DIAMOND PRICE PREDICTION

Prepared By
Tharindu Darshana

Contents

Introduction

Diamonds have long been regarded as symbols of luxury, value, and timeless elegance. The determination of a diamond's price involves a combination of factors that reflect its physical characteristics, including the four Cs: carat weight, cut, color, and clarity. However, the complexity of this pricing mechanism poses challenges for both buyers and sellers.

In this report, we aim to predict diamond prices using advanced data analysis and machine learning techniques. By leveraging a dataset that includes key attributes such as carat, cut, color, clarity, and additional specifications, we seek to identify the most significant predictors of price. The results of this project could benefit various stakeholders, including jewelers, investors, and consumers, by offering a more precise and systematic understanding of diamond pricing.

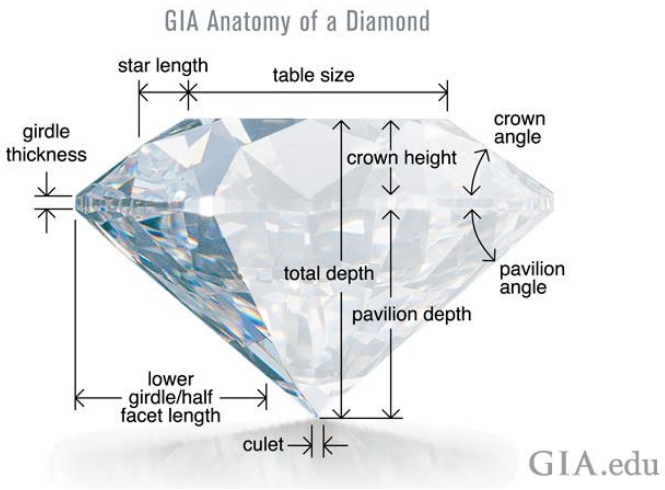
Data

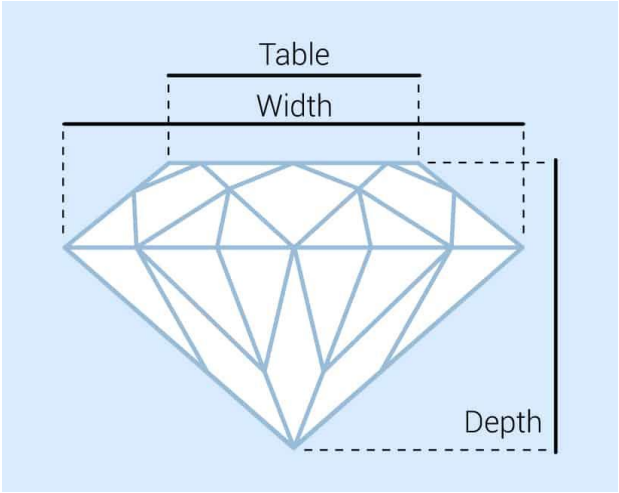
The dataset used for this study consists of 53,940 observations with 10 features. that capture various characteristics of diamonds. The key attributes in the dataset are as follows:

- **Carat**: A numeric variable representing the weight of the diamond, which is one of the most significant factors influencing its price. One carat is equal to 1/5 g.
- **Cut**: A categorical variable indicating the quality of the diamond's cut (e.g., Fair, Good, Very Good, Premium, Ideal). The cut affects the diamond's brilliance and is an important qualitative measure.
- **Color**: A categorical variable ranking the diamond's color from D (best, colorless) to J (poorest, faint yellow).
- **Clarity**: A categorical variable describing the diamond's internal and external inclusions, rated from I3 (lowest) to FL (highest, flawless).
- **Depth**: A numeric variable representing the height of the diamond as a percentage of its width.
- **Table**: A numeric variable describing the width of the diamond's top relative to its widest point (percentage).
- **Price**: A numeric variable representing the price of the diamond in US dollars, which serves as the response variable in the analysis.
- **x, y, z**: Numeric variables representing the diamond's length, width, and depth dimensions, respectively, in millimeters.

Diamond Measurements

Since there are several terms related to measurements of the diamond, let's first understand these measurements and characteristics to gain a comprehensive understanding of the dataset. Among these attributes, the **Four Cs**: Carat**,** Cut**,** Color**,** and Clarity serve as the universally accepted standard for assessing diamond quality. In addition to the Four Cs, the dataset also includes other critical measurements, such as: depth percentage, table, length, width and depth. We can better understand these measurements by the following picture.





Here are the measurement labels shown in the picture and their corresponding column names in our dataset.

table size / Table	-	table
total depth	-	depth
Width	-	y
total depth / Depth	-	z
star length	-	x

To make it more clear about cut, color and clarity levels of a diamond, let's rank them in order

Cut: Fair < Good < Very Good < Premium < Ideal

Cut	Description
Fair	Cut allows some light to escape from the bottom or sides of the diamond
Good	The diamond reflects more light than Fair but lacks optimal brilliance
Very Good	A well-cut diamond that offers excellent brilliance but may fall short of perfection
Premium	Near the top of the scale, with exceptional brilliance.
Ideal	The cut maximizes light reflection and refraction, resulting in maximum brilliance

Color: D (Colorless (Best)) < E < F < G < H < I < J (Faint Color (Worst))

Clarity: I1 < SI2 < SI1 < VS2 < VS1 < VVS2 < VVS1 < IF

Clarity	Description
I1	Included
SI2	Slightly Included
SI1	Slightly Included
VS2	Very Slightly Included
VS1	Very Slightly Included
VVS2	Very, Very Slightly Included
VVS1	Very, Very Slightly Included
IF	Internally Flawless (Best)

Methodology

The process of analyzing and predicting diamond prices in this study involves several carefully structured steps. The methodology can be summarized as follows:

Data Preprocessing:
Before any analysis, the dataset is preprocessed to ensure it is clean, consistent, and ready for further exploration. This includes handling missing values and categorical variables are encoded for compatibility with machine learning models.

Descriptive Analysis and Exploratory Data Analysis (EDA):

Once the data has been preprocessed, a detailed description of the sample data is carried out to understand the distribution and the features in the dataset. Exploratory data analysis (EDA) is performed to uncover underlying patterns and relationships among variables. Visualizations, such as histograms, scatter plots, and correlation heatmaps, are utilized to better understand how features influence diamond prices.

Outlier Detection and Removal:
During EDA, potential outliers are identified that could skew model performance. Those anomalies are removed to ensure that the model is trained on accurate and representative data, leading to more reliable predictions and improved overall performance.

Model Training and Evaluation:
Multiple machine learning models are explored to identify the most accurate predictor of diamond prices. Models such as Linear Regression, Random Forests, and Gradient Boosting are trained and evaluated using performance metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Cross-validation is employed to ensure the robustness and reliability of the models.

Web API Development:
Once the best-performing model is selected, it is used to create a web API using Flask. The API accepts user inputs for diamond features (e.g., carat, cut, color, clarity) and provides an accurate price prediction. This API is designed with a focus on user-friendliness and efficiency, ensuring seamless interaction for end-users.

Deployment:
The final step involves deploying the Flask-based web API, making it accessible for real-world use. Deployment ensures that the predictive tool can serve stakeholders as well as customers effectively, offering quick and accurate diamond price predictions based on the selected model.

Descriptive Analysis

This section focuses on preprocessing the data to make it suitable for the EDA, and perform EDA to uncover underlying patterns and finally removing potential outliers to make the data suitable to train a predictive model.

Data Preprocessing

```
df.columns
```

```
Index(['Unnamed: 0', 'carat', 'cut', 'color', 'clarity', 'depth', 'table',
       'price', 'x', 'y', 'z'],
      dtype='object')
```

```
# The column 'Unnamed: 0' is unnecessary. So we proceed to remove it
```

When checking column names it can be seen that there is an unnecessary column in the dataset, so we remove it.

```
df.describe()
```

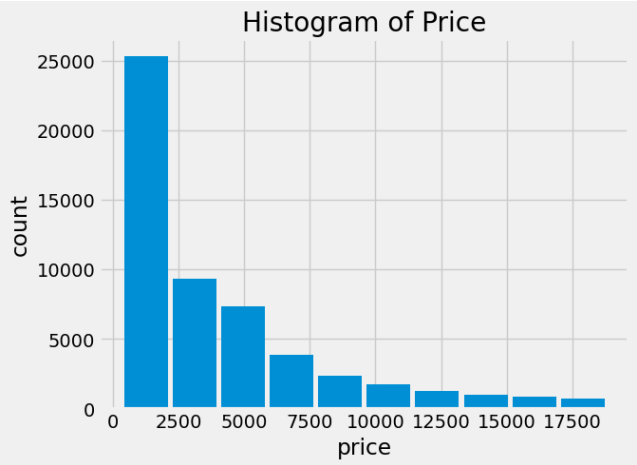
	carat	depth	table	price	x	y	z
count	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000
mean	0.797940	61.749405	57.457184	3932.799722	5.731157	5.734526	3.538734
std	0.474011	1.432621	2.234491	3989.439738	1.121761	1.142135	0.705699
min	0.200000	43.000000	43.000000	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

When we look at the summary of the numerical variables, it can be seen that there are data points with x,y,z values zero. Since these measurements are related to the length, width, and depth of the diamond, they cannot be equal to zero. So they might indicate missing values or data errors. Since we have a rich amount of data, we proceed to drop the observations with either x,y or z variables equal to zero to ensure the reliability of the analysis.

Exploratory Data Analysis (EDA)

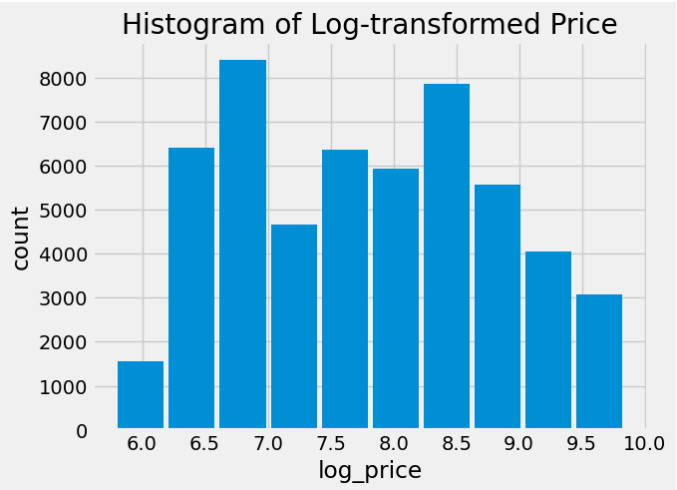
This section is dedicated to exploring distributions of variables and the relationships between those variables.

Univariate Analysis



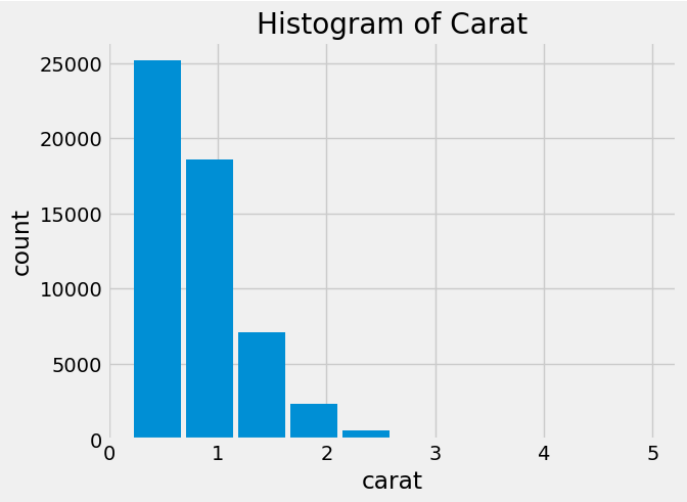
Statistic	Value
Count	53920
Min	326
Max	18823
Mean	3930.99
Variance	1.58984e+07
Skewness	1.6183
Kurtosis	2.17808

The variable diamond price is heavily right skewed meaning that the majority of diamonds have low prices while few diamonds with very high prices. Since the skewness of the price is greater than 0, (1.6) it is confirmed that the variable is right skewed. Let's apply some transformation to this variable to make it normal, which is useful in future works.



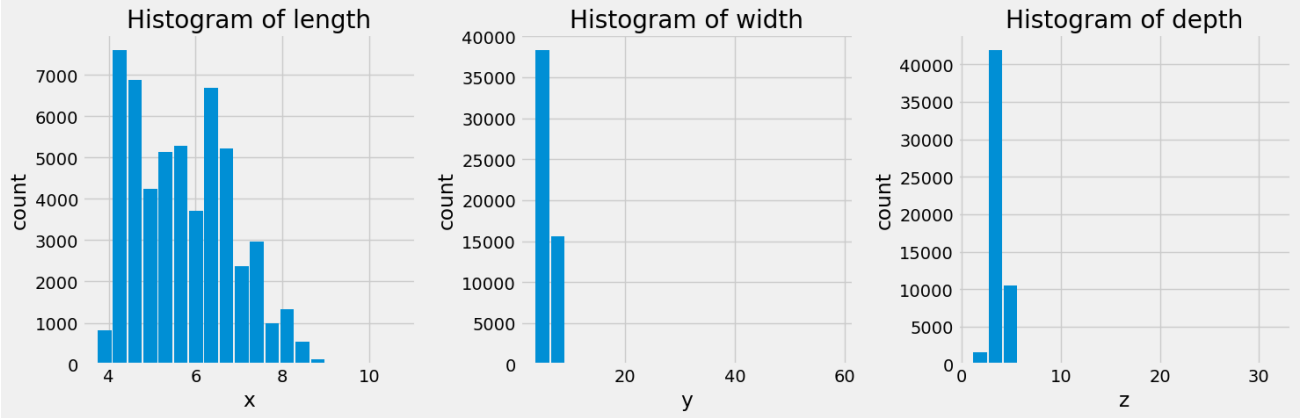
Statistic	Value
Count	53920
Min	5.7869
Max	9.84283
Mean	7.78639
Variance	1.02929
Skewness	0.115458
Kurtosis	-1.09699

It can be seen that after applying log-transformation to the variable 'price', it becomes close to normal distribution. Here we can see in the table the skewness value is closer to zero (0.11) indicating that it is less skewed.

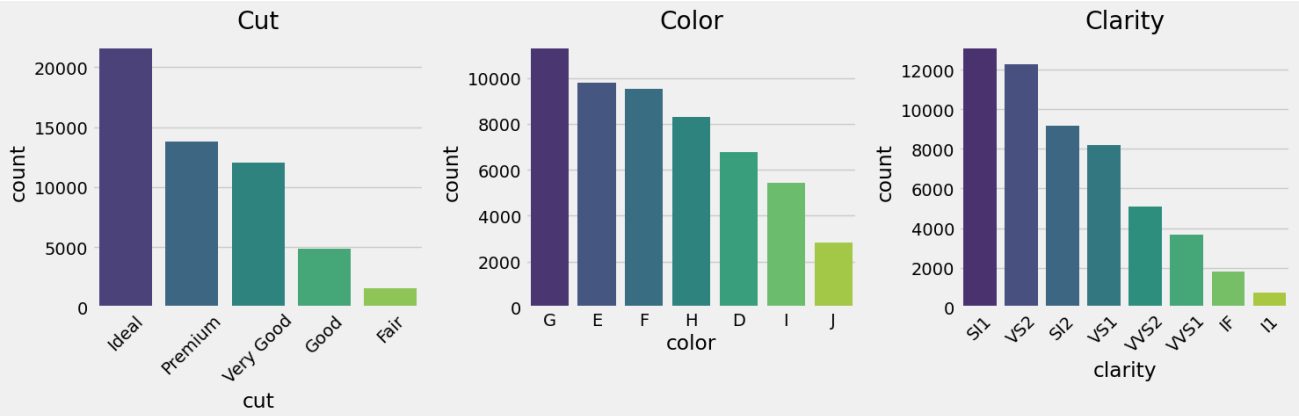


Statistic	Value
Count	53920
Min	0.2
Max	5.01
Mean	0.797698
Variance	0.224482
Skewness	1.11618
Kurtosis	1.25501

Same as the 'price', variable 'carat' also follows a fight skewed distribution with many diamonds having small carat values and few diamonds having large carat values

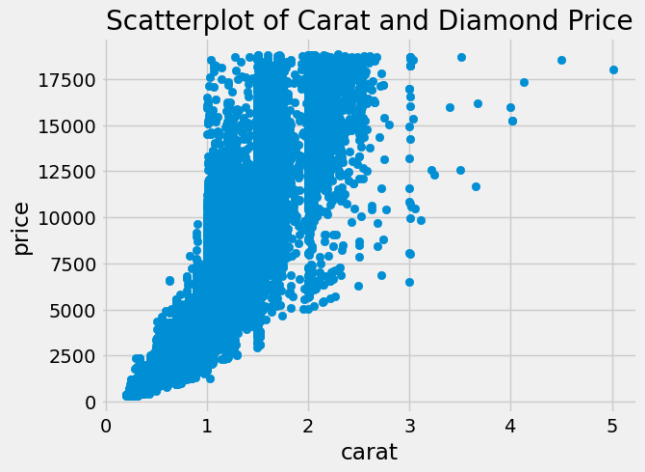


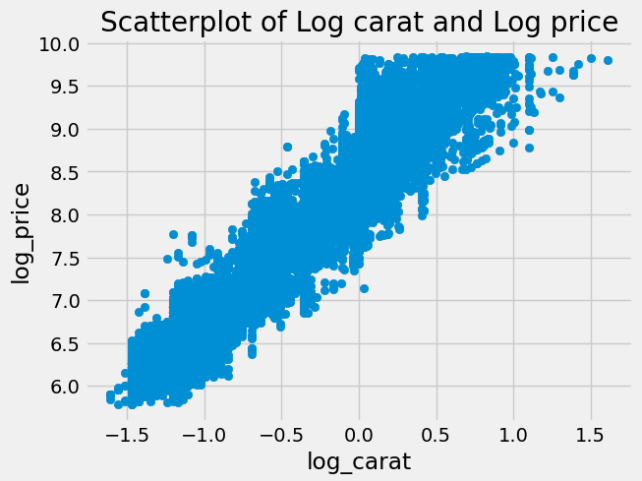
We can see that the variable 'length' of a diamond is fairly distributed but variables 'width' and 'depth' (y and z) have some extreme outliers. Therefore, special attention to be paid to this to understand what are the reasons for that. We will proceed to detect and remove those outliers at a later stage.



These are the count plots depicting the counts of categories of three categorical variables, 'cut', 'color' and 'clarity'.

Bivariate Analysis

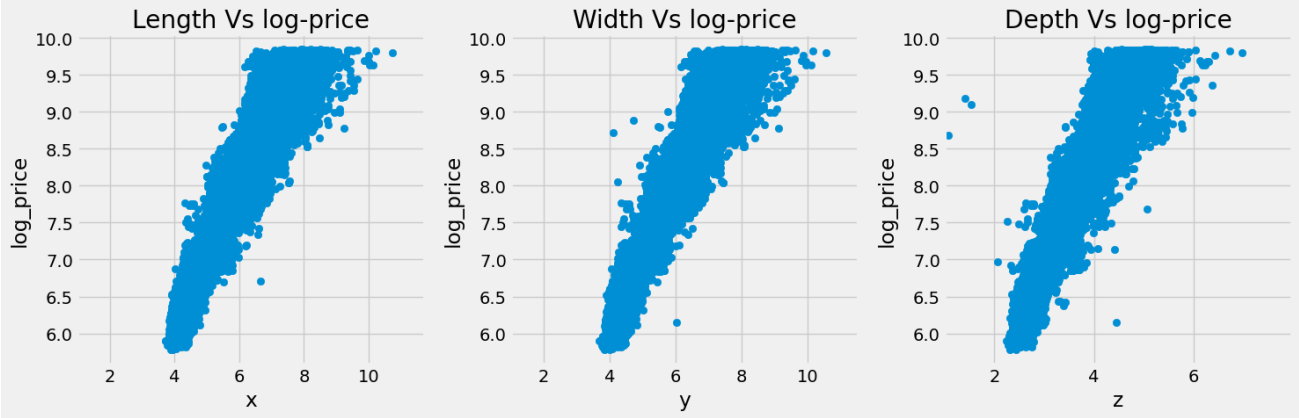




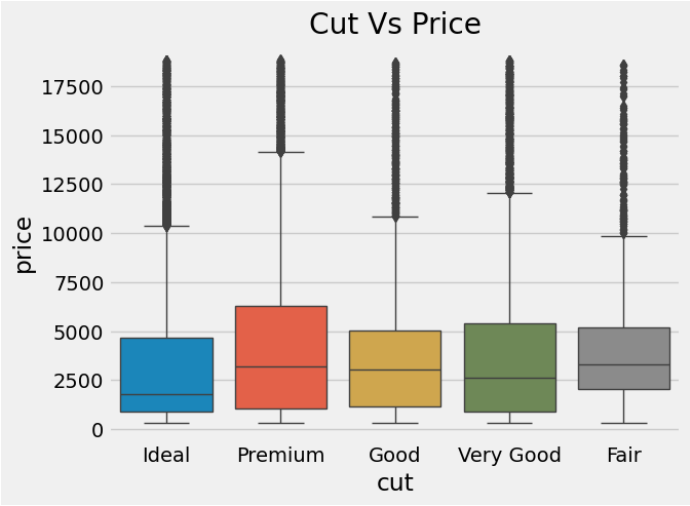
The relationship between carat value and the price is curvilinear, indicating that the rate of price increase was not constant.

Apply log transformation to both variables helps linearize the relationship

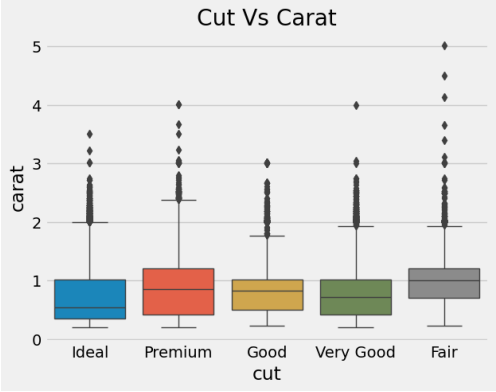
Similar to the previous relationships, these three relationships also exhibit a nonlinear (curvilinear) trend. However, it is evident that the price of a diamond increases as its length, width, and depth increase. This result is intuitive, as larger diamonds generally command higher prices due to their greater size and weight.



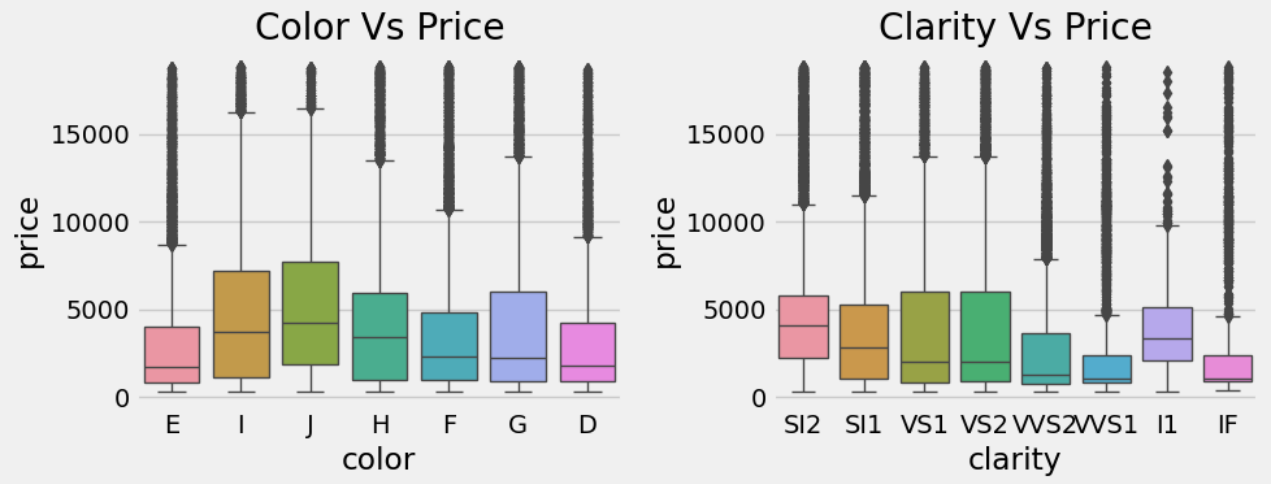
After applying log transformation to the price variable (since the variable 'price' is skewed) we can observe there is a positive linear relationship between the variables.



Here we observe the median price for the 'ideal' cut is lower than other types of cuts, which is not what we expected as the cut 'ideal' is the best cut. To identify the reason behind it we'll explore how 'cut' varies with other measurements such as carat value, length, width and depth of the diamond as these variables are highly related to the price of the diamond.



By analyzing these graphs, we observe that diamonds with higher median prices also tend to have larger measurements, including higher carat values, greater length, width, and depth. Notably, diamonds with an 'Ideal' cut are smaller in size compared to other cuts, which correlates with their lower median prices. This observation suggests that the physical measurements of a diamond play a more significant role in determining its price than the cut quality alone



Similar trends were observed for relationships between color and price, clarity and price. The best category might not show the highest price because the diamond's physical measurements play a greater role in price determination.

Outlier Detection and Removal

Since we found there are some extreme values present in the 'y' and 'z' variables in the univariate analysis, we will proceed to detect them and treat them accordingly.

	y	z
count	53920.000000	53920.000000
mean	5.734887	3.540046
std	1.140126	0.702530
min	3.680000	1.070000
25%	4.720000	2.910000
50%	5.710000	3.530000
75%	6.540000	4.040000
max	58.900000	31.800000

When inspecting the distribution of 'y' and 'z' variables, we can see their maximum values are far away from the 75% quantile. These can be valid extreme values or data errors.

```
df1[df1['y'] > 11]
```

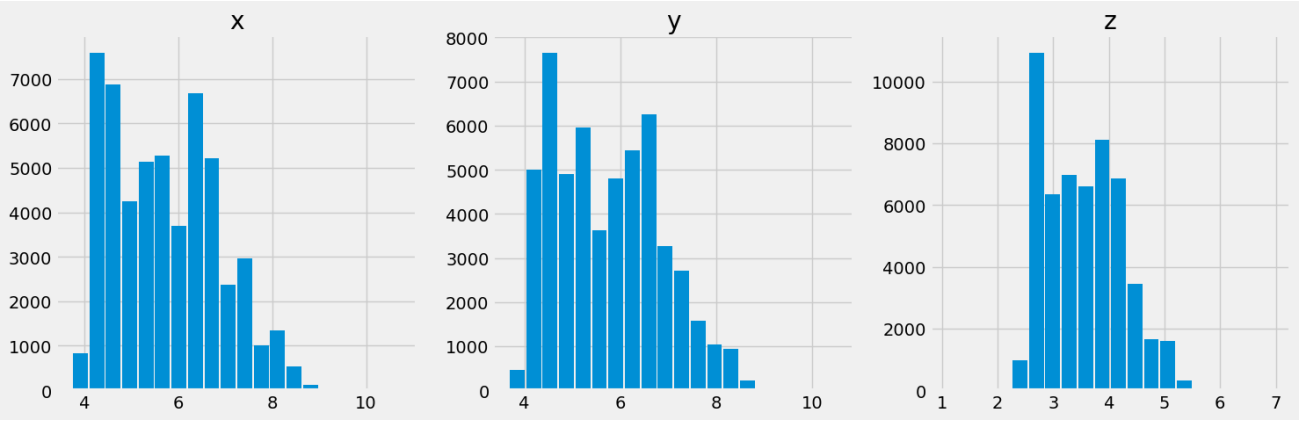
	carat	cut	color	clarity	depth	table	price	x	y	z
24067	2.00	Premium	H	SI2	58.9	57.0	12210	8.09	58.9	8.06
49189	0.51	Ideal	E	VS1	61.8	55.0	2075	5.15	31.8	5.12

Further inspecting the reason for that, we can detect there is a high chance these extreme 'y' values occur due to data errors because these measures are related to the width and depth of the diamonds. Therefore we drop those values.

```
df2[df2['z']>10]
```

	carat	cut	color	clarity	depth	table	price	x	y	z
48410	0.51	Very Good	E	VS1	61.8	54.7	1970	5.12	5.15	31.8

Similarly, we can see the extreme 'z' values also a data error. Therefore we proceed to remove it also.



After removing those outliers, the distribution of those variables became much more likely normal, suitable for using them to build a model.

Advanced Analysis

This section focuses on building a predictive model to predict the price of a diamond given its features. For that, we will train a multiple linear regression model as a baseline model and evaluate its performance using the r2 score, mean squared error and mean absolute error and find more accurate models.

Since we have categorical variables in the dataset, we have to encode them before feeding into the models. For that, the 'OrdinalEncoder' method in 'category_encoders' library will be used to encode the categorical variable because all of the categorical variables are in the ordinal level. (categories have an order)

After encoding the variables, we divide the dataset into training and testing sets such that 80% of the observations are for the training set and 20% of the observations are for the testing set. After forming the train set and test set, we first trained a multiple linear regression model on the train set and evaluated its performance on the testing set.

```
model = LinearRegression()
model.fit(x_train,y_train)
model.score(x_test,y_test)
```

```
0.9790900009174435
```

```
mean_absolute_error(y_test,model.predict(x_test))
```

```
0.1130833930285011
```

```
mean_squared_error(y_test,model.predict(x_test))
```

```
0.02143339807181714
```

We got a decent accuracy of 97.9% and very low MSE and MAE.

Fit a multiple linear regression model in R to get a detailed report. Here in the output, we can see the values of coefficients and their p-values as well.

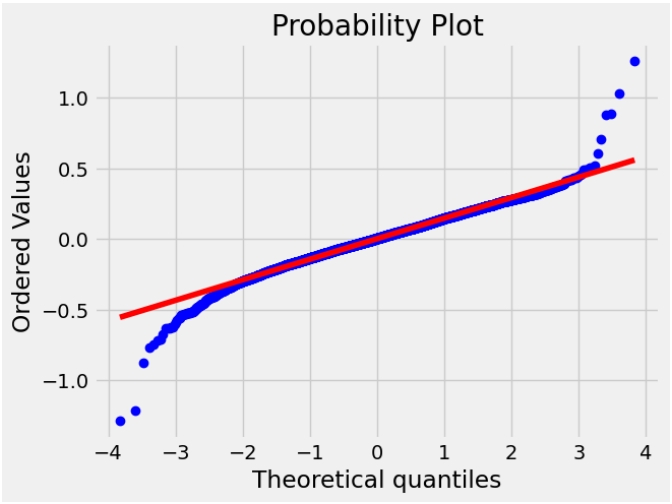
```
Call:
lm(formula = log(price) ~ ., data = diamond_cleaned)

Residuals:
     Min       1Q   Median       3Q      Max
-1.39193 -0.08617 -0.00160  0.08526  2.31765

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.8625602  0.0788905  -36.28   <2e-16 ***
carat        -1.0272041  0.0062254 -165.00   <2e-16 ***
cutGood       0.0752080  0.0040953   18.36   <2e-16 ***
cutIdeal      0.1464766  0.0040504   36.16   <2e-16 ***
cutPremium    0.1179890  0.0038702   30.49   <2e-16 ***
cutVery Good  0.1065421  0.0039601   26.90   <2e-16 ***
colorE       -0.0565412  0.0021437  -26.38   <2e-16 ***
colorF       -0.0958972  0.0021681  -44.23   <2e-16 ***
colorG       -0.1616678  0.0021229  -76.16   <2e-16 ***
colorH       -0.2562380  0.0022573 -113.52   <2e-16 ***
colorI       -0.3767812  0.0025360 -148.57   <2e-16 ***
colorJ       -0.5143004  0.0031312 -164.25   <2e-16 ***
clarityIF     1.0868623  0.0061363  177.12   <2e-16 ***
claritySI1    0.5793935  0.0052490  110.38   <2e-16 ***
claritySI2    0.4135487  0.0052680   78.50   <2e-16 ***
clarityVS1    0.7944075  0.0053589  148.24   <2e-16 ***
clarityVS2    0.7264188  0.0052743  137.73   <2e-16 ***
clarityVVS1   0.9952638  0.0056709  175.50   <2e-16 ***
clarityVVS2   0.9238927  0.0055158  167.50   <2e-16 ***
depth         0.0358341  0.0011505   31.15   <2e-16 ***
table         0.0097629  0.0003494   27.94   <2e-16 ***
x             0.7579837  0.0124156   61.05   <2e-16 ***
y             0.3527719  0.0126989   27.78   <2e-16 ***
z             0.4765600  0.0176582   26.99   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1354 on 53893 degrees of freedom
Multiple R-squared:  0.9822,    Adjusted R-squared:  0.9822
F-statistic: 1.293e+05 on 23 and 53893 DF,  p-value: < 2.2e-16
```

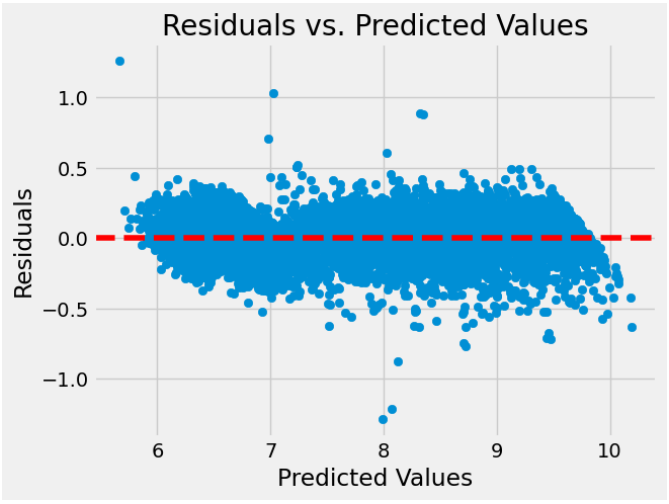
All the p-values are less than 0.05, hence all the variables are significant in predicting the diamond price.

Using above coefficients, we can write the regression model as,

log(price) = -2.86 – 1.03(carat) + 0.08(cut_good) + 0.15(cut_ideal) + 0.12(cut_premium) + 0.11(cut_very good) – 0.06(color_E) – 0.10(color_F) – 0.16(color_G) – 0.26(color_H) – 0.38(color_I) – 0.51(color_J) + 1.09(clarity_IF) + 0.58(clarity_SI1) + 0.41(clarity_SI2) + 0.79(clarity_VS1) + 0.73(clarity_VS2) + 1(clarity_VVS1) + 0.92(clarity_VVS2) + 0.04(depth) + 0.01(table) + 0.76(x) + 0.35(y) + 0.48(z)

If we are to check the assumptions of our linear regression model, we can do it using probability plot and fitted value Vs residual plots.





We can see the majority of data points lying closer to the straight line indicating that the normality of the residuals assumption is not violated. But the tails deviate from the straight line indicating that there are extreme values or outliers present

Residuals are randomly scattered around zero with a constant variance. Hence the homoscedasticity of residuals assumption is satisfied

However, we will explore some other models to check whether they perform better than the linear regression.

	model	scores	mean_score
0	Ridge()	[0.979, 0.978, 0.979, 0.979, 0.979]	0.9788
1	Lasso()	[0.15, 0.154, 0.149, 0.146, 0.151]	0.1500
2	RandomForestRegressor()	[0.992, 0.992, 0.992, 0.992, 0.992]	0.9920
3	SVR()	[0.977, 0.976, 0.978, 0.977, 0.977]	0.9770
4	KNeighborsRegressor()	[0.974, 0.972, 0.973, 0.973, 0.973]	0.9730
5	XGBRFRegressor(base_score=None, booster=None, ...	[0.978, 0.977, 0.977, 0.977, 0.978]	0.9774

Here we trained the dataset on Ridge regression, Lasso regression, Random forest regressor, Support vector regressor, K Neighbours regressor and Extreme Gradient boost regressor and evaluated its performance using cross-validation. We can see cross-validation results in the above table. We got an excellent accuracy of 99% from the random forest regressor model and other models performed similarly to the linear regression model. However, the Random forest regressor is like a 'black box' test as we cannot see its internal workings and we cannot interpret the model as well.

Since the model is being deployed, **linear regression** is preferred due to its simplicity**,** interpretability**,** and fast prediction speed. Therefore, we proceed with deploying the linear regression model.

Model Deployment

After selecting the best model, we deploy it so users can use it to predict diamond prices based on their input features in real-time. For that, we create a web application using Python flask. Python Flask is a lightweight web framework that provides useful tools and features that make creating web applications in Python easier. The web API was created as follows

- Created a flask server to predict diamond price based on its features using the trained linear regression model.
- Created a web page as a user interface to allow users to enter the input features
- Created a result page to display the predicted price based on given inputs

The created API works as follows.

- Users can input values into a form on the web page and click on the submit button to get the prediction.
- When a user clicks on the submit button, the provided values will be sent to the backend flask server.
- Then the server will calculate the predicted price using the trained linear regression model and send the predicted value to the result page.
- The predicted price will be displayed to the user on the result page.

But this is only accessible within the localhost of the PC where it is built. To make it accessible to anyone with the link, we created an **Amazon EC2 instance** and deployed the web API on it. Additionally, we installed **Nginx** on the EC2 instance to act as a **reverse proxy**, improving performance and handling incoming requests efficiently. We configured Nginx to route requests from the public URL to the locally running web application, ensuring seamless access to the prediction tool.

==========End of the Project==========