What Drives Diamond Prices?

A Data-Driven Classification Analysis

By: Allysa M. Wibowo

Table of Contents

Abstract

Diamond pricing is strongly influenced by a combination of physical and quality-related characteristics, making it an important area of analysis within the jewellery and luxury goods market. This study aims to investigate the key factors that contribute to high diamond prices and to identify patterns that can be used for effective price classification. Using a structured dataset containing diamond attributes such as carat, cut, colour, clarity, depth, table, and dimensions, the analysis applies exploratory data analysis and classification techniques to better understand pricing behaviour.

The study employs several quantitative classification models, including Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, K-Nearest Neighbours, and a Neural Network, to classify diamonds into high and non-high price categories. The findings highlight the significant role of size-related variables and quality grades in determining price classification outcomes. Results show that traditional statistical classification models perform particularly well for this dataset, achieving high predictive accuracy.

The report emphasizes the importance of data exploration, appropriate variable selection, and model comparison when conducting classification analysis. By understanding how diamond characteristics influence price categories, this study provides insights that can support pricing strategies, valuation processes, and decision-making within the diamond industry.

1.0 Introduction

This study aims to analyse how diamond characteristics influence whether a diamond belongs to a high price category using classification models and data driven interpretation. Diamonds play a significant role in the global luxury and jewellery market and are valued for their rarity, quality, and physical attributes. Diamond prices vary substantially due to a combination of factors, making pricing a complex and important process within the industry. Key determinants of diamond value are commonly referred to as the “4Cs”, carat, cut, colour, and clarity, which form a globally recognised grading system that provides objective and consistent quality assessment. In addition to these quality measures, physical attributes such as depth, table, and overall dimensions also contribute to price variation. While consumer choice may be influenced by personal preference, accurate diamond valuation requires objective and standardised evaluation.

Accurate pricing is essential for fair valuation, effective inventory management, and risk reduction in trading and retail environments. Mispricing can lead to revenue loss and reduced consumer trust, highlighting the importance of identifying the characteristics that distinguish high priced diamonds from lower priced ones. Rather than predicting exact prices, this study focuses on classifying diamonds into high priced and non-high priced categories, an approach that is often more practical and actionable in real world decision making. Price classification supports premium product identification, pricing strategy development, and high value inventory prioritisation.

The analysis is conducted using a dataset containing both numerical and categorical variables related to diamond quality and size. Exploratory Data Analysis (EDA) is performed to understand the data structure, identify key relationships, and ensure data reliability through

comprehensive quality checks. Following data preparation, multiple classification models are applied and compared, including Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, K Nearest Neighbours, and a Neural Network. The findings provide insight into how diamond characteristics influence price categories and offer practical value for industry stakeholders by supporting informed pricing decisions and improving understanding of value driven diamond attributes.

2.0 Data Description and Pre-Processing

2.1 Reading the data

The data set used in this analysis consists of 53,794 observations after the data cleaning process and contains a total of ten variables describing diamond characteristics. These variables include numerical features such as carat, depth, which represents the height of the diamond measured from the table (top surface) to the culet (bottom point), and table, which refers to the flat top surface of the diamond measured as a percentage of the diamond's average width. The variables x and y represent the length and width of the diamond respectively, while z measures the diamond's depth in millimetres, capturing its vertical dimension. In addition to these numerical measurements, the dataset also contains categorical variables representing the previously mentioned 4Cs of diamond quality: cut, colour, and clarity. This combination of numerical and categorical data is particularly well suited for classification analysis, as it allows the models to incorporate both objective measurements and quantitative grading standards. Including variables that reflect size, structure, and quality enables the classification models to better distinguish between high priced and non-high priced diamonds, making the dataset appropriate and informative for this study.

2.2 Data cleaning

A data cleaning process was carried out to ensure the quality and reliability of the dataset prior to the processing and analysis. An initial inspection confirmed that there are no missing values across any of the variables. However, 146 duplicated observations were identified and subsequently removed from the dataset. Removing duplicate rows are particularly important in classification analysis, as repeated observations a]can artificially inflate model accuracy and bias distance based methods such as the K-Nearest Neighbours classification. Additionally, the presence of duplicates increases the risk of data leakage between training and resting sets, which can lead to overly optimistic performance estimates. By eliminating duplicated records, the dataset is better suited for building classification models that generalise reliably to unseen data.

2.3 Converting categorical variables

Categorical variables in the dataset, namely cut, colour, and clarity were converted into factor variables prior to model implementation. This step is essential for classification analysis, as many statistical classification models are designed to treat categorical predictors as grouped categories rather than numerical values. Without converting these variables into factors, models such as Logistic Regression, Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA) would incorrectly interpret category levels as continuous numerical quantities, leading to invalid assumptions and misleading results. Ensuring that

categorical variables are properly encoded allows the models to correctly capture differences between quality grades and improves the overall validity and interpretability of the classification outcomes.

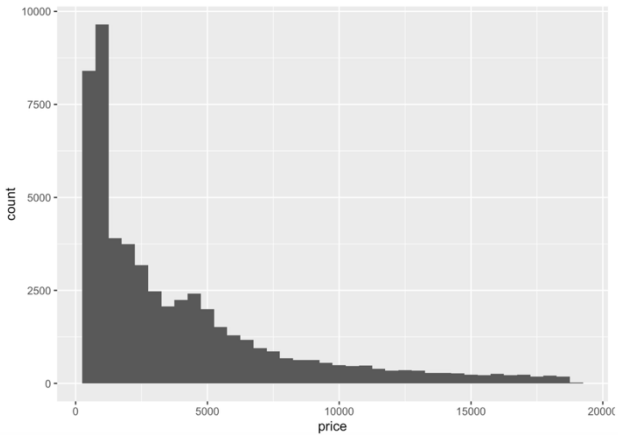
3.0 Explanatory Data Analysis

3.1 Introduction

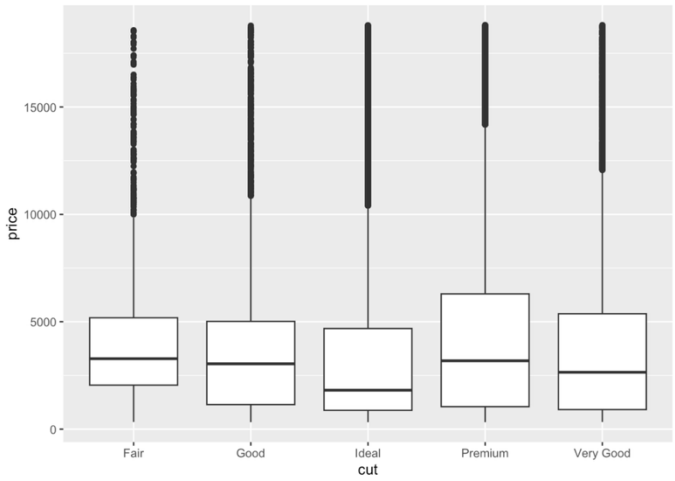
Explanatory Data Analysis (EDA) is used to gain an initial understanding of the dataset by summarising its main characteristics and identifying key patterns, trends and anomalies. In this study EDA helps examine the distribution of variables and explore relationships between diamond characteristics and price. This process is useful as it informs subsequent modelling decisions, such as variable selection and model choice. By visually and statistically exploring the data, EDA provides valuable insights that improve the reliability and interpretability of the final results.

3.2 Data Analysis

3.2.1 Distribution of Diamond Prices

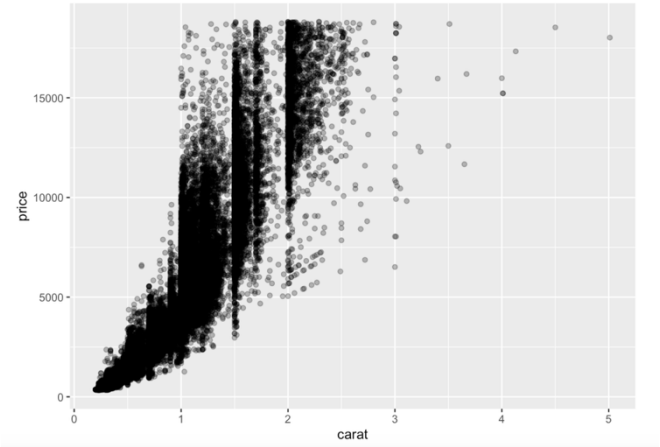


The histogram of diamond prices shows highly right skewed distribution, with most diamonds concentrated at lower price levels and a long tail of very high priced diamonds. This indicates that diamond prices are not evenly distributed and that only a small portion of diamonds fall into the premium price range. This result is important because it motivates the decision to transform price into a binary classification variable, distinguishing between high-priced and non-high-priced diamonds. Such a classification approach is more practical in real-world settings, where identifying premium products is often more valuable than predicting exact prices.



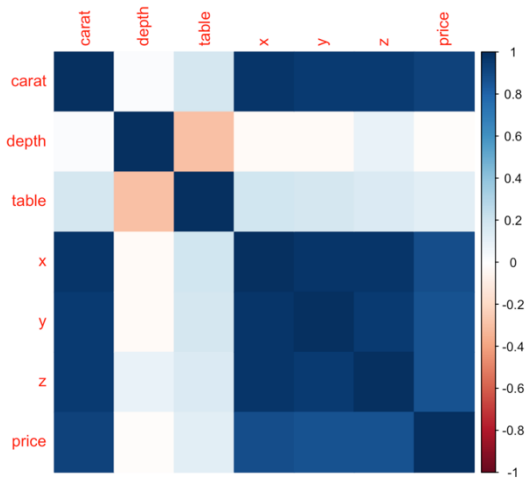
3.2.2 Relationship Between Carat and Price

The scatterplot of carat versus price reveals a strong positive relationship, where diamond prices increase substantially as carat weight increases. The relationship is clearly non-linear, with prices rising more sharply for larger diamonds. This highlights carat as one of the most influential predictors of price and suggests that size plays a dominant role in determining whether a diamond belongs to the high-price category. The observed non-linearity also indicates that flexible classification models may be better suited to capturing this relationship.



3.2.3 Effect of Cut on Price

The boxplot comparing diamond prices across different cut categories demonstrates that cut quality has a noticeable impact on pricing. Diamonds with higher-quality cuts, such as Premium and Ideal, tend to have higher median prices compared to lower-quality cuts. Although there is some overlap between categories, the difference in price distributions suggest that cut provides meaningful information for distinguishing between price groups. This supports the inclusion of cut as a categorical predictor in the classification models and reflects the real-world importance of quality grading in diamond valuation.



3.2.4 Correlation Between Numerical Variables

The correlation matrix of numerical variables highlights strong positive correlations between price and size-related features, including carat and the physical dimensions x, y, and z. This indicates that larger diamonds are generally more valuable and confirms that size-related variables are key drivers of its price. In contrast, depth and table show weaker correlations with price, suggesting that there influence is less direct. This analysis is important because it identifies potential multicollinearity among predictors and help explain the performance of different classification models used later in the study.

3.3 Conclusion

Overall, the Explanatory Data analysis Provided a clear understanding of the key characteristics and relationships within the diamond dataset. The analysis showed that diamonds prices are highly skewed, are strongly influenced by carat size, vary noticeably across different cut categories, while size related variables exhibit strong correlations with price. These findings confirm both physical dimensions and quality attributes play an important role in determining price categories. By identifying influential predictors and potential multicollinearity, the EDA informed appropriate feature selection and model choice, ensuring that the subsequent classification analysis is based on well-understood data patterns.

4.0 Creating the Classification Target

4.1 Introduction

In order to apply classification models, a categorical response variable must be defined. Since the original price variable is continuous and highly skewed, converting it into a binary outcome allows the analysis to focus on distinguishing premium diamonds from the rest of the market. This approach reflects real-world pricing practices, where diamonds are often grouped into price segments rather than evaluated solely by exact price values. Similarly, the Rapaport Diamond Report which provides benchmark price lists and segmented pricing grids for polished diamonds, with prices listed by carat size, colour, and clarity categories rather than as

single continuous values.[1]The aim of this section is to define a meaningful and practical classification target based on diamond price.

4.2 Approach

The classification target was constructed using a quantile-based threshold applied to the diamond price variable. Specifically, the 75th percentile of the empirical price distribution was selected as the cutoff, such that diamonds with prices greater than or equal to this threshold were classified as "High", and all remaining observations were classified as "NotHigh". This approach ensures that the classification boundary is determined endogenously from the data, rather than relying on an arbitrary price level that may not generalise across different market conditions.

The choice of a quantile-based threshold is particularly appropriate given the pronounced right-skewness observed in the price distribution during Exploratory Data Analysis. In markets such as the diamond industry, price distributions are typically heavy-tailed, with a small proportion of premium products accounting for a disproportionate share of total value[2]. Defining high-priced diamonds relative to the upper tail of the distribution mirrors real-world market segmentation practices, where premium diamonds are identified based on their position within the broader price spectrum rather than absolute price alone.

From a modelling perspective, the resulting class distribution of approximately 25% High and 75% NotHigh provides a balanced compromise between representativeness and model stability. This level of class imbalance reflects realistic market conditions while remaining suitable for supervised classification algorithms, reducing the risk of biased predictions toward the majority class. The binary target variable, price_class, therefore provides a robust and interpretable outcome measure that supports consistent training, evaluation, and comparison across multiple classification models.

4.3 Conclusion

Defining the classification target using a quantile-based approach provides a robust and defensible framework for price categorisation. By identifying the top 25% of diamonds as high-priced, the analysis aligns with practical market segmentation used in retail and valuation contexts. The resulting class balance is suitable for classification models and avoids extreme imbalance, allowing reliable model training and evaluation. This target definition establishes a clear and meaningful outcome variable for the subsequent classification analysis.

[1] Rapaport Group. (2007, March 1). *A guide to the Rapaport price lists*. https://rapaport.com/magazine-article/guide-to-the-rapaport-price-lists/

[2] Clauset, A., Shalizi, C. R., & Newman, M. E. J. (2009). *Power-law distributions in empirical data*. SIAM Review, 51(4), 661–703. https://doi.org/10.1137/070710111

5.0 Model Building Strategy

5.1 Introduction

This section outlines the strategy used to prepare the dataset for classification modelling. A structured modelling framework is essential to ensure that model evaluation is reliable, unbiased, and comparable across different classification techniques. Two key components of this strategy are the use of a train–test split and the careful selection of predictor variables. Together, these steps help prevent overfitting, reduce bias, and ensure that the models learn meaningful relationships between diamond characteristics and price categories.

5.2 Approach

5.2.1 Train–Test Split

To assess classification performance in an unbiased manner, the dataset was randomly portioned into disjoint training subsets. Specifically in the study, 70% of the observations were assigned to do the training set for model estimation, whereas the remaining 30% were held out as an independent test set for performance evaluation. This sampling strategy ensures that model parameters are learned from a sufficiently large dataset while preserving an adequate number of unseen observations to reliably estimate generalisation error. Evaluating model predictions on the held-out test set mitigates the risk of overfitting and provides an empirical approximation of out-of-sample performance, thereby offering a realistic assessment of how the classification models are expected to perform on new data.

5.2.2 Predictor Selection

The predictor variables were selected to capture the key physical and quality-related characteristics that influence diamond pricing. These include numerical variables such as carat, depth, table, and the physical dimensions x, y, and z, as well as categorical variables representing diamond quality: cut, colour, and clarity. The original price variable was explicitly excluded from the predictor set. Including price as an input would result in data leakage, as the classification target is directly derived from price, leading to invalid model performance and misleading conclusions. Restricting the predictors to independent diamond attributes ensures that the classification models are learning genuine relationships rather than relying on information from the target variable itself.

5.3 Conclusion

The model building strategy establishes a foundation for the classification analysis. The use of a train–test split ensures that model performance is evaluated fairly and generalises beyond the training data, while careful predictor selection prevents data leakage and supports meaningful interpretation. Together, these methodological choices ensure that the classification results are statistically valid, comparable across models, and relevant to real-world diamond pricing and valuation contexts.

6.0 Classification Models and Interpretation

6.1 Introduction

This section presents the results obtained from applying multiple classification models to the dataset. Model performance is evaluated using classification accuracy on the held-out test set and each model is interpreted in terms of its underlying assumptions, predictive behaviour, and suitability for the structure of the data. Comparing models with varying levels of complexity allows for a robust assessment of the trade-off between interpretability and predictive performance.

6.2 Approach

6.2.1 Logistic Regression

Logistic regression is used in this study as a baseline parametric classification model to quantify the relationship between diamond characteristics and the probability of a diamond belonging to the high-price category. Despite its simplicity, logistic regression proves highly effective when the underlying data structure is well defined, making it an attractive choice for interpretable and reliable classification in pricing applications.

Logistic regression demonstrates the highest classification accuracy of 97.3% amongst all models considered. This strong performance indicates that the relationship between the predictor variables and the binary price classification is strongly structured and largely linearly separable in the feature space. The estimated coefficients exhibit large magnitudes for key predictors such as carat and physical dimensions, reflecting their dominant influence on the probability of a diamond being classified as high-priced. The presence of predicted probabilities approaching 0 or 1 further indicates near perfect class separation, confirming the strength of the signal within the data.

6.2.2 Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis is used in this study as a parametric, distribution-based classification model to assess whether the price categories can be effectively separated under the assumption of multivariate normality with a shared covariance structure across classes. LDA provides a balance between interpretability and predictive power, making it suitable for evaluating linear separation between high-priced and non-high-priced diamonds.

LDA achieves a high test accuracy of 96.2%, indicating strong classification performance. This result suggests that the class means are well separated in the predictor space and that a linear decision boundary is sufficient to distinguish between the two price categories. The close performance to logistic regression further confirms that the underlying structure of the data is well captured by linear classification methods.

6.2.3 Quadratic Discriminant Analysis (QDA)

Quadratic Discriminant Analysis extends LDA by allowing each class to have its own covariance matrix, resulting in more flexible, non-linear decision boundaries. In this study, QDA is applied to evaluate whether increased model flexibility improves classification performance when class distributions differ in their variance structures.

QDA achieves a test accuracy of 93.0%, which is lower than both logistic regression and LDA. This reduction in performance suggests that the additional flexibility introduced by QDA leads to higher variance without providing substantial gains in predictive accuracy. The results indicate that the separation between high-priced and non-high-priced diamonds does not require complex non-linear boundaries and that simpler linear models are more appropriate for this dataset.

6.2.4 K-Nearest Neighbours (KNN)

The K-Nearest Neighbours algorithm is employed as a non-parametric, distance-based classification method that makes minimal assumptions about the underlying data distribution. KNN classifies observations based on the majority class among their nearest neighbours in the feature space and is included to capture potential local and non-linear patterns in the data.

The KNN model achieves a test accuracy of 90.0%, demonstrating reasonable classification performance. However, its accuracy is lower than that of the parametric models, indicating sensitivity to feature scaling, dimensionality, and local noise. While KNN is capable of modelling non-linear decision boundaries, the results suggest that global structure captured by parametric approaches is more effective for this classification task.

6.2.5 Neural Network

A neural network model is incorporated in this study to evaluate the suitability of a deep learning approach for classifying diamond prices using structured tabular data. Neural networks are designed to capture complex, non-linear relationships and higher-order interactions among predictors through multiple layers of weighted transformations. However, their performance is highly dependent on extensive hyperparameter tuning, appropriate feature representation, and large volumes of data, particularly when applied to datasets which contain both numerical and categorical variables.

In this study, the neural network achieves a test accuracy of 74.8%, which is the lowest among all classification models considered. This comparatively weaker performance can be attributed to several factors. First, the presence of high-dimensional categorical predictors, in case being cut, colour, clarity, could increase the model complexity which dilutes the learning efficiency when encoded for neural network training. Second, The underlying relationship between diamond characteristics and price classification is predominantly linear and strongly structured, limiting the advantage of non-linear function approximation offered by neural networks. As a result, simpler statistical models are able to exploit this structure more efficiently.

From a real-world perspective, this outcome reflects a common challenge in applying deep learning to structured pricing data, where domain-driven relationships are well understood and largely monotonic[3]. In such contexts, neural network may offer limited marginal benefit over interpretable parametric models, while introducing additional computational complexity and reduced transparency. These findings highlight the importance of aligning model choice with

[3] Shmuel, A., Glickman, O., & Lazebnik, T. (2025). *A comprehensive benchmark of machine and deep learning across diverse tabular datasets*. Neurocomputing, 587, 131337. https://doi.org/10.1016/j.neucom.2025.131337

data structure and practical application requirements rather than defaulting to more complex modelling approaches.

6.3 Conclusion

Overall, the results demonstrate that classical statistical classification models, particularly logistic regression and LDA, are highly effective for classifying diamond prices based on physical and quality-related attributes. More complex models, such as QDA, KNN, and Neural Networks do not provide additional predictive benefits in this context and may introduce unnecessary variance or instability. These findings emphasise the importance of aligning model complexity with data structure when developing classification solutions in real-world pricing and valuation settings.

7.0 Conclusion

This study set out to investigate whether diamond prices can be effectively classified into high and non-high categories using observable physical and quality-related characteristics. Through a comprehensive data preparation process, including data cleaning, appropriate variable encoding, and exploratory data analysis, the structure and key drivers of diamond pricing were clearly identified. The Explanatory Data Analysis revealed that diamond prices are highly right-skewed and strongly influenced by size-related attributes, particularly carat and physical dimensions, as well as quality measures such as cut, colour, and clarity. These findings justified the construction of a binary classification target and informed the selection of suitable classification models.

The application and comparison of multiple classification models demonstrated that diamond price classification is highly feasible when using these observable characteristics. Logistic regression and Linear Discriminant Analysis achieved the highest test accuracies, indicating that the relationship between predictors and price categories is strongly structured and largely linearly separable. More complex models, including Quadratic Discriminant Analysis, K-Nearest Neighbours, and a neural network, did not provide additional predictive benefits and, in some cases, underperformed due to increased variance or model complexity. These results highlight that simpler statistical models can outperform more flexible approaches when the underlying data structure is well defined.

From a real-world perspective, the findings align closely with practical pricing and valuation frameworks used in the diamond industry, where size and quality characteristics play a dominant role in determining value. The results demonstrate how statistical classification techniques can support pricing segmentation, valuation consistency, and decision-making for industry stakeholders such as retailers, valuers, and analysts. Overall, this study emphasises the importance of thorough data exploration, careful model selection, and interpretability when applying classification methods to structured pricing data.

8.0 Appendix

```
# Read Data
library(readxl)

## Warning: package 'readxl' was built under R version 4.3.3

diamonds_data <- read_excel("diamonds dataset 2.xlsx", skip = 1)
head(diamonds_data)

## # A tibble: 6 × 10
##   carat cut       color clarity depth table     x     y     z price
##   <dbl> <chr>     <chr> <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  0.23 Ideal     E     SI2      61.5    55  3.95  3.98  2.43   326
## 2  0.21 Premium   E     SI1      59.8    61  3.89  3.84  2.31   326
## 3  0.23 Good      E     VS1      56.9    65  4.05  4.07  2.31   327
## 4  0.29 Premium   I     VS2      62.4    58  4.2   4.23  2.63   334
## 5  0.31 Good      J     SI2      63.3    58  4.34  4.35  2.75   335
## 6  0.24 Very Good J     VVS2     62.8    57  3.94  3.96  2.48   336

summary(diamonds_data)

##      carat            cut               color              clarity
##  Min.   :0.2000   Length:53940       Length:53940       Length:53940
##  1st Qu.:0.4000   Class :character   Class :character   Class :characte
r
##  Median :0.7000   Mode  :character   Mode  :character   Mode  :characte
r
##  Mean   :0.7979
##  3rd Qu.:1.0400
##  Max.   :5.0100
##      depth           table             x                y
##  Min.   :43.00   Min.   :43.00   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:61.00   1st Qu.:56.00   1st Qu.: 4.710   1st Qu.: 4.720
##  Median :61.80   Median :57.00   Median : 5.700   Median : 5.710
##  Mean   :61.75   Mean   :57.46   Mean   : 5.731   Mean   : 5.735
##  3rd Qu.:62.50   3rd Qu.:59.00   3rd Qu.: 6.540   3rd Qu.: 6.540
##  Max.   :79.00   Max.   :95.00   Max.   :10.740   Max.   :58.900
##        z             price
##  Min.   : 0.000   Min.   :  326
##  1st Qu.: 2.910   1st Qu.:  950
##  Median : 3.530   Median : 2401
##  Mean   : 3.539   Mean   : 3933
##  3rd Qu.: 4.040   3rd Qu.: 5324
##  Max.   :31.800   Max.   :18823

# Call Library
library(MASS)
library(e1071)

library(class)

library(ggplot2)
library(dplyr)
```

```r
library(GGally)

library(corrplot)

# Check missing values per variable
colSums(is.na(diamonds_data))
```

```
##   carat     cut   color clarity   depth   table       x       y       z
price
##       0       0       0       0       0       0       0       0       0
0
```

```r
# Count duplicated rows
sum(duplicated(diamonds_data))
```

```
## [1] 146
```

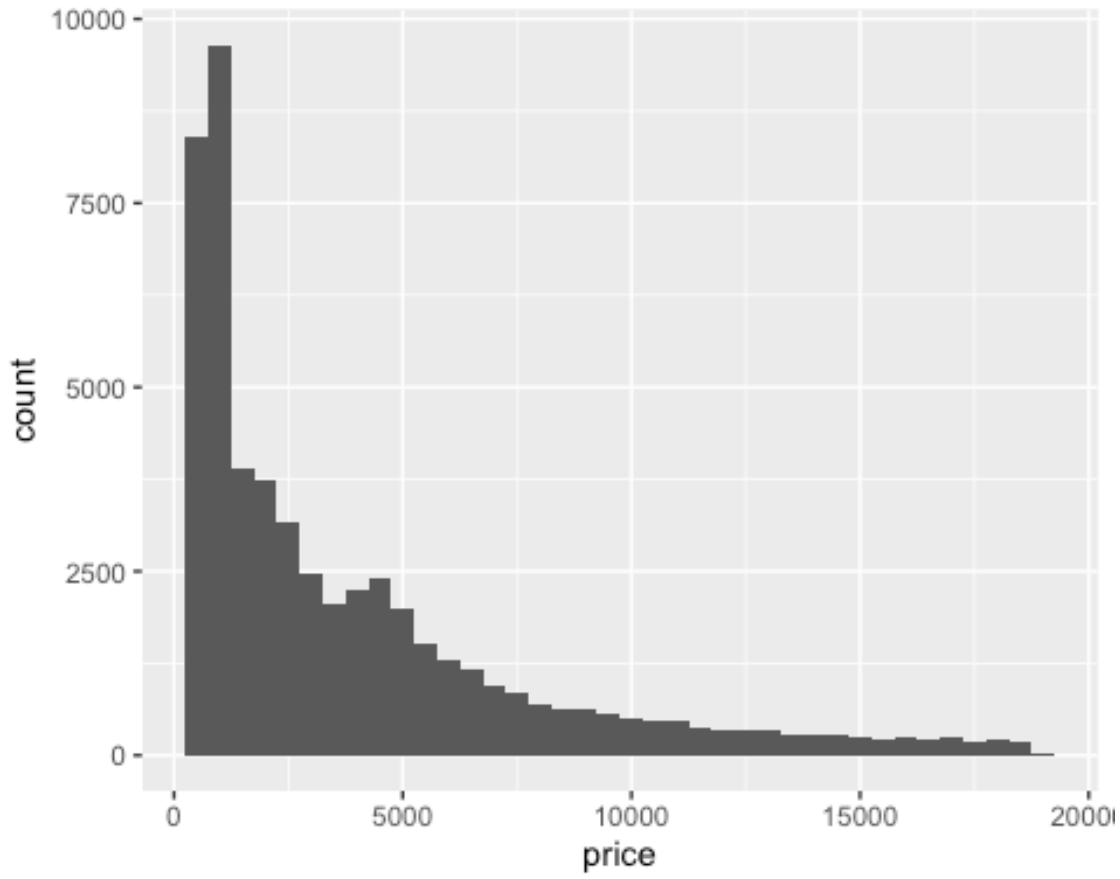
```r
# Remove duplicated rows to keep only unique athletes
diamonds_data <- diamonds_data[!duplicated(diamonds_data), ]

# Confirm duplicates are gone
sum(duplicated(diamonds_data))
```

```
## [1] 0
```

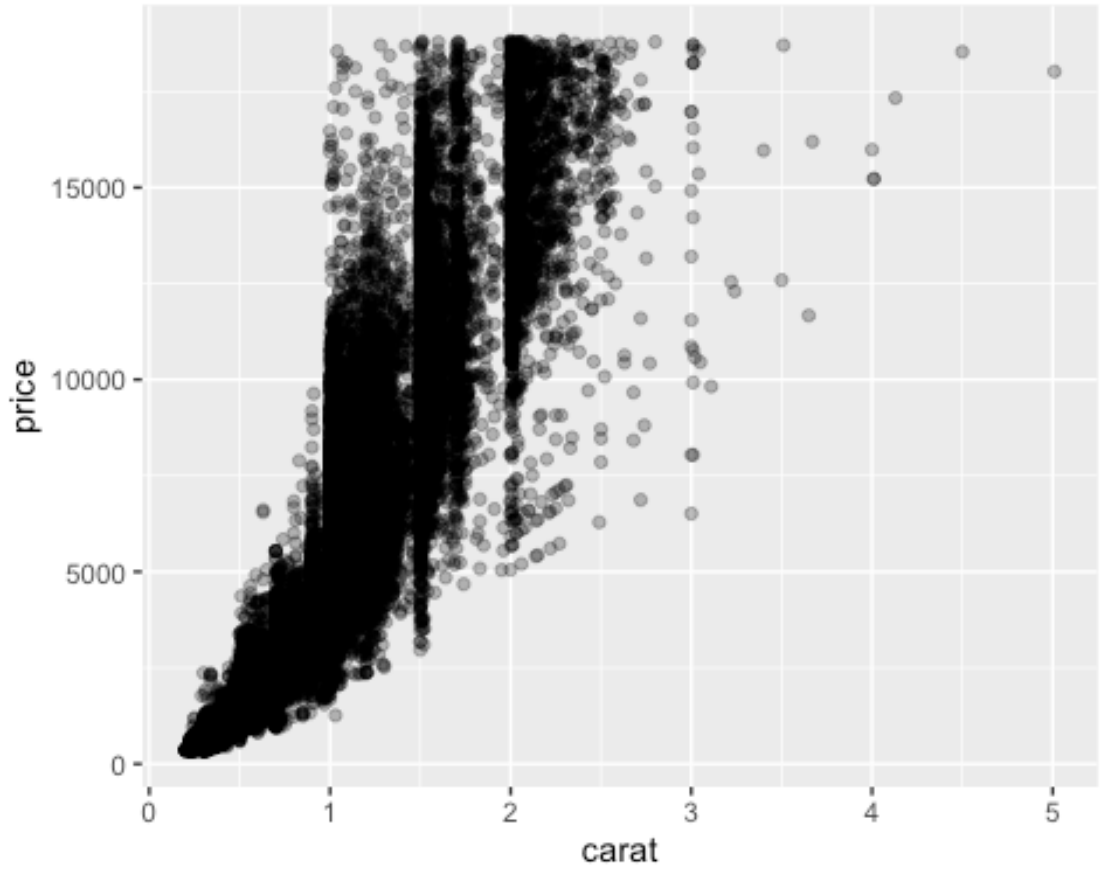
```r
# Convert categorical variables into factors
diamonds_data$cut    <- factor(diamonds_data$cut)
diamonds_data$color  <- factor(diamonds_data$color)
diamonds_data$clarity <- factor(diamonds_data$clarity)
str(diamonds_data)
```

```
## tibble [53,794 × 10] (S3: tbl_df/tbl/data.frame)
##  $ carat  : num [1:53794] 0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22
0.23 ...
##  $ cut    : Factor w/ 5 levels "Fair","Good",..: 3 4 2 4 2 5 5 5 1 5 ..
.
##  $ color  : Factor w/ 7 levels "D","E","F","G",..: 2 2 2 6 7 7 6 5 2 5
...
##  $ clarity: Factor w/ 8 levels "I1","IF","SI1",..: 4 3 5 6 4 8 7 3 6 5
...
##  $ depth  : num [1:53794] 61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1
59.4 ...
##  $ table  : num [1:53794] 55 61 65 58 58 57 57 55 61 61 ...
##  $ x      : num [1:53794] 3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4
...
##  $ y      : num [1:53794] 3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78
4.05 ...
##  $ z      : num [1:53794] 2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49
2.39 ...
##  $ price  : num [1:53794] 326 326 327 334 335 336 336 337 337 338 ...
```

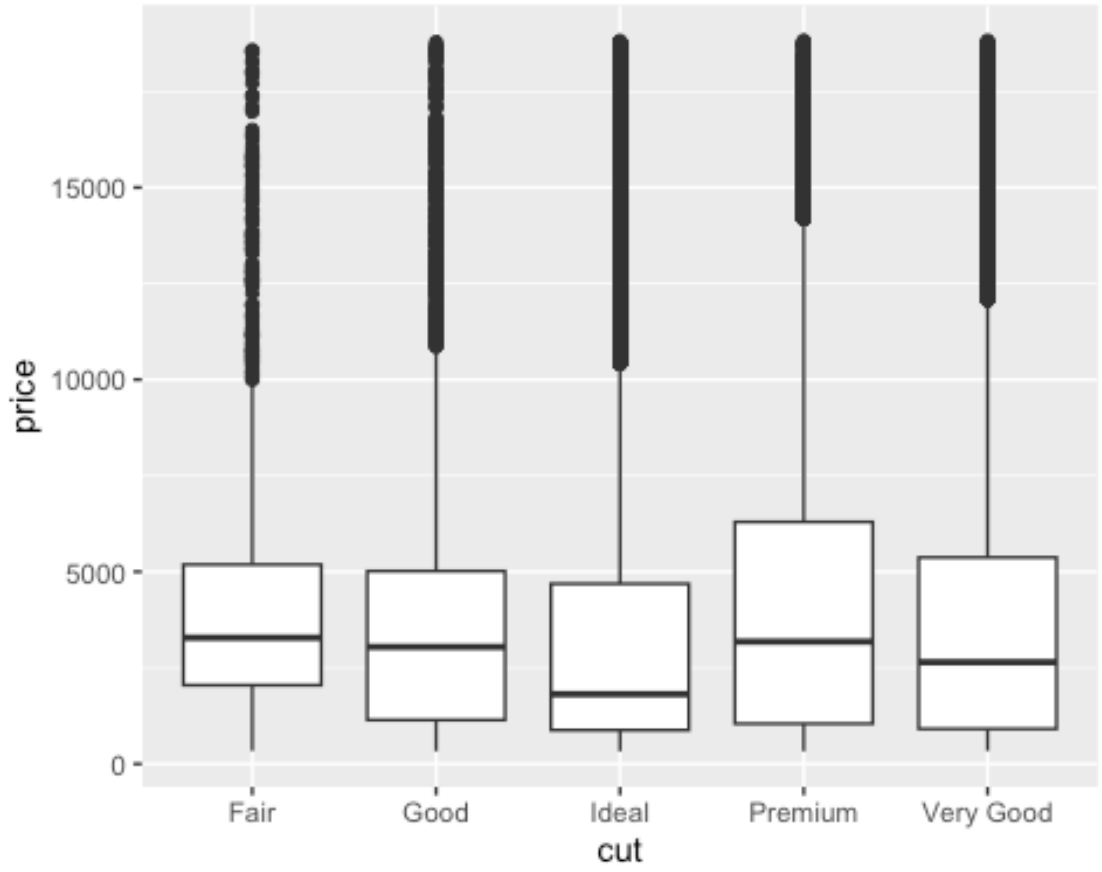
```r
# Exploratory Data Analysis

# Histogram
ggplot(diamonds_data, aes(x = price)) + geom_histogram(binwidth = 500)
```



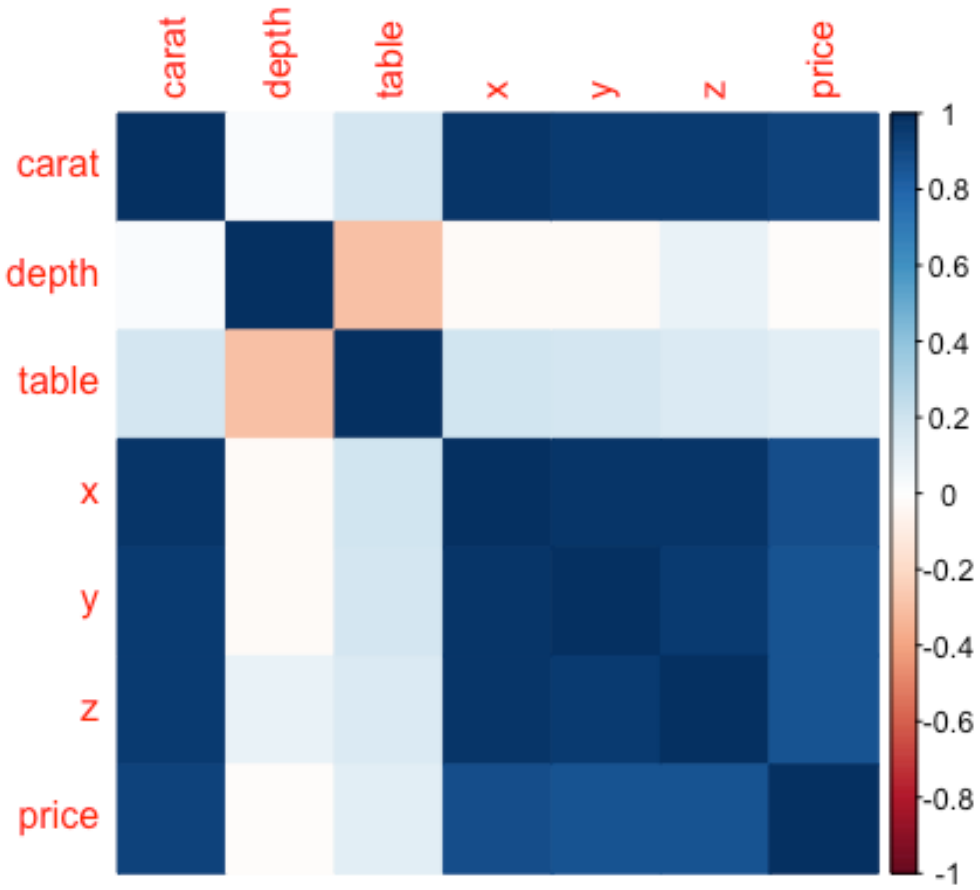

```
# Scatterplots
ggplot(diamonds_data, aes(x = carat, y = price)) + geom_point(alpha = 0.3)
```




```r
# Boxplots or Violin plots
ggplot(diamonds_data, aes(x = cut, y = price)) + geom_boxplot()
```




```
# Correlation Matrix (for numeric predictors)
corrplot(cor(diamonds_data[, c("carat","depth","table","x","y","z","price"
)]),
         method = "color")
```



Classification models

```
# CREATE A BINARY CLASSIFICATION TARGET
# Here we define "High" price as the top 25% most expensive diamonds
price_threshold <- quantile(diamonds_data$price, 0.75)

diamonds_data$price_class <- ifelse(diamonds_data$price >= price_threshold
,
                                    "High", "NotHigh")
diamonds_data$price_class <- factor(diamonds_data$price_class,
                                    levels = c("NotHigh", "High"))

table(diamonds_data$price_class)  # check class balance

##
## NotHigh    High
##   40345   13449

# TRAIN-TEST SPLIT (70% train, 30% test)
n <- nrow(diamonds_data)
set.seed(123)
train_index <- sample(seq_len(n), size = 0.7 * n)

train_data <- diamonds_data[train_index, ]
test_data  <- diamonds_data[-train_index, ]

# 3. CHOOSE PREDICTOR VARIABLES
```

```r
# We use all main predictors except the raw price (since that defines pric
e_class)
predictors <- c("carat", "cut", "color", "clarity",
                "depth", "table", "x", "y", "z")

# Formula for all models
form <- as.formula(
  paste("price_class ~", paste(predictors, collapse = " + "))
)

# 1. LOGISTIC REGRESSION (GLM, BINOMIAL)

logit_model <- glm(form,
                   data   = train_data,
                   family = binomial)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(logit_model)  # interpret coefficients, significance, etc.

##
## Call:
## glm(formula = form, family = binomial, data = train_data)
##
## Coefficients:
##                Estimate Std. Error    z value Pr(>|z|)
## (Intercept)   5.789e+15  3.277e+07  176647912   <2e-16 ***
## carat         6.379e+15  3.440e+06 1854174089   <2e-16 ***
## cutGood       9.674e+13  2.373e+06   40776212   <2e-16 ***
## cutIdeal      2.742e+14  2.366e+06  115920884   <2e-16 ***
## cutPremium    1.466e+14  2.278e+06   64381648   <2e-16 ***
## cutVery Good  1.555e+14  2.280e+06   68186335   <2e-16 ***
## colorE       -7.073e+13  1.279e+06  -55307841   <2e-16 ***
## colorF       -1.565e+14  1.287e+06 -121554739   <2e-16 ***
## colorG       -2.490e+14  1.263e+06 -197124355   <2e-16 ***
## colorH       -4.002e+14  1.348e+06 -296828817   <2e-16 ***
## colorI       -7.332e+14  1.501e+06 -488499716   <2e-16 ***
## colorJ       -1.208e+15  1.864e+06 -648136805   <2e-16 ***
## clarityIF     2.380e+15  3.615e+06  658584282   <2e-16 ***
## claritySI1    1.740e+15  3.080e+06  564915710   <2e-16 ***
## claritySI2    1.236e+15  3.091e+06  399800332   <2e-16 ***
## clarityVS1    2.123e+15  3.146e+06  674833830   <2e-16 ***
## clarityVS2    2.041e+15  3.096e+06  659212096   <2e-16 ***
## clarityVVS1   2.351e+15  3.330e+06  706035641   <2e-16 ***
## clarityVVS2   2.314e+15  3.240e+06  714200900   <2e-16 ***
## depth        -7.585e+13  4.038e+05 -187836887   <2e-16 ***
## table        -1.678e+13  2.073e+05  -80920183   <2e-16 ***
## x            -1.342e+15  3.091e+06 -434373250   <2e-16 ***
## y             1.477e+13  1.195e+06   12362358   <2e-16 ***
## z             2.265e+13  4.677e+06    4843598   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 42295  on 37654  degrees of freedom
## Residual deviance: 70934  on 37631  degrees of freedom
## AIC: 70982
##
## Number of Fisher Scoring iterations: 21
```

```
# Predicted probabilities for the test set
logit_prob <- predict(logit_model, newdata = test_data, type = "response")

# Convert probabilities to class labels using 0.5 cutoff
logit_pred <- ifelse(logit_prob >= 0.5, "High", "NotHigh")
logit_pred <- factor(logit_pred, levels = levels(train_data$price_class))

# Confusion matrix and accuracy
logit_cm <- table(Predicted = logit_pred, Actual = test_data$price_class)
logit_cm
```

```
##          Actual
## Predicted NotHigh  High
##   NotHigh   11849   205
##   High        230  3855
```

```
logit_accuracy <- sum(diag(logit_cm)) / sum(logit_cm)
logit_accuracy
```

```
## [1] 0.9730467
```

```
# 2. LINEAR DISCRIMINANT ANALYSIS (LDA)

lda_model <- lda(form, data = train_data)
lda_model
```

```
## Call:
## lda(form, data = train_data)
##
## Prior probabilities of groups:
##   NotHigh      High
## 0.7506573 0.2493427
##
## Group means:
##             carat    cutGood  cutIdeal cutPremium cutVery Good    color
E
## NotHigh 0.5890264 0.09353994 0.4173919  0.2360787    0.2225642 0.201726
5
## High    1.4303930 0.08478006 0.3362445  0.3216530    0.2275003 0.115454
3
##            colorF    colorG    colorH     colorI     colorJ  clarityIF
## NotHigh 0.1863016 0.2026109 0.1404160 0.08915305 0.04245383 0.03796080
## High    0.1593354 0.2349558 0.1843647 0.14176164 0.08062626 0.01789328
##         claritySI1 claritySI2 clarityVS1 clarityVS2 clarityVVS1 clarity
VVS2
## NotHigh  0.2453124  0.1647562  0.1473502  0.2139319  0.07977782  0.0970
4238
## High     0.2301630  0.1920332  0.1638087  0.2654170  0.03429545  0.0822
2388
```

```
##            depth    table        x        y        z
## NotHigh 61.75495 57.35227 5.257009 5.262538 3.247004
## High    61.72661 57.80552 7.166141 7.165018 4.419258
##
## Coefficients of linear discriminants:
##                      LD1
## carat         4.81140214
## cutGood       0.34842382
## cutIdeal      0.46660978
## cutPremium    0.50728039
## cutVery Good  0.46399472
## colorE       -0.12521029
## colorF       -0.16108938
## colorG       -0.09219472
## colorH       -0.36829379
## colorI       -0.67244885
## colorJ       -1.05128120
## clarityIF     2.38870420
## claritySI1    1.58906880
## claritySI2    0.86457936
## clarityVS1    2.23584749
## clarityVS2    2.14903121
## clarityVVS1   2.30939991
## clarityVVS2   2.38760195
## depth        -0.02900510
## table        -0.02940190
## x            -0.36781887
## y            -0.02139772
## z             0.08807120
```

```r
lda_pred <- predict(lda_model, newdata = test_data)$class

lda_cm <- table(Predicted = lda_pred, Actual = test_data$price_class)
lda_cm
```

```
##          Actual
## Predicted NotHigh  High
##   NotHigh   12003   530
##   High         76  3530
```

```r
lda_accuracy <- sum(diag(lda_cm)) / sum(lda_cm)
lda_accuracy
```

```
## [1] 0.9624512
```

```r
# 3. QUADRATIC DISCRIMINANT ANALYSIS (QDA)

qda_model <- qda(form, data = train_data)
qda_model
```

```
## Call:
## qda(form, data = train_data)
##
## Prior probabilities of groups:
##   NotHigh      High
```

```
## 0.7506573 0.2493427
##
## Group means:
##             carat    cutGood  cutIdeal cutPremium cutVery Good    color
E
## NotHigh 0.5890264 0.09353994 0.4173919  0.2360787    0.2225642 0.201726
5
## High    1.4303930 0.08478006 0.3362445  0.3216530    0.2275003 0.115454
3
##            colorF    colorG    colorH     colorI     colorJ  clarityIF
## NotHigh 0.1863016 0.2026109 0.1404160 0.08915305 0.04245383 0.03796080
## High    0.1593354 0.2349558 0.1843647 0.14176164 0.08062626 0.01789328
##         claritySI1 claritySI2 clarityVS1 clarityVS2 clarityVVS1 clarity
VVS2
## NotHigh  0.2453124  0.1647562  0.1473502  0.2139319  0.07977782  0.0970
4238
## High     0.2301630  0.1920332  0.1638087  0.2654170  0.03429545  0.0822
2388
##            depth    table        x        y        z
## NotHigh 61.75495 57.35227 5.257009 5.262538 3.247004
## High    61.72661 57.80552 7.166141 7.165018 4.419258
```

```
qda_pred <- predict(qda_model, newdata = test_data)$class

qda_cm <- table(Predicted = qda_pred, Actual = test_data$price_class)
qda_cm
```

```
##          Actual
## Predicted NotHigh  High
##   NotHigh   11734   779
##   High        345  3281
```

```
qda_accuracy <- sum(diag(qda_cm)) / sum(qda_cm)
qda_accuracy
```

```
## [1] 0.930355
```

```
# 4. K-NEAREST NEIGHBOURS (KNN, k = 5)

# For KNN we use only numeric predictors and scale them.
numeric_vars <- c("carat", "depth", "table", "x", "y", "z")

# Create numeric matrices
train_x <- as.matrix(train_data[, numeric_vars])
test_x  <- as.matrix(test_data[, numeric_vars])

# Scale using training set parameters
train_x_scaled <- scale(train_x)
test_x_scaled  <- scale(test_x,
                        center = attr(train_x_scaled, "scaled:center"),
                        scale  = attr(train_x_scaled, "scaled:scale"))

train_y <- train_data$price_class
test_y  <- test_data$price_class
```

```r
set.seed(123)
knn_pred <- knn(train = train_x_scaled,
                test  = test_x_scaled,
                cl    = train_y,
                k     = 5)

knn_cm <- table(Predicted = knn_pred, Actual = test_y)
knn_cm
```

```
##          Actual
## Predicted NotHigh  High
##   NotHigh   11335   718
##   High        744  3342
```

```r
knn_accuracy <- sum(diag(knn_cm)) / sum(knn_cm)
knn_accuracy
```

```
## [1] 0.909412
```

```r
# 5. NEURAL NETWORK
# Fit a neural network model on training data
library(nnet)
set.seed(123)  # for reproducibility
nn_model <- nnet(price_class ~ ., data = train_data, size = 5, maxit = 500
)
```

```
## # weights:  131
## initial  value 22691.076173
## final  value 21147.495810
## converged
```

```r
# Predict on the test set (type="class" yields factor class predictions)
nn_pred <- predict(nn_model, newdata = test_data, type = "class")

# Confusion matrix and accuracy
conf_mat_nn <- table(Predicted = nn_pred, Actual = test_data$price_class)
conf_mat_nn
```

```
##          Actual
## Predicted NotHigh  High
##   NotHigh   12079  4060
```

```r
accuracy_nn <- mean(nn_pred == test_data$price_class)
accuracy_nn
```

```
## [1] 0.7484355
```

```r
# COMPARE MODEL PERFORMANCES

model_performance <- data.frame(
  Model    = c("Logistic Regression", "LDA", "QDA", "KNN (k = 5)", "Neural
Network"),
  Accuracy = c(logit_accuracy, lda_accuracy, qda_accuracy, knn_accuracy, a
ccuracy_nn)
)
```

```
model_performance
```

```
##                 Model  Accuracy
## 1 Logistic Regression 0.9730467
## 2                 LDA 0.9624512
## 3                 QDA 0.9303550
## 4         KNN (k = 5) 0.9094120
## 5      Neural Network 0.7484355
```