PREDICTIVE MODELING PROJECT - 5 REPORT

Piyush Kumar Singh
PGP – DSBA Online
May-21 Batch

Date: 31/10/2021

Table of Contents

Problem 1

List of Figures

List of Tables

Problem 1: Linear Regression

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.

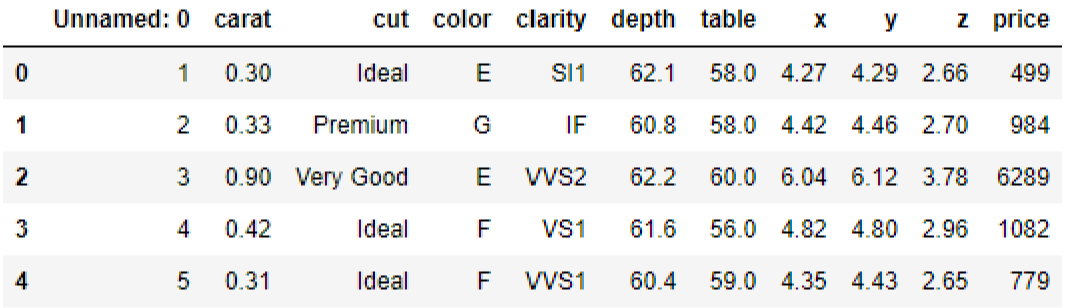
Data Dictionary

Variable Name	Description
Carat	Carat weight of the cubic zirconia.
Cut	Describe the cut quality of the cubic zirconia. Quality is increasing order Fair, Good, Very Good, Premium, Ideal.
Color	Colour of the cubic zirconia.With D being the worst and J the best.
Clarity	Clarity refers to the absence of the Inclusions and Blemishes. (In order from Worst to Best in terms of avg price) IF, VVS1, VVS2, VS1, VS2, Sl1, Sl2, l1
Depth	The Height of cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter.
Table	The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter.
Price	the Price of the cubic zirconia.
X	Length of the cubic zirconia in mm.
Y	Width of the cubic zirconia in mm.
Z	Height of the cubic zirconia in mm.

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

Exploratory Data Analysis

Dataset Sample

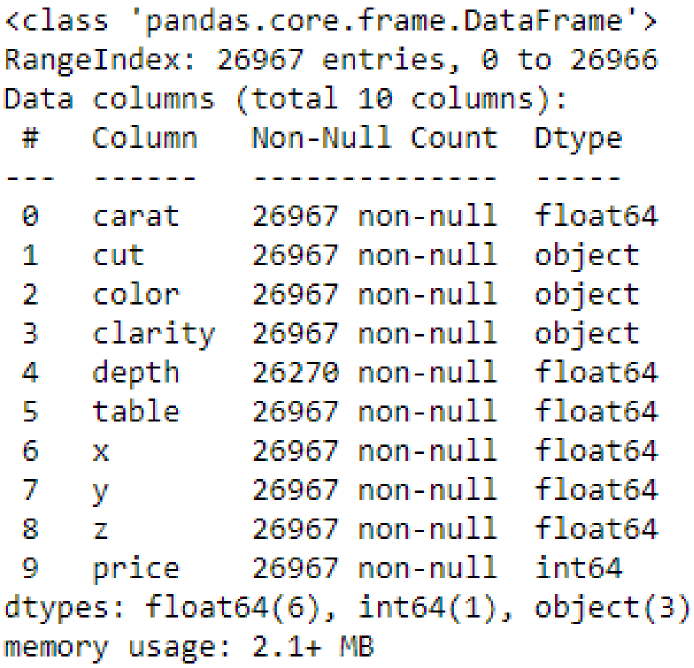


	Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
0	1	0.30	Ideal	E	SI1	62.1	58.0	4.27	4.29	2.66	499
1	2	0.33	Premium	G	IF	60.8	58.0	4.42	4.46	2.70	984
2	3	0.90	Very Good	E	VVS2	62.2	60.0	6.04	6.12	3.78	6289
3	4	0.42	Ideal	F	VS1	61.6	56.0	4.82	4.80	2.96	1082
4	5	0.31	Ideal	F	VVS1	60.4	59.0	4.35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

Table. 1

- I am dropping ‘Unnamed:0’ column here itself as it is just index numbers and will not contribute to anything in model analysis.

Dataset Info

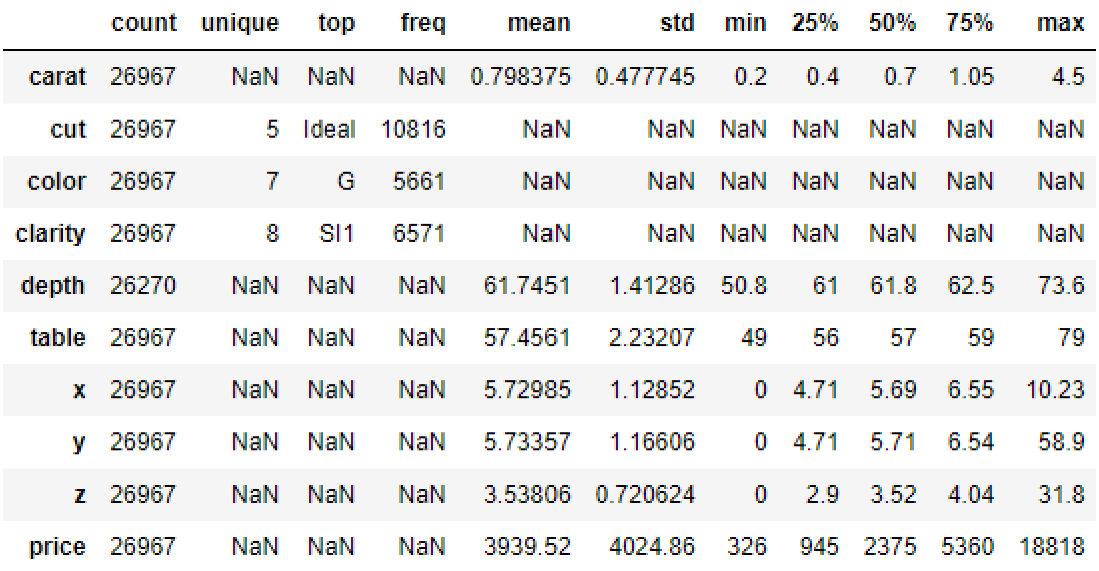


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26967 entries, 0 to 26966
Data columns (total 10 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   carat     26967 non-null  float64
 1   cut       26967 non-null  object 
 2   color     26967 non-null  object 
 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. 2

- The given dataset has now 10 columns out of following datatypes:
  - Float64 type - 6 column
  - Int64 type - 1 column
  - Object type - 3 column
- Dataset has 26967 rows and 10 column

Summary of Dataset

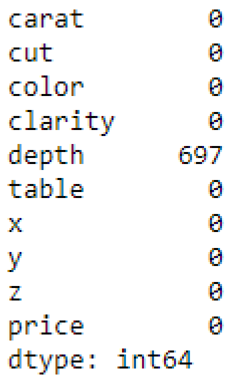


	count	unique	top	freq	mean	std	min	25%	50%	75%	max
carat	26967	NaN	NaN	NaN	0.798375	0.477745	0.2	0.4	0.7	1.05	4.5
cut	26967	5	Ideal	10816	NaN	NaN	NaN	NaN	NaN	NaN	NaN
color	26967	7	G	5661	NaN	NaN	NaN	NaN	NaN	NaN	NaN
clarity	26967	8	SI1	6571	NaN	NaN	NaN	NaN	NaN	NaN	NaN
depth	26270	NaN	NaN	NaN	61.7451	1.41286	50.8	61	61.8	62.5	73.6
table	26967	NaN	NaN	NaN	57.4561	2.23207	49	56	57	59	79
x	26967	NaN	NaN	NaN	5.72985	1.12852	0	4.71	5.69	6.55	10.23
y	26967	NaN	NaN	NaN	5.73357	1.16606	0	4.71	5.71	6.54	58.9
z	26967	NaN	NaN	NaN	3.53806	0.720624	0	2.9	3.52	4.04	31.8
price	26967	NaN	NaN	NaN	3939.52	4024.86	326	945	2375	5360	18818

Table. 3

- We can see that for there are 3 object type columns so they need to encoded to numerical type for analysis
- Also values of columns are in ones, tens, hundreds and thousands, so scaling need to performed to make the scale of each variable consistent.

Checking for missing values in dataset

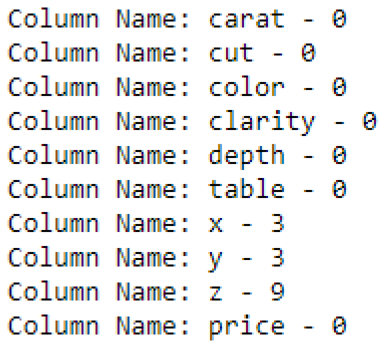


```
carat         0
cut           0
color         0
clarity       0
depth       697
table         0
x             0
y             0
z             0
price         0
dtype: int64
```

Table. 4

- There are 697 rows with 'NaN' entry in depth column.
- As per data dictionary of dataset we can say that value of this column can't be 0, so we are dropping all these rows.

Checking for entries' with 0 in dataset



```
Column Name: carat - 0
Column Name: cut - 0
Column Name: color - 0
Column Name: clarity - 0
Column Name: depth - 0
Column Name: table - 0
Column Name: x - 3
Column Name: y - 3
Column Name: z - 9
Column Name: price - 0
```

Table. 5

- Here we are checking the dataset for entries of value – 0. As per data dictionaries of the dataset values of column x, y, z can't be 0 as it make no sense.
- So we are imputing these entries with median values of respective columns.

Checking for duplicates in dataset

- There are 34 duplicate entries in the dataset, I am dropping all these duplicate records.

Univariate Analysis

We perform the univariate analysis on the data set and display distplot to check distribution for each continuous type column and use boxplot to check for outliers if any.

Continuous Type Columns

carat

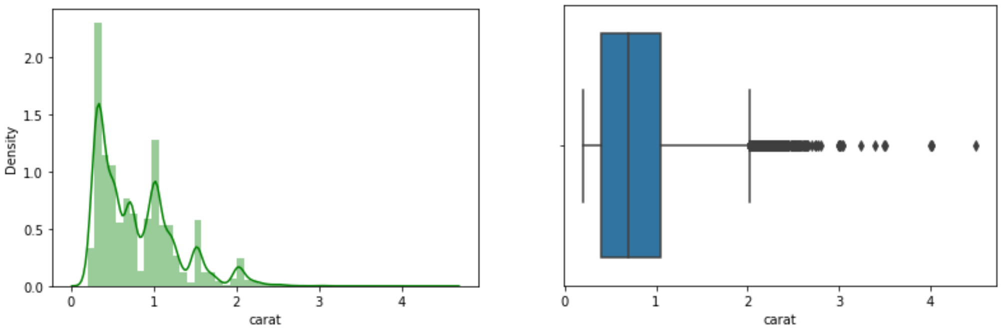


Fig.1 - Distplot and boxplot of carat

- The mean number of carat is 0.79 whereas the SD is 0.47
- The maximum value is 4.5 and min value is 0.2
- The distribution is highly skewed towards right.
- Outliers are present.

depth

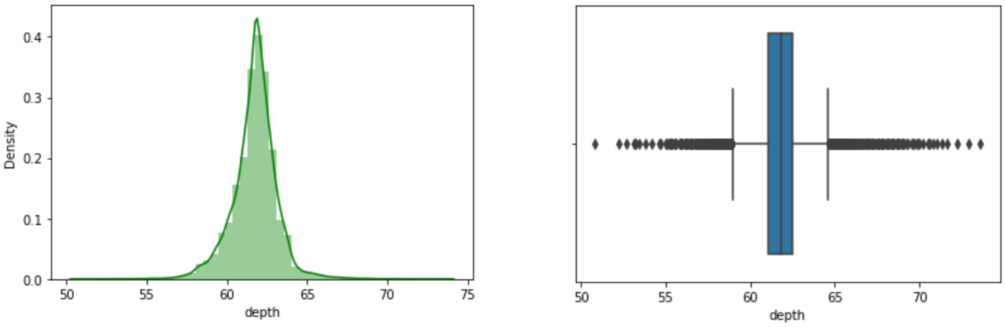


Fig.2 - Distplot and boxplot of depth

- The mean number of depth is 61.47 whereas the SD is 1.41
- The maximum value is 73.6 and min value is 50.8
- The distribution is normally distributed.
- Outliers are present.

Table

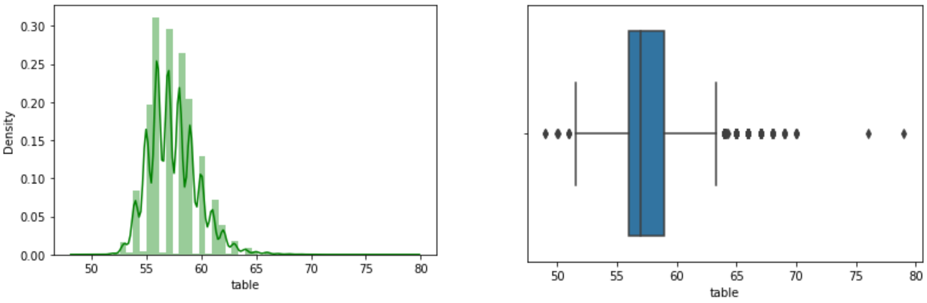


Fig.3 - Distplot and boxplot of table

- The mean number of table is 57.45 whereas the SD is 2.23
- The maximum value is 79 and min value is 49
- The distribution is somewhat normal
- Outliers are present.

x

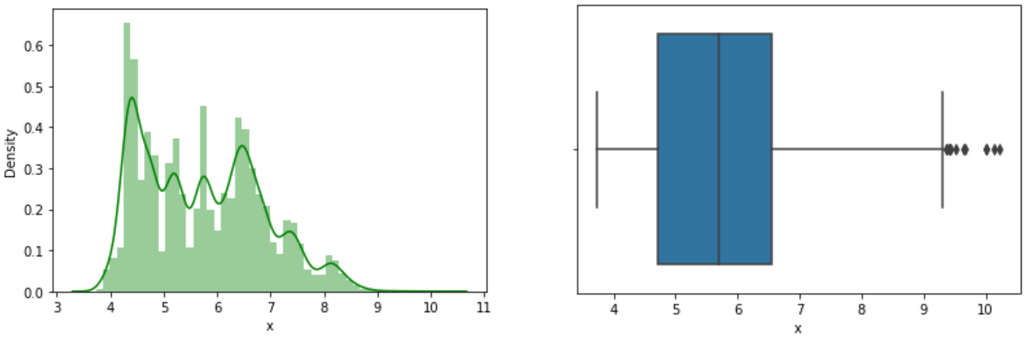


Fig.4 - Distplot and boxplot of x

- The mean number of x is 5.72 whereas the SD is 1.12.
- The maximum value is 10.23 and min value is 0
- The distribution is skewed towards right.
- Outliers are present.

y

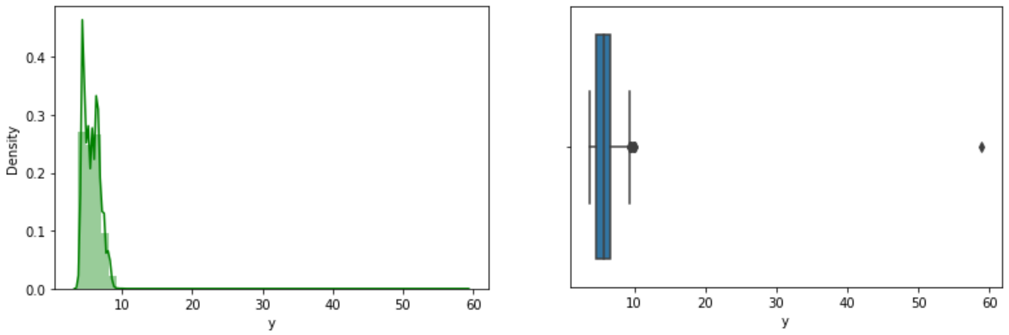


Fig.5 - Distplot and boxplot of y

- The mean number of y is 5.73 whereas the SD is 1.16
- The maximum value is 58.9 and min value is 0
- The distribution is somewhat normally distributed
- Outliers are present.

z

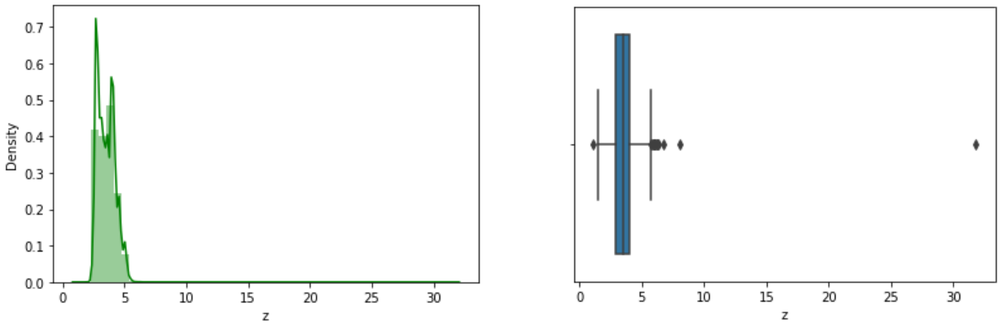


Fig.6 - Distplot and boxplot of z

- The mean number of z is 3.53 whereas the SD is 0.72
- The maximum value is 31.8 and min value is 0
- The distribution is somewhat normally distributed
- Outliers are present.

Price – Our Target Variable

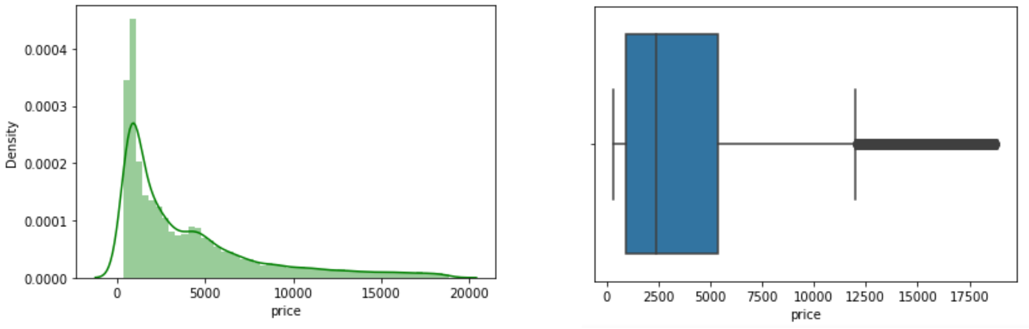
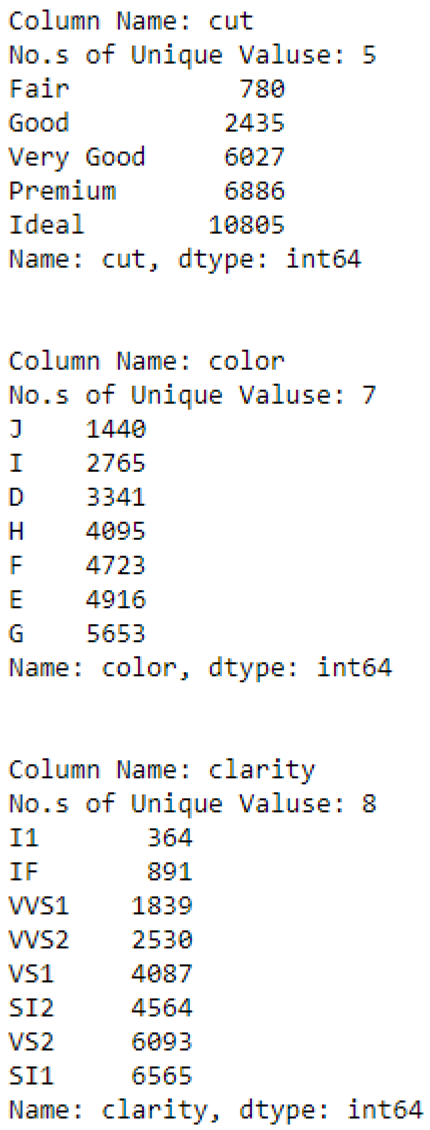


Fig.7 - Distplot and boxplot of price

- The mean number of price is 3939.52 whereas the SD is 4024.86.
- The maximum value is 18818 and min value is 326
- The distribution is highly skewed towards right.
- Outliers are present.
- **Price is our target variable for analysis.**

Categorical Type Columns



```
Column Name: cut
No.s of Unique Valuse: 5
Fair           780
Good          2435
Very Good     6027
Premium       6886
Ideal        10805
Name: cut, dtype: int64


Column Name: color
No.s of Unique Valuse: 7
J    1440
I    2765
D    3341
H    4095
F    4723
E    4916
G    5653
Name: color, dtype: int64


Column Name: clarity
No.s of Unique Valuse: 8
I1       364
IF       891
VVS1    1839
VVS2    2530
VS1     4087
SI2     4564
VS2     6093
SI1     6565
Name: clarity, dtype: int64
```

Table. 6

- The above table give details of all the types of category for the three categorical columns along with their value counts(Unit sold)
- Ideal cut diamonds are sold most out of all the cut types, and Fair is sold least.
- Diamond of Color G is sold most and J is sold least.
- Diamond of clarity SI1 is sold most and I1 is sold least

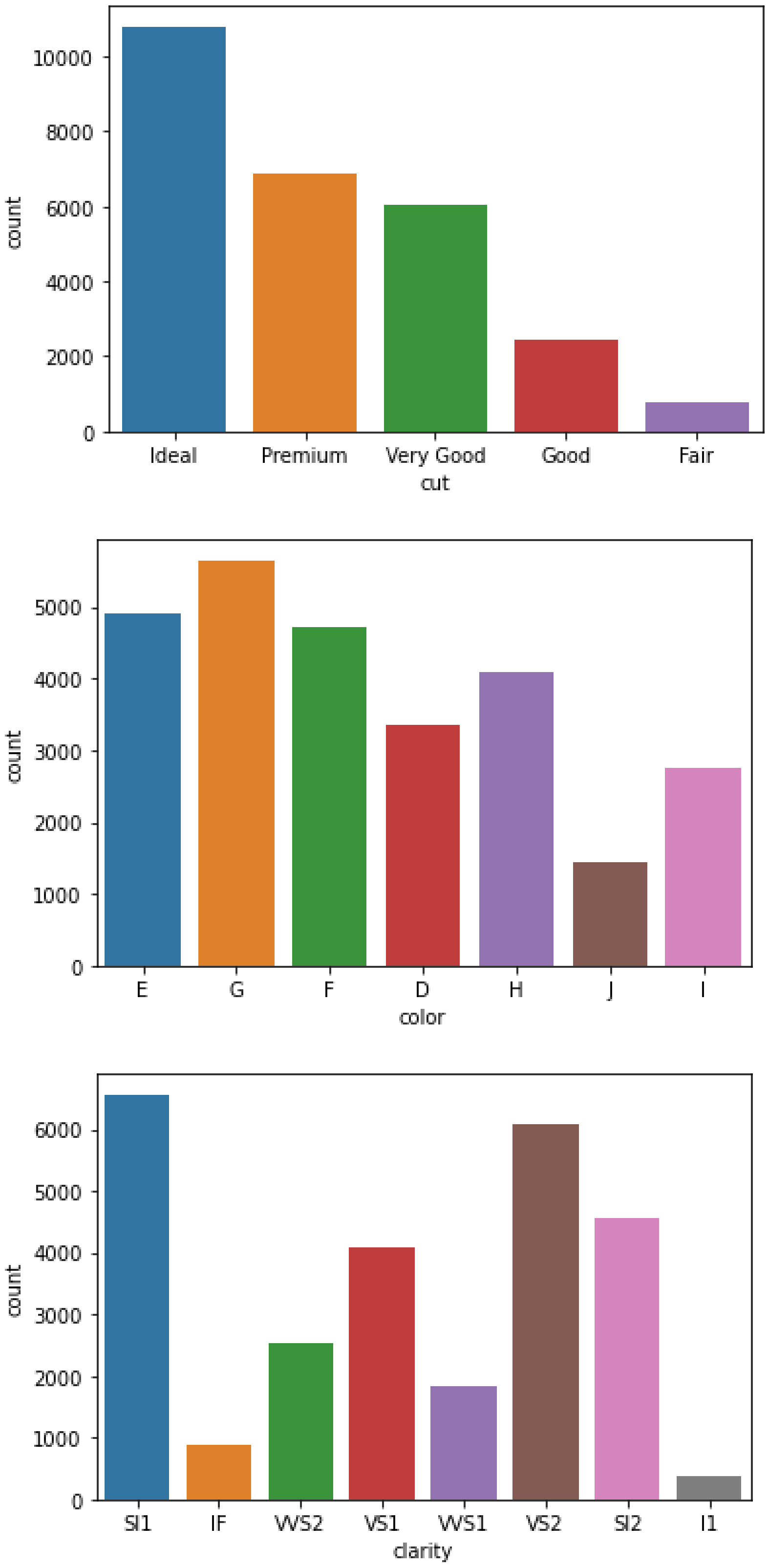


Fig. 8

- Graphical representation of the categorical columns using count plot.

Bi/Multivariate Analysis

Bi variant analysis of our target variable by comparing it with cut, clarity and color columns.

Cut vs price:

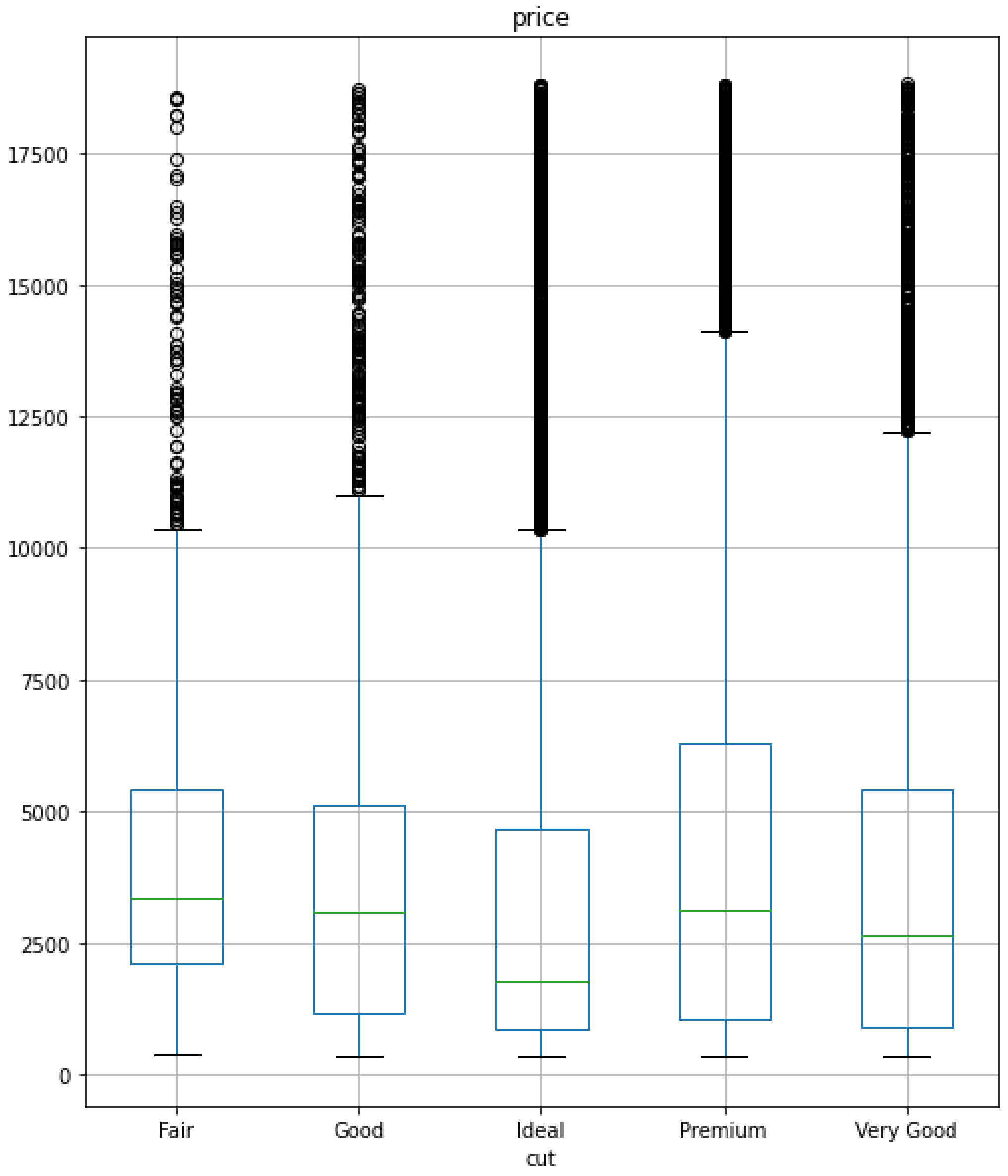


Fig. 9

- Analysing boxplot of cut with price and checking the median line , we can say price of ‘Premium’ , ‘Fair’, ‘Good’ cut diamonds is highest and ‘Ideal’ is lowest.

Color vs price:

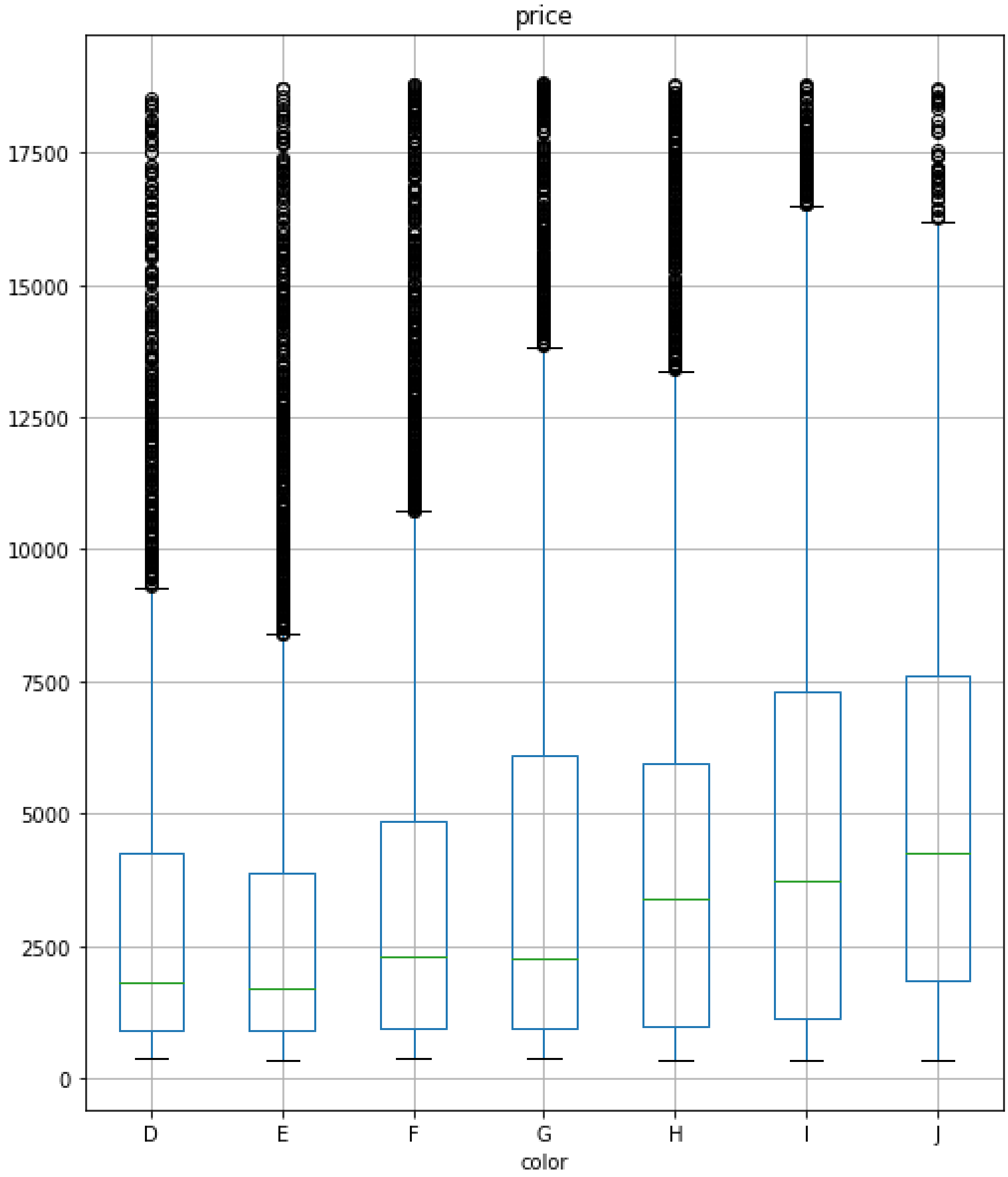


Fig. 10

- Analysing boxplot of color with price and checking the median line, we can say price of ‘J’ color diamonds is highest and ‘E’ is lowest.

Clarity vs price:

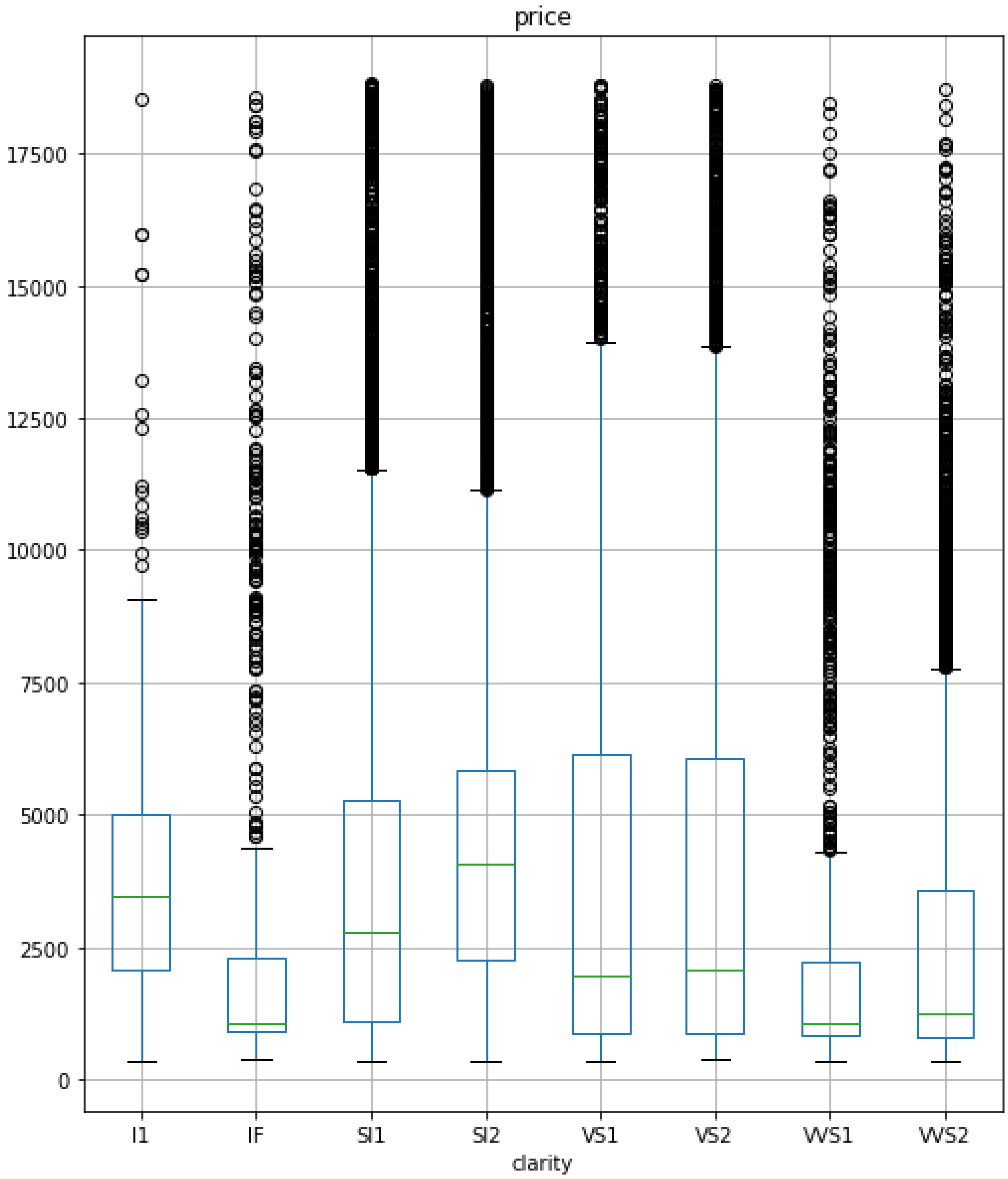


Fig. 11

- Analysing boxplot of clarity with price and checking the median line, we can say price of ‘SI2’ clarity diamonds is highest and ‘IF’ and ‘VVS1’ is lowest.

For Multivariate Analysis we use pair plot and Heatmap:

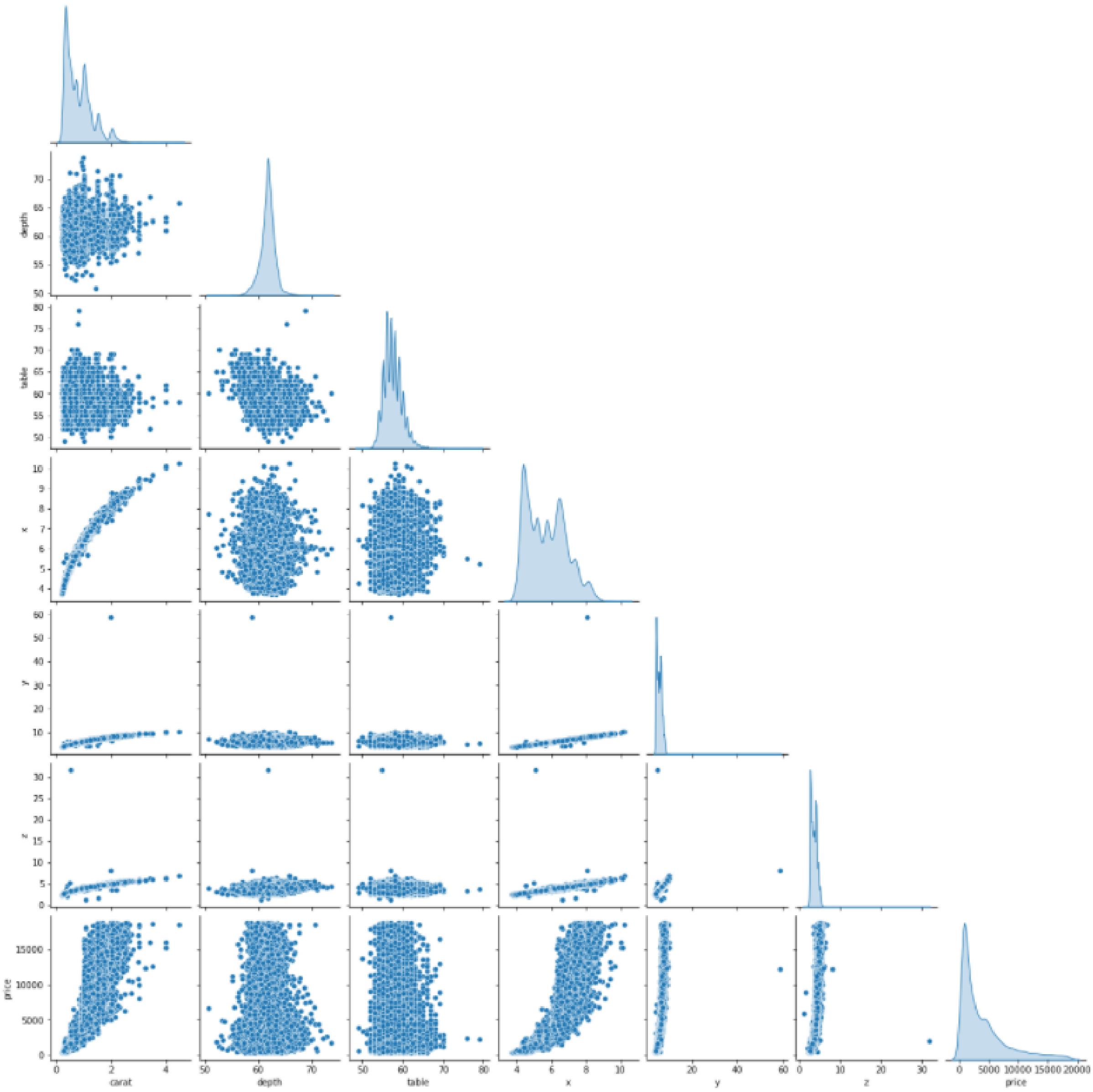


Fig. 12

- Graphs showing positive linear relationship
  - x and carat
  - y and carat
  - z and carat
  - carat and price
  - x and y
  - x and z
  - price and x

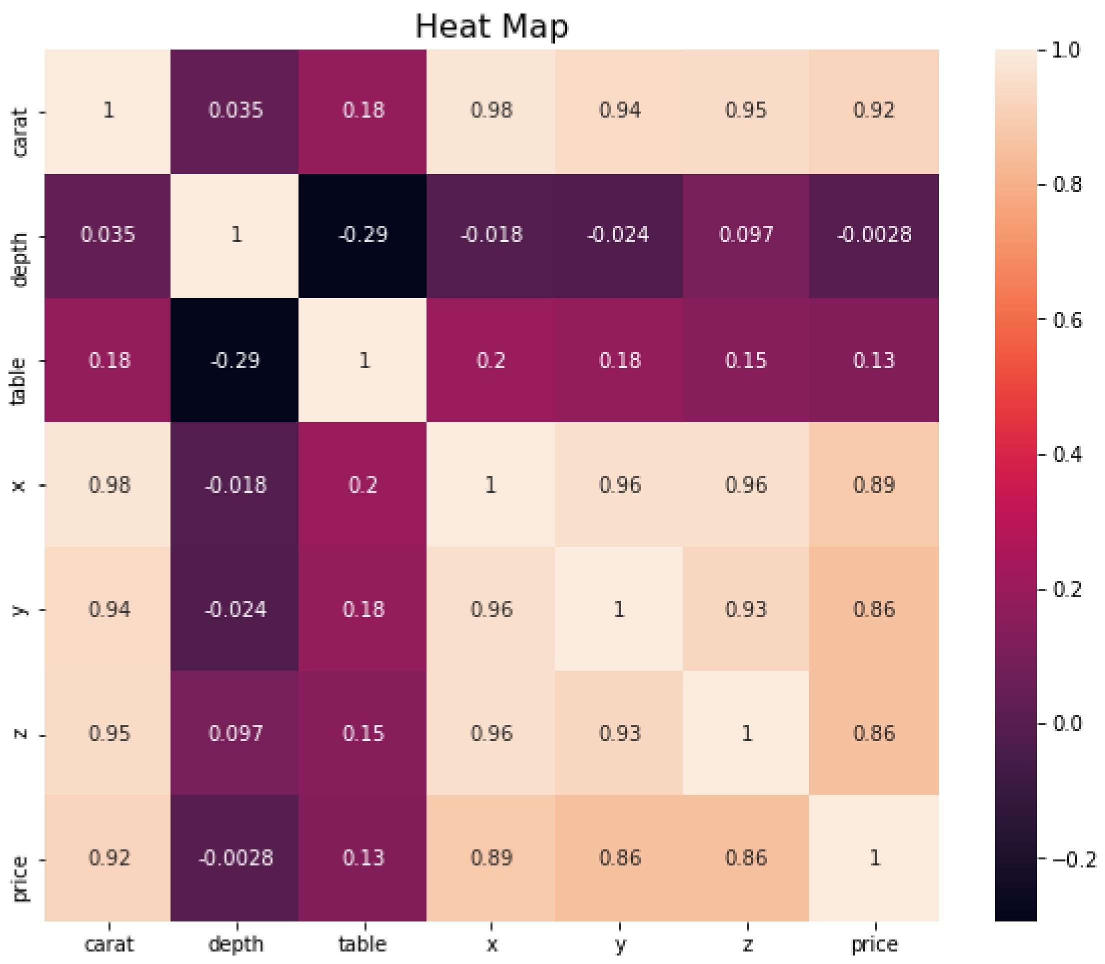


Fig. 13

- Degree of correlation between the columns is represented by the above heatmap, 1 being max value of correlation and 0 below no correlation between them.

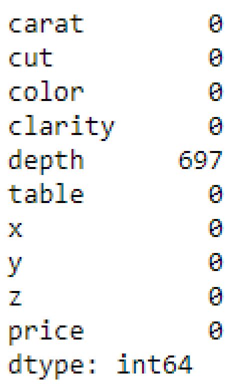
  <u>Pair with high correlation are as follows</u>:
- x and carat
- y and carat
- z and carat
- price and carat
- depth and z
- y and x
- z and x
- price and x
- z and y
- price and y
- price and z

- Presence of so many columns that are highly correlated with each other is not good for analysis, dataset is suffering from presence of multicollinearity.

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

As described in EDA part also we have checked the data set missing value, values equal to zero and duplicate values:

Checking for missing values in dataset

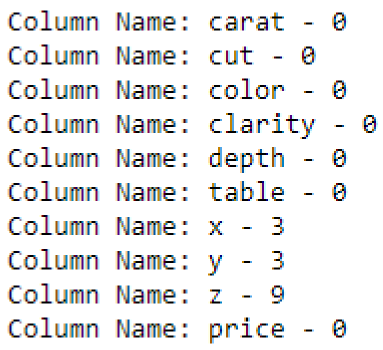


```
carat        0
cut          0
color        0
clarity      0
depth      697
table        0
x            0
y            0
z            0
price        0
dtype: int64
```

Table. 4

- There are 697 rows with 'NaN' entry in depth column.
- We are imputing the missing values with the median of the 'depth' column here, as depth of diamond is an important factor in our analysis.

Checking for entries' with 0 in dataset



```
Column Name: carat - 0
Column Name: cut - 0
Column Name: color - 0
Column Name: clarity - 0
Column Name: depth - 0
Column Name: table - 0
Column Name: x - 3
Column Name: y - 3
Column Name: z - 9
Column Name: price - 0
```

Table. 5

- Here we are checking the dataset for entries of value – 0. As per data dictionaries of the dataset values of column x, y, z can't be 0 as it make no sense.
- So we are imputing these entries with median values of their respective columns.

- Of all the categorical variable in the dataset I don't think there is any possibility of combining the sub levels of the variables as all the sub level are unique identifier and it's not possible to merge any of these levels.

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

Before encoding the object type columns we are performing outlier's treatment for the numeric columns first. As per one of the assumption of regression model building outlier's should not be present in the dataset.

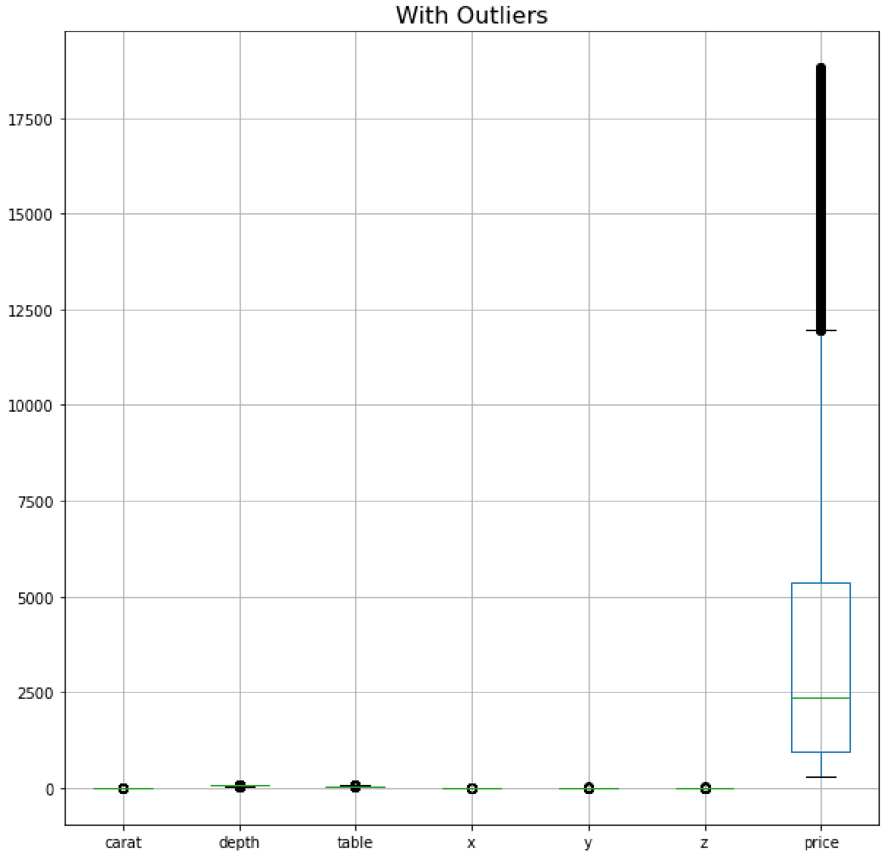


Fig. 14

- Above figure shows continuous columns with outlier's.

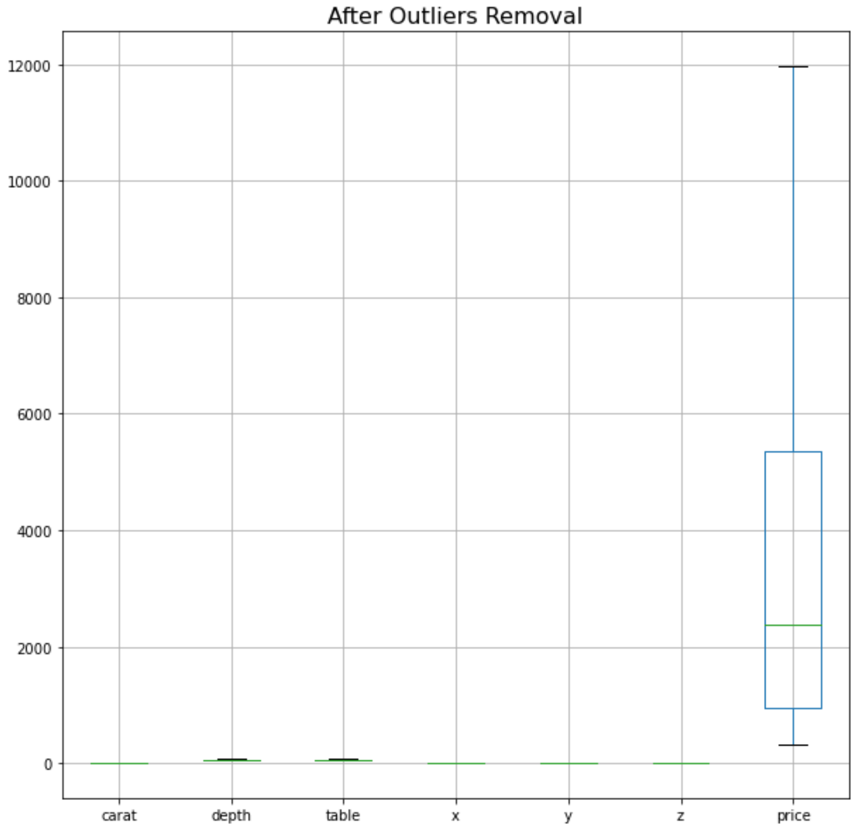
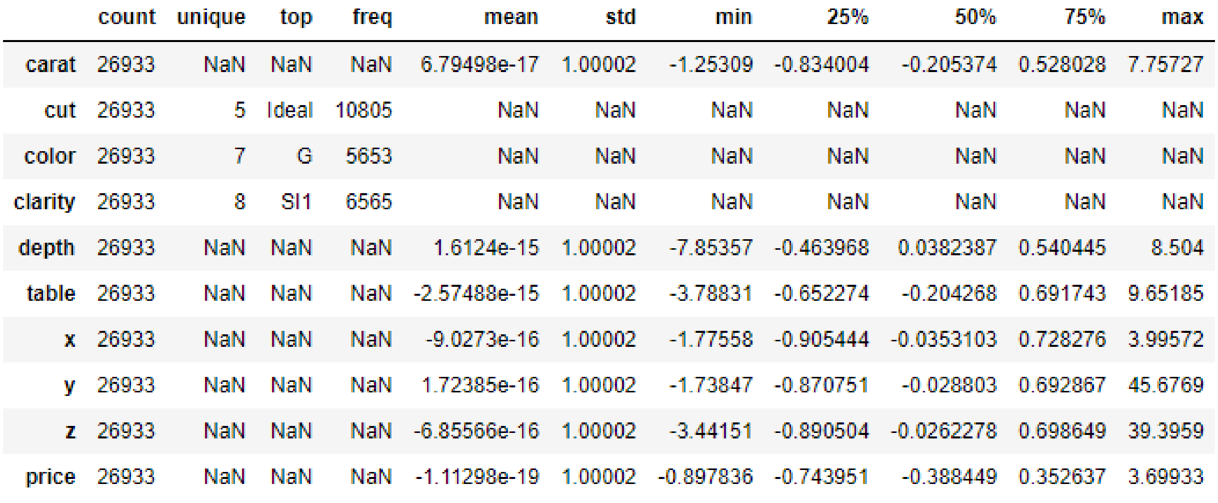


Fig. 15

- Figure after treatment of all the outlier's.

Now we are scaling the dataset, because if we apply linear regression to the dataset without scaling the data the coefficient's that are coming from the model are not stable and of very high values.

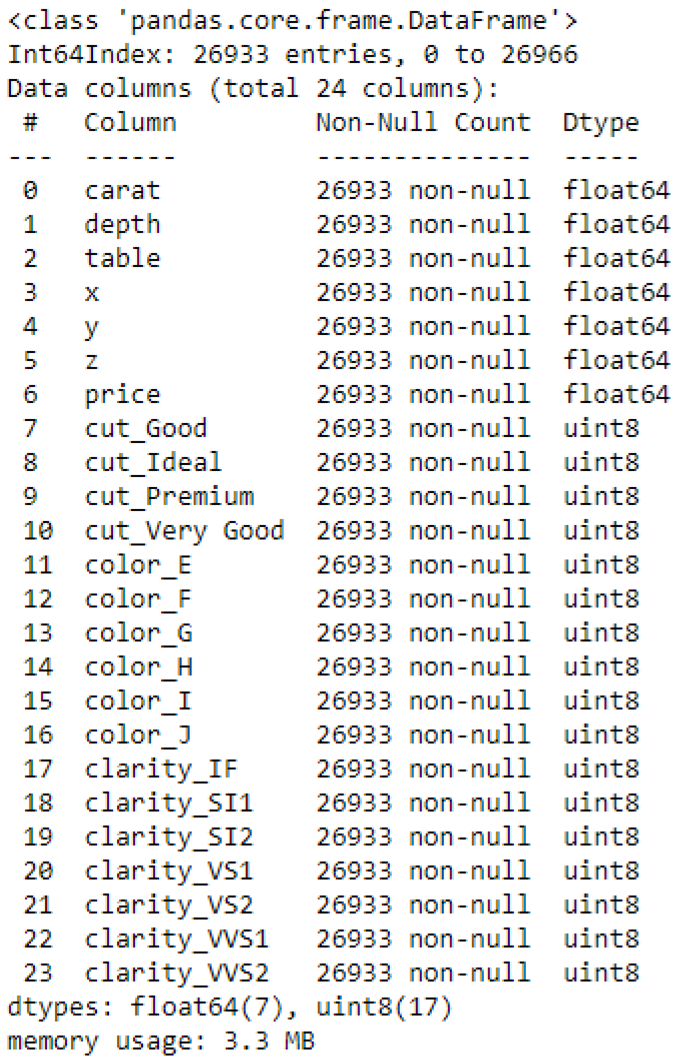
I have performed scaling using Standard Scaler, and below is the summary of dataset after scaling:



	count	unique	top	freq	mean	std	min	25%	50%	75%	max
carat	26933	NaN	NaN	NaN	6.79498e-17	1.00002	-1.25309	-0.834004	-0.205374	0.528028	7.75727
cut	26933	5	Ideal	10805	NaN	NaN	NaN	NaN	NaN	NaN	NaN
color	26933	7	G	5653	NaN	NaN	NaN	NaN	NaN	NaN	NaN
clarity	26933	8	SI1	6565	NaN	NaN	NaN	NaN	NaN	NaN	NaN
depth	26933	NaN	NaN	NaN	1.6124e-15	1.00002	-7.85357	-0.463968	0.0382387	0.540445	8.504
table	26933	NaN	NaN	NaN	-2.57488e-15	1.00002	-3.78831	-0.652274	-0.204268	0.691743	9.65185
x	26933	NaN	NaN	NaN	-9.0273e-16	1.00002	-1.77558	-0.905444	-0.0353103	0.728276	3.99572
y	26933	NaN	NaN	NaN	1.72385e-16	1.00002	-1.73847	-0.870751	-0.028803	0.692867	45.6769
z	26933	NaN	NaN	NaN	-6.85566e-16	1.00002	-3.44151	-0.890504	-0.0262278	0.698649	39.3959
price	26933	NaN	NaN	NaN	-1.11298e-19	1.00002	-0.897836	-0.743951	-0.388449	0.352637	3.69933

Table. 7

Now label Encoding is performed on all the three ‘object’ type columns to make them numeric.



```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 26933 entries, 0 to 26966
Data columns (total 24 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   carat          26933 non-null  float64
 1   depth          26933 non-null  float64
 2   table          26933 non-null  float64
 3   x              26933 non-null  float64
 4   y              26933 non-null  float64
 5   z              26933 non-null  float64
 6   price          26933 non-null  float64
 7   cut_Good       26933 non-null  uint8  
 8   cut_Ideal      26933 non-null  uint8  
 9   cut_Premium    26933 non-null  uint8  
 10  cut_Very Good  26933 non-null  uint8  
 11  color_E        26933 non-null  uint8  
 12  color_F        26933 non-null  uint8  
 13  color_G        26933 non-null  uint8  
 14  color_H        26933 non-null  uint8  
 15  color_I        26933 non-null  uint8  
 16  color_J        26933 non-null  uint8  
 17  clarity_IF     26933 non-null  uint8  
 18  clarity_SI1    26933 non-null  uint8  
 19  clarity_SI2    26933 non-null  uint8  
 20  clarity_VS1    26933 non-null  uint8  
 21  clarity_VS2    26933 non-null  uint8  
 22  clarity_VVS1   26933 non-null  uint8  
 23  clarity_VVS2   26933 non-null  uint8  
dtypes: float64(7), uint8(17)
memory usage: 3.3 MB
```

Table. 8

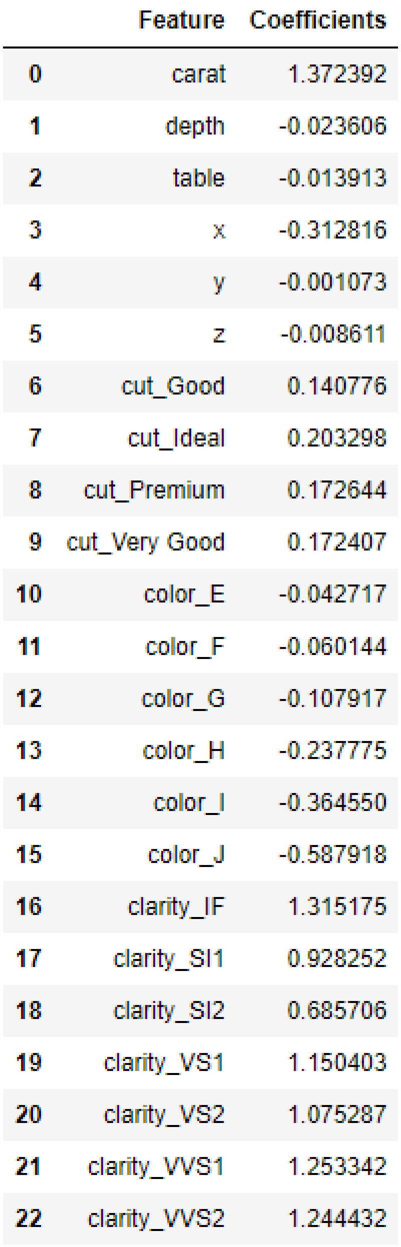
- Dataset info after label encoding.
- Number of columns has now been increased from 10 to 24 now.

Model Build using sklearn library:

We now split the scaled data into train and test set with 70:30 ration.

We have taken ‘price’ as our target variable.

We apply the Linear Regression model and explore the coefficients of each variables in the training dataset.



	Feature	Coefficients
0	carat	1.372392
1	depth	-0.023606
2	table	-0.013913
3	x	-0.312816
4	y	-0.001073
5	z	-0.008611
6	cut_Good	0.140776
7	cut_Ideal	0.203298
8	cut_Premium	0.172644
9	cut_Very Good	0.172407
10	color_E	-0.042717
11	color_F	-0.060144
12	color_G	-0.107917
13	color_H	-0.237775
14	color_I	-0.364550
15	color_J	-0.587918
16	clarity_IF	1.315175
17	clarity_SI1	0.928252
18	clarity_SI2	0.685706
19	clarity_VS1	1.150403
20	clarity_VS2	1.075287
21	clarity_VVS1	1.253342
22	clarity_VVS2	1.244432

Table. 9

- Because of the scaling that we performed earlier the coefficient's are much more stable and reliable. From the coefficient's we can identify all the variables that have positive relationship with our target variable all the variable's that are not contributing much toward ours analysis.
  - Variable showing positive values show positive relationship.
  - Variable showing negative values show negative relationship.
  - Variable showing values close to zero can be considered insignificant in our analysis.

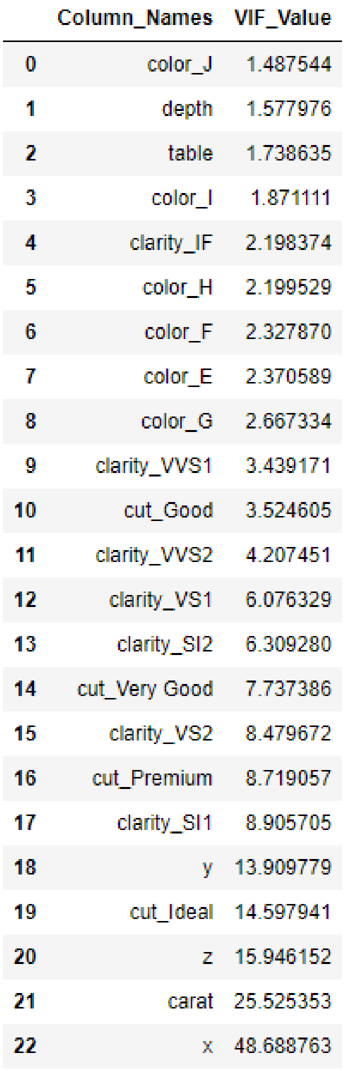
- Intercept of our Linear Regression Model is **-1.0371**

- R2 score of training dataset is **0.92102**
- R2 score of testing dataset is **0.92249**

  Since the coefficient of determinant values of both train and test set are very close, it means our model is performing well in both train dataset as well as test dataset. We can later see adjusted R2 value from stats model with is much more reliable than R2 score.

- RMSE value of training dataset is **0.2816**
- RMSE value of testing dataset is **0.2770**

To test Multicollinearity with the dataset we use Variation Inflation Factor (VIF):



	Column_Names	VIF_Value
0	color_J	1.487544
1	depth	1.577976
2	table	1.738635
3	color_I	1.871111
4	clarity_IF	2.198374
5	color_H	2.199529
6	color_F	2.327870
7	color_E	2.370589
8	color_G	2.667334
9	clarity_VVS1	3.439171
10	cut_Good	3.524605
11	clarity_VVS2	4.207451
12	clarity_VS1	6.076329
13	clarity_SI2	6.309280
14	cut_Very Good	7.737386
15	clarity_VS2	8.479672
16	cut_Premium	8.719057
17	clarity_SI1	8.905705
18	y	13.909779
19	cut_Ideal	14.597941
20	z	15.946152
21	carat	25.525353
22	x	48.688763

Table. 10

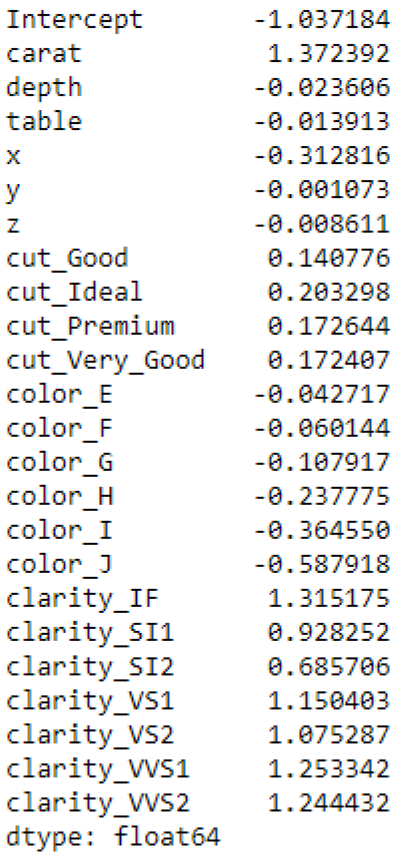
VIF start from 1 and has no upper limit.

- A value indicates that there is no correlation between the independent variable.
- VIF's between 1 and 5 suggest that there is a moderate correlation, but it is not that severe enough to warrant corrective measures.
- VIF's greater that 5 represent critical levels of multicollinearity where the coefficient's are poorly estimated and p-value are questionable.

The above VIF value table is arranged in ascending order of VIF values and all columns from index 12 to 22 have critical values and need corrective measures.

Model Build using Statsmodel:

- Now we build the linear regression model using stats method using OLS formula.
- The coefficient's we get from stats model are as follow:

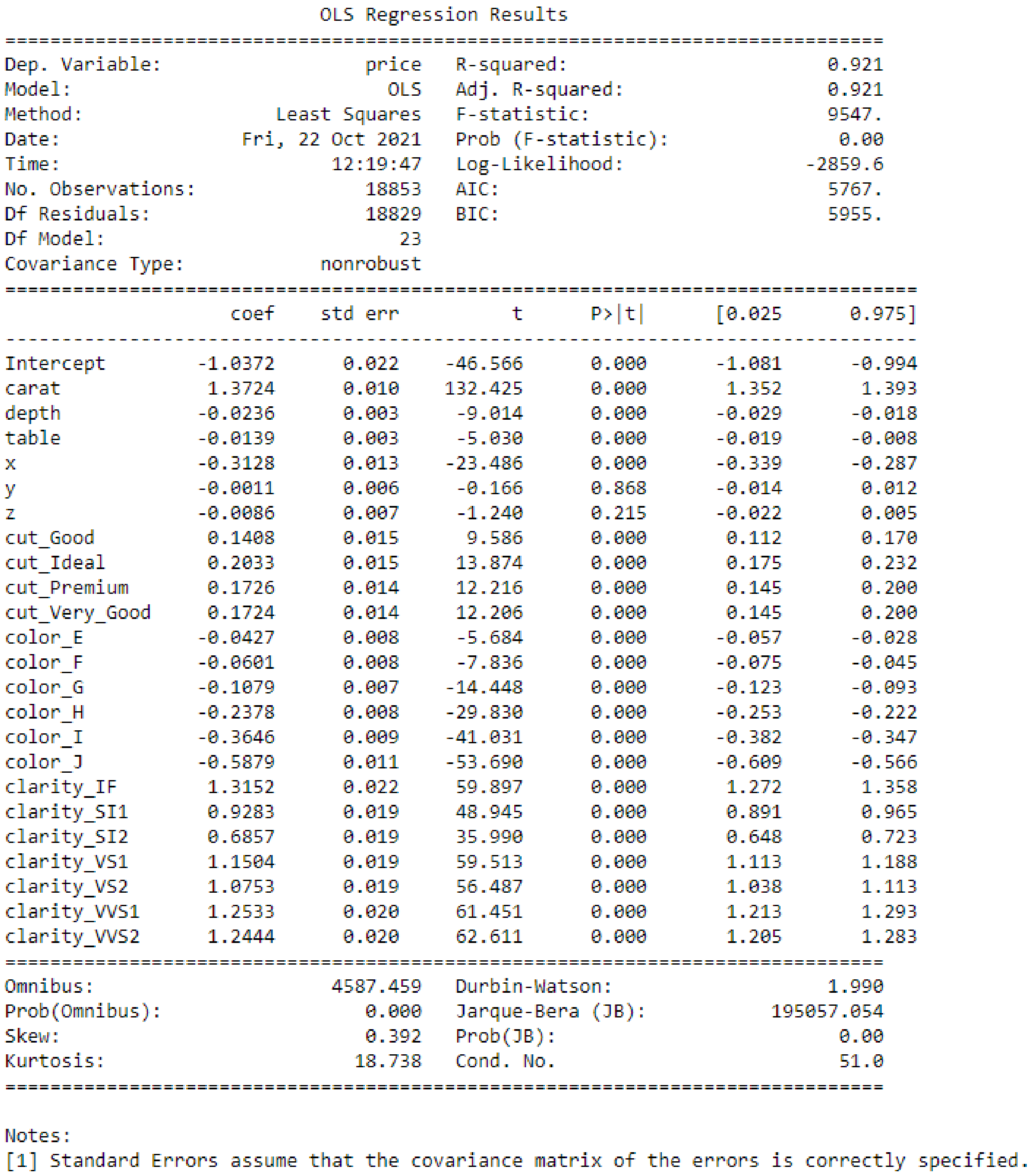


```
Intercept       -1.037184
carat            1.372392
depth           -0.023606
table           -0.013913
x               -0.312816
y               -0.001073
z               -0.008611
cut_Good         0.140776
cut_Ideal        0.203298
cut_Premium      0.172644
cut_Very_Good    0.172407
color_E         -0.042717
color_F         -0.060144
color_G         -0.107917
color_H         -0.237775
color_I         -0.364550
color_J         -0.587918
clarity_IF       1.315175
clarity_SI1      0.928252
clarity_SI2      0.685706
clarity_VS1      1.150403
clarity_VS2      1.075287
clarity_VVS1     1.253342
clarity_VVS2     1.244432
dtype: float64
```

Table. 11

- We can clearly see the coefficient's we get from stats model is exactly same as we get from sklearn regression model.

- Stats model summary for the model is as below:



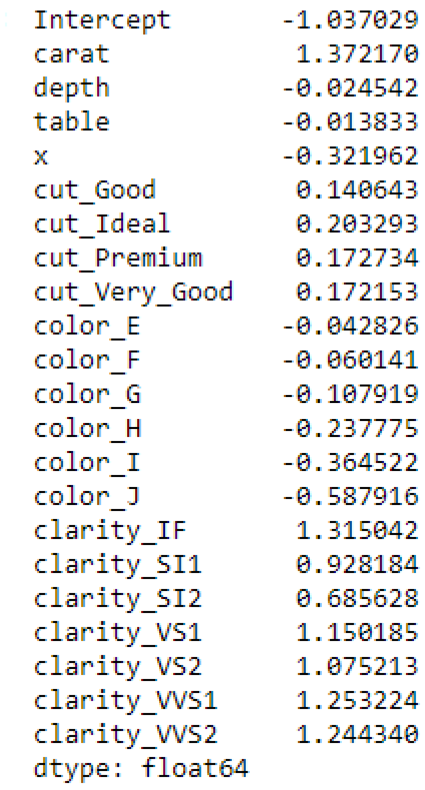
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.921
Model:                            OLS   Adj. R-squared:                  0.921
Method:                 Least Squares   F-statistic:                     9547.
Date:                Fri, 22 Oct 2021   Prob (F-statistic):               0.00
Time:                        12:19:47   Log-Likelihood:                -2859.6
No. Observations:               18853   AIC:                             5767.
Df Residuals:                   18829   BIC:                             5955.
Df Model:                          23
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
Intercept        -1.0372      0.022    -46.566      0.000      -1.081      -0.994
carat             1.3724      0.010    132.425      0.000       1.352       1.393
depth            -0.0236      0.003     -9.014      0.000      -0.029      -0.018
table            -0.0139      0.003     -5.030      0.000      -0.019      -0.008
x                -0.3128      0.013    -23.486      0.000      -0.339      -0.287
y                -0.0011      0.006     -0.166      0.868      -0.014       0.012
z                -0.0086      0.007     -1.240      0.215      -0.022       0.005
cut_Good          0.1408      0.015      9.586      0.000       0.112       0.170
cut_Ideal         0.2033      0.015     13.874      0.000       0.175       0.232
cut_Premium       0.1726      0.014     12.216      0.000       0.145       0.200
cut_Very_Good     0.1724      0.014     12.206      0.000       0.145       0.200
color_E          -0.0427      0.008     -5.684      0.000      -0.057      -0.028
color_F          -0.0601      0.008     -7.836      0.000      -0.075      -0.045
color_G          -0.1079      0.007    -14.448      0.000      -0.123      -0.093
color_H          -0.2378      0.008    -29.830      0.000      -0.253      -0.222
color_I          -0.3646      0.009    -41.031      0.000      -0.382      -0.347
color_J          -0.5879      0.011    -53.690      0.000      -0.609      -0.566
clarity_IF        1.3152      0.022     59.897      0.000       1.272       1.358
clarity_SI1       0.9283      0.019     48.945      0.000       0.891       0.965
clarity_SI2       0.6857      0.019     35.990      0.000       0.648       0.723
clarity_VS1       1.1504      0.019     59.513      0.000       1.113       1.188
clarity_VS2       1.0753      0.019     56.487      0.000       1.038       1.113
clarity_VVS1      1.2533      0.020     61.451      0.000       1.213       1.293
clarity_VVS2      1.2444      0.020     62.611      0.000       1.205       1.283
==============================================================================
Omnibus:                     4587.459   Durbin-Watson:                   1.990
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           195057.054
Skew:                           0.392   Prob(JB):                         0.00
Kurtosis:                      18.738   Cond. No.                         51.0
==============================================================================

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

Table. 12

- Value of Adjusted R-squared is same as of R-squared – 0.921
- Variance type is nonrobust – means model is not very accurate and has presence of multicollinearity.
- By checking the P-value of all the columns we can say that ‘y’ and ‘z’ are not significant in model analysis as p-value is greater than 0.05. We can drop these columns to get much better results from the model.

We again perform stats model after dropping y and z column:
2nd Run of Stats model -

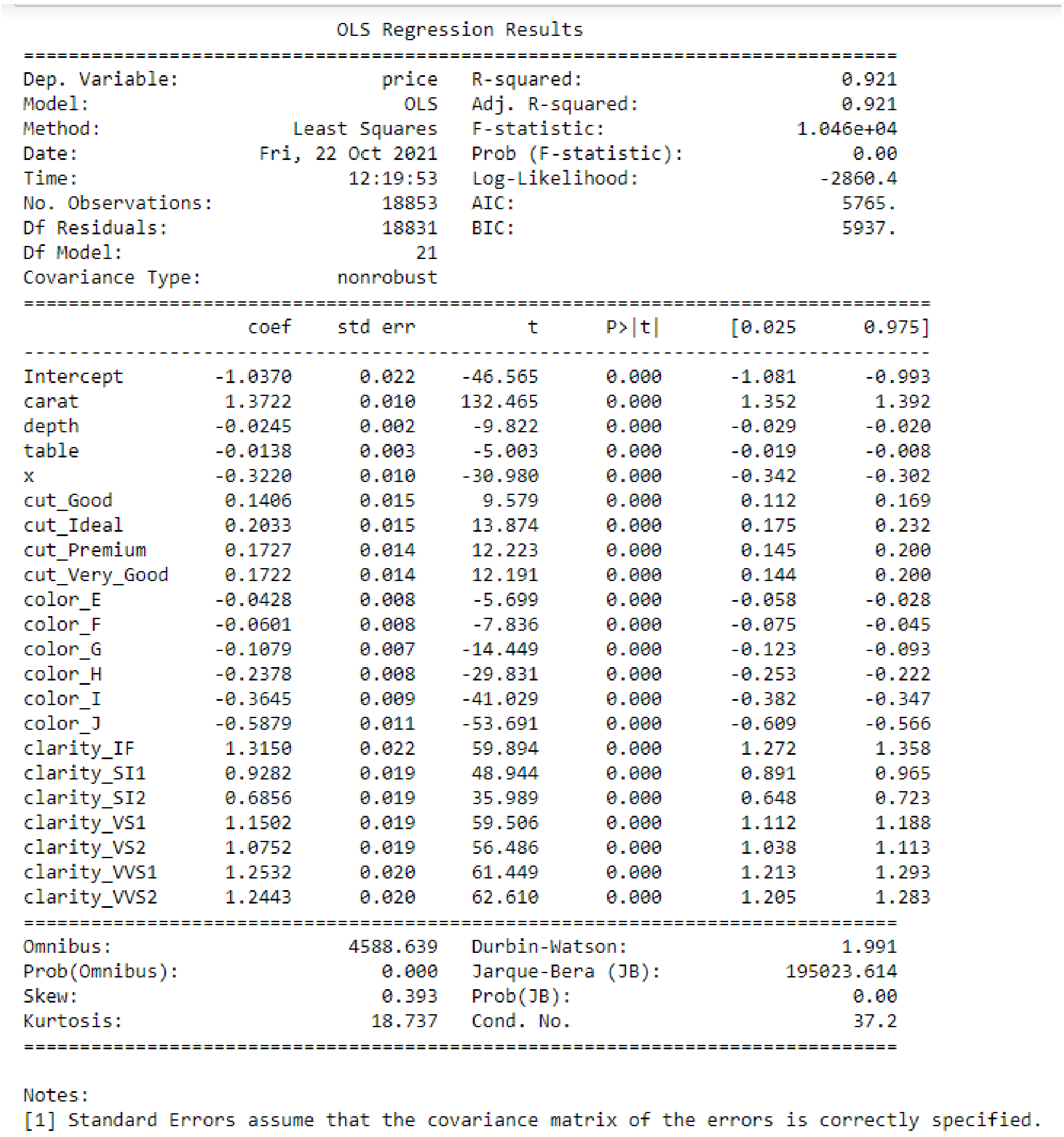
- New coefficients' :



```
Intercept        -1.037029
carat             1.372170
depth            -0.024542
table            -0.013833
x                -0.321962
cut_Good          0.140643
cut_Ideal         0.203293
cut_Premium       0.172734
cut_Very_Good     0.172153
color_E          -0.042826
color_F          -0.060141
color_G          -0.107919
color_H          -0.237775
color_I          -0.364522
color_J          -0.587916
clarity_IF        1.315042
clarity_SI1       0.928184
clarity_SI2       0.685628
clarity_VS1       1.150185
clarity_VS2       1.075213
clarity_VVS1      1.253224
clarity_VVS2      1.244340
dtype: float64
```

Table. 13

- Even after dropping the columns there was not much improvements in the values of coefficients they are more or less the same.



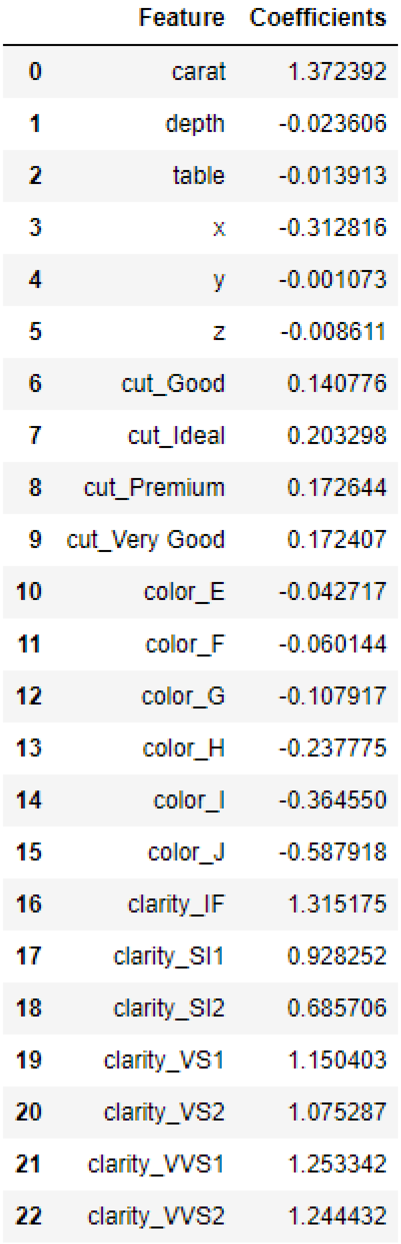
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.921
Model:                            OLS   Adj. R-squared:                  0.921
Method:                 Least Squares   F-statistic:                 1.046e+04
Date:                Fri, 22 Oct 2021   Prob (F-statistic):               0.00
Time:                        12:19:53   Log-Likelihood:                -2860.4
No. Observations:               18853   AIC:                             5765.
Df Residuals:                   18831   BIC:                             5937.
Df Model:                          21
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept       -1.0370      0.022    -46.565      0.000      -1.081      -0.993
carat            1.3722      0.010    132.465      0.000       1.352       1.392
depth           -0.0245      0.002     -9.822      0.000      -0.029      -0.020
table           -0.0138      0.003     -5.003      0.000      -0.019      -0.008
x               -0.3220      0.010    -30.980      0.000      -0.342      -0.302
cut_Good         0.1406      0.015      9.579      0.000       0.112       0.169
cut_Ideal        0.2033      0.015     13.874      0.000       0.175       0.232
cut_Premium      0.1727      0.014     12.223      0.000       0.145       0.200
cut_Very_Good    0.1722      0.014     12.191      0.000       0.144       0.200
color_E         -0.0428      0.008     -5.699      0.000      -0.058      -0.028
color_F         -0.0601      0.008     -7.836      0.000      -0.075      -0.045
color_G         -0.1079      0.007    -14.449      0.000      -0.123      -0.093
color_H         -0.2378      0.008    -29.831      0.000      -0.253      -0.222
color_I         -0.3645      0.009    -41.029      0.000      -0.382      -0.347
color_J         -0.5879      0.011    -53.691      0.000      -0.609      -0.566
clarity_IF       1.3150      0.022     59.894      0.000       1.272       1.358
clarity_SI1      0.9282      0.019     48.944      0.000       0.891       0.965
clarity_SI2      0.6856      0.019     35.989      0.000       0.648       0.723
clarity_VS1      1.1502      0.019     59.506      0.000       1.112       1.188
clarity_VS2      1.0752      0.019     56.486      0.000       1.038       1.113
clarity_VVS1     1.2532      0.020     61.449      0.000       1.213       1.293
clarity_VVS2     1.2443      0.020     62.610      0.000       1.205       1.283
==============================================================================
Omnibus:                     4588.639   Durbin-Watson:                   1.991
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           195023.614
Skew:                           0.393   Prob(JB):                         0.00
Kurtosis:                      18.737   Cond. No.                         37.2
==============================================================================

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

Table. 14

- Not much improvement in the score of R-squared and Adj R-squared as well.
- Root Mean Squared (RMSE) – **0.281**

Comparison of Sklearn and Stats model –

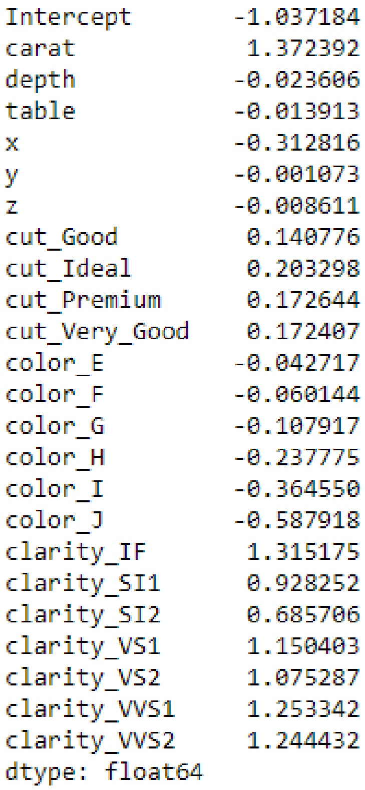
Coefficients from Sklearn Model



	Feature	Coefficients
0	carat	1.372392
1	depth	-0.023606
2	table	-0.013913
3	x	-0.312816
4	y	-0.001073
5	z	-0.008611
6	cut_Good	0.140776
7	cut_Ideal	0.203298
8	cut_Premium	0.172644
9	cut_Very Good	0.172407
10	color_E	-0.042717
11	color_F	-0.060144
12	color_G	-0.107917
13	color_H	-0.237775
14	color_I	-0.364550
15	color_J	-0.587918
16	clarity_IF	1.315175
17	clarity_SI1	0.928252
18	clarity_SI2	0.685706
19	clarity_VS1	1.150403
20	clarity_VS2	1.075287
21	clarity_VVS1	1.253342
22	clarity_VVS2	1.244432

Table.9

Coefficients from Stats Model



```
Intercept        -1.037184
carat             1.372392
depth            -0.023606
table            -0.013913
x                -0.312816
y                -0.001073
z                -0.008611
cut_Good          0.140776
cut_Ideal         0.203298
cut_Premium       0.172644
cut_Very_Good     0.172407
color_E          -0.042717
color_F          -0.060144
color_G          -0.107917
color_H          -0.237775
color_I          -0.364550
color_J          -0.587918
clarity_IF        1.315175
clarity_SI1       0.928252
clarity_SI2       0.685706
clarity_VS1       1.150403
clarity_VS2       1.075287
clarity_VVS1      1.253342
clarity_VVS2      1.244432
dtype: float64
```

Table.11

Intercept from sklearn -1.0371

Coefficients of both the models are exactly same.

RMSE value from train dataset from sklearn model is – 0.2816

RMSE value from train dataset from stats model is – 0.2816

- Both model are performing exactly the same and there is not much difference between the models. We can use any one them.

Visual representation of Actual y vs Predicted y for test data -

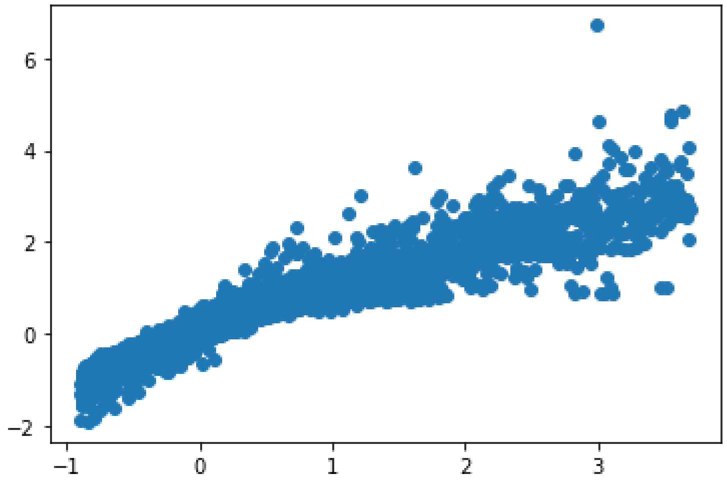


Fig. 16

Model represented in linear equation form:

-1.04 * Intercept + 1.37 * carat + -0.02 * depth + -0.01 * table + -0.32 * x + 0.14 * cut_Good + 0.2 * cut_Ideal + 0.17 * cut_Premium + 0.17 * cut_Very_Good + -0.04 * color_E + -0.06 * color_F + -0.11 * color_G + -0.24 * color_H + -0.36 * color_I + -0.59 * color_J + 1.32 * clarity_IF + 0.93 * clarity_SI1 + 0.69 * clarity_SI2 + 1.15 * clarity_VS1 + 1.08 * clarity_VS2 + 1.25 * clarity_VVS1 + 1.24 * clarity_VVS2 +

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

As per our EDA analysis we can conclude the following inferences:

For 'cut' factor Fair, Premium, Good are bringing in maximum profits and Ideal is bringing in least profits.

For 'color' factor H, I, J are bringing in more profits and lowest for color D and E.

For 'clarity' factor are SI2, l1 are bringing maximum profits and lowest for VVS1 and IF.

Coefficients from both the model are more or less same, by checking the values of coefficient's we can find the best attributes in the data.

The best attributes in the model are:

- clarity_IF
- clarity_SI1
- clarity_SI2
- clarity_VS1
- clarity_VS2
- clarity_VVS1
- clarity_VVS2

Positive value closer to 1 shows strong positive linear relationship with target variable, in our case is 'Price'.

So we can easily say the clarity is a very important factor that have high influence on price.

Cut is also showing positive relationship with price but not more than clarity.

Recommendations:

- The fair, premium, good cut types are the one which are bringing profits, marketing decisions needs to be made considering this to increase profits.
- The clarity of the diamond is the next important attribute and it has high impact on price, so business should make decision keeping clarity on mind.

Problem 2: Logistic Regression and Linear Discriminant Analysis

Executive Summary

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

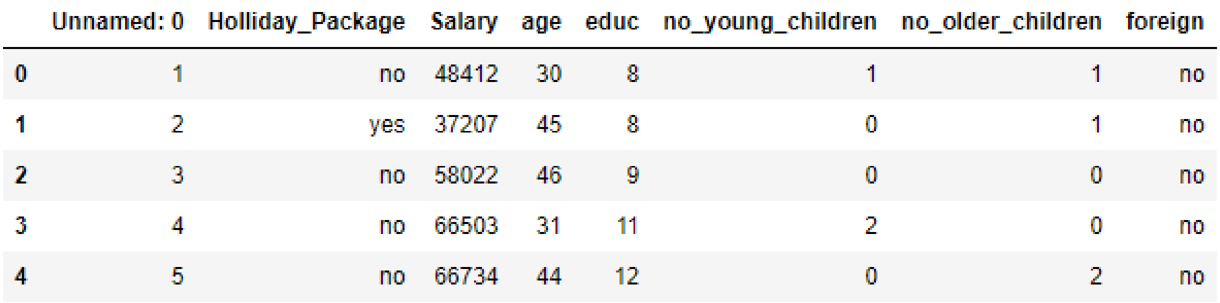
Data Dictionary

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

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

Exploratory Data Analysis

Dataset Sample

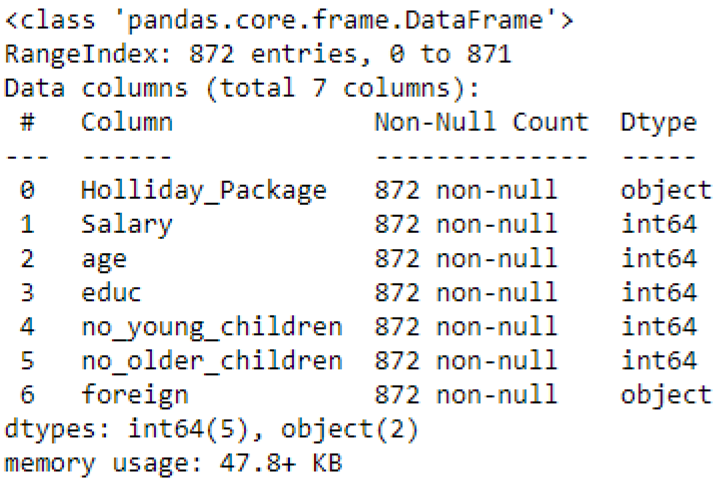


	Unnamed: 0	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
0	1	no	48412	30	8	1	1	no
1	2	yes	37207	45	8	0	1	no
2	3	no	58022	46	9	0	0	no
3	4	no	66503	31	11	2	0	no
4	5	no	66734	44	12	0	2	no

Table. 15

I am dropping ‘Unnamed:0’ column here itself as it is just index numbers and will not contribute to anything in model evaluation.

Dataset Info

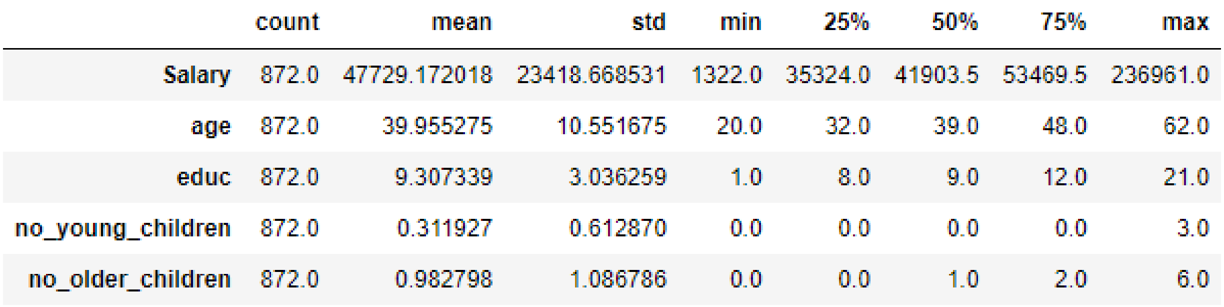


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 872 entries, 0 to 871
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Holliday_Package   872 non-null    object
 1   Salary             872 non-null    int64 
 2   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. 16

- The given dataset has now 7 columns out of which following datatypes:
  - Int64 type – 5 column
  - Object type – 2 column
- Dataset has 872 rows and 7 columns

Summary of Dataset

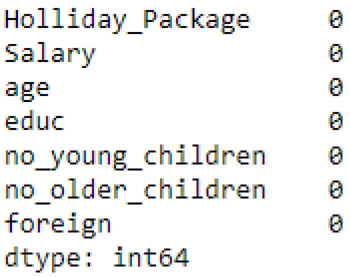


	count	mean	std	min	25%	50%	75%	max
Salary	872.0	47729.172018	23418.668531	1322.0	35324.0	41903.5	53469.5	236961.0
age	872.0	39.955275	10.551675	20.0	32.0	39.0	48.0	62.0
educ	872.0	9.307339	3.036259	1.0	8.0	9.0	12.0	21.0
no_young_children	872.0	0.311927	0.612870	0.0	0.0	0.0	0.0	3.0
no_older_children	872.0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

Table. 17

- Object type variable also needs to be encoded.
- We are not performing scaling.

Checking for missing values in dataset

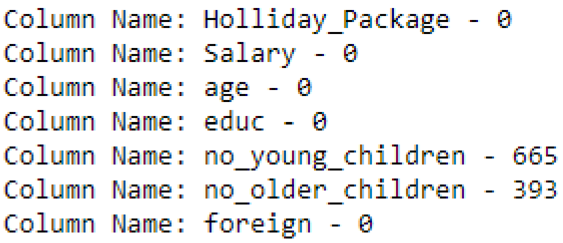


```
Holliday_Package     0
Salary               0
age                  0
educ                 0
no_young_children    0
no_older_children    0
foreign              0
dtype: int64
```

Table. 18

- There are no missing values present in our dataset.

Checking for entries' with 0 in dataset



```
Column Name: Holliday_Package - 0
Column Name: Salary - 0
Column Name: age - 0
Column Name: educ - 0
Column Name: no_young_children - 665
Column Name: no_older_children - 393
Column Name: foreign - 0
```

Table. 19

- These '0' values in both columns are valid entries.

Univariate Analysis

We perform the univariate analysis on the data set and display distplot to check distribution for each continuous type column and use boxplot to check for outliers if any.

Continuous Type Columns

salary

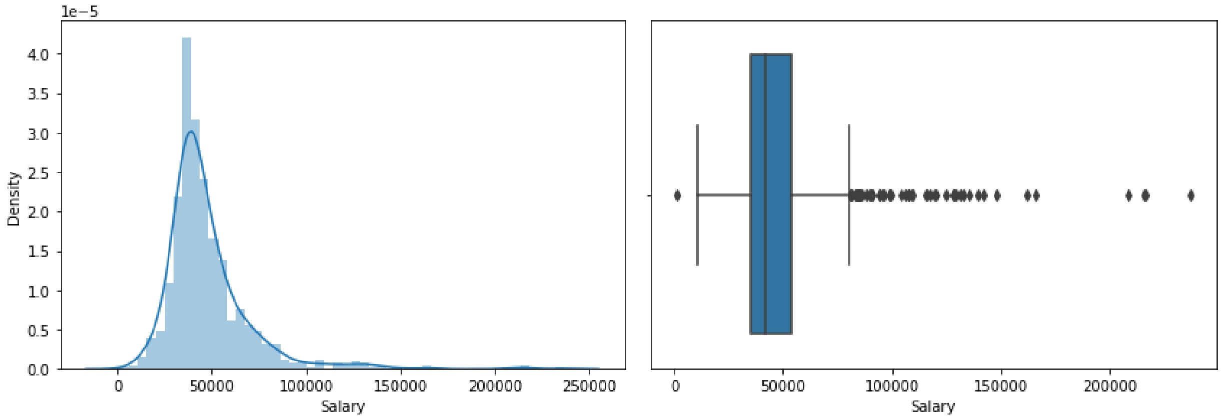


Fig. 17

- The mean number of salary is 47729.17 whereas the SD is 23418.66
- The maximum value is 236961.0 and min value is 1322
- The distribution is right skewed.
- Outliers are present.

age

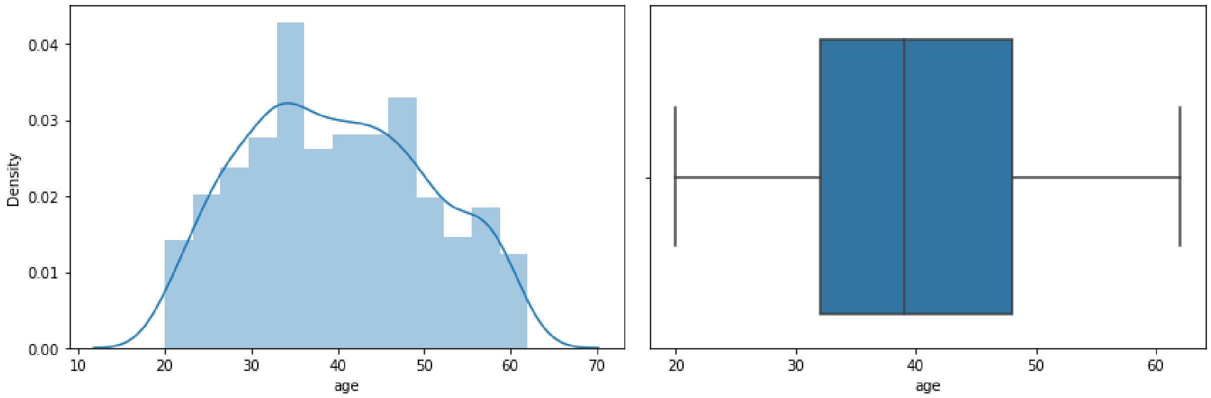


Fig. 18

- The mean number of age is 39.95 whereas the SD is 10.55
- The maximum value is 62 and min value is 20.
- The distribution is somewhat normally distributed.
- Outliers are not present in column.

educ

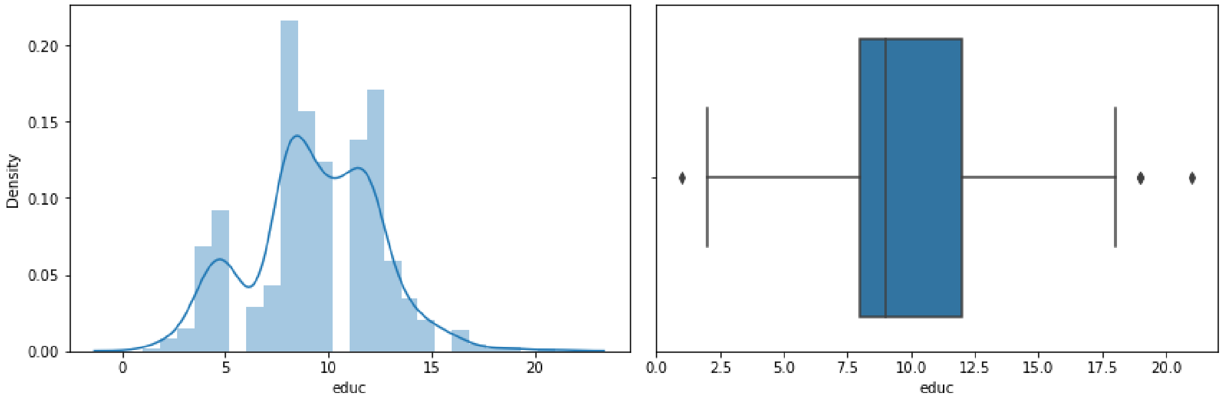


Fig. 19

- The mean number of educ is 9.30 whereas the SD is 3.03
- The maximum value is 21 and min value is 1
- The distribution is somewhat normally distributed.
- Outliers are present.

no_young_children

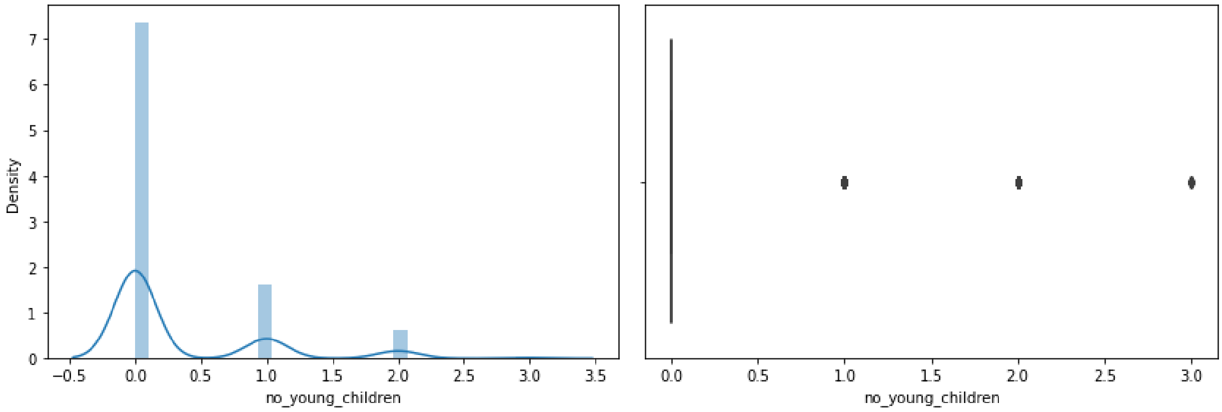


Fig. 20

- The mean number of no_young_children is 0.31 whereas the SD is 0.61
- The maximum value is 3 and min value is 0
- The distribution is skewed towards right.
- Outliers are present.

no_older_children

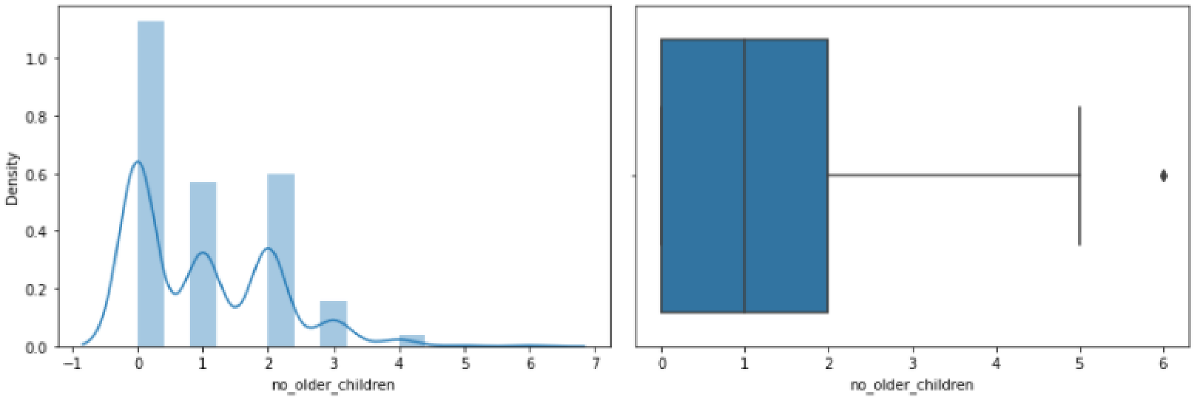


Fig. 21

- The mean number of no_older_children is 0.98 whereas the SD is 1.08
- The maximum value is 6 and min value is 0
- The distribution is skewed towards right.
- Outliers are present.

Categorical Type Columns

foreign

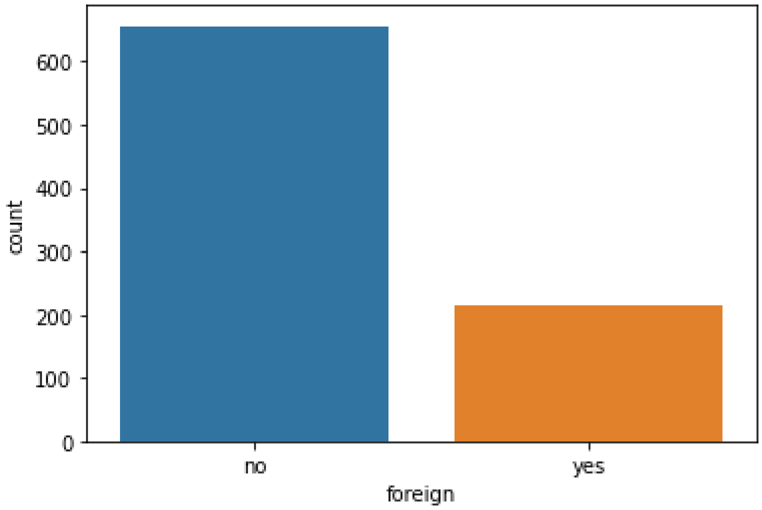


Fig. 22

- Number of unique values - 2

Holiday_Package – Our Target Variable

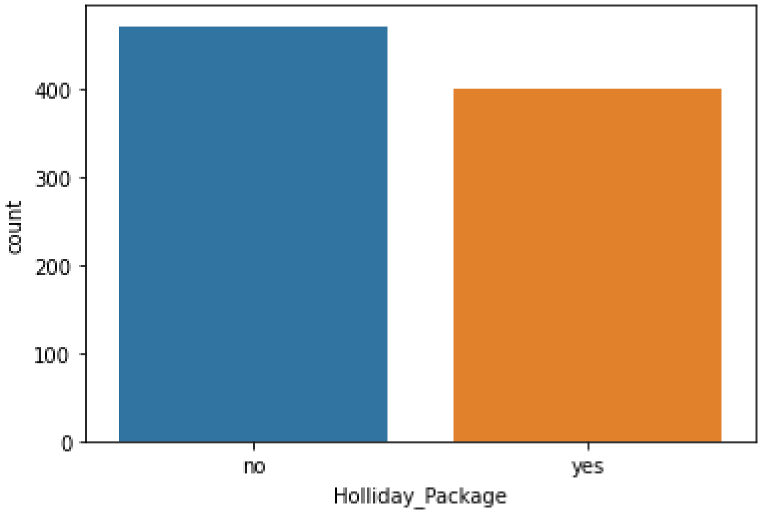


Fig. 23

- Value count of Target Variable –
  no 0.540138
  yes 0.459862
  Name: Holliday_Package, dtype: float64

- We need to make model to prediction of this variable using Logistic Regression Analysis.

Bi/Multivariate Analysis

Bi variant analysis of our target variable by comparing it with salary, age and educ, no_young_children and no_older_children columns.

Holliday_Package vs salary:

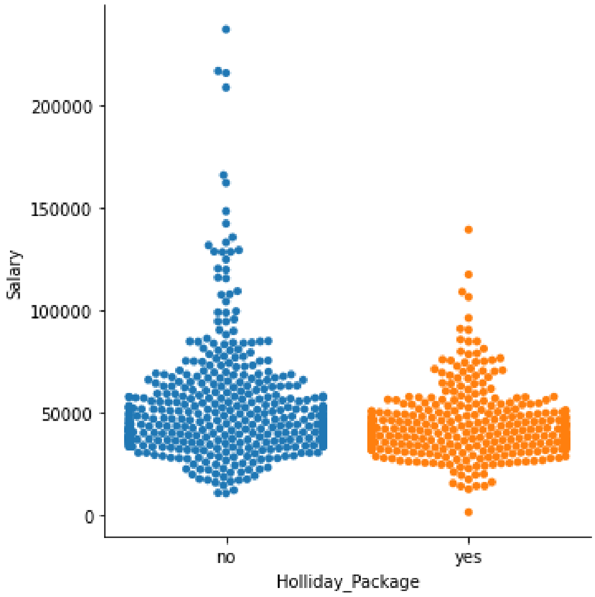


Fig. 24

- People in with salary not opting for holiday package is more.
- People with higher salary are not opting for holiday packages.

Holliday_Package vs age:

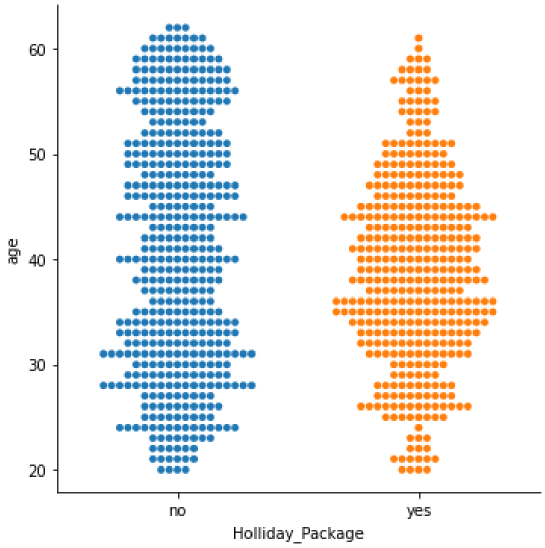


Fig. 25

- People with age between 30 and 50 aprox are opting for holidays packages more than any other age group.

Holliday_Package vs educ:

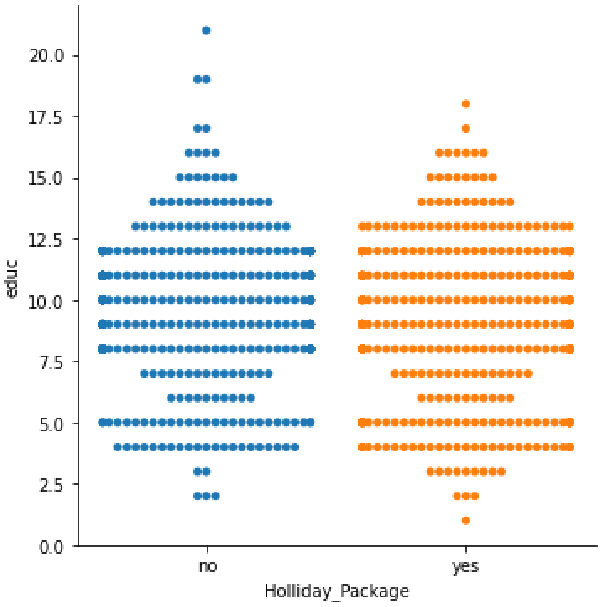


Fig. 26

- People having no.s of formal education of 4 and 5 are opting more for holiday packages
- People having no.s of formal education between 8 to 13 approx. are opting more for holiday packages.

Holliday_Package vs no_young_children:

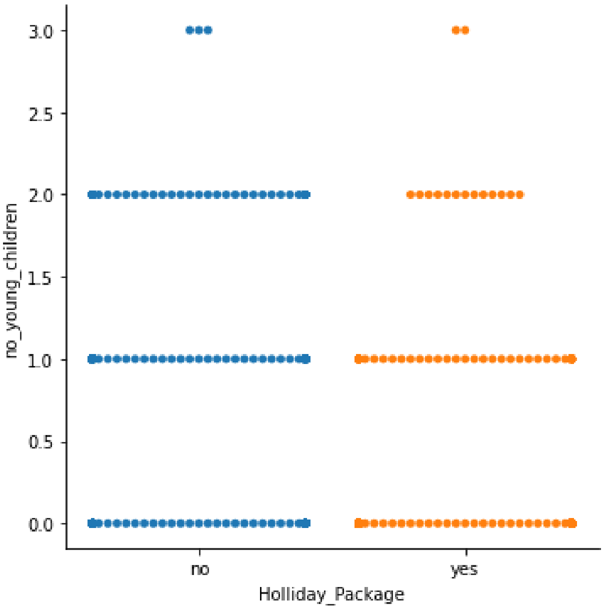


Fig. 27

- People having 0 or only 1 children opt more for holiday packages than people having 2 and 3 children.

Holliday_Package vs no_older_children:

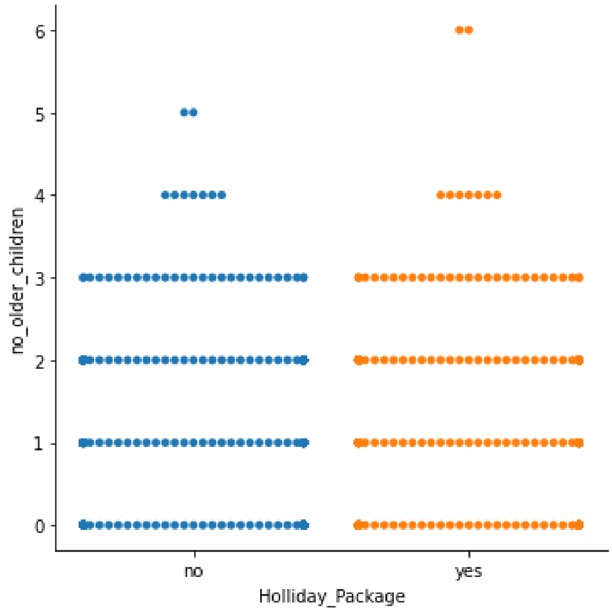


Fig. 28

- People with older children having 0,1,2,3 are opting for holiday packages more than people having 4,5 and 6 older children.

For Multivariate Analysis we use pair plot and Heatmap:

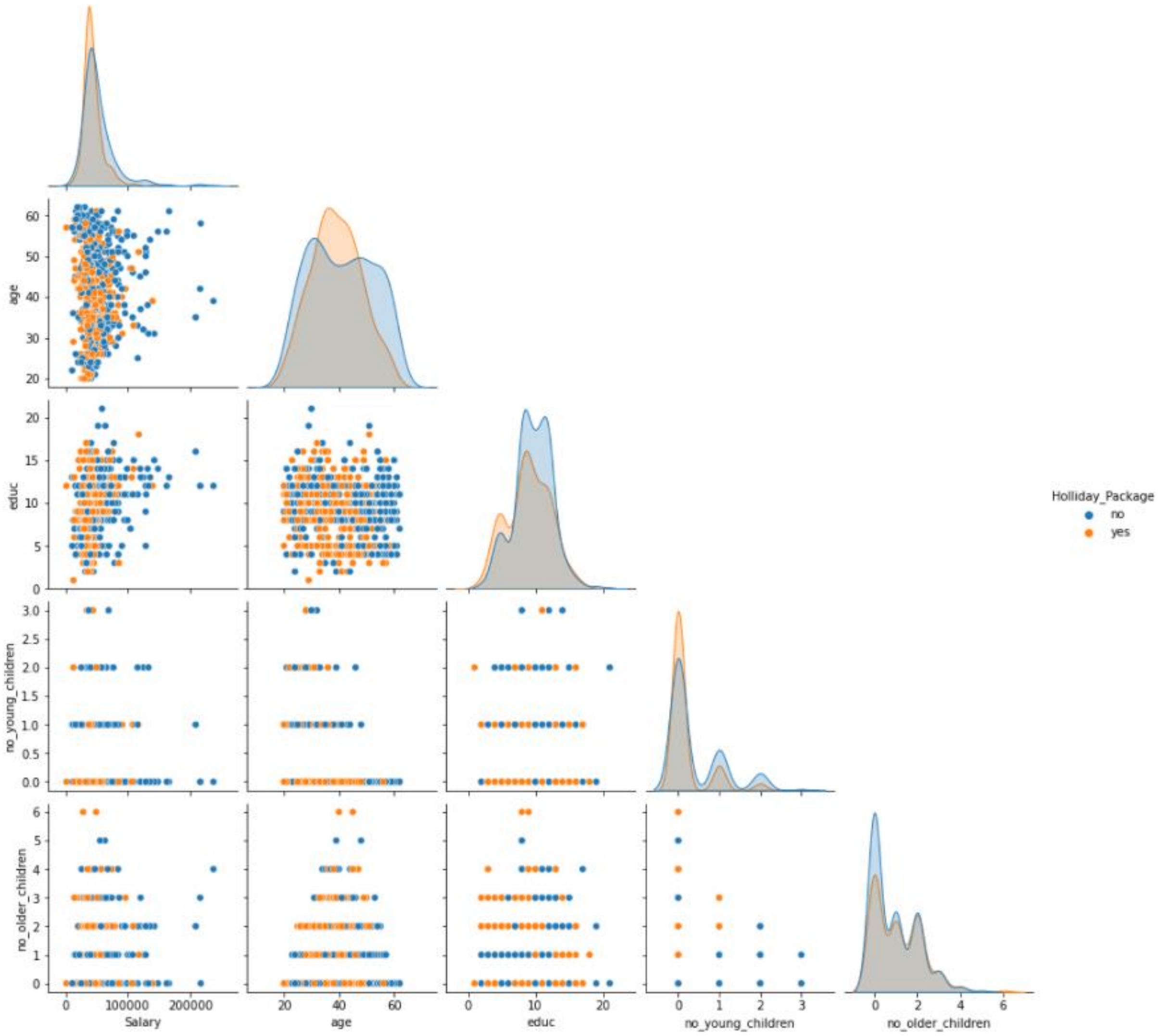


Fig. 29

- If we analyse the diagonals the for each variable, the ideal situation is that the areas of both values ‘yes’ and ’no’ do not overlap that much so that our Logistic Regression Model can perform meaningful prediction.
- We want variables that where the two classes are separated out significantly. All the variable in our dataset are weak and poor predictors. We can see that for all the variable the distribution of our target variable is overlapping.

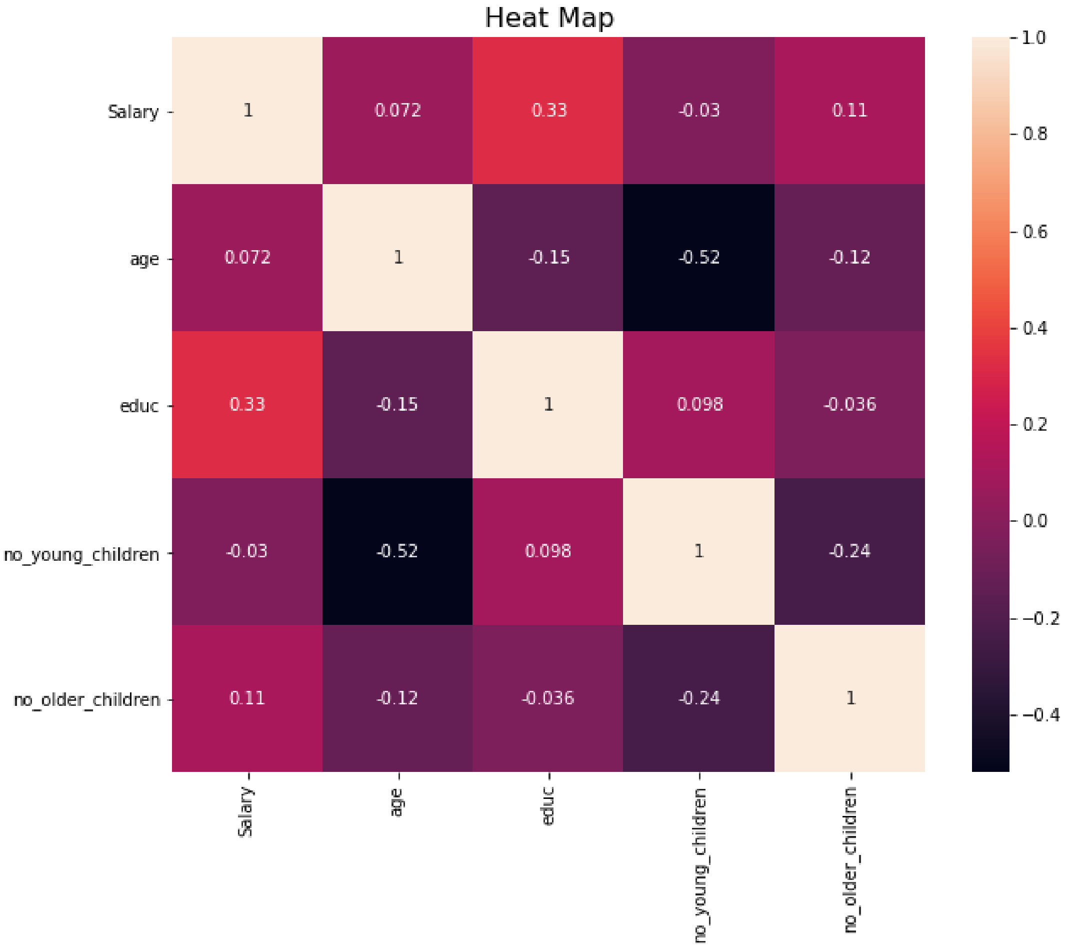


Fig. 30

- Degree of correlation between the columns is represented by the above heatmap, 1 being max value of correlation and 0 below no correlation between them.

Pair with high correlation are as follows:

- age and salary
- no_young_children and educ

Before encoding the object type columns we are performing outlier's treatment for the numeric columns first. As outlier can have adverse impacts on the log odds regression.

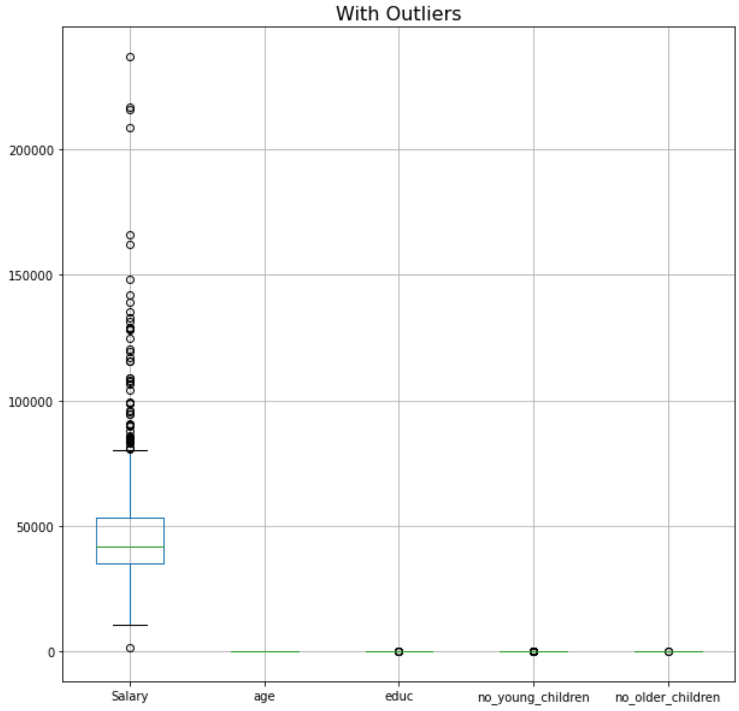


Fig. 31

- Above figure shows continuous columns with outlier's.

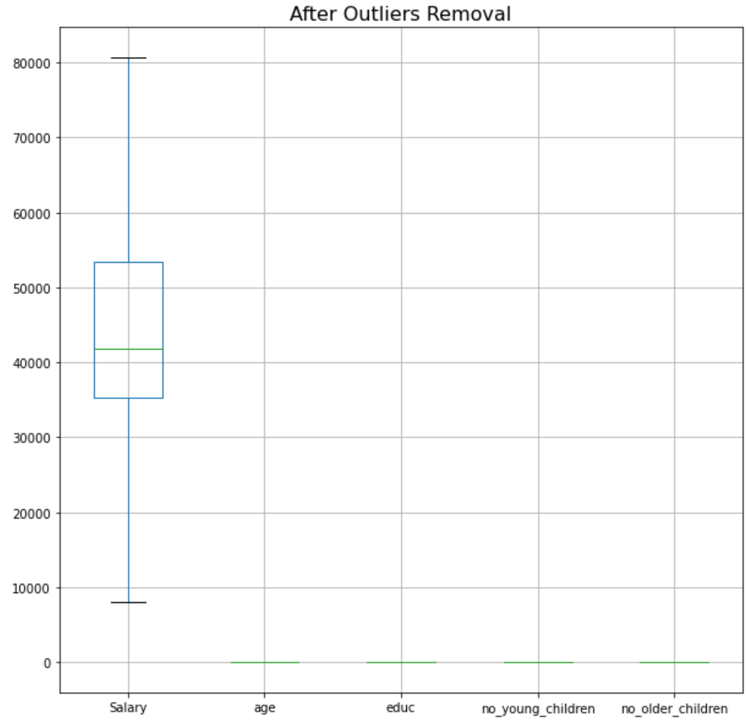
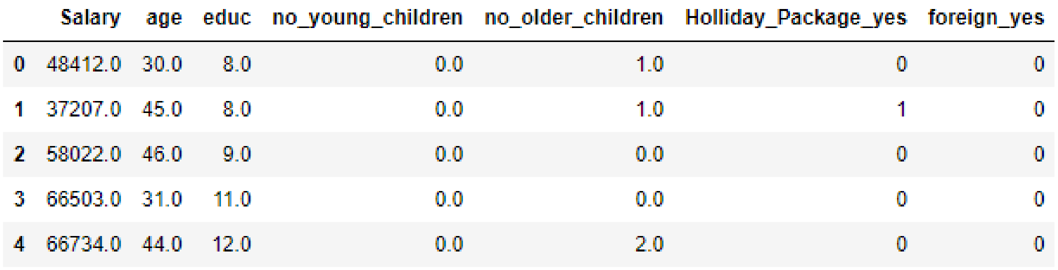


Fig. 32

- Figure after treatment of all the outlier's.

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).

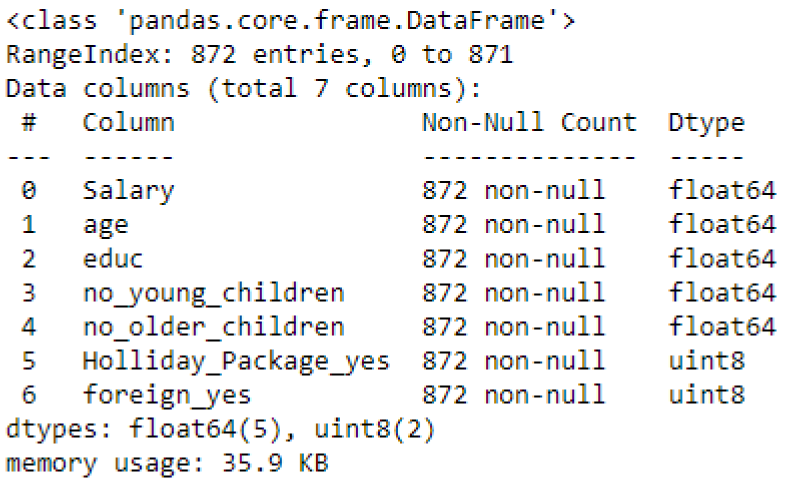
Dataset after performing label encoding for the object type variable in dataset.



	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. 20

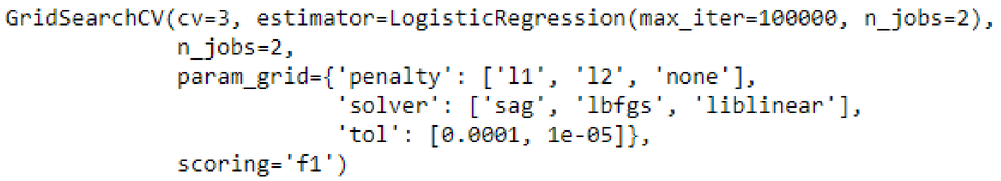
Dataset info after label encoding:



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 872 entries, 0 to 871
Data columns (total 7 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Salary                872 non-null    float64
 1   age                   872 non-null    float64
 2   educ                  872 non-null    float64
 3   no_young_children     872 non-null    float64
 4   no_older_children     872 non-null    float64
 5   Holliday_Package_yes  872 non-null    uint8  
 6   foreign_yes           872 non-null    uint8  
dtypes: float64(5), uint8(2)
memory usage: 35.9 KB
```

Table. 21

- Now using ‘Holliday_Pakage_yes’ as our target variable. We split the data into train and test set into 70:30 percent size ratio respectively.
- We are using grid search for finding the best parameters for the model.

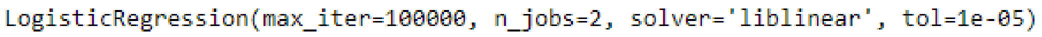


```
GridSearchCV(cv=3, estimator=LogisticRegression(max_iter=100000, n_jobs=2),
             n_jobs=2,
             param_grid={'penalty': ['l1', 'l2', 'none'],
                         'solver': ['sag', 'lbfgs', 'liblinear'],
                         'tol': [0.0001, 1e-05]},
             scoring='f1')
```

Table. 22

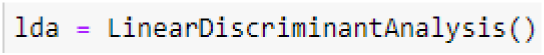
- Using the best parameters of this grid we fit it to our Logistic Regression model.

- Now using **Logistic Regression Model** we make prediction for training and testing data separately using these best estimators.



```
LogisticRegression(max_iter=100000, n_jobs=2, solver='liblinear', tol=1e-05)
```

- We also create **Linear Discriminant Analysis** model and we make prediction for training and testing data separately.



```
lda = LinearDiscriminantAnalysis()
```

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.

General Rules for performance Metrics:

- A model is said to perform well when it runs well in train as well as test data both
- The auc_ruc_score of both train and test should not differ more than 10% for the model to be valid.
- Higher the auc_ruc_score the better the model.
- If the difference between the auc_ruc_score for train and test set is greater than 10% then problem for overfitting and under fitting may arise.
  ‘Overfitting’ is when model perform well in train but not in test set.
  ‘Under-fitting’ is when model does not perform well in train but do well in test set.
  ‘Best Fit’ is when model perform well in train as well as in test to a similar level.

- Confusion matrix and classification report is made for all the models.
- In classification report - ‘1’ is our value of importance for model i.e. people who have claimed insurance.
  - Recall indicates how many of the actual data points are identified as True data points by the model.
  - Precision indicates the points that are identified as positive by the model, how many are really positive.
  - The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0.

  We will focus on recall and precision value in Classification Report for each model.

- An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:
  True Positive Rate (TPR) is a synonym for recall and is therefore defined as follows:

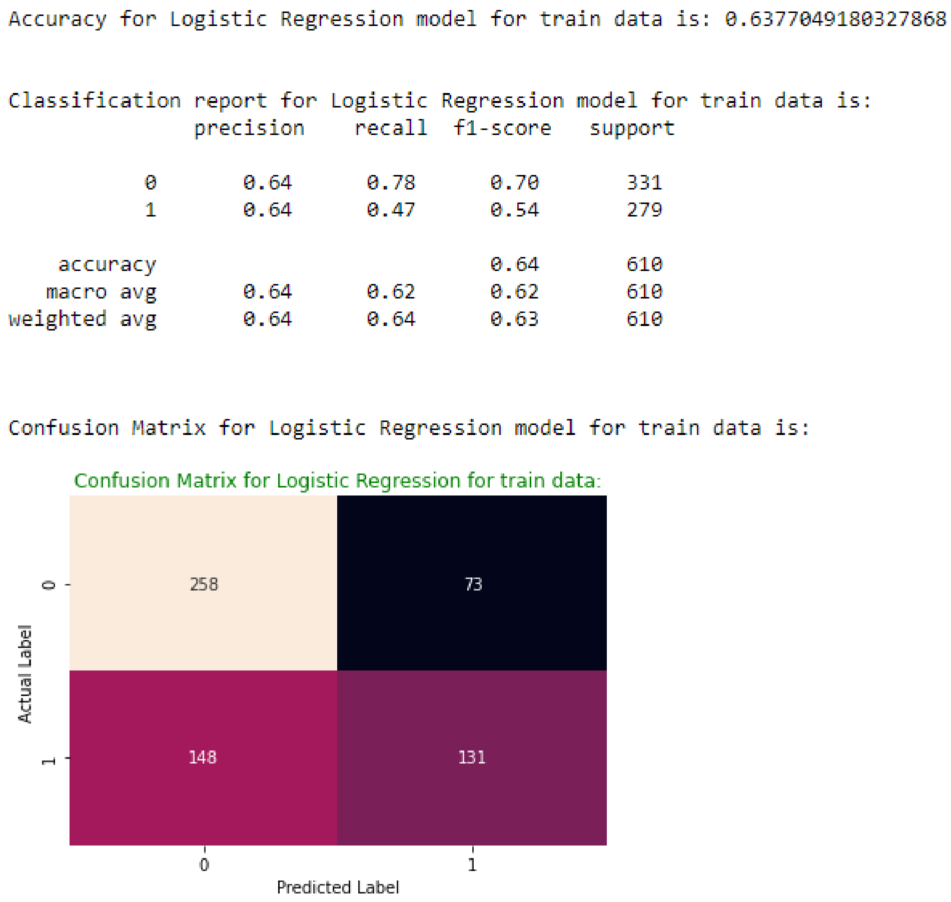
  $$TPR=TP/TP+FN$$

  False Positive Rate (FPR) is defined as follows:

  $$FPR=FP/FP+TN$$
- An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives.

Performance Metrics – Logistic Regression Model

- **Train Set**

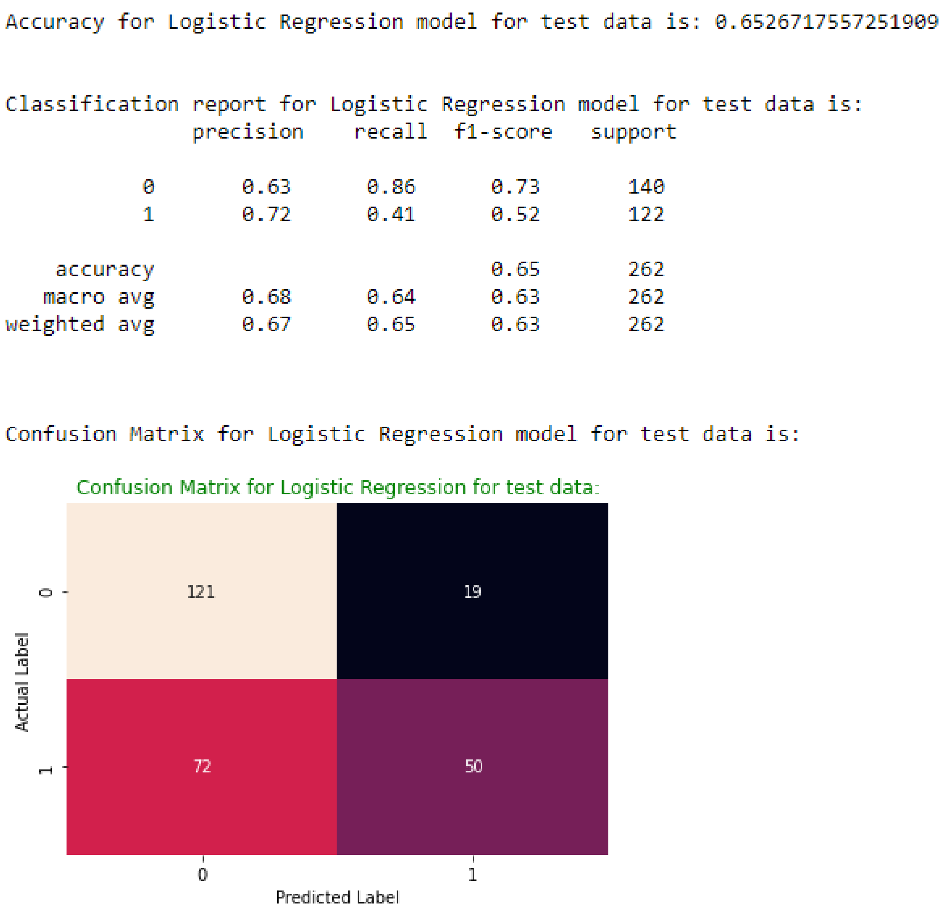


```
Accuracy for Logistic Regression model for train data is: 0.6377049180327868


Classification report for Logistic Regression model for train data is:
               precision    recall  f1-score   support

           0       0.64      0.78      0.70       331
           1       0.64      0.47      0.54       279

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



Confusion Matrix for Logistic Regression model for train data is:
```

Table. 23

AUC Score of Train set is – 0.66

- **Test Set**



```
Accuracy for Logistic Regression model for test data is: 0.6526717557251909


Classification report for Logistic Regression model for test data is:
              precision    recall  f1-score   support

           0       0.63      0.86      0.73       140
           1       0.72      0.41      0.52       122

    accuracy                           0.65       262
   macro avg       0.68      0.64      0.63       262
weighted avg       0.67      0.65      0.63       262


Confusion Matrix for Logistic Regression model for test data is:
```

Table. 24

AUC Score of Test set is – 0.68

- **ROC Curve for Train and Test set – Logistic Regression Model**

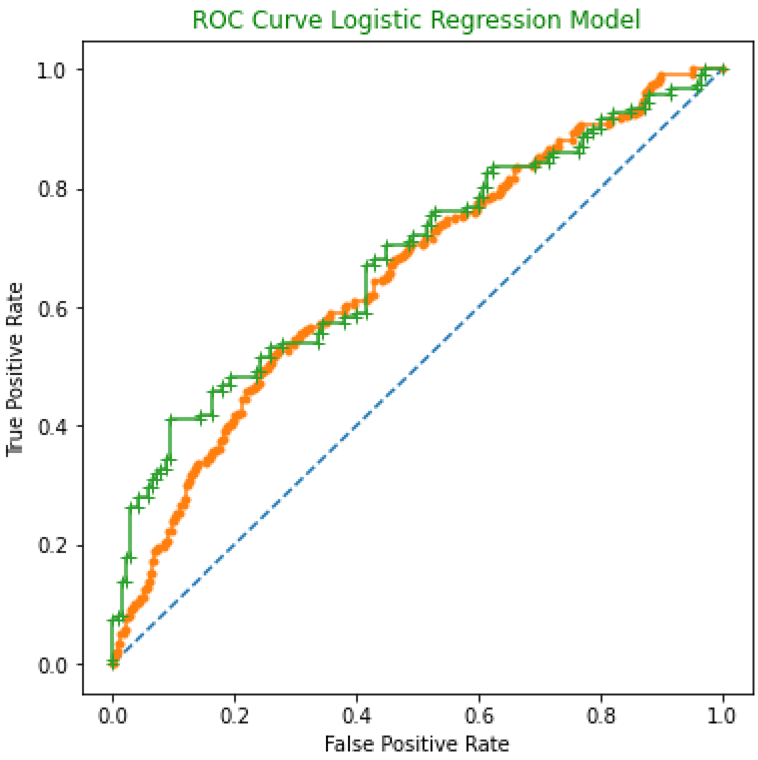


Fig. 33

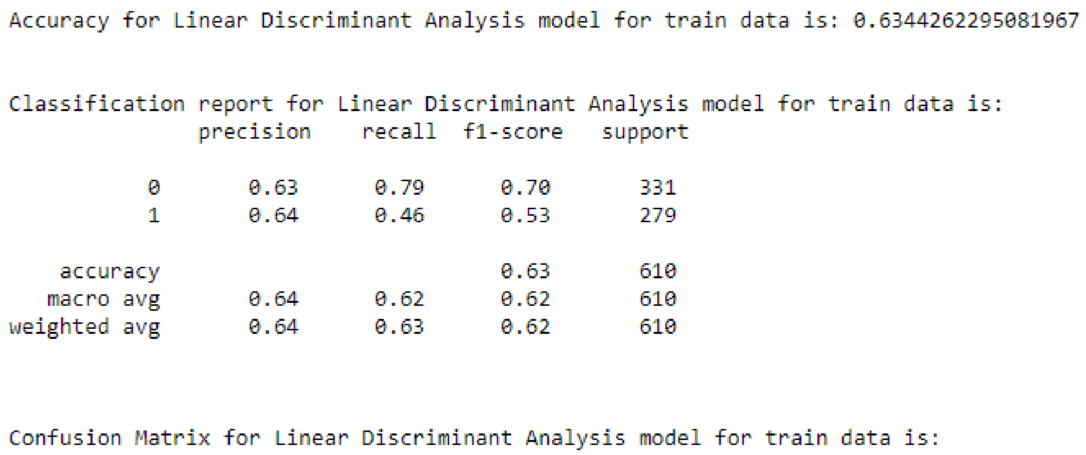
Line plot with dot "." marker is for train set and plus"+" marker is for test set.

Inferences:

- Logistic Regression model is performing well in both train and test set so it is a valid model.
  AUC score for train is 0.66
  AUC score for test is 0.68
- For test set for value '1' we have :
  Recall – 0.41
  Precision – 0.72
  This is model is performing well in test data and is able to correctly identify the labels for test set with acceptable values of recall and precision.

Performance Metrics – Linear Discriminant Analysis Model

- **Train Set**



```
Accuracy for Linear Discriminant Analysis model for train data is: 0.6344262295081967


Classification report for Linear Discriminant Analysis model for train data is:
               precision    recall  f1-score   support

           0       0.63      0.79      0.70       331
           1       0.64      0.46      0.53       279

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


Confusion Matrix for Linear Discriminant Analysis model for train data is:
```

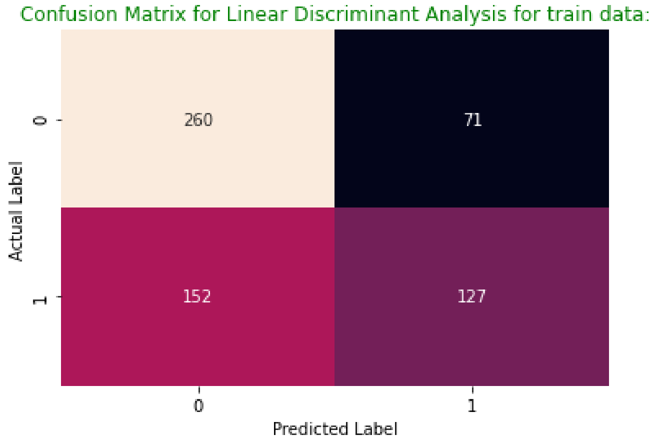
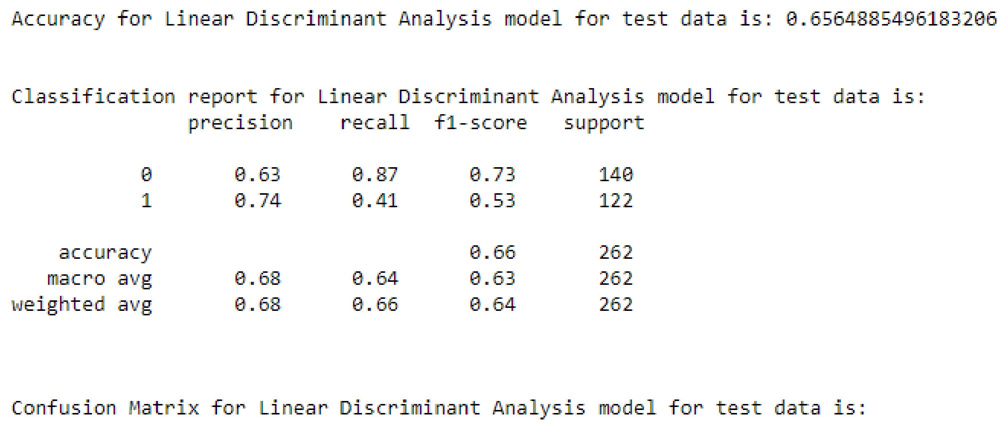


Table. 25

AUC Score of Train set is – 0.66

- **Test Set**



```
Accuracy for Linear Discriminant Analysis model for test data is: 0.6564885496183206


Classification report for Linear Discriminant Analysis model for test data is:
               precision    recall  f1-score   support

           0       0.63      0.87      0.73       140
           1       0.74      0.41      0.53       122

    accuracy                           0.66       262
   macro avg       0.68      0.64      0.63       262
weighted avg       0.68      0.66      0.64       262


Confusion Matrix for Linear Discriminant Analysis model for test data is:
```

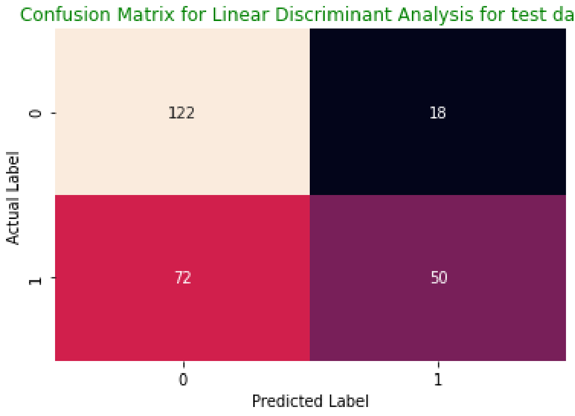


Table. 26

AUC Score of Test set is – 0.68

- **ROC Curve for Train and Test set – Linear Discriminant Analysis Model**

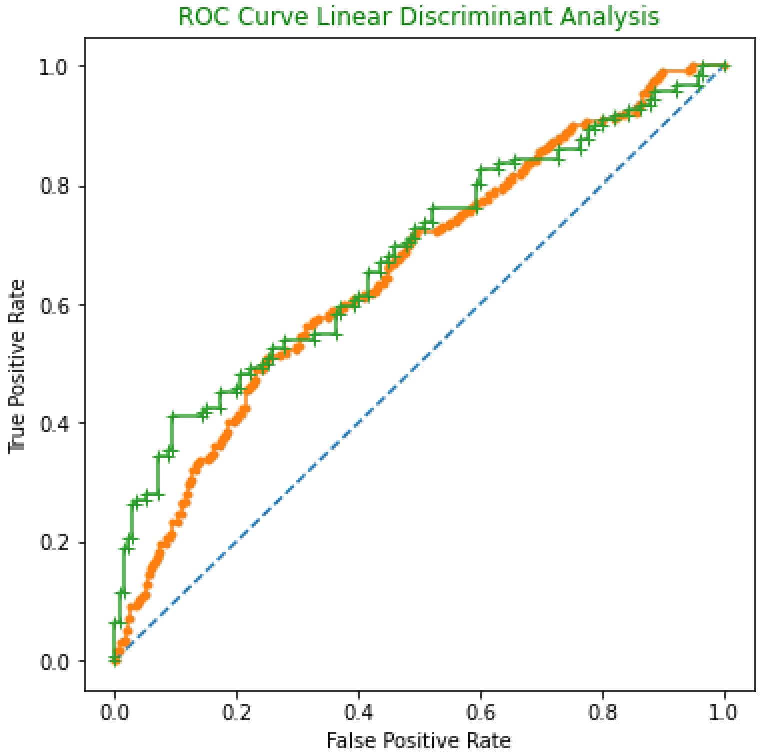


Fig. 34

Line plot with dot "." marker is for train set and plus"+" marker is for test set.

Inferences:

- Linear Discriminant Analysis model is performing well in both train and test set so it is a valid model.
  AUC score for train is 0.66
  AUC score for test is 0.68
- For test set for value '1' we have :
  Recall – 0.41
  Precision – 0.74
  This is model is performing well in test data and is able to correctly identify the labels for test set with acceptable values of recall and precision.

- **Comparing both the models metric :**

  Below table give details train and test metrics for our critical value '1' for the target variable.

	Logistic Regression Model		**Linear Discriminant Analysis**	
	Train Set	Test Set	Train Set	Test Set
Accuracy	0.63	0.65	0.63	0.65
Precision	0.64	0.72	0.64	0.74
Recall	0.47	0.41	0.46	0.41
F1-Score	0.54	0.52	0.53	0.53
AUC	0.66	0.68	0.66	0.68

Table. 27

Inferences:

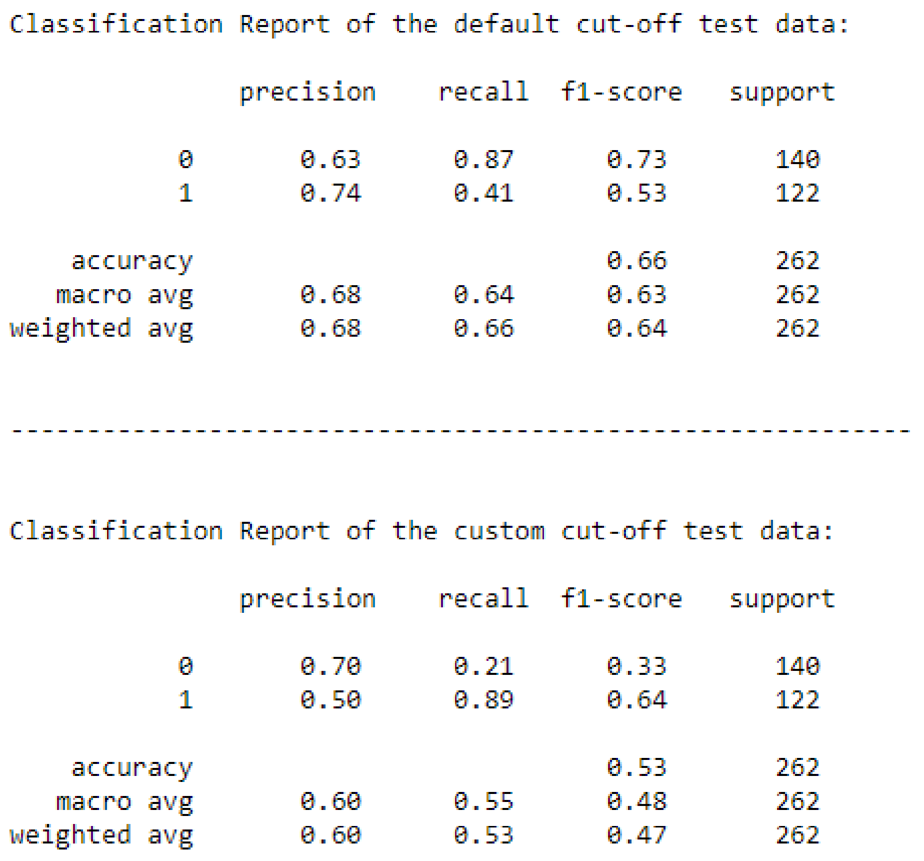
- Results from both the models is more or less same for all metrics, so both models are performing exactly same.

We can further find custom cut-off value for discriminating between class instead of using 0.5 as discriminating value

We see that for cut-off value of 0.3 we get the best score for :

F1-score – 0.64 and Accuracy Score – 0.53

Comparing the classification report using 0.5 cut-off (default) and using new cut-off of 0.3



```
Classification Report of the default cut-off test data:

              precision    recall  f1-score   support

           0       0.63      0.87      0.73       140
           1       0.74      0.41      0.53       122

    accuracy                           0.66       262
   macro avg       0.68      0.64      0.63       262
weighted avg       0.68      0.66      0.64       262

 ----------------------------------------------------------

Classification Report of the custom cut-off test data:

              precision    recall  f1-score   support

           0       0.70      0.21      0.33       140
           1       0.50      0.89      0.64       122

    accuracy                           0.53       262
   macro avg       0.60      0.55      0.48       262
weighted avg       0.60      0.53      0.47       262
```

Table. 28

Inferences:

- Even with custom cut-off there is not much significant improvement in the metrics as recall has improved from 41% to 89% but precision has fallen from 74% to 50% for 1 value. F1-score has also increased from 53% to 64%.

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

Since both the models 'Logistic Regression Model' and 'Linear Discriminant Analysis" are performing same as per there performance metrics, we can choose any one for our business recommendation analysis.

As per our EDA analysis, if we observe the diagonal of pairplot for the given dataset we can say that most of the predictors in the dataset are very poor predictors as they are not discriminating between the values of our target variable significant enough. But if we have to pick the best from all these poor predictors we can salary, age and educ are somewhat good predictors as the values of target variable are not that overlapping.

By analysing the cat plot of different variables against our target variable we can make the following inferences:

- People with age between 30 and 50 age approx. are opting more for the packages than any other age group.
- People with higher salary are not opting for holiday packages.
- People with 2 young children are opting less for holiday packages

Recommendation:

- More good predictors need to be identified to make more meaningful model.
- Need to target people with age group between 20 – 30 and age group above 50 to make them opt for packages.
- Need to identify the reason why high income group of people are not opting for holiday packages.
- How we can attract the high income group to opt for holiday packages.
- Holiday packages exclusive tailored for high income groups and their preference can make them to shift.
- Reasons why people with 2 young children are not opting more need to be identified and changes in packages need to be made cratering this group.

THE END