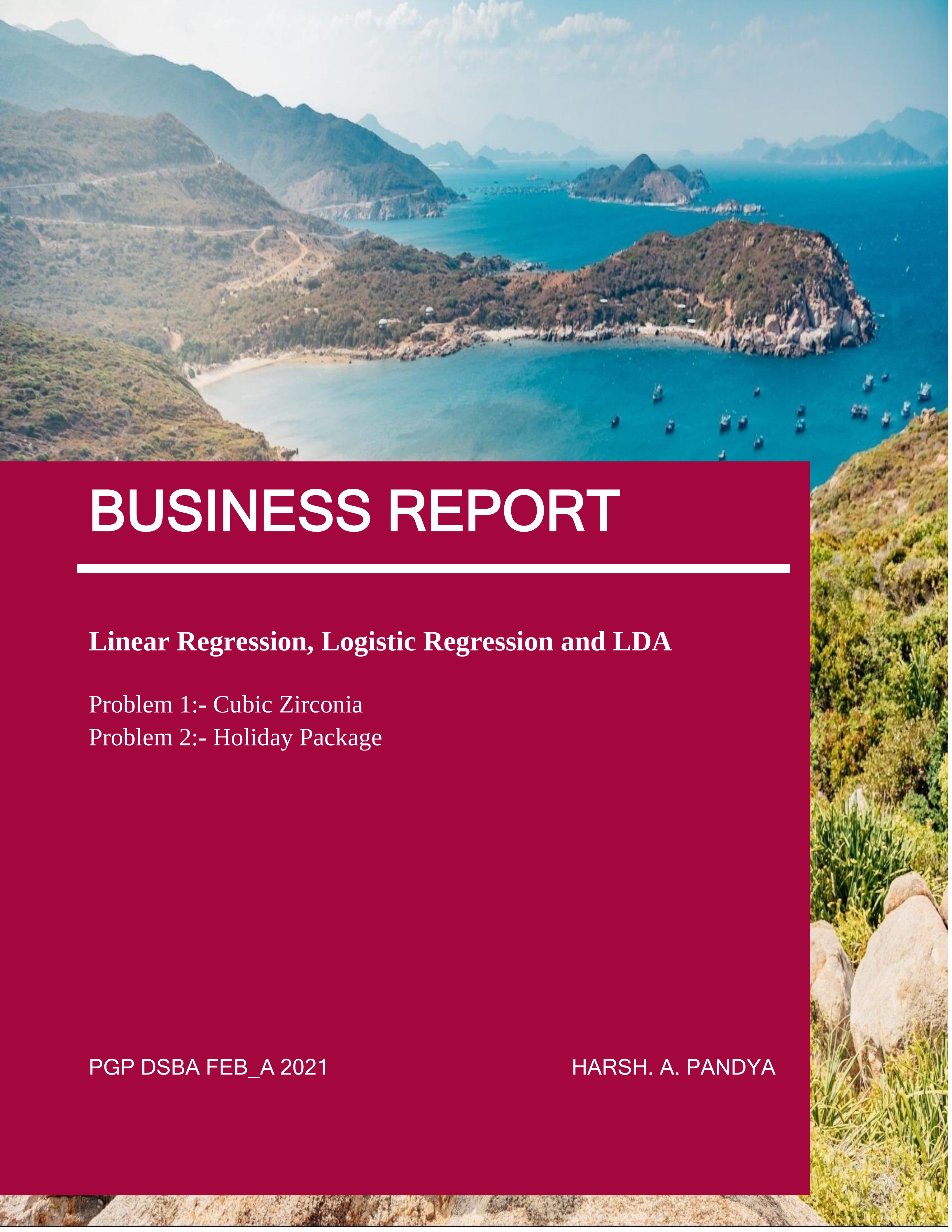


BUSINESS REPORT

Linear Regression, Logistic Regression and LDA

Problem 1:- Cubic Zirconia
Problem 2:- Holiday Package

PGP DSBA FEB_A 2021

HARSH. A. PANDYA

INDEX

Table of Content

List of Table

List of Figure

List of Figure

Problem 2:- Holiday Package

SUMMARY

Problem 1:- Linear Regression (Cubic Zirconia)

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.

Problem 2:- Logistic Regression & LDA (Holiday Package)

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.

Problem 1:- Cubic Zirconia (Linear Regression)

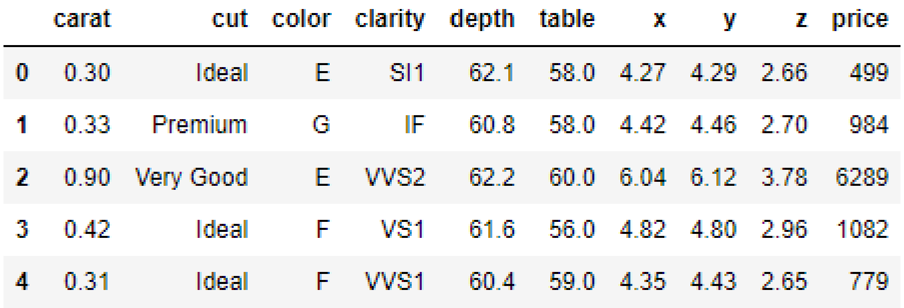
1.1 Read the data and do exploratory data analysis. Describe the data briefly. (Check the null values, Data types, shape, EDA). Perform Univariate and Bivariate Analysis.

Solution:-

Dataset has 26967 observations and 10 attributes.

- 6 attributes named **carat, depth, table, x, y, z** are of **float64 type**
- 3 attributes named **cut, color, clarity** are of **object type**
- 1 attribute named **price** is of **integer type**

Let's start the data exploration step with the head function to look at the first 5 rows.

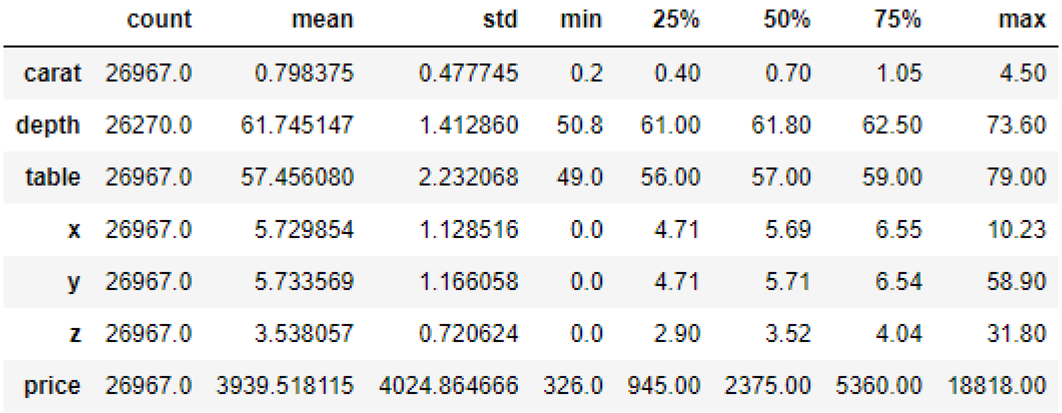


	carat	cut	color	clarity	depth	table	x	y	z	price
0	0.30	Ideal	E	SI1	62.1	58.0	4.27	4.29	2.66	499
1	0.33	Premium	G	IF	60.8	58.0	4.42	4.46	2.70	984
2	0.90	Very Good	E	VVS2	62.2	60.0	6.04	6.12	3.78	6289
3	0.42	Ideal	F	VS1	61.6	56.0	4.82	4.80	2.96	1082
4	0.31	Ideal	F	VVS1	60.4	59.0	4.35	4.43	2.65	779

(Table:-1 Data set)

Unnamed: 0 seems to be of no use for our dataset and it's dropped due to that.

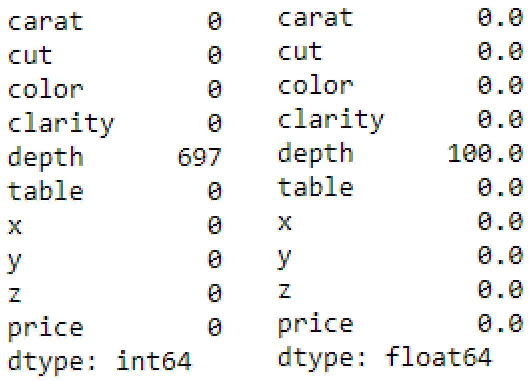
After checking the summary of the data, there seems to be some outliers in the dataset. Let's check them by doing further exploration.



	count	mean	std	min	25%	50%	75%	max
carat	26967.0	0.798375	0.477745	0.2	0.40	0.70	1.05	4.50
depth	26270.0	61.745147	1.412860	50.8	61.00	61.80	62.50	73.60
table	26967.0	57.456080	2.232068	49.0	56.00	57.00	59.00	79.00
x	26967.0	5.729854	1.128516	0.0	4.71	5.69	6.55	10.23
y	26967.0	5.733569	1.166058	0.0	4.71	5.71	6.54	58.90
z	26967.0	3.538057	0.720624	0.0	2.90	3.52	4.04	31.80
price	26967.0	3939.518115	4024.864666	326.0	945.00	2375.00	5360.00	18818.00

(Table:-2 Describe Data)

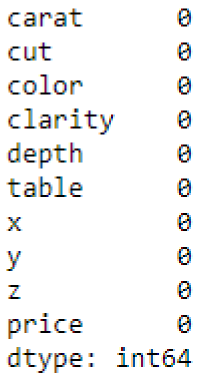
Dataset has null values in the depth column. By checking the percentage of null values, it seems to be that there are null values only in one column that is depth.



```
carat         0    carat         0.0
cut           0    cut           0.0
color         0    color         0.0
clarity       0    clarity       0.0
depth       697    depth       100.0
table         0    table         0.0
x             0    x             0.0
y             0    y             0.0
z             0    z             0.0
price         0    price         0.0
dtype: int64       dtype: float64
```

(Fig:-1 Null Value Check)

There is so maney null values are there we need to treat the values here we treat the null values.



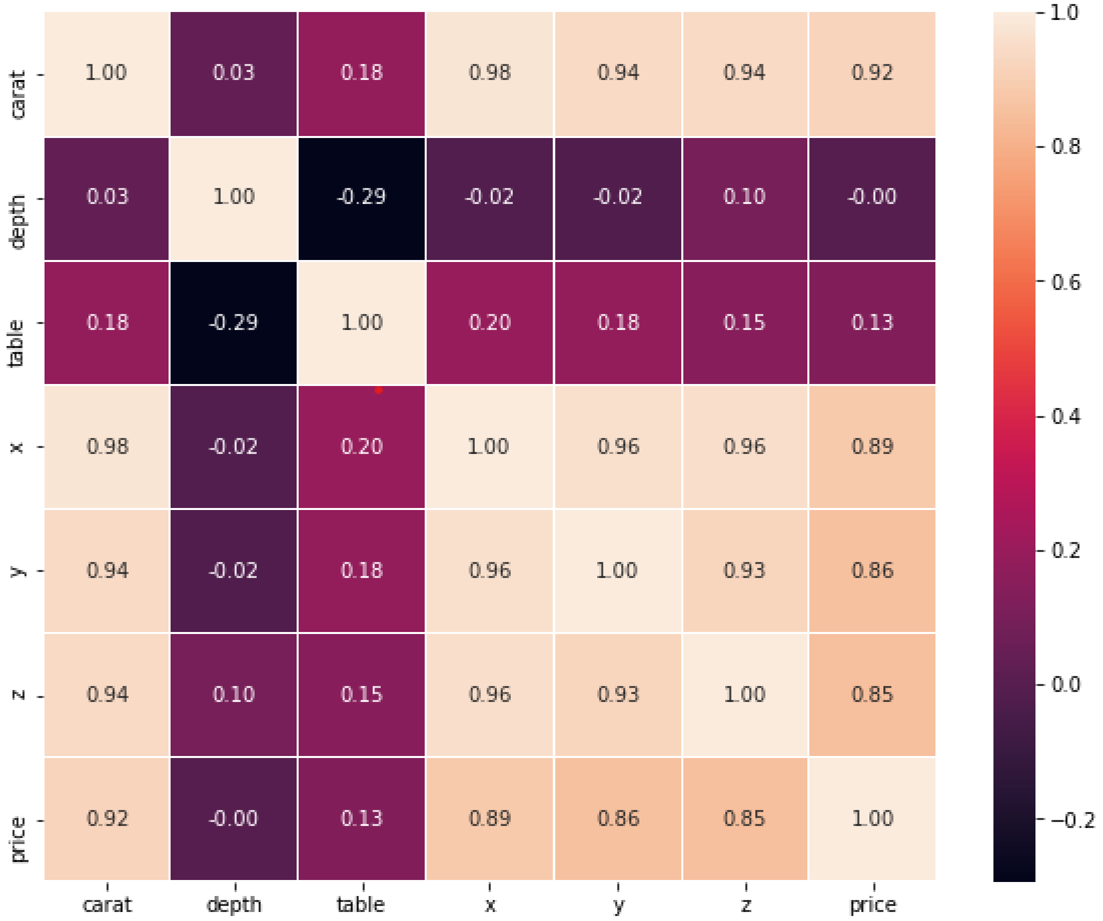
```
carat       0
cut         0
color       0
clarity     0
depth       0
table       0
x           0
y           0
z           0
price       0
dtype: int64
```

(Fig:-2 After Null Value Treatment)

Heat map shows that the below parameters of the cubic zirconia are highly correlated with each other

- Carat weight
- Length in mm.
- Width in mm.
- Height in mm.
- Price

The Height of a cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter is <u>least correlated</u> with all the attributes.



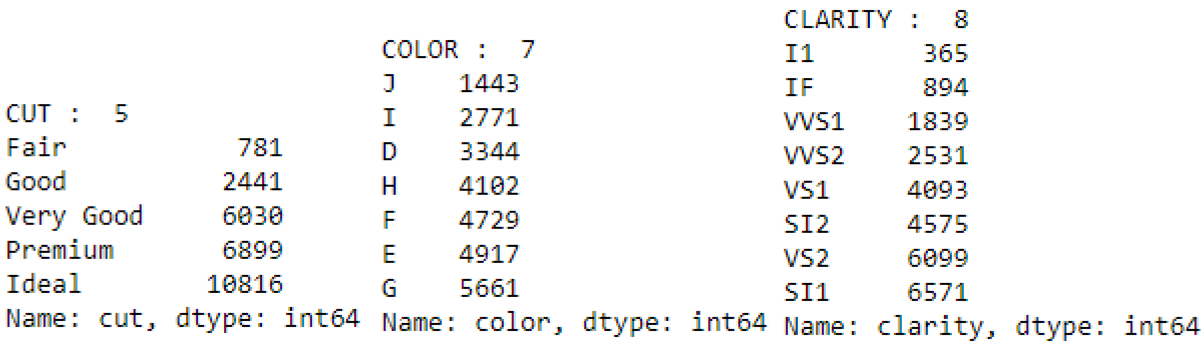
(Fig:-3 Heat Map)

Observation

Strong positive correlation between

- cart & price,
- depth & z cut,
- table & y cut
- x cut & x table
- z cut & table

Let’s check the unique values for the object data types - CUT, COLOR and CLARITY



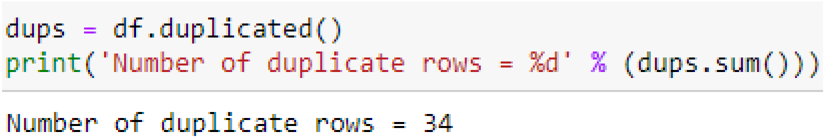
```
CUT :  5
Fair            781
Good           2441
Very Good      6030
Premium        6899
Ideal         10816
Name: cut, dtype: int64
```

```
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
```

(Fig:-4 Unique value)

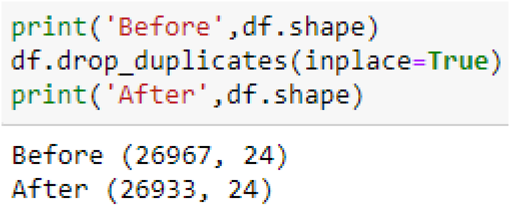
There seems to be 34 duplicates in the dataset.



```
dups = df.duplicated()
print('Number of duplicate rows = %d' % (dups.sum()))
```

```
Number of duplicate rows = 34
```

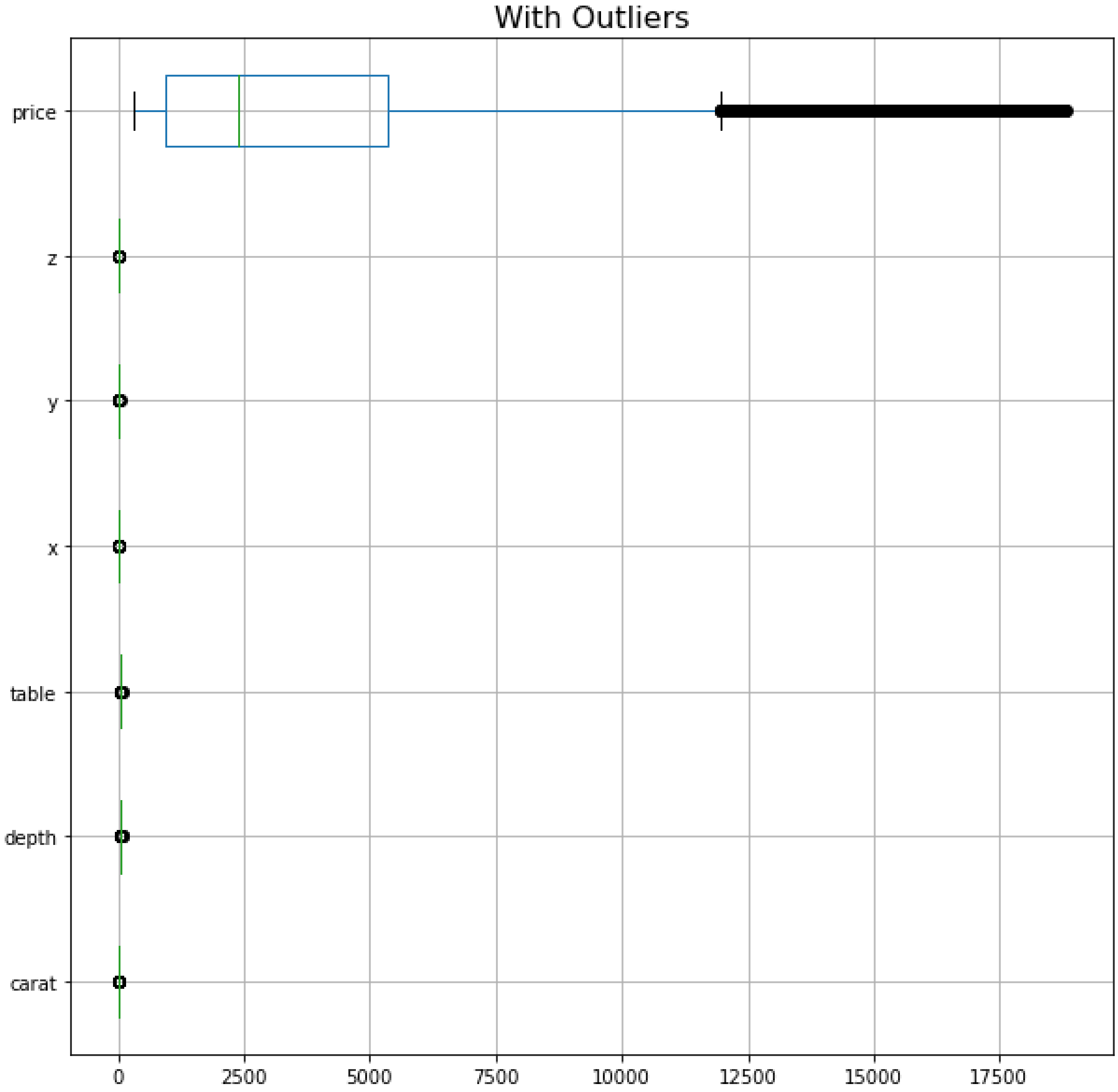
Let's drop them for now. The dataset before and after removing duplicate rows is as shown below.



```
print('Before',df.shape)
df.drop_duplicates(inplace=True)
print('After',df.shape)
```

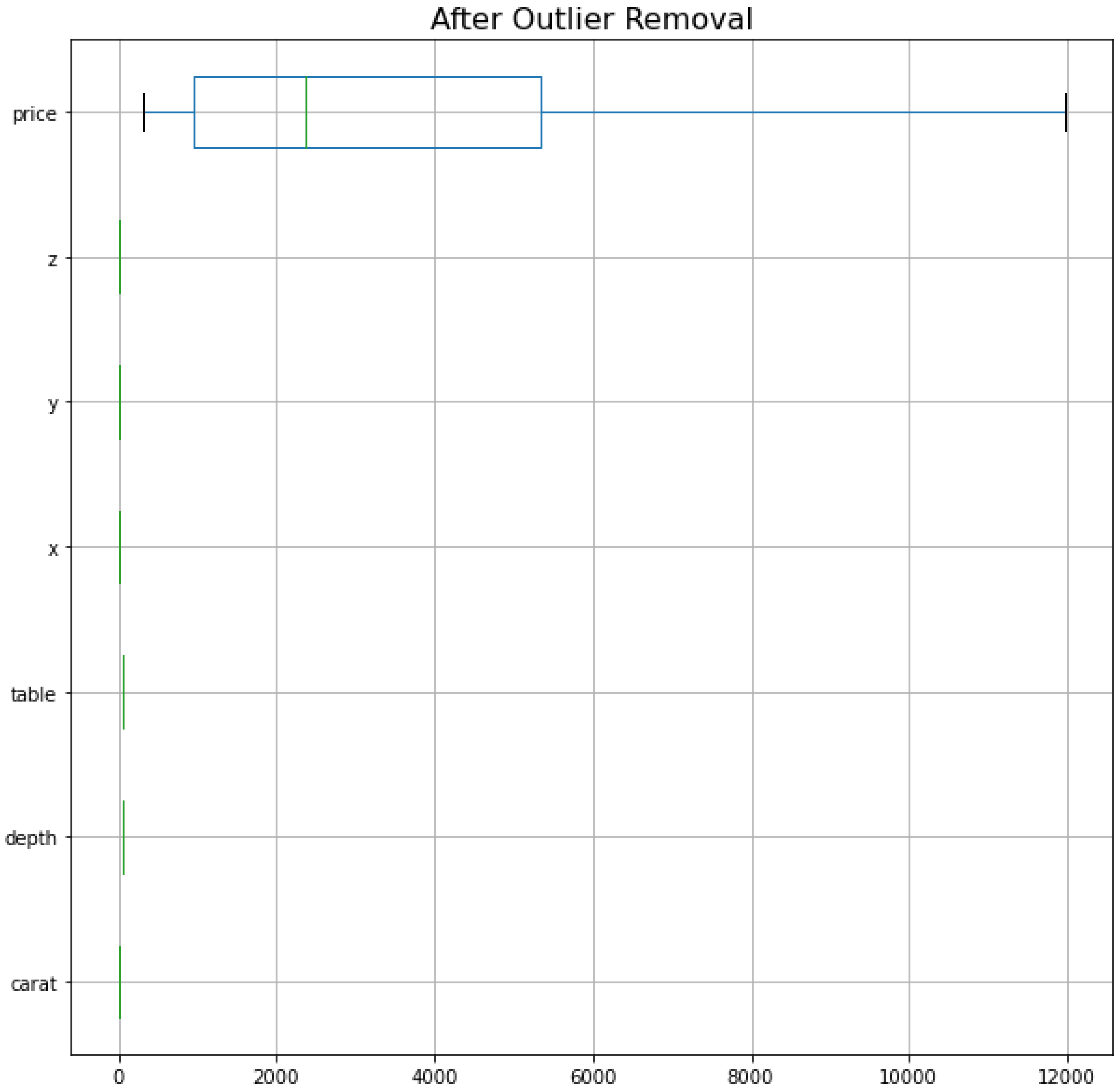
```
Before (26967, 24)
After (26933, 24)
```

We observed that there are some outliers in the dataset. Upon checking them using box plots, we found that there are outliers in almost every attribute. The box plot is given below:-



(Fig:-5 Before Outlier Treatment)

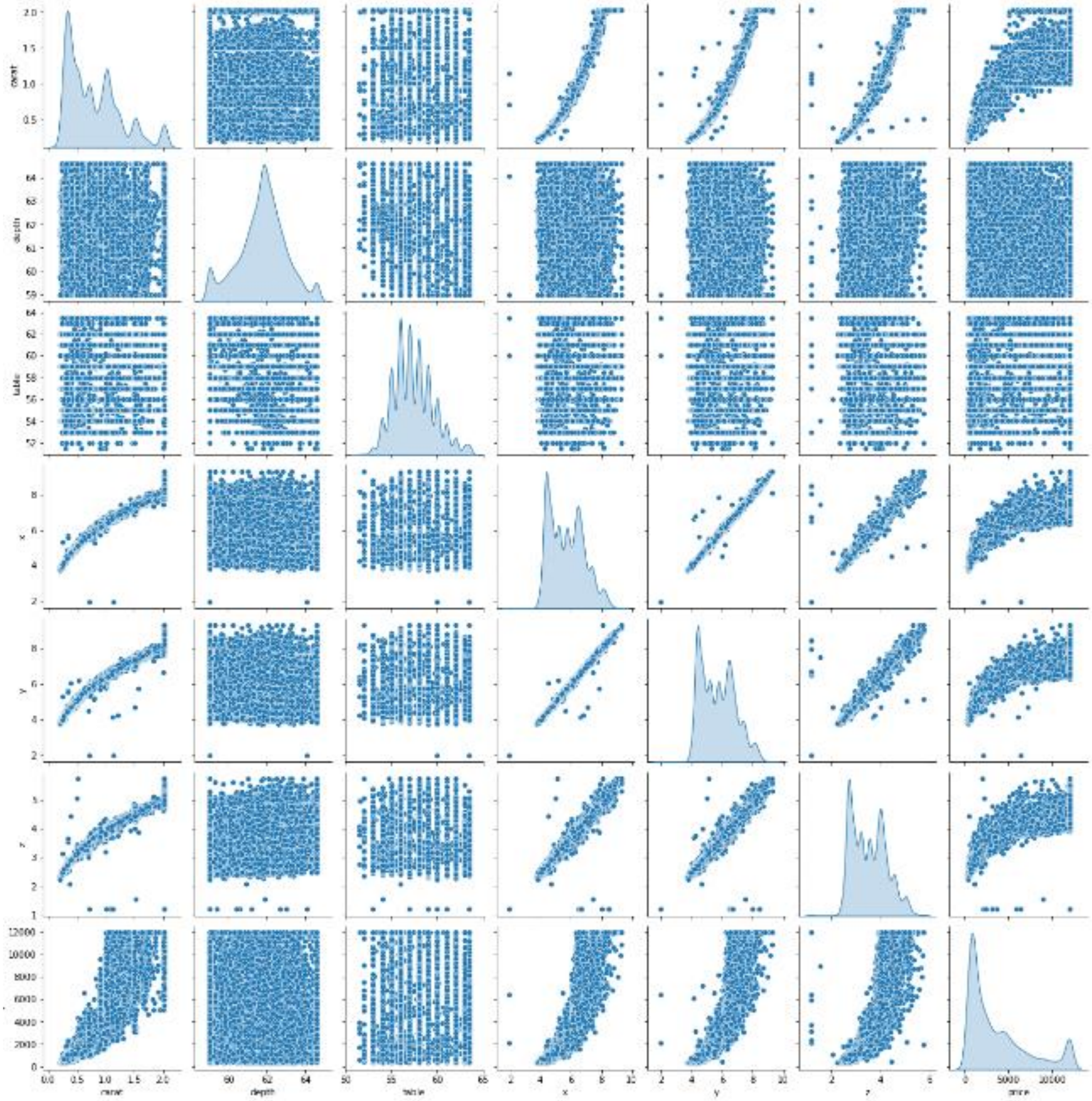
The box plot after removing outliers is as given below:-



(Fig:-6 After Outlier Treatment)

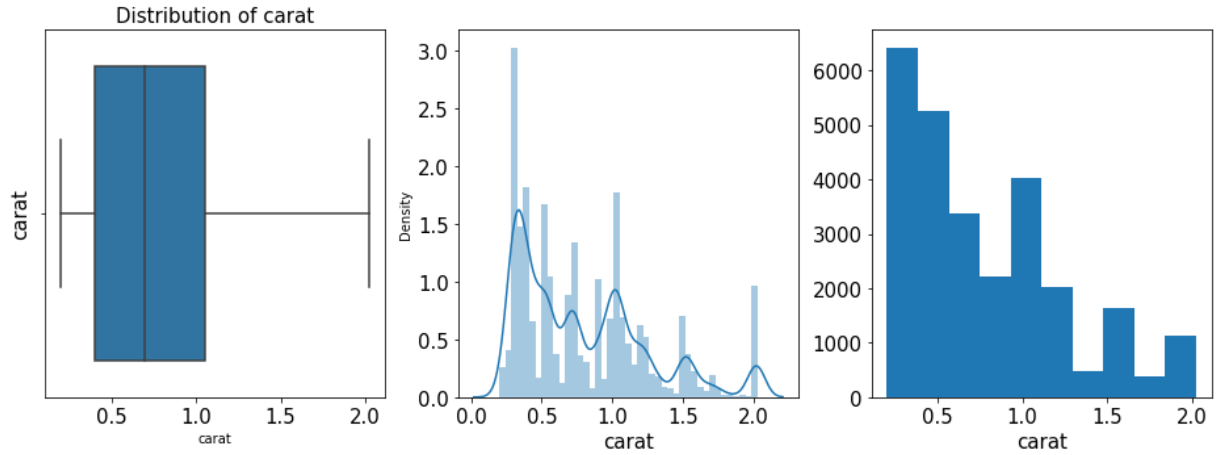
By checking the pairplot that's given below, there seems to be some clusters present. We observed that there is some negative relationship between depth and height, width and length of cubic zirconia. Carat, length, width, height and price are positively skewed.

MULTIVARIATE



(Fig:-7 Pair Plot)

UNIVARIET/BIVARIOT



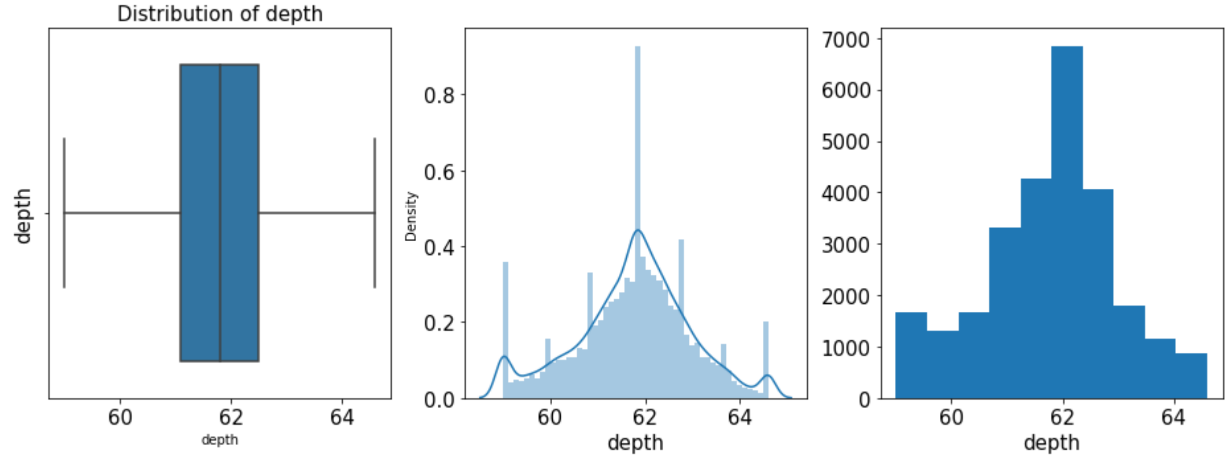
(Fig:-8 Cart plot Distribution)

The box plot of the Cart variable shows outliers.

Cart is positively skewed - 1.65421

The dist plot shows the distribution of data from 0.5 to 2.0

In the range of 0.5 to 2.0 is where the majority of the distribution lies.



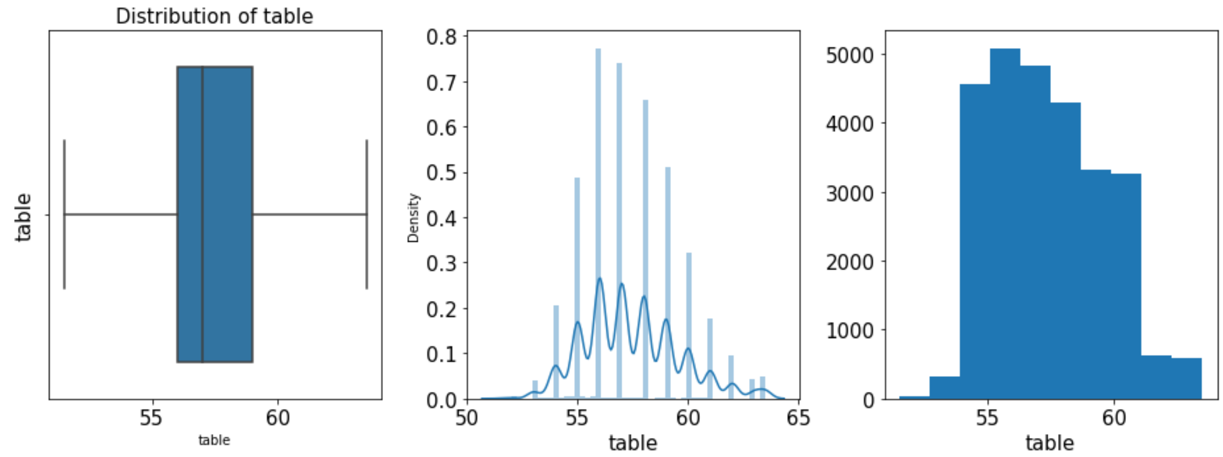
(Fig:-9 Depth Plot Distribution)

The box plot of the Cart variable shows outliers.

Cart is nutral skewed – 0.45

The dist plot shows the distribution of data from 50 to 65

In the range of 50 to 65 is where the majority of the distribution lies.



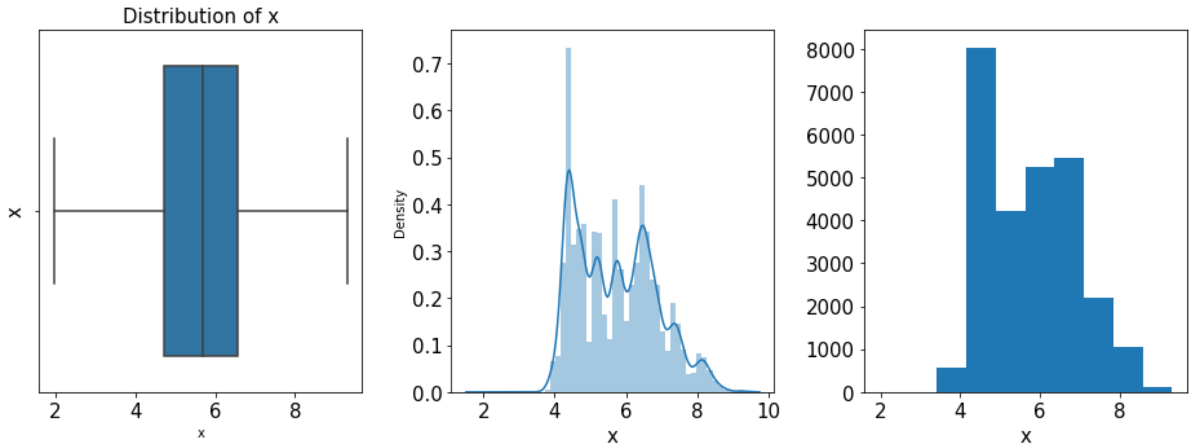
(Fig:-10 Table Distribution)

The box plot of the Cart variable shows outliers.

Cart is normal distributed skewed – 0.24548

The dist plot shows the distribution of data from 50 to 65

In the range of 0 to 69 is where the majority of the distribution lies.



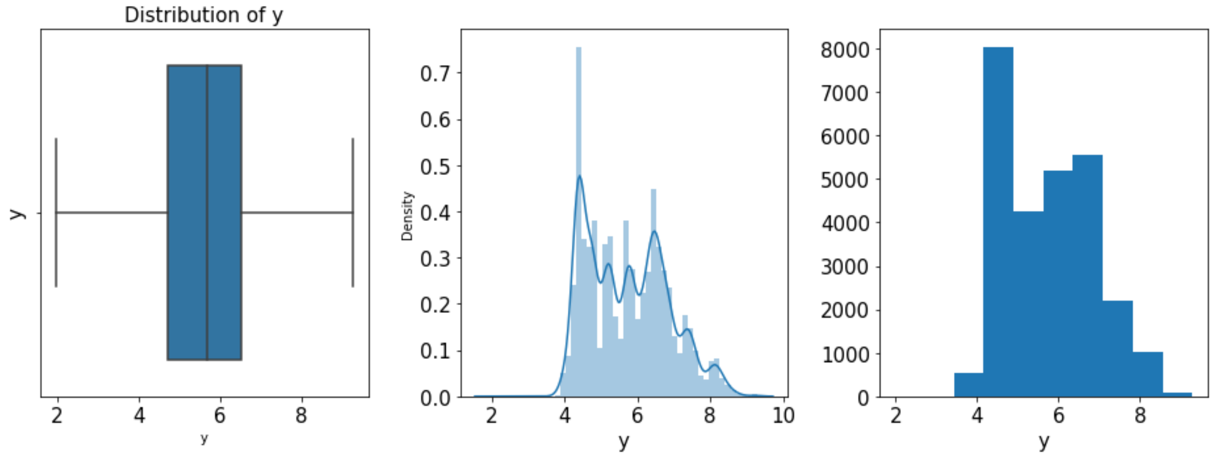
(Fig:-11 X Plot Distribution)

The box plot of the X variable shows outliers.

Cart is Negative skewed - 1.149713

The dist plot shows the distribution of data from 3.8 to 9.756

In the range of 3.5 to 8.78 is where the majority of the distribution lies.



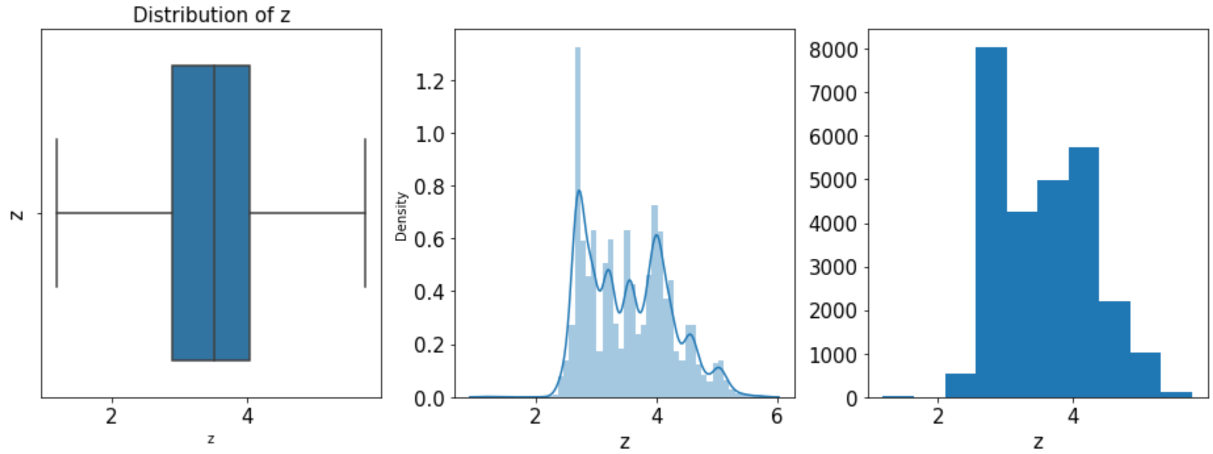
(Fig:-12 Y Plot Distribution)

The box plot of the Y variable shows outliers.

Cart is positively skewed – 0.5933

The dist plot shows the distribution of data from 4.0 to 9.55

In the range of 3.75 to 9 is where the majority of the distribution lies.



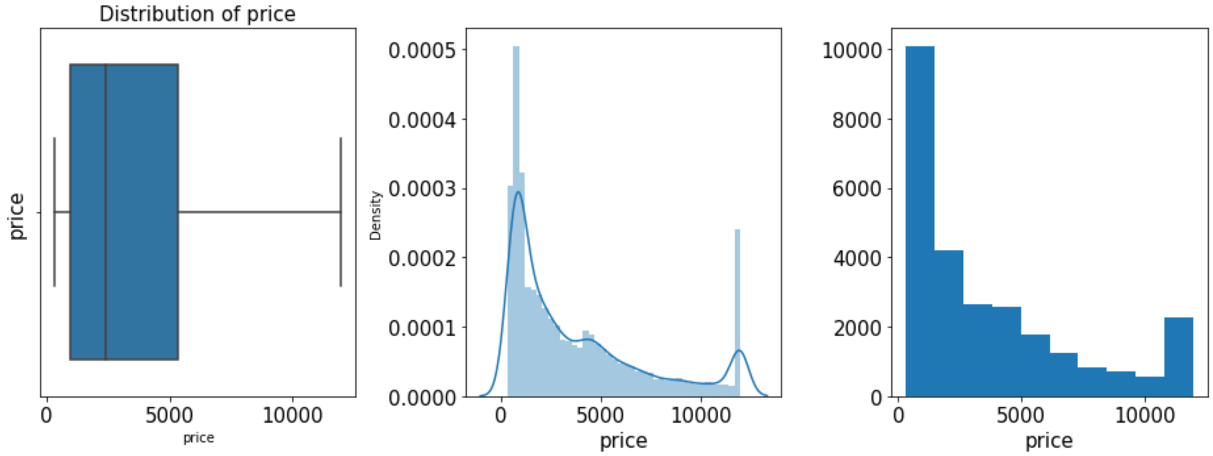
(Fig:-13 Z Plot Distribution)

The box plot of the Z variable shows outliers.

Cart is Negative skewed – 1.5713

The dist plot shows the distribution of data from 2.68 to 5.95

In the range of 2.3 to 6.65 is where the majority of the distribution lies.



(Fig:-14 Price Distribution)

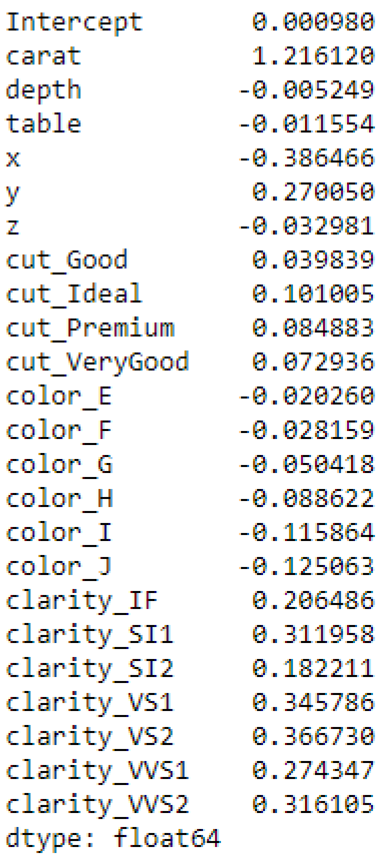
The box plot of the Price variable shows outliers.

Cart is positively skewed – 0.0005

The dist plot shows the distribution of data from 1 to 10000

In the range of 1.0 to 15000 is where the majority of the distribution lies.

After checking the skewness by using the skew() function, we can see that the depth attribute is negatively skewed. The screenshot of the result is shown below:-



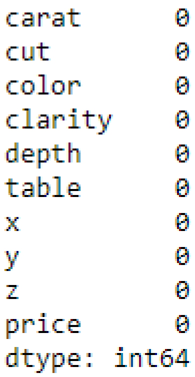
```
Intercept        0.000980
carat            1.216120
depth           -0.005249
table           -0.011554
x               -0.386466
y                0.270050
z               -0.032981
cut_Good         0.039839
cut_Ideal        0.101005
cut_Premium      0.084883
cut_VeryGood     0.072936
color_E         -0.020260
color_F         -0.028159
color_G         -0.050418
color_H         -0.088622
color_I         -0.115864
color_J         -0.125063
clarity_IF       0.206486
clarity_SI1      0.311958
clarity_SI2      0.182211
clarity_VS1      0.345786
clarity_VS2      0.366730
clarity_VVS1     0.274347
clarity_VVS2     0.316105
dtype: float64
```

(Fig:-15 Skewed results)

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?

Solution:-

There are 697 null values present in the depth column. Let's replace them with median value.



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

(Fig:-16 Null value replace with median)

From the pair plot and heat map that’s shown in problem 1.1 above, we can see that the attribute depth is least correlated with all other attributes, we can remove it or we can replace the null values with median values. The null values are replaced with its median value.

Scaling is necessary because by checking the summary, we can say that min, max and mean are not equal for all the columns. Data is scaled using preprocessing from sklearn library.

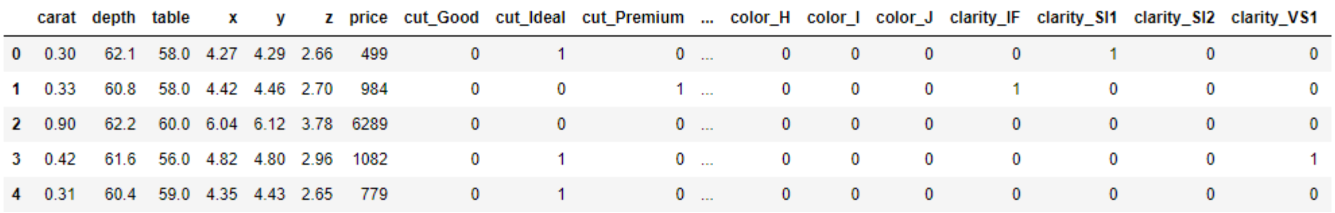
1.3 Encode the data (having string values) for Modelling. Data Split: Split the data into test and train (70:30). Apply linear regression. Performance Metrics: Check the performance of Predictions on Train and Test sets using R-Square, RMSE.

Solution:-

Let's change the object data types by encoding.

One-Hot-Encoding is used to create dummy variables to replace the categories in a categorical variable into features of each category and represent it using 1 or 0 based on the presence or absence of the categorical value in the record.

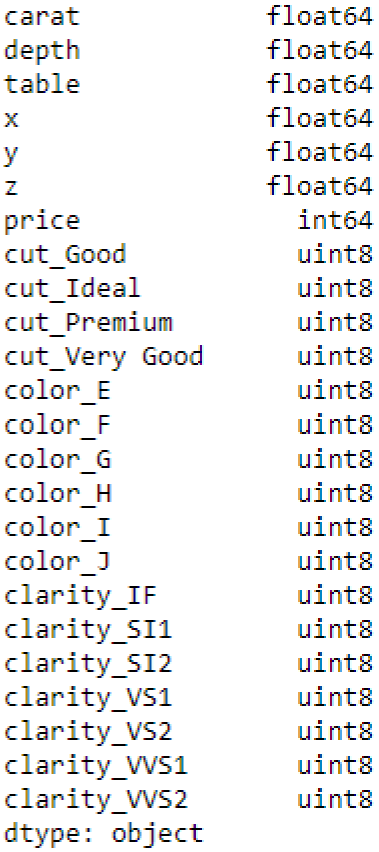
The new dataset is as shown below.



	carat	depth	table	x	y	z	price	cut_Good	cut_Ideal	cut_Premium	...	color_H	color_I	color_J	clarity_IF	clarity_SI1	clarity_SI2	clarity_VS1
0	0.30	62.1	58.0	4.27	4.29	2.66	499	0	1	0	...	0	0	0	0	1	0	0
1	0.33	60.8	58.0	4.42	4.46	2.70	984	0	0	1	...	0	0	0	1	0	0	0
2	0.90	62.2	60.0	6.04	6.12	3.78	6289	0	0	0	...	0	0	0	0	0	0	0
3	0.42	61.6	56.0	4.82	4.80	2.96	1082	0	1	0	...	0	0	0	0	0	0	1
4	0.31	60.4	59.0	4.35	4.43	2.65	779	0	1	0	...	0	0	0	0	0	0	0

(Table:-3 New Data)

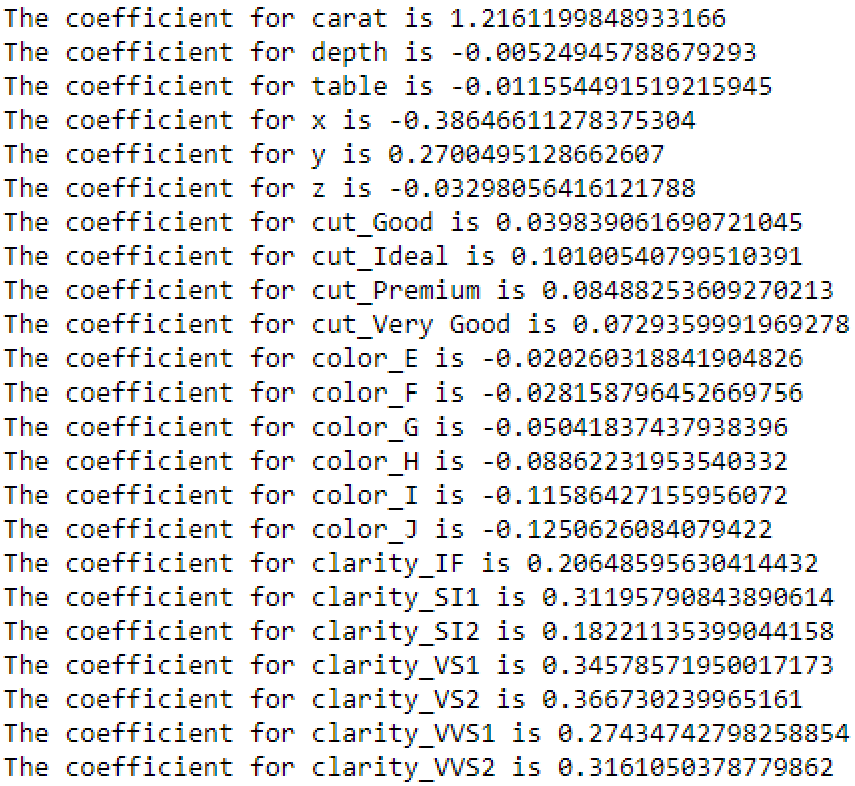
The datatypes of the new dataset are as shown below:-



```
carat            float64
depth            float64
table            float64
x                float64
y                float64
z                float64
price              int64
cut_Good           uint8
cut_Ideal          uint8
cut_Premium        uint8
cut_Very Good      uint8
color_E            uint8
color_F            uint8
color_G            uint8
color_H            uint8
color_I            uint8
color_J            uint8
clarity_IF         uint8
clarity_SI1        uint8
clarity_SI2        uint8
clarity_VS1        uint8
clarity_VS2        uint8
clarity_VVS1       uint8
clarity_VVS2       uint8
dtype: object
```

(Fig:-17 New Data Types)

After splitting data, the coefficients for each of the independent attributes are



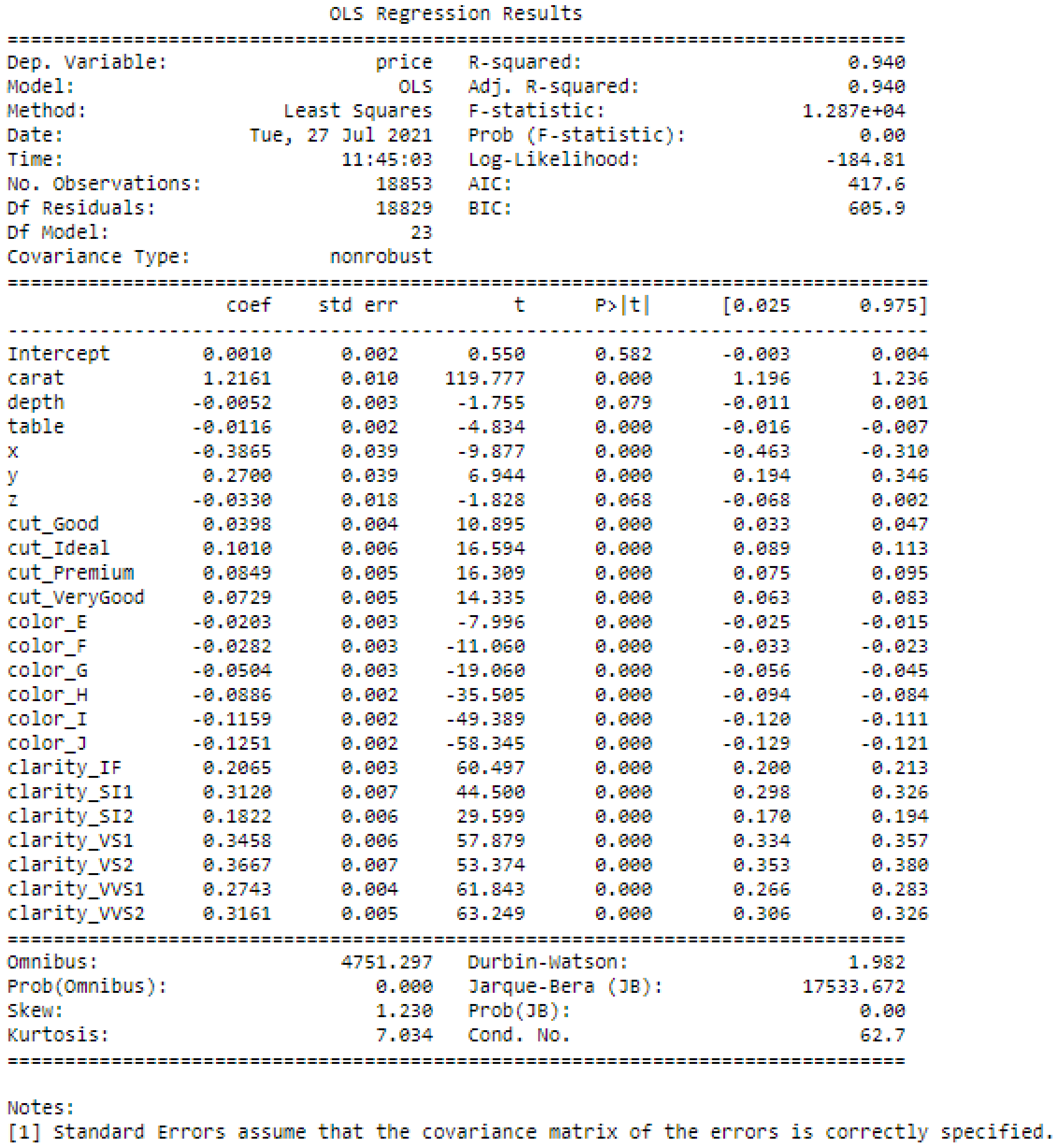
```
The coefficient for carat is 1.2161199848933166
The coefficient for depth is -0.00524945788679293
The coefficient for table is -0.011554491519215945
The coefficient for x is -0.38646611278375304
The coefficient for y is 0.2700495128662607
The coefficient for z is -0.03298056416121788
The coefficient for cut_Good is 0.039839061690721045
The coefficient for cut_Ideal is 0.10100540799510391
The coefficient for cut_Premium is 0.08488253609270213
The coefficient for cut_Very Good is 0.0729359991969278
The coefficient for color_E is -0.020260318841904826
The coefficient for color_F is -0.028158796452669756
The coefficient for color_G is -0.05041837437938396
The coefficient for color_H is -0.08862231953540332
The coefficient for color_I is -0.11586427155956072
The coefficient for color_J is -0.1250626084079422
The coefficient for clarity_IF is 0.20648595630414432
The coefficient for clarity_SI1 is 0.31195790843890614
The coefficient for clarity_SI2 is 0.18221135399044158
The coefficient for clarity_VS1 is 0.34578571950017173
The coefficient for clarity_VS2 is 0.366730239965161
The coefficient for clarity_VVS1 is 0.27434742798258854
The coefficient for clarity_VVS2 is 0.3161050378779862
```

(Fig:-18 Attributes Of New Data)

It looks like depth, table, length and width are negatively correlated. But, it shows positive relations. It seems to be that multicollinearity exists.
Let us check the intercept for the model.

- The intercept for our model = 0.0009803459694918194
- R-Square on training data = 0.9402044588687953
- R-Square on testing data = 0.9419074345242372
- RMSE on training data = 0.24435440092961688
- RMSE on testing data = 0.24143024829860243

We got the same results by using stats model

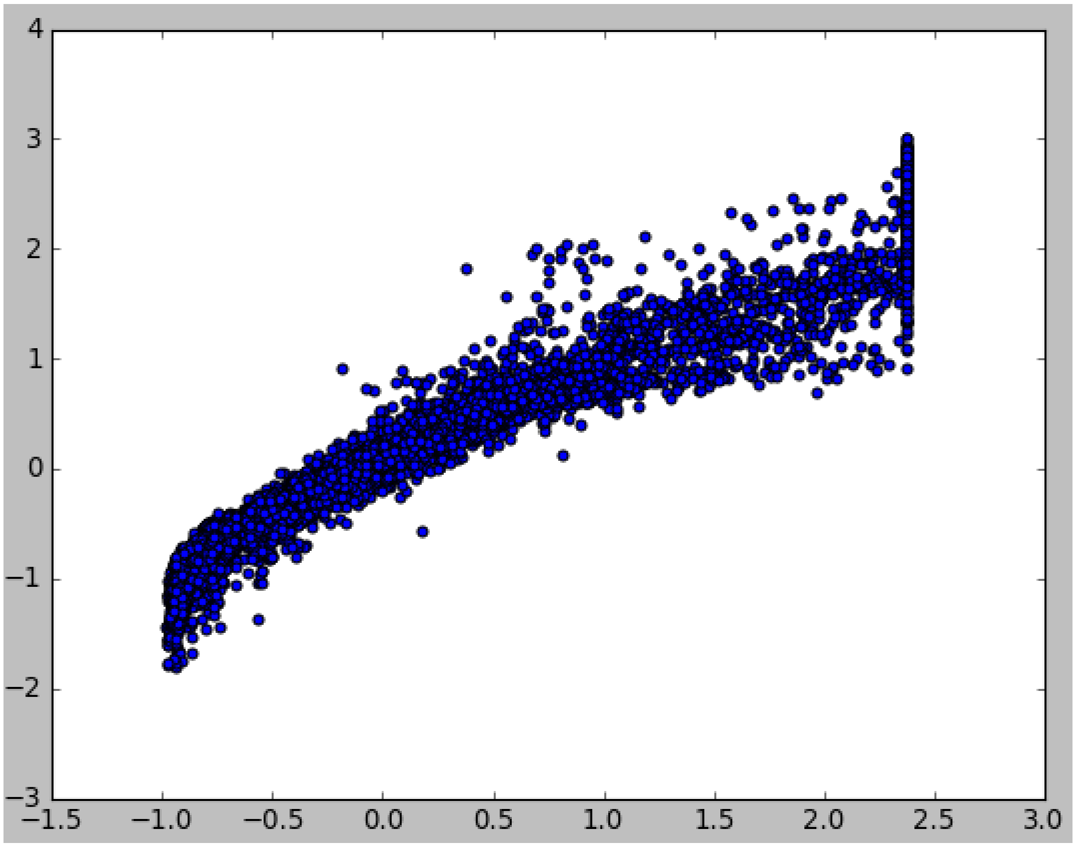


```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.940
Model:                            OLS   Adj. R-squared:                  0.940
Method:                 Least Squares   F-statistic:                 1.287e+04
Date:                Tue, 27 Jul 2021   Prob (F-statistic):               0.00
Time:                        11:45:03   Log-Likelihood:                -184.81
No. Observations:               18853   AIC:                             417.6
Df Residuals:                   18829   BIC:                             605.9
Df Model:                          23
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept        0.0010      0.002      0.550      0.582      -0.003       0.004
carat            1.2161      0.010    119.777      0.000       1.196       1.236
depth           -0.0052      0.003     -1.755      0.079      -0.011       0.001
table           -0.0116      0.002     -4.834      0.000      -0.016      -0.007
x               -0.3865      0.039     -9.877      0.000      -0.463      -0.310
y                0.2700      0.039      6.944      0.000       0.194       0.346
z               -0.0330      0.018     -1.828      0.068      -0.068       0.002
cut_Good         0.0398      0.004     10.895      0.000       0.033       0.047
cut_Ideal        0.1010      0.006     16.594      0.000       0.089       0.113
cut_Premium      0.0849      0.005     16.309      0.000       0.075       0.095
cut_VeryGood     0.0729      0.005     14.335      0.000       0.063       0.083
color_E         -0.0203      0.003     -7.996      0.000      -0.025      -0.015
color_F         -0.0282      0.003    -11.060      0.000      -0.033      -0.023
color_G         -0.0504      0.003    -19.060      0.000      -0.056      -0.045
color_H         -0.0886      0.002    -35.505      0.000      -0.094      -0.084
color_I         -0.1159      0.002    -49.389      0.000      -0.120      -0.111
color_J         -0.1251      0.002    -58.345      0.000      -0.129      -0.121
clarity_IF       0.2065      0.003     60.497      0.000       0.200       0.213
clarity_SI1      0.3120      0.007     44.500      0.000       0.298       0.326
clarity_SI2      0.1822      0.006     29.599      0.000       0.170       0.194
clarity_VS1      0.3458      0.006     57.879      0.000       0.334       0.357
clarity_VS2      0.3667      0.007     53.374      0.000       0.353       0.380
clarity_VVS1     0.2743      0.004     61.843      0.000       0.266       0.283
clarity_VVS2     0.3161      0.005     63.249      0.000       0.306       0.326
==============================================================================
Omnibus:                     4751.297   Durbin-Watson:                   1.982
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            17533.672
Skew:                           1.230   Prob(JB):                         0.00
Kurtosis:                       7.034   Cond. No.                         62.7
==============================================================================

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

(Fig:-19 Regression Model)

- R-squared = 0.940
- Adj. R-squared = 0.940
- Root Mean Squared Error - RMSE = 0.24451008200785526

Scatter plot for the predicted test data is as given below:



(Fig:-20 Scatter Plot for Predicted Data set)

After scaling and applying linear regression, we got the same results as before.
The model seems to be overfat. Let's apply ridge and lasso regression techniques which are some of the simple techniques to reduce model complexity and prevent overfitting that may result from simple linear regression.

$$\sum_{i=1}^{M}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{p} w_j \times x_{ij}\right)^2$$

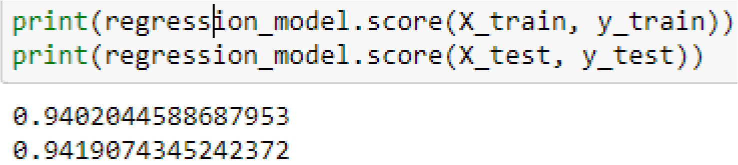
Cost function for simple linear model

Ridge regression minimizes the cost function. It puts constraint on the coefficients. The penalty term regularizes the coefficients such that if the coefficients take large values the optimization function is penalized. So, ridge regression shrinks the coefficients and it helps to reduce the model complexity and multi-collinearity.

$$\sum_{i=1}^{M}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{p} w_j \times x_{ij}\right)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

Cost function for ridge regression

Higher the alpha value, more restriction on the coefficients; low alpha > more generalization
The coefficients for ridge model are
The train and test scores for Ridge and linear regression models are almost same



```
print(regression_model.score(X_train, y_train))
print(regression_model.score(X_test, y_test))
```

```
0.9402044588687953
0.9419074345242372
```

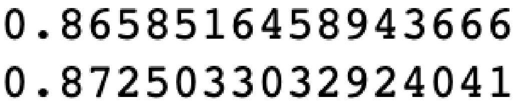
Let’s apply lasso regression technique.
The cost function for Lasso (least absolute shrinkage and selection operator) regression can be written as

$$\sum_{i=1}^{M}(y_i-\hat{y}_i)^2=\sum_{i=1}^{M}\left(y_i-\sum_{j=0}^{p}w_j\times x_{ij}\right)^2+\lambda\sum_{j=0}^{p}|w_j|$$

Cost function for Lasso regression

The only difference between ridge and lasso regression is instead of taking the square of the coefficients, magnitudes are taken into account. This type of regularization(L1) can lead to zero coefficients i.e. some of the features are completely neglected for the evaluation of output. Lasso regression not only helps in reducing overfitting but it can help us in feature selection.
The coefficients of lasso model are

We Observed that, many of the coefficients have become 0 indicating drop of those dimensions from the model
The train and test score for lasso model is



```
0.8658516458943666
0.872503303292404l
```

Further, the number of dimensions is much less in LASSO model than ridge or un-regularized model

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

Solution:-

- Linear regression models are over fit for this dataset.
- After regularizing using lasso regression technique, we can say that carat, weight, length, width and height of the cubic zirconia are the most important attributes in deciding the good stones that give more profits.

Depth attribute is least important and also lowest profitable if decided based on depth.

Problem 2:- Holiday Package (Logistic Regression and LDA)

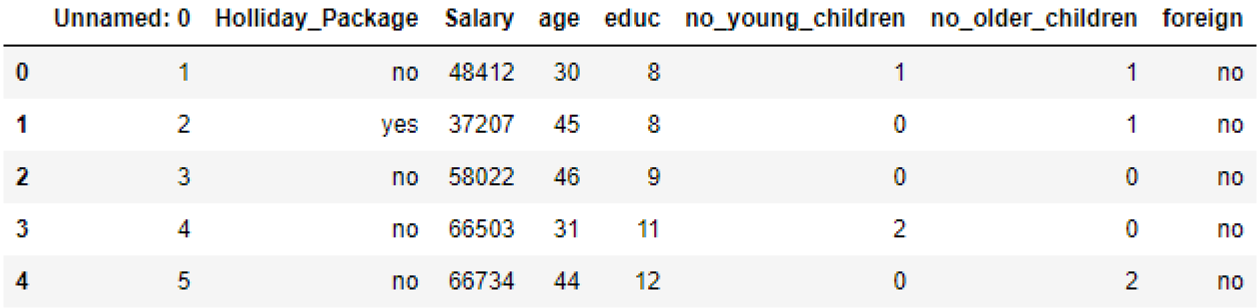
2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, write an inference on it. Perform Univariate and Bivariate Analysis. Do exploratory data analysis.

Solution:-

Dataset has 872 observations and 8 attributes.

- 6 attributes named unnamed: 0, salary, age, educ, no_young_children, no_older_children are of integer_type.
- 2 attributes named holliday_package, foriegn are of object type.

Let's start the data exploration step with the head function to look at the first 5 rows.

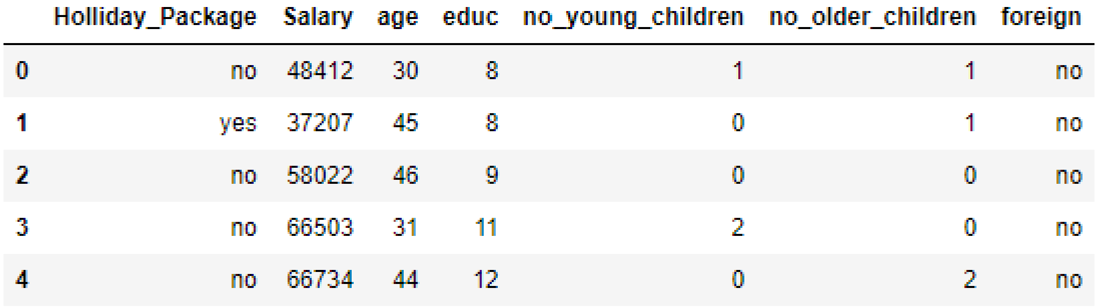


	Unnamed: 0	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
0	1	no	48412	30	8	1	1	no
1	2	yes	37207	45	8	0	1	no
2	3	no	58022	46	9	0	0	no
3	4	no	66503	31	11	2	0	no
4	5	no	66734	44	12	0	2	no

(Table: - 1 Data head)

Unnamed: 0 seems to be of no use for our dataset and it's dropped due to that.

Dataset after dropping the first column

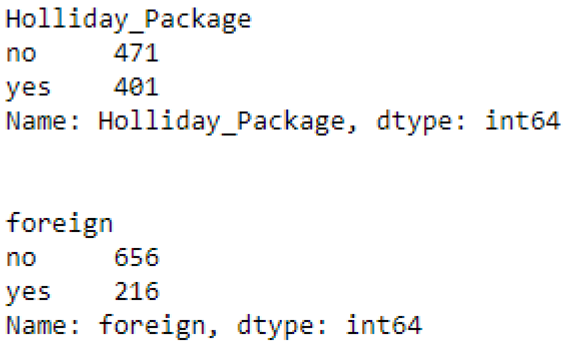


	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
0	no	48412	30	8	1	1	no
1	yes	37207	45	8	0	1	no
2	no	58022	46	9	0	0	no
3	no	66503	31	11	2	0	no
4	no	66734	44	12	0	2	no

(Table: - 2 After Drop Raw Number)

Now the shape of the dataset is (872, 7). There are 401 who opted for a Holliday_Package and 471 who did not opt for it. The proportion of yes and no are almost the same.

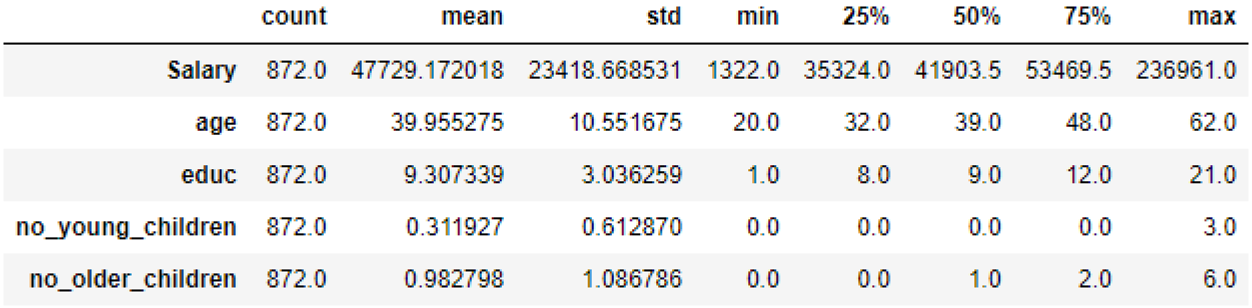
The proportion 1’s and 0’s for object data types are as shown below:-



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


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

(Fig:-1 Data Objects)

After checking the summary of the data, there seems to be some outliers in the dataset. Let’s check them by doing further exploration.

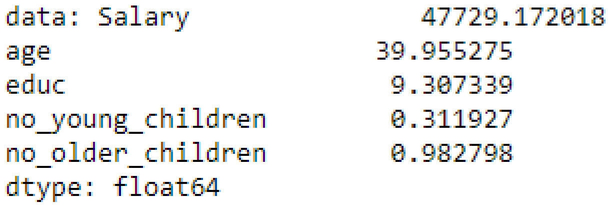


	count	mean	std	min	25%	50%	75%	max
Salary	872.0	47729.172018	23418.668531	1322.0	35324.0	41903.5	53469.5	236961.0
age	872.0	39.955275	10.551675	20.0	32.0	39.0	48.0	62.0
educ	872.0	9.307339	3.036259	1.0	8.0	9.0	12.0	21.0
no_young_children	872.0	0.311927	0.612870	0.0	0.0	0.0	0.0	3.0
no_older_children	872.0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

(Table:-3 Describe Data)

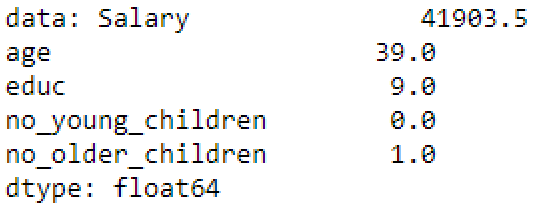
By doing descriptive statistics, we observed that mean and median are almost the same for all attributes.

Data Mean



```
data: Salary              47729.172018
age                    39.955275
educ                    9.307339
no_young_children       0.311927
no_older_children       0.982798
dtype: float64
```

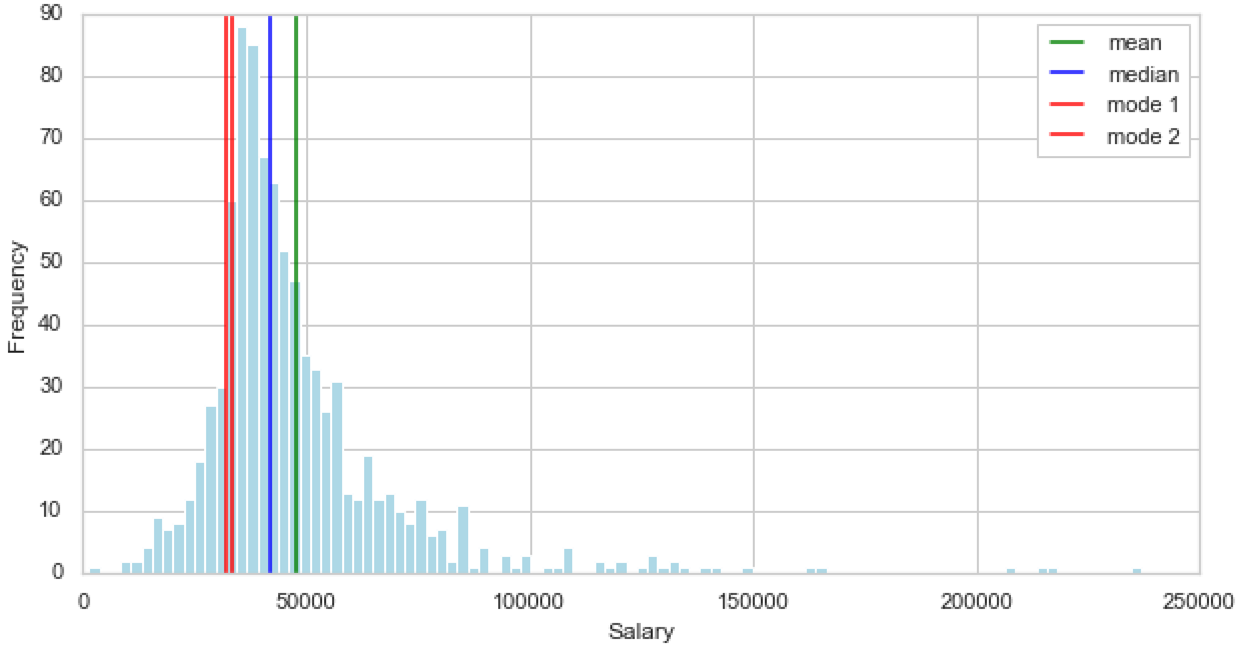
Data Median



```
data: Salary              41903.5
age                    39.0
educ                    9.0
no_young_children       0.0
no_older_children       1.0
dtype: float64
```

(Fig:-2 Descriptive Statistics)

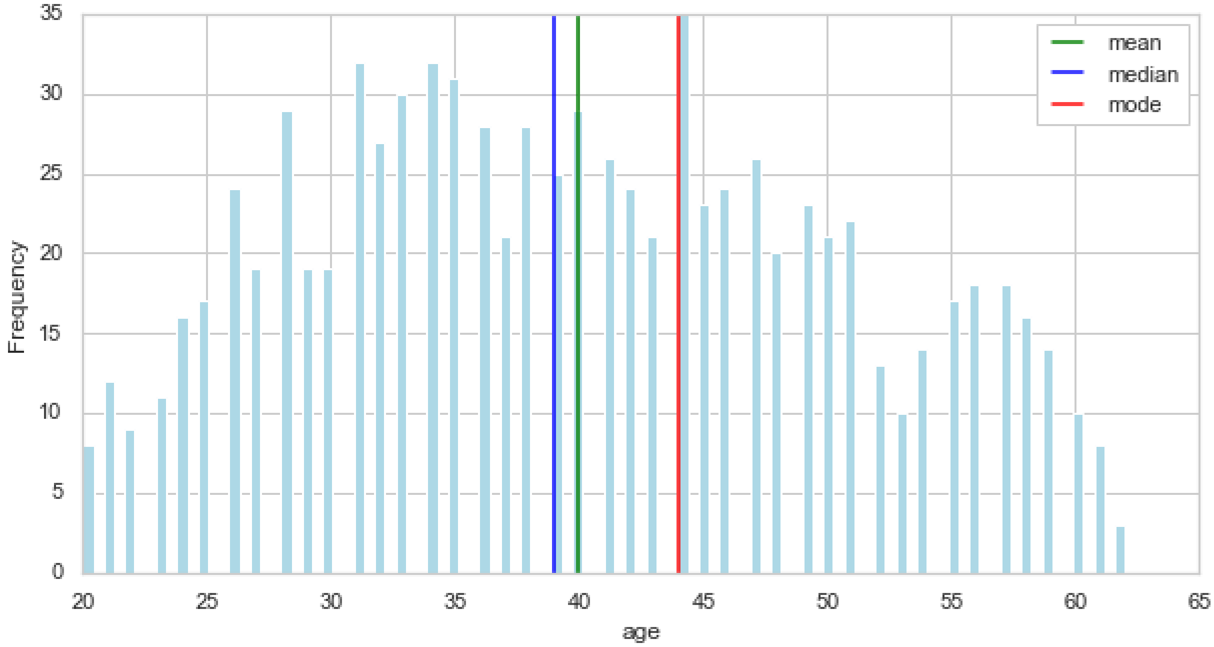
From the distribution of all attributes given in the below picture, we can see that the distribution for salary attribute seems to be positive.



(Fig:-3 Salary Frequency Distribution)

Most frequent values are low and tail is towards high values

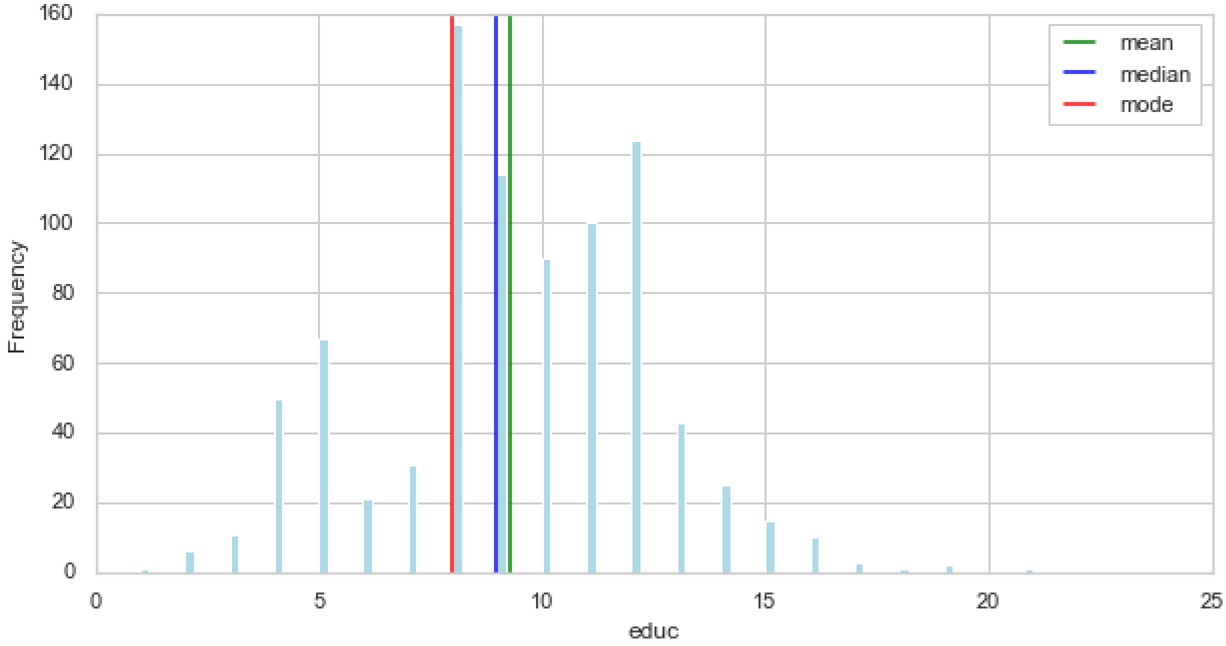
The distribution for age attribute seems to be uneven. Mean and median are almost the same.



(Fig:-4 Age Frequency Distribution)

As shown in Fig the age frequency Graph is normally distributed.

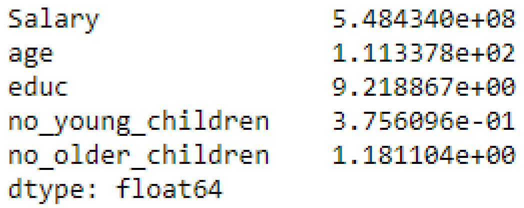
The distribution for educ attribute seems to be positive.



(Fig:-5 Educ Frequency Distribution)

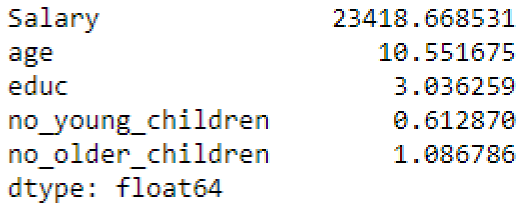
By checking the variance and standard deviation values, we can say that data is much deviated from the mean that means there may be some outliers for all the attributes.

Standard Variation



```
Salary                5.484340e+08
age                   1.113378e+02
educ                  9.218867e+00
no_young_children     3.756096e-01
no_older_children     1.181104e+00
dtype: float64
```

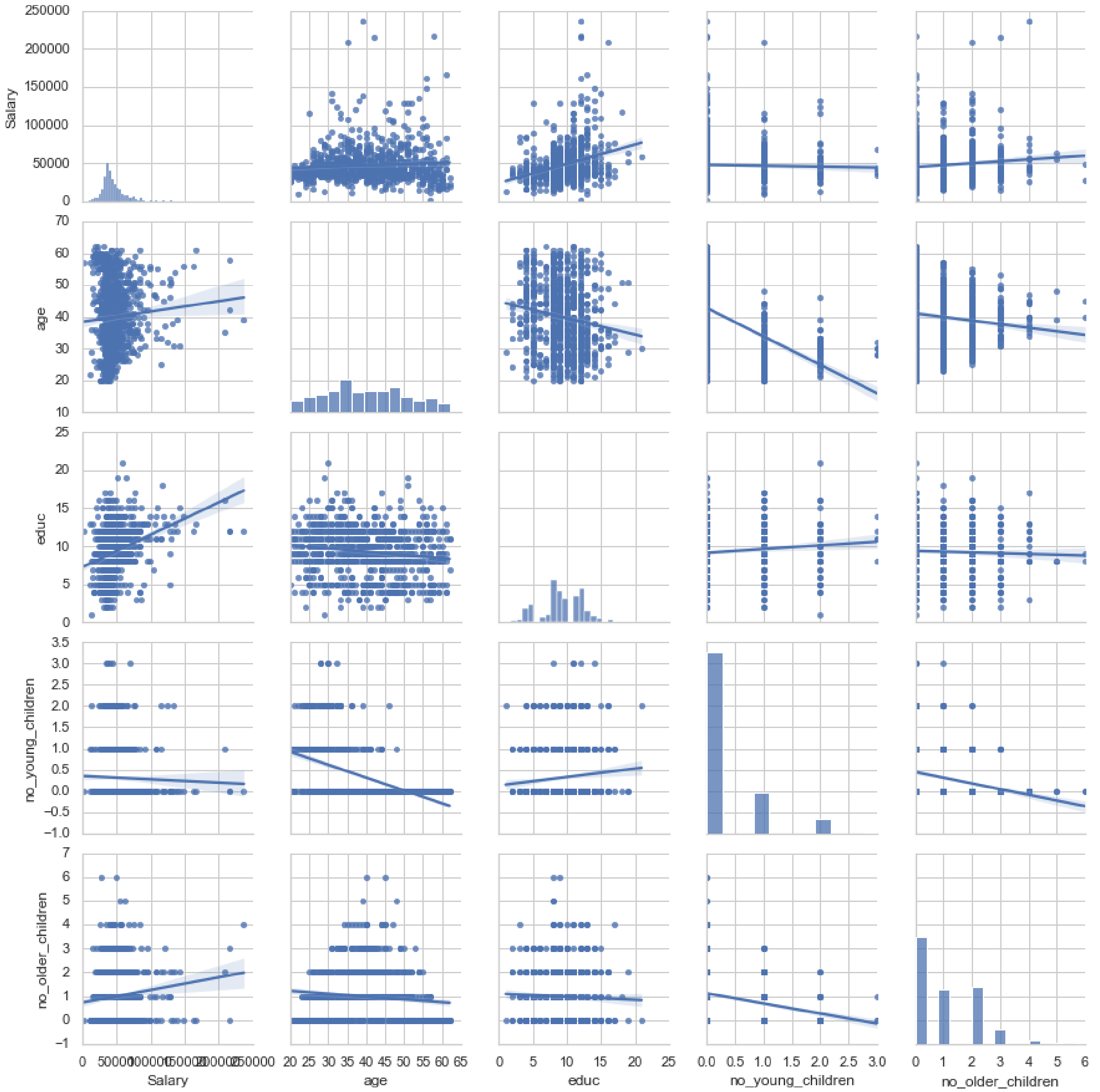
Standard Deviation



```
Salary                23418.668531
age                      10.551675
educ                      3.036259
no_young_children         0.612870
no_older_children         1.086786
dtype: float64
```

(Fig:-6 Standard deviation and variation)

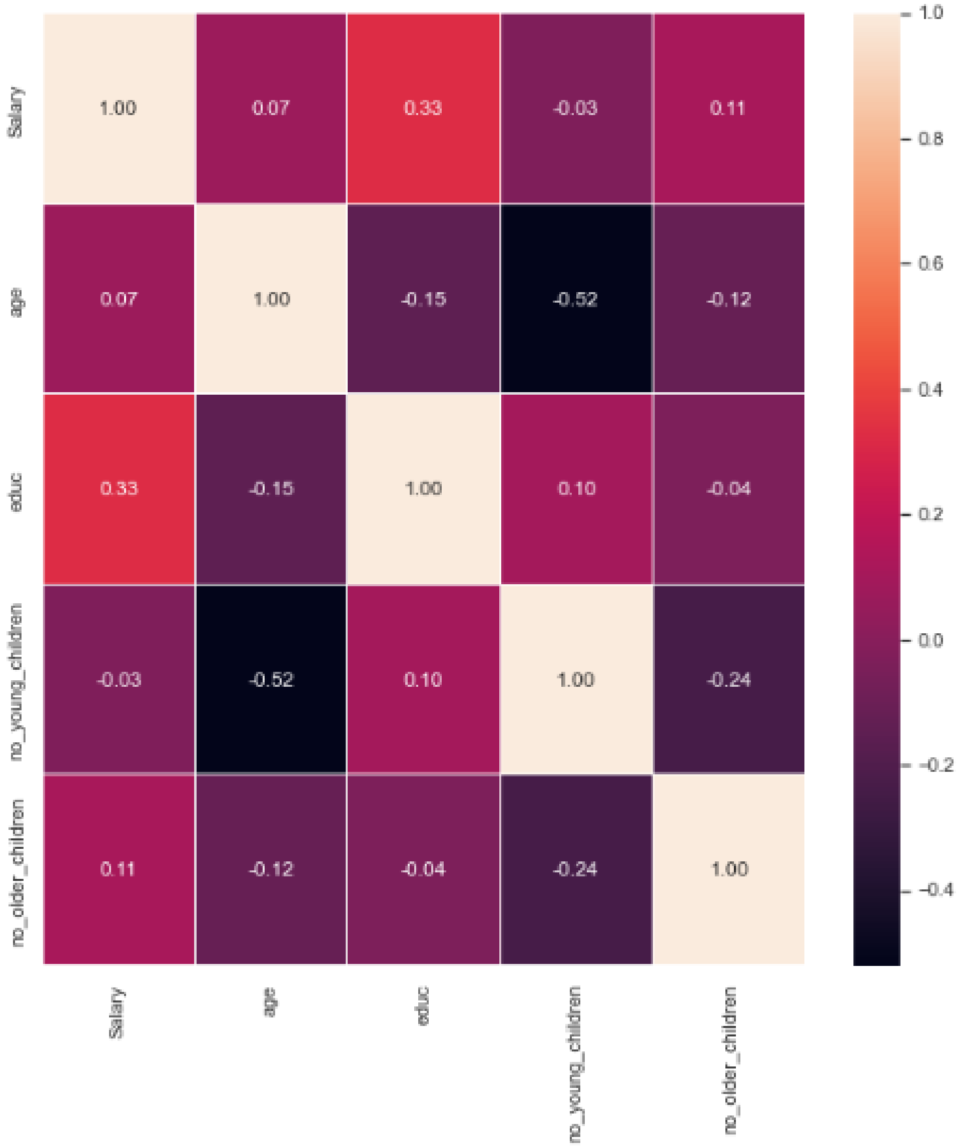
Let's check the pair plot to see the correlation between all the attributes.



(Fig:-7 Multivariate)

Here from the pair plot given above, we can say that the attributes are not that much correlated with each other.

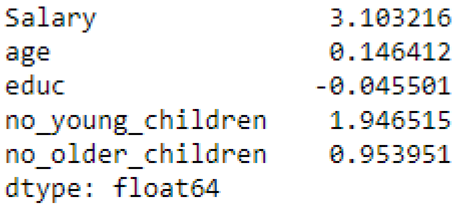
Let's check heat map given below for further analysis



(Fig:-8 Heat Map)

Employee salary and Years of formal education are somewhat correlated. But, there isn't much relation between other attributes.

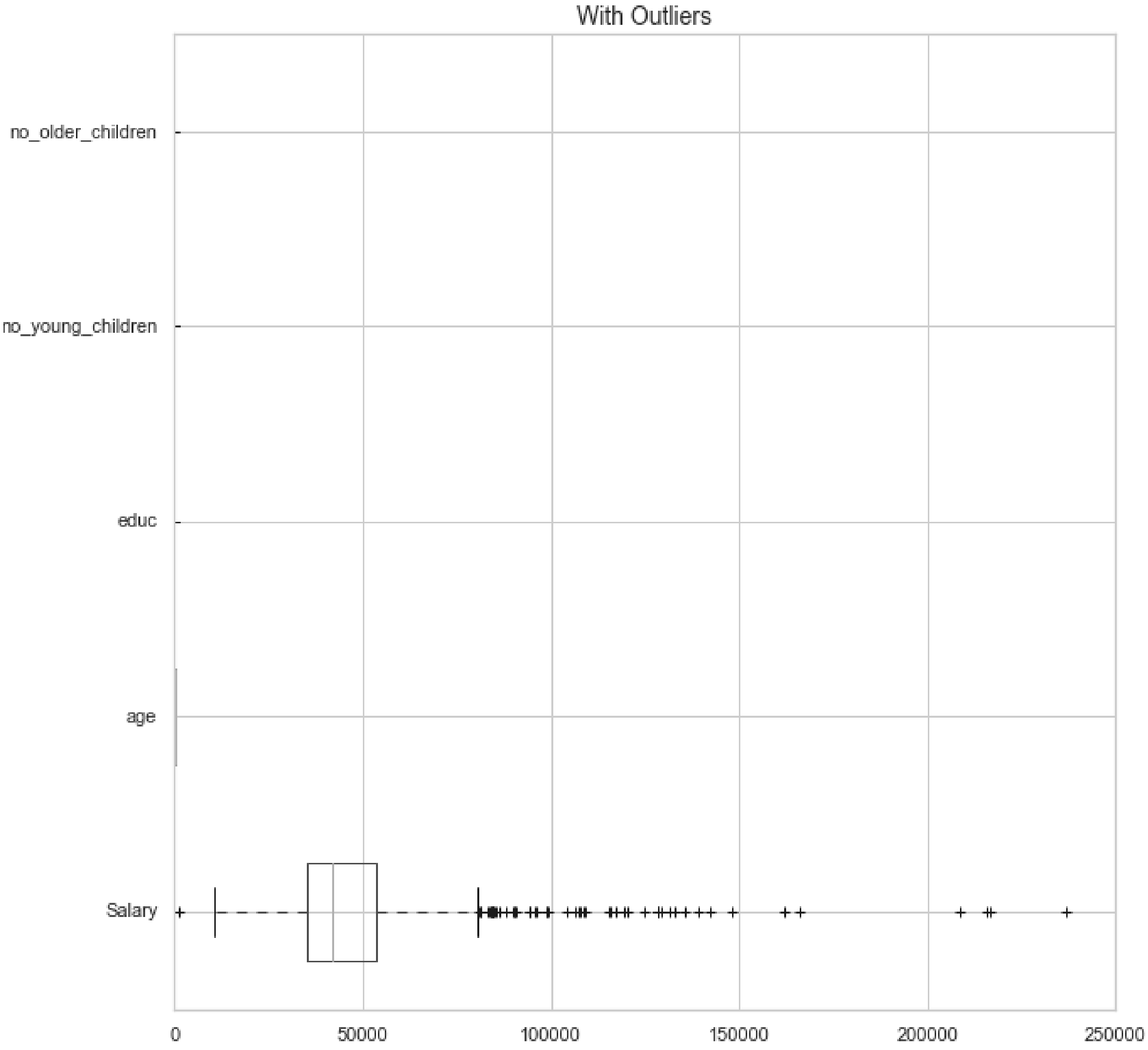
Let's check the skewness of every attribute that’s given in the picture below:-



```
Salary               3.103216
age                  0.146412
educ                -0.045501
no_young_children    1.946515
no_older_children    0.953951
dtype: float64
```

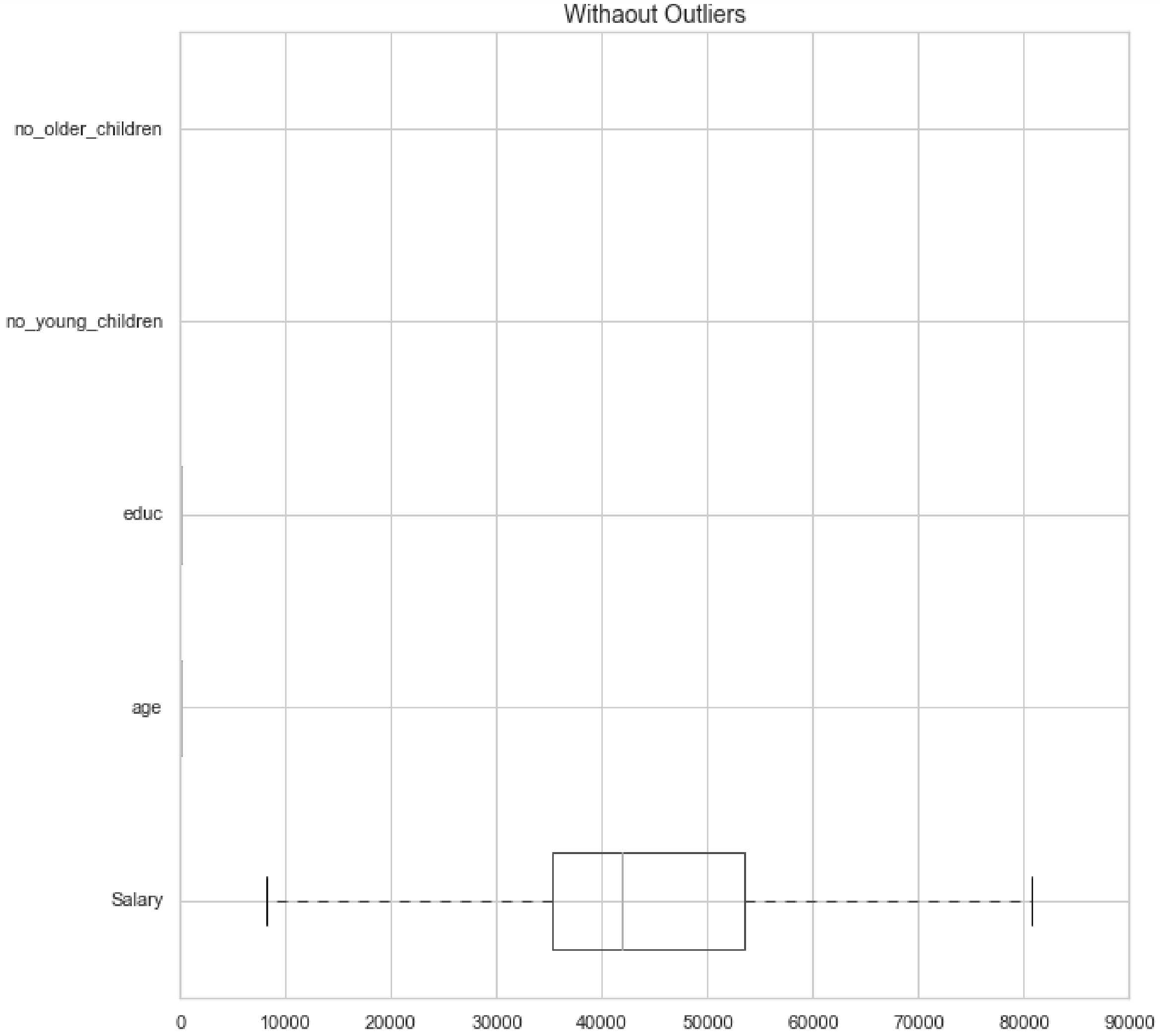
(Fig:-9 Measure Skewness)

Years of formal education is negatively skewed and rest of the attributes are positively skewed



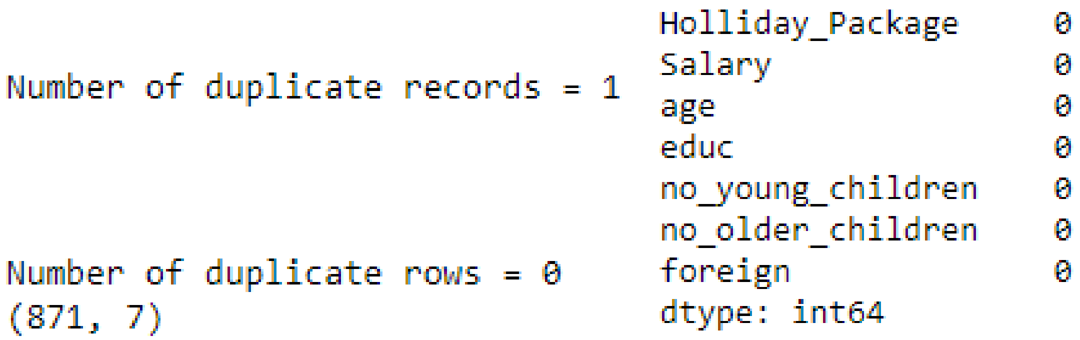
(Fig:-10 Positive skweds outlier)

In the box-plots given above, we can see that all the attributes except age have outliers. Let’s treat them by defining custom function and check them again



(Fig:-11 Positive skweds After Treatment)

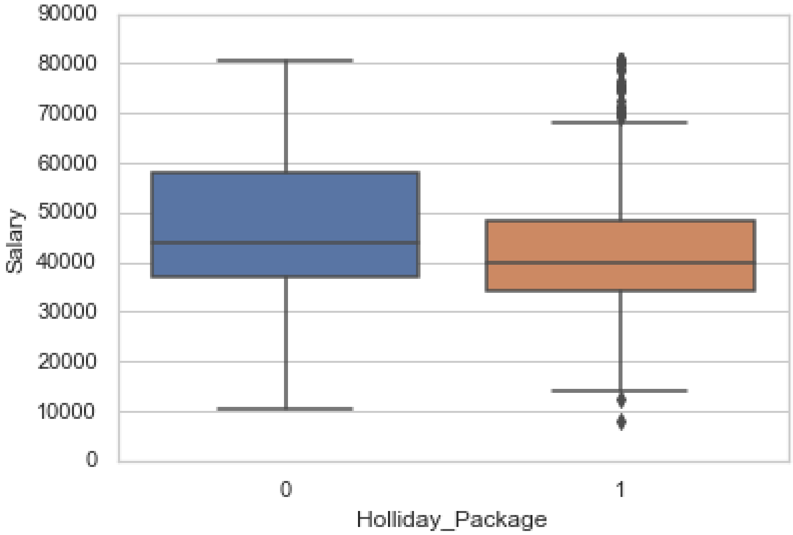
As seen in the above box-plots, there seems to be no null values. But, duplicates are in the dataset. Let’s remove them.



```
                                    Holliday_Package     0
                                    Salary               0
Number of duplicate records = 1     age                  0
                                    educ                 0
                                    no_young_children    0
                                    no_older_children    0
Number of duplicate rows = 0        foreign              0
(871, 7)                            dtype: int64
```

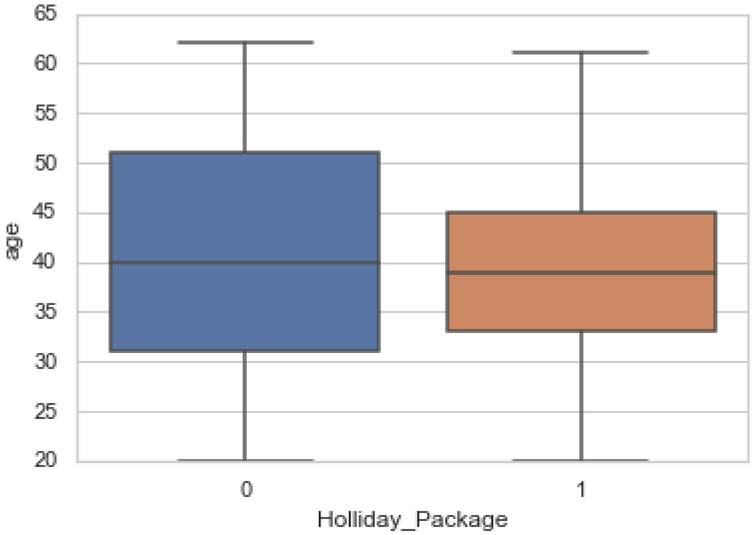
(Fig:-12 Duplicates)

Let's check the distribution of every attribute with the target variable.



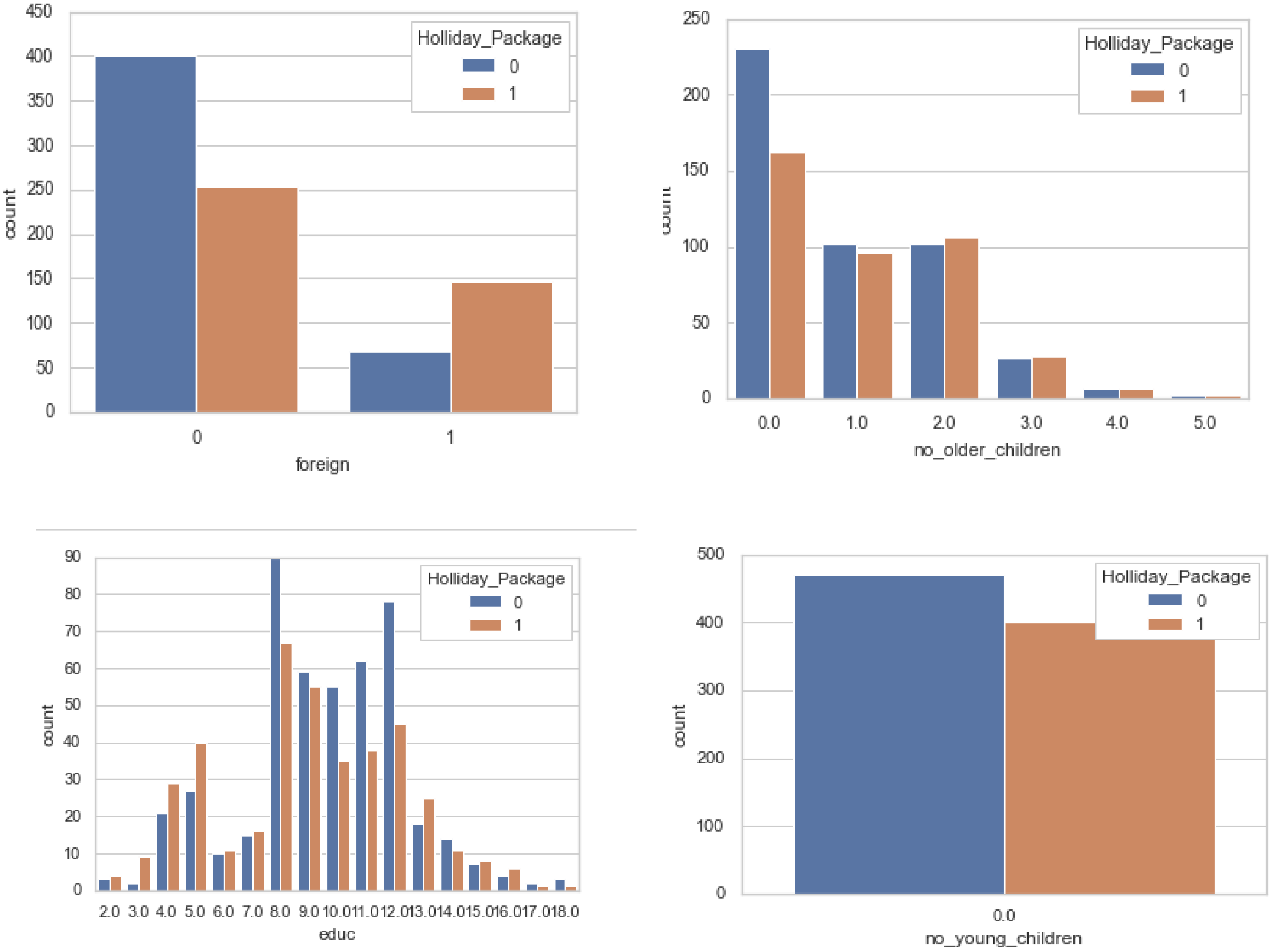
(Fig:-13 Holiday vs Salary)

For salary attribute, it seems to be the people with high salary opted for the holiday packages most than the people with lowest salary



(Fig:-14 Holiday vs Age)

The age between 35 to 45 are going more for the Holliday Package than the rest of the people.



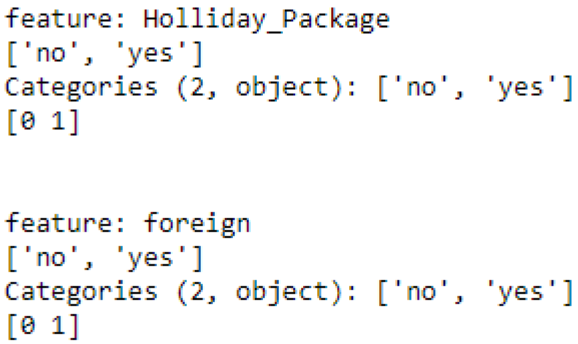
(Fig:-16 Parameters)

From the plots shown above, we can say that people with age ranged between 30 and 50 with a high salary and have less older children are opting for the holiday packages more. It seems that a number of years of formal education does not have that much impact on the target variable.

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).

Solution:-

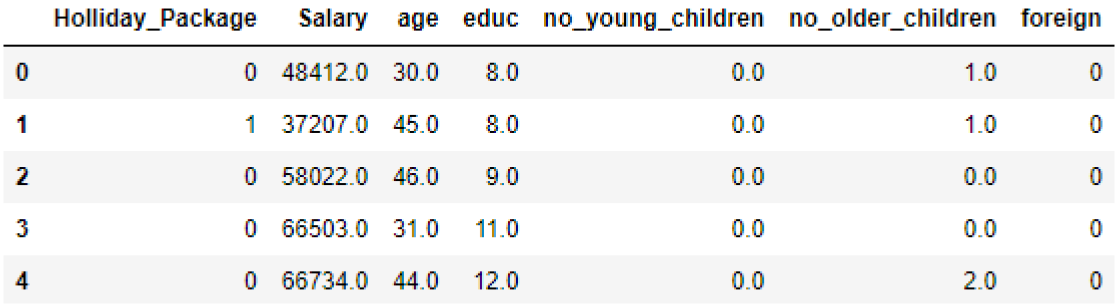
Converting object variables to categorical codes.



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


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

(Fig:-17 Categorical)

Let's check the first five rows in the dataset to check if the conversion is done or not.



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

(Table: - 4 Dataset)

After splitting data into training and test set in **70:30 ratio** and applying **logistic regression and LDA.**

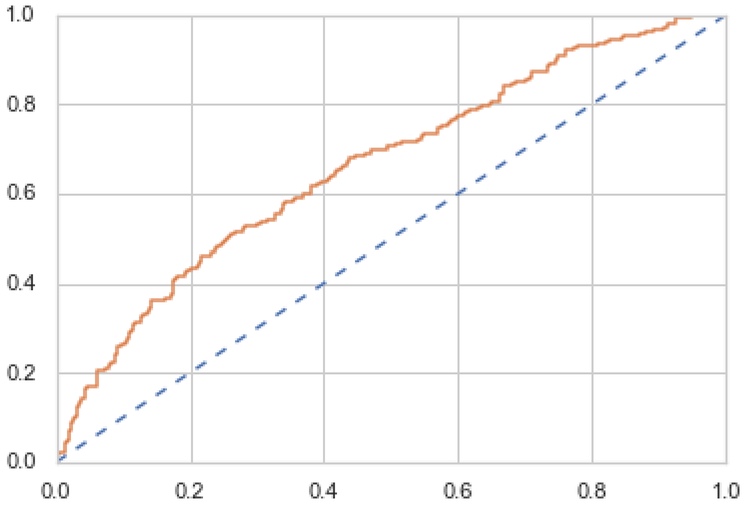
2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model Final Model: Compare both the models and write inference which model is best/optimized.

Solution:-

Logistic model performance metrics:-

The accuracy score for the training data is 0.6272577996715928

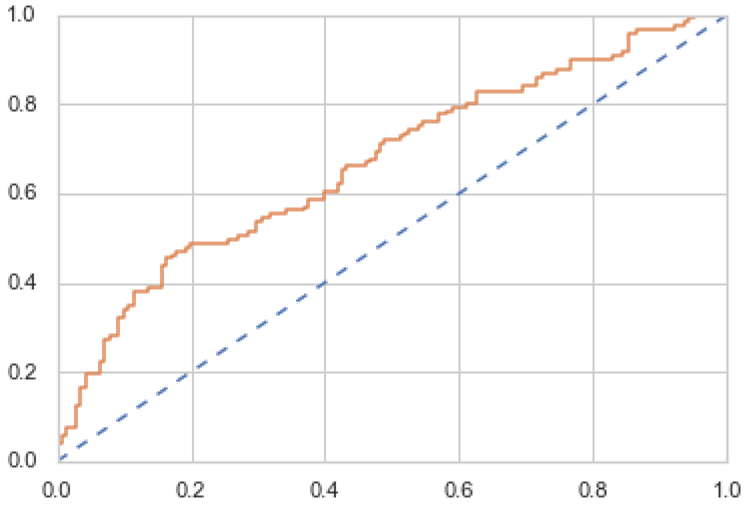
- AUC = 0.663 (# AUC for the training data)



(Fig:-18 AUC for Train Data)

The accuracy score for the test data is 0.6564885496183206

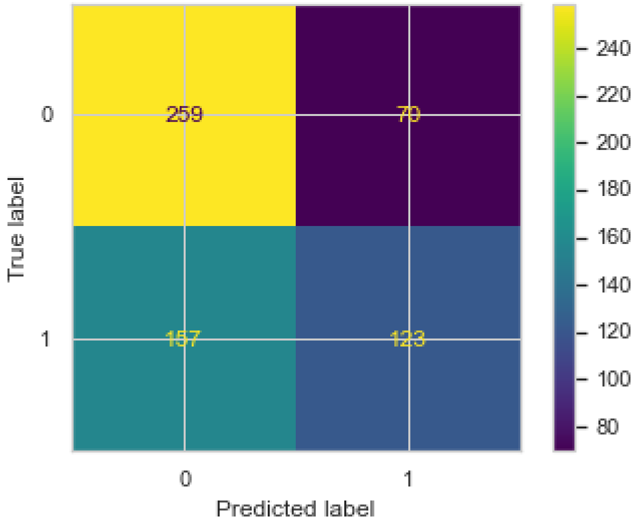
- AUC = 0.663



(Fig:-18 AUC for Train Data)

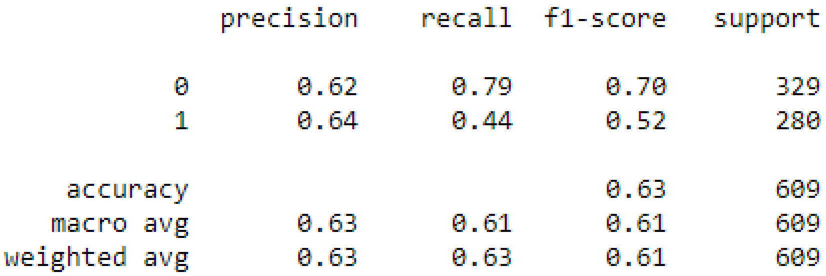
The accuracy and AUC are almost same for the train dataset and test dataset

Confusion Matrix for the training data



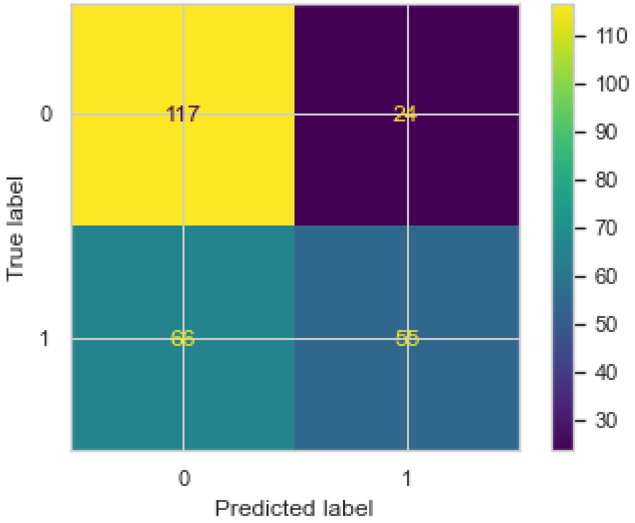
(Fig:-19 Confusion Matrix Test Data)

Classification report for the training data


```
              precision    recall  f1-score   support

           0       0.62      0.79      0.70       329
           1       0.64      0.44      0.52       280

    accuracy                           0.63       609
   macro avg       0.63      0.61      0.61       609
weighted avg       0.63      0.63      0.61       609
```

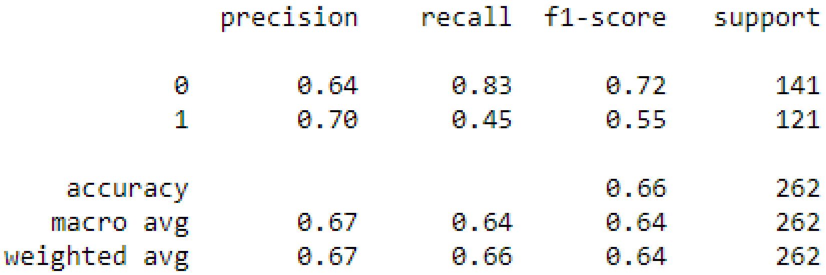
(Fig:-20 Classification Train Data)

Confusion Matrix for test data



(Fig:-21 Confusion Matrix Train Data)

Classification report for the test data



```
              precision    recall  f1-score   support

           0       0.64      0.83      0.72       141
           1       0.70      0.45      0.55       121

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

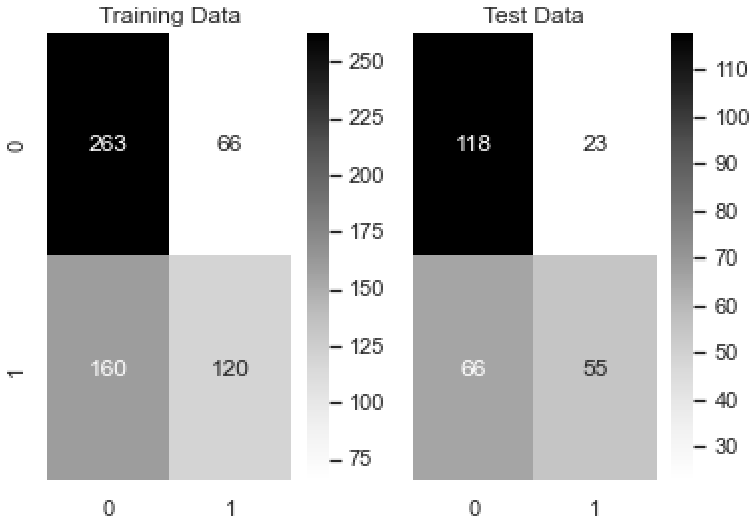
(Fig:-22 Classification Test Data)

- tn - 117 (tn means “true negative”)
- tp - 55
- fp - 24
- fn - 66

- Since all the attributes are poorly correlated with each other and with the target variable, we got the low recall value(45)
- Poor predictors implies poor features

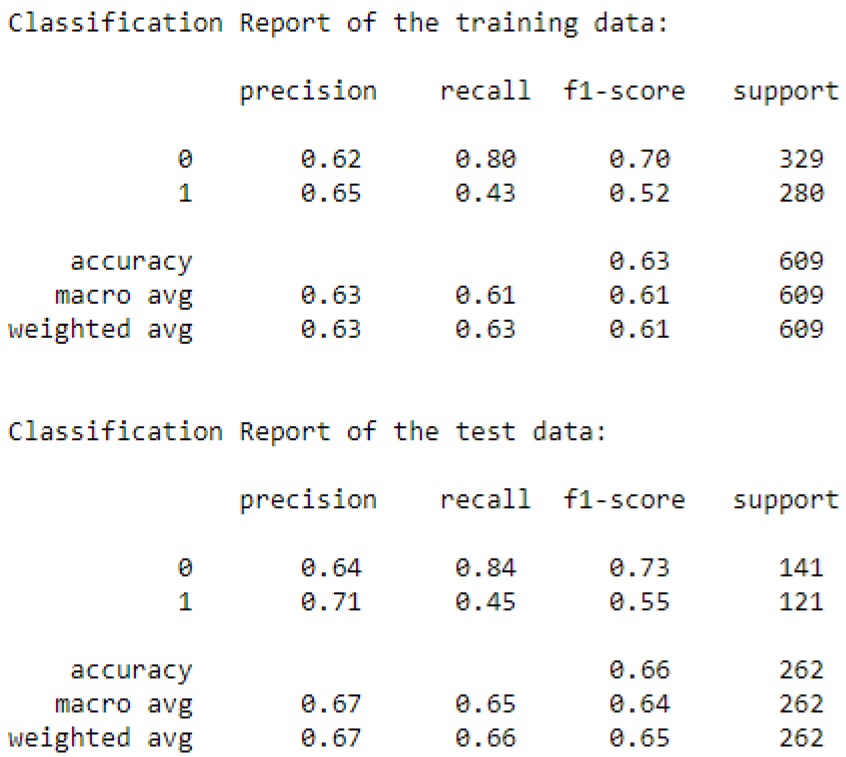
LDA performance metrics

Confusion Matrix for Training Data and Test Data



(Fig:-23 Confusion Test & Train Data)

Classification Report for Training Data and Test Data

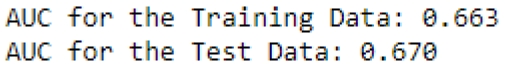


```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.62      0.80      0.70       329
           1       0.65      0.43      0.52       280

    accuracy                           0.63       609
   macro avg       0.63      0.61      0.61       609
weighted avg       0.63      0.63      0.61       609


Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.64      0.84      0.73       141
           1       0.71      0.45      0.55       121

    accuracy                           0.66       262
   macro avg       0.67      0.65      0.64       262
weighted avg       0.67      0.66      0.65       262
```

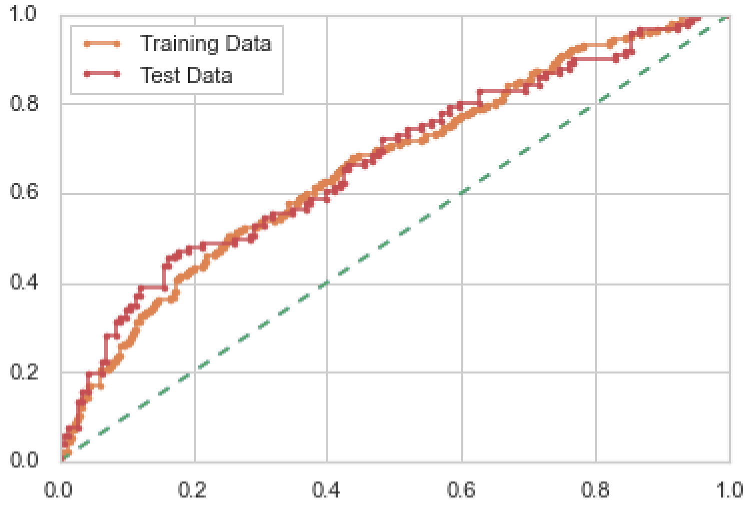
(Fig:-24 Classification Test & Train Data)

AUC for the

- Training Data = 0.663
- AUC for the Test Data = 0.670



```
AUC for the Training Data: 0.663
AUC for the Test Data: 0.670
```



(Fig:-25 AUC Test & Train Data)

- The recall value inferred from the above = 45, which is the same as the logistic model. Because of poor predictors, we got the low recall value.
- Test and training data are also almost the same as we got in logistic regression.
- We can see that there is a lot of difference between training and test data in the confusion matrix.

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

Solution:-

- The training and testing data is almost the same for both the models.
- Logistic regression gave better results than the LDA model.
- Because of the poor predictors, we didn't get a good recall value.
- We got the same recall value of 45 for the logistic model and LDA model. All the attributes aren't highly correlated with each other as well.
- The AUC score for both the training and testing dataset is almost the same for both the models.
- We can say that based on the predictions,
    - ✓ Employees with medium to high level salary and with age range between 30 and 45 are choosing holliday_package.
    - ✓ Employees with very young children are not interested in choosing holliday_package.
- It is recommended for the company to focus on employees with medium level salaries and who are in middle ages (30-45).
- They can neglect the factor years of formal education, since it is not showing much relation with the target variable or other attributes.
- Foreigners are also not the main factor but they can give the second importance to that factor.
- Employees with children who are not very young and very old are opting for holiday packages
- The important factors that company can focus on to sell their packages are salary and age.