Predictive Modelling Project

Name: Swetha Kunapuli

Batch & Course: PGP-DSBA Online June Batch

Date: 23/11/2021

Table of Contents

List of Tables:

List of Figures:

PROBLEM 1

Executive Summary:

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.

cubic_zirconia.csv

Data Introduction:

The purpose of this whole exercise is to explore the dataset and is recommended for learning and practicing our skills using Linear Regression Model.

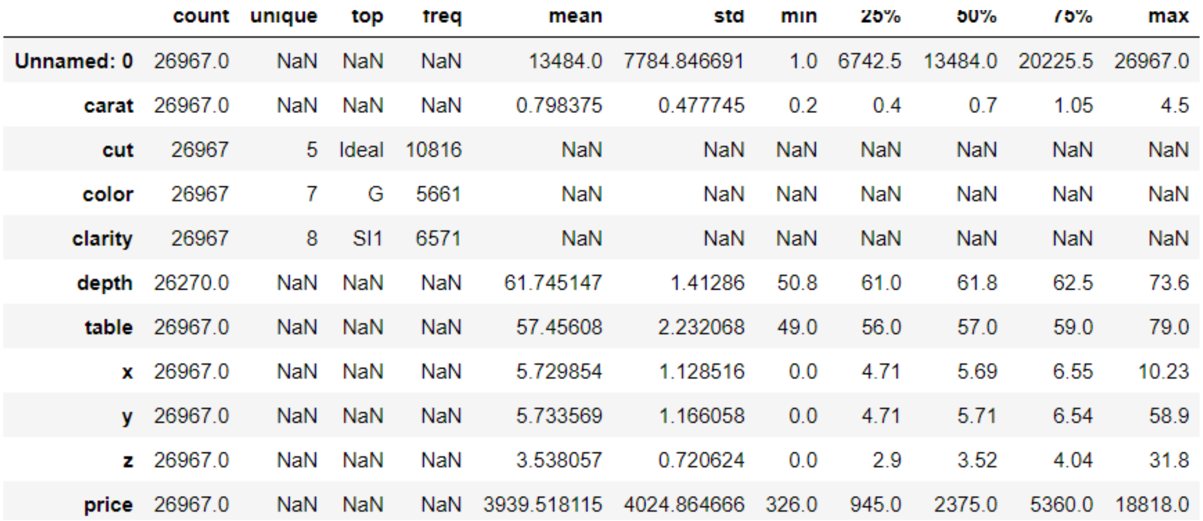
The dataset contains 26967 rows and 11 columns.

Data Description:

Description of variables is as follows:

Variable Name	**Description**
Carat	Carat weight of the cubic zirconia.
Cut	Describe the cut quality of the cubic zirconia. Quality is increasing order Fair, Good, Very Good, Premium, Ideal.

Color	Colour of the cubic zirconia. With D being the worst and J the best.
Clarity	Clarity refers to the absence of the Inclusions and Blemishes. (In order from Worst to Best in terms of avg price) IF, VVS1, VVS2, VS1, VS2, Sl1, Sl2, l1
Depth	The Height of cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter.
Table	The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter.
Price	the Price of the cubic zirconia.
X	Length of the cubic zirconia in mm.
Y	Width of the cubic zirconia in mm.
Z	Height of the cubic zirconia in mm.



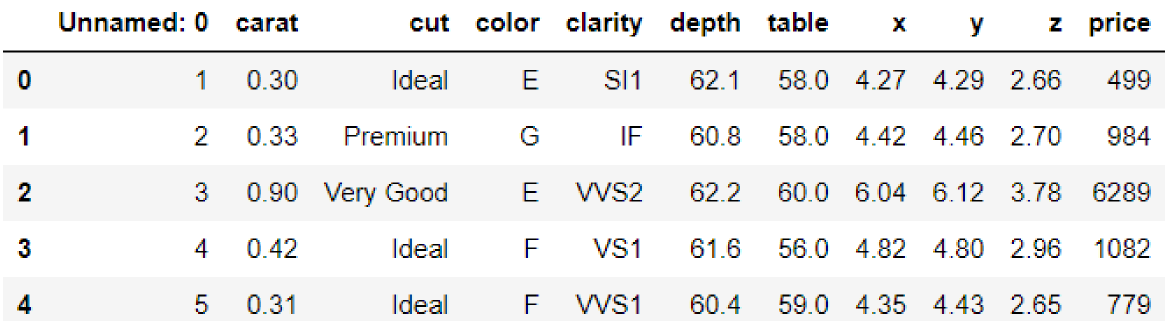
	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Unnamed: 0	26967.0	NaN	NaN	NaN	13484.0	7784.846691	1.0	6742.5	13484.0	20225.5	26967.0
carat	26967.0	NaN	NaN	NaN	0.798375	0.477745	0.2	0.4	0.7	1.05	4.5
cut	26967	5	Ideal	10816	NaN	NaN	NaN	NaN	NaN	NaN	NaN
color	26967	7	G	5661	NaN	NaN	NaN	NaN	NaN	NaN	NaN
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

Table 1.0

Observation:

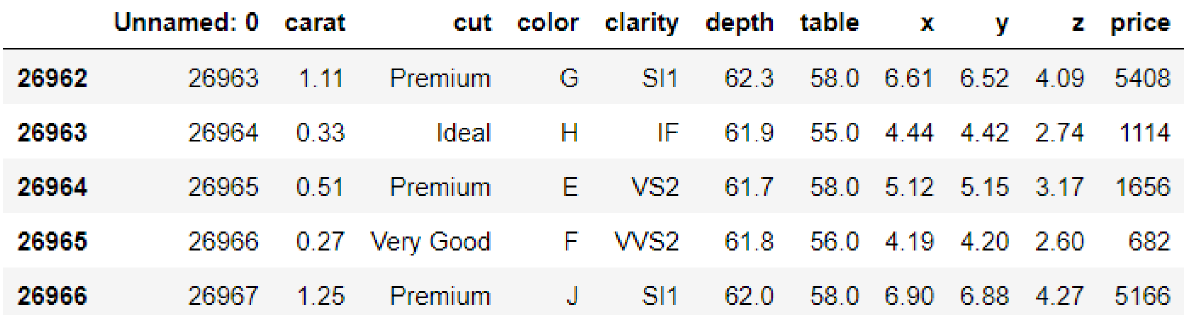
We have both categorical and continuous data, for categorical data we have cut, colour and clarity for continuous data we have carat, depth, table, x. y, z and price will be target variable.

Sample 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

Table 1.1



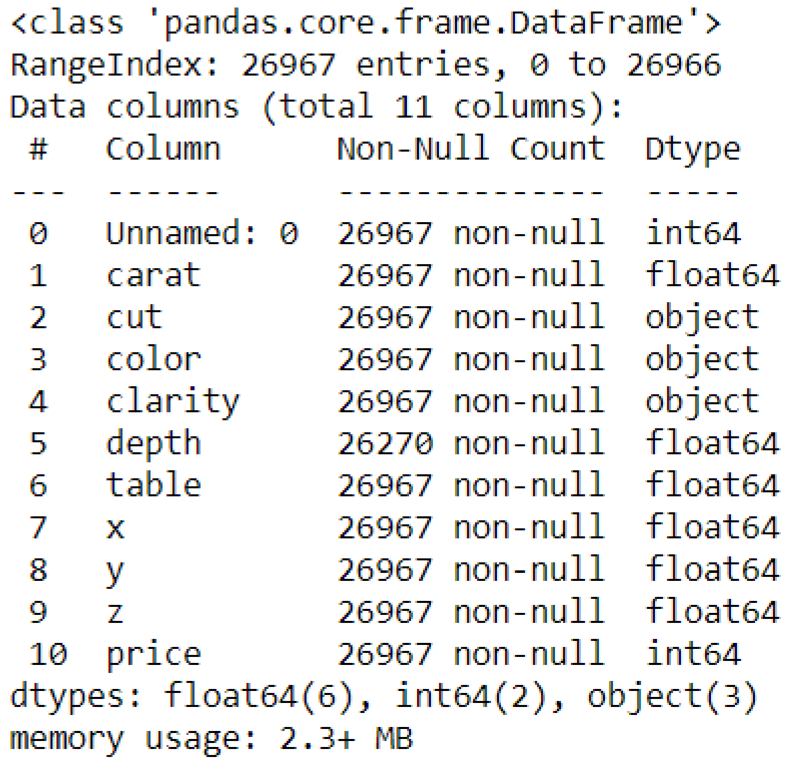
	Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
26962	26963	1.11	Premium	G	SI1	62.3	58.0	6.61	6.52	4.09	5408
26963	26964	0.33	Ideal	H	IF	61.9	55.0	4.44	4.42	2.74	1114
26964	26965	0.51	Premium	E	VS2	61.7	58.0	5.12	5.15	3.17	1656
26965	26966	0.27	Very Good	F	VVS2	61.8	56.0	4.19	4.20	2.60	682
26966	26967	1.25	Premium	J	SI1	62.0	58.0	6.90	6.88	4.27	5166

Table 1.2

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:

Check for types of variables in the data frame:



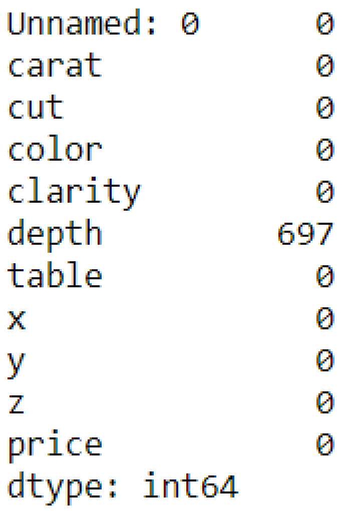
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26967 entries, 0 to 26966
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  26967 non-null  int64
 1   carat       26967 non-null  float64
 2   cut         26967 non-null  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 1.1.1

Observation:

11 variables and 26976 records. We have float, int and object data types in the given data.

Check for missing values in the dataset:



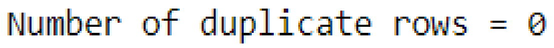
```
Unnamed: 0      0
carat           0
cut             0
color           0
clarity         0
depth         697
table           0
x               0
y               0
z               0
price           0
dtype: int64
```

Table 1.1.2

Observation:

We have 697 missing values in the given data.

Check for duplicate values in the dataset:



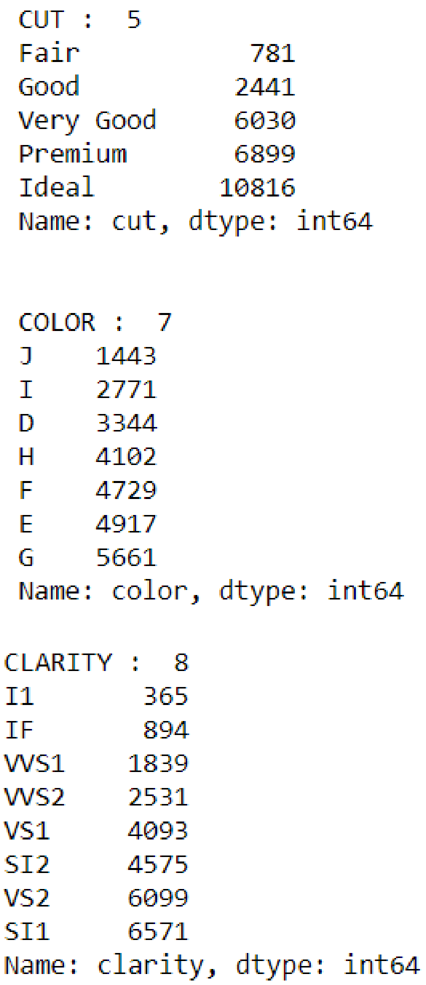
```
Number of duplicate rows = 0
```

Table 1.1.3

Observation:

No duplicate values in the dataset

Check for unique values for categorical variables



```
CUT :  5
Fair            781
Good           2441
Very Good      6030
Premium        6899
Ideal         10816
Name: cut, dtype: int64
```

```
COLOR :  7
J     1443
I     2771
D     3344
H     4102
F     4729
E     4917
G     5661
Name: color, dtype: int64
```

```
CLARITY :  8
I1       365
IF       894
VVS1    1839
VVS2    2531
VS1     4093
SI2     4575
VS2     6099
SI1     6571
Name: clarity, dtype: int64
```

Table 1.1.4

Univariate Analysis:

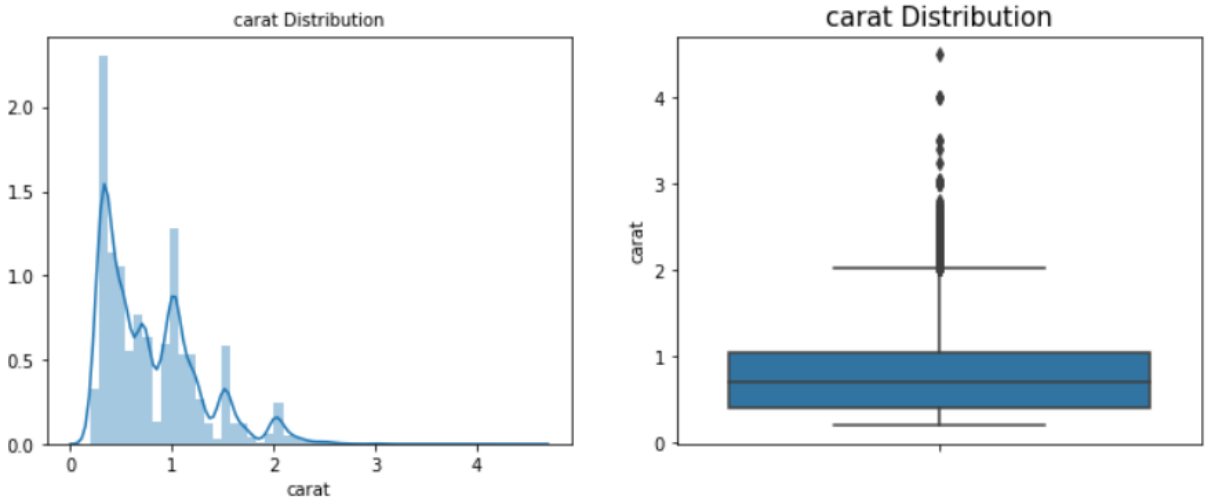


Fig 1.1.1

The distribution of data in carat seems to positively skewed, as there are multiple peaks points in the distribution there could multimode, and the box plot of carat seems to have large number of outliers. In the range of 0 to 1 where majority of data lies.

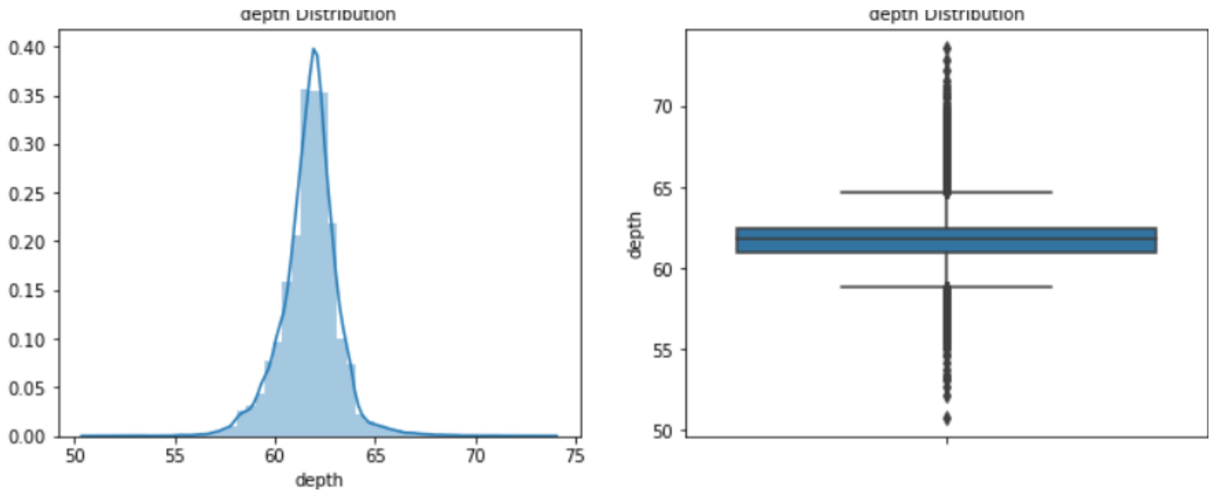


Fig 1.1.2

The distribution of depth seems to be normal distribution.

The depth ranges from 55 to 65.

The box plot of the depth distribution holds many outliers.

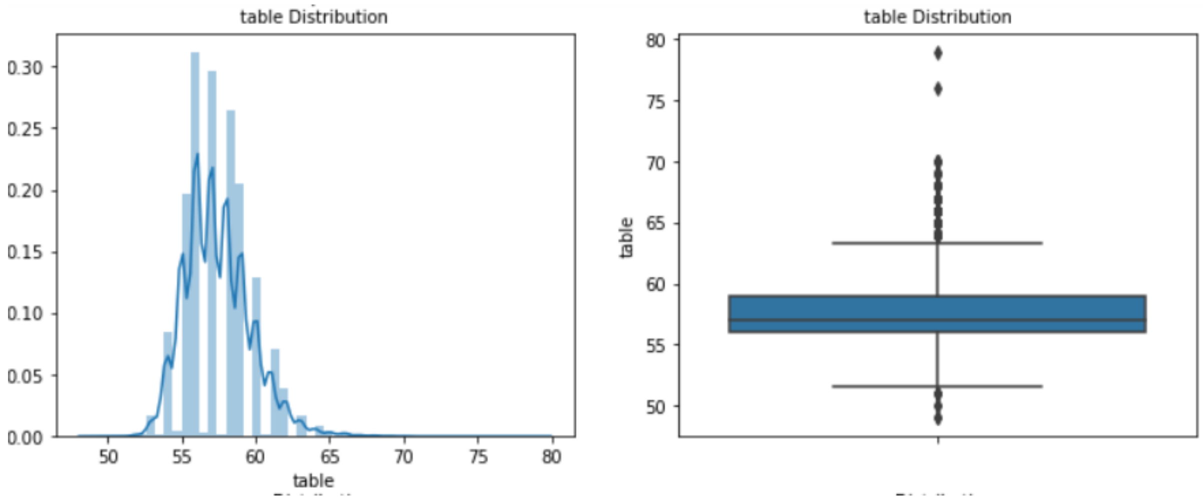


Fig 1.1.3

The distribution of table also seems to be positively skewed.

The box plot of table has outliers.

The data distribution where there is maximum distribution is between 55 to 65.

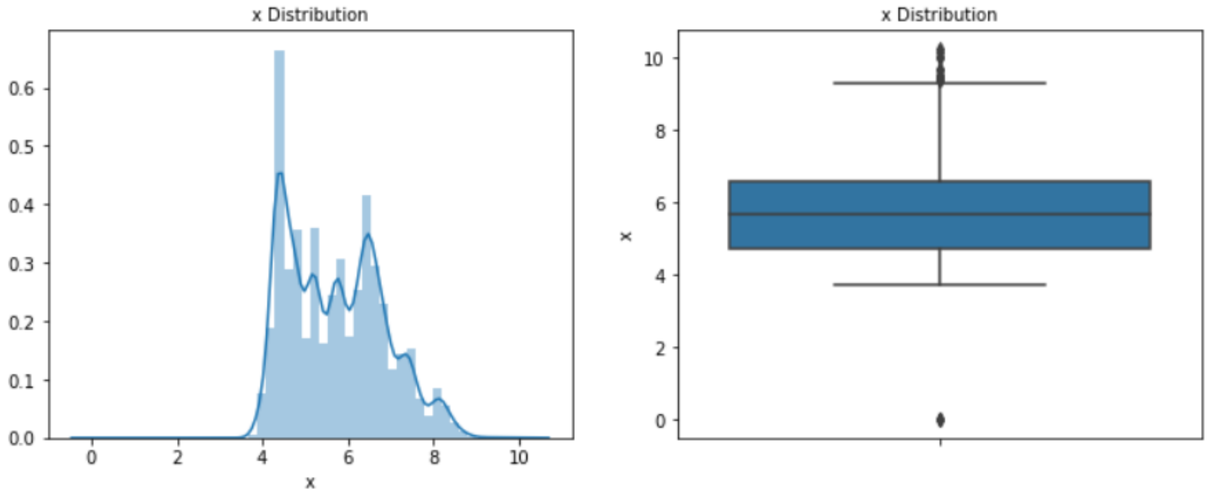


Fig 1.1.4

The distribution of x (Length of the cubic zirconia in mm.) is positively skewed.

The box plot of the data consists of many outliers.

The distribution rages from 4 to 8.

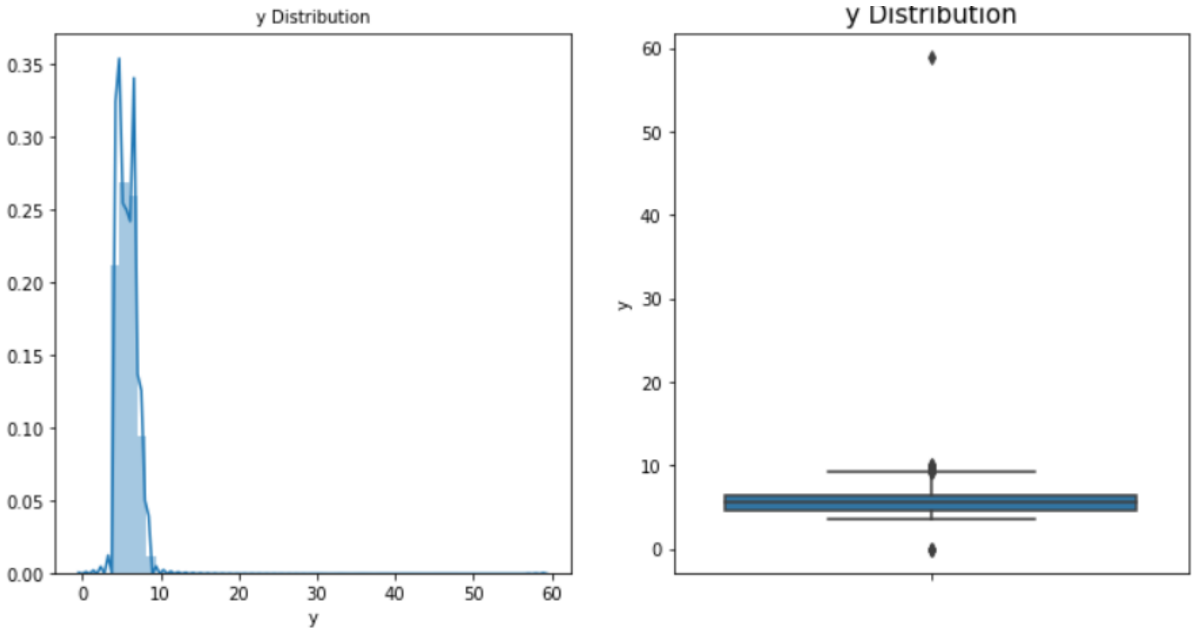


Fig 1.1.5

The distribution of Y (Width of the cubic zirconia in mm.) is positively skewed.

The box plot also consists of outliers.

The distribution too much positively skewed.

The skewness may be due to the diamonds are always made in specific shape. There might not be too many sizes in the market.

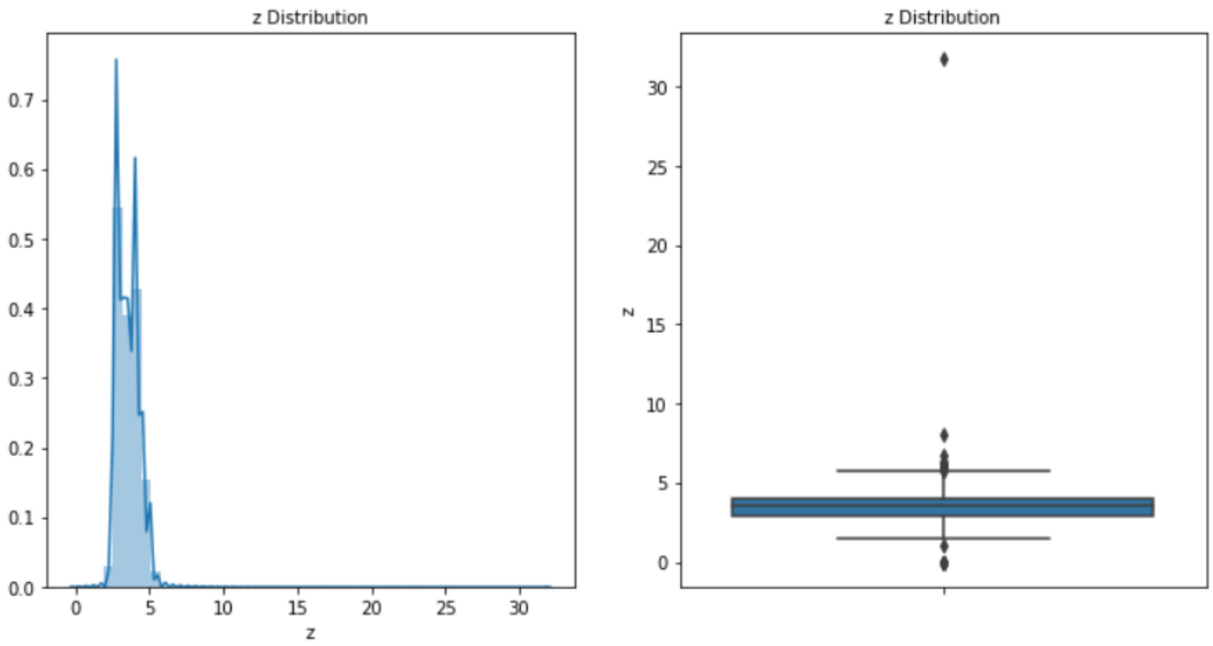


Fig 1.1.6

The distribution of z (Height of the cubic zirconia in mm.) is positively skewed.

The box plot also consists of outliers.

The distribution too much positively skewed.

The skewness may be due to the diamonds are always made in specific shape. There might not be too many sizes in the market.

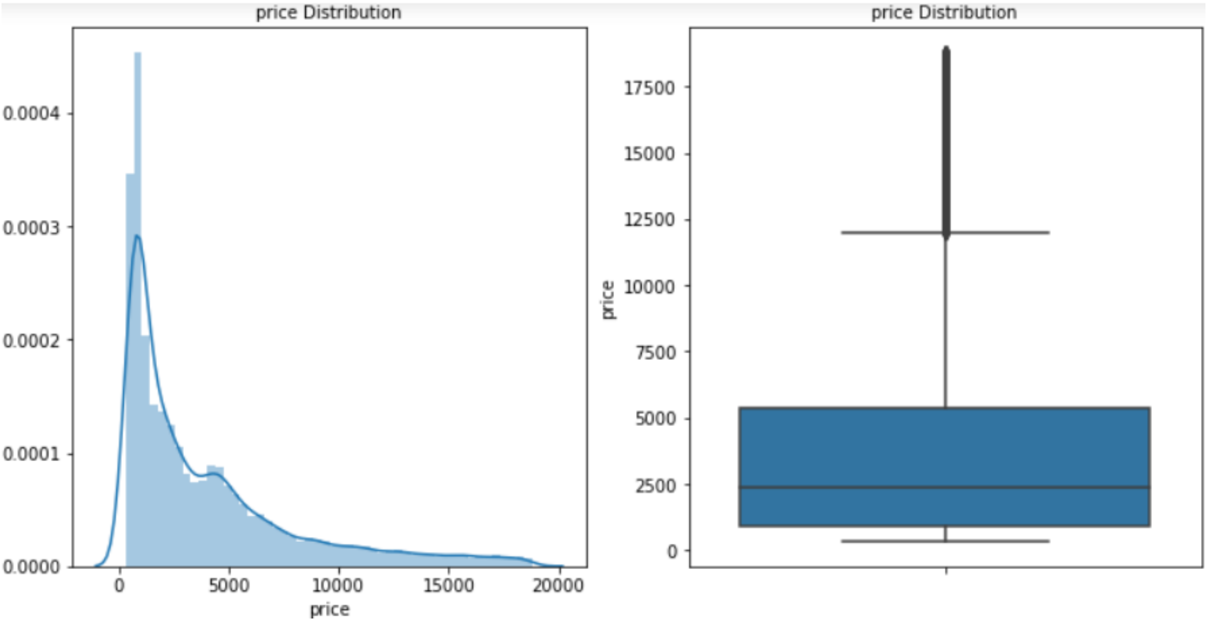


Fig 1.1.7

The price has seemed to be positively skewed.

The skew is positive.

The price has outliers in the data.

The price distribution is from Rs 100 to 8000.

PRICE – HIST

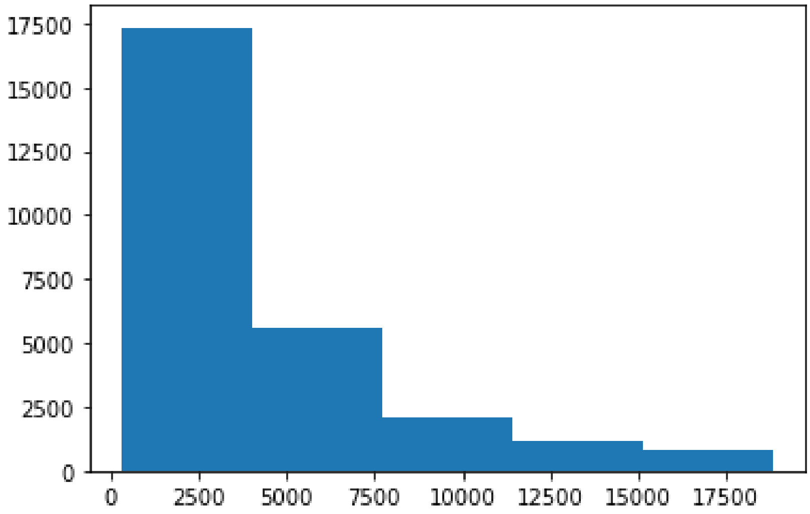
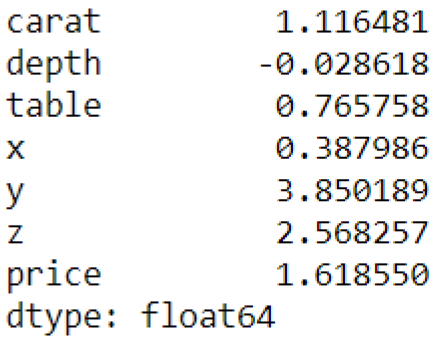


Fig 1.1.8

Skew



```
carat          1.116481
depth         -0.028618
table          0.765758
x              0.387986
y              3.850189
z              2.568257
price          1.618550
dtype: float64
```

Table 1.1.5

Bivariate Analysis

CUT :

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

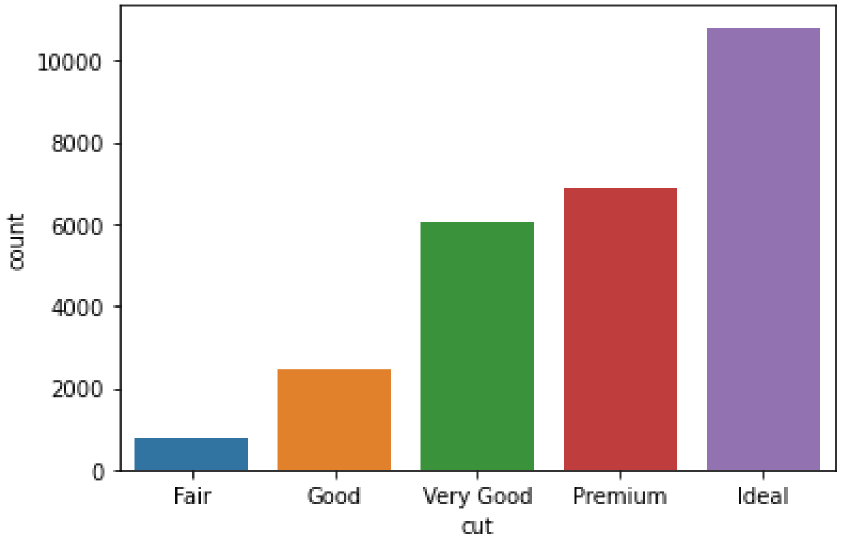


Fig 1.1.9

The most preferred cut seems to be ideal cut for diamonds.

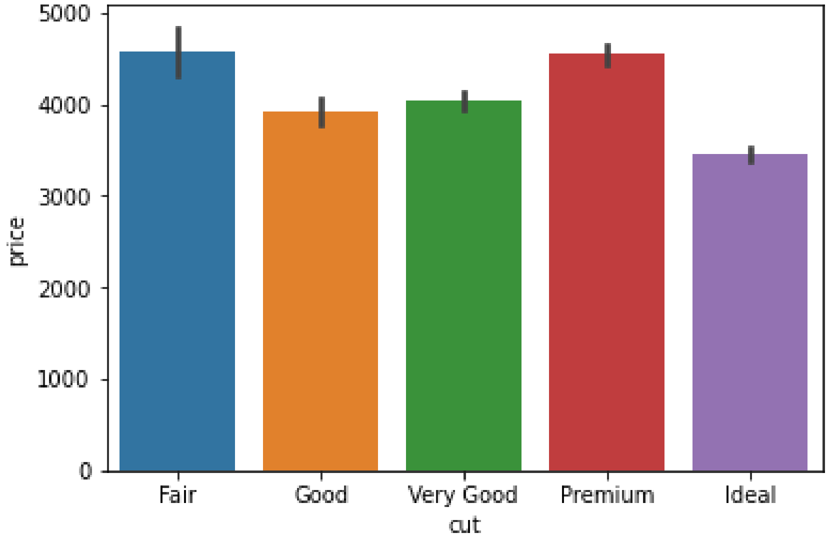


Fig 1.1.10

The reason for the most preferred cut ideal is because those diamonds are priced lower than other cuts.

COLOR:

D being the best and J the worst.

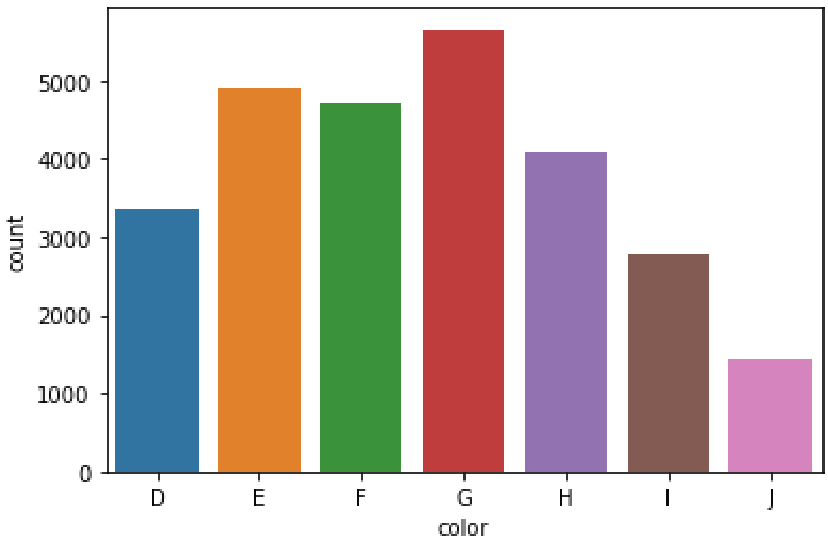


Fig 1.1.11

We have 7 colours in the data, The G seems to be the preferred colour.

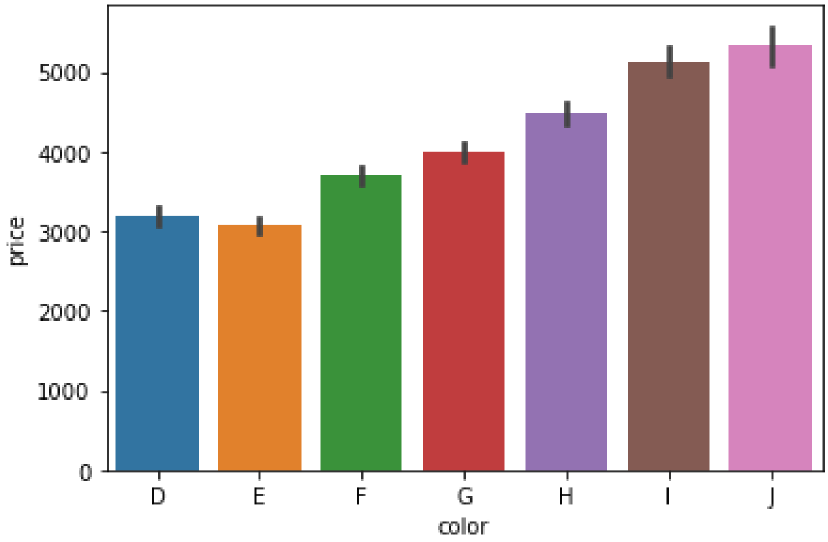


Fig 1.1.12

We see the G is priced in the middle of the seven colours, whereas J being the worst colour price seems too high.

CLARITY:

Best to Worst, FL = flawless, I3= level 3 inclusions) FL, IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1, I2, I3.

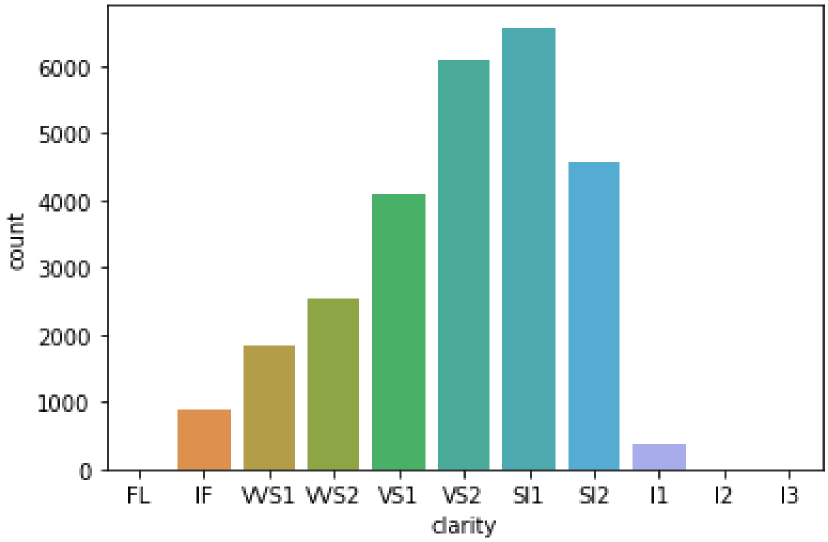


Fig 1.1.13

The clarity VS2 seems to be preferred by people.

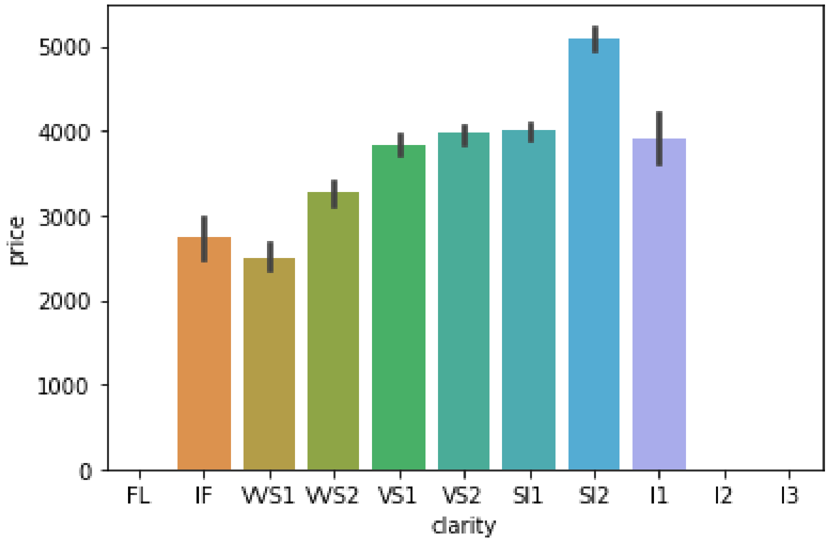


Fig 1.1.14

The data has No FL diamonds, from this we can clearly understand the flawless diamonds are not bringing any profits to the store.

More relationship between categorical variables

Cut and colour

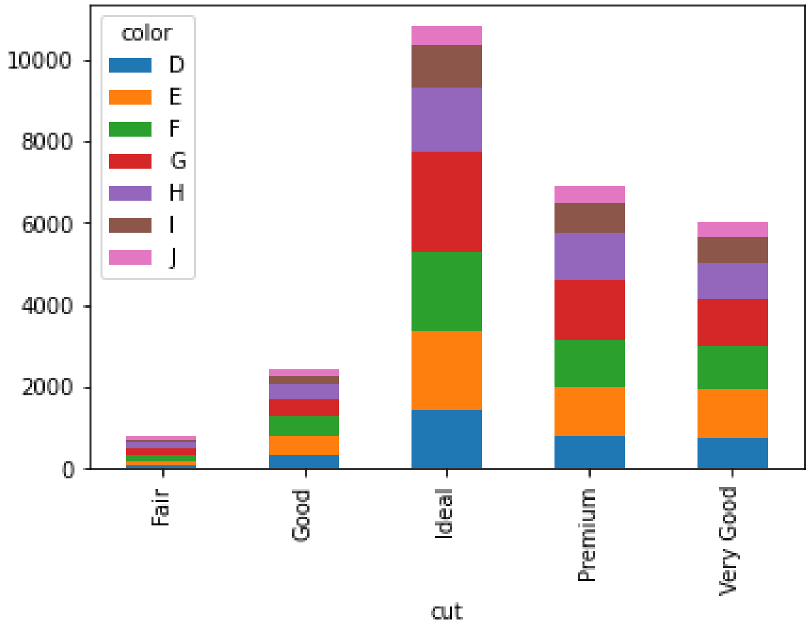


Fig 1.1.15

Cut and Clarity

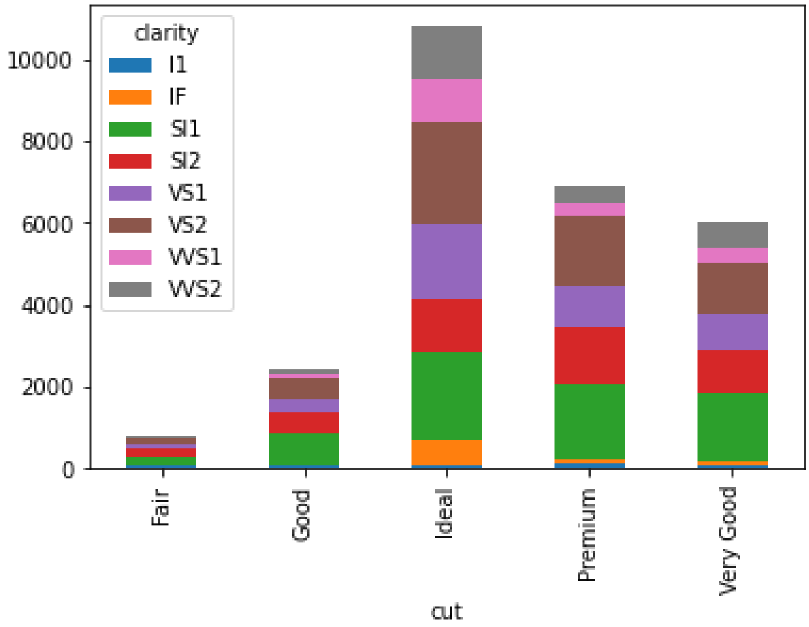


Fig 1.1.16

CORRLEATION

Carat Vs Price

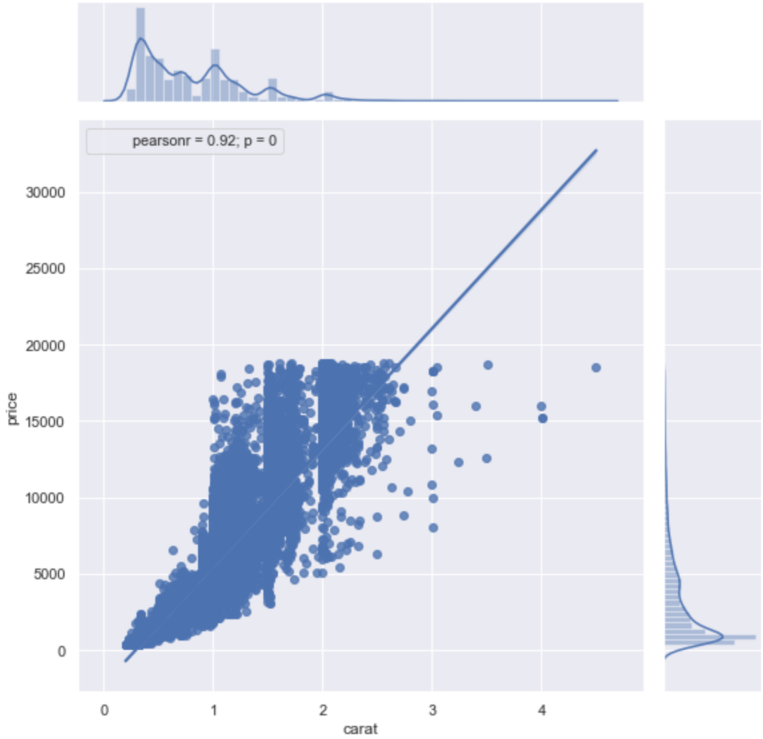


Fig 1.1.17

Depth Vs Price

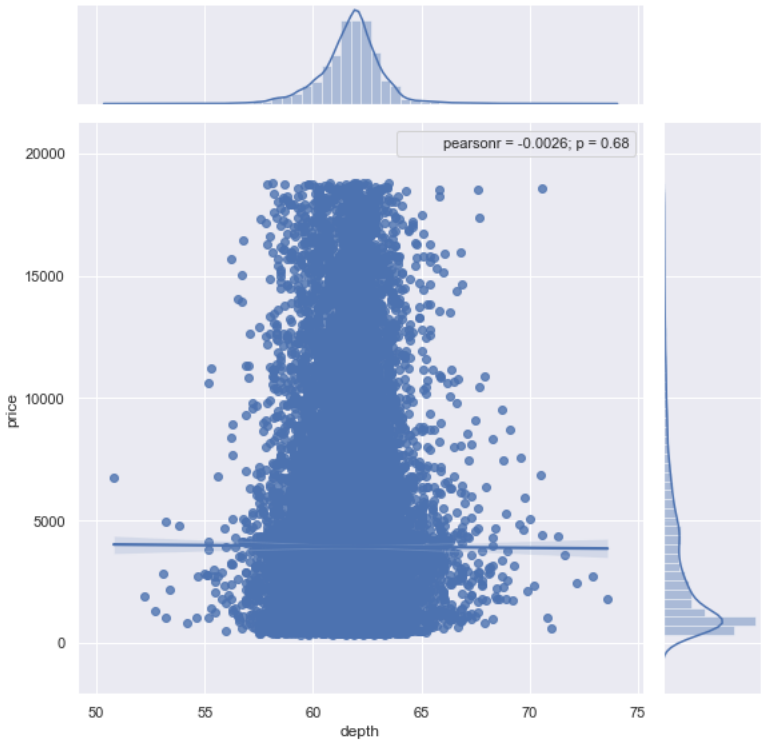


Fig 1.1.18

X Vs Price

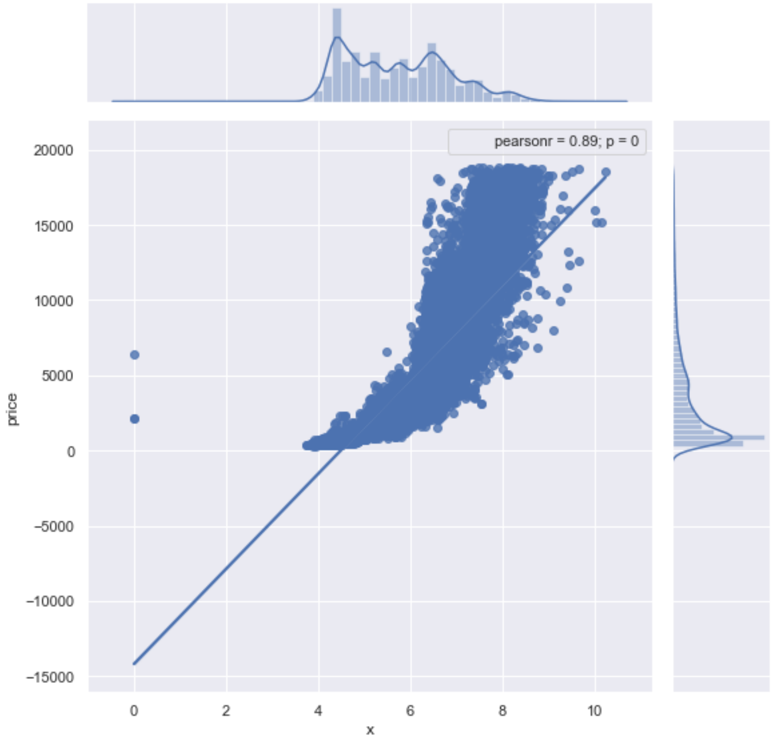


Fig 1.1.19

Y Vs Price

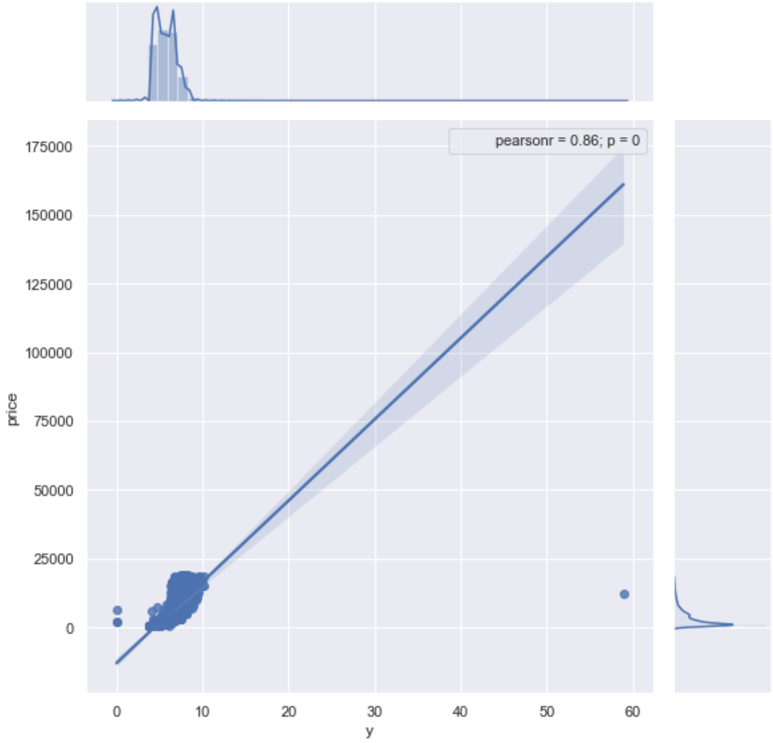


Fig 1.1.20

Z Vs Price

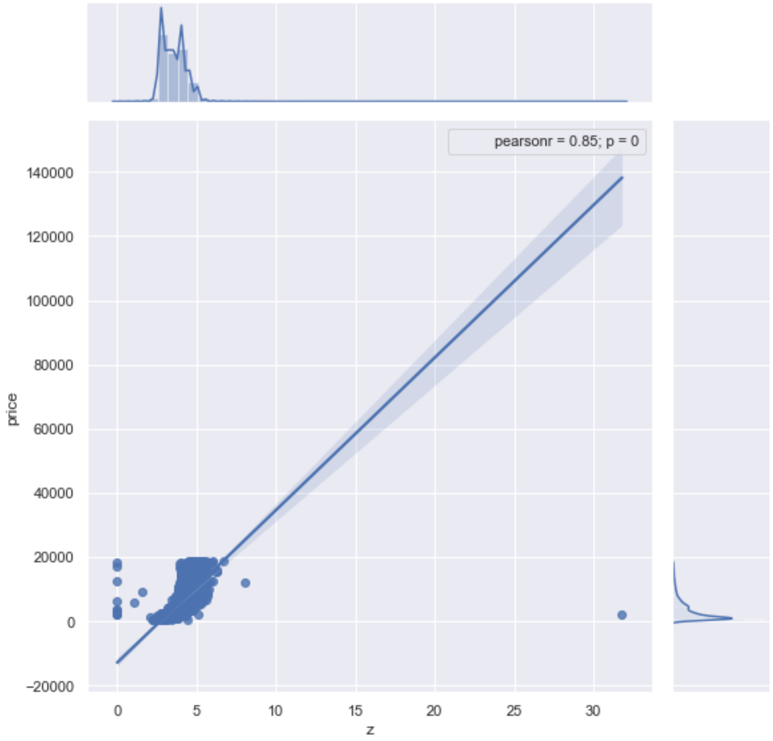


Fig 1.1.21

Observations:

Depth is the only variable which can be considered as normal distribution.

Carat, Table, x, y, z these variables have multiple modes with the spread of data.

Box Plot for each Variable

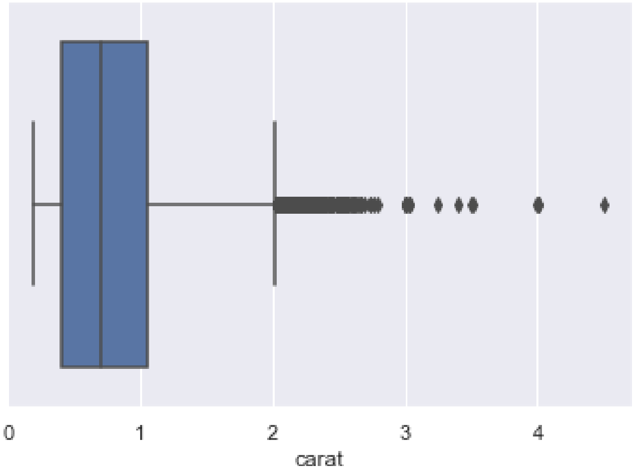


Fig 1.1.22

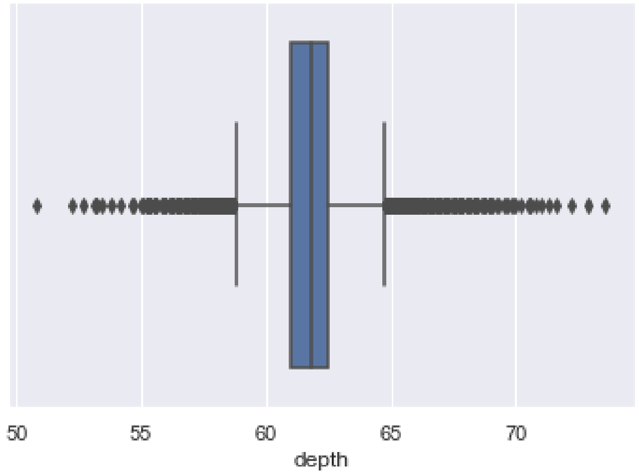


Fig 1.1.23

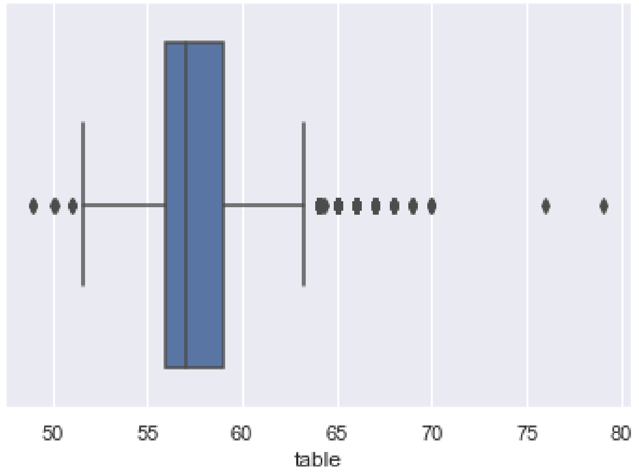


Fig 1.1.24

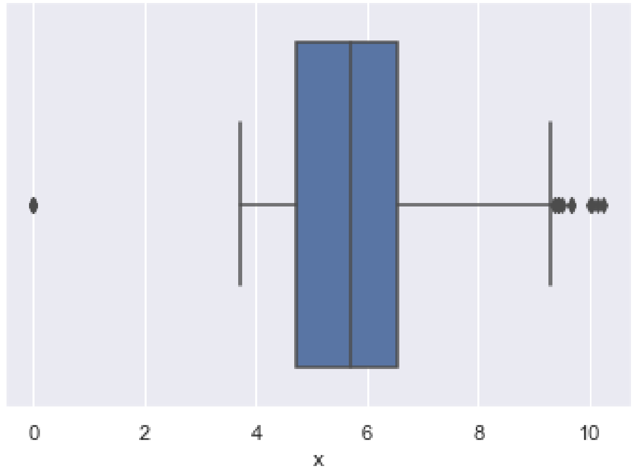


Fig 1.1.25

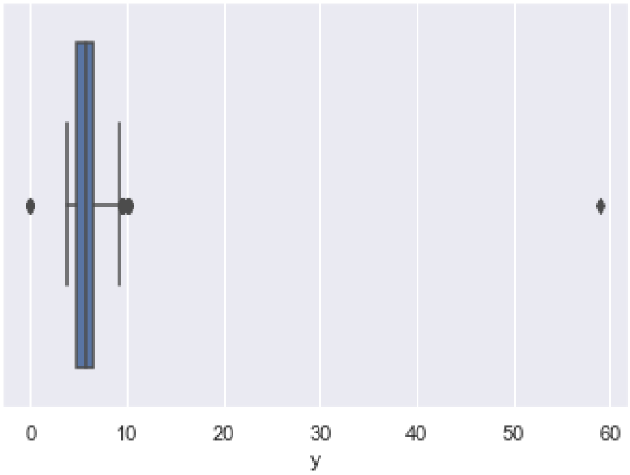


Fig 1.1.26

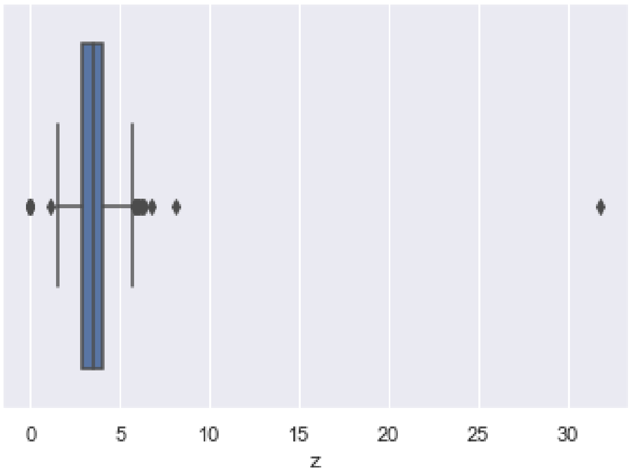


Fig 1.1.27

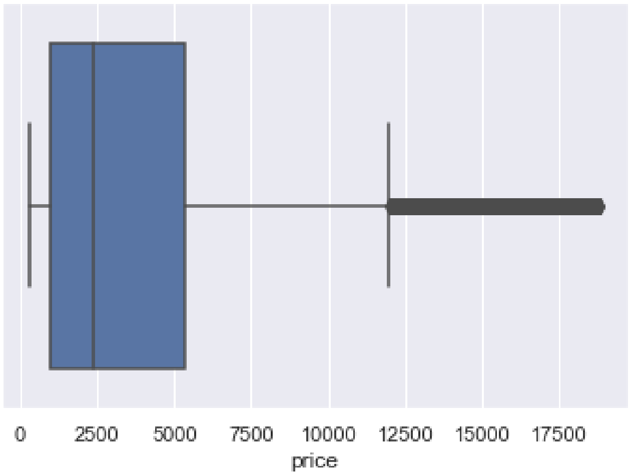


Fig 1.1.28

Outliers:

Large number of outliers are present in all the variables (Carat, Depth, Table, x, y, z).

Price will be the target variable or dependent variable, it is right skewed with large range of outliers.

Data Distribution

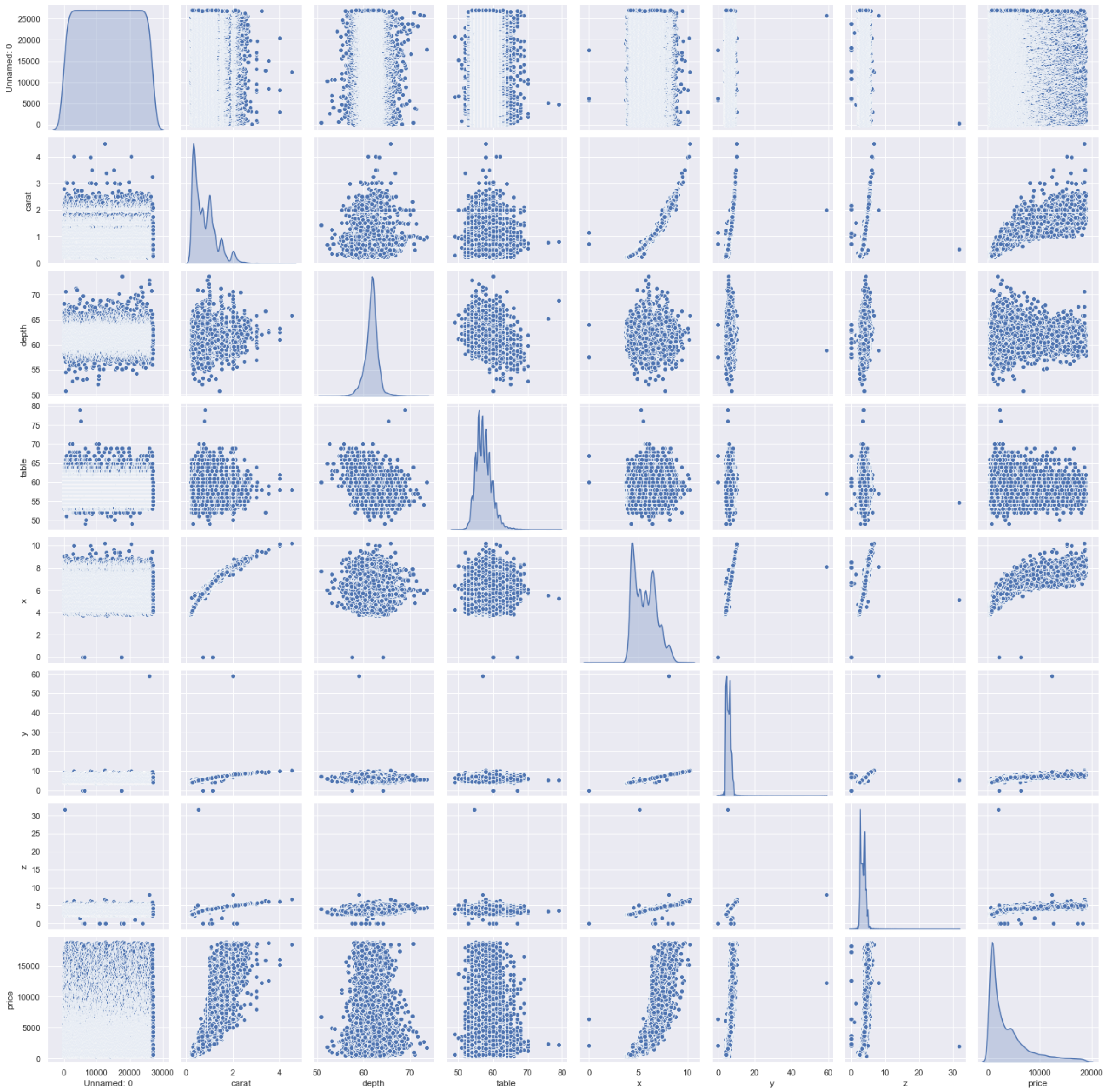
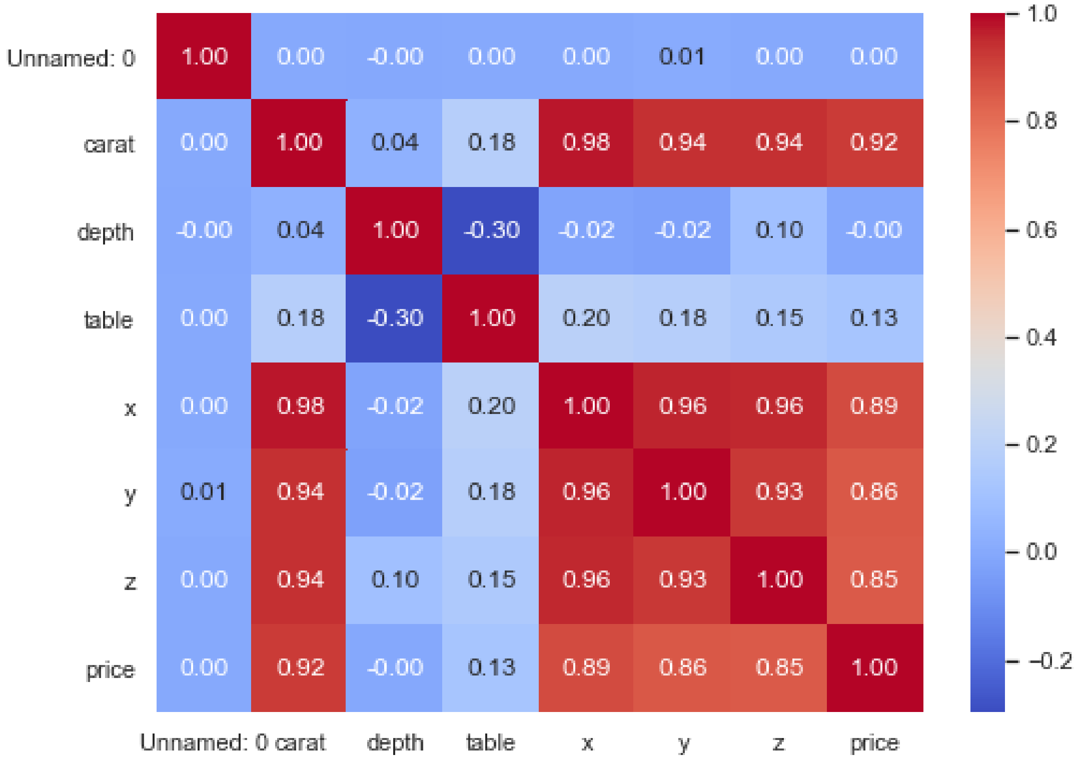


Fig 1.1.29

Observations:

Pair plot allows us to see both distribution of single variable and relationships between two variables.

Correlation Matrix



	Unnamed: 0	carat	depth	table	x	y	z	price
Unnamed: 0	1.00	0.00	-0.00	0.00	0.00	0.01	0.00	0.00
carat	0.00	1.00	0.04	0.18	0.98	0.94	0.94	0.92
depth	-0.00	0.04	1.00	-0.30	-0.02	-0.02	0.10	-0.00
table	0.00	0.18	-0.30	1.00	0.20	0.18	0.15	0.13
x	0.00	0.98	-0.02	0.20	1.00	0.96	0.96	0.89
y	0.01	0.94	-0.02	0.18	0.96	1.00	0.93	0.86
z	0.00	0.94	0.10	0.15	0.96	0.93	1.00	0.85
price	0.00	0.92	-0.00	0.13	0.89	0.86	0.85	1.00

Fig 1.1.30

This matrix clearly shows the presence of multi collinearity in the dataset.

Observations:

High correlation between the different features like carat, x, y, z and price.

Less correlation between table with the other features. Depth is negatively correlated with most the other features except for carat.

EDA Observations:

Price – This variable gives the continuous output with the price of the cubic zirconia stones; this will be our Target Variable.

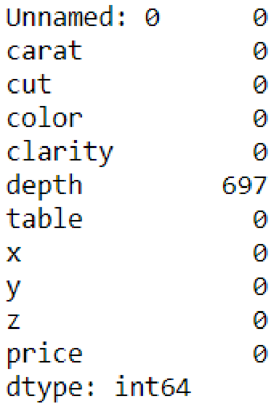
Carat, depth, table, x, y, z variables are numerical or continuous variables.

Cut, Clarity and Colour are categorical variables.

We will drop the first column ‘Unnamed: 0’ column as this is not important for our study which leaves the shape of the dataset with 26967 rows & 10 Columns

Only in ‘depth' 697 missing values are present which we will impute by its median values.

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? Do you think scaling is necessary in this case? Check for the possibility of combining the sub levels of an ordinal variables and take actions accordingly. Explain why you are combining these sub levels with appropriate reasoning.



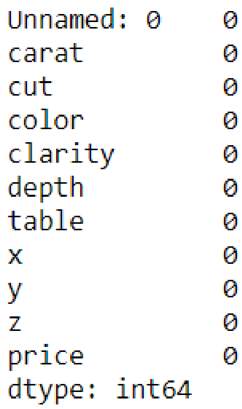
```
Unnamed: 0      0
carat           0
cut             0
color           0
clarity         0
depth         697
table           0
x               0
y               0
z               0
price           0
dtype: int64
```

Table 1.2.1

We have Null values in depth, since depth being continuous variable mean or median imputation can be done.

The percentage of Null values is less than 5%, we can also drop these if we want.

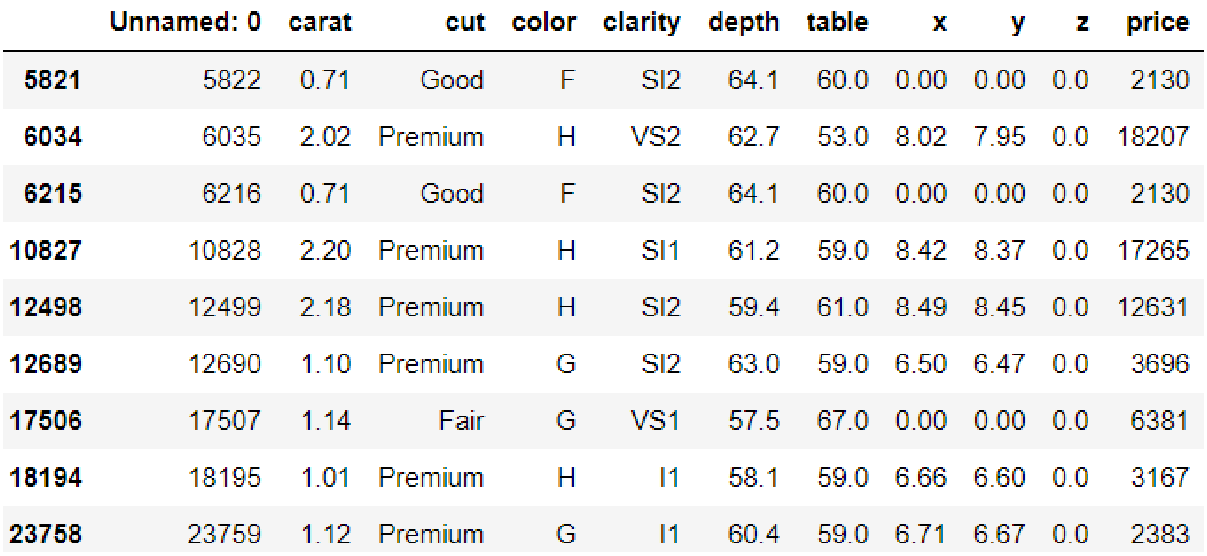
After median imputation, we don’t have any null values in the dataset.



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

Table 1.2.2

Checking if there is value that is “0”



	Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
5821	5822	0.71	Good	F	SI2	64.1	60.0	0.00	0.00	0.0	2130
6034	6035	2.02	Premium	H	VS2	62.7	53.0	8.02	7.95	0.0	18207
6215	6216	0.71	Good	F	SI2	64.1	60.0	0.00	0.00	0.0	2130
10827	10828	2.20	Premium	H	SI1	61.2	59.0	8.42	8.37	0.0	17265
12498	12499	2.18	Premium	H	SI2	59.4	61.0	8.49	8.45	0.0	12631
12689	12690	1.10	Premium	G	SI2	63.0	59.0	6.50	6.47	0.0	3696
17506	17507	1.14	Fair	G	VS1	57.5	67.0	0.00	0.00	0.0	6381
18194	18195	1.01	Premium	H	I1	58.1	59.0	6.66	6.60	0.0	3167
23758	23759	1.12	Premium	G	I1	60.4	59.0	6.71	6.67	0.0	2383

Table 1.2.3

We have certain rows having values zero, the x, y, z are the dimensions of a diamond so this can’t take into model. As there are very less rows.

We can drop these rows as don’t have any meaning in model building.

Scaling

Scaling can be useful to reduce or check the multi collinearity in the data.

So, if scaling is applied or not applied there is no difference in scores of the model or VIF values and model performance is same.

Another valid reason for scaling in regression is when one predictor variable has a very large scale. In that case, the regression coefficients may be on a very small order of magnitude which can be unclear to interpret.

The convention that we standardize predictions primarily exists so that the units of the regression coefficients are the same. More often, the dataset contains feature highly varying in magnitudes, units and range.

However, most of the machine learning algorithms use Euclidean distance between two data points in their computations, and this can be a potential problem.

Also, scaling helps to standardize the independent features present in the data in a fixed range. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

The features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes. To suppress this effect, we need to bring all features to the same level of magnitudes.

Check for the possibility of combining the sub levels of an ordinal variables and take actions accordingly. Explain why you are combining these sub levels with appropriate reasoning.

Reason we combine sub levels to reduce the levels in the dataset which will reduce complexity of the model.

For example, if dataset has multiple education levels such as pre-primary, secondary, high school, 12th, graduation, post-graduation, masters and PhD, we perform combining sub levels as below to reduce levels of the model.

pre-primary, secondary, high school, 12th into 1 category as Schooling.

post-graduation and masters into another category as PG.

By reducing the levels of variable, it can reduce complexity of the model as well.

In a given dataset, I tried combining sublevels such as ‘Good’ and ‘Very Good’ under 1 category i.e., ‘Good’ and there is no difference is model scores or performance.

This may be because there are only 5 levels and decreasing 1 level down may not be much difference in model scores or model performance.

Checking for Outliers

Before Treating Outliers

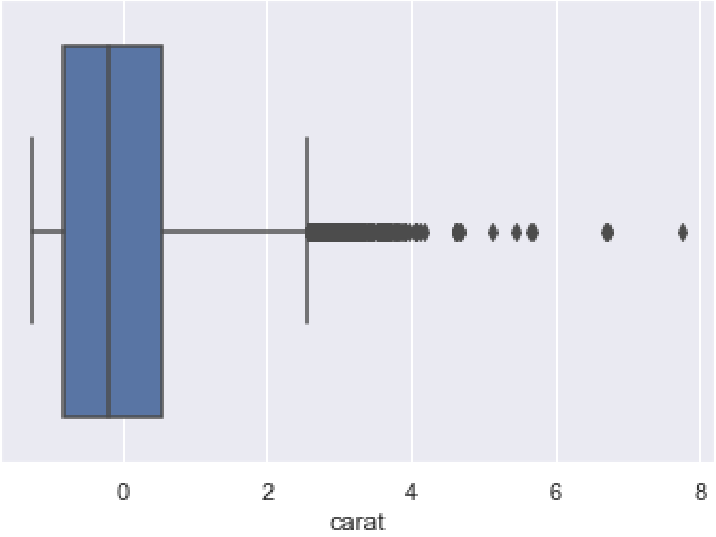


Fig 1.2.1

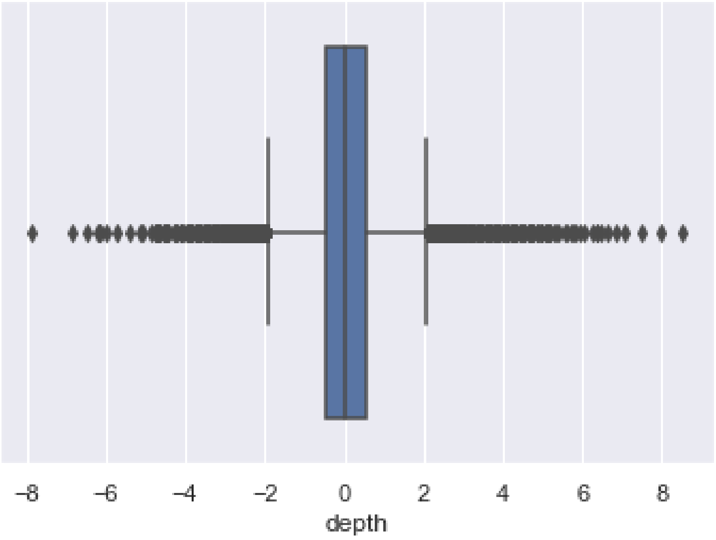


Fig 1.2.2

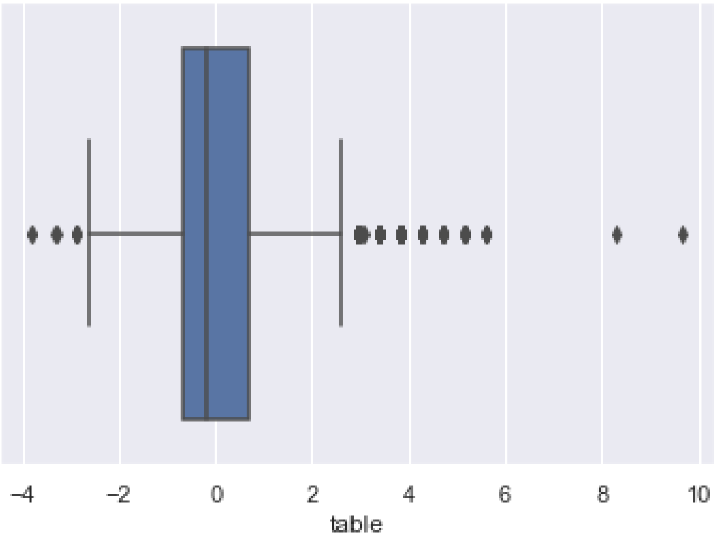


Fig 1.2.3

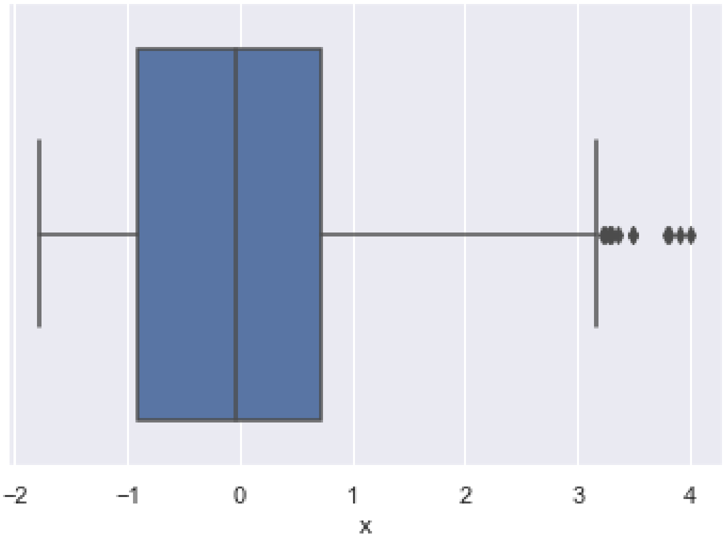


Fig 1.2.4

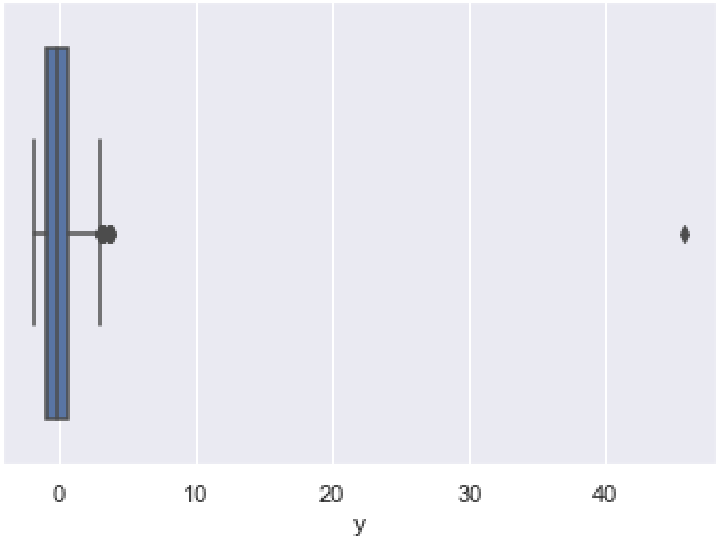


Fig 1.2.5

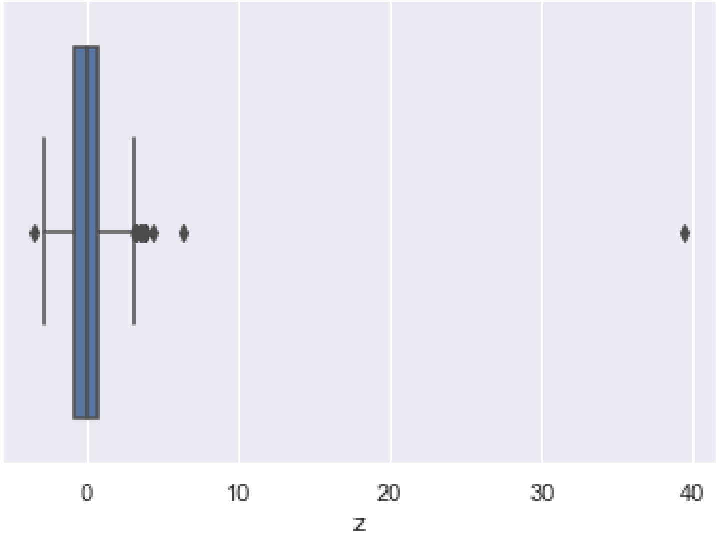


Fig 1.2.6

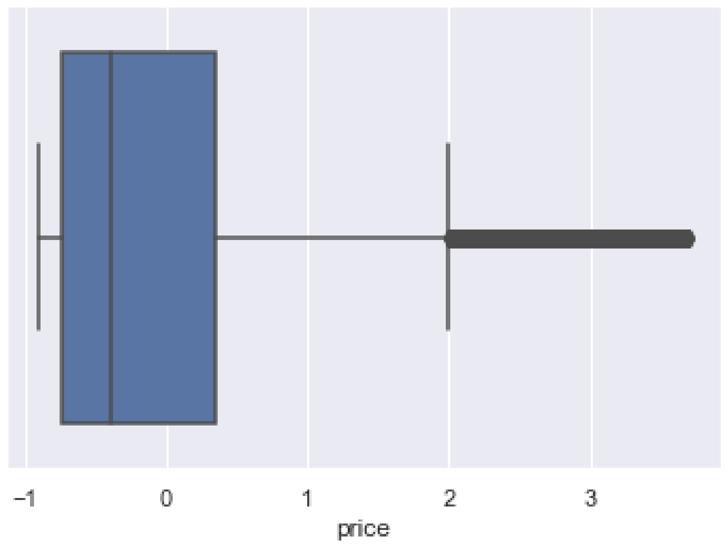


Fig 1.2.7

After Treating Outliers

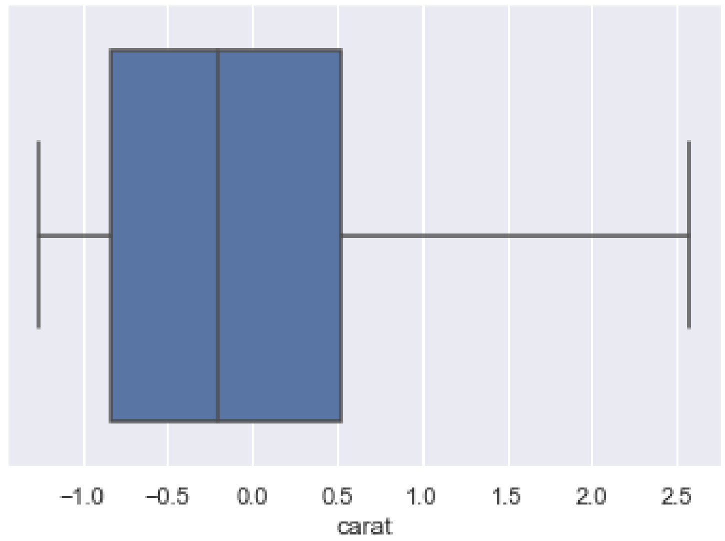


Fig 1.2.8

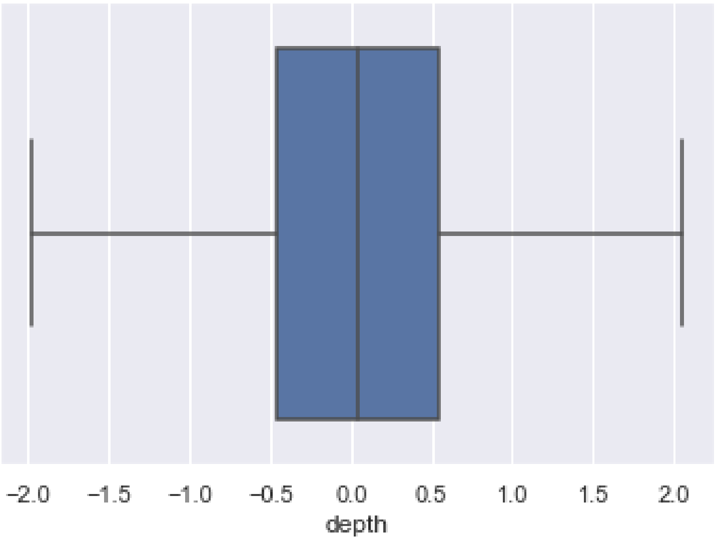


Fig 1.2.9

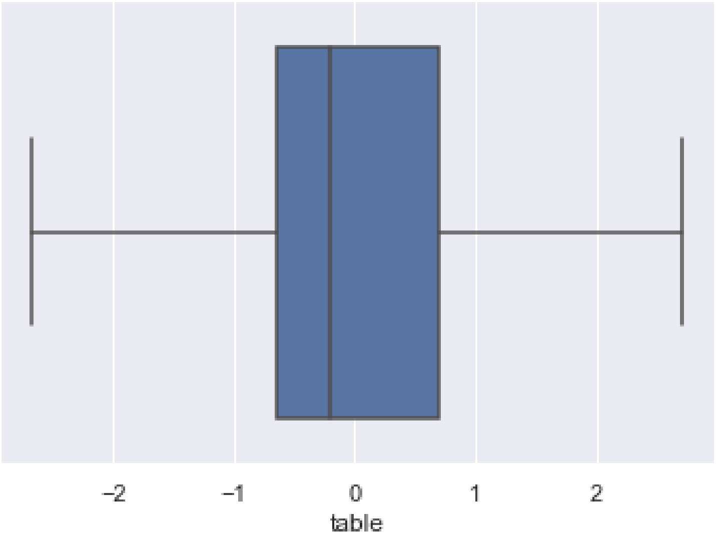


Fig 1.2.10

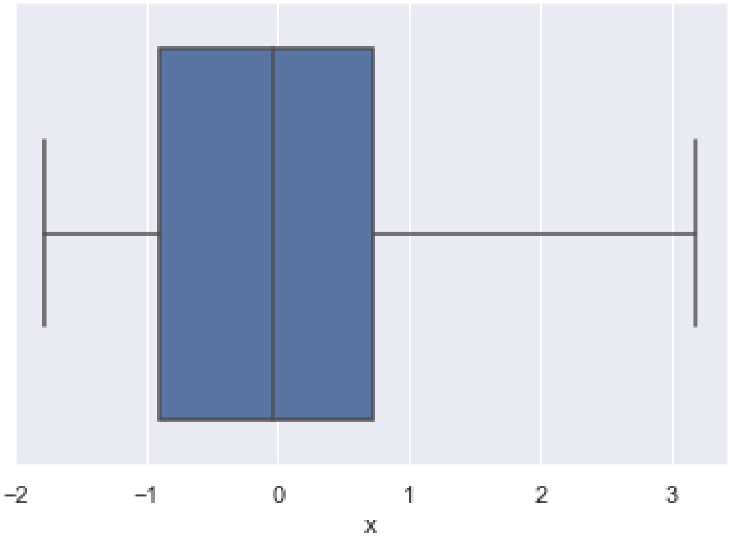


Fig 1.2.11

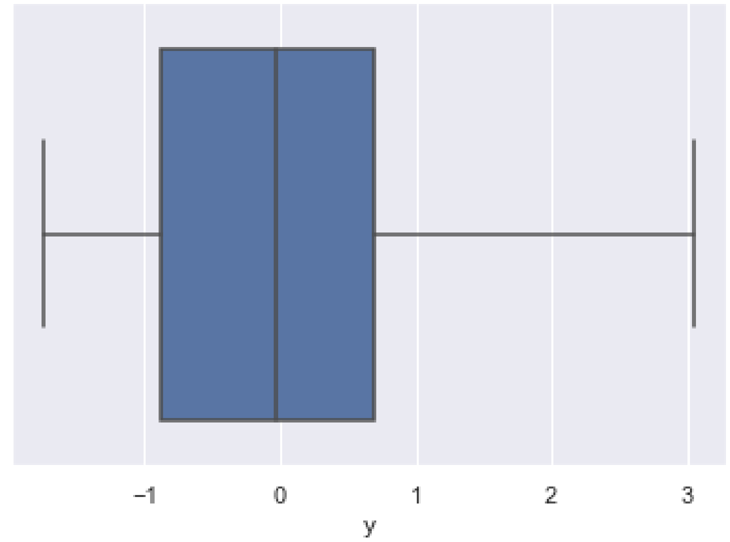


Fig 1.2.12

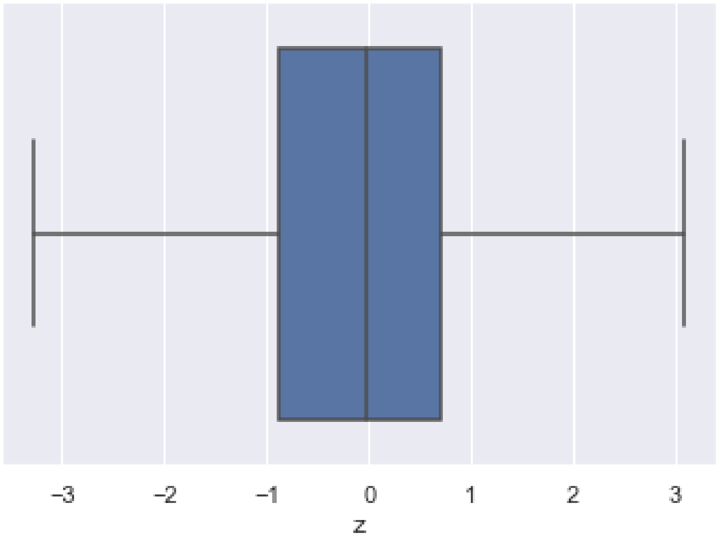


Fig 1.2.13

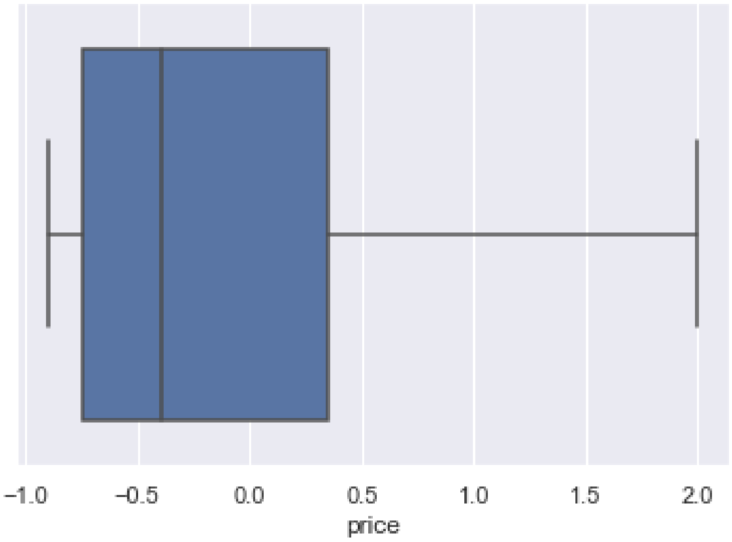
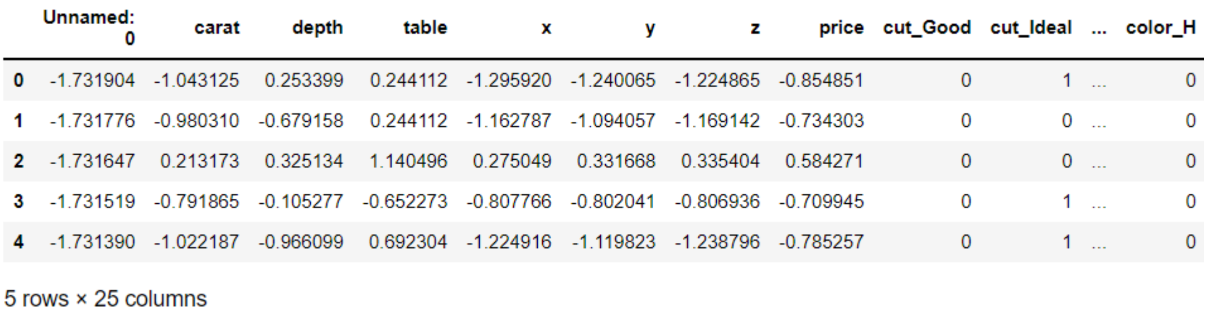


Fig 1.2.14

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 Rsquare, RMSE & Adj Rsquare. Compare these models and select the best one with appropriate reasoning.

Encoding String Values

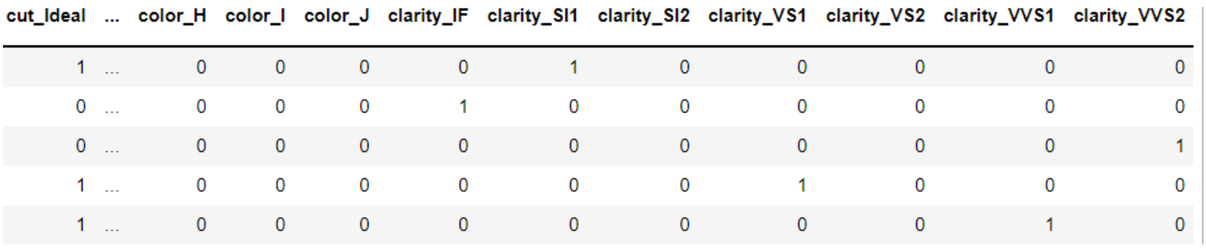
Get Dummies



	Unnamed: 0	carat	depth	table	x	y	z	price	cut_Good	cut_Ideal	...	color_H
0	-1.731904	-1.043125	0.253399	0.244112	-1.295920	-1.240065	-1.224865	-0.854851	0	1	...	0
1	-1.731776	-0.980310	-0.679158	0.244112	-1.162787	-1.094057	-1.169142	-0.734303	0	0	...	0
2	-1.731647	0.213173	0.325134	1.140496	0.275049	0.331668	0.335404	0.584271	0	0	...	0
3	-1.731519	-0.791865	-0.105277	-0.652273	-0.807766	-0.802041	-0.806936	-0.709945	0	1	...	0
4	-1.731390	-1.022187	-0.966099	0.692304	-1.224916	-1.119823	-1.238796	-0.785257	0	1	...	0

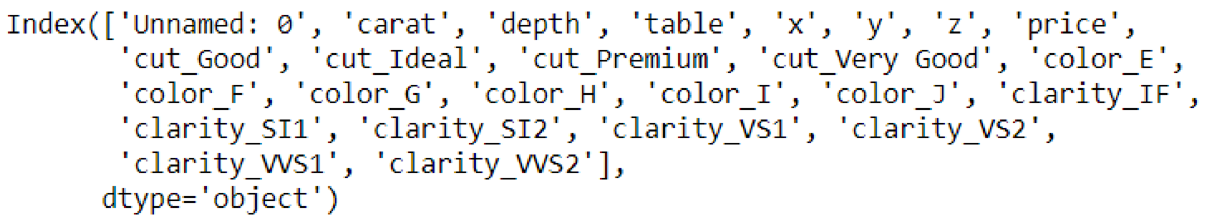
5 rows × 25 columns

Table 1.3.1



cut_Ideal	...	color_H	color_I	color_J	clarity_IF	clarity_SI1	clarity_SI2	clarity_VS1	clarity_VS2	clarity_VVS1	clarity_VVS2
1	...	0	0	0	0	1	0	0	0	0	0
0	...	0	0	0	1	0	0	0	0	0	0
0	...	0	0	0	0	0	0	0	0	0	1
1	...	0	0	0	0	0	0	1	0	0	0
1	...	0	0	0	0	0	0	0	0	1	0

Table 1.3.2



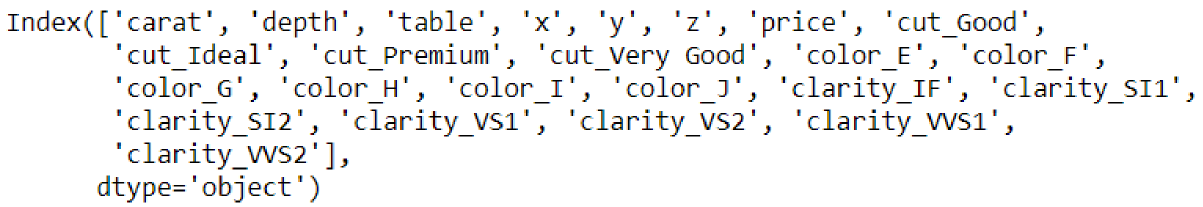
```
Index(['Unnamed: 0', 'carat', 'depth', 'table', 'x', 'y', 'z', 'price',
       'cut_Good', 'cut_Ideal', 'cut_Premium', 'cut_Very Good', 'color_E',
       'color_F', 'color_G', 'color_H', 'color_I', 'color_J', 'clarity_IF',
       'clarity_SI1', 'clarity_SI2', 'clarity_VS1', 'clarity_VS2',
       'clarity_VVS1', 'clarity_VVS2'],
      dtype='object')
```

Table 1.3.3

Dummies have been encoded.

Linear regression model does not take categorical values so that we have encoded categorical values to integer for better results.

Dropping Unwanted Columns

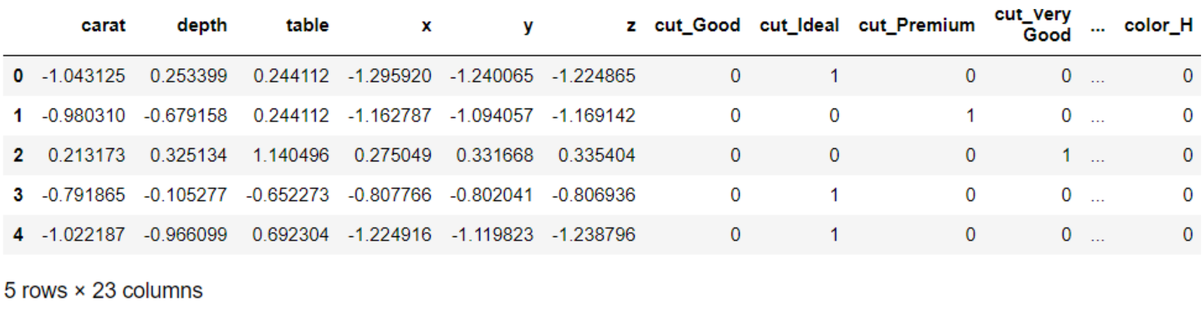


```
Index(['carat', 'depth', 'table', 'x', 'y', 'z', 'price', 'cut_Good',
       'cut_Ideal', 'cut_Premium', 'cut_Very Good', 'color_E', 'color_F',
       'color_G', 'color_H', 'color_I', 'color_J', 'clarity_IF', 'clarity_SI1',
       'clarity_SI2', 'clarity_VS1', 'clarity_VS2', 'clarity_VVS1',
       'clarity_VVS2'],
      dtype='object')
```

Table 1.3.4

Train/Test Split

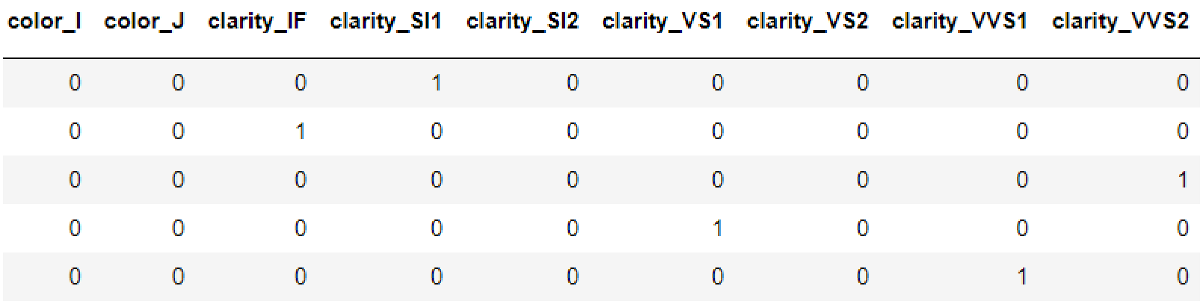
All predictor variables copied into X and below is the sample dataset head of the X.



	carat	depth	table	x	y	z	cut_Good	cut_Ideal	cut_Premium	cut_Very Good	...	color_H
0	-1.043125	0.253399	0.244112	-1.295920	-1.240065	-1.224865	0	1	0	0	...	0
1	-0.980310	-0.679158	0.244112	-1.162787	-1.094057	-1.169142	0	0	1	0	...	0
2	0.213173	0.325134	1.140496	0.275049	0.331668	0.335404	0	0	0	1	...	0
3	-0.791865	-0.105277	-0.652273	-0.807766	-0.802041	-0.806936	0	1	0	0	...	0
4	-1.022187	-0.966099	0.692304	-1.224916	-1.119823	-1.238796	0	1	0	0	...	0

5 rows × 23 columns

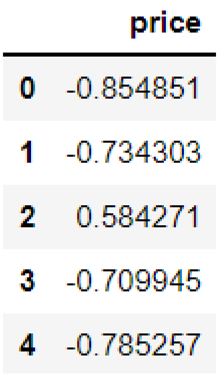
Table 1.3.5



color_I	color_J	clarity_IF	clarity_SI1	clarity_SI2	clarity_VS1	clarity_VS2	clarity_VVS1	clarity_VVS2
0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1
0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0

Table 1.3.6

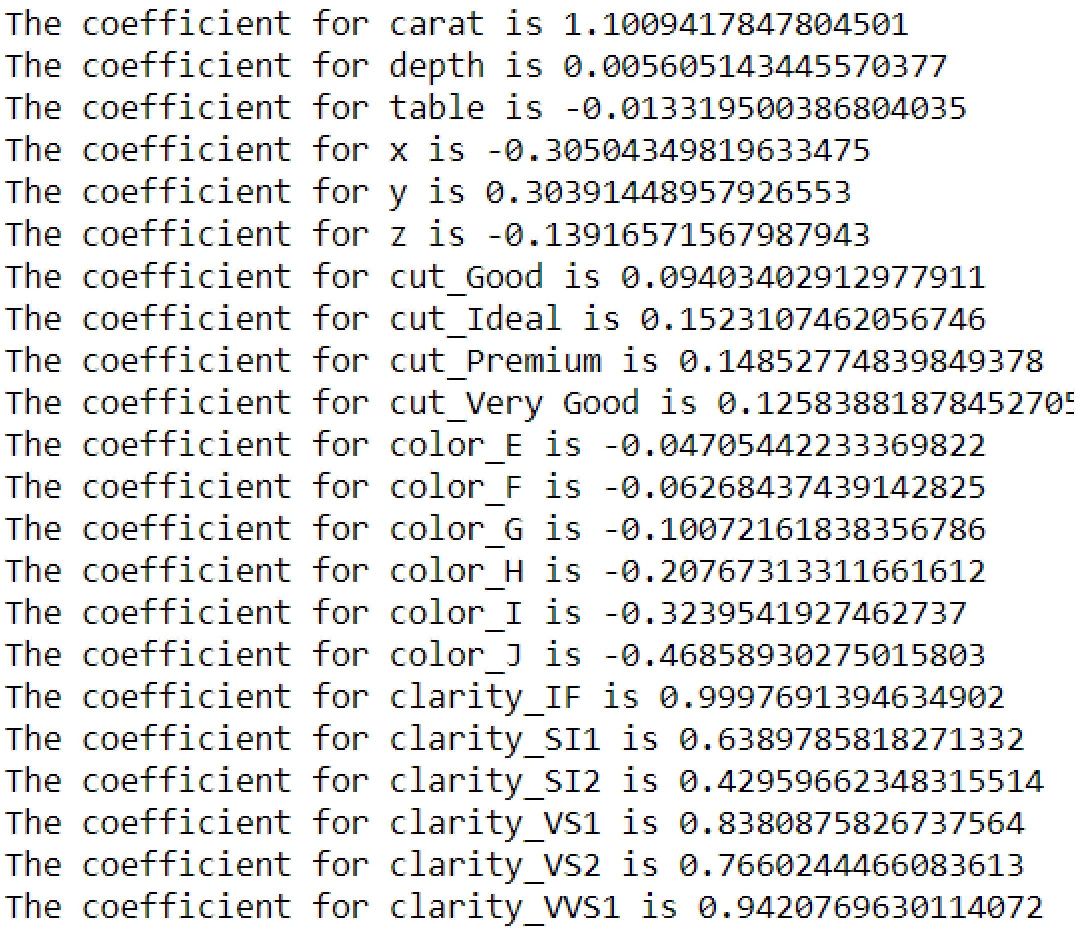
And target variable is copied into y and below is the sample dataset head of target variable y.



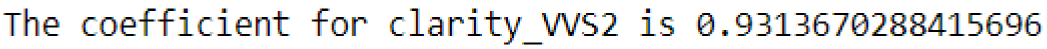
	price
0	-0.854851
1	-0.734303
2	0.584271
3	-0.709945
4	-0.785257

Table 1.3.7

Building Linear Regression Model



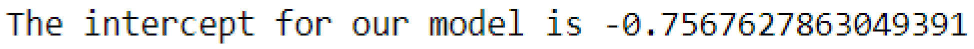
```
The coefficient for carat is 1.1009417847804501
The coefficient for depth is 0.005605143445570377
The coefficient for table is -0.013319500386804035
The coefficient for x is -0.30504349819633475
The coefficient for y is 0.30391448957926553
The coefficient for z is -0.13916571567987943
The coefficient for cut_Good is 0.09403402912977911
The coefficient for cut_Ideal is 0.1523107462056746
The coefficient for cut_Premium is 0.14852774839849378
The coefficient for cut_Very Good is 0.12583881878452705
The coefficient for color_E is -0.04705442233369822
The coefficient for color_F is -0.06268437439142825
The coefficient for color_G is -0.10072161838356786
The coefficient for color_H is -0.20767313311661612
The coefficient for color_I is -0.3239541927462737
The coefficient for color_J is -0.46858930275015803
The coefficient for clarity_IF is 0.9997691394634902
The coefficient for clarity_SI1 is 0.6389785818271332
The coefficient for clarity_SI2 is 0.42959662348315514
The coefficient for clarity_VS1 is 0.8380875826737564
The coefficient for clarity_VS2 is 0.7660244466083613
The coefficient for clarity_VVS1 is 0.9420769630114072
```



```
The coefficient for clarity_VVS2 is 0.9313670288415696
```

Table 1.3.8

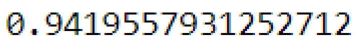
Intercept for the model



```
The intercept for our model is -0.7567627863049391
```

Table 1.3.9

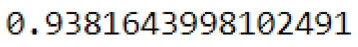
R Square on Training Data



```
0.9419557931252712
```

Table 1.3.10

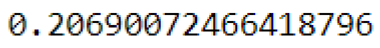
R Square on Test Data



```
0.9381643998102491
```

Table 1.3.11

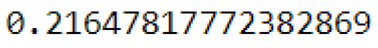
RMSE on Training Data



```
0.20690072466418796
```

Table 1.3.12

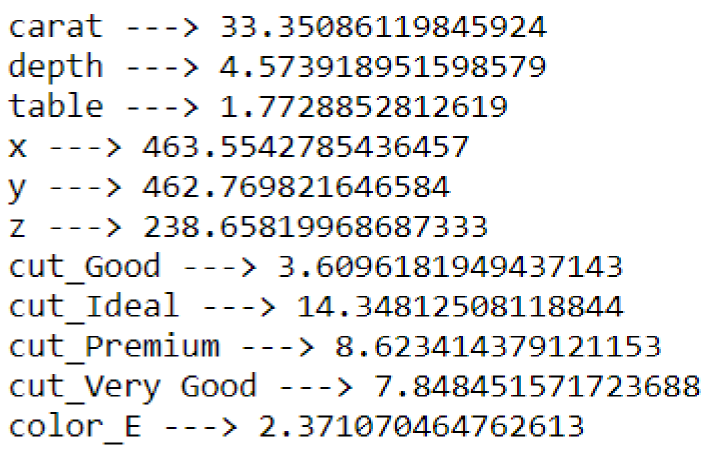
RMSE on Test Data



```
0.21647817772382869
```

Table 1.3.13

VIF Values



```
carat ---> 33.35086119845924
depth ---> 4.573918951598579
table ---> 1.7728852812619
x ---> 463.5542785436457
y ---> 462.769821646584
z ---> 238.65819968687333
cut_Good ---> 3.6096181949437143
cut_Ideal ---> 14.34812508118844
cut_Premium ---> 8.623414379121153
cut_Very Good ---> 7.848451571723688
color_E ---> 2.371070464762613
```

Table 1.3.14

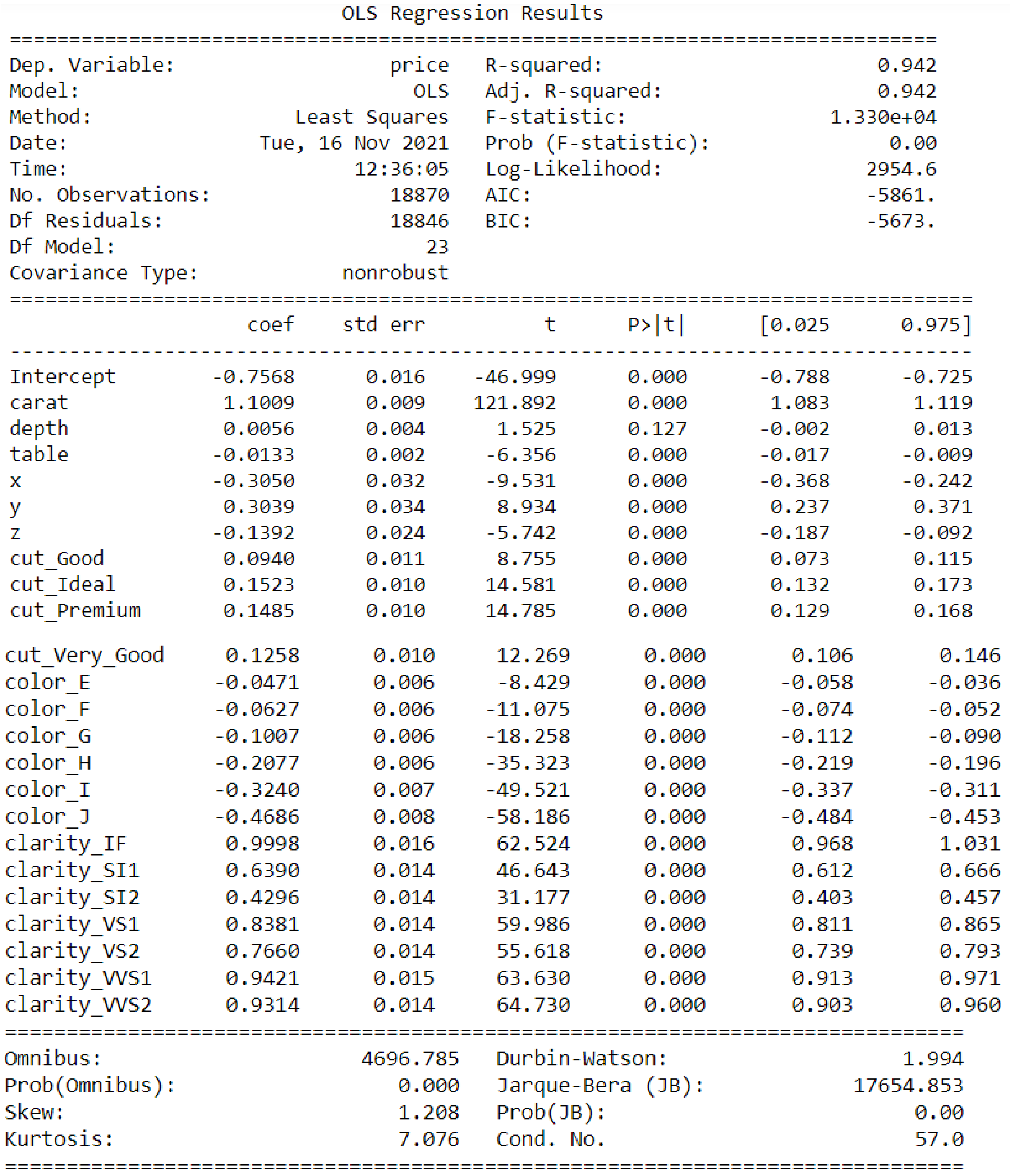
We still find we have multi collinearity in the dataset, to drop these values to lower level we can drop columns after doing stats model.

From stats model we can understand the features that do not contribute to the model, we can remove those features, after that the Vif Values will be reduced.

Ideal value of VIF is less than 5%.

STATS MODEL

Best Params Summary

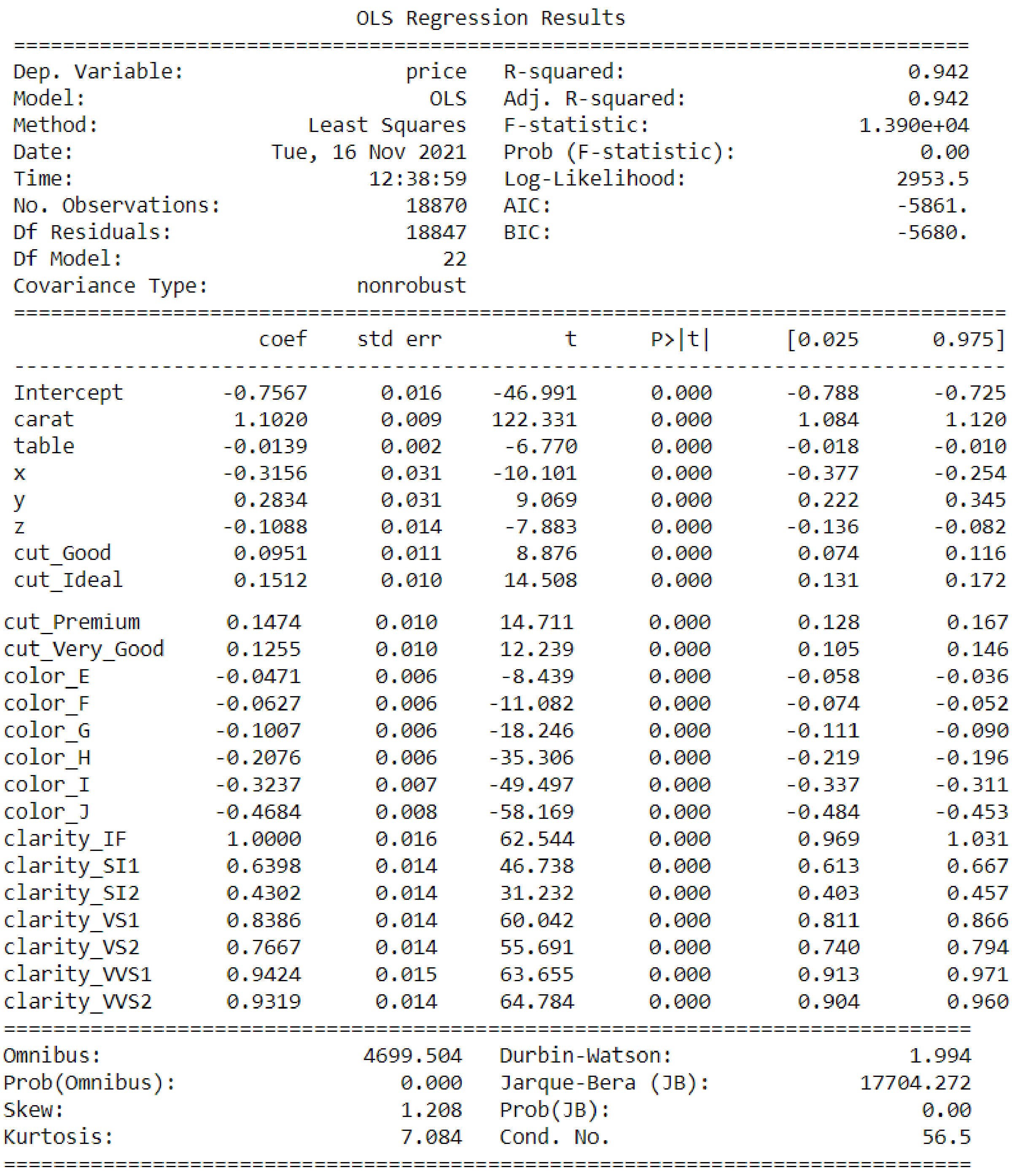


```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.942
Model:                            OLS   Adj. R-squared:                  0.942
Method:                 Least Squares   F-statistic:                 1.330e+04
Date:                Tue, 16 Nov 2021   Prob (F-statistic):               0.00
Time:                        12:36:05   Log-Likelihood:                 2954.6
No. Observations:               18870   AIC:                            -5861.
Df Residuals:                   18846   BIC:                            -5673.
Df Model:                          23
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
Intercept        -0.7568      0.016    -46.999      0.000      -0.788      -0.725
carat             1.1009      0.009    121.892      0.000       1.083       1.119
depth             0.0056      0.004      1.525      0.127      -0.002       0.013
table            -0.0133      0.002     -6.356      0.000      -0.017      -0.009
x                -0.3050      0.032     -9.531      0.000      -0.368      -0.242
y                 0.3039      0.034      8.934      0.000       0.237       0.371
z                -0.1392      0.024     -5.742      0.000      -0.187      -0.092
cut_Good          0.0940      0.011      8.755      0.000       0.073       0.115
cut_Ideal         0.1523      0.010     14.581      0.000       0.132       0.173
cut_Premium       0.1485      0.010     14.785      0.000       0.129       0.168

cut_Very_Good     0.1258      0.010     12.269      0.000       0.106       0.146
color_E          -0.0471      0.006     -8.429      0.000      -0.058      -0.036
color_F          -0.0627      0.006    -11.075      0.000      -0.074      -0.052
color_G          -0.1007      0.006    -18.258      0.000      -0.112      -0.090
color_H          -0.2077      0.006    -35.323      0.000      -0.219      -0.196
color_I          -0.3240      0.007    -49.521      0.000      -0.337      -0.311
color_J          -0.4686      0.008    -58.186      0.000      -0.484      -0.453
clarity_IF        0.9998      0.016     62.524      0.000       0.968       1.031
clarity_SI1       0.6390      0.014     46.643      0.000       0.612       0.666
clarity_SI2       0.4296      0.014     31.177      0.000       0.403       0.457
clarity_VS1       0.8381      0.014     59.986      0.000       0.811       0.865
clarity_VS2       0.7660      0.014     55.618      0.000       0.739       0.793
clarity_VVS1      0.9421      0.015     63.630      0.000       0.913       0.971
clarity_VVS2      0.9314      0.014     64.730      0.000       0.903       0.960
==============================================================================
Omnibus:                     4696.785   Durbin-Watson:                   1.994
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            17654.853
Skew:                           1.208   Prob(JB):                         0.00
Kurtosis:                       7.076   Cond. No.                         57.0
==============================================================================
```

Table 1.3.15

Based on p-value and coefficient, Dropping ‘depth’ variable as it not useful for model.

After dropping the depth variable



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.942
Model:                            OLS   Adj. R-squared:                  0.942
Method:                 Least Squares   F-statistic:                 1.390e+04
Date:                Tue, 16 Nov 2021   Prob (F-statistic):               0.00
Time:                        12:38:59   Log-Likelihood:                 2953.5
No. Observations:               18870   AIC:                            -5861.
Df Residuals:                   18847   BIC:                            -5680.
Df Model:                          22
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
Intercept        -0.7567      0.016    -46.991      0.000      -0.788      -0.725
carat             1.1020      0.009    122.331      0.000       1.084       1.120
table            -0.0139      0.002     -6.770      0.000      -0.018      -0.010
x                -0.3156      0.031    -10.101      0.000      -0.377      -0.254
y                 0.2834      0.031      9.069      0.000       0.222       0.345
z                -0.1088      0.014     -7.883      0.000      -0.136      -0.082
cut_Good          0.0951      0.011      8.876      0.000       0.074       0.116
cut_Ideal         0.1512      0.010     14.508      0.000       0.131       0.172

cut_Premium       0.1474      0.010     14.711      0.000       0.128       0.167
cut_Very_Good     0.1255      0.010     12.239      0.000       0.105       0.146
color_E          -0.0471      0.006     -8.439      0.000      -0.058      -0.036
color_F          -0.0627      0.006    -11.082      0.000      -0.074      -0.052
color_G          -0.1007      0.006    -18.246      0.000      -0.111      -0.090
color_H          -0.2076      0.006    -35.306      0.000      -0.219      -0.196
color_I          -0.3237      0.007    -49.497      0.000      -0.337      -0.311
color_J          -0.4684      0.008    -58.169      0.000      -0.484      -0.453
clarity_IF        1.0000      0.016     62.544      0.000       0.969       1.031
clarity_SI1       0.6398      0.014     46.738      0.000       0.613       0.667
clarity_SI2       0.4302      0.014     31.232      0.000       0.403       0.457
clarity_VS1       0.8386      0.014     60.042      0.000       0.811       0.866
clarity_VS2       0.7667      0.014     55.691      0.000       0.740       0.794
clarity_VVS1      0.9424      0.015     63.655      0.000       0.913       0.971
clarity_VVS2      0.9319      0.014     64.784      0.000       0.904       0.960
==============================================================================
Omnibus:                     4699.504   Durbin-Watson:                   1.994
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            17704.272
Skew:                           1.208   Prob(JB):                         0.00
Kurtosis:                       7.084   Cond. No.                         56.5
==============================================================================
```

Table 1.3.16

To ideally bring down the values to lower levels we can drop one of the variables that is highly correlated.

Dropping variables would bring down the multi collinearity level down.

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

We had a business problem to predict the price of the stone and provide insights for the company on the profits on different prize slots.

From the EDA analysis we could understand the cut, ideal cut had number profits to the company.

The colours H, I, J have bought profits for the company.

In clarity if we could see there were no flawless stones and they were no profits coming from I1, I2, I3 stones.

The ideal, premium and very good types of cuts were bringing profits whereas fair and good are not bringing profits.

The predictions were able to capture 95% variations in the price and it is explained by the predictors in the training set.

Using stats model if we could run the model again we can have P values and coefficients which will give us better understanding of the relationship, so that values more 0.05 we can drop those variables and rerun the model again for better results.

For better accuracy dropping depth column in iteration for better results.

The equation, (-0.76) * Intercept + (1.1) * carat + (-0.01) * table + (-0.32) * x + (0.2 8) * y + (-0.11) * z + (0.1) * cut_Good + (0.15) * cut_Ideal + (0.15) * cut_Premiu m + (0.13) * cut_Very_Good + (-0.05) * color_E + (-0.06) * color_F + (-0.1) * color _G + (-0.21) * color_H + (-0.32) * color_I + (-0.47) * color_J + (1.0) * clarity_IF + (0.64) * clarity_SI1 + (0.43) * clarity_SI2 + (0.84) * clarity_VS1 + (0.77) * clarity_ VS2 + (0.94) * clarity_VVS1 + (0.93) * clarity_VVS2

Table 1.4.1

PROBLEM 2

Executive Summary:

You are hired by a tour and travel agency which deals in selling holiday packages. You are provided details of 872 employees of a company. Among these employees, some opted for the package, and some didn't. You must 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 based on which the company will focus on employees to sell their packages.

Holiday_Package.csv

Data Introduction:

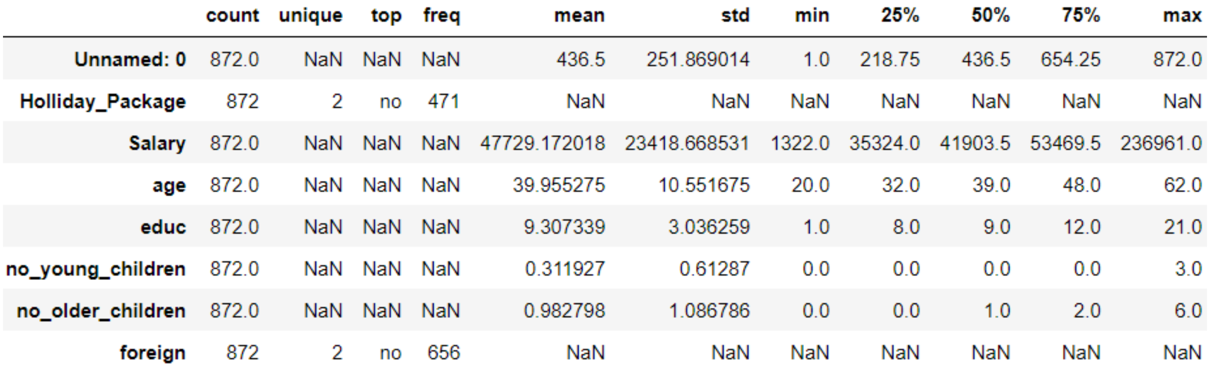
The purpose of this whole exercise is to explore the dataset and is recommended for learning and practicing our skills using Logistic Regression Model and Linear Discriminant Analysis.

The dataset contains 872 rows and 8 columns.

Data Description:

Description of variables is as follows:

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



	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Unnamed: 0	872.0	NaN	NaN	NaN	436.5	251.869014	1.0	218.75	436.5	654.25	872.0
Holliday_Package	872	2	no	471	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Salary	872.0	NaN	NaN	NaN	47729.172018	23418.668531	1322.0	35324.0	41903.5	53469.5	236961.0
age	872.0	NaN	NaN	NaN	39.955275	10.551675	20.0	32.0	39.0	48.0	62.0
educ	872.0	NaN	NaN	NaN	9.307339	3.036259	1.0	8.0	9.0	12.0	21.0
no_young_children	872.0	NaN	NaN	NaN	0.311927	0.61287	0.0	0.0	0.0	0.0	3.0
no_older_children	872.0	NaN	NaN	NaN	0.982798	1.086786	0.0	0.0	1.0	2.0	6.0
foreign	872	2	no	656	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 2.0

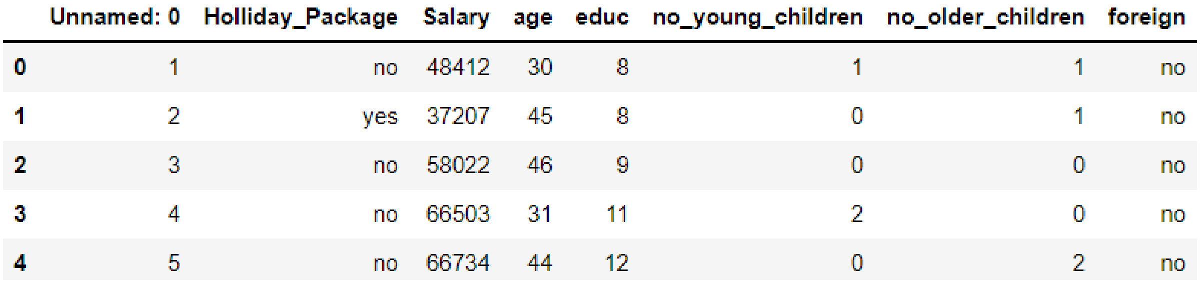
Observation:

We have integer and continuous data.

Holiday package is our target variable.

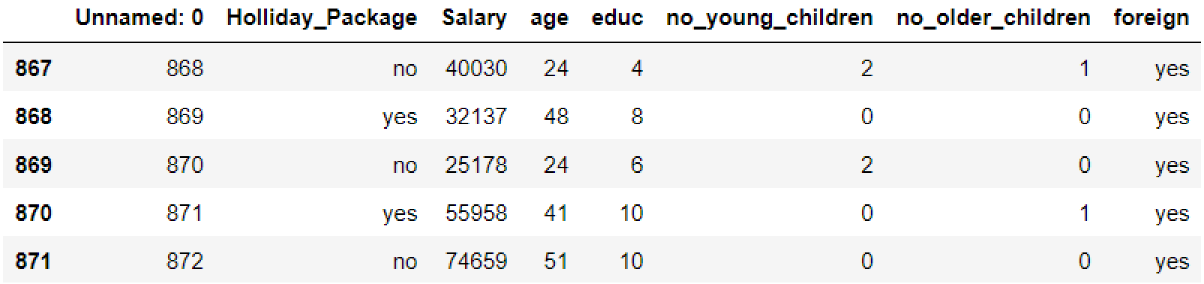
Salary, age, edu and number young children, number older children of employee have the went to foreign, these are the attributes we have to cross examine and help the company predict weather the person will opt for holiday package or not.

Sample 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

Table 2.1



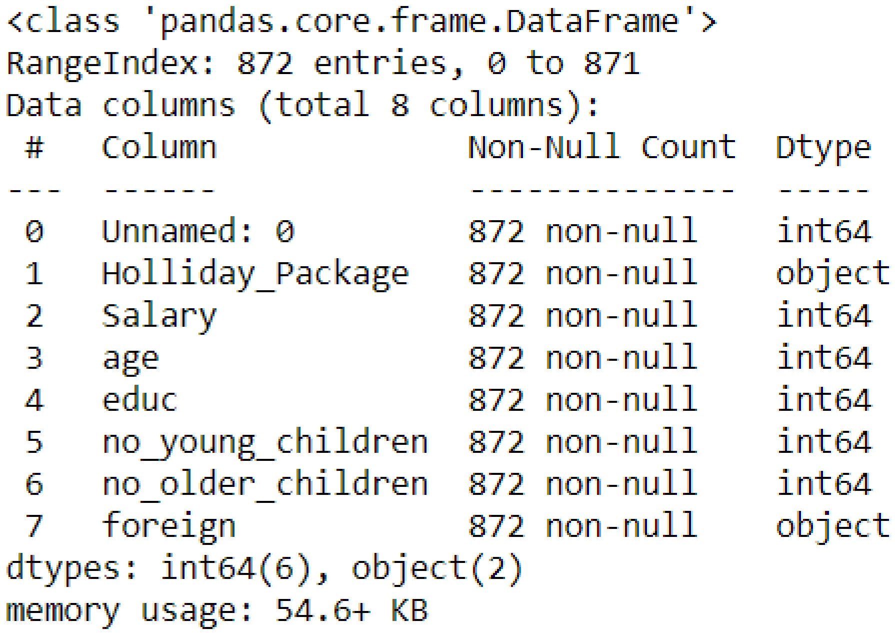
	Unnamed: 0	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
867	868	no	40030	24	4	2	1	yes
868	869	yes	32137	48	8	0	0	yes
869	870	no	25178	24	6	2	0	yes
870	871	yes	55958	41	10	0	1	yes
871	872	no	74659	51	10	0	0	yes

Table 2.2

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:

Check for types of variables in the data frame:



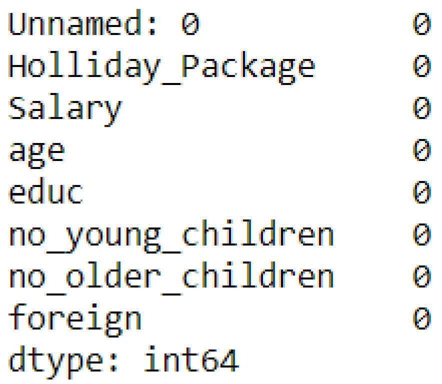
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 872 entries, 0 to 871
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Unnamed: 0         872 non-null    int64 
 1   Holliday_Package   872 non-null    object
 2   Salary             872 non-null    int64 
 3   age                872 non-null    int64 
 4   educ               872 non-null    int64 
 5   no_young_children  872 non-null    int64 
 6   no_older_children  872 non-null    int64 
 7   foreign            872 non-null    object
dtypes: int64(6), object(2)
memory usage: 54.6+ KB
```

Table 2.1.1

Observation:

We have int and object data types in the given data.
There are no null values in the dataset

Check for missing values in the dataset:



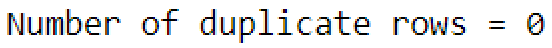
```
Unnamed: 0           0
Holliday_Package     0
Salary               0
age                  0
educ                 0
no_young_children    0
no_older_children    0
foreign              0
dtype: int64
```

Table 2.1.2

Observation:

There are no missing values in the given dataset.

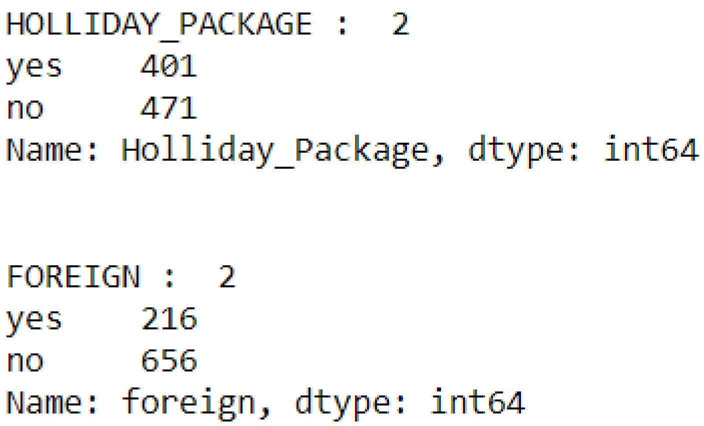
Check for duplicate values in the dataset:



```
Number of duplicate rows = 0
```

Table 2.1.3

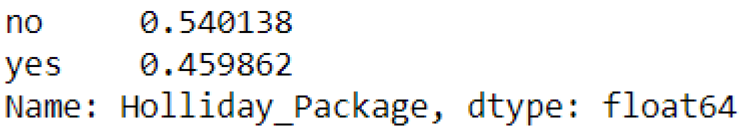
Unique values in the Categorical data:



```
HOLLIDAY_PACKAGE :  2
yes    401
no     471
Name: Holliday_Package, dtype: int64


FOREIGN :  2
yes    216
no     656
Name: foreign, dtype: int64
```

Table 2.1.4

Percentage of the Target Variable:



```
no      0.540138
yes     0.459862
Name: Holliday_Package, dtype: float64
```

Table 2.1.5

Observation:

The above split indicates that 45% of employees are interested in holiday package.

Univariate Analysis

Foreign

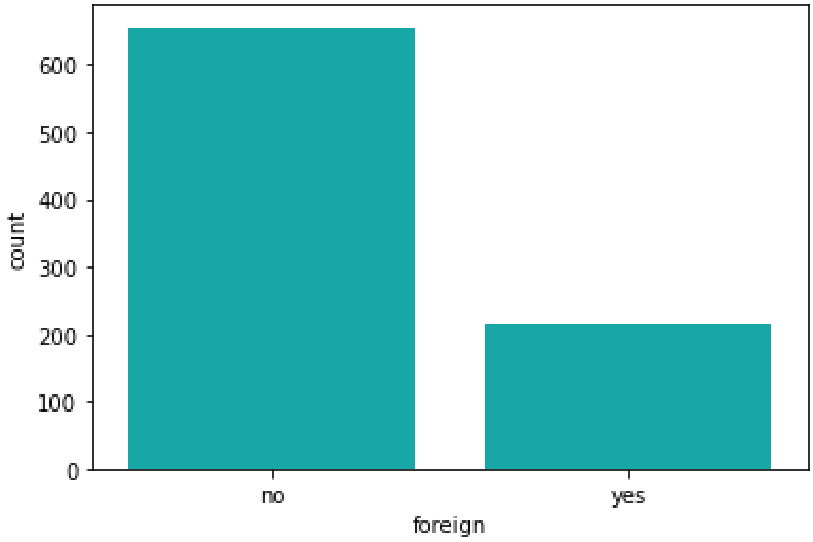


Fig 2.1.1

We can see the number of non-foreigners count is more than foreigner count.

Holiday Package

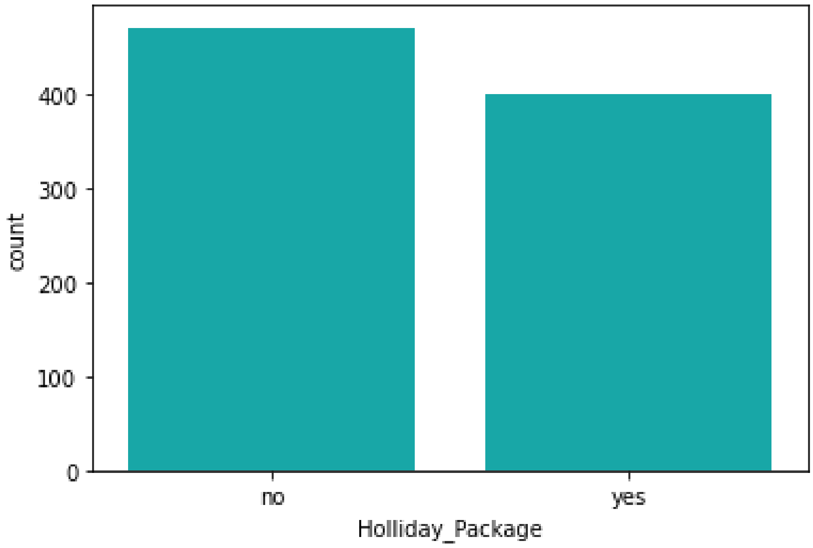


Fig 2.1.2

We can see the employees count who are not willing to opt for holiday package is more than employees who opted for holiday package.

Holiday Package Vs Salary

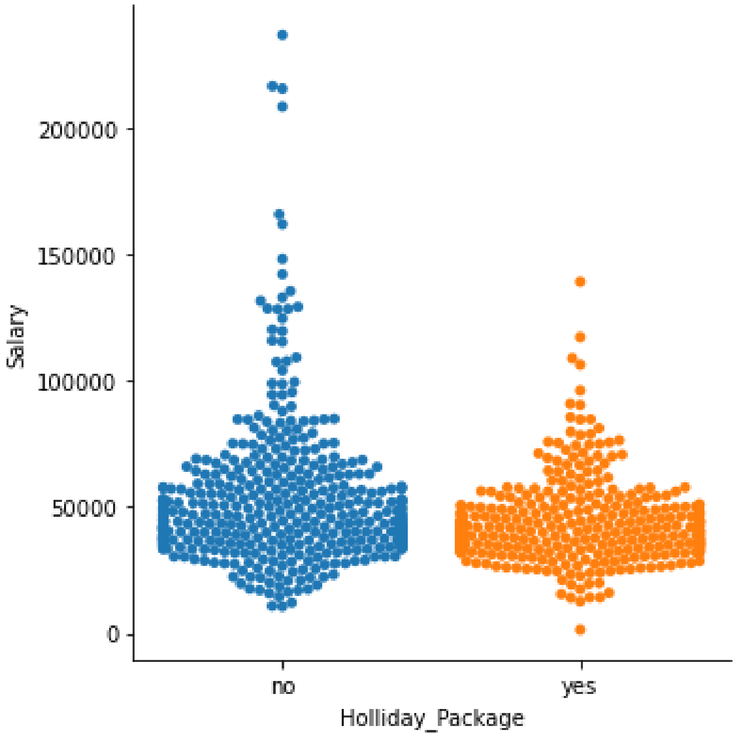


Fig 2.1.3

We can see employee below salary 150000 have always opted for holiday package.

Holiday Package Vs Age

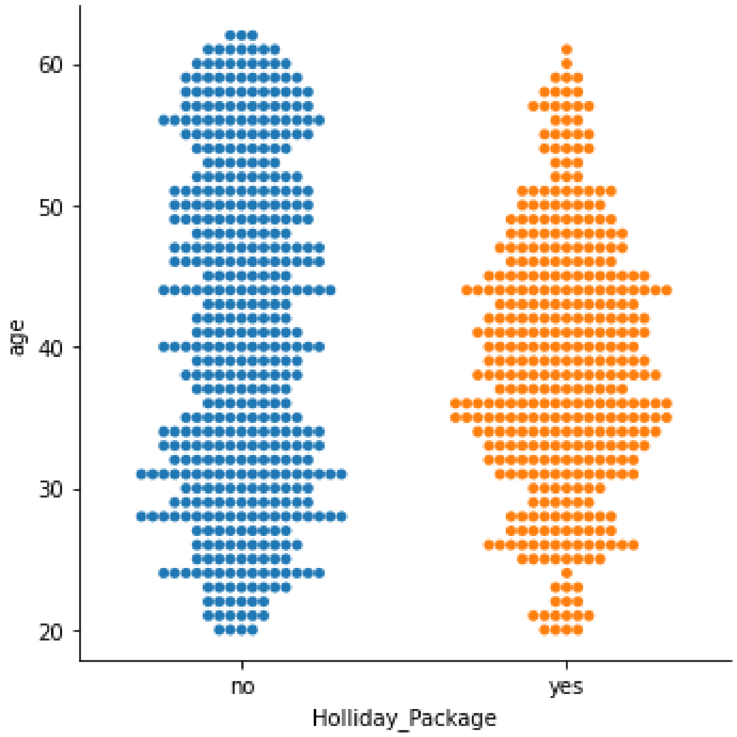


Fig 2.1.4

We can see employees with all age groups, who has not opted for holiday package and who opted holiday package are almost equal.

Holiday Package Vs Educ

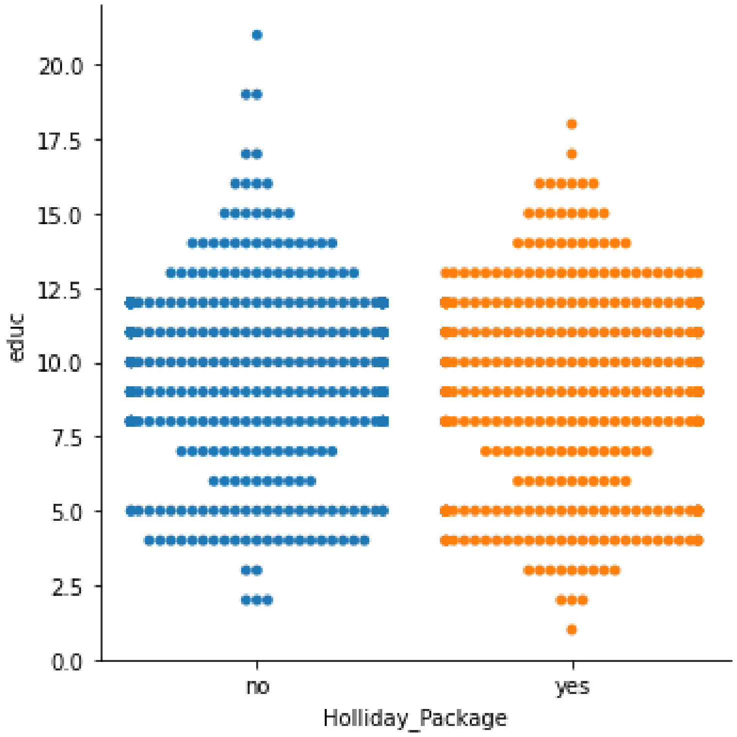


Fig 2.1.5

We can see employees with less formal education years has opted for holiday package.

Holiday Package Vs Young Children

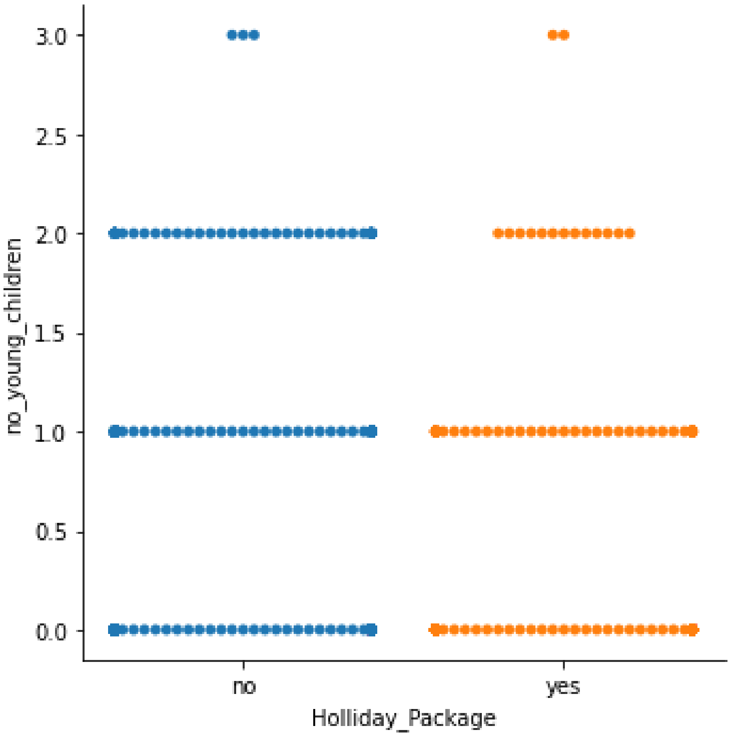


Fig 2.1.6

We can see employees with young children, who opted for holiday package and who has not opted for holiday package are equal.

Holiday Package Vs Old Children

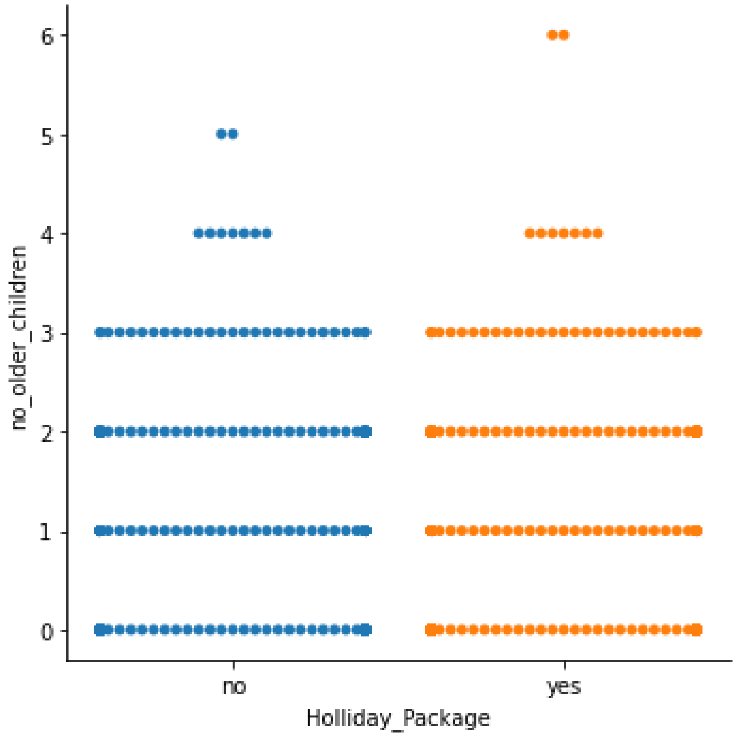


Fig 2.1.7

We can see employees with older children, who has opted for holiday package are more than who has not opted for holiday package.

Age Vs Salary Vs Holiday Package

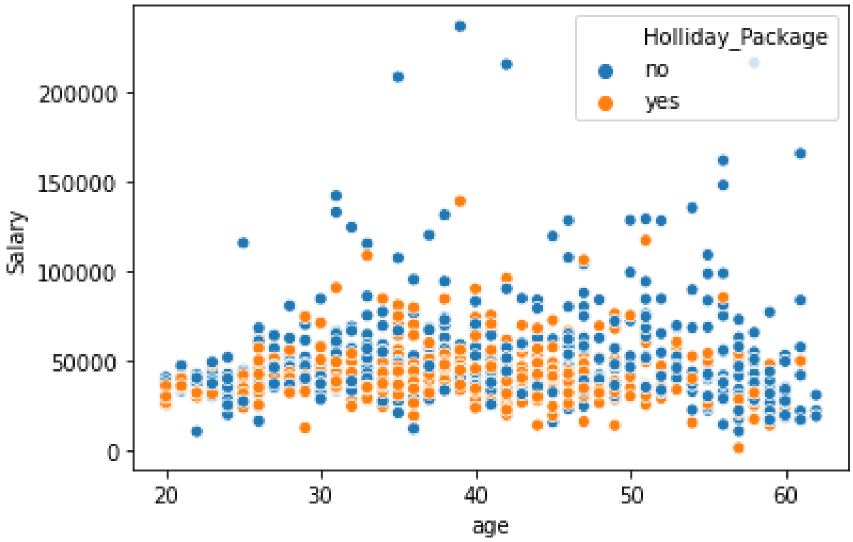


Fig 2.1.8

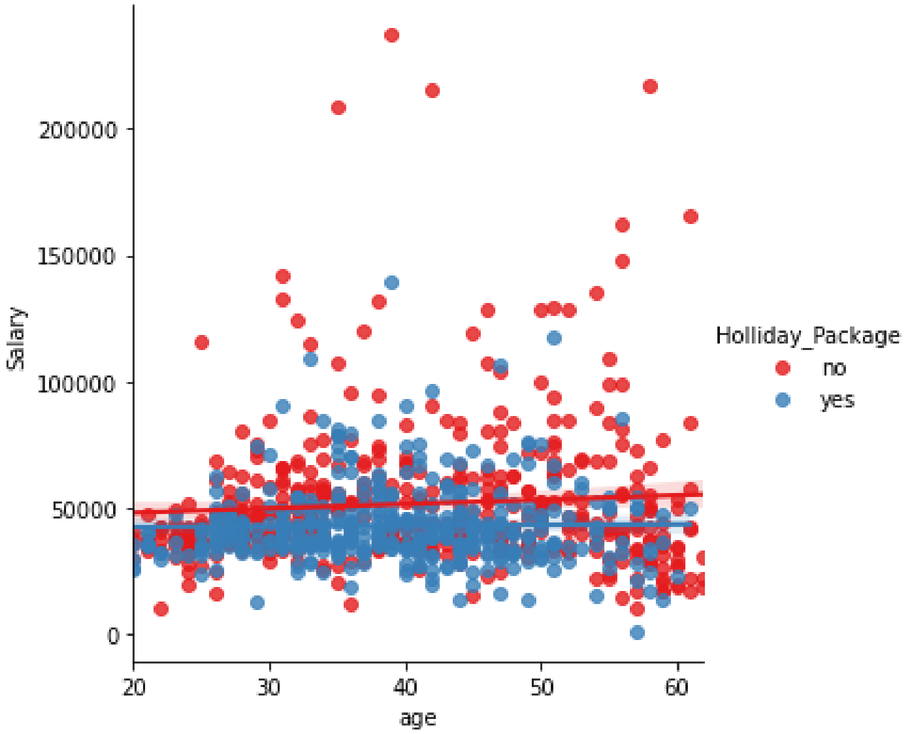


Fig 2.1.9

Employee age over 50 to 60 have seems to be not taking the holiday package, whereas in the age 30 to 50 and salary less than 50000 people have opted more for holiday package.

Educ Vs Salary Vs Holiday Package

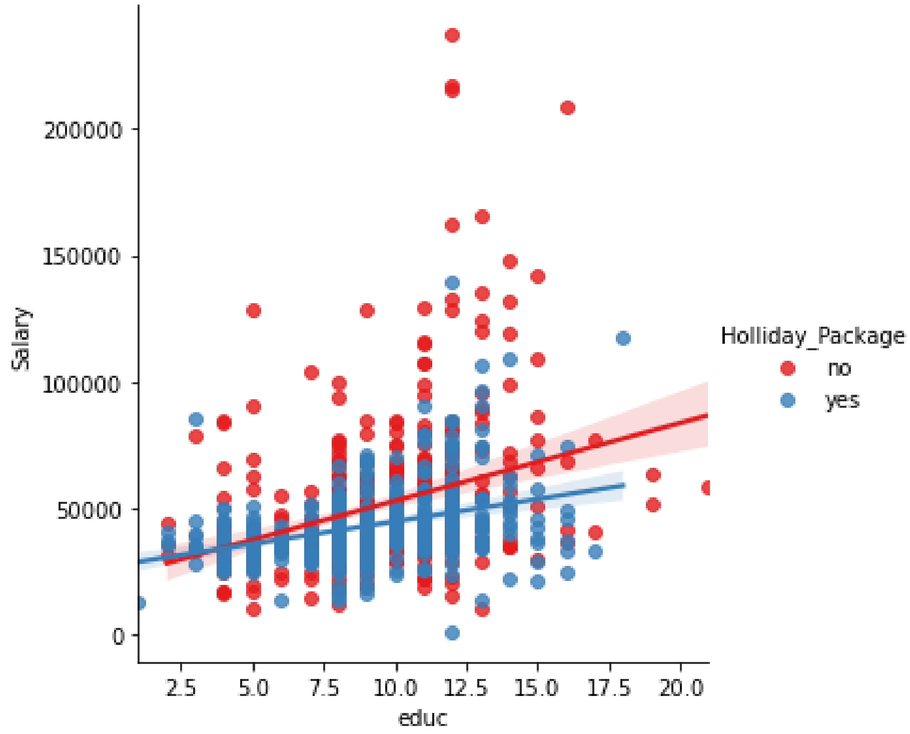


Fig 2.1.10

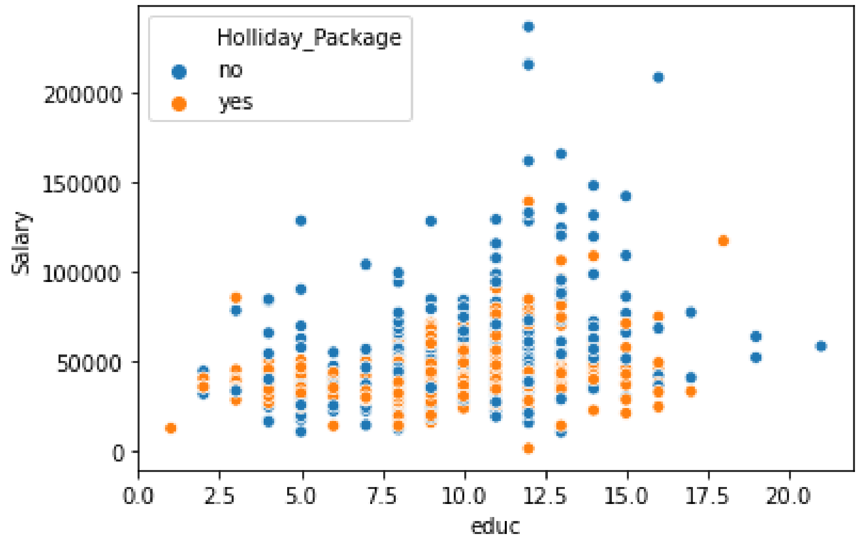


Fig 2.1.11

We can see employees with salary above 100000 and who has a greater number of years in formal education has not opted for holiday package.

Young Children Vs Age Vs Holiday Package

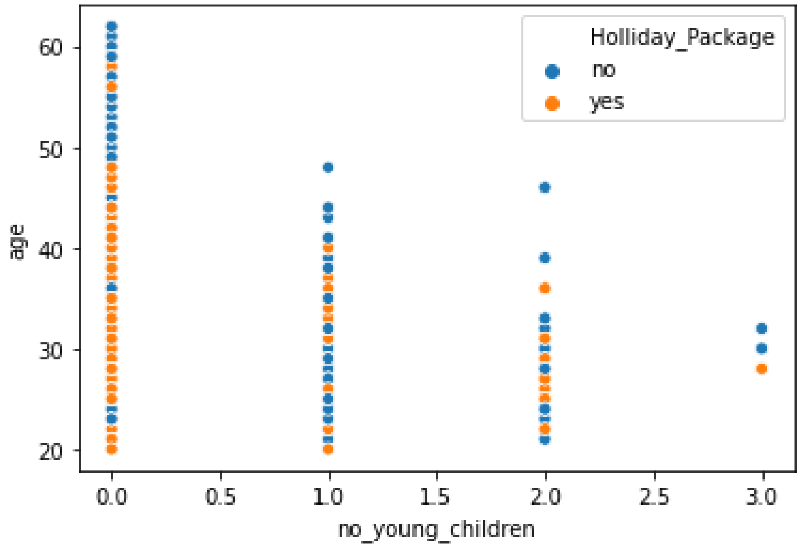


Fig 2.1.12

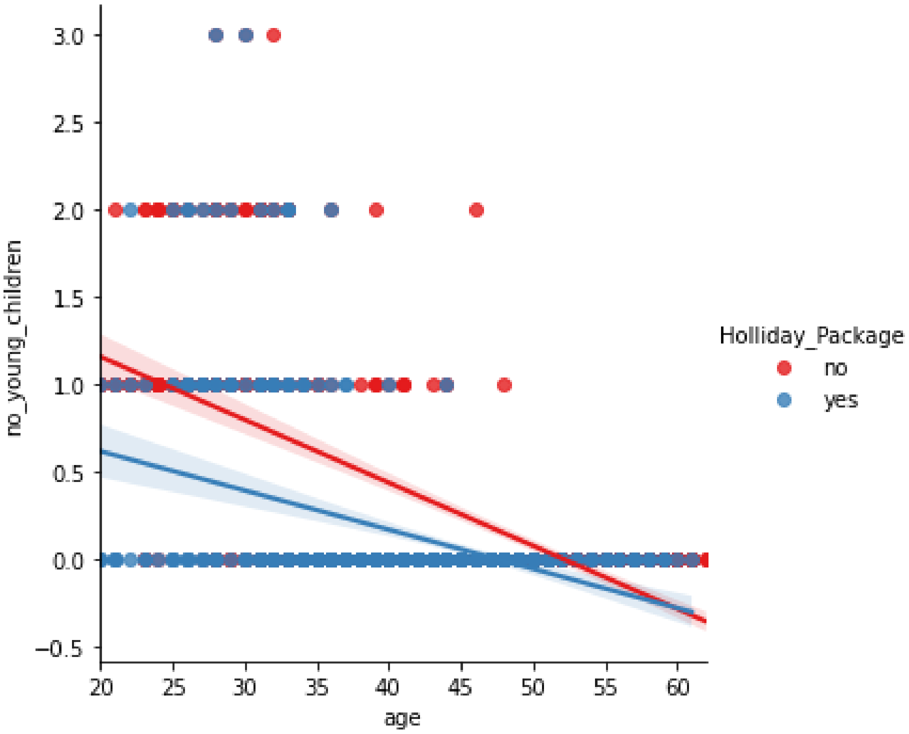


Fig 2.1.13

We can see employees with young children and whose age is 50 years and above has not opted for holiday package.

Old Children Vs Age Vs Holiday Package

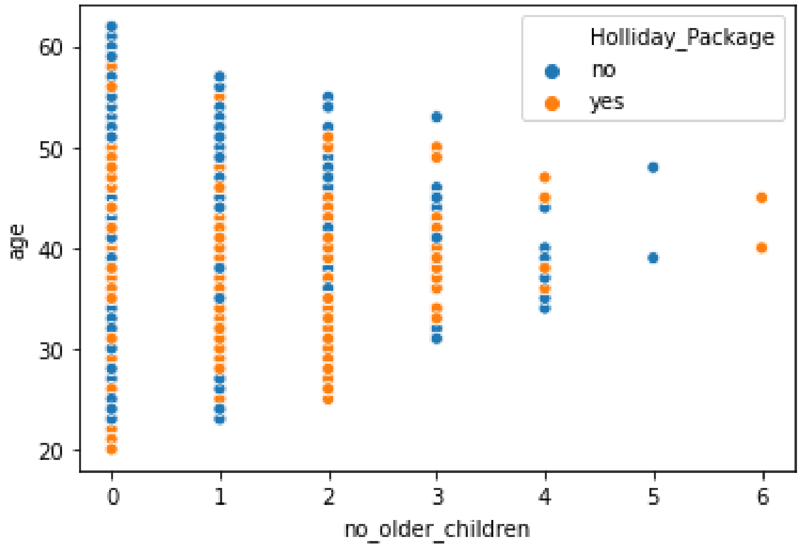


Fig 2.1.14

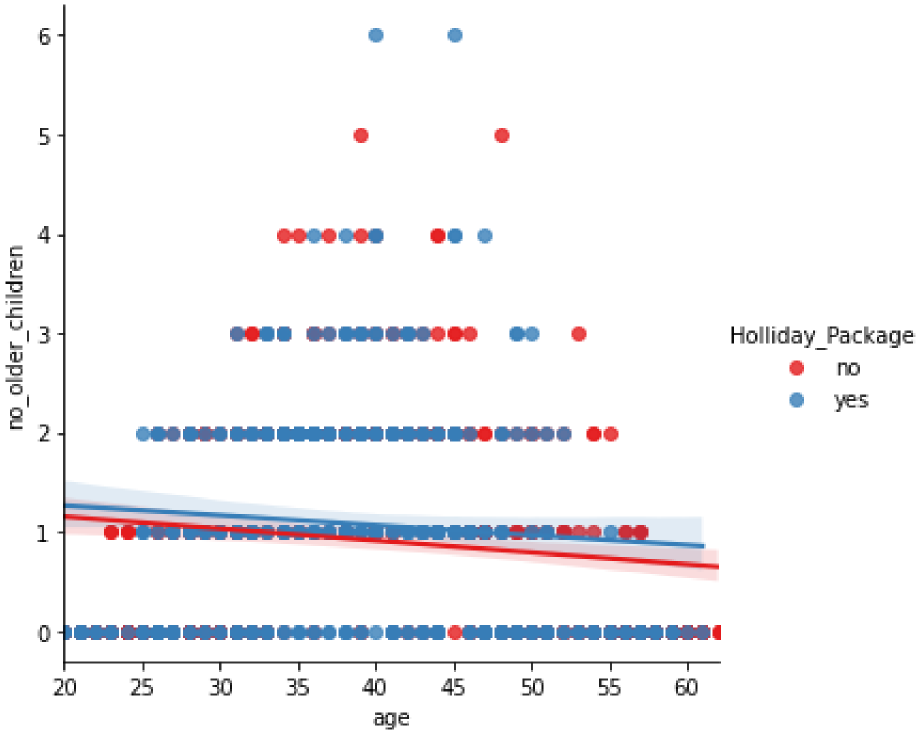


Fig 2.1.15

We can see employees with older children and with age up to 50 years has opted for holiday package compared to employees with older children with age 50 years and above.

Bivariate Analysis

Data Distribution

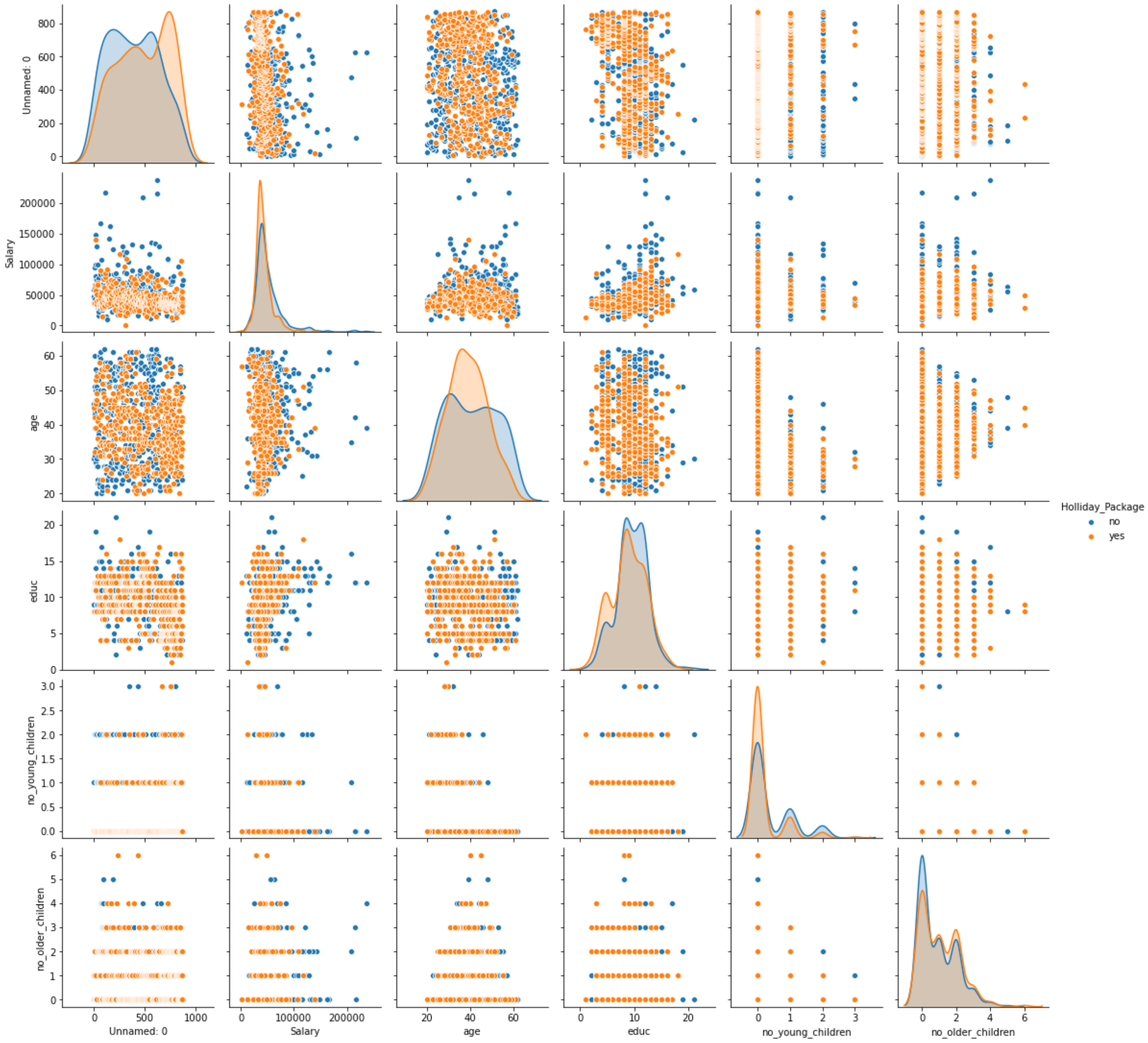


Fig 2.1.16

There is no correlation between the data, the data seems to be normal.

There is no huge difference in the data distribution among the holiday package, I don’t see any clear two different distribution in the data.

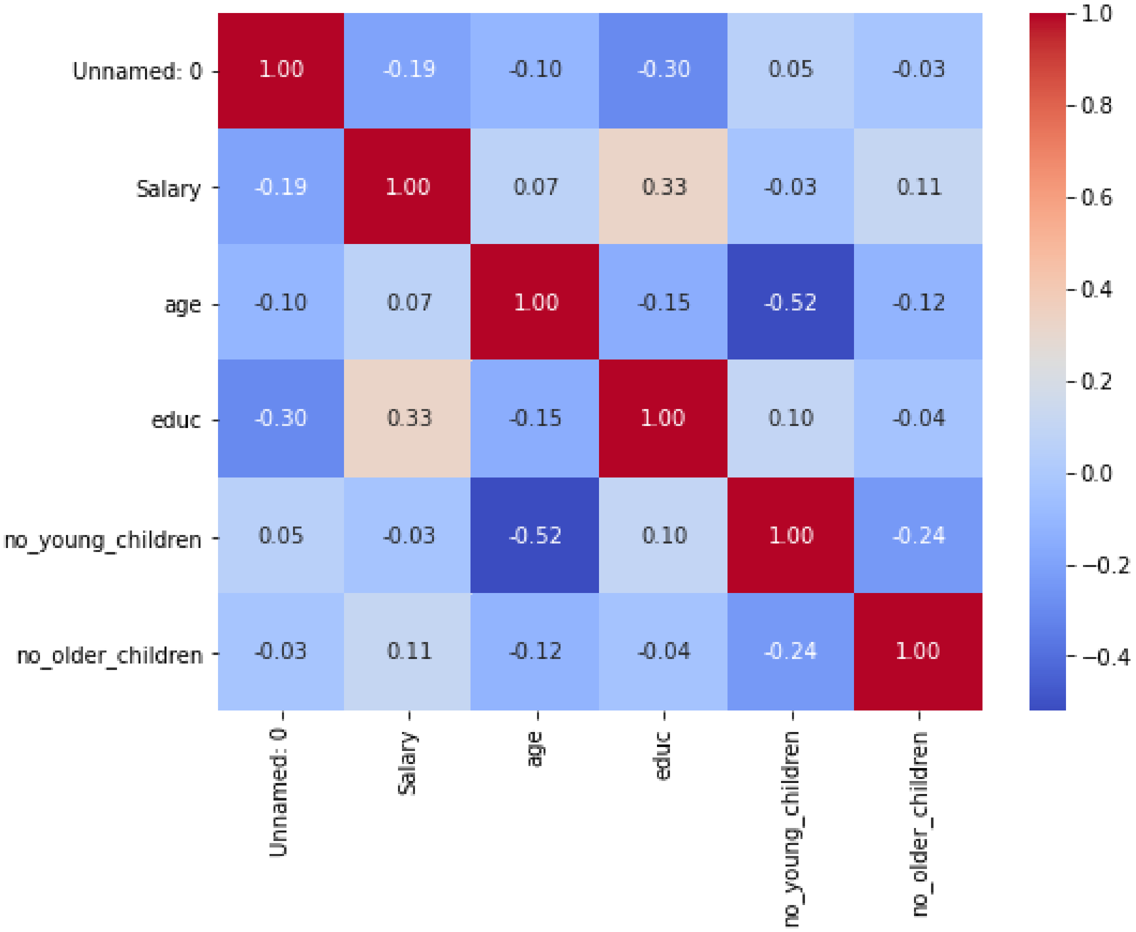


Fig 2.1.17

We can see there is no multi collinearity in the data.

Treating Outliers

Before Outlier Treatment

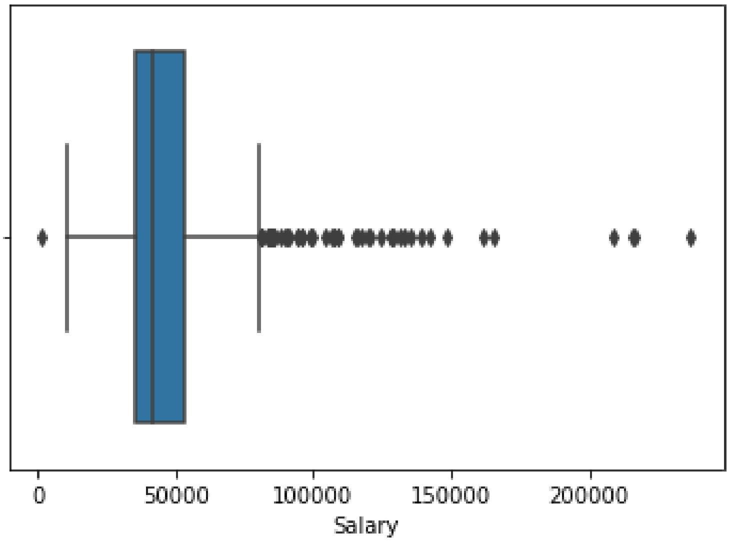


Fig 2.1.18

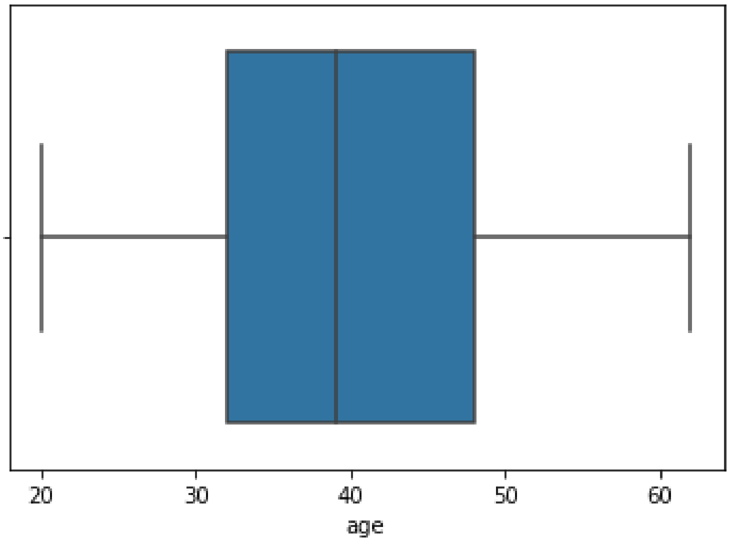


Fig 2.1.19

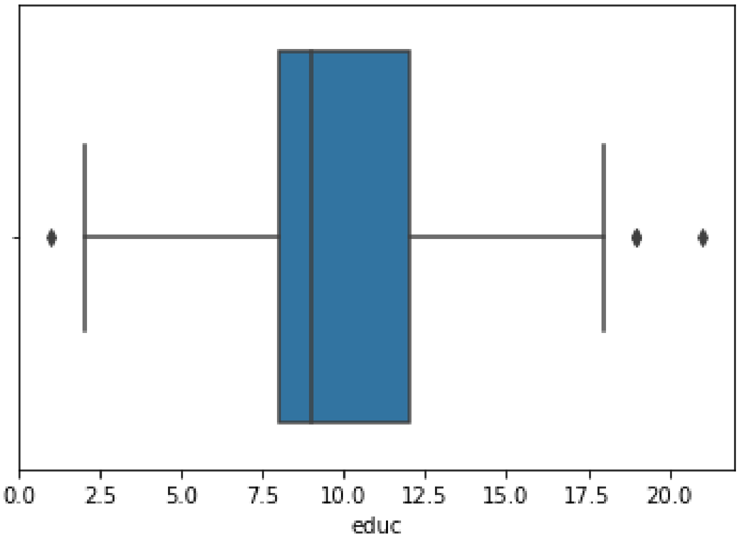


Fig 2.1.20

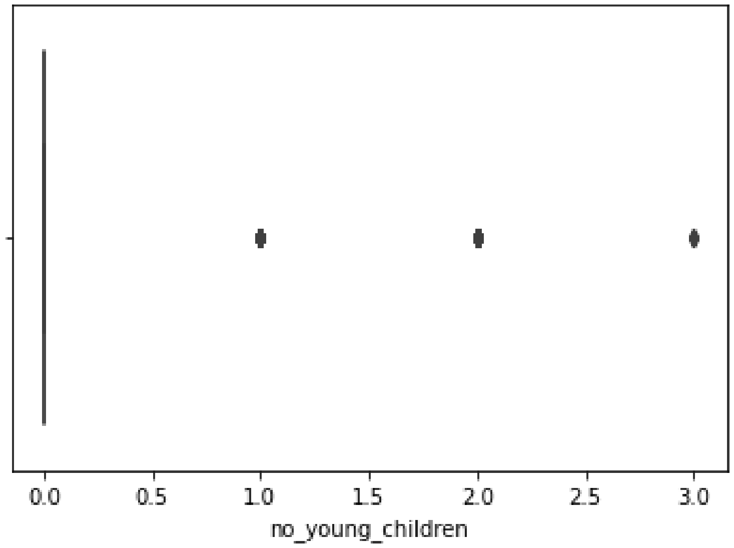


Fig 2.1.21

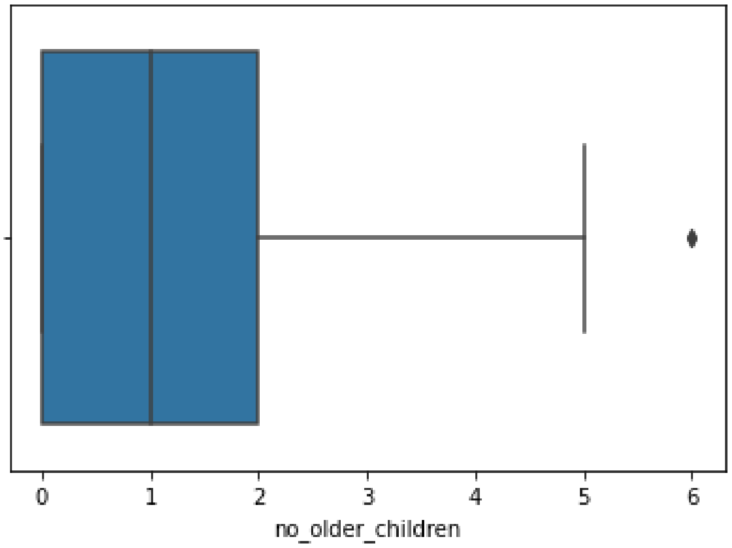


Fig 2.1.22

We have outliers in the dataset, as LDA works based on numerical computation treating outliers will help perform the model better.

After Outlier Treatment

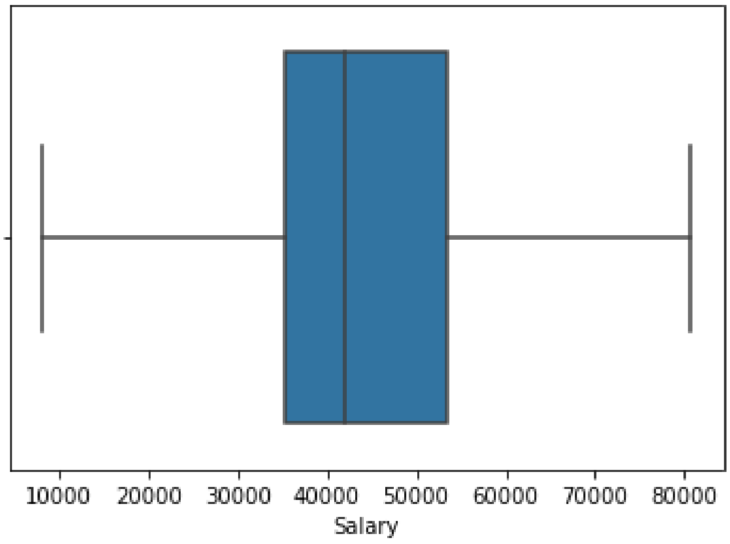


Fig 2.1.23

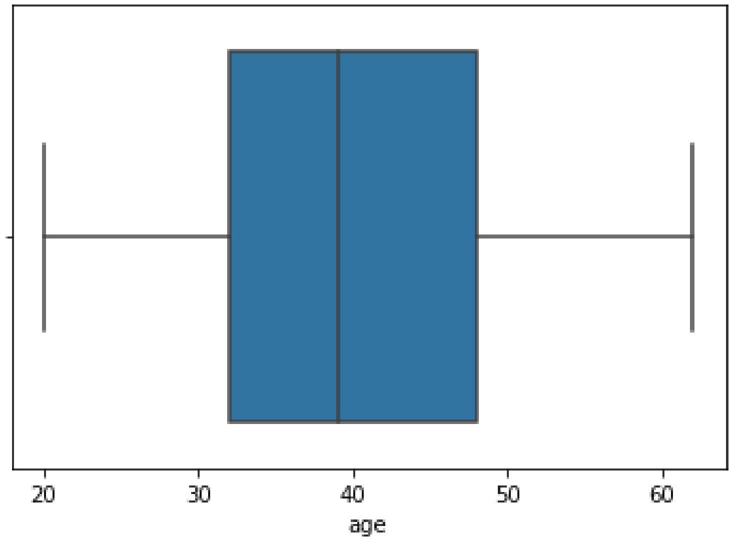


Fig 2.1.24

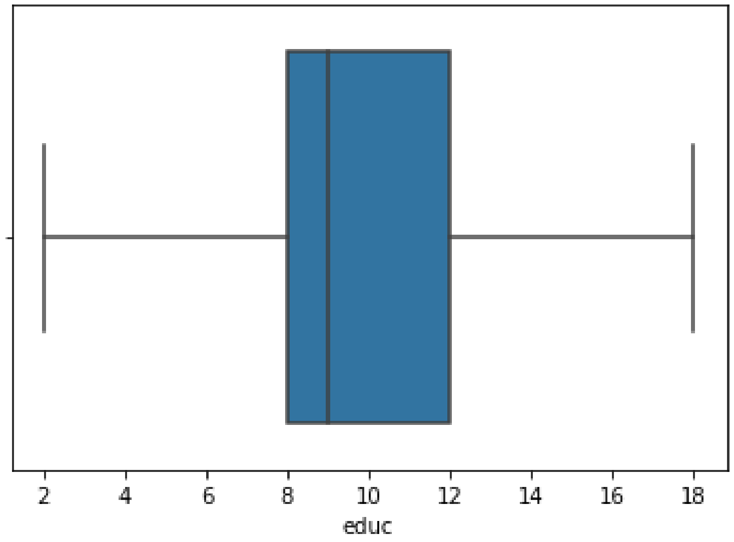


Fig 2.1.25

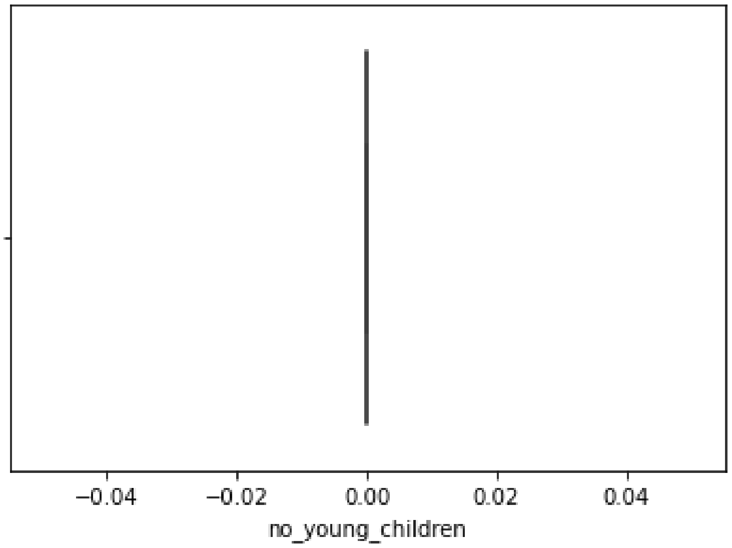


Fig 2.1.26

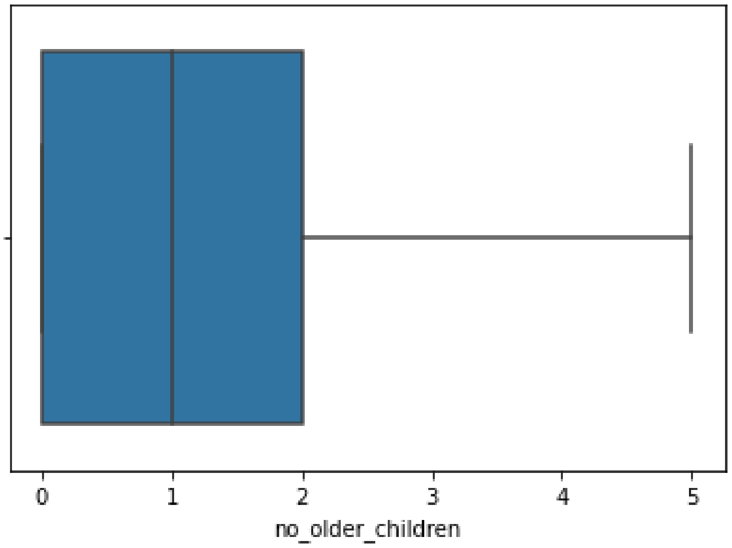
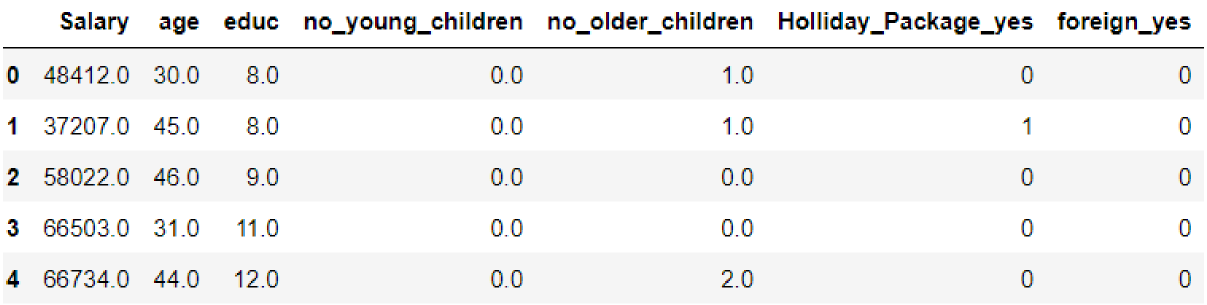


Fig 2.1.27

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).

Building Logistic Regression Model

Encoding Categorical Variable

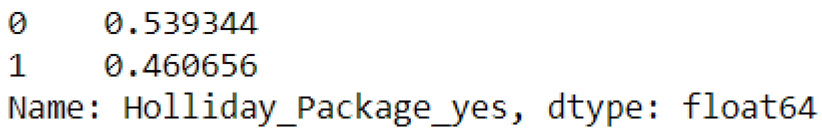


	Salary	age	educ	no_young_children	no_older_children	Holliday_Package_yes	foreign_yes
0	48412.0	30.0	8.0	0.0	1.0	0	0
1	37207.0	45.0	8.0	0.0	1.0	1	0
2	58022.0	46.0	9.0	0.0	0.0	0	0
3	66503.0	31.0	11.0	0.0	0.0	0	0
4	66734.0	44.0	12.0	0.0	2.0	0	0

Table 2.2.1

The encoding helps the logistic regression model predict better results.

Train/Test Split

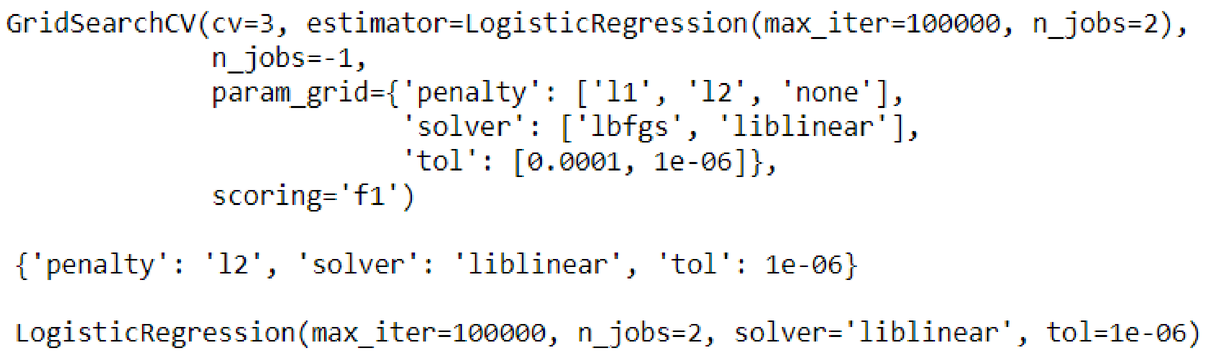


```
0    0.539344
1    0.460656
Name: Holliday_Package_yes, dtype: float64
```

Table 2.2.2

Grid Search Method

The grid search method is used for logistic regression to find the optimal solving and the parameters for solving.

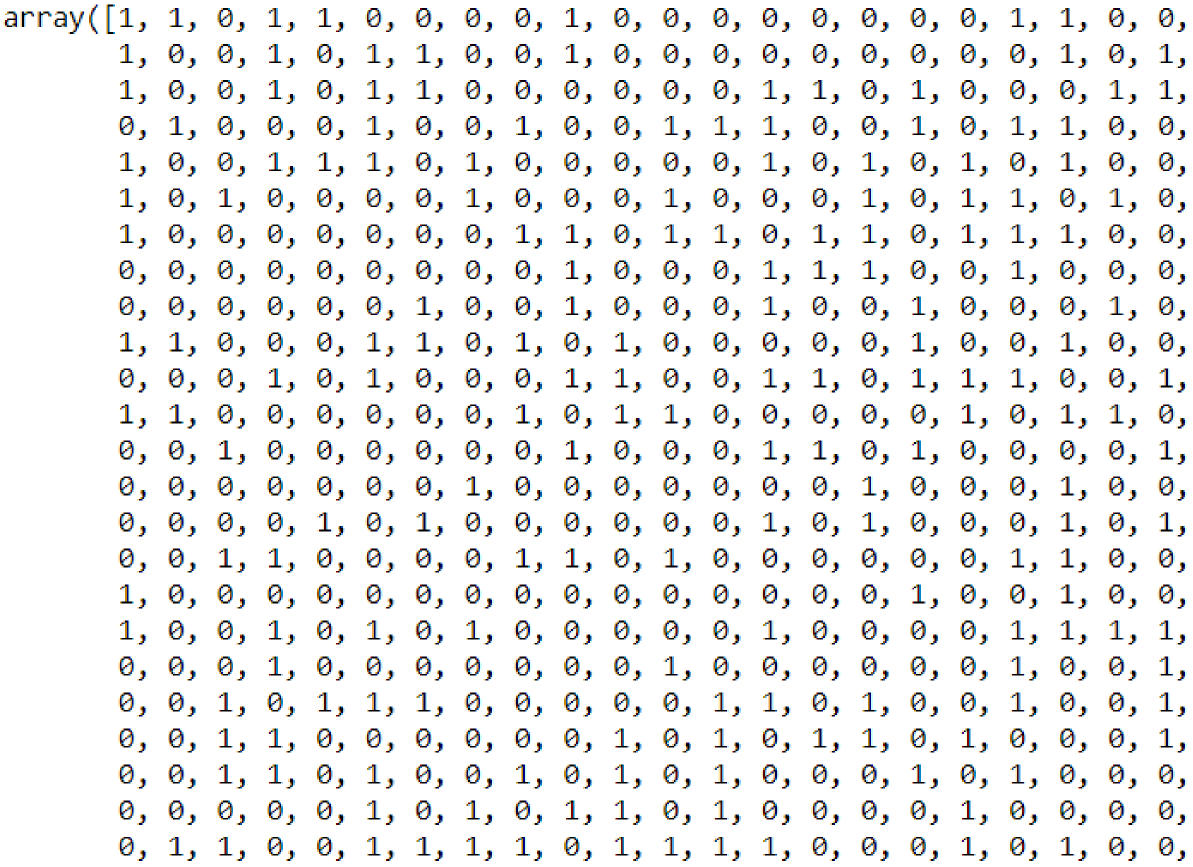


```
GridSearchCV(cv=3, estimator=LogisticRegression(max_iter=100000, n_jobs=2),
             n_jobs=-1,
             param_grid={'penalty': ['l1', 'l2', 'none'],
                         'solver': ['lbfgs', 'liblinear'],
                         'tol': [0.0001, 1e-06]},
             scoring='f1')

{'penalty': 'l2', 'solver': 'liblinear', 'tol': 1e-06}

LogisticRegression(max_iter=100000, n_jobs=2, solver='liblinear', tol=1e-06)
```

Table 2.2.3

The grid search method gives, liblinear solver which is suitable for small datasets. Tolerance and penalty have been found using grid search method

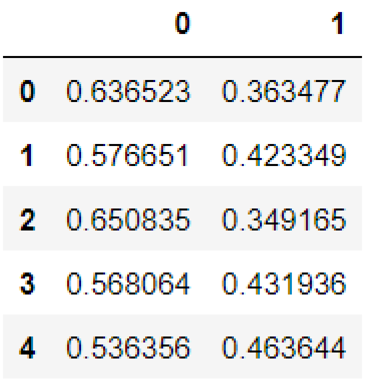
Predicting Training Data



```
array([1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
       1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1,
       0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0,
       1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0,
       1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0,
       1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1,
       1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1,
       0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
       1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1,
       0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1,
       0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0,
```

Table 2.2.4

Probabilities on Test Data



	0	1
0	0.636523	0.363477
1	0.576651	0.423349
2	0.650835	0.349165
3	0.568064	0.431936
4	0.536356	0.463644

Table 2.2.5

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.

Logistic Regression

Confusion Matrix for Train Data

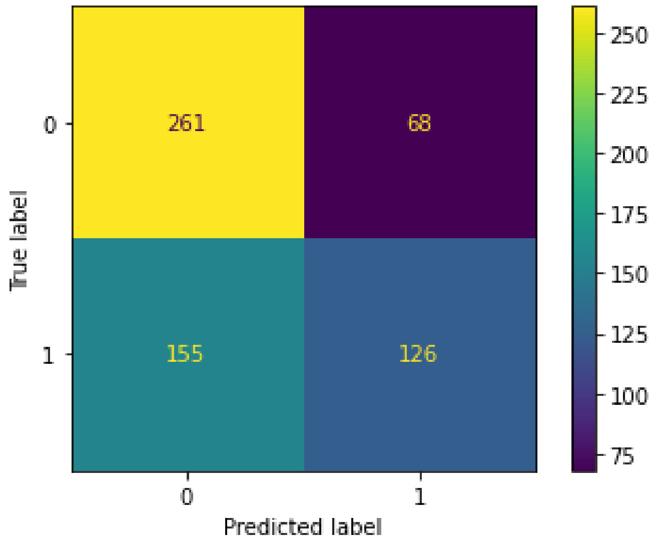


Fig 2.3.1

Confusion Matrix for Test Data

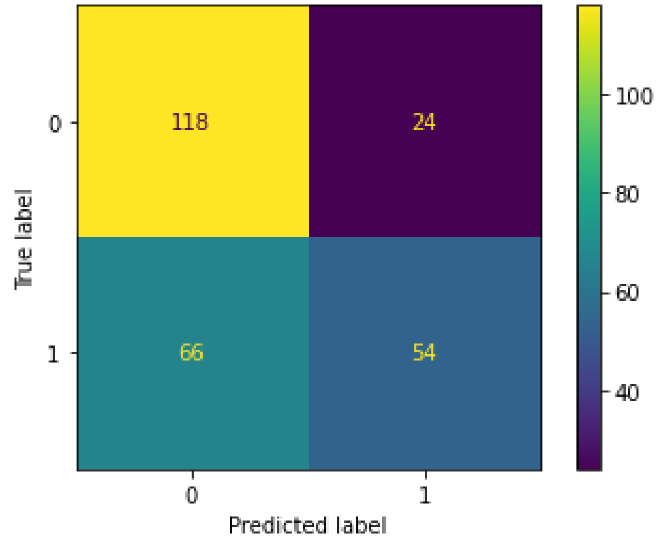
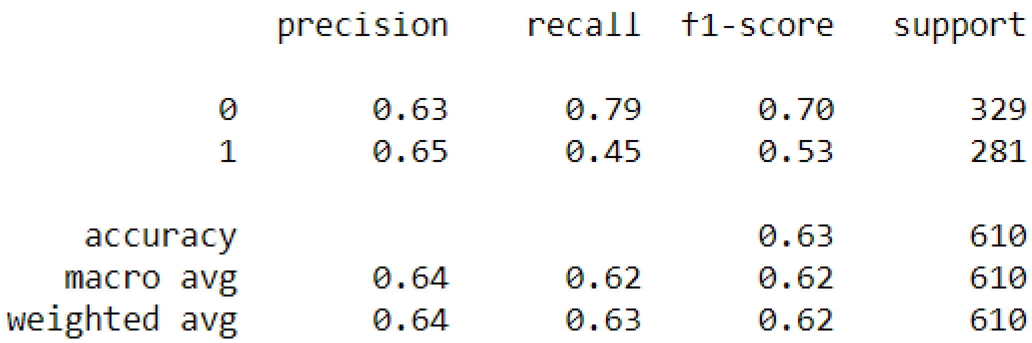


Fig 2.3.2

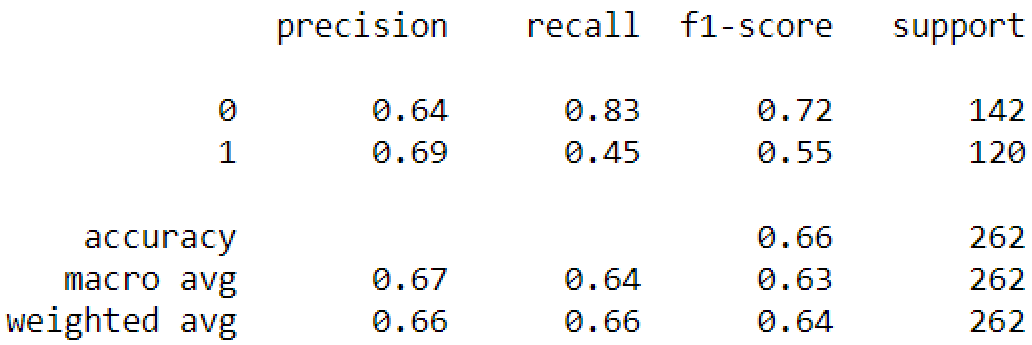
Classification report for Train Data



```
              precision    recall  f1-score   support

           0       0.63      0.79      0.70       329
           1       0.65      0.45      0.53       281

    accuracy                           0.63       610
   macro avg       0.64      0.62      0.62       610
weighted avg       0.64      0.63      0.62       610
```

Table 2.3.1

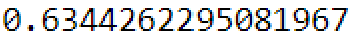
Classification report for Test Data



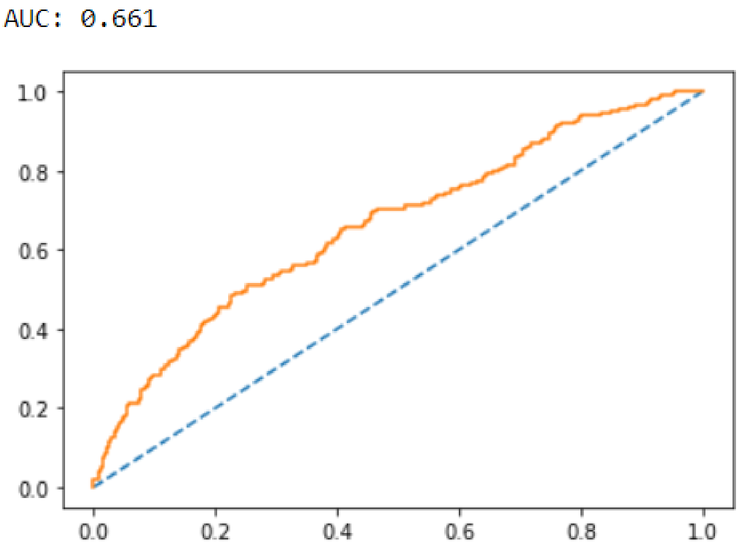
```
              precision    recall  f1-score   support

           0       0.64      0.83      0.72       142
           1       0.69      0.45      0.55       120

    accuracy                           0.66       262
   macro avg       0.67      0.64      0.63       262
weighted avg       0.66      0.66      0.64       262
```

Table 2.3.2

Accuracy, AUC and ROC for Train Data



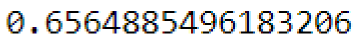
```
0.6344262295081967
```



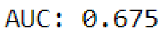
```
AUC: 0.661
```

Fig 2.3.3

Accuracy, AUC and ROC for Test Data



```
0.6564885496183206
```



```
AUC: 0.675
```

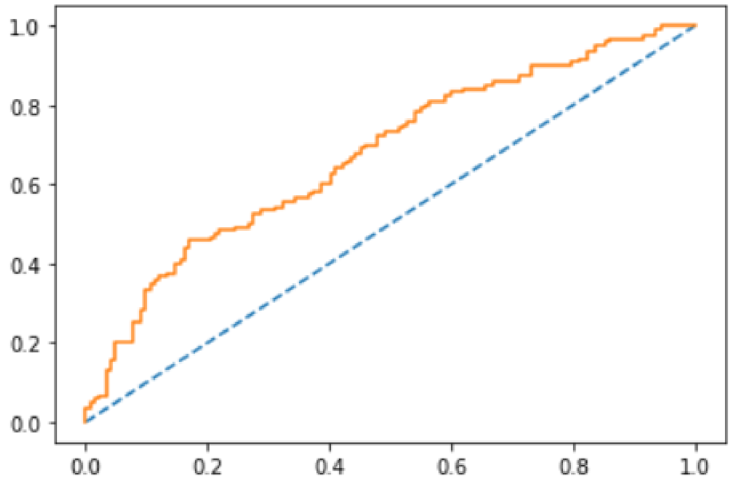
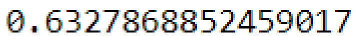


Fig 2.3.4

LDA Model

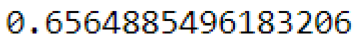
Model Score for Train Data



```
0.6327868852459017
```

Table 2.3.3

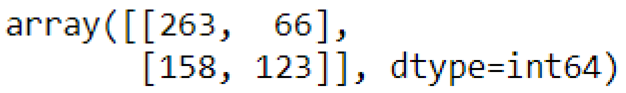
Model Score for Test Data



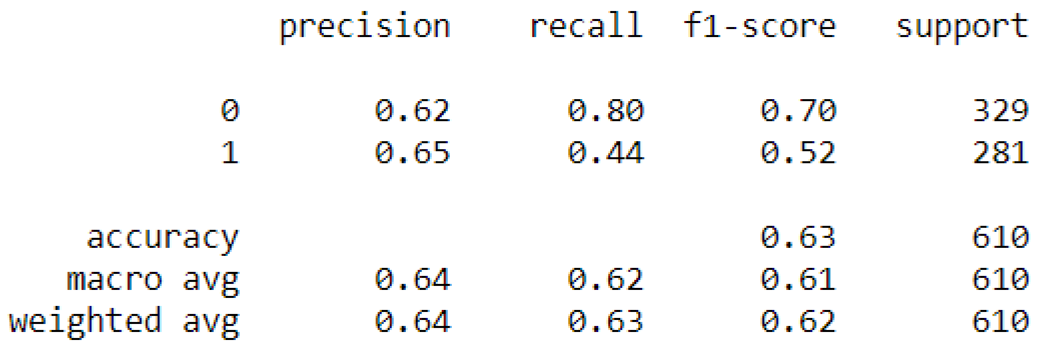
```
0.6564885496183206
```

Table 2.3.4

Confusion Matrix and Classification Report on Train Data



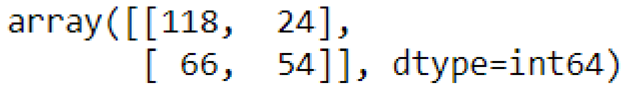
```
array([[263,  66],
       [158, 123]], dtype=int64)
```



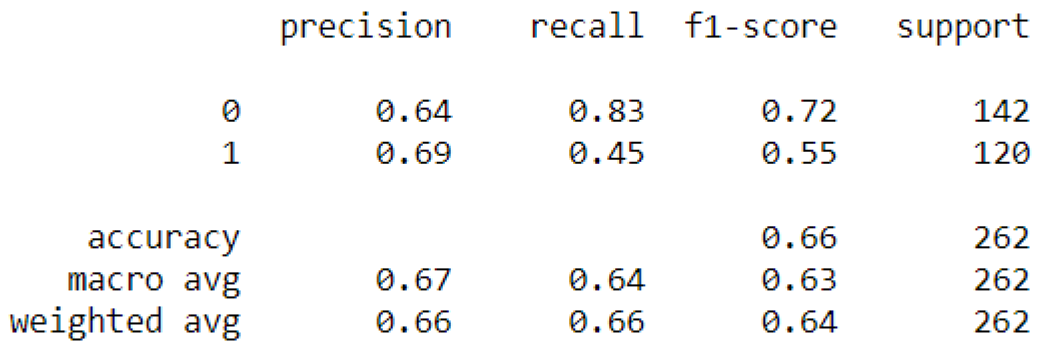
```
              precision    recall  f1-score   support

           0       0.62      0.80      0.70       329
           1       0.65      0.44      0.52       281

    accuracy                           0.63       610
   macro avg       0.64      0.62      0.61       610
weighted avg       0.64      0.63      0.62       610
```

Table 2.3.5

Confusion Matrix and Classification Report on Test Data



```
array([[118,  24],
       [ 66,  54]], dtype=int64)
```



```
              precision    recall  f1-score   support

           0       0.64      0.83      0.72       142
           1       0.69      0.45      0.55       120

    accuracy                           0.66       262
   macro avg       0.67      0.64      0.63       262
weighted avg       0.66      0.66      0.64       262
```

Table 2.3.6

Changing the cut off value to check optimal value that gives better Accuracy and F1 score

0.1

Accuracy Score 0.4607
F1 Score 0.6308

Confusion Matrix

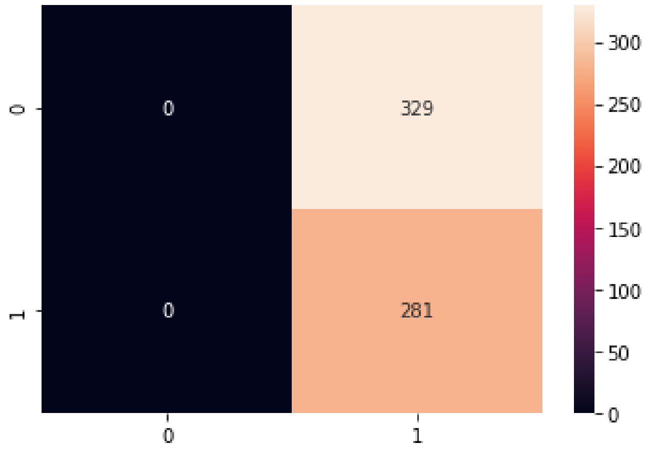


Fig 2.3.5

0.2

Accuracy Score 0.4738
F1 Score 0.6365

Confusion Matrix

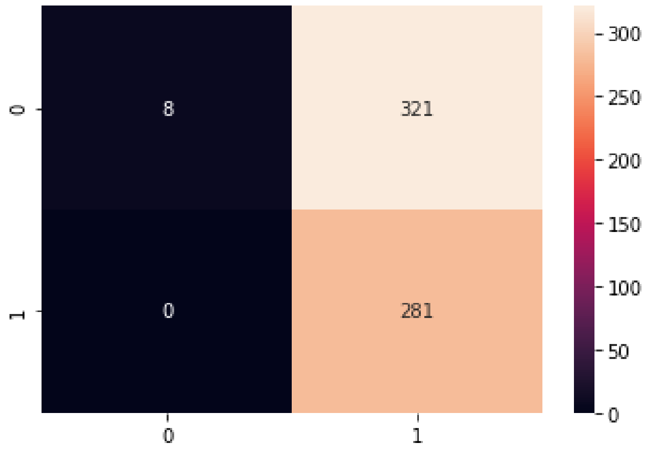


Fig 2.3.6

0.3

Accuracy Score 0.5344
F1 Score 0.6485

Confusion Matrix

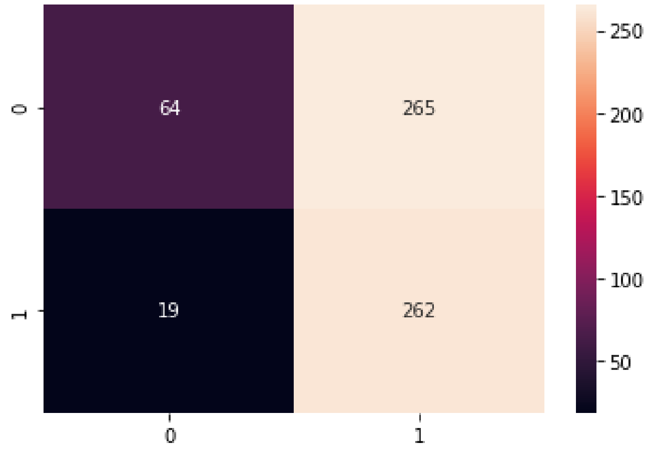


Fig 2.3.7

0.4

Accuracy Score 0.5787
F1 Score 0.6088

Confusion Matrix

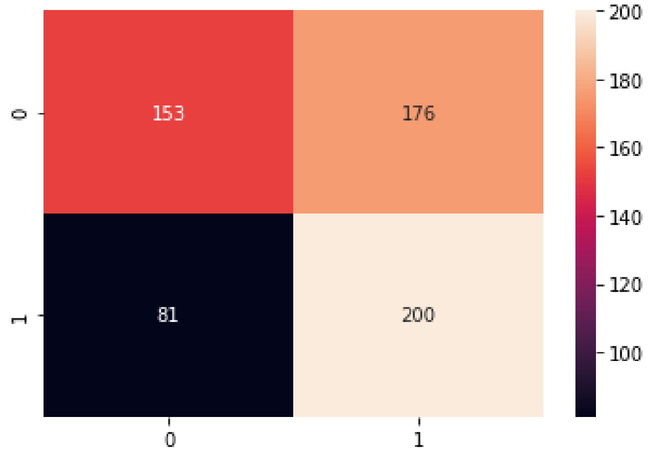


Fig 2.3.8

0.5

Accuracy Score 0.6328
F1 Score 0.5234

Confusion Matrix

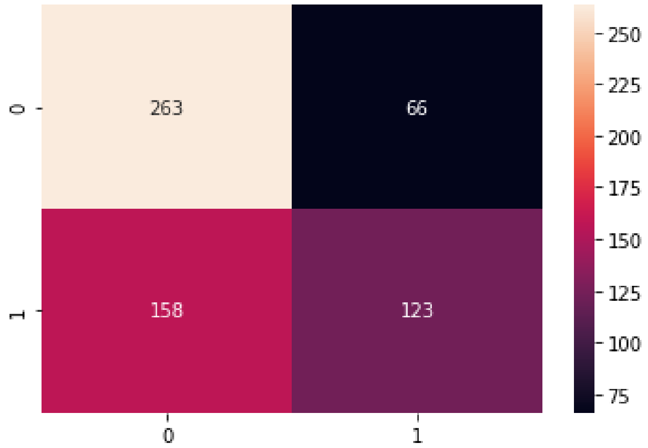


Fig 2.3.9

0.6

Accuracy Score 0.6213
F1 Score 0.446

Confusion Matrix

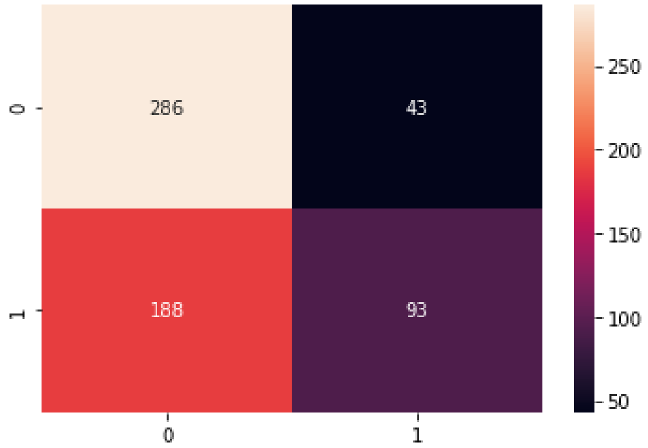


Fig 2.3.10

0.7

Accuracy Score 0.5869
F1 Score 0.2455

Confusion Matrix

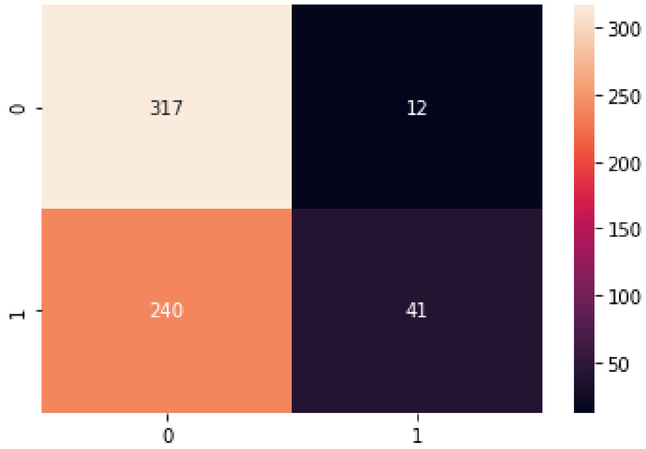


Fig 2.3.11

0.8

Accuracy Score 0.541
F1 Score 0.0071

Confusion Matrix

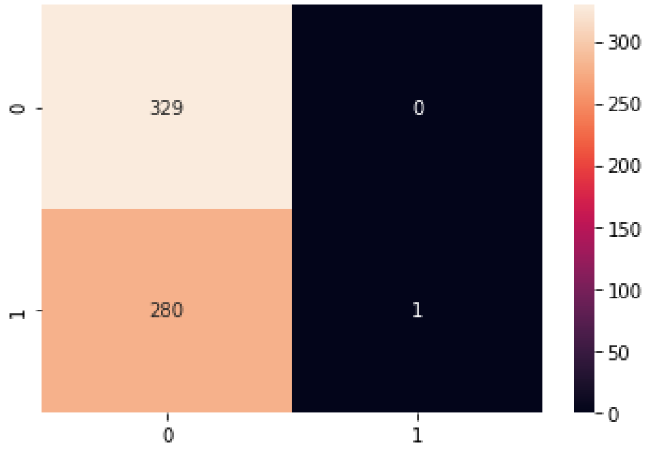


Fig 2.3.12

0.9

Accuracy Score 0.5393
F1 Score 0.0

Confusion Matrix

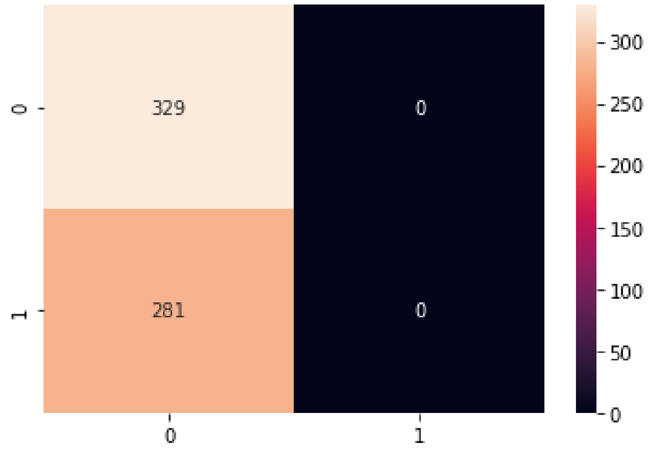
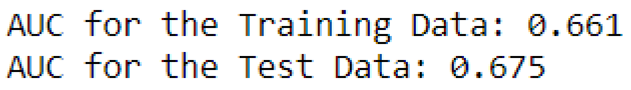


Fig 2.3.13

AUC and ROC Curve on Train Data and Test Data



```
AUC for the Training Data: 0.661
AUC for the Test Data: 0.675
```

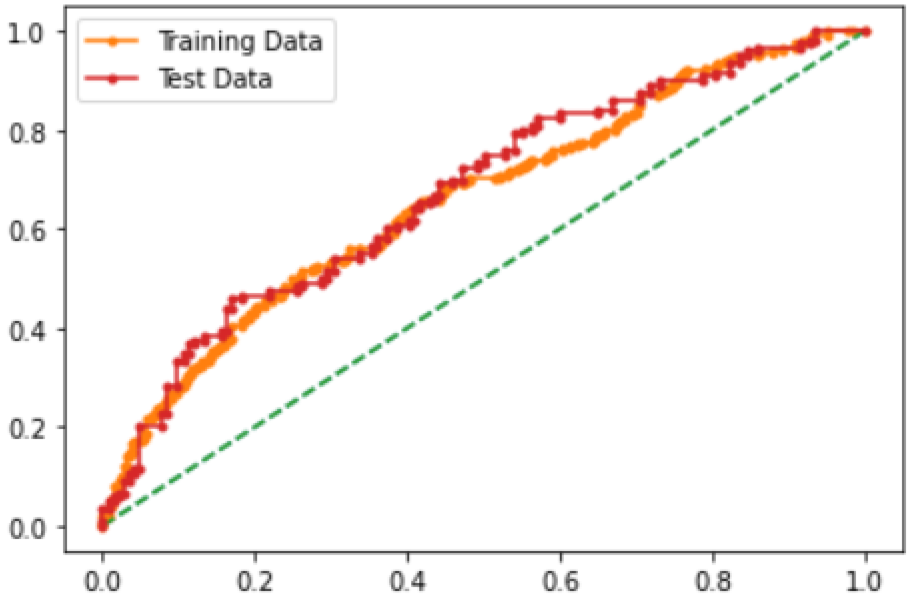
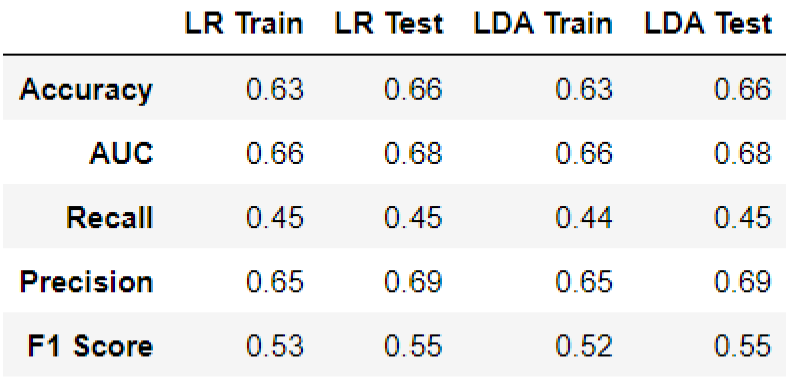


Fig 2.3.14

Comparing both Models



	LR Train	LR Test	LDA Train	LDA Test
Accuracy	0.63	0.66	0.63	0.66
AUC	0.66	0.68	0.66	0.68
Recall	0.45	0.45	0.44	0.45
Precision	0.65	0.69	0.65	0.69
F1 Score	0.53	0.55	0.52	0.55

Table 2.3.7

Comparing both these models, we find both results are same, but LDA works better when there is category target variable.

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

We had a business problem where we need predict whether an employee would opt for a holiday package or not, for this problem we had done predictions both logistic regression and linear discriminant analysis.

Since both are results are same, the EDA analysis clearly indicates certain criteria where we could find people aged above 50 are not interested much in holiday packages.

So, this is one of the we find aged people not opting for holiday packages. People ranging from the age 30 to 50 generally opt for holiday packages.

Employee age over 50 to 60 have seems to be not taking the holiday package, whereas in the age 30 to 50 and salary less than 50000 people have opted more for holiday package.

The important factors deciding the predictions are salary, age and educ.

Recommendations

1. To improve holiday packages over the age above 50 we can provide religious destination places.

2. For people earning more than 150000 we can provide vacation holiday packages.

3. For employee having more than number of older children we can provide packages in holiday vacation places.

THE END