greatlearning
Learning for Life

BUSINESS REPORT

PREPARED BY - RUPESH KUMAR

SUBMISSION DATE: 03/10/2021

Table of Contents

Problem Statement 1: Linear Regression

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.

EXECUTIVE SUMMARY

A company Gem Stones co ltd. which is a cubic zirconia manufacturer. The dataset consists of 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). Based on the different attributes / characteristics of the cubic zirconia we are helping 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.

Introduction

The purpose of this whole exercise is to explore the dataset. Do the exploratory data analysis & apply various supervised learning algorithms i.e. Linear Regression to predict the price for the cubic zirconia. Explore the dataset using central tendency and other parameters. The data consists of 26967 different stones with 10 unique activities . Analyse the different attributes of the stone which can help in predicting the price for the stone. This assignment should help the gemstone company to distinguish between higher profitable stones and lower profitable stones so as to have better profit share.In exploring the summary statistics, linear regression model will help company to predict the price for the stone.

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.

EDA - Data Description , Data Preprocessing , Data Visualization -

Checking the Records of the Dataset.

Head of the Dataset - First 10 Records of the Dataset.
Tail of the Dataset - Last 10 Records of the Dataset.

	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
5	6	1.02	Ideal	D	VS2	61.5	56.0	6.46	6.49	3.99	9502
6	7	1.01	Good	H	SI1	63.7	60.0	6.35	6.30	4.03	4836
7	8	0.50	Premium	E	SI1	61.5	62.0	5.09	5.06	3.12	1415
8	9	1.21	Good	H	SI1	63.8	64.0	6.72	6.63	4.26	5407
9	10	0.35	Ideal	F	VS2	60.5	57.0	4.52	4.60	2.76	706

	Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
26957	26958	2.09	Premium	H	SI2	60.6	59.0	8.27	8.22	5.00	17805
26958	26959	1.37	Premium	E	SI2	61.0	57.0	7.25	7.19	4.40	6751
26959	26960	1.05	Very Good	E	SI2	63.2	59.0	6.43	6.36	4.04	4281
26960	26961	1.10	Very Good	D	SI2	NaN	63.0	6.76	6.69	3.94	4361
26961	26962	0.25	Premium	F	VVS2	62.0	59.0	4.04	3.99	2.49	740
26962	26963	1.11	Premium	G	SI1	62.3	58.0	6.61	6.52	4.09	5408
26963	26964	0.33	Ideal	H	IF	61.9	55.0	4.44	4.42	2.74	1114
26964	26965	0.51	Premium	E	VS2	61.7	58.0	5.12	5.15	3.17	1656
26965	26966	0.27	Very Good	F	VVS2	61.8	56.0	4.19	4.20	2.60	682
26966	26967	1.25	Premium	J	SI1	62.0	58.0	6.90	6.88	4.27	5166

TAB:1 RECORDS OF THE DATASET HEAD & TAIL

Dropping the Unnamed: 0 Column.

We are going to drop the column unnamed:0 as it is useless for the model & checking the dataset again.

	carat	cut	color	clarity	depth	table	x	y	z	price
0	0.30	Ideal	E	SI1	62.1	58.0	4.27	4.29	2.66	499
1	0.33	Premium	G	IF	60.8	58.0	4.42	4.46	2.70	984
2	0.90	Very Good	E	VVS2	62.2	60.0	6.04	6.12	3.78	6289
3	0.42	Ideal	F	VS1	61.6	56.0	4.82	4.80	2.96	1082
4	0.31	Ideal	F	VVS1	60.4	59.0	4.35	4.43	2.65	779
5	1.02	Ideal	D	VS2	61.5	56.0	6.46	6.49	3.99	9502
6	1.01	Good	H	SI1	63.7	60.0	6.35	6.30	4.03	4836
7	0.50	Premium	E	SI1	61.5	62.0	5.09	5.06	3.12	1415
8	1.21	Good	H	SI1	63.8	64.0	6.72	6.63	4.26	5407
9	0.35	Ideal	F	VS2	60.5	57.0	4.52	4.60	2.76	706

TAB:2 RECORDS OF THE DATASET WITH FINALISED COLUMNS

Insights:

Now we have all the columns which are useful for the model.

Data Dictionary for Problem Statement 1.

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	cubic zirconia Clarity refers to the absence of the Inclusions and Blemishes. (In order from Worst to Best) IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1
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.

TAB:3 DATA DICTIONARY OF THE DATASET

Checking the Summary of the Dataset.

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
carat	26967.0	NaN	NaN	NaN	0.798375	0.477745	0.2	0.4	0.7	1.05	4.5
cut	26967	5	Ideal	10816	NaN	NaN	NaN	NaN	NaN	NaN	NaN
color	26967	7	G	5661	NaN	NaN	NaN	NaN	NaN	NaN	NaN
clarity	26967	8	SI1	6571	NaN	NaN	NaN	NaN	NaN	NaN	NaN
depth	26270.0	NaN	NaN	NaN	61.745147	1.41286	50.8	61.0	61.8	62.5	73.6
table	26967.0	NaN	NaN	NaN	57.45608	2.232068	49.0	56.0	57.0	59.0	79.0
x	26967.0	NaN	NaN	NaN	5.729854	1.128516	0.0	4.71	5.69	6.55	10.23
y	26967.0	NaN	NaN	NaN	5.733569	1.166058	0.0	4.71	5.71	6.54	58.9
z	26967.0	NaN	NaN	NaN	3.538057	0.720624	0.0	2.9	3.52	4.04	31.8
price	26967.0	NaN	NaN	NaN	3939.518115	4024.864666	326.0	945.0	2375.0	5360.0	18818.0

TAB:4 SUMMARY OF THE DATASET

Insights:

From the above table we can infer the count ,mean, std , 25% , 50% ,75% and min & max values of the all numeric variables present in the dataset.
From the above table we can infer the count, unique, top, frequency of all the categorical variables present in the dataset.
There is bad values found in the x , y , z columns of the Dataset. As x , y , z are the length , width & height of the cubic zirconia in mm and we have found minimum value of x , y , z is zero which doesn't make sense. As we know that length , width , height can't be zero. Thus, we need to treat & clean them.

Checking the Shape of the DataFrame

Shape attribute tells us number of observations and variables we have in the data set. It is used to check the dimension of data. The cubic_zirconia.csv data set has 26967 observations (rows) and 10 variables (columns) in the dataset.

Number of Rows : 26967
Number of Columns :10

Checking the Appropriateness of Datatypes & Information of the DataFrame

```
<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
```

TAB:5 APPROPRIATENESS OF DATATYPES & INFORMATION OF THE DATAFRAME

Insights:

From the above results we can see that there is null values present in the depth column of the dataset.Their are total 26967 rows & 10 columns in this dataset, indexed from 0 to 26966. Out of 10 variables 6 are float64 , 3 variables are object and 1 variable is int64. Memory used by the dataset: 2.1+ MB.

Checking for Null Values.

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

TAB:6 NULL VALUES.

Insights:

From the output we infer that only depth have null values.
697 null values are present in the depth column.

Checking for Anomalies in the Dataset

Carat -

```
array([0.3 , 0.33, 0.9 , 0.42, 0.31, 1.02, 1.01, 0.5 , 1.21, 0.35, 0.32,
       1.1 , 0.71, 1.5 , 0.34, 0.54, 1.04, 0.4 , 1.52, 1.19, 0.66, 0.52,
       0.72, 0.77, 0.51, 1.26, 1.55, 1.58, 0.43, 2.  , 0.73, 1.14, 0.78,
       0.91, 0.27, 1.8 , 1.13, 0.38, 0.57, 0.44, 0.7 , 1.22, 0.59, 1.2 ,
       2.16, 1.68, 0.76, 0.74, 0.41, 1.51, 1.69, 1.3 , 1.  , 0.28, 0.55,
       1.39, 0.36, 0.23, 1.37, 0.81, 2.02, 2.8 , 1.56, 2.2 , 0.75, 1.71,
       1.11, 0.92, 1.45, 1.16, 0.58, 0.97, 1.03, 0.26, 1.53, 1.63, 0.96,
       1.24, 0.39, 0.61, 0.24, 2.01, 1.7 , 0.79, 0.67, 1.28, 0.25, 0.56,
       1.09, 2.11, 1.17, 0.82, 0.53, 0.46, 2.43, 1.65, 0.84, 1.74, 0.8 ,
       1.83, 1.25, 1.15, 0.6 , 1.06, 0.29, 1.05, 1.18, 2.27, 2.36, 1.07,
       0.95, 0.93, 2.48, 1.23, 2.03, 1.27, 0.83, 1.43, 0.45, 1.12, 1.59,
       0.62, 1.61, 2.04, 1.33, 0.37, 1.35, 1.6 , 3.04, 1.57, 2.14, 0.94,
       0.49, 1.49, 1.76, 2.1 , 1.78, 1.34, 1.38, 2.33, 2.51, 2.05, 0.87,
       1.79, 1.73, 0.69, 1.32, 1.86, 2.61, 1.72, 1.66, 3.01, 0.63, 2.06,
       2.29, 1.47, 1.08, 0.86, 0.68, 1.31, 1.41, 0.65, 1.54, 1.91, 2.22,
       1.29, 2.49, 0.98, 1.44, 0.64, 1.87, 0.47, 2.56, 2.28, 1.9 , 1.67,
       2.45, 1.82, 1.64, 0.48, 2.39, 2.21, 2.24, 2.25, 1.99, 1.75, 2.44,
       2.09, 1.36, 0.88, 2.07, 1.62, 0.85, 2.35, 1.48, 2.13, 0.89, 4.01,
       2.23, 2.77, 2.32, 2.3 , 2.65, 1.42, 0.21, 1.85, 2.17, 2.52, 1.4 ,
       2.34, 2.5 , 2.74, 2.08, 1.95, 2.12, 1.77, 2.54, 0.99, 2.31, 2.42,
       2.53, 2.55, 0.2 , 2.37, 2.19, 1.84, 1.46, 2.18, 2.26, 4.  , 3.5 ,
       1.98, 2.15, 2.38, 3.  , 1.97, 2.66, 1.81, 1.96, 4.5 , 1.88, 3.4 ,
       2.4 , 2.64, 3.51, 1.94, 2.57, 2.7 , 1.89, 2.63, 0.22, 1.93, 2.58,
       2.6 , 2.59, 2.41, 3.24])
```

Cut -

```
array(['Ideal', 'Premium', 'Very Good', 'Good', 'Fair'], d-type=object)
```

Clarity -

```
array(['SI1', 'IF', 'VVS2', 'VS1', 'VVS1', 'VS2', 'SI2', 'I1'],
      dtype=object)
```

Depth -

```
array([62.1, 60.8, 62.2, 61.6, 60.4, 61.5, 63.7, 63.8, 60.5, 60.7, 61.1,
       66.2, 61.2, 59.8, 61.9, 60. , 62.9, 62.7, 61.7, 62.4, 61.4,  nan,
       64. , 62.3, 63. , 59.9, 62.8, 61.3, 62. , 61. , 63.9, 62.6, 62.5,
       61.8, 58. , 64.9, 60.9, 59.7, 63.2, 58.4, 59.4, 63.5, 63.1, 66.8,
       65.2, 60.6, 64.3, 60.2, 60.3, 65.5, 58.5, 68.3, 66.5, 63.3, 58.8,
       63.6, 63.4, 57.5, 59. , 58.7, 59.1, 64.1, 64.5, 64.4, 60.1, 57.6,
       70.6, 59.2, 59.3, 50.8, 58.9, 65.4, 58.6, 59.5, 56.7, 67. , 66. ,
       54.6, 59.6, 64.7, 66.9, 64.6, 64.8, 58.2, 57.9, 56.9, 66.4, 65. ,
       66.6, 57.4, 64.2, 58.1, 67.7, 55.2, 66.3, 65.3, 67.9, 67.6, 65.8,
       67.1, 65.1, 67.5, 56.6, 55.9, 57.3, 57.1, 57.8, 58.3, 65.7, 57.2,
       52.7, 56.1, 66.1, 56.3, 66.7, 54.7, 71.3, 67.3, 65.9, 71. , 57.7,
       53.4, 65.6, 56. , 68.9, 68.8, 55.3, 69.2, 53.1, 69.8, 56.5, 56.2,
       55.1, 55.5, 53.2, 56.8, 68.4, 67.8, 55.6, 67.2, 57. , 69. , 55.8,
       52.2, 53.8, 68.6, 68. , 68.7, 68.5, 70.2, 56.4, 68.1, 73.6, 55.4,
       68.2, 69.5, 55. , 69.3, 70. , 67.4, 54.2, 69.1, 69.7, 69.9, 71.6,
       70.5, 69.6, 72.9, 72.2, 70.8])
```

Table -

```
array([58. , 60. , 56. , 59. , 62. , 64. , 57. , 55. , 58.2, 53. , 61. ,
       54. , 55.7, 63. , 60.3, 65. , 54.7, 54.2, 66. , 62.2, 57.8, 60.4,
       53.8, 55.4, 54.3, 51. , 67. , 55.8, 55.1, 54.8, 56.7, 56.9, 53.4,
       60.1, 68. , 55.5, 55.2, 56.5, 56.1, 58.9, 62.3, 70. , 57.2, 57.6,
       60.2, 53.6, 53.1, 58.5, 55.9, 56.2, 52. , 59.4, 54.5, 55.6, 58.4,
       56.4, 57.4, 54.4, 53.7, 54.1, 54.9, 58.7, 53.2, 79. , 59.3, 53.3,
       53.9, 60.7, 76. , 59.9, 59.6, 58.3, 55.3, 61.6, 60.5, 57.7, 58.1,
       57.1, 60.8, 69. , 49. , 64.2, 50. , 61.5, 56.6, 58.6, 60.9, 56.8,
       57.5, 54.6, 62.8, 51.6, 56.3, 60.6, 57.9, 59.1, 62.5, 53.5, 59.8,
       59.7, 50.1, 61.8, 63.3, 58.8, 59.2, 57.3, 61.4, 62.6, 61.9, 64.3,
       62.1, 61.2])
```

X -

```
array([ 4.27,  4.42,  6.04,  4.82,  4.35,  6.46,  6.35,  5.09,  6.72,
        4.52,  4.4 ,  6.74,  5.08,  5.74,  7.12,  4.37,  4.56,  6.52,
        6.  ,  5.42,  6.54,  4.72,  7.35,  6.8 ,  5.53,  7.4 ,  4.5 ,
        5.18,  4.38,  5.65,  6.76,  5.7 ,  4.3 ,  5.95,  4.61,  6.82,
        5.16,  6.9 ,  7.42,  6.05,  4.88,  4.32,  8.01,  6.42,  4.85,
        5.21,  5.75,  6.71,  5.83,  6.14,  5.12,  6.88,  4.89,  6.44,
        5.13,  4.22,  7.81,  4.28,  6.79,  4.64,  5.36,  4.25,  5.23,
        4.9 ,  5.91,  5.39,  5.37,  5.11,  8.29,  4.39,  7.6 ,  6.08,
        4.81,  5.85,  5.92,  6.86,  4.75,  4.34,  5.06,  5.24,  4.77,
        7.3 ,  5.07,  4.47,  6.24,  5.71,  5.15,  5.02,  6.25,  5.57,
        7.59,  7.01,  6.93,  6.56,  4.2 ,  6.31,  5.31,  7.18,  4.59,
        4.03,  7.11,  5.97,  6.4 ,  8.05,  7.72,  6.01,  9.03,  6.19,
        5.76,  5.35,  4.8 ,  7.44,  4.26,  8.54,  6.15,  5.8 ,  4.78,
        4.76,  4.74,  7.65,  6.5 ,  6.06,  6.66,  6.22,  7.14,  6.78,
        6.45,  5.69,  6.27,  5.32,  5.14,  6.39,  4.98,  4.15,  6.13,
        4.36,  7.55,  5.98,  5.73,  4.48,  5.72,  4.87,  4.41,  7.64,
        6.34,  7.02,  5.1 ,  4.86,  6.21,  5.45,  5.27,  4.  ,  5.4 ,
        6.38,  6.07,  5.79,  5.64,  8.19,  6.87,  4.43,  5.33,  6.84,
        7.39,  4.21,  4.02,  6.37,  7.74,  5.29,  4.53,  4.91,  5.94,
        5.62,  6.83,  6.12,  6.18,  7.13,  4.16,  6.47,  5.01,  4.45,
        5.3 ,  4.07,  4.73,  7.43,  5.38,  4.69,  5.88,  6.6 ,  5.68,
        4.68,  6.65,  5.96,  4.67,  5.2 ,  4.04,  5.6 ,  5.81,  4.33,
        6.49,  5.82,  4.83,  4.31,  5.25,  7.62,  4.49,  4.84,  5.63,
        5.77,  6.11,  4.23,  5.59,  7.56,  6.63,  7.9 ,  6.2 ,  5.67,

        7.32,  7.16,  3.9 ,  7.  ,  7.96,  7.33,  7.63,  6.62,  6.3 ,
        7.24,  6.59,  7.34,  6.94,  5.78,  7.29,  6.03,  7.2 ,  4.44,
        6.51,  6.55,  4.95,  4.51,  4.1 ,  7.97,  8.41,  6.36,  8.51,
        6.43,  4.29,  4.58,  6.91,  6.1 ,  8.13,  6.68,  8.15,  5.05,
        6.48,  7.79,  5.48,  6.41,  8.64,  6.64,  7.54,  6.73,  7.37,
        4.18,  4.6 ,  7.51,  6.23,  6.92,  7.73,  5.26,  6.75,  6.29,
        7.28,  5.17,  4.57,  7.36,  7.52,  5.  ,  4.12,  7.21,  5.04,
        7.48,  4.55,  5.34,  5.19,  5.49,  8.36,  7.49,  6.57,  8.14,
        7.15,  6.85,  4.54,  7.03,  5.66,  8.17,  7.27,  6.53,  4.65,
        5.46,  8.09,  4.92,  6.96,  7.31,  7.05,  4.94,  9.14,  7.57,
        8.07,  8.28,  4.06,  4.66,  6.09,  6.17,  5.03,  7.61,  6.95,
        5.87,  7.7 ,  3.89,  7.58,  7.47,  4.99,  7.67,  8.32,  3.93,
        7.06,  6.98,  8.37,  6.89,  4.46,  7.46,  7.09,  4.7 ,  6.28,
        8.21,  4.11,  3.96,  5.61,  6.58,  6.32,  8.11,  7.38,  4.14,
        4.79,  4.63,  3.98,  7.53,  7.66,  7.19,  5.5 ,  3.94,  5.99,
        7.77,  7.1 ,  8.85,  4.24,  7.69,  5.52,  5.56,  8.27,  6.16,
        4.13,  8.  ,  5.84,  3.91,  8.99,  8.39,  4.71,  8.45,  7.25,
        6.81,  3.99,  6.7 ,  8.18,  7.26,  7.89,  6.77,  4.96,  8.33,
        7.17,  5.9 ,  8.02,  7.04,  7.86,  7.87,  8.42,  6.67,  8.88,
        4.09,  6.02,  5.22,  8.26,  5.28,  7.07,  4.62,  7.41,  3.95,
        8.08,  5.41,  8.62,  7.23,  8.43,  6.99,  3.87,  5.89,  5.51,
        7.93,  4.19,  7.78,  4.93,  8.25,  5.86,  6.69,  6.97,  7.45,
        8.67,  7.83,  7.76,  6.33,  5.93,  7.8 ,  7.22,  4.97,  8.3 ,
        7.5 ,  4.01,  7.84,  7.71,  6.61,  8.71,  7.68,  8.35,  7.88,
        8.38,  8.55,  5.54,  7.91,  7.95,  5.58,  8.06,  8.04,  8.58,
        7.99,  5.47,  8.23,  5.44,  8.34,  8.12,  8.2 ,  4.08,  6.26,
        7.94,  9.24,  3.92,  7.85, 10.02,  3.97,  8.93,  8.5 ,  3.88,
        8.47,  8.03,  8.82,  8.1 ,  7.98,  5.43,  8.4 ,  5.55,  8.49,
        7.75,  8.31,  8.6 ,  4.05,  8.24,  8.46,  8.87,  4.17,  8.16,
        8.44,  8.61,  8.76,  8.63,  8.9 ,  8.7 ,  7.08,  7.82,  3.74,
        0.  ,  8.52,  8.22,  8.78,  9.25,  8.56, 10.01,  8.75,  9.65,
        8.66,  8.72,  8.59,  3.81,  8.65,  9.3 ,  7.92,  8.48,  3.79,
        3.82,  8.68,  8.69, 10.23,  9.42,  3.73,  9.1 ,  3.84,  3.85,
        8.53,  9.66,  9.36,  8.77,  8.84,  9.38,  9.06,  8.8 ,  8.81,
        3.86,  3.77,  9.51, 10.14,  8.74,  8.83,  8.57,  3.83,  9.44])
```

Y -

```
array([ 4.29,  4.46,  6.12,  4.8 ,  4.43,  6.49,  6.3 ,  5.06,  6.63,
        4.6 ,  6.71,  5.12,  5.76,  7.08,  4.39,  4.53,  6.09,  5.22,
        6.51,  4.69,  7.28,  6.85,  5.56,  7.25,  4.44,  5.2 ,  4.35,
        5.59,  6.81,  5.72,  4.32,  5.9 ,  4.64,  6.43,  5.11,  6.93,
        7.47,  6.02,  7.38,  4.84,  4.34,  7.91,  4.78,  5.23,  5.81,
        5.19,  5.18,  6.68,  5.8 ,  4.85,  6.22,  6.47,  5.15,  6.8 ,
        4.94,  6.37,  5.1 ,  4.17,  7.89,  4.3 ,  6.7 ,  5.3 ,  5.17,
        4.41,  5.44,  4.93,  5.16,  5.83,  5.42,  6.83,  5.4 ,  8.22,
        4.42,  4.77,  7.54,  6.17,  4.87,  5.88,  5.97,  4.72,  4.37,
        5.08,  4.82,  7.35,  4.5 ,  6.16,  5.75,  6.18,  5.54,  7.66,
        7.05,  4.4 ,  6.94,  6.52,  4.25,  6.34,  4.81,  5.41,  7.14,
        4.62,  4.07,  7.04,  5.95,  6.36,  8.11,  4.51,  8.98,  6.13,
        5.33,  4.24,  4.83,  7.4 ,  4.33,  8.49,  5.79,  6.24,  5.82,
        5.14,  4.75,  4.74,  4.76,  4.52,  7.58,  6.44,  5.43,  6.61,
        6.11,  6.25,  7.17,  6.75,  6.4 ,  5.73,  6.32,  5.09,  6.28,
        6.48,  5.03,  5.46,  4.19,  6.21,  6.1 ,  4.38,  7.62,  6.01,
        5.77,  5.64,  4.9 ,  7.56,  6.38,  6.87,  4.66,  6.69,  4.79,
        4.88,  4.67,  6.31,  5.48,  5.25,  4.09,  4.03,  5.31,  5.61,
        8.15,  6.78,  6.84,  4.47,  5.38,  6.26,  7.37,  7.48,  6.82,
        6.41,  4.04,  7.84,  5.27,  4.56,  6.  ,  5.69,  7.49,  6.2 ,
        6.35,  7.03,  4.45,  5.32,  4.1 ,  4.7 ,  5.85,  4.63,  6.59,
        6.91,  5.13,  5.98,  4.06,  4.54,  4.92,  5.66,  8.1 ,  8.61,
        5.74,  4.28,  7.55,  4.48,  5.6 ,  5.87,  5.93,  5.67,  5.34,
        6.15,  5.51,  5.94,  7.78,  5.71,  7.3 ,  7.09,  3.93,  6.72,
        7.88,  7.68,  6.45,  5.68,  6.65,  6.46,  7.31,  6.56,  7.29,
        4.27,  7.22,  4.36,  4.18,  6.92,  6.54,  6.5 ,  4.91,  4.68,
        5.36,  6.77,  6.89,  4.13,  7.8 ,  6.55,  5.65,  4.65,  8.36,
        5.84,  8.56,  6.58,  4.31,  6.19,  5.7 ,  6.33,  5.04,  6.14,
        6.57,  8.09,  6.66,  8.2 ,  5.21,  6.27,  7.75,  5.45,  8.58,
        7.51,  6.97,  7.34,  4.23,  4.55,  7.98,  5.28,  7.46,  5.39,
        7.26,  6.29,  6.23,  5.57,  7.24,  6.6 ,  7.41,  5.05,  4.15,
        5.35,  6.08,  6.67,  5.78,  6.74,  7.43,  5.24,  5.52,  8.26,
        6.39,  7.27,  4.86,  6.76,  6.95,  4.61,  5.29,  8.24,  7.23,
        6.42,  7.32,  7.39,  8.  ,  3.99,  7.01,  7.9 ,  4.73,  7.  ,
        7.1 ,  4.98,  9.07,  7.5 ,  8.18,  4.11,  6.9 ,  7.67,  4.71,
        6.04,  7.74,  3.91,  7.53,  7.45,  6.07,  6.79,  8.12,  7.44,
        4.96,  4.26,  7.61,  4.89,  8.28,  3.97,  8.3 ,  6.88,  8.6 ,
        4.59,  8.27,  4.95,  4.  ,  5.58,  7.85,  8.07,  7.18,  3.92,
        7.99,  5.96,  7.82,  6.03,  7.94,  7.19,  5.92,  8.73,  4.97,
        5.63,  7.33,  7.63,  6.99,  5.55,  7.64,  8.21,  8.04,  3.94,
        5.89,  8.94,  5.49,  8.29,  3.95,  8.43,  4.16,  7.21,  5.02,
        5.26,  7.2 ,  8.35,  8.19,  7.15,  5.01,  6.64,  4.01,  7.13,
        8.13,  5.07,  4.21,  7.6 ,  5.  ,  7.65,  4.2 ,  5.62,  5.86,
        8.23,  6.86,  5.91,  4.49,  6.06,  7.81,  8.34,  8.64,  7.59,
        7.06,  7.12,  4.57,  8.57,  7.11,  8.46,  7.7 ,  5.5 ,  6.53,
        7.42,  7.07,  7.86,  6.62,  7.36,  5.99,  7.87,  4.05,  5.53,
        6.73,  4.58,  7.79,  7.77,  6.96,  7.57,  7.52,  5.37,  8.14,
        4.08,  8.77,  8.06,  8.33,  8.41,  8.51,  6.05,  7.96,  7.97,
        8.03,  8.62,  7.92,  7.72,  8.16,  7.02,  8.31,  8.02,  8.44,
        4.02,  8.25,  4.12,  6.98,  3.96,  3.98,  4.14,  5.47,  7.16,
        9.13,  4.22,  8.17,  9.94,  7.69,  8.01,  8.83,  8.54,  8.42,
        4.99,  8.75,  8.08,  8.37,  8.47,  8.65,  3.9 ,  8.9 ,  7.73,
        8.48,  8.05,  8.39,  8.55,  7.95,  8.66,  8.52,  8.82,  8.59,
        3.71,  8.53,  0.  ,  8.38,  7.93,  8.32,  7.76,  8.74,  7.83,
        9.2 ,  8.4 ,  7.71,  8.68,  8.69,  9.59,  8.7 ,  8.81,  3.78,
        9.14,  3.84,  3.75,  8.45,  8.76, 10.16,  8.67,  3.77,  9.34,
        8.97,  3.89,  8.5 ,  9.63,  9.31,  8.78,  9.01,  9.26,  3.8 ,
        3.88,  3.86,  8.79,  3.72,  9.46,  8.84, 10.1 ,  3.85,  8.71,
        8.88, 58.9 ,  3.87,  9.4 ])
```

Z -

```
array([ 2.66,  2.7 ,  3.78,  2.96,  2.65,  3.99,  4.03,  3.12,  4.26,
        2.76,  2.72,  4.08,  3.11,  3.54,  4.7 ,  2.78,  3.89,  3.74,
        3.19,  4.59,  4.21,  3.46,  4.49,  2.74,  3.6 ,  4.23,  2.69,
        2.77,  4.22,  4.01,  3.15,  4.29,  4.54,  3.85,  4.61,  2.94,
        2.67,  5.09,  3.01,  3.22,  3.59,  3.2 ,  4.11,  3.63,  4.02,
        3.27,  3.95,  2.58,  4.88,  2.85,  3.18,  2.73,  3.3 ,  2.99,
        3.21,  3.43,  4.25,  3.34,  4.2 ,  5.14,  2.92,  3.64,  3.52,
        4.32,  3.16,  3.24,  4.52,  3.1 ,  3.82,  3.62,  4.15,  4.74,
        2.68,  3.96,  2.62,  4.07,  2.97,  4.44,  2.81,  2.43,  4.36,
        3.69,  4. ,  5.04,  4.67,  5.5 ,  3.77,  2.91,  2.64,  5.1 ,
        3.57,  3.81,  3.61,  3.09,  2.79,  4.8 ,  3.9 ,  3.84,  4.09,
        3.8 ,  4.16,  3.48,  3.33,  2.52,  3.73,  2.71,  3.71,  3.47,
        3.55,  3.25,  4.66,  3.66,  2.95,  3.37,  3.56,  3.86,  2.49,
        2.48,  4.05,  3.87,  4.85,  3.17,  4.58,  4.5 ,  4.24,  2.47,
        4.71,  3. ,  3.65,  3.38,  4.45,  2.89,  3.97,  3.42,  3.72,
        2.9 ,  2.4 ,  3.98,  2.93,  2.75,  3.29,  2.45,  2.83,  2.98,
        3.31,  3.44,  4.1 ,  5.08,  4.19,  3.23,  2.42,  2.8 ,  3.53,
        5.34,  3.41,  2.82,  4.73,  4.78,  3.79,  3.45,  2.86,  3.83,
        4.87,  4.14,  4.93,  4.48,  4.28,  3.26,  4.17,  5.03, 31.8 ,
        3.5 ,  4.56,  3.06,  4.72,  3.49,  4.57,  4.55,  4.46,  2.6 ,
        2.54,  4.4 ,  3.13,  2.88,  5.02,  5.23,  5.3 ,  4.04,  3.76,
        4.27,  2.5 ,  4.13,  4.91,  3.92,  3.4 ,  3.35,  3.51,  5.36,
        4.06,  5.05,  3.32,  4.3 ,  3.91,  3.93,  2.57,  4.18,  2.84,
        3.94,  4.42,  5.11,  4.64,  5.01,  4.12,  4.65,  4.94,  2.63,
        2.39,  4.38,  3.14,  2.61,  3.04,  4.35,  5.75,  4.69,  3.88,
        2.53,  2.56,  3.08,  4.62,  2.87,  2.41,  4.34,  4.43,  5.13,
        3.02,  4.39,  3.67,  5.32,  5.46,  3.05,  2.59,  3.07,  5.06,
        4.33,  4.82,  4.95,  4.51,  4.47,  5.07,  3.7 ,  4.81,  4.97,
        5.9 ,  4.37,  2.46,  5.2 ,  3.58,  3.28,  4.41,  3.39,  4.75,
        5.17,  4.83,  3.36,  4.77,  5.22,  5.15,  3.68,  4.63,  4.99,
        4.68,  2.36,  4.53,  5.45,  3.03,  4.9 ,  5.47,  4.31,  4.6 ,
        4.79,  4.84,  2.51,  3.75,  4.76,  5.21,  4.96,  5.26,  2.38,
        5.41,  2.55,  5. ,  5.18,  4.98,  5.12,  2.44,  5.19,  4.92,
        5.73,  4.86,  6.24,  5.24,  5.56,  5.25,  2.3 ,  5.53,  5.29,
        4.89,  5.48,  5.4 ,  1.53,  5.27,  5.65,  5.44,  2.35,  5.16,
        0. ,  5.39,  5.43,  5.69,  6.31,  6.03,  5.33,  2.32,  5.6 ,
        2.31,  2.27,  5.31,  5.54,  5.35,  2.34,  5.37,  2.37,  5.28,
        5.55,  6.72,  2.26,  6.27,  2.33,  5.67,  5.52,  5.62,  5.38,
        5.49,  5.77,  5.58,  2.28,  5.57,  5.61,  5.51,  6.17,  1.07,
        5.42,  2.24,  8.06,  2.06,  5.85])
```

Price -

```
array([ 499,  984, 6289, ..., 8771, 3649, 6751])
```

- **TAB:7 CHECKING FOR ANOMALIES FOR VARIABLES IN THE DATASET**

Observations :
No Anomalies found in the Dataset.

Checking the Value counts on all the Categorical Column

Cut -

- As per the given Data Dictionary there are 5 cut quality of the cubic zirconia. Quality is increasing order Fair, Good, Very Good, Premium, Ideal.
- 10816 cubic zirconia are of the Ideal cut quality.
- 6899 cubic zirconia are of the Premium cut quality.
- 6030 cubic zirconia are of the Very Good cut quality.
- 2441 cubic zirconia are of the Good cut quality.
- 781 cubic zirconia are of the Fair cut quality.

Color -

- As per the given Data Dictionary color of the cubic zirconia ranges from D to J.
- D being the worst and J the best color.
- 1443 cubic zirconia are of the best color.
- 3344 cubic zirconia are of the wrost color.

Clarity -

- As per the given Data Dictionary cubic zirconia Clarity refers to the absence of the Inclusions and Blemishes (In order from Worst to Best) IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1.
- 894 cubic zirconia are of IF clarity.
- 1839 cubic zirconia are of VVS1 clarity.
- 2531 cubic zirconia are of VVS2 clarity.
- 4093 cubic zirconia are of VS1 clarity.
- 6099 cubic zirconia are of VS2 clarity.
- 6571 cubic zirconia are of SI1 clarity.
- 4575 cubic zirconia are of SI2 clarity.
- 365 cubic zirconia are of I1 clarity.

Observation :

There is no missing value & bad value present in the above categorical variables.

Checking of Bad Values :

1.Checking of bad values present in x (length of the cubic zirconia in mm)

As we found that x min value found to be inappropriate , so we need cleaned that.We know that min value for length of of the cubic zirconia in mm can't be 0. But in x (length of the cubic zirconia in mm.) we found min value of 0 (zero) this has to be cleaned.

	carat	cut	color	clarity	depth	table	x	y	z	price
5821	0.71	Good	F	SI2	64.1	60.0	0.0	0.0	0.0	2130
6215	0.71	Good	F	SI2	64.1	60.0	0.0	0.0	0.0	2130
17506	1.14	Fair	G	VS1	57.5	67.0	0.0	0.0	0.0	6381

From above records we observe that '0' in x has been entered maybe because the data was not available or by mistake of data entry operator. However, this data has to be imputed. We can either impute it with mean/median value or make some assumption.

2.Checking of bad values present in y (width of the cubic zirconia in mm).

As we found that y min value found to be inappropriate , so we need cleaned that.We know that min value for width of the cubic zirconia in mm can't be 0. But in y (width of the cubic zirconia in mm.) we found min value of 0 (zero) this has to be cleaned.

	carat	cut	color	clarity	depth	table	x	y	z	price
5821	0.71	Good	F	SI2	64.1	60.0	0.0	0.0	0.0	2130
6215	0.71	Good	F	SI2	64.1	60.0	0.0	0.0	0.0	2130
17506	1.14	Fair	G	VS1	57.5	67.0	0.0	0.0	0.0	6381

From above records we observe that '0' in y has been entered maybe because the data was not available or by mistake of data entry operator. However, this data has to be imputed. We can either impute it with mean/median value or make some assumption.

3.Checking of bad values present in z (height of the cubic zirconia in mm).

As we found that z min value found to be inappropriate , so we need cleaned that.We know that min value for height of of the cubic zirconia in mm can't be 0. But in z (width of the cubic zirconia in mm.) we found min value of 0 (zero) this has to be cleaned.

	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
6215	0.71	Good	F	SI2	64.1	60.0	0.00	0.00	0.0	2130
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

As we found that z min value found to be inappropriate , so we need cleaned that.We know that min From above records we observe that '0' in z has been entered maybe because the data was not available or by mistake of data entry operator. However, this data has to be imputed. We can either impute it with mean/median value or make some assumption.

Conclusion :

Here we infer that x(length),y(width) & z(height) have zero as min value.we observe that '0' in x(length),y(width) & z(height) has been entered maybe because the data was not available or by mistake of data entry operator. However, this data has to be imputed. As per the instruction we are going to impute bad value (0) for x(length),y(width) & z(height) with median in Question 1.2.So we are going to impute the zero present in x(length),y(width) & z(height) down the line.

Checking Duplicate Values -

Here we found the no of duplicated rows in data set i.e. 34 , as we know that duplicated rows are not useful we decided drop them by using .drop( ) function.

After using the .drop function ( ) , we drop all the duplicate rows of the data ,check the shape of the data once again.

Number of Rows : 269633
Number of Columns :10

Observation :

Number of duplicate rows = 0

Univariate Analysis of Numerical Variables - Histogram & Box-plot

A histogram takes as input a numeric variable only. The variable is cut into several bins, and the number of observation per bin is represented by the height of the bar. It is possible to represent the distribution of several variable on the same axis using this technique.

A box-plot gives a nice summary of one or several numeric variables. The line that divides the box into 2 parts represents the median of the data. The end of the box shows the upper and lower quartiles. The extreme lines show the highest and lowest value excluding outliers.

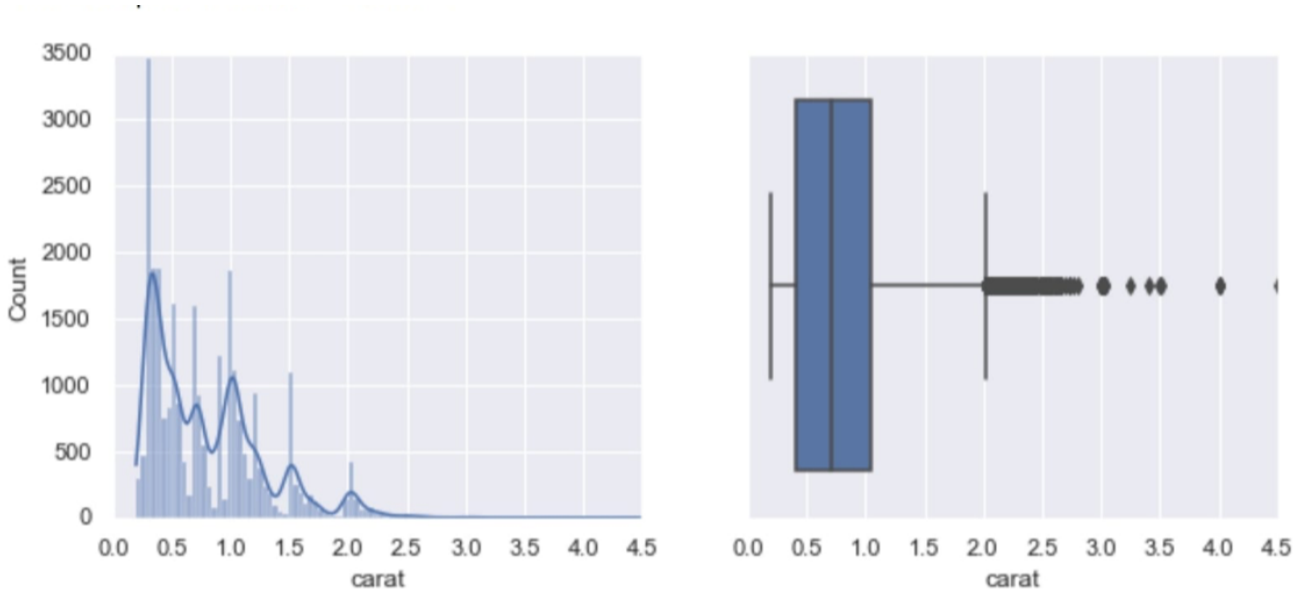


FIG:1 Histogram & Box-Plot of Carat

```
count    26933.000000
mean         0.798010
std          0.477237
min          0.200000
25%          0.400000
50%          0.700000
75%          1.050000
max          4.500000
Name: carat, dtype: float64
```

Statistical Description of Carat

Insights :

- Carat: weight of the cubic zirconia ranges from a minimum of 0.200 to maximum of 4.500.
- The average Carat: weight of the cubic zirconia is around 0.798.
- The standard deviation of the Carat: weight of the cubic zirconia is 0.477.
- 25% , 50% (median) and 75 % of the Carat: weight of the cubic zirconia are 0.400 , 0.700 and 1.050.
- Skewness indicating that the distribution is right skewed. (Skew Value - 1.11472)
- Carat: weight of the cubic zirconia have outliers.

```
<AxesSubplot:xlabel='depth'>
```

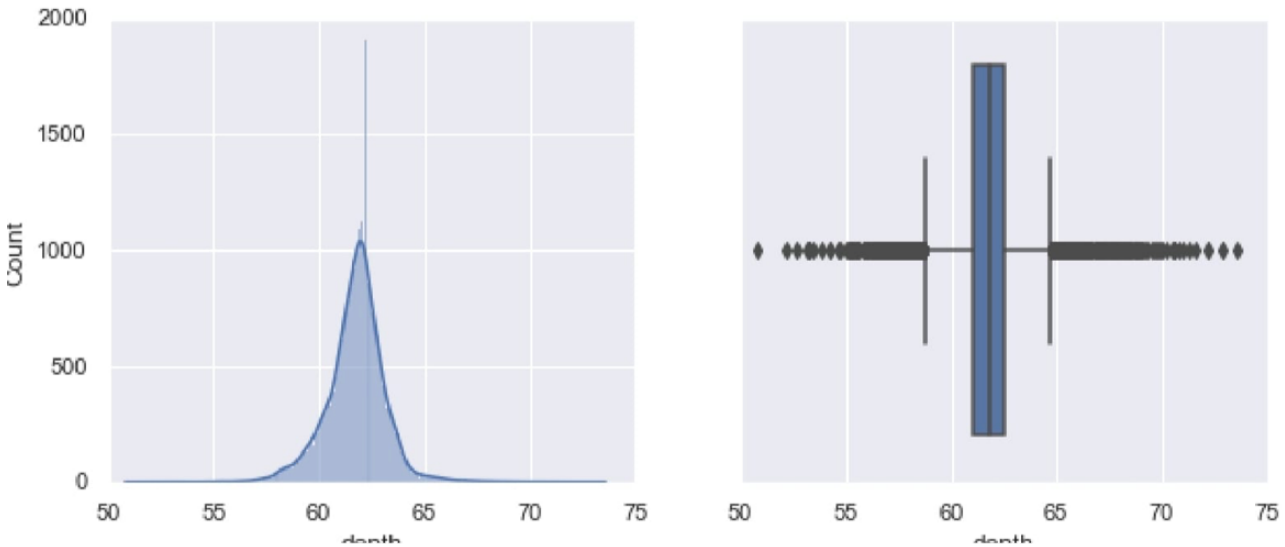


FIG:2 Histogram & Box-Plot of Depth

```
count    26236.000000
mean        61.745285
std          1.412243
min         50.800000
25%         61.000000
50%         61.800000
75%         62.500000
max         73.600000
Name: depth, dtype: float64
```

Statistical Description of Depth

Insights :

- Depth: The Height of cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter ranges from a minimum of 50.800 to maximum of 73.600.
- The average Depth: The Height of cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter is around 61.7.
- The standard deviation of the Depth: The Height of cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter is 1.412.
- 25%, 50% (median) and 75% of the Depth: The Height of cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter are 61.000, 61.800 and 62.500.
- Skewness indicating that the distribution is normal distributed.(SkewValue - o)
- Depth: The Height of cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter have outliers.

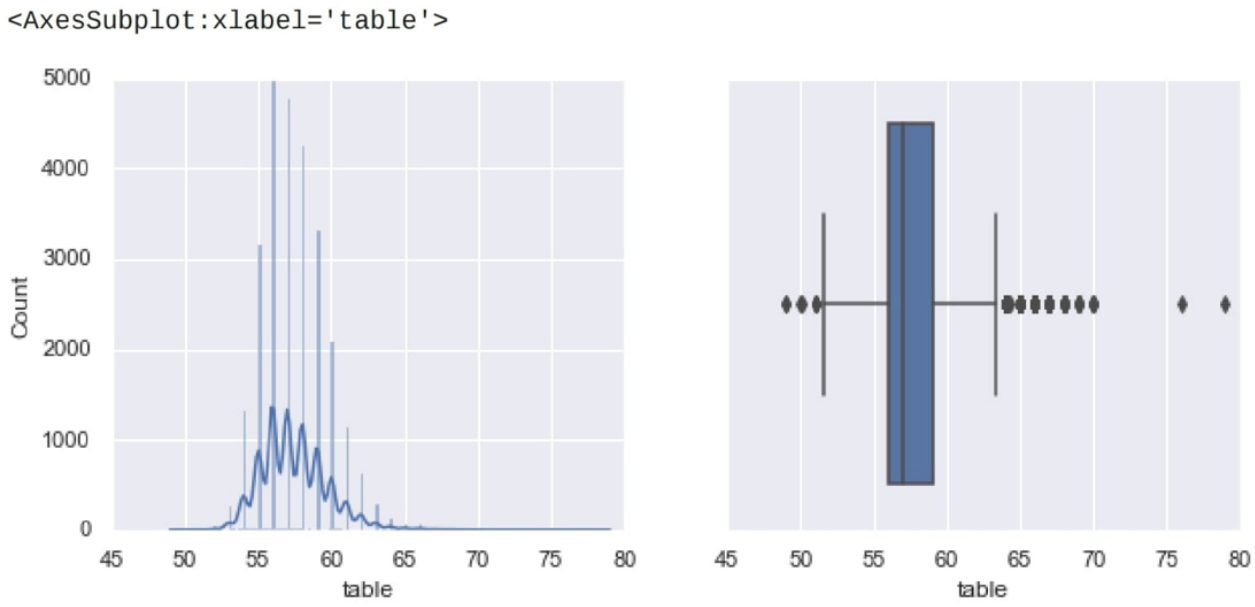


FIG:3 Histogram & Box-Plot of Table

```
count    26933.000000
mean        57.455950
std          2.232156
min         49.000000
25%         56.000000
50%         57.000000
75%         59.000000
max         79.000000
Name: table, dtype: float64
```

Statistical Description of Table

Insights:

- Table: The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter ranges from a minimum of 49.000 to maximum of 79.000.
- The average Table: The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter is around 57.455.
- The standard deviation of the Table: The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter is 2.232.
- 25% , 50% (median) and 75 % of the Table: The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter are 56.000 , 57.000 and 59.000.
- Skewness indicating that the distribution is slightly right skewed.(Skew Value - 0.76576)
- Table: The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter have outliers.

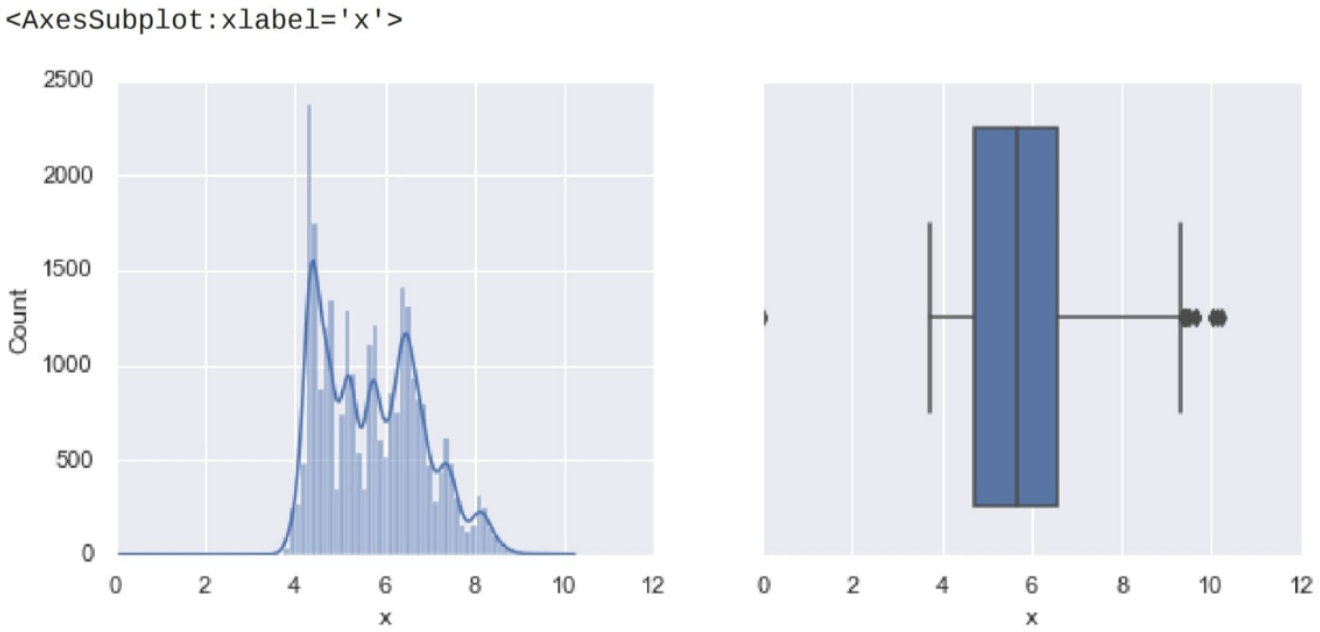


FIG:4 Histogram & Box-Plot of x

```
count    26933.000000
mean         5.729346
std          1.127367
min          0.000000
25%          4.710000
50%          5.690000
75%          6.550000
max         10.230000
Name: x, dtype: float64
```

Statistical Description of X

Insights :

- X: Length of the cubic zirconia in mm ranges from a minimum of 0 to maximum of 10.230.
- The average X: Length of the cubic zirconia in mm is around 5.729.
- The standard deviation of the X: Length of the cubic zirconia in mm is 1.127.
- 25% , 50% (median) and 75 % of the X: Length of the cubic zirconia in mm are 4.710 , 5.690 and 6.550.
- Skewness indicating that the distribution is slightly right skewed.(Skew Value -0.392)
- X: Length of the cubic zirconia in mm have outliers.

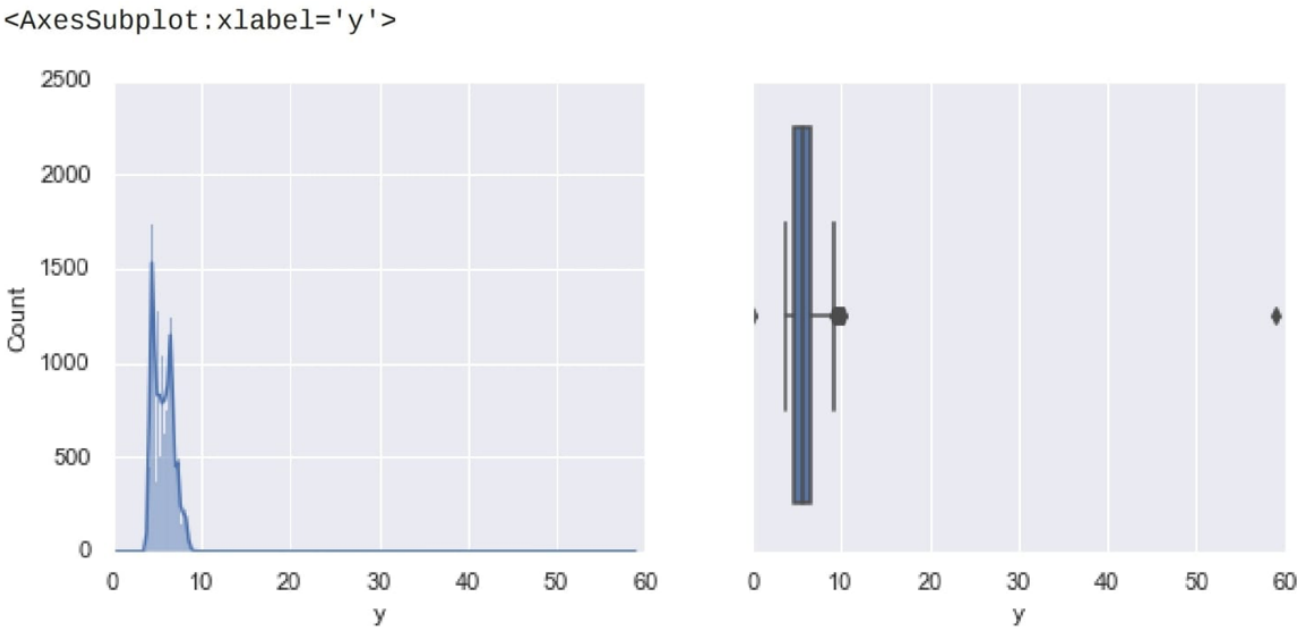


FIG:5 Histogram & Box-Plot of y

```
count    26933.000000
mean         5.733102
std          1.165037
min          0.000000
25%          4.710000
50%          5.700000
75%          6.540000
max         58.900000
Name: y, dtype: float64
```

Statistical Description of Y

Insights :

- Y : Width of the cubic zirconia in mm ranges from a minimum of 0 to maximum of 58.900.
- The average Y: Width of the cubic zirconia in mm is around 5.733.
- The standard deviation of the Y: Width of the cubic zirconia in mm is 1.165.
- 25% , 50% (median) and 75 % of the Y: Width of the cubic zirconia in mm are 4.720 , 5.700 and 6.540.
- Skewness indicating that the distribution is right skewed.(Skew Value - 3.867)
- Y: Width of the cubic zirconia in mm have outliers.

```
<AxesSubplot:xlabel='z'>
```

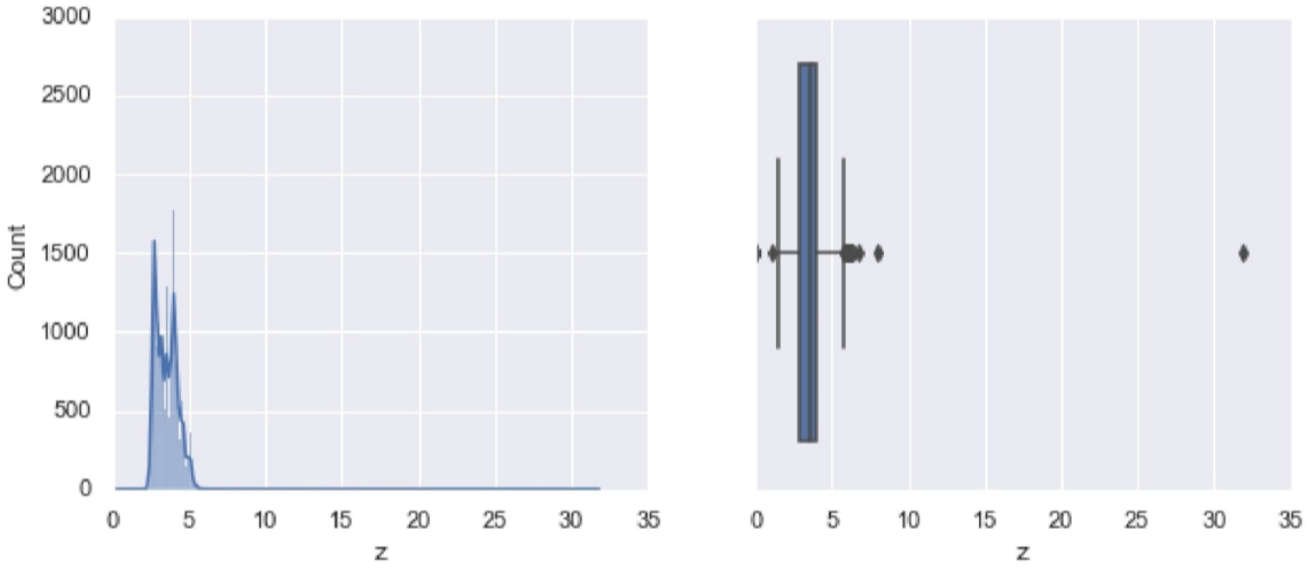


FIG: 6 Histogram & Box-Plot of z

```
count    26933.000000
mean         3.537769
std          0.719964
min          0.000000
25%          2.900000
50%          3.520000
75%          4.040000
max         31.800000
Name: z, dtype: float64
```

Statistical Description of Z

Insights :

- Z: Height of the cubic zirconia in mm ranges from a minimum of 0 to maximum of 31.800.
- The average Z: Height of the cubic zirconia in mm is around 3.538.
- The standard deviation of the Z: Height of the cubic zirconia in mm is 0.719.
- 25% , 50% (median) and 75 % of the Z: Height of the cubic zirconia in mm are 2.900 , 3.520 and 4.040.
- Skewness indicating that the distribution is right skewed.(Skew Value - 2.5805)
- Z: Height of the cubic zirconia in mm have outliers.

```
<AxesSubplot:xlabel='price'>
```

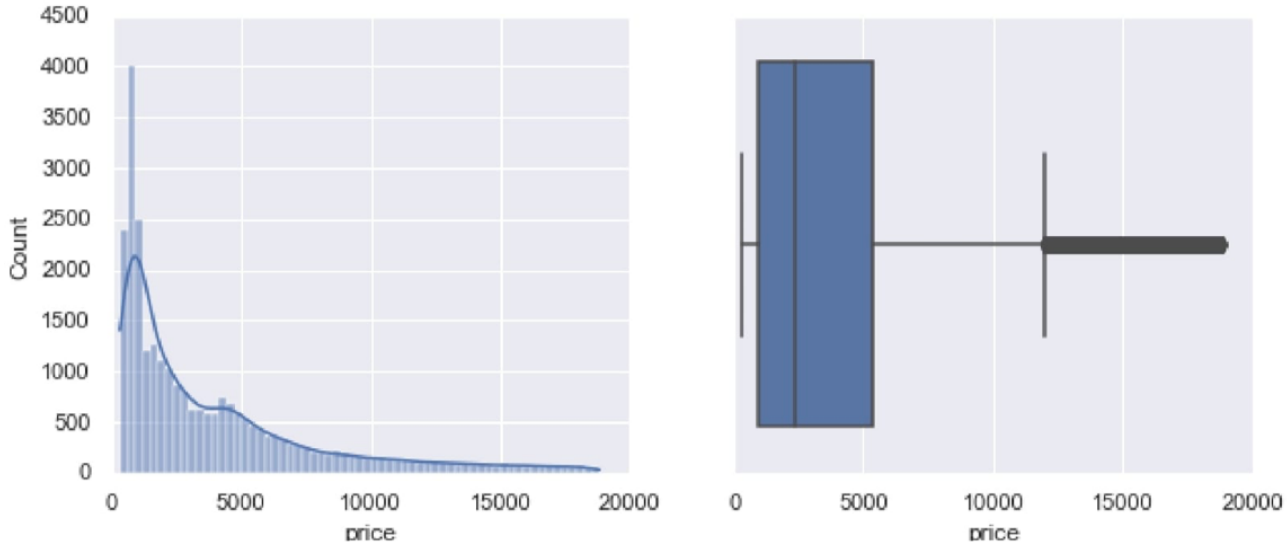


FIG: 7 Histogram & Box-Plot of Price

```
count    26933.000000
mean      3937.526120
std       4022.551862
min        326.000000
25%        945.000000
50%       2375.000000
75%       5356.000000
max      18818.000000
Name: price, dtype: float64
```

Statistical Description of Price

Insights :

- Price:The Price of the cubic zirconia ranges from a minimum of 326.00 to maximum of 18818.00.
- The average Price: The Price of the cubic zirconia is around 3937.526.
- The standard deviation of the Price: The Price of the cubic zirconia is 4022.55.
- 25% , 50% (median) and 75 % of the Price: The Price of the cubic zirconia are 945.00 , 2375.00 and 5356.00.
- Skewness indicating that the distribution is right skewed.(Skew Value - 1.619)
- Price: The Price of the cubic zirconia have outliers.

Univariate Analysis of Categorical Variables -

*** Countplot :**
A countplot is kind of like a histogram or a bar graph for categorical variables.

*** Piechart :**
A pie chart is a circle divided into sectors that each represent a proportion of the whole. It is often used to show proportion, where the sum of the sectors equal 100%.

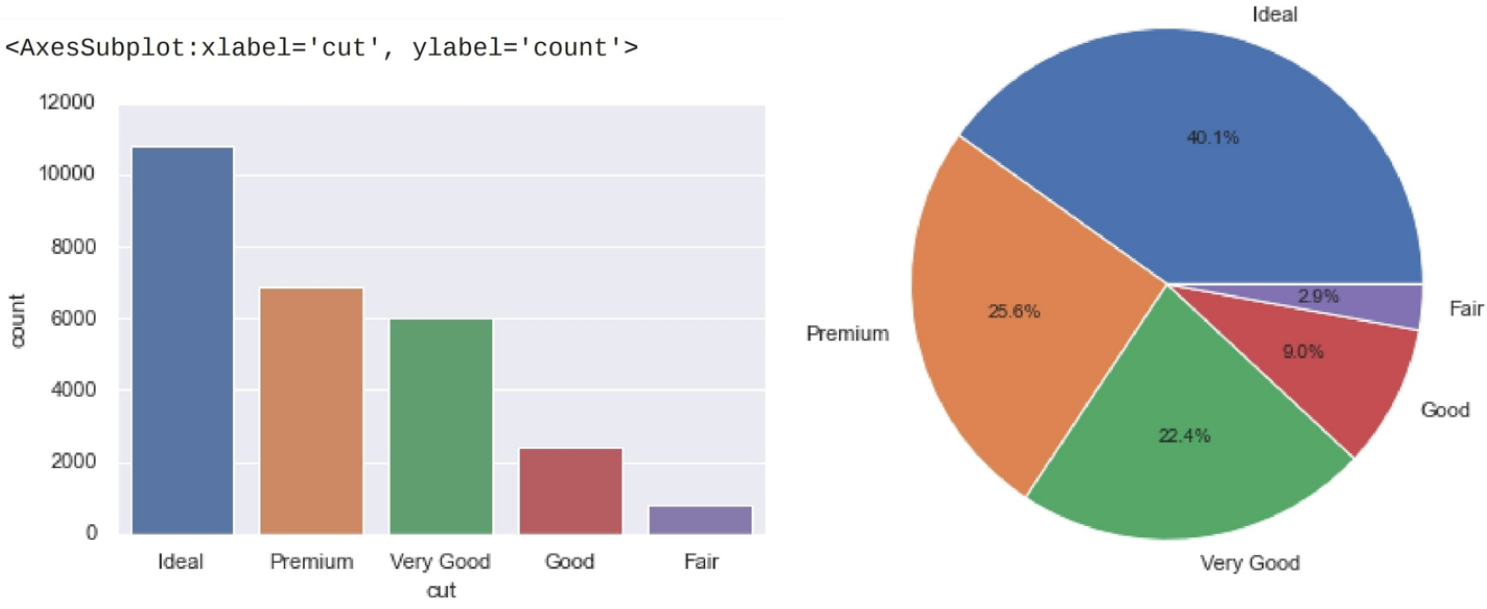


FIG: 8 Count Plot & Pie Chart of Cut

Insights :

- There are 5 type of cut quality of the cubic zirconia present in the data set named as 'Ideal', 'Premium', 'Very Good', 'Good' & 'Fair'.
- Cut Quality is increasing order Fair, Good, Very Good, Premium, Ideal.
- 40.11% cubic zirconia have Ideal cut quality which is the max among all 5 cut quality present in the data.
- 25.6% cubic zirconia have Premium cut quality.
- 22.4% cubic zirconia have Very Good cut quality.
- 9.0% cubic zirconia have Good cut quality.
- Only 2.9% cubic zirconia have Fair cut quality which is the min among all 5 cut quality present in the data.

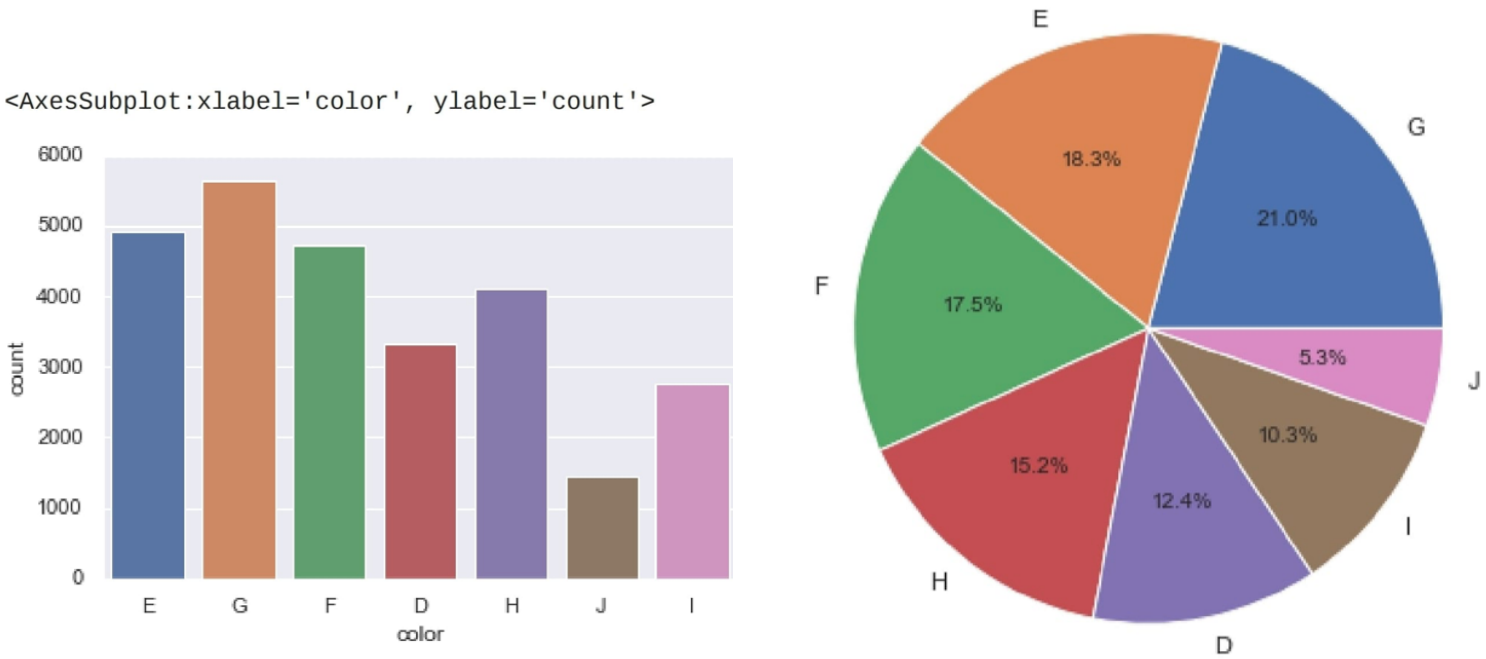


FIG: 9 Count Plot & Pie Chart of Color

Insights :

- There are 7 type of color of the cubic zirconia present in the data set named as 'D', 'E', 'F', 'G', 'H', 'I' & 'J'.
- With D being the worst and J the best.
- 21% cubic zirconia are of G color which is the max among all 7 color present in the data.
- 18.3% cubic zirconia are of E color.
- 17.5% cubic zirconia are of F color.
- 15.2% cubic zirconia are of H color.
- 12.4% cubic zirconia are of D color.
- 10.3% cubic zirconia are of I color.
- 5.3% cubic zirconia are of J color which is the min among all 7 color present in the data.

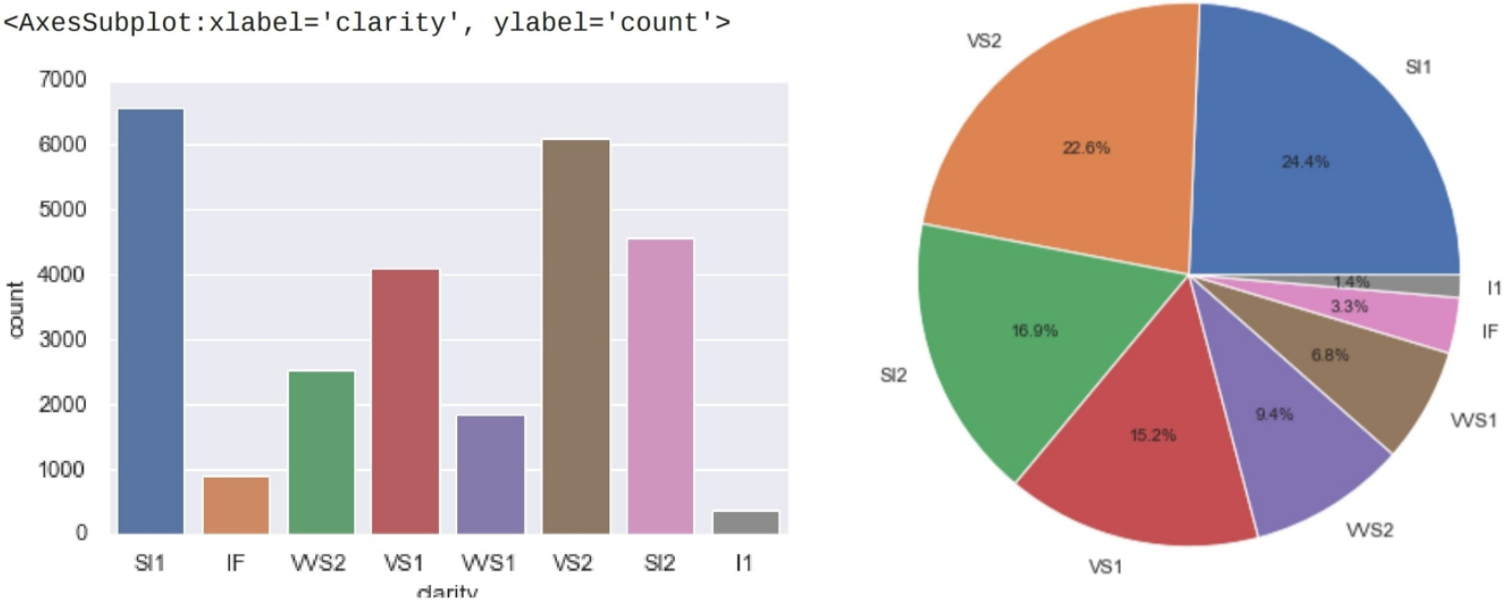


FIG: 10 Count Plot & Pie Chart of Clarity

Insights :

- There are 8 type of clarity of the cubic zirconia present in the data set named as 'IF', 'VVS1', 'VVS2', 'VS1', 'VS2', 'SI1', 'SI2', 'I1'.
- cubic zirconia clarity refers to the absence of the Inclusions and Blemishes. (In order from Worst to Best) IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1.
- 24.4% of cubic zirconia are of SI1 clarity which is the max among all 8 clarity quality present in the data.
- 22.6% of cubic zirconia are of VS2 clarity.
- 16.9% of cubic zirconia are of SI2 clarity.
- 15.2% of cubic zirconia are of VS1 clarity.
- 9.4% of cubic zirconia are of VVS2 clarity.
- 6.8% of cubic zirconia are of VVS1 clarity.
- 3.3% of cubic zirconia are of IF clarity.
- 1.4% of cubic zirconia are of I1 clarity which is the min among all 8 clarity quality present in the data.

Bivariate Analysis

*** Scatter Plot**

A scatter plot (aka scatter chart, scatter graph) uses dots to represent values for two different numeric variables. The position of each dot on the horizontal and vertical axis indicates values for an individual data point. Scatter plots are used to observe relationships between variables.

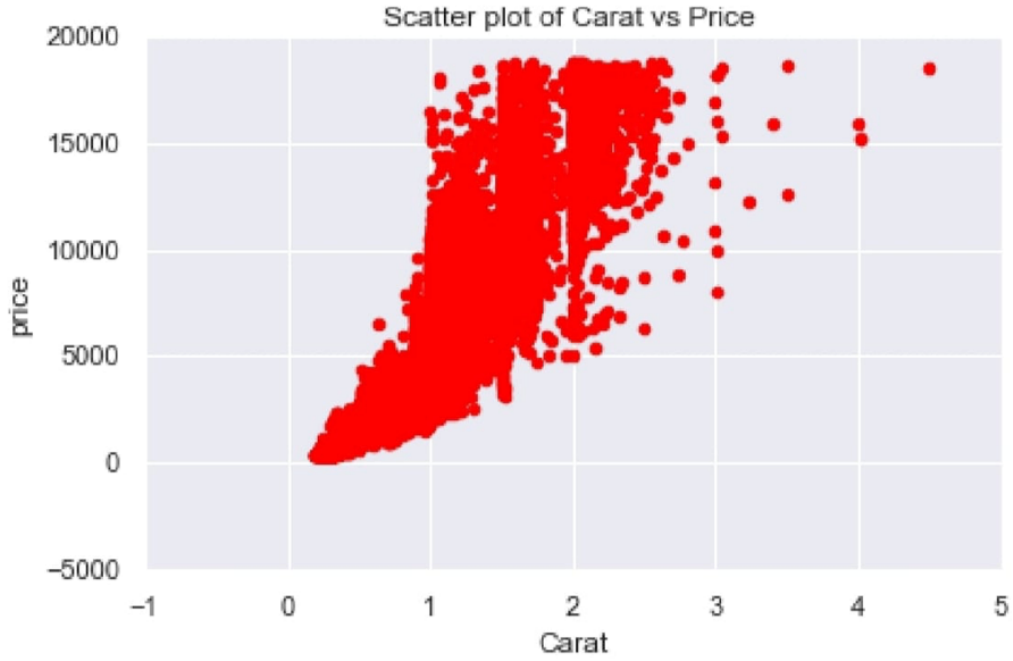


FIG: 11 Scatter Plot Carat VS Price

Insights:

From the above plot we see that the carat and the price is showing a strong relationship, with increase in carat price is also increases.

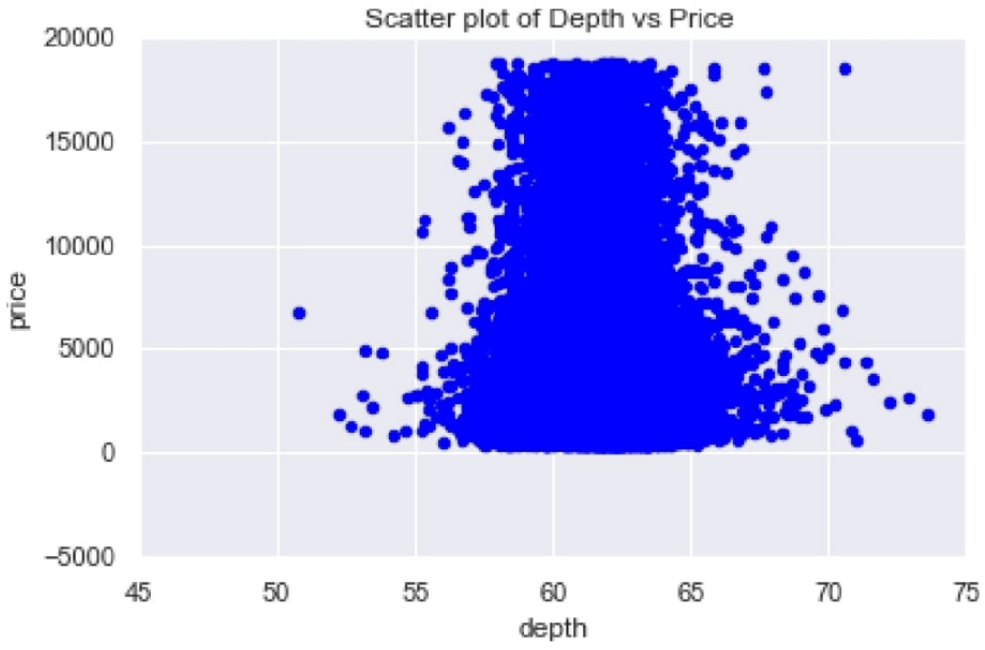


FIG: 12 Scatter Plot Depth VS Price

Insights :

From the above plot we see that the depth and the price is showing a no relationship as all the data points are scatter as cloud around its mean.

FIG: 13 Scatter Plot Table VS Price

Insights :

From the above plot we see that the table and the price is showing a very poor relationship.

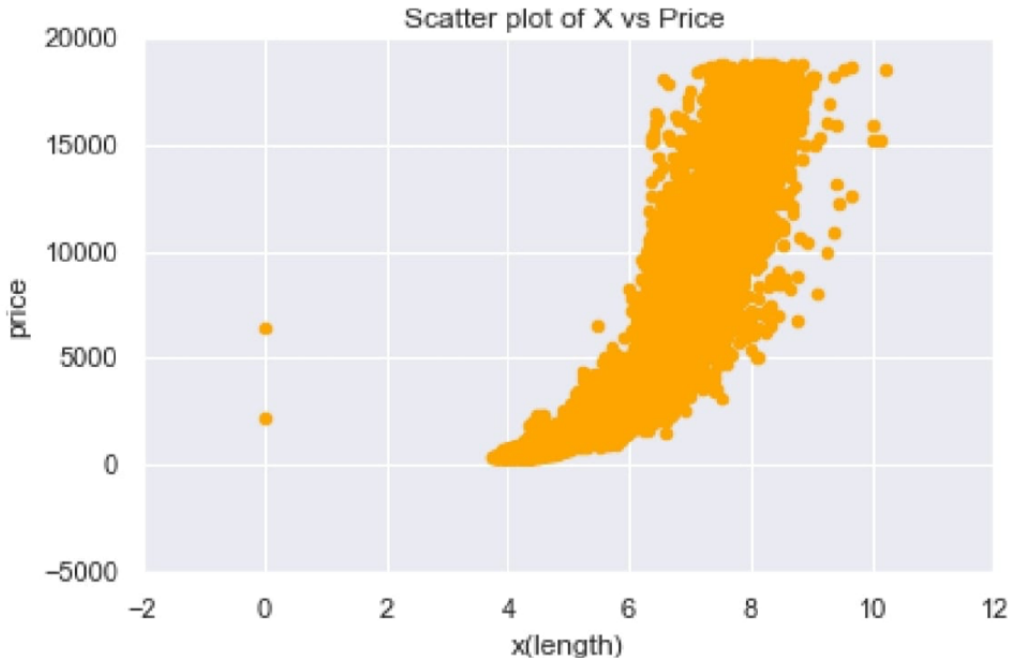


FIG: 14 Scatter Plot X VS Price

Insights :

From the above plot we see that the x and the price is showing a strong relationship with increase in x price is also increases

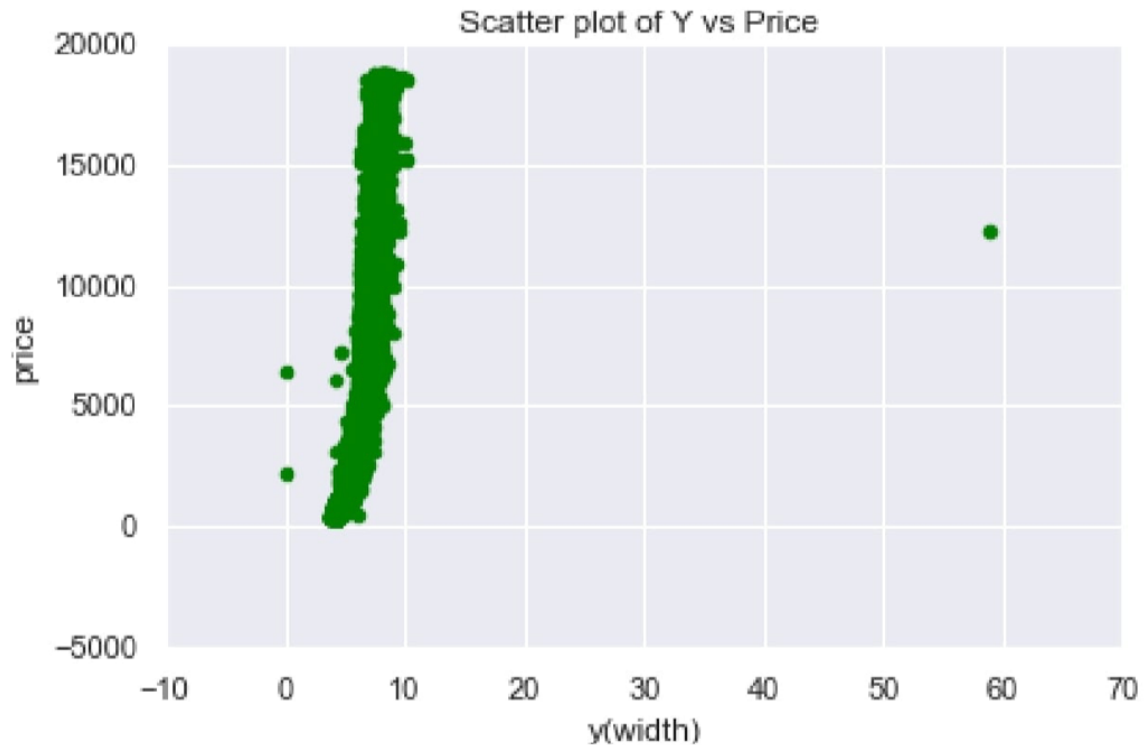


FIG: 15 Scatter Plot Y VS Price

Insights :

From the above plot we see that the y (width) and the price is showing a positive relationship with increase in y (width) price is also increases.

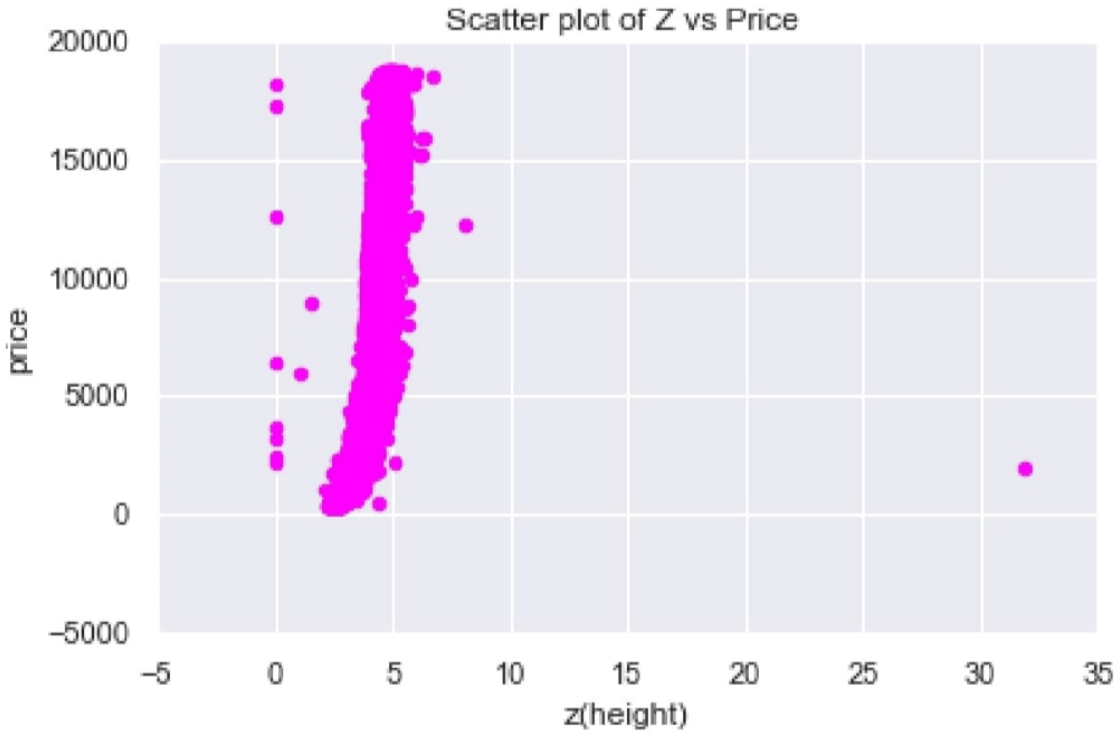


FIG: 16 Scatter Plot Z VS Price

Insights :

From the above plot we see that the z(height) and the price is showing a positive relationship with increase in z(height) price is also increases

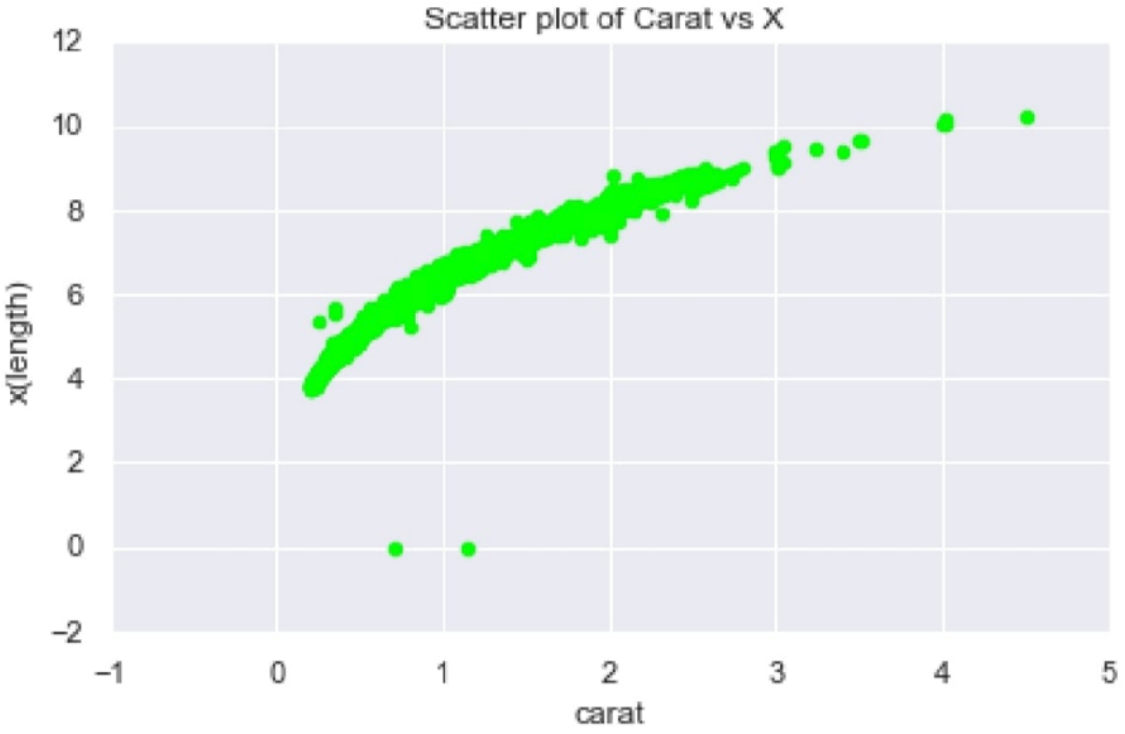


FIG: 17 Scatter Plot Carat VS X

Insights :

From the above plot we see that the carat and the x (length) is showing a positive relationship with increase in carat x (length) is also increases.

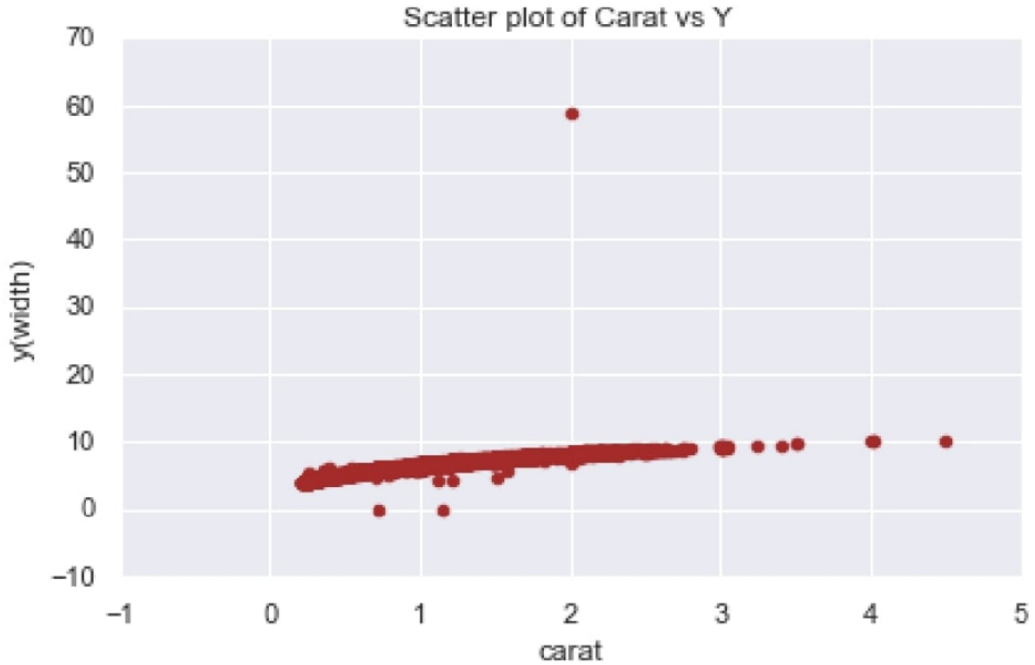


FIG: 18 Scatter Plot Carat VS Y

Insights :

From the above plot we see that the carat and the y(width) is showing a positive relationship with increase in carat y(width) is also increases.

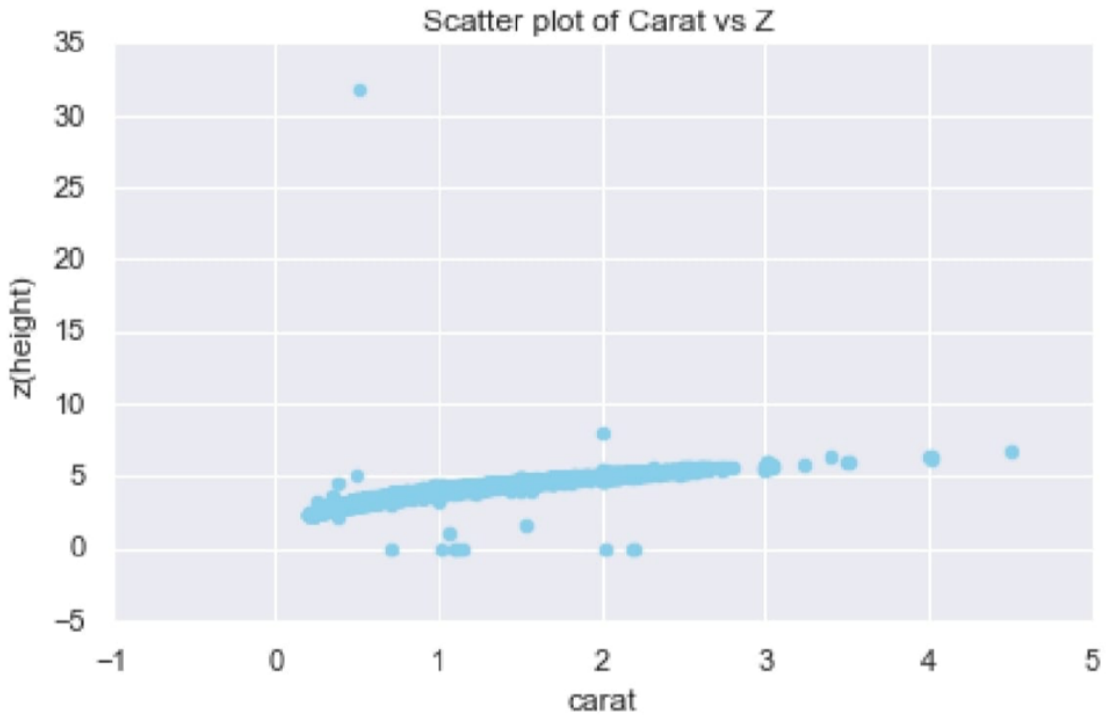


FIG: 19 Scatter Plot Carat VS Z

Insights :

From the above plot we see that the carat and the z(height) is showing a positive relationship with increase in carat z(height) is also increases.

***Count-Plot with Hue.**

A count-plot is kind of like a histogram or a bar graph for categorical variables.
Hue : This parameter take column name for colour encoding.

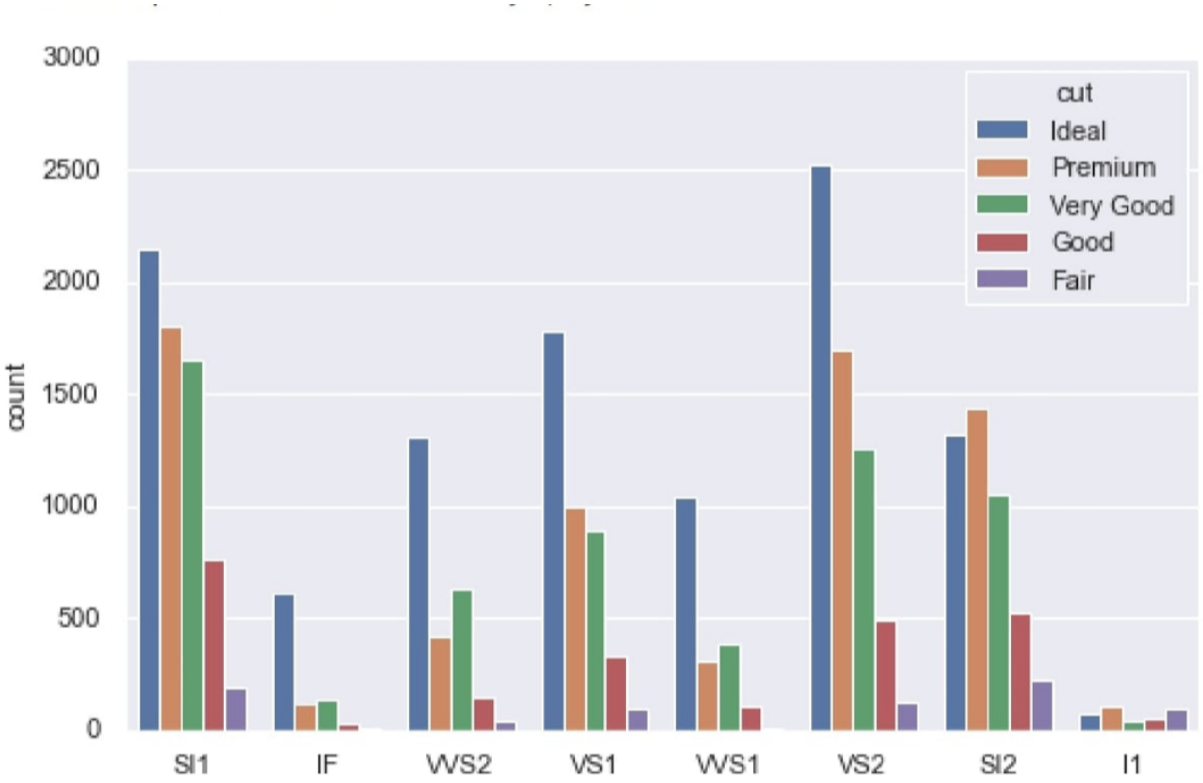


FIG: 20 Count Plot with Hue Clarity VS Cut

Insights :

- Note - cubic zirconia clarity refers to the absence of the Inclusions and Blemishes. (In order from Worst to Best) IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1.
- VS2 clarity cubic zirconia have maximum ideal cut quality.
- I1 clarity cubic zirconia have minimum ideal cut quality.
- SI1 clarity cubic zirconia have maximum premium cut quality.
- I1 clarity cubic zirconia have minimum premium cut quality.
- SI1 clarity cubic zirconia have maximum very good cut quality.
- I1 clarity cubic zirconia have minimum very good cut quality.
- SI1 clarity cubic zirconia have maximum good cut quality.
- IF clarity cubic zirconia have minimum good cut quality.
- SI2 clarity cubic zirconia have maximum fair cut quality.
- IF clarity cubic zirconia have minimum fair cut quality.

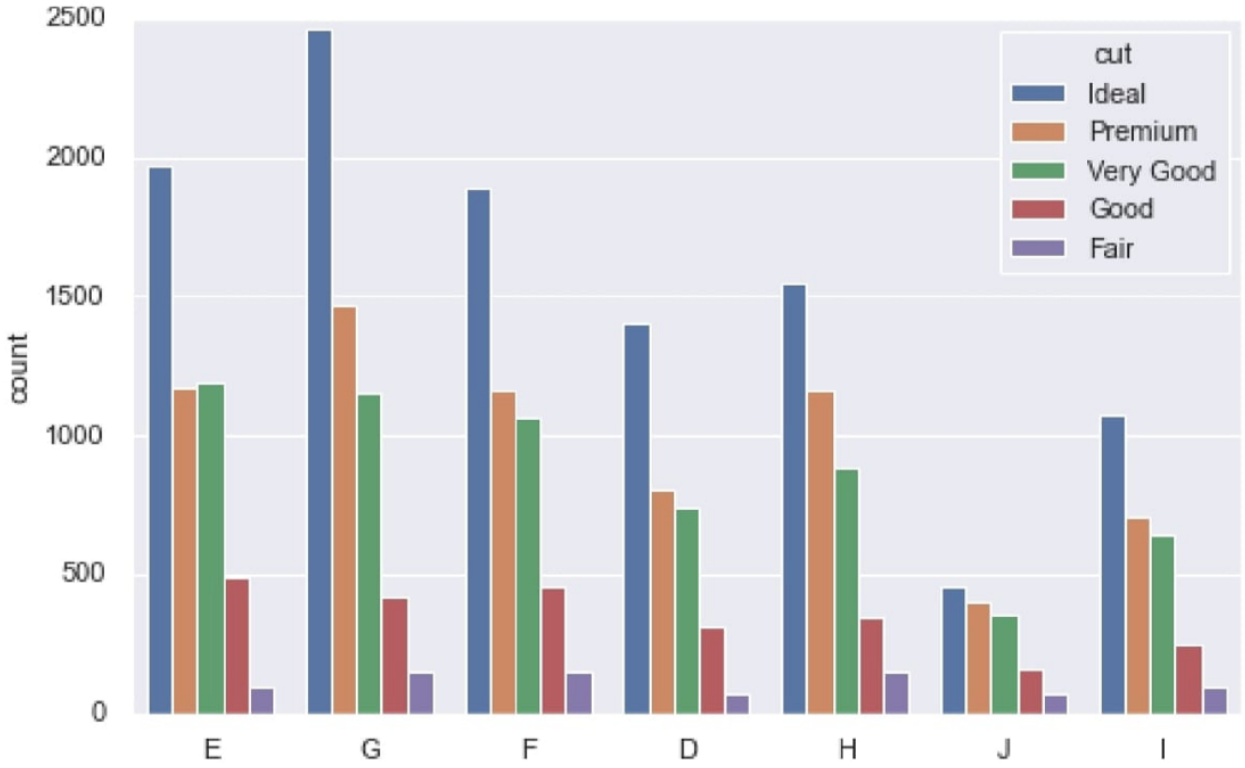


FIG: 21Count Plot with Hue Color VS Cut

Insights :

- G color cubic zirconia have maximum ideal cut quality.
- J color cubic zirconia have minimum ideal cut quality.
- G color cubic zirconia have maximum premium cut quality.
- J color cubic zirconia have minimum premium cut quality.
- E color cubic zirconia have maximum very good cut quality.
- J color cubic zirconia have minimum very good cut quality.
- E color cubic zirconia have maximum good cut quality.
- J color cubic zirconia have minimum good cut quality.
- H color cubic zirconia have maximum fair cut quality.
- J color cubic zirconia have minimum fair cut quality.

*** Heatmap**

A correlation heatmap uses colored cells, typically in a monochromatic scale, to show a 2D correlation matrix (table) between two discrete dimensions or event types.correlation heatmaps are ideal for comparing the measurement for each pair of dimension values.Darker Shades have higher correlation, while lighter shades have smaller values of correlation as compared to darker shades values.Correlation values near to 1 or -1 are highly positively correlated and highly negatively correlated respectively. Correlation values near to 0 are not correlated to each other.

Checking for Correlations :

	carat	depth	table	x	y	z	price
carat	1.000000	0.035240	0.181539	0.976858	0.941442	0.940982	0.922409
depth	0.035240	1.000000	-0.297768	-0.018401	-0.024453	0.101973	-0.002895
table	0.181539	-0.297768	1.000000	0.196254	0.182352	0.148994	0.126844
x	0.976858	-0.018401	0.196254	1.000000	0.962601	0.956490	0.886554
y	0.941442	-0.024453	0.182352	0.962601	1.000000	0.928725	0.856441
z	0.940982	0.101973	0.148994	0.956490	0.928725	1.000000	0.850682
price	0.922409	-0.002895	0.126844	0.886554	0.856441	0.850682	1.000000

TAB:8 CORRELATION TABLE

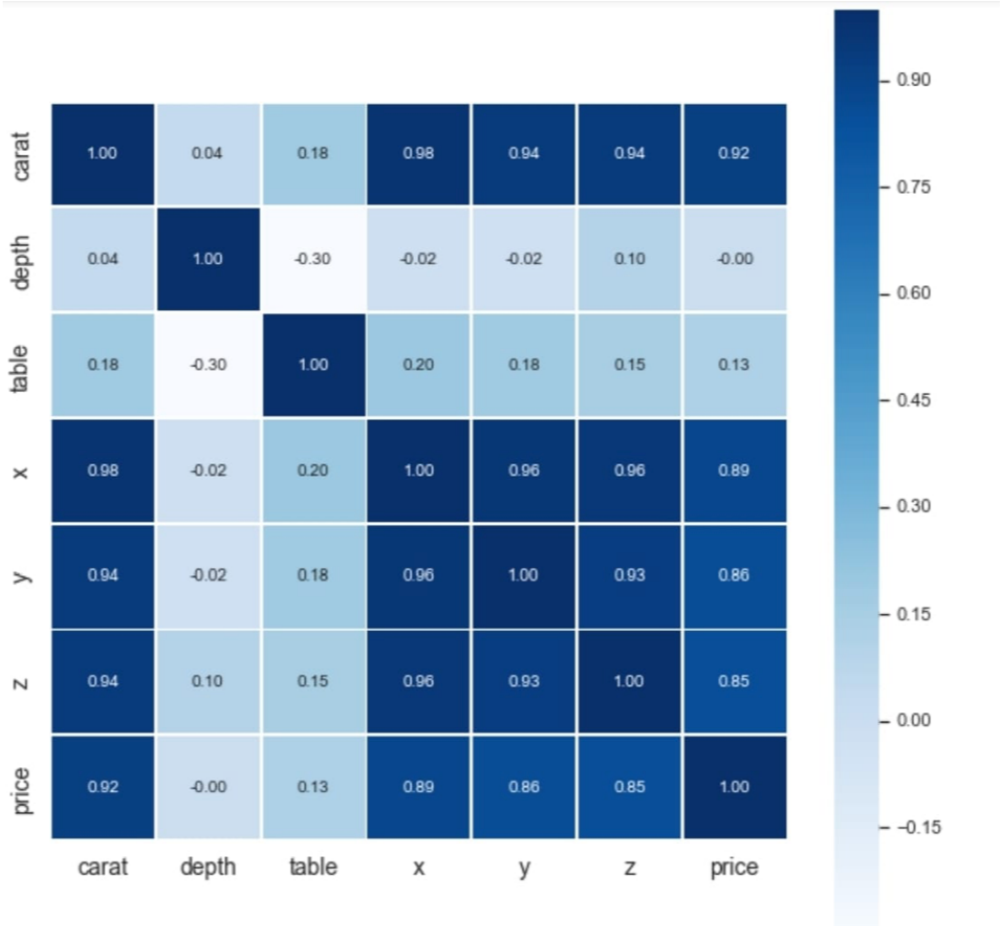


FIG: 22 Heat Map of Problem - 1

Insights :

- From the above correlation table & plot we conclude that -
- carat with x(length), y(width), z(height) have strong correlation i.e. 0.98 ,0.94 ,0.94.
- carat with price also have strong correlation i.e. 0.92.
- x(length), y(width), z(height) with price have strong correlation i.e. 0.89 ,0.86 ,0.85.
- carat with table have poor correlation i.e. 0.18.
- table with x(length), y(width), z(height) have poor correlation. i.e. 0.20, 0.18, 0.15.
- carat with depth have very poor correlation i.e. 0.04.
- table with price have poor correlation i.e. 0.13.
- table with depth shows negative correlation. i.e. -0.30.

***Pairplot**

Pairplot shows the relationship between the variables in the form of scatterplot and the distribution of the variable in the form of histogram.

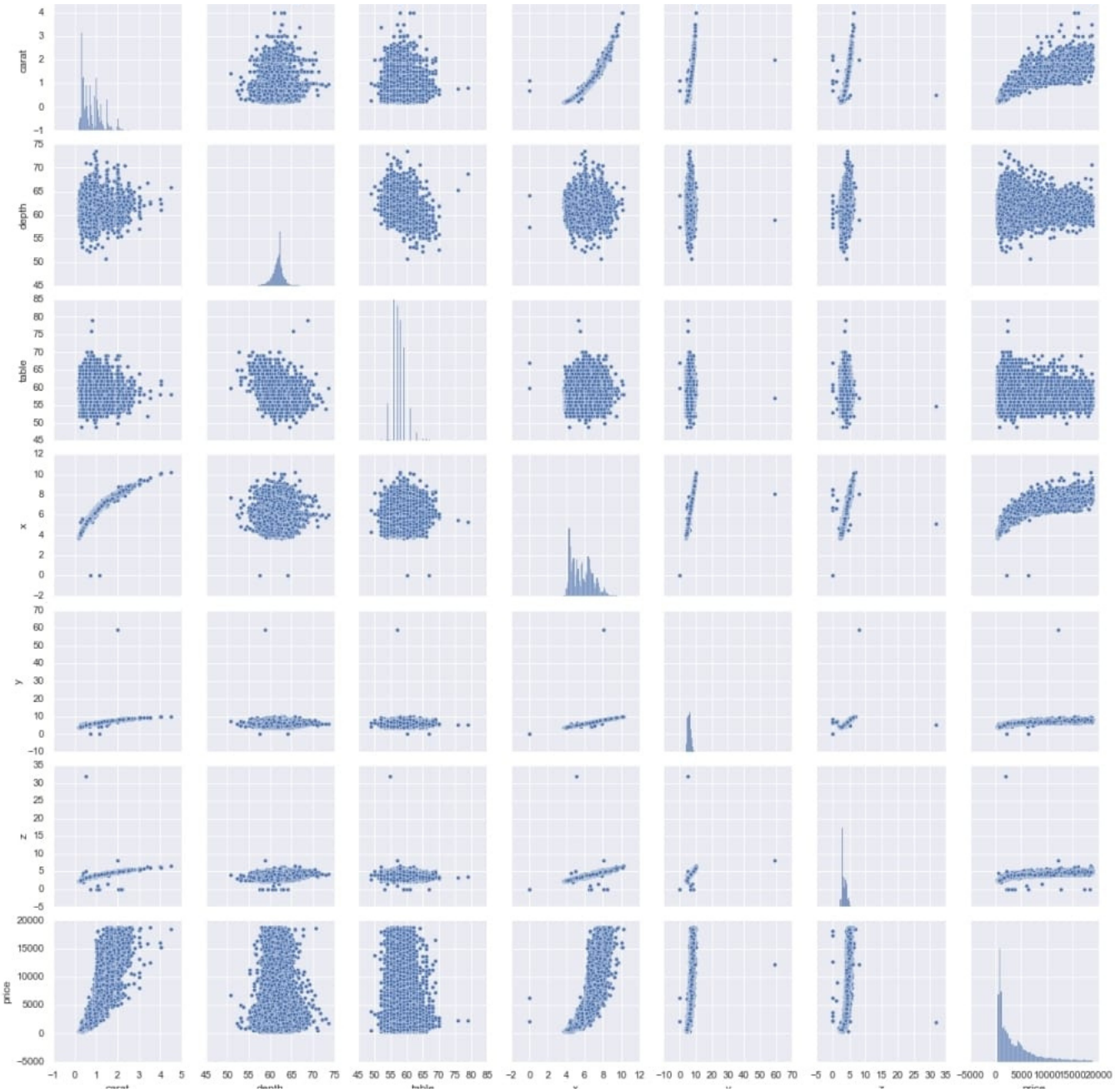


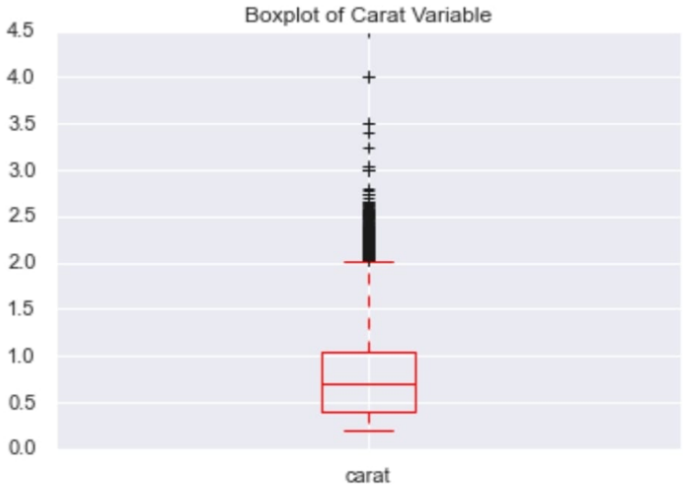
FIG: 23 Pairplot of Problem - 1

Insights

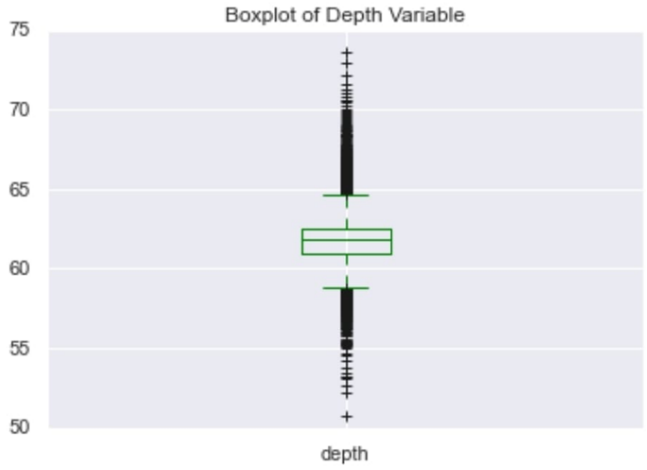
- carat with x(lenght), y(width), z(height) show positive relationship , carat increases the x(lenght), y(width), z(height) is also increasing.
- carat with price show positive relationship , carat increases the price is also increasing.
- price with x(lenght) show positive relationship ,price increases the x(lenght),is also increasing.
- price with y(width), z(height) show some positive relationship.

Checking for Outliers in the dataset.

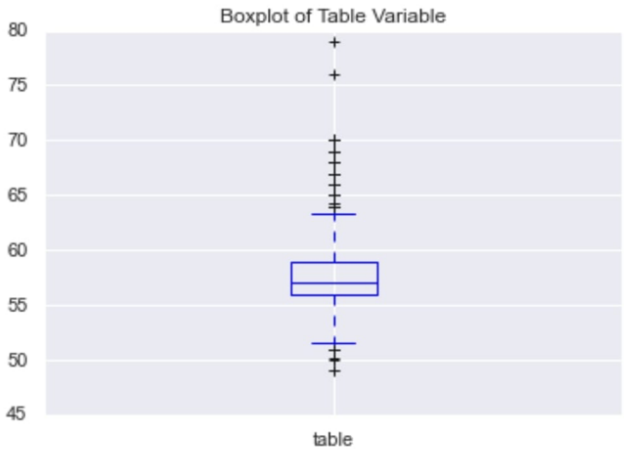
An observation is considered to be an outlier if that particular has been mistakenly captured in the data set. To check for outliers, we will be plotting the box plots.



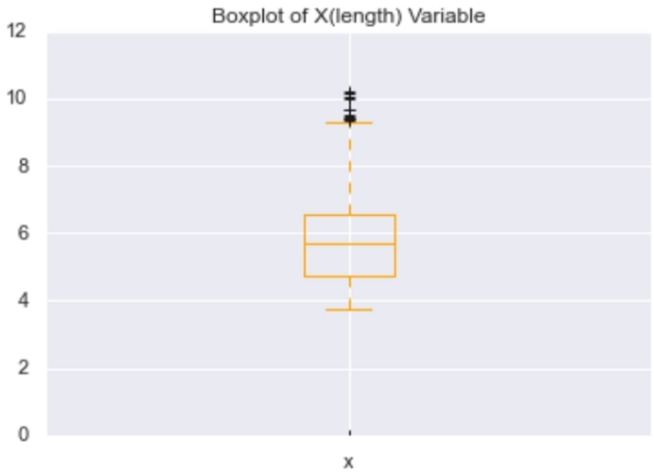
Box-Plot of Carat Variable



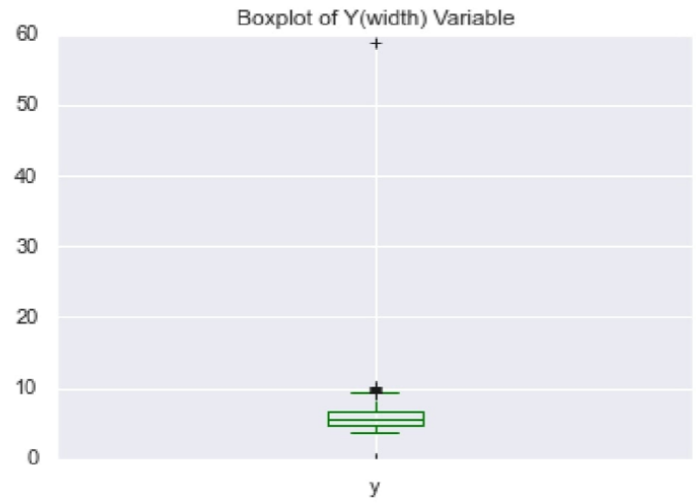
Box-Plot of Depth Variable



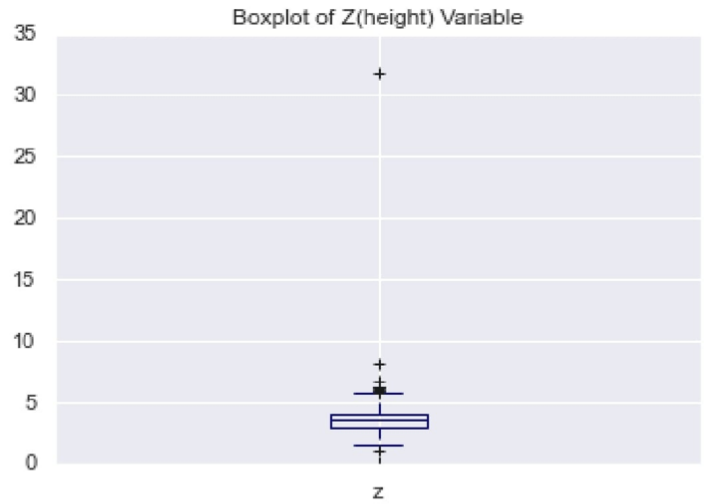
Box-Plot of Table Variable



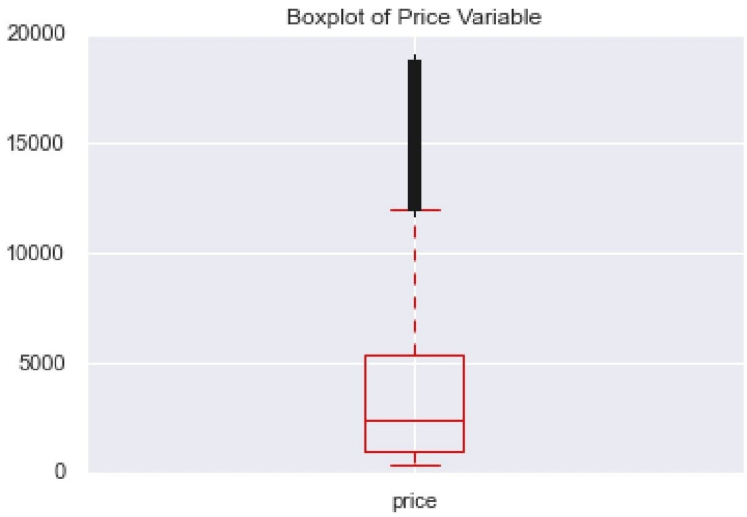
Box-Plot of X (length) Variable



Box-Plot of Y (Width) Variable



Box-Plot of Z (Height) Variable



Box-Plot of price

FIG: 24 OUTLIER DETECTION OF PROBLEM - 1

Insights

Looking at the box plot, it seems that their are 6 independent variables Carat, Depth , Table and X(length) , Y(width) , Z(height) and 1 dependent variable Price have outliers present in the variables.

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

Checking for Null Values.

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

Insights:

From the above function we infer that only depth variable have null values i.e. 697.As we know that Theoretically, 25 to 30% is the maximum missing values are allowed, beyond which we might want to drop the variable from analysis.Here we have 2.58% (approx) null values in the the depth variable we are going to impute the null values by median by using numpy .replace func( ).

Imputation of Null Values by .replace() function with median :

Imputation is the process of replacing missing data with substituted values like mean / median , if outliers are present then we impute with median ,if outliers are not present then we impute with the mean.Because missing data can create problems for analysing data, imputation is seen as a way to avoid pitfalls involved with list wise deletion of cases that have missing values.For imputation we are going to use the numpy .replace func( ).

replace() function, each element in array, return a copy of the string with all occurrences of substring old replaced by new.

We can see that we have various missing values in depth column. There are various ways of treating your missing values in the data set. And which technique to use when is actually dependent on the type of data you are dealing with.

In this exercise, we will use .replace( ) function for the numerical columns and replace the null values with the median value.

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

Insights:

We successfully impute the null values with median value by using the .replace() function , now we don't have any null values present in the given cubic zirconia dataset.

Check for the values which are equal to zero.

	count	mean	std	min	25%	50%	75%	max
carat	26933.0	0.798010	0.477237	0.2	0.40	0.70	1.05	4.50
depth	26933.0	61.746701	1.393875	50.8	61.10	61.80	62.50	73.60
table	26933.0	57.455950	2.232156	49.0	56.00	57.00	59.00	79.00
x	26933.0	5.729346	1.127367	0.0	4.71	5.69	6.55	10.23
y	26933.0	5.733102	1.165037	0.0	4.71	5.70	6.54	58.90
z	26933.0	3.537769	0.719964	0.0	2.90	3.52	4.04	31.80
price	26933.0	3937.526120	4022.551862	326.0	945.00	2375.00	5356.00	18818.00

There is bad values found in the x(lenght) , y(width) , z(height) columns of the Dataset. As x(lenght) , y(width) , z(height) are the length , width & height of the cubic zirconia in mm and we have found mininum value of x(lenght) , y(width) , z(height) is zero which doesnot make sense. As we know that length , width , height can't be zero. Thus, we need to treat & clean them.

Treatment of Bad Values or Zeros present in x(lenght) , y(width) , z(height) columns of the Dataset.

1.Treatment of bad values present in x (length of the cubic zirconia in mm) -

As we found that x min value found to be inappropriate , so we need cleaned that.We know that min value for length of of the cubic zirconia in mm can't be 0. But in x (length of the cubic zirconia in mm.) we found min value of 0 (zero) this has to be cleaned.

	carat	cut	color	clarity	depth	table	x	y	z	price
5821	0.71	Good	F	SI2	64.1	60.0	0.0	0.0	0.0	2130
17506	1.14	Fair	G	VS1	57.5	67.0	0.0	0.0	0.0	6381

From above records we observe that '0' in x has been entered maybe because the data was not available or by mistake of data entry operator. However, this data has to be imputed. We can either impute it with mean/median value or make some assumption.Here we are impute this with median by using the np.where ( ) function.

2.Treatment of bad values present in y (width of the cubic zirconia in mm) -

As we found that y min value found to be inappropirate , so we need cleaned that.We know that min value for width of of the cubic zirconia in mm can't be 0. But in y (width of the cubic zirconia in mm.) we found min value of 0 (zero) this has to be cleaned.

	carat	cut	color	clarity	depth	table	x	y	z	price
5821	0.71	Good	F	SI2	64.1	60.0	0.0	0.0	0.0	2130
17506	1.14	Fair	G	VS1	57.5	67.0	0.0	0.0	0.0	6381

From above records we observe that '0' in y has been enterd maybe because the data was not available or by mistake of data entry operator. However, this data has to be imputed. We can either impute it with mean/median value or make some assumption.Here we are impute this with median.

3.Treatment of bad values present in z (height of the cubic zirconia in mm).

As we found that z min value found to be inappropriate , so we need cleaned that.We know that min value for height of of the cubic zirconia in mm can't be 0. But in z (width of the cubic zirconia in mm.) we found min value of 0 (zero) this has to be cleaned.

	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
6215	0.71	Good	F	SI2	64.1	60.0	0.00	0.00	0.0	2130
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

From above records we observe that '0' in z has been entered maybe because the data was not available or by mistake of data entry operator. However, this data has to be imputed. We can either impute it with mean/median value or make some assumption.Here we are impute this with median by using the np.where ( ) function.

Result :

	count	mean	std	min	25%	50%	75%	max
carat	26933.0	0.798010	0.477237	0.20	0.40	0.70	1.05	4.50
depth	26933.0	61.746701	1.393875	50.80	61.10	61.80	62.50	73.60
table	26933.0	57.455950	2.232156	49.00	56.00	57.00	59.00	79.00
x	26933.0	5.729769	1.126285	3.73	4.71	5.69	6.55	10.23
y	26933.0	5.733525	1.163989	3.71	4.72	5.70	6.54	58.90
z	26933.0	3.538815	0.717377	1.07	2.90	3.52	4.04	31.80
price	26933.0	3937.526120	4022.551862	326.00	945.00	2375.00	5356.00	18818.00

We have successfully impute the bad value present in x , y & z with median value.Now we clearly infer that min value of x(lenght) , y(width) , z(height) is not zero anymore & we have appropirate min values for x(lenght) , y(width) , z(height) columns of the Dataset.

Geting unique counts of all Objects

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


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


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

TAB:9 UNIQUE COUNTS OF ALL OBJECTS

Combining the sub levels of a ordinal variables :

We combine the sub levels of a categorical ordinal variables in order to reduce the labels before encoding lets us assume if we do the one hot encoding then more new dummy cols of the label will be created and it can increase the dimensions of the dataset to keep ourself safe from the curse of dimensionality. We usually combine the sub levels of a variables.

We combine the sub levels of a categorical ordinal variables in order to reduce the labels before encoding because if we do the label encoding the label encoder or pd.codes func ( ) allots number from 0 upto the number of labels present in that variable so model will might be confuse & so this is advised for the model building too if we have some labels which can club into each other so we can proceed with clubbing of labels . That's why we combine the sub levels of a ordinal variables.

Cut

Cut - Quality is increasing order Fair, Good, Very Good, Premium, Ideal. We are going to club Good and Very Good labels.

```
Ideal        10805
Very Good     8462
Premium       6886
Fair           780
Name: cut, dtype: int64
```

After groupping we have 4 labels in the 'cut' labelled as - Quality is increasing order Fair, Very Good, Premium, Ideal.

Color -

Colour of the cubic zirconia.With D being the worst and J the best. Here we labelled the color first as :
'D' = 'Worst',
'E' ='Very Bad',
'F' ='Bad',
'G' ='Fair',
'H' ='Good',
'I' = 'Very Good',
'J' ='Best'

```
Best         1440
Very Good    2765
Worst        3341
Good         4095
Bad          4723
Very Bad     4916
Fair         5653
Name: color, dtype: int64
```

in order to understand better & will encode with label encoding for model building. After doing this we are going to club the Fair and Good to reduce the labels.

Here we are going to club the Fair and Good to reduce the labels.

```
Best         1440
Very Good    2765
Worst        3341
Bad          4723
Very Bad     4916
Good         9748
Name: color, dtype: int64
```

Now we have 6 labels in the 'color' labelled as - Quality is increasing in order 'Worst' , 'Very Bad' , 'Bad' , 'Good' , 'Very Good' , 'Best'.

Clarity
cubic zirconia Clarity refers to the absence of the Inclusions and Blemishes. (In order from Worst to Best) IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1. Here we labelled the clarity in order to understand better & will encode with label encoding for model building.

Here we labelled the clarity first as :
'IF'= 'Worst',
'VVS1'= 'Very Bad',
'VVS2'= 'Bad',
'VS1'= 'Fair',
'VS2'= 'Good',
'SI1'= 'Better',
'SI1'= 'Very Good'
'I1',='Best'

```
Best          364
Worst         891
Very Bad     1839
Bad          2530
Fair         4087
Very Good    4564
Good         6093
Better       6565
Name: clarity, dtype: int64
```

in order to understand better & will encode with label encoding for model building. After doing this we are going to club the Fair & Good and Better & Very Good to reduce the labels.

Here we are going to club the Fair & Good and Better & Very Good to reduce the labels.

```
Best          364
Worst         891
Very Bad     4369
Good        10180
Very Good   11129
Name: clarity, dtype: int64
```

Now we have 5 labels in the in the 'clarity' labelled as - Quality is increasing order 'Worst' , 'Very Bad' , 'Good' , 'Very Good' , 'Best'.

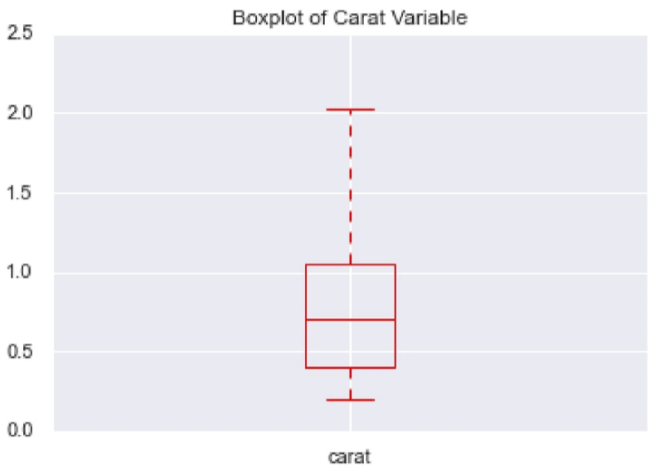
1.3 Encode the data (having string values) for Modelling. Split the data into train and test (70:30). Apply Linear regression using scikit learn. Perform checks for significant variables using appropriate method from statsmodel. Create multiple models and check the performance of Predictions on Train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

Preprocessing Steps for Linear Regression Model Building -

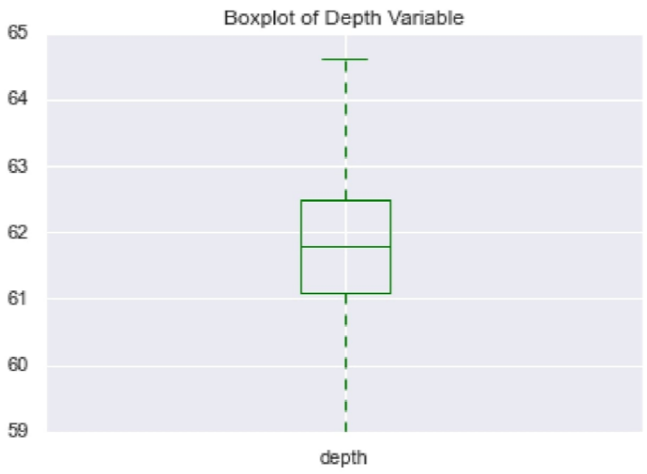
Treatment of Outliers-

Outliers are unusual values in your dataset, and they can distort statistical analyses and violate their assumptions. Outliers increase the variability in your data, which decreases statistical power. Consequently, excluding outliers can cause your results to become statistically significant. That's why are doing the outlier treatment.:

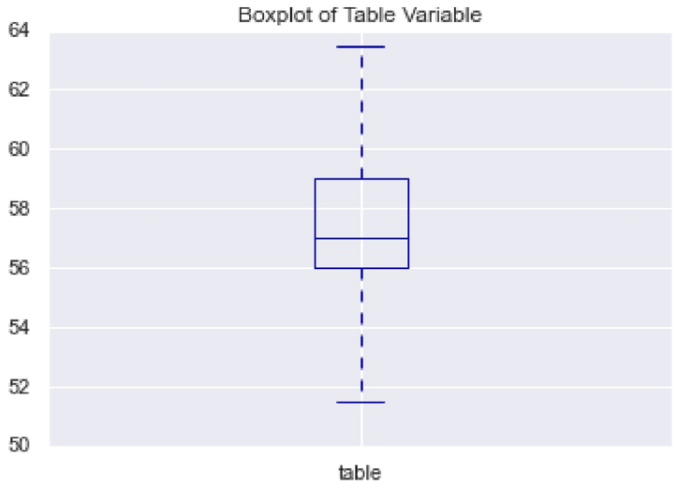
Here we treating the outlier for linear regression model , we treat the outliers of independent variables only because if we did the treatment of the dependent variable then our model predictions might be affected & we will not able to get right predictions. That's why we only treat the outliers of independent variables. We use 25% / 75% IQR imputation method for outliers treatment as we saw in the above box plots we have outliers in all independent features .Now here we treating the outliers.



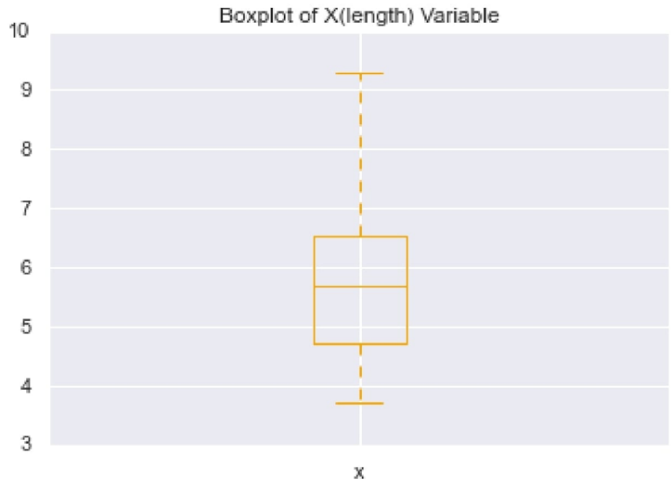
BoxPlot of Carat



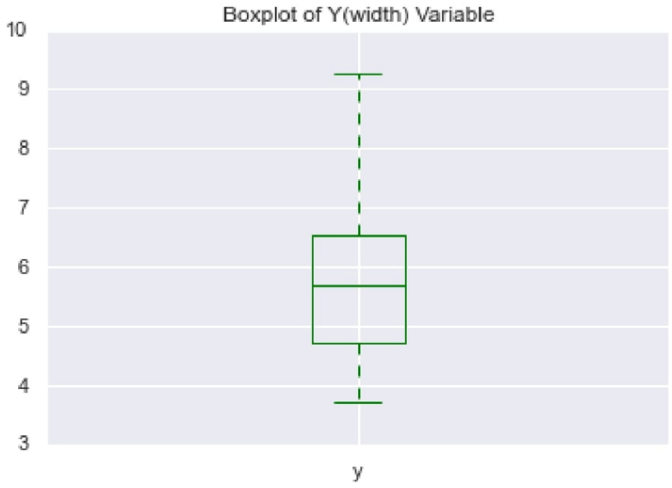
BoxPlot of Depth



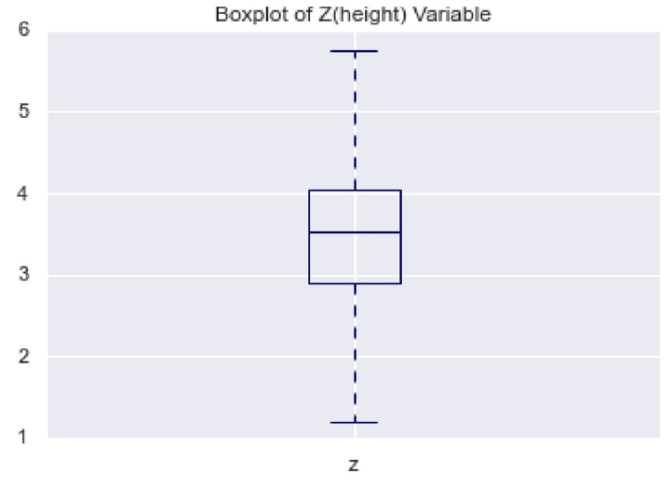
BoxPlot of Table



BoxPlot of X (length)



BoxPlot of Y



BoxPlot of Z

FIG: 25 OUTLIER TREATMENT OF PROBLEM - 1

We have successfully treat the outliers , now we donot have any outliers present in the independent variables of the cubic zirconia dataset.
Here we didn't treat the outliers of the dependent variable { Price } because if we did the treatment of the dependent variable then our model predictions might be affected will not able to get right predictions. That's why we only treat the outliers of independent variables.

Encoding for Linear Regression Model -

There are three Ordinal categorical Variable.

In Cut we have order of Quality is increasing order Fair, Very Good, Premium, Ideal.
In Color we have order of Quality is increasing order Quality is increasing order 'Worst' , 'Very Bad' , 'Bad' , 'Good' , 'Very Good' , 'Best'.
In Clarity we have order of Quality is increasing order Quality is increasing order 'Worst' , 'Very Bad' , 'Good' , 'Very Good' , 'Best'.
Here, we convert these into Codes by using labelling encoding instead of one hot encoding to avoid high dimensionality because of OHE(get_dummies).

Label Encoding -

Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

Label Encoding of Cut -

```
Ideal        10805
Very Good     8462
Premium       6886
Fair           780
Name: cut, dtype: int64
```

Before Encoding

```
0      780
2     6886
1     8462
3    10805
Name: cut, dtype: int64
```

After Encoding

Label Encoding of Color -

```
Best         1440
Very Good    2765
Worst        3341
Bad          4723
Very Bad     4916
Good         9748
Name: color, dtype: int64
```

Before Encoding

```
5    1440
4    2765
0    3341
2    4723
1    4916
3    9748
Name: color, dtype: int64
```

After Encoding

Label Encoding of Clarity -

```
Best          364
Worst         891
Very Bad     4369
Good        10180
Very Good   11129
Name: clarity, dtype: int64
```

Before Encoding

```
4      364
0      891
1     4369
2    10180
3    11129
Name: clarity, dtype: int64
```

After Encoding

TAB:10 ENCODED VARIBLES TABLE

Checking the Dataset after Encoding

	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
1	0.33	2	3	0	60.8	58.0	4.42	4.46	2.70	984
2	0.90	1	1	1	62.2	60.0	6.04	6.12	3.78	6289
3	0.42	3	2	2	61.6	56.0	4.82	4.80	2.96	1082
4	0.31	3	2	1	60.4	59.0	4.35	4.43	2.65	779

TAB:11 RECORDS OF DATASET AFTER ENCODING

Linear Regression Model -

In statistics, linear regression is a linear approach for modelling the relationship between a scalar response and one or more explanatory variables (also known as dependent and independent variables). The case of one explanatory variable is called simple linear regression; for more than one, the process is called multiple linear regression.

[1] This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

[2]In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models.

.[3] Most commonly, the conditional mean of the response given the values of the explanatory variables (or predictors) is assumed to be an affine function of those values; less commonly, the conditional median or some other quantile is used. Like all forms of regression analysis, linear regression focuses on the conditional probability distribution of the response given the values of the predictors, rather than on the joint probability distribution of all of these variables, which is the domain of multivariate analysis. Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications.

[4] This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine.

Linear regression has many practical uses. Most applications fall into one of the following two broad categories:

- If the goal is prediction, forecasting, or error reduction,[clarification needed] linear regression can be used to fit a predictive model to an observed data set of values of the response and explanatory variables. After developing such a model, if additional values of the explanatory variables are collected without an accompanying response value, the fitted model can be used to make a prediction of the response.

- If the goal is to explain variation in the response variable that can be attributed to variation in the explanatory variables, linear regression analysis can be applied to quantify the strength of the relationship between the response and the explanatory variables, and in particular to determine whether some explanatory variables may have no linear relationship with the response at all, or to identify which subsets of explanatory variables may contain redundant information about the response.

- Linear regression models are often fitted using the least squares approach, but they may also be fitted in other ways, such as by minimizing the "lack of fit" in some other norm (as with least absolute deviations regression), or by minimizing a penalized version of the least squares cost function as in ridge regression (L2-norm penalty) and lasso (L1-norm penalty). Conversely, the least squares approach can be used to fit models that are not linear models. Thus, although the terms "least squares" and "linear model" are closely linked, they are not synonymous.

Note :
A linear regression model describes the relationship between a dependent variable, y, and one or more independent variables, X. The dependent variable is also called the response variable. ... Continuous predictor variables are also called covariates, and categorical predictor variables are also called factors.Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable.

Train-Test Split for Linear Regression Model -

The train-test split is a technique for evaluating the performance of a machine learning algorithm. It can be used for classification or regression problems and can be used for any supervised learning algorithm. The procedure involves taking a dataset and dividing it into two subsets.

In the given problem, we are advised to split the training and the testing data in the ratio of (70: 30).Here we are split the data into train and test part , like x_train , x_test , train_labels & test_labels ,by using train_test_split func() from sk-learn library here , we are taking 70 % data for training and 30 % data for testing.

Building A Linear Regression Model -

Let's start with building a linear model. Instead of simple linear regression, where you have one predictor and one outcome, we will go with multiple linear regression, where you have more than one predictors and one outcome.

Multiple linear regression follows the formula :

$$yi = \beta0 + \beta1xi1 + \beta2xi2 + ... + \beta pxip + \epsilon$$

Where:

- yi is the dependent or predicted variable
- β0 is the y-intercept, i.e., the value of y when both xi and x2 are 0.
- β1 and β2 are the regression coefficients representing the change in y relative to a one-unit change in xi1 and xi2, respectively.
- βp is the slope coefficient for each independent variable
- ϵ is the model's random error (residual) term.

The coefficients in this linear equation denote the magnitude of additive relation between the predictor and the response. In simpler words, keeping everything else fixed, a unit change in x1 will lead to change of β1 in the outcome, and so on.

Invoke the Linear Regression function (from sklearn.linear_model import LinearRegression) fit the function on the train & test data and build the linear regression model. In this problem we are advised to build various linear regression model and check the performance of Predictions on Train and Test sets using Rsquare, RMSE & Adj Rsquare & atlast we need to compare these models and select the best one.

Reference:

AS I made 4 models in the code - file . As instructed here I showcase the best model which selected on the basis of Train and Test sets using Rsquare, RMSE & Adj Rsquare. I also make comparison table for comparing 4 model at the bottom so that will give you clear understanding why choosing this model. I made model in multiple model in my code file by using sklearn and stats model ols technique *For codes please referred code file.

Linear Regression Model : 1 (with outlier treatment) - In this model the train and test data which I feed is having no outlier , outlier treatment of the dataset is done before the model building process (for codes please refer my code file). Lets check the results of Linear Regression Model 1 : (By Sklearn library & Statsmodel) .

Explore the coefficients for each of the independent attributes.

- The coefficient for carat is 13571.683740760513
- The coefficient for cut is 135.98605320137753
- The coefficient for color is -376.55070332051673
- The coefficient for clarity is -892.462765659155
- The coefficient for depth is -17.13901605079258
- The coefficient for table is -35.40418175538726
- The coefficient for x is -2459.0892736346445
- The coefficient for y is 1601.7602050587698
- The coefficient for z is -1721.6619421768203

The coefficients in this linear equation denote the magnitude of additive relation between the predictor and the response. In simpler words, keeping everything else fixed, a unit change in x1 will lead to change of β1 in the outcome, and so on.

Intercept for the model -

The regression constant is also known as the intercept thus, regression models without predictors are also known as intercept only models. ... As such, we will begin with intercept only models for OLS regression and then move on to logistic regression models without predictors. Intercept for the model is **9829.25572257.**

R square - 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. 100% indicates that the model explains all the variability of the response data around its mean.

R square on training data - 0.9107855334149219
R square on testing data - 0.9136525749360112

R-Squared value of 0.9 would indicate that 90% of the variance of the dependent variable being studied is explained by the variance of the independent variable.

RMSE - The root mean square error (RMSE) for a regression model is similar to the standard deviation (SD) for the ideal measurement model. The SD estimates the deviation from the sample mean x. The RMSE estimates the deviation of the actual y-values from the regression line.

Root mean square error, which is a metric that tells us the average distance between the predicted values from the model and the actual values in the dataset. The lower the RMSE, the better a given model is able to “fit” a dataset.

RMSE on Training data - 1197.576490998623
RMSE on Testing data - 1190.87764451318

Linear Regression using Stats-Model ols : 1

This is the expression of model 1 :
expr = 'price ~ carat + cut + color +clarity + depth + table + x + y + z'

Explore the coefficients for each of the independent attributes.

- The coefficient for carat is 13571.683740760513
- The coefficient for cut is 135.98605320137753
- The coefficient for color is -376.55070332051673
- The coefficient for clarity is -892.462765659155
- The coefficient for depth is -17.13901605079258
- The coefficient for table is -35.40418175538726
- The coefficient for x is -2459.0892736346445
- The coefficient for y is 1601.7602050587698
- The coefficient for z is -1721.6619421768203

The coefficients in this linear equation denote the magnitude of additive relation between the predictor and the response. In simpler words, keeping everything else fixed, a unit change in x1 will lead to change of β1 in the outcome, and so on.

Intercept for the model -

The regression constant is also known as the intercept thus, regression models without predictors are also known as intercept only models. ... As such, we will begin with intercept only models for OLS regression and then move on to logistic regression models without predictors. Intercept for the model is **9829.25572257.**

*****As Linear regression model with sklearn and stats model both showing the same results . That’s why we get the same values for intercept & coefficient.**

OLS Regression Results

OLS is a common technique used in analyzing linear regression. In brief, it compares the difference between individual points in your data set and the predicted best fit line to measure the amount of error produced. ... ols() function requires two inputs, the formula for producing the best fit line, and the dataset.

OLS Regression Results

Dep. Variable:	price	R-squared:	0.911
Model:	OLS	Adj. R-squared:	0.911
Method:	Least Squares	F-statistic:	2.137e+04
Date:	Mon, 04 Oct 2021	Prob (F-statistic):	0.00
Time:	00:34:33	Log-Likelihood:	-1.6038e+05
No. Observations:	18853	AIC:	3.208e+05
Df Residuals:	18843	BIC:	3.209e+05
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	9829.2557	1031.528	9.529	0.000	7807.367	1.19e+04
carat	1.357e+04	108.601	124.969	0.000	1.34e+04	1.38e+04
cut	135.9861	11.689	11.633	0.000	113.074	158.898
color	-376.5507	6.758	-55.718	0.000	-389.797	-363.304
clarity	-892.4628	11.346	-78.656	0.000	-914.703	-870.223
depth	-17.1390	14.382	-1.192	0.233	-45.328	11.050
table	-35.4042	5.149	-6.876	0.000	-45.496	-25.312
x	-2459.0893	162.909	-15.095	0.000	-2778.406	-2139.773
y	1601.7602	160.410	9.985	0.000	1287.342	1916.178
z	-1721.6619	180.188	-9.555	0.000	-2074.847	-1368.477

Omnibus:	3589.100	Durbin-Watson:	1.979
Prob(Omnibus):	0.000	Jarque-Bera (JB):	26762.413
Skew:	0.717	Prob(JB):	0.00
Kurtosis:	8.658	Cond. No.	1.01e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.01e+04. This might indicate that there are strong multicollinearity or other numerical problems.

TAB:12 SUMMARY OF LINEAR REGRESSION MODEL - 1

Insights :

We get the intercept & coefficient values from the summary .

- Looking at p value of depth we conclude that their is no relationship between depth & price (dependent variable) , so we can drop it from sample , will do further analysis.
- R-sqd value for this model is 0.911 which is good.
- Adj.- R-sqd value for this model is 0.911 which good.
- The Durbin Watson (DW) statistic is a test for autocorrelation in the residuals from a statistical model or regression analysis. The Durbin-Watson statistic will always have a value ranging between 0 and 4. A value of 2.0 indicates there is no autocorrelation detected in the sample , here Durbin Watson value is 1.979 this shows there is no autocorrelation detected in the sample.
- Here Kurtosis value found be 8.658 which indicates he dataset has heavier tails than a normal distribution (more in the tails).
- Here skew is 0.717 indicated data slightly right skewed.

- Prob(Omnibus) – a test of the skewness and kurtosis of the residual (characteristic #2). We hope to see a value close to zero which would indicate normalcy. The Prob (Omnibus) performs a statistical test indicating the probability that the residuals are normally distributed. Here omnibus value is 0.000 indicates normalcy.
- Standard Errors assume that the covariance matrix of the errors is correctly specified.
- The condition number is large, 1.01e+04. This might indicate that there are strong multicollinearity or other numerical problems , we can do treatment of multicollinearity for better results.

RMSE on Training data - 1197.576490998623
RMSE on Testing data - 1190.87764451318

Prediction on Test data -

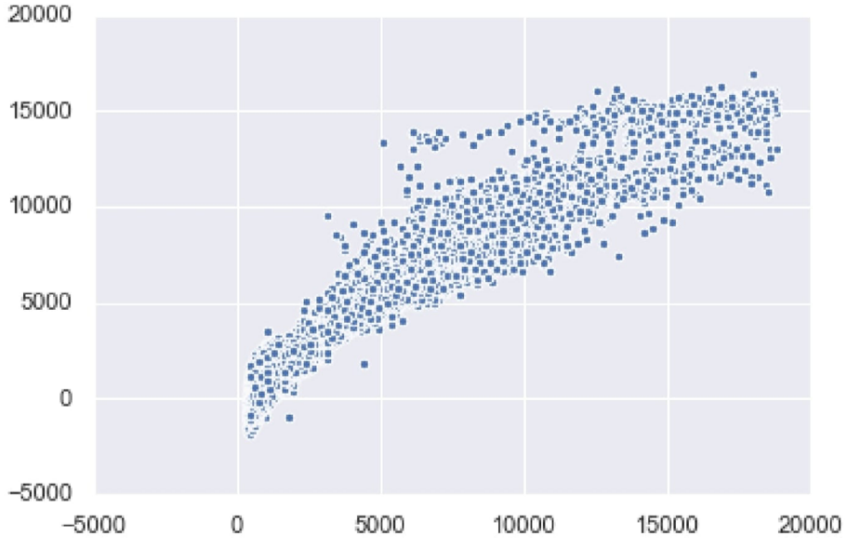


FIG: 26 Prediction on Test data

Scatter plots shows the distribution of actual y and predicted y.

Linear Regression Model in terms of Equation :
Y= c + m1x1 + m2x2 + m2X3————————mnxn
(9829.26) * Intercept + (13571.68) * carat + (135.99) * cut + (-376.55) * color + (-892.46) * clarity + (-17.14) * depth + (-35.4) * table + (-2459.09) * x + (1601.76) * y + (-1721.66) * z +

VIF - Variance inflation factor (VIF) is used to detect the severity of multicollinearity in the ordinary least square (OLS) regression analysis.

- carat ---> 121.0393389522025
- cut ---> 7.825817600127252
- color ---> 4.270949713420288
- clarity ---> 9.503246843639163
- depth ---> 1155.5475921943837
- table ---> 871.3738716144344
- x ---> 10646.829638778272
- y ---> 9357.38254505309
- z ---> 2998.8096429554753

Insights :

VIF value must be in between 1 to 5 , but we saw here all the VIF are very high which that their is a strong multicollinearity between the variables. If we want better performance so need to treat the the multicollinearity . As per instructions we are advised not treat that's why I did not treat it , but we can treat multicollinearity by various techniques like by Remove some of the highly correlated independent variables.Linearly combine the independent variables, such as adding them together.Perform an analysis designed for highly correlated variables, such as principal components analysis or partial least squares regression.

Linear Rregression Model : 2 (without outlier treatment)

Explore the coefficients for each of the independent attributes.

- The coefficient for carat is 10995.374364620247
- The coefficient for cut is 103.42154931763986
- The coefficient for color is -376.89482791221263
- The coefficient for clarity is -931.9613456294825
- The coefficient for depth is -94.13589656418183
- The coefficient for table is -38.02170910227073
- The coefficient for x is -982.7154555711769
- The coefficient for y is 3.8425597551378203
- The coefficient for z is -31.718499390264405

The coefficients in this linear equation denote the magnitude of additive relation between the predictor and the response. In simpler words, keeping everything else fixed, a unit change in x1 will lead to change of β1 in the outcome, and so on.

Intercept for the model -

The regression constant is also known as the intercept thus, regression models without predictors are also known as intercept only models. ... As such, we will begin with intercept only models for OLS regression and then move on to logistic regression models without predictors. Intercept for the model is **11601.82863906**

R square - 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. 100% indicates that the model explains all the variability of the response data around its mean.

R square on training data - 0.9028908224702554
R square on testing data - 0.9027027269979324

R-Squared value of 0.9 would indicate that 90% of the variance of the dependent variable being studied is explained by the variance of the independent variable.

RMSE - The root mean square error (RMSE) for a regression model is similar to the standard deviation (SD) for the ideal measurement model. The SD estimates the deviation from the sample mean x. The RMSE estimates the deviation of the actual y-values from the regression line.

Root mean square error, which is a metric that tells us the average distance between the predicted values from the model and the actual values in the dataset. The lower the RMSE, the better a given model is able to “fit” a dataset.

RMSE on Training data - 1249.4410151857392
RMSE on Testing data -1264.1330364514727

Linear Regression using Stats-Model ols :2

This is the expression of model 2 :
expr_1 = 'price ~ carat + cut + color +clarity + depth + table + x + y + z'

Explore the coefficients for each of the independent attributes.

- The coefficient for carat is 10995.374364620247
- The coefficient for cut is 103.42154931763986
- The coefficient for color is -376.89482791221263
- The coefficient for clarity is -931.9613456294825
- The coefficient for depth is -94.13589656418183
- The coefficient for table is -38.02170910227073
- The coefficient for x is -982.7154555711769
- The coefficient for y is 3.8425597551378203
- The coefficient for z is -31.718499390264405

The coefficients in this linear equation denote the magnitude of additive relation between the predictor and the response. In simpler words, keeping everything else fixed, a unit change in x1 will lead to change of $\beta 1$ in the outcome, and so on.

Intercept for the model -

The regression constant is also known as the intercept thus, regression models without predictors are also known as intercept only models. ... As such, we will begin with intercept only models for OLS regression and then move on to logistic regression models without predictors. Intercept for the model is **11601.82863906**

OLS Regression Results

OLS is a common technique used in analyzing linear regression. In brief, it compares the difference between individual points in your data set and the predicted best fit line to measure the amount of error produced. ... ols() function requires two inputs, the formula for producing the best fit line, and the dataset.

OLS Regression Results

Dep. Variable:	price	R-squared:	0.903
Model:	OLS	Adj. R-squared:	0.903
Method:	Least Squares	F-statistic:	1.947e+04
Date:	Mon, 04 Oct 2021	Prob (F-statistic):	0.00
Time:	01:44:27	Log-Likelihood:	-1.6118e+05
No. Observations:	18853	AIC:	3.224e+05
Df Residuals:	18843	BIC:	3.225e+05
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1.16e+04	716.803	16.186	0.000	1.02e+04	1.3e+04
carat	1.1e+04	96.195	114.303	0.000	1.08e+04	1.12e+04
cut	103.4215	12.103	8.545	0.000	79.699	127.144
color	-376.8948	7.054	-53.433	0.000	-390.720	-363.069
clarity	-931.9613	11.782	-79.104	0.000	-955.054	-908.869
depth	-94.1359	8.001	-11.766	0.000	-109.818	-78.454
table	-38.0217	5.127	-7.416	0.000	-48.071	-27.973
x	-982.7155	52.281	-18.797	0.000	-1085.190	-880.241
y	3.8426	24.510	0.157	0.875	-44.200	51.885
z	-31.7185	42.819	-0.741	0.459	-115.647	52.210

Omnibus:	3875.523	Durbin-Watson:	1.981
Prob(Omnibus):	0.000	Jarque-Bera (JB):	122643.226
Skew:	0.241	Prob(JB):	0.00
Kurtosis:	15.486	Cond. No.	6.70e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.7e+03. This might indicate that there are strong multicollinearity or other numerical problems.

TAB:13 SUMMARY OF LINEAR REGRESSION MODEL - 2

Insights :

We get the intercept & coefficient values from the summary .

- Looking at p value of y, z we conclude that their is no relationship between y , z & price (dependent variable) , so we can drop it from sample , will do further analysis.
- R-sqd value for this model is 0.903 which is good.
- Adj.- R-sqd value for this model is 0.903 which good.
- The Durbin Watson (DW) statistic is a test for autocorrelation in the residuals from a statistical model or regression analysis. The Durbin-Watson statistic will always have a value ranging between 0 and 4. A value of 2.0 indicates there is no autocorrelation detected in the sample , here Durbin Watson value is 1.981 this shows there is no autocorrelation detected in the sample.
- Here Kurtosis value found be 15.486 which indicates he dataset has heavier tails than a normal distribution (more in the tails).
- Here skew is 0.241 indicated data very slightly right skewed.

- Prob(Omnibus) – a test of the skewness and kurtosis of the residual (characteristic #2). We hope to see a value close to zero which would indicate normalcy. The Prob (Omnibus) performs a statistical test indicating the probability that the residuals are normally distributed. Here omnibus value is 0.000 indicates normalcy.
- Standard Errors assume that the covariance matrix of the errors is correctly specified.
- The condition number is large, 6.70e+03. This might indicate that there are strong multicollinearity or other numerical problems , we can do treatment of multicollinearity for better results.

RMSE on Training data - 1249.4410151857392
RMSE on Testing data -1264.1330364514727

Prediction on Test data -

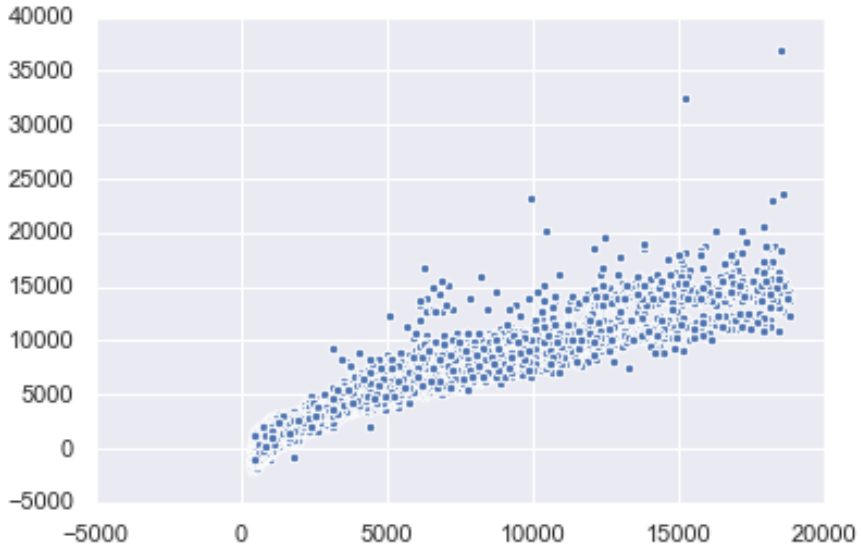


FIG: 27 PREDICTION ON TEST DATA MODEL-2

Scatter plots shows the distribution of actual y and predicted y.

Linear Regression Model in terms of Equation :
Y= c + m1x1 + m2x2 + m2X3————————mnxn
(11601.83) * Intercept + (10995.37) * carat + (103.42) * cut + (-376.89) * color + (-931.96) * clarity + (-94.14) * depth + (-38.02) * table + (-982.72) * x + (3.84) * y + (-31.72) * z +

VIF - Variance inflation factor (VIF) is used to detect the severity of multicollinearity in the ordinary least square (OLS) regression analysis.

- carat ---> 81.5033418559768
- cut ---> 6.860729682922422
- color ---> 4.27389351773964
- clarity ---> 9.369000024698023
- depth ---> 558.577146731718
- table ---> 556.1039170515155
- x ---> 1133.1266837484013
- y ---> 347.88572635041317
- z ---> 382.0311260961392

Insights :

VIF value must be in between 1 to 5 , but we saw here all the VIF are very high which that their is a strong multicollinearity between the variables. If we want better performance so need to treat the the multicollinearity . As per instructions we are advised not treat that's why I did not treat it , but we can treat multicollinearity by various techniques like by Remove some of the highly correlated independent variables.Linearly combine the independent variables, such as adding them together.Perform an analysis designed for highly correlated variables, such as principal components analysis or partial least squares regression.

Linear Rregression Model : 3 (Z-Score)

Explore the coefficients for each of the independent attributes.

- The coefficient for carat is 1.3020432866410232
- The coefficient for cut is 0.023419132324343093
- The coefficient for color is -0.12752572043584115
- The coefficient for clarity is -0.1966930919244331
- The coefficient for depth is -0.03245062536483056
- The coefficient for table is -0.02116667988395818
- The coefficient for x is -0.2750666364781017
- The coefficient for y is 0.001130399870640189
- The coefficient for z is -0.005727300197063211

The coefficients in this linear equation denote the magnitude of additive relation between the predictor and the response. In simpler words, keeping everything else fixed, a unit change in x1 will lead to change of β1 in the outcome, and so on.

Intercept for the model -

The regression constant is also known as the intercept thus, regression models without predictors are also known as intercept only models. ... As such, we will begin with intercept only models for OLS regression and then move on to logistic regression models without predictors. Intercept for the model is **-8.306971752742142e-17 , as we apply scaling intercept become nearly equal to 0.**

R square - 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. 100% indicates that the model explains all the variability of the response data around its mean.

R square on training data - 0.9028908224702554

R square on testing data - 0.9026286235959851

R-Squared value of 0.9 would indicate that 90% of the variance of the dependent variable being studied is explained by the variance of the independent variable.

RMSE - The root mean square error (RMSE) for a regression model is similar to the standard deviation (SD) for the ideal measurement model. The SD estimates the deviation from the sample mean x. The RMSE estimates the deviation of the actual y-values from the regression line.

Root mean square error, which is a metric that tells us the average distance between the predicted values from the model and the actual values in the dataset. The lower the RMSE, the better a given model is able to “fit” a dataset.

RMSE on Training data - 1249.4410151857392

RMSE on Testing data - 1256.332600024064

Linear Regression using Stats-Model ols :3

This is the expression of model 3 :
expr_2 = 'price ~ carat + cut + color +clarity + depth + table + x + y + z'

Explore the coefficients for each of the independent attributes.

- The coefficient for carat is 1.3020432866410232
- The coefficient for cut is 0.023419132324343093
- The coefficient for color is -0.12752572043584115
- The coefficient for clarity is -0.1966930919244331
- The coefficient for depth is -0.03245062536483056
- The coefficient for table is -0.02116667988395818
- The coefficient for x is -0.2750666364781017
- The coefficient for y is 0.001130399870640189
- The coefficient for z is -0.005727300197063211

The coefficients in this linear equation denote the magnitude of additive relation between the predictor and the response. In simpler words, keeping everything else fixed, a unit change in x1 will lead to change of β1 in the outcome, and so on.

Intercept for the model -

The regression constant is also known as the intercept thus, regression models without predictors are also known as intercept only models. ... As such, we will begin with intercept only models for OLS regression and then move on to logistic regression models without predictors. Intercept for the model is **-8.306971752742142e-17 , as we apply scaling intercept become nearly equal to 0.**

OLS Regression Results

OLS is a common technique used in analyzing linear regression. In brief, it compares the difference between individual points in your data set and the predicted best fit line to measure the amount of error produced. ... ols() function requires two inputs, the formula for producing the best fit line, and the dataset.

OLS Regression Results

Dep. Variable:	price	R-squared:	0.903
Model:	OLS	Adj. R-squared:	0.903
Method:	Least Squares	F-statistic:	1.947e+04
Date:	Mon, 04 Oct 2021	Prob (F-statistic):	0.00
Time:	03:47:53	Log-Likelihood:	-4769.4
No. Observations:	18853	AIC:	9559.
Df Residuals:	18843	BIC:	9637.
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-5.855e-18	0.002	-2.58e-15	1.000	-0.004	0.004
carat	1.3020	0.011	114.303	0.000	1.280	1.324
cut	0.0234	0.003	8.545	0.000	0.018	0.029
color	-0.1275	0.002	-53.433	0.000	-0.132	-0.123
clarity	-0.1967	0.002	-79.104	0.000	-0.202	-0.192
depth	-0.0325	0.003	-11.766	0.000	-0.038	-0.027
table	-0.0212	0.003	-7.416	0.000	-0.027	-0.016
x	-0.2751	0.015	-18.797	0.000	-0.304	-0.246
y	0.0011	0.007	0.157	0.875	-0.013	0.015
z	-0.0057	0.008	-0.741	0.459	-0.021	0.009

Omnibus:	3875.523	Durbin-Watson:	1.981
Prob(Omnibus):	0.000	Jarque-Bera (JB):	122643.226
Skew:	0.241	Prob(JB):	0.00
Kurtosis:	15.486	Cond. No.	15.9

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

TAB:14 SUMMARY OF LINEAR REGRESSION MODEL - 3

Insights :

We get the intercept & coefficient values from the summary .

- Looking at p value of y, z we conclude that their is no relationship between y , z & price (dependent variable) , so we can drop it from sample , will do further analysis.
- R-sqd value for this model is 0.903 which is good.
- Adj.- R-sqd value for this model is 0.903 which good.
- The Durbin Watson (DW) statistic is a test for autocorrelation in the residuals from a statistical model or regression analysis. The Durbin-Watson statistic will always have a value ranging between 0 and 4. A value of 2.0 indicates there is no autocorrelation detected in the sample , here Durbin Watson value is 1.981 this shows there is no autocorrelation detected in the sample.
- Here Kurtosis value found be 15.486 which indicates he dataset has heavier tails than a normal distribution (more in the tails).
- Here skew is 0.241 indicated data very slightly right skewed.

- Prob(Omnibus) – a test of the skewness and kurtosis of the residual (characteristic #2). We hope to see a value close to zero which would indicate normalcy. The Prob (Omnibus) performs a statistical test indicating the probability that the residuals are normally distributed. Here omnibus value is 0.000 indicates normalcy.
- Standard Errors assume that the covariance matrix of the errors is correctly specified.
- The condition number is large, 15.7.

RMSE on Training data - 1249.4410151857392

RMSE on Testing data -1256.332600024064

Prediction on Test data -

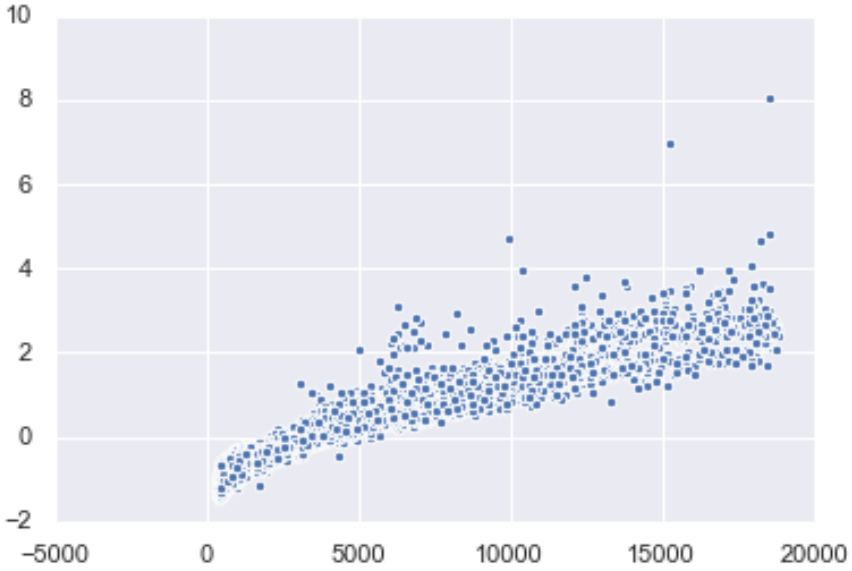


FIG: 28 PREDICTION ON TEST DATA MODEL-3

Scatter plots shows the distribution of actual y and predicted y.

VIF - Variance inflation factor (VIF) is used to detect the severity of multicollinearity in the ordinary least square (OLS) regression analysis.

- carat ---> 81.5033418559768
- cut ---> 6.860729682922422
- color ---> 4.27389351773964
- clarity ---> 9.369000024698023
- depth ---> 558.577146731718
- table ---> 556.1039170515155
- x ---> 1133.1266837484013
- y ---> 347.88572635041317
- z ---> 382.0311260961392

Insights :

VIF value must be in between 1 to 5 , but we saw here all the VIF are very high which that their is a strong multicollinearity between the variables. If we want better performance so need to treat the the multicollinearity . As per instructions we are advised not treat that's why I did not treat it , but we can treat multicollinearity by various techniques like by Remove some of the highly correlated independent variables.Linearly combine the independent variables, such as adding them together.Perform an analysis designed for highly correlated variables, such as principal components analysis or partial least squares regression.

Linear Regression Model 4 (Applied Feature Engineering)

Explore the coefficients for each of the independent attributes.

- The coefficient for carat is 8683.924280963514
- The coefficient for cut is 105.90184220433494
- The coefficient for color is -372.323745419193
- The coefficient for clarity is -977.1567159616646
- The coefficient for depth is -51.699714493156755
- The coefficient for table is -37.6839687737022
- The coefficient for area is 0.13193675940098695

The coefficients in this linear equation denote the magnitude of additive relation between the predictor and the response. In simpler words, keeping everything else fixed, a unit change in x1 will lead to change of β1 in the outcome, and so on.

Intercept for the model -

The regression constant is also known as the intercept thus, regression models without predictors are also known as intercept only models. ... As such, we will begin with intercept only models for OLS regression and then move on to logistic regression models without predictors. Intercept for the model is **- 5151.29267584**

R square - 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. 100% indicates that the model explains all the variability of the response data around its mean.

R square on training data - 0.8998127112974966

R square on testing data - 0.9003498265899168

R-Squared value of 0.9 would indicate that around 90% of the variance of the dependent variable being studied is explained by the variance of the independent variable.

RMSE - The root mean square error (RMSE) for a regression model is similar to the standard deviation (SD) for the ideal measurement model. The SD estimates the deviation from the sample mean x. The RMSE estimates the deviation of the actual y-values from the regression line.

Root mean square error, which is a metric that tells us the average distance between the predicted values from the model and the actual values in the dataset. The lower the RMSE, the better a given model is able to “fit” a dataset.

RMSE on Training data - 1269.0885689301178

RMSE on Testing data - 1279.3267371088534

Linear Regression using Stats-Model ols : 4

This is the expression of model : 4
expr_3 = 'price ~ carat + cut + color +clarity + depth + table + 'area'

Explore the coefficients for each of the independent attributes.

- The coefficient for carat is 8683.924280963514
- The coefficient for cut is 105.90184220433494
- The coefficient for color is -372.323745419193
- The coefficient for clarity is -977.1567159616646
- The coefficient for depth is -51.699714493156755
- The coefficient for table is -37.6839687737022
- The coefficient for area is 0.13193675940098695

The coefficients in this linear equation denote the magnitude of additive relation between the predictor and the response. In simpler words, keeping everything else fixed, a unit change in x1 will lead to change of β1 in the outcome, and so on.

Intercept for the model -

The regression constant is also known as the intercept thus, regression models without predictors are also known as intercept only models. ... As such, we will begin with intercept only models for OLS regression and then move on to logistic regression models without predictors. Intercept for the model is - **5151.29267584**

OLS Regression Results

OLS is a common technique used in analyzing linear regression. In brief, it compares the difference between individual points in your data set and the predicted best fit line to measure the amount of error produced. ... ols() function requires two inputs, the formula for producing the best fit line, and the dataset.

OLS Regression Results

Dep. Variable:	price	R-squared:	0.900
Model:	OLS	Adj. R-squared:	0.900
Method:	Least Squares	F-statistic:	2.418e+04
Date:	Mon, 04 Oct 2021	Prob (F-statistic):	0.00
Time:	04:38:06	Log-Likelihood:	-1.6148e+05
No. Observations:	18853	AIC:	3.230e+05
Df Residuals:	18845	BIC:	3.230e+05
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	5151.2927	661.409	7.788	0.000	3854.871	6447.714
carat	8683.9243	61.135	142.045	0.000	8564.094	8803.754
cut	105.9018	12.284	8.621	0.000	81.824	129.979
color	-372.3237	7.161	-51.990	0.000	-386.361	-358.287
clarity	-977.1567	11.817	-82.693	0.000	-1000.318	-953.995
depth	-51.6997	7.536	-6.860	0.000	-66.472	-36.928
table	-37.6840	5.208	-7.236	0.000	-47.891	-27.476
area	0.1319	0.351	0.376	0.707	-0.556	0.819

Omnibus:	3913.195	Durbin-Watson:	1.975
Prob(Omnibus):	0.000	Jarque-Bera (JB):	45759.450
Skew:	0.662	Prob(JB):	0.00
Kurtosis:	10.517	Cond. No.	1.22e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.22e+04. This might indicate that there are strong multicollinearity or other numerical problems.

TAB:15 SUMMARY OF LINEAR REGRESSION MODEL - 4

Insights :

We get the intercept & coefficient values from the summary .

- Looking at p value of area we conclude that their is no relationship between area & price (dependent variable) , so we can drop it from sample , will do further analysis.
- R-sqd value for this model is 0.90 which is good.
- Adj.- R-sqd value for this model is 0.90 which good.
- The Durbin Watson (DW) statistic is a test for autocorrelation in the residuals from a statistical model or regression analysis. The Durbin-Watson statistic will always have a value ranging between 0 and 4. A value of 2.0 indicates there is no autocorrelation detected in the sample , here Durbin Watson value is 1.975 this shows there is no autocorrelation detected in the sample.
- Here Kurtosis value found be 10.517 which indicates he dataset has heavier tails than a normal distribution (more in the tails).
- Here skew is 0.662 indicated data very slightly right skewed.
- Prob(Omnibus) – a test of the skewness and kurtosis of the residual (characteristic #2). We hope to see a value close to zero which would indicate normalcy. The Prob (Omnibus) performs a statistical test indicating the probability that the residuals are normally distributed. Here omnibus value is 0.000 indicates normalcy.
- Standard Errors assume that the covariance matrix of the errors is correctly specified.
- The condition number is large --- 1.22e+04.

RMSE on Training data - 1269.0885689301178

RMSE on Testing data - 1279.3267371088534

Prediction on Test data -

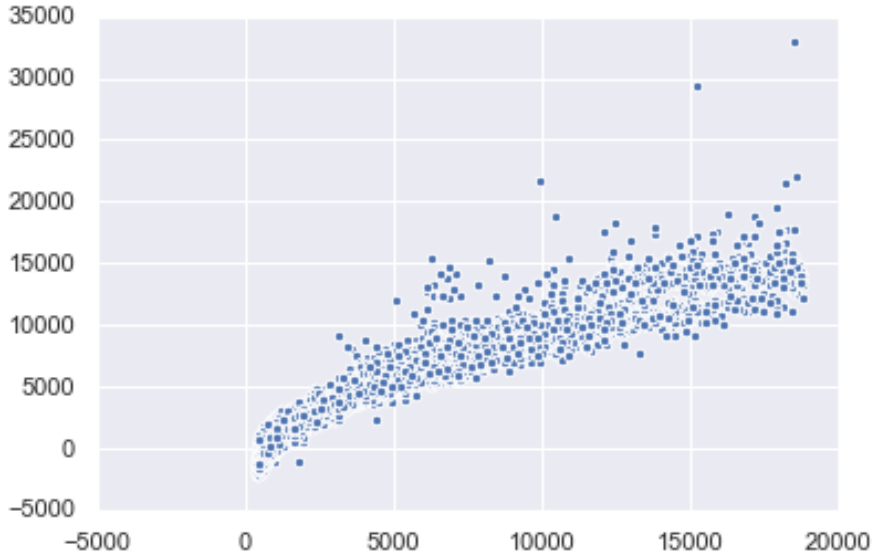


FIG: 29 PREDICTION ON TEST DATA MODEL-4

Scatter plots shows the distribution of actual y and predicted y.

Linear Regression Model in terms of Equation :
Y= c + m1x1 + m2x2 + m2X3————————mnxn
(5151.29) * Intercept + (8683.92) * carat + (105.9) * cut + (-372.32) * color + (-977.16) * clarity + (-51.7) * depth + (-37.68) * table + (0.13) * area +

VIF - Variance inflation factor (VIF) is used to detect the severity of multicollinearity in the ordinary least square (OLS) regression analysis.

- carat ---> 50.8593205612385
- cut ---> 6.38981578579258
- color ---> 4.271191049858925
- clarity ---> 9.187574857372661
- depth ---> 454.0528564411028
- table ---> 427.92455431733987
- area ---> 47.34501022665412

Insights :

VIF value must be in between 1 to 5 , but we saw here all the VIF are very high which that their is a strong multicollinearity between the variables. If we want better performance so need to treat the the multicollinearity . As per instructions we are advised not treat that's why I did not treat it , but we can treat multicollinearity by various techniques like by Remove some of the highly correlated independent variables.Linearly combine the independent variables, such as adding them together.Perform an analysis designed for highly correlated variables, such as principal components analysis or partial least squares regression.

Comparison Among the Various Linear Regression Model :

Here we are comparing the 4 Linear Regression Models & check the performance of Predictions on Train and Test sets using Rsquare, RMSE & Adj Rsquare.

Training Data

Factors of Comparison	Linear Regression Model 1 (with outlier treatment) Train Data	Linear Regression Model 2 (without outlier treatment)Train Data	Linear Regression Model 3 (with z score apply)Train Data	Linear Regression Model 4 (with feature engineering) Train Data
R square	0.911	0.903	0.903	0.899
Adj. R square	0.911	0.903	0.903	0.90
RMSE	1197.576	1249.441	1248.441	1269.088

Test Data

Factors of Comparison	Linear Regression Model 1 (with outlier treatment) Test Data	Linear Regression Model 2 (without outlier treatment)Test Data	Linear Regression Model 3 (with z score apply)Test Data	Linear Regression Model 4 (with feature engineering) Test Data
R square	0.9136	0.9027	0.9026	0.900
Adj. R square	0.911	0.903	0.903	0.90
RMSE	1190.877	1264.1330	1256.3326	1279.326

TAB:16 COMPARISON TABLE OF LINEAR REGRESSION MODEL ON TRAIN & TEST FOR R.SQUARE ,ADJ.R.SQUARE & RMSE

Results :

From the above table we clearly infer that Linear Regression Model 1 have good Rsquare, RMSE & Adj Rsquare values on the basis of Train and Test sets. That's why we we choose **linear regression model 1** because it have least RMSE among all & have max R.square & adj.R.square on train & test set with good accuracy score. Model is neither overfit nor underfit.

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

Insights :

Linear Regression Model 1 in terms of Equation :

Y= c + m1x1 + m2x2 + m2X3————————mnxn
(9829.26) * Intercept + (13571.68) * carat + (135.99) * cut + (-376.55) * color + (-892.46) * clarity + (-17.14) * depth + (-35.4) * table + (-2459.09) * x + (1601.76) * y + (-1721.66) * z +

When carat increases by 1 unit, price increases by 13571.68 units, keeping all other predictors constant.
There are also some negative co-efficient values, for cut , has its corresponding co-efficient as -376.55 .

Comparing the Model Scores from the Linear Regression models , LM1 sees to be an optimum model. From the Analysis, we can get a clear picture that on basis of the cut,Ideal cut had a significant number of turnover to the company.The Colours G,E, F,H are also comparatively superior over the other colours & have provided profits.We could see that the dimensional features had strong positive correlation among themselves & with the price feature. From our result,the computed R-Square is 91.07% for the model which means that 91.07% of the variance of the target variable price is explained by the predictors (independent variables) in the training data set. Also,for better accuracies we have dropped the columns depth & depth as the had p values more than 0.05 for better results. The final Linear Regression equation is as follows:
Carat is the highest co-efficient in the predicting the price, change in 1 unit of the carat will have an impact on the price 13571 times, henceforth it is the most important deciding factor for the price of the diamond. The Top Attributes reasonable & influencing the price are : Carat, ,Cut , Color & Clarity.

Recommendations:

The company should focus on key strengths & develop marketing strategies to promote diamond's carat , cut and clarity & also be competitive about their prices. More Carat & clear the stone the profits are more. The cuts of the diamond are also a significant factor in the prices .Company should focus on the carat , cut ,color and clarity for the zirconia stone in order to make more profits.Cubic zirconia manufacturer should manufacture zirconia with the following specifications for higher price range and higher profits:
Length > =6 mm Width >= 6 mm Height > =4 mm Carat weight> 0.8 Clarity either I1, IF, SI1, SI2, VS2 or VVS2.

Problem Statement 2: Logistic Regression and LDA (Linear discriminant analysis).

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.

Executive Summary

A tour and travel agency which deals in selling holiday packages.. The dataset consists of details of 872 employees of a company. Based on the different attributes / characteristics of these employees we are helping the company in predicting whether an employee will opt for the holiday package or not on the basis of the information given in the data set. Also help company to analyse important factors on the basis of which the company will focus on particular employees to sell their packages.

Introduction

The purpose of this whole exercise is to explore the dataset. Do the exploratory data analysis & apply various supervised learning algorithms i.e. Logistic Regression and LDA (Linear discriminant analysis) to predicting whether an employee will opt for the package or not. Explore the dataset using central tendency and other parameters. The data consists of 872 different employees with 7 unique activities . Analyse the different attributes of the employees which can help in predicting whether an employee will opt for the holiday package or not.. This assignment should help the tour and travel agency to distinguish between higher profitable and lower profitable employees so as to have better profit share.In exploring the summary statistics, Logistic Regression and LDA model will help company to predict the whether an employee will opt for the holiday package or not.

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.

EDA - Data Description , Data Preprocessing , Data Visualization -

Checking the Records of the Dataset.

Head of the Dataset - First 10 Records of the Dataset.

Tail of the Dataset - Last 10 Records of the Dataset.

	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
5	6	yes	61590	42	12	0	1	no
6	7	no	94344	51	8	0	0	no
7	8	yes	35987	32	8	0	2	no
8	9	no	41140	39	12	0	0	no
9	10	no	35826	43	11	0	2	no

	Unnamed: 0	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
862	863	no	66900	35	10	1	1	yes
863	864	no	35290	51	9	0	1	yes
864	865	no	25527	41	5	1	0	yes
865	866	yes	44057	35	9	0	2	yes
866	867	yes	22643	42	14	0	0	yes
867	868	no	40030	24	4	2	1	yes
868	869	yes	32137	48	8	0	0	yes
869	870	no	25178	24	6	2	0	yes
870	871	yes	55958	41	10	0	1	yes
871	872	no	74659	51	10	0	0	yes

TAB:1 RECORDS OF THE DATASET HEAD & TAIL

Dropping the Unnamed: 0 Column.

We are going to drop the column unnamed:0 as it is useless for the model & checking the dataset again.

	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
0	no	48412	30	8	1	1	no
1	yes	37207	45	8	0	1	no
2	no	58022	46	9	0	0	no
3	no	66503	31	11	2	0	no
4	no	66734	44	12	0	2	no
5	yes	61590	42	12	0	1	no
6	no	94344	51	8	0	0	no
7	yes	35987	32	8	0	2	no
8	no	41140	39	12	0	0	no
9	no	35826	43	11	0	2	no

TAB:2 RECORDS OF THE DATASET WITH FINALISED COLUMNS

Insights:

Now we have all the columns which are useful for the model.

Data Dictionary for Problem Statement 2.

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

TAB:3 DATA DICTIONARY OF THE DATASET

Checking the Summary of the Dataset.

	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

TAB:4 SUMMARY OF THE DATASET

Insights:

From the above table we can infer the count,mean, std , 25% , 50% ,75% and min & max values of the all numeric variables present in the dataset.

From the above table we can infer the count,unique,top,freq of all the categorical variables present in the dataset.

Checking the Shape of the DataFrame

Shape attribute tells us number of observations and variables we have in the data set. It is used to check the dimension of data. The Holiday_Package.csv data set has 872 observations (rows) and 7 variables (columns) in the dataset.

Number of Rows :872
Number of Columns :7

Checking the Appropriateness of Datatypes & Information of the DataFrame

```
<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
```

TAB:5 APPROPRIATENESS OF DATATYPES & INFORMATION OF THE DATAFRAME

Insights :

From the above results we can see that there is no null values present in the dataset.Their are total 872 rows & 7 columns in this dataset, indexed from 0 to 871. Out of 7 variables 5 are int64 , 2 variables are object. Memory used by the dataset: 47.8+ KB.

Checking for Null Values.

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

TAB:6 NULL VALUES.

Insights:

From the above output we infer that only their is no null values in the dataset.

Checking for Anomalies in the Dataset

Holiday_Package -

array(['no', 'yes'], d-type=object)

Salary -

array([48412, 37207, 58022, 66503, 66734, 61590, 94344, 35987,
41140, 35826, 42643, 35157, 75327, 148221, 98870, 80297,
52117, 139253, 62858, 57400, 52059, 66711, 51463, 35682,
35741, 40183, 37821, 74927, 51077, 38804, 129262, 42920,
41136, 37690, 68572, 30192, 31973, 99423, 70022, 34646,
66588, 42497, 36998, 44331, 15818, 55391, 76831, 23183,
40991, 41967, 33523, 35136, 35533, 34045, 33768, 35305,
68770, 30299, 29901, 52268, 35089, 27554, 165895, 90324,
36855, 36072, 45187, 83859, 49748, 30225, 65950, 35344,
39663, 107442, 40904, 36959, 50291, 54718, 31684, 15462,
25773, 29139, 38991, 47265, 37831, 42386, 51652, 29308,
84031, 75798, 35302, 48398, 40133, 29166, 51060, 63102,
42334, 30899, 44721, 31201, 61159, 40373, 42172, 32432,
46220, 37838, 64928, 47624, 216630, 47972, 43031, 48282,
78418, 84682, 50309, 60713, 14395, 37370, 46234, 46856,
50552, 35982, 31678, 49647, 47254, 44363, 34424, 34543,
42760, 55589, 52336, 90827, 50003, 57723, 10551, 46195,
83203, 71377, 68302, 42895, 65273, 38927, 62091, 40967,
52641, 42820, 142183, 41740, 42835, 56241, 67218, 32616,
55055, 37444, 62935, 57618, 48177, 53807, 41667, 49225,
53162, 162024, 74327, 36681, 34878, 47286, 46900, 43887,
63934, 44055, 40334, 32900, 40855, 50699, 22611, 46454,
55994, 51034, 45014, 40635, 64656, 68051, 37156, 42612,
124627, 55269, 36832, 64181, 40913, 38641, 54486, 57010,
36538, 62810, 39111, 72621, 58939, 83693, 51634, 33010,
59165, 52277, 37169, 50058, 25909, 73040, 75032, 53846,
41828, 51546, 42639, 84882, 76304, 70236, 54823, 36607,
58451, 31844, 38837, 34386, 36997, 44394, 40001, 72947,
56374, 38482, 46847, 35056, 57208, 49052, 17580, 62509,
56915, 22366, 98668, 40406, 42078, 36813, 75120, 44302,
52736, 36976, 32826, 67573, 25852, 72394, 39355, 49382,
17350, 43256, 43855, 26516, 117276, 48778, 58227, 38424,
44552, 33814, 63019, 51621, 35700, 48458, 37638, 25118,
33300, 38185, 43426, 51306, 30539, 46951, 43790, 44153,
34469, 60491, 44544, 54915, 27895, 128276, 35780, 69406,
69808, 42004, 47944, 59062, 41298, 55065, 44476, 57098,
32287, 44280, 63053, 42482, 38352, 119644, 96072, 115431,
44207, 50995, 48786, 32197, 53444, 86113, 68325, 54169,
70279, 23345, 45979, 16001, 69565, 1322, 47233, 41856,
50066, 120228, 52419, 108836, 84673, 34206, 48376, 44070,
41756, 32573, 65782, 21000, 53164, 56699, 45393, 47956,
33222, 30685, 62232, 33789, 20000, 33357, 104085, 45565,
45248, 41582, 40454, 54076, 38725, 60409, 57888, 65834,
56933, 69011, 57269, 22018, 53282, 45523, 32191, 36967,
57561, 60930, 41382, 53100, 38420, 52753, 58369, 74137,
38774, 46960, 27394, 47658, 42020, 40524, 36210, 79732,
44962, 64877, 47361, 131576, 89759, 39888, 38800, 69827,
41962, 57694, 57073, 60265, 36449, 37144, 34330, 31693,
52488, 44728, 57523, 38018, 41585, 43254, 40160, 28602,
40889, 23964, 75562, 40726, 53738, 59108, 54246, 59692,
39099, 37448, 46666, 29846, 64330, 56007, 47381, 49420,
68190, 25327, 34412, 36211, 36409, 49673, 49993, 57829,
56432, 44293, 38478, 52566, 35825, 42260, 46223, 41528,
39218, 80764, 35600, 34654, 35306, 17047, 41769, 40998,
51165, 77421, 28158, 57197, 59050, 35582, 40794, 35039,
35678, 44023, 24608, 47827, 36729, 54626, 78447, 81307,
84482, 38683, 47038, 47995, 48018, 36946, 90272, 37726,
33071, 55506, 38145, 50801, 33954, 76446, 44994, 64239,
128381, 38598, 44653, 46241, 35093, 51573, 36596, 27040,
45385, 44334, 56148, 208561, 51070, 49487, 78892, 42594,
37069, 37795, 67611, 24573, 36115, 44861, 34420, 51666,
49302, 41282, 36308, 45419, 59852, 28596, 19266, 39346,
38196, 41267, 35070, 35095, 41397, 34017, 41167, 43502,
53759, 27745, 56874, 53003, 37094, 25434, 39765, 36016,
14119, 46115, 41165, 45698, 84592, 53189, 46110, 41421,
36800, 55880, 74580, 44580, 43204, 62674, 50640, 47175,
45436, 41098, 36099, 34644, 24186, 94490, 51543, 49686,
47249, 61441, 23792, 42626, 49080, 40090, 34521, 84656,
115842, 20682, 63833, 37733, 43940, 42443, 12070, 42188,
37293, 27909, 27598, 62777, 81148, 38343, 28573, 87897,
46063, 53546, 135489, 21909, 46846, 34286, 22244, 39460,
49582, 57464, 20869, 36267, 39963, 37266, 48209, 67248,
69658, 18486, 19845, 42425, 17396, 19514, 68528, 32910,
46595, 30735, 50202, 30378, 42448, 42508, 51337, 64744,
34649, 42369, 42042, 79280, 52631, 28656, 64029, 36949,
71170, 40005, 132984, 38268, 48737, 25324, 51832, 57662,
46988, 14108, 36522, 19072, 48578, 42311, 37797, 84103,
74925, 66780, 52386, 72164, 26696, 236961, 35330, 215675,
37685, 54136, 42562, 34260, 33112, 44098, 19656, 35575,
64345, 34731, 42505, 34632, 33126, 28058, 39904, 128548,
50428, 28446, 44520, 55009, 37152, 28892, 27624, 28736,
60894, 40453, 44872, 50598, 47077, 39807, 40784, 32577,
48800, 32225, 38741, 40544, 43664, 41229, 33178, 37947,
75333, 36450, 44193, 33574, 39857, 37462, 39593, 34463,
41496, 40909, 49756, 41510, 36146, 43953, 32509, 36551,
49317, 40363, 34187, 33608, 34096, 41395, 56695, 54335,
44630, 38874, 28425, 39256, 38663, 54055, 26415, 32584,
38226, 39097, 43839, 35841, 24543, 41705, 31838, 43417,
40095, 70629, 40710, 44595, 33963, 30383, 39116, 16939,
40124, 29067, 35341, 31241, 38533, 39648, 31681, 33591,
38624, 107741, 34923, 109150, 35045, 36369, 36564, 41290,
40549, 40646, 44223, 44819, 34100, 38880, 38489, 32175,
35458, 40270, 35989, 40138, 37665, 50584, 34947, 36006,
34357, 37458, 77241, 41951, 68467, 70513, 46544, 33662,
31517, 38053, 26045, 27789, 12578, 37274, 26088, 32977,
32072, 72963, 32455, 39821, 38332, 39809, 33060, 43484,
10741, 37418, 37215, 42008, 40552, 80233, 36839, 31701,
43398, 43333, 34610, 36095, 56036, 27459, 29119, 32112,
38105, 34814, 36598, 35386, 30513, 32828, 30942, 37155,
24690, 37129, 33336, 36023, 26980, 35076, 13993, 33544,
60047, 34407, 30215, 32891, 16342, 35646, 24582, 42465,
37324, 28271, 46300, 34205, 33578, 38974, 44146, 31715,
30575, 26317, 27630, 33448, 95460, 31662, 33274, 33434,
35726, 34608, 26564, 37853, 46488, 34293, 29968, 36679,
29809, 36595, 42741, 42478, 38344, 36345, 30253, 45641,
22257, 35416, 85549, 35730, 46747, 38536, 27332, 106302,
49318, 56718, 32206, 47837, 37380, 35191, 66900, 35290,
25527, 44057, 22643, 40030, 32137, 25178, 55958, 74659])

Age -

```
array([30, 45, 46, 31, 44, 42, 51, 32, 39, 43, 60, 33, 56, 47, 50, 53, 29,
       20, 54, 28, 49, 55, 24, 36, 23, 41, 38, 37, 22, 34, 58, 59, 35, 61,
       40, 52, 21, 48, 62, 27, 57, 26, 25])
```

Educ -

```
array([ 8,  9, 11, 12, 14, 19, 10, 13, 15,  4, 17,  7, 16,  6,  5, 21,  2,
       18,  3,  1])
```

no_young_children -

```
array([1, 0, 2, 3])
```

no_older_children -

```
array([1, 0, 2, 4, 3, 5, 6])
```

foreign -

```
array(['no', 'yes'], dtype=object)
```

TAB:7 CHECKING FOR ANOMALIES FOR VARIABLES IN THE DATASET

Observations

No Anomalies found in the Dataset.

Checking the Value counts on all the Categorical Column

Holiday_Package -

- As per the given Data Dictionary there are 2 label in Holliday_Package variable yes and no.
- 471 employees opted "no" for the Holiday Package.
- 401 employees opted "yes" for the Holiday Package.

foreign -

- As per the given Data Dictionary there are 2 label in foreign variable yes and no.
- 656 employees are not foreigner.
- 216 employees are foreigner.

Observation

There is no missing value & bad value present in the above categorical variables.

Checking Duplicate Values -

Observation :

Number of duplicate rows = 0

Number of Rows :872
Number of Columns :7

Univariate Analysis of Numerical Variables - Histogram & Box-plot

A histogram takes as input a numeric variable only. The variable is cut into several bins, and the number of observation per bin is represented by the height of the bar. It is possible to represent the distribution of several variable on the same axis using this technique.

A box-plot gives a nice summary of one or several numeric variables. The line that divides the box into 2 parts represents the median of the data. The end of the box shows the upper and lower quartiles. The extreme lines show the highest and lowest value excluding outliers.

```
<AxesSubplot:xlabel='Salary'>
```

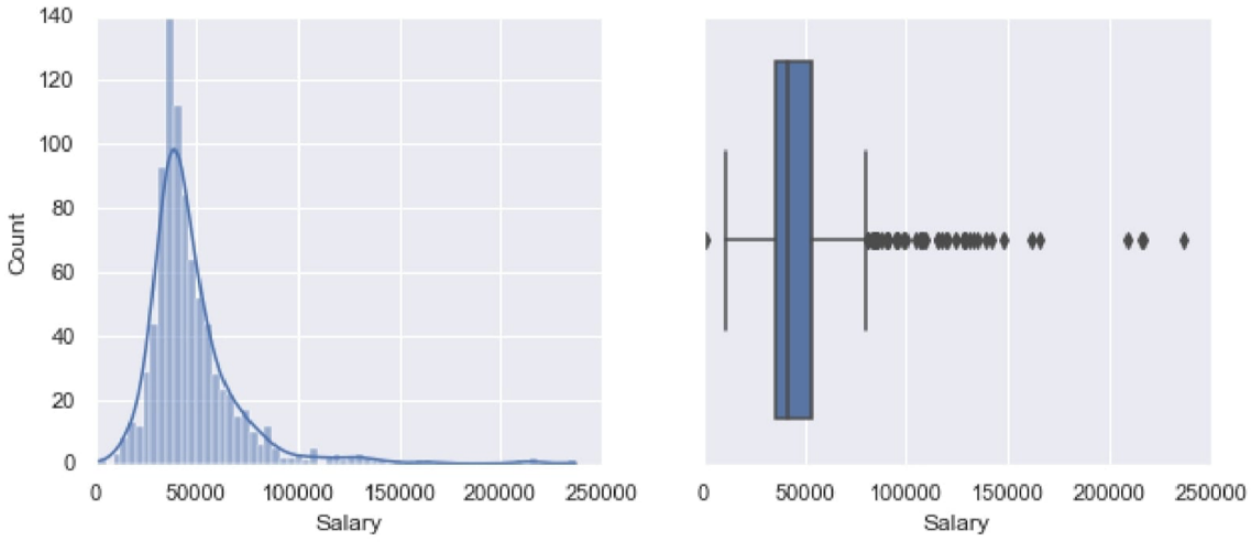


Fig: 1 Histogram & Box-Plot of Salary

```
count       872.000000
mean      47729.172018
std       23418.668531
min        1322.000000
25%       35324.000000
50%       41903.500000
75%       53469.500000
max      236961.000000
Name: Salary, dtype: float64
```

Statistical Description of Salary

- **Insights**
- Salary : Employee salary ranges from a minimum of 1322 to maximum of 236961.
- The average Salary : Employee salary is around 47729.172.
- The standard deviation of the Salary: Employee salary is 23418.668.
- 25% , 50% (median) and 75 % of the Salary : Employee salary are 35324 , 41903.500 and 53469.500.
- Skewness indicating that the distribution is slightly right skewed.(Skew Value - 3.0978)
- Salary:Employee salary have outliers.

```
<AxesSubplot:xlabel='age'>
```

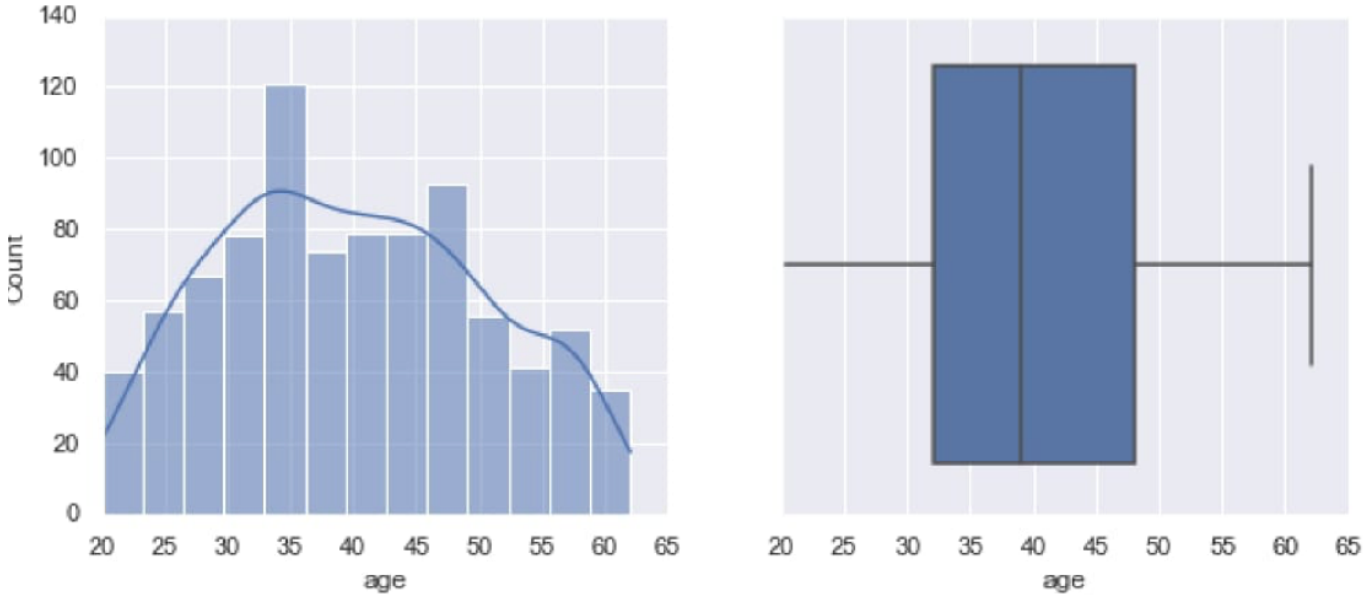


Fig: 2 Histogram & Box-Plot of Age

```
count    872.000000
mean      39.955275
std       10.551675
min       20.000000
25%       32.000000
50%       39.000000
75%       48.000000
max       62.000000
Name: age, dtype: float64
```

Statistical Description ofAge

- **Insights**
- Age : Age in years ranges from a minimum of 20 to maximum of 62.
- The average Age : Age in years is around 39.95.
- The standard deviation of the Age: Age in years is 10.55.
- 25% , 50% (median) and 75 % of the Age : Age in years are 32 , 39 and 48.
- Skewness indicating that the distribution is slightly right skewed.(Skew Value - 0.146160)
- Age : Age in years don't have outliers.

```
<AxesSubplot:xlabel='educ'>
```

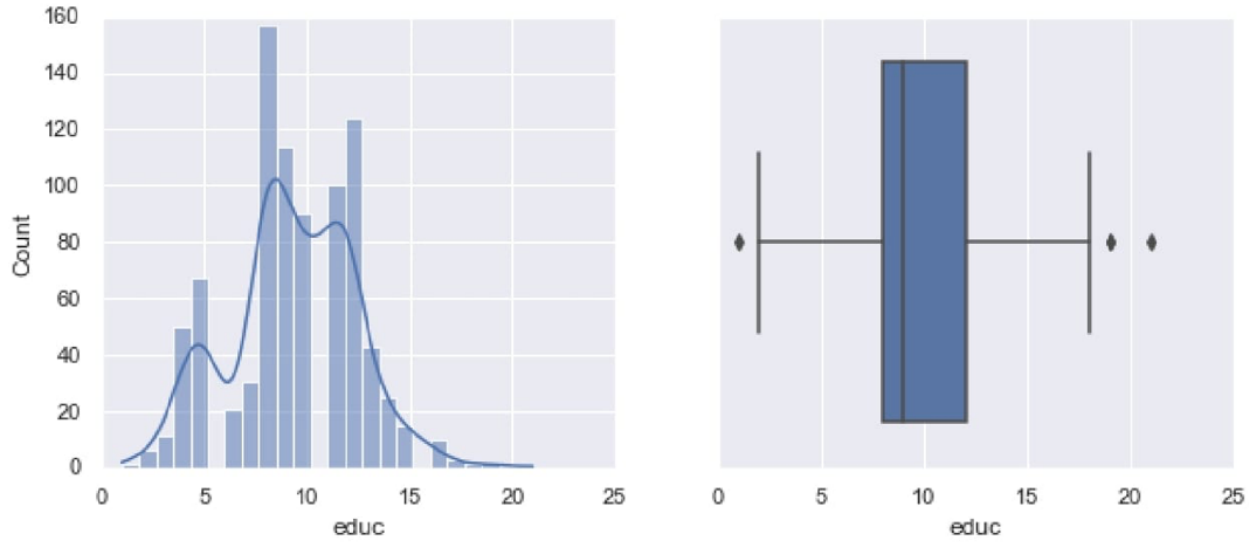


Fig: 3 Histogram & Box-Plot of Education

```
count    872.000000
mean       9.307339
std        3.036259
min        1.000000
25%        8.000000
50%        9.000000
75%       12.000000
max       21.000000
Name: educ, dtype: float64
```

Statistical Description of Educ

- **Insights**
- Educ : Years of formal education ranges from a minimum of 1 to maximum of 62.
- The average Educ : Years of formal education is around 9.30.
- The standard deviation of the Educ: Years of formal education is 3.03.
- 25% , 50% (median) and 75 % of the Educ : Years of formal are 8 , 9 and 12.
- Skewness indicating that the ditribution is slightly left skewed.(Skew Value - -0.045423)
- Educ : Years of formal education have outliers.

Univariate Analysis of Discrete Variables.

***Countplot**

A countplot is kind of like a histogram or a bar graph for Discrete & categorical variables.

```
<AxesSubplot:xlabel='no_young_children', ylabel='count'>
```

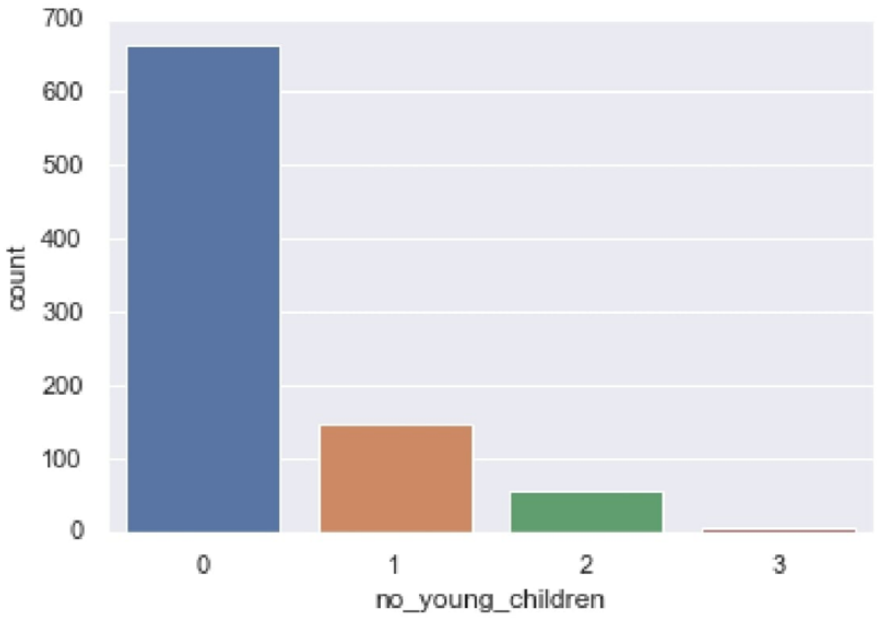


Fig: 4 Count Plot of no_young_children

- **Insights**

- 76.26% employee have 0 young children (younger than 7 years).
- 16.85% employee have 1 young children (younger than 7 years).
- 6.3% employee have 2 young children (younger than 7 years).
- 0.57% employee have 3 young children (younger than 7 years).

```
<AxesSubplot:xlabel='no_older_children', ylabel='count'>
```

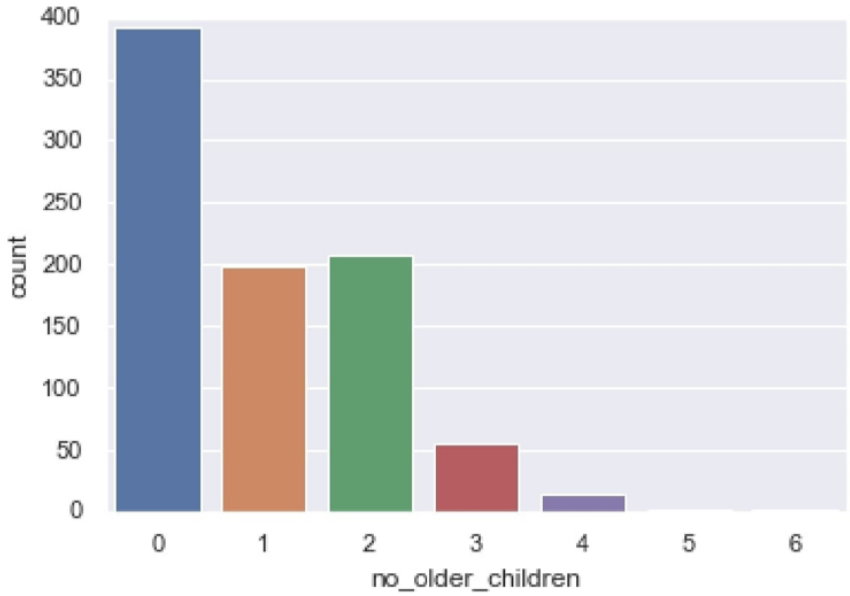


Fig: 5 Count Plot of no_older_children

- **Insights:**
- 45.06% employee have 0 older children.
- 22.70% employee have 1 older children.
- 23.85% employee have 2 older children.
- 6.3% employee have 3 older children.
- 1.6% employee have 4 older children.
- 0.22% employee have 5 older children.
- 0.22% employee have 6 older children.

Univariate Analysis of Categorical Variables.

***Countplot**

A countplot is kind of like a histogram or a bar graph for categorical variables.

```
<AxesSubplot:xlabel='Holliday_Package', ylabel='count'>
```

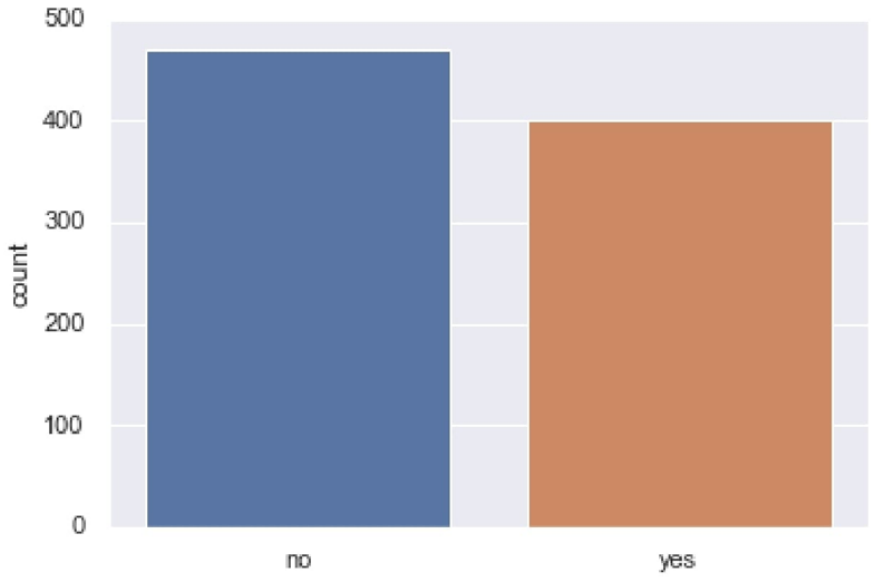


Fig: 6 Count Plot of Holiday_Package

- **Insights**

- 54.0138% employee Opted 'no' for Holiday Package.
- 45.9862% employee Opted 'yes' for Holiday Package

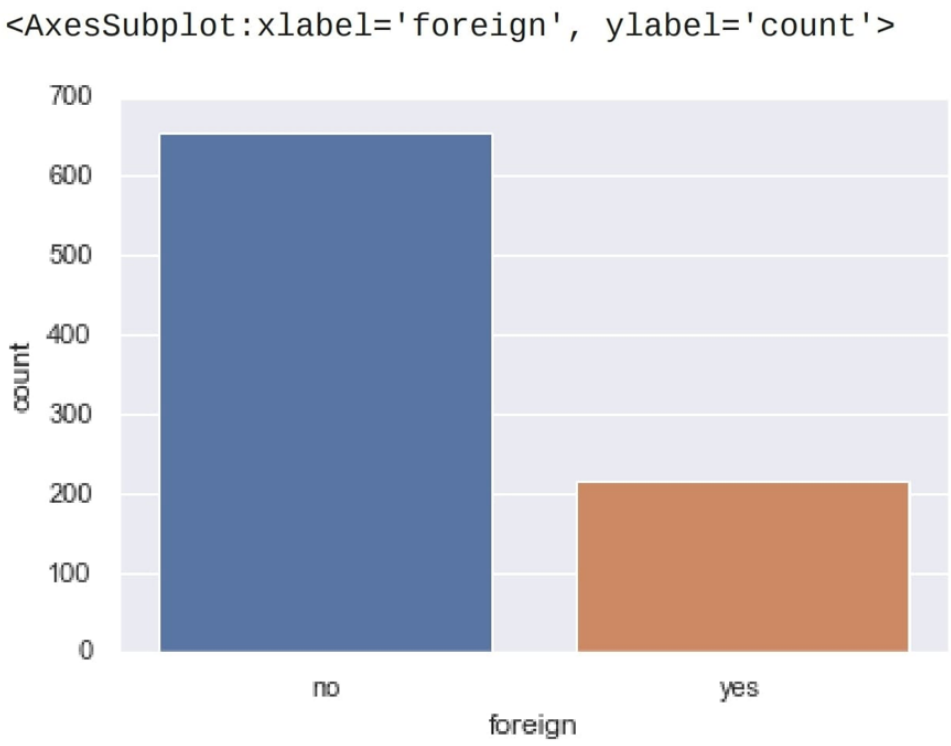


Fig: 7 Count Plot of Foreign

- **Insights:**

- 75.22% employee are not foreigner.
- 24.78% employee are foreigner.

Bivariate Analysis

A scatter plot (aka scatter chart, scatter graph) uses dots to represent values for two different numeric variables. The position of each dot on the horizontal and vertical axis indicates values for an individual data point. Scatter plots are used to observe relationships between variables.

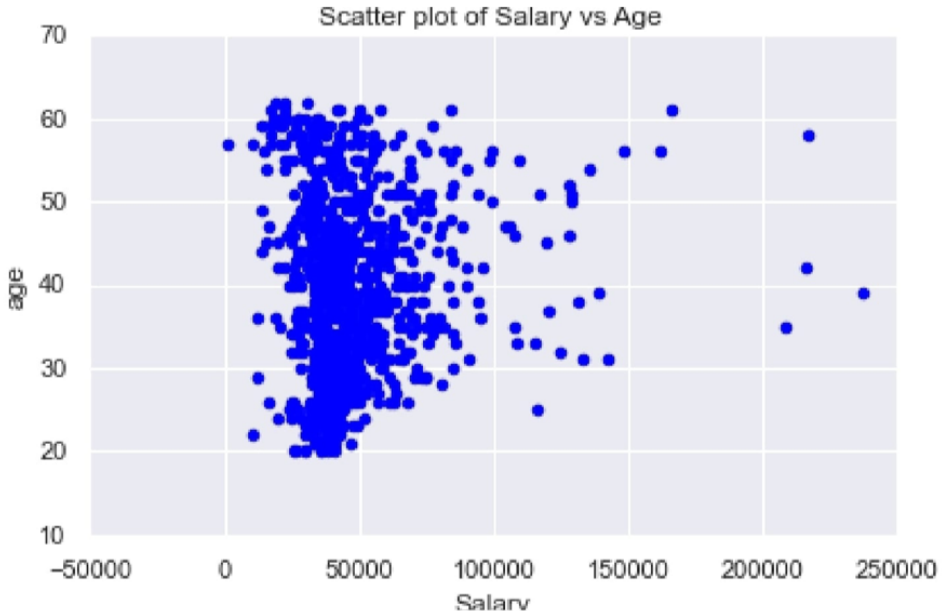


Fig: 8 Scatter Plot of Salary VS Age

Insights

There is no such significant relationship between salary and age.

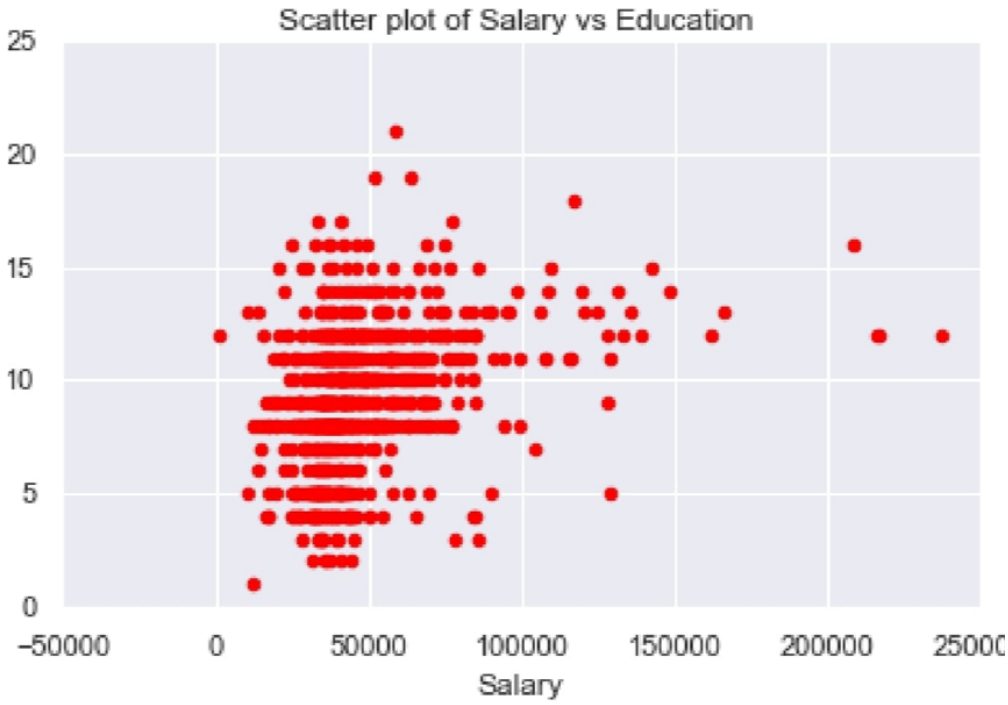


Fig: 9 Scatter Plot of Salary VS Education

Insights:
There is weak relation between salary and education.

***Countplot with Hue.**

A countplot is kind of like a histogram or a bar graph for categorical variables.

Hue :This parameter take column name for colour encoding

```
<AxesSubplot:xlabel='foreign', ylabel='count'>
```

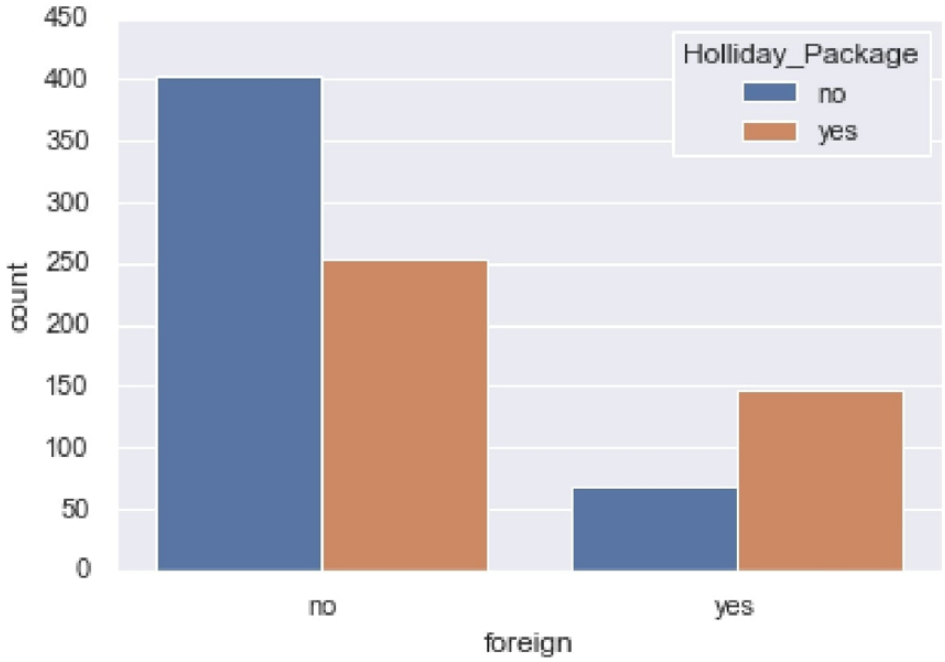


Fig: 10 Count Plot of Foreign VS Holiday_Package

- **Insights**

- Around 61.2% employee who are not foreigner opted no for holiday package.
- Around 38.71% employee who are not foreigner opted yes for holiday package.
- Around 68.05% employee who are foreigner opted yes for holiday package.
- Around 31.94% employee who are foreigner opted no for holiday package.

```
<AxesSubplot:xlabel='Holliday_Package', ylabel='Salary'>
```

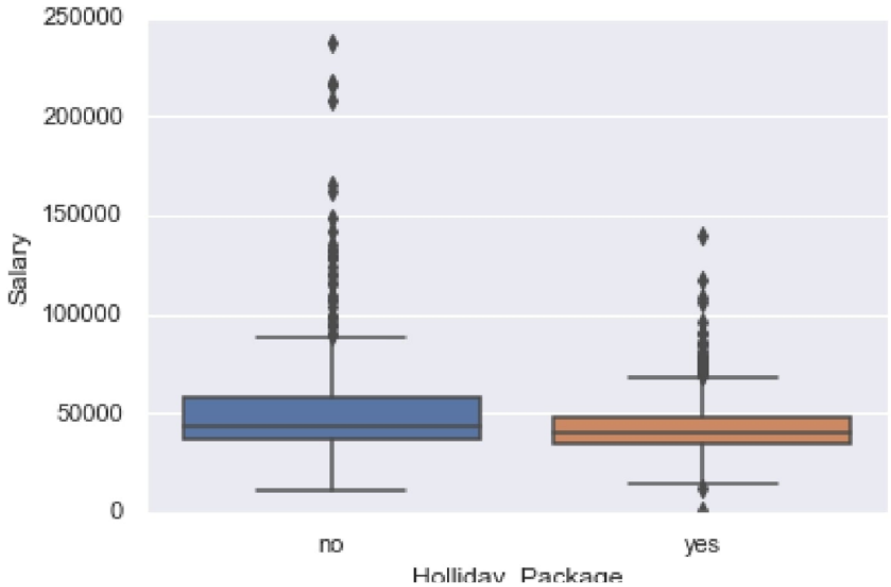


Fig: 11 Box Plot of Holiday_Package VS Salary

- **Insights**

- 50% employee who opted yes for the holiday package having the salary level of 40K-45K.
- 50% employee who opted no for the holiday package having the salary level of 45K-50K.
- Distribution is wider for no and lesser for yes. More Salary indicates more employee opted no for holiday package.

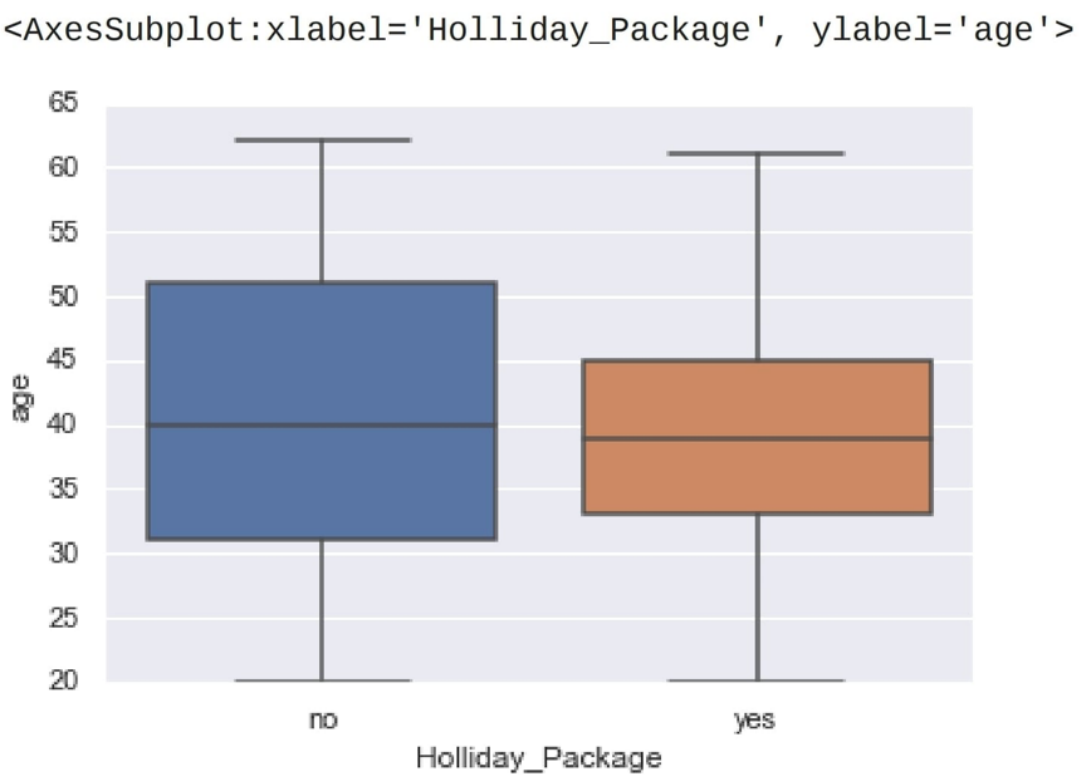


Fig: 12 Box Plot of Holiday_Package VS Age

- **Insights**

- 50% employee who opted yes for the holiday package are in the age level of 37-38.
- 50% employee who opted no for the holiday package are in the age level of 40.
- Distribution is much wider for no and lesser for yes . More age indicates more employee opted no for holiday package.

```
<AxesSubplot:xlabel='Holliday_Package', ylabel='educ'>
```

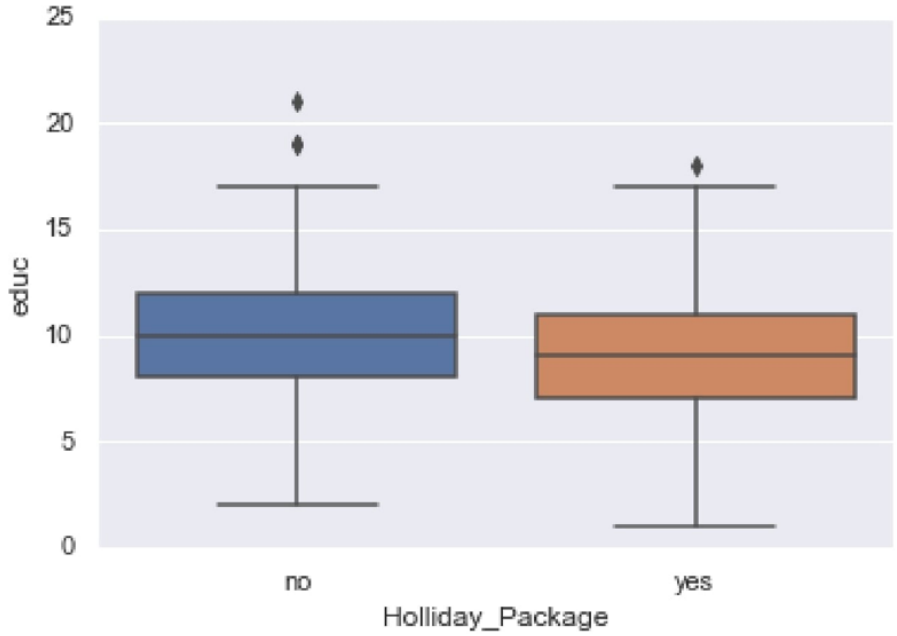


Fig: 13 Box Plot of Holiday_Package VS Education

- **Insights**

- Distribution is almost similar between no and yes opted for holiday package. Medians are almost same.
- 50% employee who opted no for holiday package having educ of 10.
- 50% employee who opted yes for holiday package having educ of 9.

```
<AxesSubplot:xlabel='Holliday_Package', ylabel='no_young_children'>
```

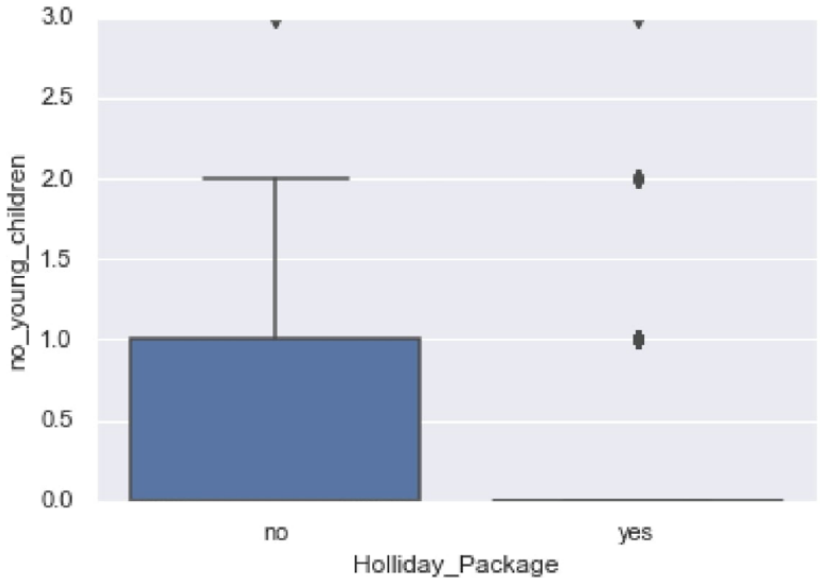


Fig: 14 Box Plot of Holiday_Package VS no_young_children

Insights

- no_young_children shows clear distinction between no and yes opted for holiday package.Employee who has opted no for holiday pacakge shows a wider distribution indicating indicating more no_young_children. Whereas employee who has opted yes for holiday package smaller distribution (mostly near no_young_children 0) with many outliers indicating few customers who has more no_young_children still has opted yes for holiday package.

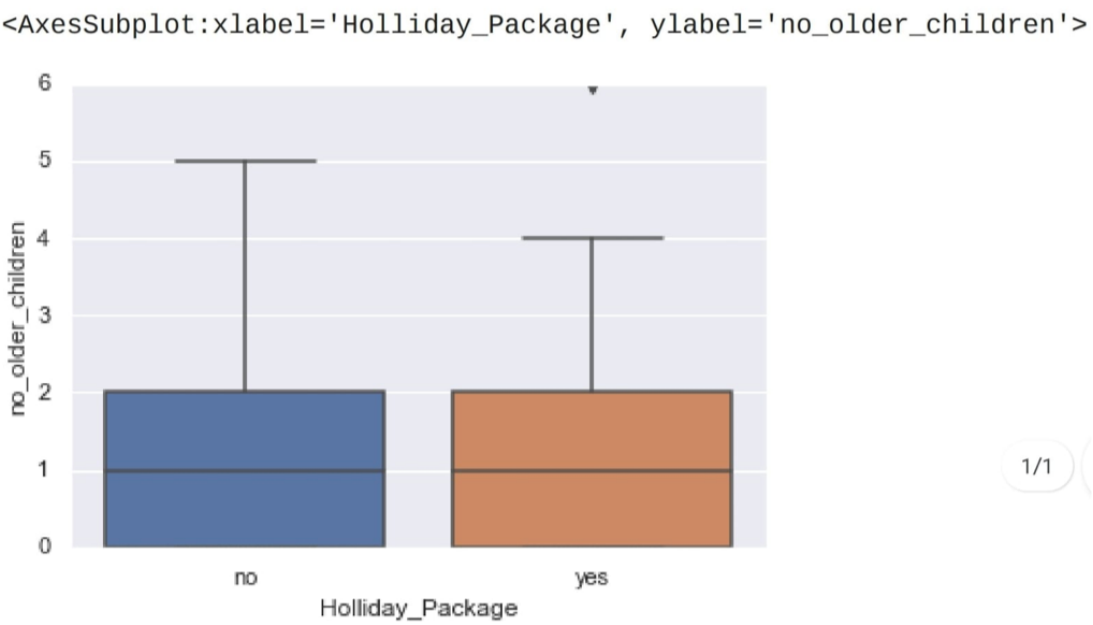


Fig: 15 Box Plot of Holiday_Package VS no_older_children

- **Insights**

- Distribution is almost similar between no and yes opted for holiday package.
- 50% employee who opted no and yes for holiday package have 1 no_older_children.
- 75% employee who opted no and yes for holiday package have 2 no_older_children.

*** Heatmap**

A correlation heatmap uses colored cells, typically in a monochromatic scale, to show a 2D correlation matrix (table) between two discrete dimensions or event types.Correlation heatmaps are ideal for comparing the measurement for each pair of dimension values.Darker Shades have higher correlation, while lighter shades have smaller values of correlation as compared to darker shades values.Correlation values near to 1 or -1 are highly positively correlated and highly negatively correlated respectively. Correlation values near to 0 are not correlated to each other.

Checking for Correlations

	Salary	age	educ	no_young_children	no_older_children
Salary	1.000000	0.071709	0.326540	-0.029664	0.113772
age	0.071709	1.000000	-0.149294	-0.519093	-0.116205
educ	0.326540	-0.149294	1.000000	0.098350	-0.036321
no_young_children	-0.029664	-0.519093	0.098350	1.000000	-0.238428
no_older_children	0.113772	-0.116205	-0.036321	-0.238428	1.000000

TAB:8 CORRELATION TABLE

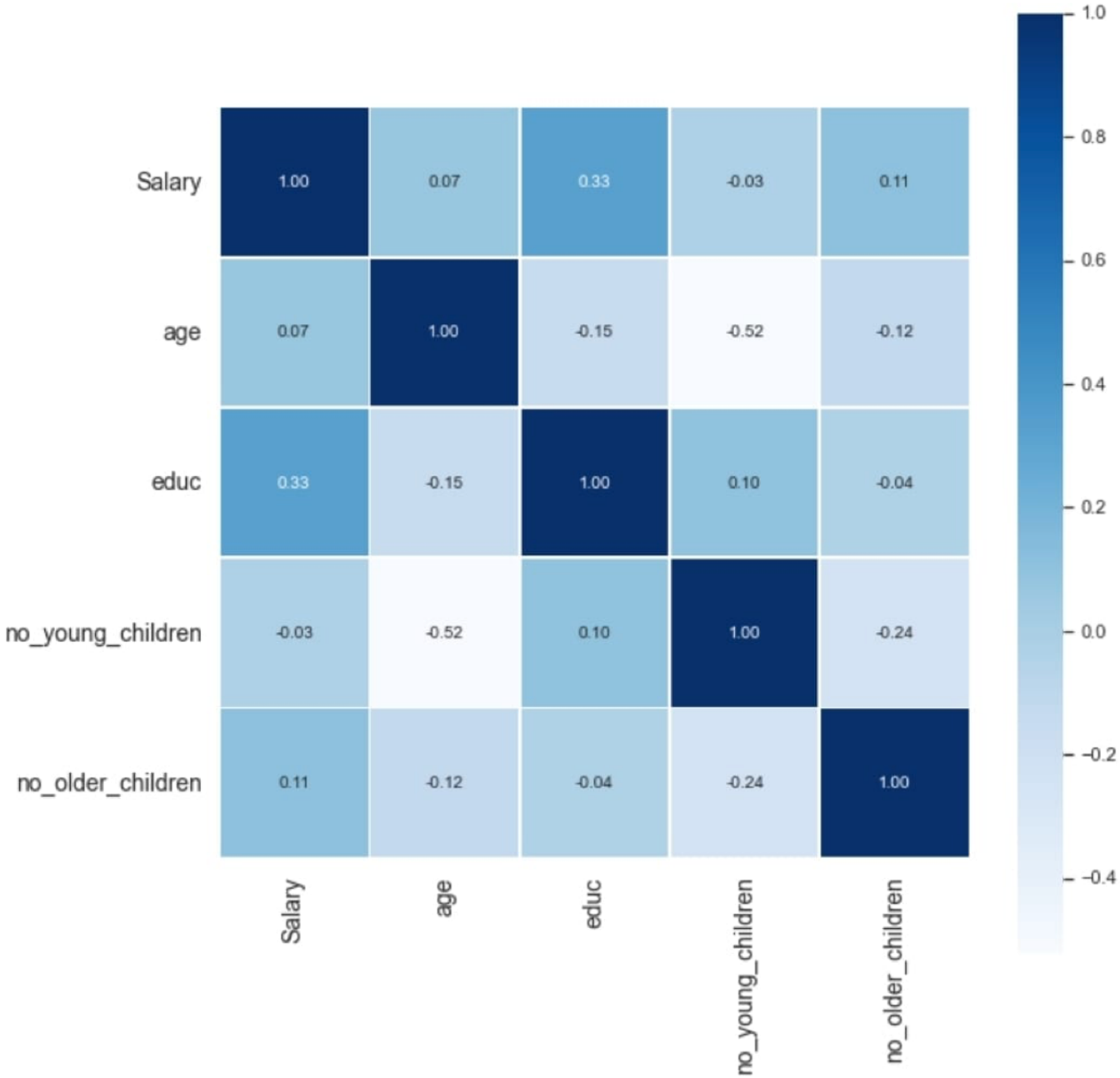


Fig: 16 HeatMap of Problem 2

Insights

From the above correlation table we conclude that -

- Salary with age shows weak correaltion (not so significant)i.e. 0.07.
- Salary with educ shows max correlation i.e. 0.33
- age with no_young_children shows least correlation i.e. -0.52.

*Pairplot

Pairplot shows the relationship between the variables in the form of scatterplot and the distribution of the variable in the form of histogram.

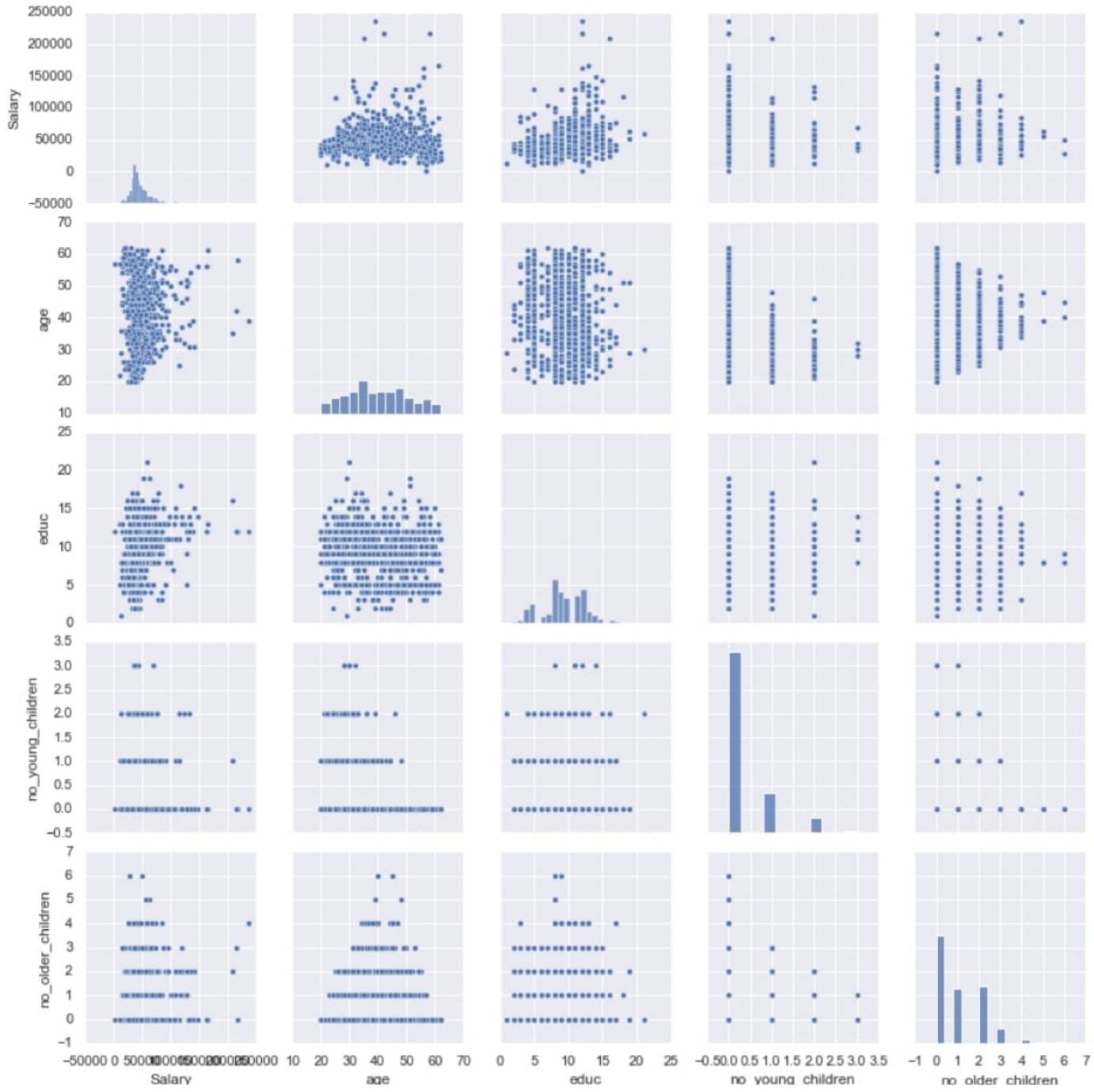


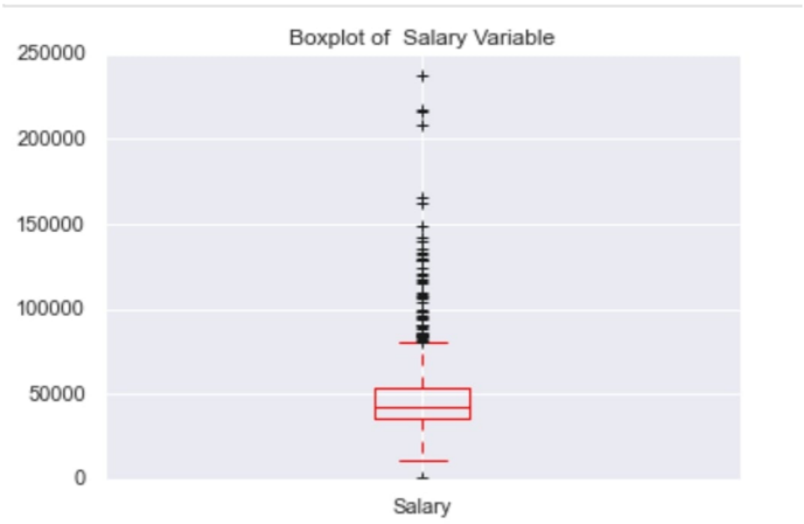
Fig: 17 PairPlot of Problem 2

Insights

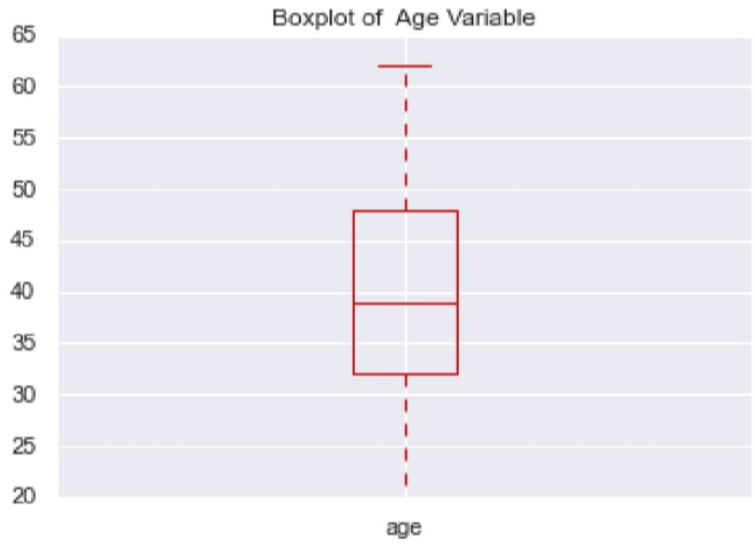
There is no such strong relationship between the variables.

Checking for Outliers in the dataset.

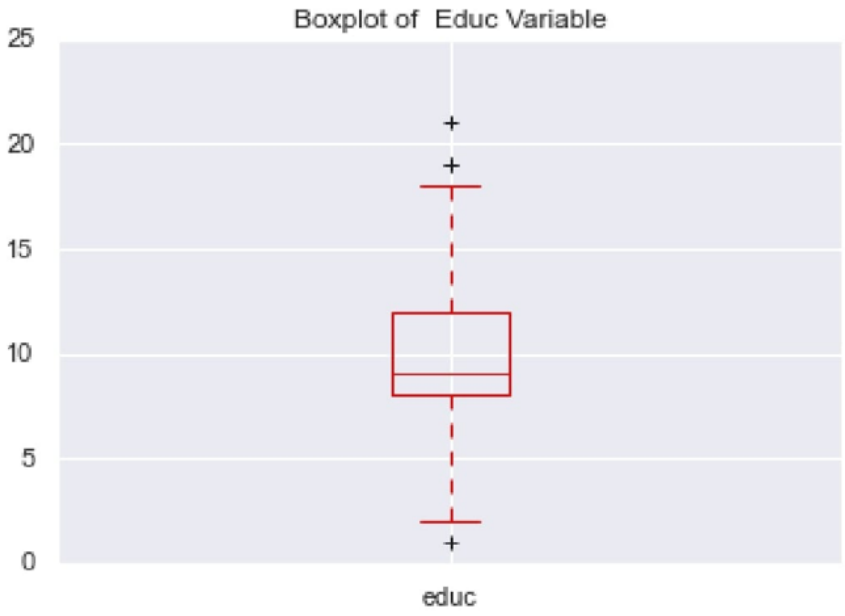
To check for outliers, we will be plotting the box plots.



BoxPlot of Salary Variable



BoxPlot of Age Variable



BoxPlot of Educ Variable

Fig: 18 BoxPlot for Outlier Check Problem 2

Treatment of Outliers in the dataset.

Although outliers exists as per the boxplot, by looking at the data distribution in describe(), the values are not too far away. Treating the outliers by converting them to min/max values will cause most variables to have values to be the same. So, outliers are not treated in this case.Only salary have outliers to extereme high values so we need treat outliers of salary only.

	count	mean	std	min	25%	50%	75%	max
Salary	872.0	47729.172018	23418.668531	1322.0	35324.0	41903.5	53469.5	236961.0
age	872.0	39.955275	10.551675	20.0	32.0	39.0	48.0	62.0
educ	872.0	9.307339	3.036259	1.0	8.0	9.0	12.0	21.0
no_young_children	872.0	0.311927	0.612870	0.0	0.0	0.0	0.0	3.0
no_older_children	872.0	0.982798	1.086786	0.0	0.0	1.0	2.0	6.0

From the above we saw Only salary have outliers to extreme high values so we need treat outliers of salary only.We use 25% / 75% IQR imputation method for outliers treatment as we saw in the above box plots we have outliers in salary independent features .Now here we treating the outliers after treatment no outliers present in the salary independent variables. So let's check their box plot once again confirm the same.

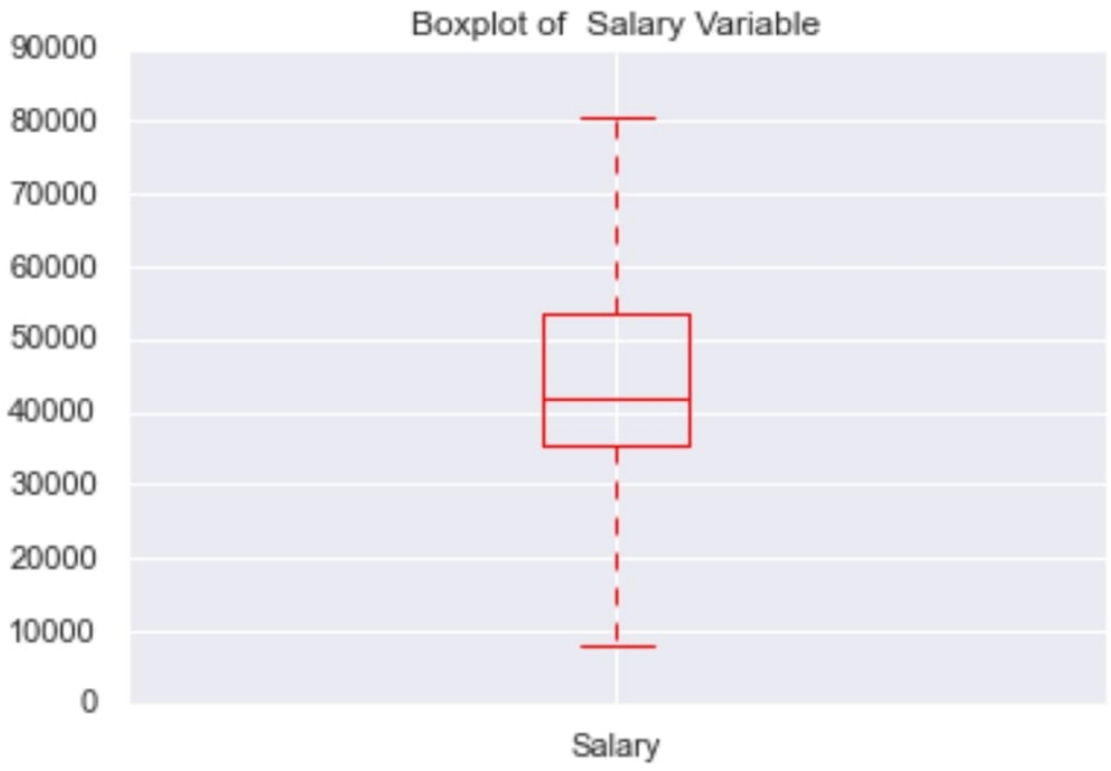


Fig: 19 BoxPlot of After Outlier Treatment (Salary) Problem 2

Insights

As we sucessfully treated the outliers and from the above plotted boxplot, we clearly infer that their is no more outliers present in the data.

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).

ENCODING

By using label encoding .codes func( ) , we are going to encode the two categorical variables present in the data one is Holiday_Package and foreign.

Holiday_Package

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

Before Encoding

```
Holliday_Package
 0    471
1    401
Name: Holliday_Package, dtype: int64
```

After Encoding

foreign -

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

Before Encoding

```
foreign
 0    656
1    216
Name: foreign, dtype: int64
```

After Encoding

TAB:9 ENCODING TABLE

Checking Original Dataset after Encoding -

	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
0	0	48412.00	30	8	1	1	0
1	1	37207.00	45	8	0	1	0
2	0	58022.00	46	9	0	0	0
3	0	66503.00	31	11	2	0	0
4	0	66734.00	44	12	0	2	0
5	1	61590.00	42	12	0	1	0
6	0	80687.75	51	8	0	0	0
7	1	35987.00	32	8	0	2	0
8	0	41140.00	39	12	0	0	0
9	0	35826.00	43	11	0	2	0

TAB:10 DATASET AFTER ENCODING

Checking the Appropriateness of Datatypes & Information of the DataFrame

```
<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    float64
 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: float64(1), int64(4), int8(2)
memory usage: 35.9 KB
```

Result

- Label Encoding has been done for categorical columns and all columns are now in number.
- After performing EDA , various data preprocessing & data preparation steps. Our dataset is now ready for supervised modelling algorithms like Logistic Regression & LDA (Linear Discriminant Analysis).

Proportion of 1s and 0s.

0s	1s
0.540138	0.459862

TAB:11 PROPORTION OF 0S & 1S

There is no issue of class imbalance here as we have reasonable proportions in both the classes.

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

Here we separating the independent and target column for performing the modelling. We have 6 independent variables named as salary , age , edu , no_young_children , no_older_children & foreign , Holiday_Package is out target variable.

Train-Test Split for Linear Regression Model -

The train-test split is a technique for evaluating the performance of a machine learning algorithm. It can be used for classification or regression problems and can be used for any supervised learning algorithm. The procedure involves taking a dataset and dividing it into two subsets.

In the given problem, we are advised to split the training and the testing data in the ratio of (70: 30).Here we are split the data into train and test part , like x_train , x_test , train_labels & test_labels ,by using train_test_split func() from sk-learn library here , we are taking 70 % data for training and 30 % data for testing.

Checking the dimensions of the training and test data.

Train Data Shape	Test Data Shape
X_train (610, 6)	X_test (262, 6)
train_labels (610,)	test_labels (262,)

TAB:12 SHAPE OF TRAIN & TEST DATA

Logistic Regression Model :

Logistic regression is a statistical analysis method used to predict a data value based on prior observations of a data set. ... A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.

Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes. For example, it can be used for cancer detection problems. It computes the probability of an event occurrence.

The logistic function, also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.
1 / (1 + e^-value)
Where e is the base of the natural logarithms (Euler's number or the EXP() function in your spreadsheet) and value is the actual numerical value that you want to transform.
Below is a plot of the numbers between -5 and 5 transformed into the range 0 and 1 using the logistic function.

Representation Used for Logistic Regression :

Logistic regression uses an equation as the representation, very much like linear regression.
Input values (x) are combined linearly using weights or coefficient values (referred to as the Greek capital letter Beta) to predict an output value (y). A key difference from linear regression is that the output value being modeled is a binary values (0 or 1) rather than a numeric value.
Below is an example logistic regression equation:
y = e^(b0 + b1*x) / (1 + e^(b0 + b1*x))
Where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x). Each column in your input data has an associated b coefficient (a constant real value) that must be learned from your training data.
The actual representation of the model that you would store in memory or in a file are the coefficients in the equation (the beta value or b's).

Logistic Regression Predicts Probabilities (Technical Interlude)

Logistic regression models the probability of the default class (e.g. the first class).
For example, if we are modeling people's sex as male or female from their height, then the first class could be male and the logistic regression model could be written as the probability of male given a person's height, or more formally:
P(sex=male|height)
Written another way, we are modeling the probability that an input (X) belongs to the default class (Y=1), we can write this formally as:
P(X) = P(Y=1|X)
We're predicting probabilities? I thought logistic regression was a classification algorithm?
Note that the probability prediction must be transformed into a binary values (0 or 1) in order to actually make a probability prediction. More on this later when we talk about making predictions.

Logistic regression is a linear method, but the predictions are transformed using the logistic function. The impact of this is that we can no longer understand the predictions as a linear combination of the inputs as we can with linear regression, for example, continuing on from above, the model can be stated as:
p(X) = e^(b0 + b1*X) / (1 + e^(b0 + b1*X))

I don't want to dive into the math too much, but we can turn around the above equation as follows (remember we can remove the e from one side by adding a natural logarithm (ln) to the other):
ln(p(X) / 1 – p(X)) = b0 + b1 * X
This is useful because we can see that the calculation of the output on the right is linear again (just like linear regression), and the input on the left is a log of the probability of the default class.
This ratio on the left is called the odds of the default class (it's historical that we use odds, for example, odds are used in horse racing rather than probabilities). Odds are calculated as a ratio of the probability of the event divided by the probability of not the event, e.g. 0.8/(1-0.8) which has the odds of 4. So we could instead write:
ln(odds) = b0 + b1 * X
Because the odds are log transformed, we call this left hand side the log-odds or the probit. It is possible to use other types of functions for the transform (which is out of scope_, but as such it is common to refer to the transform that relates the linear regression equation to the probabilities as the link function, e.g. the probit link function.
We can move the exponent back to the right and write it as:
odds = e^(b0 + b1 * X)
All of this helps us understand that indeed the model is still a linear combination of the inputs, but that this linear combination relates to the log-odds of the default class.

The coefficients (Beta values b) of the logistic regression algorithm must be estimated from your training data. This is done using maximum-likelihood estimation.
Maximum-likelihood estimation is a common learning algorithm used by a variety of machine learning algorithms, although it does make assumptions about the distribution of your data (more on this when we talk about preparing your data).
The best coefficients would result in a model that would predict a value very close to 1 (e.g. male) for the default class and a value very close to 0 (e.g. female) for the other class. The intuition for maximum-likelihood for logistic regression is that a search procedure seeks values for the coefficients (Beta values) that minimize the error in the probabilities predicted by the model to those in the data (e.g. probability of 1 if the data is the primary class).
We are not going to go into the math of maximum likelihood. It is enough to say that a minimization algorithm is used to optimize the best values for the coefficients for your training data. This is often implemented in practice using efficient numerical optimization algorithm (like the Quasi-newton method).
When you are learning logistic, you can implement it yourself from scratch using the much simpler gradient descent algorithm.

Making Predictions with Logistic Regression

Making predictions with a logistic regression model is as simple as plugging in numbers into the logistic regression equation and calculating a result.
Let's make this concrete with a specific example.
Let's say we have a model that can predict whether a person is male or female based on their height (completely fictitious). Given a height of 150cm is the person male or female.

We have learned the coefficients of b0 = -100 and b1 = 0.6. Using the equation above we can calculate the probability of male given a height of 150cm or more formally P(male|height=150). We will use EXP() for e, because that is what you can use if you type this example into your spreadsheet:
y = e^(b0 + b1*X) / (1 + e^(b0 + b1*X))
y = exp(-100 + 0.6*150) / (1 + EXP(-100 + 0.6*X))
y = 0.0000453978687
Or a probability of near zero that the person is a male.
In practice we can use the probabilities directly. Because this is classification and we want a crisp answer, we can snap the probabilities to a binary class value, for example:
0 if p(male) < 0.5
1 if p(male) >= 0.5
Now that we know how to make predictions using logistic regression, let's look at how we can prepare our data to get the most from the technique.

Building Logistic Regression Model -

Grid Search for finding out the optimal values for the hyper parameters-

Grid search builds a model for every combination of hyperparameters specified and evaluates each model. A more efficient technique for hyperparameter tuning is the Randomized search — where random combinations of the hyperparameters are used to find the best solution.
As per the industries standards we are taking various hyper parameters to build our logistic regression ,hyperparametrs are listed below.

penalty	'l2','none'
solver	'sag','lbfgs','newton-cg'
tol	0.0001,0.00001

TAB:13 GRID SEARCH PARAMETR FOR LOGISTIC REGRESSION MODEL

The best estimator for building our decision tree are obtained by using the grid search cv function() are tabluate below:

penalty	'l2'
solver	newton-cg'
tol	0.0001

TAB:14 BEST GRID SEARCH PARAMETR FOR LOGISTIC REGRESSION MODEL

Result :

From grid search we get penalty = 'l2' , solver = newton-cg' , tol = 0.0001 , we are going to use these parameter to build our logistic regression model. We can create logistic regression model by the help of sklearn lib by import LogisticRegression. Now fit these parameters into the model with train data , check predictions on train and test data.

LDA (linear discriminant analysis) :

Linear Discriminant Analysis, or LDA for short, is a predictive modeling algorithm for multi-class classification. It can also be used as a dimensionality reduction technique, providing a projection of a training dataset that best separates the examples by their assigned class.

The representation of LDA is straight forward.
It consists of statistical properties of your data, calculated for each class. For a single input variable (x) this is the mean and the variance of the variable for each class. For multiple variables, this is the same properties calculated over the multivariate Gaussian, namely the means and the covariance matrix.
These statistical properties are estimated from your data and plug into the LDA equation to make predictions. These are the model values that you would save to file for your model.

LDA makes some simplifying assumptions about your data:
That your data is Gaussian, that each variable is is shaped like a bell curve when plotted.
That each attribute has the same variance, that values of each variable vary around the mean by the same amount on average.
With these assumptions, the LDA model estimates the mean and variance from your data for each class. It is easy to think about this in the univariate (single input variable) case with two classes.
The mean (mu) value of each input (x) for each class (k) can be estimated in the normal way by dividing the sum of values by the total number of values.

muk = 1/nk * sum(x)
Where muk is the mean value of x for the class k, nk is the number of instances with class k. The variance is calculated across all classes as the average squared difference of each value from the mean.
sigma^2 = 1 / (n-K) * sum((x – mu)^2)
Where sigma^2 is the variance across all inputs (x), n is the number of instances, K is the number of classes and mu is the mean for input x.

LDA makes predictions by estimating the probability that a new set of inputs belongs to each class. The class that gets the highest probability is the output class and a prediction is made.
The model uses Bayes Theorem to estimate the probabilities. Briefly Bayes' Theorem can be used to estimate the probability of the output class (k) given the input (x) using the probability of each class and the probability of the data belonging to each class:
P(Y=x|X=x) = (PIk * fk(x)) / sum(PIl * fl(x))
Where PIk refers to the base probability of each class (k) observed in your training data (e.g. 0.5 for a 50-50 split in a two class problem). In Bayes' Theorem this is called the prior probability.
PIk = nk/n
The f(x) above is the estimated probability of x belonging to the class. A Gaussian distribution function is used for f(x). Plugging the Gaussian into the above equation and simplifying we end up with the equation below. This is called a discriminate function and the class is calculated as having the largest value will be the output classification (y):

Dk(x) = x * (muk/siga^2) – (muk^2/(2*sigma^2)) + ln(PIk)
Dk(x) is the discriminate function for class k given input x, the muk, sigma^2 and PIk are all estimated from your data.

Linear Discriminant Analysis is a simple and effective method for classification. Because it is simple and so well understood, there are many extensions and variations to the method. Some popular extensions include:
Quadratic Discriminant Analysis (QDA): Each class uses its own estimate of variance (or covariance when there are multiple input variables).
Flexible Discriminant Analysis (FDA): Where non-linear combinations of inputs is used such as splines.
Regularized Discriminant Analysis (RDA): Introduces regularization into the estimate of the variance (actually covariance), moderating the influence of different variables on LDA.
The original development was called the Linear Discriminant or Fisher's Discriminant Analysis. The multi-class version was referred to Multiple Discriminant Analysis. These are all simply referred to as Linear Discriminant Analysis now.

Building A LDA Model : -

We can create LDA model by the help of sklearn lib by import import Linear Discriminant Analysis. Now fit train data into the model, check predictions on train and test data.

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.

Logistics Regression Model

Predicting on Training and Test dataset -
y_train_predict -

```
array([0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0,
       0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1,
       0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0,
       0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1,
       0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0,
       1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1,
       0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1,
       1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0,
       0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0,
       1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0,
       0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1,
       0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1,
       0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0,
       0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0,
       1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0,
       1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0], d-type=int8)
```

TAB:15 LOGISTIC REGRESSION PREDICTION ON TRAIN DATA

y_test_predict_ -

```
array([0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1,
       1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0,
       1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1,
       1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0,
       1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
       1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0,
       0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1,
       0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0],
      dtype=int8)
```

TAB:16 LOGISTIC REGRESSION PREDICTION ON TEST DATA

Getting the Predicted Probability -

Train Data -

	0	1
0	0.731299	0.268701
1	0.269331	0.730669
2	0.950614	0.049386
3	0.690274	0.309726
4	0.448412	0.551588

Test Data -

	0	1
0	0.767056	0.232944
1	0.286662	0.713338
2	0.888828	0.111172
3	0.952326	0.047674
4	0.498776	0.501224

Model Evaluation

AUC and ROC for the Training Data ------0.741

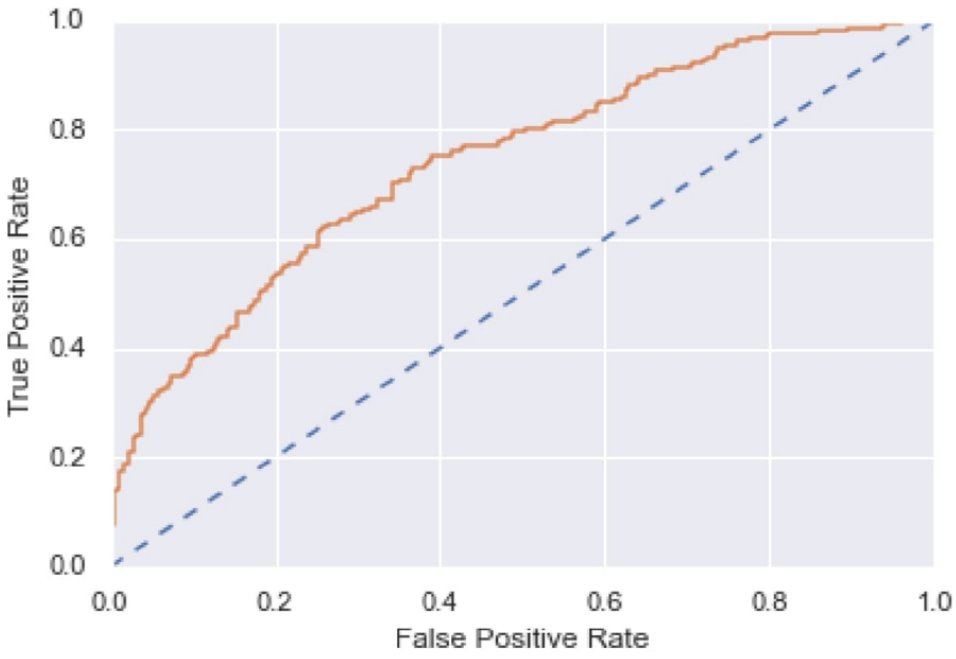


Fig: 20 LR Model - AUC AND ROC for Training Data

Confusion Matrix for the Training Data -

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7ff838beca60>
```

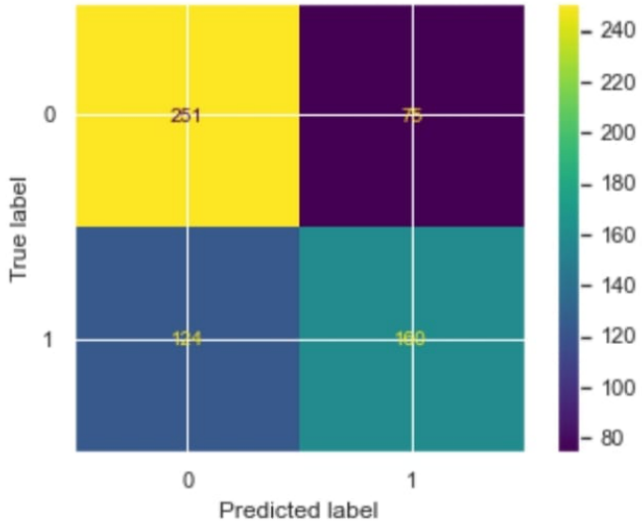


Fig: 21 LR Model - Confusion Matrix for Training Data

Train Data Accuracy---------- 0.6737704918032786

Classification Report Train Data -

	precision	recall	f1-score	support
0	0.67	0.77	0.72	326
1	0.68	0.56	0.62	284
accuracy			0.67	610
macro avg	0.68	0.67	0.67	610
weighted avg	0.67	0.67	0.67	610

TAB:17 CLASSIFIACTION REPORT OF LOGISTIC REGRESSION TRAIN DATA

AUC and ROC for the Test Data----------0.705

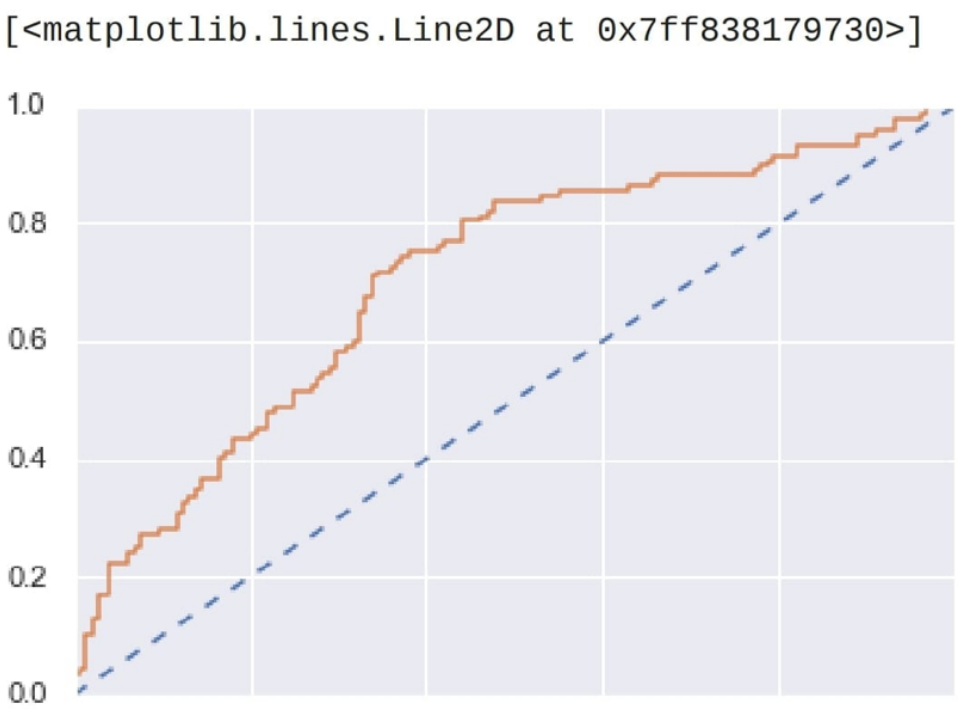


Fig: 22 LR Model - AUC AND ROC for Test Data

Confusion Matrix for Test Data

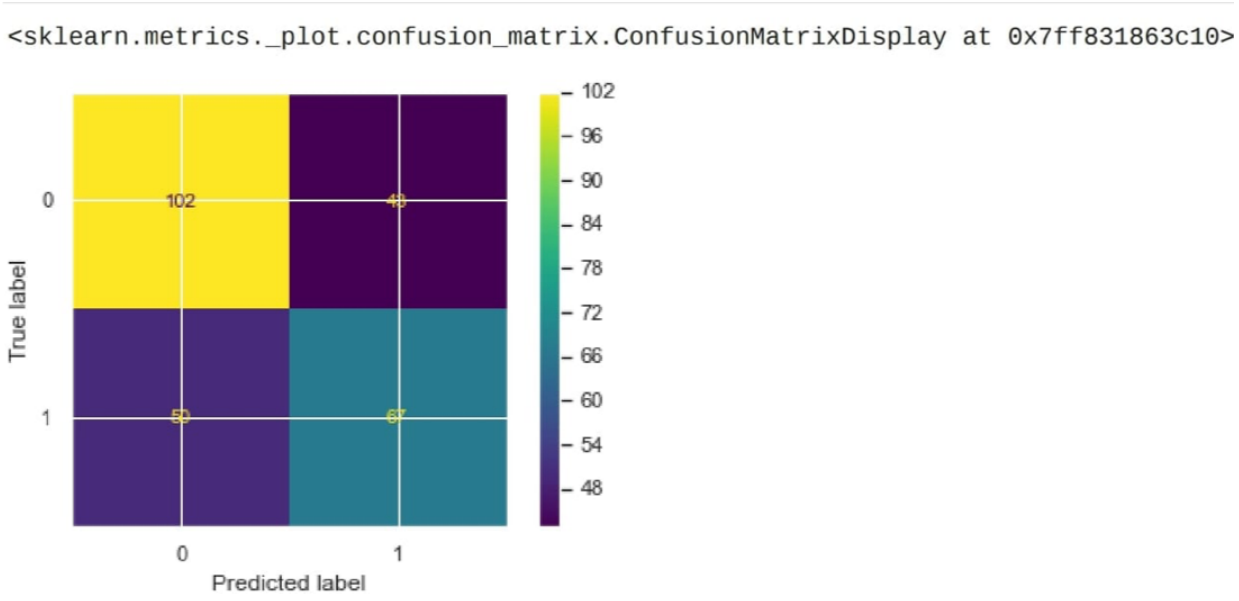


Fig: 23 LR Model - Confusion Matrix for Test Data

Test Data Accuracy----------0.6450381679389313

Classifiaction Report

	precision	recall	f1-score	support
0	0.67	0.70	0.69	145
1	0.61	0.57	0.59	117
accuracy			0.65	262
macro avg	0.64	0.64	0.64	262
weighted avg	0.64	0.65	0.64	262

TAB:18 CLASSIFIACTION REPORT OF LOGISTIC REGRESSION TEST DATA

Logistic Regression Model Conclusion:

Train Data:
AUC: 74.1%
Accuracy: 67.37%
Precision: 68%
Recall: 56%
f1-Score: 62%

Test Data:
AUC: 70.5%
Accuracy: 65%
Precision: 61%
Recall: 57%
f1-Score: 59%

LDA Model

Predicting the Training and Testing data

```
array([0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0,
       0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1,
       0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0,
       0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0,
       1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1,
       0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
       1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0,
       0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0,
       1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0,
       0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1,
       0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1,
       0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0,
       0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0,
       1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0,
       1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0], dtype=int8)
```

TAB:19 LDA PREDICTION ON TRAIN DATA

y_test_predict -

```
array([0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1,
       1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0,
       0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1,
       1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0,
       1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
       1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0,
       0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1,
       0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0],
      dtype=int8)
```

TAB:20 LDA PREDICTION ON TEST DATA

Getting the Predicted Probability

Train Data -

	0	1
0	0.751451	0.248549
1	0.247976	0.752024
2	0.949134	0.050866
3	0.696704	0.303296
4	0.453222	0.546778

Test Data -

	0	1
0	0.764441	0.235559
1	0.277406	0.722594
2	0.887380	0.112620
3	0.950494	0.049506
4	0.508190	0.491810

Model Evaluation

AUC and ROC for the Training Data----------0.739

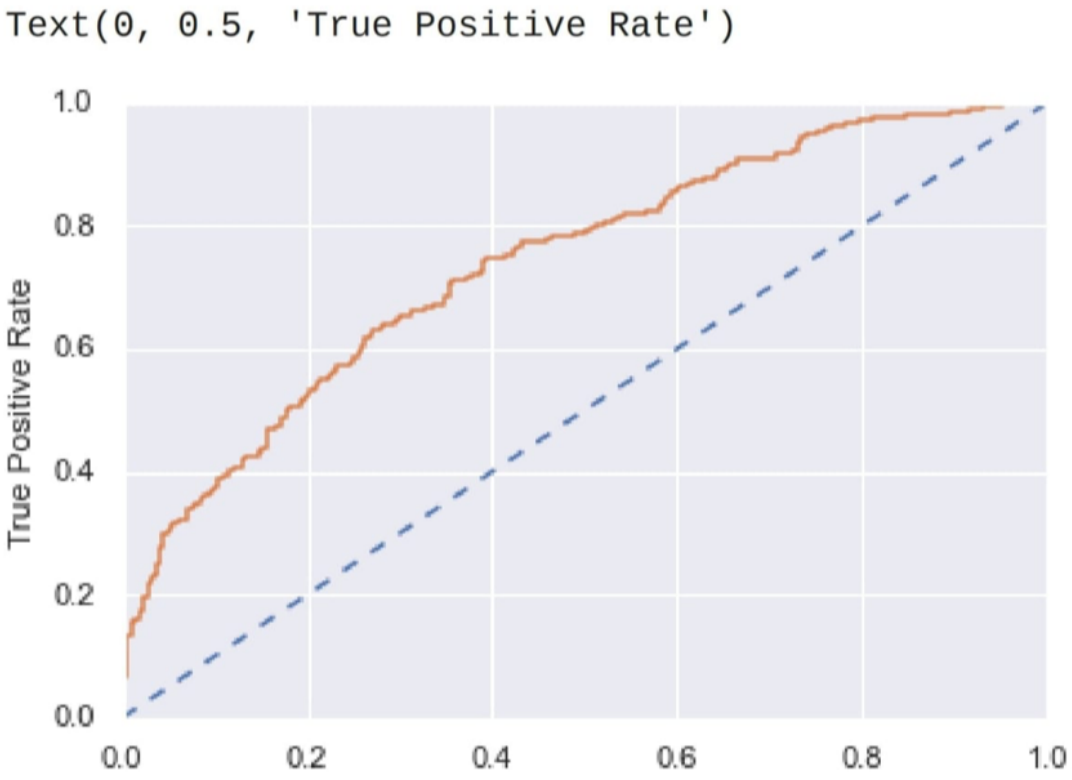


Fig: 24 LDA Model - AUC AND ROC for Training Data

Confusion Matrix for the Training Data

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7ff839f3cbb0>
```

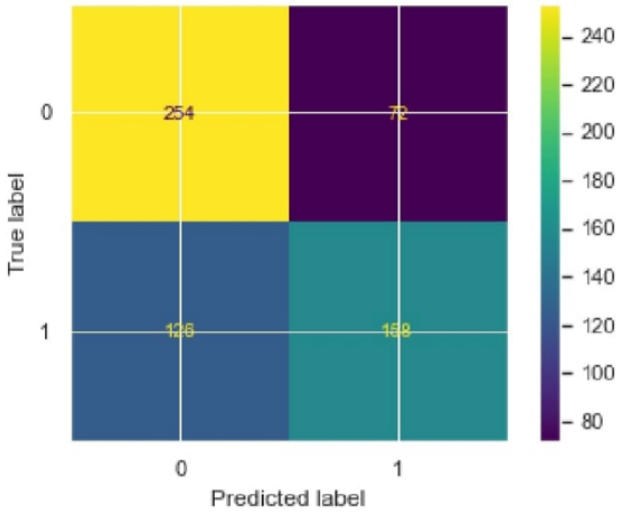


Fig: 25 LDA Model - Confusion Matrix for Training Data

Train Data Accuracy----------0.6754098360655738

Classification Report

	precision	recall	f1-score	support
0	0.67	0.78	0.72	326
1	0.69	0.56	0.61	284
accuracy			0.68	610
macro avg	0.68	0.67	0.67	610
weighted avg	0.68	0.68	0.67	610

TAB:21 CLASSIFICATION REPORT OF LDA ON TRAIN DATA

AUC and ROC for the Test Data-----0.705

```
[<matplotlib.lines.Line2D at 0x7ff8399e7ac0>]
```

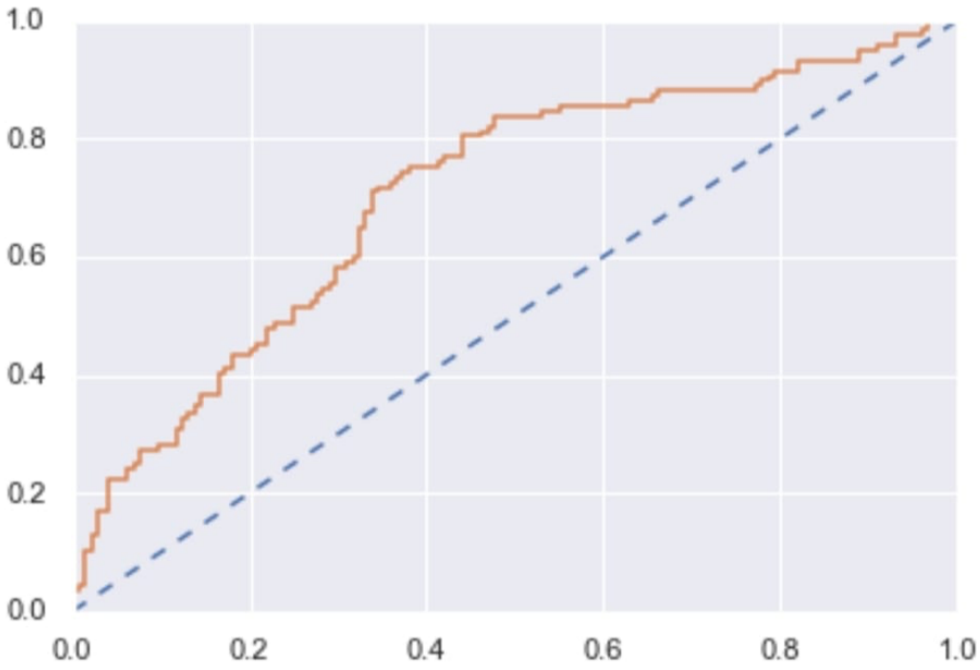


Fig: 26 LDA Model - AUC AND ROC for Test Data

Confusion Matrix for Test Data

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7ff839c7e640>
```

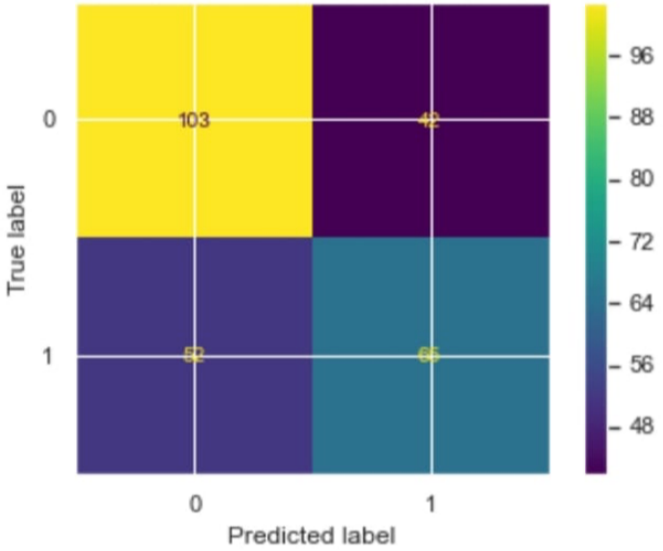


Fig: 27 LDA Model - Confusion Matrix for Test Data

Test Data Accuracy----------0.6412213740458015

Classifiaction Report

	precision	recall	f1-score	support
0	0.66	0.71	0.69	145
1	0.61	0.56	0.58	117
accuracy			0.64	262
macro avg	0.64	0.63	0.63	262
weighted avg	0.64	0.64	0.64	262

TAB:22 CLASSIFICATION REPORT OF LDA ON TEST DATA

LDA Conclusion:

Train Data:
AUC: 74%
Accuracy: 68%
Precision: 69%
Recall:56%
f1-Score: 61%

Test Data:
AUC: 70.5%
Accuracy: 64%
Precision: 61%
Recall:56%
f1-Score: 58%

Comparison of the performance metrics from the 2 models

	Logistic Regression Train	Logistic Regression Test	LDA Train	LDA Test
Accuracy	0.67	0.65	0.68	0.64
AUC	0.74	0.70	0.74	0.70
Recall	0.56	0.57	0.56	0.56
Precision	0.68	0.61	0.69	0.61
F1 Score	0.62	0.59	0.61	0.58

TAB:23 COMPARISON LOGISTIC REGRESSION VS LDA

ROC Curve for the 2 models on the Training data

```
<matplotlib.legend.Legend at 0x7ff83a593eb0>
```

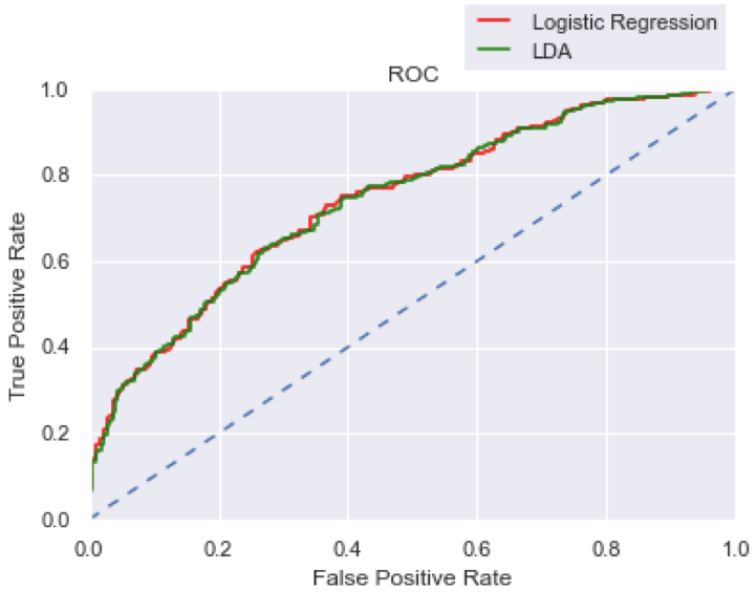


Fig: 28 ROC Curve for the 2 models on the Training data

ROC Curve for the 2 models on the Test data

```
<matplotlib.legend.Legend at 0x7ff839b8e820>
```

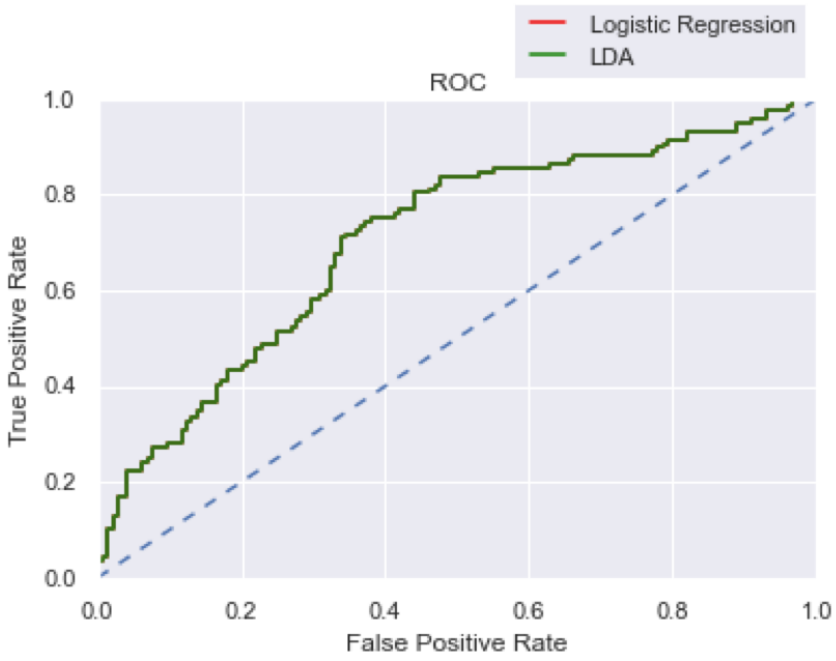


Fig: 29 ROC Curve for the 2 models on the Training data

Conclusion:

Out of 2 models Logistic Regression and LDA , LDA performs better as from the above table we conclude that accuracy , recall , precision and f1 score of LDA model is better than the Logistic Regression Model .Majorly the factors influencing the predictions are Salary, Age and Education.

Change the cut-off values for maximum accuracy for LDA Model :

We will do this exercise only on the training data.By doing this we are changing the default cutoff of 0.5 and try to check the model on various cutoff from 0.1 to 1, and find on which cutoff we get the best result of accuracy , precision , recall and f1 score and will that cut-off and make predictions on that cutoff to improve our perfomance

Result @ 0.4 cutoff

Accuracy Score 0.6656

F1 Score 0.6699

Recall Score 0.7289

Precision Score 0.6198

Confusion Matrix

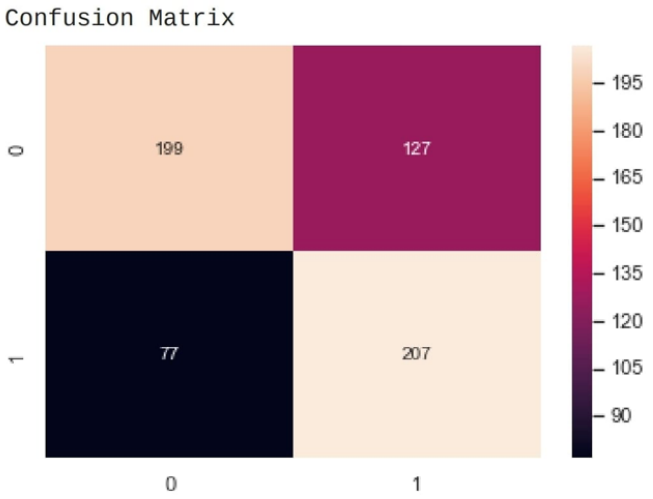


Fig: 30 LDA Custom Cutoff– Confusion Matrix

At cutoff 0.4 we get the better value for accuracy score , F1 score , recall & Precision.

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

Insights :

- In the given business problem we need to predict whether an employee would like to opt for a Holiday Package or not based on some given key factors.
- We have carried out some predictions using both Logistic Regression and Linear discriminant analysis And found that both are giving us similar results.
- The EDA analysis clearly indicates that Employees that are aged above 50 are not considering the Holiday packages available.
- Whereas employees in the range from 35 to 50 are generally opting for holiday packages and even if the Salary is less than 50000 they have opted for Holiday packages.
- Majorly the factors influencing the predictions are Salary, Age and Education.

Recommendations :

- For older age group they can add some new trip packages like Special packages to religious destinations and it might get the required traction
- Also for Employees having higher Salary package like above 150000, they can introduce some special offers on Holiday packages
- Employees having more number of Older children can be suggested and given some special packages with great discount for vacations.

Note : For more deatils please check the code .