



DIAMONDS

IN DEPTH ANALYSIS WITH R

NORAH ALYABS – nxa164430

KETKI CHAUDHARY – kxc170730

ASWIN KRISHNAMOORTHY - axk180011

ROHIT GURJAR – rsg180002

AMEY VANMALI – akv170000

Table of Contents

KEY QUESTION	2
IMPORTANCE OF DIAMONDS	2
ANALYSIS, RESULTS AND INFERENCE.....	3
<i>DATA COLLECTION</i>	3
<i>INSTALLING LIBRARIES</i>	3
<i>IMPORTING DATASET</i>	4
<i>DATA EXPLORATION</i>	4
<i>CHECKING FOR MISSING DATA</i>	4
<i>DESCRIPTIVE STATISTICS</i>	5
<i>DATA VISUALISATION</i>	5
<i>DIVIDING ENTIRE DATASET INTO TRAINING AND VALIDATION DATASET</i>	11
<i>PRINCIPAL COMPONENT ANALYSIS</i>	11
<i>LINEAR REGRESSION</i>	12
<i>LINEAR DISCRIMINANT ANALYSIS</i>	16
RECOMMENDATIONS	21
CITATIONS	21
APPENDIX.....	23
Individual Report.....	24

KEY QUESTION

The primary objective of this project is to predict the price of diamonds by applying statistical techniques and R Programming, using a comprehensive data set. The impact of this work would transcend mainly to the industries who use diamonds as a raw material or a catalyst in their process of production. An added application of our work would be predicting prices of diamonds from an investment perspective, since they are a significant to public.

This project is mainly focused on prediction of diamond prices based on few of its characteristic, which are detailed in the following sections. Our aim is to put ourselves in the shoes of an industry who is looking to procure diamonds for manufacturing goods, an individual who is keen on buying a good quality diamond whose value will potentially increase. We would be understanding the various questions that come into our minds from different perspectives and try to devise a solution which is applicable and scalable to almost all practical scenarios pertaining to procurement of diamonds.

IMPORTANCE OF DIAMONDS

The hardest substance on the planet. While some use it to accessorize themselves, there is a category who sees the fascinating properties of it and widen the range of its applications. Diamonds maybe the luxurious thing that people crave for and consider as a status symbol. But its chemical properties make it a considerable candidate for industrial applications as well.

When the existence of such an invaluable substance is established, it is imperative that its demand will be proportionate. The fact that diamonds are considered as an investment is a contributing factor for their demand. The longer a diamond is preserved, the rarer it gets and more will be its value. In case of industries, diamond being the hardest substance, can easily cut almost anything. They are also used for grinding, lapping and polishing. Apart from being a tool used in processing and finishing products, they are directly used as a components or catalysts in optics and electronics.

The industries who are involved in diamonds are not known to many since their association to general public is less and industries are more. Companies like Lackmond and Husqvarna are major players who produce cutting and grinding tools where diamonds are used as blades and drill bits. They are primarily used sawing/polishing concrete floors. An interesting fact is that Bosch is also a producer of diamond abrasive tools which are used extensively for cutting and drilling. Diamonds in electronics is a growing topic today, where researches think that the properties of diamonds could replace silicon in the electronics industry. This means that from the smallest components like semiconductors to electronics used in transportation, communication etc. AKHAN SEMI is the company who was one of the key revolutionary companies which implemented technology for introducing diamonds in electronics. They continue to produce diamond-based electronics components and wires which yield high quality and durability

The guide to identify a good quality diamond is the 4 Cs – Cut, Clarity, Colour and carat weight. Invariably, the price of a diamond is also decided by these 4 Cs. Although shape and certification are also influential, the 4 Cs are the major factors.

Cut is the most important factor since it defines the diamond's sparkle. A bad cut will result in the diamond being dull. **Colours** of diamonds are graded from D to Z. The D graded diamonds are purest in from, being colourless. Gradually moving forward in the grading chart, the diamonds move from

colourless to gaining a hue and light yellow/brown (Z grade). Naturally, a D grade diamond will be the costliest of the lot. Diamond's **Clarity** is more or a measure of "how less the imperfections are" rather than a metric of its accuracy. The blemishes in a diamond are microscopic and can be rectified to only some extent. This might be the reason clarity is considered as the least important factor in assessing the quality of a diamond. Finally, the Carat, which means weight and not the size of the diamond is the 4th C. Carat and cut goes hand in hand when it comes to buying a diamond.

ANALYSIS, RESULTS AND INFERENCE

DATA COLLECTION

The source of our dataset is Kaggle which is a widely used platform of numerous datasets consisting of various types of data. The data is available in terms of its file type such as Comma Separated Variable, JavaScript Object Notation, SQLite, Big Query etc. We came across a range of topics for our project which included datasets of App store, FIFA 18, and Diamonds which were later shortlisted.

As the project involved analysis in R, the commonly used and supported file type CSV was preferred which made importing the data easier. The advantage of a CSV for a tabular data is that it also has associated column descriptions and column metadata. The column description which is assigned to individual columns made it easier to understand what each column means.

Another aspect while choosing the dataset was concerned with the significant use of R concepts or techniques that we could explore in the project. We intended on using techniques like regression models, Principle Component Analysis, Visualizations and Performance Analysis etc. on the primary level. In accordance to this, the dataset of Diamonds was chosen as it had about 54 thousand rows with 11 attributes which was a blend of numerical and categorical variables. The dataset was clean with no missing values in it.

As mentioned earlier, the dataset had no missing values, however some attributes had impractical values which were not useful for the analysis. Hence those values were removed from the dataset by creating a subset of required values. Also an unwanted column was deleted from the dataset which was not required in the analysis by indexing.

INSTALLING LIBRARIES

The analysis, visualisation and algorithms applied on the data required importing and installing many libraries. The libraries used in the project are mentioned below:

1. utils
2. dplyr
3. MASS
4. gtools
5. lattice
6. ggplot2
7. corrplot
8. GGally
9. caret
10. RColorBrewer
11. klaR

IMPORTING DATASET

The dataset is in csv format and is imported using read.csv command. Moreover, commands like summary, View and head are used to have better insights of the data in hand.

```
diamonds <- read.csv("diamonds.csv")
attach(diamonds)
head(diamonds)
```

DATA EXPLORATION

The data has about 53K observations on 11 variables, 3 of them are Categorical variables (Cut, Colour, and Clarity) while the others are Quantitative variables.

The description of each variable is given below:

1. X_1: Index: counter.
2. Carat: Weight of the diamond in Carats.
3. Cut: Describes the cut quality of the diamond. Quality in increasing order are: Fair, Good, Very Good, Premium, Ideal.
4. Color: Gives the color of the diamond, with D being the best and J the worst.
5. Clarity: States the inclusions within the diamond. The order from best to worst is: FL = flawless, I3= level 3 inclusions) FL,IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1, I2, I3.
6. Depth: The height of a diamond, measured from the culet to the table, divided by its average girdle diameter.
7. Table: The width of the diamond's table expressed as a percentage of its average diameter.
8. Price: The price of the diamond.
9. X: Length of the diamond in mm.
10. Y: Width of the diamond in mm.
11. Z: Depth of the diamond in mm.

CHECKING FOR MISSING DATA

There are no missing values in the data.

On the other hand, the dimension variables (x, y, z) and Carat indicates weight, which cannot take a value less than or equal to Zero. However, some dimension variables have the value Zero which doesn't make sense, hence we will remove those values and keep only the sensible values.

Additionally, we will also remove the first column X_1 because we don't need it in the analysis.

```
diamonds[!complete.cases(diamonds),]
diamonds <- subset(diamonds, z > 0)
diamonds <- diamonds[, -1]
```

DESCRIPTIVE STATISTICS

The R command `sapply` is used to gather insights into data by determining the Mean, Standard Deviation, Minimum value, Maximum value, Median and Length.

```
summ1 <- data.frame(mean=sapply(diamonds, mean),
                    sd=sapply(diamonds, sd),
                    min=sapply(diamonds, min),
                    max=sapply(diamonds, max),
                    median=sapply(diamonds, median),
                    length=sapply(diamonds, length),
                    miss.val=sapply(diamonds, function(x)
                                   sum((is.na(x)))))
options(scipen = 999)
print(summ1, digits=1)
```

	mean <dbl>	sd <dbl>	min <fctr>	max <fctr>	median <dbl>	length <int>
Carat	0.7976983	0.4737953	0.2	5.01	0.70	53920
Cut	NA	NA	Fair	Very Good	NA	53920
Color	NA	NA	D	J	NA	53920
Clarity	NA	NA	I1	VVS2	NA	53920
Depth	61.7495141	1.4323311	43	79	61.80	53920
Table	57.4568342	2.2340642	43	95	57.00	53920
Price	3930.9932307	3987.2804460	326	18823	2401.00	53920
X	5.7316269	1.1194228	3.73	10.74	5.70	53920
Y	5.7348871	1.1401258	3.68	58.9	5.71	53920
Z	3.5400464	0.7025303	1.07	31.8	3.53	53920

Table 1: Descriptive Analysis

DATA VISUALISATION

Data is visualised using various graphs including bar charts, scatterplots, boxplots, line charts etc.

These visualisations help us infer key insights into the data and have been explained below.

1. Visualising Colour:

```
colfunc <- colorRampPalette(c("navy", "white"))
barplot(table(diamonds$color), main = "color distribution", xlab = "color", col = colfunc(7))
```

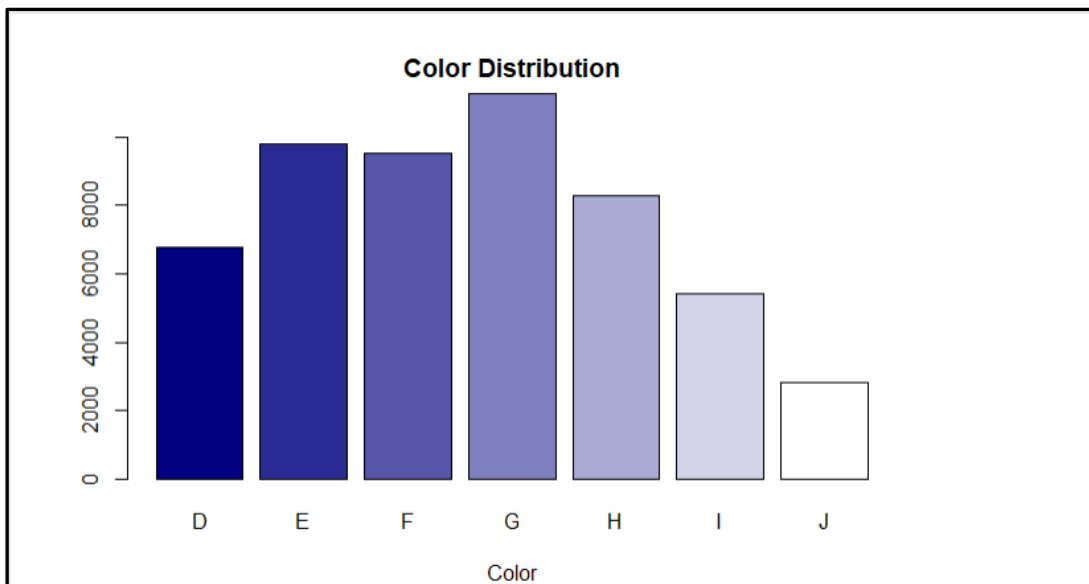


Figure 1: Visualizing colour

This bar chart represents the Colour of the diamond, with D being the most crystal clear and J being the least. Colour G has the highest price comparing with other colors.

2. Visualising Cut:

```
colfunc <- colorRampPalette(c("green", "white"))
barplot(table(diamonds$cut), main = "cut distribution", xlab = "color", col = colfunc(5))
```

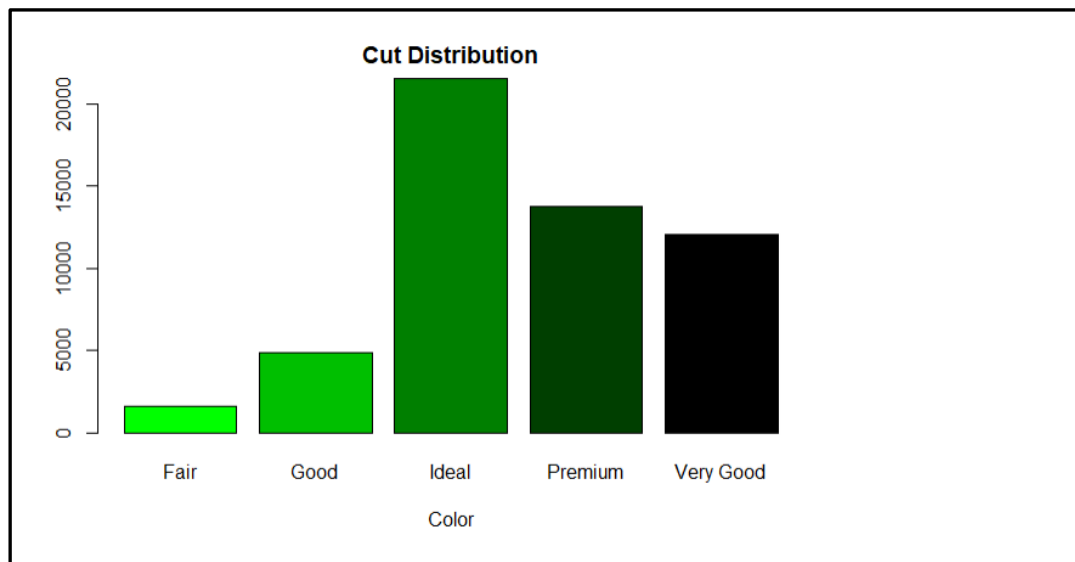


Figure 2: Visualizing Cut

This chart displays the cut quality of the diamond and shows that the most prominent cut is the ideal cut.

3. Visualising Clarity:

```
colfunc <- colorRampPalette(c("white", "black"))  
barplot(table(diamonds$clarity), main = "clarity distribution", xlab = "color", col = colfunc(11))
```

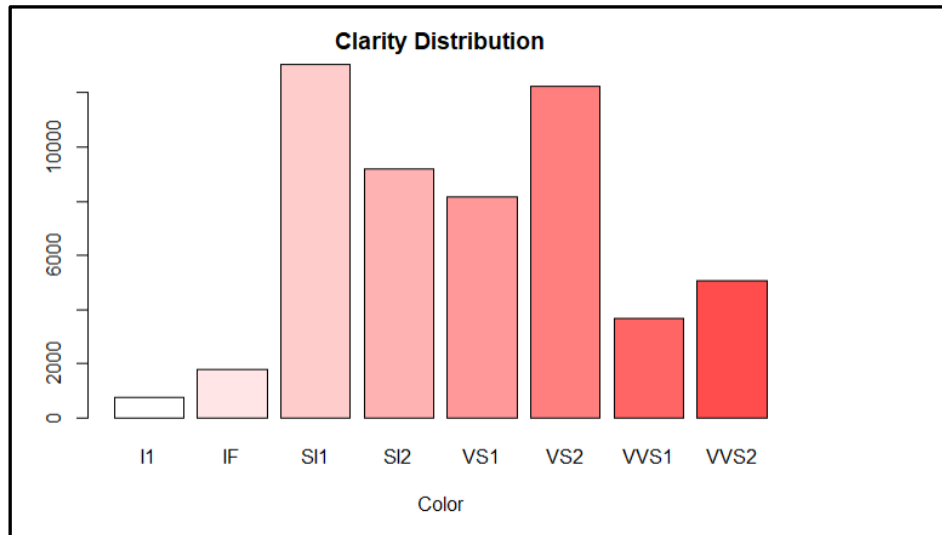


Figure 3: Visualizing clarity

This visualisation presents how obvious the inclusions are within the diamond. SI1 and VS2 are leading among all of them.

4. Visualising Cut and Clarity:

```
ggplot(data = diamonds) +  
  geom_bar(mapping = aes(x = cut, fill = clarity), alpha = 1/2, position = "dodge")
```

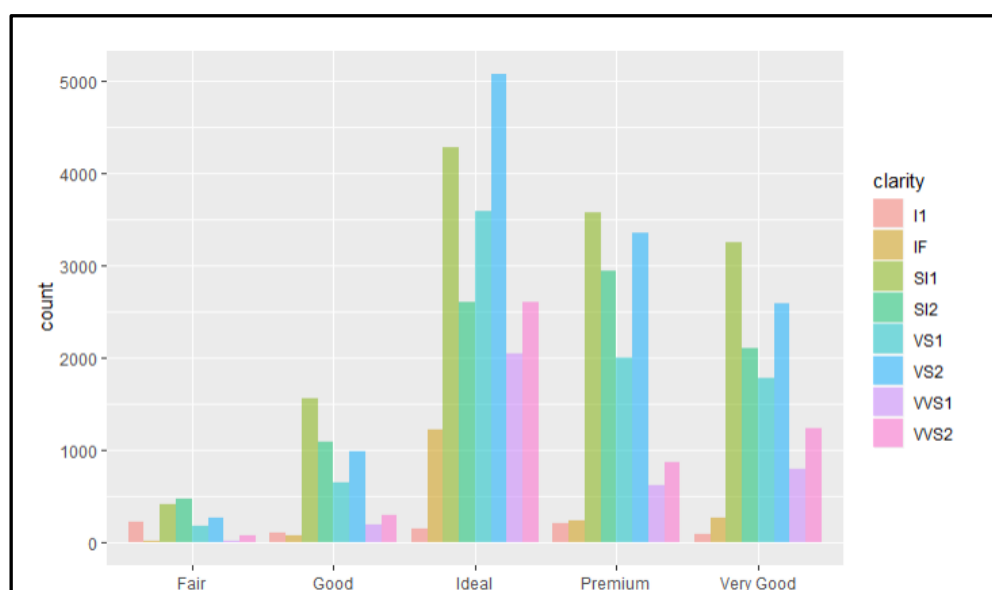


Figure 4: Visualizing cut and clarity

The above ggplot compares 2 categorical data - Clarity and Cut. As visible from the graph, maximum VS2 are present in the ideal cut and the minimum VS2 are present in Fair cut. Similar inferences can be made from this graph.

5. Visualising Clarity Using Boxplot:

```
ggplot(diamonds, aes(x=color, y=price, fill=color)) + geom_boxplot()+
scale_fill_brewer(palette="Set1")
ggplot(diamonds, aes(x=cut, y=price, fill=cut)) + geom_boxplot()+
scale_fill_brewer(palette="Set2")
ggplot(diamonds, aes(x=clarity, y=price, fill=clarity)) + geom_boxplot()+
scale_fill_brewer(palette="Set3")
```

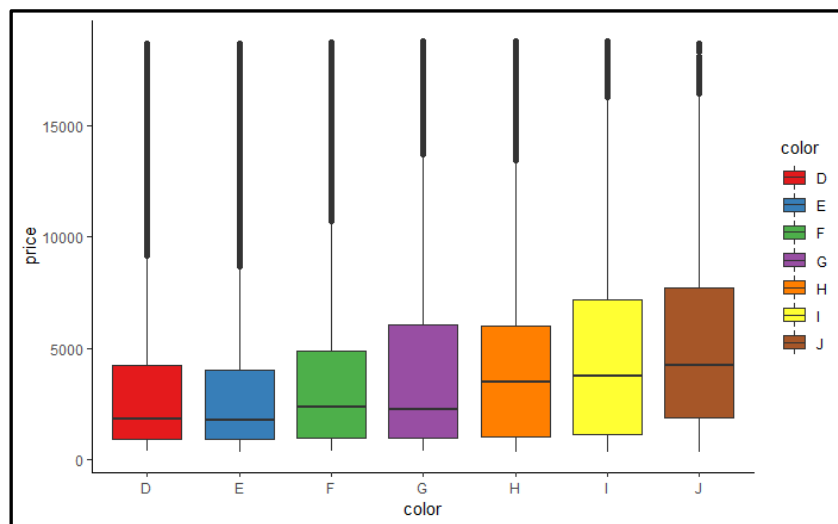


Figure 5: Visualizing Clarity using Boxplots -1

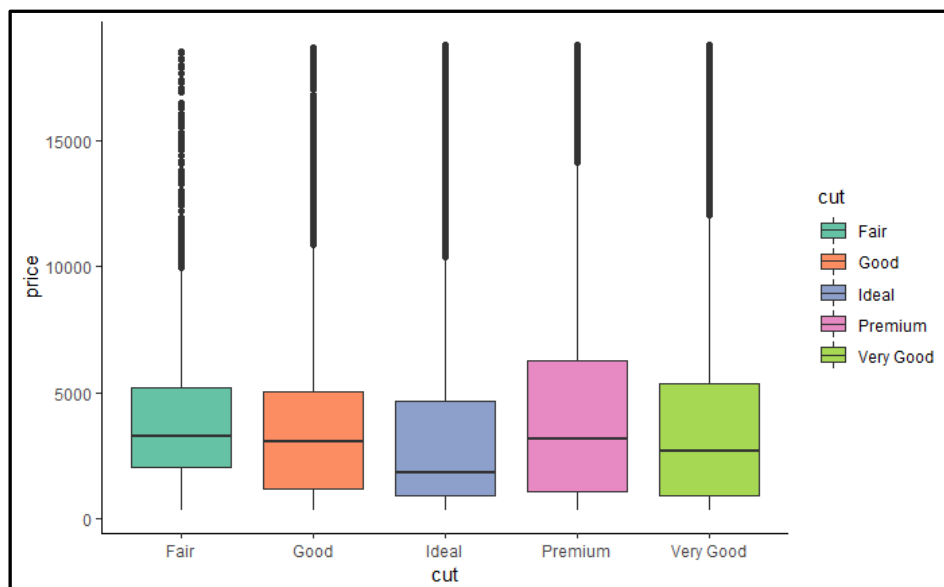


Figure 6: Visualizing Clarity using Boxplots -2

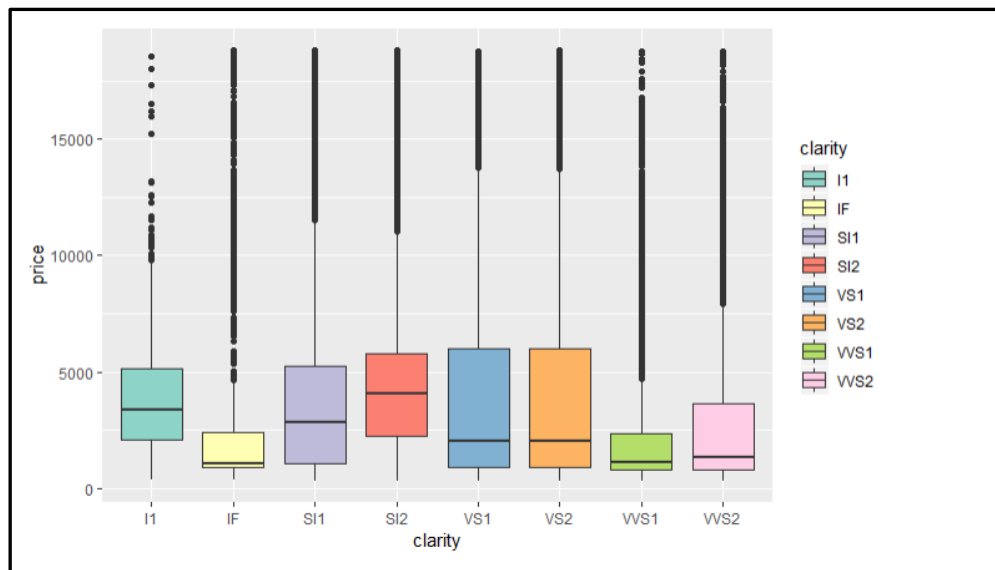


Figure 7: Visualizing Clarity using Boxplots -3

Boxplots are an excellent way to detect outliers in data, these are presented in the above visualisations. All the three plots show the prices for different categories has extreme values, that is extremely expensive diamonds could be from any cut, colour and clarity.

6. Visualising Price and Carat:

The scatterplot displays high correlation between price and carat. Hence, we can conclude that higher weight of diamonds result in higher price.

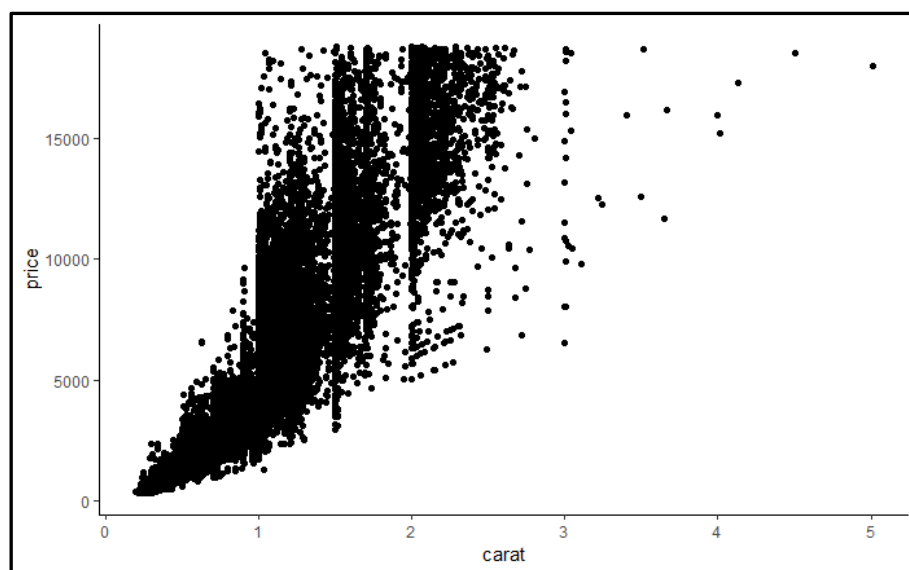


Figure 8: Scatter plot for correlation between price and carat

7. Correlation Matrix of Quantitative Variables:

```
diamondsM <- diamonds[,c(1,5:10)]  
cor.matrix <- cor(diamondsM)  
corrplot(cor.matrix)
```

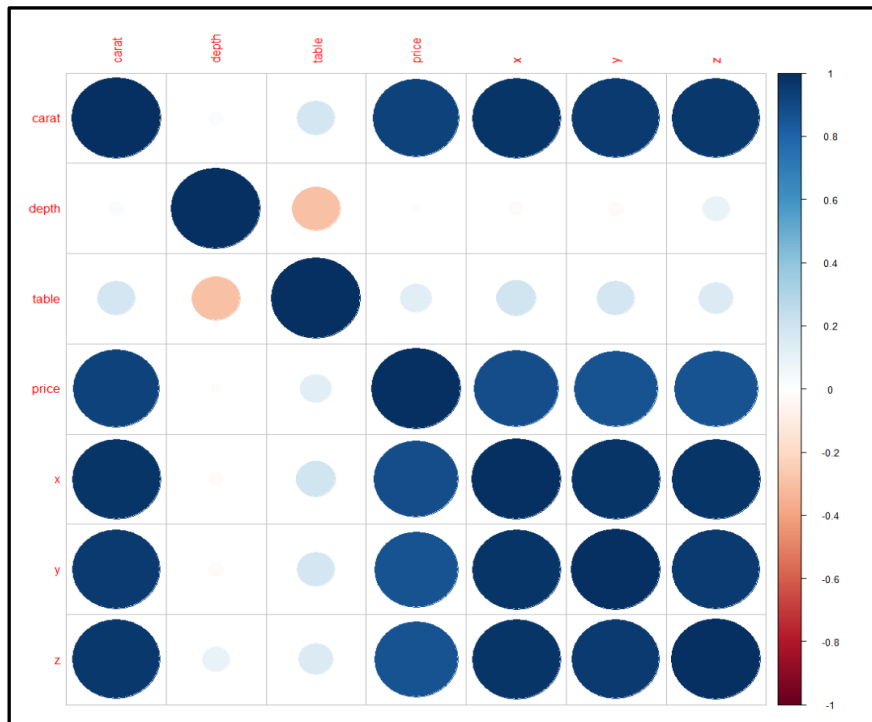


Figure9: Correlation Matrix of quantitative variables

The results of this plot show that price has high positive correlation with the variables carat, x, y, z which makes sense and explains the common logic.

As size of the diamond or carat increases, the price will invariably increase. Similarly, the size will also dictate the cost of a diamond.

While size and carat play the major role, table and depth does not have a significant say in deciding price. They are actually not important at all. This can also be explained since these are characteristics of a diamond which provides information about the significant properties of a diamond. For example, lower the depth, the larger the carat will appear.

Should we use them in predicting the price or no? This question will be answer after fitting the linear regression model.

DIVIDING ENTIRE DATASET INTO TRAINING AND VALIDATION DATASET

Following the ideal approach, the data sets are divided into 2 parts. First will be create a training dataset on which we test our models and then we create the validation dataset on which we would apply the best model to predict results.

The training set for this analysis will consist of 60% of complete dataset and validation data set will consist of the rest 40%.

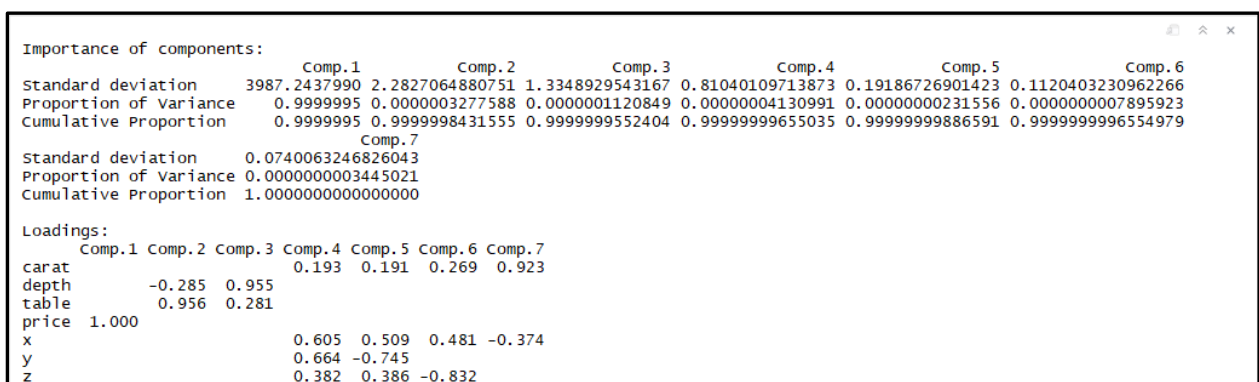
```
set.seed(123)
training.index <- createDataPartition(diamonds$price, p = 0.6, list = FALSE)
dim.train <- diamonds[training.index, ]
dim.valid <- diamonds[-training.index, ]
```

PRINCIPAL COMPONENT ANALYSIS

Principal Components Analysis is used to reduce the number of numerical variables by removing the overlap of information between them. This analysis is done prior to running regression models to identify significant variables and pick them for analysis. Choosing redundant variables would potentially decrease the accuracy of model and will not yield optimum results.

```
pca <- princomp(diamondsM)
summary(pca, loadings=TRUE)
```

Output:



```
Importance of components:
              Comp.1          Comp.2          Comp.3          Comp.4          Comp.5          Comp.6
Standard deviation 3987.2437990 2.2827064880751 1.3348929543167 0.81040109713873 0.19186726901423 0.1120403230962266
Proportion of Variance 0.9999995 0.0000003277588 0.0000001120849 0.00000004130991 0.00000000231556 0.0000000007895923
Cumulative Proportion 0.9999995 0.9999998431555 0.999999952404 0.99999999655035 0.99999999886591 0.9999999996554979

              Comp.7
Standard deviation 0.0740063246826043
Proportion of Variance 0.0000000003445021
Cumulative Proportion 1.0000000000000000

Loadings:
              Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
carat              1.000
depth             -0.285  0.955
table              0.956  0.281
price              0.605  0.509  0.481 -0.374
x                  0.664 -0.745
y                  0.382  0.386 -0.832
z
```

Figure 90: Principal Component Analysis Results

We see that for our data set, just a single component can be used to convey data in the entire dataset.

But we interpret this result as redundant because considering the business scenarios of the problem statement, it's imperative to consider all the variables for regression analysis.

On the contrary, this analysis reinforces the credibility of the data set. The analysis from this will be reliable as the data set is clean and crisp.

LINEAR REGRESSION

Linear regression is an approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The relationships are modelled using linear predictor functions whose unknown model parameters are estimated from the data.

Before we predict the price of diamonds, the regression assumptions must be verified for this dataset to ensure whether we can apply Linear Regression on it. We should delve into the following aspects:

1. Linearity of the data.
2. Normality of residuals.
3. Homogeneity of residuals variance.
4. Independence of residuals error terms.

```
ggplot(data=dim.train, aes(price)) + geom_histogram(fill = "firebrick" )+ ggtitle("Frequency  
distribution of the price")
```

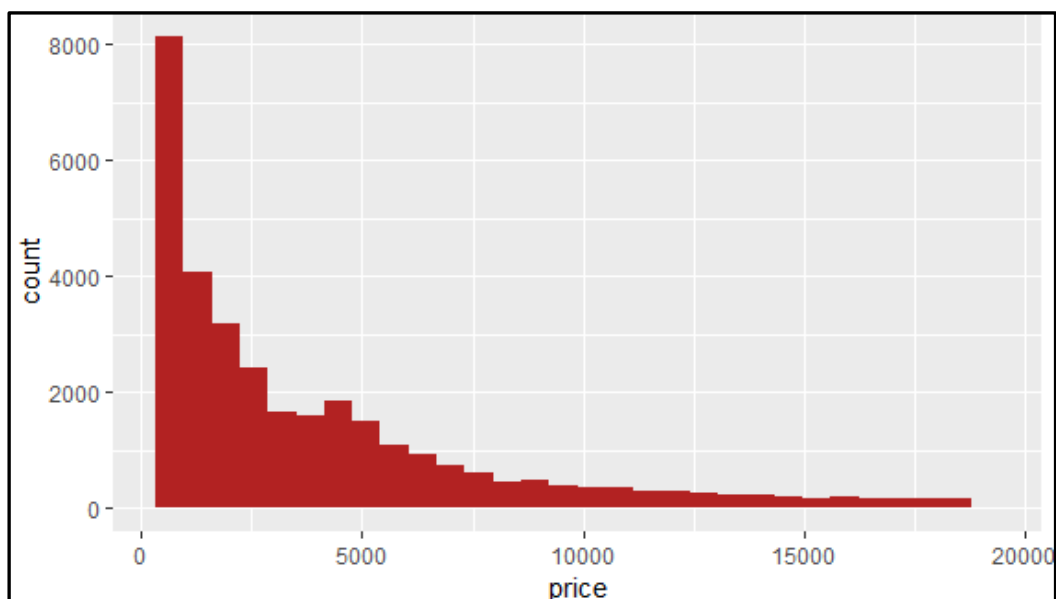


Figure 101: Histogram plot to check Normal distribution

The plot shows that the variable Price does not follow a Normal distribution. It shows a right skewed plot which means that there are diamonds which are on the extremely expensive side and there are diamonds which are comparatively less expensive as well.

The data should be normalized before applying regression techniques to produce optimum results. In order to do this, we will apply a logarithmic transformation on Price to produce normalized data.

```
dim.train1<-dim.train %>% mutate(logy=log(price))
dim.train1<-dim.train1[,-7]
ggplot(data=dim.train1, aes(logy)) + geom_histogram(fill = "firebrick" )+ ggtitle("Frequency
distribution of the log(price)")
```

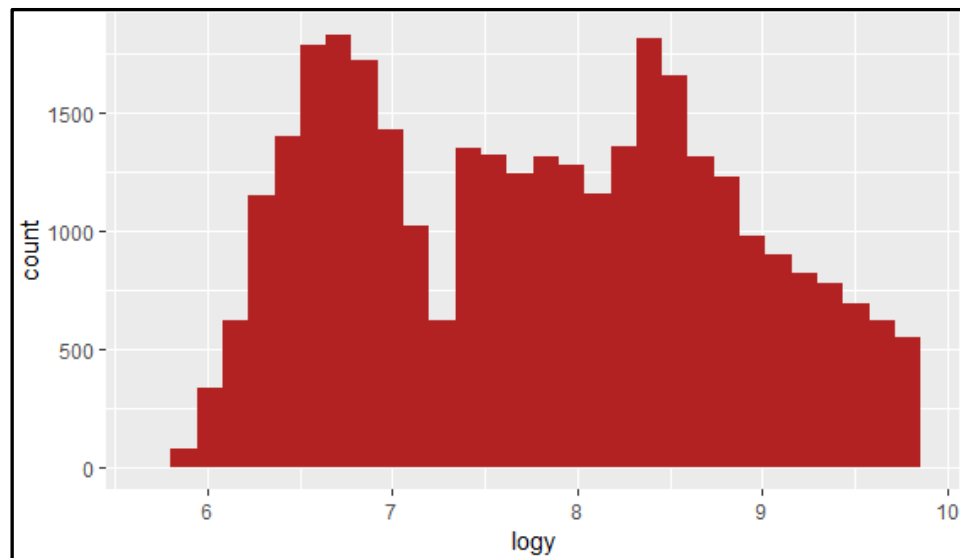


Figure 112: Histogram Plot of Logarithmic Transformation

The Logarithmic transformation looks like two normal curves. To validate this and attain an assurance, we shall use the Boxcox power transformation.

```
model <- lm(price ~., data= dim.train)
model
boxcox(model)
```

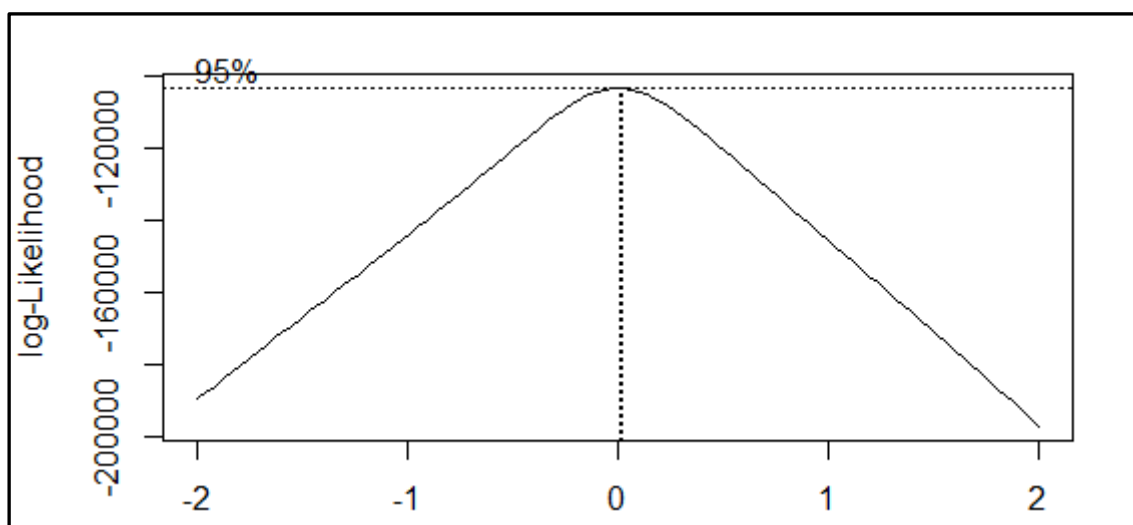


Figure 123: Boxcox plot

The results of Boxcox validates with our decision of using logarithmic transformation.

Now, we run Linear Regression on training dataset to test our model's credibility and accuracy.

```
Call:
lm(formula = logy ~ ., data = dim.train1)

Residuals:
    Min       1Q   Median       3Q      Max
-2.58890 -0.08651 -0.00128  0.08778  2.62326

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.1786919   0.0637522  -65.546 < 0.0000000000000002 ***
carat       -0.9932355   0.0081439  -121.961 < 0.0000000000000002 ***
cutGood      0.0954221   0.0053580   17.809 < 0.0000000000000002 ***
cutIdeal     0.1599585   0.0053374   29.969 < 0.0000000000000002 ***
cutPremium   0.1149086   0.0051532   22.299 < 0.0000000000000002 ***
cutVery Good 0.1331637   0.0051552   25.831 < 0.0000000000000002 ***
colorE      -0.0556732   0.0028314   -19.663 < 0.0000000000000002 ***
colorF      -0.0939733   0.0028508   -32.963 < 0.0000000000000002 ***
colorG      -0.1598780   0.0028003   -57.093 < 0.0000000000000002 ***
colorH      -0.2574434   0.0029696   -86.693 < 0.0000000000000002 ***
colorI      -0.3805250   0.0033493  -113.614 < 0.0000000000000002 ***
colorJ      -0.5117734   0.0041317  -123.866 < 0.0000000000000002 ***
clarityIF    1.0929196   0.0080869   135.147 < 0.0000000000000002 ***
claritySI1   0.5812847   0.0069179   84.027 < 0.0000000000000002 ***
claritySI2   0.4135329   0.0069516   59.487 < 0.0000000000000002 ***
clarityVS1   0.7974239   0.0070588   112.969 < 0.0000000000000002 ***
clarityVS2   0.7280054   0.0069551   104.672 < 0.0000000000000002 ***
clarityVVS1  1.0031580   0.0074744   134.212 < 0.0000000000000002 ***
clarityVVS2  0.9270567   0.0072663   127.583 < 0.0000000000000002 ***
depth       0.0588389   0.0006991    84.161 < 0.0000000000000002 ***
table       0.0088152   0.0004586    19.221 < 0.0000000000000002 ***
x           1.3505027   0.0048369   279.208 < 0.0000000000000002 ***
y           0.0135886   0.0024069     5.646  0.0000000166 ***
z           0.0405002   0.0046997     8.618 < 0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1384 on 32329 degrees of freedom
Multiple R-squared:  0.9814,    Adjusted R-squared:  0.9814
F-statistic: 7.436e+04 on 23 and 32329 DF,  p-value: < 0.0000000000000002
```

Figure 134: Linear Regression

The regression results above show us that our model is indeed good and can provide ideal results. In statistical terms, the estimated coefficient explains the change on the price based on the change on the independent variable. For example the coefficient of the carat equals -0.99 which means, if the carat increased by one unit the log(price) will decrease by almost 1, which means the price will increase by $e^{-0.99} = \$0.372$. For the categorical variables; R used dummy variable to illustrate the effect of each category. For example, there are 4 dummy variables associated with cut, that has 5 categories, the log price of ideal cut is better by 0.16 comparing the log price of good cut, by the money language ideal cut is more expensive by $e^{0.16} = \$1.17$ than good cut, when other variables hold.

R square (R²) measures the ratio of variation explained by regression model to the total variation. In other words, it shows how well the data is being explained by the regression model. A higher value of R² and adjusted R² means that the regression model is good. R² & Adjusted R² = 0.9814 which means about 98% of the variation in the log(price) is explained by the model.

The predictors for which p-value are large should not be included in the model. According to the regression results, in this model all the p-values are too small which indicate that all the predictors are significant.

Checking the model:

1- stepwise

Even though we have ample proof to solidify our belief in the strength of this model, we will use Stepwise method as a validation to ensure there is no possibility of a better model.

The stepwise regression (or stepwise selection) is a process of iteratively adding and removing predictors, in the predictive model. It is used to find the subset of variables in the data set resulting in the best performing model, where the errors would be minimal.

```
model1 <- lm(logy ~., data= dim.train1)
summary(model1)
```

Output:

```
Start:  AIC=-127949.2
logy ~ carat + cut + color + clarity + depth + table + x + y +
      z
```

	Df	Sum of Sq	RSS	AIC
<none>			619.05	-127949
- y	1	0.61	619.66	-127919
- z	1	1.42	620.47	-127877
- table	1	7.07	626.13	-127584
- cut	4	21.33	640.38	-126861
- depth	1	135.63	754.68	-121542
- carat	1	284.82	903.88	-115706
- color	6	535.76	1154.81	-107789
- clarity	7	1075.69	1694.74	-95381
- x	1	1492.76	2111.81	-88251

Figure 145: Step wise method

Stepwise method agrees that all the variables are significant in predicting the log(price) of the diamonds.

```
step <- step(model1, direction="both")
par(mfrow = c(2, 2))
plot(model1)
```


2- Residuals:

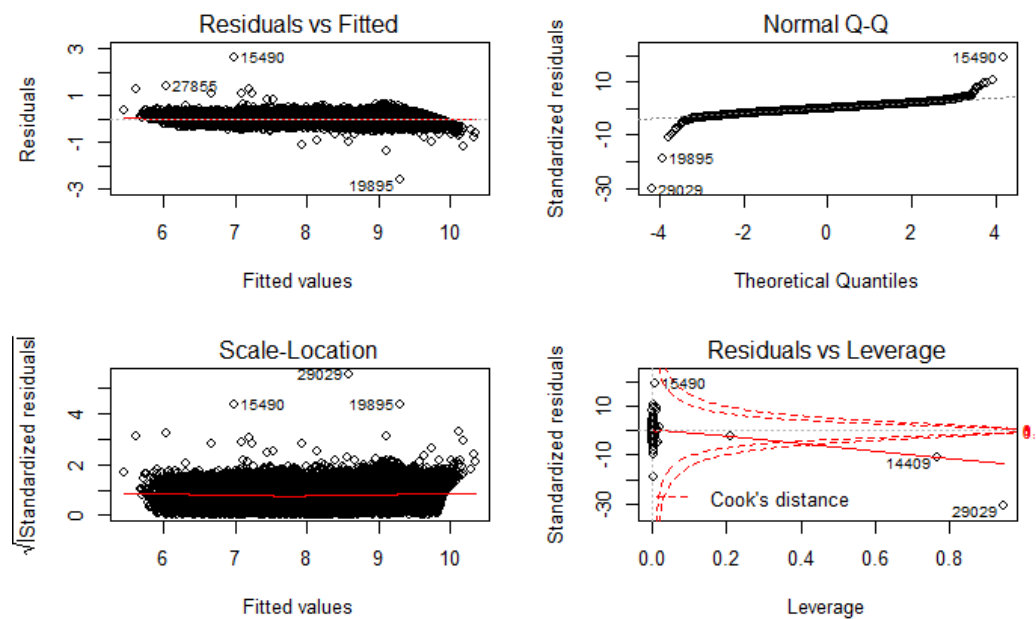


Figure 156: residuals vs Leverage plot

The residuals look perfect! They follow normal distribution from QQ plot with few outliers, the variance seems constant based on the scale location, a linear relationship assumption by comparing residuals vs. fitted values since there is a clear horizontal line with no obvious trend.

Therefore, the multiple linear regression model is good to predict the $\log(\text{price})$ of the diamond based on the predictors, using exponential function, the real price could be calculated.

LINEAR DISCRIMINANT ANALYSIS

Linear Discriminant Analysis (also abbreviated as LDA) is a commonly used technique for data classification and dimensionality reduction (i.e., reducing the number of predictors in the model). It focuses on maximizing the separability among the known labels/categories. It helps towards maximizing the separation between classes and minimizing the variance within the class thereby guaranteeing increased separation. Some popular use cases involve face recognition, Bankruptcy prediction.

The use of Linear Discriminant analysis for data classification is applied in classifying the diamonds. We decided to implement this classification algorithm in hopes of achieving better classification compared to the Principal Component Analysis (PCA) performed earlier. As we may know, the PCA does more of feature classification and LDA does data classification. We have tried to make use of this feature in our model by classifying the diamonds into derived price categories/classes.

We derived the price categories (low, medium, expensive, & extremely expensive) from the price field in the data. In order to avoid dominance of any single field while running the model, we normalized

the dataset and dropped the price field (as we already created labels out of it). Further, we implemented the algorithm using the R function and following is the output.

From the output, we have the prior group probabilities, group means, coefficients linear discriminants, & Proportion of trace. We are interested in the 'Proportion of trace' value and we can see that LD1 is **94.15%** successful in separating the data labels w.r.t LD2(0.0823) and LD3(0.035).

```
fivenum(diamonds$price)
diamonds$priceCat[diamonds$price < 949 ] <- "low"
diamonds$priceCat[diamonds$price >= 949 & diamonds$price <2401 ] <- "medium"
diamonds$priceCat[diamonds$price >= 2401 & diamonds$price <5323 ] <- "expensive"
diamonds$priceCat[diamonds$price >= 5323 ] <- "extremely expensive"

# dropping the price column
diamonds.p <- diamonds[,-7]
set.seed(123)
training.index <- createDataPartition(diamonds.p$priceCat, p = 0.6, list = FALSE)
dim.train <- diamonds.p[training.index, ]
dim.valid <- diamonds.p[-training.index, ]

# normalize the data
# Estimate preprocessing parameters
norm.values <- preprocess(dim.train, method = c("center", "scale"))

# Transform the data using the estimated parameters
dim.train.norm <- predict(norm.values, dim.train)
dim.valid.norm <- predict(norm.values, dim.valid)

# run lda
lda2 <- lda(priceCat~., data = dim.train.norm)

# output
lda2

# predict - using training data and plot
pred2.train <- predict(lda2, dim.train.norm)
```

```
# generate lda plot
lda2.plot <- cbind(dim.train.norm, predict(lda2)$x)
ggplot(lda2.plot, aes(LD1, LD2)) + geom_point(aes(color = priceCat))

# LDA hist|
par(mar=c(1,1,1,1))
ldahist(pred2.train$x[,1],g = dim.train.norm$priceCat)
ldahist(pred2.train$x[,2],g = dim.train.norm$priceCat)
ldahist(pred2.train$x[,3],g = dim.train.norm$priceCat)

# predict - using validation data
pred2 <- predict(lda2, dim.valid.norm)

# check model accuracy
table(pred2$class, dim.valid.norm$priceCat) # pred v actual
mean(pred2$class == dim.valid.norm$priceCat) # percent accurate
```

Output:

```
Call:
lda(priceCat ~ ., data = dim.train.norm)

Prior probabilities of groups:
      expensive 0.2500850
      extremely 0.2501159
      low       0.2495597
      medium    0.2502395

Group means:
      carat      cutGood      cutIdeal      cutPremium      cutVery Good      colorE      colorF      colorG      colorH      colorI
expensive      0.2312546 0.12194218 0.3197430 0.2578453 0.2524092 0.1834692 0.1891525 0.1643192 0.1712380 0.10872251
extremely expensive 1.3236081 0.08387894 0.3425571 0.3179741 0.2265596 0.1201977 0.1581223 0.2322304 0.1830760 0.14193947
low            -0.9874600 0.07700879 0.4618051 0.2166646 0.2380834 0.2096075 0.1707317 0.2156741 0.1457224 0.09025628
medium        -0.5692885 0.08124460 0.4693172 0.2275590 0.1881714 0.2134831 0.2006421 0.2217558 0.1069268 0.06753920
      colorJ      clarityIF      claritySI1      claritySI2      clarityVS1      clarityVS2      clarityVVS1      clarityVVS2      depth      table
expensive      0.05819125 0.01284902 0.3047937 0.30429948 0.1063751 0.1614776 0.03508772 0.05040771 0.0642392757 0.1865201
extremely expensive 0.07856702 0.02137122 0.2290303 0.19221742 0.1645460 0.2664608 0.03360099 0.07918468 -0.0187564260 0.1398175
low            0.02946639 0.04197103 0.2195122 0.07799926 0.1708555 0.2515786 0.10053238 0.13297016 0.0001011355 -0.2203736
medium        0.03914063 0.05247561 0.2122484 0.11544635 0.1652056 0.2308927 0.09877763 0.11063094 -0.0455533153 -0.1063785
      x      y      z
expensive      0.3926089 0.3903002 0.3856663
extremely expensive 1.2727260 1.2638390 1.2288248
low            -1.1599113 -1.1495573 -1.1224642
medium        -0.5077033 -0.5068394 -0.4942309

Coefficients of linear discriminants:
      LD1      LD2      LD3
carat      2.22227435 -5.386351739 -1.07457538
cutGood    -0.32183008 -0.365991166 -1.15097895
cutIdeal   -0.44330805 -0.470324018 -0.71576435
cutPremium -0.32114394 -0.629895221 -0.82465983
cutVery Good -0.41069131 -0.492973184 -1.42051515
colorE     0.12832379 0.048142712 -0.22835932
colorF     0.25983921 -0.002291062 -0.31073407
colorG     0.46876681 -0.223458829 -0.23539466
colorH     0.70399979 -0.018600975 -1.14067852
colorI     1.07189744 0.140066795 -1.04049182
colorJ     1.45573901 0.486200884 -0.51690219
clarityIF  -3.00225649 -1.304514259 2.34633551
claritySI1 -1.50552220 -1.329372121 -0.08810821
claritySI2 -0.99495357 -0.621882317 -0.86529460
clarityVS1 -2.18156147 -1.734880541 1.00072885
clarityVS2 -1.98322851 -1.762734840 0.82364876
clarityVVS1 -2.76736784 -1.327295933 1.69504932
clarityVVS2 -2.52603186 -1.664254846 1.10244607
depth      -0.31827614 0.296470807 -0.14661154
table      -0.07173441 0.162716839 -0.14253528
x          -5.07370486 4.555014910 1.57507704
y          -0.44669433 0.369071891 -0.42647503
z          -0.05587520 0.048331216 0.12915734

Proportion of trace:
      LD1      LD2      LD3
0.9415 0.0549 0.0036
```

Figure 167: Linear Discriminant Analysis

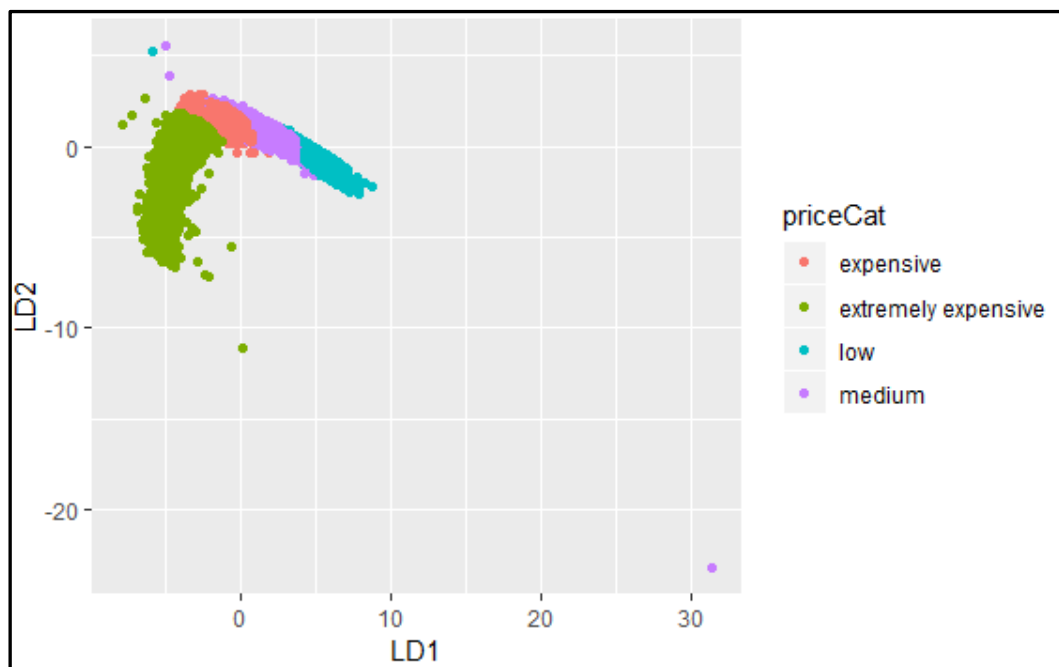


Figure18: Plot to show separation achieved using LDA

Let us visualize the separation achieved by LD1 and LD2 from the figure above. We can see along x-axis that LD1 has been successful in separating of the price categories whereas LD2 is not helping in

achieving the same. We knew this from the 'Proportion of trace' results (from Figure **) of both LD1 and LD2.

Following are the histograms for LD1, LD2, & LD3 to visualize the separation between each category. Looking at the figure 20, we can see that LD1 has managed to separate the classes to a greater extent. However, there are few overlaps between expensive and extremely expensive class and a chunk of overlap between low and medium class price. From figure 21 and 22, we see that these two coefficients are not helping in discriminating the price categories. We see there is a lot of overlap between all the 4 price categories in the data and doesn't provide significant separation.

Confusion Matrix

	expensive	extremly expensive	low	medium
expensive	5123	777	1	557
extremly expensive	93	4615	0	0
low	2	4	5112	614
medium	178	0	271	4228

Figure 17: Confusion Matrix

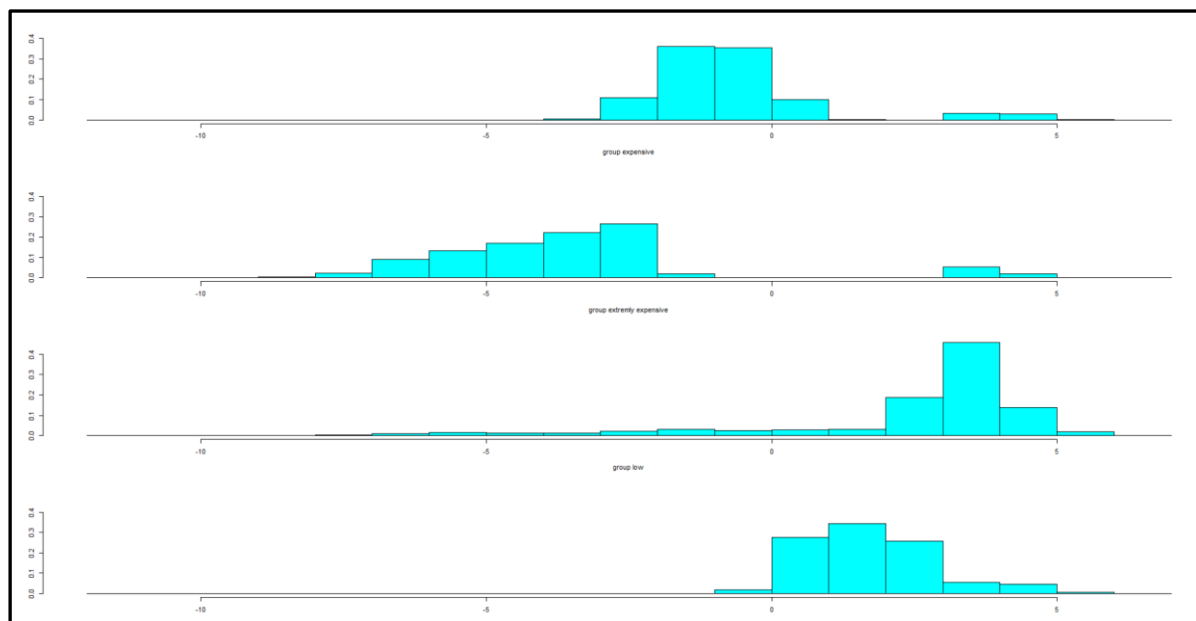


Figure 180: Histogram plot of LD1

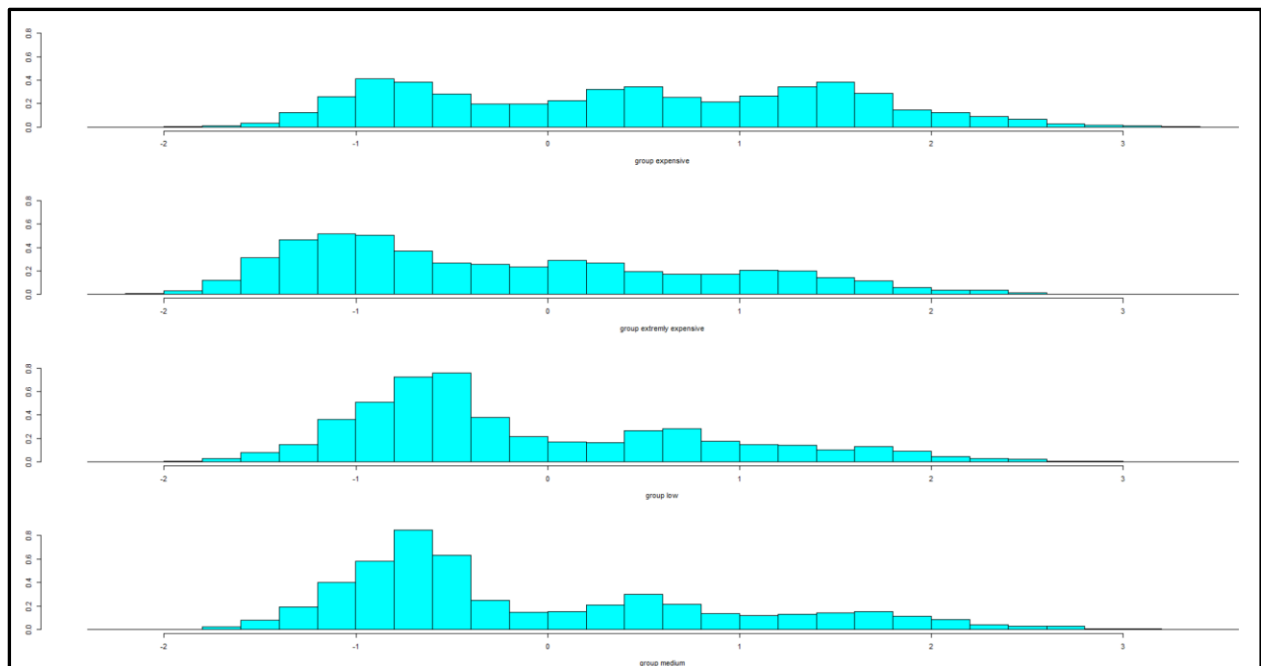


Figure 191: Histogram plot of LD2

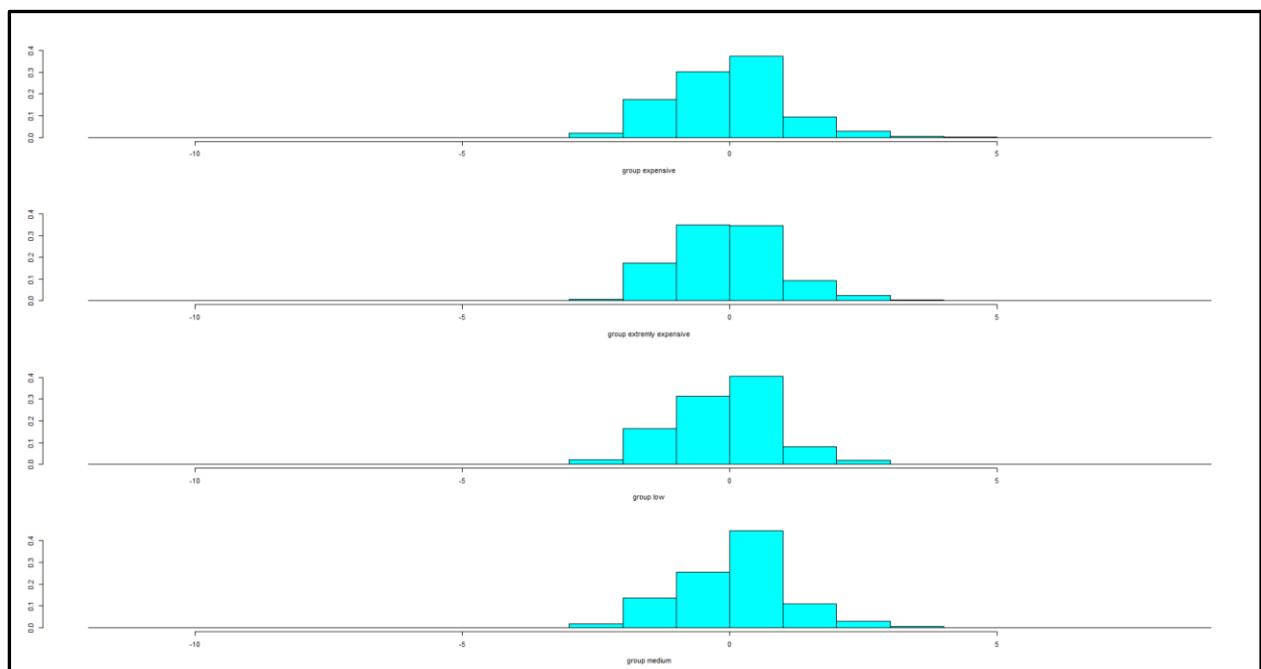


Figure 202: Histogram plot of LD3

The confusion matrix obtained by running the model on the validation dataset. From the matrix, it's evident that the model performance is good. Though we have seen from the plots that there are some anomalies in the output of the model, but the confusion matrix says it all. The diagonal values of the matrix suggest that each class has been predicted as desired. The other values in the matrix are values that have been wrongly predicted/identified by the model in other groups. After looking this, the immediate question comes to mind is the accuracy of the model. We have got an accuracy of **88.42%** from the validation dataset.

So, we have been successful in implementing and categorizing the price categories of diamonds based on the predictors with an accuracy of 88.42%

RECOMMENDATIONS

1. People can foresee the trends in the prices of diamonds in the near future using the prediction models we have designed and plan their lavish investments accordingly.
2. Clarity and Carat value are two factors that play a sizeable role while purchasing a diamond. While investing on a diamond compromising on clarity then on carat value would be a good cost cutting ploy.
3. To enrich the dataset for fitter analysis, year and month of purchase for diamonds should also be taken into consideration. A newly designed predictive model using this seasonal information will help us in assimilating a healthy seasonality trend.
4. Implementing other variables that may affect the price of the diamond. Could the store name effect the price? for example, is the same ring in Tiffani &Co. has the same price of second-hand used jewellery store? How about the age of the diamond? Does it affect the price?

CITATIONS

Dataset:

<https://www.kaggle.com/fuzzywizard/diamonds-in-depth-analysis/data>

Linear Regression:

https://en.wikipedia.org/wiki/Linear_regression

Linear Discriminant Analysis:

https://www.isip.piconepress.com/publications/reports/1998/isip/lda/lda_theory.pdf

https://en.wikipedia.org/wiki/Linear_discriminant_analysis

APPENDIX

R code:

```
#Diamond In Depth Analysis
```

```
#Used packages:
```

```
library(utils)
```

```
library(dplyr)
```

```
library(MASS)
```

```
library(gtools)
```

```
library(lattice)
```

```
library(ggplot2)
```

```
library("corrplot")
```

```
library(GGally)
```

```
library(caret)
```

```
library(RColorBrewer)
```

```
library(klaR)
```

```
#Upload data:
```

```
diamonds <- read.csv("diamonds.csv")
```

```
attach(diamonds)
```

```
head(diamonds)
```

```
# Check if there is missing values:
```

```
diamonds[!complete.cases(diamonds),]
```

```
diamonds <- subset(diamonds, z > 0 )
```

```
diamonds <-diamonds[,-1]
```

```
#Descriptive statistics:
```

```
summ1 <- data.frame(mean=sapply(diamonds, mean),
```

```
                      sd=sapply(diamonds, sd),
```

```
                      min=sapply(diamonds, min),
```

```
                      max=sapply(diamonds, max),
```

```
                      median=sapply(diamonds, median),
```

```

length=sapply(diamonds, length),
miss.val=sapply(diamonds, function(x)
sum((is.na(x))))
options(scipen = 999)
print(summ1, digits=1)

#Visualising data:
#Bar Chart for Categorical Variable
#Color:
colfunc <- colorRampPalette(c("navy", "white"))
barplot(table(diamonds$color), main = "color distribution", xlab = "color", col = colfunc(7))

# Cut:
colfunc <- colorRampPalette(c("green", "white"))
barplot(table(diamonds$cut), main = "cut distribution", xlab = "color", col = colfunc(5))

# Clarity:
colfunc <- colorRampPalette(c("white", "black"))
barplot(table(diamonds$clarity), main = "clarity distribution", xlab = "color", col = colfunc(11))

# Comparing categorical variables:
ggplot(data = diamonds) + geom_bar(mapping = aes(x = cut, fill = clarity),alpha = 1/2, position =
"dodge")

#Relation of categorical variable with the price:
ggplot(diamonds, aes(x=color, y=price, fill=color)) + geom_boxplot()+
scale_fill_brewer(palette="Set1")
ggplot(diamonds, aes(x=cut, y=price, fill=cut)) + geom_boxplot()+ scale_fill_brewer(palette="Set2")
ggplot(diamonds, aes(x=clarity, y=price, fill=clarity)) + geom_boxplot()+
scale_fill_brewer(palette="Set3")

# Creating a variable volume using diamond dimensions x,y and z
diamonds$volume<-diamonds$x*diamonds$y*diamonds$z
volume<-lm(price~volume, data = diamonds)

```



```

summary(volume)

ggplot(aes(x = price, y = volume), data = diamonds)+geom_point(alpha = 0.5, size = 1)


#Scatter Plot and correlation matrix for quantitative variables:

diamondsM <- diamonds[,c(1,5:10)]
cor.matrix <- cor(diamondsM)
corrplot(cor.matrix)


ggpairs(diamonds[, c(1,5:10)])


# Divide the Dataset into Train and Validation sets
# Train for fitting algorithm, and Validation for checking.
set.seed(123)
training.index <- createDataPartition(diamonds$price, p = 0.6, list = FALSE)
dim.train <- diamonds[training.index, ]
dim.valid <- diamonds[-training.index, ]


# Principal Components Analysis
pca <- princomp(diamondsM)
summary(pca,loadings=TRUE)


# Linear regression

ggplot(data=dim.train, aes(price)) + geom_histogram(fill = "firebrick") + ggtitle("Frequency distripution
of the price")

dim.train1<-dim.train %>% mutate(logy=log(price))
dim.train1<-dim.train1[,-7]

ggplot(data=dim.train1, aes(logy)) + geom_histogram(fill = "firebrick") + ggtitle("Frequency
distripution of the log(price)")


model <- lm(price ~., data= dim.train)

model

boxcox(model)

```

```

model1 <- lm(logy ~., data= dim.train1)
summary(model1)
step <- step(model1, direction="both")

#Checking the model:
par(mfrow = c(2, 2))
plot(model1)

#Linear Discriminant Analysis:
fivenum(diamonds$price)
diamonds$priceCat[diamonds$price < 949 ] <- "low"
diamonds$priceCat[diamonds$price >= 949 & diamonds$price <2401 ] <- "medium"
diamonds$priceCat[diamonds$price >= 2401 & diamonds$price <5323 ] <- "expensive"
diamonds$priceCat[diamonds$price >= 5323 ] <- "extremely expensive"

# Dropping the price column
diamonds.p <- diamonds[,-7]
set.seed(123)
training.index <- createDataPartition(diamonds.p$priceCat, p = 0.6, list = FALSE)
dim.train <- diamonds[training.index, ]
dim.valid <- diamonds[-training.index, ]

# Normalize the data
# Estimate preprocessing parameters
norm.values <- preProcess(dim.train, method = c("center", "scale"))

# Transform the data using the estimated parameters
dim.train.norm <- predict(norm.values, dim.train)
dim.valid.norm <- predict(norm.values, dim.valid)

# Run lda
lda2 <- lda(priceCat~., data = dim.train.norm)

```

```
# Output
```

```
lda2
```

```
# Predict - using training data and plot
```

```
pred2.train <- predict(lda2, dim.train.norm)
```

```
# Generate lda plot
```

```
lda2.plot <- cbind(dim.train.norm, predict(lda2)$x)
```

```
ggplot(lda2.plot, aes(LD1, LD2)) + geom_point(aes(color = priceCat))
```

```
# Predict - using validation data
```

```
pred2 <- predict(lda2, dim.valid.norm)
```

```
# Check model accuracy
```

```
table(pred2$class, dim.valid.norm$priceCat) # prediction vs actual
```

```
mean(pred2$class == dim.valid.norm$priceCat) # percent accuracy
```

```
# LDA hist
```

```
ldahist(pred2.train$x[,1], g = dim.train.norm$priceCat)
```

```
ldahist(pred2.train$x[,2], g = dim.train.norm$priceCat)
```

```
ldahist(pred2.train$x[,3], g = dim.train.norm$priceCat)
```

```
# Data partition
```

```
AS <- as.data.frame(dim.train.norm)
```

```
AS$priceCat <- as.factor(AS$priceCat)
```

```
partimat(formula = priceCat ~ carat + depth, method = "lda", prec = 200, data = AS, nplots.hor = 1, nplots.vert = 1)
```

Individual Report

Amey Vanmali

My project team members are Norah Alyabs, Ketki Chaudhary, Rohit Gurjar, & Ashwin Krishnamoorthy and myself. The first task in any team project needs to search a considerable topic to start with and so we started searching for a good topic on Kaggle. After searching for number of topics, we shortlisted 3 potential topics that the team showed interest on working. They are as follows:

1. Predict and analyse who will be the best soccer player in FIFA 2018
2. Analysis of applications in App store
3. Predict Diamond price based on some variables

We started our initial findings with the first two topics independently. Moving forward, Ketki and I worked on the data exploration for topic 2 and found that it would not be feasible to proceed with it. After facing challenges with the first two topics, the team decided to move on to the third topic.

We found the dataset for the Diamonds from Kaggle website. The initial findings started, and we realized that we can go ahead with this topic as the dataset had considerable information to work on our objective of predicting price of the diamonds.

As a team, we all had one thing to look for and that was to implement the topics taught in the syllabus and learn to interpret them wisely by working on a real case problem. I worked on the aspect of trying to reduce the dimension by working on Principal component analysis. Me and Norah had discussion on the results obtained by running PCA on the data. From the results, it turned out that the results are less effective and hence there was no reduction in the number of variables in further analysis. Working on the PCA helped me a lot on improving my knowledge because I tried to go through online videos and correct my thoughts regarding this technique.

Further, I was working with Norah on implementing Linear Discriminant Analysis (LDA) method in hope to predict the price category (low, medium, expensive, extremely expensive) derived from the price variable using the numerical predictors in the data. We did succeed in classifying a given record into the derived price categories using this method. We also tested the model on the validation dataset and got an overall accuracy of 87%. In the process, I got to learn different visualization methods and various plots. Unfortunately, couldn't include everything because of constraint in the dataset. While creating the final report for the project, I wrote the inferences for the visualizations and gave background about the same.

To sum up, I had a great time working on this project with my peers. It also gave me a good experience of working in a healthy team environment where each one of us is enthusiastic and wants to contribute to the project effectively.