PREDICTIVE MODELLING PROJECT

Created by-

SHUBHASREE SARKAR

INDEX

Topic – **Cubic_Zirconia**

List of Figures

List of Tables

Topic – Holiday Package

List of Figures

List of Tables

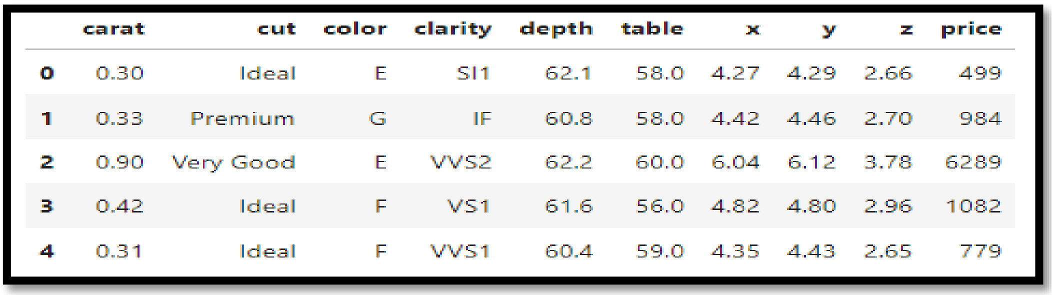
Topic: Cubic_Zirconia

You are hired by a company Gem Stones co ltd, which is a cubic zirconia manufacturer. You are provided with the dataset containing the prices and other attributes of almost 27,000 cubic zirconia (which is an inexpensive diamond alternative with many of the same qualities as a diamond). The company is earning different profits on different prize slots. You have to help the company in predicting the price for the stone on the bases of the details given in the dataset so it can distinguish between higher profitable stones and lower profitable stones so as to have better profit share. Also, provide them with the best 5 attributes that are most important.

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.
Color	Colour of the cubic zirconia. With D being the worst and J the best.
Clarity	Clarity refers to the absence of the Inclusions and Blemishes. (In order from Worst to Best in terms of Average 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.

1.1. Read the data and do exploratory data analysis. Describe the data briefly. (Check the null values, Data types, shape, EDA, duplicate values). Perform Univariate and Bivariate Analysis.

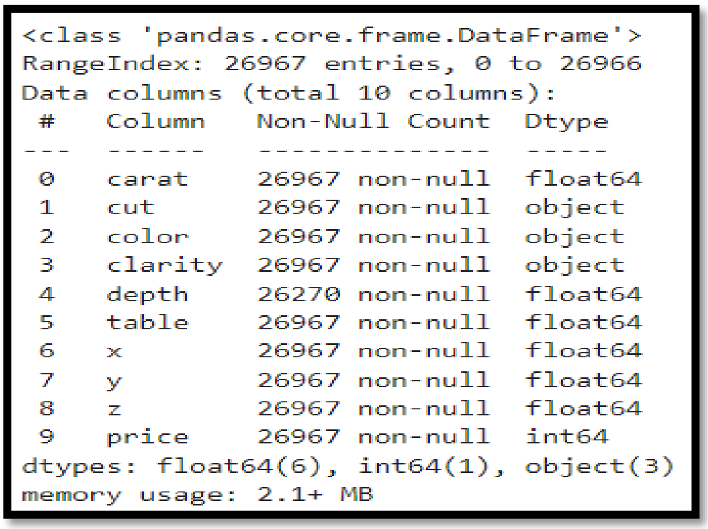
#Check for Head/Sample of the dataset – Table 1



	carat	cut	color	clarity	depth	table	x	y	z	price
0	0.30	Ideal	E	SI1	62.1	58.0	4.27	4.29	2.66	499
1	0.33	Premium	G	IF	60.8	58.0	4.42	4.46	2.70	984
2	0.90	Very Good	E	VVS2	62.2	60.0	6.04	6.12	3.78	6289
3	0.42	Ideal	F	VS1	61.6	56.0	4.82	4.80	2.96	1082
4	0.31	Ideal	F	VVS1	60.4	59.0	4.35	4.43	2.65	779

The above figure shows the output of the head function, showing top 5 records with the total of 10 variables or attributes

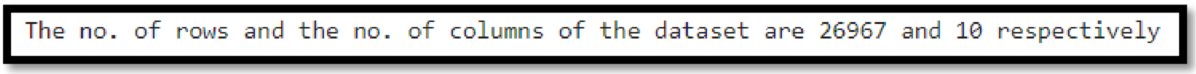
#Check for info – Table 2



```
<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
```

The above figure shows the different data types using the info function and here in this case we only have three data types, i.e float64(6 variables), int64(1 variables) and object(3 variables)

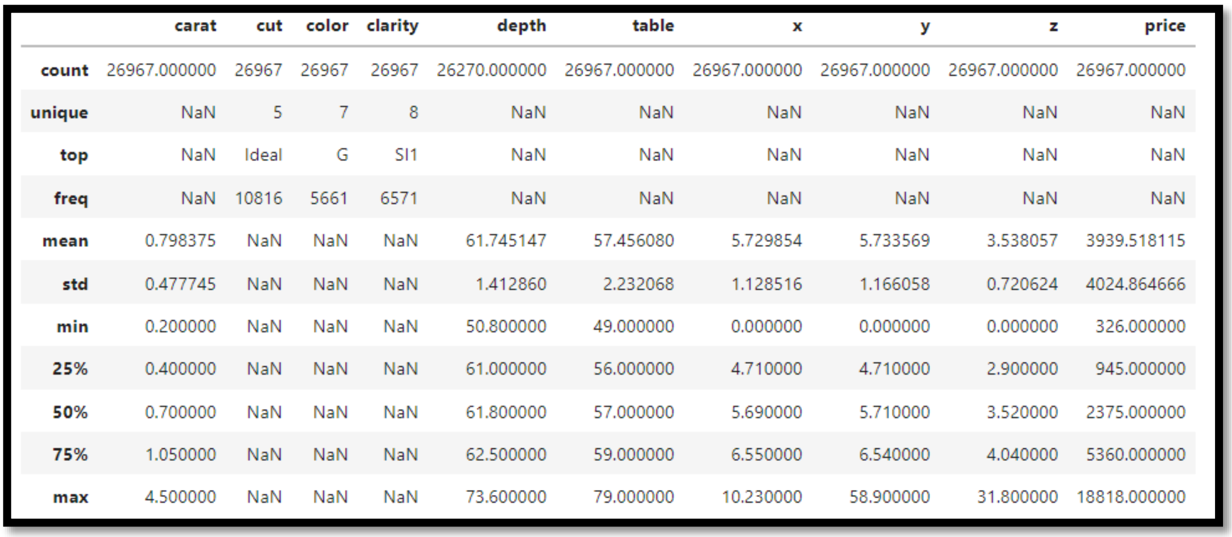
#Check the shape of the dataset



```
The no. of rows and the no. of columns of the dataset are 26967 and 10 respectively
```

The dimensions of the whole dataset are shown in the above figure

Check for Summary stats- Table 3



	carat	cut	color	clarity	depth	table	x	y	z	price
count	26967.000000	26967	26967	26967	26270.000000	26967.000000	26967.000000	26967.000000	26967.000000	26967.000000
unique	NaN	5	7	8	NaN	NaN	NaN	NaN	NaN	NaN
top	NaN	Ideal	G	SI1	NaN	NaN	NaN	NaN	NaN	NaN
freq	NaN	10816	5661	6571	NaN	NaN	NaN	NaN	NaN	NaN
mean	0.798375	NaN	NaN	NaN	61.745147	57.456080	5.729854	5.733569	3.538057	3939.518115
std	0.477745	NaN	NaN	NaN	1.412860	2.232068	1.128516	1.166058	0.720624	4024.864666
min	0.200000	NaN	NaN	NaN	50.800000	49.000000	0.000000	0.000000	0.000000	326.000000
25%	0.400000	NaN	NaN	NaN	61.000000	56.000000	4.710000	4.710000	2.900000	945.000000
50%	0.700000	NaN	NaN	NaN	61.800000	57.000000	5.690000	5.710000	3.520000	2375.000000
75%	1.050000	NaN	NaN	NaN	62.500000	59.000000	6.550000	6.540000	4.040000	5360.000000
max	4.500000	NaN	NaN	NaN	73.600000	79.000000	10.230000	58.900000	31.800000	18818.000000

The above figure shows the statistical measures associated with the continuous variables.

The description of the same is shown below-

Carat

Here Carat is actually *Carat weight of the cubic zirconia.*
Average Carat weight is **0.798375 units** with **standard deviation** of **0.477745 units**
The Carat weight range is between **0.20 units (Minimum)** and **4.50 units (Maximum)**
The **median** of the Carat weight is **0.70 units**
Total count is **26967.**

Depth

Here, the Depth is described as *the Height of cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter*.
Average Depth is **61.745147 units** with **standard deviation** of **1.412860 units**
The Depth range is between **50.80 units (Minimum)** and **73.60 units (Maximum)**
The **median** of the Depth is **61.80 units**
Total count is **26270**

Table

Here, the Table is described as *the Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter.*
Average Table is **57.456080 units** with **standard deviation** of **2.232068 units**
The **Table** range is between **49.00 units (Minimum)** and **79.00 units (Maximum)**
The **median** of the **Table** is **57.00 units**
Total count is **26967**

X

Here X is basically *the length of the cubic zirconia in mm.*
Average X is 5.729854 units with **standard deviation** of **1.128516 units**
The X range is between **0.00 units (Minimum)** and **10.23 units (Maximum)**
The **median** of the X is **5.69 units**
Total count is **26967**

Y

Here Y is basically *the Width of the cubic zirconia in mm.*
Average Y is **5.733569 units** with **standard deviation** of **1.166058 units**
The Y range is between **0.00 units (Minimum)** and **58.90 units (Maximum)**
The **median** of the Y is **5.71 units**
Total count is **26967**

Z

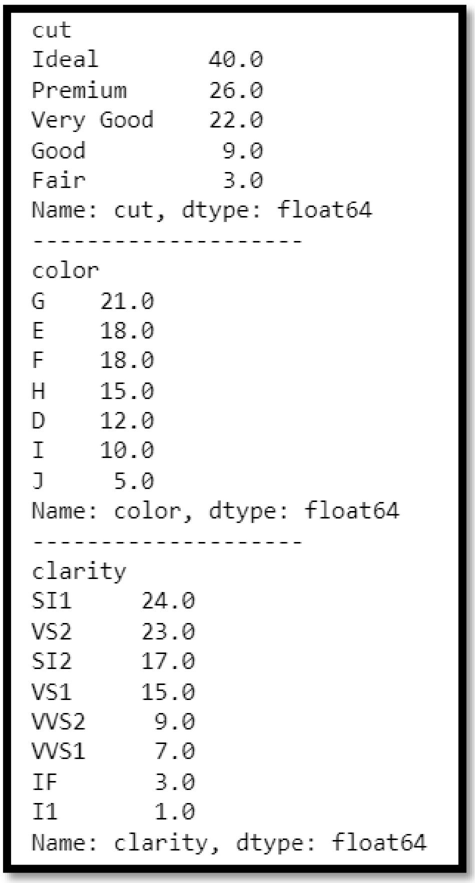
Here Z is basically *the Height of the cubic zirconia in mm.*
Average Z is **3.538057 units** with **standard deviation** of **0.720624 units**
The Z range is between **0.00 units (Minimum)** and **31.80 units (Maximum)**
The **median** of the Z is **3.52 units**
Total count is **26967**

Price

Here Price is *the Price of the cubic zirconia.*
Average Price is **3939.518115 units** with **standard deviation** of **4024.864666 units**
The **Price** range is between **326.00 units (Minimum)** and **18818.00 units (Maximum)**
The **median** of the **Price** is **2375.00 units**
Total count is **26967**

The data description also shows the description of the categorical variables.

The detailed categories are shown below-



```
cut
Ideal        40.0
Premium      26.0
Very Good    22.0
Good          9.0
Fair          3.0
Name: cut, dtype: float64
--------------------
color
G    21.0
E    18.0
F    18.0
H    15.0
D    12.0
I    10.0
J     5.0
Name: color, dtype: float64
--------------------
clarity
SI1     24.0
VS2     23.0
SI2     17.0
VS1     15.0
VVS2     9.0
VVS1     7.0
IF       3.0
I1       1.0
Name: clarity, dtype: float64
```

Cut

Here Cut **is actually** *description of the cut quality of the cubic zirconia.*
Quality **is** increasing **order** Fair, Good, Very Good, Premium, Ideal.
Hence there are **5 unique** categories.

Highest count(40%) –Ideal type

Lowest count(3%)- Fair type

Color

Here Color **is actually** *the Colour of the cubic zirconia.*
With D **being the** worst **and** J **the** best.
There are **7 unique** categories**.**

Highest count(21%) –G type

Lowest count(5%)- J type

Best color type has the lowest count.

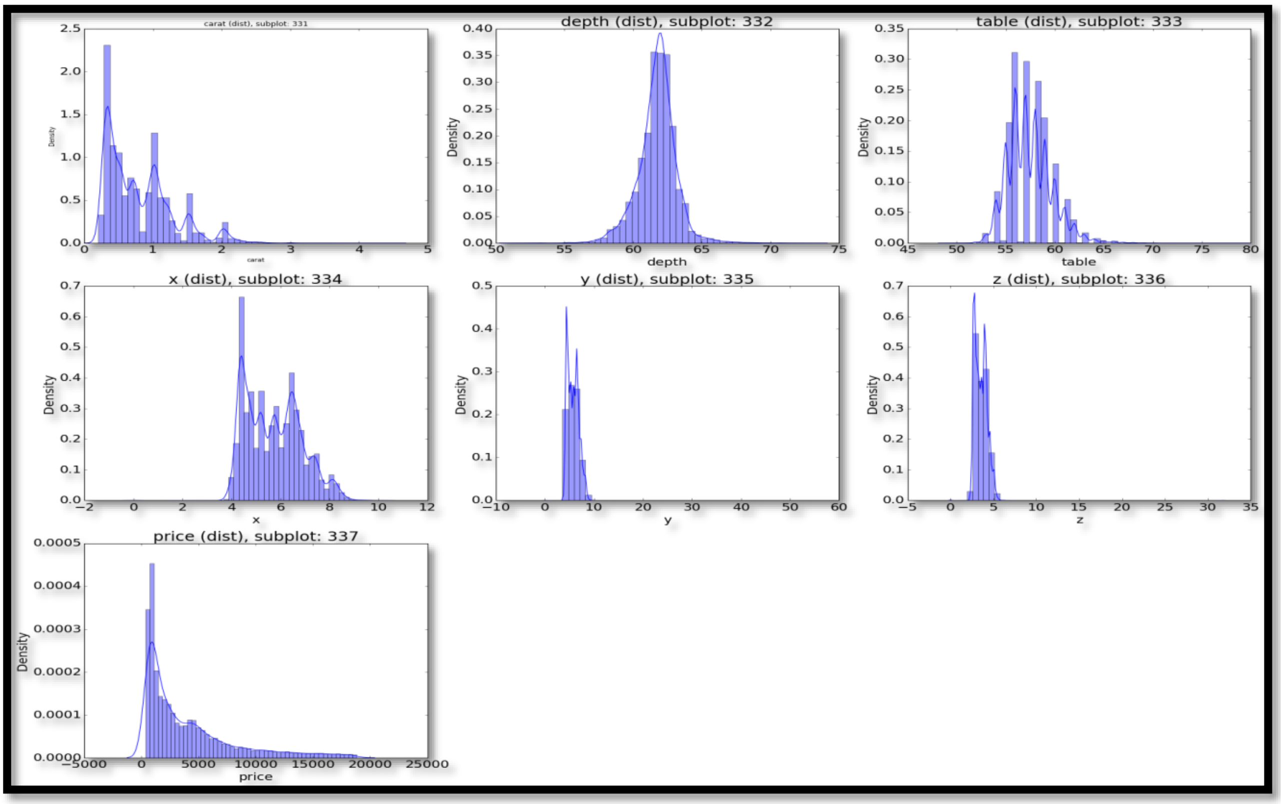
Clarity

Here Clarity **actually refers to** *the absence of the Inclusions and Blemishes.*
(**In order from** Worst **to** Best **in terms of** Average price)
IF, VVS1, VVS2, VS1, VS2, Sl1, Sl2, l1
There are **8 unique** categories.

Highest count(24%) – Sl1 type

Lowest count(1%)- I1 type

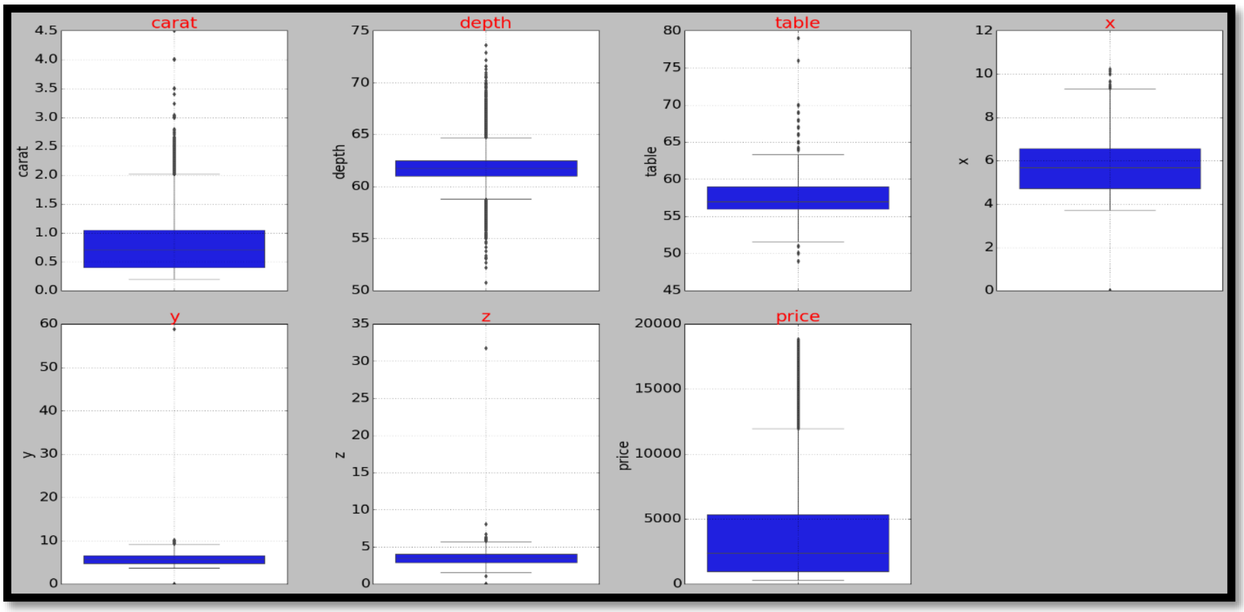
Distribution plots(histogram) or similar plots for the continuous columns. - Figure 1



After plotting the histogram plots for all the numerical variables, the following can be concluded-

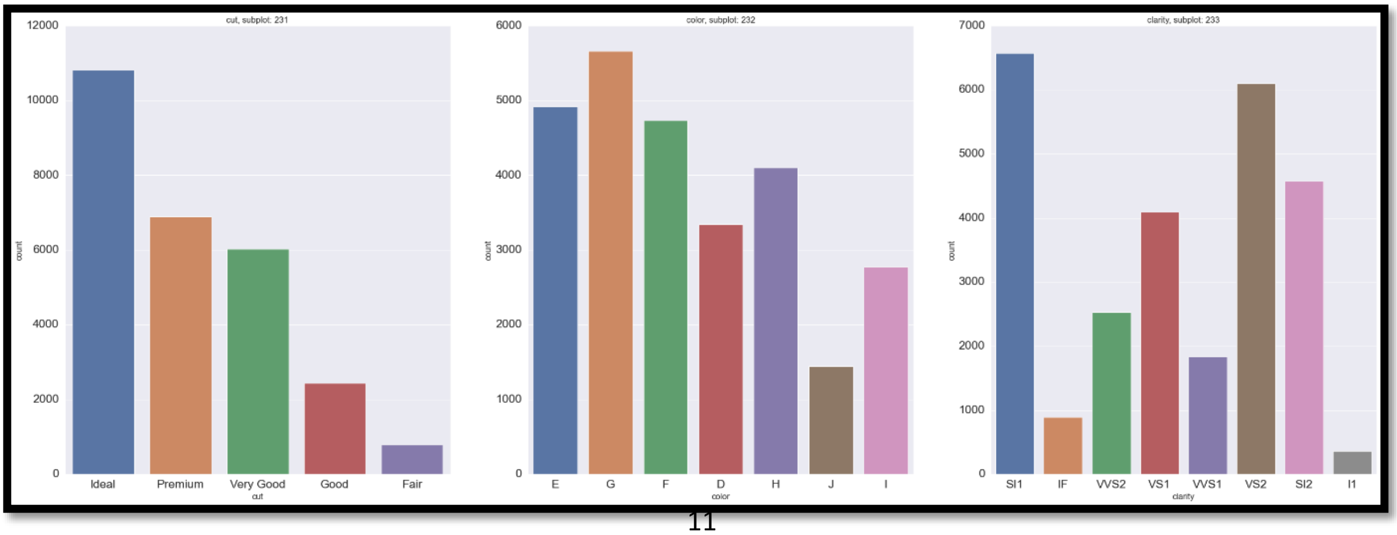
- Multi modal distribution can be seen in few of the variables.
- Carat- The distribution here seems to have positive skewness. There is also presence of multi-modes as multiple peak points are there in the distribution.
- Depth- The distribution here seems to be a normal distribution. The range is approximately between 55-65 units.
- Table- The distribution here seems to have positive skewness. The maximum distribution is between the ranges 55-65 units.
- X- The distribution here seems to have positive skewness. The maximum distribution is between the ranges 4-8 units.
- Y- The distribution here seems to have positive skewness along with it being leptokurtic (positive excess kurtosis, with heavy tails indicating larger outliers). The reason may be due to the fact that the diamonds are mostly made in specific shape and more or less a little variation in width and hence there might not be too many sizes.
- Z- The distribution here seems to have positively skewness along with it being leptokurtic like Y. The reason may be due to the fact that the diamonds are mostly made in specific shape and more or less a little variation in height and hence there might not be too many sizes available.
- Price- The distribution here seems to have positive skewness. The maximum distribution is between the ranges 1000-10000 units

#Check for Box plots(without outlier treatment) for the continuous columns- Figure 2



- Carat- The boxplot shows a large no. of outliers in this category, with positive skewness.
- Depth- The boxplot shows a large no. of outliers in this category on the either side, even though the distribution is normal.
- Table- The boxplot shows a large no. of outliers in this category, with positive skewness.
- X- The boxplot shows a few outliers in this category because the variation is low, with positive skewness.
- Y- The boxplot shows a few outliers in this category because the variation is low, with positive skewness.
- Z- The boxplot shows a few outliers in this category because the variation is low, with positive skewness.
- Price- The boxplot has too many outliers.

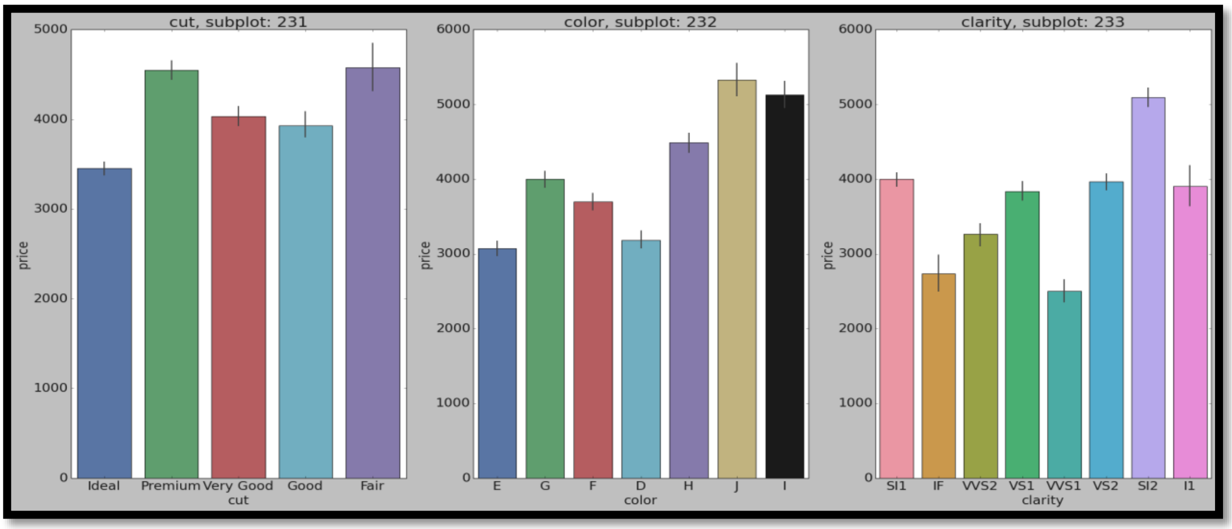
#Check for Bar plots for the categorical columns- Figure 3



The conclusions that can be drawn from the above bar plots are-

- In the first bar graph, we can observe different categories of 'Cut' feature. Most frequent is the Ideal type, followed by Premium, with least frequency in the Fair type.
- In the second bar graph, we can observe different categories of 'Color' feature. Most frequent is the G type, followed by E, with least frequency in the J type.
- In the third bar graph, we can observe different categories of 'Clarity' feature. Most frequent is the SI1 type, followed by VS2, with least frequency in the I1 type.

# Check Bar plots for the categorical columns w.r.t. price- Figure 4

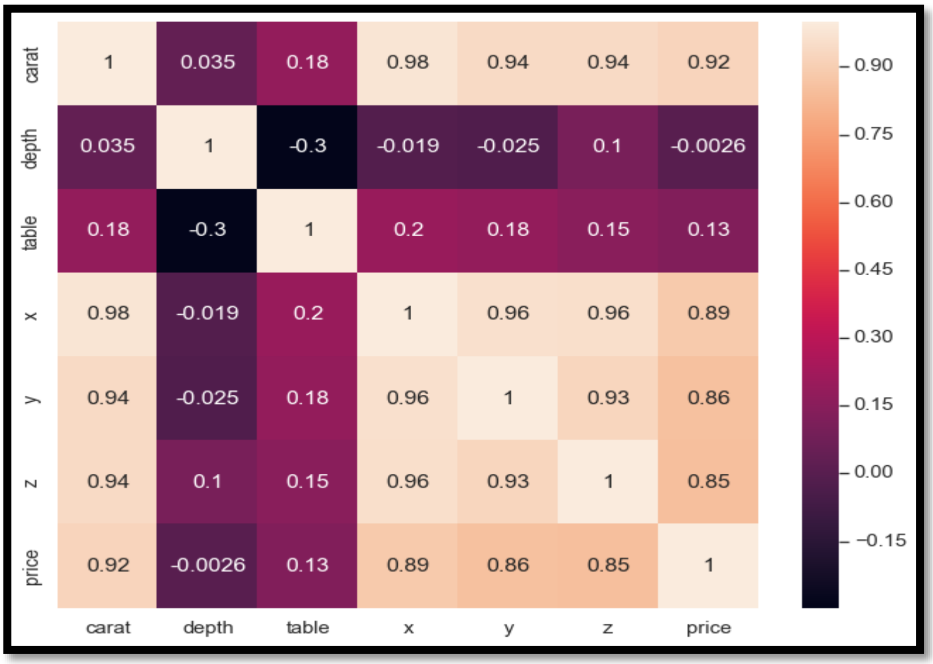


The conclusions that can be drawn from the above bar plots are-

- In the first bar graph, we can observe different categories of 'Cut' feature against the Price variable. Most costly cut type is the Fair type, followed by Premium, with least costly being the Ideal type.
- In the second bar graph, we can observe different categories of 'Color' feature against the Price variable. Most costly color is the J type, followed by I, with least costly being the E type.
- In the third bar graph, we can observe different categories of 'Clarity' feature against the Price variable. Most costly is the SI2 type, followed by SI1, with least costly being the VVS1 type.

<u># Check the heatmap with only continuous variables- Figure 5</u>

Heatmap or Corrleation plot is basically being used to evaluate the relationship between the different numeric variables within a dataset

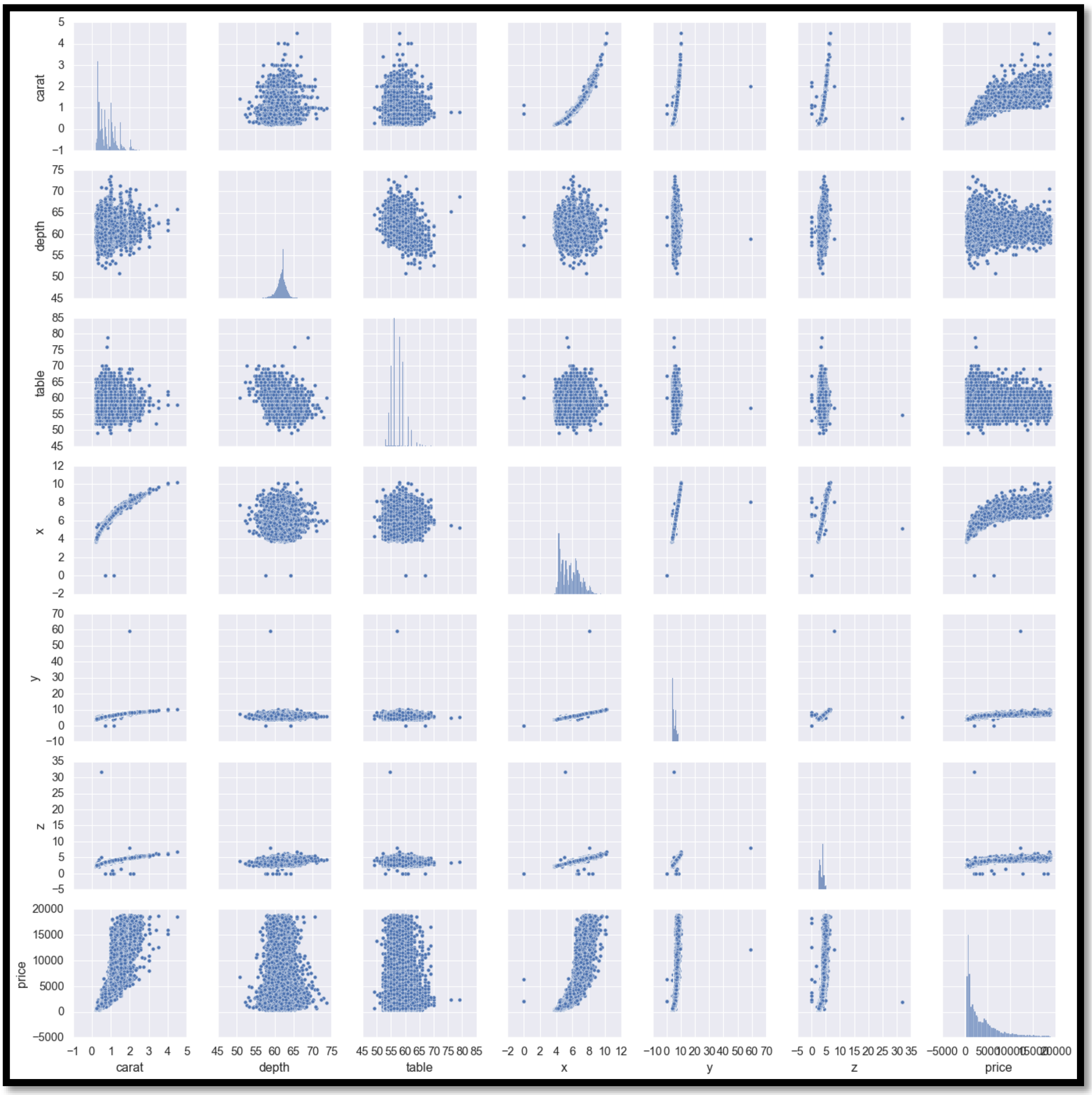


	carat	depth	table	x	y	z	price
carat	1	0.035	0.18	0.98	0.94	0.94	0.92
depth	0.035	1	-0.3	-0.019	-0.025	0.1	-0.0026
table	0.18	-0.3	1	0.2	0.18	0.15	0.13
x	0.98	-0.019	0.2	1	0.96	0.96	0.89
y	0.94	-0.025	0.18	0.96	1	0.93	0.86
z	0.94	0.1	0.15	0.96	0.93	1	0.85
price	0.92	-0.0026	0.13	0.89	0.86	0.85	1

From the above Correlation plots using Heatmap, the following facts can be concluded-

- Presence of correlation can be observed for price with that of the features like- Carat, X,Y,Z, while that with table and depth it is significantly low.
- In case of multicollinearity, it can be observed in most of the independent variables.
- Carat has high correlation with that of the independent features like X, Y, Z.
- Depth and table has very low correlation with the rest of the features.
- X, Y, Z features have very high correlation with all the independent features except Depth and Table. They have high correlation with the target variable i.e., Price.

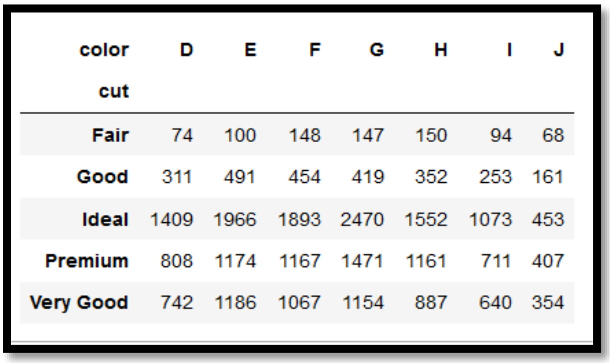
# Check with the pair plots for Bi variate data- Figure 6



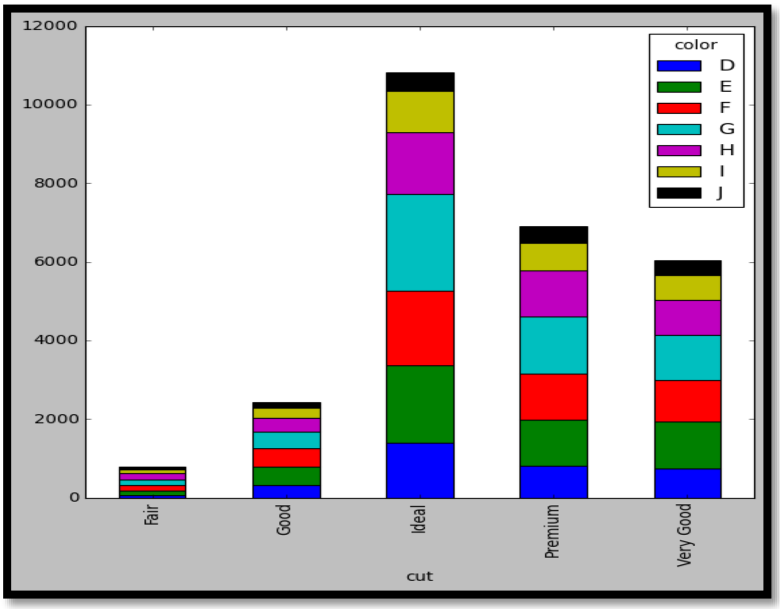
With the help of the pairplot , we can understand all the univariate and bivariate trend of the datapoints/ variables in the dataset

# Check with the count plots for Bi variate data (Categorical)- Figure 7

CUT vs COLOR



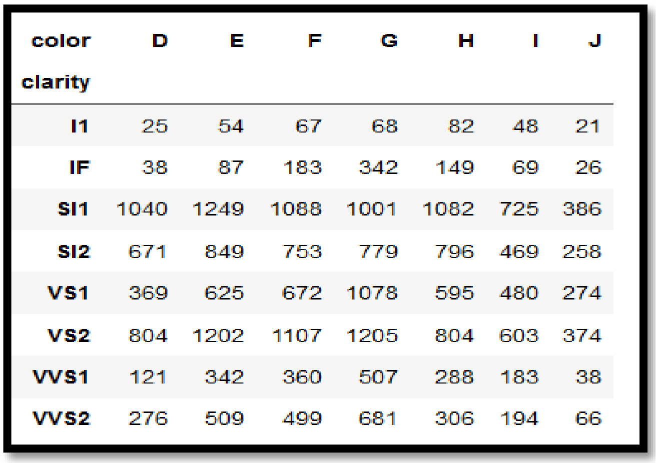
color cut	D	E	F	G	H	I	J
Fair	74	100	148	147	150	94	68
Good	311	491	454	419	352	253	161
Ideal	1409	1966	1893	2470	1552	1073	453
Premium	808	1174	1167	1471	1161	711	407
Very Good	742	1186	1067	1154	887	640	354



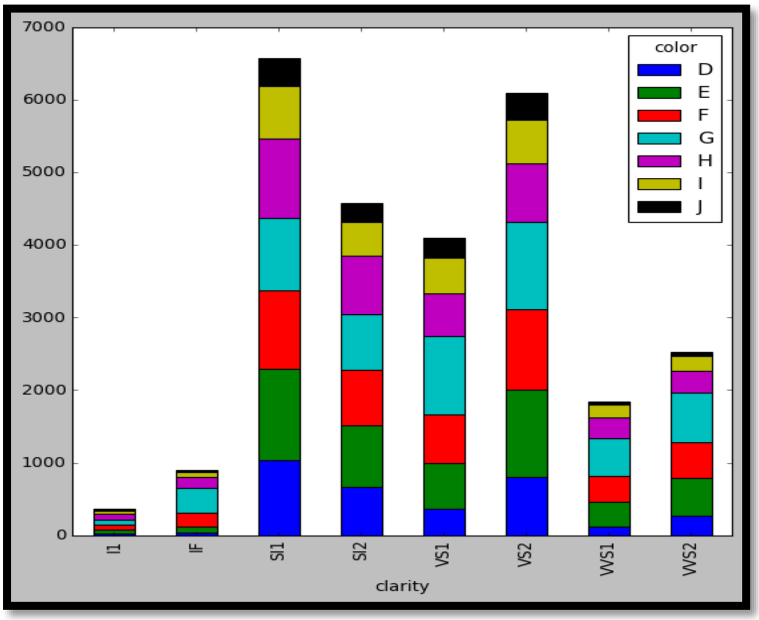
Conclusions from the above Graph

- For the D type color, which is the worst type, most frequent cut is the Ideal one(the best quality.
- For the J type color, which is the best one, most frequent cut type is Ideal. If we check from the range of color, worst to best, Ideal type is the most frequent one used in each color type, followed by either Premium or very good cut type.

CLARITY vs COLOR



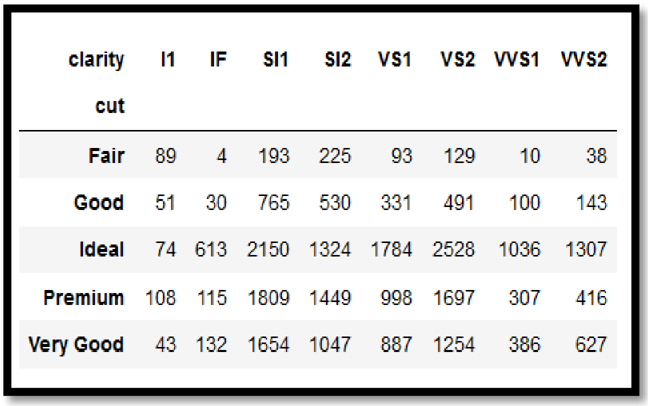
color clarity	D	E	F	G	H	I	J
I1	25	54	67	68	82	48	21
IF	38	87	183	342	149	69	26
SI1	1040	1249	1088	1001	1082	725	386
SI2	671	849	753	779	796	469	258
VS1	369	625	672	1078	595	480	274
VS2	804	1202	1107	1205	804	603	374
VVS1	121	342	360	507	288	183	38
VVS2	276	509	499	681	306	194	66



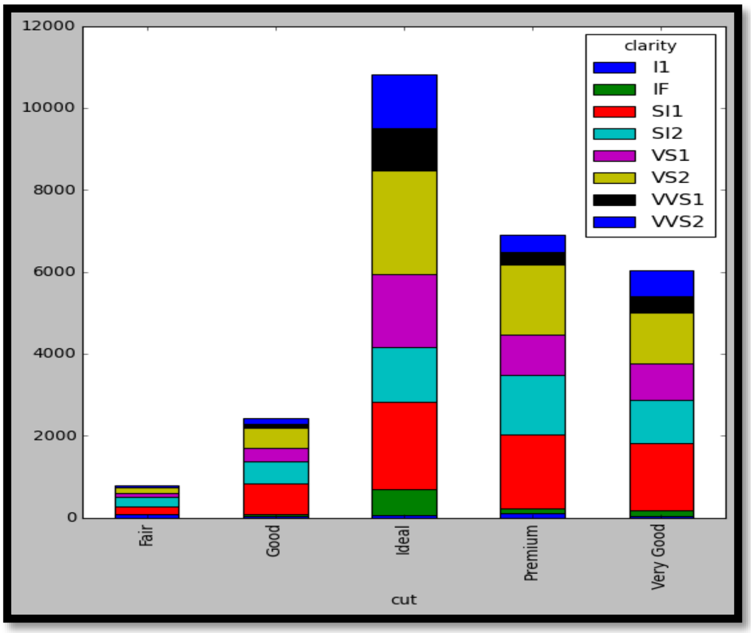
Conclusions from the above Graph

- For the D type color, which is the worst type, most frequent clarity is the SI1 (the third best in terms of price) and least frequent is the I1 (which is the best one).
- For the J type color, which is the best one, most frequent clarity is the SI1, followed by VS2 (the fourth best as clarity).
  If we check from the range of color, worst to best, SI1 type is the most frequent one used in each clarity wise feature, followed by VS2 clarity type.

CLARITY vs CUT



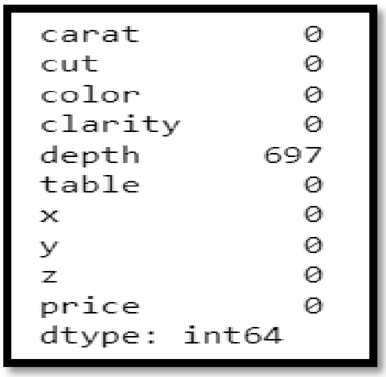
clarity cut	I1	IF	SI1	SI2	VS1	VS2	VVS1	VVS2
Fair	89	4	193	225	93	129	10	38
Good	51	30	765	530	331	491	100	143
Ideal	74	613	2150	1324	1784	2528	1036	1307
Premium	108	115	1809	1449	998	1697	307	416
Very Good	43	132	1654	1047	887	1254	386	627



Conclusions from the above Graph

- For the IF type clarity, which is the worst type in terms of average price, most frequent cut is the Premium one (the second best quality type) and least frequent is the Very Good type (which is the third best one).
- For the IF type clarity, which is the worst one, most frequent cut is the Ideal (the best quality), followed by Very Good type.
  If we check from the range of clarity, worst to best, Ideal cut type is the most frequent used irrespective of the price involved, followed by Premium cut type.

Check for the Null Values- Table 4



```
carat        0
cut          0
color        0
clarity      0
depth      697
table        0
x            0
y            0
z            0
price        0
dtype: int64
```

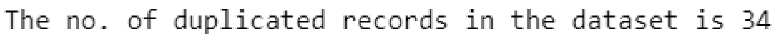
The missing values or 'NaN' or 'NA' values are in general to be cleaned during the data cleaning process.

They are basically –

1. Removed
2. Replaced with mode function (Categorical variables)
3. Replaced with Median function (Continuous variables)
4. Replace with other processes

In this dataset, **697 missing values** are present in the **depth** feature.

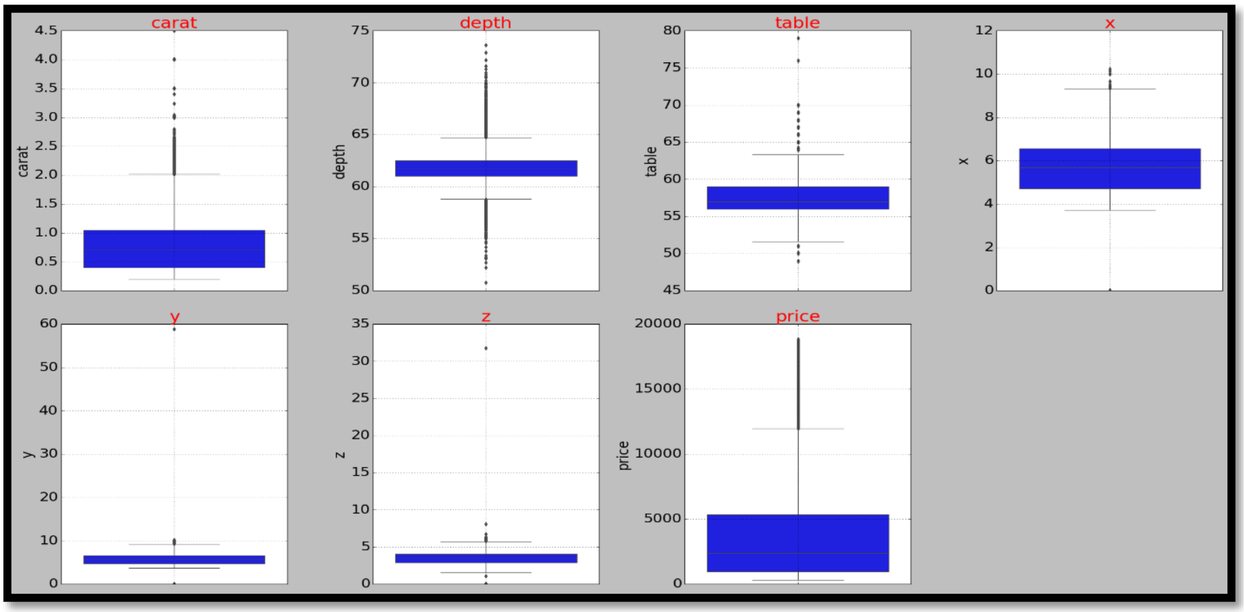
Check for the Duplicated Values



```
The no. of duplicated records in the dataset is 34
```

Check for the Outliers Values-

This is also being shown in the Boxplot section before, where outliers can be identified.



****

1.2 Impute null values if present, also check for the values which are equal to zero. Do they have any meaning or do we need to change them or drop them? Check for the possibility of combining the sub levels of a ordinal variables and take actions accordingly. Explain why you are combining these sub levels with appropriate reasoning.

Null or Missing values

The null values are imputed with the median in this case .Replacement of the null values occurred with median value due to the fact that the variable depth is continuous in nature and also there are presence of outliers.

Duplicate values

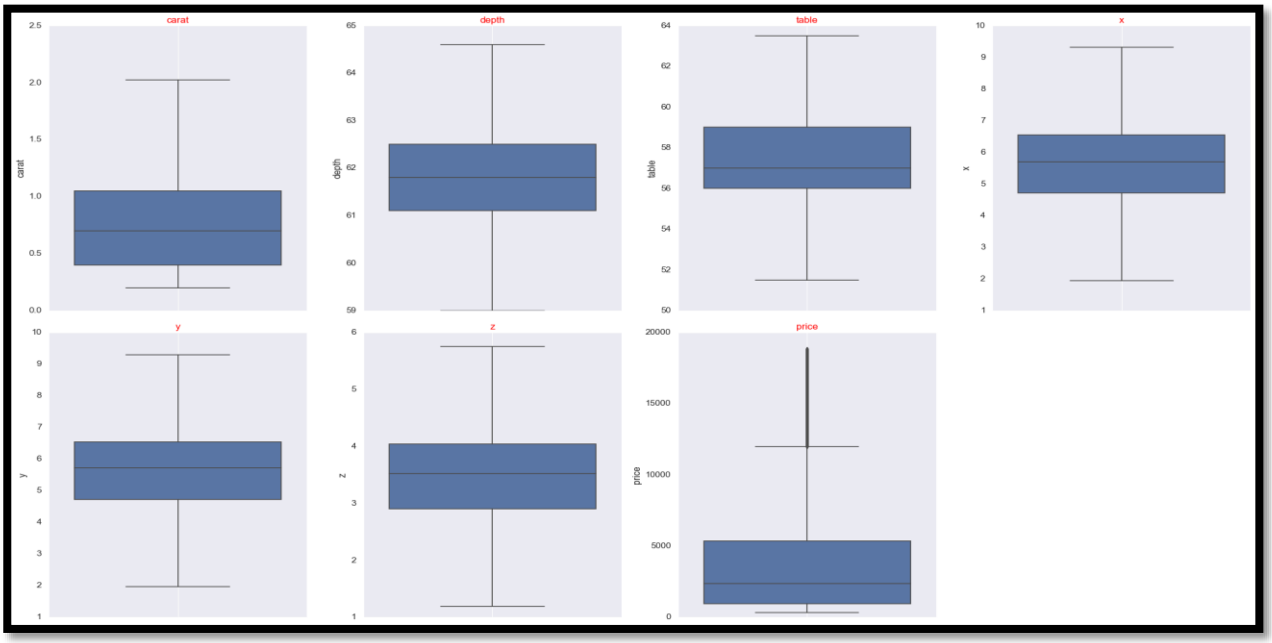
The duplicate values are dropped from the dataset.

Outliers

Outliers are basically the extreme values in the dataset. Outliers increase the variability in your data, which decreases statistical power.

As the dataset has huge no. of outliers in most of the continuous variables, we created a user defined function to remove the outliers from the data. The boxplot after outlier removal is show below.

#Check for Box plots (with outlier treatment) for the continuous columns- Figure 8



We can combine sublevels of an ordinal variable, to understand if it increases the efficiency of the model. For example combination of Good and Very Good variety for the Cut category. But we haven't used the same in this particular model because, there can issue where we won't be able to understand, if they have contributed individually.

****

1.3 Encode the data (having string values) for Modelling. Split the data into train and test (70:30). Apply Linear regression using scikit learn. Perform checks for significant variables using appropriate method from statsmodel. Create multiple models and check the performance of Predictions on Train and Test sets using Rsquare, RMSE & Adj Rsquare. Compare these models and select the best one with appropriate reasoning.

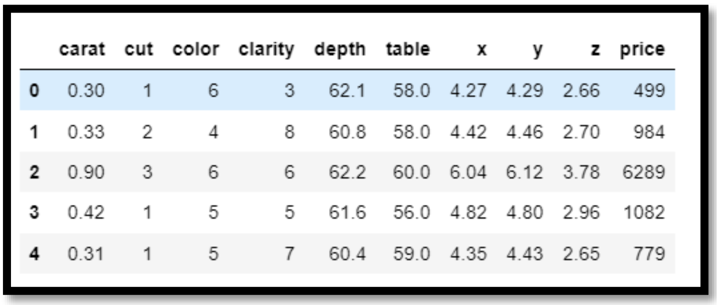
Before splitting the data into train test, we opted for the Encoding process of the categorical variables.

Here in this case, we have used a user defined encoder instead of Label encoder or One hot encoder.

The reasons being-

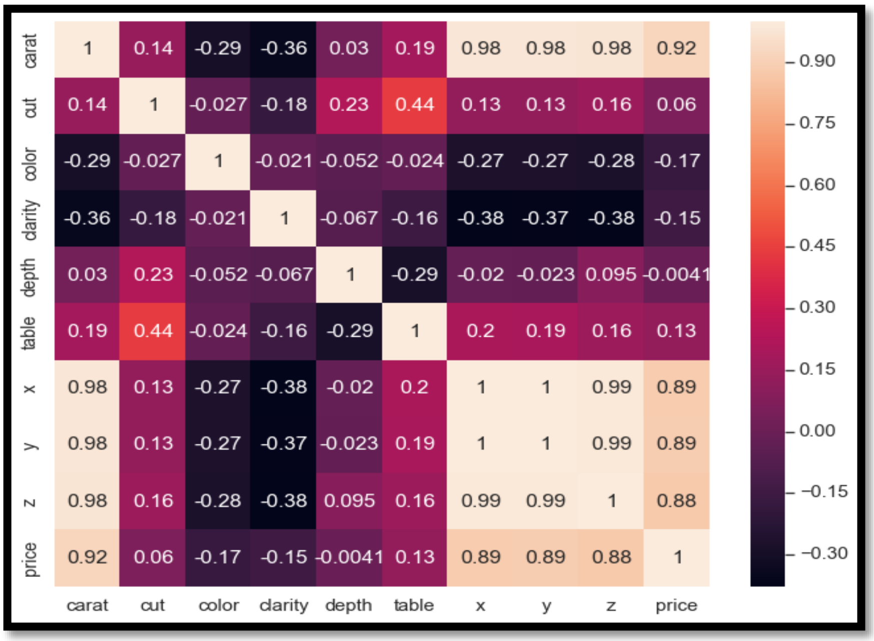
- The categories are in a particular order which are to be manually encoded.
- Considering the data being ordinal, we can't use Label encoding as it will encode based on the categories in Alphabetical order, which is not at all required.
- Even, the creation of dummy variables through one hot encoding is not preferable as it will increase the column nos. and it won't serve the purpose required here.

Check for the head Values after Label Encoding- Table 5



	carat	cut	color	clarity	depth	table	x	y	z	price
0	0.30	1	6	3	62.1	58.0	4.27	4.29	2.66	499
1	0.33	2	4	8	60.8	58.0	4.42	4.46	2.70	984
2	0.90	3	6	6	62.2	60.0	6.04	6.12	3.78	6289
3	0.42	1	5	5	61.6	56.0	4.82	4.80	2.96	1082
4	0.31	1	5	7	60.4	59.0	4.35	4.43	2.65	779

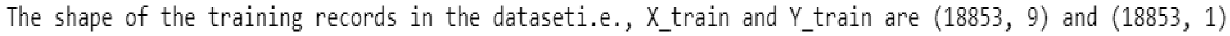
# Check the heatmap with only continuous variables after encoding- Figure 9



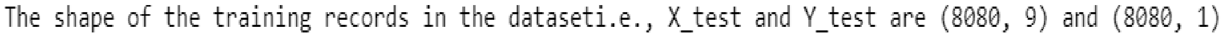
	carat	cut	color	clarity	depth	table	x	y	z	price
carat	1	0.14	-0.29	-0.36	0.03	0.19	0.98	0.98	0.98	0.92
cut	0.14	1	-0.027	-0.18	0.23	0.44	0.13	0.13	0.16	0.06
color	-0.29	-0.027	1	-0.021	-0.052	-0.024	-0.27	-0.27	-0.28	-0.17
clarity	-0.36	-0.18	-0.021	1	-0.067	-0.16	-0.38	-0.37	-0.38	-0.15
depth	0.03	0.23	-0.052	-0.067	1	-0.29	-0.02	-0.023	0.095	-0.0041
table	0.19	0.44	-0.024	-0.16	-0.29	1	0.2	0.19	0.16	0.13
x	0.98	0.13	-0.27	-0.38	-0.02	0.2	1	1	0.99	0.89
y	0.98	0.13	-0.27	-0.37	-0.023	0.19	1	1	0.99	0.89
z	0.98	0.16	-0.28	-0.38	0.095	0.16	0.99	0.99	1	0.88
price	0.92	0.06	-0.17	-0.15	-0.0041	0.13	0.89	0.89	0.88	1

The Heatmap of the correlation for the dataset after manual encoding of the categorical values is shown in the figure on the last page.
The categorical variables i.e., cut, color and clarity does not have significant correlation with that of Target variable price as well as with the other independent variables.

After this the model is being split into train and test dataset.



```
The shape of the training records in the dataseti.e., X_train and Y_train are (18853, 9) and (18853, 1)
```



```
The shape of the training records in the dataseti.e., X_test and Y_test are (8080, 9) and (8080, 1)
```

After this we have scaled the data using Standard scalar on the continuous variables.

Type of Scaling Technique used -

We will scale the data based on Z-Score method or Standard Scalar in SkLearn
• Standard Scalar standardizes a feature by subtracting the mean and then scaling to unit variance. Unit variance means dividing all the values by the standard deviation.
• Standard Scalar results in a distribution with a standard deviation equal to 1. The variance is equal to 1 also, because variance = standard deviation squared. And 1 squared = 1.
• Standard Scalar makes the mean of the distribution 0. About 68% of the values will lie be between -1 and 1.

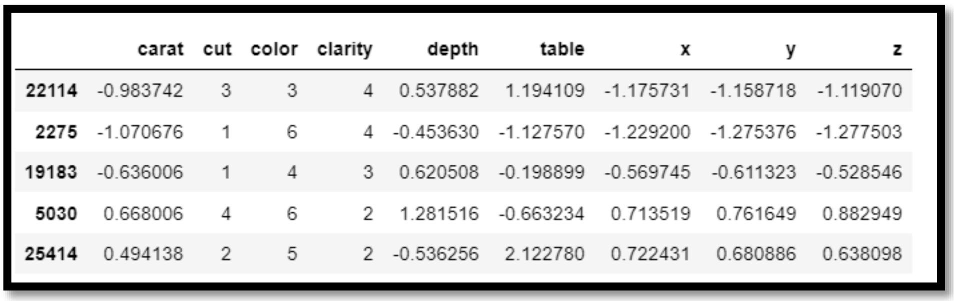
Standard Scalar normalizes the data using the formula (x-mean)/standard deviation.

Z = (value -mean)/standard deviation

Why it is to be used?
If your variables are of incomparable units (e.g. height in cm and weight in kg) then you should standardize variables, of course. Even if variables are of the same units but show quite different variances it is still a good idea to standardize before K-means. You see, K-means clustering is "isotropic" in all directions of space and therefore tends to produce more or less round (rather than elongated) clusters. In this situation leaving variances unequal is equivalent to putting more weight on variables with smaller variance, so clusters will tend to be separated along variables with greater variance.

Explanation with respect to the Variables given
As per the data provided, standard scaling is required as all the different variables are provided in different units, for example, Carat weight, Depth and Tables don't have any units associated with them, while the features like X, Y, Z are in mm. As there are differences in units, hence other values expressed in higher units will outweigh the variables in lower units and can give varied results. This is why scaling is important, and Standard scalar, as mentioned above, normalise the data points with mean 0 and standard deviation 1.

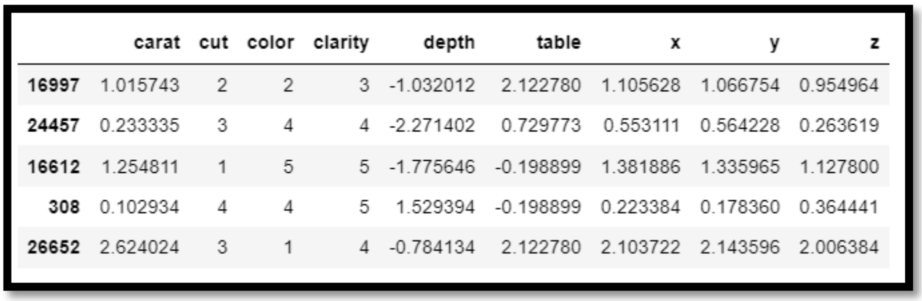
#Check with the output of scaled data-Trained data- Table 6



	carat	cut	color	clarity	depth	table	x	y	z
22114	-0.983742	3	3	4	0.537882	1.194109	-1.175731	-1.158718	-1.119070
2275	-1.070676	1	6	4	-0.453630	-1.127570	-1.229200	-1.275376	-1.277503
19183	-0.636006	1	4	3	0.620508	-0.198899	-0.569745	-0.611323	-0.528546
5030	0.668006	4	6	2	1.281516	-0.663234	0.713519	0.761649	0.882949
25414	0.494138	2	5	2	-0.536256	2.122780	0.722431	0.680886	0.638098

The above table shows the columns with scaled value for the Continuous data types in the trained dataset

#Check with the output of scaled data-Test data- Table 7



	carat	cut	color	clarity	depth	table	x	y	z
16997	1.015743	2	2	3	-1.032012	2.122780	1.105628	1.066754	0.954964
24457	0.233335	3	4	4	-2.271402	0.729773	0.553111	0.564228	0.263619
16612	1.254811	1	5	5	-1.775646	-0.198899	1.381886	1.335965	1.127800
308	0.102934	4	4	5	1.529394	-0.198899	0.223384	0.178360	0.364441
26652	2.624024	3	1	4	-0.784134	2.122780	2.103722	2.143596	2.006384

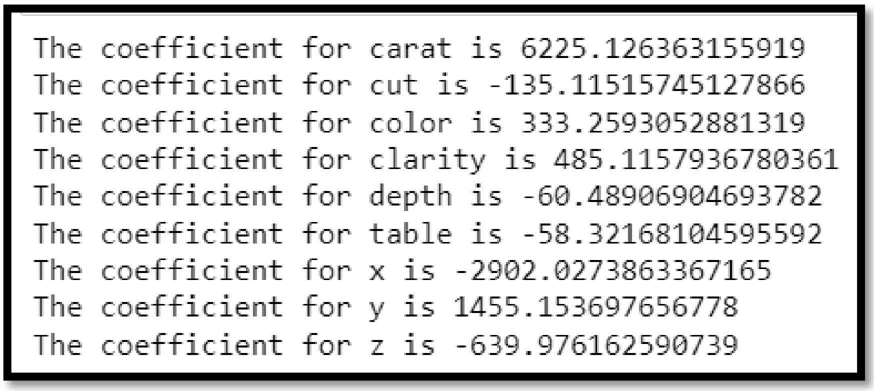
The above table shows the columns with scaled value for the Continuous data types in the test dataset

Then after this part, we have tried two different process of predictive modelling process.

#Method 1-Using Sklearn

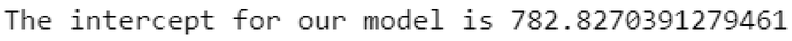
The first process being the standard one where the dataset is being applied with the Regression model, which is done with help of the library Sklearn and the different modules as required.

#Check with the coefficients of different features - Table 8



```
The coefficient for carat is 6225.126363155919
The coefficient for cut is -135.11515745127866
The coefficient for color is 333.2593052881319
The coefficient for clarity is 485.1157936780361
The coefficient for depth is -60.48906904693782
The coefficient for table is -58.32168104595592
The coefficient for x is -2902.0273863367165
The coefficient for y is 1455.153697656778
The coefficient for z is -639.976162590739
```

The value of the intercept is shown below-



```
The intercept for our model is 782.8270391279461
```

The different statistics obtained are-

❖ ***R^2 measure-***

- R-squared (R^2) is a statistical measure that **represents the proportion of the variance for a dependent variable** that's explained by an independent variable or variables in a regression model.
- It is also known as the coefficient of determination is used to evaluate the performance of a linear regression model.
- What qualifies as a "good" R-Squared value will depend on the context. In some fields, such as the social sciences, even a relatively low R-Squared such as 0.5 could be considered relatively strong. In other fields, the standards for a good R-Squared reading can be much higher, such as 0.9 or above.
- Essentially, an R-Squared value of 0.9 would indicate that 90% of the variance of the dependent variable being studied is explained by the variance of the independent variable.

Regression score or R-sqaured for the Train data model: **0.9160043267788675** or **91.60%**

Regression score or R-sqaured for the Test data model: **0.9180929919842764** or **91.80%**

❖ ***Adjusted R^2 measure-***

- Adjusted R-squared is a **modified version of R-squared** that has been adjusted for the number of predictors in the model.
- The adjusted R-squared increases when the new term improves the model more than would be expected by chance. It decreases when a predictor improves the model by less than expected.
- Typically, the adjusted R-squared is positive, not negative. It is always **lower than the R-squared**.

The value of Adjusted R^2 from the statsmodel output is 0.916

❖ *RMSE*

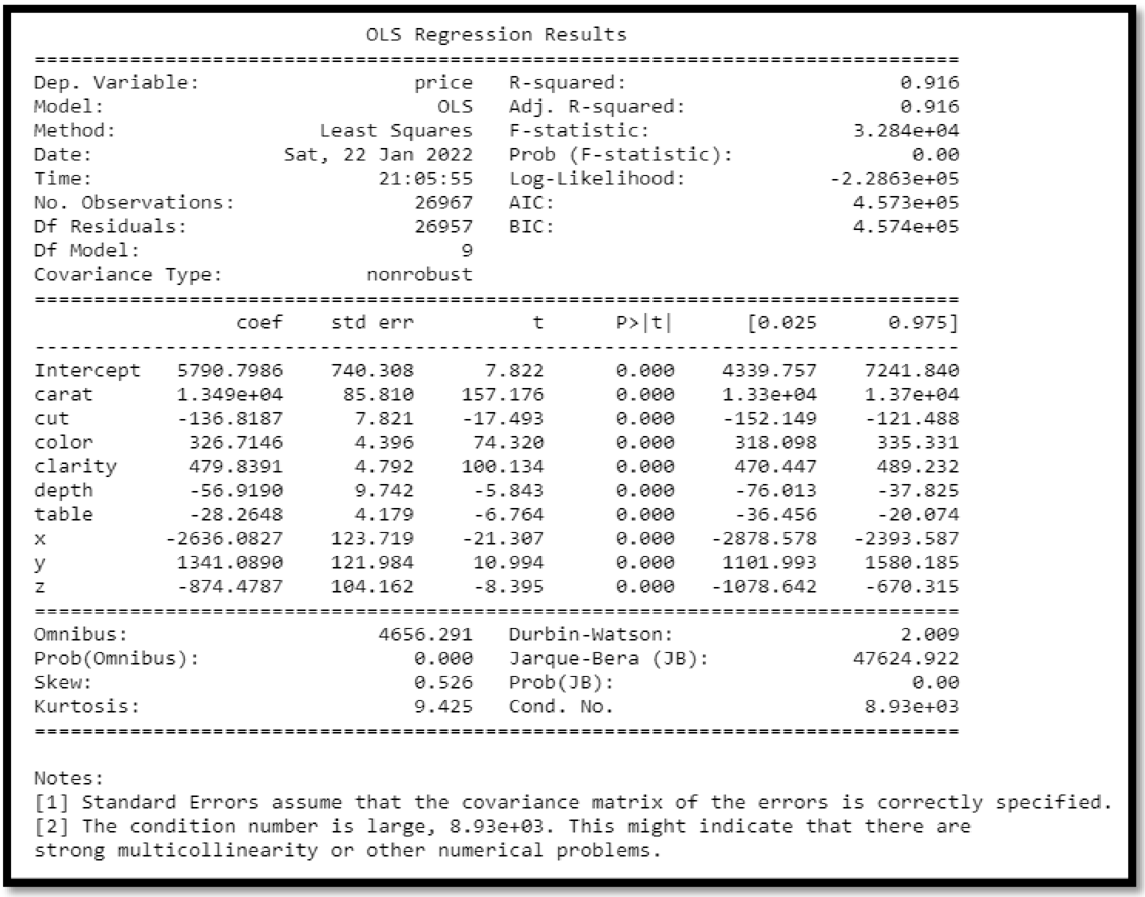
- **Root Mean Square Error** (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are.
- RMSE is a measure of how spread out these residuals are.
- In other words, it tells you how concentrated the data is around the line of best fit.
- Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results.
- Based on a rule of thumb, it can be said that RMSE values **between 0.2 and 0.5** shows that the model can relatively predict the data accurately.

The RMSE value from the model is 0.34 for both the training and the test set.

<u>#Method 2-Using Statsmodel</u>

The second process being the different one where the dataset is being applied with the Regression model, which is done with help of the library Stats model.
Scikit does not provide a facility for adjusted R^2. So we have used statsmodel, a library that gives results similar to what you obtain in R language
This library expects the X and Y to be given in one single data frame.

<u>#Check the statsmodel output - Table 9</u>



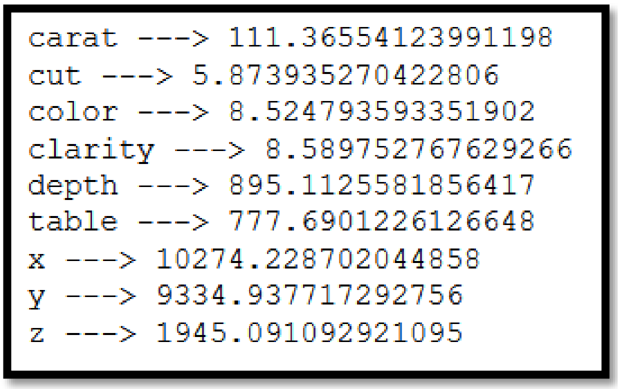
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.916
Model:                            OLS   Adj. R-squared:                  0.916
Method:                 Least Squares   F-statistic:                 3.284e+04
Date:                Sat, 22 Jan 2022   Prob (F-statistic):               0.00
Time:                        21:05:55   Log-Likelihood:            -2.2863e+05
No. Observations:               26967   AIC:                         4.573e+05
Df Residuals:                   26957   BIC:                         4.574e+05
Df Model:                           9
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   5790.7986    740.308      7.822      0.000    4339.757    7241.840
carat       1.349e+04     85.810    157.176      0.000    1.33e+04    1.37e+04
cut         -136.8187      7.821    -17.493      0.000    -152.149    -121.488
color        326.7146      4.396     74.320      0.000     318.098     335.331
clarity      479.8391      4.792    100.134      0.000     470.447     489.232
depth        -56.9190      9.742     -5.843      0.000     -76.013     -37.825
table        -28.2648      4.179     -6.764      0.000     -36.456     -20.074
x          -2636.0827    123.719    -21.307      0.000   -2878.578   -2393.587
y           1341.0890    121.984     10.994      0.000    1101.993    1580.185
z           -874.4787    104.162     -8.395      0.000   -1078.642    -670.315
==============================================================================
Omnibus:                     4656.291   Durbin-Watson:                   2.009
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            47624.922
Skew:                           0.526   Prob(JB):                         0.00
Kurtosis:                       9.425   Cond. No.                     8.93e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 8.93e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Output from VIF

- Variance inflation factor (vif) is a measure of the amount of multicollinearity in a set of multiple regression variables.
- Mathematically, the vif for a regression model variable is equal to the ratio of the overall model variance to the variance of a model that includes only that single independent variable.
- This ratio is calculated for each independent variable. A high vif indicates that the associated independent variable is highly collinear with the other variables in the model.

#Check the VIF output - Table 10



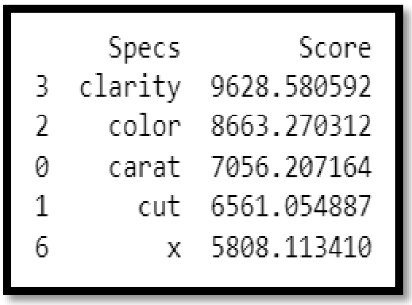
```
carat ---> 111.36554123991198
cut ---> 5.873935270422806
color ---> 8.524793593351902
clarity ---> 8.589752767629266
depth ---> 895.1125581856417
table ---> 777.6901226126648
x ---> 10274.228702044858
y ---> 9334.937717292756
z ---> 1945.091092921095
```

The vif values in this dataset is considerably high, considering presence of multicollinearity within data features.

Output from the Feature importance

We have used the 'SelectKBest' and 'Chi' from the 'Sklearn-feature selection' library, in order to find the top 5 features that are significant in this dataset.

#Check the Feature importance output - Table 11



```
      Specs        Score
3  clarity  9628.580592
2    color  8663.270312
0    carat  7056.207164
1      cut  6561.054887
6        x  5808.113410
```

We can clearly observe, that **the Clarity feature is the most important, followed by Color, Carat, Cut and X feature.**

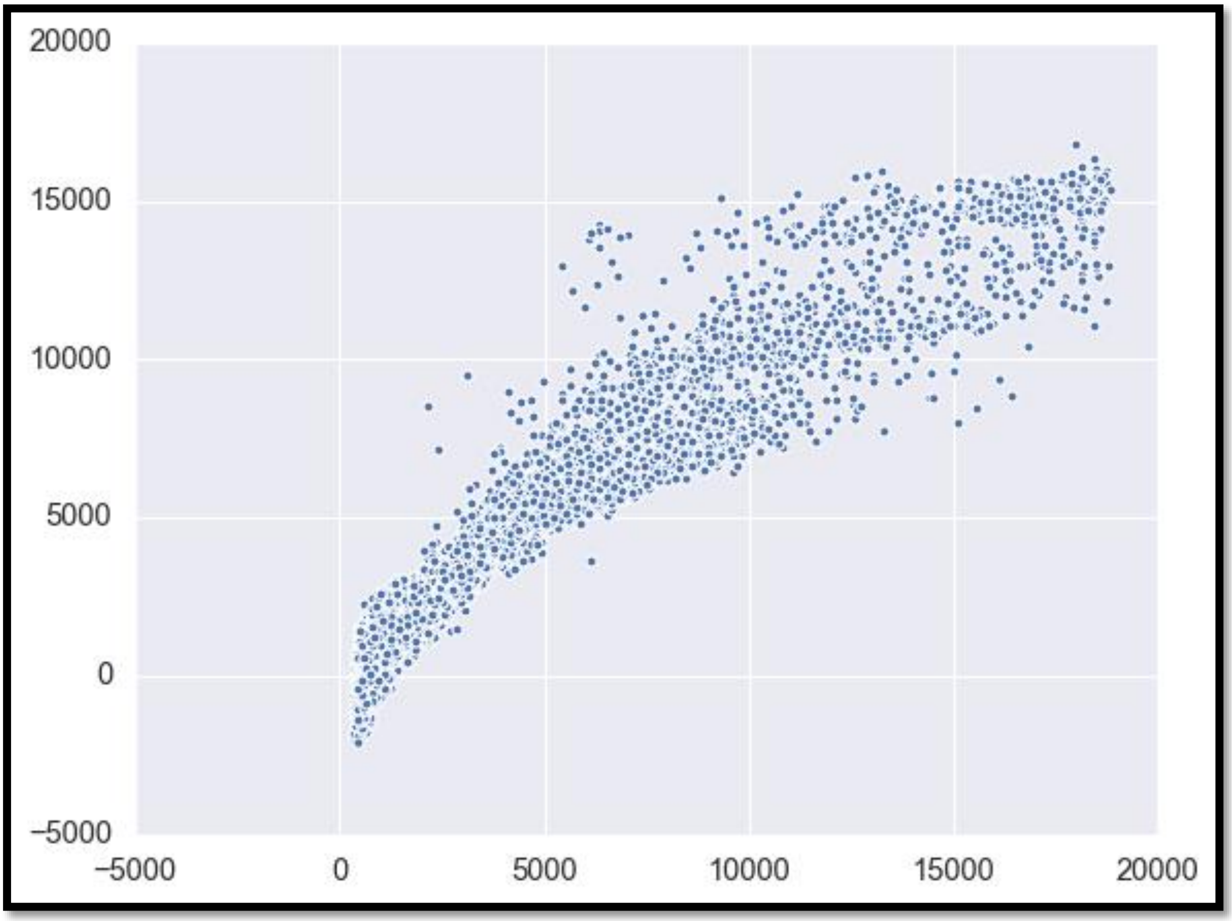
This also means that the length (X) plays a major role than the width (Y) and height (Z) of Zirconia while predicting its price.

<u>Output in terms of the Actual vs Predicted price plot generated</u>

The output from the qq plot has shown that the predicted and actual price values almost follow a curve line or a area in general, with low-moderate disparity or variance to be precise.

But to presence of multicollinearity, the residual errors are high, causing the dispersion of the data, as observed from the graph.

# Check the Scatter plot of Actual vs Predicted price- Figure 10



****

1.4 Inference: Basis on these predictions, what are the business insights and recommendations.
Please explain and summarise the various steps performed in this project.
There should be proper business interpretation and actionable insights present.

On the basis of the business problem provided to us, the price is to be determined for the stone and the features which make it profitable.
We all know that the data is not an ideal dataset considering the performance as it doesn't follow all the assumptions of Linear Regression.

Understanding from EDA

From the EDA analysis, we could understand –

- Ideal Cut type is the most profitable cut, followed by Premium type, considering being used frequently irrespective of the prices involved. But it is the cheapest amongst all the five cut types.
- From the cost point of view, the costlier ones in color type are H, I, J. But those were not being used frequently, rather, E, G, F were the most frequent ones, which are not great as compared to H, I, J in terms of price and profit.
- In clarity, we have observed, that the most frequent ones being used are SI1, SI2, VS2, which are in the top 4 clarity type based on average pricing. We consider these three to bring the most amount of profits.
- In terms of price, we have observed, high correlation with that of Carat and the different dimensional features like the length (X), width (Y) and height (Z).

Understanding from Feature importance

- The most significant features are Clarity, Color, Carat, Cut and X.

Recommendation

- The Ideal, Premium and Very good Cut types are the most frequent ones opted for, which is why they contribute majorly to the profits generated. More marketing strategies can be opted for using these types, in terms of the revenues are concerned to bring more customers and make awareness of the product.
  The company can also learn about the festive seasons or other opportunities, by providing discounts as compared to the peers, for capturing more customers and for increasing market share.

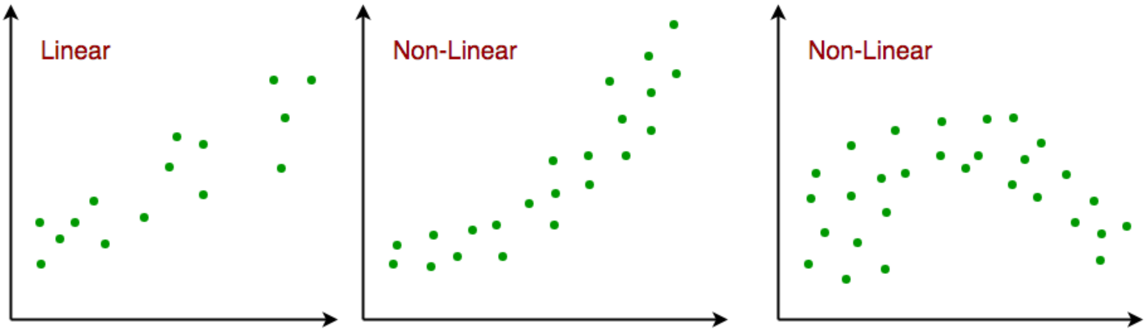
- The clarity being a significant feature, should also be considered from the revenue generation point of view, along with the carat weight.

- The company should also consider having retention policies for the existing customers, by offering them next best offer while they buy any product the next time.

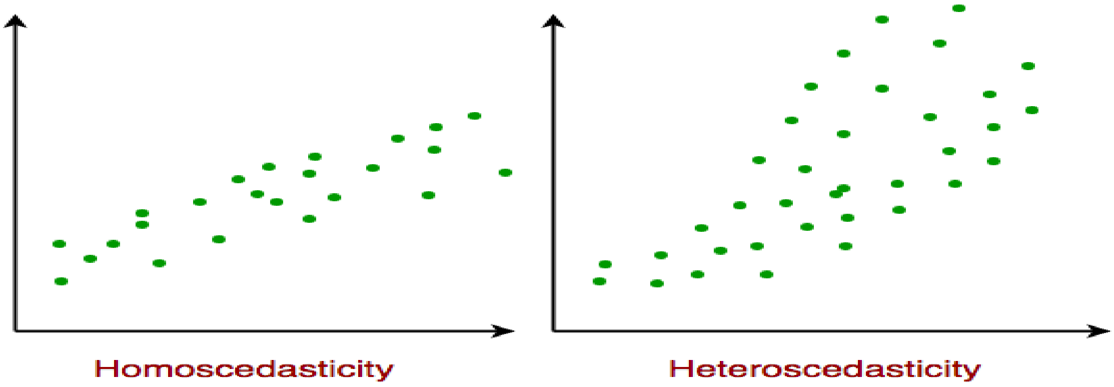
Assumptions:

Given below are the basic assumptions that a linear regression model makes regarding a dataset on which it is applied:

- **Linear relationship**: Relationship between response and feature variables should be linear. The linearity assumption can be tested using scatter plots. As shown below, 1st figure represents linearly related variables whereas variables in the 2nd and 3rd figures are most likely non-linear. So, 1st figure will give better predictions using linear regression.



- **Little or no multi-collinearity**: It is assumed that there is little or no multicollinearity in the data. Multicollinearity occurs when the features (or independent variables) are not independent of each other.
- **Little or no auto-correlation**: Another assumption is that there is little or no autocorrelation in the data. Autocorrelation occurs when the residual errors are not independent of each other.
- **Homoscedasticity**: Homoscedasticity describes a situation in which the error term (that is, the "noise" or random disturbance in the relationship between the independent variables and the dependent variable) is the same across all values of the independent variables. As shown below, figure 1 has homoscedasticity while figure 2 has heteroscedasticity.



****

Topic: Holday Package

Executive Summary

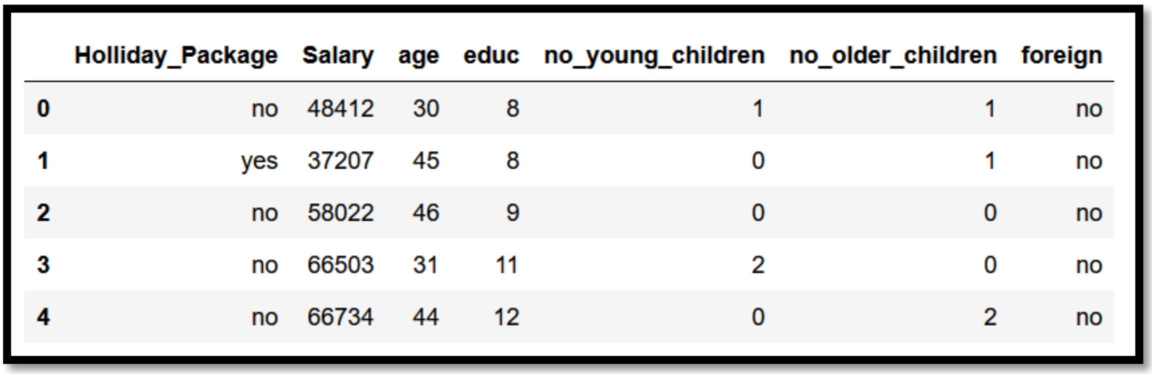
You are hired by a tour and travel agency which deals in selling holiday packages. You are provided details of 872 employees of a company. Among these employees, some opted for the package and some didn't. You have to help the company in predicting whether an employee will opt for the package or not on the basis of the information given in the data set. Also, find out the important factors on the basis of which the company will focus on particular employees to sell their packages.

Data Description

Variable Name	**Description**
Holiday_Package	Opted for Holiday Package yes/no?
Salary	Employee salary
age	Age in years
edu	Years of formal education
no_young_children	The number of young children (younger than 7 years)
no_older_children	Number of older children
foreign	foreigner Yes/No

2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, write an inference on it. Perform Univariate and Bivariate Analysis. Do exploratory data analysis.

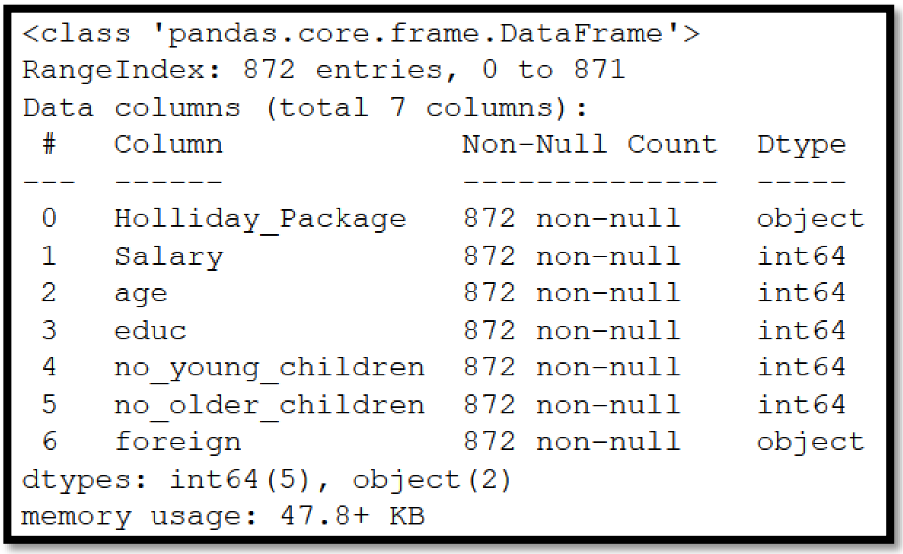
#Check for Head/Sample of the dataset – Table 1



	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
0	no	48412	30	8	1	1	no
1	yes	37207	45	8	0	1	no
2	no	58022	46	9	0	0	no
3	no	66503	31	11	2	0	no
4	no	66734	44	12	0	2	no

The above figure shows the output of the head function, showing top 5 records with the total of 7 variables or attributes

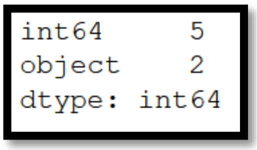
#Check for info – Table 2



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 872 entries, 0 to 871
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Holliday_Package   872 non-null    object
 1   Salary             872 non-null    int64 
 2   age                872 non-null    int64 
 3   educ               872 non-null    int64 
 4   no_young_children  872 non-null    int64 
 5   no_older_children  872 non-null    int64 
 6   foreign            872 non-null    object
dtypes: int64(5), object(2)
memory usage: 47.8+ KB
```

The above figure shows the different datatypes using the info function. To have a better count of the datatypes, we have the output of the value counts function as follows.

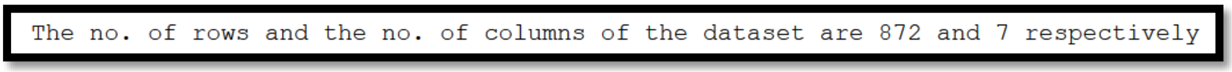
#Check for datatypes- Table 3



```
int64     5
object    2
dtype: int64
```

In the above figure we can observe that there is total 2 object data type and 5 integer data types.We will later observe that two more integer columns are actually having discrete values, which is why they are required to be converted into categorical type.

#Check the shape of the dataset

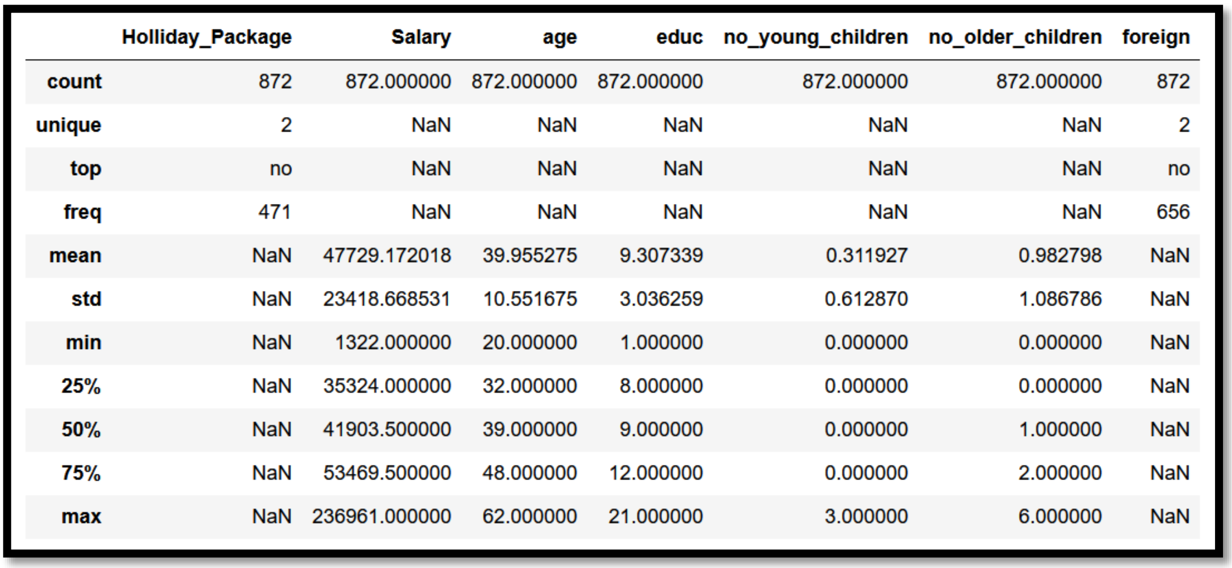


```
The no. of rows and the no. of columns of the dataset are 872 and 7 respectively
```

The dimensions of the whole dataset is shown in the above figure

----

# Check for Summary stats- Table 4



	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
count	872	872.000000	872.000000	872.000000	872.000000	872.000000	872
unique	2	NaN	NaN	NaN	NaN	NaN	2
top	no	NaN	NaN	NaN	NaN	NaN	no
freq	471	NaN	NaN	NaN	NaN	NaN	656
mean	NaN	47729.172018	39.955275	9.307339	0.311927	0.982798	NaN
std	NaN	23418.668531	10.551675	3.036259	0.612870	1.086786	NaN
min	NaN	1322.000000	20.000000	1.000000	0.000000	0.000000	NaN
25%	NaN	35324.000000	32.000000	8.000000	0.000000	0.000000	NaN
50%	NaN	41903.500000	39.000000	9.000000	0.000000	1.000000	NaN
75%	NaN	53469.500000	48.000000	12.000000	0.000000	2.000000	NaN
max	NaN	236961.000000	62.000000	21.000000	3.000000	6.000000	NaN

The above figure shows the statistical measures associated with the continuous variables as well as the categorical ones. The description of the same is shown below

SALARY

Here, the Salary is basically the Employee salary
Average Salary is 47729.172018 units with standard deviation of 23418.668531 units
The Salary range is between 1322.00 units (Minimum) and 236961.00 units (Maximum)
The median of the Commission category is 41903.50 units
Total count is 872

AGE

Here, the Age described is basically Age in years
Average Age is 39.955275 units with standard deviation of 10.551675 units
The Age range is between 20.00 units (Minimum) and 62.00 units (Maximum)
The median of the Age category is 39.00 units
Total count is 872

EDUCATION

Here, the Education is basically Years of formal education

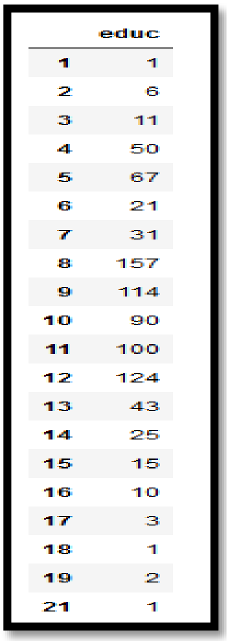
Average Education is 9.307339 years with standard deviation of 3.036259 years

The Education range is between 1 year (Minimum) and 21 years (Maximum)

The median of the Education category is 9 years

Total count is 872

Even though education is first interpreted as integer, but it is actually discrete in nature, because of the categories of each year type as far as formal education is concerned(refer the table on the right), which is why we won't be considering it in the numerical type.

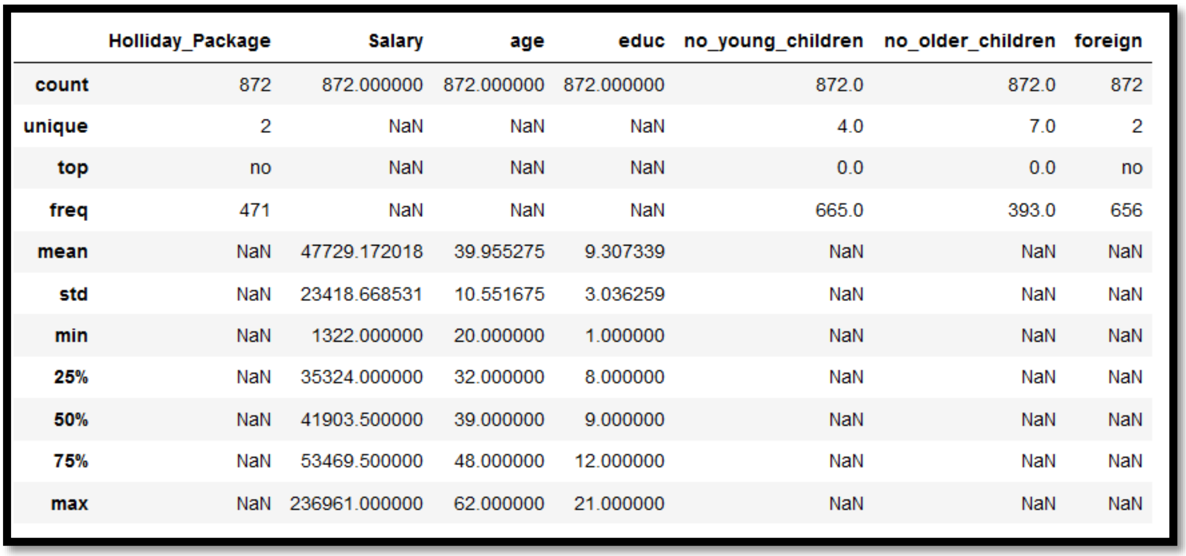


	educ
1	1
2	6
3	11
4	50
5	67
6	21
7	31
8	157
9	114
10	90
11	100
12	124
13	43
14	25
15	15
16	10
17	3
18	1
19	2
21	1

But we won't be converting it into object and later do encoding because the classes are already represented in number. Rather we will keep it as it is, for the efficiency of the process.

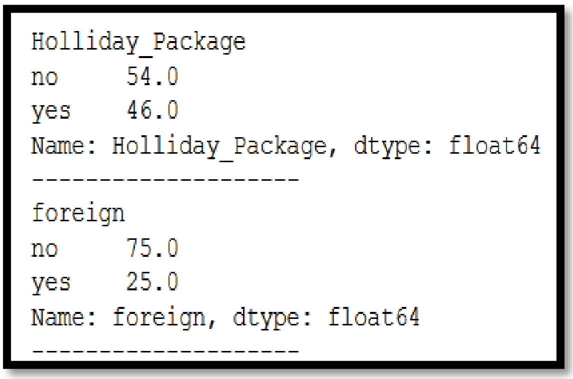
Moreover the two variables (No. of Young children, No. of older children) are having discrete values and having categorical expression, which is why they are required to be converted into categorical type using .astype('object') function. Hence the no. of object data type will increase to 4.We will observe the same in the modified description part.

<u># Check for modified Summary stats- Table 5</u>

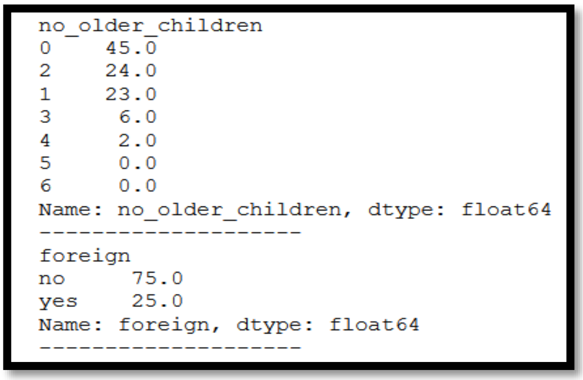


	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
count	872	872.000000	872.000000	872.000000	872.0	872.0	872
unique	2	NaN	NaN	NaN	4.0	7.0	2
top	no	NaN	NaN	NaN	0.0	0.0	no
freq	471	NaN	NaN	NaN	665.0	393.0	656
mean	NaN	47729.172018	39.955275	9.307339	NaN	NaN	NaN
std	NaN	23418.668531	10.551675	3.036259	NaN	NaN	NaN
min	NaN	1322.000000	20.000000	1.000000	NaN	NaN	NaN
25%	NaN	35324.000000	32.000000	8.000000	NaN	NaN	NaN
50%	NaN	41903.500000	39.000000	9.000000	NaN	NaN	NaN
75%	NaN	53469.500000	48.000000	12.000000	NaN	NaN	NaN
max	NaN	236961.000000	62.000000	21.000000	NaN	NaN	NaN

# Check for categorical variables- Table 6



```
Holliday_Package
no      54.0
yes     46.0
Name: Holliday_Package, dtype: float64
--------------------
foreign
no      75.0
yes     25.0
Name: foreign, dtype: float64
--------------------
```



```
no_older_children
0      45.0
2      24.0
1      23.0
3       6.0
4       2.0
5       0.0
6       0.0
Name: no_older_children, dtype: float64
--------------------
foreign
no      75.0
yes     25.0
Name: foreign, dtype: float64
--------------------
```

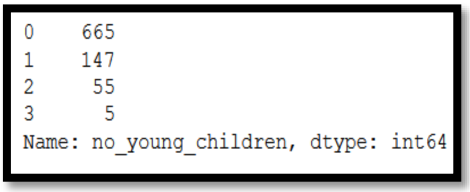
NO. OF YOUNG CHILDREN

Here, the no. of Young children is basically the number of children younger than 7 years

This is actually categorical variable present in the dataset as discrete data type.

Total. No of categories are 4-

- No young children present-0 : 665 employees
- No young children present-1 : 147 employees
- No young children present-2 : 55 employees
- No young children present-3 : 5 employees



```
0    665
1    147
2     55
3      5
Name: no_young_children, dtype: int64
```

Total count is 872

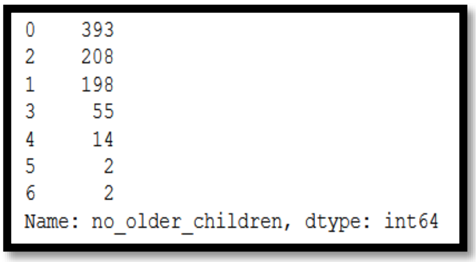
NO. OF OLDER CHILDREN

No. of older children is basically the number of older children with age>7 years

This is actually categorical variable present in the dataset as discrete data type.

Total. No of categories are 7-

- No older children present-0 : 393 employees
- No older children present-1 : 198 employees
- No older children present-2 : 208 employees
- No older children present-3 : 55 employees
- No older children present-4 : 14 employees
- No older children present-5 : 2 employees
- No older children present-6 : 2 employees

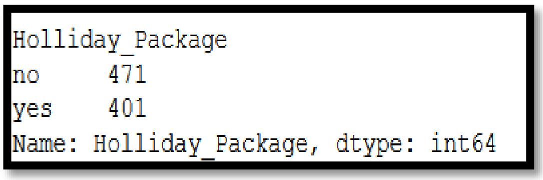


```
0    393
2    208
1    198
3     55
4     14
5      2
6      2
Name: no_older_children, dtype: int64
```

Total count is 872

HOLLIDAY_PACKAGE (Target variable)

Here Holiday Package is actually tells whether opted for Holiday Package or not (yes/no)
It is the target variable in this business problem.
It is a binary data type with values in categorical form-Yes (46%) and No (54%).
Total count is 872



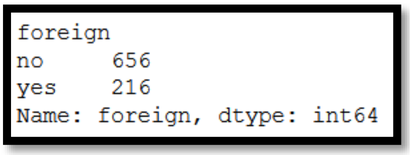
```
Holliday_Package
no     471
yes    401
Name: Holliday_Package, dtype: int64
```

FOREIGN

Here Foreign variable actually tells whether the individual is a foreigner or not (yes/no)

It is a binary data type with values in categorical form-Yes (25%) and No (75%).

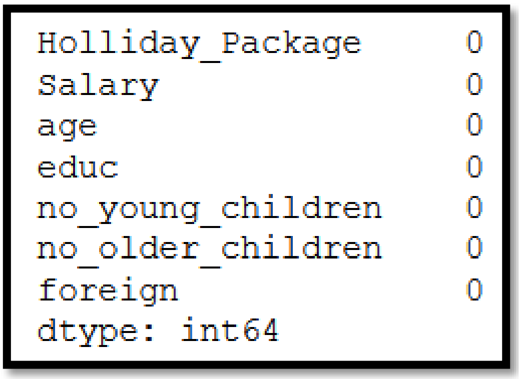
Total count is 872



```
foreign
no     656
yes    216
Name: foreign, dtype: int64
```

NULL OR MISSING VALUES

#Check for the Null Values- Table 7



```
Holliday_Package     0
Salary               0
age                  0
educ                 0
no_young_children    0
no_older_children    0
foreign              0
dtype: int64
```

The missing values or 'NaN' or 'NA' values are in general to be cleaned during the data cleaning process.

They are basically –

1. Removed or dropped
2. Replaced with mode function (Categorical variables)
3. Replaced with Median function (Continuous variables)
4. Replace with other processes

In this dataset, no missing value is present.

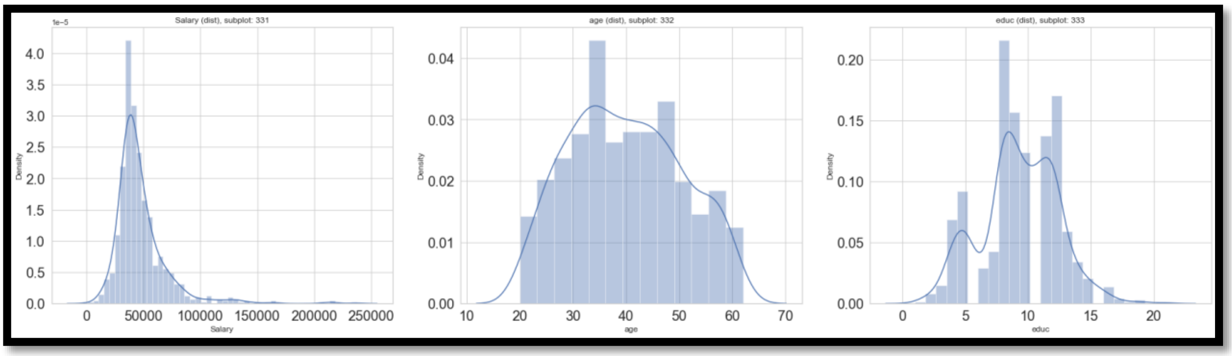
DUPLICATE VALUES

There are no duplicate values in this dataset.

OUTLIERS

We will check with the outliers in the boxplot of continuous variables while performing EDA.

Distribution plots(histogram) or similar plots for the continuous columns. - Figure 1



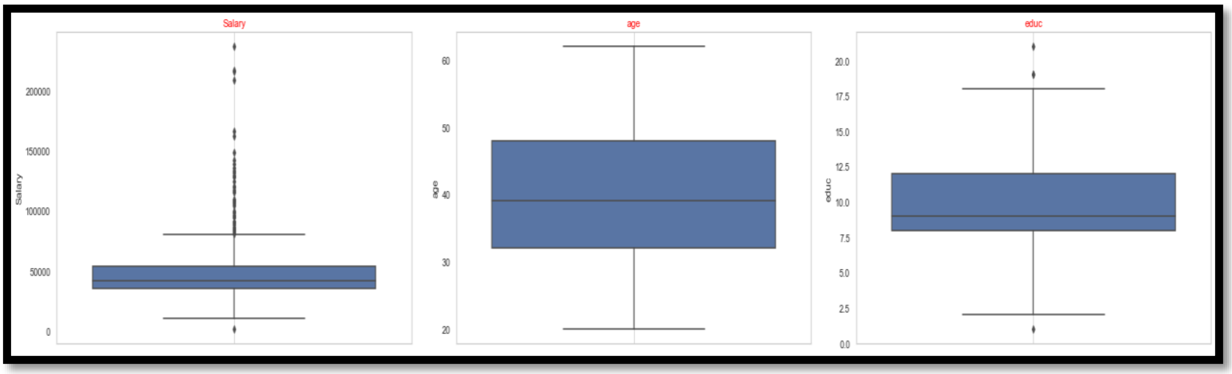
After plotting the histogram plots for all the numerical variables, the following can be concluded-

- Salary - For Salary, a high frequency/density can be seen for the salary range between 0-100000, and with positive skewness.

- Age - For Age, we can see, that the distribution is more or less normally distributed. The distribution is dense in between the minimum and maximum values

- Education- For Education, we should observe the count plot to be specific, in order to understand the categories with the highest and lowest frequencies, which is being shown and explained later.

#Check for Box plots for the continuous columns- Figure 2



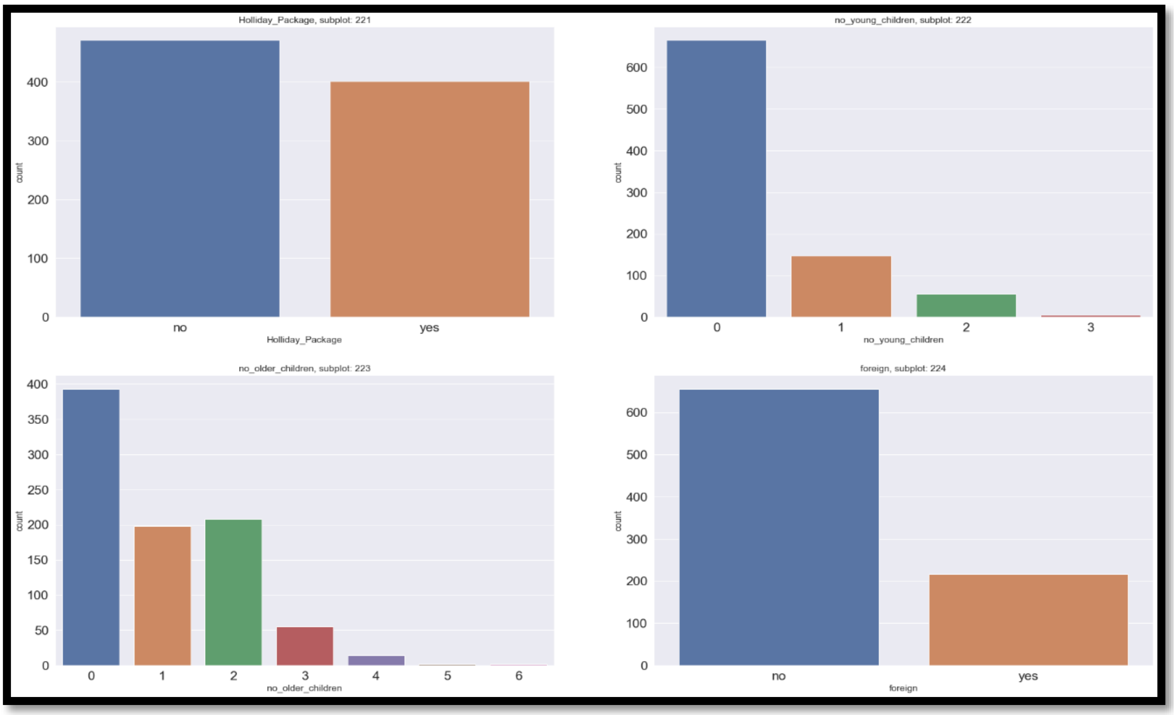
From the above Box plots, the following facts can be concluded-

- For the Salary column we can see, the median is majorly around the salary amount of 50000.Presence of outliers can be observed mostly towards higher side (beyond 85000 units) and a few below 5000 units approx.

- For the Age column we can see, the median is towards the age of 40 years. Outliers are not observed here.

- We won't be considering the outliers in the Education attribute as these are nothing but the categories with Extreme values. We will explore the Education while doing the count plot on the same.

#Check for Bar plots for the categorical columns- Figure 3

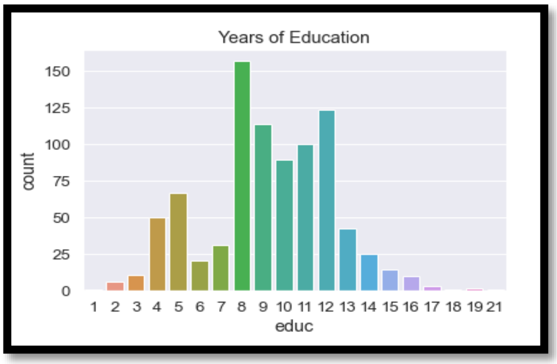


The conclusions that can be drawn from the above bar plots are-

- In the first bar graph, we can observe different categories of 'Holiday Package' feature. Most of the employees have not opted for the package (No: 54%, Yes: 46%). This is the target variable.
- In the second bar graph, we can observe different categories of 'No. of Young Children' feature. Most frequent are the employees with no younger children (age<7), followed by those with single young child, with least frequency with those employees having at least 3 younger children.
- In the third bar graph, we can observe different categories of 'No. of older Children' feature. Most frequent are the employees with no older child, followed by those with 2 older children, with least being those employees having at least 6 older children.
- In the fourth bar graph, we can observe different categories of 'Foreign' feature. Most of the employees are not foreigners (No: 75%, Yes: 25%)

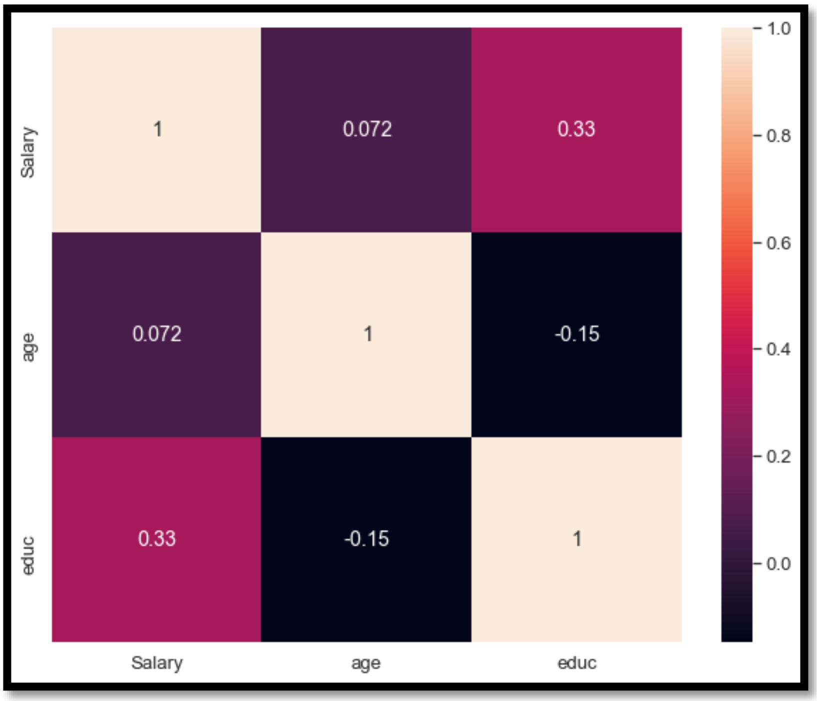
#Check for Count plot for the Education column- Figure 4

Like previously mentioned, we have used here the countplot to understand the structure of the column Education.
As clearly shown on the graph in the right, majority of the Employees have 8 years of formal education, followed by those with 12 years of education.
The least no. of employees have 1 year/18 years/21 years of formal education. This can be referred to the description of the Education Variable (pg-27)



<u># Check the heatmap with only continuous variables- Figure 5</u>

Heatmap or Corrleation plot is basically being used to evaluate the relationship between the different numeric variables within a dataset



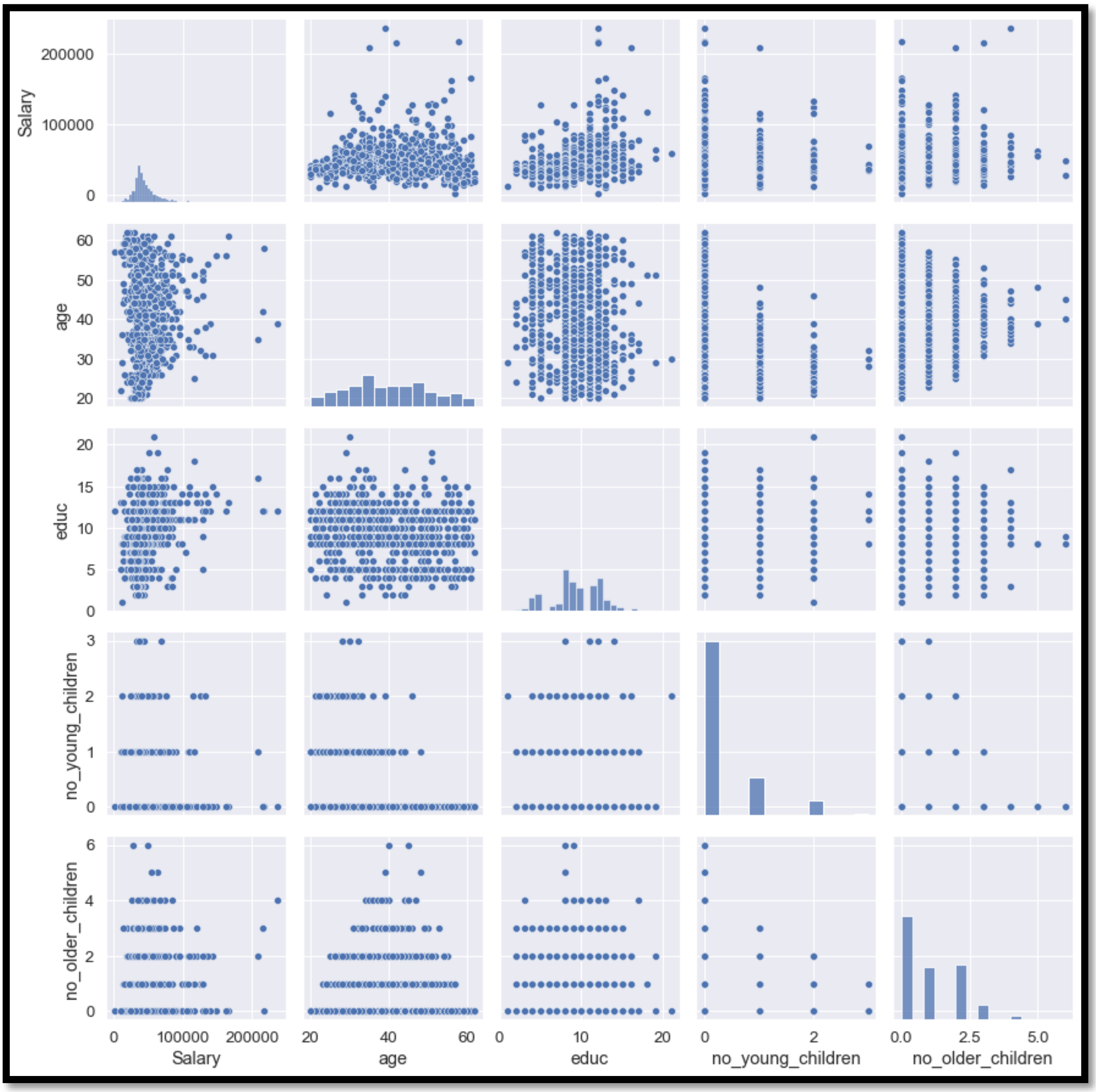
From the above Correlation plots using Heatmap, the following facts can be concluded-

None of the continuous variables have high correlation, which means no significant mutlicollinearity is observed.

We will check correlation once again after converting all the categorical variables into numeric using label encoding and normal .astype() function

--

<u># Check with the pair plots for bi variate data- Figure 6</u>



From the above Pair plots / Scatter plots, the following facts can be concluded-

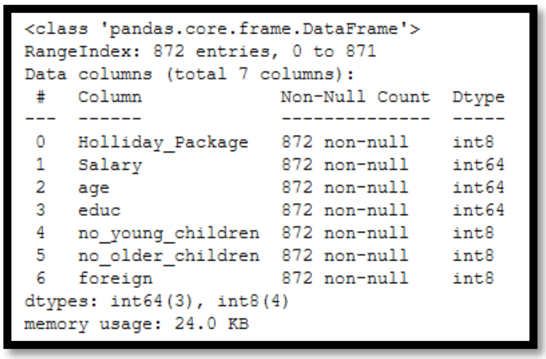
There is no correlation that can be observed, with respect to continuous and discrete variables.

****

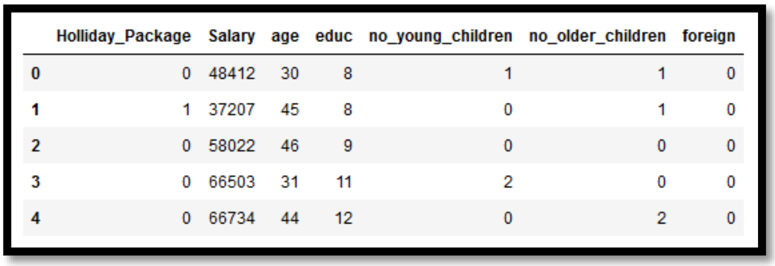
2.2 Do not scale the data. Encode the data (having string values) for Modelling. Data Split: Split the data into train and test (70:30). Apply Logistic Regression and LDA (linear discriminant analysis). Object data should be converted into categorical/numerical data to fit in the models.

#Converting Categorical variables into numeric ones

Before splitting the data into test and train, we first converted all the categorical attributes into numeric ones using pd.Categorical().codes.



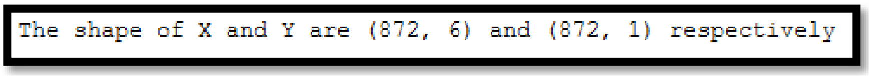
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 872 entries, 0 to 871
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Holliday_Package   872 non-null    int8
 1   Salary             872 non-null    int64
 2   age                872 non-null    int64
 3   educ               872 non-null    int64
 4   no_young_children  872 non-null    int8
 5   no_older_children  872 non-null    int8
 6   foreign            872 non-null    int8
dtypes: int64(3), int8(4)
memory usage: 24.0 KB
```



	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
0	0	48412	30	8	1	1	0
1	1	37207	45	8	0	1	0
2	0	58022	46	9	0	0	0
3	0	66503	31	11	2	0	0
4	0	66734	44	12	0	2	0

#Splitting the dataset into Feature and Target variable

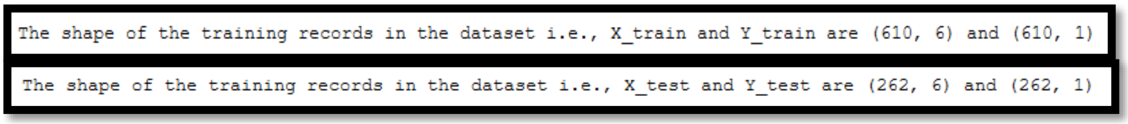
After converting the categorical ones into numeric, we then have separated the features and the target variables and stored them as 'X' and 'Y' respectively.



```
The shape of X and Y are (872, 6) and (872, 1) respectively
```

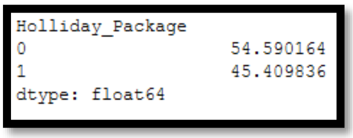
#Splitting the data into train and test set

After the above steps, the data is being split into train and test using the train_test_split function by a ratio of 70:30



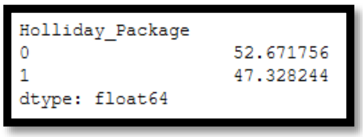
```
The shape of the training records in the dataset i.e., X_train and Y_train are (610, 6) and (610, 1)
```

```
The shape of the training records in the dataset i.e., X_test and Y_test are (262, 6) and (262, 1)
```



```
Holliday_Package
0                    54.590164
1                    45.409836
dtype: float64
```

Train Data Class



```
Holliday_Package
0                    52.671756
1                    47.328244
dtype: float64
```

Test Data Class

LOGISTIC REGRESSION

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Some of the examples of classification problems are Email spam or not spam, Online transactions Fraud or not Fraud

Logistic regression transforms its output using the logistic sigmoid function to return a probability value.

Logistic Regression is thus a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and is based on the concept of probability.

What are the types of logistic regression

- Binary (eg. Tumor Malignant or Benign)
- Multi-linear functions failsClass (eg. Cats, dogs or Sheep's)

#Steps involved

Using the train dataset, we have created Logistic model and then further testing the same on the test dataset.

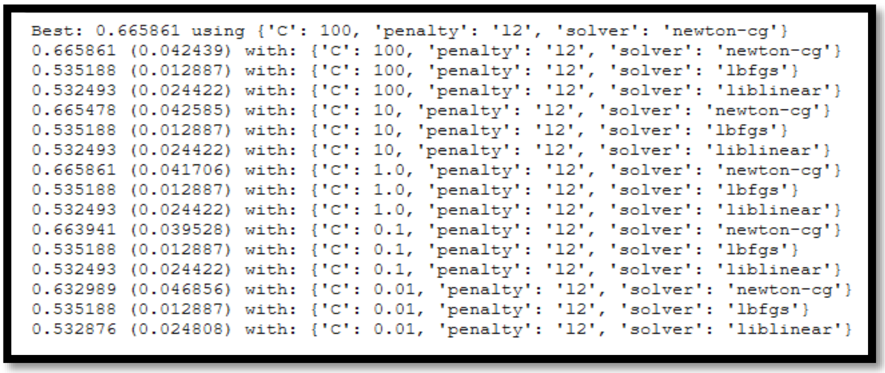
For the Sigmoid formation, we have imported the LogisticRegression module from the ‘sklearn’ package

With the help of afore mentioned package, Logistic regression model is created, in order to fit the training data into this model.

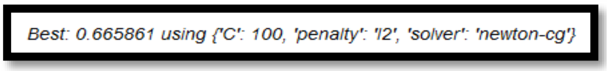
In the following step, we have used Grid Search CV method to evaluate the best parameters for the model in order to get a better performed Logistic Regression.

GRID Search CV-

GridSearchCV, can be briefly defined as a library function that is a member of sklearn's model_selection package. **It helps to loop through predefined hyper parameters and fit your estimator (model) on your training set**. So, in the end, one can select the best parameters from the listed hyper parameters.



```
Best: 0.665861 using {'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'}
0.665861 (0.042439) with: {'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'}
0.535188 (0.012887) with: {'C': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
0.532493 (0.024422) with: {'C': 100, 'penalty': 'l2', 'solver': 'liblinear'}
0.665478 (0.042585) with: {'C': 10, 'penalty': 'l2', 'solver': 'newton-cg'}
0.535188 (0.012887) with: {'C': 10, 'penalty': 'l2', 'solver': 'lbfgs'}
0.532493 (0.024422) with: {'C': 10, 'penalty': 'l2', 'solver': 'liblinear'}
0.665861 (0.041706) with: {'C': 1.0, 'penalty': 'l2', 'solver': 'newton-cg'}
0.535188 (0.012887) with: {'C': 1.0, 'penalty': 'l2', 'solver': 'lbfgs'}
0.532493 (0.024422) with: {'C': 1.0, 'penalty': 'l2', 'solver': 'liblinear'}
0.663941 (0.039528) with: {'C': 0.1, 'penalty': 'l2', 'solver': 'newton-cg'}
0.535188 (0.012887) with: {'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'}
0.532493 (0.024422) with: {'C': 0.1, 'penalty': 'l2', 'solver': 'liblinear'}
0.632989 (0.046856) with: {'C': 0.01, 'penalty': 'l2', 'solver': 'newton-cg'}
0.535188 (0.012887) with: {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
0.532876 (0.024808) with: {'C': 0.01, 'penalty': 'l2', 'solver': 'liblinear'}
```



```
Best: 0.665861 using {'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'}
```

Using the above mentioned GridSearchCV package, we have identified the best parameters and an Optimised Logistic Regression model is being built after doing few iterations with the values we have received in each step. But even with this the model couldn’t generate better accuracy, precision and recall. We will discuss about the same in the next questions

Note – Kindly refer the code file for the steps involved in the CART formation.

LDA

Linear Discriminant Analysis or **Normal Discriminant Analysis** or **Discriminant Function Analysis** is a dimensionality reduction technique that is commonly used for supervised classification problems. It is used for modelling differences in groups i.e. separating two or more classes. It is used to project the features in higher dimension space into a lower dimension space.

For example, we have two classes and we need to separate them efficiently. Classes can have multiple features. Using only a single feature to classify them may result in some overlapping as shown in the below figure. So, we will keep on increasing the number of features for proper classification.

Two criteria are used by LDA to create a new axis:

- Maximize the distance between means of the two classes.
- Minimize the variation within each class.

Extensions to LDA:

1. **Quadratic Discriminant Analysis (QDA):** Each class uses its own estimate of variance (or covariance when there are multiple input variables).
2. **Flexible Discriminant Analysis (FDA):** Where non-linear combinations of inputs are used such as splines.

3. **Regularized Discriminant Analysis (RDA):** Introduces regularization into the estimate of the variance (actually covariance), moderating the influence of different variables on LDA.

#Steps involved

First, we have used the train dataset to create the `Linear Discriminants` and to test on the test dataset.

For the Random Forest formation, we have imported the `LinearDiscriminantAnalysis` module from the 'sklearn' package

With the help of afore mentioned package, LDA model is created, in order to fit the training data into this model.

A LDA model is thus being created which is being further used for model performance evaluation.

QDA

We have also used the QDA method, to improve the performance of the dataset.

The codes for the same is being attached with the Jupyter notebook (Logistic + LDA)

For the model evaluation and comparison, we have stated the same in the later half of the question, where model performances are being compared.

****

2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model Final Model: Compare Both the models and write inference which model is best/optimized.

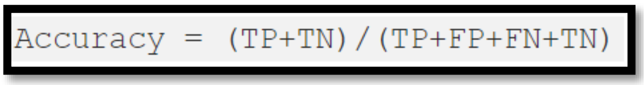
After doing all the necessary steps for model fitting and generating predictions, we came to the very outcome i.e, the part of model evaluation.

ACCURACY

It is a part of metrices derived from confusion matrix which is basically a **NxN** matrix, where **N** is the **number of classes to be predicted**

It is the **proportion** of the total number of **predictions** that were **correct**.

It is easily suited for **binary** as well as a **multiclass classification problem** which are **well balanced** and **not skewed** or **No class imbalance**.



```
Accuracy = (TP+TN)/(TP+FP+FN+TN)
```

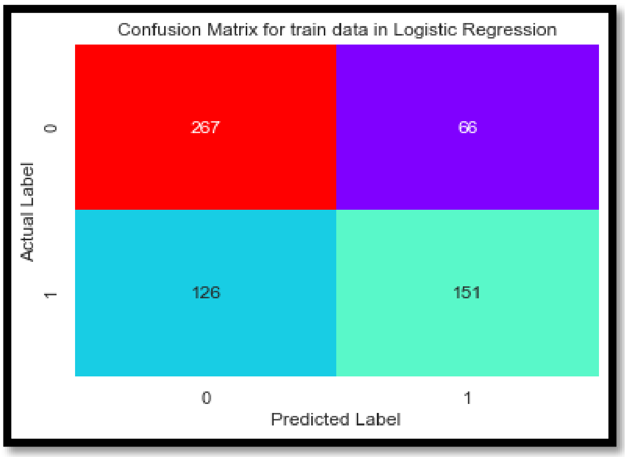
#Train and Test accuracies for each model – Table - 8

Model Types	Train Data Accuracy	Test Data Accuracy
Logistic Regression(Grid Search)	**0.6852**	**0.6335**
LDA	**0.6721**	**0.6259**
QDA	**0.6803**	**0.6450**

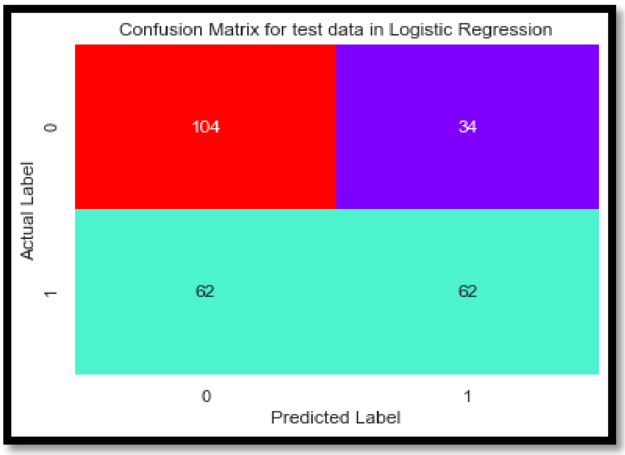
As per the model validation is concerned, we can observe that once after using the Hyper Tuning Parameters, through Grid Search Cross Validation, the models show generalisation, which means they are no longer Under fitted or Over fitted.

#Confusion Matrix for Each Model- Figure 7

Logistic Reg. Confusion Matrix- Train

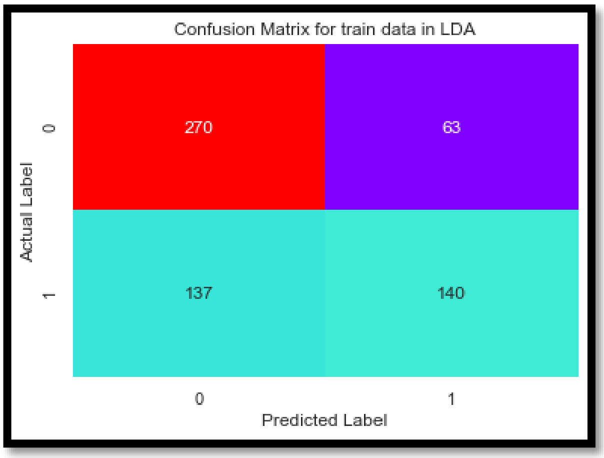


Logistic Reg. Confusion Matrix - Test

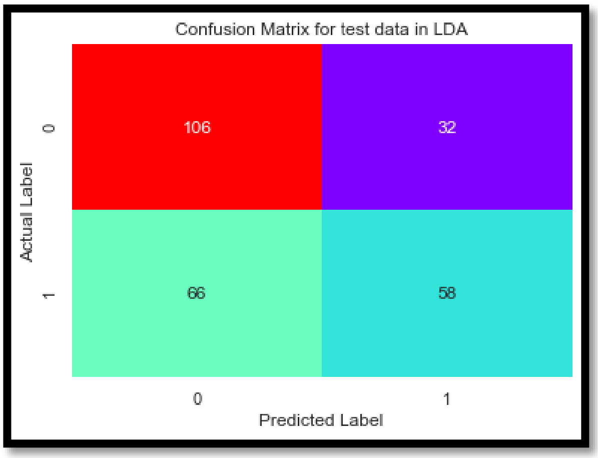


- **Confusion Matrix from Train data**- Majority of the data belong to True positives and True Negatives as compared to the False ones, which are responsible for Moderate Accuracy. But overall, False positives and False negatives are very high, which are the causes of concern here, due to which the metrics are not well performed.
- **Confusion Matrix from Test data-** Even though the majority of the data shows the reasons of moderate accuracy, but there has been considerable increase in False Negatives and Positives, which brings down the accuracy score, precision and recall.

LDA Confusion Matrix- Train Data



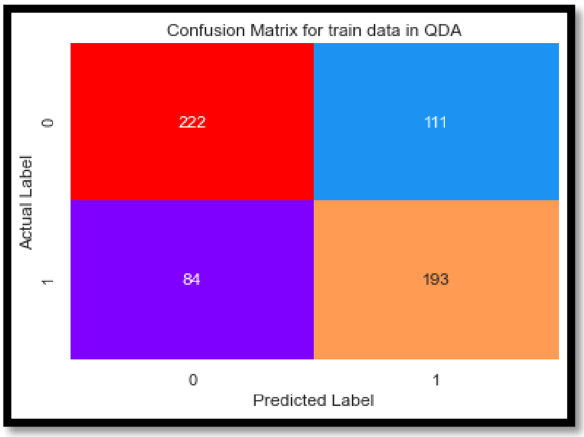
LDA Confusion Matrix - Test Data



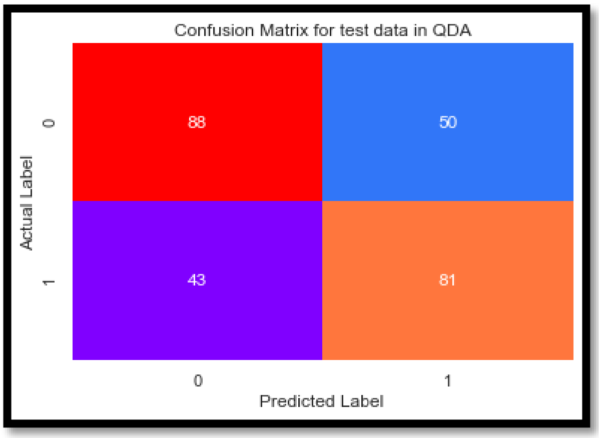
- **Confusion Matrix from Train data**- Similar to logistic regression, majority of the data belong to True positives and True Negatives as compared to the False ones, which are responsible for Moderate Accuracy. But overall, False positives and False negatives are very high, which are the causes of concern here, due to which the metrics are not well performed.

- **Confusion Matrix from Test data-** Even though the majority of the data shows the reasons of moderate accuracy, but there has been considerable increase in False Negatives and Positives, which brings down the accuracy score, precision and recall.

QDA Confusion Matrix- Train Data



QDA Confusion Matrix - Test Data



- **Confusion Matrix from Train data**- Similar to logistic regression and LDA, majority of the data belong to True positives and True Negatives as compared to the False ones, which are responsible for Moderate Accuracy. But overall, False positives and False negatives are very high, which are the causes of concern here, due to which the metrics are not well performed.

- **Confusion Matrix from Test data-** Even though the majority of the data shows the reasons of moderate accuracy, but there has been considerable increase in False Negatives and Positives, which brings down the accuracy score, precision and recall.

ROC AUC Score

This is again one of the popular metrics used in the industry.

The ROC (**Receiver operating characteristic**) curve is the plot between **sensitivity** and (**1-specificity**). (**1- specificity**) is also known as **false positive rate** and **sensitivity** is also known as **True Positive rate**.

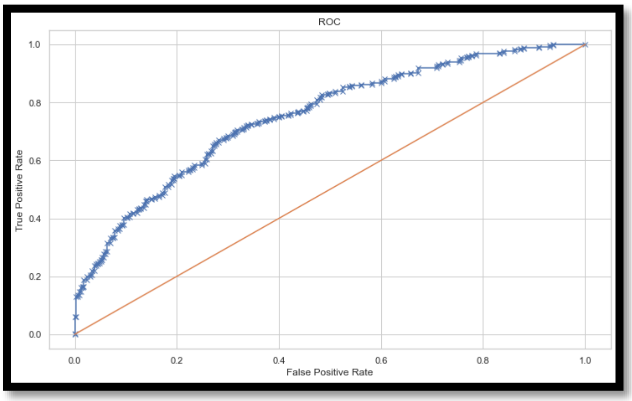
The biggest advantage of using ROC curve is that it is **independent** of the **change in proportion of responders.**

#ROC -AUC Score for each model- Table -9

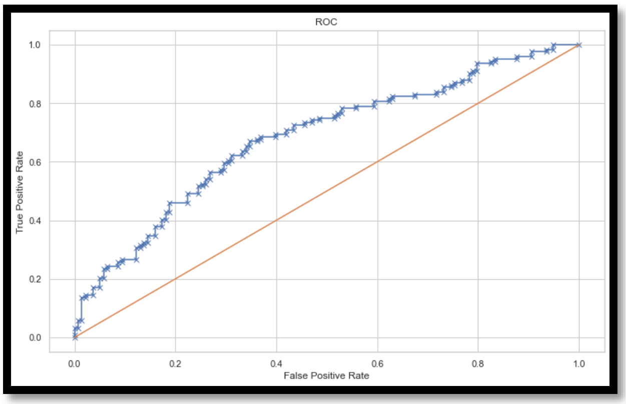
Model Types	Train Data ROC AUC Score	Test Data ROC AUC Score
Logistic Regression(Grid Search)	**0.7509**	**0.6823**
LDA	**0.7526**	**0.6809**
QDA	**0.7483**	**0.6867**

#ROC -AUC Curve for each model – Figure 8

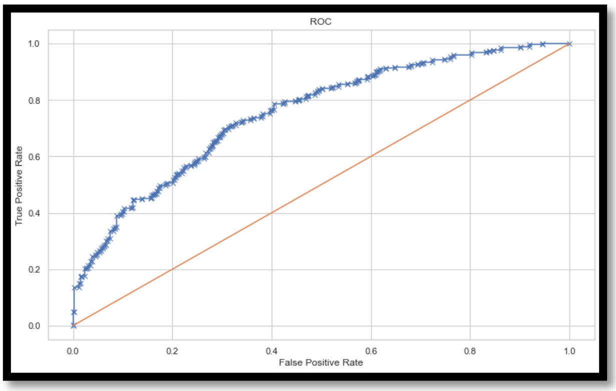
Logistic Reg. AUC Curve- Train



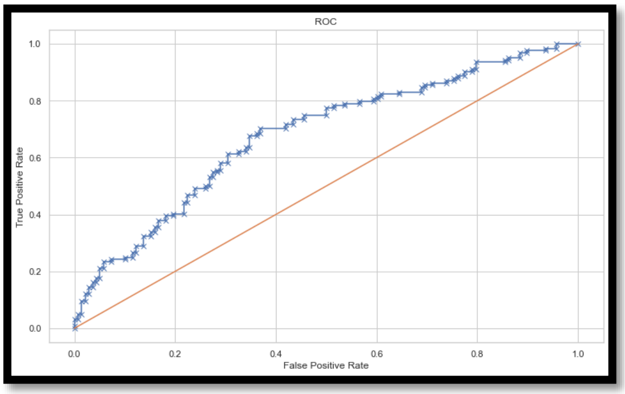
Logistic Reg. AUC Curve- Test



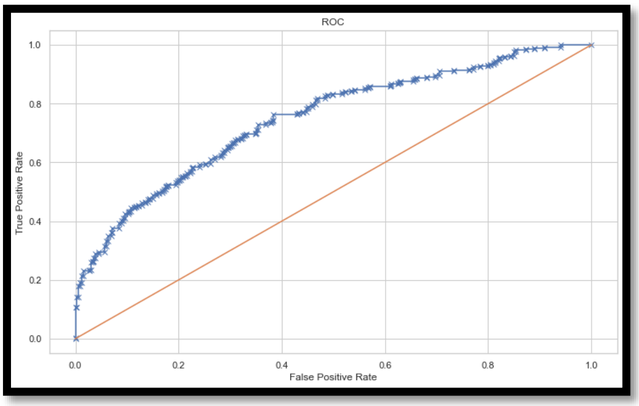
LDA ROC AUC Curve- Train Data



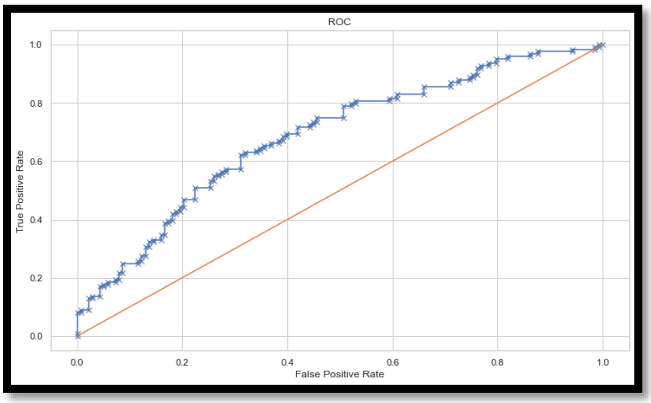
LDA ROC AUC Curve- Test Data



QDA ROC AUC Curve- Train Data



QDA ROC AUC Curve- Test Data



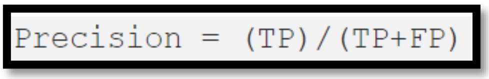
PRECISION

Similar to the Accuracy, it is a metric derived from confusion matrix

Positive Predictive Value or Precision is also defined as the proportion of positive cases that were correctly identified.

In other words, it determines the proportion of **predicted Positives** which is truly Positive

Precision is a valid choice of evaluation metric when we want to be very sure of our prediction.



```
Precision = (TP)/(TP+FP)
```

#Train and Test Precision for each model- Table -10

Model Types	Train Data Precision	Test Data Precision
Logistic Regression(Grid Search)	**0.71**	**0.65**
LDA	**0.69**	**0.64**
QDA	**0.63**	**0.62**

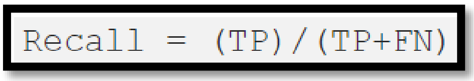
RECALL

Similar to the Precision, it is a metric derived from confusion matrix.

Sensitivity or Recall is also defined as the proportion of actual positive cases which are correctly identified.

In other words, it determines the proportion of **actual Positives** is correctly classified

Recall is a valid choice of evaluation metric when we want to capture as many positives as possible.



```
Recall = (TP)/(TP+FN)
```

#Train and Test Recall for each model- Table -11

Model Types	Train Data Recall	Test Data Recall
Logistic Regression(Grid Search)	**0.55**	**0.50**
LDA	**0.51**	**0.47**
QDA	**0.70**	**0.65**

F1 SCORE

Similar to the Precision and Recall, it is a metric derived from confusion matrix.
The **F1 score** is a **number between 0** and **1** and is the **harmonic mean** of **precision** and **recall** values for a **classification** problem.
F1 score sort of maintains a balance between the precision and recall for your classifier. If your precision is low, the F1 is low and if the recall is low again your F1 score is low.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

#Train and Test F1 score for each model- Table -12

Model Types	Train Data F1 Score	Test Data F1 Score
Logistic Regression(Grid Search)	**0.61**	**0.56**
LDA	**0.58**	**0.54**
QDA	**0.66**	**0.64**

****

#Train and Test scores for each model- Table -13

		Accuracy	Auc_Roc_Score	Precision	Recall	F1 Score
Logistic Regression(Grid Search)	Train	68%	75%	71%	55%	61%
	Test	63%	68%	65%	50%	56%
LDA	Train	67%	75%	69%	51%	58%
	Test	62%	68%	64%	47%	54%
QDA	Train	68%	74%	63%	70%	66%
	Test	64%	68%	62%	65%	64%
QDA is considered to be the best model among these three, followed by the Logistics regression. Although, the model is not well classified, due to presence of high amount of False interpretation of the data.						

2.4 Inference: Basis on these predictions, what are the insights and recommendations.

Please explain and summarise the various steps performed in this project. There should be proper business interpretation and actionable insights present.

INSIGHTS

From the above comparable performance metrics or evaluators we can observe, QDA is the more or less a better model followed by the Logistic Regression, although we have to be concerned about the dataset provided to us.

Reasons

For both the Train and Test datasets-

- The Precision is better in Logistics than QDA.
- Accuracy is better than other models
- AUC-ROC score is better than other models
- Though Recall and F1 Scores are moderate here

Recommendations

All the three models are more or less similar in terms or performance, barring a few metrices.

But overall, due to high amount of misinterpretation of the classes, the performance of the metrics are not in the acceptable range.

Which is why, it is recommended to cross-check the dataset thoroughly.

Look for any data entry error, which have given rise to such insufficiency, inaccuracy, etc.

It is necessary to make use of proper resources, to avoid such misinterpretation or overlapping of data values in classes.

****

Reference

- *https://github.com/krishnaik06/Feature-Selection-techniques/blob/master/Feature%20Selection.ipynb*
- *https://scikit-learn.org/stable/modules/preprocessing.html*
- *https://towardsdatascience.com/what-and-why-behind-fit-transform-vs-transform-in-scikit-learn-78f915cf96fe*
- *https://www.investopedia.com/terms/r/r-squared.asp#:~:text=R%2Dsquared%20(R2),variables%20in%20a%20regression%20model.*
- *https://www.geeksforgeeks.org/python-coefficient-of-determination-r2-score/#:~:text=R2%20%3D%201%2D%20600%2F200%20%3D%20%2D2&text=metrics%20in%20Python%20to%20compute%20R2%20score.*
- *https://www.statisticshowto.com/probability-and-statistics/regression-analysis/*
- *https://machinelearningmastery.com/hyperparameters-for-classification-machine-learning-algorithms/*
- *https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html*
- *https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RepeatedStratifiedKFold.html*
- *https://adataanalyst.com/data-analysis-resources/visualise-categorical-variables-in-python/#:~:text=Visualise%20Categorical%20Variables%20in%20Python%20using%20Bivariate%20Analysis,a%20pre%2Ddefined%20significance%20level.*
- *https://www.investopedia.com/terms/v/variance-inflation-factor.asp#:~:text=Variance%20inflation%20factor%20(VIF)%20is,only%20that%20single%20independent%20variable.*