TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

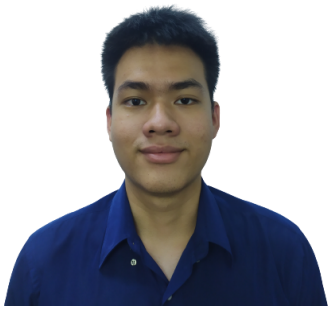
FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY
ACADEMIC YEAR 2023/2024 Session 202305

BACS3013 DATA SCIENCE: Assignment Documentation

Title: Diamond Price Prediction

Programme and Tutorial Class: RSWG3/G6
Tutor Name: Noor Aida binti Husaini
Submission Date: 17/09/2023

Team Member:



Student Name: Gabriel Teoh Kang Wei
Student ID:22WMR05647
Module-in-Charge: KNN Algorithm

Student Name: Nee Mei Yi
Student ID: 22WMR09856
Module-in-Charge: Bayesian Ridge Algorithm

Student Name: Lee Wee Harn
Student ID: 22WMR05673
Module-in-Charge: Linear Regression Algorithm

Table of Content

1.0 Business Understanding

1.1 Company Background

Figure 1.1 Head Quarter of SK Jewellery Group

SK Jewellery was founded in 1991 by Ms. Christina Lee and Mr. Daniel Lim in Singapore. The company started as a small jewellery store under the brand name Soo Kee Jewellery in Bedok Central and gradually expanded its operations, gaining a reputation for its elegant designs and exquisite pieces. Over the years, SK Jewellery has grown to become one of Singapore's leading jewellery brands. The company is recognized for its craftsmanship, innovative designs, and commitment to quality. SK Jewellery has positioned itself as a retailer that caters to a diverse customer base with various preferences and budgets. SK Jewellery offers a wide variety of jewellery, including diamond engagement rings, wedding bands, necklaces, bracelets, earrings, and other accessories. The company is known for its focus on quality diamonds and its ability to incorporate them into unique and fashionable designs.

The headquarters of SK Jewellery has a central location in Singapore, a city where tradition and modernity coexist in harmony. The environment is ideal for the brand's principles, drawing inspiration from the city's global energy and rich cultural past. In this dynamic environment, SK Jewellery is able to infuse its designs with a unique blend of inspirations, resulting in jewellery that appeals to customers from all walks of life. They are a well-known and well-established jeweller based in Singapore. In addition, they have one of the most extensive retail networks in Singapore and Malaysia. Their retail outlets in Singapore are strategically located in major downtown shopping malls like ION Orchard and VivoCity, as well as popular heartland malls like Tampines Mall and Jurong Point. They have a presence in important Malaysian states and cities such as Kuala Lumpur, Selangor, Penang, Ipoh, and Johor Bahru. Their significant retail presence allows them to provide fine jewellery and mementoes to better serve their clients (*Anon, 2015*).

Guided by its vision, SK Jewellery aims to be the preferred choice for individuals seeking jewellery that perfectly blends elegance and modernity. The company defines itself as setting industry standards by merging historical designs with modern trends, making jewellery that stands the test of time. At the heart of its mission is the desire to create items that capture life's most treasured events, becoming a part of the stories that customers appreciate. SK Jewellery is dedicated to creating long-term connections by providing a personalised shopping experience and delivering outstanding quality. Based on its legacy, the brand keeps shining on lives across generations, enlightening them with the warmth of wonderfully created jewels (*Find Out More About SK Jewellery Singapore, 2020*).

1.2 Problem Background

- **Problem 1: Market Trends and Fluctuations**

  Diamond prices are tightly linked to a web of market trends, economic trends, and geopolitical events, making it difficult to predict their impact on diamond prices. The diamond market is subject to consumer attitude, which can be influenced by changing fashion tastes and cultural influences. Consumer purchasing power and willingness to invest in luxury items such as diamonds are directly affected by economic conditions such as global recessions or periods of economic expansion. Furthermore, geopolitical variables such as trade agreements, political instability, and movements in diamond-producing regions can disrupt supply chains and affect market dynamics. Given their frequently rapid and unpredictable character, predicting how these multiple external variables will interact and impact diamond prices is difficult *(Diamondregistry, 2021)*.

- **Problem 2: Emergence of Lab-Grown Diamonds**

  The introduction of lab-grown diamonds has caused a major change in the diamond industry by providing an innovative alternative to traditionally harvested natural diamonds. Lab-grown diamonds have the same physical and chemical features as natural diamonds, but they are created in a controlled laboratory setting. This regulated environment enables efficient and predictable diamond production, resulting in diamonds that are frequently less expensive than naturally created counterparts. The introduction of lab-grown diamonds has brought a new level of complication to the pricing dynamics of the diamond market. The advent of lab-grown diamonds at lower price points has given consumers more options, causing the usual supply-demand balance to be disrupted. As lab-grown diamonds gain popularity, their market share and acceptance impact natural diamonds' perceived worth, potentially impacting pricing patterns in both categories. Consumer preferences for ethically produced and environmentally sustainable solutions, as well as the industry's response to this disruptive technology, all contribute to the changing landscape of diamond pricing and market behaviour *(Labrilliante, 2022)*.

1.3 Aims

1. **Optimised Pricing Strategy**

   A diamond price prediction model-based optimised pricing strategy can transform the company's approach to selling diamonds. The model provides a comprehensive perspective of pricing dynamics by carefully evaluating a wide range of elements such as diamond characteristics, market trends, historical sales data, and economic indicators. With this level of detail, our organisation can effectively calculate the most accurate and competitive prices for each diamond. Customers like the accuracy, which increases their trust and purchases. Furthermore, the enhanced pricing strategy increases revenue streams by collecting the full value of each diamond while simultaneously protecting profit margins through intelligent price adjustments. Overall, implementing such a methodology enables our organisation to negotiate the complex diamond market landscape with ease, resulting in better sales, healthier profitability, and a stronger market presence.

2. **Inventory Management**

   Diamond pricing trends can be predicted using an advanced price prediction model, allowing for highly effective inventory management strategies. The model predicts whether diamond prices will rise, fall, or remain stable in the near future by evaluating historical pricing data, market trends, and different influencing factors. With this knowledge, our organisation may change the inventory levels proactively, ensuring that the right types of diamonds are supplied in acceptable numbers at the right times. This avoids the costly issue of overstocking, which ties up cash and storage space, while simultaneously reducing the risk of understocking, which may result in lost sales opportunities. Therefore, the company can reduce carrying costs associated with excess inventory and maximise working capital efficiency. As a result, inventory is accurately customised to market demand and price expectations, enhancing financial performance and operational agility.

3. Supplier Negotiations

Using an accurate diamond price prediction model can improve supplier negotiations significantly by giving our organisation useful insights and leverage. Negotiations with diamond suppliers become more informed and strategic when we have historical pricing trends, current market situations, and predictive data. The ability to predict expected price movements allows our team to begin discussions with confidence, allowing us to gain better pricing arrangements. We establish ourselves as a smart and discerning buyer by showing a thorough understanding of market dynamics and trends. As a result,our organisation is better positioned to increase profit margins through cost-effective procurement. Finally, the combination of data-driven insights and negotiation expertise enables our company to strengthen supplier partnerships and enhance diamond sourcing methods.

1.4 Objectives

The objective of this project is to create and put into use a sophisticated diamond cut prediction technology that will revolutionise jewellery design accuracy. This involves using data analysis and machine learning techniques to build a reliable prediction system that will be seamlessly incorporated into the company's conceptual infrastructure. To develop prediction models that precisely prescribe the best diamond cuts, thus boosting their brilliance and aesthetic appeal, a broad collection of diamonds will be gathered and examined. These suggestions can help jewellery designers create items that are aesthetically pleasing and adhere to consumer expectations for magnificent and dazzling designs. The system's correctness will be guaranteed by rigorous validation and ongoing improvement, and designer training will make it easier to use its insights effectively. By achieving this goal, the company intends to improve the accuracy of jewellery design, leading to productions that routinely exceed expectations and raise customer happiness.

The secondary objective of this project is to realise considerable time and cost savings into our jewellery design and production processes. We intend to accelerate and streamline our design and manufacturing workflows by putting this cutting-edge solution in place. By using automated decision assistance, designers will be given the ability to quickly and wisely choose diamond cuts that are in line with their artistic visions, doing away with the necessity for laborious manual evaluations. We will be able to introduce items to the market more rapidly and in a responsive manner because of the efficiency that results in shorter design cycles and quicker manufacturing schedules. Additionally, by optimising resource allocation as a result of these streamlined processes, we will be able to concentrate more on perfecting intricate design elements and creating one-of-a-kind settings, which will ultimately produce higher-quality finished goods that uphold our brand's commitment to excellence.

The third objective of this project is to establish our company as a trendsetter and market leader in the competitive jewellery sector. We seek to differentiate ourselves from the competition and create a distinctive brand identity by seamlessly incorporating an innovative diamond price prediction technology. This tactical choice exemplifies our unwavering dedication to embracing cutting-edge technologies and promoting an innovative culture. In addition to showcasing our

skill in jewellery creation, the combination of aesthetic workmanship and data-driven tools strengthens our ability to use cutting-edge solutions. This initiative aspires to develop a distinctive brand that appeals to customers looking for contemporary and forward-thinking jewellery options. We anticipate greater market acceptance, a larger market share, and fresh business opportunities as we develop this avant-garde identity, which will advance our brand to the forefront of the sector's development.

1.5 Motivation

For our organisation, the adoption of a diamond price prediction system is of utmost importance as it promises to have a favourable impact on societal progress as well as economic growth. From an economic standpoint, this innovation gives us a definite competitive edge and strategically positions us as industry leaders in the jewellery industry. Utilising this technology's predictive capabilities allows us to improve our pricing strategies and increase our agility in order to more effectively respond to market needs. This leads to speedier response times, cost savings, and a greater capacity to acquire market share.

The societal effects of our acceptance of technology are also encouraging. Automated pricing forecasts' effectiveness results in resource conservation and a smaller environmental impact, which is perfectly in line with consumer demands for environmentally friendly products. Additionally, we set a standard for the jewellery business with our dedication to fusing creativity with data-driven insights. This dedication fosters a more ecologically conscious and forward-thinking industry while also raising standards within the industry. In conclusion, implementing diamond price prediction technology has positive effects on our company, customers, and society at large. This effort illustrates our goal to influence a more sustainable and forward-thinking industrial landscape by enhancing our ability to achieve economic success and reaffirming our commitment to responsible innovation.

1.6 Milestone

Description	Assign to	Start date and end date
Identify the dataset	Gabriel Teoh Kang Wei , Nee Mei Yi , Lee Wee Harn	12/07/2023 - 15/07/2023
Brief description about the report	Gabriel Teoh Kang Wei , Nee Mei Yi , Lee Wee Harn	16/07/2023 - 19/07/2023
Data understanding	Gabriel Teoh Kang Wei , Nee Mei Yi , Lee Wee Harn	20/07/2023 - 24/07/2023
Data preparation	Gabriel Teoh Kang Wei , Nee Mei Yi , Lee Wee Harn	25/07/2023 - 28/07/2023
Data cleaning	Gabriel Teoh Kang Wei , Nee Mei Yi , Lee Wee Harn	29/07/2023 - 31/07/2023
Normalisation	Gabriel Teoh Kang Wei , Nee Mei Yi , Lee Wee Harn	01/08/2023 - 03/08/2023
Data transformation	Gabriel Teoh Kang Wei , Nee Mei Yi , Lee Wee Harn	04/08/2023 - 10/08/2023
Modelling	Gabriel Teoh Kang Wei , Nee Mei Yi , Lee Wee Harn	11/08/2023 - 20/08/2023
Evaluation	Gabriel Teoh Kang Wei , Nee Mei Yi , Lee Wee Harn	21/08/2023 - 25/08/2023
Deployment	Gabriel Teoh Kang Wei , Nee Mei Yi , Lee Wee Harn	26/08/2023 - 02/09/2023
Complete Final Report	Gabriel Teoh Kang Wei , Nee Mei Yi , Lee Wee Harn	03/09/2023 - 14/09/2023
Check Plagiarism for final report	Gabriel Teoh Kang Wei , Nee Mei Yi , Lee Wee Harn	15/09/2023 - 16/09/2023
Revise Final Report	Gabriel Teoh Kang Wei , Nee Mei Yi , Lee Wee Harn	17/09/2023 - 17/09/2023
Submission of report	Gabriel Teoh Kang Wei , Nee Mei Yi , Lee Wee Harn	17/09/2023 - 17/09/2023

2 .0 Data Understanding

Name	Type	Dependent/ Independent Variable	Description
Carat	Continuous	Independent Variable	**The weight of the diamond in carat**
Cut	Categorical	Independent Variable	**Cut of a diamond refers to how well the diamond's facets interact with light.** 1= b'Ideal' 2 = b'Premium' 3 = b'Very Good' 4 = b'Good' 5 = b'Fair'
Color	Categorical	Independent Variable	**The colour grade of a diamond indicates the presence of colour .** 1 = b'G' 2 = b'E' 3 = b'F' 4 = b'H' 5 = b'D' 6 = b'I' 7 = b'J'
Clarity	Categorical	Independent Variable	**Clarity refers to the presence of internal or external flaws in a diamond** 1 = b'SI1' 2 = b'VS2' 3 = b'SI2' 4 = b'VS1' 5 = b'VVS2' 6 = b'VVS1' 7 = b'IF' 8 = b'I1'
Depth	Continuous	Independent Variable	**Depth refers to the height of the gemstone measured from the culet(bottom) to the table(top) as a percentage of its average girdle diameter.**

Table	Continuous	Independent Variable	**The table percentage is the width of the top facet of the diamond divided by the average girdle diameter .**
Price	Continuous	Dependent Variable	**The selling price of the diamond**
X	Continuous	Independent Variable	**X indicates the diamond length in millimetres**
Y	Continuous	Independent Variable	**Y indicates the diamond width in millimetres**
Z	Continuous	Independent Variable	**Z indicates the diamond depth in millimetres**

Table 1 List of Variables and Specifications

In this project, we have chosen diamonds as our datasets. Based on ***Table 1***, the dataset consists of 10 columns and 53940 rows of data. The 10 columns include 9 independent variables and 1 dependent variable.

	carat	cut	color	clarity	depth	table	price	'x'	'y'	'z'
0	0.23	b'Ideal'	b'E'	b'SI2'	61.5	55.0	326.0	3.95	3.98	2.43
1	0.21	b'Premium'	b'E'	b'SI1'	59.8	61.0	326.0	3.89	3.84	2.31
2	0.23	b'Good'	b'E'	b'VS1'	56.9	65.0	327.0	4.05	4.07	2.31
3	0.29	b'Premium'	b'I'	b'VS2'	62.4	58.0	334.0	4.20	4.23	2.63
4	0.31	b'Good'	b'J'	b'SI2'	63.3	58.0	335.0	4.34	4.35	2.75
5	0.24	b'Very Good'	b'J'	b'VVS2'	62.8	57.0	336.0	3.94	3.96	2.48
6	0.24	b'Very Good'	b'I'	b'VVS1'	62.3	57.0	336.0	3.95	3.98	2.47
7	0.26	b'Very Good'	b'H'	b'SI1'	61.9	55.0	337.0	4.07	4.11	2.53
8	0.22	b'Fair'	b'E'	b'VS2'	65.1	61.0	337.0	3.87	3.78	2.49
9	0.23	b'Very Good'	b'H'	b'VS1'	59.4	61.0	338.0	4.00	4.05	2.39

Figure 2.0.1 Overview of Dataset

	carat	depth	table	price	'x'	'y'	'z'
count	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000
mean	0.797940	61.749405	57.457184	3932.799722	5.731157	5.734526	3.538734
std	0.474011	1.432621	2.234491	3989.439738	1.121761	1.142135	0.705699
min	0.200000	43.000000	43.000000	326.000000	0.000000	0.000000	0.000000
25%	0.400000	61.000000	56.000000	950.000000	4.710000	4.720000	2.910000
50%	0.700000	61.800000	57.000000	2401.000000	5.700000	5.710000	3.530000
75%	1.040000	62.500000	59.000000	5324.250000	6.540000	6.540000	4.040000
max	5.010000	79.000000	95.000000	18823.000000	10.740000	58.900000	31.800000

Figure 2.0.2 Overview of Dataset

2.1 Dependent variable

2.1.1 Price

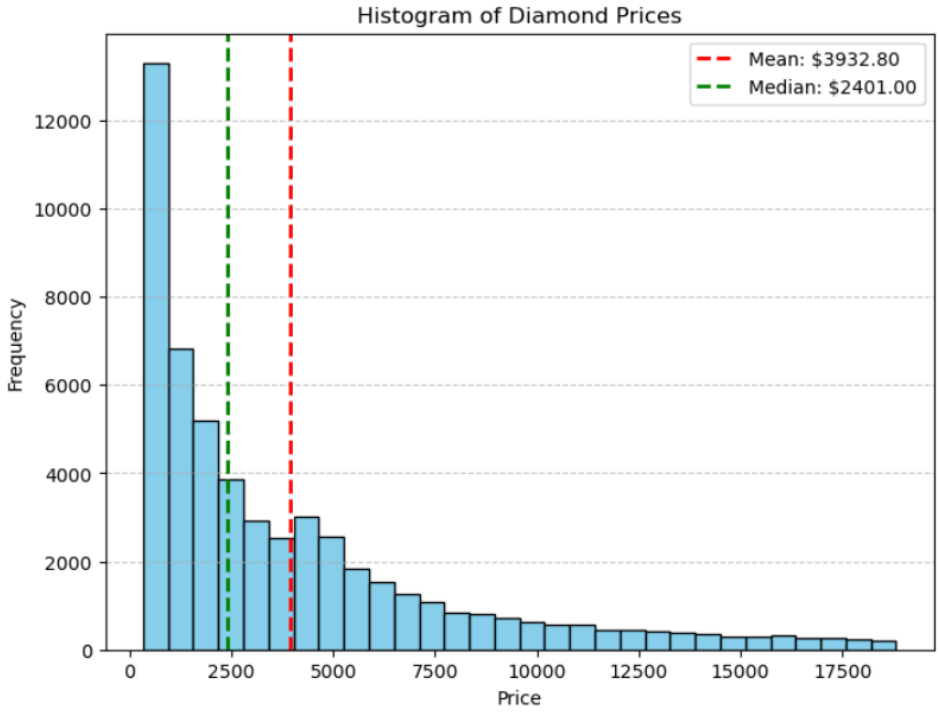


Figure 2.1.1.1: Diamond prices results in histogram

Based on ***Figure 2.1.1.1***, several important statistical measurements were computed in the study of a dataset involving 53,940 diamond price records. The mean price, $3,932.80, is the average price of the dataset, offering an overall impression of central tendency. Meanwhile, the median price, at $2,401.00, represents the centre value when sorted, acting as a reliable fundamental measurement that is less influenced by outliers. The information displayed a wide range of prices, from $326.00 to $18,823.00, illustrating the variance in diamond pricing. Furthermore, the standard deviation of $3,989.44 indicated a significant price spread, showing variability within the dataset. Particularly, the fact that the median is lower than the mean shows a potential right-skewness, which could be influenced by higher-priced outliers.

2.2 Independent variable

2.2.1 Carat

```
Summary of Diamond Carat
----------------------------
count    53940.000000
mean         0.797940
std          0.474011
min          0.200000
25%          0.400000
50%          0.700000
75%          1.040000
max          5.010000
Name: carat, dtype: float64
```

Figure 2.2.1.1: Description of diamond carat

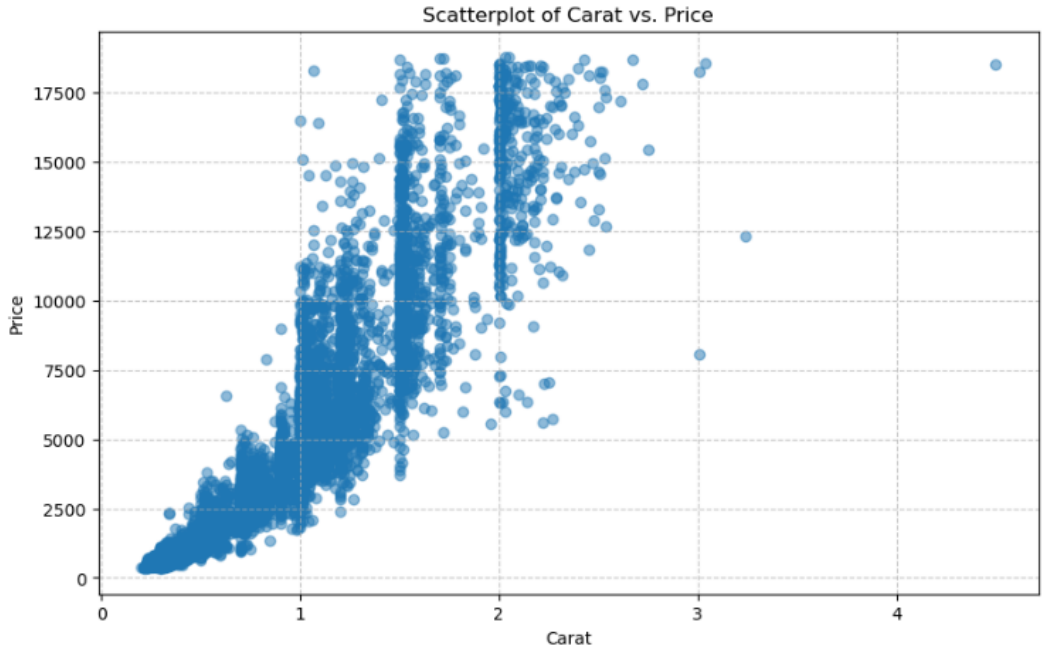


Figure 2.2.1.2: Diamond carat with price in scatterplot

Based on ***Figure 2.2.1.1***, and ***Figure 2.2.1.1***, the statistics for the "carat" variable in the diamond dataset provide useful information on the distribution among the 53,940 diamonds. The mean carat weight of approximately 0.798 carats shows that the average diamond in the dataset is close to 0.8 carats. Carat weights have moderate variation, with a standard deviation of roughly 0.474

carats, indicating that while many diamonds cluster around the mean, there are also important variances. According to the quartiles, the majority of diamonds have carat weights ranging from 0.4 to 1.04 carats, with a median carat weight of roughly 0.7 carats, implying that half of the diamonds fall below this figure. The following summary provides useful background for understanding the carat vs. price scatterplot. The scatterplot illustrates the visual link between carat and price for the sampled diamonds. Typically, a positive connection would be expected, implying that as carat increases, so does the price of the diamond.

2.2.2 Cut

```
b'Ideal'        21551
b'Premium'      13791
b'Very Good'    12082
b'Good'          4906
b'Fair'          1610
Name: cut, dtype: int64
```

Figure 2.2.2.1: Total number of diamond cuts

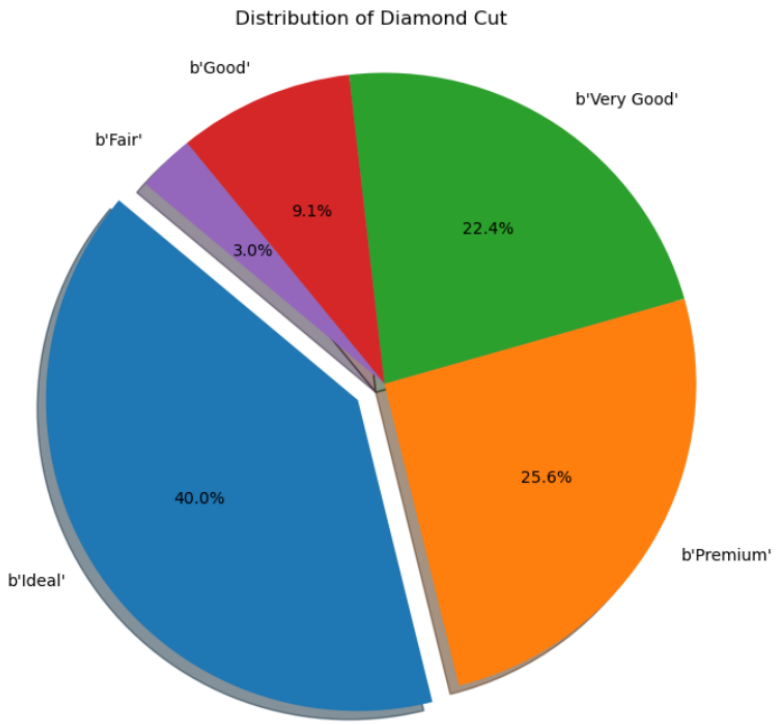


Figure 2.2.2.2: Distribution of diamond cut in pie chart

Based on ***Figure 2.2.2.1***, we can see that out of 53940 of records 21551 of them are graded as 'Ideal', 13791 of them are graded as 'Premium', 12082 of the are graded as 'Very Good', 4906 of them are graded as 'Good', and 1610 of them are graded as 'Fair'. Therefore, for better visualisation, we distribute the diamond cut in a pie chart. Based on ***Figure 2.2.2.2***, we can see

that 40% of the diamonds are graded as ‘Ideal’, 25.6% of the diamonds are graded as ‘Premium’, 22.4% of the diamonds are graded as ‘Very Good’, 9.1% of the diamonds are graded as ‘Good’ and 3% of them are graded as ‘Fair’.

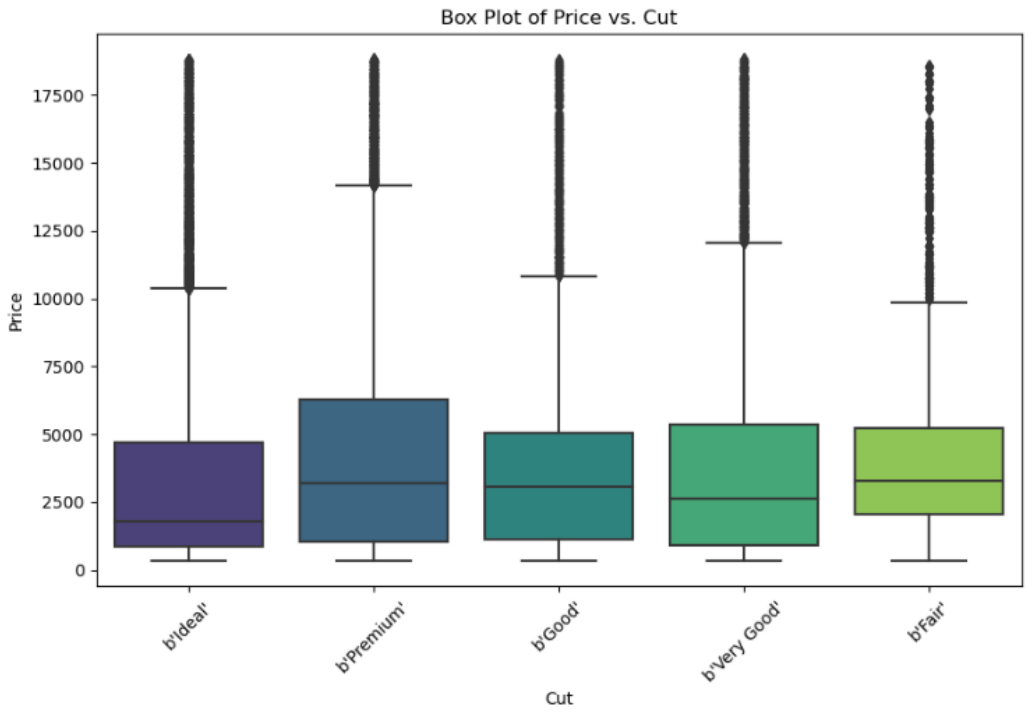


Figure 2.2.2.3: Distribution of diamond cut with price in box plot

Based on ***Figure 2.2.2.3***, the 'Premium' cut grade having the highest median price suggests that, on average, diamonds with a 'Premium' cut tend to command higher prices compared to other cut grades. On the other hand, the ‘Ideal’ cut grade has the lowest median price which shows that, on average, diamonds with a ‘Ideal’ cut tend to command lower prices compared to other cut grades.

2.2.3 Color

```
b'G'    11292
b'E'     9797
b'F'     9542
b'H'     8304
b'D'     6775
b'I'     5422
b'J'     2808
Name: color, dtype: int64
```

Figure 2.2.3.1: Total number of diamonds in different colors

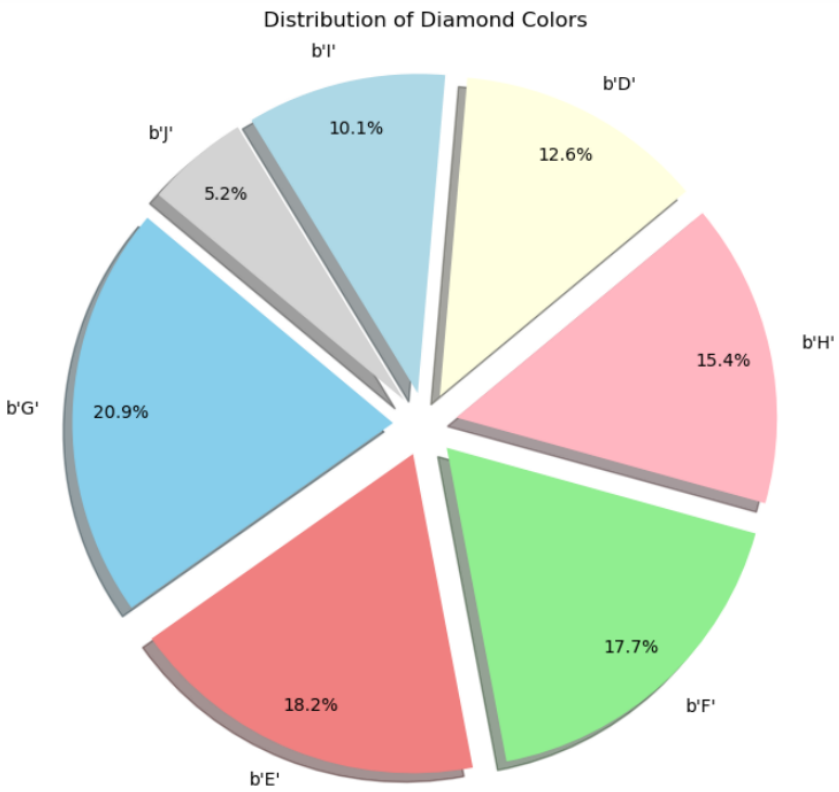


Figure 2.2.3.2: Distribution of diamonds colour in pie chart

Based on ***Figure 2.2.3.1***, the color grade 'D' is the highest and rarest grade in the GIA color scale, color grade 'E' and 'F' are considered "colorless" and represent diamonds with very minute traces of color while color grade 'G' and 'H' are are in the "near-colorless" range.

Finally, the color grade ‘I’ and ‘J’ represent diamonds in the "near-colorless" to "faint" color range. Therefore, 11292 diamonds are graded as color ‘G’, 9797 diamonds are graded as color ‘E’, 9542 diamonds are graded as color ‘F’, 8304 diamonds are graded as color ‘H’, 6775 diamonds are graded as color ‘D’, 5422 diamonds are graded as color ‘I’ and 2802 of them are graded as colour ‘J’.Therefore, for better visualisation we distribute the diamonds color in a pie chart. Based on ***Figure 2.2.3.2***, 20.9% of the diamonds are graded as color ‘G’, 18.2% of the diamonds are graded as color ‘E’, 17.7% of the diamonds are graded as color ‘F’, 15.4% of the diamonds are graded as color ‘H’, 12.6% of the diamonds are graded as color ‘D’, 10.1% of the diamonds are graded as color ‘I’ and 5.2% of the diamonds are graded as color ‘J’.

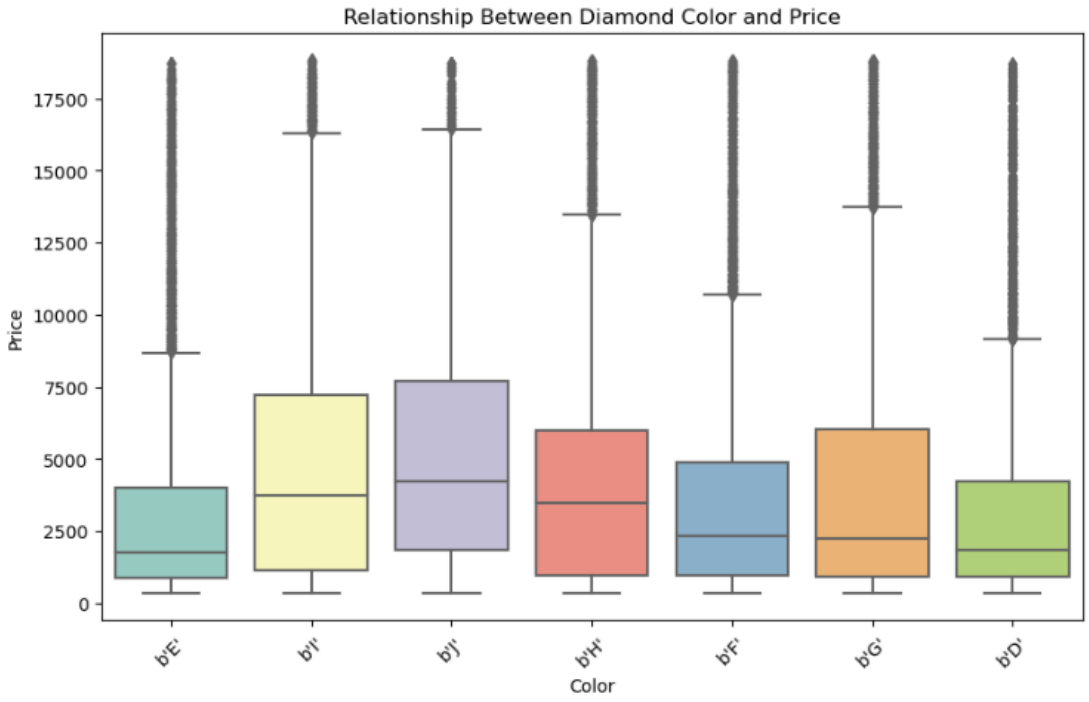


Figure 2.2.3.3: Relationship between diamond colour and price in box plot

Based on ***Figure 2.2.3.3***, the color grade ‘J’ has the highest median value in the box plot which suggests that on average, diamonds with a color grade ‘J’ tend to command higher prices compared to other colour grades. On the other hand, the color grade ‘E’ and ‘D’ has similar median value in the box plot which is quite low and suggesting that diamonds with color grade ‘E’ and color grade ‘D’ tend to command lower prices compared to other colour grades.

2.2.4 Clarity

```
b'SI1'     13065
b'VS2'     12258
b'SI2'      9194
b'VS1'      8171
b'VVS2'     5066
b'VVS1'     3655
b'IF'       1790
b'I1'        741
Name: clarity, dtype: int64
```

Figure 2.2.4.1: Total number of varying degrees of clarity in diamond

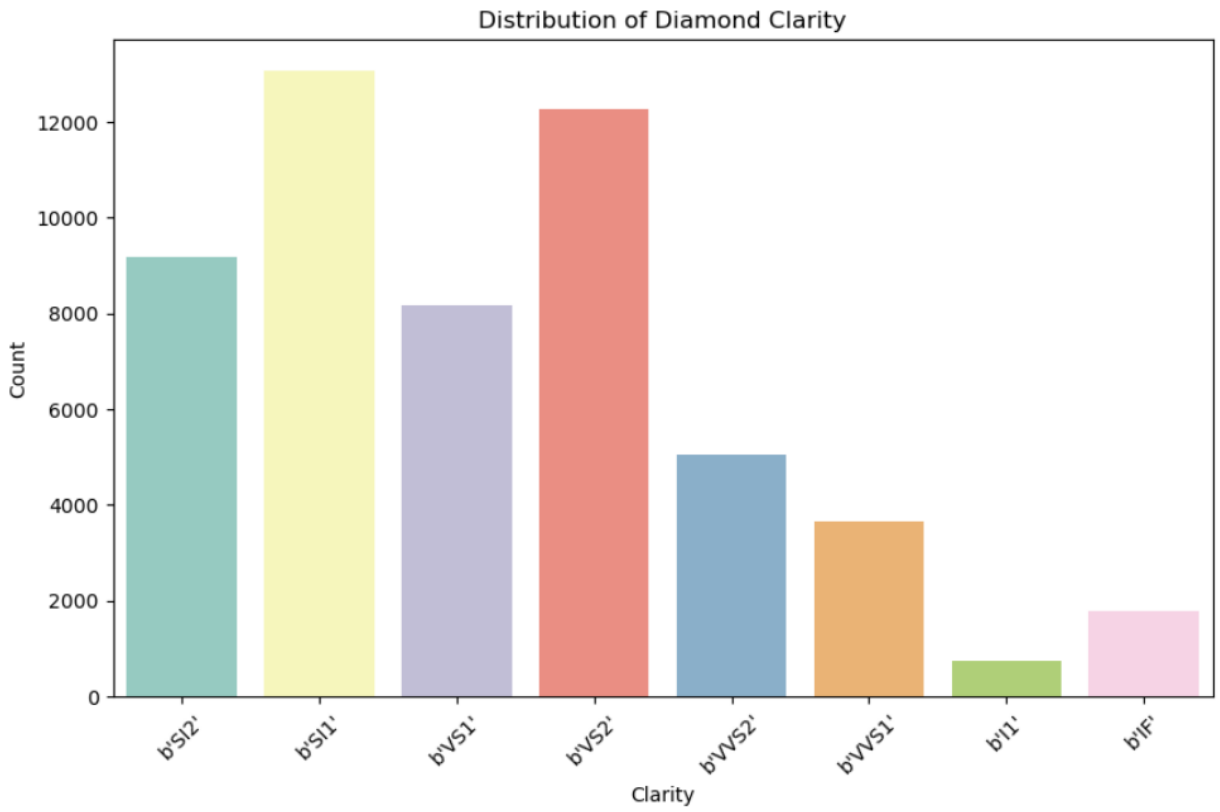


Figure 2.2.4.2: Distribution of diamonds clarity in bar graph

From ***Figure 2.2.4.2*** above, we know that there are 8 groups of different clarity which are 'SI2', 'SI1', 'VS1', 'VS2', 'VVS2', 'VVS1', 'I1' and b'IF'. The most common clarity grade is SI1

followed by VS2, SI2 ,VS1, VVS2, VVS1 and IF . The least common for the diamond clarity in the dataset is I1 with the record of 741. Therefore , 13,065 count of diamonds clarity in the dataset is SI1, 12258 count of diamonds clarity in the dataset is VS, 9194 count of diamonds clarity in the dataset is SI2, 8171 count of diamonds clarity in the dataset is VS1, 5066 count of diamonds clarity in the dataset is VVS2, 3655 count of diamonds clarity in the dataset is VVS1, 1790 count of diamonds clarity in the dataset is IF.

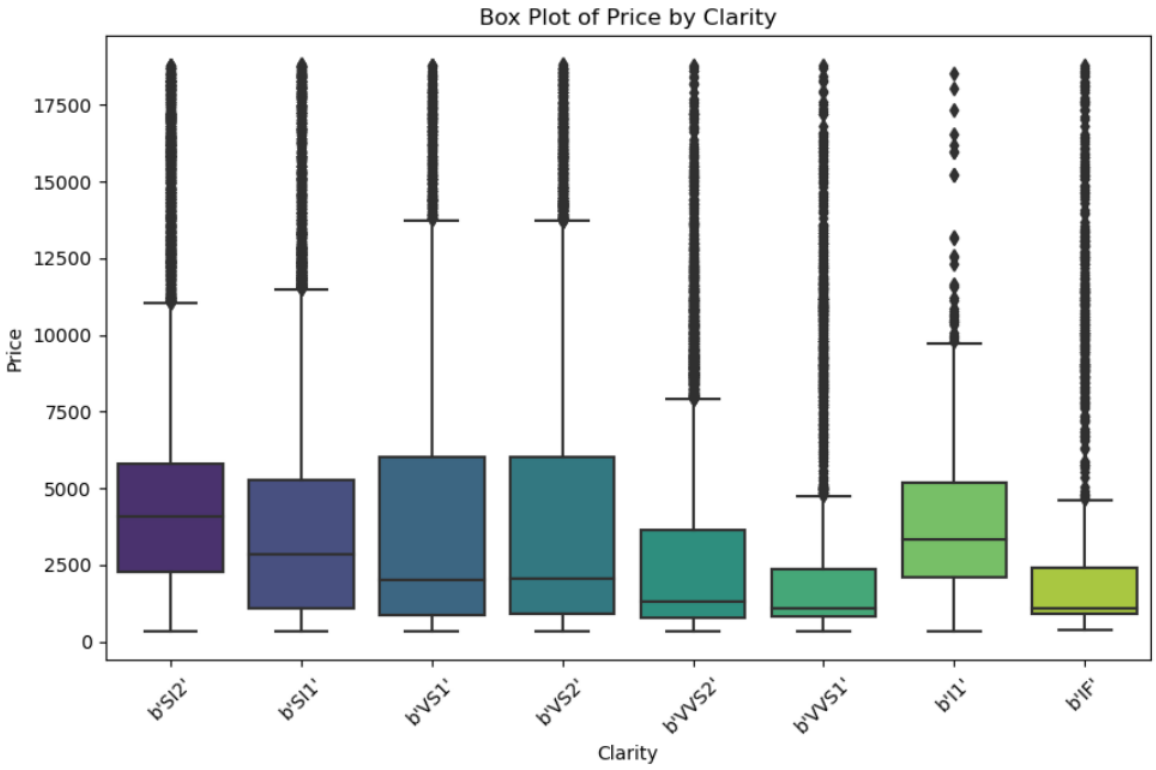


Figure 2.2.4.3: Diamonds price by clarity in box plot

Based on the ***Figure 2.2.4.3***, the clarity 'SI2' has the highest median price, suggesting that, on average, diamonds with a 'SI2' clarity tend to command higher prices compared to other clarity grades. On the other hand, the IF clarity has the lowest median price which shows that, on average, diamonds with the clarity IF will tend to command lower prices compared to other clarity.

2.2.5 Depth

```
Summary of Diamond Depth
----------------------------
count    53940.000000
mean        61.749405
std          1.432621
min         43.000000
25%         61.000000
50%         61.800000
75%         62.500000
max         79.000000
Name: depth, dtype: float64
```

Figure 2.2.5.1: Description of diamond depth

Figure 2.2.5.2: Diamond depth with price in scatterplot

Based on ***Figure 2.2.5.2***, the scatter plot of diamond depth and price shows a weak positive correlation between the two variables. This means that as the depth of a diamond increases, The price of the diamond tends to increase as well but this relationship is not very strong . There are a

few possible reasons for the correlation. One possibility is that deeper diamonds are more valuable because they are more rare. Another possibility is that deeper diamonds are more valuable because they can be cut into larger diamonds. There are also a number of outliers in the scatterplot , which are diamonds that have prices that are much higher or lower than expected given their depth. These outliers could be due to a number of factors, such as the other characteristics of the diamonds like carat, weight, x, y and z or the specific market the diamonds are sold in.

2.2.6 Table

```
Summary of Diamond Table
----------------------------
count    53940.000000
mean        57.457184
std          2.234491
min         43.000000
25%         56.000000
50%         57.000000
75%         59.000000
max         95.000000
Name: table, dtype: float64
```

Figure 2.2.6.1: Description of diamonds table

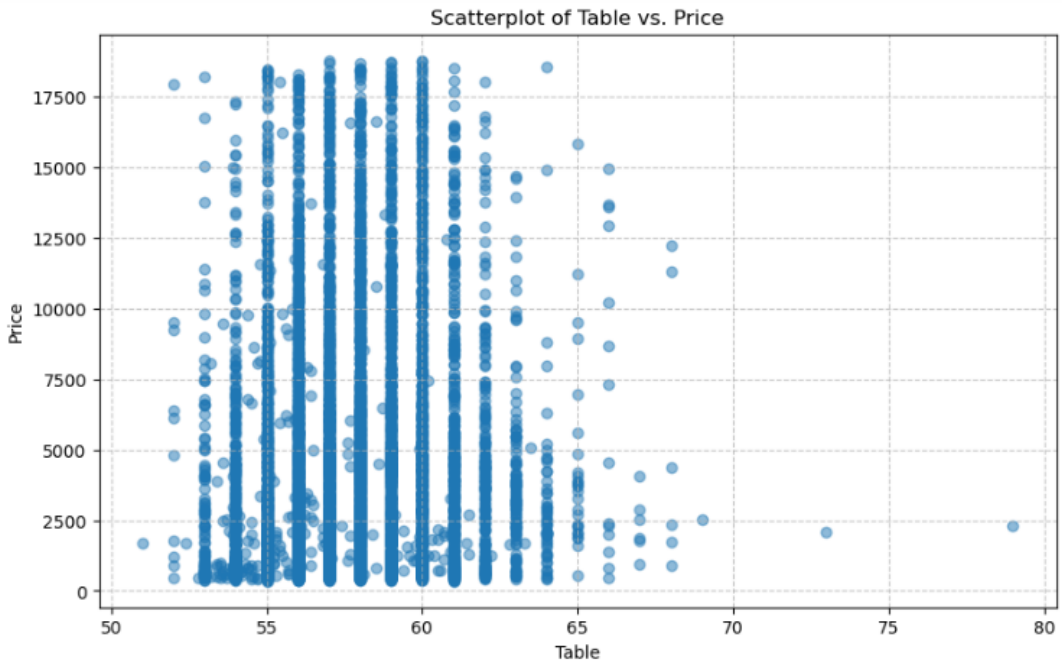


Figure 2.2.6.2: Diamond table with price in scatterplot

This ***Figure 2.2.6.2*** shows the diamond table size and price. This scatterplot shows a positive correlation which means as the table size increases the price of the diamonds tends to increase as well. However,the correlation is not very strong, so there is a lot of variation in the prices of diamonds with table sizes. There are also outliers in the scatterplot , which are diamonds that have prices that are much higher or lower than expected given their table size. These outliers could be due to a number of factors, such as the other characteristics of the diamond.

2.2.7 X

```
Summary of x
--------------------------
count    53940.000000
mean         5.731157
std          1.121761
min          0.000000
25%          4.710000
50%          5.700000
75%          6.540000
max         10.740000
Name: x, dtype: float64
```

Figure 2.2.7.1: Description of X

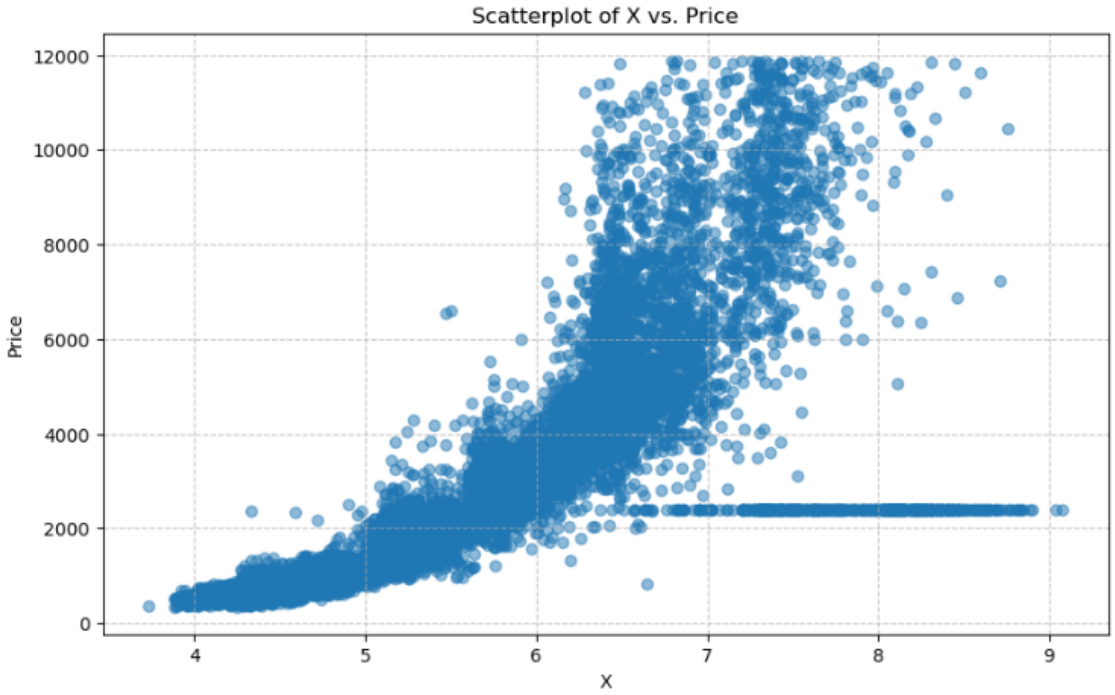


Figure 2.2.7.2: X with price in scatterplot

Based on ***Figure 2.2.7.1***, the statistics for the "X" variable in the diamond dataset offer useful information on how the 53,940 diamonds are distributed. The average diamond in the dataset has a length of about 5.731 millimetres, according to the dataset's mean diamond length. The quartiles show that the majority of diamonds have lengths between 4.71 and 6.54 millimetres, with a median length of about 5.7 millimetres, meaning that half of the diamonds are shorter than

this measurement. The following summary offers valuable information for comprehending the scatterplot of X (length) and price. For the sampled diamonds, the scatterplot shows a visual correlation between X (length) and price. Typically, there is a positive correlation between each pair of diamonds, indicating that the price of the diamond rises as X (length) grows.

2.2.8 Y

```
Summary of y
----------------------------
count    53940.000000
mean         5.734526
std          1.142135
min          0.000000
25%          4.720000
50%          5.710000
75%          6.540000
max         58.900000
Name: y, dtype: float64
```

Figure 2.2.8.1: Description of Y

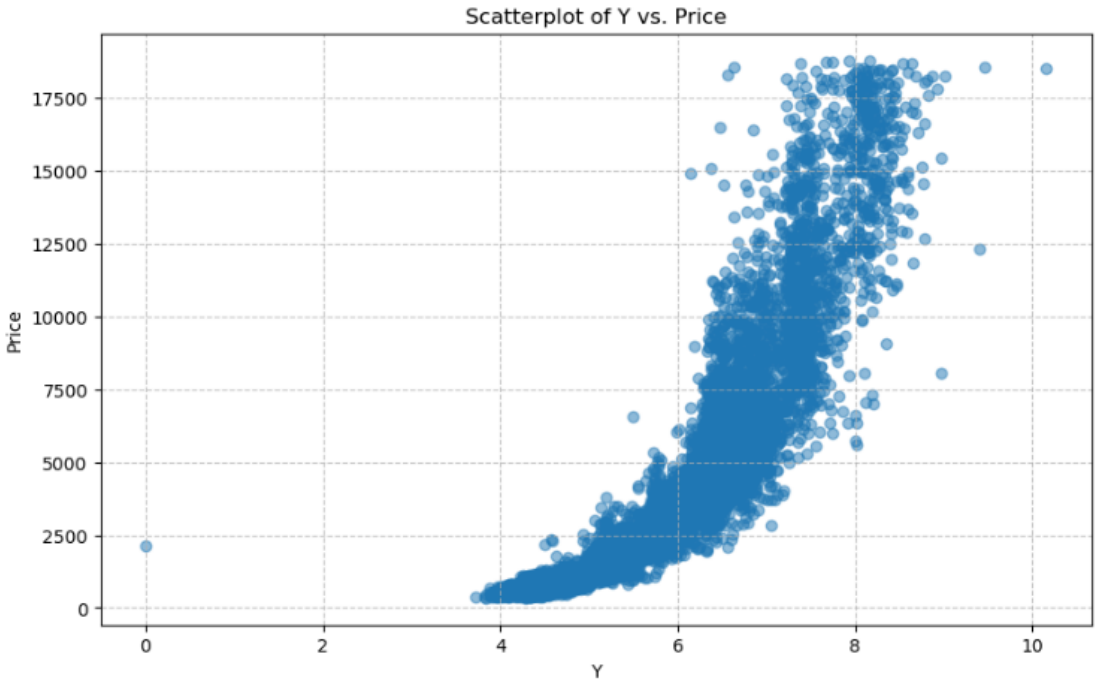


Figure 2.2.8.2: Y with price in scatterplot

The statistics for the "Y" variable in the diamond dataset, as shown in ***Figure 2.2.8.1***, offer important details about the distribution among the 53, 940 diamonds. The typical diamond in the dataset has a width of around 5.7 millimetres, according to the mean of Y (width in millimetres), which is roughly 5.735. With a standard deviation of around 1.122 millimetres, the diamond's

width shows moderate variation, indicating that while many diamonds tend to cluster around the mean, there are also significant deviations. The quartiles show that the majority of diamonds range in width from 4.71 to 6.54 millimetres, with a median width of approximately 5.7 millimetres, meaning that half of the diamonds are smaller than this measurement. However, the largest width of the diamonds that can be found is 58.90 millimetres. For a better understanding of the Y (width) with the price scatterplot, refer to the following summary. For the sampled diamonds, the scatterplot shows a visible relationship between Y (width) and price. Usually, there is a positive correlation between each pair of diamonds, indicating that the price of the diamond rises as Y (width) does.

2.2.9 Z

```
Summary of z
--------------------------
count    53940.000000
mean         3.538734
std          0.705699
min          0.000000
25%          2.910000
50%          3.530000
75%          4.040000
max         31.800000
Name: z, dtype: float64
```

Figure 2.2.9.1: Description of Z

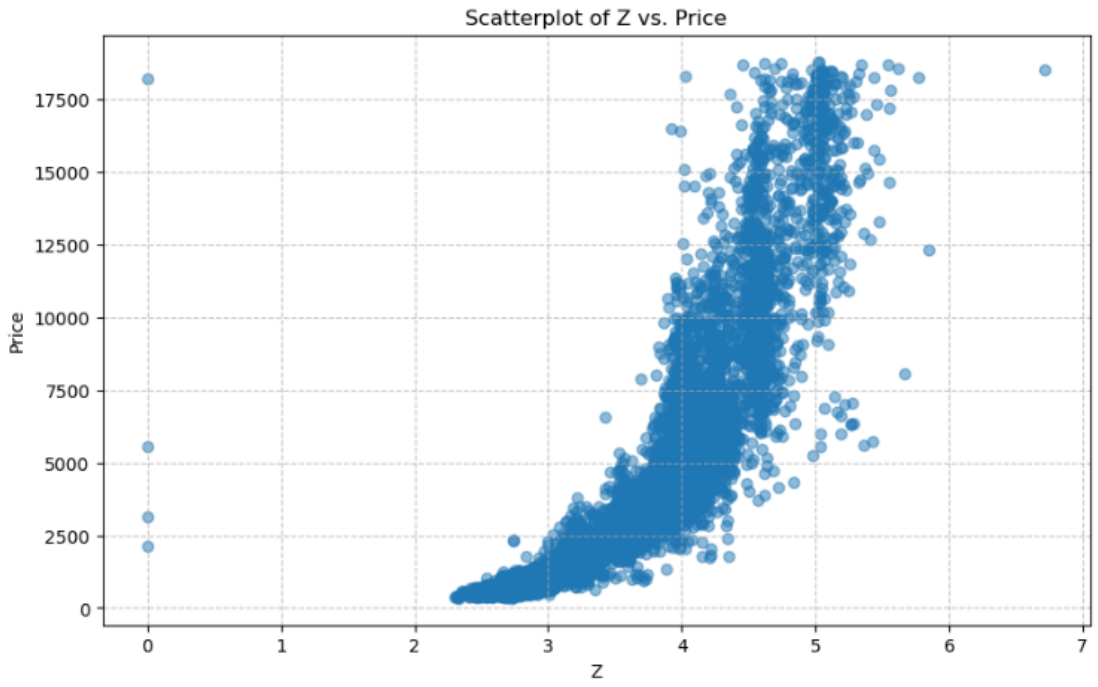


Figure 2.2.9.2: Z with price in scatterplot

The distribution of the 53, 940 diamonds is usefully revealed by the statistics for the "Z" variable in the diamond dataset, according to ***Figure 2.2.9.1***. The average diamond in the dataset has a depth of about 3.5 millimetres, according to the dataset's mean Z (depth in millimetres), which is roughly 3.539. Although many diamonds cluster around the mean, there are also significant variances, as seen by the diamond's major fluctuation in depth, which has a standard deviation of

about 0.706 millimetres. The quartiles show that the majority of diamonds have a depth between 2.91 and 4.04 millimetres, with a median depth of about 3.53 millimetres, indicating that half of the diamonds are shallower than this measurement. The overview provided in the following summary is helpful for comprehending the Z (depth) in the pricing scatterplot. The scatterplot shows a visual relationship between diamond depth and price. Typically, a positive correlation would be anticipated, indicating that as Z (depth) increases, so does the diamond's price.

2.3 Analysis of selected tool

Tools comparison	Remark	Visual Studio Code	Jupyter Notebook	PyCharm Community
Type of licence and open source licence	State all types of licence	MIT License	Modified BSD licence	Apache 2 licensed
Year founded	When is this tool being introduced?	April 2015	2014	3 February 2010
Founding company	Owner	Microsoft	Fernando Pérez and Brian Granger	JetBrains
Licence Pricing	Compare the prices if the licence is used for development and business/commercialization	Open-source and free	Open-source and free	$ 199/1st year $ 159/2nd year
Supported features	What features does it offer?	• It supports cross-platform. • Provides powerful code editing features. • Includes a built-in debugger. • Has a vast library of extensions that provide additional functionality	• Support markdown language in code • Ability to display plots that are the output of running code cells	•A graphical debugger •An integrated unit tester •Integration support for version control systems (VCSs) •Support for data science with Anaconda
Common applications	In what areas is this tool usually used?	Used in a wide range of software development areas such as web development, cloud development, devOps, python development.	Used for all sorts of analysis, visualisations, rapid prototyping, ML and various code practices	Develop python application
Customer support	How the customer support is given, e.g. proprietary, online community, etc.	Online Community	Online Community	Support provided by JetBrains
Limitations	The drawbacks of the software	Not a full-featured integrated development environment (IDE) like PyCharm	No auto-complete	Can’t just run a specific code segment like Jupyter Notebook

2.3.1 Justification of why the selected tool is suitable

It's critical to choose the most appropriate platform for our Python-based project before we go out on an exploration to construct an accurate and effective diamond prediction system. Jupyter Notebook is the technology that, in this situation, perfectly complements the intricate details and ambitious targets of our project. This adaptable system offers a variety of features and advantages that make it the best option for our project to predict diamonds. For instance, we can create interactive charts and graphs in real-time by integrating Jupyter Notebook with data visualisation libraries like Matplotlib and Seaborn. This visual representation of our data makes it easier to see outliers, correlations, and possible model-improvement areas. Furthermore, the cell-based design of Jupyter Notebook encourages modular development. We are able to divide our project into small chunks according to this modular strategy, which enables iterative testing and improvement. We can quickly change and repeat particular sections as we create and improve our prediction models without affecting the workflow as a whole. In summary, Jupyter Notebook proved to be an invaluable tool in our efforts to create an accurate and effective diamond prediction system. Its various features, such as real-time data visualisation and modular programming, perfectly suit the complex requirements of our project. As we move on with this project, utilising Jupyter Notebook's advantages guarantees that we will be well-equipped to handle the challenges of data analysis and model development while keeping a flexible and structured workflow. The technology we have chosen, Jupyter Notebook, will allow us to gain insights that will help our project predict diamonds successfully.

3.0 Data Preparation

The process of preparing raw data for further processing and analysis is known as data preparation. The key processes are to gather, clean, and label raw data in order to make it appropriate for machine learning (ML) algorithms, and then to explore and visualise the data *(Amazon Web Services, Inc., 2021)*. One of the key goals of data preparation is to guarantee that raw data being processed and analysed is correct and consistent so that the outcomes are valid. Missing numbers, inaccuracies, and other errors are common in data, and different sets of data frequently have different formats that must be resolved when joined. Correction of data mistakes, validation of data quality, and integration of data sets are all important aspects of data preparation efforts *(Stedman, 2022)*. Therefore, in our case, first we check for missing values in each column in the dataset. Next, we find duplicate records and create a DataFrame containing only the duplicate records. By doing this, we have a better visualisation on the duplicate records and are able to drop all of the duplicated data effectively.

3.1 Data Selection

	carat	cut	color	clarity	depth	table	price	'x'	'y'	'z'
0	0.23	b'Ideal'	b'E'	b'SI2'	61.5	55.0	326.0	3.95	3.98	2.43
1	0.21	b'Premium'	b'E'	b'SI1'	59.8	61.0	326.0	3.89	3.84	2.31
2	0.23	b'Good'	b'E'	b'VS1'	56.9	65.0	327.0	4.05	4.07	2.31
3	0.29	b'Premium'	b'I'	b'VS2'	62.4	58.0	334.0	4.20	4.23	2.63
4	0.31	b'Good'	b'J'	b'SI2'	63.3	58.0	335.0	4.34	4.35	2.75
...	...	...	...	...	...	...	...	...	...	...
53935	0.72	b'Ideal'	b'D'	b'SI1'	60.8	57.0	2757.0	5.75	5.76	3.50
53936	0.72	b'Good'	b'D'	b'SI1'	63.1	55.0	2757.0	5.69	5.75	3.61
53937	0.70	b'Very Good'	b'D'	b'SI1'	62.8	60.0	2757.0	5.66	5.68	3.56
53938	0.86	b'Premium'	b'H'	b'SI2'	61.0	58.0	2757.0	6.15	6.12	3.74
53939	0.75	b'Ideal'	b'D'	b'SI2'	62.2	55.0	2757.0	5.83	5.87	3.64

53940 rows × 10 columns

Figure 3.1.1 Description of Variables

The process of identifying the proper data type, source, and instruments to gather data is referred to as data selection. Data selection comes before actual data collecting. Data selection is distinguished from selective data reporting and interactive or active data selection by this definition *(Data Selection, 2020)*. Based on ***Figure 3.1***, the datasets are first displayed for decision making on selecting the features which are related to the diamond price. In this case, all of the 10 features are important and have a relevant relationship with the diamond price.

3.1.1 Missing Value

```
Missing Value Information:
         Missing Values Percentage
carat                 0       0.0%
cut                   0       0.0%
color                 0       0.0%
clarity               0       0.0%
depth                 0       0.0%
table                 0       0.0%
price                 0       0.0%
'x'                   0       0.0%
'y'                   0       0.0%
'z'                   0       0.0%
```

Figure 3..1.1.1 Description of the missing value

To provide an overview whether there is any missing value in each column of the dataset, we apply **'missing_values = df.isnull().sum()'** to check it and print it to have a better visualisation on whether there is any missing value. Besides that, we also show the percentage of missing value and based on ***Figure 3.1.1.1***, the percentage of missing value is 0.00% indicates that there is no missing value in each column of the dataset.

3.1.2 Duplicate Records

```
Duplicate Records:
       carat       cut color  clarity  depth  table   price   'x'   'y'   'z'
1005    0.79  b'Ideal'  b'G'   b'SI1'   62.3   57.0  2898.0  5.90  5.85  3.66
1006    0.79  b'Ideal'  b'G'   b'SI1'   62.3   57.0  2898.0  5.90  5.85  3.66
1007    0.79  b'Ideal'  b'G'   b'SI1'   62.3   57.0  2898.0  5.90  5.85  3.66
1008    0.79  b'Ideal'  b'G'   b'SI1'   62.3   57.0  2898.0  5.90  5.85  3.66
2025    1.52   b'Good'  b'E'    b'I1'   57.3   58.0  3105.0  7.53  7.42  4.28
...      ...       ...   ...      ...    ...    ...     ...   ...   ...   ...
47969   0.52  b'Ideal'  b'D'   b'VS2'   61.8   55.0  1919.0  5.19  5.16  3.20
49326   0.51  b'Ideal'  b'F'  b'VVS2'   61.2   56.0  2093.0  5.17  5.19  3.17
49557   0.71   b'Good'  b'F'   b'SI2'   64.1   60.0  2130.0  0.00  0.00  0.00
50079   0.51  b'Ideal'  b'F'  b'VVS2'   61.2   56.0  2203.0  5.19  5.17  3.17
52861   0.50   b'Fair'  b'E'   b'VS2'   79.0   73.0  2579.0  5.21  5.18  4.09

[146 rows x 10 columns]
Percentage of Duplicate Records: 0.27%
```

Figure 3.1.2.1 Description of duplicate value

To provide an overview of the duplicate records among the 52861 rows in the dataset, we apply **'duplicate_mask = df.duplicated()'** to find if there are any duplicate records in the dataset. Next, to create a DataFrame that contains only the duplicate records, we apply **'duplicate_rows = df[duplicate_mask]'** then we print it in order to know how many duplicate records. Besides that, we also show the percentage of duplicate rows and based on ***Figure 3.1.2.1***, 0.27% of rows are duplicated.

3.2 Data Cleaning

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 53794 entries, 0 to 53939
Data columns (total 10 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   carat    53794 non-null  float64
 1   cut      53794 non-null  object
 2   color    53794 non-null  object
 3   clarity  53794 non-null  object
 4   depth    53794 non-null  float64
 5   table    53794 non-null  float64
 6   price    53794 non-null  float64
 7   'x'      53794 non-null  float64
 8   'y'      53794 non-null  float64
 9   'z'      53794 non-null  float64
dtypes: float64(7), object(3)
memory usage: 4.5+ MB

Percentage of Duplicate Rows After Dropping: 0.00%
```

Figure 3.2.1 Dropping Duplicate Records

The process of understanding and correcting wrong information is known as data cleaning. The data could be in the wrong format, have duplicates, be corrupted, incorrect, incomplete, or irrelevant. Various corrections can be done to the data values that represent errors in the data *(knowledgehut, 2021)*. Based on ***Figure 3.2.1***, the process of data cleaning in our case, is dropping all of the duplicate records. Moreover, the duplicated records are successfully dropped to 0.00% where there are no duplicated records anymore in the dataset.

3.3 Data Normalisation

```
      carat        cut color clarity     depth     table     price         x  \
0 -1.199402    b'Ideal'  b'E'  b'SI2' -0.173495 -1.100486 -0.904462 -1.589399
1 -1.241651  b'Premium'  b'E'  b'SI1' -1.362393  1.585691 -0.904462 -1.642938
2 -1.199402     b'Good'  b'E'  b'VS1' -3.390512  3.376475 -0.904211 -1.500168
3 -1.072656  b'Premium'  b'I'  b'VS2'  0.455922  0.242603 -0.902456 -1.366321
4 -1.030407     b'Good'  b'J'  b'SI2'  1.085338  0.242603 -0.902205 -1.241397

          y         z
0 -1.537553 -1.572574
1 -1.660231 -1.742780
2 -1.458689 -1.742780
3 -1.318485 -1.288899
4 -1.213332 -1.118694
```

Figure 3.3.1 Data Normalisation

The process of reorganising data within a database so that users can use it for additional queries and analysis is known as data normalisation. Basically, it is the procedure for producing clean data. This includes removing unnecessary and unstructured data and making all entries and fields appear comparable *(Simplilearn.com, 2021)*. Therefore, in our case, we apply data normalisation by using a-score scaling. Z-score scaling is the process of normalising each value in a dataset so that the mean of all values is zero and the standard deviation is one *(Zach, 2021)*. Therefore, in our case, standardisation guarantees that the scaling of all numerical features in our dataset is consistent. In our situation, we have 10 columns that include various units and ranges such as, carat, depth, and price. Standardisation makes it easier for our model to accurately compare and evaluate the relevance of these features. Moreover, using z-score scaling as our data normalisation techniques can have a lower impact on outliers. Diamond price prediction can be affected by outliers such as extraordinarily large or small diamonds. Because it is based on the mean and standard deviation, Z-score scaling is resistant to outliers, making our model more resilient to extreme values.

3.4 Outliers

```
Outliers Records:
          carat           cut color clarity     depth     table     price  \
2     -1.199402       b'Good'  b'E'  b'VS1' -3.390512  3.376475 -0.904211
8     -1.220527       b'Fair'  b'E'  b'VS2'  2.344171  1.585691 -0.901704
24    -1.030407  b'Very Good'  b'J'  b'SI1' -2.551291  2.033387 -0.897692
35    -1.199402       b'Good'  b'F'  b'VS1' -2.481355  0.690299 -0.885405
42    -1.136029       b'Good'  b'D'  b'VS2'  2.414106 -0.652790 -0.885155
...         ...           ...   ...     ...       ...       ...       ...
53882 -0.185430       b'Fair'  b'D'  b'VS1'  2.553976  0.690299 -0.297403
53886 -0.206555       b'Good'  b'D'  b'VS2' -2.621226  2.033387 -0.296901
53890 -0.143181       b'Good'  b'E'  b'SI1' -2.691161 -1.100486 -0.296901
53895 -0.206555       b'Good'  b'F'  b'VS1' -2.761096  1.585691 -0.296400
53927 -0.016435       b'Good'  b'F'  b'SI1' -2.551291  0.690299 -0.295146

              x         y         z
2     -1.500168 -1.458689 -1.742780
8     -1.660784 -1.712807 -1.487472
24    -1.152166 -1.108180 -1.345634
35    -1.491245 -1.449926 -1.657677
42    -1.553706 -1.502502 -1.317266
...         ...       ...       ...
53882 -0.099238 -0.135518  0.172030
53886  0.043532  0.118600 -0.225116
53890  0.239841  0.197465 -0.111646
53895  0.088148  0.048499 -0.253483
53927  0.293379  0.346431  0.001825

[6378 rows x 10 columns]
Percentage of Outliers: 11.86%
```

Figure 3.4.1 Detect Outliers using IQR

An outlier is a value in a random sampling from a population that is significantly different from the other values *(What are outliers in the data?, 2019)*. Hence, based on ***Figure 3.4.1*** to detect the outliers in our data, we've decided to use IQR. The interquartile range (IQR) is the range of values located in the midpoint of the score distribution. When a distribution is skewed and the median is utilised instead of the mean to demonstrate a central tendency, the Interquartile range is the appropriate measure of variability *(BYJUS, 2020)*. The reason why we apply IQR as our method to detect outliers is The diamond price prediction dataset is likely to include prices ranging from exceedingly high to extremely low. Because it analyses the data's quartiles and is less affected by individual outliers, the IQR approach is resistant to extreme price values. Moreover, detecting outliers is critical for recognizing price inconsistencies in the context of

diamond price prediction. Outliers could be extremely rare or unique diamonds, pricing errors, or fraud attempts. We can easily highlight these odd price points for additional inquiry by using the IQR approach.we managed to detect outliers in each column as displayed above. There are a total of 6378 rows and 11.86% of outliers in total.

```
# Replace outliers with the median value
for column in columns:
    median_value = df[column].median()
    df[column] = df[column].where(~outlier[column], median_value)
```

Figure 3.4.2 Replace outliers with median value

Next, we handle the outliers of our data by replacing outliers with the median value. This is because outliers in the diamond price prediction dataset can have a major impact on the predictive model's performance. We can make our predictions more strong and resistant to the influence of extreme price values by replacing them with the median, resulting in more accurate price estimates for the majority of diamonds. Some outliers in diamond pricing may reflect exceptional or rare diamonds with reasonable prices. Removing them totally may result in the loss of important details. By replacing outliers with the median, we keep data integrity by recognizing the existence of these unusual cases while minimising their impact on the overall analysis.

3.5 Correlation

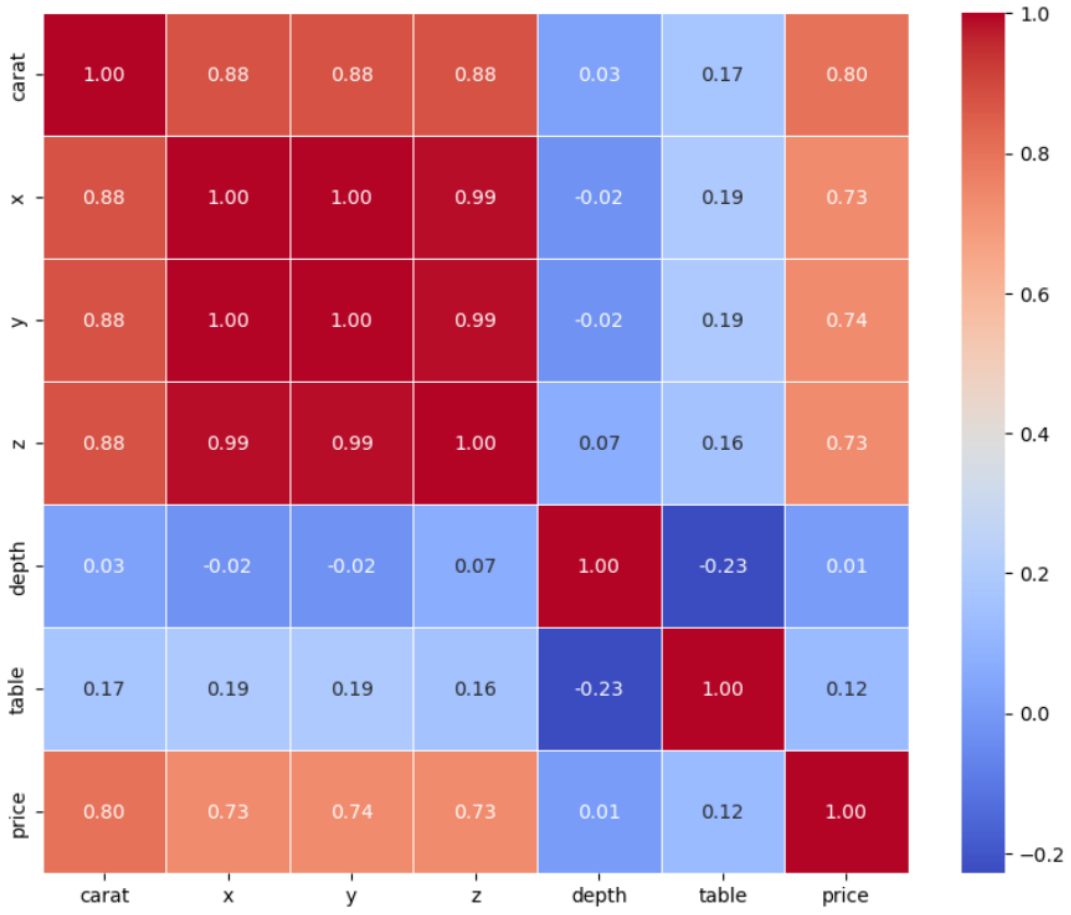


Figure 3.5.1 Correlation

Correlation is used to determine the connection between two variables, which is useful in real life since we may predict the value of one variable using other factors that are connected with it. Because there are two variables involved, it is a sort of two-way statistics *(GOYAL, 2021)*. Therefore, in our case, firstly we calculate the correlation for variables that are in numeric values. Next, we visualise the correlation by plotting a heatmap. We can interpret the heatmap by looking at the colour of the cells and the corresponding values. Darker colours such as dark red represent stronger correlations, while lighter colours such as light blue represent weaker or no correlations. Therefore, based on ***Figure 3.5.1***, we can conclude that carat, x, y and z have stronger correlation while table and depth have weaker correlation.

4.0 Modelling

4.1 Convert Categorical Variables Into Numerical Values

	carat	cut	color	clarity	depth	table	price	x	y	z
0	-1.199402	2.0	1.0	3.0	-0.173495	-1.100486	-0.904462	-1.589399	-1.537553	-1.572574
1	-1.241651	3.0	1.0	2.0	-1.362393	1.585691	-0.904462	-1.642938	-1.660231	-1.742780
2	-1.199402	1.0	1.0	4.0	0.036311	-0.205093	-0.904211	-1.500168	-1.458689	-1.742780
3	-1.072656	3.0	5.0	5.0	0.455922	0.242603	-0.902456	-1.366321	-1.318485	-1.288899
4	-1.030407	1.0	6.0	3.0	1.085338	0.242603	-0.902205	-1.241397	-1.213332	-1.118694

Figure 4.1.1 Convert Categorical Variables Into Numerical Values

Before proceeding to separating data into training and testing sets, we convert categorical variables into numerical values. Based on ***Figure 4.1.1***, categorical variables such as cut, color and clarity have been converted to numerical values. The way to perform this is we used the fitted **'OrdinalEncoder (oe)'** to transform the values in the current column from categorical to numerical. The transformed values replace the original values in the DataFrame **'df'**. Therefore, the purpose of this code is to prepare the categorical data for use in machine learning models, which typically require numerical input features.

4.2 Splitting Dataset into Training and Testing Set

```
from sklearn.model_selection import train_test_split

X = df.drop('price', axis = 1)
y = df['price']

xtrain, xtest, ytrain, ytest = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

Figure 4.2.1 Splitting Dataset into Training and Testing Set

Before performing data modelling using different algorithms, we must split the rows of data into two different sets: training and testing. A specific percentage of the data will be used to feed and train the system or machine so that it may learn the pattern of the data and produce better results when fed different data. While another part of the data will be utilised as a testing set to see if the system can deal with different types of data and give reliable and precise results. Therefore,based on ***Figure 4.2.1***, we have decided to fix 75% of the dataset as the training data while 25% as the testing data.

4.3 Algorithm

4.3.1 Linear Regression

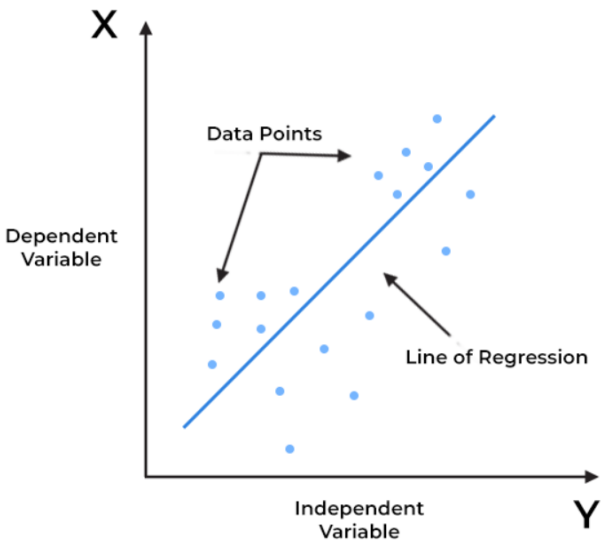


Figure 4.3.1.1 Best Fit Line for a Linear Regression Model (kanade, 2022)

Linear regression is an algorithm that predicts future events by establishing a linear relationship between an independent variable and a dependent variable. It is a statistical method used for predictive analysis in data science and machine learning. The independent variable is also the predictor or explaining factor that remains constant when other factors change. The dependent variable, on the other hand, can differ according to changes in the independent variable. The regression model predicts the value of the dependent variable, which is the response or outcome variable under investigation *(kanade, 2022)*.

Linear regression is used to clarify and predict the behaviour of a variable in various fields, including finance, economics, and psychology. In finance, for example, linear regression may be used to understand the relationship between a company's stock price and profitability, or to predict the future value of a currency based on its historical performance *(Gupta, 2018)*.

Based on ***Figure 4.3.1.1***, the X-axis represents the independent variable, the Y-axis represents the output or dependent variable while the line of regression represents the line of best fit for a model. For the given data points, a line is shown that fits all of the problems. As a result, it is known as the 'best fit line.' The linear regression algorithm's purpose is to identify the best fit line, as seen in the figure above *(kanade, 2022)*.

Therefore, using a linear regression model for predicting diamond prices is consequently a wise decision for various reasons. For example, linear regression is a simple and easy-to-understand method that can reveal significant information into the correlations between characteristics and the target variable, in this case, diamond pricing. We use linear regression to better determine which individual variables, like carat weight, cut quality, and color, contribute to price changes. Furthermore, linear regression is well-suited for cases in which the connection between predictions and the target variable is assumed to be nearly linear, as is the case for diamond pricing, where certain parameters such as carat weight have a linear impact on price. Furthermore, linear regression is relatively efficient and simple to implement, making it an appropriate choice for companies wishing to construct pricing models quickly. Overall, the simplicity, interpretability, and ability to provide valuable pricing insights make linear regression a valuable technique for predicting diamond prices.

4.3.1.1 GridSearchCV for Parameter Setting

```
import numpy as np
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LinearRegression

# Define the parameter grid for GridSearchCV
param_grid = {
    'fit_intercept': [True, False],
    'positive': [True, False],
    'copy_X': [True, False]
}

# Create a Linear Regression model
model = LinearRegression()

# Create the GridSearchCV object
grid_search = GridSearchCV(model, param_grid, cv=5, scoring='neg_mean_squared_error')

# Fit the GridSearchCV object to the data
grid_search.fit(xtrain, ytrain)

# Print the best parameters and best score
print("Best Parameters:", grid_search.best_params_)
print("Best Score (Negative Mean Squared Error):", grid_search.best_score_)

# Get the best model from GridSearchCV
best_model = grid_search.best_estimator_

# Evaluate the best model on the test set
test_predictions = best_model.predict(xtest)

# You can now use test_predictions for evaluation metrics, analysis, etc.
```

```
Best Parameters: {'copy_X': True, 'fit_intercept': True, 'positive': False}
Best Score (Negative Mean Squared Error): -0.15424787285626254
```

Figure 4.3.1.1.1 GridSearchCV for Parameter Setting

Parameter	Best Value
fit_intercept	True
copy_X	True
Negative Mean Square Error	- 0.1542

Table 4.3.1.1.1 Linear Regression Parameter Setting

The use of linear regression Grid search in predicting diamond prices is critical for improving the model's prediction accuracy. The best grid search parameters, specifically **'copy_X'** set to **'True'** and **'fit_intercept'** set to **'True,'** give clarity to the ideal established of the linear regression

model. The **'copy_X'** parameter specifies whether a copy of the input data should be made before fitting the model, and in this case, it indicates that using a copy enhances the model's performance. Meanwhile, **'fit_intercept'** controls whether the model should calculate an intercept or bias term. The fact that it is set to **'True'** suggests that accounting for an intercept term is advantageous in reflecting the inherent variability in diamond pricing. The **'Best Score'** which is shown as a negative mean squared error of around **-0.1542**, demonstrates the predictive quality of the model. The closer this value is to zero, the more closely the model's predictions match actual diamond prices. As a result, the grid search improves the hyperparameters of the linear regression model to ensure that it gives accurate and important insights into diamond price.

4.3.1.2 Accuracy for Test and Training Set

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

# Predictions on the test set using the best model
test_predictions = best_model.predict(xtest)

# Calculate the Mean Squared Error (MSE) for the test set
mse_test = mean_squared_error(ytest, test_predictions)

# Calculate the Mean Absolute Error (MAE) for the test set
mae_test = mean_absolute_error(ytest, test_predictions)

# Calculate the R-squared (R^2) value for the test set
r2_test = r2_score(ytest, test_predictions)

# Calculate the Signal-to-Noise Ratio (SNR) for the test set
snr_test = 10 * np.log10(np.var(ytest) / mse_test)

# Print the calculated metrics
print("Mean Squared Error (MSE) for Testing Set:", f"{mse_test:.2f}")
print("Mean Absolute Error (MAE) for Testing Set:", f"{mae_test:.2f}")
print("Signal-to-Noise Ratio (SNR) for Testing Set:", f"{snr_test:.2f} dB")
print("R-squared for Testing Set:", f"{r2_test:.2f}")
```

```
Mean Squared Error (MSE) for Testing Set: 0.15
Mean Absolute Error (MAE) for Testing Set: 0.22
Signal-to-Noise Ratio (SNR) for Testing Set: 4.61 dB
R-squared for Testing Set: 0.65
```

Figure 4.3.1.2.1 Accuracy for Test and Training Set

Based on ***Figure 4.3.2.1***, the values, including a Mean Squared Error (MSE) of 0.15, a Mean Absolute Error (MAE) of 0.22, a Signal-to-Noise Ratio (SNR) of 4.61 dB, and an R-squared (R^2) of 0.65, offer important insights into the utility and advantages of using linear regression for diamond price prediction. These numerical evaluation metrics shed light on the model's performance. The low MSE of 0.15 signifies that, on average, the model's predictions deviate from the actual prices by a relatively small squared difference, highlighting its ability to make accurate predictions. The MAE of 0.22 provides an additional measure of prediction accuracy, indicating that the model's absolute prediction errors are typically within 22% of the actual price, making it useful for price guidance. Furthermore, the SNR of 4.61 dB underscores the model's signal strength relative to the noise, implying a robust ability to distinguish relevant patterns in the data. Lastly, an R-squared of 0.65 illustrates that approximately 65% of the variance in diamond prices is explained by the model, showcasing its effectiveness in capturing price trends.

4.3.2 KNN

A machine learning approach called K-Nearest Neighbours (KNN) regression is used to forecast continuous numeric values. Finding the K data points in the training dataset that are physically closest to the input data point you want to forecast for is the key idea underlying KNN regression. Usual methods for measuring distance between data points include Euclidean distance. Once the nearest neighbours have been located, a prediction for the query point is then made using their target values, or the values you wish to forecast.

The average (or weighted average) of the target values of the K nearest neighbours is frequently used to calculate the prediction in KNN regression. The anticipated value for the question point is therefore effectively a guess based on the nearest data points' values. In KNN regression, the choice of K is a key hyperparameter; a smaller K will make predictions more susceptible to noise and oscillations in the data, whereas a bigger K will result in smoother predictions but may overlook finer features in the data. The accuracy of the predicted values relative to the actual target values is measured using regression-specific metrics like Mean Absolute Error (MAE) or Mean Squared Error (MSE), which are commonly used to assess the model's performance.

When there is a geographical or local pattern in the data, KNN regression is very helpful since it makes the assumption that data points tend to have similar target values. In high-dimensional areas, though, it could not work as well, and Its performance can be greatly affected by the distance measure and K value that are chosen. KNN regression can be a useful tool for regression jobs when the data demonstrates local dependencies and linkages, despite its simplicity and sensitivity to hyperparameters.

4.3.2.1 GridSearchCV for Parameter Setting

```
from sklearn.model_selection import GridSearchCV
from sklearn.neighbors import KNeighborsRegressor

# Create a KNN regressor object
knn = KNeighborsRegressor()

# Define the hyperparameters and their possible values
param_grid = {
    'n_neighbors': [1, 3, 5, 7, 9],  # List of 'k' values to try
    'p': [1, 2]  # Distances: 1 (Manhattan), 2 (Euclidean)
}

# Create the GridSearchCV object
grid_search = GridSearchCV(estimator=knn, param_grid=param_grid, scoring='neg_mean_squared_error', cv=5)

# Fit the grid search to your data
grid_search.fit(xtrain, ytrain)  # Assuming you have training data (X_train, y_train)

# Get the best hyperparameters from the grid search
best_k = grid_search.best_params_['n_neighbors']
best_p = grid_search.best_params_['p']

# Print the best hyperparameters
print("Best 'k' value:", best_k)
print("Best distance metric 'p':", best_p)

# You can also get the best model from the grid search
best_model_k = grid_search.best_estimator_
```

Parameter	Values to try
n_neighbors	1,3,5,7,9
‘p’	1,2

GridSearchCV, a powerful tool from scikit-learn, is used to find the best hyperparameters for a K-Nearest Neighbors (KNN) regression model. KNN is a simple yet effective algorithm, but the choice of hyperparameters, such as the number of neighbours (k) and the distance metric (p), can significantly impact its performance. GridSearchCV automates the process of systematically trying different combinations of these hyperparameters to find the best configuration.First, a KNeighborsRegressor object (knn) is created, representing the KNN regression model. Then, a dictionary named param_grid is defined, which lists the hyperparameters you want to tune. In this case,exploring different values of 'n_neighbors' (k) and 'p,' where 'p' can be 1 for Manhattan distance or 2 for Euclidean distance.The grid_search object is then fitted to your training data (xtrain and ytrain), which means it systematically trains and evaluates the KNN models with

different combinations of hyperparameters using cross-validation. After the grid search is complete, you can retrieve the best hyperparameters using grid_search.best_params_. In this code, 'best_k' and 'best_p' store the best values for 'n_neighbors' and 'p,' respectively. Additionally, best-performing model can be accessed usinggrid_search.best_estimator_.Overall, this code efficiently explores a range of hyperparameter values to determine the combination that results in the best KNN regression model for your specific dataset, as measured by the chosen scoring metric

4.3.2.2 Accuracy for Test and Training Set

```
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error

# Assuming you have a test dataset (X_test, y_test)
# Make predictions using the best KNN model
y_pred = best_model_k.predict(xtest)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(ytest, y_pred)

# Calculate Mean Absolute Error (MAE)
mae = mean_absolute_error(ytest, y_pred)

# Calculate R-squared (R2) score
r2 = r2_score(ytest, y_pred)

# Calculate Signal-to-Noise Ratio (SNR)
# SNR = Variance of Predicted Values / Variance of Residuals
variance_predicted = np.var(y_pred)
variance_residuals = np.var(ytest - y_pred)
snr = variance_predicted / variance_residuals

# Print the results with two decimal places
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"R-squared (R2) Score: {r2:.2f}")
print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"Signal-to-Noise Ratio (SNR): {snr:.2f}")
```

```
Mean Squared Error (MSE): 0.06
R-squared (R2) Score: 0.87
Mean Absolute Error (MAE): 0.10
Signal-to-Noise Ratio (SNR): 6.77
```

Figure 4.3.2.2.1 Accuracy for test and training dataset

Based on Figure 4.3.2.2.1, the metrics, such as a Mean Squared Error(MSE) of 0.06, R-squared(RS) of 0.87, a Mean Absolute Error (MAE) of 0.10 and the signal to noise ratio (SNR) is 6.77 db. The average squared difference between predicted values and actual values differences is very low. This shows that the predictions are close to the actual prices. R-square measures the proportion of the varian in the dependent variable(price) that is predictable from the independent variable. An R-squared of 0.87 is quite high and it indicates that the model explains a significant portion of the variance in the diamond prices. The model fits the data well and captures most of the price variability. SNR signal to noise ratio suggests that the signal is much stronger than the noise in the predictions. This is good for the model as the model can make reliable predictions.

4.3.3 Bayesian Ridge

By constructing regression models using probability distributors rather than point estimates, Bayesian regression enables a natural mechanism to endure a lack of data or data with an uneven distribution. Instead of being estimated as a single value, the output or response 'y' is supposed to be chosen from a probability distribution.

Mathematically, the answer y is assumed to be Gaussian distributed around $XwXas$ follows in order to construct a completely probabilistic model.

$$p(y \mid X, w, \alpha) = N(y \mid X_w, \alpha)$$

Bayesian Ridge regression, which calculates a probabilistic model of the regression issue, is one of the most practical types of Bayesian regression. Here, spherical Gaussian provides the prior for the coefficient w as follows:

$$p(w \mid \lambda) = N\left(w \mid 0, \lambda^{-1} I_p\right)$$

The model that is produced is known as Bayesian Ridge Regression and is found in the scikit-learn library as sklearn.linear_model. For Bayesian Ridge Regression, use the BeyesianRidge module. *(www.tutorialspoint.com, n.d.)*

For some situations, Bayesian Regression is a desirable option due to its numerous advantages. It excels at handling small datasets and delivers quick and accurate results. It also works well for online learning, removing the need to save a large amount of data in instances when the complete dataset isn't immediately accessible. It can be applied effectively without having in-depth understanding of the dataset due to its mathematical resilience, making it a useful tool for a variety of data analysis jobs. *(Anon, 2022)*

Therefore, choosing a Bayesian Ridge Regression model for predicting diamond prices turns out to be a wise decision for a number of compelling reasons. In this case, the target variable is diamond pricing, and Bayesian Ridge Regression provides an easy-to-understand methodology

that reveals key insights into the correlations between numerous qualities and the target variable. We obtain a better understanding of how specific elements, like carat weight, cut quality, and colour, relate to price variances by using Bayesian Ridge Regression. Additionally, Bayesian Ridge Regression is particularly useful in circumstances when it is anticipated that the relationship between predictions and the target variable would follow a linear or nearly linear pattern, as is the case with diamond pricing, where factors like carat weight have a linear impact on price. Additionally, Bayesian Ridge Regression is effective and simple to use, making it a great option for companies looking to quickly develop pricing models. In summary, Bayesian Ridge Regression is a useful technique for predicting diamond prices due to its combination of simplicity, interpretability, and the ability to produce insightful pricing data.

4.3.3.1 GridSearchCV for Parameter Setting

```
import numpy as np
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import BayesianRidge
from sklearn.metrics import mean_squared_error, mean_absolute_error, make_scorer

import warnings

# Ignore all warnings
warnings.filterwarnings("ignore")

# Define the parameter grid for GridSearchCV
param_grid = {
    'n_iter': [100, 200, 300],  # Number of iterations
    'alpha_1': [1e-6, 1e-7, 1e-8],  # Alpha_1 hyperparameter
    'alpha_2': [1e-6, 1e-7, 1e-8],  # Alpha_2 hyperparameter
    'lambda_1': [1e-6, 1e-7, 1e-8],  # Lambda_1 hyperparameter
    'lambda_2': [1e-6, 1e-7, 1e-8]  # Lambda_2 hyperparameter
}

# Create a Bayesian Ridge Regression model
model = BayesianRidge()

# Create the GridSearchCV object
grid_search = GridSearchCV(model, param_grid, cv=5, scoring=make_scorer(mean_squared_error))

# Fit the GridSearchCV object to the training data
grid_search.fit(xtrain, ytrain)

# Print the best parameters and best score
print("Best Parameters:", grid_search.best_params_)
print("Best Mean Squared Error (MSE):", grid_search.best_score_)

# Get the best model from GridSearchCV
best_bayesian_ridge_model = grid_search.best_estimator_

# Make predictions using the best Bayesian Ridge model
y_pred = best_bayesian_ridge_model.predict(xtest)
```

```
Best Parameters: {'alpha_1': 1e-06, 'alpha_2': 1e-08, 'lambda_1': 1e-08, 'lambda_2': 1e-06, 'n_iter': 100}
Best Mean Squared Error (MSE): 0.14994496491510187
```

Figure 4.3.3.1.1 GridSearchCV for Parameter Setting

Parameter	Best Value
alpha_1	1e-06
alpha_2	1e-08
lambda_1	1e-08
lambda_2	1e-06
n_iter	100
Mean Squared Error	0.1499

Table 4.3.3.1.1 Bayesian Ridge Parameter Setting

Bayesian ridge implementation for the model to be more accurate at predicting diamond values, grid search is essential. The ideal established by the bayesian ridge model is clarified by using the best grid search parameters, especially **'alpha_1'** set to **'1e-06', 'alpha_2'** set to **'1e-08', 'lambda_1'** set to **'1e-08'**, **'lambda_2'** set to **'1e-06'**, and **'n_iter'** set to **'100'**. The hyperparameter **'alpha_1'** and **'alpha_2'** controls how the distribution of regression coefficients is shaped. It's a phrase for regularisation. The regularisation terms **'lambda_1'** and **'lambda_2'** regulate the accuracy of the distribution across the noise. However, **'n_iter'** represents the number of optimisation process iterations. The number of iterations the algorithm will go through to discover the best model parameters, such as the **'100'** iteration defined in this code, is determined by this. The optimisation algorithm's performance may be impacted by the selection of this parameter. Larger values might make it possible for the algorithm to extensively search the solution space, but they might also increase computation time. The precise values selected in the parameter grid are frequently determined by particular or specific knowledge. As an illustration, the values **[1e-6, 1e-7, 1e-8]** are frequently used as a range for regularisation hyperparameters since they vary from relatively weak regularisation **'1e-6'** to higher regularisation **'1e-8'**. However, depending on the dataset and the issue that has to be resolved, different values may be chosen. The **'Best Score'** which is represented as a mean squared error of roughly **0.1499**, shows the model's ability for prediction. The more closely the model's forecasts match actual diamond prices, the nearer this number is to zero. In order to ensure that

the Bayesian ridge model provides precise and significant insights about diamond pricing, the grid search enhances the hyperparameters of the model.

4.3.3.2 Accuracy for Test and Training Set

```
import numpy as np
from sklearn.linear_model import BayesianRidge
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

# Create a Bayesian Ridge Regression model
bayesian_ridge = BayesianRidge()

# Fit the model on the training data
bayesian_ridge.fit(xtrain, ytrain)

# Make predictions on the test set
y_pred = bayesian_ridge.predict(xtest)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(ytest, y_pred)

# Calculate Mean Absolute Error (MAE)
mae = mean_absolute_error(ytest, y_pred)

# Calculate R-squared (R2) score
r2 = r2_score(ytest, y_pred)

# Calculate Signal-to-Noise Ratio (SNR)
# SNR = Variance of Predicted Values / Variance of Residuals
variance_predicted = np.var(y_pred)
variance_residuals = np.var(ytest - y_pred)
snr = variance_predicted / variance_residuals

# Print the results with two decimal places using f-strings
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"R-squared (R2) Score: {r2:.2f}")
print(f"Signal-to-Noise Ratio (SNR): {snr:.2f}")
```

```
Mean Squared Error (MSE): 0.15
Mean Absolute Error (MAE): 0.21
R-squared (R2) Score: 0.66
Signal-to-Noise Ratio (SNR): 2.01
```

Figure 4.3.3.2.1 Accuracy for Test and Training Set

Based on ***Figure 4.3.3.2.1***, the metrics, such as a Mean Squared Error (MSE) of 0.15, a Mean Absolute Error (MAE) of 0.21, an R-squared (R2) of 0.66, and a Signal-to-Noise Ratio (SNR) of 2.01 dB, provide significant insights into the utility and benefits of using a bayesian ridge for predicting diamond prices. These numerical evaluation measures provide information about the model's effectiveness. The model's ability to make accurate predictions is demonstrated by the low MSE of 0.15, which shows that, generally, the model's forecasts and actual prices differ by a minimal squared difference. The absolute prediction errors of the model are often within 21% of

the actual price, making it helpful for price guiding, according to the MAE of 0.21, which gives an additional indicator of forecast accuracy. In addition, the model's SNR of 4.61 dB emphasises its signal-to-noise ratio, indicating an ability to identify important patterns in the data. Last but not least, an R-squared of 0.65 shows that the model accurately represents price patterns, explaining around 65% of the variance in diamond prices.

5.0 Evaluation

5.1 Mean Squared Error (MSE)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

Figure 5.1.1 Mean Squared Error Formula

Mean Squared Error (MSE) is a widely used mathematical metric for evaluating the performance of a predictive model, particularly in regression tasks. It measures the average squared difference between the predicted values and the actual (observed) values in a dataset. MSE stands for the mean squared error , n stands for the data points in the dataset , ŷi stands for the value of the target variable for the i-th data point , Σ stands for the summation symbol , the sum of the squared differences for all data points and then divide by the number of data points. The key reason for using MSE in the system is to quantify how well a predictive model, such as a regression model, is performing.

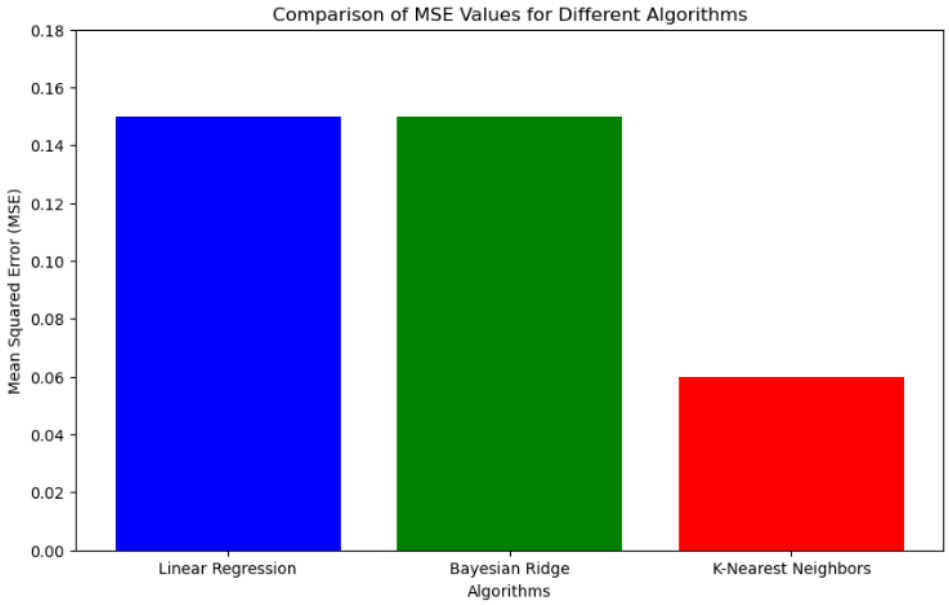


Figure 5.1.2 Comparison of MSE Values for different algorithms

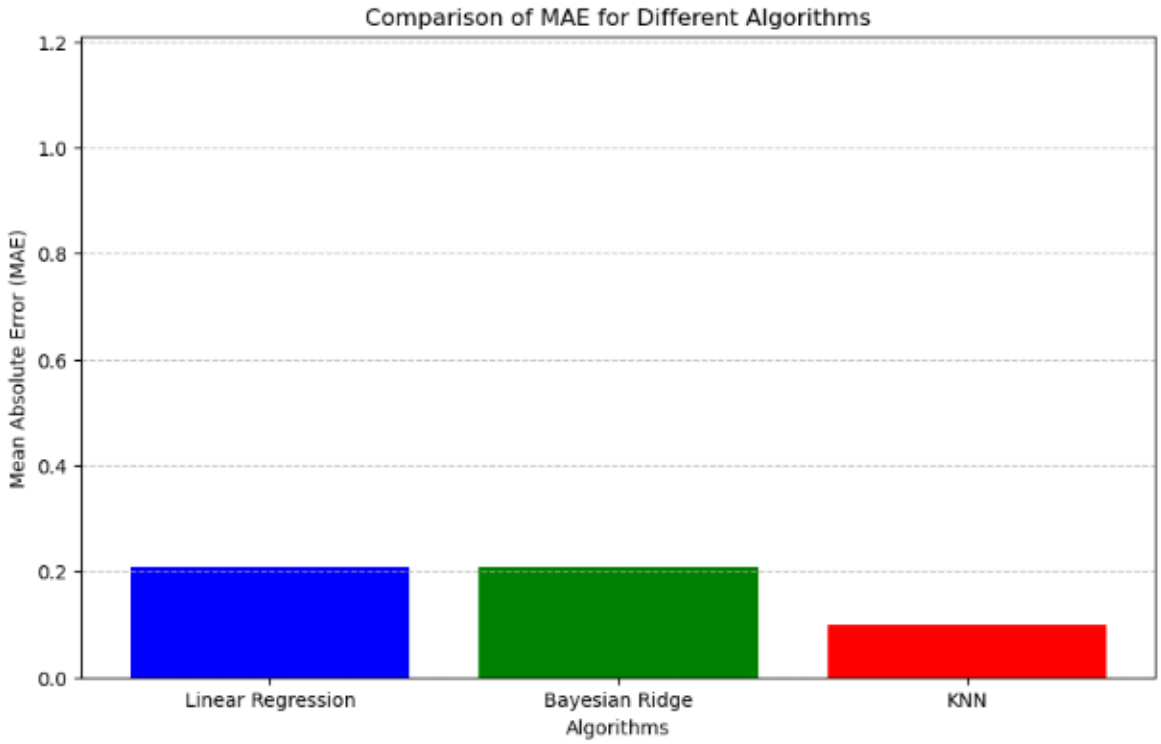
The graph shows the MSE values for three different machine learning algorithms : linear regression , Bayesian ridge, and k-nearest neighbours(KNN). The MSE values are measured on a dataset of diamonds, and the lower the MSE value, the better the algorithm is at predicting the diamond prices.The KNN algorithm has the lowest MSE value, followed by the Bayesian ridge algorithm and then the linear regression algorithm. This means that the KNN algorithm is the best at predicting the diamond prices in this dataset, followed by the Bayesian ridge algorithm and then the linear regression algorithm. . One possibility is that the KNN algorithm is more flexible than the other two algorithms, and it is able to better capture the complex relationships between the different features of the diamonds. Another possibility is that the KNN algorithm is less prone to overfitting than the other two algorithms. Overfitting occurs when an algorithm learns the training data too well and is unable to generalise to new data. Overall, the graph shows that the KNN algorithm is the best at predicting the diamond prices in this dataset, followed by the Bayesian ridge algorithm and then the linear regression algorithm. This is likely because the KNN algorithm is more flexible and less prone to overfitting than the other two algorithms.

5.2 Mean Absolute Error (MAE)

$$MAE = (1/n)\ \Sigma(i=1\ to\ n)\ |y\_i - \hat{y}\_i|$$

Figure 5.2.1 Mean Absolute Error Formula

Mean Absolute Error (MAE) is a measurement of the average size of errors in a set of predictions without taking into account their direction. It is used to determine whether a regression model is working properly and is calculated as the average absolute difference between the predicted values and the actual values *(Deepchecks, n.d.)*. Based on ***Figure 5.2.1***, **n** is the number of observations in the dataset, **y_i** is the true value and **ŷ_i** is the predicted value. Since the MAE is a linear score, every individual variation contributes equally to the mean. The amount of the error is estimated, but not its direction.




```
Linear Regression : 0.21
Bayesian Ridge : 0.21
KNN : 0.10
```

Figure 5.2.2 MAE Value of each algorithm

The MAE amount for various algorithms gives information on how well the model for each approach fits the data. The average error size (MAE), which fails to adjust for the direction of the errors, is measured for a set of estimations. In this case, the independent variables in the model are responsible for explaining the anticipated price value. According to ***Figure 5.2.2*** and an MAE of 0.21, the Bayesian Ridge model and linear regression account for around 21% of the variance in the data. Consequently, predicted values are reasonably near to the actual values and accurate price estimates, and the models successfully capture a significant percentage of the underlying patterns in the data. On the other hand, the MAE of the KNN algorithm is 0.10, which is much lower than those of the other algorithms. This suggests that, in comparison to the linear model and the bayesian ridge model, the KNN model describes a smaller proportion of the variance in the data. This result indicates that KNN beats the linear regression-based models in terms of predicting accuracy for this particular dataset and task. As a result, when selecting the best regression model, it is important to examine the particular needs and goals of the application. KNN stands out as a strong competitor in situations when reducing prediction errors is crucial. In conclusion, MAE is an useful metric for evaluating how well various regression models forecast diamond prices. Although Bayesian Ridge and linear regression both have quite lower values for this prediction, the K-Nearest Neighbours model greatly outperforms them, demonstrating KNN's better predictive accuracy in this situation.

5.3 R-Squared

$$\text{R-Squared} = \frac{SS_{regression}}{SS_{total}}$$

Figure 5.3.1 R-Squared Formula (Taylor, 2020)

In a regression model, R-Squared is a measure of statistics which calculates the proportion of variance in the dependent variable that can be explained by the independent variable. To put it another way, r-squared indicates how well the data match the regression model *(Taylor, 2020)*. Based on ***Figure 5.3.1***, '**SSregression'** is the sum of squares due to regression and **'SStotal'** is the total sum of squares. The sum of squares due to regression measures how well the regression model represents the data used for modelling while the total sum of squares measures the variation in the observed data.

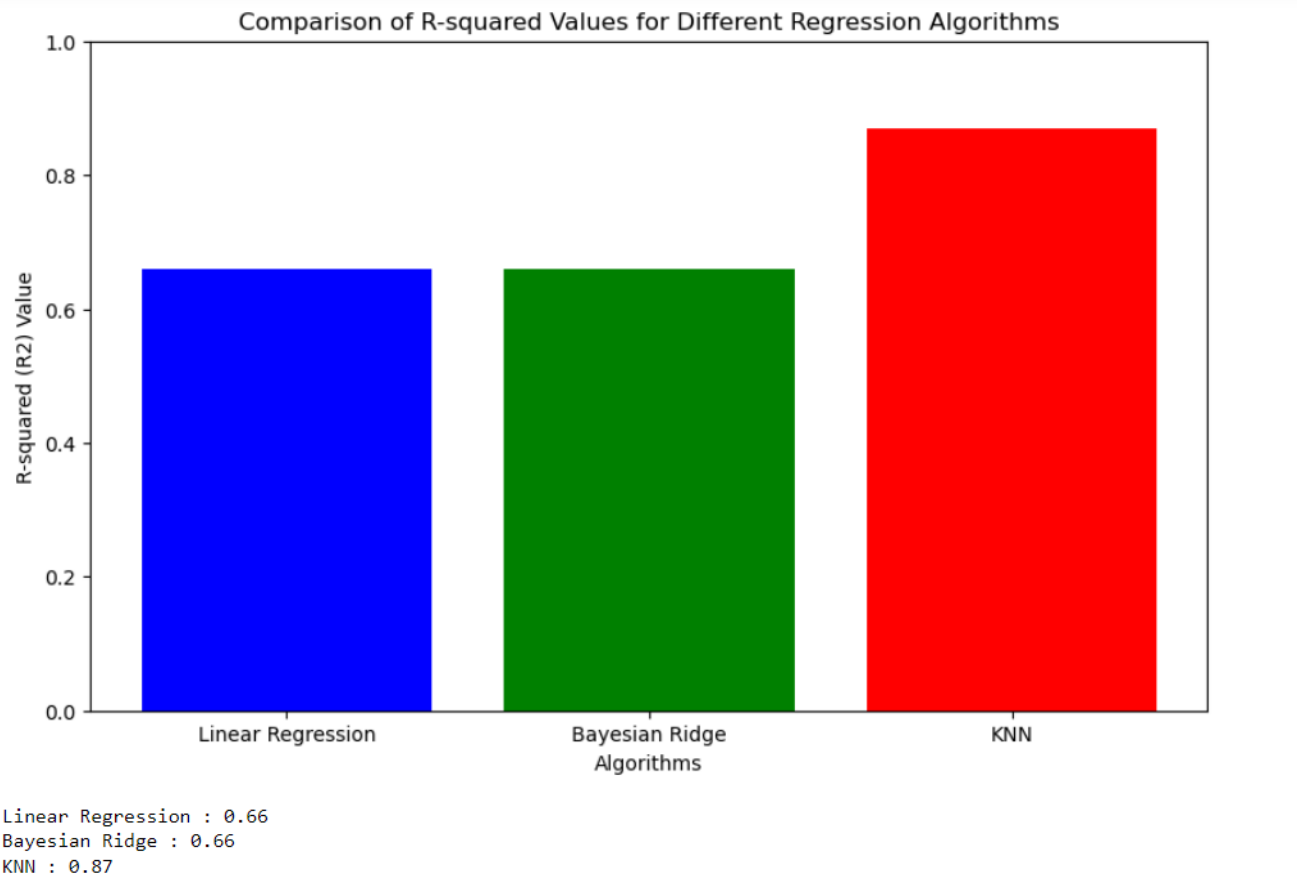


Figure 5.3.2 R-Squared Value of each algorithm

The R-squared (R^2) values for different algorithms provide insights into how well each algorithm's model fits the data. R^2 is a statistical measure that represents the proportion of variance in the dependent variable. In this case, the predicted price values are explained by the independent variables in the model. Based on ***Figure 5.3.2***, an R^2 of 0.66 indicates that the linear regression and Bayesian Ridge model explains approximately 66% of the variance in the data. In other words, the models capture a substantial portion of the underlying patterns in the data, and the predicted values are reasonably close to the actual values. On the other hand, the KNN algorithm demonstrates a significantly higher R^2 of 0.87. This implies that the KNN model explains a larger portion of the variance in the data compared to the linear models. KNN works by considering the similarity of data points, which can be effective when the underlying relationship between features and the target variable is non-linear or complex. The higher R^2

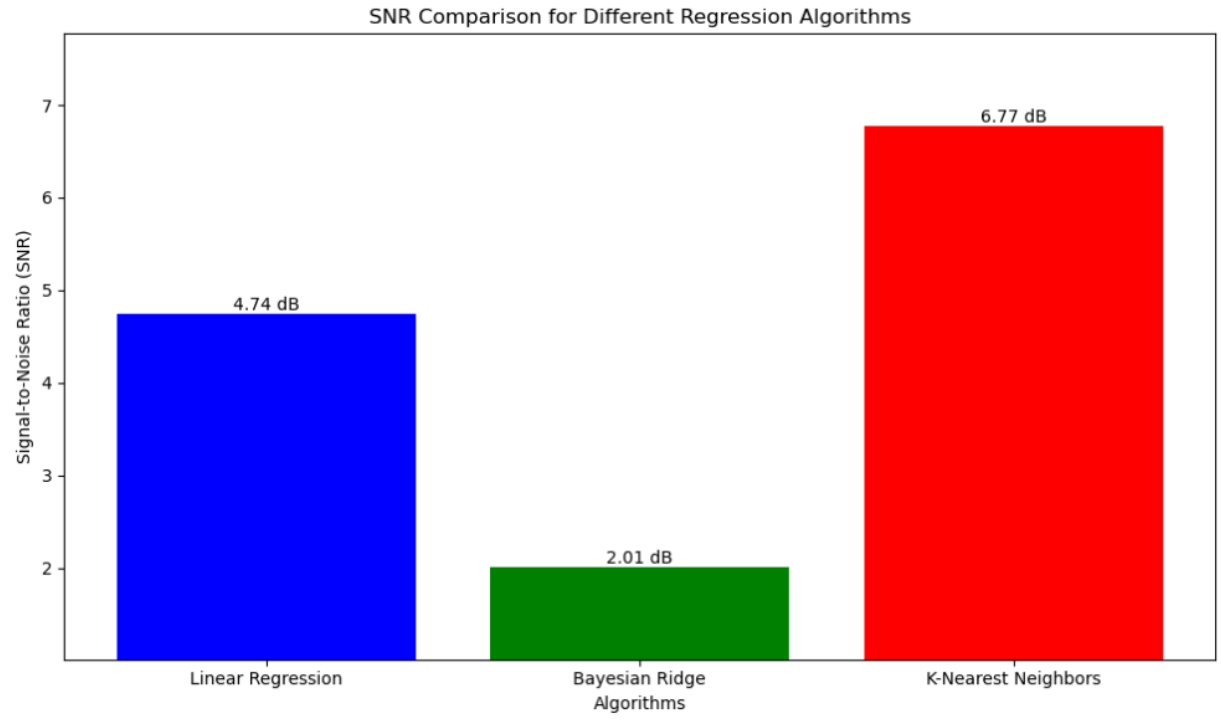
suggests that KNN captures more of the data's complexity, resulting in better predictions. In conclusion, R-squared values provide a quantitative evaluation of model performance, with larger values suggesting better fit. While both linear regression and Bayesian Ridge provide plausible explanations for the data, the K-Nearest Neighbors model significantly exceeds them, suggesting a more effective capture of the underlying data patterns.

5.4 Signal-to-Noise Ratio (SNR)

$$SNR = (P_{signal})/(P_{noise}) = \mu / \sigma$$

Figure 5.4.1 SNR Formula (Cuemath, 2021)

The signal-to-noise ratio quantifies the quantity of background noise compared to the principal input signal. It is formally defined as the signal-to-noise power ratio and is commonly given in decibels. A ratio of 1:1, for example, shows that there is more signal than noise and is greater than 0 decibels (dB) (DeepAI, 2019). Based on ***Figure 5.4.1***, '**Psignal'** is the power of the signal, **'Pnoise'** is the power of the noise, **'μ'** is the signal mean and **'σ'** is the standard deviation of the noise *(Cuemath, 2021)*.




```
Linear Regression : 4.74db
Bayesian Ridge : 2.01db
KNN : 6.77db
```

Figure 5.4.2 R-Squared Value of each algorithm

The provided Signal-to-Noise Ratio (SNR) values for each algorithm offer insights into their respective performances. In this context, higher SNR values signify better signal quality and a superior ability to distinguish useful information from unwanted noise. Based on ***Figure 5.4.2***, Among the three algorithms, K-Nearest Neighbors (KNN) stands out with the highest SNR of 6.77 dB, indicating its effectiveness in classifying data points, particularly when they are closely clustered in feature space. Linear Regression follows with an SNR of 4.74 dB, suggesting reasonable performance for linear relationships between variables. Bayesian Ridge, with an SNR of 2.01 dB, indicates comparatively lower signal quality but may excel in handling data uncertainty. Ultimately, the choice of algorithm should align with the specific dataset characteristics and analysis goals, where KNN proves valuable for tasks demanding robust signal discrimination, while Linear Regression and Bayesian Ridge offer more modest but potentially adequate performance on diamond price prediction.

6.0 Deployment

6.1 Selection of Best Model

By making comparisons with all the models we tested, KNN algorithm has the best results among all the algorithms in terms of MSE, MAE, R-squared and SNR.

In the Evaluation part, the Linear Regression, Bayesian Ridge, KNN Algorithms had these results:

Evaluation	Linear Regression	KNN	Bayesian Ridge
MSE	0.15	0.06	0.15
MAE	0.21	0.10	0.21
R-squared	0.66	0.87	0.66
SNR	4.74	6.77	2.01

Table 6.1.1 Evaluation of each algorithms

Based on ***Table 6.1.1***, it shows the summary of the MSE, MAE, Rsquared and SNR of the Linear Regression, KNN, and Bayesian Ridge. K-Nearest Neighbors (KNN) stands out as the best-performing model for predicting diamond prices among the three algorithms based on a comprehensive evaluation of multiple metrics.

Firstly, let's examine the MSE and MAE metrics. These metrics provide insights into the predictive accuracy of the models. KNN boasts the lowest MSE of 0.06 and MAE of 0.10, signifying that it, on average, generates predictions with the smallest errors compared to Linear Regression and Bayesian Ridge. A lower MSE and MAE are favourable as they indicate superior precision in predicting diamond prices. KNN's remarkable performance in minimising prediction errors highlights its effectiveness in capturing complex patterns within the data.

Moving on to the R-squared (R^2) metric, which measures the goodness of fit of the model, KNN again shines. With an R^2 value of 0.87, KNN demonstrates its prowess in explaining the variance

in diamond prices. This high R^2 value indicates that KNN's predictions closely align with the actual diamond prices, offering an impressive level of explanatory power. In contrast, while Linear Regression and Bayesian Ridge also achieve a reasonable R^2 value of 0.66, KNN's outperformance suggests that it excels in capturing intricate relationships among the input features and the target variable.

Lastly, the Signal-to-Noise Ratio (SNR) metric provides insights into the model's ability to distinguish signals from undesirable inaccuracies. KNN emerges as the leader in this regard, with an SNR value of 6.77 dB, the highest among the three algorithms. A higher SNR indicates superior signal quality, reflecting KNN's proficiency in effectively identifying meaningful patterns in the data while minimising the impact of noise. On the other hand, Bayesian Ridge lags behind with an SNR of 2.01 dB, suggesting a relatively lower ability to filter out noise while Linear Regression also lags behind with an SNR of 4.64db, suggesting an average performance on filter out noise.

In conclusion, K-Nearest Neighbors (KNN) establishes itself as the most promising model for predicting diamond prices due to its impressive performance across multiple evaluation metrics. It excels in minimising prediction errors (MSE and MAE), explaining the variance in prices (R^2), and exhibiting the highest signal quality (SNR) among the three algorithms. These results highlight KNN's adaptability and effectiveness in handling the complexities of the diamond price prediction task, making it a compelling choice for practitioners seeking a reliable predictive model in this domain.

6.2 Actual Price vs Predicted Price

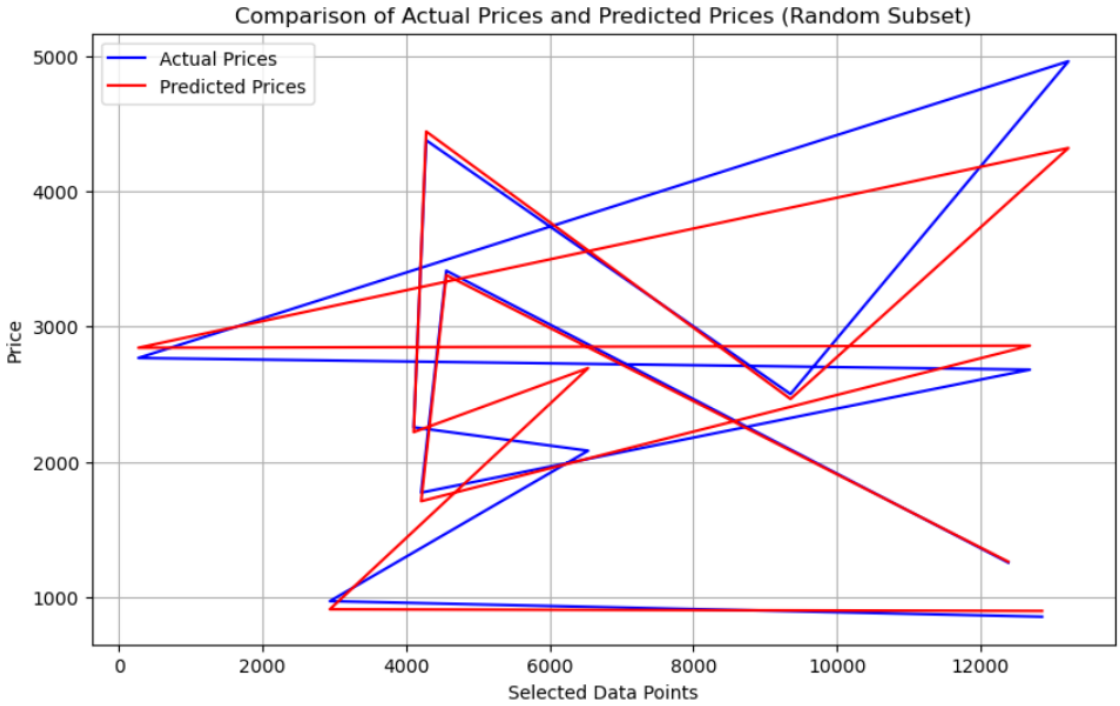


Figure 6.2.1 Line Chart of Actual Price vs Predicted Price

Based on ***Figure 6.2.1***, the line chart above illustrates the comparison between actual prices and predicted prices. Due to the large amount of records in our dataset, we've decided to use random data to make comparisons on actual prices and predicted prices. The x-axis of the chart represents the indices of the selected data points, while the y-axis represents the price values. The blue line on the chart corresponds to the actual prices, with each point representing the real price for a specific data point. In contrast, the red line represents the predicted prices generated by our model for the same selected data points. The purpose of this chart is to visually compare how well our model's predicted prices align with the actual prices for a randomly selected subset of data points. If the red line closely follows the blue line, it indicates that our model's predictions are accurate for the selected data points.

6.3 Model Deployment

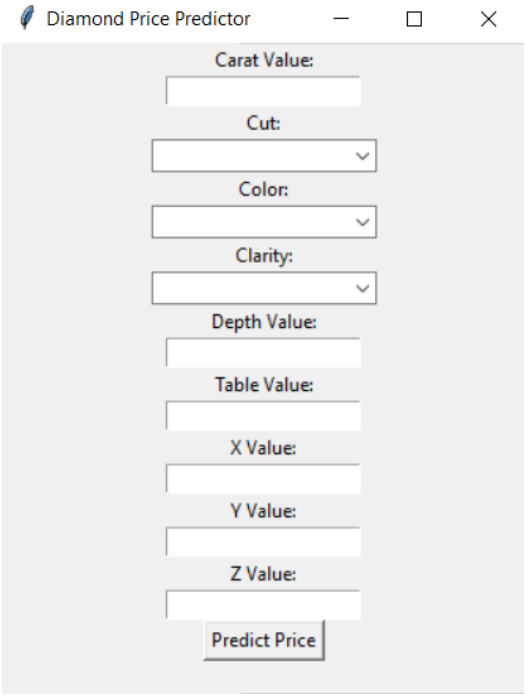


Figure 6.3.1 Diamond Price Predictor using Tkinter

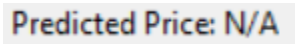


Figure 6.3.2 Error message for wrong data value inside field

Tkinter is a standard Python library for creating Graphical User Interfaces (GUIs). It provides a set of tools and widgets (UI elements) for building windows, dialogs, buttons, labels, text entry fields, and other graphical elements that allow users to interact with a program. Tkinter is easy to use and is particularly well-suited for creating simple to moderately complex desktop applications with GUIs. There are 9 fields for the user to enter the value and choose the value. When the user does not choose anything or the value is incorrect the GUI will show a N/A value which stands for not available. The field Cut, Color, Clarity will let the user choose the type they want and it will automatically convert it into numerical values before calculating the value. The value will be normalised , after that it will send the normalised value to the KNN regression model and it will calculate the best predicted price to the user. KNN regression is chosen because it has the most accurate predicted value compared to the other model.

7.0 Conclusion

In conclusion, the K-Nearest Neighbours (KNN) model is the one that should be used to estimate the price of diamonds. The KNN consistently outperforms the other models in terms of accuracy, even when additional conditions are utilised, such as deleting records with or without outliers. Moreover, the model's performance is clearly enhanced by incorporating the IQR to identify outliers and replace them with the median value, as well as by removing duplicate data that was discovered and using z-score scaling to normalise the data so that outliers can be less impactful. This is to ensure the model can more precisely forecast diamond price results for independent variables entered by the user by increasing the adaptability.

The first objective has been accomplished because the deployment results demonstrate that the KNN is capable of correctly predicting the majority of diamond price outputs after training on the dataset provided by SK Jewellery. Thus, by supplying the KNN model with information about diamond carat, cut, colour, clarity, depth, table, X, Y, and Z, SK Jewellery may be able to predict the price of diamonds. The second objective has been accomplished effectively, allowing us to extract the key characteristics such as diamond carat, cut, clarity, depth, X, Y, Z, and price results. Furthermore, we effectively preprocess the data supplied by SK Jewellery by removing the outliers and substituting the outlier with the median using regression. All information is displayed using visualisation techniques such as bar graph, pie charts, box plots, scatterplot and histogram. The third objective was to establish an accurate model to estimate diamond prices in order to leverage our organisation's position and improve supplier negotiations. As a result, we think that by enhancing supplier relationships and diamond sourcing techniques, our data model could help SK Jewellery increase its revenue.

By completing this assignment, we are able to utilise the knowledge we learned in the lecture and practical class to use in building a data model based on actual data supplied by SK Jewellery. This enables us to comprehend the complete procedure of managing data, assessing it, training it, and successfully constructing a data model. Although we initially encountered considerable challenges in understanding and processing the data, with the help of our tutor, Miss Noor Aida, and the team's weekly meetings, we were eventually able to get a deeper knowledge and create a high-accuracy data model. Finally, we would want to express our gratitude for being able to

finish this assignment on time and to the best of our abilities. In the future, we want to investigate a wider range of datasets and enhance our data model with a variety of preprocessing techniques.

Reference

1. Anon, (2015). *SOOKEE HQ | SK JEWELLERY GROUP*. [online] Available at: https://www.skjewellerygroup.com/sookee-hq/ [Accessed 19 Aug. 2023]

2. Find Out More About SK Jewellery Singapore, (2020). [online] Available at: https://www.skjewellery.com/about/ [Accessed 19 Aug. 2023]

3. Diamondregistry (2021). Analysing the Decline: Factors Influencing the Fall in Diamond Prices. [online] Available at: https://www.diamondregistry.com/news/analyzing-the-decline-factors-influencing-the-fall-in-diamond-prices/ [Accessed 28 Aug. 2023].

4. labrilliante.com. (2022). How Lab-Grown Diamonds Could Impact the Industry In the Nearest Future - Labrilliante. [online] Available at: https://labrilliante.com/about-us/labrilliante-blog/how-lab-grown-diamonds-could-impact-the-industry-in-the-nearest-future.

5. Amazon Web Services, Inc. (2021). What is Data Preparation - Data Preparation Explained - AWS. [online] Available at: https://aws.amazon.com/what-is/data-preparation/#:~:text=Data%20preparation%20is%20the%20process.

6. Stedman, C. (2022). What is Data Preparation? An In-Depth Guide to Data Prep. [online] SearchBusinessAnalytics. Available at: https://www.techtarget.com/searchbusinessanalytics/definition/data-preparation.

7. Data Selection. (2020). Javatpoint. [online] Available at: https://www.javatpoint.com/data-selection-in-data-mining.

8. knowledgehut. (2021). Data Cleaning in Data Science: Process, Benefits and Tools. [online] Available at: https://www.knowledgehut.com/blog/data-science/data-cleaning#what-is-data-cleaning-in-data-science?-%C2%A0 [Accessed 30 Aug. 2023].

9. Simplilearn.com. (2021). What is Normalization of Data in Database? | Simplilearn. [online] Available at: https://www.simplilearn.com/automated-recruiting-in-companies-article#:~:text=Data%20normalization%20is%20the%20process.

10. Zach (2021). Z-Score Normalization: Definition & Examples. [online] Statology. Available at: https://www.statology.org/z-score-normalization/#:~:text=Z%2Dscore%20normalization%20refers%20to.

11. What are outliers in the data? (2019). www.itl.nist.gov What are outliers in the data? [online] Available at: https://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm#:~:text=Definition%20of%20outliers.

12. BYJUS. (2020). Interquartile Range (IQR) | Definition, Formula & Examples. [online] Available at: https://byjus.com/maths/interquartile-range/#:~:text=The%20interquartile%20range%20(IQR)%20is.

13. GOYAL, C. (2021). Correlation | Intuition Behind Correlation - Definition and It's Types. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/04/intuition-behind-correlation-definition-and-its-types/#:~:text=Correlation%20is%20used%20to%20find.

14. kanade, vijay (2022). What Is Linear Regression? Types, Equation, Examples, and Best Practices for 2022. [online] Spiceworks. Available at: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-linear-regression/

15. Gupta, M. (2018). ML | Linear Regression - GeeksforGeeks. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/ml-linear-regression/.

16. Taylor, S. (2020). R-Squared. [online] Corporate Finance Institute. Available at: https://corporatefinanceinstitute.com/resources/data-science/r-squared/.

17. Cuemath. (2021). Signal to Noise Ratio Formula - Learn Formula Signal to Noise Ratio. [online] Available at: https://www.cuemath.com/signal-to-noise-ratio-formula/.

18. DeepAI. (2019). Signal-to-noise Ratio. [online] Available at: https://deepai.org/machine-learning-glossary-and-terms/signal-to-noise-ratio#:~:text=A%20Signal%2Dto%2Dnoise%20ratio%20is%20a%20measure%20of%20the [Accessed 15 Sep. 2023].

19. www.tutorialspoint.com. (n.d.). Scikit Learn - Bayesian Ridge Regression. [online] Available at: https://www.tutorialspoint.com/scikit_learn/scikit_learn_bayesian_ridge_regression.htm [Accessed 16 Sep. 2023].

20. Simplilearn.com. (2022). Introduction To Bayesian Linear Regression | Simplilearn. [online] Available at: https://www.simplilearn.com/tutorials/data-science-tutorial/bayesian-linear-regression#advantages_of_bayesian_regression [Accessed 16 Sep. 2023].

21. Deepchecks. (n.d.). What is Mean Absolute Error. [online] Available at: https://deepchecks.com/glossary/mean-absolute-error/ [Accessed 16 Sep. 2023].