PREDICTIVE MODELING

Cubic Zirconia Manufacturer Analysis - Linear Regression

Tour & Travel Agency Analysis - Logistic Regression & LDA

Cubic Zirconia Manufacturer Analysis - Linear Regression

Table of Contents

List of Tables……3

List of Figures……3

Problem Statement……4

Questions:

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……..4

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 an ordinal variables and take actions accordingly. Explain why you are combining these sub levels with appropriate reasoning……16

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……19

1.4 Inference: Basis on these predictions, what are the business insights and recommendations……44

Tabel List:

Table 1 : Sample Data 5
Table 2 : Descriptive Statistics.....6
Table 3 : Strongest Correlation Variables.....15
Table 4 : Moderate Correlation Variables.....15
Table 5 : Poor & negative Correlation Variables.....16
Table 6 : Sub grouped table (categorical columns).....17
Table 7 : Descriptive Statistics (cut with price).....18
Table 8 : Descriptive Statistics (color with price).....18
Table 9 : Descriptive Statistics (clarity with price) 19
Table 10 : Encoded table (categorical columns) 20
Table 11 : Model - 1 OLS stats table 26
Table 12 : Model - 2 OLS stats table 30
Table 13 : Model - 3 OLS stats table 35
Table 14 : Model - 4 OLS stats table 40
Table 15 : Model Comparison (4 models) 42

List of Figure:

Fig 1 : Box & hist Plot – Univariant Analysis (Carat) 7
Fig 2 : Box & hist Plot – Univariant Analysis (depth).....7
Fig 3 : Box & hist Plot – Univariant Analysis (table) 8
Fig 4 : Box & hist Plot – Univariant Analysis (x) 9
Fig 5 : Box & hist Plot – Univariant Analysis (y) 9
Fig 6 : Box & hist Plot – Univariant Analysis (z) 10
Fig 7 : Box & hist Plot – Univariant Analysis (price) 10
Fig 8 : Count Plot - Univariant Analysis (cut) 11
Fig 9 : Count Plot - Univariant Analysis (color) 11
Fig 10 : Count Plot - Univariant Analysis (clarity) 11
Fig 11 : Scatter Plot - Bivariant Analysis (continuous columns) 12
Fig 12 : Count Plot - Bivariant Analysis (Categorical columns) 12
Fig 13 : Pair Plot - Multivariant Analysis (continuous columns) 14
Fig 14 : Heatmap Plot – Correlation analysis (continuous columns) 15
Fig 15 : Box plot - After outlier treatment.....17
Fig 16 : Model 1 - Scatter Plot (Actual vs Predicted) 24
Fig 17 : Model 2 - Scatter Plot (Actual vs Predicted) 29
Fig 18 : Model 3 - Scatter Plot (Actual vs Predicted) 34
Fig 19 : Model 4 - Scatter Plot (Actual vs Predicted) 34

Problem 1A:

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.

Data Dictionary:

Variable Name	Description
Carat	Carat weight of the cubic zirconia.
Cut	Describe the cut quality of the cubic zirconia. Quality is increasing order Fair, Good, Very Good, Premium, Ideal.
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 avg price) IF, VVS1, VVS2, VS1, VS2, Sl1, Sl2, l1
Depth	The Height of cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter.
Table	The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter.
Price	the Price of the cubic zirconia.
X	Length of the cubic zirconia in mm.
Y	Width of the cubic zirconia in mm.
Z	Height of the cubic zirconia in mm.

1.1. 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.

The objectives of EDA can be summarized as follow:

- Maximize insight into the data/understand the data structure.
- EDA is an approach to analyse date using non-visual and visual techniques.
- EDA involves through analyse of data to understand the current business situation.
- EDA objective is to extract "'Gold' from the "'Data mine" based on domain understanding.

As a first step, importing all the necessary libraries, we think that will be requiring to perform the EDA.

Loading the data set – Loading the 'cubic_zirconia.csv ' file using pandas. For this we will be using read excel file.

EDA Exploration: Following outputs are retrieved from Jupyter.

Head of the dataset: After reading the CSV file, the head command option gives the below output.

	Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
0	1	0.30	Ideal	E	SI1	62.1	58.0	4.27	4.29	2.66	499
1	2	0.33	Premium	G	IF	60.8	58.0	4.42	4.46	2.70	984
2	3	0.90	Very Good	E	VVS2	62.2	60.0	6.04	6.12	3.78	6289
3	4	0.42	Ideal	F	VS1	61.6	56.0	4.82	4.80	2.96	1082
4	5	0.31	Ideal	F	VVS1	60.4	59.0	4.35	4.43	2.65	779

Table 1 : Sample Data

From the above head table, we infer that,

Dataset contains of 11 variables such as 'Unnamed: 0', 'carat', 'cut', 'color', 'clarity', 'depth', 'table', 'x', 'y', 'z', 'price'. Variable **'Unnamed: 0'** is not useful for our analysis and it will be dropped in future.

There are 10 Independent variables and 1 dependent variable as 'Price'.

Shape of the dataset : Output from shape command is –

```
The dataset has 26967 rows and 11 columns.
```

info() is used to check the Information about the data and the datatypes of each respective attributes: Output from Info command is –

```
<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
```

From Info() function we infer that, the dataset has 26967 instances with 10 attributes. 1 integer type, 6 float type and 3 object type (Strings in the column).

Duplication check : Output from duplicated with sum command is –

```
The dataset has 34 duplication.
```

Though there is no customer ID or any unique identifier to determine whether it is true duplication or not, here we are dropping the duplicate value since it is less in number compared to size of the data and this will further avoid bias in analysis.

Duplication check after dropping: Output from duplicated with sum command is –

```
The dataset has 0 duplication.
```

Shape of the dataset after dropping duplicate values : Output from shape command is –

```
The dataset has 26933 rows and 10 columns
```

Null value check : Output from isnull with sum command is –

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

Dataset contains 697 null values in variable 'depth' and that needs to be validated and imputation treatment required.

Descriptive Analytics : Describe method will help us see how data is spread for the numerical values, also we can see the minimum value, mean values, different percentile values and maximum values.

Output from Describe with Transpose option is –

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
carat	26933.0	NaN	NaN	NaN	0.79801	0.477237	0.2	0.4	0.7	1.05	4.5
cut	26933	5	Ideal	10805	NaN	NaN	NaN	NaN	NaN	NaN	NaN
color	26933	7	G	5653	NaN	NaN	NaN	NaN	NaN	NaN	NaN
clarity	26933	8	SI1	6565	NaN	NaN	NaN	NaN	NaN	NaN	NaN
depth	26236.0	NaN	NaN	NaN	61.745285	1.412243	50.8	61.0	61.8	62.5	73.6
table	26933.0	NaN	NaN	NaN	57.45595	2.232156	49.0	56.0	57.0	59.0	79.0
x	26933.0	NaN	NaN	NaN	5.729346	1.127367	0.0	4.71	5.69	6.55	10.23
y	26933.0	NaN	NaN	NaN	5.733102	1.165037	0.0	4.71	5.7	6.54	58.9
z	26933.0	NaN	NaN	NaN	3.537769	0.719964	0.0	2.9	3.52	4.04	31.8
price	26933.0	NaN	NaN	NaN	3937.52612	4022.551862	326.0	945.0	2375.0	5356.0	18818.0

Table 2: Descriptive Statistics

From the above descriptive table, we infer that,

- Among 10 attributes there are 7 features that are in numbers and remaining 3 attributes are object type.
- Variable 'cut' has 5 unique values in that 'Ideal' has the top value count of 10805, variable 'color' has 7 unique values in that 'G' has the most value count of 5653 and variable 'clarity' has 8 unique values in that 'SI1' has the most value count of 6565.
- Feature 'X' - Length of the cubic zirconia in mm, 'Y' - Width of the cubic zirconia in mm and 'Z' - Height of the cubic zirconia in mm contain min value as 0 which is invalid ie Length, width & Height value cannot be as zero, it needs to validate, and imputation/drop treatment required.

- There is a drastic change in terms of min and max values in variable 'Price', this an indication of outliers.
- Variable 'Carat' has the mean value of 0.798 with the standard deviation of 0.47 and the min Carat weight of the cubic zirconia is 0.2 to the max of 4.50.
- Variable 'depth' has the mean value of 61.75 with the standard deviation of 1.41 and the min value as 50.8 to the max of 73.60.
- Variable 'table' has the mean value of 57.45 with the standard deviation of 2.23 and the min value as 49.0 to the max of 79.0.
- There seems to be drastic difference of values in Upper and Lower range for most of the variables, this indicates the presence of outliers.

Univariant Analysis :- Histogram & Boxplot (Numeric Columns)

The objective of univariant analysis is to derive the data, define, analyze and summarize the pattern present in it. In a dataset, it explores each variable separately such as Numerical variable and Categorical variable. Some of the patterns that can be easily identified with univariant analysis are Central Tendency (mean, mode and median), Dispersion (range, variance), Quartiles (interquartile range), and Standard deviation. Univariant analysis can be described and visualize with the help of most used plots of Histogram/Distplot and Barplot.

Column : Carat (Carat weight of the cubic zirconia)

```
Skewness of carat: 1.11
Kurtosis of carat: 1.21
Outliers of carat: 6.57
```

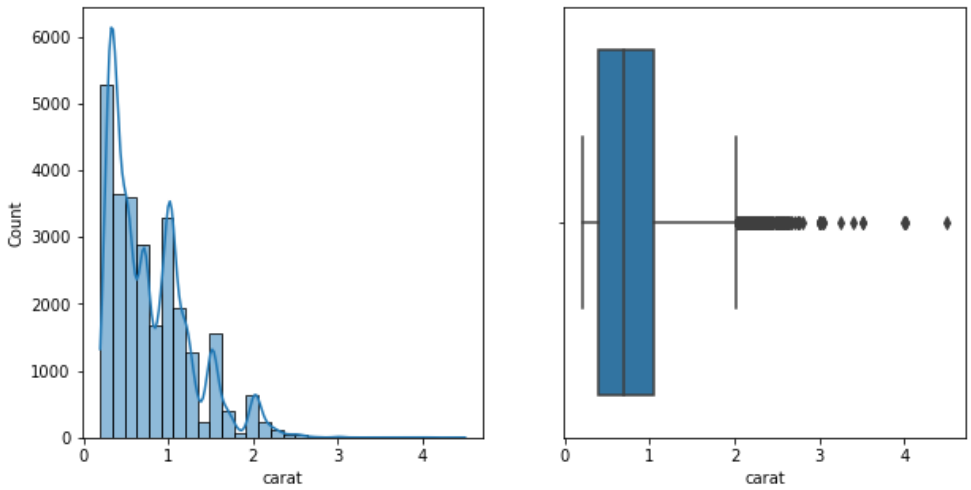


Fig 1 : Box Plot – Univariant Analysis (Carat)

Observation:

- The above graph represents, 'carat' has positive skewness i.e., to a longer on the right side of the distribution.
- The histplot shows the distribution of data ranges from 0 to 4 and has positive kurtosis.
- The boxplot of the variable 'carat' has 6.57 % of outliers but mean & median values are not far away from each other.

Column : Depth (The Height of cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter)

```
Skewness of depth: -0.03
Kurtosis of depth: 3.68
Outliers of depth: 12.19
```

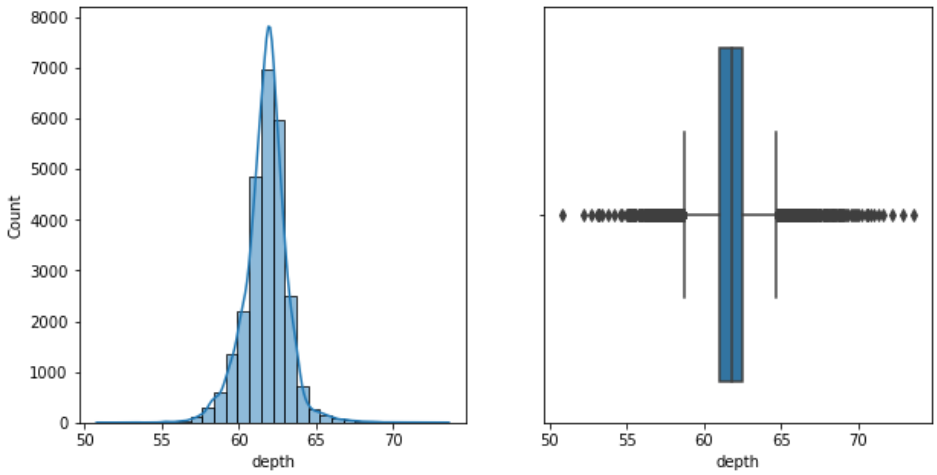


Fig 2 : Box Plot – Univariant Analysis (depth)

Observation:

- The above graph represents, 'depth' has low skewness and has almost normal distribution.
- The histplot shows the distribution of data ranges from 50 to 70 and has positive kurtosis.
- The boxplot of the variable 'carat' has 12.19 % of outliers but mean & median values are not far away from each other.

Column : Table (The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter)

```
Skewness of table: 0.77
Kurtosis of table: 1.58
Outliers of table: 3.18
```

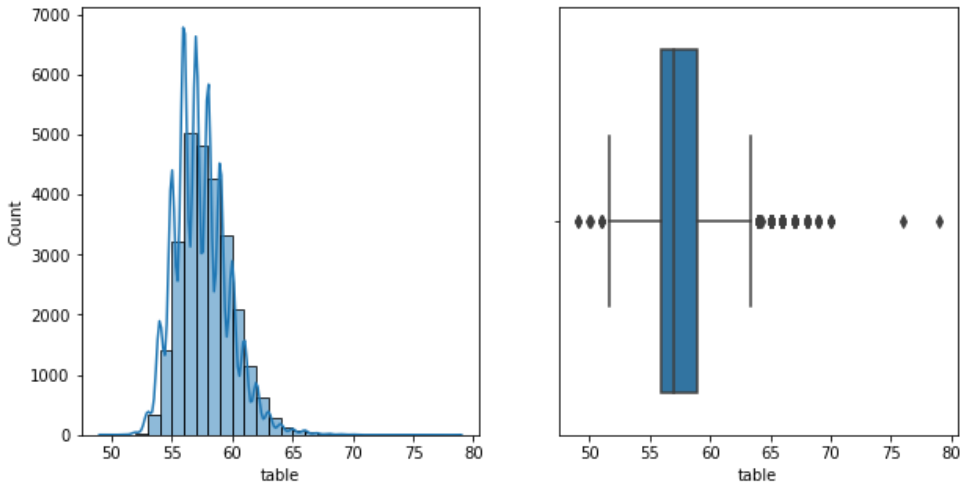


Fig 3 : Box & hist Plot – Univariant Analysis (table)

Observation:

- ❖ The above graph represents, ‘table’ has low skewness and slightly right skewed
- ❖ The histplot shows the distribution of data ranges from 50 to 80 and has positive kurtosis.
- ❖ The boxplot of the variable ‘carat’ has 3.18 % of outliers but mean & median values are not far away from each other.

Column : X (Length of the cubic zirconia in mm)

```
Skewness of x: 0.39
Kurtosis of x: -0.68
Outliers of x: 0.14
```

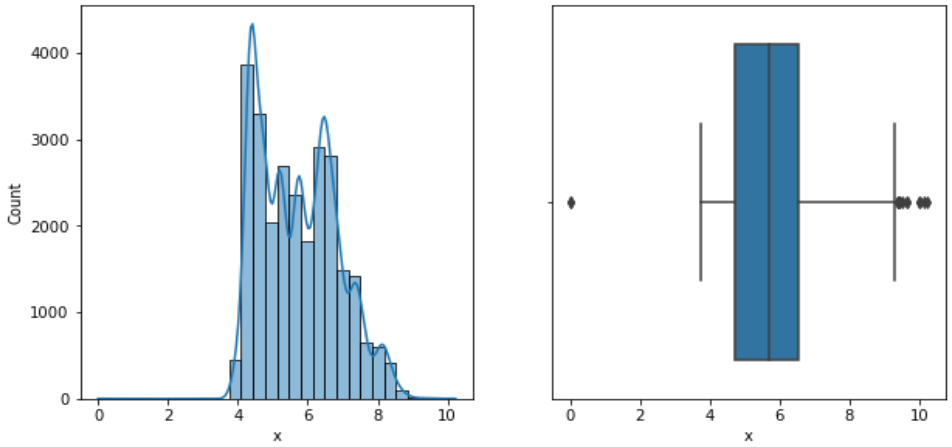


Fig 4 : Box & hist Plot – Univariant Analysis (x)

Observation:

- ❖ The above graph represents, ‘X’ has low skewness and skewed towards left.
- ❖ The histplot shows the distribution of data ranges from 0 to 10 and has very low kurtosis.
- ❖ The boxplot of the variable ‘carat’ has low outliers ie 0.14%, but mean & median values are not far away from each other.

Column : Y (Length of the cubic zirconia in mm)

```
Skewness of y: 3.87
Kurtosis of y: 160.04
Outliers of y:0.14
```

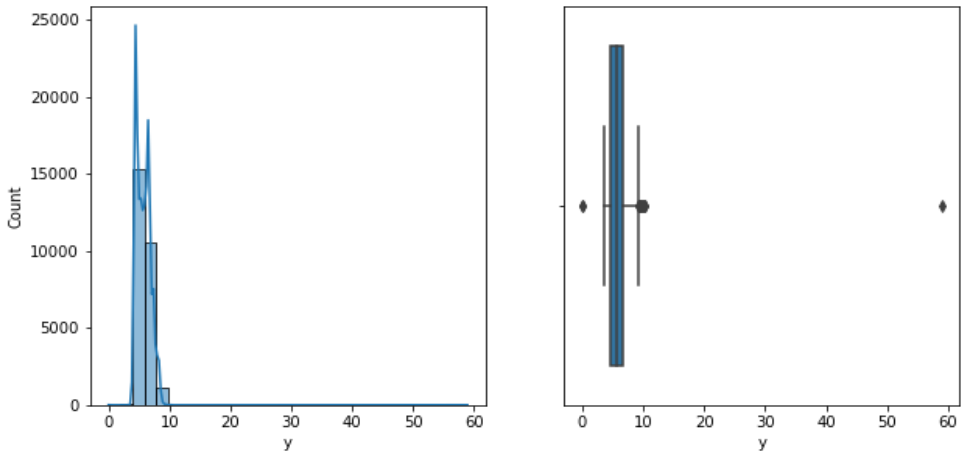


Fig 5 : Box & hist Plot – Univariant Analysis (y)

Observation:

- ❖ The above graph represents, 'Y' has positive skewness and skewed towards right.
- ❖ The histplot shows the distribution of data ranges from 0 to 60 and has very high positive kurtosis.
- ❖ The boxplot of the variable 'Y' has minimum outliers ie 0.14%, but mean & median values are not far away from each other and got squeezed.

Column : Z (Width of the cubic zirconia in mm)

```
Skewness of z: 2.58
Kurtosis of z: 87.42
Outliers of z:0.22
```

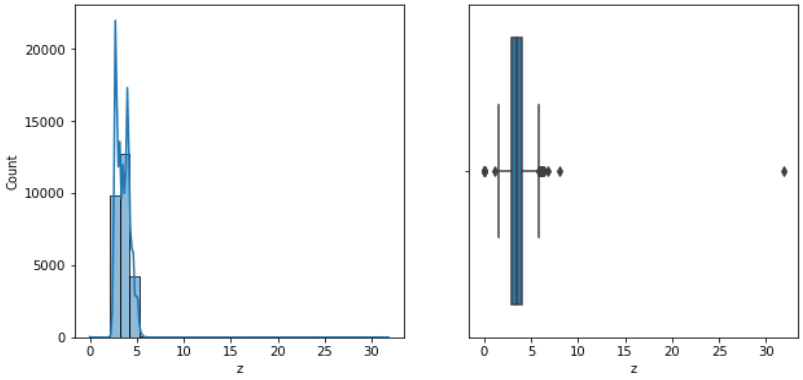


Fig 6 : Box & hist Plot – Univariant Analysis (z)

Observation:

- ❖ The above graph represents, 'Z' has positive skewness and skewed towards right.
- ❖ The histplot shows the distribution of data ranges from 0 to 30 and has high positive kurtosis.
- ❖ The boxplot of the variable 'Z' has minimum outliers ie 0.22%, but mean & median values are not far away from each other.

Column : price (the Price of the cubic zirconia)

```
Skewness of price: 2.58
Kurtosis of price: 2.15
Outliers of price: 17.78
```

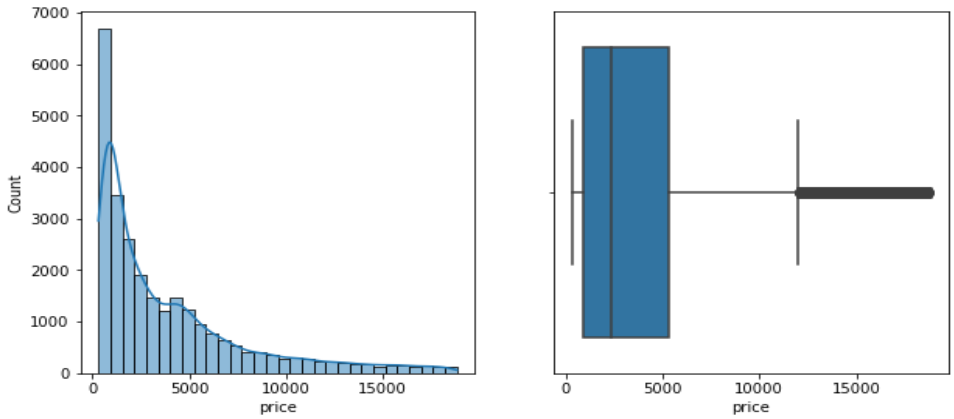


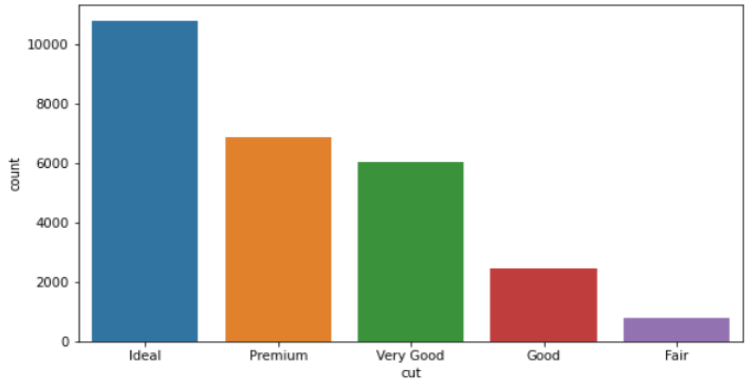
Fig 7 : Box & hist Plot – Univariant Analysis (price)

Observation:

- The above graph represents, 'Price' has positive skewness and skewed towards right.
- The histplot shows the distribution of data ranges from 0 to 15000 and has positive kurtosis.
- The boxplot of the variable 'Price' has more outliers ie 17.78%, but mean & median values are not far away from each other.

Univariant Analysis :- Countplot (Categorical Columns)

Column :- Cut

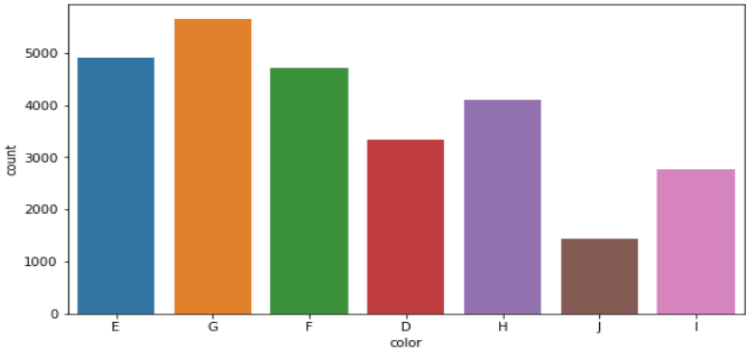


Value Count

```
Ideal        10805
Premium       6886
Very Good     6027
Good          2435
Fair           780
Name: cut, dtype: int64
```

Fig 8 : Count Plot - Univariant Analysis (cut)

Column :- Color

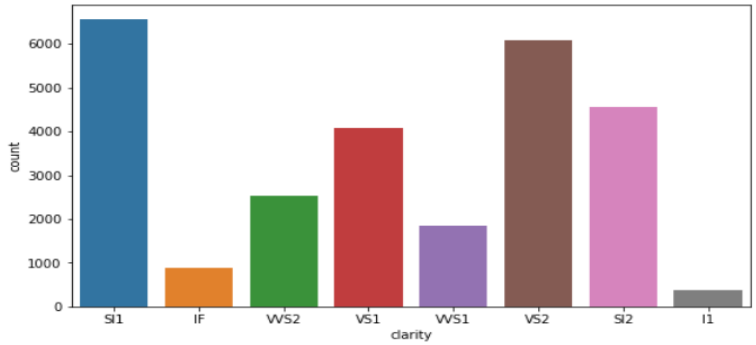


Value Count

```
G    5653
E    4916
F    4723
H    4095
D    3341
I    2765
J    1440
Name: color, dtype: int64
```

Fig 9 : Count Plot - Univariant Analysis (color)

Column:- Clarity



Value Count

```
SI1     6565
VS2     6093
SI2     4564
VS1     4087
VVS2    2530
VVS1    1839
IF       891
I1       364
Name: clarity, dtype: int64
```

Fig 10 : Count Plot - Univariant Analysis (clarity)

Observation:-

Cut:- Describe the cut quality of the cubic zirconia. Quality is increasing order Fair, Good, Very Good, Premium, Ideal.

Per plot we infer that most of cubic zirconia are of ideal cut quality with the value count of 10805, followed by Premium cut of 6886 and only 780 cubic zirconia has fair cut quality.

Color:- Colour of the cubic zirconia With D being the worst and J the best.

There are maximum number of Cubic zirconia with 'G' color and their count is 5653 and Cubic zirconia with 'J' color has the least count of 1440.

Clarity:- Clarity refers to the absence of the Inclusions and Blemishes. (In order from Worst to Best in terms of avg price) IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1.

Among 8 clarity types, the SI1 type has 6565 cubic zirconia without an Inclusions and Blemishes, followed by VS2 type with the count of 6093 and I1 type has only 364 cubic zirconia without Inclusions and Blemishes.

Bivariant Analysis:- For Bi-variant analysis of Target Vs continuous variables, we shall use Scatter plot.

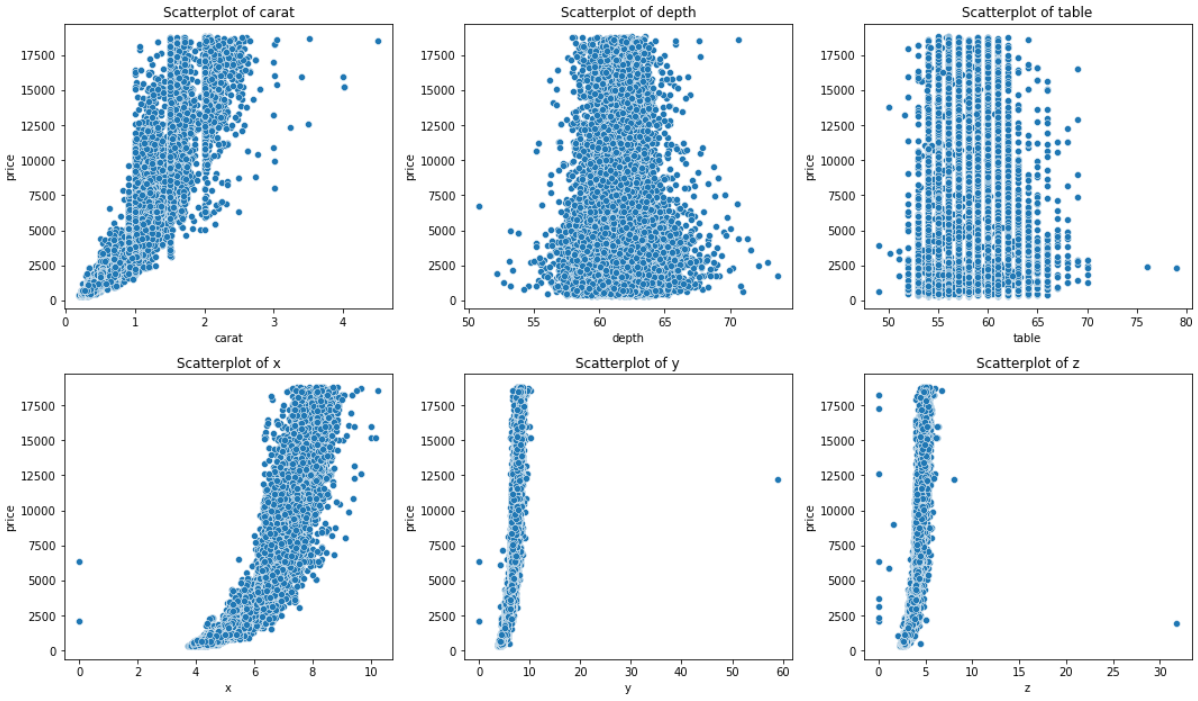


Fig 11 : Scatter Plot - Bivariant Analysis (continuous columns)

Observation :-

From the above scatter plot we infer that,

- Variable carat and price show a strong positive relationship, when carat weight of the cubic zirconia increases their price also simultaneously increases.

- ❖ Variable depth and price show no relationship and all the data points are scattered as cloud its mean.
- ❖ Variable width of the cubic zirconia's table and price shows very poor relationship.
- ❖ Variable x (Length of the cubic zirconia in mm) and price shows a strong positive relationship, where length of the cubic zirconia increases their price also increases simultaneously.
- ❖ Variable y (Width of the cubic zirconia in mm) and price show slightly positive relationship, where width of the cubic zirconia increases their price also seems increases simultaneously.
- ❖ Variable z (Height of the cubic zirconia in mm) and price show slightly positive relationship, where Height of the cubic zirconia increases their price also seems increases simultaneously.

According to the assumptions of Linear regression model, the independent variables should not be linearly corelated with each other which leads to the high multicollinearity between the independent variables Carat, X, Y and Z respectively.

Bivariant Analysis:- For Bi-variant analysis of categorical variables, we shall use count plot using hue command.

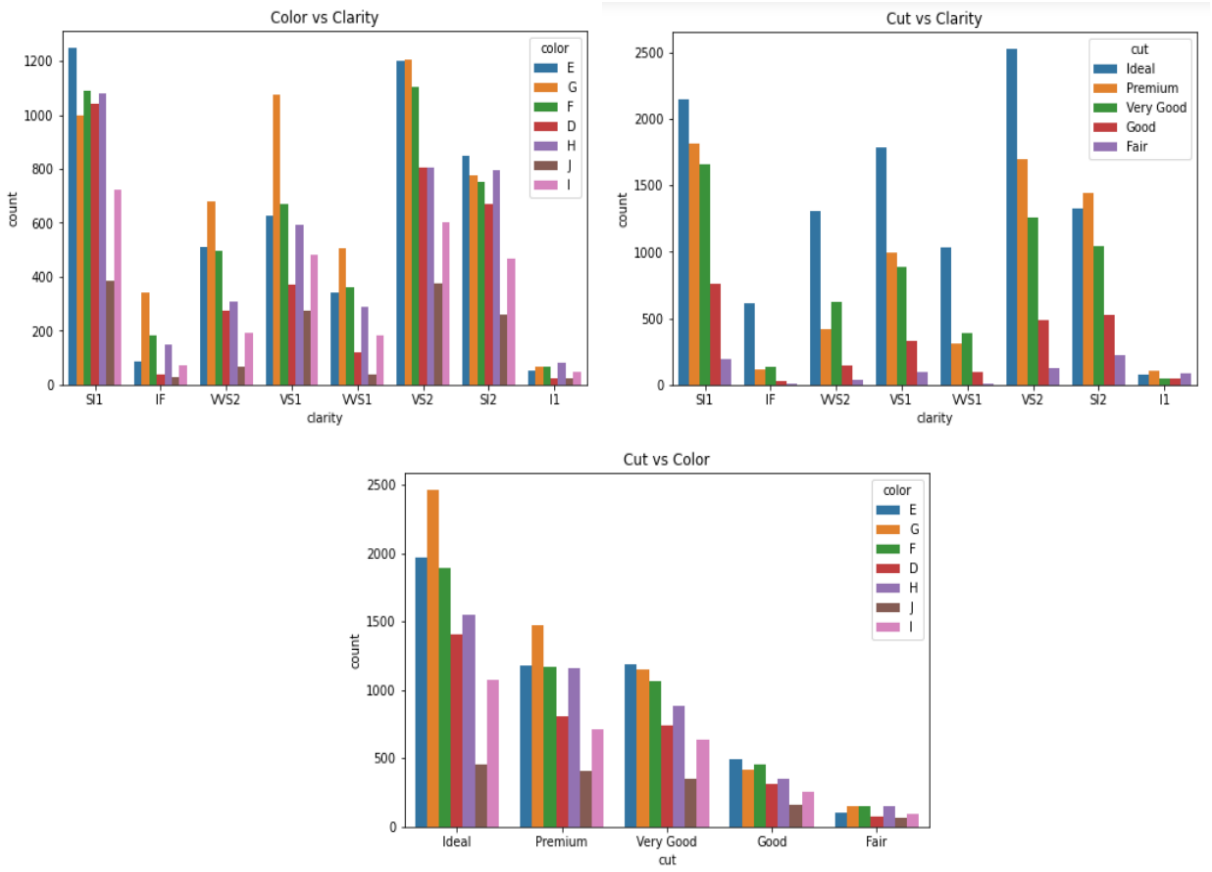


Fig 12 : Count Plot - Bivariant Analysis (Categorical columns)

Observation :-

Clarity vs Color :- Among 7 cubic zirconia colors, SI1 germs with G color has more clarity followed by VS2 germs and I1 germs shows less clarity across 7 colors followed by IF germs.

Clarity vs Cut:- High ideal quality cuts are found in VS2 germs followed by SI1 germs, Premium quality cuts also found in V52 & SI2 germs and most of germs has maximum of ideal and premium

quality cut. Low ideal quality germs found in I1 & IF germs. Across all the clarity of germs a fair quality cut seems to be very low.

Cut vs Color:- G & J color cubic zirconia has maximum ideal & Premium cut quality. Across 6 cut quality of the cubic zirconia maximum fair quality germs are of G, F & H color.

Multivariant Analysis :- Pair Plot & Heat Map

Pair Plot:- A pair plot gives us correlation graphs between all numerical variables in the dataset. Thus, from the graphs we can identify the relationships between all numerical variables.

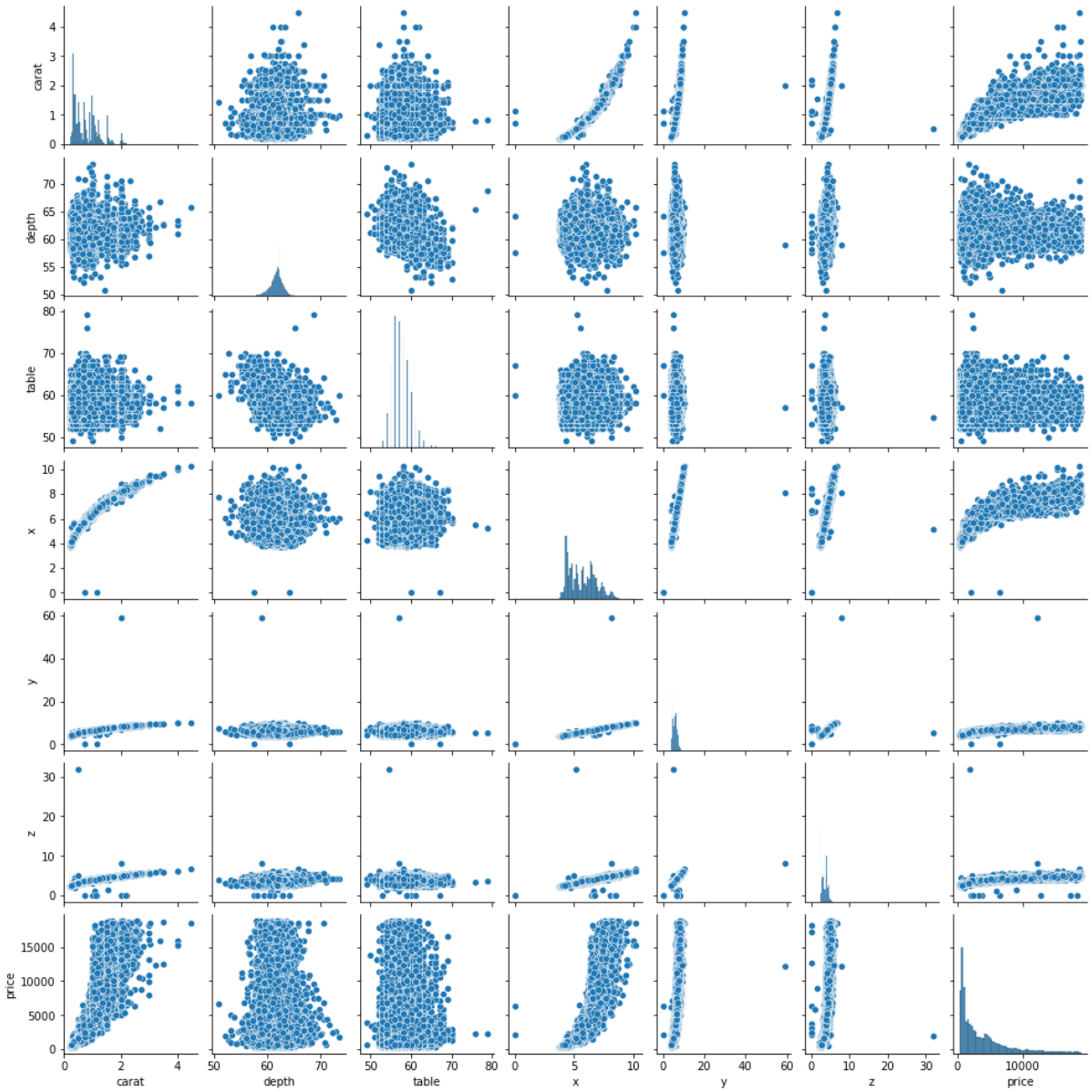


Fig 13 : Pair Plot - Multivariant Analysis (continuous columns)

HeatMap:-

A heatmap gives us the correlation between numerical variables. If the correlation value is tending to 1, the variables are highly positively correlated whereas if the correlation value is close to 0, the variables are not correlated. Also, if the value is negative, the correlation is negative. That means, higher the value of one variable, the lower is the value of another variable and vice-versa.

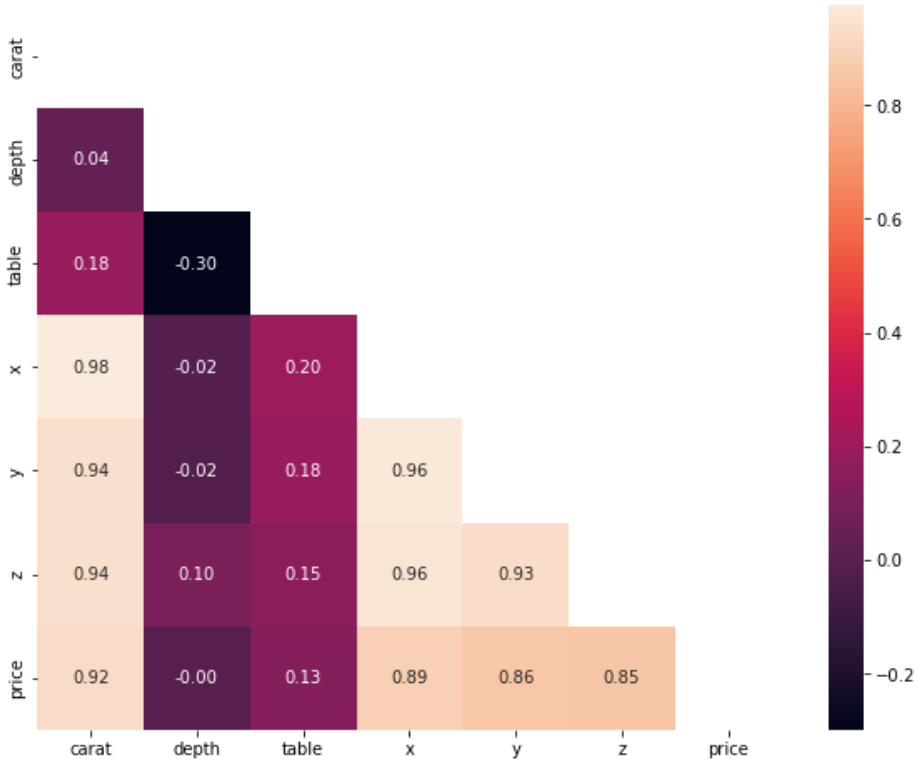


Fig 14 : Heatmap Plot – Correlation analysis (continuous columns)

Observation:-

Correlation is a statistical measure that expresses the extent to which two variables are linearly related.

- In reference to above heatmap and correlation matrix table we can infer the **high positive correlation** among following variables :-

Carat and X	0.98
Carat and Y	0.94
Carat and Z	0.94
Carat and price	0.92
X and Z	0.96
X and Y	0.96

Table 3 : Strongest Correlation Variables

- In reference to above heatmap and correlation matrix table we can infer the **Moderate correlation** among following variables :-

X and Price	0.89
Y and Price	0.86
Z and Price	0.85

Table 4 : Moderate Correlation Variables

- In reference to above heatmap and correlation matrix table we can infer the **poor and negative correlation** among following variables :-

Carat and depth	0.04	Carat and table	0.18
Z and depth	0.10	Depth and price	-0.00
Table and Price	0.13	Depth and Y	-0.02
Table and Z	0.15	Depth and X	-0.02
Table and Y	0.18	Depth and table	-0.30
Table and X	0.20		

Table 5 : Poor & negative Correlation Variables

Observation:-

- From the above heatmap we infer that, when weight of the cubic zirconia carat increases price of the cubic zirconia also increases and their length, width & Height also increases. Carat attribute is the best predictor of price.
- Variable Table and depth not correlated to any of the variables hence the chance of affecting stones in terms of high and low profitable prices are less.
- An above pair plot also gives the same insights of heatmap and scatterplot.

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 an ordinal variables and take actions accordingly. Explain why you are combining these sub levels with appropriate reasoning.

Checking Null Values :- Output from isnull with sum command is –

Dataset contains 697 null values in variable ‘depth’, here we are using median values for imputing missing values, due to the presence of outliers here mean value imputation is not recommended.

Checking Null Values after imputation :- Output from replace() of np.nan with median command is-

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

Now the dataset doesn’t contain any missing value In variable ‘depth’ after median value imputation.

Checking for the values which are equal to zero:-

During EDA performance of descriptive analysis, we found that, variable ‘x’ , ’y’ & ’z’ holds min value as 0 which is invalid i.e., length, Width & Height of the cubic zirconia in mm, cannot be as zero and this might be occurred due to the manual error.

	carat	cut	color	clarity	depth	table	x	y	z	price
5821	0.71	Good	F	SI2	64.1	60.0	0.00	0.00	0.0	2130
6034	2.02	Premium	H	VS2	62.7	53.0	8.02	7.95	0.0	18207
10827	2.20	Premium	H	SI1	61.2	59.0	8.42	8.37	0.0	17265
12498	2.18	Premium	H	SI2	59.4	61.0	8.49	8.45	0.0	12631
12689	1.10	Premium	G	SI2	63.0	59.0	6.50	6.47	0.0	3696
17506	1.14	Fair	G	VS1	57.5	67.0	0.00	0.00	0.0	6381
18194	1.01	Premium	H	I1	58.1	59.0	6.66	6.60	0.0	3167
23758	1.12	Premium	G	I1	60.4	59.0	6.71	6.67	0.0	2383

Table 6 : Sub grouped table (categorical columns)

From the above Jupyter output:- We infer that variable 'x', 'y' & 'z' holds eight bad values which are equal to zero. Since the number of observation that holds bad values are less than 1% compared to the total number of observation of 26933, here we are dropping the bad value observation and that will not have more impact during model building.

Shape of the dataset after dropping bad values : Output from shape command is –

```
The dataset has 26925 rows and 10 columns
```

Outlier Treatment :-

The present of outliers in the dataset may affect the output during Clustering. That is because each centroid is a mean, that is measure of central tendency whose value is affected by extreme values.

- As per univariant analysis (Boxplot) we deducted the presence of outliers in all the variables
- Using IQR Capping method, Imputing the Outlier values by replacing the observations outside the lower limit with the value of 25th percentile; and the observations that lie above the upper limit, with the value of 75th percentile of the same dataset.

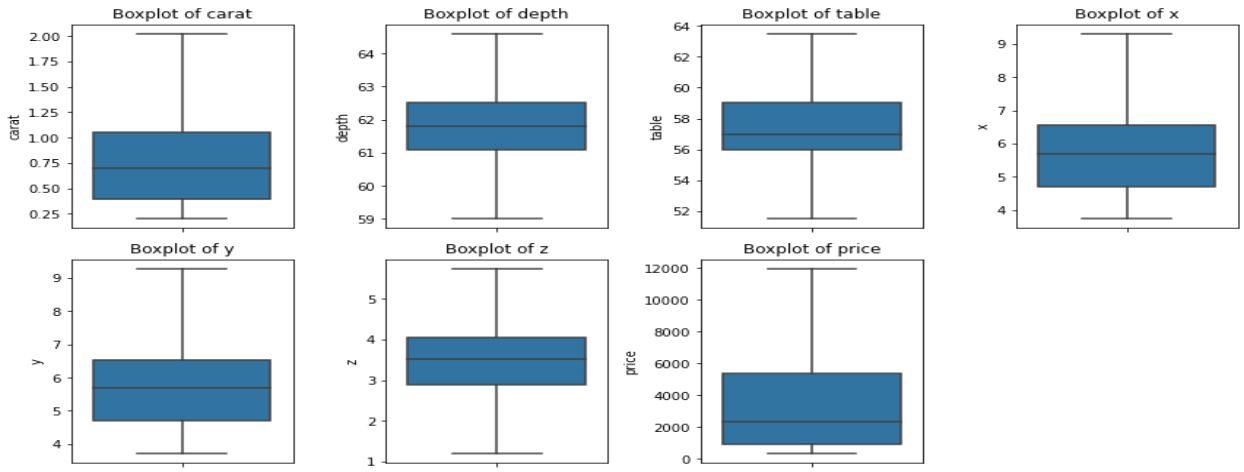


Fig:15 Box plot - After outlier treatment

The below boxplot shows that, after IQR imputation no outliers got deducted across the variables.

Checking for the possibility of combining the sub levels of an ordinal variables and taking actions, accordingly, also explaining why we are combining these sub levels.

Here we combine the sub levels of categorical ordinal variables in order to reduce the labels before performing an encoding so that it will reduce the curse of dimensionality.

Variable :- Cut

Variable cut describes the quality of the cubic zirconia, and the quality is increasing in an order of Fair, Good, Very Good, Premium, Ideal.

Below descriptive output is retrieved from Jupyter by grouping the variable 'cut' with respective to the target variable 'price':-

cut	Fair	Good	Ideal	Premium	Very Good
count	779.000000	2434.000000	10805.000000	6880.000000	6027.000000
mean	4364.383825	3770.679540	3282.618788	4276.784593	3829.352912
std	3193.788413	3149.222492	3353.661103	3679.608382	3461.492179
min	369.000000	335.000000	326.000000	326.000000	336.000000
25%	2117.000000	1157.000000	872.000000	1037.500000	910.000000
50%	3337.000000	3092.500000	1762.000000	3108.000000	2633.000000
75%	5407.500000	5112.250000	4668.000000	6257.250000	5438.000000
max	11965.000000	11965.000000	11965.000000	11965.000000	11965.000000

Table 7 : Descriptive Statistics (cut with price)

- From the above summary we infer that, mean and median values of category Good and Very good with respective to variable 'price' are close to each other. Here we are combining the sub level of Good and Very good and naming it as 'Good'.
- Our final Sub-categories of variable 'cut' order increasing from 'Fair', Good', Premium & Ideal'

Variable:- Color

Feature Color describes the color of the cubic zirconia, where D being the worst and J be the best.

Below descriptive output is retrieved from Jupyter by grouping the variable 'color' with respective to the target variable 'price':-

color	D	E	F	G	H	I	J
count	3341.000000	4916.000000	4722.000000	5650.000000	4091.000000	2765.000000	1440.000000
mean	3067.771027	2956.374288	3537.406184	3810.162301	4215.174529	4730.496926	5008.376389
std	3022.221311	2993.071116	3326.115989	3534.772080	3596.514528	3881.685718	3812.362655
min	357.000000	326.000000	357.000000	361.000000	337.000000	336.000000	335.000000
25%	910.000000	882.000000	947.250000	931.250000	990.000000	1145.000000	1843.000000
50%	1799.000000	1698.000000	2282.000000	2273.500000	3394.000000	3733.000000	4234.500000
75%	4265.000000	3892.750000	4862.000000	6096.750000	5949.500000	7292.000000	7592.000000
max	11965.000000	11965.000000	11965.000000	11965.000000	11965.000000	11965.000000	11965.000000

Table 8 : Descriptive Statistics (color with price)

- Here we decided not to sub levels of an ordinal color variables, because cubic zirconia stone can be of different colors and cannot be grouped with one color to another color.

Variable:- Clarity

Feature Clarity describes the absence of the Inclusions and Blemishes and ordering from Worst to Best in terms of avg price) IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1.

Below descriptive output is retrieved from Jupyter by grouping the variable 'Clarity' with respective to the target variable 'price':-

clarity	I1	IF	SI1	SI2	VS1	VS2	VVS1	VVS2
count	362.000000	891.000000	6564.000000	4561.000000	4086.000000	6092.000000	1839.000000	2530.000000
mean	3850.662983	2592.427609	3812.165143	4738.905722	3652.068527	3746.075837	2424.065797	3165.168379
std	2542.831763	3252.252509	3292.539153	3448.488611	3542.705105	3537.589627	3063.527052	3547.754089
min	345.000000	369.000000	326.000000	326.000000	338.000000	357.000000	336.000000	336.000000
25%	2071.000000	891.000000	1090.000000	2273.000000	877.000000	876.000000	814.000000	791.750000
50%	3494.000000	1063.000000	2795.000000	4077.000000	1949.000000	2066.000000	1066.000000	1253.000000
75%	5031.000000	2291.000000	5266.000000	5828.000000	6120.250000	6069.750000	2217.500000	3583.750000
max	11965.000000	11965.000000	11965.000000	11965.000000	11965.000000	11965.000000	11965.000000	11965.000000

Table 9 : Descriptive Statistics (clarity with price)

- The subcategory of VS1 and VS2 stone's mean and median in terms of price are very close to each other. Let's combine the category of VS1 & VS2 and name it as VS.
- Next categories which can be grouped are VVS1 & VVS2 their mean and median in terms of price seems to be slightly different, but they are close to each other. Let's combine the category of VVS1 & VVS2 and name it as VVS.
- At last, the categories that can be grouped are SI1 & SI2 and their mean and median in terms of price are to be almost same. Given that SI1 has a larger stone count but has lower mean value in terms of price, mean while SI2 has a lower stone count, but has high mean value in terms of price. Here we conclude that two stones are balanced, and we combine the category of SI1 & SI2 and name it as SI.
- Our final Sub-categories of variable 'Clarity' order increasing IF, VVS, VS, Sl, l1.

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.

1. Encoding the categorical variables:-

- The given dataset categorical variables are having defined ordinal sub-categories, as per data dictionary we are already aware of relationship between the categories, so ordinal encoding is appropriate technique to convert categorical features into numerical values so that machine learning algorithm can understand.
- An Ordinal Encoder is used to encode categorical features into an ordinal numerical value (ordered set). This approach transforms categorical value to numerical value in ordered sets.
- This encoding technique appears almost similar to Label Encoding. But label encoding would not consider whether a variable is ordinal or not, but in the case of ordinal encoding, it will assign a sequence of numerical values as per the order of data.

Ordinal encoding for variable 'cut'–

Variable cut describes the cut quality of the cubic zirconia. Quality is increasing order Fair, Good, Premium, Ideal. Here with the help of Pandas, we will map each row for the variable as per the dictionary, where 0 being the worst and 3 being the best :-

- Fair:0
- Good:1
- Premium:2
- Ideal:3

Ordinal encoding for variable 'Color'–

Variable color describes the color of the cubic zirconia, where D being the worst and J be the best. Here with the help of Pandas, we will map each row for the variable as per the dictionary, where 0 being the worst and 6 being the best :-

- D :0
- E :1
- F :2
- G :3
- H :4
- I :5
- J :6

Ordinal encoding for variable 'clarity'–

Feature Clarity describes the absence of the Inclusions and Blemishes and ordering from Worst to Best in terms of avg price) IF, VVS, VS, Sl, l1. Here with the help of Pandas, we will map each row for the variable as per the dictionary, where 0 being the worst and 4 being the best :-

- IF : 0
- VVS : 1
- VS : 2
- SI : 3
- I1 : 4

2. Checking the head after encoding the categorical variables in juypter using head() command –

	carat	cut	color	clarity	depth	table	x	y	z	price
0	0.30	3	1	3	62.1	58.0	4.27	4.29	2.66	499.0
1	0.33	2	3	0	60.8	58.0	4.42	4.46	2.70	984.0
2	0.90	1	1	1	62.2	60.0	6.04	6.12	3.78	6289.0
3	0.42	3	2	2	61.6	56.0	4.82	4.80	2.96	1082.0
4	0.31	3	2	1	60.4	59.0	4.35	4.43	2.65	779.0

Table 10 : Encoded table (categorical columns)

From above output we infer that all categorical columns got encoded with numeric values and now the dataset is ready for model building.

3. Checking the Info of the dataset after ordinal encoding:-

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 26925 entries, 0 to 26966
Data columns (total 10 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   carat    26925 non-null  float64
 1   cut      26925 non-null  int64
 2   color    26925 non-null  int64
 3   clarity  26925 non-null  int64
 4   depth    26925 non-null  float64
 5   table    26925 non-null  float64
 6   x        26925 non-null  float64
 7   y        26925 non-null  float64
 8   z        26925 non-null  float64
 9   price    26925 non-null  float64
dtypes: float64(7), int64(3)
memory usage: 2.3 MB
```

Linear Regression Model :-

Linear regression is a supervised Machine Learning model and way to identify a relationship between two or more variables. We use this relationship to predict the values for one variable for a given set of value(s) of the other variable(s). The variable, which is used in prediction is termed as independent/explanatory/regress or variable where the predicted variable is termed as dependent/target/response/regress and variable. Linear regression assumes that the dependent variable is linearly related to the estimated parameter(s).

Main goal of Linear Regression model is to find the best fit linear line and the optimal values of intercept and coefficients such that the error is minimized.

4. Creating multiple models and checking their performance of Predictions on Train and Test sets using Rsquare, RMSE & Adj Rsquare:-

Here we build a Multiple Linear Regression model and check their model performance metrics, at the end we will compare the created models and select the best fit model to create a final linear equation.

In machine learning and Multiple Linear Regression literature the above equation is used in the form: -

Y= $m1x1$ + $m2x2$+ $m3x3$+ ……+ mnxn+C+e

- y - Dependent/target/predicted variable
- xi – Independent/Predictor varible
- mi- Co-efficients for the ith independent/Predictor variable
- C – Constant/intercept/bias
- e – Residual error/unexplained variance/difference between actual and prediction
-

$$\text{y (price)} = \text{C+ m1 * carat + m2 * cut + m3 * color + m4 * clarity + m5 * depth + m6 * table + m7 * x+ m8 * y+ m9 * z.}$$

Model 1 : Considering all the variables in the dataset and fitting in to linear regression.

Model 2 : Dropping high multicollinearity attributes of independent variables ‘x’, ‘y’,’z’ and fitting in to linear regression.

Model 3 : Dropping high multicollinearity attributes of independent variables ‘x’, ‘y’,’z’, and fitting in to linear regression.

Model 4 : Scaling and fitting the data in to linear regression.

Model 1 : Considering all the variables in the dataset and fitting in to linear regression.

Linear Regression Model 1 using Sklearn -

In this model, all the attributes are considered, and the dataset is not scaled since the accuracy and model performance does not get influence by scaling the dataset.

1. **Importing the necessary library of Linear regression from scikit learn package**
2. **Extracting the target column into separate vectors for training set and test set.**
   - Here we store the independent features in variable X ('carat', 'cut', 'color', 'clarity', 'depth', 'table', 'x', 'y', 'z') dependent feature/Target feature in Y variable('Price').
   - Train data will hold an independent variables whereas test data will hold a dependent variable of the dataset.
3. **Splitting data into training and test set.**
   - Inorder to perform this step, from the package sklearn.model_selection we imported train_test_split.
   - Now we split the data into 70 -30 ratio, where the train data hold 70% of the data and test data holds 30% of the data. The random state mentioned here is 1.
4. **Checking the dimensions of the training and test data.**
   Below output is retrieved from Jupter using shape command.

```
X_train (18847, 9)
X_test (8078, 9)
train_labels (18847,)
test_labels (8078,)
```

Now we have our train and test data ready. We will start building our Linear regression for model:1.

5. **Fitting the Linear regression model from sklearn linear models to Training set.**
6. **Finding the coefficient of determinants for each of the independent attributes in Jupyter using coef_ command.**

The coefficients in linear equation denote the magnitude of additive relation between the predictor and the response variable. In another word, keeping everything else fixed, to see for every one-unit increase/ decrease in X, how much does Y changes i.e., increase/decrease.

```
The coefficient for carat is 8876.101584466329
The coefficient for cut is 115.90214179948099
The coefficient for color is -261.96553649230464
The coefficient for clarity is -795.8798497789428
The coefficient for depth is 22.44659598777949
The coefficient for table is -21.259559425363648
The coefficient for x is -1483.1681321511865
The coefficient for y is 1655.4821489743026
The coefficient for z is -949.5464732378701
```

- The above coefficients of determinants output describe among 9 independent variables, variable 'carat' has most weightage and acts as a good predictor for target variable 'price'. Ie When carat increases by 1 unit, price increases by 8876.10 units, keeping all other predictors constant.
- We also infer that, variable 'X', 'Z', 'table', 'clarity', 'color' has negative coefficient, and it acts as a week predictor for target variable 'price'. Variable 'cut', 'depth' & 'Y' has some moderate weightage in predicting target variable 'price'.

7. Intercept of the model

The **intercept** (sometimes called the "constant") in a regression model represents the mean value of the predictor variable when all of the response variables in the model are equal to zero. If X(response) variable never equals 0, then the intercept has no intrinsic meaning and doesn't tell anything about the relationship between X and Y.

Intercept for our model is 1104.9988552259715.

The value for the intercept term in our model is **1104.99**. This means the average price is **1104.99** when the number of (cubic zirconia) diamond is equal to zero.

Here intercept doesn't make any sense, since it's not possible for a diamond to be zero priced.

8. R Squared

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.
The definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model.

R-squared = Explained variation / Total variation
R-squared is always between 0 and 100%

- 0% indicates that the model explains none of the variability of the response data around its mean.
- 100% indicates that the model explains all the variability of the response data around its mean.

In general, the higher the R-squared, the better the model fits your data.

Below output retrieved from Jupyter on training and testing data using score command -

```
R squared for training set is 0.9272782479041014
R squared for testing set is 0.9278137643972159
```

We see that, R squared value for train and set close to 90% and we can assume that data are close enough to the fitted regression line, but this not a reliable metrices since it might contain some statistical fluke. To arrest this statistical fluke, we will consider the adjusted R squared.

9. Evaluation of RMSE (Root Mean Square Error)

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. In general, a lower RMSD and close to 0 would indicate a perfect fit to the data. However, comparisons across different types of data would be invalid because the measure is dependent on the scale of the numbers used.

Below output retrieved from Jupyter on training and testing data from Sklearn package using metrics.mean_squared_error command -

```
RMSE for training set is 932.5837368395347
RMSE for testing set is 936.3569120438865
```

From the output we infer that, RMSE value for train & test set is far away from zero, this might be because the data is not scaled, in model-4 we will evaluate and observe whether scale data change the RMSE score or not.

10. Evaluation of plot between actual and predicted price for linear regression-

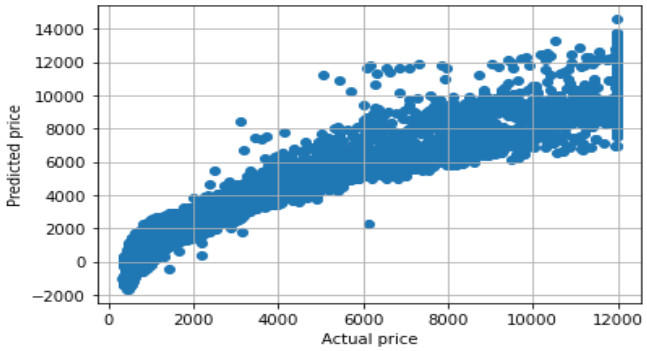


Fig 16 : Model 1 - Scatter Plot (Actual vs Predicted)

From the above scatter plot we infer that, data points got slightly scattered but doesn't not look cloudy. This indicates that, our model did reasonable prediction and forms more or less a positive linear line.

Linear Regression Model 1 using statsmodel –

Statsmodel uses OLS (ordinary least square method) to predict the best fit plane. OLS also minimizes the sum of squared error between the observed and predicted values by estimating its coefficients and bias.

One of the main difference between sci-kit Learn and statsmodel in Linear Regression is, that stats model provides a stronger emphasis on statistical inference such as adjusted R square and statistical hypothesis testing and with the help of these metrics we can compare and decide whether to drop the week attributes or not.

- **Adjusted R square :-**

Adjusted r^2=r^2- statistical fluke

- ❖ Here we remove statistical fluke from r^2 and get adjusted r^2, when we compare the model with r^2 and adjusted r^2, the adjusted r^2 will always be less than a r^2.
- ❖ When we can more redundant variables to the dataset, the r^2 will either go up or remain static and will approach to the range of (0-1), whereas adjusted r^2 will go down when we add useless dimension to the model and will get increase only when we add good dimension to our model.
- ❖ From this we conclude that, adjusted r^2 is more reliable than r^2 for evaluating a model.

- **Hypothesis testing:-**

In OLS model, to establish the reliability of the coefficients, we need hypothesis testing. Assuming Ho to be true, what is the probability of finding the coefficients found in the sample if the sample is drawn from the universe in which Ho is true.

The null hypothesis (H0) claims that there is no relation between dependent and independent variables, that means the co-efficient is 0 in the universe.

- ❖ At 95% confidence level if the p value is > = 0.5, we do not have enough evidence in data to reject the Ho and hence we believe Ho is likely to be true in the universe.
- ❖ if p value is < 0.5, we reject null hypothesis, we have enough evidence to accept Ho and believe there is relationship between dependent and independent variables.

1. Checking the OLS summary for model 1-

OLS Regression Results

Dep. Variable:	price	R-squared:	0.927
Model:	OLS	Adj. R-squared:	0.927
Method:	Least Squares	F-statistic:	2.669e+04
Date:	Fri, 20 May 2022	Prob (F-statistic):	0.00
Time:	05:55:40	Log-Likelihood:	-1.5562e+05
No. Observations:	18847	AIC:	3.113e+05
Df Residuals:	18837	BIC:	3.113e+05
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1104.9989	807.913	1.368	0.171	-478.583	2688.580
carat	8876.1016	84.961	104.473	0.000	8709.571	9042.632
cut	115.9021	9.105	12.729	0.000	98.055	133.749
color	-261.9655	4.210	-62.224	0.000	-270.218	-253.713
clarity	-795.8798	8.833	-90.107	0.000	-813.192	-778.567
depth	22.4466	11.311	1.984	0.047	0.276	44.617
table	-21.2596	4.030	-5.276	0.000	-29.158	-13.361
x	-1483.1681	139.421	-10.638	0.000	-1756.447	-1209.890
y	1655.4821	136.886	12.094	0.000	1387.173	1923.792
z	-949.5465	143.028	-6.639	0.000	-1229.894	-669.199

Omnibus:	2691.417	Durbin-Watson:	2.000
Prob(Omnibus):	0.000	Jarque-Bera (JB):	8724.595
Skew:	0.732	Prob(JB):	0.00
Kurtosis:	5.995	Cond. No.	1.02e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.02e+04. This might indicate that there are strong multicollinearity or other numerical problems.

Table 11 : Model - 1 OLS stats table

From the above OLS summary, we infer that -

- ❖ R-squared and adjusted R-squared values are same which is equal to 0.927
- ❖ Overall p value for the model is 0.00 which is less that 0.05, this indicates that, there is some relation between dependent and independent variables.
- ❖ Standard Errors assume that the covariance matrix of the errors is correctly specified.
- ❖ The condition number is large, 1.02e+04. This might indicate that there are strong multicollinearity or other numerical problems. As we already noticed in EDA process that few variable has positive multicollinearity.
- ❖ Kurtosis value found be 5.995 which indicates the dataset has slightly high positive peak tail and fatter tails than normal distribution.
- ❖ The skew value is 0.732 indicated data slightly right skewed
- ❖ The Prob(JB) test value is 0.00 which indicates that, the sample data is normally distributed.

2. **Evaluation of RMSE (Root Mean Square Error)**

Below output is retrieved from Jupyter using math.sqrt command by predicting value of y for training & testing cases and subtracting from the actual y for the training & testing cases.

- ❖ RMSE for training set in OLS model is 932.5837368395352
- ❖ RMSE for testing set in OLS model is 936.3569120438865

From the output we infer that both Sklearn and statsmodel gives the same RMSE score.

3. **Checking Multi-collinearity using VIF technique**

1. The variance inflationfactor(VIF) identifies correlation between independent variables and the strength of that correlation.

a. **VIF = 1 / 1 –r2**
b. As the collinearity between variables increase, r2increases, the denominator approaches 0 and as a result VIF approaches infinity.

2. VIFs start at 1 and have no upper limit.
A value of 1 indicates that there is no correlation between this independent variable and any others

a. VIFs between 1 and 5 suggest that there is a moderate correlation, but it is not severe enough to warrant corrective measures
b. VIFs greater than 5 represent critical levels of multicollinearity where the coefficients are poorly estimated, and the p-values are questionable.

Below output is retrieved from Jupyter on independent variable using the command variance_inflation_factor –

```
carat ---> 123.0827435038358
cut ---> 7.887496814302903
color ---> 3.711149940023802
clarity ---> 18.620853151109056
depth ---> 1201.143350402747
table ---> 893.1456864311363
x ---> 10722.054330582225
y ---> 9355.993077375968
z ---> 3234.5376185766945
```

- ❖ From the above output we infer that, for all the variables, the VIF values are greater than 5 this represent critical levels of multicollinearity where the coefficients are poorly estimated, and the p-values are questionable. Making an appropriate change and dropping highly correlated variables in the data can help us to arrest the Multicollinearity.
- ❖ Compared to other variables, variable ‘x’,’y’,’z’,’table’ & depth has highest VIF score, so in model 2 we will drop the highest multicollinearity varible ‘x’,’y’ & ‘z’ and check for the further performance.

4. **Linear Equation**

$$y\ (price) = m0 + m1 * carat + m2 * cut + m3 * color + m4 * clarity + m5 * depth + m6 * table + m7 * x + m8 * y + m9 * z.$$

```
price = (1900.88) * Intercept + (8876.1) * carat + (115.9) * cut + (-261
.97) * color + (-795.88) * clarity + (22.45) * depth + (-21.26) * table
+ (-1483.17) * x + (1655.48) * y + (-949.55) * z
```

- ❖ Variable ‘carat’ has most weightage and acts as a good predictor for target variable ‘price’. Ie When carat increases by 1 unit, price increases by 8876.10 units, keeping all other predictors constant.
- ❖ Based on other performing metrics, we conclude that, this is not the best model for predicting price slots for zirconia stones. But based on this model we came to know what the changes are to be done, in order to create a best fit model.

Model 2 : Dropping high multicollinearity attributes of independent variables ‘x’, ‘y’,’z’ and fitting in to linear regression.

Linear Regression Model 2 using Sklearn -

In this model, we are dropping the highly correlated attributes ‘x’,’y’,’z’, and the dataset is not scaled since the accuracy and model performance does not get influence by scaling the dataset.

1. **Importing the necessary library of Linear regression from scikit learn package**
2. **Extracting the target column into separate vectors for training set and test set.**
   - ❖ Here we store the impendent features in variable X (‘carat', 'cut', 'color', 'clarity', 'depth', 'table') dependent feature/Target feature in Y variable(‘Price’).
   - ❖ Train data will hold an independent variables whereas test data will hold a dependent variable of the dataset.
3. **Splitting data into training and test set.**
   - ❖ Inorder to perform this step, from the package sklearn.model_selection we imported train_test_split.
   - ❖ Now we split the data into 70 -30 ratio, where the train data hold 70% of the data and test data holds 30% of the data. The random state mentioned here is 1.
4. **Checking the dimensions of the training and test data.**
   Below output is retrieved from Jupter using shape command.

```
X_train (18847, 9)
X_test (8078, 9)
train_labels (18847,)
test_labels (8078,)
```

Now we have our train and test data ready. We will start building our Linear regression for model:1.

5. **Fitting the Linear regression model from sklearn linear models to Training set.**
6. **Finding the coefficient of determinants for each of the independent attributes in Jupyter using coef_ command.**

The coefficients in linear equation denote the magnitude of additive relation between the predictor and the response variable. In another word, keeping everything else fixed, to see for every one-unit increase/ decrease in X, how much does Y changes i.e., increase/decrease.

```
The coefficient for carat is 7877.03817022081
The coefficient for cut is 103.89057343274295
The coefficient for color is -260.2731684668061
The coefficient for clarity is -823.6817195374326
The coefficient for depth is -28.523685266241426
The coefficient for table is -27.237961378651267
```

- ❖ The above coefficients of determinants output describe among 7 independent variables, variable 'carat' has most weightage and acts as a good predictor for target variable 'price'. Ie When carat increases by 1 unit, price increases by 7877 units, keeping all other predictors constant.
- ❖ We also infer that, variable 'table', 'depth' 'clarity', 'color' in this model also gives negative coefficient, and it acts as a week predictor for target variable 'price'. Variable 'cut' has some moderate weightage in predicting target variable 'price'.

7. Intercept of the model

The **intercept** (sometimes called the "constant") in a regression model represents the mean value of the predictor variable when all of the response variables in the model are equal to zero. If X(response) variable never equals 0, then the intercept has no intrinsic meaning and doesn't tell anything about the relationship between X and Y.

Intercept for our model is 3101.317022281597

The value for the intercept term in our model is **3101.32,** which is lower than model 1. This means the average price is **3101.32** when the number of (cubic zirconia) diamond is equal to zero.

Here intercept doesn't make any sense since it's not possible for a diamond to be zero priced.

8. R Squared

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.
The definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model.

R-squared = Explained variation / Total variation
R-squared is always between 0 and 100%

- 0% indicates that the model explains none of the variability of the response data around its mean.
- 100% indicates that the model explains all the variability of the response data around its mean.

In general, the higher the R-squared, the better the model fits your data.

Below output retrieved from Jupyter on training and testing data using score command -

```
R squared for training set is 0.9260825413502574

R squared for testing set is  0.9266320935192703
```

We see that, R squared value for train and set for this model also close to 90% and no drastic difference found compared to model 1 and we can assume that data are close enough to the fitted regression line, but this not a reliable metrices since it might contain some statistical fluke. To arrest this statistical fluke, we will consider the adjusted R squared.

9. Evaluation of RMSE (Root Mean Square Error)

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. In general, a lower RMSD and close to 0 would indicate a perfect fit to the data. However, comparisons across different types of data would be invalid because the measure is dependent on the scale of the numbers used.

Below output retrieved from Jupyter on training and testing data from Sklearn package using metrics.mean_squared_error command -

```
RMSE for training set is 940.2193485412608
RMSE for testing set is 943.9897677390762
```

From the output we infer that, RMSE value for train & test set in this model also far away from zero, this might be because the data is not scaled, in model-4 we will evaluate and observe whether scale data change the RMSE score or not.

10. Evaluation of plot between actual and predicted price for linear regression-

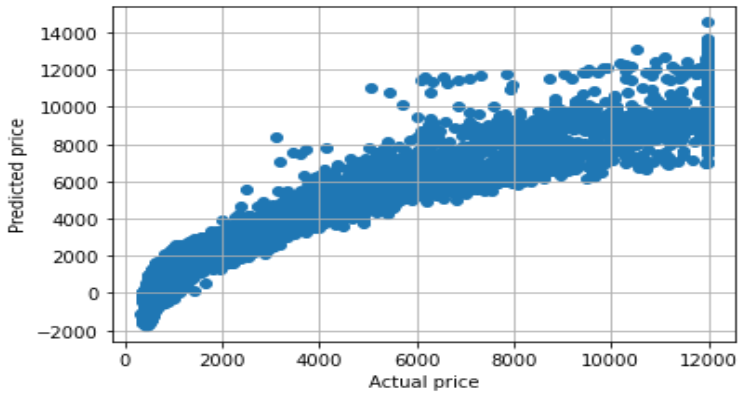


Fig 17 : Model 2 - Scatter Plot (Actual vs Predicted)

From the above scatter plot we infer that, both model 1 & model 2 data points got slightly scattered but doesn't not look cloudy. This indicates that, our model did reasonable prediction and forms more or less a positive linear line.

Linear Regression Model 2 using statsmodel –

Statsmodel uses OLS (ordinary least square method) to predict the best fit plane. OLS also minimizes the sum of squared error between the observed and predicted values by estimating its coefficients and bias.

One of the main difference between sci-kit Learn and statsmodel in Linear Regression is, that stats model provides a stronger emphasis on statistical inference such as adjusted R square and statistical hypothesis testing and with the help of these metrics we can compare and decide whether to drop the week attributes or not.

- **Adjusted R square :-**

Adjusted r^2=r^2- statistical fluke

- Here we remove statistical fluke from r^2 and get adjusted r^2, when we compare the model with r^2 and adjusted r^2, the adjusted r^2 will always be less than a r^2.
- When we can more redundant variables to the dataset, the r^2 will either go up or remain static and will approach to the range of (0-1), whereas adjusted r^2 will go down when we add useless dimension to the model and will get increase only when we add good dimension to our model.
- From this we conclude that, adjusted r^2 is more reliable than r^2 for evaluating a model.

- **Hypothesis testing:-**

In OLS model, to establish the reliability of the coefficients, we need hypothesis testing. Assuming Ho to be true, what is the probability of finding the coefficients found in the sample if the sample is drawn from the universe in which Ho is true.

The null hypothesis (H0) claims that there is no relation between dependent and independent variables, that means the co-efficient is 0 in the universe.

- At 95% confidence level if the p value is > = 0.5, we do not have enough evidence in data to reject the Ho and hence we believe Ho is likely to be true in the universe.
- if p value is < 0.5, we reject null hypothesis, we have enough evidence to accept Ho and believe there is relationship between dependent and independent variables.

1. **Checking the OLS summary for model 1-**

OLS Regression Results

Dep. Variable:	price	R-squared:	0.926
Model:	OLS	Adj. R-squared:	0.926
Method:	Least Squares	F-statistic:	3.934e+04
Date:	Fri, 20 May 2022	Prob (F-statistic):	0.00
Time:	12:11:20	Log-Likelihood:	-1.5577e+05
No. Observations:	18847	AIC:	3.116e+05
Df Residuals:	18840	BIC:	3.116e+05
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	3924.9987	537.405	7.304	0.000	2871.636	4978.362
carat	7877.0382	16.804	468.766	0.000	7844.101	7909.975
cut	103.8906	9.100	11.417	0.000	86.055	121.727
color	-260.2732	4.240	-61.387	0.000	-268.584	-251.963
clarity	-823.6817	8.734	-94.302	0.000	-840.802	-806.561
depth	-28.5237	6.327	-4.508	0.000	-40.926	-16.121
table	-27.2380	3.996	-6.816	0.000	-35.071	-19.405

Omnibus:	2395.483	Durbin-Watson:	2.002
Prob(Omnibus):	0.000	Jarque-Bera (JB):	6827.675
Skew:	0.691	Prob(JB):	0.00
Kurtosis:	5.605	Cond. No.	6.63e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.63e+03. This might indicate that there are strong multicollinearity or other numerical problems.

Table 12 : Model - 2 OLS stats table

From the above OLS summary, we infer that -

- ❖ R-squared and adjusted R-squared values are same which is equal to 0.926.
- ❖ Overall p value for the model is 0.00 which is less than 0.05, this indicates that, there is some relation between dependent and independent variables.
- ❖ Standard Errors assume that the covariance matrix of the errors is correctly specified.
- ❖ The condition number is large, 6.63e+03, but less than model 1 value 1.02e+04. This might indicate that there are strong multicollinearity or other numerical problems. As we already noticed in EDA process that few variable has positive multicollinearity.
- ❖ Kurtosis value found be slightly less than model-1 that is 5.995 and the model 2 value is 5.605 which indicates the dataset has slightly high positive peak tail and fatter tails than normal distribution.
- ❖ The skew value also slightly less compared to model 1 and the value is 0.691 indicated data slightly right skewed
- ❖ The Prob(JB) test value is 0.00 which indicates that, the sample data is normally distributed.

2. Evaluation of RMSE (Root Mean Square Error)

Below output is retrieved from Jupyter using math.sqrt command by predicting value of y for training & testing cases and subtracting from the actual y for the training & testing cases.

- ❖ RMSE for training set in OLS model is 940.2193485412616
- ❖ RMSE for testing set in OLS model is 943.9897677390751

From the output we infer that both Sklearn and statsmodel gives the same RMSE score.

3. Checking Multi-collinearity using VIF technique

1. The variance inflationfactor(VIF) identifies correlation between independent variables and the strength of that correlation.

c. **VIF = 1 / 1 –r2**

d. As the collinearity between variables increase, r2increases, the denominator approaches 0 and as a result VIF approaches infinity.

2. VIFs start at 1 and have no upper limit.
A value of 1 indicates that there is no correlation between this independent variable and any others

c. VIFs between 1 and 5 suggest that there is a moderate correlation, but it is not severe enough to warrant corrective measures

d. VIFs greater than 5 represent critical levels of multicollinearity where the coefficients are poorly estimated, and the p-values are questionable.

Below output is retrieved from Jupyter on independent variable using the command variance_inflation_factor –

```
carat ---> 5.00143380530788
cut ---> 6.427197235863787
color ---> 3.7034419766661943
clarity ---> 17.996785779953473
depth ---> 523.2411991065859
table ---> 492.6503248587516
```

- From the above output we infer that, compared to model 1, model 2 VIF score got decreased to very good extend for the variable 'carat' & 'color' and their VIF score is less than 5 this represent there is no critical levels of multicollinearity, and the coefficients are correctly estimated.
- Variable 'cut' VIF score is slightly greater than 5, but variable 'depth' & 'table' in this model also gives VIF values as greater than 5 this represent critical levels of multicollinearity where the coefficients are poorly estimated, and the p-values are questionable. Making an appropriate change and dropping highly correlated variables in the data can help us to arrest the Multicollinearity. Making an appropriate change and dropping highly correlated variables in the data can help us to arrest the Multicollinearity.

4. **Linear Equation**

$$y\ (\text{price}) = m0 + m1 * \text{carat} + m2 * \text{cut} + m3 * \text{color} + m4 * \text{clarity} + m5 * \text{depth} + m6 * \text{table}$$

```
price = (3925.0) * Intercept + (7877.04) * carat + (103.89) * cut + (-26
0.27) * color + (-823.68) * clarity + (-28.52) * depth + (-27.24) * tab
le +
```

- Variable 'carat' still holds the more weightage and acts as a good predictor for target variable 'price'. Ie When carat increases by 1 unit, price increases by 7877.04 units, keeping all other predictors constant.
- Based on other performing metrics, we conclude that, From this model we can infer that, the strong multicollinearity, which was due to x, y & z is reduced to greater extend. The remaining collinearity is due to the 'depth' variable. The depth variable can also be dropped since it's not a good predictor for model building. In next model we built by dropping the depth variable and compare its effect on the model performance with the previous models and choose the best fit model for prediction of price slots for a company.

Model 3 : Dropping high multicollinearity attributes of independent variables 'x', 'y','z' & depth and fitting in to linear regression.

Linear Regression Model 3 using Sklearn -

In this model, we are dropping the highly correlated attributes 'x','y','z', and the dataset is not scaled since the accuracy and model performance does not get influence by scaling the dataset.

1. **Importing the necessary library of Linear regression from scikit learn package**
2. **Extracting the target column into separate vectors for training set and test set.**
   - Here we store the impendent features in variable X ('carat', 'cut', 'color', 'clarity', 'table') dependent feature/Target feature in Y variable('Price').
   - Train data will hold an independent variables whereas test data will hold a dependent variable of the dataset.
3. **Splitting data into training and test set.**
   - Inorder to perform this step, from the package sklearn.model_selection we imported train_test_split.
   - Now we split the data into 70 -30 ratio, where the train data hold 70% of the data and test data holds 30% of the data. The random state mentioned here is 1.
4. **Checking the dimensions of the training and test data.**
   Below output is retrieved from Jupter using shape command.

```
X_train (18847, 5)
X_test (8078, 5)
train_labels (18847,)
test_labels (8078,)
```

Now we have our train and test data ready. We will start building our Linear regression for model:1.

5. **Fitting the Linear regression model from sklearn linear models to Training set.**
6. **Finding the coefficient of determinants for each of the independent attributes in Jupyter using coef_ command.**

The coefficients in linear equation denote the magnitude of additive relation between the predictor and the response variable. In another word, keeping everything else fixed, to see for every one-unit increase/ decrease in X, how much does Y changes i.e., increase/decrease.

```
The coefficient for carat is 7875.192000639305
The coefficient for cut is 118.18545468751158
The coefficient for color is -261.22788912393196
The coefficient for clarity is -826.5812448828494
The coefficient for table is -19.544000671766163
```

- The above coefficients of determinants output describe among 5 independent variables, variable 'carat' has most weightage and acts as a good predictor for target variable 'price'. Ie When carat increases by 1 unit, price increases by 7875 units, keeping all other predictors constant.
- We also infer that, variable 'table', 'clarity', 'color' in this model also gives negative coefficient, and it acts as a week predictor for target variable 'price'. Variable 'cut' has some moderate weightage in predicting target variable 'price'.

7. **Intercept of the model**

The **intercept** (sometimes called the "constant") in a regression model represents the mean value of the predictor variable when all of the response variables in the model are equal to zero. If X(response) variable never equals 0, then the intercept has no intrinsic meaning and doesn't tell anything about the relationship between X and Y.

Intercept for our model is 879.4299486234354

The value for the intercept term in our model is **879.43,** which is lower than model 1 & 2. This means the average price is **879.43** when the number of (cubic zirconia) diamond is equal to zero.

Here intercept doesn't make any sense since it's not possible for a diamond to be zero priced.

8. **R Squared**

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.
The definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model.

R-squared = Explained variation / Total variation
R-squared is always between 0 and 100%

- 0% indicates that the model explains none of the variability of the response data around its mean.
- 100% indicates that the model explains all the variability of the response data around its mean.

In general, the higher the R-squared, the better the model fits your data.

Below output retrieved from Jupyter on training and testing data using score command -

R squared for training set is 0.9260028120890724
R squared for testing set is 0.9265177098296845

We see that, R squared value for train and set for this model also close to 90% and no drastic difference found compared to model 1 & model 2 and we can assume that data are close enough to the fitted regression line, but this not a reliable metrices since it might contain some statistical fluke. To arrest this statistical fluke, we will consider the adjusted R squared.

9. Evaluation of RMSE (Root Mean Square Error)

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. In general, a lower RMSD and close to 0 would indicate a perfect fit to the data. However, comparisons across different types of data would be invalid because the measure is dependent on the scale of the numbers used.

Below output retrieved from Jupyter on training and testing data from Sklearn package using metrics.mean_squared_error command -

RMSE for training set is 940.7262841964081
RMSE for testing set is 944.7253412706062

From the output we infer that, RMSE value for train & test set in this model also far away from zero, this might be because the data is not scaled, in model-4 we will evaluate and observe whether scale data change the RMSE score or not.

10. Evaluation of plot between actual and predicted price for linear regression-

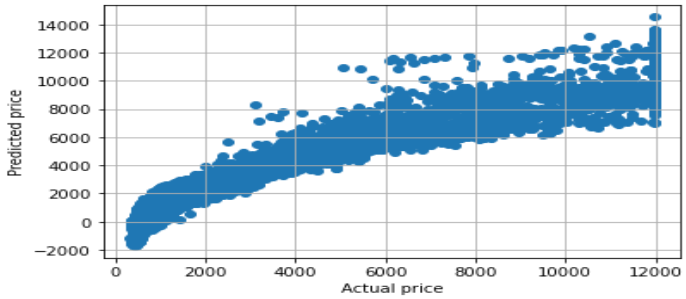


Fig 18 : Model 3 - Scatter Plot (Actual vs Predicted)

Linear Regression Model 1 using statsmodel –

Statsmodel uses OLS (ordinary least square method) to predict the best fit plane. OLS also minimizes the sum of squared error between the observed and predicted values by estimating its coefficients and bias.

One of the main difference between sci-kit Learn and statsmodel in Linear Regression is, that stats model provides a stronger emphasis on statistical inference such as adjusted R square and statistical hypothesis testing and with the help of these metrics we can compare and decide whether to drop the week attributes or not.

- **Adjusted R square :-**

Adjusted r^2=r^2- statistical fluke

- Here we remove statistical fluke from r^2 and get adjusted r^2, when we compare the model with r^2 and adjusted r^2, the adjusted r^2 will always be less than a r^2.
- When we can more redundant variables to the dataset, the r^2 will either go up or remain static and will approach to the range of (0-1), whereas adjusted r^2 will go down when we add useless dimension to the model and will get increase only when we add good dimension to our model.
- From this we conclude that, adjusted r^2 is more reliable than r^2 for evaluating a model.

- **Hypothesis testing:-**

In OLS model, to establish the reliability of the coefficients, we need hypothesis testing. Assuming Ho to be true, what is the probability of finding the coefficients found in the sample if the sample is drawn from the universe in which Ho is true.

The null hypothesis (H0) claims that there is no relation between dependent and independent variables, that means the co-efficient is 0 in the universe.

- At 95% confidence level if the p value is > = 0.5, we do not have enough evidence in data to reject the Ho and hence we believe Ho is likely to be true in the universe.
- if p value is < 0.5, we reject null hypothesis, we have enough evidence to accept Ho and believe there is relationship between dependent and independent variables.

1. **Checking the OLS summary for model 3-**

OLS Regression Results

Dep. Variable:	price	R-squared:	0.926
Model:	OLS	Adj. R-squared:	0.926
Method:	Least Squares	F-statistic:	4.716e+04
Date:	Fri, 20 May 2022	Prob (F-statistic):	0.00
Time:	14:14:27	Log-Likelihood:	-1.5578e+05
No. Observations:	18847	AIC:	3.116e+05
Df Residuals:	18841	BIC:	3.116e+05
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1706.0112	215.754	7.907	0.000	1283.115	2128.907
carat	7875.1920	16.807	468.555	0.000	7842.248	7908.136
cut	118.1855	8.534	13.850	0.000	101.459	134.912
color	-261.2279	4.237	-61.657	0.000	-269.532	-252.923
clarity	-826.5812	8.715	-94.843	0.000	-843.664	-809.499
table	-19.5440	3.615	-5.406	0.000	-26.630	-12.458

Omnibus:	2388.908	Durbin-Watson:	2.001
Prob(Omnibus):	0.000	Jarque-Bera (JB):	6839.758
Skew:	0.688	Prob(JB):	0.00
Kurtosis:	5.611	Cond. No.	1.82e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.82e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

Table 13 : Model - 3 OLS stats table

From the above OLS summary, we infer that -

- ❖ R-squared and adjusted R-squared values are same which is equal to 0.926.
- ❖ Overall p value for the model is 0.00 which is less than 0.05, this indicates that, there is some relation between dependent and independent variables.
- ❖ Standard Errors assume that the covariance matrix of the errors is correctly specified.
- ❖ The condition number is large, 1.82e+03, which is greater than model 1 & model 2. This might indicate that there is strong multicollinearity between encoded categorical variable.
- ❖ Kurtosis value found be slightly less than model 1 that is 5.995 and the model 3 value is 5.611 which indicates the dataset has slightly high positive peak tail and fatter tails than normal distribution.
- ❖ The skew value also slightly less compared to model 1 and the value is 0.688 indicated data slightly right skewed
- ❖ The Prob(JB) test value is 0.00 which indicates that, the sample data is normally distributed.

2. Evaluation of RMSE (Root Mean Square Error)

Below output is retrieved from Jupyter using math.sqrt command by predicting value of y for training & testing cases and subtracting from the actual y for the training & testing cases.

- ❖ RMSE for training set in OLS model is 940.7262841964077
- ❖ RMSE for testing set in OLS model is 944.725341270607

From the output we infer that both Sklearn and statsmodel gives the same RMSE score.

3. Checking Multi-collinearity using VIF technique

1. The variance inflationfactor(VIF) identifies correlation between independent variables and the strength of that correlation.

e. **VIF = 1 / 1 –r2**

f. As the collinearity between variables increase, r2increases, the denominator approaches 0 and as a result VIF approaches infinity.

2. VIFs start at 1 and have no upper limit.
A value of 1 indicates that there is no correlation between this independent variable and any others

e. VIFs between 1 and 5 suggest that there is a moderate correlation, but it is not severe enough to warrant corrective measures

f. VIFs greater than 5 represent critical levels of multicollinearity where the coefficients are poorly estimated, and the p-values are questionable.

Below output is retrieved from Jupyter on independent variable using the command variance_inflation_factor –

```
carat ---> 4.961312068959281
cut ---> 5.626324193587377
color ---> 3.676877410640053
clarity ---> 17.77780120020934
table ---> 25.054007989087257
```

- ❖ From the above output we infer that, compared to model 1 & model 2 VIF score got decreased to very good extend for the variable ‘carat’ ‘cut’ & color and their VIF score is less

than 5 this represent there is no critical levels of multicollinearity, and the coefficients are correctly estimated.

- Variable 'clarity' & 'table' VIF values as greater than 5 this represent critical levels of multicollinearity where the coefficients are poorly estimated, and the p-values are questionable.

Linear Equation

$$y\ (price) = m0 + m1 * carat + m2 * cut + m3 * color + m4 * clarity + m5 * table$$

```
(1706.01) * Intercept + (7875.19) * carat + (118.19) * cut + (-261.23)
* color + (-826.58) * clarity + (-19.54) * table +
```

- Variable 'carat' still holds the more weightage and acts as a good predictor for target variable 'price'. Ie When carat increases by 1 unit, price increases by 7875.19 units, keeping all other predictors constant.
- Variable 'cut' states, when variable 'cut' increases by 1 unit, price increases by 118.19 units, keeping all other predictors constant.
- Based on other performing metrics, we conclude that, From this model we can infer that, the strong multicollinearity, which was due to x, y, z & depth is reduced to greater extend.
- The VIF scores got reduced almost to 5 for most of the variables. RMSE has not shown any major change in three models. The p values for all the variables are under 0.05. Condition number is also reduced considerably.

Now we see that, the RMSE score is high, and coefficients are not balanced. So, we need to bring the variables to a comparable state in order to validate the RMSE and coefficient values. In model 4 we are fitting the scaled data to the linear model.

Model 4 : Scaling and fitting the data in to linear regression.

In this model, we are dropping the highly correlated attributes 'x','y','z', and the dataset is scaled.

Importing the necessary library of Linear regression from scikit learn package

1. **Extracting the target column into separate vectors for training set and test set.**
   - Here we store the impendent features in variable X ('carat', 'cut', 'color', 'clarity', 'table','depth') dependent feature/Target feature in Y variable('Price').
   - Train data will hold an independent variables whereas test data will hold a dependent variable of the dataset.
2. **Splitting data into training and test set.**
   - Inorder to perform this step, from the package sklearn.model_selection we imported train_test_split.
   - Now we split the data into 70 -30 ratio, where the train data hold 70% of the data and test data holds 30% of the data. The random state mentioned here is 1.
3. **Checking the dimensions of the training and test data.**
   Below output is retrieved from Jupter using shape command.

```
X_train (18847, 5)
X_test (8078, 5)
train_labels (18847,1)
test_labels (8078,1)
```

Now we have our train and test data ready. We will start building our Linear regression for model:1.

4. **Fitting the Linear regression model from sklearn linear models to Training set.**

5. **Finding the coefficient of determinants for each of the independent attributes in Jupyter using coef_ command.**

The coefficients in linear equation denote the magnitude of additive relation between the predictor and the response variable. In another word, keeping everything else fixed, to see for every one-unit increase/ decrease in X, how much does Y changes i.e., increase/decrease.

```
The coefficient for carat is 1.0474828391594044
The coefficient for cut is 0.02737027068588608
The coefficient for color is -0.12825981522265867
The coefficient for clarity is -0.20220071390379005
The coefficient for depth is -0.010055361836416428
The coefficient for table is -0.016984655817500194
```

- From the above coefficients of determinants output, we infer that among 5 independent variables, variable 'carat' has most weightage and acts as a good predictor for target variable 'price'. Ie When carat increases by 1 unit, price increases by 1.04 units, keeping all other predictors constant. Variable 'cut' increases by 1 unit, price increases by 0.027 units, keeping all other predictors constant.
- We also infer that, variable 'table', 'clarity', 'color' in this model also gives negative coefficient, and it acts as a week predictor for target variable 'price'. Variable 'cut' has some moderate weightage in predicting target variable 'price'.

6. **Intercept of the model**

The **intercept** (sometimes called the "constant") in a regression model represents the mean value of the predictor variable when all of the response variables in the model are equal to zero. If X(response) variable never equals 0, then the intercept has no intrinsic meaning and doesn't tell anything about the relationship between X and Y.

Intercept for our model is -1.7524403108336342e-16

The value for the intercept term in our model is almost equal to zero**,** which is lower than model 1, 2 & 3. This means the average price is **0** when the number of (cubic zirconia) diamond is equal to zero.

7. **R Squared**

R-squared is a statistcal measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.
The definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model.

R-squared = Explained variation / Total variation
R-squared is always between 0 and 100%

- 0% indicates that the model explains none of the variability of the response data around its mean.
- 100% indicates that the model explains all the variability of the response data around its mean.

In general, the higher the R-squared, the better the model fits your data.

Below output retrieved from Jupyter on training and testing data using score command -

```
R squared for training set is 0.9260825413502574
R squared for testing set is 0.9266125346369597
```

We see that, R squared value for train and set for this model also close to 90% and no drastic difference found compared to model 1, 2 & 3 and we can assume that data are close enough to the fitted regression line, but this not a reliable metrices since it might contain some statistical fluke. To arrest this statistical fluke, we will consider the adjusted R squared.

8. Evaluation of RMSE (Root Mean Square Error)

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. In general, a lower RMSD and close to 0 would indicate a perfect fit to the data. However, comparisons across different types of data would be invalid because the measure is dependent on the scale of the numbers used.

Below output retrieved from Jupyter on training and testing data from Sklearn package using metrics.mean_squared_error command -

```
RMSE for training set is 0.2718776538256549
RMSE for testing set is 0.27090120960054836
```

From the output we infer that, RMSE value for train & test set in this model is close to zero.

9. Evaluation of plot between actual and predicted price for linear regression-

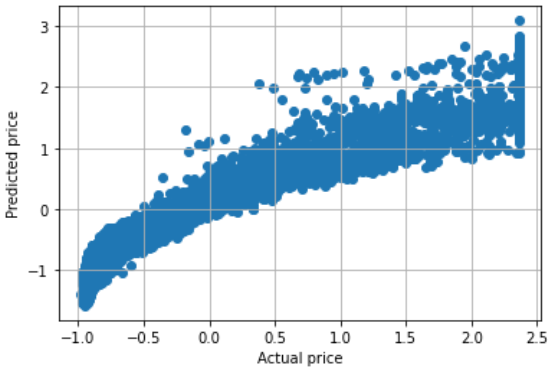


Fig 19 : Model 4 - Scatter Plot (Actual vs Predicted)

From the above scatter plot we infer that, both model 1 2 & 3 data points got slightly scattered but doesn't not look cloudy. This indicates that, our model did reasonable prediction and forms more or less a positive linear line.

Linear Regression Model 4 using statsmodel –

Statsmodel uses OLS (ordinary least square method) to predict the best fit plane. OLS also minimizes the sum of squared error between the observed and predicted values by estimating its coefficients and bias.

One of the main difference between sci-kit Learn and statsmodel in Linear Regression is, that stats model provides a stronger emphasis on statistical inference such as adjusted R square and statistical

hypothesis testing and with the help of these metrics we can compare and decide whether to drop the week attributes or not.

- **Adjusted R square :-**

Adjusted r^2=r^2- statistical fluke

- ❖ Here we remove statistical fluke from r^2 and get adjusted r^2, when we compare the model with r^2 and adjusted r^2, the adjusted r^2 will always be less than a r^2.
- ❖ When we can more redundant variables to the dataset, the r^2 will either go up or remain static and will approach to the range of (0-1), whereas adjusted r^2 will go down when we add useless dimension to the model and will get increase only when we add good dimension to our model.
- ❖ From this we conclude that, adjusted r^2 is more reliable than r^2 for evaluating a model.

- **Hypothesis testing:-**

In OLS model, to establish the reliability of the coefficients, we need hypothesis testing. Assuming Ho to be true, what is the probability of finding the coefficients found in the sample if the sample is drawn from the universe in which Ho is true.

The null hypothesis (H0) claims that there is no relation between dependent and independent variables, that means the co-efficient is 0 in the universe.

- ❖ At 95% confidence level if the p value is > = 0.5, we do not have enough evidence in data to reject the Ho and hence we believe Ho is likely to be true in the universe.
- ❖ if p value is < 0.5, we reject null hypothesis, we have enough evidence to accept Ho and believe there is relationship between dependent and independent variables.

1. **Checking the OLS summary for model 4-**

OLS Regression Results

Dep. Variable:	price	R-squared:	0.926
Model:	OLS	Adj. R-squared:	0.926
Method:	Least Squares	F-statistic:	3.934e+04
Date:	Sun, 22 May 2022	Prob (F-statistic):	0.00
Time:	05:26:59	Log-Likelihood:	-2196.3
No. Observations:	18847	AIC:	4407.
Df Residuals:	18840	BIC:	4462.
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-3.209e-17	0.002	-1.62e-14	1.000	-0.004	0.004
carat	1.0475	0.002	468.766	0.000	1.043	1.052
cut	0.0274	0.002	11.417	0.000	0.023	0.032
color	-0.1283	0.002	-61.387	0.000	-0.132	-0.124
table	-0.0170	0.002	-6.816	0.000	-0.022	-0.012
depth	-0.0101	0.002	-4.508	0.000	-0.014	-0.006
clarity	-0.2022	0.002	-94.302	0.000	-0.206	-0.198

Omnibus:	2395.483	Durbin-Watson:	2.002
Prob(Omnibus):	0.000	Jarque-Bera (JB):	6827.675
Skew:	0.691	Prob(JB):	0.00
Kurtosis:	5.605	Cond. No.	2.20

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Table 14 : Model - 4 OLS stats table

From the above OLS summary, we infer that -

- ❖ R-squared and adjusted R-squared values are same which is equal to 0.926.
- ❖ Overall p value for the model is 0.00 which is less than 0.05, this indicates that, there is some relation between dependent and independent variables.
- ❖ Standard Errors assume that the covariance matrix of the errors is correctly specified.
- ❖ Here we can see condition number got reduced to a max.
- ❖ Kurtosis value found be slightly less 5.605 compared with other 3 models, kurtosis here indicates the dataset has slightly high positive peak tail and fatter tails than normal distribution.
- ❖ The skew value also slightly less compared to model 3 and the value is 0.691 indicated data slightly right skewed
- ❖ The Prob(JB) test value is 0.00 which indicates that, the sample data is normally distributed.

2. Evaluation of RMSE (Root Mean Square Error)

Below output is retrieved from Jupyter using math.sqrt command by predicting value of y for training & testing cases and subtracting from the actual y for the training & testing cases.

- ❖ RMSE for training set in OLS model is 0.2709012096005484
- ❖ RMSE for testing set in OLS model is 0.27187765382565576

From the output we infer that both Sklearn and statsmodel gives the same RMSE score.

3. Checking Multi-collinearity using VIF technique

1. The variance inflationfactor(VIF) identifies correlation between independent variables and the strength of that correlation.

g. **VIF = 1 / 1 –r2**

h. As the collinearity between variables increase, r2increases, the denominator approaches 0 and as a result VIF approaches infinity.

2. VIFs start at 1 and have no upper limit.
A value of 1 indicates that there is no correlation between this independent variable and any others

g. VIFs between 1 and 5 suggest that there is a moderate correlation, but it is not severe enough to warrant corrective measures

h. VIFs greater than 5 represent critical levels of multicollinearity where the coefficients are poorly estimated, and the p-values are questionable.

Below output is retrieved from Jupyter on independent variable using the command variance_inflation_factor –

```
carat ---> 1.2734251828224157
cut ---> 1.4663781347483635
color ---> 1.1118919479125764
clarity ---> 1.1704646394510017
depth ---> 1.2696184119559804
table ---> 1.5823293378075485
```

- ❖ From the above output we infer that, compared to model 1, model 2 & model 2 VIF score got decreased to very good extend for the variable which is less than 5 this represent there is no critical levels of multicollinearity, and the coefficients are correctly estimated.

4. Linear Equation

$$y\ (price) = m0 + m1 * carat + m2 * cut + m3 * color + m4 * table + m5 * depth + m6 * clarity$$

```
(-0.0) * Intercept + (1.05) * carat + (0.03) * cut + (-0.13) * color +
(-0.02) * table + (-0.01) * depth + (-0.2) * clarity +
```

- Variable 'carat' still holds the more weightage and acts as a good predictor for target variable 'price'. Ie When carat increases by 1 unit, price increases by 1.05 units, keeping all other predictors constant.
- Variable 'cut' states, when variable 'cut' increases by 1 unit, price increases by 0,03 units, keeping all other predictors constant.
- When 'color' decreases by 1 unit, price decreases by -0.13 units, keeping all other predictors constant.
- Based on other performing metrics, we conclude that, strong multicollinearity, got reduced to greater extend.
- The VIF scores got reduced almost to 5 for most of the variables. RMSE has not shown any major change in three models. The p values for all the variables are under 0.05. Condition number is also reduced considerably.

Now we see that, the RMSE score is high, and coefficients are not balanced. So, we need to bring the variables to a comparable state in order to validate the RMSE and coefficient values. In model 4 we are fitting the scaled data to the linear model.

Model Comparison :-

Model 1 : Considering all the variables in the dataset and fitting in to linear regression.

Model 2 : Dropping high multicollinearity attributes of independent variables 'x', 'y','z' and fitting in to linear regression.

Model 3 : Dropping high multicollinearity attributes of independent variables 'x', 'y','z' and fitting in to linear regression.

Model 4 : Scaling and fitting the data in to linear regression.

	Model 1	Model 2	Model 3	Model 4
Intercept	1105.00	3101.32	879.43	-0.00
Rsq_train	0.93	0.93	0.93	0.93
Rsq_test	0.93	0.93	0.93	0.93
Adj_Rsq	0.93	0.93	0.93	0.93
RMSE_train	932.58	940.22	940.73	0.27
RMSE_test	936.36	943.99	944.73	0.27
Stats_Rsq_train	869712.43	884012.42	884965.94	0.07
Stats_Rsq_test	876764.27	891116.68	892505.97	0.07
Stats_RMSE_train	932.58	940.22	940.73	0.27
Stats_RMSE_test	936.36	943.99	944.73	0.27
VIF_Max	1202.41	519.67	17.29	1.59
VIF_Min	3.71	3.70	3.68	1.11

Table 15 : Model Comparison (4 models)

Inferences:

- Accuracy (R square) is same for all models for both sklearn as well as stat models.
- R-Square and adjusted R square gives the same value.
- All the four model give us the best R-squared values.
- Model 4 gives the best RMSE score, this is because of the scaled data.
- VIF max and VIF min values are lowest for models 5,since that data is scaled.
- Compared to 4 models, model 4 gives the reasonable intercept value.

Here we conclude that, our model doesn't perform more effectively, this is due to high multicollinearity between independent variables and useless variables and as per the size of the rows, most of the features seems to be useless, due to the lack of feature model didn't perform great extend to predict the price. Still, we violated some of the linear model assumptions and created 4 models.

As per comparison of 4 models, we chose Model 3 moderately to be our best fit model since this model gives the best VIF & RMSE score and reasonable co-efficient for most variables and other parameters doesn't give any drastic difference with other 3 models.

Model 4 – Final Linear Equation

$$y\ (\text{price}) = m0 + m1 * \text{carat} + m2 * \text{cut} + m3 * \text{color} + m4 * \text{table} + m5 * \text{depth} + m6 * \text{clarity}$$

```
(-0.0) * Intercept + (1.05) * carat + (0.03) * cut + (-0.13) * color +
(-0.02) * table + (-0.01) * depth + (-0.2) * clarity +
```

Variable 'carat' still holds more weightage and acts as a good predictor for target variable 'price'

- When carat increases by 1 unit, price increases by 1.05 units, keeping all other predictors constant.
- when variable 'cut' increases by 1 unit, price increases by 0,03 units, keeping all other predictors constant.
- When 'color' decreases by 1 unit, price decreases by -0.13 units, keeping all other predictors constant.
- When 'table' decreases by 1 unit, price decreases by -0.02 units, keeping all other predictors constant.
- When 'depth' decreases by 1 unit, price decreases by -0.01 units, keeping all other predictors constant.
- When 'clarity' decreases by 1 unit, price decreases by -0.02 units, keeping all other predictors constant.

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

According to the problem statement, Gemstones co ltd, a cubic zirconia manufacturer earns varying profits on different pricing slots. The company wants to predict the stone's price based on the data provided so that the company may distinguish between higher profitable and lower profitable stones and maximise the company's profit share. Also require the top five attributes which are most essential in price prediction.

Here we have thoroughly examined the historical data and developed a model that predicts different price slots based on the characteristics. We first look at the key points in our past data suggest some recommendations to the firm.

To have a better profit share, the business value is to distinguish between higher profitable stones and lower profitable stones. Our model has more than 90% of an accuracy score and that can be acceptable in this business, and properly predicted the price for more than 90% of the stones.

Following are the insights and recommendations to help the firm to solve the business objective:

Carat: Carat weight of the cubic zirconia Insights:

- ❖ As per analysis we found that variable Carat is the best predictor for the price.
- ❖ It has positive linear relationship with price. The price increases when carat of zirconia stone increases.
- ❖ Carat is measure of weight which has direct correlation with physical dimensions (x, y, z).

Recommendations:

- ✓ According to the best fit model, variable carat is the best predictor in terms of price.
- ✓ The firm can manufacture the zirconia stone with high carat value and advertise to high class customer with some luxury benefits.
- ✓ For low budgets should be offered lower carat stones or we can introduce yearly savings for low budget customers with some presentation, so that they will pay the monthly instalments and might buy the high carat stones. On the other hand, firm will get interest during the monthly instalments, this will be double beneficial to the firm.

Cut: Quality of the cubic zirconia

- ❖ For cut attribute, we see that Ideal cut type is the most selling and the average price of Ideal is slightly less prices compared to premium cut type which is slightly more expensive.
- ❖ ‘Fair’ and ‘Good’ have a lower count of sales and have a relatively higher average price.
- ❖ The ideal, premium, very good cut types have better profits.

Recommendations:

- ✓ Cut quality ideal, premium, very good cut types are the one which brings more profits, proper marketing of the products may increase the sales to greater extend.
- ✓ The best quality cut, 'Ideal,' has a lower average price comparatively. However, their value counts are high. The firm can try and increase the price of the ideal cut category a little to see whether it affects the sales or not. If sales got reduced, they can return to the current marking price or can increase their carat value.
- ✓ As per analysis, 'Fair' & 'Good' cut quality has low counts and small quantities got sold, but their average price is comparably high compared to other cuts. The firm can attempt to lower the average price or increase the quality of these cuts so that customers might try to avail 'Fair' & 'Good' cut quality stones.
- ✓ ‘Fair’ and ‘Good’ cut types is advisable to eschew as the number of sales and profits are very less.

‘X’, ‘Y’ & ‘Z’ - length, width, and height of the cubic zirconia

- ❖ X, Y and Z having linear relation among each other and with target variable 'price'.
- ❖ All three have a strong relation to the price variable, this relation might, change the price value. At the same time, these variables end up causing a high multicollinearity, which affect the performance of our price prediction

Recommendations:

- ✓ The dimensions are having negative effect on the stones, smaller dimension's i.e., balanced size is more expensive.
- ✓ If a stone has smaller dimensions along with high carat value and superior clarity, their value will increase more than a high dimensions stone with low carat & clarity.
- ✓ Firm can focus more on balancing the stone size for high quality stones.

Depth and Table

- ❖ Bothe the variable Depth and table are poor predictors in terms of price.
- ❖ From EDA analysis we observed that variable depth & table has no defined relationship with target variable 'price' and its spreads looks like a cloud, which is not useful for model building.

Clarity: Absence of the Inclusions and Blemishes

- ❖ S1 variable is most expensive category in terms of absence of the Inclusions and Blemishes and has emerged as a strong predictor of price, followed by VS2 and S2 which are good category stones in terms of price, but I1 and lF are the cheap stones and might have presence of Inclusions and Blemishes .
- ❖ S1 type of clarity is the most selling category followed by VS2 and I1 being the least selling category.
- ❖ Clarity of stone types Sl1, VS2 and Sl2 has high value counts this variable might help the firm to gain more profit, so firm can make expensive price cap for these stone.

Recommendations:

- ✓ Price 'I1' has highest mean value in terms of price, but lowest sales rate.
- ✓ Firm can take little risk by lowering its price for a period of time, check for sales grow, if it goes up, then they can raise the price to the former level.
- ✓ 'VVS',' 'VVS1', and 'VVS2' are performing better in price prediction than other I1 & I1. The firm should put greater emphasis on them to increase the sale.

Color

- ❖ Most of the people opted for G color gem, they are more expensive and has high selling. Color 'E' and 'F' nearly falling into the same range
- ❖ 'J' color gem are the least selling stone.

Recommendations:

- ✓ Color of the stones H, I, and J, will not help the company if they cap them for high price.

- ✓ Instead, the firm should concentrate on stones that has D, E, and F colors, in order to fetch for high prices and for boosting the sales.
- ✓ This might be a signal for the firm to explore unique color stones, such as transparent stones that has Absence of the Inclusions and Blemishes this might help in boosting the pricing & sales.
- ✓ 'J' and 'I' color stones can be priced lower. This might make the customers to get attracted and sales might get increased.

As per EDA & model performance, the best 5 attributes that makes good prediction in terms of price are as follows:

1. Carat
2. Clarity
3. Color
4. Cut
5. Table

Key performance indicators:

- ❖ Sales promotion can be effective in terms of changing short terms buyers' behaviour.
- ❖ Advertisement is one of the best way in terms of reaching/covering more people & potential buyers. For example, advertising campaign & discount can be offered during festival & occasional days, i.e., New year, Christmas, Valentine's Day, Mother's Day, etc.
- ❖ The company can make a segment to target the customer based on their income/paying Capacity and give offers & discounts accordingly.
- ❖ A virtual point system could be used in which customers can get certain number of points during the time of high stone purchase and need to allow the customers to avail the gained points at the time of next purchase.
- ❖ We can introduce yearly savings for high & low budget customers based on their income with some presentation & making charge reduction, so that they will pay the monthly instalments and might buy the high carat stones. On the other hand, firm will get interest during customers monthly instalment time, this will be double beneficial to the firm
- ❖ Customization of products can be initiated for better sales.
- ❖ Firm can introduce online shopping with more collection and can provide some discounts & exclusive offer for low selling stone. In today's busy schedule most of the people will opt for online shopping rather than in door shopping. This might increase the sales are of low selling stones.

Tour & Travel Agency Analysis - Logistic Regression & LDA

Table of Contents

List of Tables……48

List of Figures……48

Problem Statement 2……..48

Questions:-

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….49

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)….58

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……63

2.4 Inference: Basis on these predictions, what are the insights and recommendations……..72

Tabel List:

Table 1 : Sample Data ……49
Table 2 : Descriptive Statistics……51
Table 3 : Label encoded table (categorical columns) ……59
Table 4 : Probability cut-off range table (0.1 - 0.9) ……63
Table 5 :Logistic Regression model comparison table (Train & Test) ……66
Table 6 : LDA model comparison table – Train & Test (cut off 0.5)……66
Table 7: LDA model comparison table – Train & Test (cut off 0.4)……69
Table 8: Comparison of cut-off probability(0.5 and 0.4) ……70
Table 9: Comparison of LR & LDA model performance……70

List of Figure:

Fig 1 : Box & hist Plot – Univariant Analysis (Salary) ……32
Fig 2 : Box & hist Plot – Univariant Analysis (age)……32
Fig 3 : Box & hist Plot – Univariant Analysis (edu) ……53
Fig 4 : Box & hist Plot – Univariant Analysis (no_young_children) ……53
Fig 5 : Box & hist Plot – Univariant Analysis (no_older_children) ……54
Fig 6 : Count Plot - Univariant Analysis (Holiday package & foreign)……54
Fig 7 : Count Plot - Univariant Analysis (no_young_children & no_older_children)……54
Fig 8 : Box Plot - Bivariant Analysis (Numeric columns) ……55
Fig 9 : Count Plot - Bivariant Analysis (Categorical columns) ……56
Fig 10 : Pair Plot - Multivariant Analysis (continuous columns)……57
Fig 11 : Heatmap Plot – Correlation analysis (continuous columns) ……58
Fig 12 : Logistic Regression Confusion Matrix ……65
Fig 13: Logistic Regression Classification report ……65
Fig 14: Logistic Regression of ROC curve ……65
Fig 15 : LDA Confusion Matrix(cut-off 0.5) ……67
Fig 16: LDA Classification report (cut-off 0.5)……67
Fig 17: LDA ROC curve(cut-off 0.5) ……67
Fig 18 : LDA Confusion Matrix (cut-off 0.4)……68
Fig 19: LDA Classification report(cut-off 0.4) ……69
Fig 20: LDA ROC curve (cut-off 0.4)……69
Fig 20: Comparison of LR & LDA ROC curve……71

Problem 2 - Logistic Regression and LDA

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 Dictionary

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.

The objectives of EDA can be summarized as follow:

- Maximize insight into the data/understand the data structure.
- EDA is an approach to analyse date using non-visual and visual techniques.
- EDA involves through analyse of data to understand the current business situation.
- EDA objective is to extract "'Gold' from the "'Data mine'' based on domain understanding.

As a first step, importing all the necessary libraries, we think that will be requiring to perform the EDA.

Loading the data set – Loading the ' **Holiday_Package.csv '** file using pandas. For this we will be using read excel file.

EDA Exploration: Following is the output from Jupyter.

Head of the dataset: After reading the CSV file, the head command with Transpose option gives the bellow output.

	Unnamed: 0	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
0	1	no	48412	30	8	1	1	no
1	2	yes	37207	45	8	0	1	no
2	3	no	58022	46	9	0	0	no
3	4	no	66503	31	11	2	0	no
4	5	no	66734	44	12	0	2	no

Table 1 : Sample Data

From the above head table, we infer that,

Dataset contains of 11 variables such as ‘Unnamed: 0', 'Holliday_Package', 'Salary', 'age', 'educ','no_young_children', 'no_older_children', 'foreign'. Variable **‘Unnamed: 0'** is not useful for our analysis and it will be dropped in future.

There are 7 Independent variables and 1 dependent variable as ‘Holliday_Package’.

Shape of the dataset : Output from shape command is –

```
The dataset has 872 rows and 8 columns.
```

info() is used to check the Information about the data and the datatypes of each respective attributes: Output from Info command is –

```
<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
```

From Info() function we infer that, the dataset has 872 instances with 7 attributes. 5 integer type and 2object type (Strings in the column).

Duplication check : Output from duplicated with sum command is –

```
The dataset has 0 duplication.
```

Null value check : Output from isnull with sum command is –

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

Dataset doesn’t contain any null values.

Descriptive Analytics : Describe method will help us see how data is spread for the numerical values, also we can see the minimum value, mean values, different percentile values and maximum values.

Output from Describe with Transpose option is –

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Holliday_Package	872	2	no	471	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Salary	872.0	NaN	NaN	NaN	47729.172018	23418.668531	1322.0	35324.0	41903.5	53469.5	236961.0
age	872.0	NaN	NaN	NaN	39.955275	10.551675	20.0	32.0	39.0	48.0	62.0
educ	872.0	NaN	NaN	NaN	9.307339	3.036259	1.0	8.0	9.0	12.0	21.0
no_young_children	872.0	NaN	NaN	NaN	0.311927	0.61287	0.0	0.0	0.0	0.0	3.0
no_older_children	872.0	NaN	NaN	NaN	0.982798	1.086786	0.0	0.0	1.0	2.0	6.0
foreign	872	2	no	656	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 2:Descriptive Statistics

From the above descriptive table, we infer that,

- ❖ Among 8 attributes there are 6 features that are in numbers and remaining 2 attributes are object type.
- ❖ Variable 'Holliday_Package' has 2 unique values in that 'no' has top value count of 471, In variable 'foreign' has 2 unique values in that 'no' has top value count of 656.
- ❖ Except variable 'Salary', there is no drastic change in terms of min and max value, and this indicates variable 'Salary' is affected with outliers.
- ❖ Based on descriptive summary, we infer that data looks good, also we see that for most of the variables the mean/medium are nearly equal to each other.

Checking Value counts for Categorical Column - Holiday_Package & foreign

Below output is retrieved from Jupyter using value_counts() command –

- **Holiday_Package**

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

From the above observation we infer that, variable 'Holiday_Package has 2 label i.e., no & yes.

- ❖ There are 471 employee who opted 'no' for Holliday package.
- ❖ There are 401 employee who opted 'yes' for Holliday package.

- **Foreign**

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

From the above observation we infer that, variable 'foreign' has 2 label i.e., no & yes.

- ❖ There are 656 non foreign employees observed in the dataset.
- ❖ There are only 216 foreign employees observed in the dataset.

Univariant Analysis :- Histogram & Boxplot (Numeric Columns)

The objective of univariant analysis is to derive the data, define, analyze and summarize the pattern present in it. In a dataset, it explores each variable separately such as Numerical variable and Categorical variable. Some of the patterns that can be easily identified with univariant analysis are Central Tendency (mean, mode and median), Dispersion (range, variance), Quartiles (interquartile range), and Standard deviation. Univariant analysis can be described and visualize with the help of most used plots of Histogram/Distplot and Barplot.

Column : Salary (Employee salary)

```
Skewness of Salary: 3.10
Kurtosis of Salary: 15.85
Outliers of Salary: 0.57
```

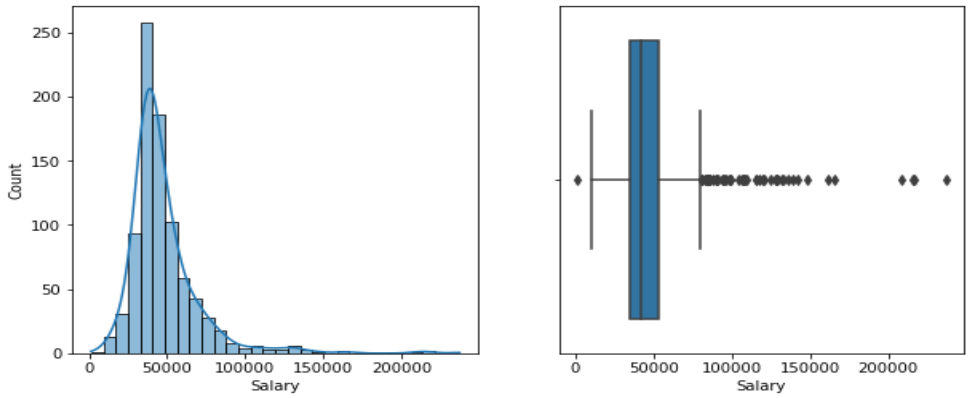


Fig 1 : Box & hist Plot – Univariant Analysis (Salary)

Observation:-

- From the above graphs, skewness & Kurtosis score, we can infer that distribution of ' advance_payments ' is slightly right skewed and has positive kurtosis.
- The histplot infer that variable'Salary' range of the tail extends up to 200000.
- Boxplot denotes variable 'Salary' has 0.57% of outliers.

Column : age (Age in years)

```
Skewness of age: 0.15
Kurtosis of age: -0.91
Outliers of age: 0.0
```

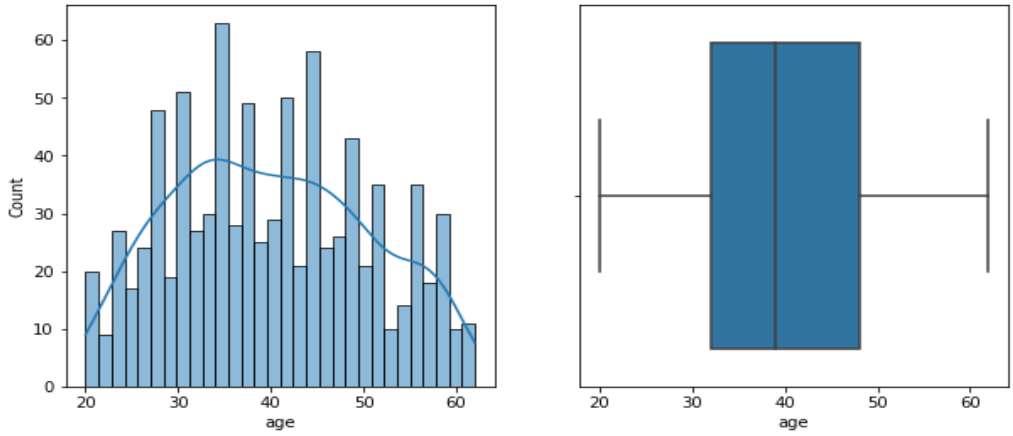


Fig 2 : Box & hist Plot – Univariant Analysis (age)

Observation:-

- From the above graphs, skewness & Kurtosis score, we can infer that distribution of ' age' almost normally distributed and has slightly negative kurtosis.
- The histplot infer that variable'Salary' range of max tail extends up to 62.
- Boxplot denotes variable 'Salary' has no outliers.

Column : educ (years of formal education)

```
Skewness of educ: -0.05
Kurtosis of educ: 0.01
Outliers of educ: 0.04
```

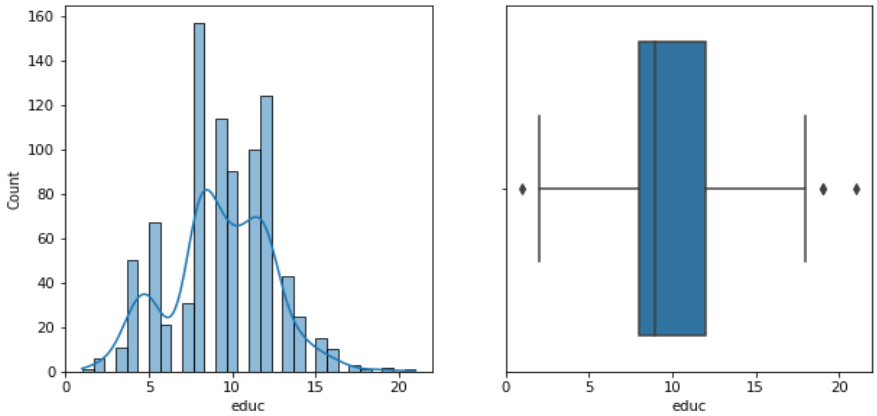


Fig 3 : Box & hist Plot – Univariant Analysis (edu)

Observation:-

- From the above graphs, skewness & Kurtosis score, we can infer that distribution of ' educ ' is slightly left skewed and has normal kurtosis.
- The histplot infer that 'educ' range of the tail extends up to 21.
- Boxplot denotes variable 'Salary' has few outliers ie 0.04%.

Column : no_young_children (The number of young children (younger than 7 years))

```
Skewness of no_young_children: 1.95
Kurtosis of no_young_children: 3.11
Outliers of no_young_children: 2.07
```

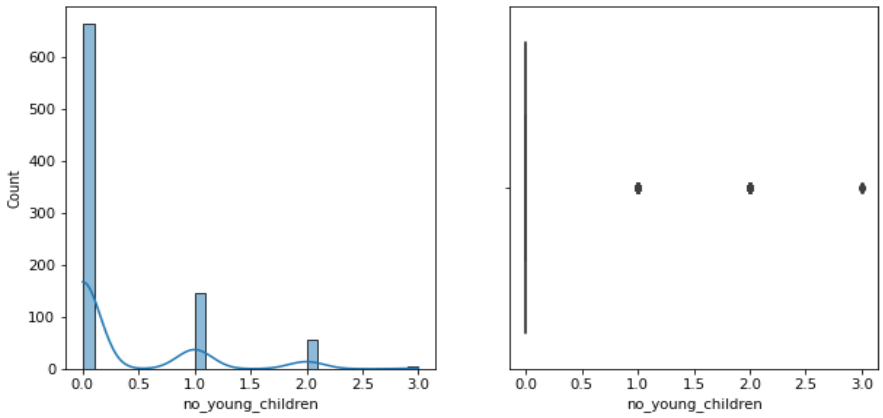


Fig 4 : Box & hist Plot – Univariant Analysis (no_young_children)

Observation:-

- From the above graphs, we infer that, variable 'no_young_children' is not a continuous numerical values instead it considered as discrete/categorical value that ranges from 0,1,2,3.

Column : no_older_children (Number of older children))

```
Skewness of no_older_children: 0.95
Kurtosis of no_older_children: 0.68
Outliers of no_older_children: 0.02
```

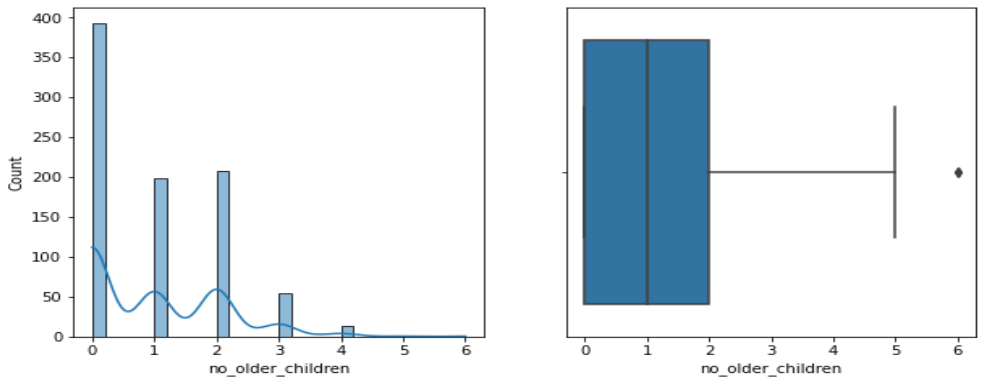


Fig 5 : Box & hist Plot – Univariant Analysis (no_older_children)

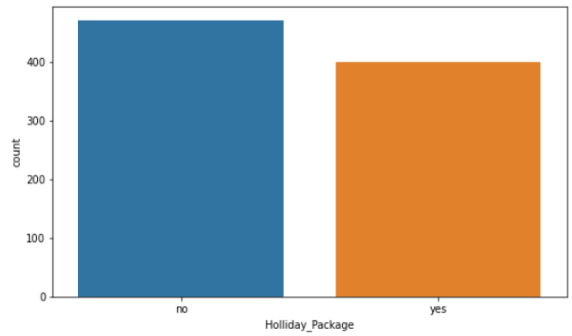
Observation:-

- From the above graphs, we infer that, variable 'no_older_children' is not a continuous numerical values instead it considered as discrete/categorical value that ranges from 0,1,2,3,5,6.

Univariant Analysis :- Count Plot (Categorical Features)

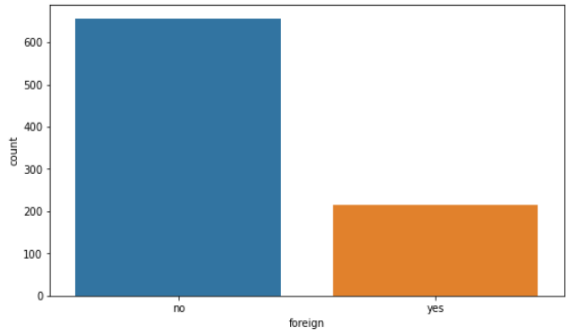
A count plot is kind of histogram or a bar graph used to visualize the categorical variables.

Column : Holliday_Package
(Opted for Holiday Package yes/no?)




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

Column : foreign
(foreigner Yes/No)




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

Fig 6 : Count Plot - Univariant Analysis (Holiday package & foreign)

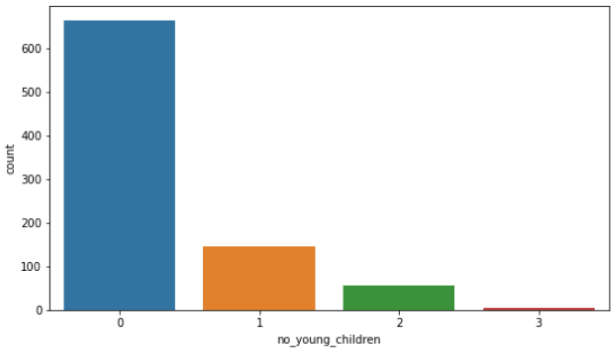
Observation :-

From the above countplot we infer that,

- There's only 0.70 % difference between the employee who opted for Holliday_Package and the employee who not opted for Holliday_Package.

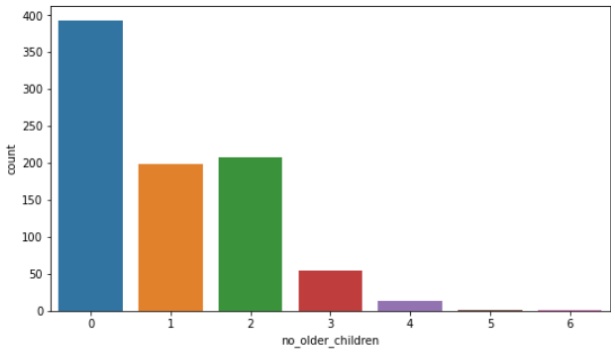
- Variable 'foreign' states, employees who are foreigner is very less from employee who are non-foreginer. Ie is because most the employees belong to their own state.

Column: no_young_children
(The number of young children (younger than 7 years)




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

Column: no_older_children
(Number of older children)




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

Fig 7 : Count Plot - Univariant Analysis (no_young_children & no_older_children)

Observation:-

From the above count plot for the variable 'no_young_children' & no_older_children,

- There are 665 employee who doesn't have younger than 7 years child that is quite higher than employee who has 7-year younger child. There are only 5 employees who have 3 children, and they are younger than 7 years.

- There are 393 employee who doesn't have older children, that is quite high than employee who has older child. There are 208 employees who have 2 older children, there are 4 employees who have 5 -6 older children.

Bivariant Analysis:- Boxplot

Bi-variant analysis of Target ('Holliday_package') vs continuous variables –

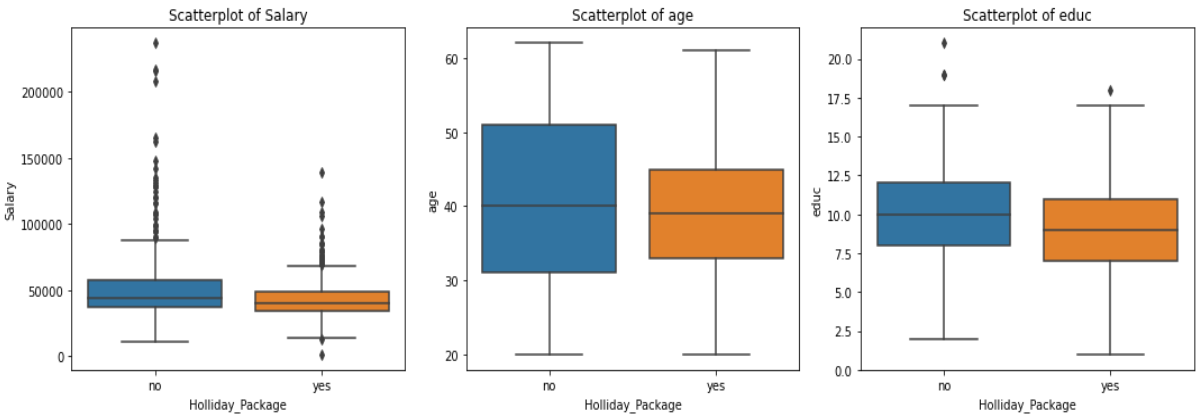


Fig 8 : Box Plot - Bivariant Analysis (Numeric columns)

Observation:-

- ❖ Employee who doesn't opt for holliday package, earns more salary than the employees who opt for the holliday package.

- ❖ As per age feature we see that, employees who are opting for holiday package and not opting are similar in nature, though the number of people who opted are less than who doesn't opted for holliday package, but they mostly fall in range of 35-45 age group.

- ❖ In terms of 'edu', we see that years of formal education count is showing a similar pattern for the employees who opted and not opted for holliday package. This means education is likely not a variable that influences for opting of holiday packages for employees.

Bivariant Analysis:- Countplot (Categorical columns)

Bi-variant analysis of categorical variables(foreign, no_young_children & no_older_children) with Target ('Holliday_package') as Hue element –

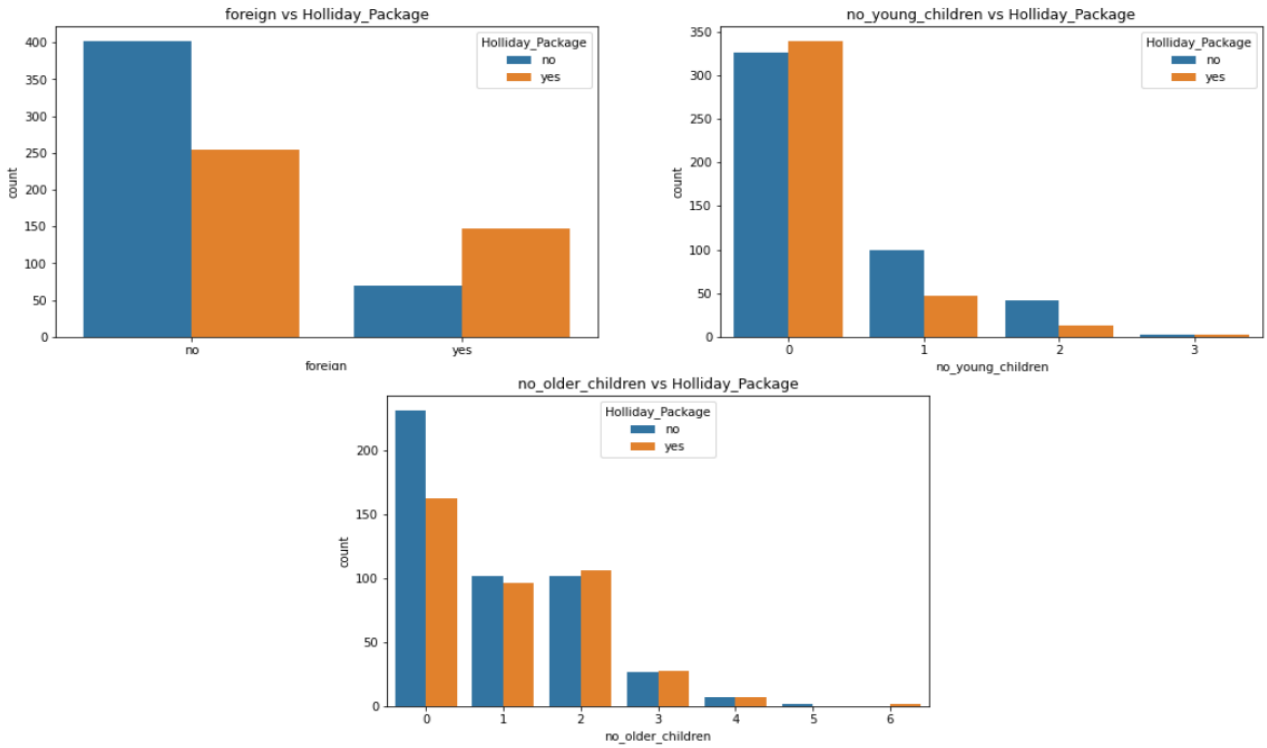


Fig 9 : Count Plot - Bivariant Analysis (Categorical columns)

Observation:-

From the above countplot we infer that,

- ❖ Most of employee who not opted for holliday package are non-foreigner and there are very few foreign employees who opted for holliday package.
- ❖ Employees who have younger than 7-year children and opted for holliday package are less compared to the employees who doesn't have younger than 7-year child and they are quite high.

- In terms of older children, employees who doesn't have older children and opted for holliday package are high then employees who have older children.

Multivariant Analysis :- Pair Plot & Heat Map

Pair Plot:- A pair plot gives us correlation graphs between all numerical variables in the dataset. Thus, from the graphs we can identify the relationships between all numerical variables.

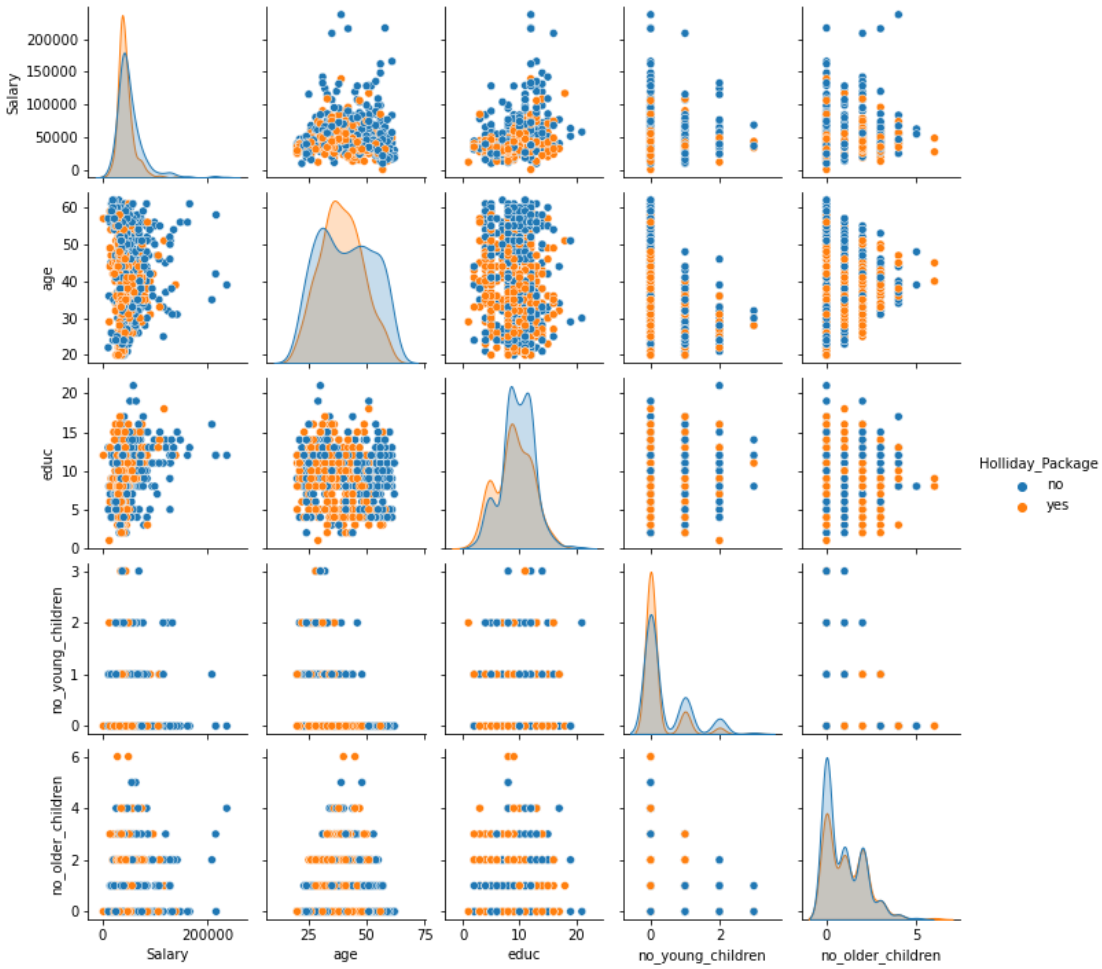


Fig 10 : Pair Plot - Multivariant Analysis (continuous columns)

Observation :-

- From the above displot diagonals & off diagonals scatter plot we infer that for all the variables, the distribution & scatter points got overlapped, this is because variables fail to differentiate 'no' & 'yes' classes of target variable.
- As per multicollinearity we infer that, almost of the variables are not correlation to each other.

Heatmap:-

A heatmap gives us the correlation between numerical variables. If the correlation value is tending to 1, the variables are highly positively correlated whereas if the correlation value is close to 0, the variables are not correlated. Also, if the value is negative, the correlation is negative. That means, higher the value of one variable, the lower is the value of another variable and vice-versa.

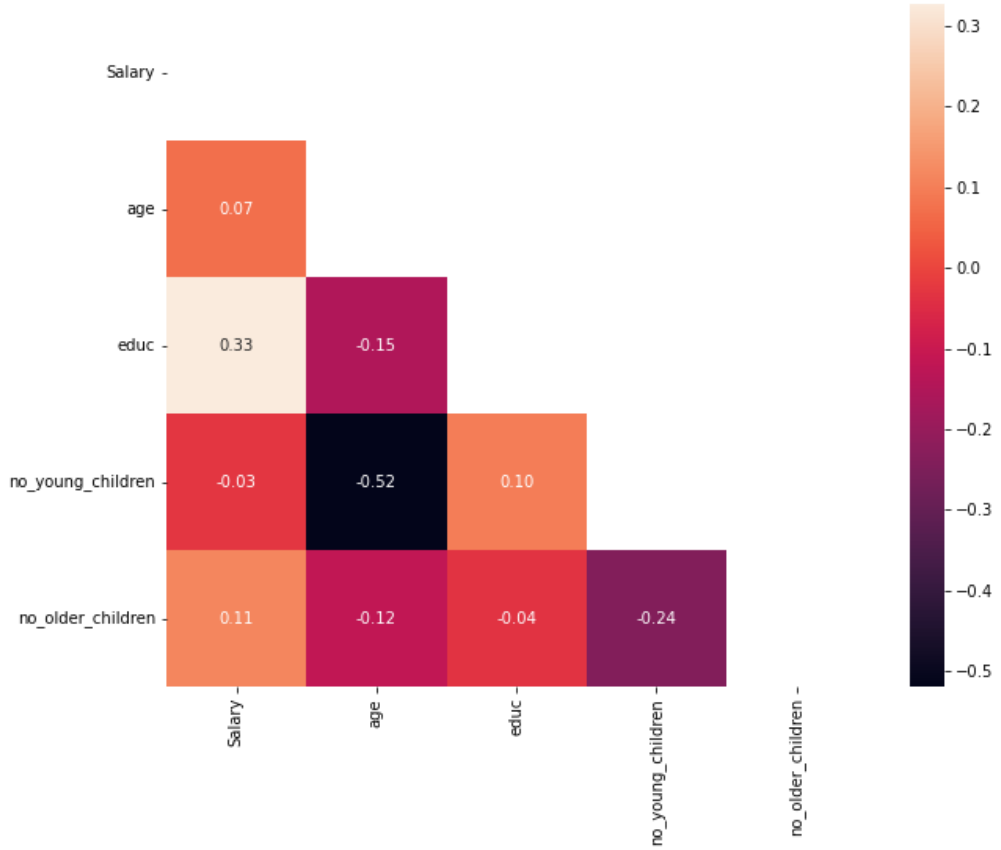


Fig 11 : Heatmap Plot – Correlation analysis (continuous columns)

Observation:-

- There is a weak correlation found between the variables 'Salary & educ (0.33).
- Between variable Salary & age, Salary & no_older_children and educ &no_younger_children correlation is very poor and remaining combination shows negative correlation.

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).

1. **Feature Engineering**

Feature engineering is a technique used to encode categorical features into numerical values so that machine learning algorithm can understand. Most popular categorical converting technique is One hot encoding or Label encoding. Here, we use label encoding for categorical values to converted into simple numerical values without losing an information. During Label encoding all categorical features are labelled in a numeric values by alphabetical order.

Below is the output retrieved from Jupyter:-

```
feature: Holliday_Package
['no', 'yes']
Categories (2, object): ['no', 'yes']
[0 1]


feature: foreign
['no', 'yes']
Categories (2, object): ['no', 'yes']
[0 1]
```

2. **Checking the head of the dataset after label encoding:-**

	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

Table 3 : Label encoded table (categorical columns)

3. **Checking the Info of the dataset after label encoding:-**

```
<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    int64
 5   no_older_children  872 non-null    int64
 6   foreign            872 non-null    int8
dtypes: int64(5), int8(2)
memory usage: 35.9 KB
```

By checking the head and info of the dataset after label encoding all object types got converted to number.

i. **Checking the proportion of observations using Target variable 'Holliday_Package'**

	Holliday_Package
0	0.54
1	0.46

By observing an above output retrieved from Jupyter, we can say that there is no issue of class imbalance, and we have reasonable proportions in both the classes, the dataset is now ready for train and test.

j. Outlier Treatment

Here we are not treating the outlier treatment for the most affected 'Salary' column since there seems to be genuine, employee can earn a salary more than 1lk. We build a model without treating an outlier.

k. Extracting the target column into separate vectors for training set and test set.

- Here we store the independent features in variable X ('Salary', 'age', 'educ', 'no_young_children', 'no_older_children', 'foreign') and dependent feature/Target feature in Y variable('Holliday_package').
- Train data will hold an independent variables whereas test data will hold a dependent variable of the dataset.

l. Splitting data into training and test set.

- Inorder to perform this step, from the package sklearn.model_selection we imported train_test_split.
- Now we split the data into 70 -30 ratio, where the train data hold 70% of the data and test data holds 30% of the data. The random state mentioned here is 1.

m. Checking the dimensions of the training and test data.

Below output is retrieved from Jupter using shape command

```
X_train_1 (610, 6)
X_test_1 (262, 6)
Y_train_1 (610,)
Y_test_1 (262,)
```

- Train dataset has 610 records i.e., 70% of the total dataset.
- Test dataset contains 262 records i.e., 30% of the total dataset.

Now we have our train and test data ready. We will start building our Logistic Regression and LDA (linear discriminant analysis).

Logistic Regression:-

- Logistic regression is Supervised Learning technique for solving the classification problems. It is used for predicting the categorical dependent variable using a given set of independent variables and it predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). The S-form curve is called the Sigmoid function or the logistic function. It maps any real value into probabilistic values which lie between 0 and 1. On the basis of the categories, Logistic Regression can be classified into three types: **Binomial, Multinomial & Ordinal**.

Logistic Regression Equation:

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

- ✓ We know the equation of the straight line can be written as:

$$y = b_1x_1 + b_2x_2 + \dots + b_nx_n + c$$

- ✓ In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y) :

$$\frac{y}{1-y}$$ **; 0 for y=0 & infinity for y=1**

- ✓ But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$\log\left[\frac{y}{1-y}\right] = b_1x_1 + b_2x_2 + \dots + b_nx_n + c$$

The above equation is the final equation for Logistic Regression.

1. **Building a Logistic Regression Model**

Import the necessary library of LogisticRegression from sklearn.linear_model and import GridSearchCV from the package sklearn.model_selection.

In this step we fit the train data and labels in the CART model, based on model performance, model will be tunned using Grid search.

2. **Hyperparameter Tuning**

```
param_grid = {
'solver':['newton-cg','liblinear','lbfgs'],
'max_iter':[10000,15000],
'penalty':['none','l1','l2'],
'verbose':[True],
'n_jobs':[2],
'tol':[0.001,0.00000001],
}
```

- ❖ **Penalized** logistic regression imposes a penalty to the logistic model for having too many variables. This results in shrinking the coefficients of the less contribute variables toward zero. This is also known as regularization. In our grid search, we take 'L2' and 'none' as our arguments and check which is preferred by grid search.

- ❖ **Solver** is a process that runs for the optimization of the weights in the model. The solver uses a Coordinate Descent (CD) algorithm that solves optimization problems by successively performing approximate minimization along coordinate directions or coordinate hyperplanes. Different solvers take a different approach to get the best fit model. In our case, we have taken 'sag', 'lbfgs', 'liblinear' and 'newton-cg' as our arguments. We will check which is preferred by grid search.
- ❖ **Tol** is the tolerance of optimization. When the training loss is not improved by at least the given tol on consecutive iterations, convergence is considered to be reached and the training stops. We will be checking for tolerance of 0.0001 and 0.00001.

- ❖ The logistic regression uses an iterative maximum likelihood algorithm to fit the data. There are no set criteria for **maximum iterations**. The solver will run the model till it reaches convergence or till the max iterations, you have provided. In this case, we have given 10000 and 15000 as inputs. We will see which fits better.
- ❖ Here we take cross-validation as 3 and scoring as F1 for our grid search.

3. Best Estimator

Below output is obtained from Jupyter that results the best estimator for building our decision tree and that is obtained using a grid search cv function.

LogisticRegression(max_iter=10000, n_jobs=2, penalty='l1', random_state=1, solver='liblinear', tol=1e-08, verbose=True)

4. Checking the Coefficients

```
The coefficient for Salary is -1.6088717566770156e-05
The coefficient for age is -0.04977888040047271
The coefficient for educ is 0.06809199197180006
The coefficient for no_young_children is -1.226457876057361
The coefficient for no_older_children is -0.010272545996478099
The coefficient for foreign is 1.248742465494623
```

LDA Model (linear discriminant analysis)

Linear Discriminant 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.

LDA works when the measurements made on independent variables for each observation are continuous quantities. When dealing with categorical independent variables, the equivalent technique is discriminant correspondence analysis.

When we fit our training data into Linear Discriminant model. By default, LDA takes a custom cut-off probability as 0.5. At first we'll create our LDA model with cut-off probability as 0.5 and validate the performance, then we'll check the performs with multiple cut-off probabilities and decide which one performs best.

1. Building a LDA Model (linear discriminant analysis)

Import the necessary library of LinearDiscriminantAnalysis from sklearn.discriminant_analysis.

2. Fitting the train dataset to LDA Model

Once we fit our training data into Linear Discriminant model. By default, LDA takes a custom cut-off probability as 0.5. At first we'll create our LDA model with cut-off probability as 0.5 and validate the performance, then we'll check the performs with multiple cut-off probabilities and decide which one performs best.

Below output is retrieved from Jupyter on training data with cut-off probability ranges from 0.1- 0.9

Cut of probability	Recall	F1 Score
0.1	0.9964	0.9964
0.2	0.9644	0.6499
0.3	0.8932	0.6693
0.4	0.7580	0.6762
0.5	0.5765	0.6125
0.6	0.4235	0.5336
0.7	0.2989	0.4398
0.8	0.4398	0.1981
0.9	0.0071	0.0141

Table 4 : Probability cut-off range table (0.1 - 0.9)

From the above table we infer that, cut -off value 0.4 gives reasonable best result compared to probability cut-off of 0.5, but we need to first test our model with several cut-off probabilities to choose the one of the best cut off.

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.

Model performance:

This helps us to understand how good our model got trained. Model performance can be done only after the prediction of training and testing dataset . Here we validate if the model is underfitting or overfitting by checking certain parameters. Following methods are used to evaluate the model performance:

- Confusion Matrix

A confusion matrix is a table that is used to define the performance of a classification algorithm. A confusion matrix visualizes and summarizes the performance of a classification algorithm.

		Predicted	
Actual		Negative	Positive
	Negative	TN	FP
	Positive	FN	TP

-TN,TP - Correct Prediction (True Negative, True Positive)
- FP,FN - Incorrect prediction (False Positive, False Negative)

- Classification Report

1. Accuracy : Accuracy are used to identify how accurately/Cleanly , the model classifies the data point. Lesser the false predictions, more the accuracy.
Accuracy = (TP + TN) / (TP + TN + FP + FN).

2. Precision: Among the points identified as positive by the model, but how many points are actual positive.
If type(I) error is low pression will he high. Type(I) error and precision are inversely proportional to each other.
Precision = TP/(TP + FP).

3. Recall (Sensitivity): How many of the actual true data points are identified as True data points by the model. False Negative are those points should have been identified as True. Higher the sensitivity lowers the false negative(Type(II) error). Type(II) error and sensitivity are inversely proportional to each other
Recall = TP/(TP + FN)

4. Score: F1 Score computes an hormonic mean between Precision and Recall. It tells us both Type(I) and Type(II) error in a particular model is higher or lower on an average. If the F1 is good, that indicated model contains less false positives and less false negatives. F1 score is considered to be perfect when it tends to be 1 and model is a total failure when it tends to be 0.
F1 score = 2 x [(Precision x Recall) / (Precision + Recall)

- ROC Curve: ROC curve is a graphic representation of classifier performance. This curve plots two parameters: True Positive Rate. False Positive Rate. Higher the cure stronger the model, flatter the ROC curve weakest the model.

- AUC Score: AUC score gives the value of area under the ROC curve . The higher the AUC score, the better the performance of the model at distinguishing between the positive and negative.

Logistic Regression Model – Performance Metrics

1. **Prediction of training and testing dataset using predict command in train and test data.**

2. **Getting the Predicted Classes and Probs.**

Below is the output retrieved in Jupyter using proba command on train data:-

	0	1
0	0.677210	0.322790
1	0.565096	0.434904
2	0.688802	0.311198
3	0.516298	0.483702
4	0.545854	0.454146

3. Confusion Matrix

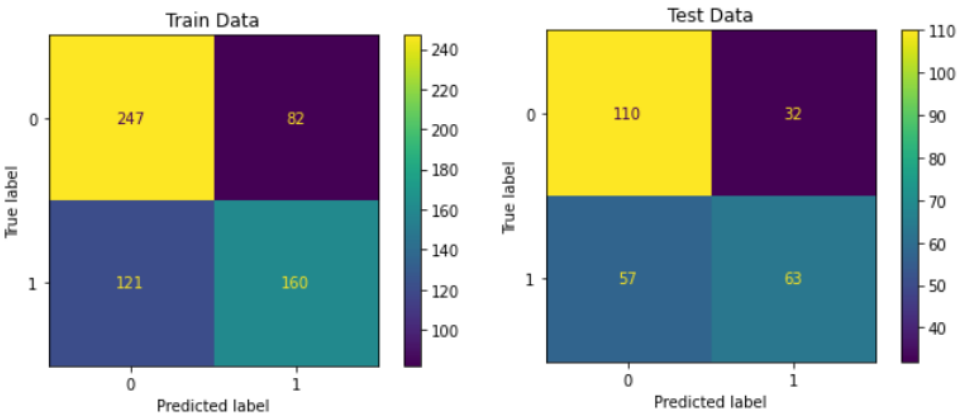


Fig 12: Logistic Regression Confusion Matrix

4. Classification Report

Train Report

	precision	recall	f1-score	support
0	0.67	0.75	0.71	329
1	0.66	0.57	0.61	281
accuracy			0.67	610
macro avg	0.67	0.66	0.66	610
weighted avg	0.67	0.67	0.66	610

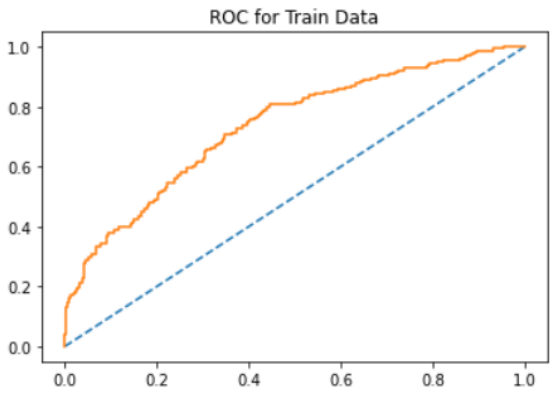
Test Report

	precision	recall	f1-score	support
0	0.66	0.77	0.71	142
1	0.66	0.53	0.59	120
accuracy			0.66	262
macro avg	0.66	0.65	0.65	262
weighted avg	0.66	0.66	0.65	262

Fig 13: Logistic Regression Classification report

5. ROC Curve and AUC score

AUC: 0.735



AUC: 0.718

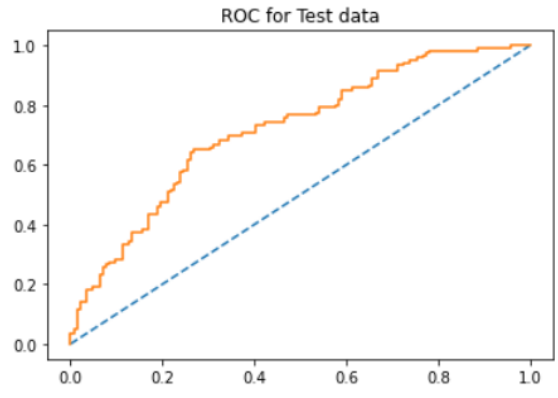


Fig 14: Logistic Regression of ROC curve

6. **Train and Test report comparison table**

Logistic Regression Model			
Sl.No	Index	Train Data	Test Data
1	TN	247	110
2	TP	160	63
3	FN	121	57
4	FP	82	32
5	Accuracy	0.67	0.66
6	Precision	0.66	0.66
7	Recall	0.57	0.53
8	F1 Score	0.61	0.59
9	AUC Score	0.74	0.72

Table 5 :Logistic Regression model comparison table (Train & Test)

Observation:

- Cases employees opted holiday package, there are 121 instances where model predicted as not opted.
- Cases where the employees not opted holiday package, but model predicted them to be opted are 82.
- In both train and test data an accuracy, precision and recall not much difference found and they are nearly identical to each other. This shows that, model is neither overfitting nor underfitting.

Linear discriminant analysis – Model performance based on a default cut-off probability (i.e., 0.5).

1. **Prediction of training and testing dataset using predict command in train and test data.**
2. **Getting the Predicted Classes and Probs.**

Below is the output retrieved in Jupyter using proba command on train data:-

	0	1
0	0.261849	0.738151
1	0.710383	0.289617
2	0.617657	0.382343
3	0.235165	0.764835
4	0.533171	0.466829

Below is the output retrieved in Jupyter using proba command on test data:-

	0	1
0	0.708475	0.291525
1	0.533448	0.466552
2	0.717871	0.282129
3	0.504865	0.495135
4	0.555863	0.444137

3. Confusion Matrix

Below is the output retrieved from Jupyter using the confusion matrix command on train data and test data:-

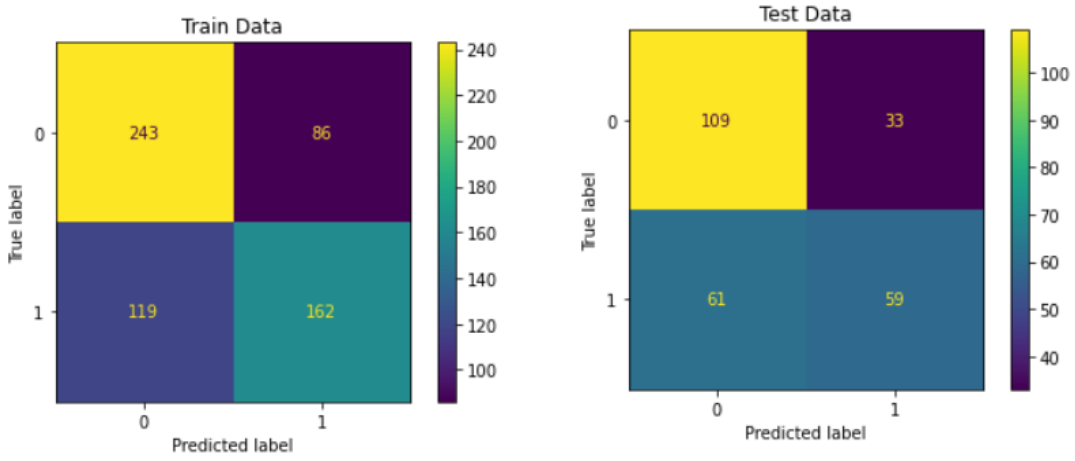


Fig 15 : LDA Confusion Matrix(cut-off 0.5)

4. Classification Report

Training Data report :

	precision	recall	f1-score	support
0	0.67	0.74	0.70	329
1	0.65	0.58	0.61	281
accuracy			0.66	610
macro avg	0.66	0.66	0.66	610
weighted avg	0.66	0.66	0.66	610

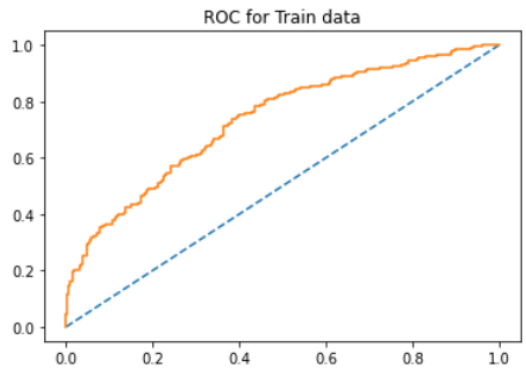
Testing Data report:

	precision	recall	f1-score	support
0	0.64	0.77	0.70	142
1	0.64	0.49	0.56	120
accuracy			0.64	262
macro avg	0.64	0.63	0.63	262
weighted avg	0.64	0.64	0.63	262

Fig 16: LDA Classification report(cut-off 0.5)

5. ROC Curve and AUC score

AUC: 0.733



AUC: 0.714

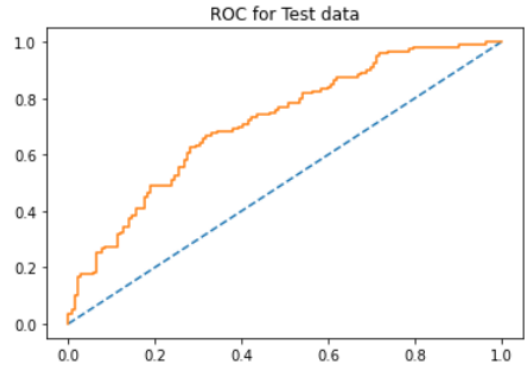


Fig 17: LDA ROC curve(cut-off 0.5)

6. Train and Test report comparison table

LDA Model – Cut-off (0.5)			
Sl.No	Index	Train Data	Test Data
1	TN	243	109
2	TP	162	61
3	FN	119	61
4	FP	86	33
5	Accuracy	0.67	0.64
6	Precision	0.65	0.64
7	Recall	0.58	0.49
8	F1 Score	0.61	0.56
9	AUC Score	0.73	0.71

LDA model comparison table – Train & Test (cut off 0.5)

Observation:

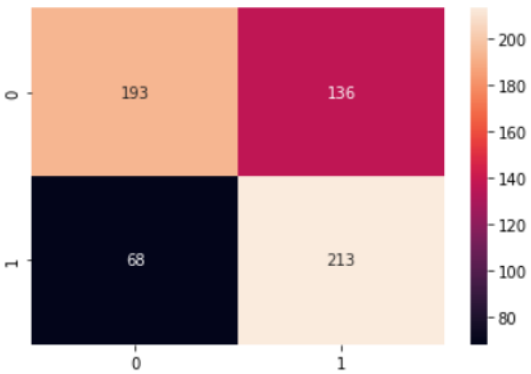
- Cases employees opted holiday package, there are 119 instances where model predicted as not opted.
- Cases where the employees not opted holiday package, but model predicted them to be opted are 86.
- In both train and test data an accuracy, precision and recall not much difference found and they are nearly identical to each other. This shows that, model is neither overfitting nor underfitting.

Linear discriminant analysis – Model performance based on a cut-off probability (i.e., 0.4).

1. Confusion Matrix

Below is the output retrieved from Jupyter using the confusion matrix command on train data and test data:-

Train Data



Test Data

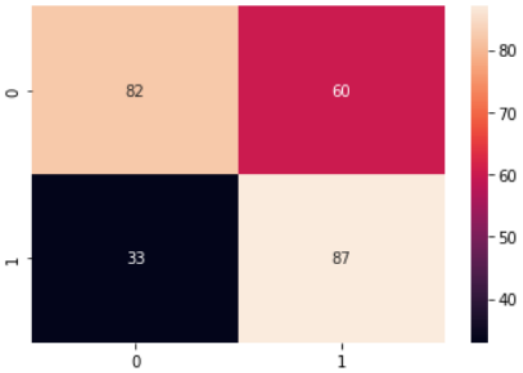


Fig 18 : LDA Confusion Matrix(cut-off 0.4)

2. Classification Report

Training Data report :

	precision	recall	f1-score	support
0	0.74	0.59	0.65	329
1	0.61	0.76	0.68	281
accuracy			0.67	610
macro avg	0.67	0.67	0.67	610
weighted avg	0.68	0.67	0.66	610

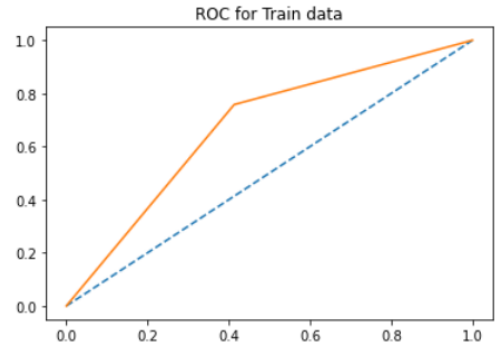
Testing Data report:

	precision	recall	f1-score	support
0	0.71	0.58	0.64	142
1	0.59	0.72	0.65	120
accuracy			0.65	262
macro avg	0.65	0.65	0.64	262
weighted avg	0.66	0.65	0.64	262

Fig 19: LDA Classification report(cut-off 0.4)

3. ROC Curve and AUC score

AUC: 0.672



AUC: 0.651

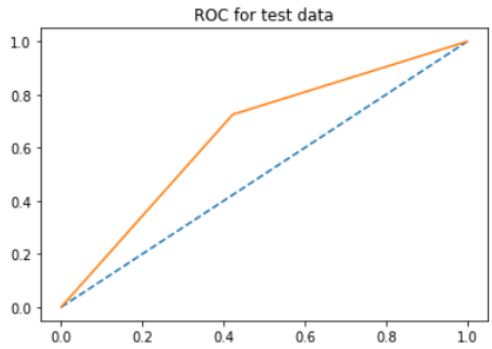


Fig 20: LDA ROC curve (cut-off 0.4)

4. Train and Test report comparison table

LDA Model - Cut off (0.4)			
Sl.No	Index	Train Data	Test Data
1	TN	193	82
2	TP	213	87
3	FN	68	33
4	FP	136	60
5	Accuracy	0.67	0.65
6	Precision	0.61	0.59
7	Recall	0.76	0.72
8	F1 Score	0.68	0.65
9	AUC Score	0.67	0.65

Table 7 : LDA model comparison table – Train & Test (cut off 0.4)

5. **Comparison of cut-off probability 0.5 and 0.4**

	LDA Train(cut_of_0.5)	LDA Test(cut_of_0.5)	LDA Train(cut_of_0.4)	LDA Test(cut_of_0.4)
Accuracy	0.66	0.64	0.66	0.64
AUC	0.73	0.71	0.67	0.65
Precision	0.65	0.64	0.61	0.59
Recall	0.58	0.49	0.76	0.72
F1 Score	0.61	0.56	0.68	0.65

Table 8: Comparison of cut-off probability (0.5 and 0.4)

Observation:-

- Problem statement, states that "We need to help the company in predicting whether an employee will opt for the package or not", so we need to have clear understanding on False positive and False Negative.
  - ✓ False positives are the employee who actually did not opt for holiday package, but the algorithm predicted that opted for holiday package.
  - ✓ False Negatives are the employee who actually opt for holiday package, but the model predicted that won't opt for holiday package.
- As a result, we can conclude that false positives will not have a significant impact on our firm, however false negative will impact the prediction. As per the model sensitivity/ recall will be more crucial.

From the above cut-off probability comparison table, we infer that, for probability cut off - 0.4, the recall score in test data increased from 0.49 to 0.72, here we decided to choose the probability cut off - 0.4.

Compare Both the models and write inference which model is best/optimized.

1. **Comparison of Logistic regression & Linear discriminative analysis models to infer which model is best/optimized.**

	LR Train	LR Test	LDA Train	LDA Test
Accuracy	0.67	0.66	0.66	0.64
AUC	0.73	0.72	0.67	0.65
Precision	0.66	0.66	0.61	0.59
Recall	0.57	0.52	0.76	0.72
F1 Score	0.61	0.59	0.68	0.65

Table 9: Comparison of LR & LDA model performance

2. **Comparison of ROC Curve and AUC scores of two models for Training data:**

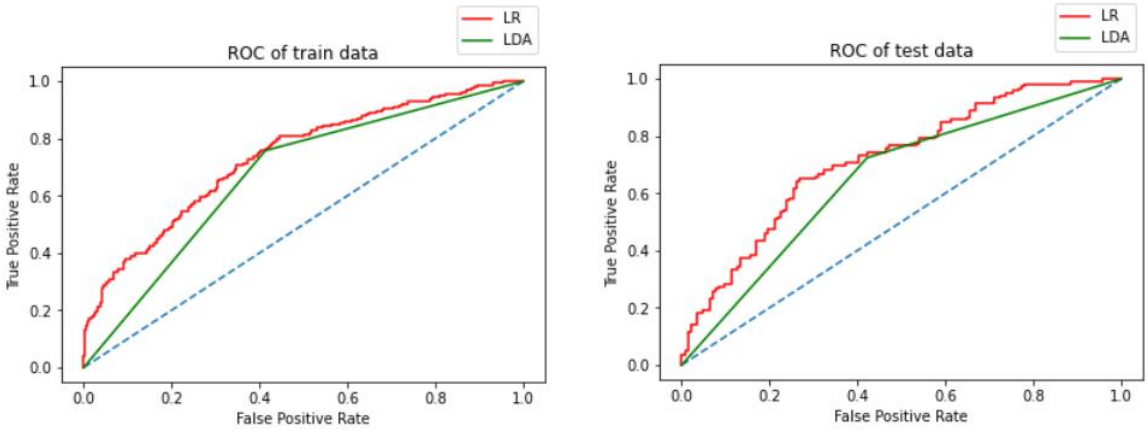


Fig 21: Comparison of LR & LDA ROC curve

Inferences on Comparison of Model performance:

- Here we built 2 models namely logistic regression and linear discriminative analysis. As we discussed earlier, as per problem statement false positives will not have a significant impact on our firm, however false negative will impact the prediction. As per the model sensitivity/ recall will be more crucial.
- Both the models performed reasonably stable enough to be used for making any future predictions. The train and test values aren't that so far for all the three modules, thus there seems to be no concern in overfitting or underfitting.
- Comparing the recall value among two models, LDA gives high recall test value as 0.76 than Logistic regression value as 0.52.
- By Comparing the Precision Score among two models, LR gives sightly high precision Score value as 0.66 than LDA model as 0.59.
- Comparing the Accuracy value among two models, Logistic regression gives sightly high accuracy score as 0.66 than LDA model as 0.64.
- By Comparing the F1 Score among two models, LDA gives sightly high F1 Score value as 0.65 than Logistic regression model as 0.59.
- By Comparing the AUC score among two models, Logistic regression gives sightly high F1 Score value as 0.72 than LDA model as 0.65.
- By visualizing the ROC curves of the two models, Logistic regression has slightly high line than LDA models.
- From the above interpretation we conclude that LDA model gives reasonable recall value & F1 score, also there doesn’t seem to have drastic difference between two models in terms of Accuracy, AUC score, Precision, Recall and F1 score. From this we conclude LDA model seems to be optimised model as per problem statement.

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

Our business problem states, we need to predict whether an employee would opt for a holiday package or not. Also, we predicted the results using both logistic regression and linear discriminant analysis.

In our extensive analysis so far, we have thoroughly examined given data and developed a model that predicts the classification of whether the employee opts for holiday package or no, based on the attributes in our dataset. Let us now look at the key points in our past data first and try to suggest some recommendations for the firm.

Insights retrieved from EDA:

Holiday package:

- We can observe that 54% of the employees are not opting for the holiday package and 46% are interested in the package. This implies we have a dataset which is fairly balanced.

Salary

- The average 'Salary' of employees opting for holiday package and not opting for holiday package is similar in nature.
- The coefficient for Salary is -1.6088717566770156e-05. There is almost no relation with the Holiday package, so we can say that Salary is not a good predictor for model building.
- Employees who earn high salary are more prone to not opt for holiday package.

Foreign :

- Foreign is a good predictor of dependent variable with a high positive coefficient 1.248742465494623.
- The frequency distribution of foreign implies that the employees are mostly from the same country which is around 75% and they are considered as non-foreign employees and foreigner employees are around 25% of them.
- We can see that the percentage of foreigners accepting the holiday package is substantially higher compared to native employees.
- The mean salary of foreign people is slightly less than natives.

Age :

- We can see that, the age distribution for employees who are opting for holiday package and not opting are similar in nature, though the number of people opting are less in number and mostly fall in range of 35-45 age group.
- We can see that, employees in middle range (34 to 45 years) are opting for holiday package are more as compared to older and younger employees.

Education :

- The variable 'educ' the number of years of formal education is showing a similar pattern. This means education is likely not a variable that influences for opting of holiday packages for employees. We can see that employee with less years of formal education (1 to 7 years) and higher education are not opting for the Holiday package as compared to employees with formal education of 8 year to 12 years
- Across education we can observe that the employees with higher number of years of formal education have a lower tendency to opt for the holiday package relative to employees with lesser years of formal education

No. of young children :

- We can see that people with have younger children and opting for holiday packages are very few in number compared to employees who do not have young children.

No. of older children :

- The distribution for opting or not opting for holiday packages looks same for employees with older children and this might not be a good predictor for model building, it slightly fails to differentiate between the 2 different classes of dependent variable.

Recommendations:

- The firm should concentrate its efforts on foreigners in order to increase sales of vacation packages, as this is where the majority of conversions will occur.
- The firm might try to target their marketing efforts or offers towards foreigners in order to increase the number of people who choose vacation packages.
- Focus on Foreign variable for good prediction while building the classification model.
- To improve the likelihood of lower-wage employees the firm can provide certain incentives, discounts & EMI option to them while selecting for a vacation package.
- For the employees who earn high salary the firm can provide premium to luxury holiday package.
- The company should target on the employees who doesn't have younger & older children and refuse to take holiday package might belong to low wage people, so firm can target on those people by providing discounts and some complement benefits.
- Employees with older children and not opt for vacation package might be targeted by providing holiday packages during vacation months along with some fun filled activities.
- The firm can give some additional benefit to opted employee when referring the non-opted workers, so that non-opted employee might opt for the package to avail the reference benefit. The firm can also make a tie up with company by providing business tour package to the employees.

Key performance indicators:

- Highlight the benefits of Holiday package and services and educate the employees about it.
- Company can come up with lucrative enchantments in holiday packages.
- Customer satisfaction should be topmost priority.
- Engage with employees through social media.
- New destinations can be added.
- Video projection is a great way to engage and inspire potential travellers.
- Travel influencers can promote destinations, activities, and businesses by using their social media influence.
- Get feedback from employees who took the holiday package and work on the betterment of package accordingly.