Predictive Modelling Project Report

Nandha Keshore Utti
PG-DSBA Online
March’ 22

Date: 28/08/2022

Contents

List of figures

List of tables

Problem 1 (Linear Regression)

Problem Statement:

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.

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.

Exploratory Data Analysis:

- **Data description:**

Reading the data file and loading first five records:

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

Table. 01

Dataset information:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26967 entries, 0 to 26966
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  26967 non-null  int64
 1   carat       26967 non-null  float64
 2   cut         26967 non-null  object
 3   color       26967 non-null  object
 4   clarity     26967 non-null  object
 5   depth       26270 non-null  float64
 6   table       26967 non-null  float64
 7   x           26967 non-null  float64
 8   y           26967 non-null  float64
 9   z           26967 non-null  float64
 10  price       26967 non-null  int64
dtypes: float64(6), int64(2), object(3)
memory usage: 2.3+ MB
```

Table. 02

Interpretation:

- There are null values in 'depth' feature.
- Total 26967 records and 11 features are in the given dataset.
- There are no duplicated records.

Data types:

```
Unnamed: 0      int64
carat         float64
cut            object
color          object
clarity        object
depth         float64
table         float64
x             float64
y             float64
z             float64
price           int64
dtype: object
```

Table. 03

- There are 8 numeric features and 3 object features.

Dataset description:

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	26967.0	13484.00	7784.85	1.0	6742.50	13484.00	20225.50	26967.00
carat	26967.0	0.80	0.48	0.2	0.40	0.70	1.05	4.50
depth	26270.0	61.75	1.41	50.8	61.00	61.80	62.50	73.60
table	26967.0	57.46	2.23	49.0	56.00	57.00	59.00	79.00
x	26967.0	5.73	1.13	0.0	4.71	5.69	6.55	10.23
y	26967.0	5.73	1.17	0.0	4.71	5.71	6.54	58.90
z	26967.0	3.54	0.72	0.0	2.90	3.52	4.04	31.80
price	26967.0	3939.52	4024.86	326.0	945.00	2375.00	5360.00	18818.00

Table. 04

Interpretation:

- If we check 'Unnamed: 0' feature, they are representing number of records, for whi ch index column is sufficient. So, let's drop this feature and re-check the data dim ensions and interpret.

➢ **Data description: (after dropping 'Unnamed: 0' feature)**

Reading the data file and loading first five records:

	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

Table. 05

Dataset information:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26967 entries, 0 to 26966
Data columns (total 10 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   carat    26967 non-null  float64
 1   cut      26967 non-null  object
 2   color    26967 non-null  object
 3   clarity  26967 non-null  object
 4   depth    26270 non-null  float64
 5   table    26967 non-null  float64
 6   x        26967 non-null  float64
 7   y        26967 non-null  float64
 8   z        26967 non-null  float64
 9   price    26967 non-null  int64
dtypes: float64(6), int64(1), object(3)
memory usage: 2.1+ MB
```

Table. 06

Interpretation:

- There are null values in ‘depth’ feature.
- Total 26967 records and 10 features are in the given dataset.
- There are 34 duplicated records.

Data types:

```
carat      float64
cut         object
color       object
clarity     object
depth      float64
table      float64
x          float64
y          float64
z          float64
price        int64
dtype: object
```

Table. 07

- There are 7 numeric features and 3 object features.

Dataset description:

	count	mean	std	min	25%	50%	75%	max
carat	26967.0	0.80	0.48	0.2	0.40	0.70	1.05	4.50
depth	26270.0	61.75	1.41	50.8	61.00	61.80	62.50	73.60
table	26967.0	57.46	2.23	49.0	56.00	57.00	59.00	79.00
x	26967.0	5.73	1.13	0.0	4.71	5.69	6.55	10.23
y	26967.0	5.73	1.17	0.0	4.71	5.71	6.54	58.90
z	26967.0	3.54	0.72	0.0	2.90	3.52	4.04	31.80
price	26967.0	3939.52	4024.86	326.0	945.00	2375.00	5360.00	18818.00

Table. 08

Interpretation:

Let us interpret by each feature individually.

Numeric features:

1) **'Carat':** Mean weight of all zirconia cubes is 0.80 carats.

2) **'Depth':** Average height of zirconia cubes is 61.75.
3) **'Table':** Average width is 57.46.

4) **Parameter x, y, z:** Average length and width is same, i.e., 5.73. But, cube avg height is 3.54, less compared to length and width.

5) **'Price':**
   Average price is ~3939
   Max price is ~18818
   Min price is ~326

   - All the features have minimum variation except 'Carat' and 'Price'.

Object features:

6) **'Cut':** Total 5 kinds of variety cuts are available.

```
array(['Ideal', 'Premium', 'Very Good', 'Good', 'Fair'], dtype=object)
```

Fig. 01

   - Quality is increasing order Fair, Good, Very Good, Premium, Ideal.

7) **'Color':** Total 7 kinds of variety colors are available.

```
array(['E', 'G', 'F', 'D', 'H', 'J', 'I'], dtype=object)
```

Fig. 02

- With D being the worst and J the best.

8) 'Clarity': Total 8 kinds of variety cuts are available.

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

Fig. 03

- In order from Worst to Best in terms of average price: IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1

➢ **Data pre-processing:**

Null treatment:

- There are null values in 'depth' feature.

```
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])
```

Fig. 04

- Let us treat the null values by imputing it with mean.

```
<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    26967 non-null  float64
 5   table    26967 non-null  float64
 6   x        26967 non-null  float64
 7   y        26967 non-null  float64
 8   z        26967 non-null  float64
 9   price    26967 non-null  int64  
dtypes: float64(6), int64(1), object(3)
memory usage: 2.1+ MB
```

Table. 09

- Null values treated successfully.

Duplicated records check:

- There are 34 duplicated records as shown below.

	carat	cut	color	clarity	depth	table	x	y	z	price
4756	0.35	Premium	J	VS1	62.4	58.0	5.67	5.64	3.53	949
6215	0.71	Good	F	SI2	64.1	60.0	0.00	0.00	0.00	2130
8144	0.33	Ideal	G	VS1	62.1	55.0	4.46	4.43	2.76	854
8919	1.52	Good	E	I1	57.3	58.0	7.53	7.42	4.28	3105
9818	0.35	Ideal	F	VS2	61.4	54.0	4.58	4.54	2.80	906
10473	0.79	Ideal	G	SI1	62.3	57.0	5.90	5.85	3.66	2898
10500	1.00	Premium	F	VVS2	60.6	54.0	6.56	6.52	3.96	8924
12894	1.21	Premium	D	SI2	62.5	57.0	6.79	6.71	4.22	6505
13547	0.43	Ideal	G	VS1	61.9	55.0	4.84	4.86	3.00	943
13783	0.79	Ideal	G	SI1	62.3	57.0	5.90	5.85	3.66	2898
14389	0.60	Premium	D	SI2	62.0	57.0	5.43	5.35	3.34	1196
14410	1.00	Very Good	D	SI1	63.1	56.0	6.34	6.30	3.99	5645
15798	0.90	Very Good	I	VS2	58.4	62.0	6.29	6.35	3.69	3334
16852	0.79	Ideal	G	SI1	62.3	57.0	5.90	5.85	3.66	2898
17263	1.04	Premium	I	SI2	62.0	57.0	6.53	6.47	4.03	3774
18025	1.51	Good	I	SI1	63.8	57.0	7.21	7.18	4.59	6046
18777	0.32	Premium	H	VS2	60.6	58.0	4.47	4.44	2.70	648
18837	1.01	Premium	H	VS1	61.2	61.0	6.44	6.41	3.93	5294
19731	0.30	Good	J	VS1	63.4	57.0	4.23	4.26	2.69	394
19877	2.01	Premium	I	VS2	60.3	62.0	8.13	8.08	4.89	15939
20301	0.30	Ideal	H	SI1	62.2	57.0	4.26	4.29	2.66	450
20760	1.80	Ideal	H	VS1	62.3	56.0	7.79	7.76	4.84	15105

Table. 10

- It can be inferred that duplicated records are significant to keep for analysis.

Anomalies:

- No anomalies are observed.

- **Data visualization:**

Univariate analysis:

- Let's visualize all the numeric columns using hist plot and check the distribution natur e of the features.

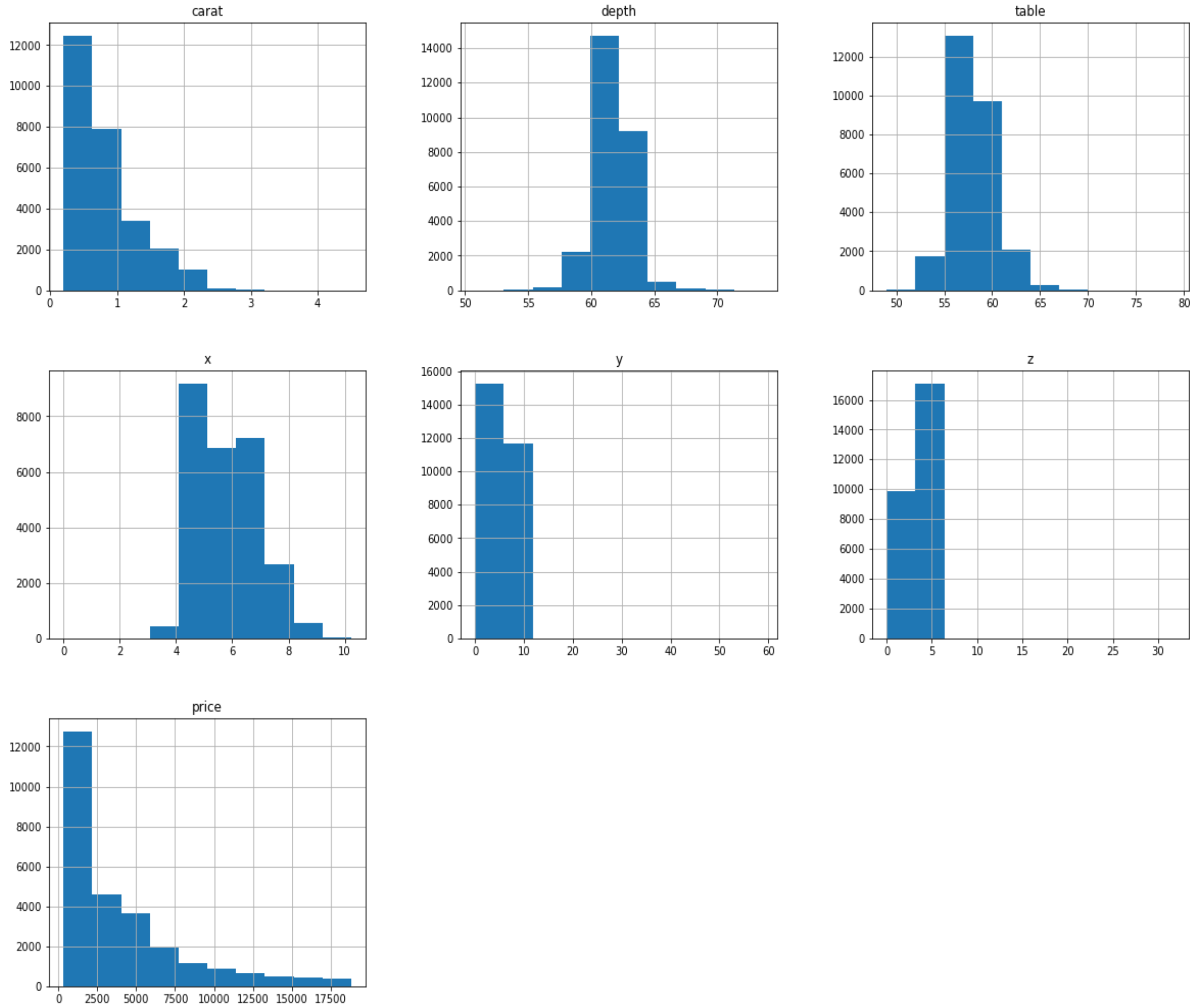

Fig. 05

Checking skewness:

```
carat     1.12
depth    -0.03
table     0.77
x         0.39
y         3.85
z         2.57
price     1.62
dtype: float64
```

Table. 11

Interpretations:

- Normally distributed features: 'depth', 'x'
- Moderately right skewed features: 'table'
- Highly right skewed features: 'carat', 'y', 'z', 'price'

Outliers check:

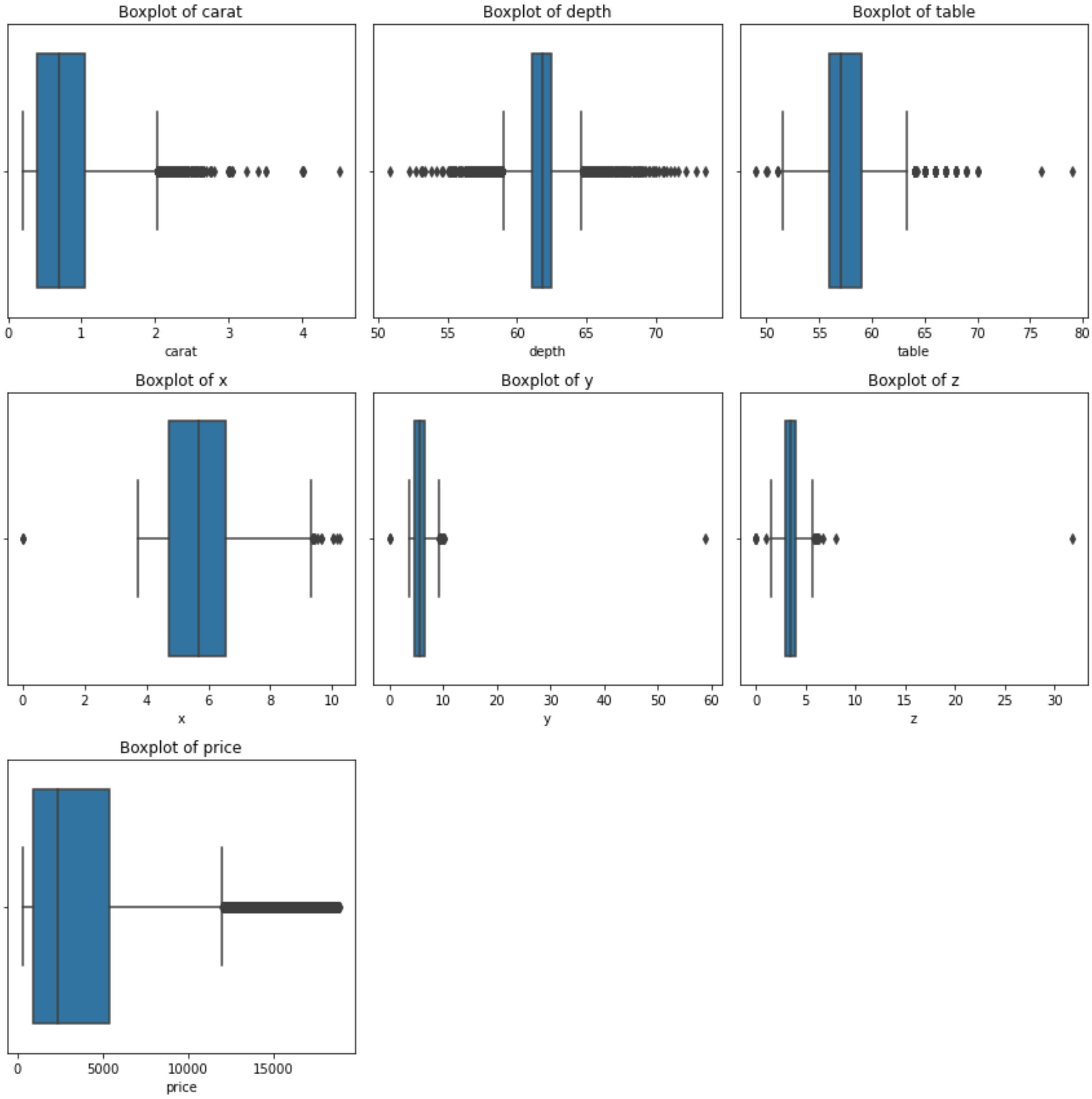

Fig. 06

Interpretations:

- Every feature has outliers and need to be treated

Outlier treatment:

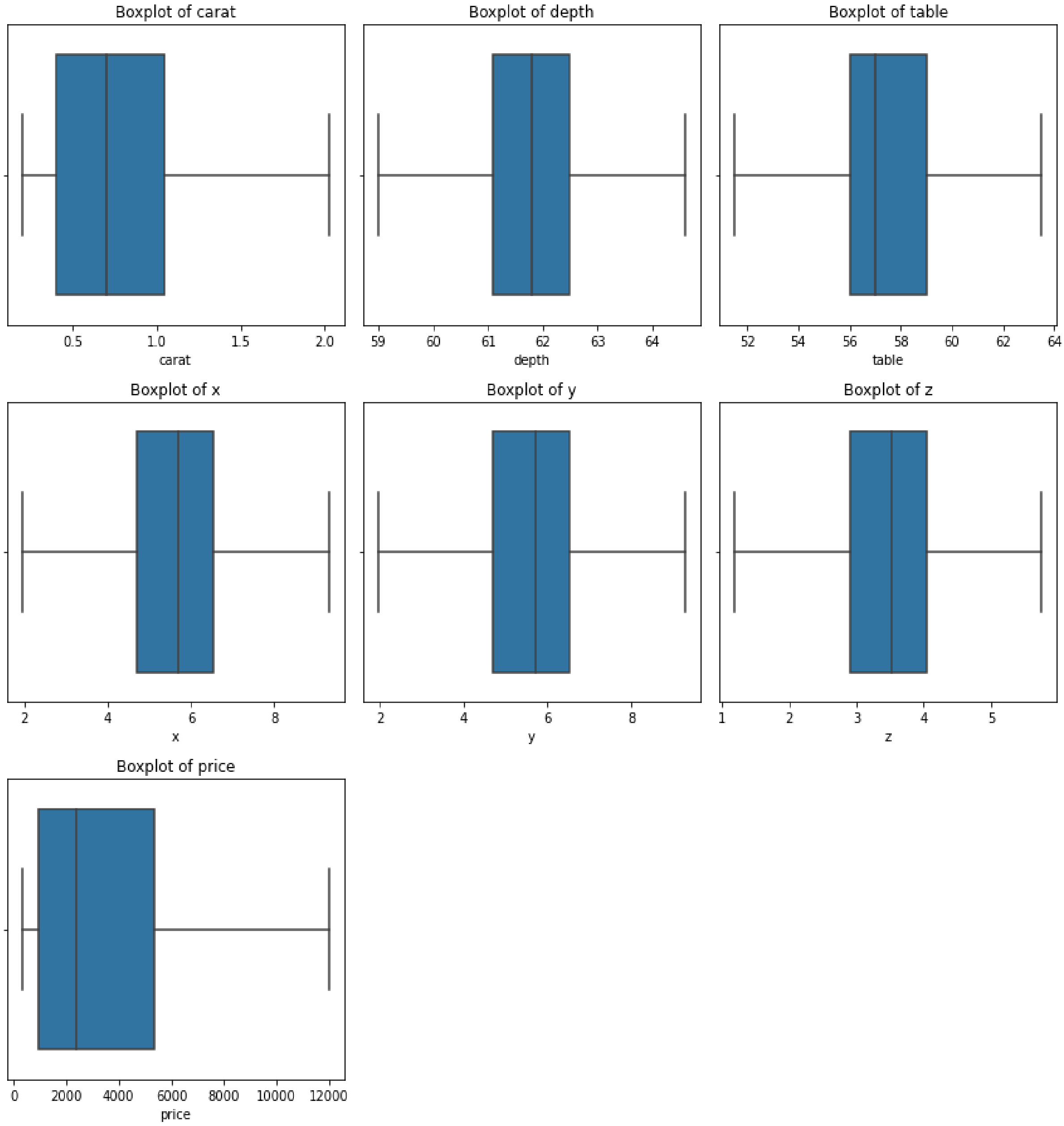

Fig. 07

- Outliers treated successfully.

Let us visualize the categorical variables and its classes: (by using count plot)

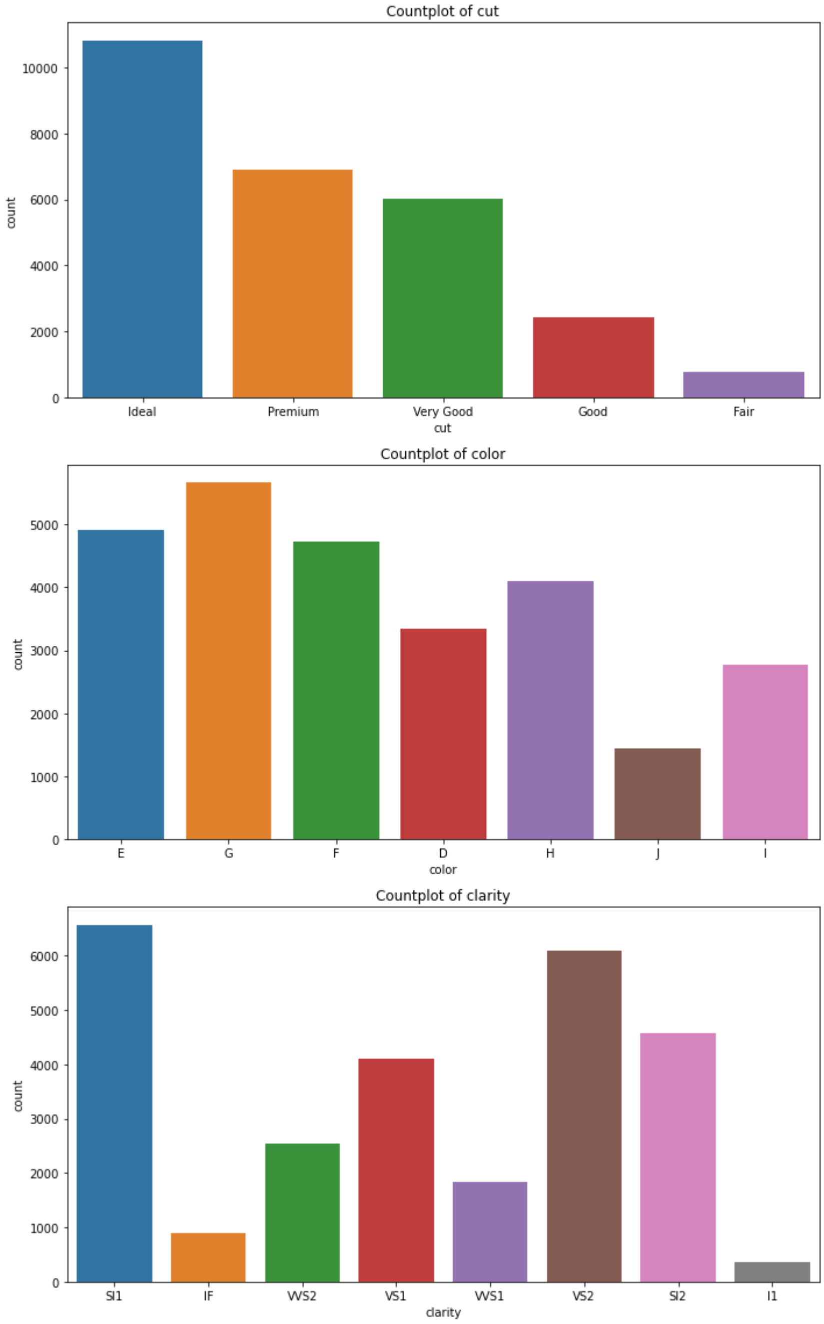

Fig. 08

Interpretation:

1) ‘Cut’: ‘Ideal’ cut zirconia cubes are maximum and ‘Fair’ cut zirconia cubes are minim um
2) ‘Color’: ‘G’ color zirconia cubes are maximum and ‘J’ color zirconia cubes are minim um
3) ‘Clarity’: ‘SI1’ clarity zirconia cubes are maximum and ‘I1’ clarity zirconia cubes are minimum.

Bivariate analysis:

- Let's plot the pair plot and heatmap to check correlation b/w the data features

Pair plot:

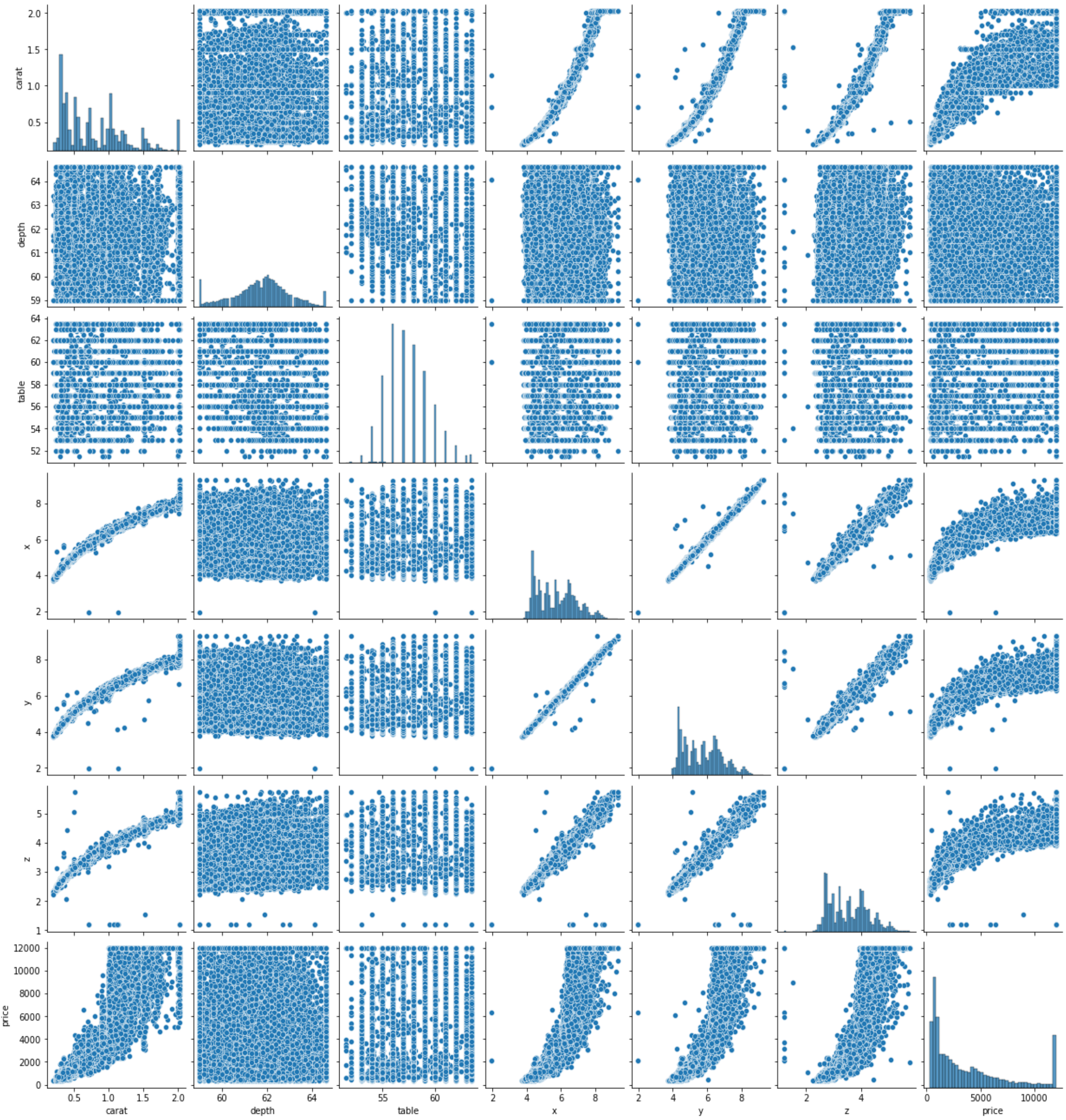

Fig. 09

Heatmap:

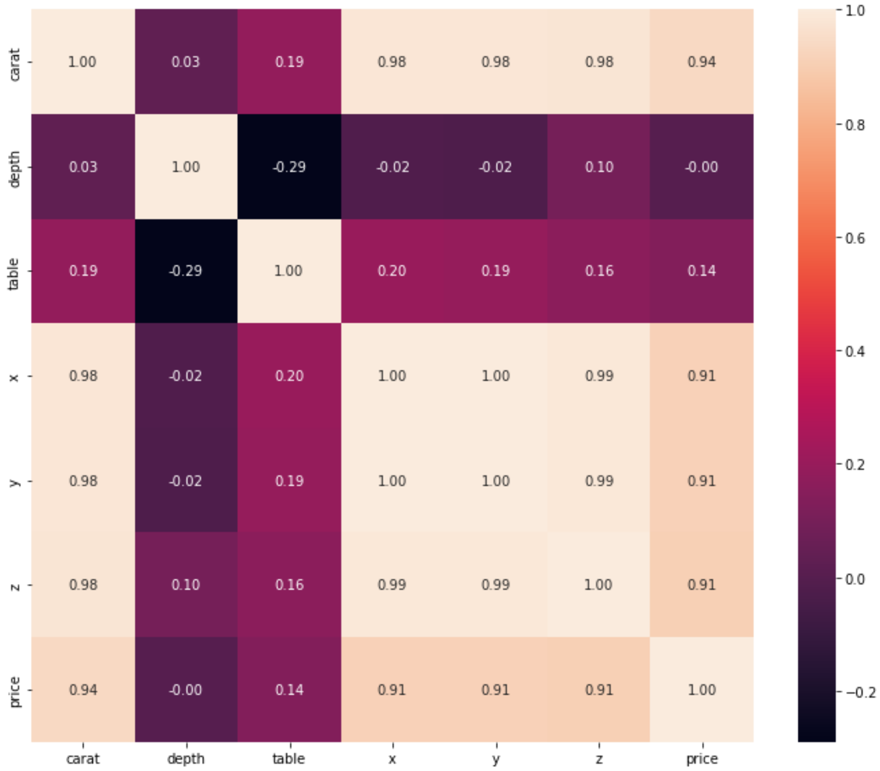

Fig. 10

Interpretation (From both pairplot and heatmap):

- There is high correlation of price with carat, x, y, z

Let us visualize the available categorical variables vs price:

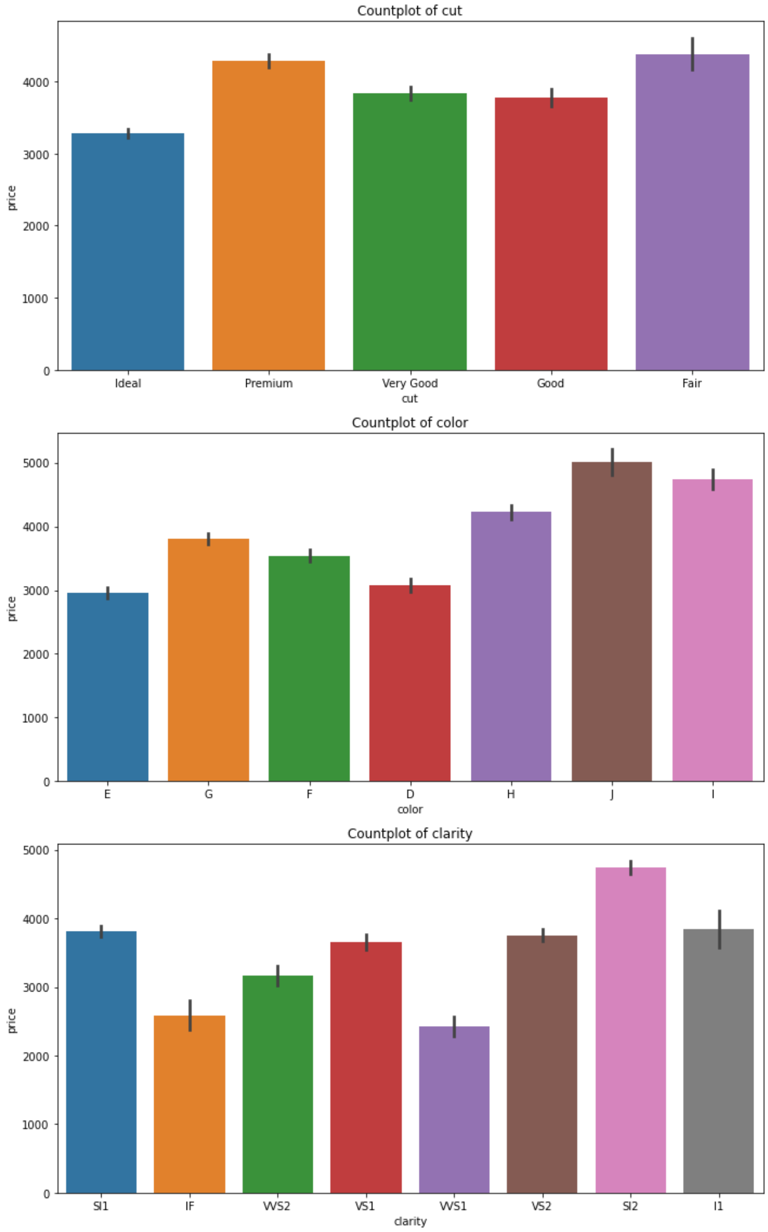

Fig. 11

Interpretation:

1) ‘Cut’: ‘Premium’ & ‘Fair’ cut zirconia cubes have maximum average price (approximately) and ‘Ideal’ cut zirconia cubes have minimum average price.
2) ‘Color’: ‘J’ color zirconia cubes have maximum average price and ‘E’ & ‘D’ color zirconia cubes have minimum average price.
3) ‘Clarity’: ‘SI2’ clarity zirconia cubes have maximum average price and ‘VVS1’ clarity zirconia cubes have minimum average price.

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

Null values analysis and treatment:

- There are null values in 'depth' feature.

```
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])
```

Fig. 12

- There are no zero values.

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

Table. 12

- It has 697 null values in depth feature. Sample is shown below:

	carat	cut	color	clarity	depth	table	x	y	z	price
26	0.34	Ideal	D	SI1	NaN	57.0	4.50	4.44	2.74	803
86	0.74	Ideal	E	SI2	NaN	59.0	5.92	5.97	3.52	2501
117	1.00	Premium	F	SI1	NaN	59.0	6.40	6.36	4.00	5292
148	1.11	Premium	E	SI2	NaN	61.0	6.66	6.61	4.09	4177
163	1.00	Very Good	F	VS2	NaN	55.0	6.39	6.44	3.99	6340
...	...	...	...	...	...	...	...	...	...	...
26848	1.22	Very Good	H	VS1	NaN	59.0	6.91	6.85	4.29	7673
26854	1.29	Premium	I	VS2	NaN	58.0	7.12	7.03	4.27	6321
26879	0.51	Very Good	E	SI1	NaN	58.0	5.10	5.13	3.12	1343
26923	0.51	Ideal	D	VS2	NaN	57.0	5.12	5.09	3.18	1882
26960	1.10	Very Good	D	SI2	NaN	63.0	6.76	6.69	3.94	4361

697 rows × 10 columns

Table. 13

- We can see, null value records have significance.
- So, let us treat the null values by imputing it with mean.

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

Table. 14

- Null values treated successfully.

Categorical variables' sub-levels analysis:

- Out of three categorical features available, all three are ordinal categorical variables.
- Let us check the possibility of combining these sub-levels by taking price as factor.

1) 'Cut':

- Average price of different cut varieties is as shown below:

```
cut
Fair         4377.907810
Good         3773.801721
Ideal        3282.754993
Premium      4284.055443
Very Good    3832.066003
Name: price, dtype: float64
```

Table. 15

Interpretations:

- We can club 'fair' and 'premium' cut classes into one category as they have very minimum difference (31) in average price and name it as common category ‘Premium’

Note: Quality is increasing order Fair, Good, Very Good, Premium, Ideal. Average price of each cut variety and quality are contradictory as per the given data.

- We can also club 'Good' and 'Very Good' cut classes into one category as they have very minimum difference (104) in average price and name it as common category ‘Good’

Now, let us check the new categories defined and its average price:

```
cut
Good       3815.276591
Ideal      3282.754993
Premium    4293.599544
Name: price, dtype: float64
```

Table. 16

- Now, we can clearly differentiate sub-levels appropriately for ‘cut’ variable as per average price.

2) ‘Color’:

- Average price of different color varieties is as shown below:

```
color
D    3069.740580
E    2957.119890
F    3538.160922
G    3809.397191
H    4228.015968
I    4736.916997
J    5009.480596
Name: price, dtype: float64
```

Table. 17

Interpretation:

- We can club 'D' and 'E' color classes into one category as they have very minimum difference (112) in average price and name it as common category ‘E’

Note: In original dataset, D being the worst and J the best. After clubbing, E being the worst and J the best.

Now, let us check the new categories defined and its average price:

```
color
E    3002.708026
F    3538.160922
G    3809.397191
H    4228.015968
I    4736.916997
J    5009.480596
Name: price, dtype: float64
```

Table. 18

- Now, we can clearly differentiate sub-levels appropriately for 'color' variable as per average price.

3) 'Clarity':

- Average price of different clarity varieties is as shown below:

```
clarity
I1      3843.109589
IF      2588.392617
SI1     3814.173337
SI2     4746.349945
VS1     3653.389812
VS2     3750.660108
VVS1    2424.598695
VVS2    3168.175030
Name: price, dtype: float64
```

Table. 19

Interpretation:

- We can club 'VS1' and 'VS2' clarity classes into one category as they have very minimum difference () in average price and name it as common category 'VS'
- We can club 'I1' into 'SI1' clarity classes into one category as they have very minimum difference () in average price

Note: In original dataset, in order from Worst to Best in terms of avg price: IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1. In new dataset, in order from Worst to Best in terms of avg price: IF, VVS1, VVS2, VS, SI1, SI2

Now, let us check the new categories defined and its average price:

```
clarity
IF      2588.392617
SI1     3815.696078
SI2     4746.349945
VS      3711.597380
VVS1    2424.598695
VVS2    3168.175030
Name: price, dtype: float64
```

Table. 20

- Now, we can clearly differentiate sub-levels appropriately for ‘clarity’ variable as per average price.

We have successfully clubbed some sub-levels in the existing categorical variables by taking ‘price’ as main factor.

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 stats-model. Create multiple models and check the performance of Predictions on Train and Test sets using R-square, RMSE & Adj R-square. Compare these models and select the best one with appropriate reasoning

Data Encoding:

Sample data frame after encoding:

	carat	depth	table	x	y	z	price	cut_Ideal	cut_Premium	clarity_SI1	clarity_SI2	clarity_VS	clarity_VVS1	clarity_VVS2	color_F	color_G
0	0.30	62.1	58.0	4.27	4.29	2.66	499.0	1	0	1	0	0	0	0	0	0
1	0.33	60.8	58.0	4.42	4.46	2.70	984.0	0	1	0	0	0	0	0	0	1
2	0.90	62.2	60.0	6.04	6.12	3.78	6289.0	0	0	0	0	0	0	1	0	0
3	0.42	61.6	56.0	4.82	4.80	2.96	1082.0	1	0	0	0	1	0	0	1	0
4	0.31	60.4	59.0	4.35	4.43	2.65	779.0	1	0	0	0	0	1	0	1	0

Table. 21

- Shape of encoded data frame is (26967, 19)

Data Splitting:

- Let us split the data into train and test set in 70:30 ratio.
- Shapes of spitted train and test sets are as shown below:

```
Shape of X_train is (18876, 18)
Shape of y_train is (18876, 1)
Shape of X_test is (8091, 18)
Shape of y_test is (8091, 1)
```

Fig. 13

Linear Regression model building using sklearn:

- Train dataset is fit into the Linear Regression model.
- Coefficients of each feature are as shown below:

```
The coefficient for carat is 9100.022010627665
The coefficient for depth is -51.44629690492242
The coefficient for table is -38.761583252547595
The coefficient for x is -1846.7167323400931
The coefficient for y is 1539.2890640405776
The coefficient for z is -275.3966645445477
The coefficient for cut_Ideal is 120.82067054560483
The coefficient for cut_Premium is 58.491834270812774
The coefficient for clarity_SI1 is -1518.8486949902524
The coefficient for clarity_SI2 is -2191.6206490924437
The coefficient for clarity_VS is -752.3996333316974
The coefficient for clarity_VVS1 is -186.98331617970882
The coefficient for clarity_VVS2 is -211.0758449633182
The coefficient for color_F is -171.2358689228729
The coefficient for color_G is -296.6699179961387
The coefficient for color_H is -723.8071514639645
The coefficient for color_I is -1189.9203494466656
The coefficient for color_J is -1764.9417436727822
```

Fig. 14

Interpretation: ‘carat’ has highest weightage among all and ‘table’ has the least weightage.

- Intercept of the model is 6093.44

Performance metrics:

Train dataset:

1) Accuracy score (R^2) – 0.932
2) Adjusted R^2 – 0.932
3) RMSE value – 907.11

Test dataset:

1) Accuracy score (R^2) – 0.929
2) Adjusted R^2 – 0.929
3) RMSE value – 921.24

Linear Regression model building using stats-model (by OLS):

- Train and test dataset which are used for sklearn model are used combined for stats-model
- Combined dataset fit by using OLS method.
- Parameters obtained are as below:

Intercept	6093.443970
carat	9100.022011
depth	-51.446297
table	-38.761583
x	-1846.716732
y	1539.289064
z	-275.396665
cut_Ideal	120.820671
cut_Premium	58.491834
clarity_SI1	-1518.848695
clarity_SI2	-2191.620649
clarity_VS	-752.399633
clarity_VVS1	-186.983316
clarity_VVS2	-211.075845
color_F	-171.235869
color_G	-296.669918
color_H	-723.807151
color_I	-1189.920349
color_J	-1764.941744
dtype: float64	

Table. 22

Interpretation:

- Coefficients and intercept obtained are matching with parameters obtained by sklearn.
- Here also, ‘carat’ has highest weightage among all and ‘table’ has the least weightage.
- Other results by OLS regression are as below:

OLS Regression Results

Dep. Variable:	price	R-squared:	0.932
Model:	OLS	Adj. R-squared:	0.932
Method:	Least Squares	F-statistic:	1.436e+04
Date:	Fri, 26 Aug 2022	Prob (F-statistic):	0.00
Time:	16:31:11	Log-Likelihood:	-1.5533e+05
No. Observations:	18876	AIC:	3.107e+05
Df Residuals:	18857	BIC:	3.109e+05
Df Model:	18		
Covariance Type:	nonrobust		

Fig. 15

Performance metrics:

1) Accuracy score (R^2) – 0.932
2) Adjusted R^2 – 0.932
3) RMSE value – 907.11

Actual price vs Predicted price graph:

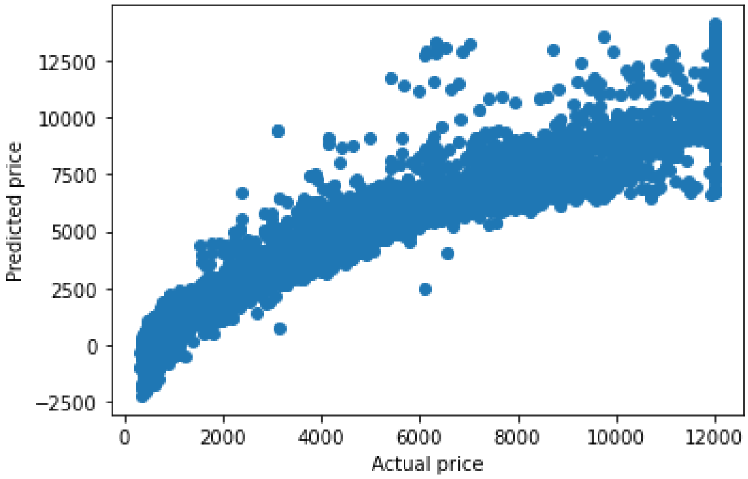

Fig. 16

- We can observe that, there is linear relationship between the Actual vs Predicted

Linear equation for this model is as below:

```
(6093.44) * Intercept +(9100.02) * carat +(-51.45) * depth +(-38.76) * table +(-1846.72) * x +(1539.29) * y +(-275.4) * z +(12
0.82) * cut_Ideal +(58.49) * cut_Premium +(-1518.85) * clarity_SI1 +(-2191.62) * clarity_SI2 +(-752.4) * clarity_VS +(-186.98)
* clarity_VVS1 +(-211.08) * clarity_VVS2 +(-171.24) * color_F +(-296.67) * color_G +(-723.81) * color_H +(-1189.92) * color_I +
(-1764.94) * color_J +
```

Fig. 17

Ridge Model:

- Train dataset is fit into the Ridge model.
- Coefficients of each feature are as shown below:

```
The coefficient for carat is 9078.143357280667
The coefficient for depth is -51.31668904471771
The coefficient for table is -38.77118771639951
The coefficient for x is -1818.671274950101
The coefficient for y is 1518.6877024092414
The coefficient for z is -273.50240021007045
The coefficient for cut_Ideal is 120.38502650076374
The coefficient for cut_Premium is 57.20041345154347
The coefficient for clarity_SI1 is -1517.2651209498488
The coefficient for clarity_SI2 is -2189.7251097957637
The coefficient for clarity_VS is -750.6975905950619
The coefficient for clarity_VVS1 is -184.67627461749106
The coefficient for clarity_VVS2 is -208.9816599531266
The coefficient for color_F is -171.0092826641814
The coefficient for color_G is -296.2632282091902
The coefficient for color_H is -723.1242905043367
The coefficient for color_I is -1188.757929021815
The coefficient for color_J is -1763.1762621677842
```

Fig. 18

Interpretation:

- 'carat' has highest weightage among all and 'table' has the least weightage.
- Intercept of the model is 6093.44

Performance metrics:

Train dataset:

1) Accuracy score (R^2) – 0.932
2) Adjusted R^2 – 0.932
3) RMSE value – 907.11

Test dataset:

4) Accuracy score (R^2) – 0.929
5) Adjusted R^2 – 0.929
6) RMSE value – 921.15

Lasso Model:

- Train dataset is fit into the Lasso model.
- Coefficients of each feature are as shown below:

```
The coefficient for carat is 9060.730734866442
The coefficient for depth is -52.5915032862889
The coefficient for table is -39.45164849840563
The coefficient for x is -1516.6994372195247
The coefficient for y is 1226.8727897794213
The coefficient for z is -280.610472729601
The coefficient for cut_Ideal is 112.82386603427734
The coefficient for cut_Premium is 38.256997151327774
The coefficient for clarity_SI1 is -1507.848565856717
The coefficient for clarity_SI2 is -2180.962004191366
The coefficient for clarity_VS is -740.7264757393582
The coefficient for clarity_VVS1 is -171.4428348904038
The coefficient for clarity_VVS2 is -196.686801454295
The coefficient for color_F is -169.13322443294498
The coefficient for color_G is -293.91070459831917
The coefficient for color_H is -721.0279158164391
The coefficient for color_I is -1185.5712284847384
The coefficient for color_J is -1758.1133698270135
```

Fig. 19

Interpretation:

- 'carat' has highest weightage among all and 'table' and 'cut_Premium' have the least weightage.
- Intercept of the model is 6148.61

Performance metrics:

Train dataset:

1) Accuracy score (R^2) – 0.932
2) Adjusted R^2 – 0.932
3) RMSE value – 907.25

Test dataset:

7) Accuracy score (R^2) – 0.929
8) Adjusted R^2 – 0.929
9) RMSE value – 921.83

Comparison of the models and conclusion:

- We have built total 4 models and all are having same R^2 score, adjusted R^2 score and RMSE values for the both train and test datasets.
- ‘carat’ feature is having the highest weightage and ‘cut_premium’ is having the lowest weightage in all the models.
- So, it can be concluded that we follow any of the model for the business analysis.

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

Business insights:

Based on the EDA analysis,

- It is clear that premium cut brings the maximum profit to the company, additionally, the ideal and good cuts are not bringing any profit to the company.
- SI2 clarity cube has high demand in the market.
- The colors H, I and J bring in profit whereas the other colors don’t.
- Price has high positive correlation with ‘carat’, dimensions of the cube (x, y, z)

Recommendations:

- From LR models, we can see carat of a zirconia cube has highest weightage. So, company should focus on giving high quality of cube to the customers. It affects the price and so business.
- Company should focus on carat and clarity of the stone to increase pricing and thereby the profit.
- Good customer base and marketing strategy needs tobe adopted to attract customers to buy the stones which gives more profit.

Problem 2 (Logistic Regression and LDA)

Problem statement:

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.

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

Exploratory Data Analysis:

- **Data description:**

Reading the data file and loading first five records:

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

Table. 23

- 'Unnamed: 0' has no significance, so let us drop this column.

Dataset data types:

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

Table. 24

- There are 5 numeric and 2 object type features.

Dataset information:

```
<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
```

Table. 25

Interpretation:

- There are no null values.
- There is total 872 records and 7 features in the dataset.
- Dataset is small
- There are no duplicated records.

Dataset description:

	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

Table. 26

Insights:

Let's describe each feature below:

1) **'Salary':** Variation is high in this feature with mean salary of ~47729
2) **'Age':** Employees age is b/w 20 and 62. There is no variation in this feature.
3) **'educ':** Average formal education of employees is ~9.3 years. There is slight variatio n in this feature.
4) **'no_young_children':** Employees are having 0 to 3 number of young children. Amon g them, majority of the employees are having zero number of young children.
5) **'no_older_children':** Employees are having 0 to 6 number of young children. Appro ximately, half majority of employees are having no older children and another half ma jority of employees are having one and two older children.

➢ Data pre-processing:

- Null treatment not required as there are no null values.
- Duplicated records treatment not required as there are no duplicated records.
- No anomalies are observed.

➢ Data visualization:

Univariate analysis:

- Let’s visualize all the numeric columns using hist plot and check the distribution natur e of the features.

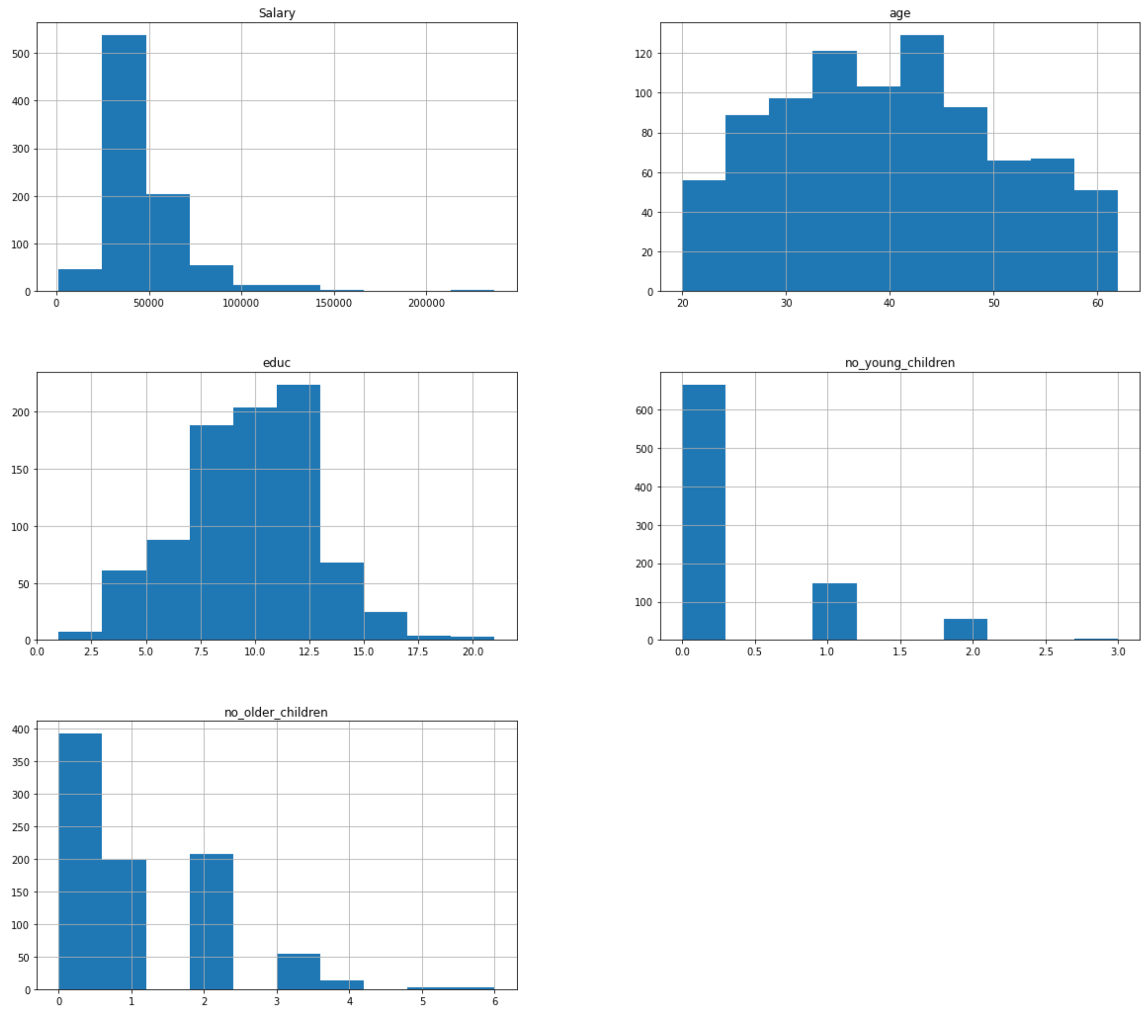

Fig. 20

Checking skewness:

```
Salary                 3.10
age                    0.15
educ                  -0.05
no_young_children      1.95
no_older_children      0.95
dtype: float64
```

Table. 27

Interpretation:

- Normally distributed features: ‘age’, ‘educ’
- Moderately right skewed features: ‘no_older children’
- Highly right skewed features: ‘salary’, ‘no_young children’

Outliers’ check:

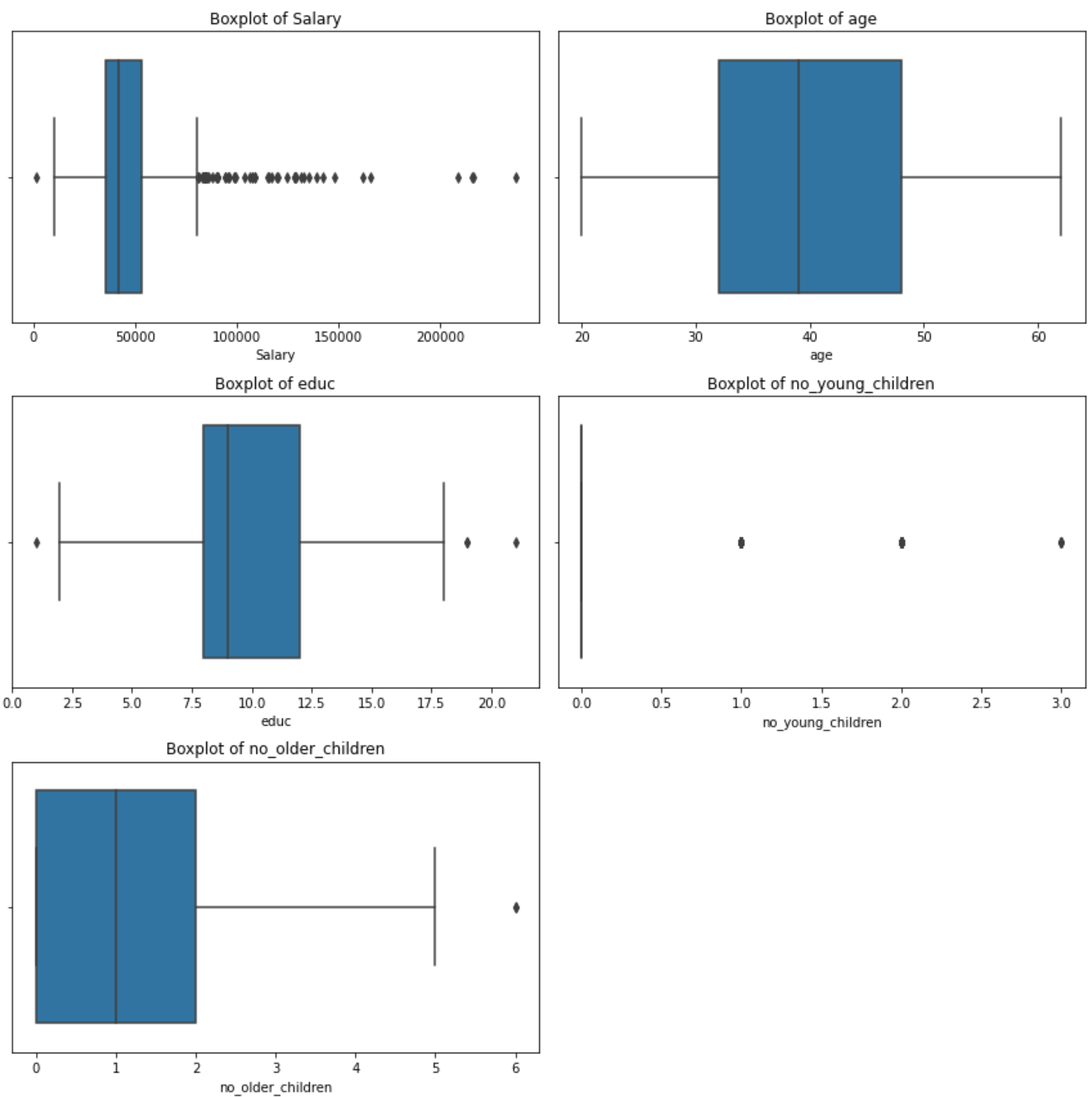

Fig. 21

Interpretation:

- Outliers are present in every feature.
- Although outliers exists as per the boxplot, by looking at the data distribution in describe(), categorical variables data will be lost and model will not be appropriate in view of different classes available in different categorical variables available.
- So, outliers are not treated in this case.

Count plots of categorical variables:

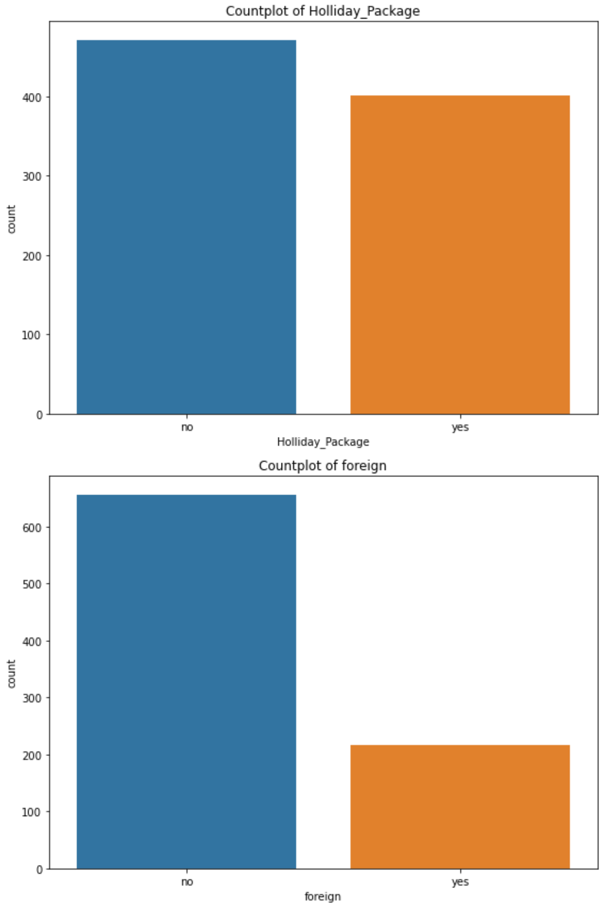

Fig. 22

Interpretation:

Holiday package:

- Percentage of employees opted for holiday package is 0.46
- percentage of employees not opted for holiday package is 0.54

Foreign:

- Percentage of foreign employees is 0.25
- Percentage of non-foreign employees is 0.75

Bivariate analysis:

- Let's plot the pair plot and heatmap to check correlation b/w the data features

Pair plot:

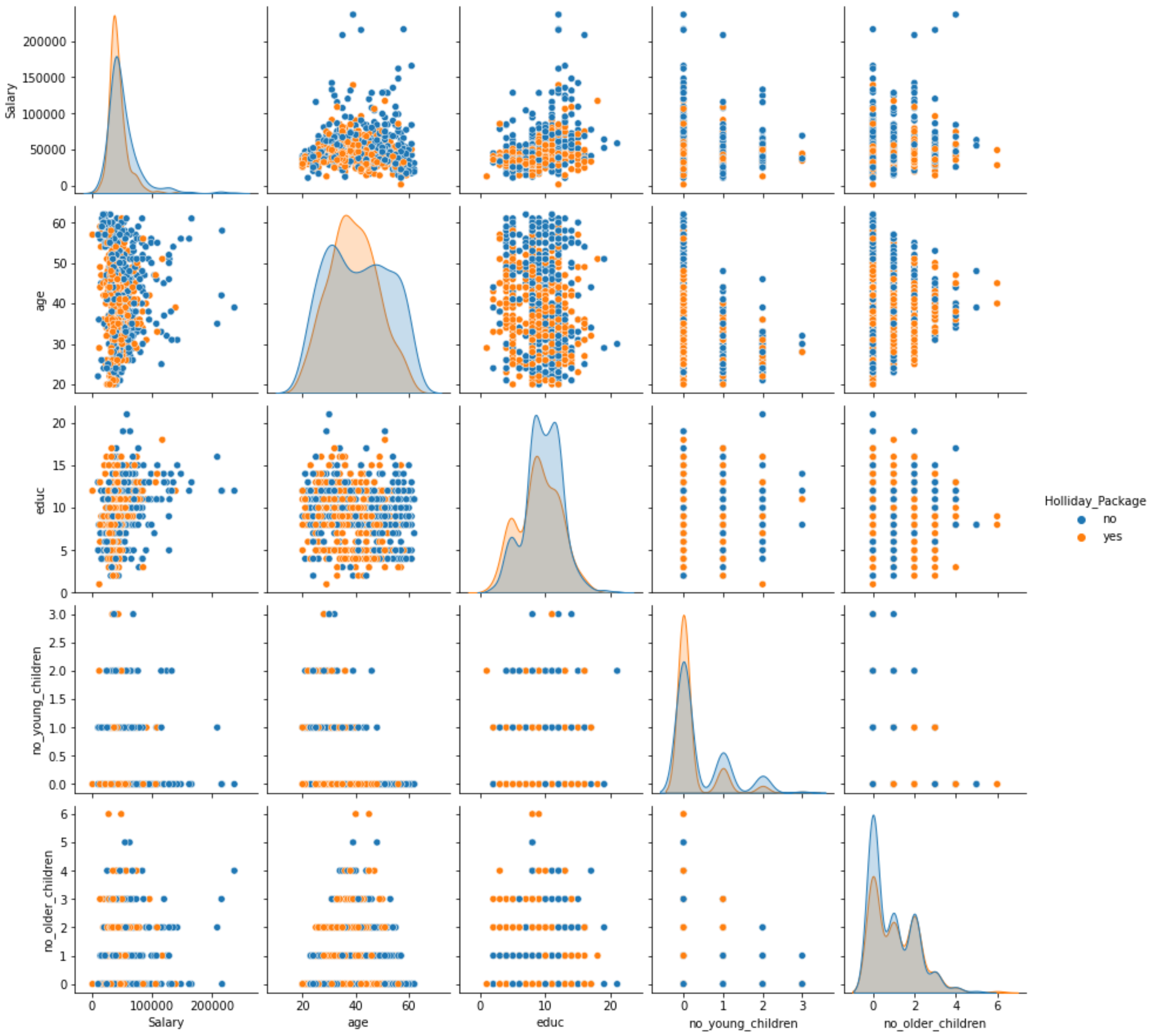

Fig. 23

Heatmap:

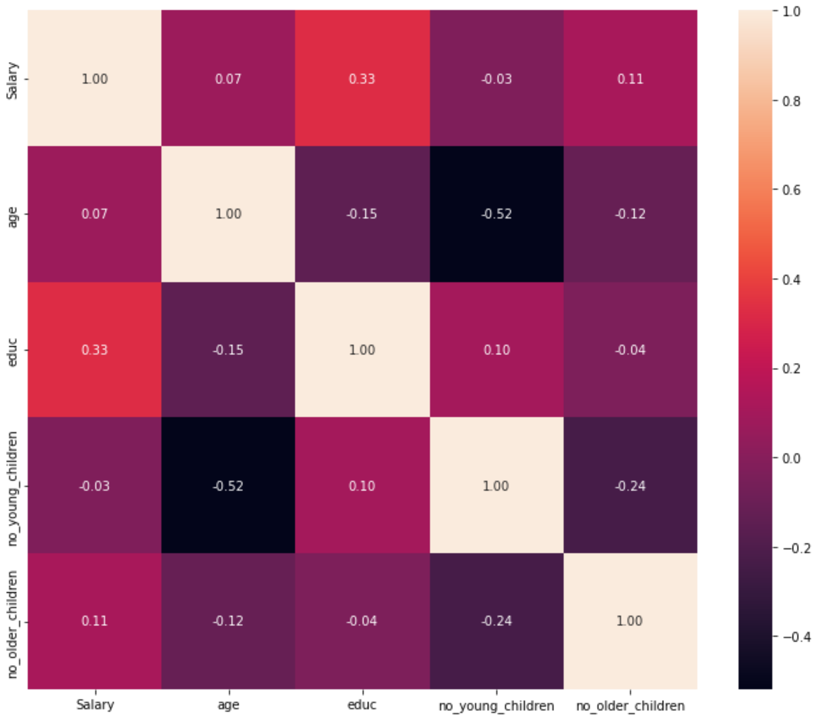

Fig. 24

Insights (From both pairplot and heatmap):

- There is no appreciable correlation between the features

Holiday package vs Foreign:

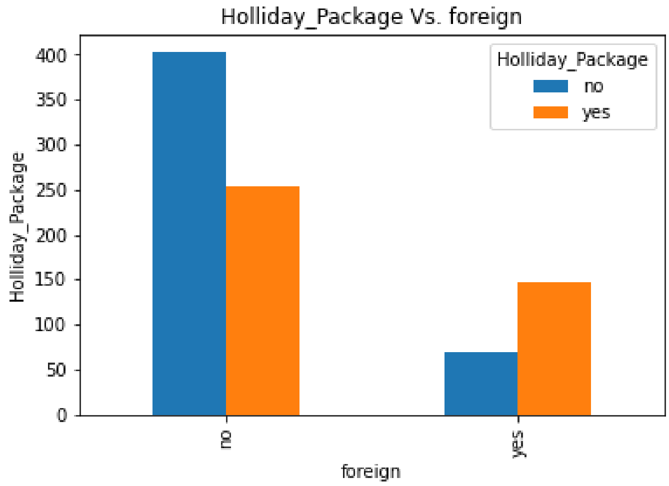

Fig. 25

Interpretation:

- Overall, Non-foreign employees are opting for holiday package more.
- Among non-foreign employees, percentage of employees not willing to opt for holida y package is more.

- Among foreign employees, percentage of employees willing to opt for holiday packag e is more.

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).

- **Data encoding:**

Converting object data into categorical/numercial data:

- Let us encode the target variable i.e, ‘Holliday_Package’ by using ‘LabelEncoder’ method.
- Sample dataset after encoding target variable is shown below:

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

Table. 28

- Now, let us encode the other categorical variable i.e., ‘foreign’ by ‘one hot’ encoding method.
- Sample dataset after encoding target variable is shown below:

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

Table. 29

- Dataset encoded successfully.

Splitting the data into Train and Test set:

- Target variable is ‘Holliday_Package’
- Let’s drop ‘Holliday_Package’ variable for train dataset and pop it for test dataset
- Dataset splitting is done with 30% test dataset and 70% train dataset.
- Let’s check the shapes of splitted dataset

```
X1_train (610, 6)
X1_test (262, 6)
y1_train (610, 1)
y1_test (262, 1)
```

Fig. 26

1) Logistic Regression model:

- Let us fit the train dataset by using ‘Logistic Regression’ model.

```
LogisticRegression(class_weight='balanced', max_iter=10000)
```

Fig. 27

2) LDA model:

- Let us fit the train dataset by using ‘Logistic Regression’ model.

```
LinearDiscriminantAnalysis()
```

Fig. 28

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.

1) Logistic Regression model:

- Train and test datasets are predicted using defined Logistic Regression model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
array([[222, 107],
       [176, 105]], dtype=int64)
```

Fig. 29

- Classification report:

	precision	recall	f1-score	support
0	0.56	0.67	0.61	329
1	0.50	0.37	0.43	281
accuracy			0.54	610
macro avg	0.53	0.52	0.52	610
weighted avg	0.53	0.54	0.53	610

Fig. 30

- Accuracy score is 53.61%
- ROC_AUC score is 0.566
- ROC curve:

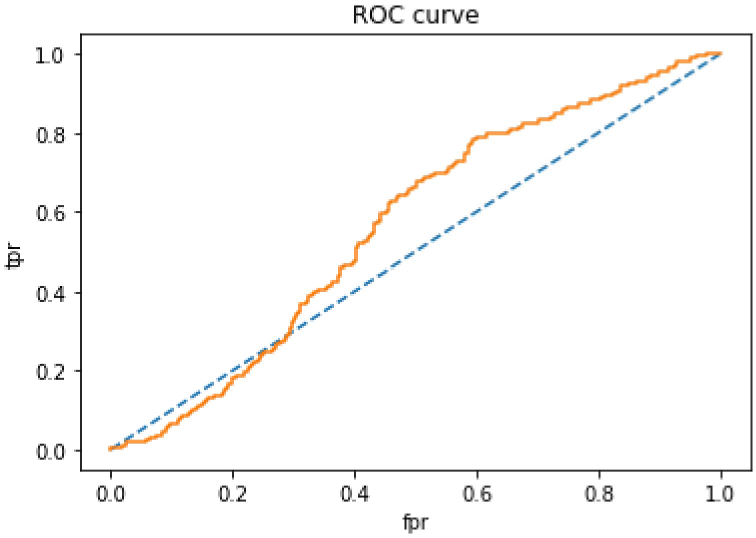

Fig. 31

Test dataset:

- Confusion matrix:

```
array([[86, 56],
       [61, 59]], dtype=int64)
```

Fig. 32

- Classification report:

	precision	recall	f1-score	support
0	0.59	0.61	0.60	142
1	0.51	0.49	0.50	120
accuracy			0.55	262
macro avg	0.55	0.55	0.55	262
weighted avg	0.55	0.55	0.55	262

Fig. 33

- Accuracy score is 55.34%
- ROC_AUC score is 0.599
- ROC curve:

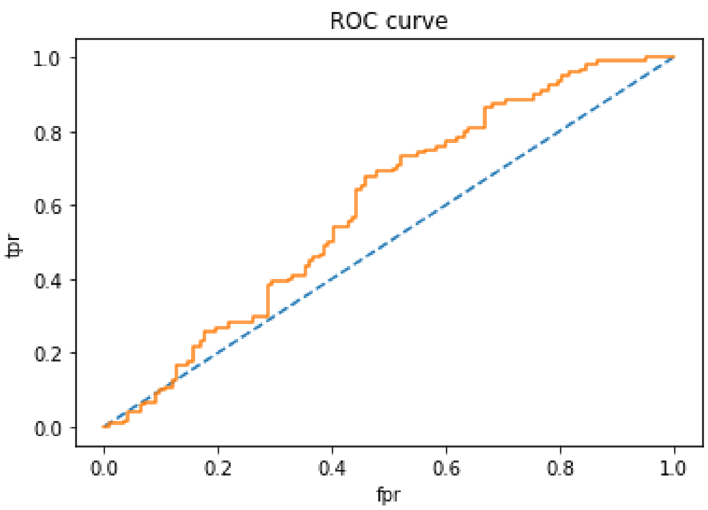

Fig. 34

2) LDA Model:

- Train and test datasets are predicted using defined Linear Discriminant Analysis model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
array([[243,  86],
       [119, 162]], dtype=int64)
```

Fig. 35

- Classification report:

```
              precision    recall  f1-score   support

           0       0.67      0.74      0.70       329
           1       0.65      0.58      0.61       281

    accuracy                           0.66       610
   macro avg       0.66      0.66      0.66       610
weighted avg       0.66      0.66      0.66       610
```

Fig. 36

- Accuracy score is 66.39%
- ROC_AUC score is 0.733
- ROC curve:

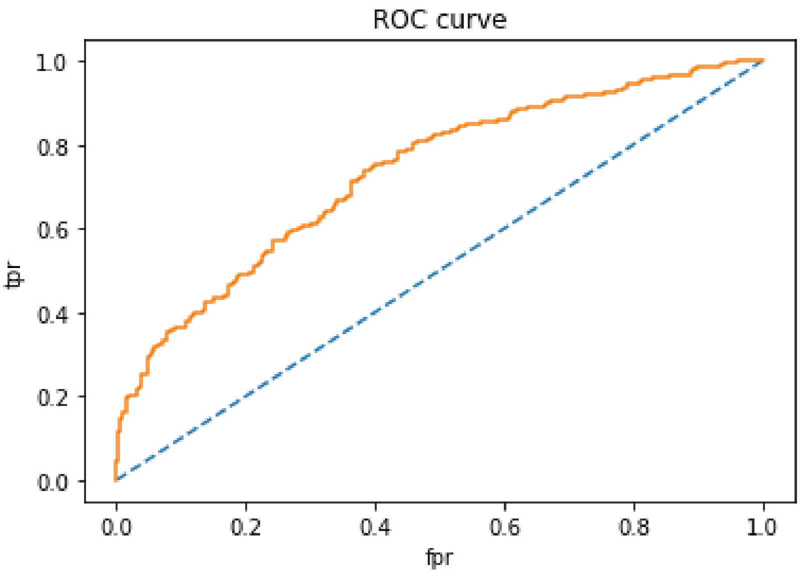

Fig. 37

Test dataset:

- Confusion matrix:

```
array([[109,  33],
       [ 61,  59]], dtype=int64)
```

Fig. 38

- Classification report:

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

Fig. 39

- Accuracy score is 64.12%
- ROC_AUC score is 0.714
- ROC curve:

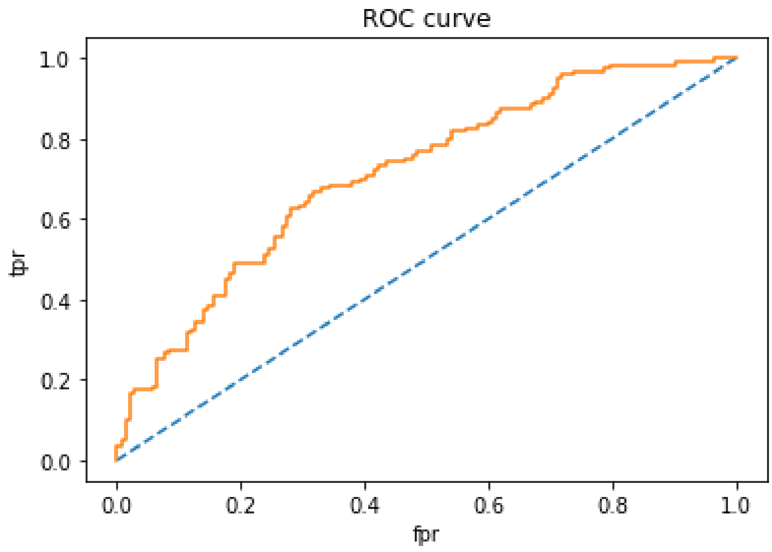

Fig. 40

Comparison of the models and conclusion:

- LDA is best model for this case study compared to Logistic Regression model.
- Dataset is small which is one of the main drawbacks of Logistic Regression model building, so LDA is giving better classification and good accuracy score.
- ROC-AUC score also better for LDA compared to Logistic Regression model.
- Precision, Recall, F1 score etc, all the parameters are better for LDA than Logistic Regression model.

2.4. Inference: Based on the whole Analysis, what are the business insights and recommendations.

Business insights:

From EDA analysis,

- Employees who high salary are opting for holiday package more compared employees who have less salary. This is expected trend.
- Less aged employees are preferring holiday package more compared to more aged employees.
- Employees who have a lesser number of young children are preferring holiday package and employees who have a lesser number of older children have a reverse trend.

Note: Comparison which is done in the above insights is from pair plot visualization. And difference between each class of target variable for the above considered features is less.

- Non-foreign employees are opting for holiday package more.
- Among non-foreign employees, percentage of employees not willing to opt for holida y package is more.

- Among foreign employees, percentage of employees willing to opt for holiday package is more

Recommendations:

- First of all, more data need to be collected for the better analysis and further study, by which accuracy also can be improved furthermore.
- Money is the prior most asset in opting for the package. So, less salaried employees should be given special consideration. It is better for company to have different policies based on salary which in turn will affect in encouraging less salaried employees to opt for the holiday package.
- There is slight drop in ~40 years aged employees opting for package compared to compared to ~30 and ~50 years. This can be considered as a special case.
- Employees who have young and old children are less opting for the package. This might be due to their children education and etc. So, holiday packages can be encouraged for these employees.

THE END