Diamonds: Predictive Analysis of Price

Francisco Arrieta, Emily Schmidt and Lucia Camenisch

21 December 2022

Contents

1 Introduction

The following report provides the result of using machine learning as a tool to estimate diamond prices for a jewelry company. Using snapshot information from their asset database, various methods were applied to train predictive models using regression analysis. The main objective was to use supervised learning methods to predict prices. These models are later compared to measure their effectiveness by using error measurements to quantify the distance between the prediction and the actual price. Finally, conclusions on the capability of each model are made and suggestions are given on which model to apply for the problem at hand.

The product being analyzed is diamonds. Being a luxury object with a long history, industry standards have been developed to serve as guidelines for estimating the value of the product. A wide variety of characteristics affect the overall price of diamonds, but this analysis will focus on their physical qualities, such as size dimensions, size ratios and color. Other factors inherent to scarce products in high demand of a capitalist economy will not be considered in our machine learning exercise.

The choice of methods shown in the report is not all inclusive and responds to the fact of only being some of the most used methods to this day. By implementing them, the analysts hope to provide a glimpse of the effects and importance in choosing the right model as well as displaying the differences between each one.

2 Data Exploration

In the early stages of any analysis, data exploration is a critical process aimed at understanding and analyzing the data set to gain insight and make valid decisions. The overarching goal of examining the data is to obtain intuition, identify questionable values, and strategize how to answer the problem statement. Therefore, let's summarize the findings that explain the main data characteristics, and dive into the relations between variables.

2.1 Dimension Summary

There is a variety of ways an analyst could approach understanding the observations in the data set. For starters, there are 53,940 records and 10 variables.

- `price`: The response variable which is measured in USD and ranges between $326 to $18,823
- `carat`: Continuous value that gives information about the weight of the diamond (1 ct = 200 mg)
- `cut`: Quality of the circular cut and has five levels (worst: `Fair` | best: `Ideal`)
- `color`: The less color within the diamond the better (yellow-ish: `J` | clear: `D`)
- `clarity`: A categorical value that measures how clear the diamond appears without impurities (worst: `I1` | best: `IF`)
- `x`: A numeric variable that relates to the diamond `length` in millimeters (mm), which ranges from 0.00 mm to 10.74 mm
- `y`: A numeric variable that relates to the diamond `width` in millimeters (mm), which ranges from 0 mm to 58.90 mm
- `z`: A numeric variable that relates to the diamond `depth`/height in millimeters (mm), which ranges from 0 mm to 31.80 mm
- `depth total`: A calculated value that is computed with `x`, `y`, and `z` in terms of a percentage. This feature does not exceed 79.00%
- `table`: The width of the top of diamond relative to widest point that can be seen when the stone is viewed face up

The outcome variable (price) is a an integer. There are an additional six numeric values and three ordinal/categorical features that have a factored structure. Those non-numeric values (`cut`, `color`, and `clarity`) need to be appropriately ordered, and renamed if there are any spaces within their naming convention as later this could obstruct certain code.

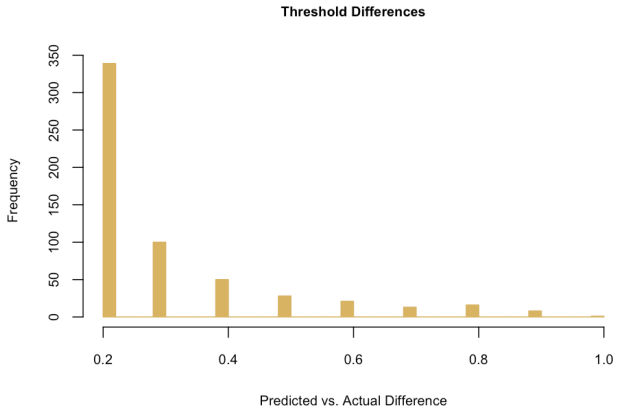

Figure 1: Amount of records by difference between computed depth ratio and actual depth ratio

To dive deeper, we consider removing records that are invalid because they add noise. For instance, price against all features were reviewed for possible impurities. In total, there are 275 (0.5%) rows removed for the following reasons:

1. 20 rows had a depth of 0.00. This was considered inaccurate because a diamond needs to have this dimension specified.
2. The difference between length and width should be almost identical since the diamond should be cut approximately circular. If they were not, those rows were removed. In this case, only two records were not included in the data set at differences above 36.00 mm.
3. The depth_ratio is a calculation between `length` (x), `width` (y), and `depth` (z). To investigate this feature further, our group computed `depth_ratio` and saw there were differences between the actual and predicted values. Therefore, the executive decision was to remove any differences above a threshold of 0.3. A total of 253 rows were removed.

In the remaining 53,918 observations, there were no missing values.

In continuation of exploring the diamonds dataset, we now look at the relationships between variables. By isolating price against the other continuous variables, it is seen that `carat`, `length`, `width`, and `depth`, all have a positively linear relation to the response feature. Some of the models used in the analysis to predict price will show these effects later as a possible issue of collinearity could occur. In addition, the relationship between carat and price is unique because their relation is non-linear. We were also interested in showing how the categorical variables affect price. `Clarity`, `color`, and `cut` have distinct patterns seen between the four continuous variables mentioned above.

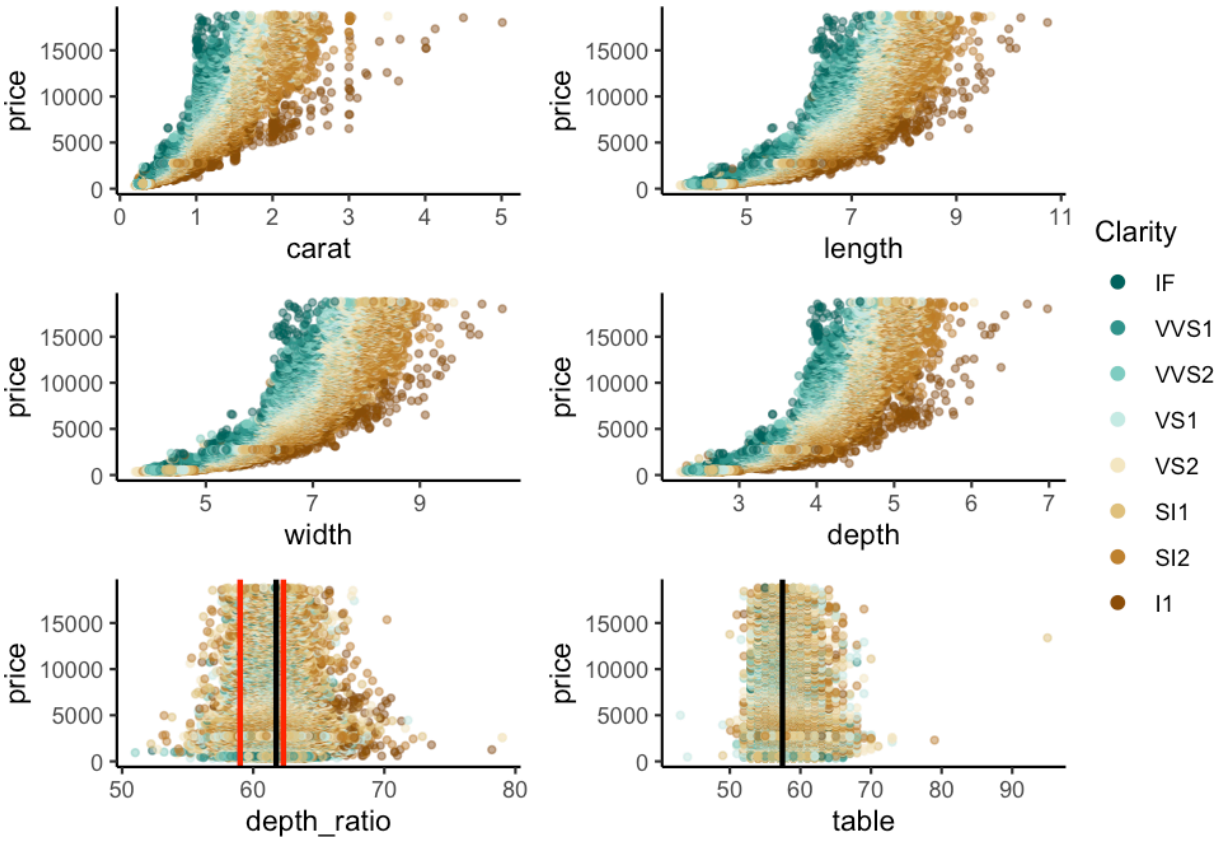

Figure 2: Price with respect to numerical variables, by Clarity

When plotting `price` by `carat`, `width`, `length`, and `depth`, with `clarity` as the color dimension, there is a clear distinction between how these observations relate. Each scatter plot shows a strong, positive correlation to `price`. When reviewing the x-axis for each explanatory variable, the lower tiered option is typically observed more. The cheaper the gemstone, the worse quality. Within each plot, there are outliers that can be explained by how the diamond was cut, which impacts the `weight`, `length`, and `depth`. By the color scale, it is noticed how there are more diamonds with a worse clarity (SI2/SI1) than there are with the best (IF/VVS1). Intuitively, this makes sense because perfecting a diamond's clarity is quite difficult. Therefore, a majority of the diamonds lie between the second and third best, and second and third worst clarity factors. In addition, the relation between `depth_ratio` and `table` to `price` is not distinct as those features are mixed throughout the relative ranges. Within all of the `depth_ratio` plots, there are vertical lines that show the `depth_ratio` mean (black) and range (red) of what is considered the prime value for `depth_ratio` as long as it is above 59.00% but does not exceed 62.30%.

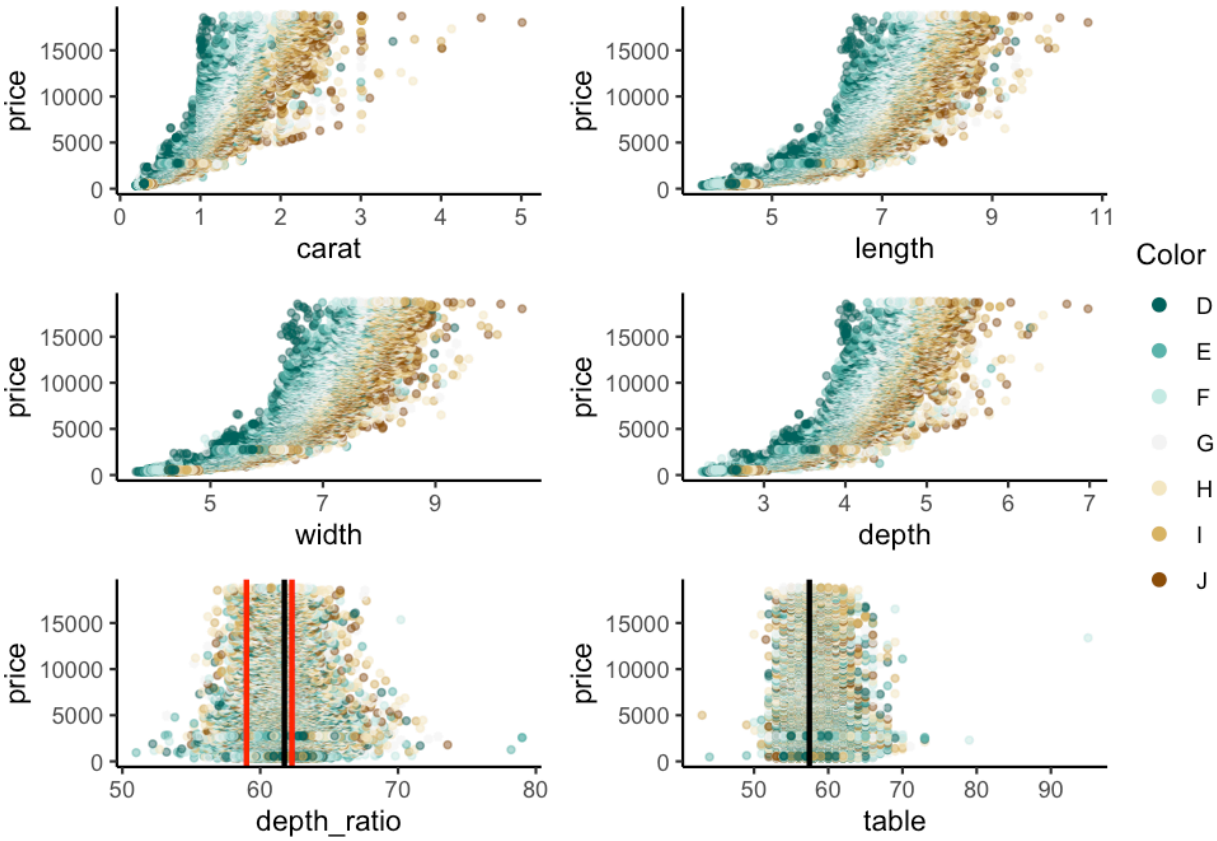

Figure 3: Price with respect to numerical variables, by Color

A similar relationship can be seen regarding `price` by the continuous variables, but as `color` as the color dimension. Instead of a majority the observations falling closer to the best and worst `clarify` features, here the reader will notice how it appears that the colors E, F and G dominate (the white to light blue) the space.

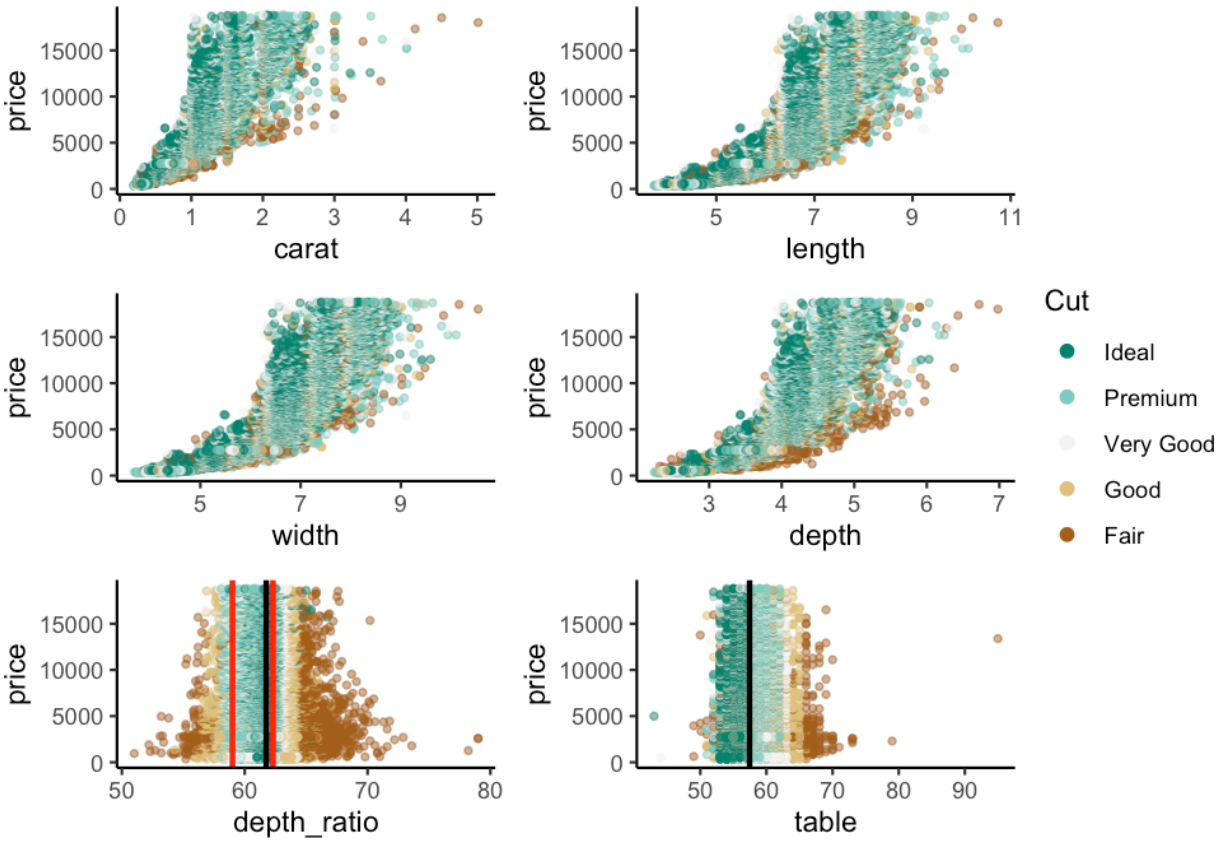

Figure 4: Price with respect to numerical variables, by Cut

`Cut` has completely different trends than `clarity` and `color` as a majority of the diamonds are either `ideal` or `premium`. This can especially be seen within the `depth_ratio` plot. As stated earlier, the red lines indicate the optimal space in which the luxury brand attempts to get the best ratio. Most of all of the prime cuts are within those bounds while the `fair`, good, and `very good` are outside. There is a clear distinction on how `cut` relates to the `depth_ratio` along `price`.

2.2 Variable Visualisation

This section involves creating graphical representations of data to visually communicate insights and trends. It allows us to quickly and easily identify patterns that may not be immediately apparent from looking at raw data. There are several graphics that demonstrate how these qualitative and quantitative variables are distributed. Since some models take into account transformation, those details will be described later.

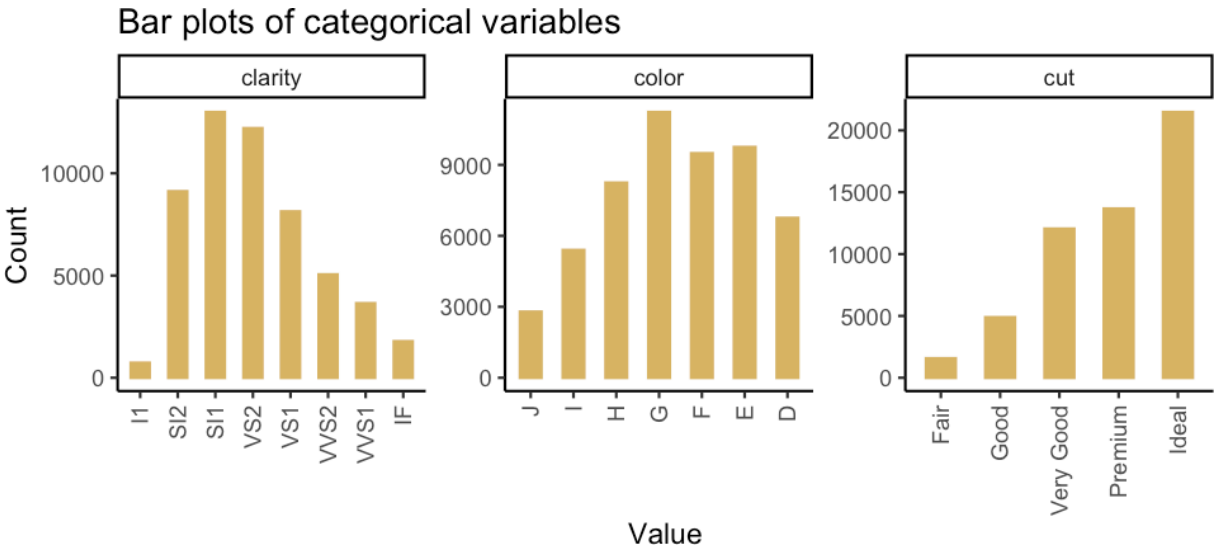

Figure 5: Bar plots of categorical variables

Recall the variable descriptions in the beginning of the exploratory data analysis.

- `clarity`: After the worst clarity (`I1`), the bulk of diamonds range from `SI2` and `VS1`. This correlates to the scatter plot with the goldish toned points as there appear to be more of those diamonds than ones with better clarity.
- `cut`: The less yellow the diamond, the better. The dataset shows that most diamonds are either G, F, or E, which are the closest three to the least colored diamond.
- `color`: There are over 20,000 diamonds that have an `ideal` cut which `premium` and `very good` between 10,000 and 15,000.

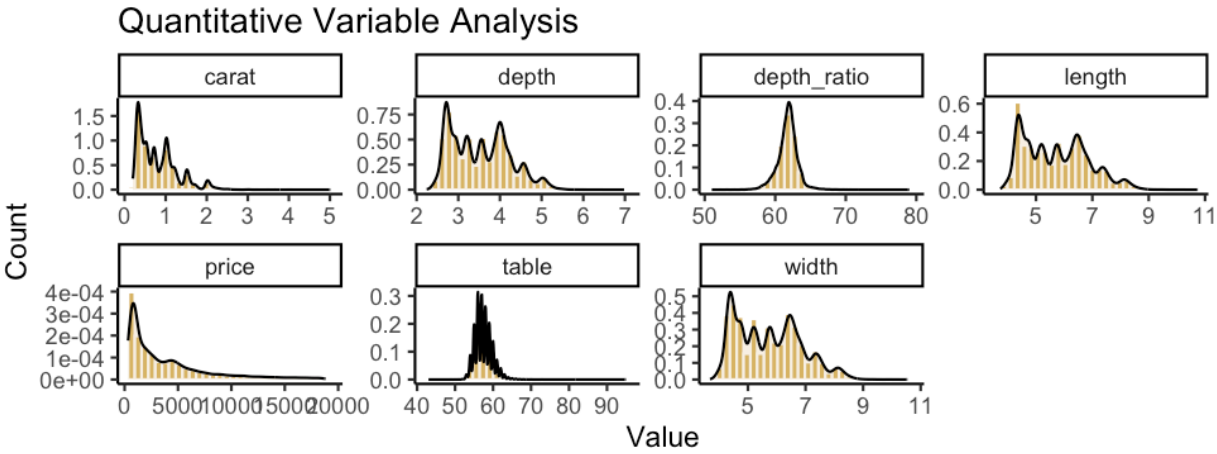

Figure 6: Histograms of numerical variables

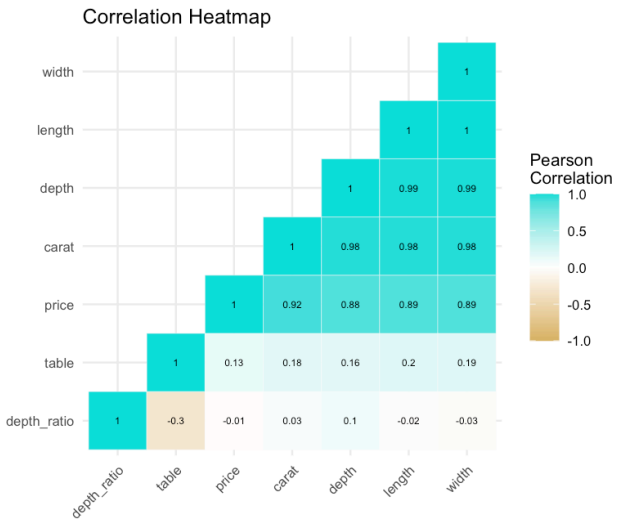

Figure 7: Correlation heatmap of numerical variables

At a high-level, `price` and `carat` can be seen as positively skewed. In the scatter plot regarding `price` by `cut` for the `depth_ratio`, the histogram also displays the same conclusion that the cut features fall into approximately 60 to 65 mm. The relationships in the Correlation Heat Map visually show how `width`, `length`, `depth`, and `carat` continue to be highly correlated with `price` while `depth_ratio` and `table` are closer to 0.

Through exploration, we can gain a deeper understanding of the data and how it can be used to answer business questions or solve real-world problems, like predicting the price of a diamond.

3 Variable Prediction and Model Performance Evaluation

3.1 Linear Regression

3.1.1 Data preparation

We begin by performing linear regressions on our data. As we saw during the exploration phase, some predictors (`carat`, `length`, `width` and `depth`) are highly correlated. Therefore, multicollinearity might be an issue.

We use the Generalized Variation Inflation Factors (GVIF) to measure the multicollinearity level of our data. This generalized version of the VIF allows us to take into account numerical and categorical predictors together. The GVIF clearly confirms that there is an issue, as `length`, `width` and `depth` all have coefficients above 1000. `carat` and `depth_ratio` also have high values above 25, but they aren't as high extreme as the other three.

After removing `length`, `width` and `depth`, the GVIF coefficients of the remaining predictors are all under 2, which indicates the multicollinearity problem is solved. We display correlation ellipses of numerical variables.

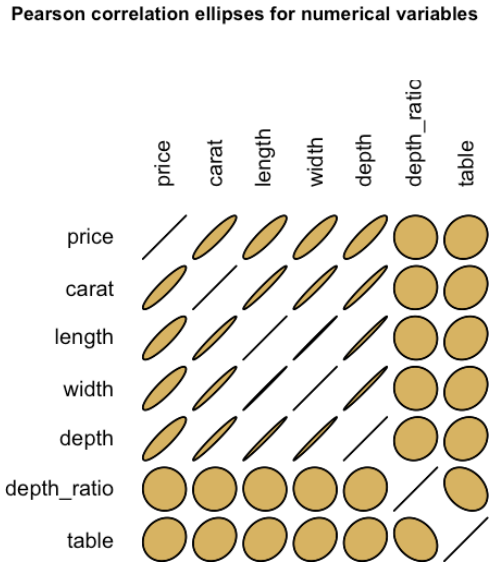

Figure 8: Correlation ellipses for numerical variables

Thus, we will perform linear regressions on two different models:

1. `LM_complete` which contains all predictors
2. `LM_minus_corr` which has `length`, `width` and `depth` removed.

These two models will serve as basis for variable selection procedures later.

However, before starting to build our models, we also need to account for skewed variables. Linear regression might perform worse when dealing with skewed variables and it is common to use transformations such as a logarithm or a nth root to make variables more symmetrical.

We use an estimator of skewness called b_1, whose definition can be found here.

The value of b_1 is interpreted as follows:

- $0 \leq |b_1| < 0.5$: variable is symmetrical;
- $0.5 \leq |b_1| < 1$: variable is moderately skewed;
- $|b_1| \geq 1$: variable is highly skewed.

We compute b_1 on our numerical variables and get the following results.

`price` and `carat` are highly skewed and `table` is moderately skewed.

Table 1: Skewness estimator for numerical variables

	b_1
price	1.6259747
carat	1.0967514
length	0.3984689
width	0.3922008
depth	0.3921093
depth_ratio	0.0182498
table	0.8728869

We apply a logarithmic transformation to all three variables. The improvement can also be seen in the histograms, as they look more symmetrical now.

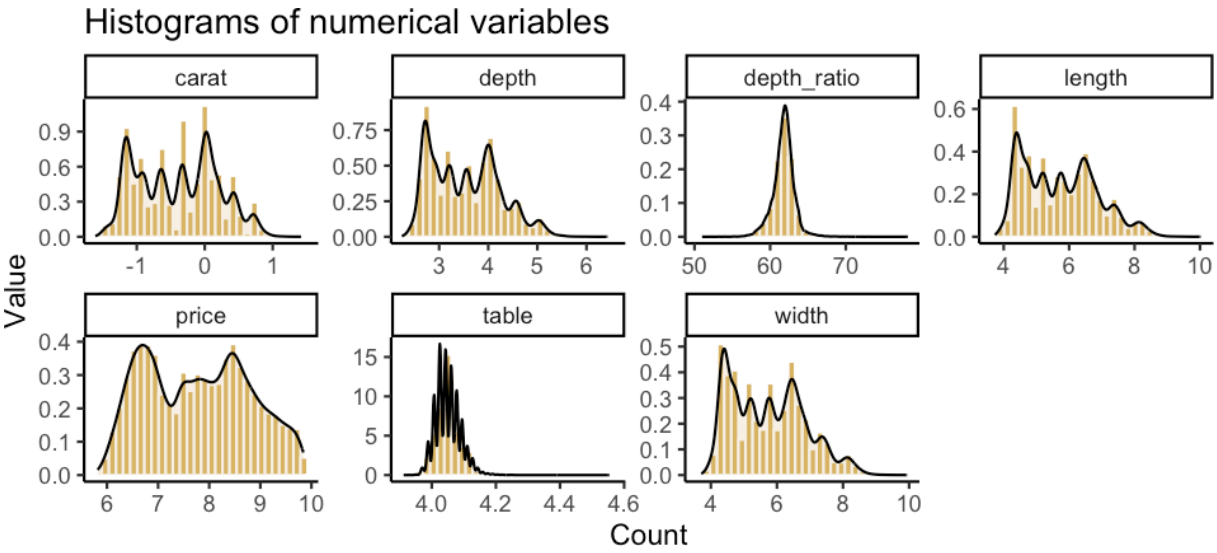

Figure 9: Histograms of numerical variables after unskewing

We also standardize numerical variables by subtracting their mean and dividing by their standard deviation. This makes the comparison of β coefficients between variables easier.

3.1.2 Linear models

The data is now ready for our linear models. For both `LM_complete` and `LM_minus_corr`, we perform the following linear regressions:

1. Linear regression on the whole model
2. Forward selection on the model (iterative method)

3. Backward selection on the model (iterative method)
4. Stepwise selection on the model (iterative method)
5. Mallows's C_p and AIC selection on the model (global method)

3.1.3 Summary of models

For each of these models, we summarise which variables are used as predictors in the following table.

Table 2: Predictors used in each linear model

Model	Cut	Color	Clarity	Carat	Length	Width	Depth	Depth Ratio	Table
LM_complete	X	X	X	X	X	X	X	X	X
LM_forward_complete	X	X	X	X	X	X	X	X	X
LM_backward_complete	X	X	X	X	X		X	X	
LM_stepwise_complete	X	X	X	X	X		X	X	
LM_CpAIC_complete	X	X	X	X	X		X	X	
LM_minus_corr	X	X	X	X				X	X
LM_forward_minus_corr	X	X	X	X				X	X
LM_backward_minus_corr	X	X	X	X				X	
LM_stepwise_minus_corr	X	X	X	X				X	
LM_CpAIC_minus_corr	X	X	X	X				X	

For both basis models, forward selection doesn't discard any variables, whereas backward, stepwise and global selections all choose the same model with less variables than initially.

Thus, we have four distinct models in total. We assess the predictive performance of these four models on our validation set by computing the five accuracy measures seen during the course.

Let us recall the definitions and meaning of the two main measures we will use (taken from Data Mining for Business Analytics - Concepts, Techniques, and Applications in R, chapter 5.2, page 119). We denote the residuals by $r = y - \hat{y}$.

1. **ME** (Mean Error) gives an indication of whether the predictions are on average over- or under-predicting the outcome variable.
$$ME = \frac{1}{n}\sum_{i=1}^{n} r_i$$
2. **RMSE** (Root Mean Squared Error) is similar to the standard error of estimate in linear regression, except that it is computed on the validation data rather than on the training data.
$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} r_i^2}$$

3.1.4 Accuracy measures of models on validation set

The accuracy measures for our four models are summarized below. In order to give meaningful results, the outcome variable `price` has been rescaled to its original scale and retransformed by taking the exponential (to cancel out the logarithm). Thus, the ME, RMSE and MAE can be interpreted in the price currency, dollars.

Table 3: Accuracy measures of linear models

Model	ME	RMSE	MAE	MPE	MAPE
LM_complete	36.36648	846.5888	408.5364	-0.8377036	10.33643
LM_CpAIC_complete	36.32774	845.5018	408.1349	-0.8408540	10.33659
LM_minus_corr	50.38994	810.1522	405.0486	-0.8420621	10.39559
LM_CpAIC_minus_corr	50.39878	810.1610	405.0616	-0.8418121	10.39568

The mean error is bigger in the models without multicollinearity issues, but their RMSE is smaller. Since the mean error is positive in all four models, we are under-predicting the price of diamonds by $36 or $50 on average depending on the model.

An increase of $14 in the mean error is a good trade-off to reducing the RMSE, which indicates how much the predictions will fluctuate from real values. The other three measures are quite close in all four models.

Considering the parsimony principle and the RMSE, the best choice is our fourth linear model, which contains `cut`, `color`, `clarity`, `carat` and `depth_ratio`.

Overall, the RMSE for linear regression remains quite high and as we will see in the following chapters, some models will achieve much better predictive performances.

3.2 k-NN

3.2.0.1 Overview k-nearest neighbors (k-NN) is a simple algorithm used for classification (categorical) and regression (continuous). Since the response variable (`price`) is a numeric outcome, this analysis will use k-NN to predict the `price` of a diamond. In a regression setting, the algorithm relies on finding the most 'similar' records in the training data. Then, one would calculate the weighted average of the numerical target of the k-nearest neighbors for the data point.

According to the Data Mining textbook, the value of k is based on a nonparametric method since it 'draws information from similarities between the predictor values of the records in the dataset.' There are several ways one could choose this hyperparameter, but this analysis will focus on one method by the optimization of RMSE. k is a critical input within the k-NN function because it determines how

many neighbors will be considered when making a prediction. Penn State states that a 'larger k leads to a smoother boundary but may also introduce noise.' On the other side, a small k 'can increase the complexity of k-NN.'

The k-nearest neighbors algorithm is very useful when predicting `price` because the data does not need to be transformed or predictors selected in a particular way.

3.2.0.2 Model Structure There are several steps that need to be completed prior to running the first k-NN model with k = 1. First, you clean the data to ensure that missing values are either removed or explained, and that all variables are essential to the response variable (see EDA section). Next, the data is normalized so that the output remains unbiased. Normalizing simply means that the raw data is put on all the same scale ('subtracting the mean and dividing by the standard deviation'). Once the data is cleaned and normalized, it can be split into the training and validation sets. Now that the foundation of the model has been built, let's start constructing the first k-NN model.

The value of k = 1 is used as a starting point to see how well the model will perform. The RMSE is of particular interest because that gives the standard deviation of how far the predicted value is away from the actual price. Seen in the k-NN error table below, ME ($20.60) and RMSE ($954.60) could potentially get better if we optimized k.

The optimal k is 4 because it produces the lowest RMSE at $814.59. After rerunning the k-NN model, the results improved in one aspect, but not the other.

Table 4: RMSE value with respect to k

k	1.0	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.0	11.0	12.00	13.00	14.00	15.00	16.00	17.00	18.00	19.00	20.00
accuracy	954.6	861.09	827.12	814.59	819.29	818.55	822.76	833.21	839.93	847.4	850.7	854.83	860.49	867.35	874.86	879.91	886.07	891.98	896.35	900.09

Table 5: Accuracy measures of models with k = 1 and k = 4

Model_kNN	ME	RMSE	MAE	MPE	MAPE
k = 1	20.59665	954.5954	487.1664	-1.833533	13.73317
k = 4	40.86142	814.5940	431.2913	-2.452640	12.13511

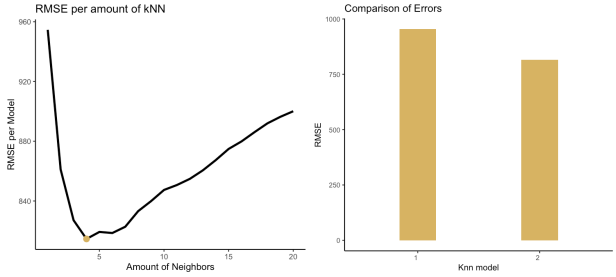

Figure 10: Evolution of RMSE with respect to k and difference of RMSE for k = 1 and k = 4

As stated earlier, the table above shows how the ME got worse for k = 4, but better with RMSE once the model was optimized. In this analysis, one of the main goals is to reduce the variation between predicted and real price. Therefore, the difference of $140.00 is more important even when the ME slightly increases. Between k = 1 and k = 4 for the k-NN model, k = 4 with an RMSE of $814.60 will predict prices better.

3.3 Decision Trees

There are four methods within this section that show various ways on how trees are built. They include a regression tree, boosted tree, bagged tree, and a random forest. The regression tree is most transparent and easy to interpret while the other three combine results from multiple trees.

3.3.0.1 Regression Tree Overview A regression tree is a flexible data-driven method that can be used for prediction of a continuous variable. The tree separates 'records into subgroups by creating splits on predictors. These splits create logical rules that are homogeneous.' Those splits 'divide the data into subsets, that is, branches, nodes, and leaves. Like decision trees, regression trees select splits that decrease the dispersion of target attribute values. Thus, the target attribute values can be predicted from their mean values in the leaves' which reduces the variance of the target variable.

3.3.0.2 Model Structure and Analysis Since the tree proactively takes into consideration the most important attributes to split on, multiple trees are created to check its validity. This idea stemmed from wanting to ensure that by removing variables due to their weak relationship to the predictor variable or by pruning, the results produced the best ME and RMSE. Within the three regression models, all have the same error rates. In addition, there are eight terminal nodes for each model and all had `width`, `length`, `carat`, and `depth` being the most important features to predict `price`. To exemplify this, we show the original regression tree that first splits on `width` < 6.33, seven splits total and uses the following for the primary splits: `width` < 6.325, `carat` < 0.985, `length` < 6.325 and `depth` < 3.935.

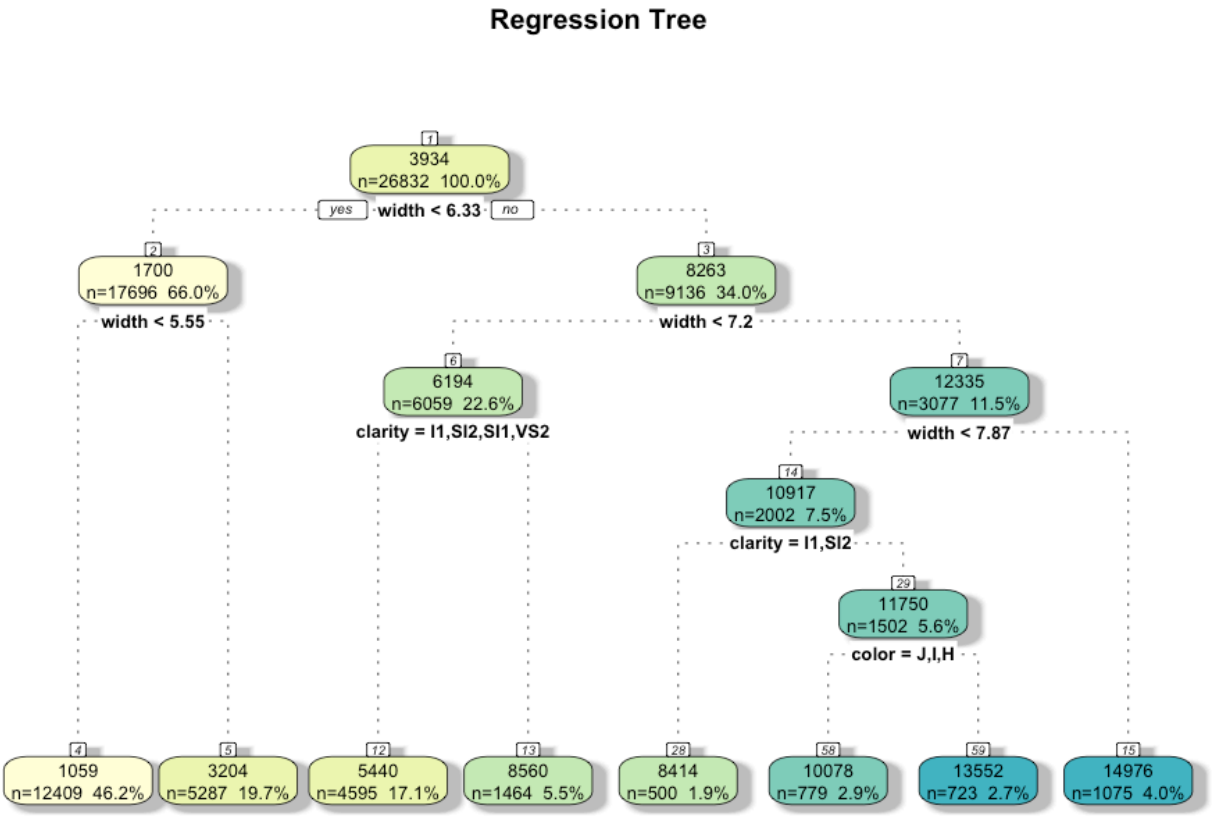

Figure 11: Regression tree diagram

3.3.1 Boosted Tree

3.3.1.1 Overview Boosted trees are a type of ensemble model that 'transforms weak decision trees into strong learners. Each new tree is built considering the errors of previous trees.' The idea behind boosting is to train a series of weak models additively, with each model attempting to correct the errors made by the previous model. Since this model is prone to overfitting, the parameters need to be carefully considered so that the lowest error rates can be achieved.

3.3.1.2 Model Structure and Analysis There is a wide variety of parameters that can be chosen for a boosted tree. Therefore, those options were used differently to choose the best model based off the number of predictors and how deep each tree would be allowed to interact. The main difference between each model is the number of trees ran. As the size of the tree increases, the better the model because it only considered the most important factors.

The best boosted model was the one with 100 trees and three cross-validation folds. The tree used six predictors to produce a model with a ME of $9.00 and RMSE of $1,170.30. What is fascinating about

this tree comes from the importance variables chart. In the regression tree, `clarity` was one of the three least important factors, but here we see that it as the fourth which is above `length`.

3.3.2 Bagged Tree

3.3.2.1 Overview The third method is bagged trees. This type of model 'combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model.' The objective with bagging is to train a large number of decision trees on different subsets of the training data, and then average the predictions of all the trees to make a final prediction. This model has the ability to reduce variance because it introduces randomness into the training process.

3.3.2.2 Model Structure and Analysis This model is pretty straight forward because the best model was built by sticking with the basics. 'The only parameters when bagging decision trees is the number of samples and hence the number of trees to include. This can be chosen by increasing the number of trees on run after run until the accuracy begins to stop showing improvement.' Overall, the bagging tree predicts `price` better than the regression tree, but worse than boosting. The ME was the lowest at $0.93, but what is more important to consider is the RMSE which is $1,248.77.

3.3.3 Random Forest

3.3.3.1 Overview 'Random forests are a special case of bagging, a method for improving predictive power by combining multiple classifiers or prediction algorithms.' They are an ensemble based on bagged trees which involves training each tree on a bootstrapped sample of the original data.

3.3.3.2 Model Structure Similar to the bagged trees, random forests are pretty simplistic. The main focus in this section was to check how many trees should be run within the model. There is a trade-off between computational power and RMSE. For example, what is the difference in RMSE if the model was built off of 100 trees versus 60? The model with 100 trees was the best model with an RMSE of $575.42.

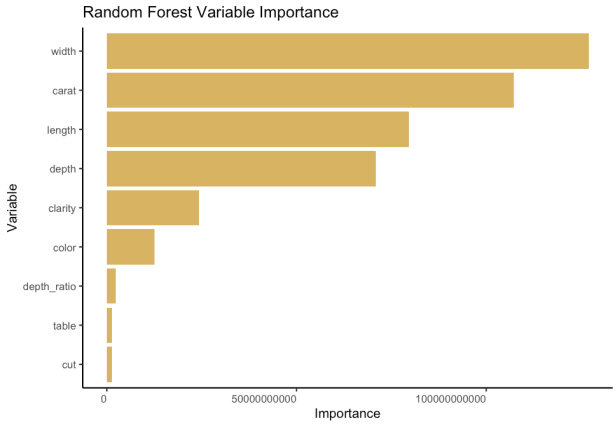

Figure 12: Variable importance for random forest

Throughout this entire section, typically the lower RMSE produced the best model. Additionally up to this point, there have been three or four variables that were most importance to the model. In the random forest though, there is much more weight on features such as `clarity` and `color`. In random forests, we proposed a trade-off with picking a model with less trees because an analyst would want to choose the parsimonious model. Therefore, the random forest with 60 trees had a slightly higher RMSE at a difference of approximately $2 which gave a $577.63 for the RMSE.

3.3.4 Decision Tree Summary Table

Table 6: Accuracy measures of decision tree models

	Model_Tree	ME	RMSE	MAE	MPE	MAPE
Reg. Tree	Reg. Tree	6.1	1267.15	846.93	-14.54	33.08
Exclus. RT	Exclus. RT	6.1	1267.15	846.93	-14.54	33.08
Pruned RT	Pruned RT	6.1	1267.15	846.93	-14.54	33.08
Boost10	Boost10	-3.9	3660.42	2765.96	-144.49	171.65
Boost30	Boost30	2.54	1531.52	985.7	-35.26	46.86
Boost100	Boost100	9	1170.3	680.22	-12.85	26.22
Bagging	Bagging	0.93	1248.77	808.93	-14.35	31.45
RndmFrst	RndmFrst	2.97	577.63	289.51	-1.4	7.24

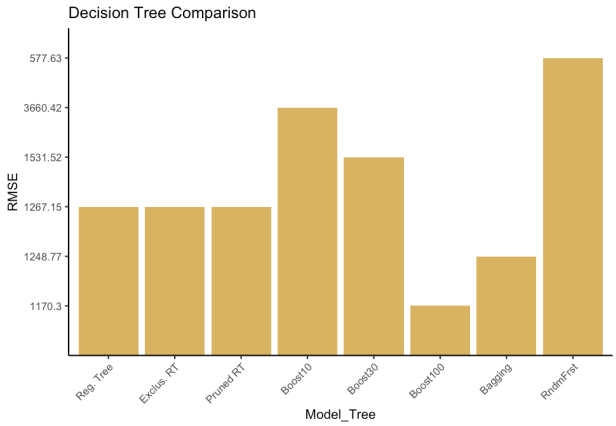

Figure 13: RMSE of decision trees

Throughout the four methods mentioned, the random forest model proved to be the most effective in predicting `price` with a RMSE of $577.63, which is more than two times smaller than the best boosting model. Although not all of the models results are shown, a breakdown of their error results can be seen within the table above. Overall, the regression tree, bagging and boosting were all within $100 RMSE difference, but no match for the random forest.

3.4 NeuralNetworks

3.4.1 Basic Concept

Artificial neural networks are a method of machine learning that receives the name due to a comparison made with actual neurons in the human brain. They consist various nodes that communicate with each other to predict an outcome based on the relationships that were determined in the process. Neural nets have proven to be very good at predicting values, through regression or classification and have been the center of much research in the past years. Though this method was once considered un useful, with the improvement of computational power, it has acquired new popularity for prediction of images, speech recognition and non-explicit trends. (Hardesty, 2017)

The basic parts of a neural networks are: the structure, composed of layers and nodes, the weights, and the activation function. The first consists of the different components used to train a neural network, ie the explanatory variables and their connection. These independent variables are considered the *input nodes* and the outcome variable the *output node*. The layers in between these to are called the *hidden layers* as these are used to calculate the output but have no easily association of there value with regards to the outcome. This is why neural networks are considered by some like a **black box**, where the exact method for predicting is not easily explained to those who are not well informed. The weights are used to pass a certain amount of a value to the next node which after passing through

the activation function will pass on to the next, and so on. This weighting can be assigned randomly or by specific methods, depending on the problem at hand and analyst discretion. Similarly, the activation function calculates the position in a curve, ie the expected value of the prediction. This as well changes per prediction problem and analyst discretion. While some are considered better for certain tasks there is no limiting factor in the way a neural net is structured.

3.4.2 Data preprocessing

Due to the fact that in every node we calculate the position of the prediction in a curve, the scale of the values used affects the output of every node. Moreover, since we use all types of variables for prediction in neural networks (continuous and categorical) the difference in amplitude is very important. This is why it is standard procedure to normalize the data so they are all in the same scale.

3.4.3 Model Structures

There is no standard or ideal way of setting up the amount of layers in a network, but common rule of thumb is to start with the same amount of input variables and then reduce to see if this improves. (Shmueli, et al. 2018: 286) In this case, it was decided to follow the following structures:

- The first model is composed of all the variables (26 explanatory) and a single hidden layer of one node. All models have only one output node as the desired outcome is a single prediction of price. This model is considered to be the most basic and should in theory have the least predictive performance.
- The second model includes a hidden layer of 26 nodes, so it equals the input nodes.
- To see if there is an improvement, we do another model with just 13 nodes in a single hidden layer.

Of these single layer models, the best is the one with 26 nodes. We measure this by looking at the error in prediction, also called the accuracy. There are many ways of measuring this, but as mentioned before, the two we looked at are the Mean Error, as a measure of accuracy, and Root Mean Squared Error, as a measure of precision. For the 26 node model, the RMSE is $745.16 vs $744.04 for the 13 node net. While just looking at this configuration one could infer using less nodes is better, after multiple tests it is concluded that keeping the 26 nodes as the first hidden layer is best for reducing the RMSE going forward.

The next configuration tested is adding 2 additional hidden layers, one of 26 additional nodes, and another of 13. Just by using this configuration, the model RMSE reduced to $627.65. As this is already a rather low RMSE, in comparison to previous neural networks, as well as other models seen above,

it was decided to keep this structure as the definitive one for establishing the most accurate prediction.

The next two adjustments made are regarding the weights and the activation function. This structure was adjusted to use the "*Glorot Normal*" weight initiation method. This initiation method consists of assigning weights to the nodes using a de probability curve of a truncated normal distribution as so:

Glorot Normal Initialization:

$$w_i \sim Gaussian\left(\mu = 0, \sigma^2 = \sqrt{\frac{2}{u_{in} + u_{out}}}\right)$$

where:

u_{in} = number of input nodes

u_{out} = number of output nodes

This initiation method demonstrated much better results than others such as the normal distribution, uniform distribution of the Glorot Uniform, which is why we kept it for the final model.

Finally regarding the activation function, many options are available as well. Depending on the problem at hand, one can use different functions that draw different curves and hence produce different predictions. The one used for our case was a *sigmoid* activation which by different literature is the most commonly used due to its good performance results. Though we tested others like *"relu"* or "*softplus"*, sigmoid ended being the most accurate predicting.

3.4.4 Neural Net Summary Table

Table 7: Accuracy measures of neural network models

	ME	RMSE	MAE	MPE	MAPE
L1 N1	-246.53	1175.41	890.47	-53.75	63.14
L1 N26	-88.66	760.74	465.61	-11.44	18.28
L1 N13	-90.67	749.84	452.97	-11.41	17.76
L2 N26	22.48	636.79	364.38	-4.75	12.82
L2 N26 G	-45.24	662.5	383.93	-8.36	14.19
L2 N26 G LR	-51.05	591.6	340.12	-5.26	11.16

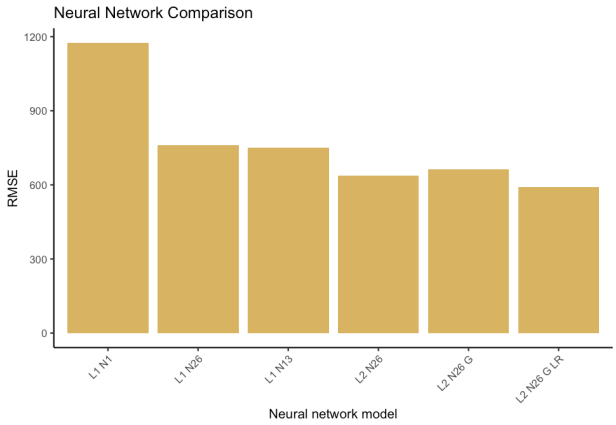

Figure 14: RMSE of neural networks

As can be seen, the predictive performance of the neural networks can vary a lot depending on the structure and parameters chosen. The negative point in this method is the requirement of human input to adjust the model to learn more precisely. Nevertheless, once the many trials have been executed, the results are very positive for a good predictive model.

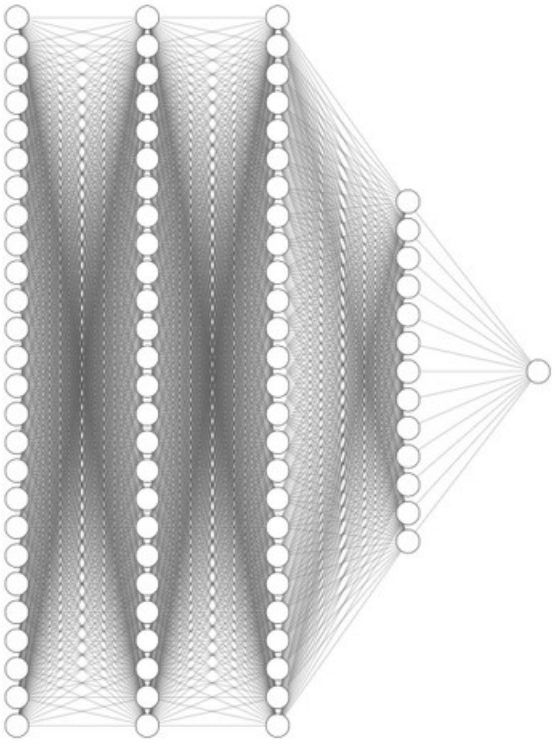

Figure 15: Visual description of Neural Network

3.5 Ensembles

3.5.0.1 Overview, Model Structure and Analysis 'Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model.' The simplest approach is to combine predictions from each method above that had the best/lowest RMSE. To accurately compare models, the mean is taken over the four models to compute the average predicted `price` and find the error rates between the calculated `price` and real prices. As the table reveals, the RMSE is $588.34 which is pretty close to random forests and the best neural network in terms of variance.

Table 8: Accuracy measures of five main models

	ME	RMSE	MAE	MPE	MAPE
Multiple Linear Regression	50.4	810.16	405.06	-0.84	10.4
Random Forest	2.97	577.63	289.51	-1.4	7.24
k-Nearest Neighbor	40.86	814.59	431.29	-2.45	12.14
Neural Network	-51.05	591.6	340.12	-5.26	11.16
Ensemble	10.8	583.21	308.04	-2.49	8.67

4 Conclusion

After reviewing the results and performance of each model in the previous section, we can conclude that the best models are the Random Forest and Neural Network. This is because they have the lowest error measures that we take into consideration throughout this analysis. In addition, more can be deducted by looking at the errors plotted in a graph.

4.1 Plots of Predicted Price

We can see how the predicted diamond price behaves with respect to the real price by utilizing each of the methods and mapping the three categorical variables. This recognizes the models' respective precision. When there are more observations closer to the diagonal line, the precision is higher. Within the scatter plots, the Random Forest and the Neural Network are more condensed. On the other hand, k-NN and MLR are more sparse. A similar situation happens when adding each of the categorical values as a color differentiator.

4.1.1 Predicted Price vs Real Price by Clarity

The multiple linear regression seems to have more issues predicting price of diamonds with better `clarity` in the mid to high priced diamonds. However, there is generally no clear discernible pattern per `clarity`. The abundance of the browner colored points is due to the existence of mid to low level clarities in the data set.

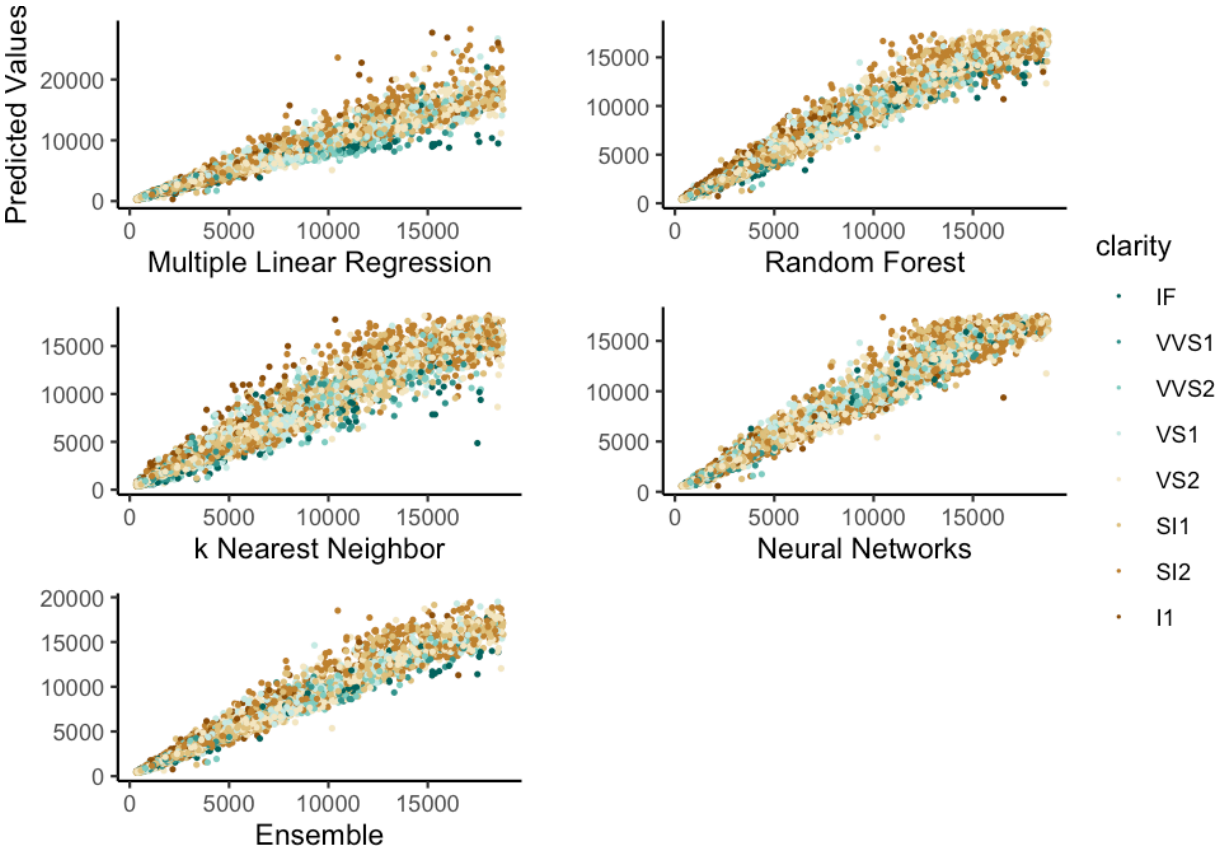

Figure 16: Predicted Price vs Real Price by Clarity

4.1.2 Predicted Price vs Real Price by Color

With regards to `color`, we see more of the more teal-colored points. However, there is still no pattern that indicates better or worse performance per `color` type.

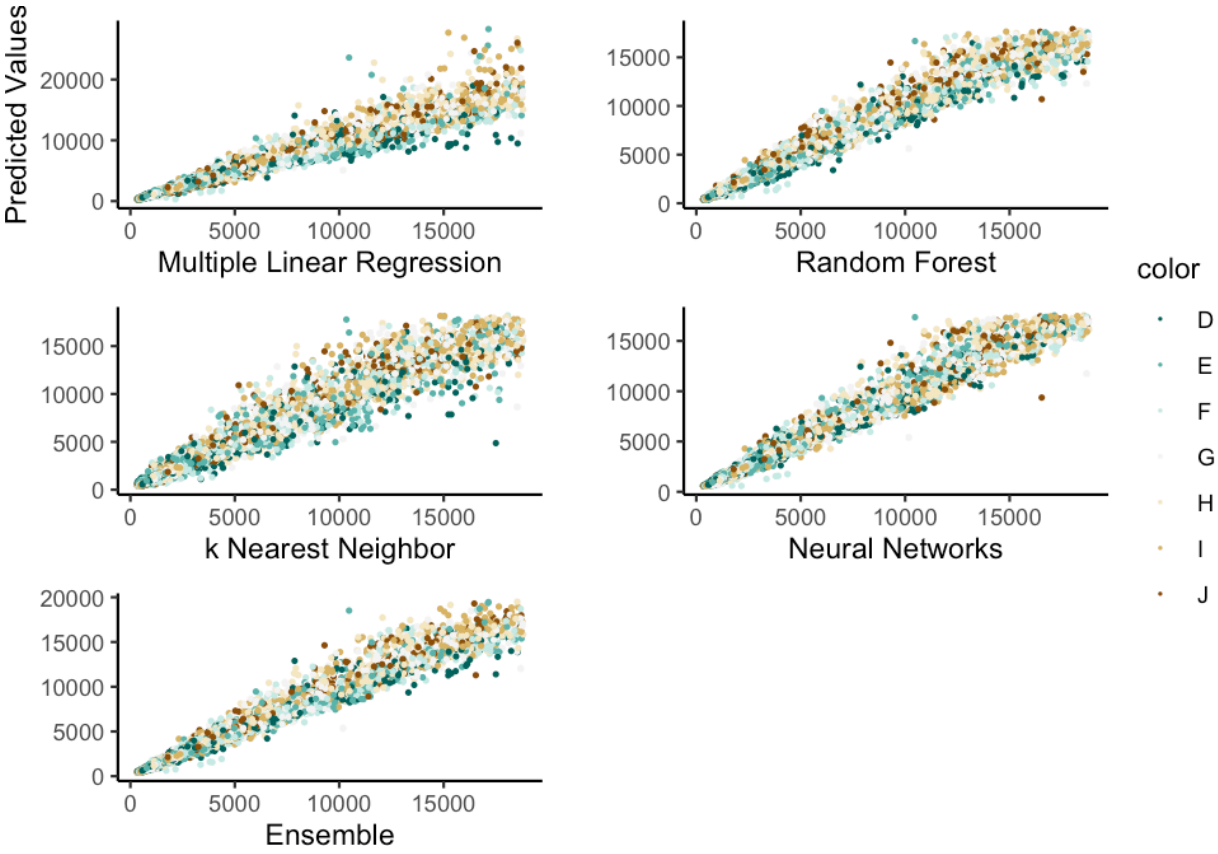

Figure 17: Predicted Price vs Real Price by Color

4.1.3 Predicted Price vs Real Price by Cut

Finally, the `cut` variable shows more `Ideal` and `Premium` cuts. Following the same trend as `clarity` and `color`, there is no pattern for the particular variable.

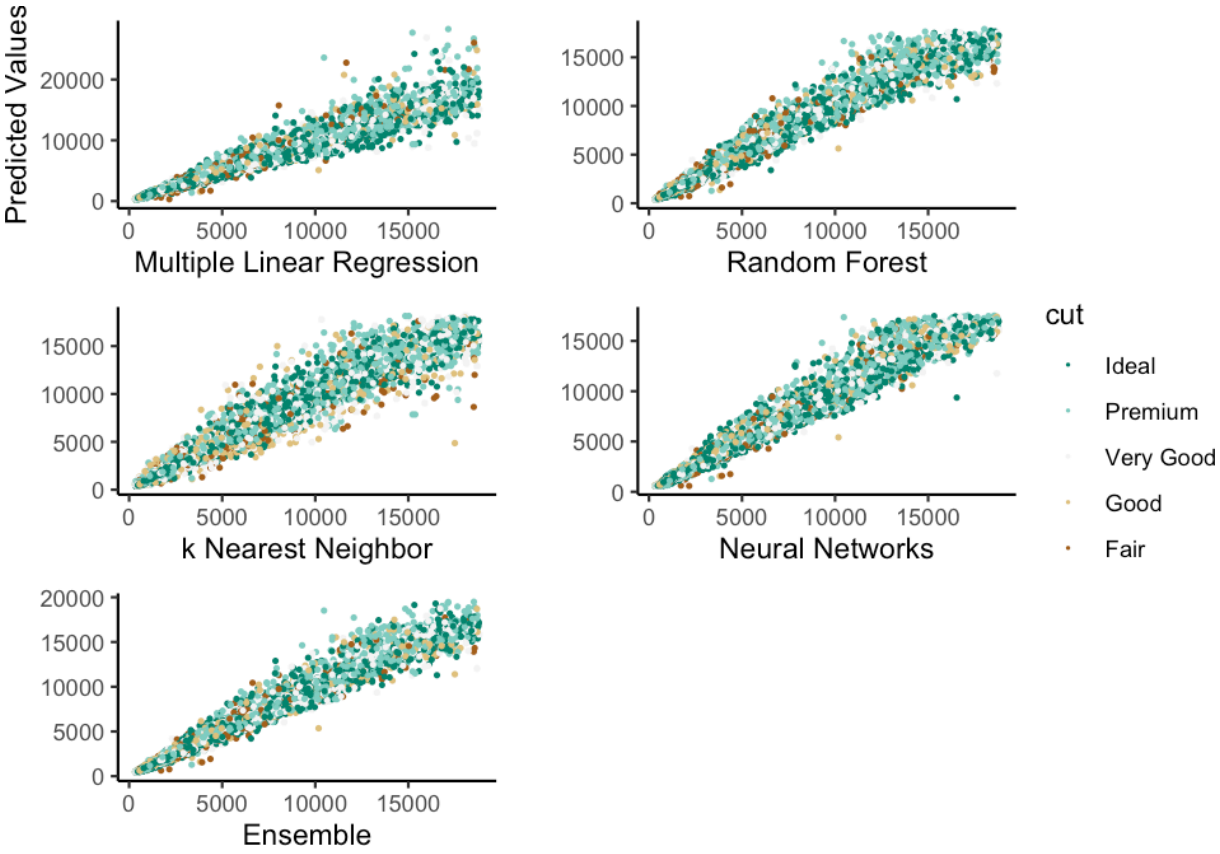

Figure 18: Predicted Price vs Real Price by Cut

The previous graphs show no discernible pattern regarding the categorical variables. Therefore, the performance of the Random Forest and the Neural Network is evident with smaller dispersion of the points. As the difference in error is quite close between these two models, our final model decision comes down to a matter of analyst preference or computational power available. We consider, however, that due to the smaller ME, our recommendation is to use the Random Forest for predicting diamond prices.

5 References

Shmueli et al. (2018). *Data Mining for Business Analytics*. Wiley.

Hardesty, Larry. (2017). [Explained: Neural Networks] MIT News. (https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414)

EDA

- https://r-graph-gallery.com/199-correlation-matrix-with-ggally.html
- http://www.sthda.com/english/wiki/ggplot2-quick-correlation-matrix-heatmap-r-software-and-data-visualization

k-NN

- https://online.stat.psu.edu/stat508/lesson/k#:~:text=The%20larger%20k%20is%2C%20the,Nearest%20Neigh
- https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

Regression Tree

- https://www.ibm.com/docs/en/db2-warehouse?topic=procedures-regression-trees

Boosted Tree

- https://towardsdatascience.com/introduction-to-boosted-trees-2692b6653b53
- https://leonlok.co.uk/blog/decision-trees-random-forests-gradient-boosting-whats-the-difference/

Bagging Tree

- https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/

Ensemble

- https://towardsdatascience.com/ensemble-methods-in-machine-learning-what-are-they-and-why-use-them-68ec3f9fef5f