ST 3082

Group 10

ADVANCED ANALYSIS ON Diamond Price Prediction

s15089 - Sanjani Wickramasinghe
s14982 - Poornima Dissanayake
s15006 - Pasindu Sachintha
s14953 - Buddhima Senarathna

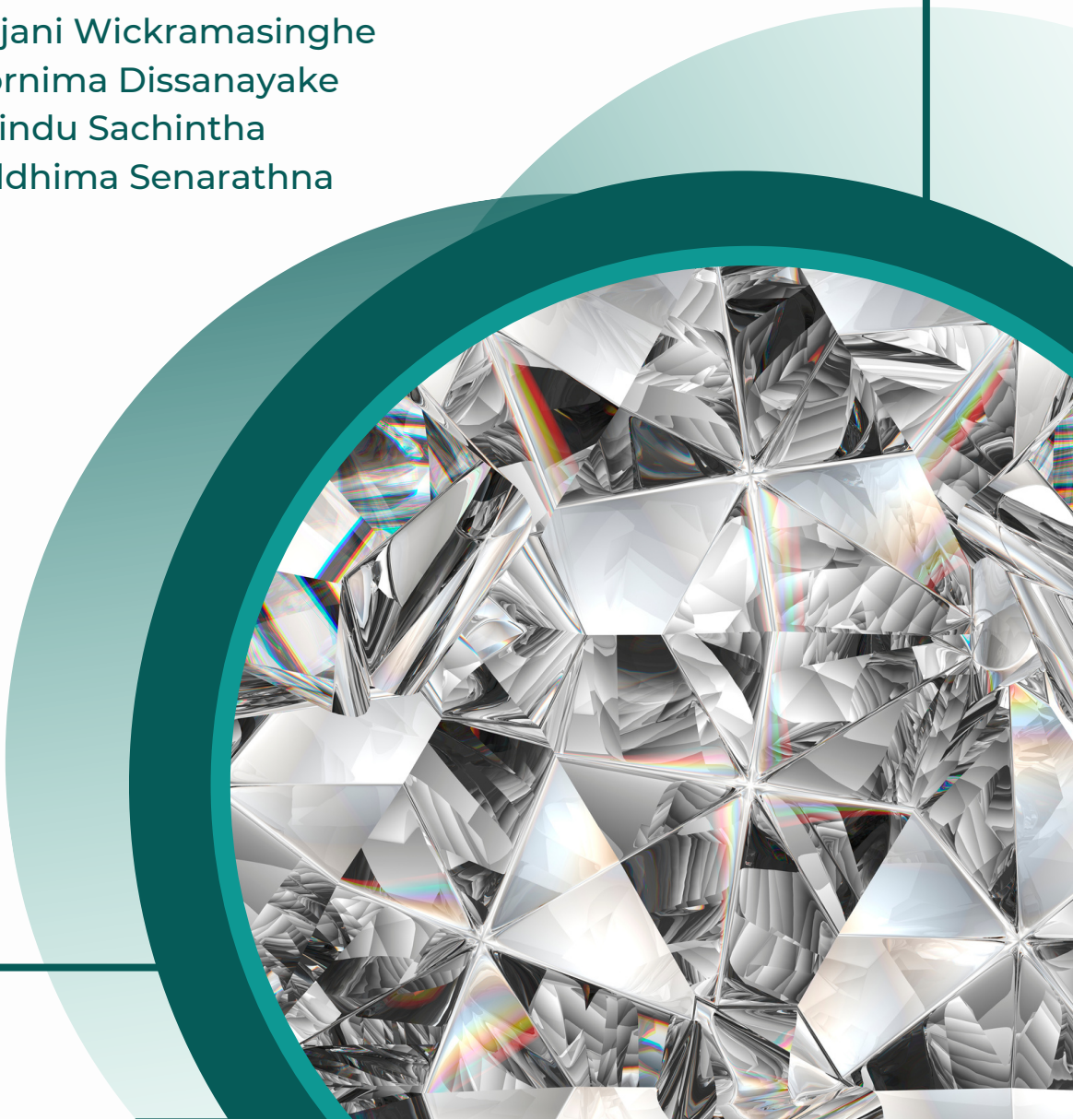
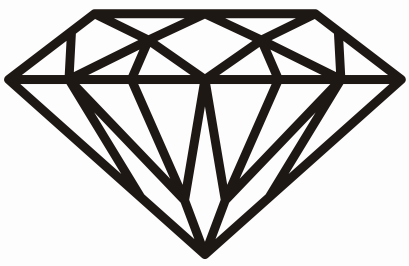


Table of Content

List of figures

List of tables



Multiple Linear Regression

Multiple linear regression (MLR), is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of multiple linear regression is to model the linear relationship between the explanatory (independent) variables and response (dependent) variables.

As our initial model of the analysis, we will fit a MLR model by considering log(Price) as the response variable and log(Carat), color, cut and clarity as the explanatory variables. Then, the predicted log(Price) values would be transformed to obtain the predicted prices of the diamonds.

The summary of the fitted model is as follows.

	ln_price		
Predictors	*Estimates*	*std. Error*	*p*
(Intercept)	7.3388	0.0067	**<0.001**
ln carat	1.8840	0.0013	**<0.001**
color [I]	0.1410	0.0035	**<0.001**
color [H]	0.2602	0.0033	**<0.001**
color [G]	0.3531	0.0032	**<0.001**
color [F]	0.4169	0.0033	**<0.001**
color [E]	0.4573	0.0033	**<0.001**
color [D]	0.5112	0.0034	**<0.001**
cut [Good]	0.0821	0.0044	**<0.001**
cut [Very Good]	0.1170	0.0041	**<0.001**
cut [Premium]	0.1403	0.0040	**<0.001**
cut [Ideal]	0.1615	0.0040	**<0.001**

	ln_price		
Predictors	*Estimates*	*std. Error*	*p*
clarity [SI2]	0.4325	0.0058	**<0.001**
clarity [SI1]	0.5989	0.0058	**<0.001**
clarity [VS2]	0.7480	0.0058	**<0.001**
clarity [VS1]	0.8178	0.0059	**<0.001**
clarity [VVS2]	0.9538	0.0061	**<0.001**
clarity [VVS1]	1.0248	0.0062	**<0.001**
clarity [IF]	1.1162	0.0067	**<0.001**
Observations	43072		
R^2 / R^2 adjusted	0.983 / 0.983		

Table 1 : Multiple Linear regression summary

All the estimated coefficients are statistically significant and the model performs well on the training dataset as it has a quite high adj.R2 value which is 98.3%. But, our goal is to predict the values of unseen data with high accuracy. Hence, we will examine how well this estimated model fits our test dataset.

Performance of the Estimated MLR model on the Test Data

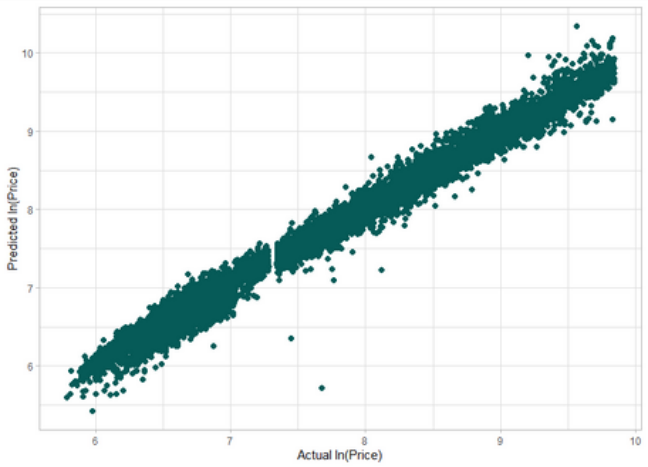


Figure 1: Plot of the relationship between Actual log(Price) vs Predicted log(Price)

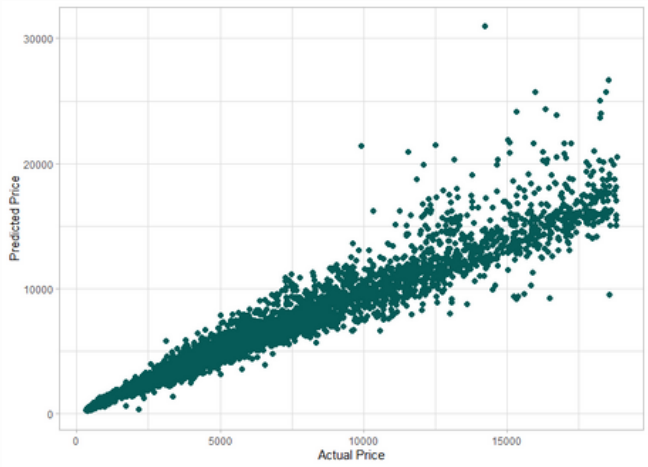


Figure 2: Plot of the relationship between Actual Price vs Predicted Price

The correlation between estimated log(Price) and actual log(Price) tends to be 0.99 which implies that the model fits the unseen data quite well. But, in our study, we need to predict the price of the diamond. Hence by applying the exponential transformation, we can obtain the predicted price values.

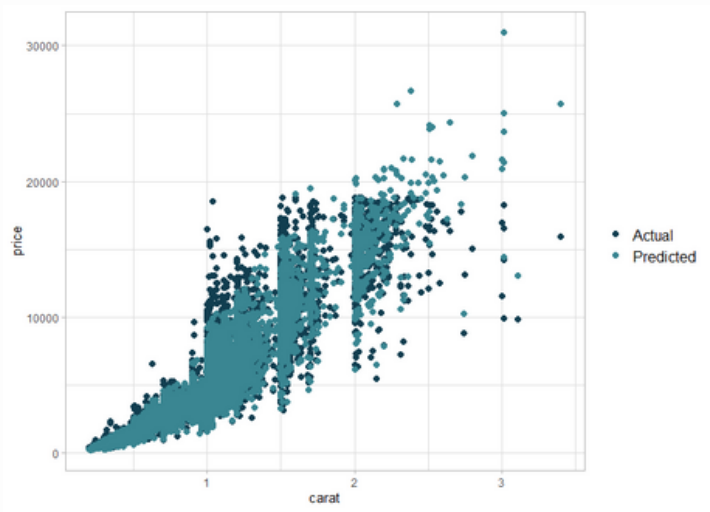


Figure 3: Plot of Carat vs Price grouped by the actual and the predicted values

We could examine that the correlation between the actual prices and the predicted prices is 0.979. Although it is a quite strong correlation, the variation of the predicted values tends to increase as the price increases which implies that the error of the prediction increases when price increases. This is clearly visible in figure 2 and figure 3.

Model Evaluation on the Training Data and the Test Data

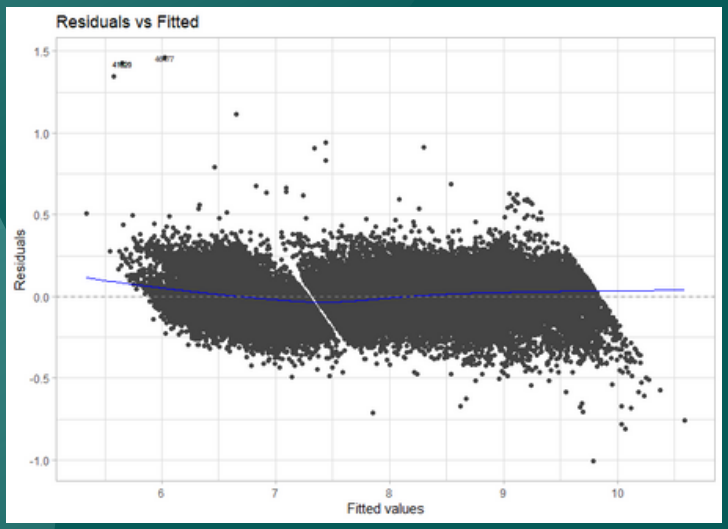
	Training Data	*Test Data*
n	43072	10768
Adj. R^2	0.983	0.982
RMSE	0.133	0.135

Table 2 : Summary of MLR model evaluation

The resulted RMSE values suggests that the model is not overfitted and it fits the unseen data quite well which implies the estimated model has a high accuracy.

Checking the Assumptions of the MLR model

- *Linearity - The relationship between the predictors and response is assumed to be linear.*



The residual plot shows no fitted pattern. The blue line is approximately horizontal at zero. This suggests that we can assume a linear relationship between the predictors and the response variable.

Figure 4: Scatter plot between residuals vs Fitted values to check linearity of MLR

- *Residuals are normally distributed with mean zero.*

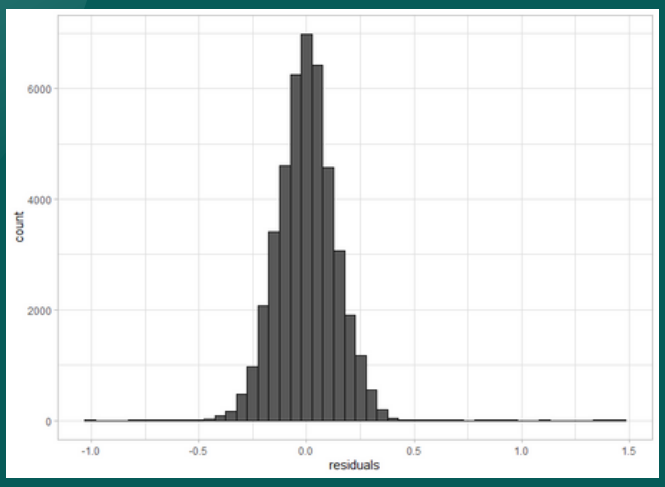


Figure 5: Histogram between residuals and the count to check normality of MLR

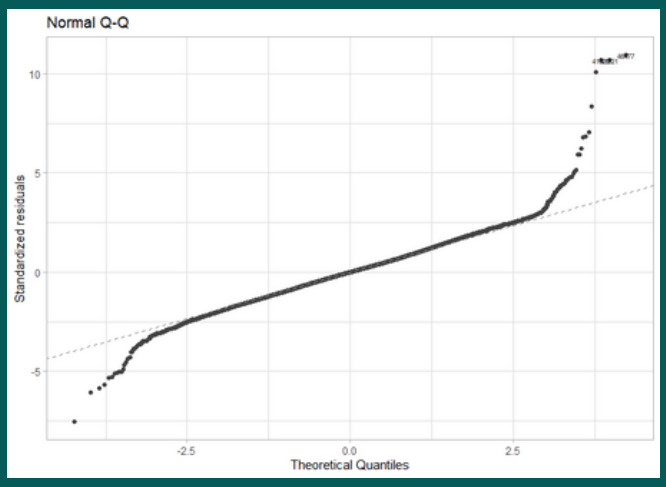


Figure 6: Plot between theoretical quantiles and the standardized residuals to check normality of MLR

Histogram shows that the residuals are normally distributed in the middle range where the mean is approximately zero, while there are few observations in the tails of the distribution. The QQ plot suggests the same behavior as most of the points are scattered on the reference line and there are few points deviating from the line. QQ plot can suggest that the normality assumption of the residuals is satisfied.

- *Independence of the residual error terms*

We performed a Durbin Watson Test using the durbinWatsonTest function in r and it provided the Durbin Watson test statistic as 1.27885 with a p-value of 0 which implies that the independence assumption is not satisfied.

- *Homoscedasticity which implies the residuals have constant variance.*

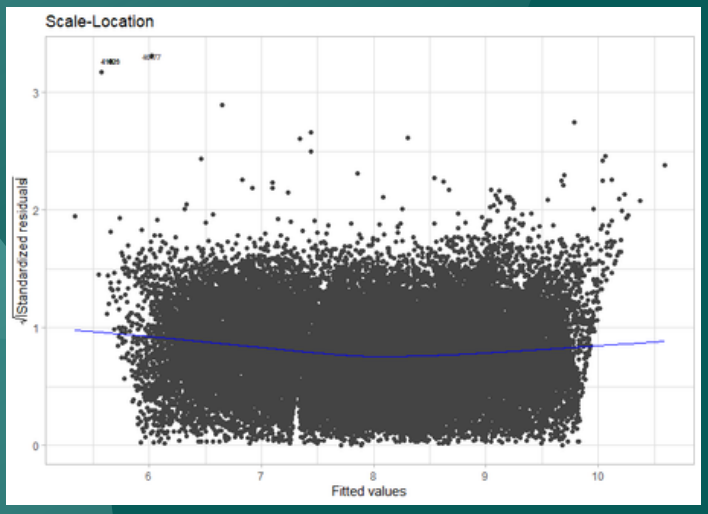


Figure 7: Plot between Fitted values vs Standardized residuals to check the Homoscedasticity of MLR

This assumption can be checked by examining the scale-location plot, also known as the spread-location plot. This plot shows that the residuals are not spread equally along the ranges of predictors as the range above the horizontal line is high. This implies that the residuals may not have constant variance. Further, we used the ncvTest and it indicated a significant result which implies the constant variance assumption is not satisfied.

- *Multicollinearity is not present*

Predictor	*VIF*
ln carat	1.318
Color	1.023
Cut	1.026
Clarity	1.042

Table 3: VIF values of variables

As all the VIF values are closer to 1, presence of multicollinearity is not an issue in our model.

Remarks :

Although our MLR model fits the data well, we found out that some of the model assumptions are not satisfied. Violating the assumptions of multiple linear regression can lead to biased and inefficient estimates, incorrect conclusions, and inaccurate predictions. Therefore more different models will be fitted and evaluated.

Regularization Techniques

Regularization techniques such as Lasso, Ridge, and Elastic Net have become increasingly important in the field of machine learning and statistical modeling. These techniques are used to prevent overfitting, improve model accuracy, and handle multicollinearity in the data.

Lasso Regression

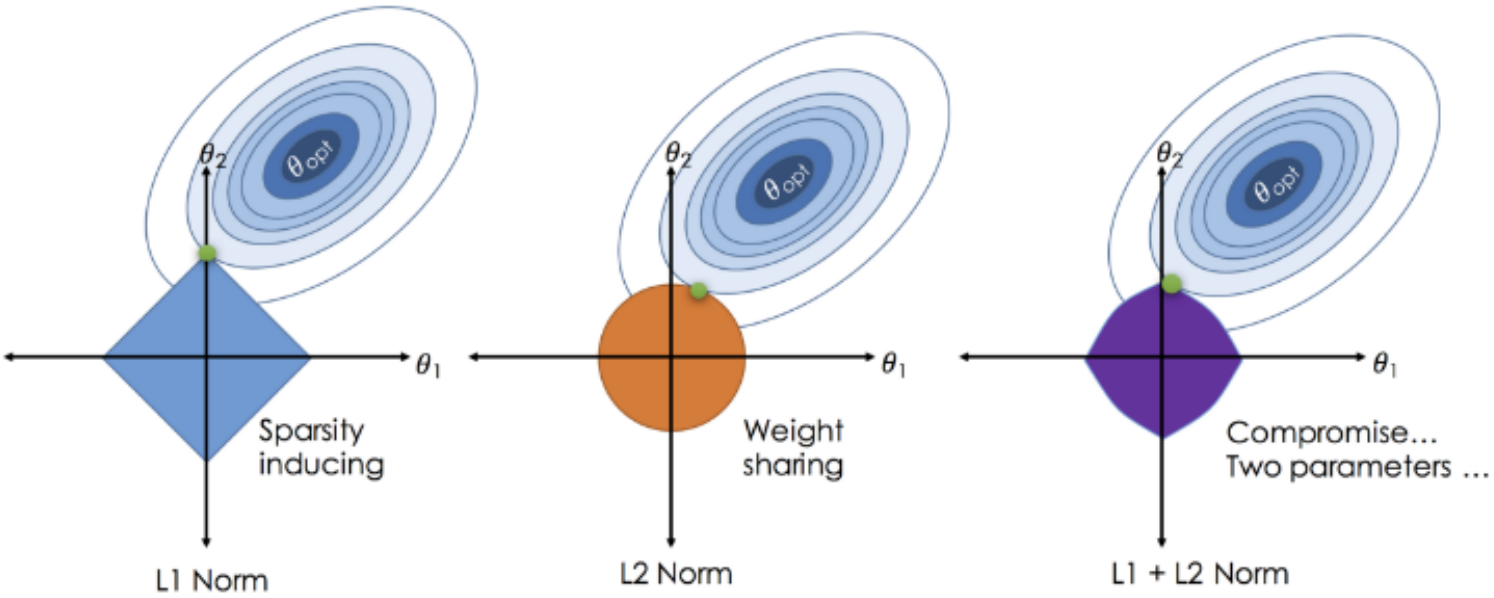
Lasso, or Least Absolute Shrinkage and Selection Operator, is a regularization technique used to perform variable selection in linear regression. Lasso adds a penalty term to the regression equation that forces some of the coefficients to be zero, effectively eliminating some of the variables from the model.

Ridge Regression

Ridge regression is another regularization technique used in linear regression. Similar to Lasso, Ridge adds a penalty term to the regression equation. However, the penalty term in Ridge is the sum of the squares of the coefficients, as opposed to the absolute values in Lasso. This results in Ridge shrinking the coefficients towards zero, but not necessarily to zero.

Elastic Net Regression

Elastic Net is a combination of Lasso and Ridge regression. Elastic Net adds both L1 (Lasso) and L2 (Ridge) penalties to the regression equation. This allows Elastic Net to handle multicollinearity in the data and perform variable selection at the same time. Elastic Net is useful when there are many correlated predictor variables in the model.



Model Evaluation on the Training Data and the Test Data

Model	Training RMSE	Test RMSE	Test R2
Ridge	1372.6	1395.69	0.88
Lasso	1203.31	1199.21	0.903
Elastic Net	1205.43	1202.57	0.905

Table 4: Model evaluation of regularization techniques

So according to the results obtained through running regression methods such as Lasso, Ridge and Elastic-Net, it shows that the machine learning algorithm on the data set has given accurate results because of the low difference between the Training RMSE and Test RMSE and we have evidence to conclude that the models have not been overfitted.

By having a thorough look at the results obtained we can rank the regression techniques according to the RMSE values obtained and the R squared values obtained. By considering only the RMSE values we could say that the Elastic-Net approach gives the lowest RMSE and the Ridge approach has the highest RMSE averagely. Also by R squared value, the Elastic-Net model has got the highest R squared value while the Ridge model has the lowest.

RMSE is the root mean square error and it measures the average difference between the predicted values and the actual values. The R squared value is the coefficient of determination and it speaks how much of a variation of a dependent variable is explained by the independent variables in the model. Both measures the goodness of a regression model.

Based on the findings presented, it is clear that the Lasso and Elastic-Net models outperform the Ridge model in terms of Training RMSE and Test RMSE, and R squared value. Specifically, the Elastic Net model achieved the lowest Training RMSE and Test RMSE values, indicating that it has the best predictive performance among the four models.

Furthermore, the Elastic-Net and Lasso models both achieved high R squared values, which suggests that they explain a large proportion of the variance in the target variable.

To sum up, it is recommended to use the Elastic-Net model apart from the Lasso model because the Elastic-Net model provides the lowest RMSE value even though both have quite similar R squared values for predicting the target variable in the dataset.

Random Forest Regression

Random Forest is a strong ensemble learning method that may be used to solve a wide range of prediction problems, including classification and regression. Because the method is based on an ensemble of decision trees, it offers all of the benefits of decision trees, such as high accuracy, ease of use, and the absence of the need to scale data. Furthermore, it has a significant advantage over ordinary decision trees in that it is resistant to overfitting as the trees are joined.

So, we'll use a Random Forest Regressor in R to try to forecast the price of diamonds using the Diamonds dataset

Loading Data for Random Forest

The dataset contains information on 53,940 diamonds. It contains the price as well as 9 other attributes. When examining the dataset, some unusual observations were noticed. It's a fact that measurements such as length(x), width(y) and depth(z) cannot take 0 values. Hence, the records containing 0 in any of these variables were removed. Some features are in the text format, and we encoded them in numerical format. We also dropped the unnamed index column.

Training the model and making predictions

At this point, we have to split our data into training and test sets. As a training set, we will take 80% of all rows and use 20% as test data.

One of the advantages of the Random Forest algorithm is that it does not require data scaling, as previously stated. To apply this random forest technique, all we need to do is define the features and the target we're attempting to predict.

We now have a model that has been pre-trained and can predict values for the test data. The model's accuracy is then evaluated by comparing the predicted value to the actual values in the test data. We will present this comparison in the form of a table and plot the price and carat value to make it more illustrative.

Below table represents only first 20 observations of testing data.

	carat	cut	color	clarity	depth	table	x	y	z	price	price_pred
40855	0.50	Ideal	G	SI1	62.2	56.0	5.13	5.08	3.17	1173	1337.5869
40925	0.30	Ideal	F	SI1	62.0	55.1	4.29	4.32	2.67	499	484.1267
42037	0.53	VeryGood	F	SI1	62.9	57.0	5.15	5.18	3.25	1268	1310.7997
15269	1.20	Premium	H	SI1	62.1	58.0	6.77	6.72	4.19	6129	5748.9787
33768	0.32	VeryGood	D	VVS2	61.9	54.0	4.43	4.46	2.75	841	847.6931
35783	0.32	Ideal	F	VVS1	61.0	56.0	4.42	4.43	2.70	912	883.3783
17518	1.04	Premium	E	VS2	61.1	59.0	6.54	6.56	4.00	7047	7106.8375
15248	1.04	Premium	G	VS2	62.2	58.0	6.49	6.40	4.01	6122	6151.1322
19874	1.52	Premium	J	VVS2	58.3	62.0	7.61	7.49	4.40	8427	8347.3874
2626	0.90	Fair	H	VS2	65.5	57.0	6.05	6.01	3.95	3226	3543.1142
53223	0.74	VeryGood	E	VS2	61.4	63.0	5.79	5.77	3.55	2638	3017.8233
3005	0.71	Ideal	E	VS2	62.0	56.0	5.72	5.75	3.55	3304	3183.0082
17411	1.03	Premium	D	VS1	59.6	61.0	6.62	6.57	3.93	6974	7539.4045
33312	0.35	Ideal	D	SI1	62.3	55.0	4.56	4.52	2.83	827	802.8649
49869	0.70	VeryGood	G	SI1	62.1	56.0	5.64	5.67	3.51	2175	2221.9050
16993	1.01	VeryGood	E	VS2	63.5	58.0	6.36	6.34	4.03	6787	6489.4523
53010	0.70	Premium	F	SI1	59.4	59.0	5.80	5.75	3.43	2596	2386.9856
42437	0.53	Ideal	D	SI2	61.4	56.0	5.22	5.24	3.21	1314	1350.8174
40823	0.39	Ideal	G	VVS1	62.3	57.0	4.70	4.64	2.91	1170	1124.5366
32963	0.30	Ideal	F	SI1	62.3	56.0	4.27	4.30	2.67	461	491.9059

Table 5: Testing data with predictions

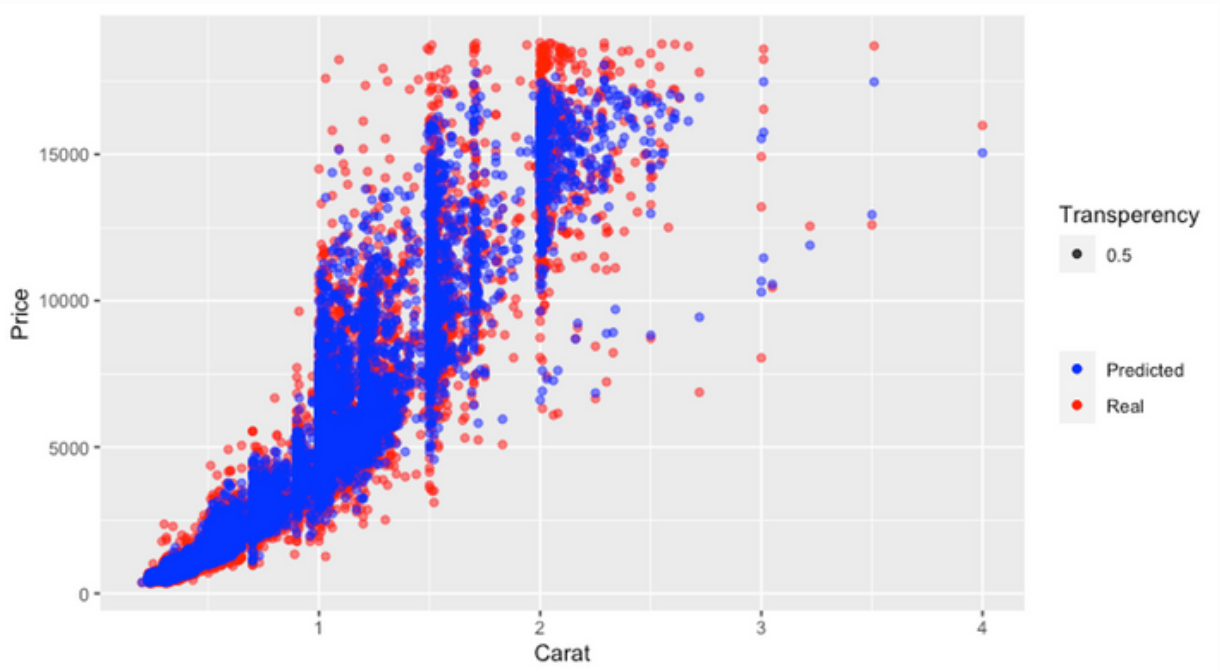


Figure 8: Plot between Carat and Price for Random Forest

The figure displays that predicted prices (blue scatters) coincide well with the real ones (red scatters), especially in the region of small carat values. But to estimate the accuracy our model more precisely, we will look at Mean absolute error (MAE), Mean squared error (MSE), and R-squared scores of the test data.

Mean absolute error (MAE): 271.02
Mean squared error (MSE): 285827.32
R-squared scores : 0.98

Summary of our random forest model:

randomForest(formula = price ~ ., data = training)

Type of random forest: regression
Number of trees: 500
No. of variables tried at each split: 3

Mean of squared residuals: 308678.4
% Var explained: 98.06

Correlation between actual and predicted values in test set: 99.1%

Defining and visualizing variables importance

For this algorithm, we used all available diamond features, but some of them contain more predictive power than others.

Let's build the plot with a features list on the y axis. On the X-axis we'll have an incremental decrease in node impurities from splitting on the variable, averaged over all trees, it is measured by the residual sum of squares and therefore gives us a rough idea about the predictive power of the feature.

Generally, it is important to keep in mind, that random forest does not allow for any causal interpretation.

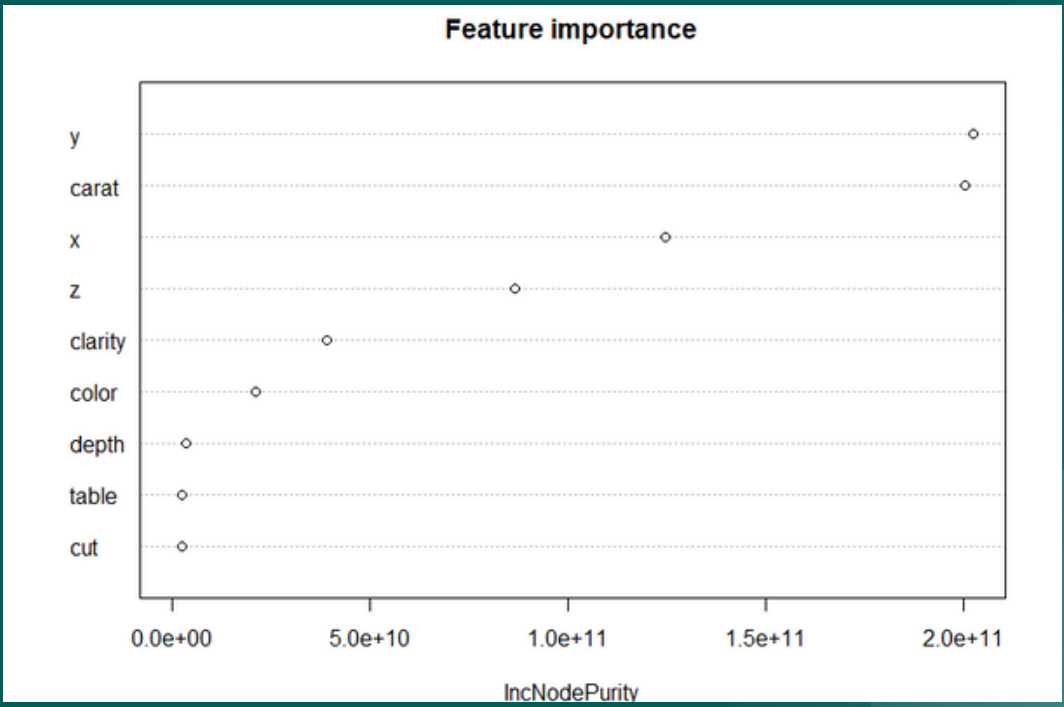


Figure 9: Plot of feature importance in Random Forest

From the figure above you can see that the size of the diamond (x,y,z refer to length, width, depth) and the weight (carat) contains the major part of the predictive power.

Selecting the Best Model

As our objective is to predict the price of a diamond given its characteristics, we will compare the accuracy of the models in predicting price.

Model	Training Data		Test Data	
	RMSE	MAPE	RMSE	MAPE
MLR	795.29	10.43%	809.66	10.5%
Ridge	1372.6	53.52%	1395.69	55.96%
Lasso	1203.31	43.03%	1199.21	44.22%
Elastic Net	1205.43	43.08%	1202.57	44.33%
Random Forest	555.59	6.9%	534.63	6.74%

Table 6: Overall Summary of advanced analysis techniques

The above table summarizes the RMSE and MAPE(mean absolute percentage error) of each model. We can examine that the lowest RMSE and MAPE scores are obtained using the Random Forest algorithm. Hence, a random forest model with same parameters will be fitted to the whole dataset obtained by combining the training and testing dataset and will be used as the final model. This final model will be used for further predictions.

Appendix

R Code: https://github.com/st3082group10/price_prediction_on_diamonds