Contents

Contents 1

Problem Statement and Objective: 3

Understanding the dataset 3

1. Data Dictionary and data information: 3
2. EDA – Descriptive analysis. 4
   - 2.1. Datatypes: 4
   - 2.2. Data distribution: 5
   - 2.3. Data discrepancy: 6
   - 2.4. Missing data: 'depth' 8
   - 2.5. Skewness: 9
   - 2.6. Mode: All variables are unimodal 9
   - 2.7. Outliers: 10
3. EDA – Visual Analysis 11
   - 3.1. Univariate analysis of variable with distplot and boxplot: 11
   - 3.2. Bivariate analysis - object type variables 14
   - 3.3. Bivariate analysis – numeric variables 15
   - 3.4. Multivariate analysis 18
4. Data Pre-processing 19
   - 4.1. Missing Values Treatment: 19
   - 4.2. Outliers Treatment 20
5. Build various iterations of the Linear Regression model using appropriate variable selection techniques for the full data. 21

5.1. Assumptions of Linear Regression 21

5.2. Steps to Regression Model without train test split 22

5.3. Steps to Regression Model with train test split 25

5.4. Significant Predictors: 26

6. Evaluation Metrics Comparison: 27

6.1. Models Evaluation Metrics Statistics 27

6.2. RSquare 28

6.3. Mean Square Error (MSE) and Root Mean Square Error (RMSE) 28

6.4. Mean Absolute Error (MAE) 29

7. Assumptions of Linear Regression 30

7.1. Test of Assumption 3 of Linear Regression: The error terms have a constant variance i.e. they are homoscedastic in nature. 30

7.2. Test of Assumption 4: No auto-correlation between the error terms. (One value of the error term should not predict the next value of the error term) 32

7.3. Test of Assumption 5 of Linear Regression: The errors are assumed to be normally distributed 32

8. Conclusion 32

Problem Statement and Objective:

A company named Gem Stones Co Ltd, which is a cubic zirconia manufacturer. We are provided with the dataset containing the prices and other attributes of approximately 27,000 pieces of cubic zirconia (which is an inexpensive synthesized diamond alternative with similar qualities of a diamond).

The objective is to:

- *accurately predict prices* of the zircon pieces and,
- to understand the *predictors which are more important* in determining the price.

Understanding the dataset

1. Data Dictionary and data information:

Data has ***26967 rows and 11 columns*** as given below excluding an "Unnamed: 0" column:

Variable Name	Description	Variable Type	Variable role
Carat	Carat weight of the cubic zirconia.	Float	Predictor
Cut	Describes the cut quality of the cubic zirconia. Quality is in increasing order: Fair, Good, Very Good, Premium, Ideal.	Object	Predictor
Colour	Colour of the cubic zirconia.	Object	Predictor
Clarity	Cubic zirconia Clarity refers to the absence of the Inclusions and	Object	Predictor

	Blemishes. (In order from Best to Worst, FL = flawless, I3= level 3 inclusions) FL, IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1, I2, I3		
Depth	The Height of a cubic zirconia piece, measured from the Culet to the table, divided by its average Girdle Diameter.	Float	Predictor
Table	The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter.	float	Predictor
Price	Price of the cubic zirconia.	Integer	Dependent
X	Length of the cubic zirconia in mm.	Float	Predictor
Y	Width of the cubic zirconia in mm.	Float	Predictor
Z	Height of the cubic zirconia in mm.	Float	Predictor

2. EDA – Descriptive analysis.

2.1. Datatypes:

a. There are **6 float, 2 integer and 3 object** datatypes in the given dataset.

b. From describe() method, we may see that column *Unnamed: 0 seems to be the serial number* as it starts from 1 to 26967 without duplicates. We will *set this column as index* as it will not have any role in the data analysis. *This **will make effective columns as 10**.*

c. There are ***no duplicates*** in the dataset before removing the unnamed. After removing the 'Unnamed:0' there are 34 records with same features, but

since there is no unique identification Id and there could be more than one zirconia with same measures, we will not consider the records as duplicates. Hence, effective number of records remain 26967

d. Within object types, we have cut colour and clarity with following unique records. There seem to be ***no redundancy*** *in object categorization*.

```
Column cut has 5 unique values:
Ideal        10816
Premium       6899
Very Good     6030
Good          2441
Fair           781
Name: cut, dtype: int64
============================================================
Column color has 7 unique values:
G    5661
E    4917
F    4729
H    4102
D    3344
I    2771
J    1443
Name: color, dtype: int64
============================================================
Column clarity has 8 unique values:
SI1     6571
VS2     6099
SI2     4575
VS1     4093
VVS2    2531
VVS1    1839
IF       894
I1       365
Name: clarity, dtype: int64
============================================================
```

2.2. Data distribution:

Five-points summary: from describe() we may observe following

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Unnamed: 0	26967	NaN	NaN	NaN	13484	7784.85	1	6742.5	13484	20225.5	26967
carat	26967	NaN	NaN	NaN	0.798375	0.477745	0.2	0.4	0.7	1.05	4.5
cut	26967	5	Ideal	10816	NaN	NaN	NaN	NaN	NaN	NaN	NaN
color	26967	7	G	5661	NaN	NaN	NaN	NaN	NaN	NaN	NaN
clarity	26967	8	SI1	6571	NaN	NaN	NaN	NaN	NaN	NaN	NaN
depth	26270	NaN	NaN	NaN	61.7451	1.41286	50.8	61	61.8	62.5	73.6
table	26967	NaN	NaN	NaN	57.4561	2.23207	49	56	57	59	79
x	26967	NaN	NaN	NaN	5.72985	1.12852	0	4.71	5.69	6.55	10.23
y	26967	NaN	NaN	NaN	5.73357	1.16606	0	4.71	5.71	6.54	58.9
z	26967	NaN	NaN	NaN	3.53806	0.720624	0	2.9	3.52	4.04	31.8
price	26967	NaN	NaN	NaN	3939.52	4024.86	326	945	2375	5360	18818

- column *Unnamed: 0* could be a serial number and has no role in analysis.
- Minimum value for dimension variables *'x', 'y' or 'z' is zero* and must be inspected further.
- *Max value is very high* for a few variables e.g., 'y' (having 54.19) and 'z' (having 28.9)
- *Range for 'carat' (4.1) is high* with min value is 0.4 and max value is 4.5
- Price has a big range too but as we know the price is dependent variable and may rise with the quality of independent variables.

2.3. Data discrepancy:

2.3.1. Investigate where 'x' or 'y' or 'z'= 0

There are ***nine records*** where either x, y or z values are zero.

```
# as we had noticed x, y, z are the dimensions, they can not be zero for cz

df[(df['x']==0)|((df['y']==0) | (df['z']==0))]    # below 9 records seem to be wrong entries, we can drop them
```

Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
5822	0.71	Good	F	SI2	64.1	60.0	0.00	0.00	0.0	2130
6035	2.02	Premium	H	VS2	62.7	53.0	8.02	7.95	0.0	18207
6216	0.71	Good	F	SI2	64.1	60.0	0.00	0.00	0.0	2130
10828	2.20	Premium	H	SI1	61.2	59.0	8.42	8.37	0.0	17265
12499	2.18	Premium	H	SI2	59.4	61.0	8.49	8.45	0.0	12631
12690	1.10	Premium	G	SI2	63.0	59.0	6.50	6.47	0.0	3696
17507	1.14	Fair	G	VS1	57.5	67.0	0.00	0.00	0.0	6381
18195	1.01	Premium	H	I1	58.1	59.0	6.66	6.60	0.0	3167
23759	1.12	Premium	G	I1	60.4	59.0	6.71	6.67	0.0	2383

Since there could be no physical object with zero dimensions, it is necessarily an entry error. Also, there is no available features from which we can directly predict the length or breadth or height, so we should drop these 9 records. Hence, the ***effective number of records are 26958***.

2.3.2. Investigate 'y'.max() = 58.9 , 'z'.max()= 31.8

Before we analyze this, we can keep two facts in mind: First, we know that x,y,z are the dimensions and depict the size of the piece and when a piece size gets bigger it should be more or less on all axes. Secondly, in CZ industry ***a size 11mm is a one of the biggest pieces***.

```
df[(df['x'] == df['x'].max())|(df['y'] == df['y'].max())|(df['z'] == df['z'].max())]
```

Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
345	0.51	Very Good	E	VS1	NaN	54.7	5.12	5.15	31.80	1970
12501	4.50	Fair	J	I1	65.8	58.0	10.23	10.16	6.72	18531
25796	2.00	Premium	H	SI2	58.9	57.0	8.09	58.90	8.06	12210

In our case, we see that maximum of y (58.9mm) and maximum of z (31.8mm) are for different pieces and the other two dimensions in each piece are very small and well within industry range. Hence, we may drop

these two records as outliers. Hence, the ***effective number of records are 26956***.

After removing the two extreme values of 'y' and 'z', once again we check the max values of x, y and z.

```
# lets check again the maximum values of x, y and z

df[(df['x'] == df['x'].max())|(df['y'] == df['y'].max())|(df['z'] == df['z'].max())]
```

Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
12501	4.5	Fair	J	I1	65.8	58.0	10.23	10.16	6.72	18531

Now, we may notice that biggest size CZ is within range of 11mm dimension only and ***same piece has highest values for all three dimensions***.

2.3.3. Investigate 'carat' range which seems high at 4.1, with min value is 0.4 and max value is 4.5

From industry, minimum carat weight may start from 0.3 and maximum weight may go well beyond 100 and hence, there seems ***no discrepancy with the entries in carat*** and we must keep the values as such in data for carat.

2.4. Missing data: 'depth'

```
# check for null values per columns, column depth has 697 null values

df.isnull().sum()
```

```
carat        0
cut          0
color        0
clarity      0
depth      696
table        0
x            0
y            0
z            0
price        0
dtype: int64
```

There is only one variable ***'depth' column has 696 missing values***. From the data dictionary we may observe that *depth value depends upon height of culet from the table and girdle diameter*. Although, we do not have a direct calculation parameters but *we understand that diameter will depend on dimension of the piece i.e. x,y,z values*. Also, we have the *table value*, hence, we may use the available features to ***predict the depth using knn technique***.

Knn technique here is more appropriate as *it will take into account all the relevant measures (like, x,y,z, price, table) in 'k' similar records* in the data to predict the depth. Here, the optimal k value is taken as square root of 26956 (rounded to odd number) i.e., 165.

2.5. Skewness:

All variables are skewed to right except 'depth' which is negatively skewed.

```
carat    1.116485
depth   -0.027847
table    0.764272
x        0.402495
y        0.397798
z        0.371889
price    1.618694
```

- *Strongly skewed* – only 'carat' and 'price' are strongly skewed as their skewness value is greater than 1.
- *Moderately skewed* - 'table' is moderately skewed with skewness of 0.76.
- *Symmetric* – rest all variables are more or less symmetric having skewness value less than 0.5.

2.6. Mode: All variables are unimodal

Variable	Mode Value
carat	0.3
cut	Ideal
color	G
clarity	SI1
depth	62
table	56
x	4.38

```
y            4.35
z            2.69
price         544
```

2.7. Outliers:

We may observe that all columns have outliers. Since, price is response variable, we need to concern ourselves about the outliers in the predictors.

```
Calculating Number of Outliers in each column with the help of formula : (Q1-1.5*IQR) and (Q3+1.5*IQR)
price    1777
depth    1223
carat     660
table     317
z          12
x          12
y          11
dtype: int64
```

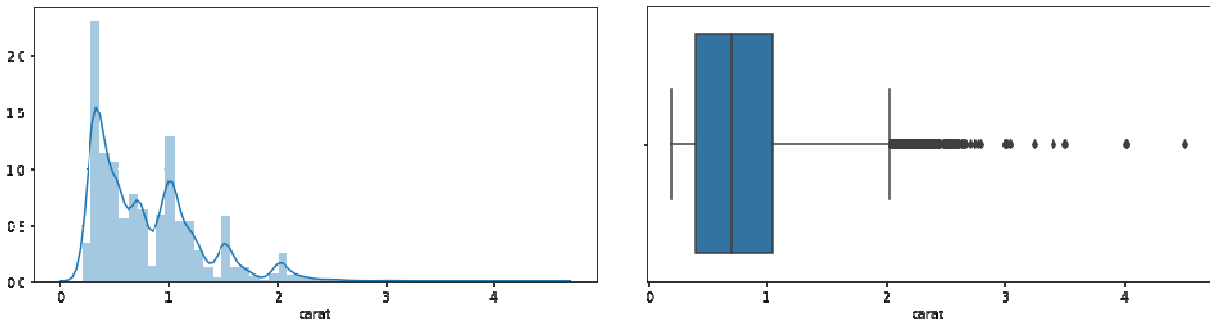
While analyzing the outliers, we need to keep in mind following points:

- *Whether the outlier is valid or a discrepancy* – this will be explored in our visual EDA in next section.
- If the outlier is valid, then *how much does it impact the whole data* – it can be noted that the *percentage of the outliers for all variables in overall data is very small.* The ***highest number of outliers is for depth which is 1223 which is less than 0.05% of the whole data***.

3. EDA – Visual Analysis

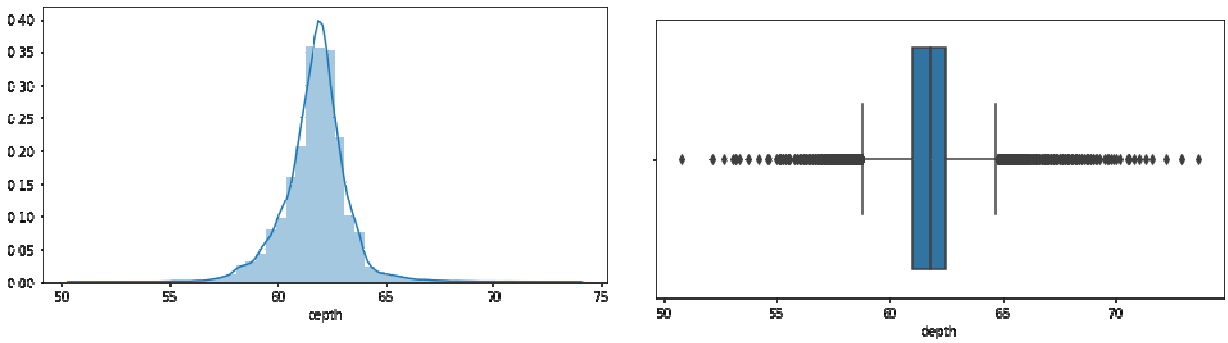
3.1. Univariate analysis of variable with distplot and boxplot:

3.1.1. Carat:



- From the graph it may seem that carat is multimodal but as calculated above, the modal value is single at 0.3.
- Distribution is skewed strongly to right.
- ***Although there are 660 outliers values but maximum value for carat is less than 4.5 which is well possible range as per industry. Hence, the outliers here are valid.***

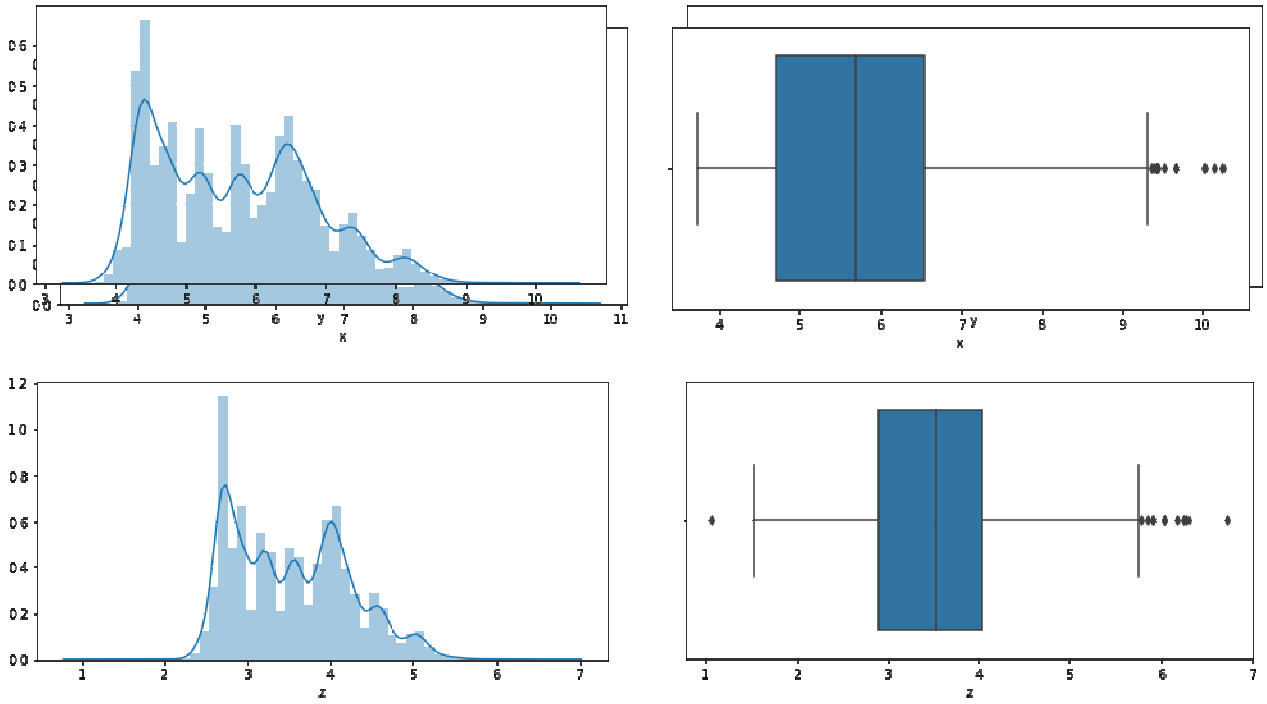
3.1.2. Depth:



- Distribution of depth is very close to normal. Modal value as we have seen is 62 which is close to both mean and median of 61.7.
- 'depth' has 1223 outliers in total on both sides and in a symmetrical manner.

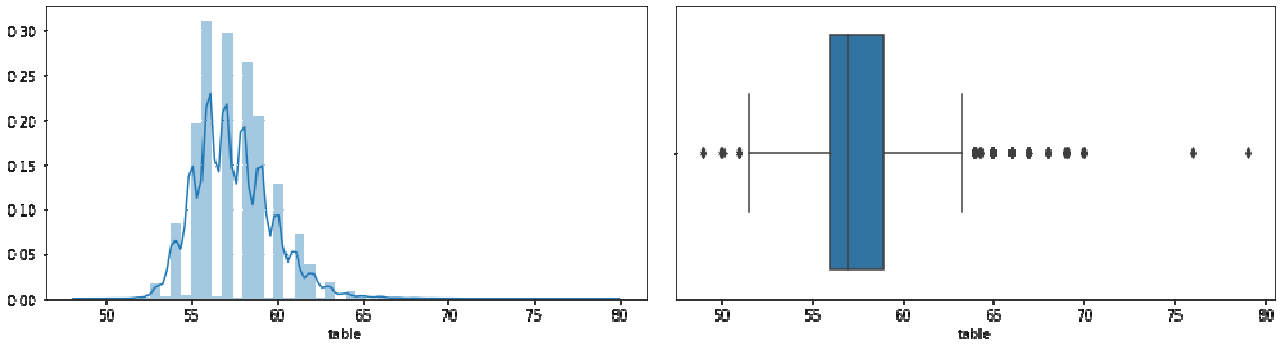
- The ‘depth’ depends on the other physical parameters of the piece and ***the range of depth is well within industry possible parameters i.e., 53 to 75, we may consider these value as valid entries.***

3.1.3. x,y,z



- All three dimensions, ’x’, ‘y’, ‘z’, are more or less symmetrically distributed.
- From histogram, it seems there are multiple peaks, but the mode value suggests only single mode for each variable with values x=4.38, y=4.35 and z=2.69
- All three have outliers on upper side except z, which has one outlier on lower side too.
- ***The maximum and minimum values are well in range as per possible industry size, hence these are valid outliers.***
- The number of outliers in case of all three dimensions is very few with x=y=12 and z=11. This makes sense as bigger pieces are rare.

3.1.4. Table:



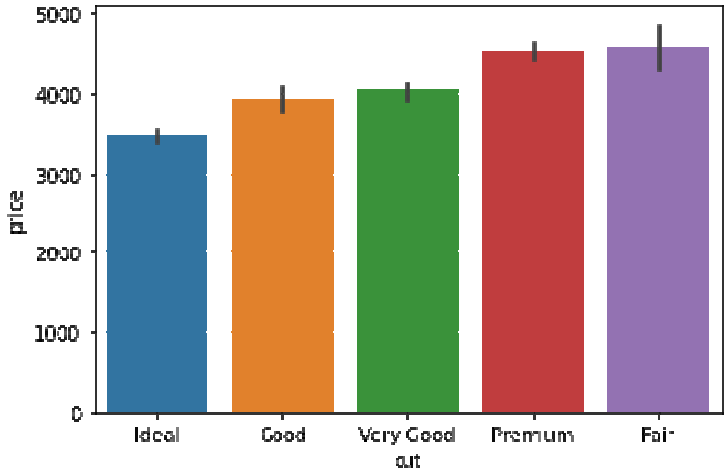
- From the graph it may seem that 'table' is multimodal but as calculated above, it is unimodal with value 56.
- There are a total of 317 outliers on both sides in 'table', but distribution is slightly skewed to right.
- Maximum for 'table' is 79 and minimum is 49 in our data but as we know that table is not a direct measurement but is derived as *percentage of average diameter* of the piece. ***Here, we cannot determine table straight away, but from our earlier analysis we have found that all outliers related to physical dimensions above are valid, hence logically, table value being a percentage of it should be valid too.***

3.2. Bivariate analysis - object type variables

Checking the dependence of response on various predictors.

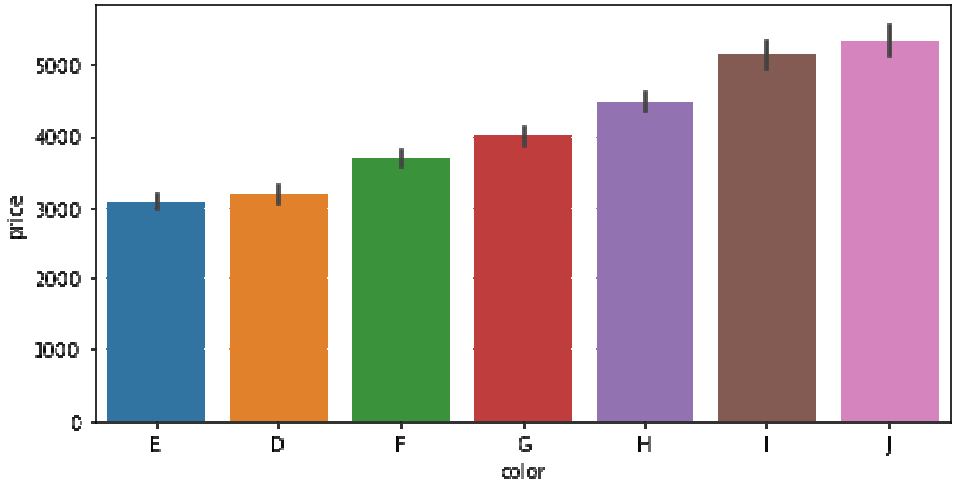
3.2.1. Cut

We may see that there is a distinguishing pattern in type of cut and price e.g., average pricing is highest for Fair, followed by Premium, Very Good, Good and Ideal in descending order.



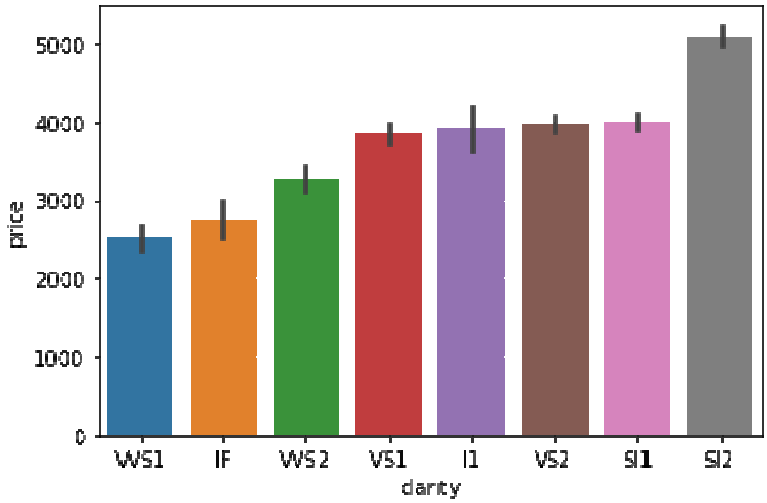
3.2.2. Color:

Color too shows a pattern with different color types e.g., color E has lowest average price followed by D , F, G, H, I and J being highest average price value



3.2.3. clarity

clarity feature also shows a distinguishing trend for different values from low to high. SI2 has highest pricing followed by SI1, VS2, I1, VS1, WS2, IF and WS1 in decreasing order.



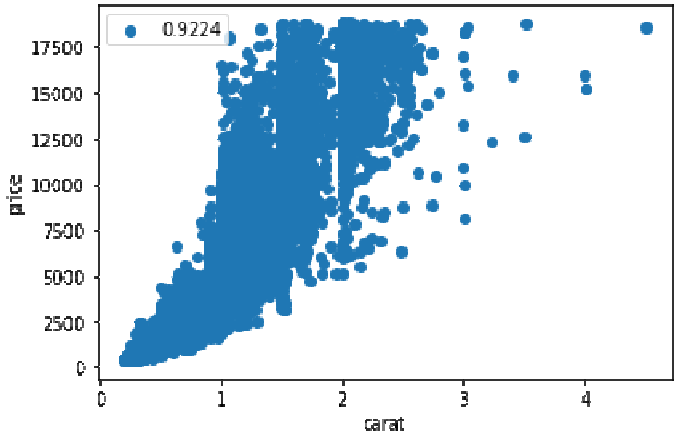
3.3. Bivariate analysis – numeric variables

Checking the dependence of response on various predictors.

3.3.1. Carat

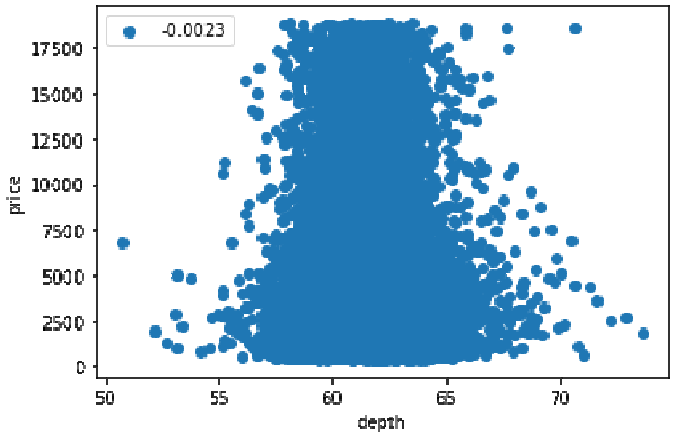
- Carat shows a strong linear relationship with price.

- Pearson's correlation coefficient here is .9224



3.3.2. Depth:

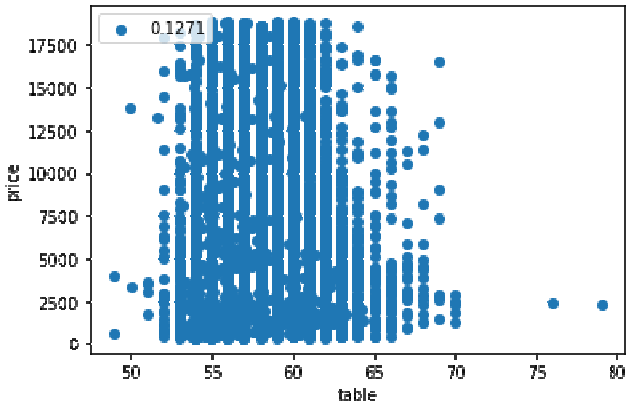
- We may observe the distribution of depth levels do not show linear trend but a distributed all over meaning a weak relationship with price
- Pearson's correlation coefficient is very low at -0.0023 which also suggest an inverse relationship between depth and price



3.3.3. table:

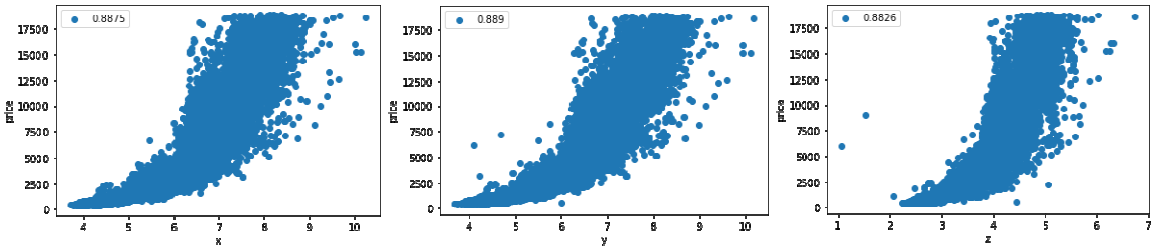
- From the plot table too show no particular trend with price variable. An increase or decrease in table value does not show a linear increase or decrease in price. Hence, we may suppose table to be not linearly related with price.

- Pearson's correlation coefficient is low at -0.1271 which confirms our observation from the plot.



3.3.4. x, y, z variables:

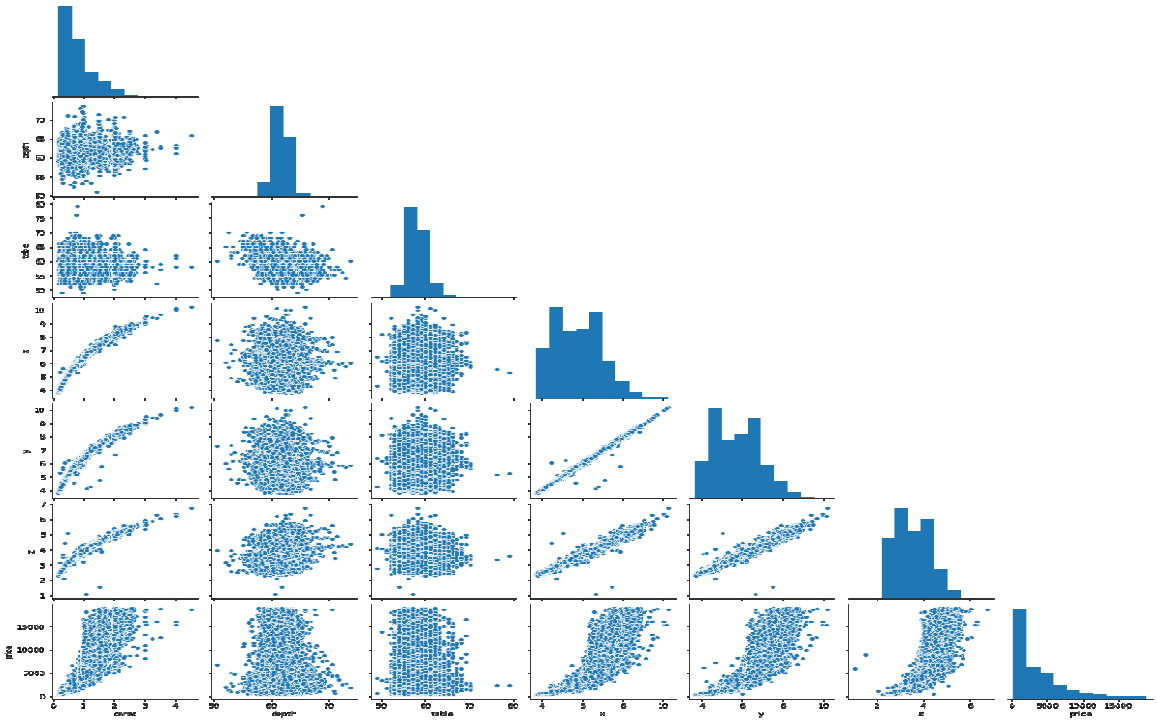
- Since x,y,z are all axes of the piece, we may look into all three together.
- From the three plots, we see linearity between x,y,z and price variable. Also, the pattern in all three plots is similar which indicates the strength of linear relationship should also be similar.
- From Pearson's correlation coefficient, our observation is confirmed as all three variables have strong and similar coefficient values i.e., x –.8875, y-.889, z-8826.



3.4. Multivariate analysis

3.4.1. Pair plot

From pair-plot we may check the linear relationship between the variables.



- As observed earlier, we may see that depth and table do not have a linear relationship with price or any other independent variables.
- Also, price has a good linear pattern with carat, x, y, z.
- We may also notice that carat and x, y, z too have strong linear relation with each other.

4. Data Pre-processing

4.1. Missing Values Treatment:

```
# check for null values per columns, column depth has 697 null values

df.isnull().sum()
```

```
carat        0
cut          0
color        0
clarity      0
depth      696
table        0
x            0
y            0
z            0
price        0
dtype: int64
```

- Only 'depth' column has 696 missing values. From the data dictionary we may observe that depth value depends upon height of culet from the table and girdle diameter. Although there is no direct calculation parameters for depth given, but we understand that diameter will depend on dimension of the piece i.e. x,y,z values and also the table value, hence, we may predict the depth from these available features.
- Here, we prefer to use Knn technique as it will take into account all the available measures (like, x,y,z, price, table) in dataset and predict using 'k' similar records with relevant features. For the optimal 'k' value, we have taken as square root of 26956 (rounded to odd number) i.e., 165.
- We may see the distribution of depth after and before knn is almost same.

```
print('\t\tbefore knn\tafter knn')
print('Mean\t\t{:.5f}\t{:.5f}'.format(df.depth.mean(),df_knn.depth.mean()))
print('Mode\t\t{}\t\t{}'.format(df.depth.mode()[0],df_knn.depth.mode()[0]))
print('Median\t\t{}\t\t{}'.format(df.depth.median(),df_knn.depth.median()))
```

```
                before knn      after knn
Mean            61.74545        61.74569
Mode            62.0            62.0
Median          61.8            61.8
```

4.2. Outliers Treatment

In point 4. Outliers, we discussed the basis of analyzing outliers on following two points:

4.2.1. Whether the outlier is valid or a discrepancy?

From our detailed analysis in boxplots, we have concluded that outliers in all the features are valid. The validation has been made on industry standards or logical interpretation of available data.

Since, all the outliers are valid, we must now look into the second condition

4.2.2. What is the impact of outliers on data?

If the outliers are valid and not significant in numbers, the effect of such outliers would be negligible and will not impact the slope too much. In present data, we can see the *percentage of the outliers for all variables in overall data is very small.* The ***highest number of outliers is for depth which is 1223 which is less than 0.05% of the whole data***. Hence, we keep the outliers as it is without treatment.

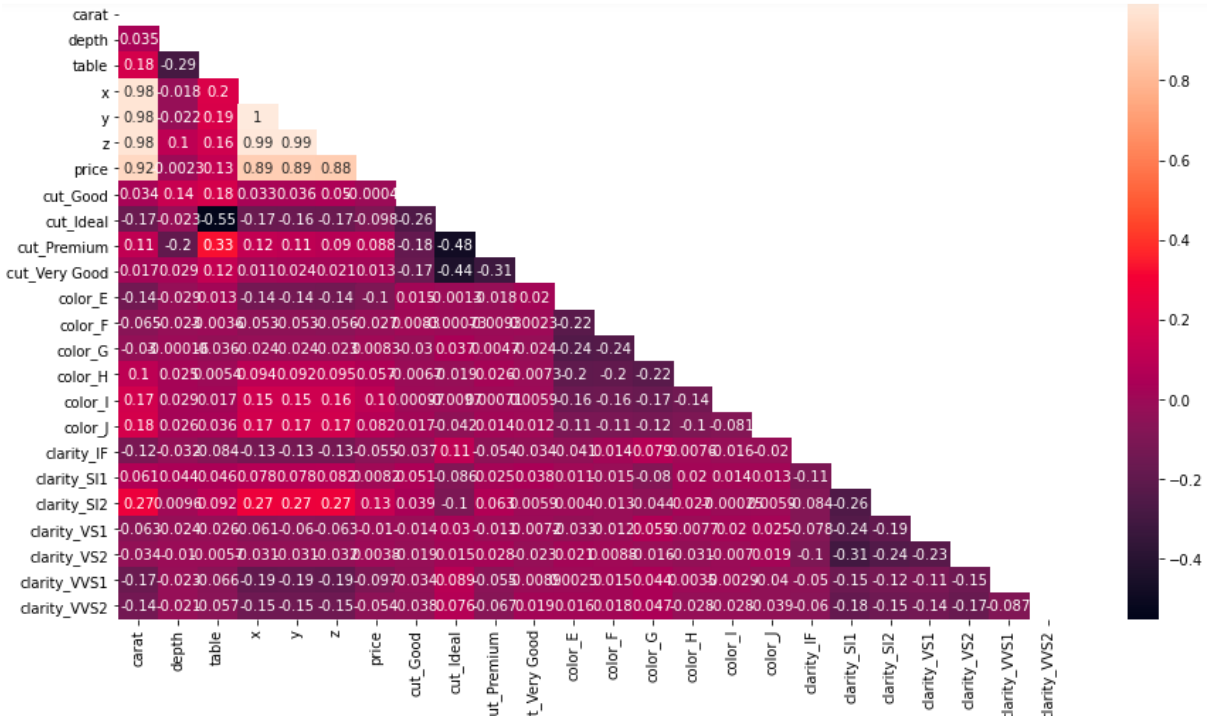
```
Calculating Number of Outliers in each column with the help of formula : (Q1-1.5*IQR) and (Q3+1.5*IQR)
price    1777
depth    1223
carat     660
table     317
z          12
x          12
y          11
dtype: int64
```

5. Build various iterations of the Linear Regression model using appropriate variable selection techniques for the full data.

5.1. Assumptions of Linear Regression

5.1.1. Test of Assumption 1 of Linear Regression: The independent variables should not be correlated.

- It is important for us to check now how variables are related to each other. For regression, one of the assumptions is that predictors should be independent of each other.
- Since correlation can only be calculated for numeric variable and we need to assess the relation between all predictors including object, we will convert the object variables into numeric type and then create the heat map or correlation metrics.



We may notice some important conclusions from above metrics:

- x and y are perfectly correlated to each other. In such case, we can use either x or y in the dataset as both will contribute in the same manner.
- Z has very strong correlation with x,y of about .99. Since, this value is also close to one to other axes, we may check their vif and keep only one out of three.
- Carat is strongly related to axes variables x,y,y with .98 value
- Rest all variables have very weak correlation with others. It is although good that predictors are slightly correlated to each other but on the other side, they are also not strongly correlated to response too.
- We may take the help of Variance Inflation factor and p-value in such cases to decide removal or keeping a variable in the model.

5.1.2. Test of Assumption 2 of Linear Regression: Linear Regression means that the dependent variable should be linearly related with the coefficients.

- We do not need to explicitly test this assumption as the Python code for Linear Regression makes sure that this particular assumption is met.
- Rest 3 assumptions are dependent on error terms which we can see after model building.

5.2. Steps to Regression Model without train test split

5.2.1. Standardize the data using zscore

```
from scipy.stats import zscore

df_scaled = stats.zscore(df_model)
df_scaled = pd.DataFrame(df_scaled, columns=['carat', 'depth', 'table', 'x', 'y', 'z', 'price', 'cut_Good',
       'cut_Ideal', 'cut_Premium', 'cut_Very Good', 'color_E', 'color_F',
       'color_G', 'color_H', 'color_I', 'color_J', 'clarity_IF', 'clarity_SI1',
       'clarity_SI2', 'clarity_VS1', 'clarity_VS2', 'clarity_VVS1',
       'clarity_VVS2'])
df_scaled.head()
```

	carat	depth	table	x	y	z	price	cut_Good	cut_Ideal	cut_Premium	...	color_H	color_I	color_J	clari
0	-1.043145	0.254143	0.244056	-1.295927	-1.289212	-1.260868	-0.854832	-0.315408	1.221570	-0.586089	...	-0.423355	-0.33849	-0.237822	-0.18
1	-0.980325	-0.678330	0.244056	-1.162787	-1.137212	-1.203427	-0.734279	-0.315408	-0.818619	1.706224	...	-0.423355	-0.33849	-0.237822	5.39
2	0.213263	0.325871	1.140433	0.275120	0.347024	0.347488	0.584356	-0.315408	-0.818619	-0.586089	...	-0.423355	-0.33849	-0.237822	-0.18
3	-0.791863	-0.104500	-0.652321	-0.807748	-0.833212	-0.830059	-0.709919	-0.315408	1.221570	-0.586089	...	-0.423355	-0.33849	-0.237822	-0.18
4	-1.022205	-0.965244	0.692245	-1.224919	-1.164036	-1.275228	-0.785234	-0.315408	1.221570	-0.586089	...	-0.423355	-0.33849	-0.237822	-0.18

5.2.2. Divide data into X and y

Here, df_model is our dataset after one-hot encoding and standardization.

X - all predictors

y – response variable price

```
# divide data into X and y

X = df_model[['carat', 'depth', 'table', 'x', 'y', 'z', 'cut_Good',
       'cut_Ideal', 'cut_Premium', 'cut_Very_Good', 'color_E', 'color_F',
       'color_G', 'color_H', 'color_I', 'color_J', 'clarity_IF', 'clarity_SI1',
       'clarity_SI2', 'clarity_VS1', 'clarity_VS2', 'clarity_VVS1',
       'clarity_VVS2']]
y = df_model.price
```

5.2.3. Run regression and VIF

- We have seen that ‘x’ and ‘y’ have a correlation of 1 and hencethey are perfectly correlated. In such case we may drop on of them on our own, however, for clarity sake we will run the VIF score for all and eliminate the variables one by one on the basis of VIF score, pvalues, importance of variables based on adjusted Rsquare and model performance Rsquare.
- Here, if the VIF values are close or seem important, we have interchangeably eliminated and tried the models to ensure which variable elimination gives better results. E.g., in last iterations, we have interchanged possible elimination for better results.

Iterations	Variables dropped	R Squared	Adjusted R Squared	Highest VIF values (first few close values)
Iteration 1	None	0.922	0.922	x,y,z
Iteration 2	x	0.921	0.921	z,y,depth
Iteration 3	x, z	0.921	0.921	table, y , depth
Iteration 4.a	X,z,y	0.917	0.917	Depth, table, cut_Ideal
Iteration 4.b	X,z,depth	0.920	0.920	Y, table,carat
Iteration 4.c	X, z, table	0.921	0.920	Y, depth, carat
Iteration 5.a	X,z,table,depth	0.920	0.920	Y,carat,claritySI1
Iteration 5.b	X,z,table,y	0.917	0.917	Depth,clariySI1,ClarityVS2
Iteration 6	'x','z','table','depth','y'	0.917	0.917	Cut_Ideal,Clarity_SI1,Cut_Premium, cut_Very_Good, clarity_VS2
Iteration 7.a	'x','z','table','y','cut_Ideal'	0.916	0.915	All under value 5
Iteration 7.b	'x','z','table','depth','y',cut_Premium	0.916	0.916	All under value 5
Iteration 7.c	'x','z','table','depth','y',cut_Very_Good	0.916	0.916	All under value 5
Iteration 7.d	'x','z','table','depth','y',clarity_VS2	0.903	0.903	All under value 5
Iteration 7.e	'x','z','table','depth','y',clarity_SI1	0.907	0.907	All under value 5
Iteration 7.f	x','z','table','depth','y',cut_Good	0.917	0.917	All under value 5

- After removing y in iteration 6, there are five variables with close and highest VIF values i.e., cut_Ideal, clarity_SI1, cut_Very_Good, clarity_VS2 and cut_Premium. Hence, instead of dropping by order of VIF , we have tried one over other and check the results. Best results are obtained by dropping 'cut_Premium' (iteration7.b) with R-square and Adj-RSquare of 0.916.
- As we can notice, cut is an important feature for the model effectiveness, we would like to try all the categories in cut to ensure only the least important variable is dropped from the model. We tried the model once again by dropping cut_Good in Iteration7.f at a R-square and Adjusted R-Square of 0.917.
- We may ***finalize the 7.f iteration as our final model***.

5.3. Steps to Regression Model with train test split

5.3.1. Split into train test

- Split the data into 30:70 ratio with train being 70% and test being 30% of the data.
- Here, X represent predictors and y represent response as we did earlier.

```
# Split X and y into training and test set in 75:25 ratio

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30 , random_state=1)
```

Iterations	Variables dropped	R Squared	Adjusted R Squared	Three highest VIF
Iteration 1	None	0.921	0.921	x,y,z
Iteration 2	x	0.921	0.921	z,y,depth
Iteration 3	x, z	0.920	0.920	table, y , depth
Iteration 4.a	X,z,y	0.916	0.916	Depth, table, clarity_SI1
Iteration 4.b	X,z,depth	0.920	0.920	Y, table,carat
Iteration 4.c	X, z, table	0.920	0.920	Y, depth, carat
Iteration 5.a	X,z,table,y	0.916	0.916	Depth,clariySI1,ClarityVS2
Iteration 5.b	X,z,table,depth	0.920	0.920	Y, carat, clarity_SI1
Iteration 6	'x','z','table','depth','y'	0.916	0.916	cut_Ideal', 'clarity_SI1' and 'cut_Premium', 'clarity_VS2', 'cut_Very_Good'
Iteration 7.a	'x','z','table','depth','y' and'cut_Ideal'	0.915	0.915	All under value 5
Iteration 7.b	x','z','table','depth','y', 'cut_Premium'	0.915	0.915	All under value 5
Iteration 7.c	x','z','table','depth','y', 'cut_Very_Good'	0.915	0.915	All under value 5
Iteration 7.d	'x','z','table','depth','y' and 'clarity_VS2'	0.902	0.902	All under value 5
Iteration 7.e	'x','z','table','depth','y' and 'clarity_SI2'	0.911	0.911	All under value 5
Iteration 7.f	'x','z','table','depth','y' and 'cut_Good'	0.916	0.916	All under value 5

- After removing y in iteration 6, there are six variables with highest and close VIF values i.e., cut_Ideal, clarity_SI1, cut_Premium, clarity_VS2 and

cut_Very_Good. So, we have tried the same iteration by dropping one at a time and measure model stats.

- As we can see the values have dropped by .01 on dropping cut_Ideal or cut_Premium (7.a and 7.b) but in case of dropping cut_Very_Good and carity_VS2 (7.c and 7.d), values of R-Square and Adjusted R-Square both dropped drastically.
- Since cut is an important feature for the model effectiveness, we would like to check what impact will dropping cut_Good do to the model, keeping others intact (iteration 7.f). Here, we observe the model effeiciency has improved by keeping others and only dropping cut_Good. Infact, the values for R-square and adjusted R-square remains same at 0.916 and all VIF values and pvalues are in expected limits.
- ***Hence, we may finalize the 7.f iteration as our final model***.

5.3.2. Check Model efficiency on train-test

For model robustness on train and test data, we have evaluated the models on different metrics like R-square, Adjusted R-square, MAE, RMSE, MSE in following section 6.

5.4. Significant Predictors:

- Below is the significant predictors according to the best models in two cases i.e., without split and with split.
- We may notice a great similarity between orders of predictors by importance. Also, the coefficient values are same in both cases upto 3-4 decimal values.

Predictor	Coefficient
1. carat	8425.976338
2. cut_Ideal	16.703568
3. cut_Premium	-111.215927
4. cut_Very_Good	-100.569709
5. color_E	-831.436647
6. color_F	-912.373501
7. color_G	-1064.392466
8. color_H	-1548.438197
9. color_I	-1966.175923
10. color_J	-2755.662204
11. clarity_IF	-253.638764
12. clarity_SI1	-2005.533995
13. clarity_SI2	-2867.837384
14. clarity_VS1	-1044.279681
15. clarity_VS2	-1354.189289
16. clarity_VVS1	-557.486261
17. clarity_VVS2	-655.263075
18. error	1.000000

Predictor	Coefficient
1. carat	8405.997461
2. cut_Ideal	-25.792392
3. cut_Premium	-184.379682
4. cut_Very_Good	-137.127340
5. color_E	-813.709819
6. color_F	-914.582507
7. color_G	-1052.468839
8. color_H	-1527.337041
9. color_I	-1955.300260
10. color_J	-2724.339358
11. clarity_IF	-152.786910
12. clarity_SI1	-1973.347071
13. clarity_SI2	-2817.579243
14. clarity_VS1	-1006.114441
15. clarity_VS2	-1304.496109
16. clarity_VVS1	-508.530698
17. clarity_VVS2	-623.525204

- Important predictors' list has same order by value of coefficient in both models. There are 17 important predictors in each model. The coefficient values are different but not by too large.
- Most important feature in both models is carat and it is directly related to profitability.
- Coefficient value of carat is not only highest, but the magnitude (absolute) value is way higher than any other predictors.

6. Evaluation Metrics Comparison:

6.1. Models Evaluation Metrics Statistics

Best models from above two cases are summarized as below:

Evaluation Parameter	Without split (for complete data)	With split (for train data)	With split (for test data)
R-Square	0.917	0.916	
Adjusted R-Square	0.917	0.916	

Mean Absolute Error	809.8819991538392	809.8819991538392	808.7043959545088
Mean Squared Error	1358545.7379267507	1345151.2424391385	1362951.745889934
Root Mean Squared Error	1167.4552436346046	1159.8065538869569	1167.4552436346046

6.2. RSquare

R Square is the measure of variability in dependent variable that can be explained by the model.

Here, the value of R-square is 0.917 without split and 0.916 after spilt which means without split best model can explain 91.7% variability and with split best model can explain 91.6% variability in predictors which is a very small difference to judge the better out of two.

6.3. Mean Square Error (MSE) and Root Mean Square Error (RMSE)

MSE is the sum of square of prediction errors (y-yhat) divided by the number of data points. It gives an absolute figure about the degree of deviation of prediction from actuals. A high MSE means bad model.

Here, the ***MSE for without split is 1358545.7379267507*** and ***MSE for train data in split model is 1345151.2424391385*** and MSE for ***test data in split model is 1362951.745889934.***

Here, the value of mean square error is least for model with split on train data followed by model without split which indicates errors are higher in first model without split as compared to second with split. On comparing test and train data errors for split model, train is slightly lower than train which is obvious for the model.

However, since the MSE values are squared values and the difference between three values is not large enough, it is recommended to compare the results by RMSE values too.

RMSE is the square root of MSE. The use of RMSE is more common than MSE because when MSE is too, it is difficult to compare. Square root of MSE brings it comparable to the same level as of prediction error and makes it easier for interpretation.

Similarly, ***RMSE for without split is 1167.4552436346046*** *and* MSE for ***train data in split model is 1159.8065538869569*** and MSE for ***test data in split model is 1167.4552436346046.*** Here, the least errors are given by the model with split for train data followed by model without split and test dat. Hence, we can consider, second model to be better here. Although, we notice that the difference in error values is not very large for three data (total, train and test).

6.4. Mean Absolute Error (MAE)

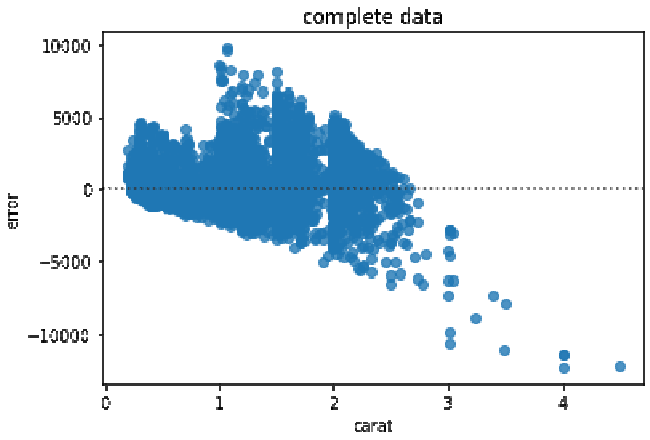
MAE is similar to MSE, only that MAE is takes the sum of absolute value of error. Compare to MSE or RMSE, MAE is a more direct representation of sum of error terms as it is absolute value. MSE penalizes errors by larger scale than MAE but it is easy to interpret MAE from the original dataset than MSE.

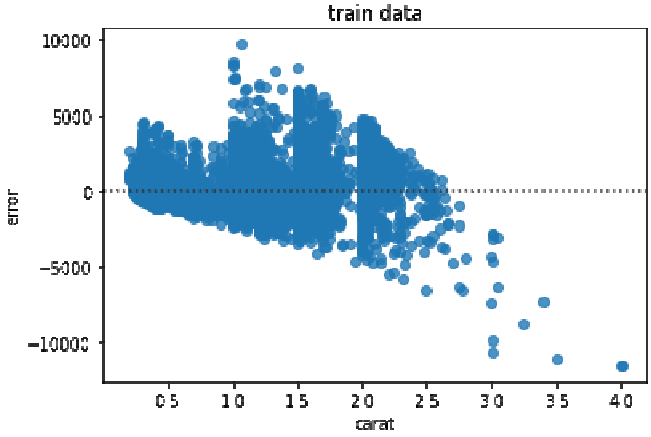
In our case, MAE for ***without split is 809.8819991538392*** and MAE for ***train data in split model is 809.8819991538392*** and ***MAE for test data in split model is 808.7043959545088.*** As we understand that here the errors in prediction are measure at absolute scale we can notice that model without split has least errors of approx. 810 and train at 810 and test at 809. We may say that from this observation too, second model is better as the test is only slightly lower than train data. Also, the model without split is giving same MAE as train dataset for split model.

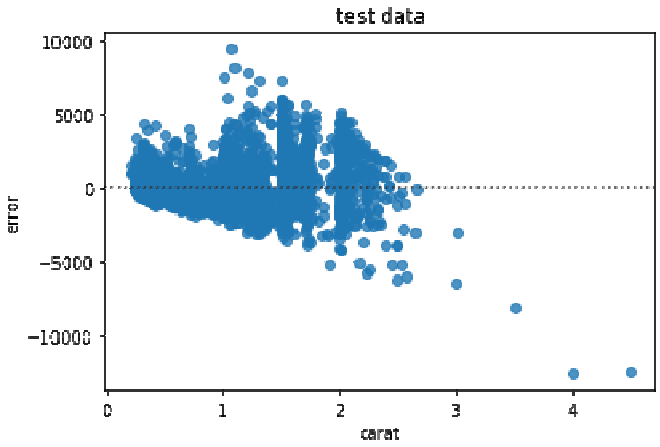
7. Assumptions of Linear Regression

7.1. Test of Assumption 3 of Linear Regression: The error terms have a constant variance i.e. they are homoscedastic in nature.

- For illustration we have shown the residual plot between carat and residuals: As may be noticed that the residuals and variable carat does not have a particular pattern which indicates residuals / errors are homoscedastic in nature. The same is true for other variables plots too.






test data
10000
5000
0
−5000
−10000
error
0
1
2
3
4
carat

7.2. Test of Assumption 4: No auto-correlation between the error terms. (One value of the error term should not predict the next value of the error term)

```
import statsmodels
print('Result of Durbin Watson test on Complete data',statsmodels.stats.stattools.durbin_watson(y_res, axis=0))
print('Result of Durbin Watson test on Train data',statsmodels.stats.stattools.durbin_watson(X_train['error'], axis=0))
print('Result of Durbin Watson test on Test data',statsmodels.stats.stattools.durbin_watson(X_test['error'], axis=0))
```

```
Result of Durbin Watson test on Complete data 2.0030030994782697
Result of Durbin Watson test on Train data 1.997084223263287
Result of Durbin Watson test on Test data 2.025419247034357
```

Here, we see that the Durbin-Watson test statistic is close to 2 and thus we can say there is no autocorrelation

- As we may notice, the value of Durbin Watson test for all three cases is similar and clos to 2 or more. Hence, this assumption stands true for the model.

7.3. Test of Assumption 5 of Linear Regression: The errors are assumed to be normally distributed

```
from scipy.stats import shapiro
print('Shapiro test result for complete data',shapiro(np.abs(y_res)))
print('Shapiro test result for train data',shapiro(np.abs(X_train['error'])))
print('Shapiro test result for test data',shapiro(np.abs(X_test['error'])))
```

```
Shapiro test result for complete data ShapiroResult(statistic=0.7257522344589233, pvalue=0.0)
Shapiro test result for train data ShapiroResult(statistic=0.7267448306083679, pvalue=0.0)
Shapiro test result for test data ShapiroResult(statistic=0.7173851132392883, pvalue=0.0)
```

- As may be observed that the p-value in all three cases is less than .05, which suggest Null Hypothesis is rejected meaning errors are not normally distributed. Hence, this assumption does not holds true.

8. Conclusion

1. We may consider the second model i.e., with split as the better model for following reasons:
2. The R-square values for the model is very slightly less than the wihout split model i.e., by 0.1%

3. All the evaluation metrics like MSE, RMSE and MAE for with split model in train data are better than without split model
4. Comparing the train and test data, test data predictions are close to train prediction and are only slightly lower which is ideally the case for robust models.
5. Assumptions of Linear Regression have very similar outputs on both the models, hence, like R-square they are in acceptable limits for both.
6. Important predictors' list has same order with 18 important predictors and has either equal or very close coefficient values for all features.
7. Most important feature in both models is carat and it is directly related to profitability.
8. Coefficient value of carat is not only highest, but the magnitude (absolute) value is way higher than any other predictors.