price had a heavily right-skewed distribution as noted in figure 7.1 with a standard deviation of $3989.4, sample mean of $3932.8 with a corresponding standard error of $62.3 and a sample median of $2401. Price was natural log transformed so that no diamonds would be predicted to have a negative value of price and any errors in predicting the more expensive diamonds would not have a disproportionate influence on the model.

The other numerical features of the diamonds were also examined as seen in figure 7.2 and figure 7.3.

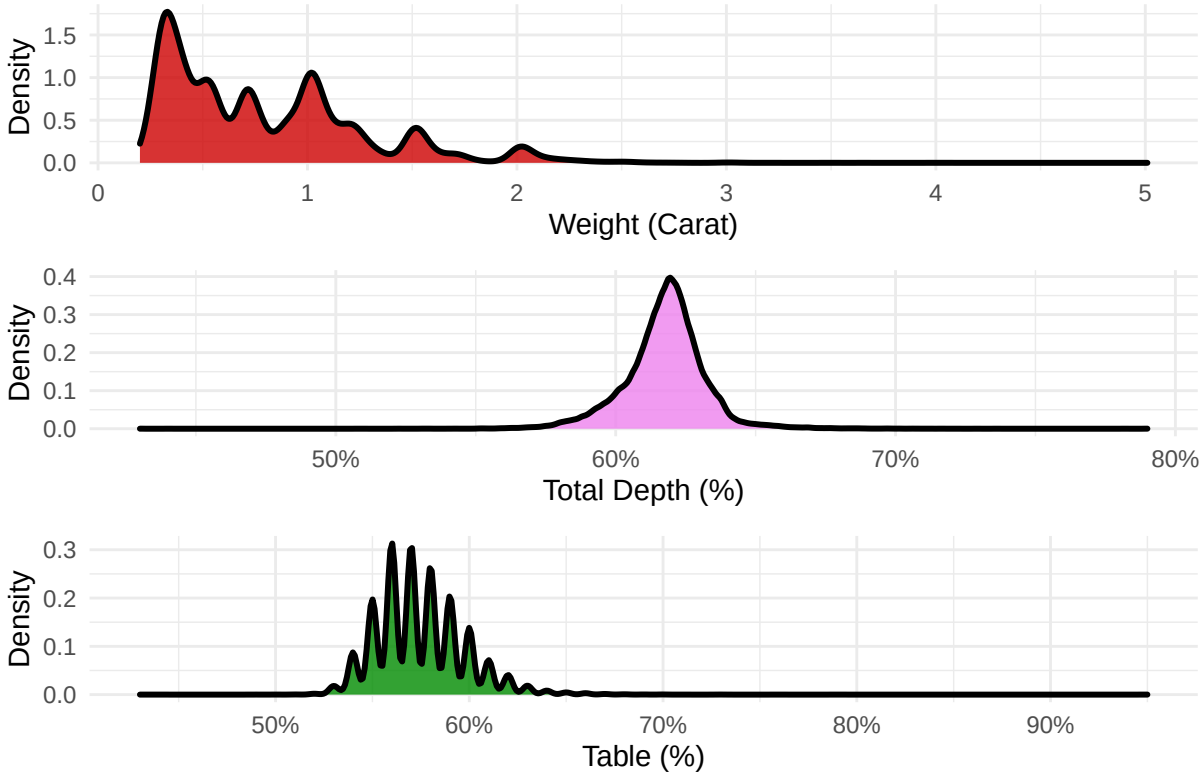

Figure 7.2: Distribution of carat, total depth percentage, and table percentage.

Like price, carat appeared to be right-skewed while the total depth percentage and table percentage were fairly close to symmetrical. On the other hand, the distributions of x and y were similar which was to be expected for round cut diamonds. The distribution of z was also noticeably centered on smaller values than x or y which again was in-line with previous assumptions.

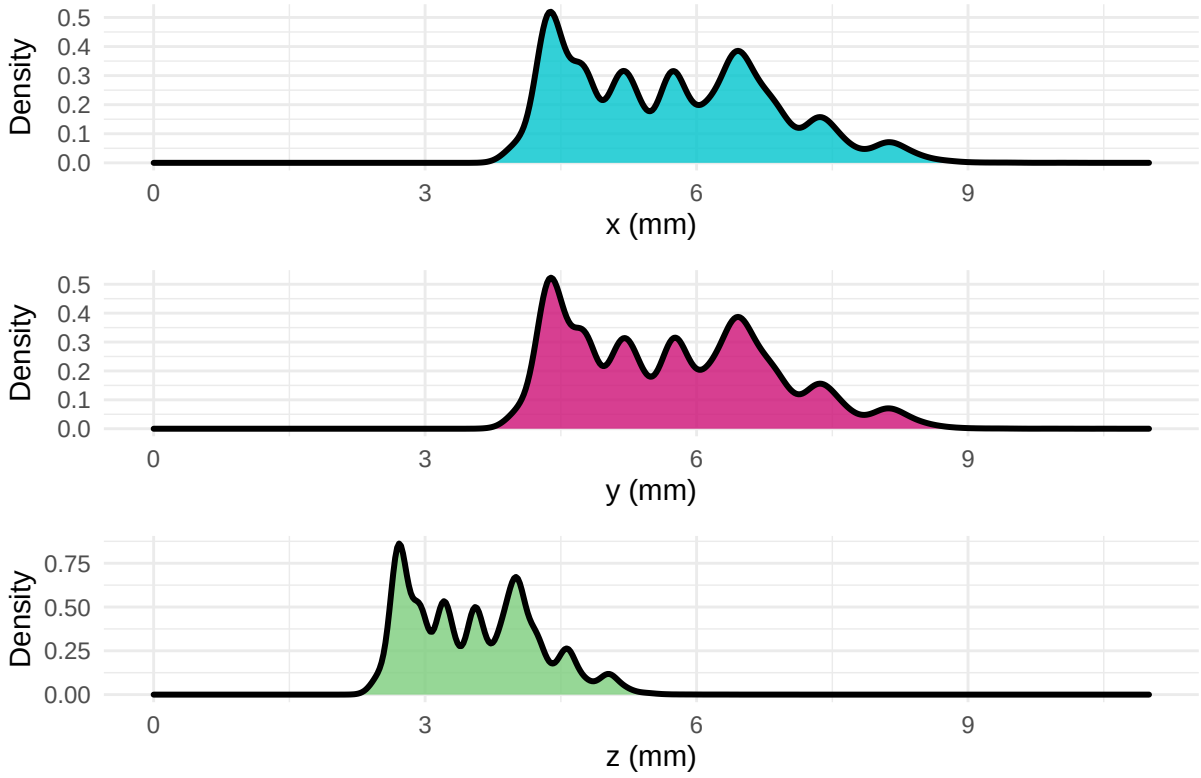

Figure 7.3: Distribution of x, y, and z (excluding eight observations with missing values).

The proportion of diamonds for each level of cut, clarity and color was also examined (figure 7.4). The worst levels of cut (Fair), clarity (I1), and color (J) made up a significantly smaller proportion of the data set compared to the other levels. Furthermore, the best level of clarity (IF) was only present in 3.32% of the data. These proportions indicated that predictions on diamonds having these levels may not be as robust due to the relatively small amount of data used to train a model.

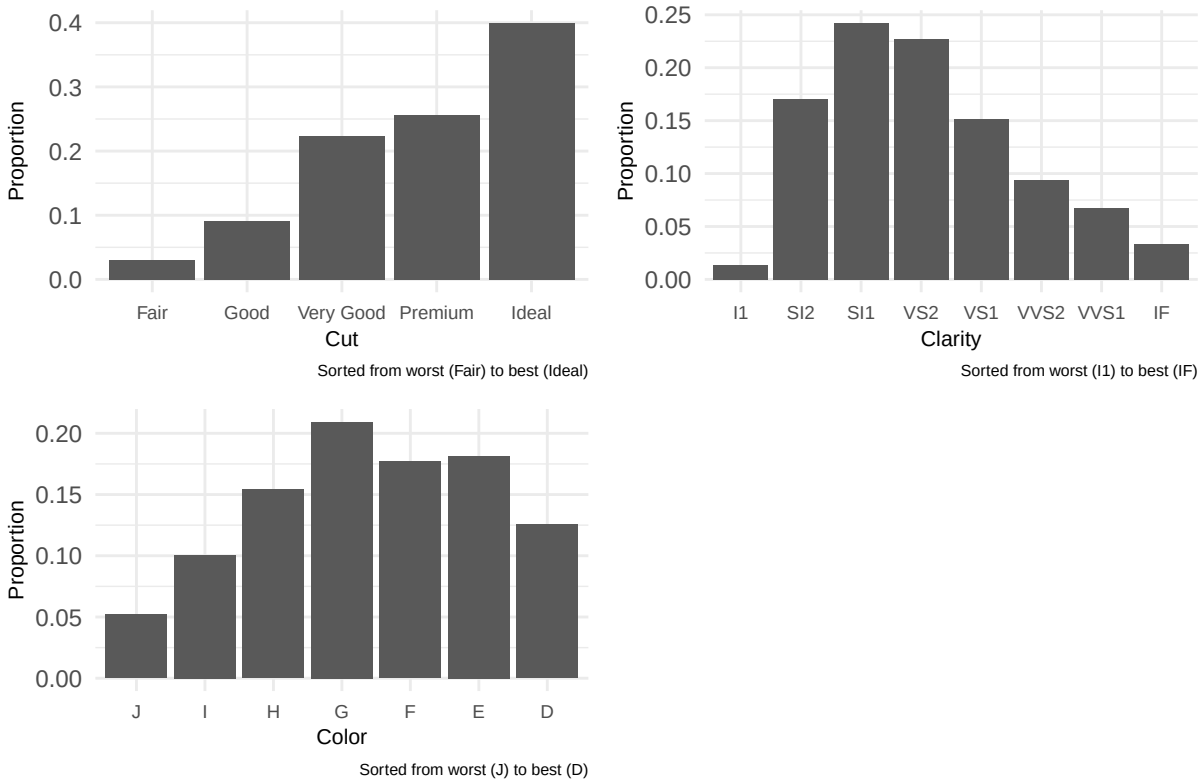

Figure 7.4: Distribution of cut, clarity, and color.

This was was further explored by examining the bivariate proportions for cut, color, and clarity (figure 7.5). If any two-way interactions existed between these categorical features, then the sparse amount of diamonds for certain combinations of these features may impact the accuracy of a predictive model on similar diamonds. For instance, there were only nine diamonds in total with a cut equal to "Fair" and a clarity equal to "IF". Predictions for diamonds similar to these may not be as accurate as the other combinations.

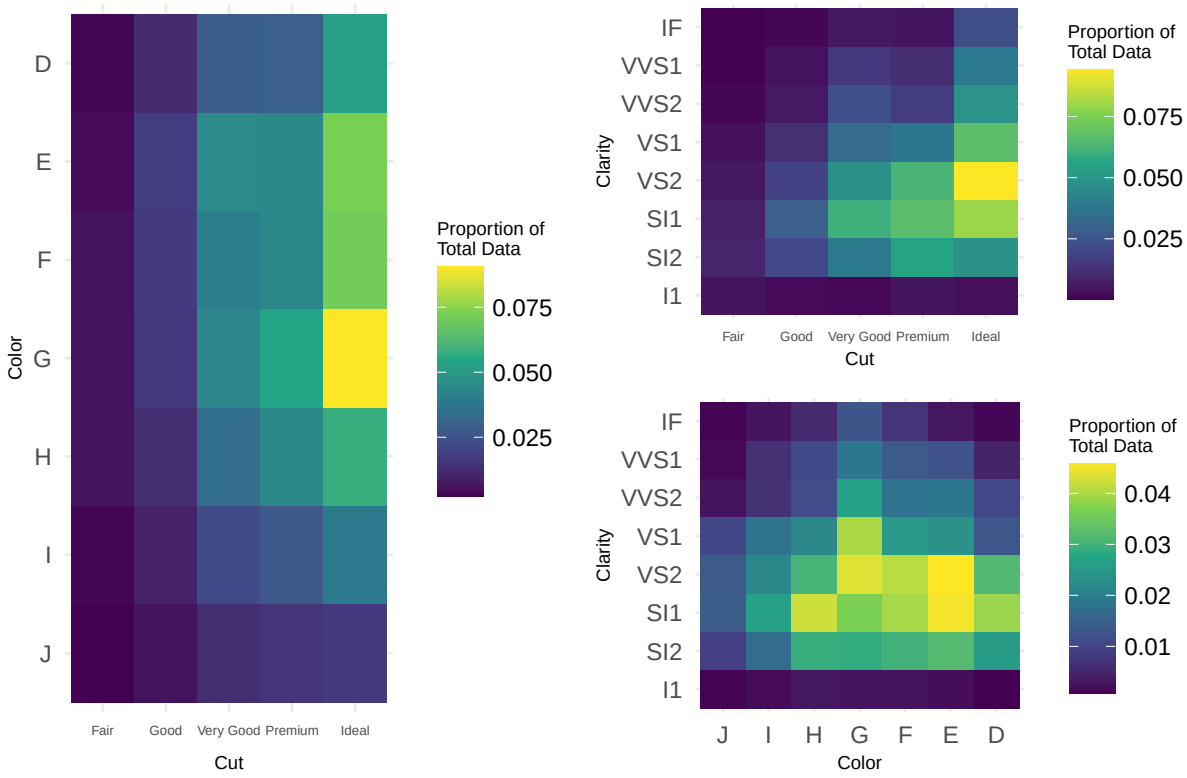

Figure 7.5: Bivariate proportions of the total data for cut, color, and clarity.

When the distribution of the natural log of price was compared across the levels of cut, clarity, and color (figure 7.6), it became obvious that many of the levels that made up a smaller proportion of the data had a much tighter distribution than the other levels. Some such as the I1 and IF level of clarity also had a fair amount of outliers. This may be due to how the data was sampled and may not be representative of how these types of diamonds are regularly priced. It was determined that depending on importance of these categorical variables, more diamonds with these rarer features may be needed in the future to increase the predictive accuracy of the model created.

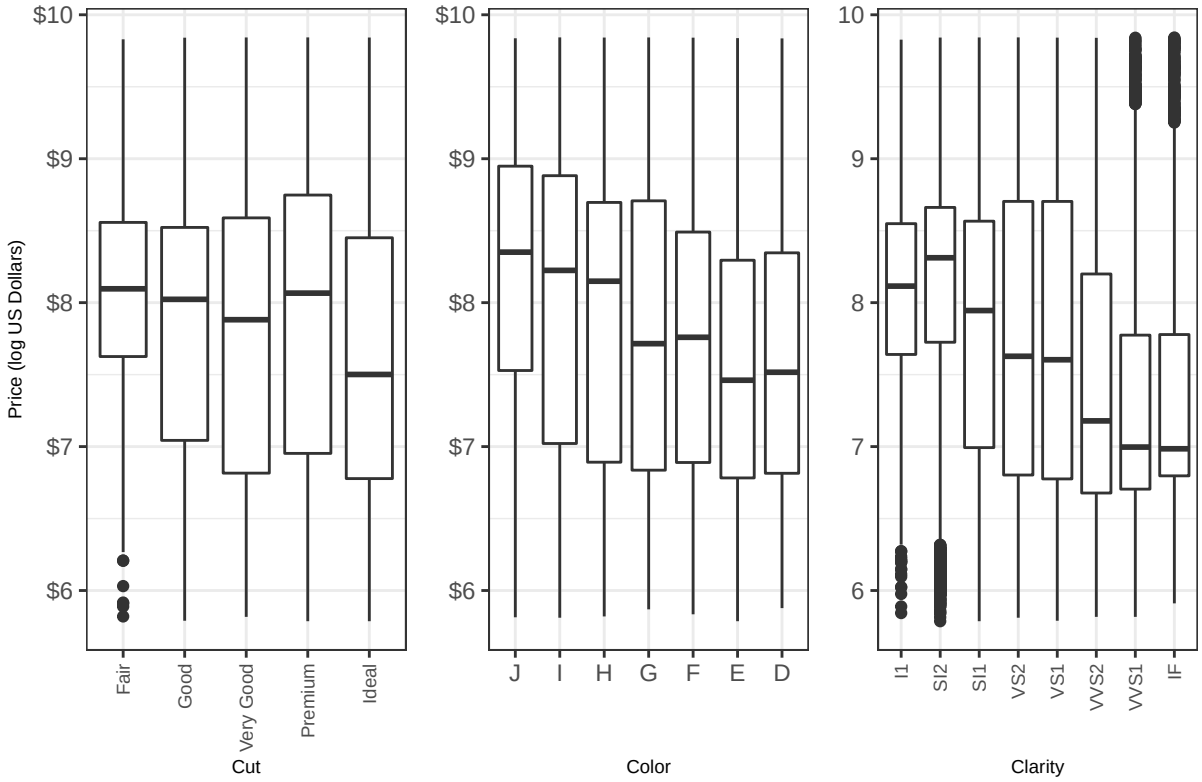

Figure 7.6: Distribution of log price for levels of cut, color, and clarity.

Bivariate examinations between each numerical predictor, and examinations between the numerical predictors and the response were held off until after splitting the data. This was done to avoid biasing the model using information from the test set.

Splitting the data

Before splitting the data, the price was natural log transformed as mentioned in the previous section and the remainder of the report will simply refer to the variable as "price" unless otherwise stated. The data was split using stratified random sampling (stratified on price) and an 80/20 split so that there were 43,156 observations in the training data and 10,784 observations in the test data set. The stratification was done so that random sampling was performed in each of the four quartile bins of price. This was performed in order to avoid having an inordinate amount of observations from the tails of the distribution in the train or test set. The test set was left untouched until the final model was trained on the training data set and predictions were made for the test set.

Pearson correlations were computed for the numerical variables in the training data and multiple variables were found to be highly correlated (Figure 8).

Pairwise examinations of the numerical variables in training set (Figure 9) also found that there existed some nonlinear association between price and many of the numerical predictors. A natural log transformation of carat, x, y, and z corrected this association such that there was both a linear relationship between the response and these predictors, and a linear relationship between each of these predictors (Figure 10). These transformations was determined to be necessary for any linear predictive model tested.

One item that was noted was that both table and depth did not appear to have any significant correlation with the response variable price. Also, based on the pairwise plots, these predictors did not appear to have any noticeable relationship with any of the other predictors.

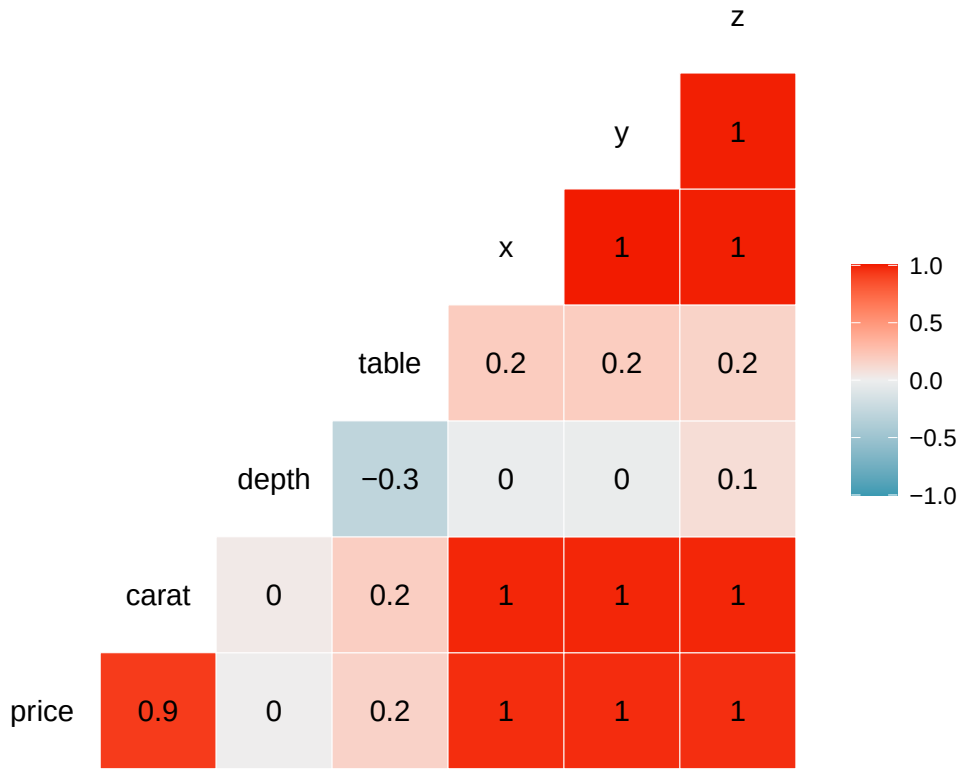

Figure 8: Pearson correlations of numerical variables in the training data.

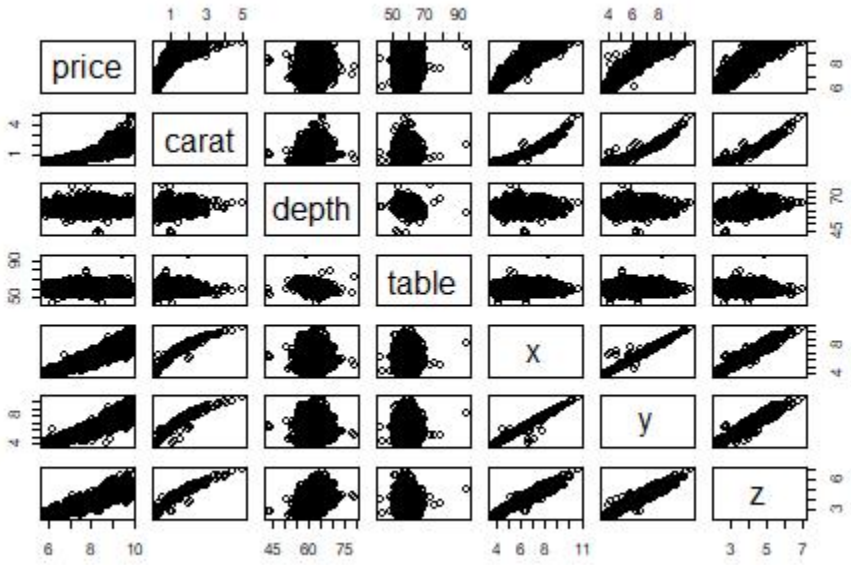

Figure 9: Bivariate relationships of numerical variables in the training data.

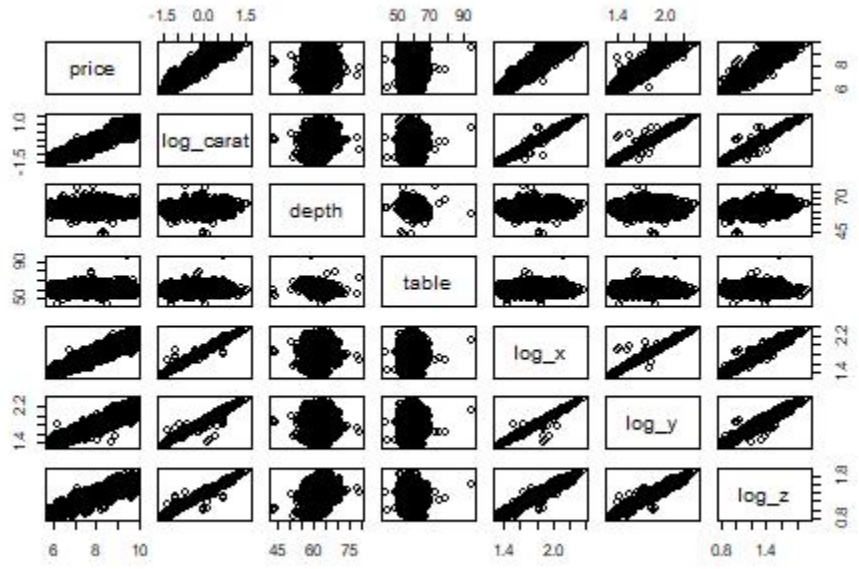

Figure 10: Bivariate relationships with natural log transformations.

Preliminary models and tuning

As seen in the previous sections, there were two main hurdles to overcome in building an accurate predictive model for the diamonds in this data set. The first was that many of the numerical predictors were highly correlated with one another. The second obstacle was that due to the large amount of levels for each categorical variable, there may exist two-way or even three-way interactions that may impact the accuracy of the predictions. The size of the interaction effects was not deemed important for predicting the price, instead these interactions needed to just be considered in building a model.

Four different types of preliminary predictive models were tuned and tested using 10-fold cross-validation in an attempt to overcome these hurdles: ridge regression, glmnet regression, single regression tree model, and a random forest model.

A ridge regression was considered as the squared L2 penalty on coefficients of a linear model can help overcome multicollinearity between the predictors. This penalty shrinks the regression coefficients, but never truly makes any of them equal to zero.

A glmnet regression model on the other hand has both the squared L2 penalty that induces shrinkage on the parameters and an L1 penalty that allows coefficients to equal zero. In other words, the glmnet regression model allows for model selection to happen while the ridge regression model does not (Hastie, Tibshirani and Friedman 2009, 61-69).

A regression tree (using the CART method) recursively partitions the feature space using binary splits. These models can capture capture complex structures in the data and have relatively low bias if grown sufficiently deep (Hastie, Tibshirani and Friedman 2009, 587-588). However, there are two main issues usually with using a regression tree for prediction: trees can easily overfit the data (though this can be limited by adjusting the size of a tree) and regression trees on their own tend to have high variance as often a slight change in the data can lead to different splits (Hastie, Tibshirani and Friedman 2009, 307 - 312).

A technique for reducing the variance of methods like regression trees is bootstrap aggregation (bagging) where a regression tree is fit many times to bootstrapped sampled versions of the training data and results are averaged. A random forest model is an extension of this method where a large collection of de-correlated trees are built and the results are averaged (Hastie, Tibshirani and Friedman 2009, 587).

Table 4: Linear models considered for ridge and glmnet tuning.

Model	Formula
Fit1	Linear combination of: cut, clarity, color, x, y, and z
Fit2	Linear combination of: cut, clarity, color, x, y, z, depth, and table
Fit3	Same as Fit1 except two-way interactions cut:carat, clarity:carat, and color:carat are included.
Fit4	Same as Fit1 except all two-way interactions between cut, clarity, and color are included.
Fit5	Same as Fit4 except all three-way interactions between cut, clarity, and color are included.
Fit6	Same as Fit4 except two-way interactions cut:carat, clarity:carat, and color:carat are also included.
Fit7	Same as Fit5 except two and three-way interactions between carat and the categorical variables cut, clarity, and color are included.

For all the preliminary models, missing values of x, y, and z were imputed using a bagged tree method as described in a previous section and carat, x, y, and z were natural log transformed. For the ridge regression and glmnet regression prediction models, the cut, color, and clarity were encoded using ordinal score encoding to maintain the order of the factors and all numerical predictors were normalized. The tree models did not require encoding (categorical predictors were initially setup as ordinal factors) or normalization.

Seven different linear models were formulated for the preliminary ridge and glmnet regression models (table 4). A regular grid of 50 ridge penalties was supplied for the ridge regression models. For the glmnet models a regular grid of 50 penalty values and 50 mixture values was supplied. The mixture value is the proportion of the L2 penalty (LASSO) portion of the glmnet model to the L1 (ridge) portion of the model such that a mixture = 1 means that the glmnet model is a pure LASSO model and a mixture = 0 means that the glmnet model is a pure ridge model.

To tune the regression tree, a regular grid of 20 values for the cost complexity, the maximum tree depth, and min_n (the minimum number of data points for a node to be split further) were supplied. The cost complexity parameter is a penalty that limits the size of trees. A larger value of the cost complexity parameter results in a smaller tree while a smaller value results in a larger tree.

The initial random forest was tuned using a regular grid of 20 values for mtry (the number of predictors that will be randomly sampled at each split when creating the trees) and min_n which is the same parameter as used in the regression tree. A total of 1000 trees was produced for each random forest model tested. Due to only having 43,156 observations in the training data and 9 predictors, it was determined that 1000 trees should be a sufficient amount based on the number of trees required for prediction error stabilization on other similarly sized data sets (Hastie, Tibshirani and Friedman 2009, 371 & 591).

These preliminary models were tuned and tested using 10-fold cross-validation. A K-fold cross-validation procedure seeks to provide an estimate of the test error by splitting the training data into k roughly equal folds. Each fold is tested such that a model is trained on k-1 of the folds and predictions are made on the remaining fold. An estimate of the prediction error is computed and the procedure is repeated such that predictions for each fold are only computed once. If you let $k : \{1, ..., N\} -> \{1, ..., K\}$ be an indexing function that indicates the fold to which the observation i is allocated and let $\hat{f}^{-k}(x)$ be the fitted function

with the k part of the data removed, then the cross-validation estimate of the prediction error is:

$$CV(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f^{-k(i)}(x_i))$$

Typically a K value of 5 or 10 is chosen (Hastie, Tibshirani and Friedman 2009, 241 - 242). A K value of 10 was chosen for the diamonds data due to the size of the data and the number of different combinations of the categorical values. The prediction error estimate chosen for each step of the 10-fold cross-validation (the loss function $L()$) was the root mean squared error (RMSE) which as the name implies is just the square root of the common error metric the mean squared error:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}$$

Here n is the sample size of the fold left out for prediction, Y_i is the observed price, and $\hat{Y}_i$ is the predicted price. This statistic helps assess the accuracy of a model and has the advantage of using the RMSE instead of just the MSE is that the prediction error is in the same units as the response. The best model as determined by the smallest mean RMSE for each glmnet and ridge regression fit is shown in figure 11 along with the seven best regression tree and random forest models. It was apparent that the linear models fared consistently worse than many of the tree models and the random forest models fared much better than any of the other models tested. It was decided that a random forest model would provide the best predictions for the round cut diamonds.

The Random Forest Model and Further Tuning

As mentioned before, a random forest is a form of bootstrap aggregation of regression or classification trees. A simple regression tree like the one shown in figure 12 seeks to recursively partition the feature space using binary splits and model the response as a constant in each region. A greedy algorithm for creating such a tree is described in The Elements of Statistical Learning as follows (Hastie, Tibshirani and Friedman 2009, 307):

- Starting with all the data, consider a splitting variable j and split point s, and define the half-planes:

$R_1(j,s) = \{X|X_j <= s\}$ and $R_2(j,s) = \{X|X_j > s\}$

- Then we seek the splitting variable j and split point s that solve

$$min_{j,s}[min_{c_1}\sum_{x_i \in R_1(j,s)}(y_i - c_1)^2 + min_{c_2}\sum_{x_i \in R_2(j,s)}(y_i - c_2)^2]$$

- For any choice j and s, the inner minimization is solved by:

$$\hat{c}_1 = ave(y_i|x_i \in R_1(j,s))$$
$$\hat{c}_2 = ave(y_i|x_i \in R_2(j,s))$$

- Having found the best split, we partition the data into two resulting regions and repeat the splitting process on each of the two regions.

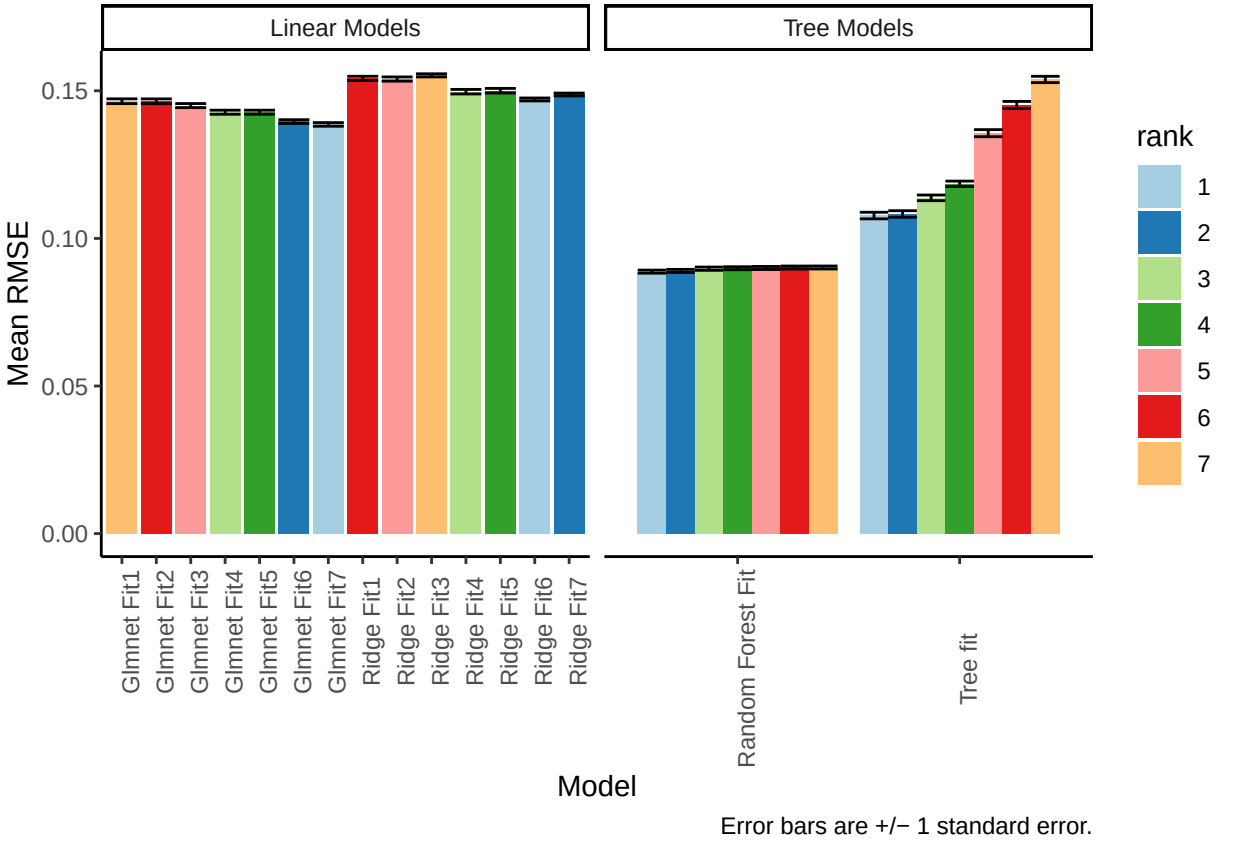

Figure 11: Comparison of the seven best linear and tree based models found using 10-fold CV. The best model was chosen for each linear fit and compared to the seven best regression trees and random forests.

- This is done until the tree is grown to a sufficient size as dictated by parameters such as the cost complexity and the minimum number of observations required for a node to be split further. If the feature space is partitioned into M regions $(R_1, R_2, ..., R_M)$ then the final prediction is the $\hat{c}_m$ of the region that an observation resides in.

A tree is a relatively simple model and is easily interpretable. As seen in figure 12, the tree starts at the root node at the top and binary splits subject to a condition are made on various predictors. If the condition is true then observations follow the split to the left while the other observations follow the split to the right. The terminal nodes at the bottom of the tree are the resulting predictions of the response for any observations that end up in these final regions.

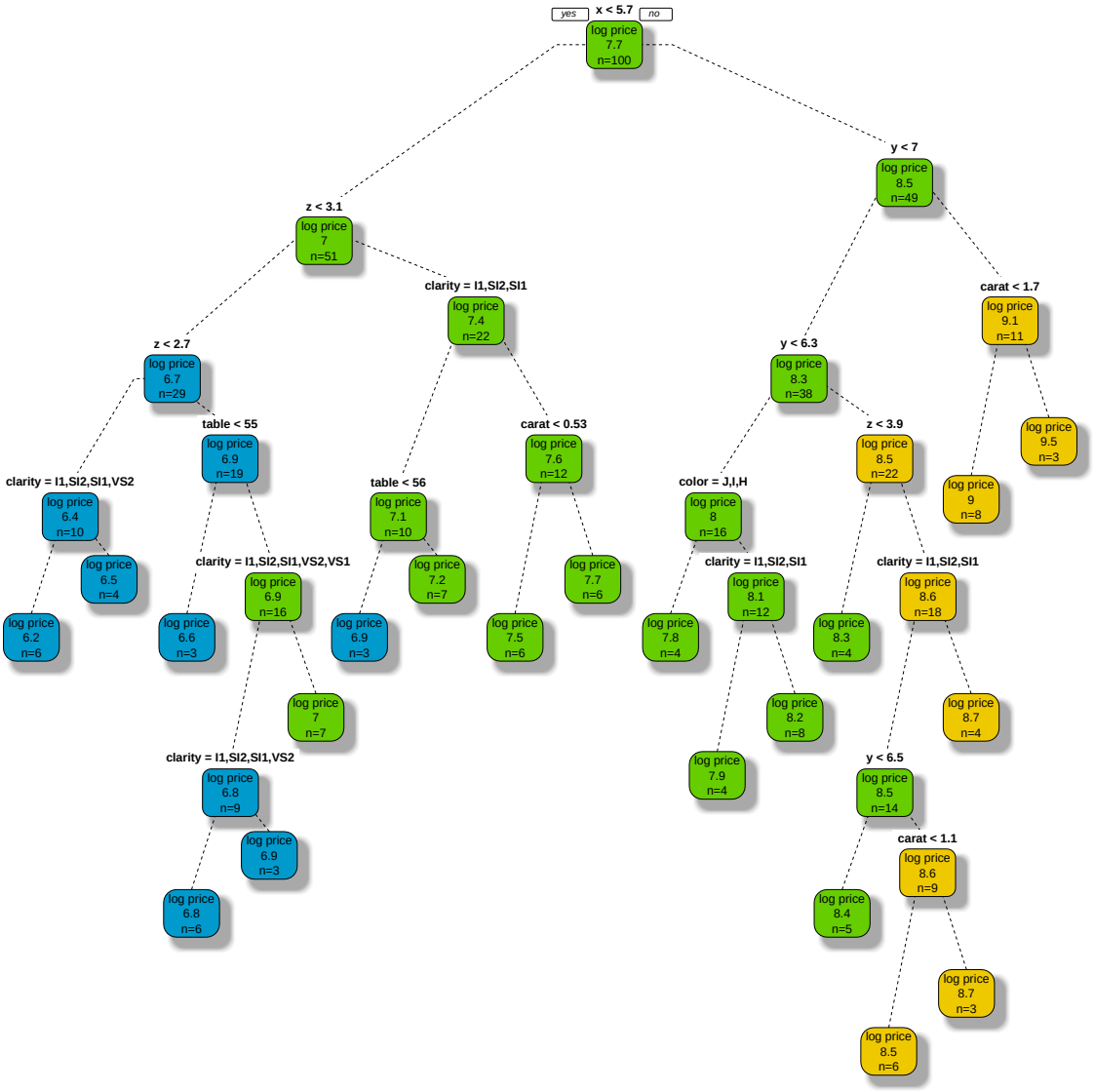

Figure 12: Simple regression tree of N = 100 random samples from the training data.

A random forest is an ensemble of hundreds or even thousands of these trees. To build each tree, a bootstrap sample (a sample with replacement) the same size of the training data is taken from the training data. The tree is then recursively grown the normal way a regression tree is grown except that each terminal node is split only considering m variables out of the p total number of variables. The best variable/split point is picked among the m variables and the tree is recursively grown until the minimum node size min_n is reached. Final predictions are made by averaging all of the trees (Hastie, Tibshirani and Friedman 2009, 588). A random forest has the same bias as an individual sampled tree and improvements in prediction are due solely to a reduction in variance (Hastie, Tibshirani and Friedman 2009, 600 - 601).

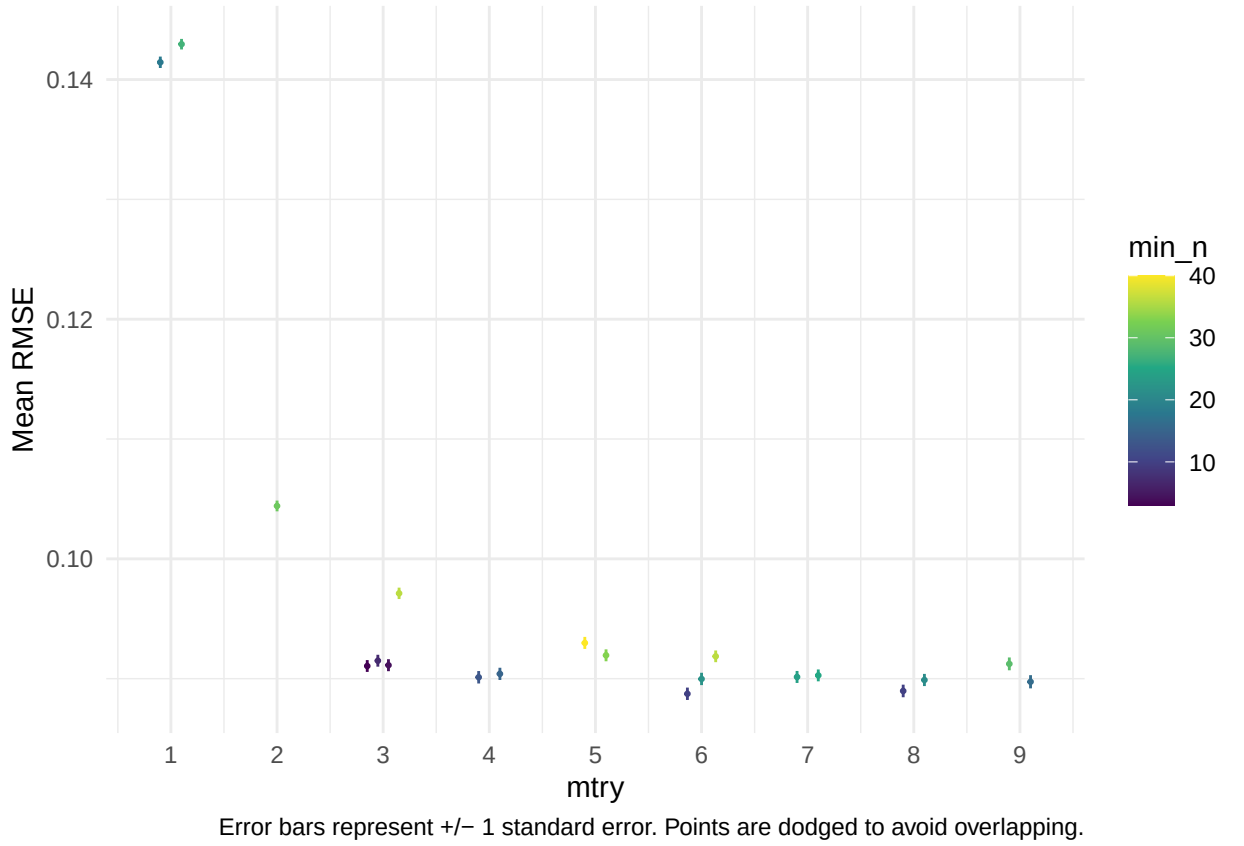

Figure 13: Evaluation of the random forest models built using various values of mtry and min_n.

It was apparent for the initial random forest models tested using cross-validation that higher values of m (mtry) and lower values of min_n produced better predictions for the diamonds (Figure 13). A finer grid of values for these parameters was tested again on the same cross-validation folds (Figure 14) and the final random forest model was chosen with an mtry = 6 and a min_n = 5. This model had a cross-validation mean RMSE of 0.0885 with a standard error of 0.000526. A separate, final measure of the estimated test error was computed after training this random forest model on the entire training set using the out-of-bag (OOB) samples. The OOB predictions are the predictions made on observations using only the regression trees grown that did not contain the observation in the bootstrap sample. The OOB error estimate is almost identical to an N-fold cross-validation and in this case was exactly identical with the OOB estimate of the RMSE equal to 0.0885 (Hastie, Tibshirani and Friedman 2009, 592-593). This trained model was then used to make predictions on the diamonds in the test data set.

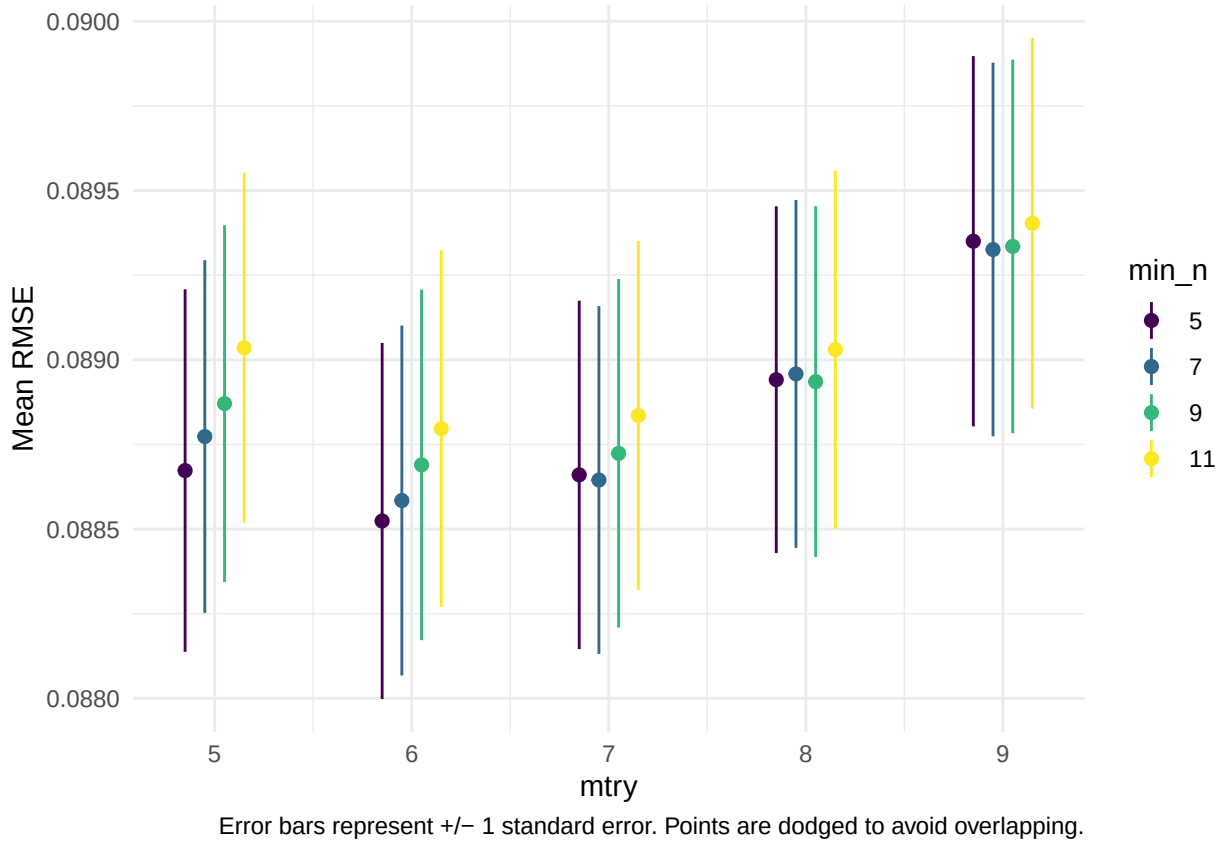

Figure 14: Evaluation of the tuning parameters for the random forest model focusing on higher values of mtry and lower values of min_n.

Results

Accuracy of the Model

The final RMSE calculated using the test set was 0.0867 log US Dollars which means the model made better predictions for the diamonds then the cross-validation approach originally assumed it would. To fully understand how well the model predicted the price of the diamonds in the data set, the predictions were transformed back into US dollars and rounded to the nearest dollar to match the format of the original price of the diamonds. With the predictions back in their original format, the RMSE was calculated to be $514. As seen in figure 15, the model performed better for lower priced diamonds, but fared worse as the price of the diamond increased. There also appeared to be a few lower priced diamonds where the model did not perform as well as it did for similarly priced diamonds. The predictions for the observations in the test set that contained missing values did not appear to be extremely different than other observations close in price with maybe the exception of the highest priced observation containing missing values.

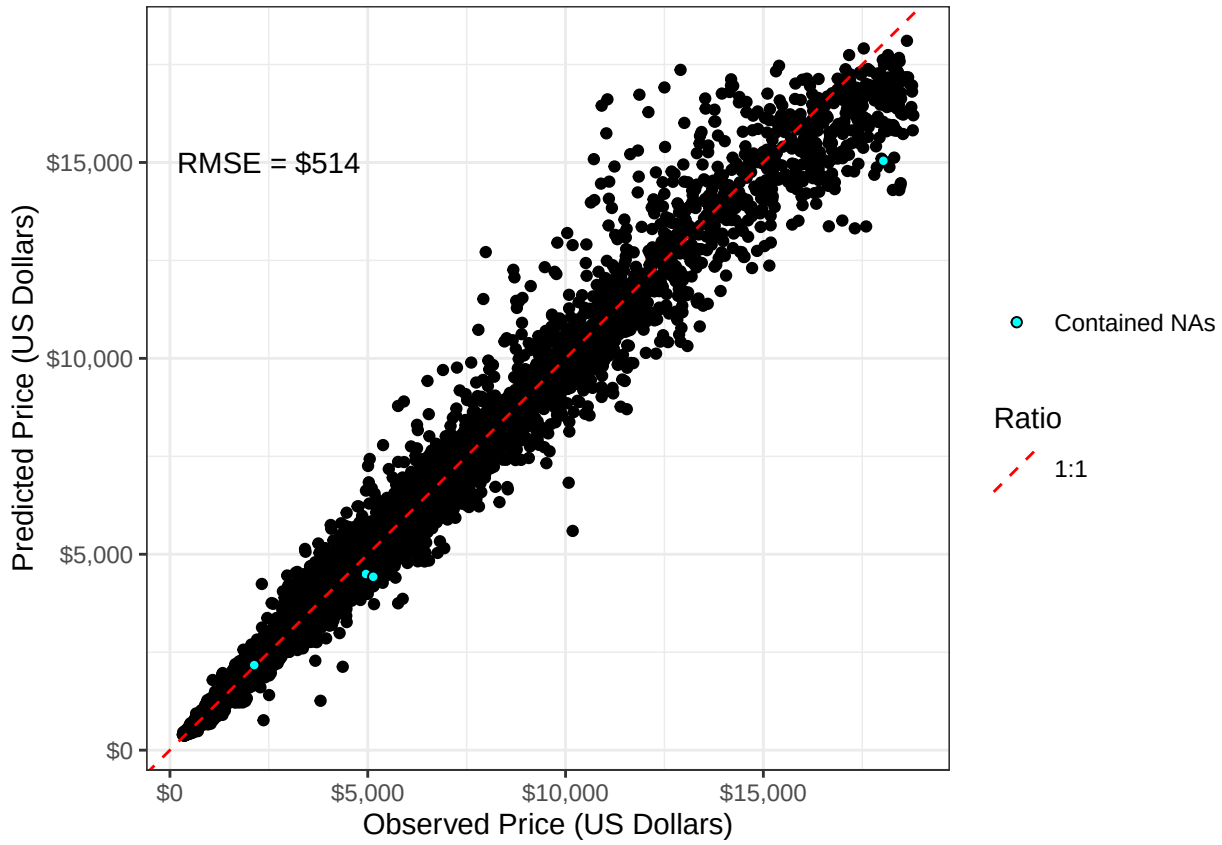

Figure 15: Predicted vs observed prices in the original units (US dollars). Observations in the test set that contained missing values are noted.

To further understand how well the model predicted the price of the diamonds, the distribution of the absolute difference between the predicted and observed prices was examined (Figure 16). It was found that this distribution had, rounded to the nearest dollar, a sample standard deviation of $444.9, a sample mean of $257.5 with a standard error of $12.2, a sample median of $93.0, and a sample 75th percentile (the point at which 75% of the data is at or smaller than this value) of $286.0. Based on this measure it appeared that the model worked relatively well for a significant portion of the data and further cemented the notion that a particular subset of the predictions fared much worse than others.

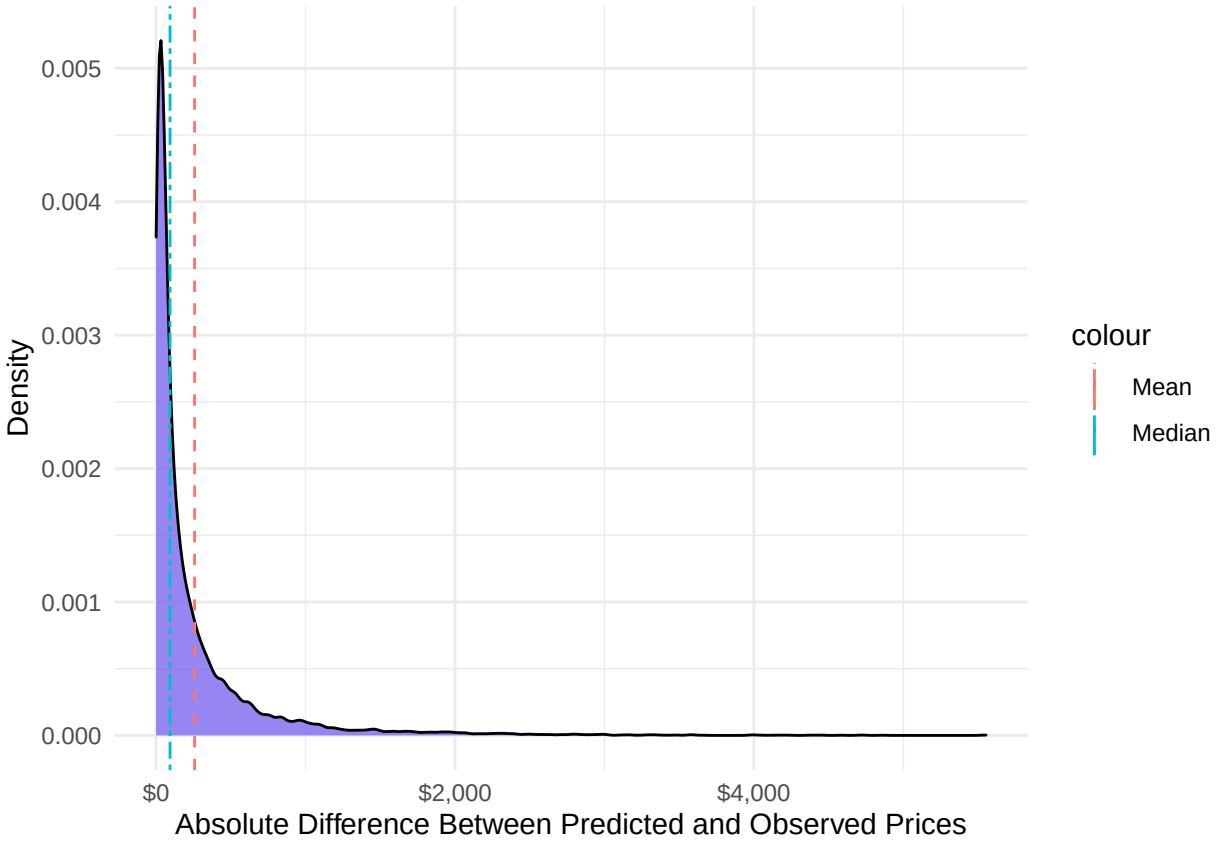

Figure 16: Distribution of the absolute difference between the predicted and observed prices.

In total, there were only 6 predictions that exactly matched the price of the observed diamond. Although the absolute difference provided insight into how close each prediction was to the original price point, practically speaking an absolute difference of $100 may be worse for a $400 diamond than a $8000 diamond depending on the buyer or seller. The percent error between the observed and predicted diamond prices can be calculated as:

$$\%error = \frac{Price_{predicted} - Price_{observed}}{Price_{observed}}$$

It was thought that this calculation may be more useful for some buyers and sellers of round cut diamonds as there may be an acceptable range for the percent error of diamonds' prices predicted using this model. Also, the distribution of this calculation proved useful in determining if the model typically undervalued the diamonds or overvalued the diamonds (Figure 17). For this distribution, the sample standard deviation was 8.73%, the sample 10th percentile was -9.06%, the median was -0.39%, the mean was 0.27% with an estimated standard error of 0.09%, and the sample 90th percentile was 10.30%. Overall, it appeared that the model tended to slightly undervalue the diamonds. The symmetry of the distribution with the middle being extremely close to zero indicated that this model produced relatively reasonable predictions.

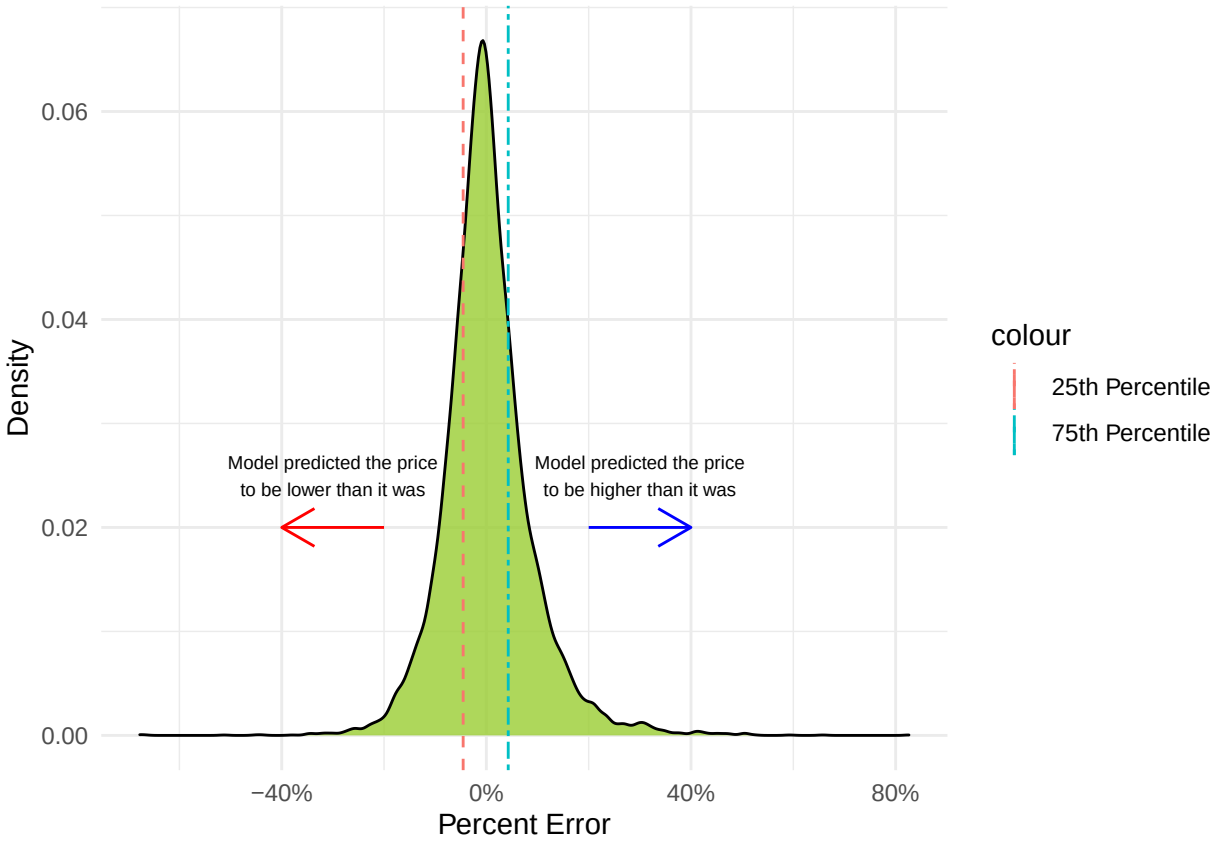

Figure 17: Distribution of percent error between the predicted and observed prices.

As the original data set, the training set, and the test set mostly contained lower priced diamonds and lower to medium priced round cut diamonds may be more routinely bought and sold for the use of everyday jewelry, it was thought that the accuracy of the random forest model should be examined for certain price ranges. Tables 5 and 6 split the original prices of the diamonds into three categories based on the price of the observed diamonds: price < $5000, $5000 <= price < $10000, and price >= $10000. As expected, the predictions for the diamonds < $5000 performed better overall compared to the other two categories when it came to the absolute difference between predicted and observed prices. Notably, 65.4% of the diamonds

Table 5: Prediction error for each price group along with the proportions in each data set and the proportion of observations whose absolute difference falls under a certain limit.

Price Group	% in Diamonds	% in Train	% in Test	RMSE	|Diff| < $10	|Diff| < $50	|Diff| < $100	|Diff| < $200	|Diff| < $300
price < $5000	72.7%	72.7%	72.6%	$217	11.7%	45.8%	65.4%	81.9%	89.4%
$5000 <= price < $10000	17.6%	17.5%	17.9%	$632	2.3%	9.5%	17.4%	33.7%	48.9%
$10000 <= price	9.7%	9.7%	9.5%	$1292	1.1%	3.9%	9.8%	17.2%	24.5%

Table 6: Proportion of observations for each price group whose absolute percent error falls below a certain limit.

Price Group	|%error| < 1%	|%error| < 5%	|%error| < 10%	|%error| < 15%	|%error| < 20%
price < $5000	13.7%	56%	81.9%	92.3%	96.7%
$5000 <= price < $10000	12.5%	53.3%	80.3%	92.2%	96.2%
$10000 <= price	11.4%	47.8%	76.8%	89.8%	95.9%

in this category had an absolute difference of less than $100. Also, 48.9% of the diamonds in the middle category had an absolute difference of less than $300. When it came to the the absolute percent error, over 50% of the diamonds in the lower and medium price categories had an absolute error of less than 5%. Interestingly enough, over 75% of the diamonds in each category had an absolute error of less than 10%.

Based on the information above, this model was reasonably accurate for a majority of the diamonds in the test set. There may need further improvements to account for outlying data points and higher priced diamonds. This predictive model may prove most useful for traders who are more likely to deal with diamonds usually in the lower to medium price range and will rarely if ever see a diamond in the higher price range.

Importance of the Predictors

All the predictors were kept in the random forest model in case there were any complex interactions between the variables that may lead to better predictions. However, the depth (total depth percentage) and table (table percentage) seemed unlikely to help on their own based on the previous bivariate plots between these variables and price. The importance of these variables and the other predictors in this random forest model were studied using a *permutation variable-importance measure* constructed from the OOB samples. When the bth tree is grown the prediction accuracy is recorded from the OOB samples. The values for the jth variable are then randomly permuted in the OOB samples and the prediction accuracy is again recorded. This decrease in accuracy is measured for every tree and averaged over all the trees (Hastie, Tibshirani and Friedman 2009, 593). This permuted average decrease in accuracy is shown for all variables in figure 18 with the OOB prediction error measure being the MSE for the natural log of the original prices. Table and depth were the least important variables in this model with cut only having a slightly larger importance than depth or table. Notably, the most important categorical variable in this model appeared to be clarity. From the information gathered, predictive models built on this or similar data may benefit from excluding table and depth.

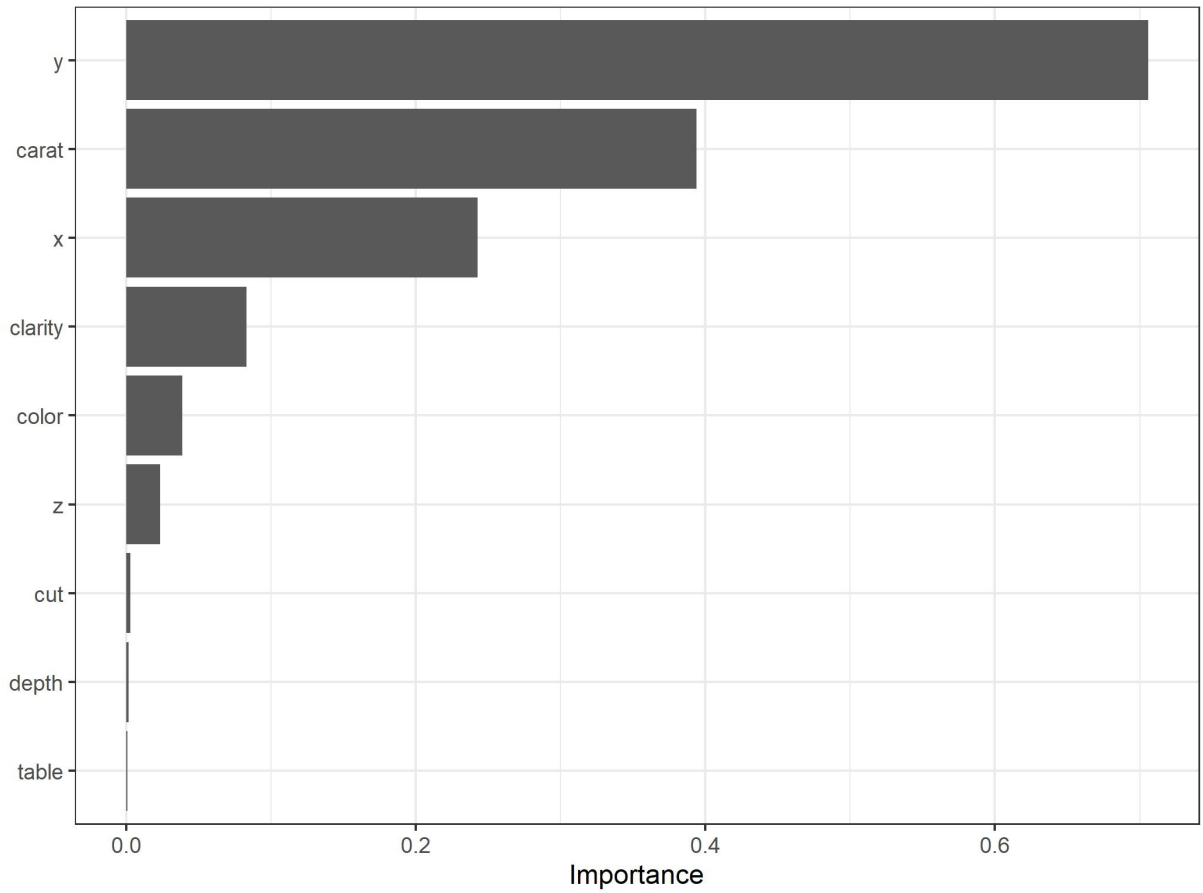

Figure 18: Permutation importance plot for the fitted random forest. Importance is measured using the average decrease in OOB MSE.

Conclusion

The random forest model created was a relatively accurate model to predict the price of round cut diamonds. It appeared to perform considerably better than other linear based models. It predicted the price of the diamonds to be within $286 of the original price for 75% of the data in the test set. Also, the absolute percent error of the predictions was less than 10% for 81% of the predictions. The model worked better for lower priced diamonds than it did for higher priced diamonds. Future models may need to use better outlier detection and handling methods to deal with this issue. Also improvements could be possibly made even with the current model's parameters if a larger sample set containing more of these higher priced diamonds is found. As the random forest model was a tree ensemble model, other similar models such as an extreme gradient boosted tree model may also improve accuracy of the predictions.

It was also found that the table percentage and the total depth percentage were not particularly important when it came to predictions. Future research should look into possibly excluding these predictors when predicting the price of round cut diamonds.

It was the hope of the author that this model proves useful in both helping to predict round cut diamonds and to act as a stepping stone for future models on similar data.

References

- "About GIA Facetware®." Accessed March 20, 2021. https://www.gia.edu/facetware-about.
- "AGS Diamond Charts | Find Diamond Rating, Scales, & Grading Charts." *American Gem Society.* Accessed May 2, 2021. https://www.americangemsociety.org/buying-diamonds-with-confidence/ags-diamond-grading-system/.
- Blodgett, T, Geurts, R, Gilbertson, A, Lucas, A, Pay, D, Reinitz, I, Shigley, J, Yantzer, K, Zink, C. *Estimating a Cut Grade Diamond Using the GIA Diamond Cut Grading System.* Gemelogical Institute of America. 2009.
- "Diamond Anatomy, Explained." *GIA 4Cs,* April 9, 2014. https://4cs.gia.edu/en-us/blog/diamond-anatomy-explained/.
- "Diamond Inclusions Defined." *GIA 4Cs,* November 25, 2013. https://4cs.gia.edu/en-us/blog/diamond-inclusions-defined/.
- Hastie, Trevor, Robert Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* 2nd ed. Springer Series in Statistics. New York, NY: Springer, 2009.
- "History of the 4Cs of Diamond Quality." *GIA 4Cs,* October 17, 2017. https://4cs.gia.edu/en-us/blog/history-4cs-diamond-quality/.
- "How Diamonds Shape Up." *GIA 4Cs,* June 19, 2012. https://4cs.gia.edu/en-us/blog/how-diamonds-shape-up/.
- Kuhn, Max, and Kjell Johnson. *Feature Engineering and Selection: A Practical Approach for Predictive Models,* 2020.
- "Learn to Calculate Diamond Prices So You Don't Get Ripped Off." *The Diamond Pro.* Accessed May 1, 2021. https://www.diamonds.pro/education/diamond-prices/.
- Pisani, Bob. "The Billion Dollar Business of Diamonds, From Mining to Retail." *CNBC,* August 27, 2012. https://www.cnbc.com/2012/08/27/the-billion-dollar-business-of-diamonds-from-mining-to-retail.html.
- "Sample Natural Diamond Reports." Accessed May 1, 2021. https://www.gia.edu/analysis-grading-sample-report-diamond?reporttype=diamond-grading-report&reporttype=diamond-grading-report.
- Zheng, Yuleng. *An Introduction to R, LaTeX, and Statistical Inference.* May 21, 2020. https://bookdown.org/Yuleng/polimethods/

Software

- Barret Schloerke, Di Cook, Joseph Larmarange, Francois Briatte, Moritz Marbach, Edwin Thoen, Amos Elberg and Jason Crowley (2021). *GGally: Extension to 'ggplot2'.* R package version 2.1.1. https://CRAN.R-project.org/package=GGally
- Brandon M. Greenwell and Bradley C. Boehmke (2020). *Variable Importance Plots—An Introduction to the vip Package.* The R Journal, 12(1), 343–366. URL https://doi.org/10.32614/RJ-2020-013.
- Hao Zhu (2021). *kableExtra: Construct Complex Table with 'kable' and Pipe Syntax.* R package version 1.3.4. https://CRAN.R-project.org/package=kableExtra
- H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

- Jeroen Ooms (2021). *magick: Advanced Graphics and Image-Processing in R.* R package version 2.7.1. https://CRAN.R-project.org/package=magick
- Kuhn et al., (2020). *Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles.* https://www.tidymodels.org
- Microsoft Corporation and Steve Weston (2020). *doParallel: Foreach Parallel Adaptor for the 'parallel' Package.* R package version 1.0.16. https://CRAN.R-project.org/package=doParallel
- R Core Team (2021). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.
- RStudio Team (2020). *RStudio: Integrated Development for R.* RStudio, PBC, Boston, MA URL http://www.rstudio.com/.
- Stephen Milborrow (2020). *rpart.plot: Plot 'rpart' Models: An Enhanced Version of 'plot.rpart'.* R package version 3.0.9. https://CRAN.R-project.org/package=rpart.plot
- Thomas Lin Pedersen (2020). *patchwork: The Composer of Plots.* R package version 1.1.1. https://CRAN.R-project.org/package=patchwork
- Wickham et al., (2019). *Welcome to the tidyverse.* Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686