**Problem 1**: Linear Regression

**Data Dictionary:**

|  |  |
| --- | --- |
| **Variable Name** | **Description** |
| Carat | Carat weight of the cubic zirconia. |
| Cut | Describe the cut quality of the cubic zirconia. Quality is increasing order Fair, Good, Very Good, Premium, Ideal. |
| Colour | Colour of the cubic zirconia.With D being the best and J the worst. |
| Clarity | Clarity refers to the absence of the Inclusions and Blemishes. (In order from Best to Worst in terms of avg price) IF, VVS1, VVS2, VS1, VS2, Sl1, Sl2, l1 |
| Depth | The Height of cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter. |
| Table | The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter. |
| Price | the Price of the cubic zirconia. |
| X | Length of the cubic zirconia in mm. |
| Y | Width of the cubic zirconia in mm. |
| Z | Height of the cubic zirconia in mm. |

1. **The very first step of any data analysis assignment is to do the exploratory data analysis (EDA). Once you have understood the nature of all the variables, identified the response and the predictors, apply appropriate methods to determine whether there is any duplicate observation or missing data and whether the variables have symmetric or skewed distribution. Note that data may contain various types of attributes and numerical and/or visual data summarization techniques need to be appropriately decided. Both univariate and bivariate analyses and pre-processing of data are important. Check for outliers and comment on removing or keeping them while model building. Since this is a regression problem, the dependence of the response on the predictors needs to be thoroughly investigated.**

* **Lets check the shape of the data-**

(26967, 10)

We can see that our data set is having 26967 rows and 10 columns.

* **Lets check the information –**

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 26967 entries, 0 to 26966

Data columns (total 10 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 carat 26967 non-null float64

1 cut 26967 non-null object

2 color 26967 non-null object

3 clarity 26967 non-null object

4 depth 26270 non-null float64

5 table 26967 non-null float64

6 x 26967 non-null float64

7 y 26967 non-null float64

8 z 26967 non-null float64

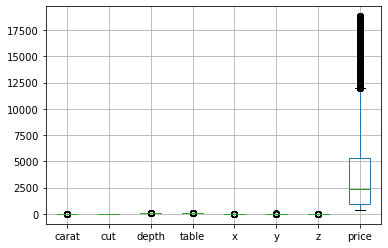
9 price 26967 non-null int64

dtypes: float64(6), int64(1), object(3)

memory usage: 2.1+ MB

We can conclude that total float64(6), int64(1), object(3) in our data set after droping 'Unnamed: 0' column.

* **Now lets check unique values in the object columns-**
* CUT : 5
* Ideal 10816
* Premium 6899
* Very Good 6030
* Good 2441
* Fair 781
* Name: cut, dtype: int64
* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* COLOR : 7
* G 5661
* E 4917
* F 4729
* H 4102
* D 3344
* I 2771
* J 1443
* Name: color, dtype: int64
* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* CLARITY : 8
* SI1 6571
* VS2 6099
* SI2 4575
* VS1 4093
* VVS2 2531
* VVS1 1839
* IF 894
* I1 365
* Name: clarity, dtype: int64
* **Let check outliers in our data set-**



We can conclude from the above graph that price has most number of outliers in the data.

* **Let’s check about Null values in the data-**

carat 0

cut 0

color 0

clarity 0

depth 697

table 0

x 0

y 0

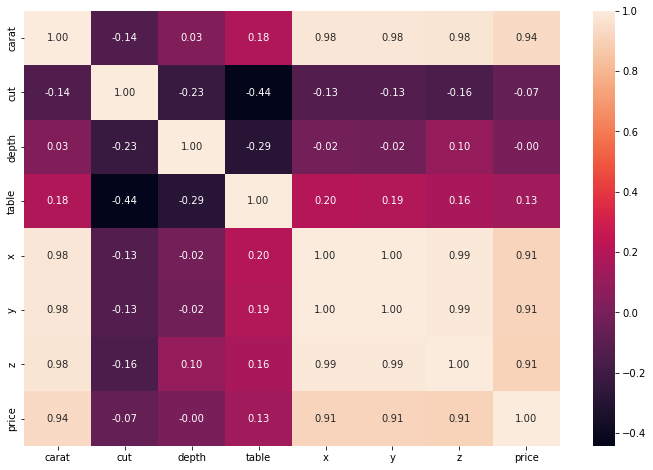
z 0

price 0

dtype: int64

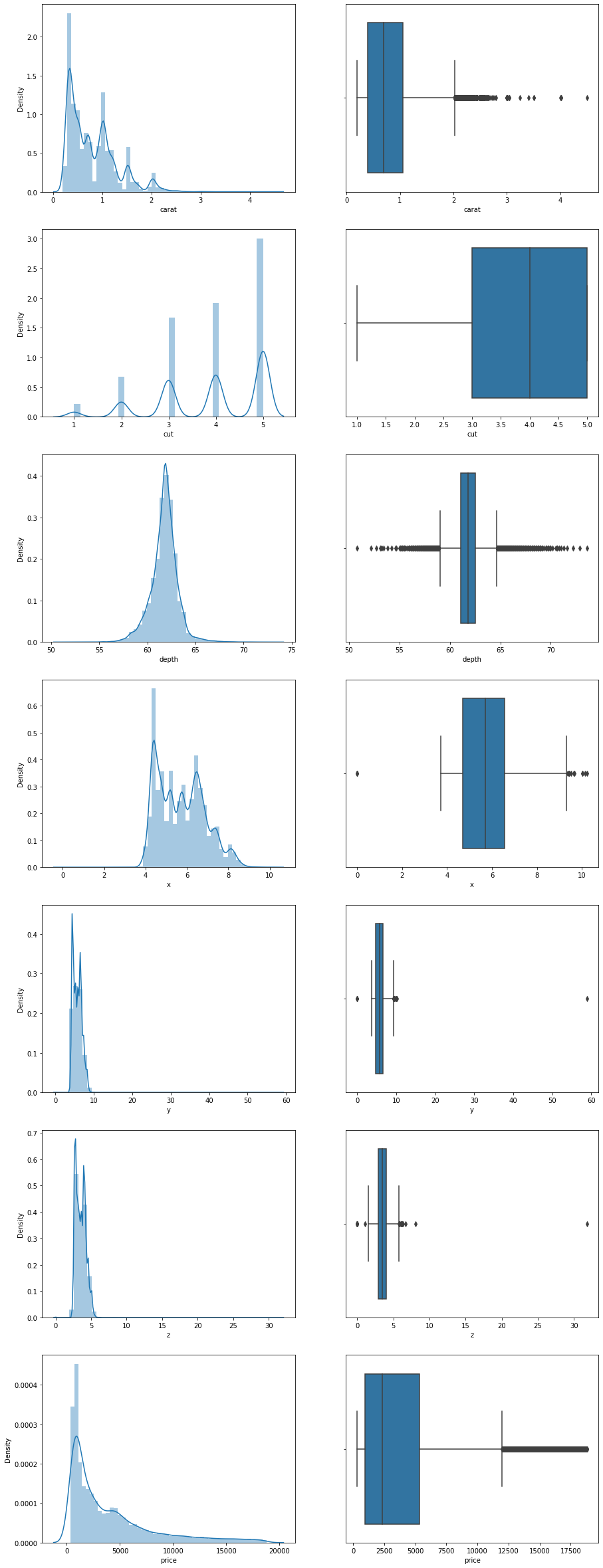
We can see that depth is having a 2.58464% null values.

* **Lets check correlation among the variables-**



We can observe from the above grapfh there is a high correlation exist among the variables.

* **Lets check about the distribution and outliars among all the variables-**



* **Data prepration for Model building-**

**Data transformation-**

We have 3 Object type columns in our data set which needs to be transformed into numerical values for Model building-

* Now lets check unique values in the object columns-
* CUT : 5
* Ideal 10816
* Premium 6899
* Very Good 6030
* Good 2441
* Fair 781
* Name: cut, dtype: int64
* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* COLOR : 7
* G 5661
* E 4917
* F 4729
* H 4102
* D 3344
* I 2771
* J 1443
* Name: color, dtype: int64
* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* CLARITY : 8
* SI1 6571
* VS2 6099
* SI2 4575
* VS1 4093
* VVS2 2531
* VVS1 1839
* IF 894
* I1 365
* Name: clarity, dtype: int64

As we can see cut is having the ordinal values so we will use 1,2,3,4,5 values to replace the ordinal values.

For color and clarity we will use Label Encoder method.

* For outlier treat meant we will use flooring and ceiling method to transform the outliers-
* Null values treatment in Depth column-

We will impute median value which is 61.80 to fill the nulls.

Now after transforming all the columns we can again look at our data information-

class 'pandas.core.frame.DataFrame'>

RangeIndex: 26967 entries, 0 to 26966

Data columns (total 10 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 carat 26967 non-null float64

1 cut 26967 non-null float64

2 color 26967 non-null int32

3 clarity 26967 non-null int32

4 depth 26967 non-null float64

5 table 26967 non-null float64

6 x 26967 non-null float64

7 y 26967 non-null float64

8 z 26967 non-null float64

9 price 26967 non-null float64

dtypes: float64(8), int32(2)

memory usage: 1.9 MB

1. **Use the Pre-processed Full Data to develop a model to identify significant predictors. Check whether the proposed model is free of multicollinearity. Apply variable selection method as required. Show all intermediate models leading to the final model. Justify your choice of the final model. Which are the significant predictors?**

**Lets build our first model with following variables-**

**1)- Model 1-** 'carat', 'cut', 'color', 'clarity', 'depth', 'table', 'x', 'y', 'z'

|  |  |  |  |
| --- | --- | --- | --- |
| **Dep. Variable:** | price | **R-squared:** | 0.911 |
| **Model:** | OLS | **Adj. R-squared:** | 0.911 |
| **Method:** | Least Squares | **F-statistic:** | 3.058e+04 |
| **Date:** | Sat, 24 Sep 2022 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 20:25:16 | **Log-Likelihood:** | -2.2552e+05 |
| **No. Observations:** | 26967 | **AIC:** | 4.511e+05 |
| **Df Residuals:** | 26957 | **BIC:** | 4.511e+05 |
| **Df Model:** | 9 |  |  |
| **Covariance Type:** | nonrobust |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **coef** | **std err** | **t** | **P>|t|** | **[0.025** | **0.975]** |
| **Intercept** | 3400.6825 | 668.944 | 5.084 | 0.000 | 2089.518 | 4711.847 |
| **carat** | 9109.5278 | 76.317 | 119.365 | 0.000 | 8959.943 | 9259.112 |
| **cut** | 142.2309 | 7.002 | 20.312 | 0.000 | 128.506 | 155.956 |
| **color** | -229.3457 | 3.884 | -59.048 | 0.000 | -236.959 | -221.733 |
| **clarity** | 248.9832 | 3.789 | 65.705 | 0.000 | 241.556 | 256.411 |
| **depth** | -40.7407 | 8.666 | -4.701 | 0.000 | -57.726 | -23.755 |
| **table** | -28.5761 | 3.620 | -7.893 | 0.000 | -35.672 | -21.480 |
| **x** | -2221.4816 | 109.948 | -20.205 | 0.000 | -2436.985 | -2005.978 |
| **y** | 1794.0650 | 108.640 | 16.514 | 0.000 | 1581.124 | 2007.006 |
| **z** | -339.6936 | 92.823 | -3.660 | 0.000 | -521.631 | -157.756 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Omnibus:** | 6873.925 | **Durbin-Watson:** | 2.011 |
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 34904.645 |
| **Skew:** | 1.142 | **Prob(JB):** | 0.00 |
| **Kurtosis:** | 8.084 | **Cond. No.** | 9.05e+03 |

**VIF score with all the variables-**

carat VIF = 332.37

cut VIF = 332.37

color VIF = 332.37

clarity VIF = 332.37

depth VIF = 332.37

table VIF = 332.37

x VIF = 332.37

y VIF = 332.37

z VIF = 332.37

**Model 2- 'carat', 'cut', 'color', 'clarity', 'depth', 'table', 'x', 'y'**

|  |  |
| --- | --- |
| **R-squared:** | 0.911 |
| **Adj. R-squared:** | 0.911 |

**Model 3- 'carat', 'cut', 'color', 'clarity', 'depth', 'table', 'x'**

|  |  |
| --- | --- |
| **R-squared:** | 0.910 |
| **Adj. R-squared:** | 0.910 |

Training Data RMSE of model\_base: 0.23

Test Data RMSE of model\_base: 0.23

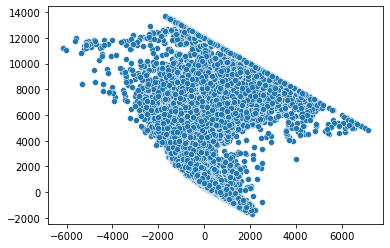
**Model 4- - 'carat', 'cut', 'color', 'clarity', 'depth', 'table'**

Training Data RMSE of model\_4: 0.35

Test Data RMSE of model\_4: 0.35

|  |  |
| --- | --- |
| **R-squared:** | 0.908 |
| **Adj. R-squared:** | 0.908 |

* Linear Relationship b/w Dependent and Independent Varaibles-



1. 3- Alternatively, if prediction accuracy of the price is the only objective, then you may want to divide the data into a training and a test set, chosen randomly, and use the training set to develop a model and test set to validate your model. Use the models developed in Part (2) to compare accuracy in training and test sets. Compare the final model of Part (2) and the proposed one in Part (3). Which model provides the most accurate prediction? If the model found in Part (2) is different from the proposed model in Part (3), give an explanation.

Lets split the data into train & test and find the RMSE score for mdels-

Model 1 RMSE score-

Which is having all the 9 independent variables 'carat', 'cut', 'color', 'clarity', 'depth', 'table', 'x', 'y', 'z'.

Training Data RMSE of model\_base: 0.22

Test Data RMSE of model\_base: 0.23

Model 4 RMSE score-

Which is having all the 6 independent variables 'carat', 'cut', 'color', 'clarity', 'depth', 'table'.

Training Data RMSE of model\_base: 0.35

Test Data RMSE of model\_base: 0.35

Model 3 RMSE score-

Which is having all the 7 independent variables 'carat', 'cut', 'color', 'clarity', 'depth', 'table',’x’.

Training Data RMSE of model\_base: 0.23

Test Data RMSE of model\_base: 0.23

Conclusion-

After comparing all the models Model 3 comes out to be the best model with following scores-

|  |  |
| --- | --- |
| **R-squared:** | 0.910 |
| **Adj. R-squared:** | 0.910 |

Training Data RMSE of model\_base: 0.23

Test Data RMSE of model\_base: 0.23