Business Report

PREDICTIVE MODELING

Prepared By: ARUNKUMAR S **Date:** 03.07.2022

Batch Name: PGPDSBA Online Jan_E 2022

TABLE CONTENTS

S.NO		CONTENTS	Qty
3		**Figures and table**	
		Total no of Figures	62 No’s
		Total no of Tables	3 No’s

Predictive Modeling project report

Problem 1: Linear Regression

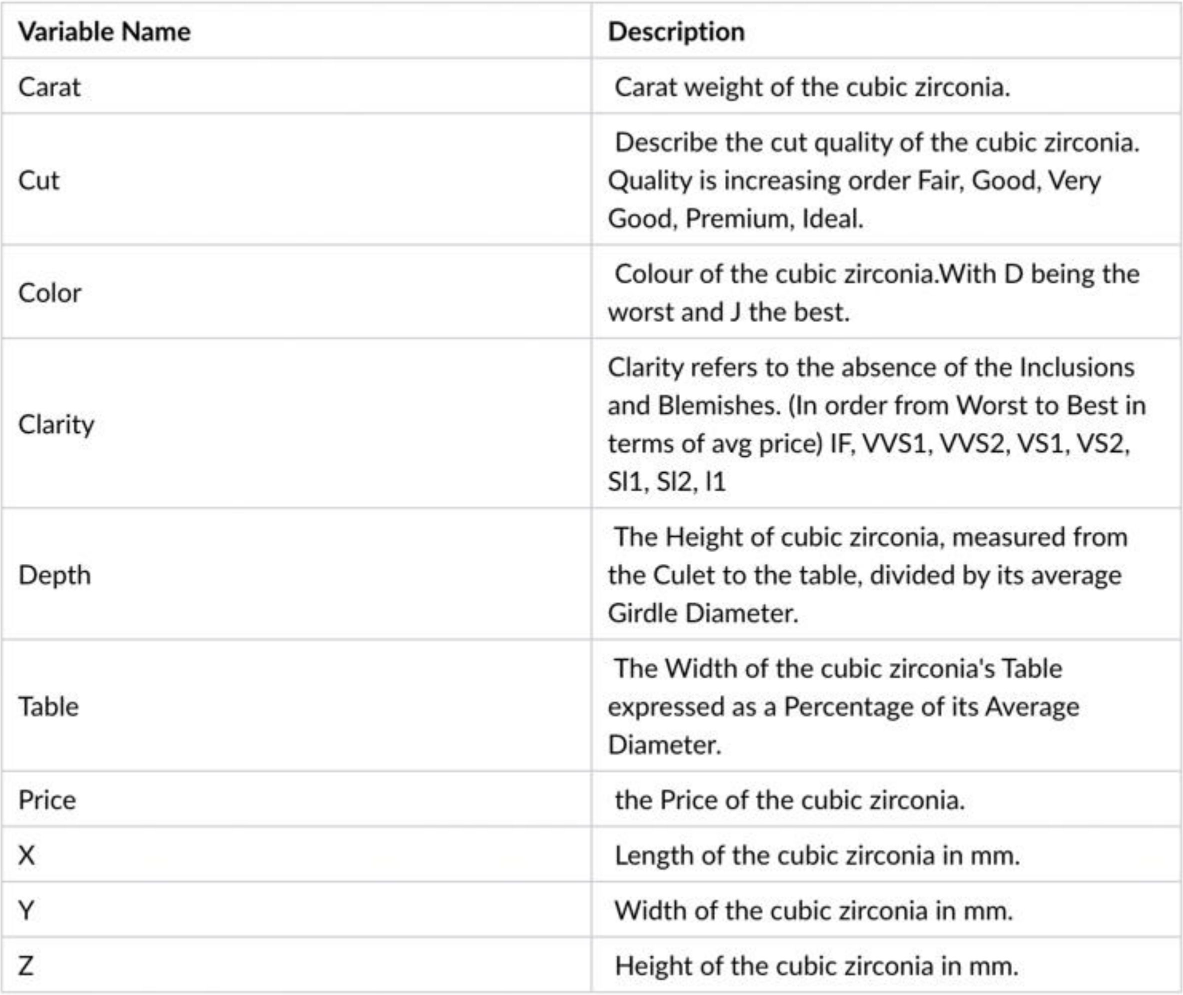
Problem Statement:

You are hired by a company Gem Stones co ltd, which is a cubic zirconia manufacturer. You are provided with the dataset containing the prices and other attributes of almost 27,000 cubic zirconia (which is an inexpensive diamond alternative with many of the same qualities as a diamond). The company is earning different profits on different prize slots. You have to help the company in predicting the price for the stone on the bases of the details given in the dataset so it can distinguish between higher profitable stones and lower profitable stones so as to have better profit share. Also, provide them with the best 5 attributes that are most important.

Domain:

Company Gem Stones co ltd

Data Dictionary:

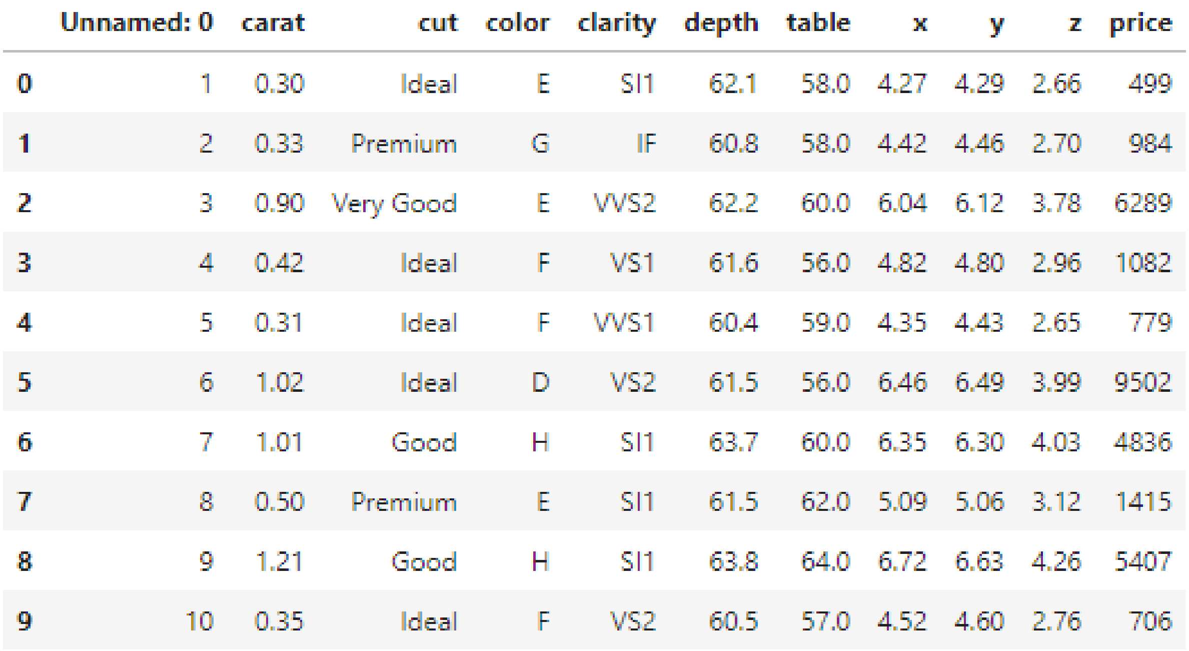


Variable Name	Description
Carat	Carat weight of the cubic zirconia.
Cut	Describe the cut quality of the cubic zirconia. Quality is increasing order Fair, Good, Very Good, Premium, Ideal.
Color	Colour of the cubic zirconia.With D being the worst and J the best.
Clarity	Clarity refers to the absence of the Inclusions and Blemishes. (In order from Worst to Best in terms of avg price) IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1
Depth	The Height of cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter.
Table	The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter.
Price	the Price of the cubic zirconia.
X	Length of the cubic zirconia in mm.
Y	Width of the cubic zirconia in mm.
Z	Height of the cubic zirconia in mm.

Figure no: 1 – Data dictionary of given ‘cubic_zirconia’ data set

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

The head of given data set "cubic_zirconia.csv”

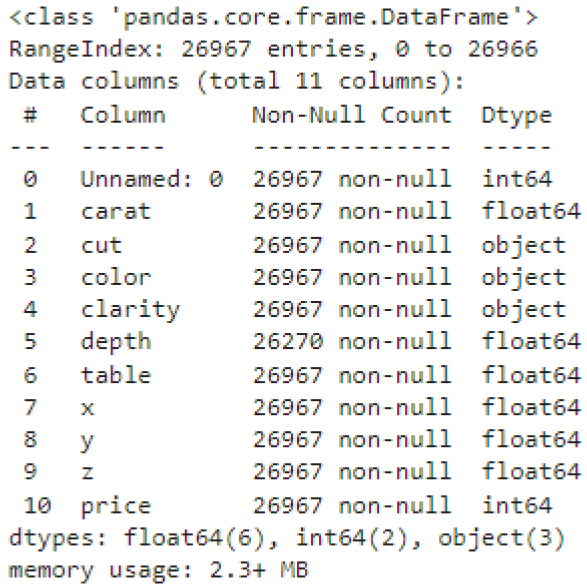


	Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
0	1	0.30	Ideal	E	SI1	62.1	58.0	4.27	4.29	2.66	499
1	2	0.33	Premium	G	IF	60.8	58.0	4.42	4.46	2.70	984
2	3	0.90	Very Good	E	VVS2	62.2	60.0	6.04	6.12	3.78	6289
3	4	0.42	Ideal	F	VS1	61.6	56.0	4.82	4.80	2.96	1082
4	5	0.31	Ideal	F	VVS1	60.4	59.0	4.35	4.43	2.65	779
5	6	1.02	Ideal	D	VS2	61.5	56.0	6.46	6.49	3.99	9502
6	7	1.01	Good	H	SI1	63.7	60.0	6.35	6.30	4.03	4836
7	8	0.50	Premium	E	SI1	61.5	62.0	5.09	5.06	3.12	1415
8	9	1.21	Good	H	SI1	63.8	64.0	6.72	6.63	4.26	5407
9	10	0.35	Ideal	F	VS2	60.5	57.0	4.52	4.60	2.76	706

Figure no: 2 – Head of given ‘cubic_zirconia’ data set

Shape of the dataset:
Rows – 26967
Column - 11

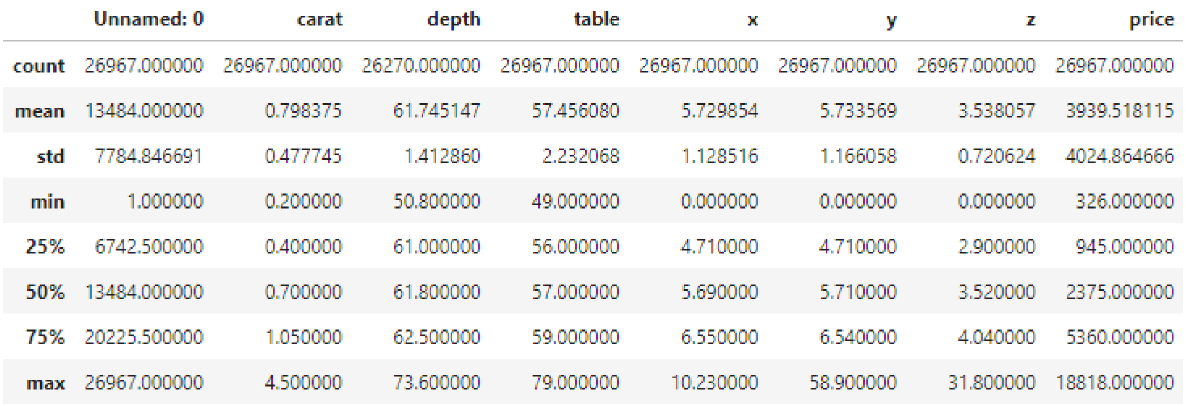
Information about the dataset:



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

Figure no: 3 – Info of ‘cubic_zirconia’ data set

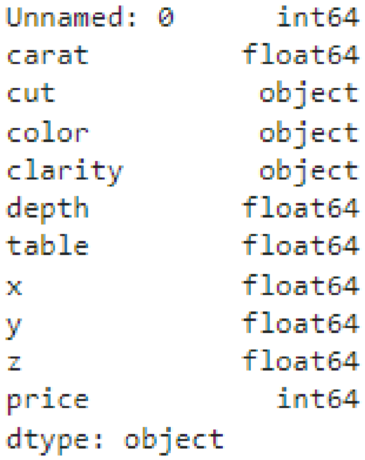
Description of the dataset:



	Unnamed: 0	carat	depth	table	x	y	z	price
count	26967.000000	26967.000000	26270.000000	26967.000000	26967.000000	26967.000000	26967.000000	26967.000000
mean	13484.000000	0.798375	61.745147	57.456080	5.729854	5.733569	3.538057	3939.518115
std	7784.846691	0.477745	1.412860	2.232068	1.128516	1.166058	0.720624	4024.864666
min	1.000000	0.200000	50.800000	49.000000	0.000000	0.000000	0.000000	326.000000
25%	6742.500000	0.400000	61.000000	56.000000	4.710000	4.710000	2.900000	945.000000
50%	13484.000000	0.700000	61.800000	57.000000	5.690000	5.710000	3.520000	2375.000000
75%	20225.500000	1.050000	62.500000	59.000000	6.550000	6.540000	4.040000	5360.000000
max	26967.000000	4.500000	73.600000	79.000000	10.230000	58.900000	31.800000	18818.000000

Figure no: 4 – Description of 'cubic_zirconia' data set

Data types of given dataset values:



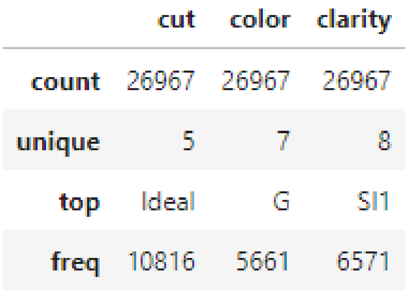
```
Unnamed: 0      int64
carat         float64
cut            object
color          object
clarity        object
depth         float64
table         float64
x             float64
y             float64
z             float64
price           int64
dtype: object
```

Figure no: 5 – Data types of 'cubic_zirconia' data set values

Inference from the Oberservation-1

- The data set contains 26967 row, 11 columns.
- In the given data set there are 2 Integer type features,6 Float type features. 3 Object type features. Where 'price' is the target variable and all other are predictor variable.
- The first column is an index ("Unnamed: 0") as this only serial no, we can remove it.
- Except depth, in all the column non null count is 26967.

The summary statistics of the object variable



	cut	color	clarity
count	26967	26967	26967
unique	5	7	8
top	Ideal	G	SI1
freq	10816	5661	6571

Figure no: 6 – Summary statistics of the object variables

Inference from the Oberservation-2

- On the given data set the mean and median values does not have much difference.
- We can observe Min value of "x", "y", & "z" are zero this indicates that they are faulty values.
- As we know dimensionless or 2-dimensional diamonds are not possible. So we need to filter out those as it clearly faulty data entries.
- There are three object data type 'cut', 'color' and 'clarity'.

Performing EDA: We will follow the below mentioned steps to perform EDA

Step 1: Checking & removing duplicates.

Step 2: Checking and treating Missing value.

Step 3: Outlier Treatment.

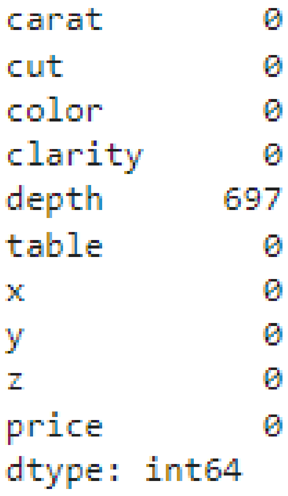
Step 4: Univariate Analysis.

Step 5: Bivariate Analysis

EDA-Step-1: Checking for duplicate records in the data

The dataset contains number of duplicate rows = 33

EDA-Step 2: Checking Missing value



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

Figure no: 7– Missing value counts of given dataset

Inference from the Oberservation-3

- We can observe there are 697 missing value in the ‘depth’ column
- Missing value treatment will be done in section 1.2.

EDA-Step 3: Outlier Checks.

Using boxplot function we can see the outliers presence in before and after stage as follows

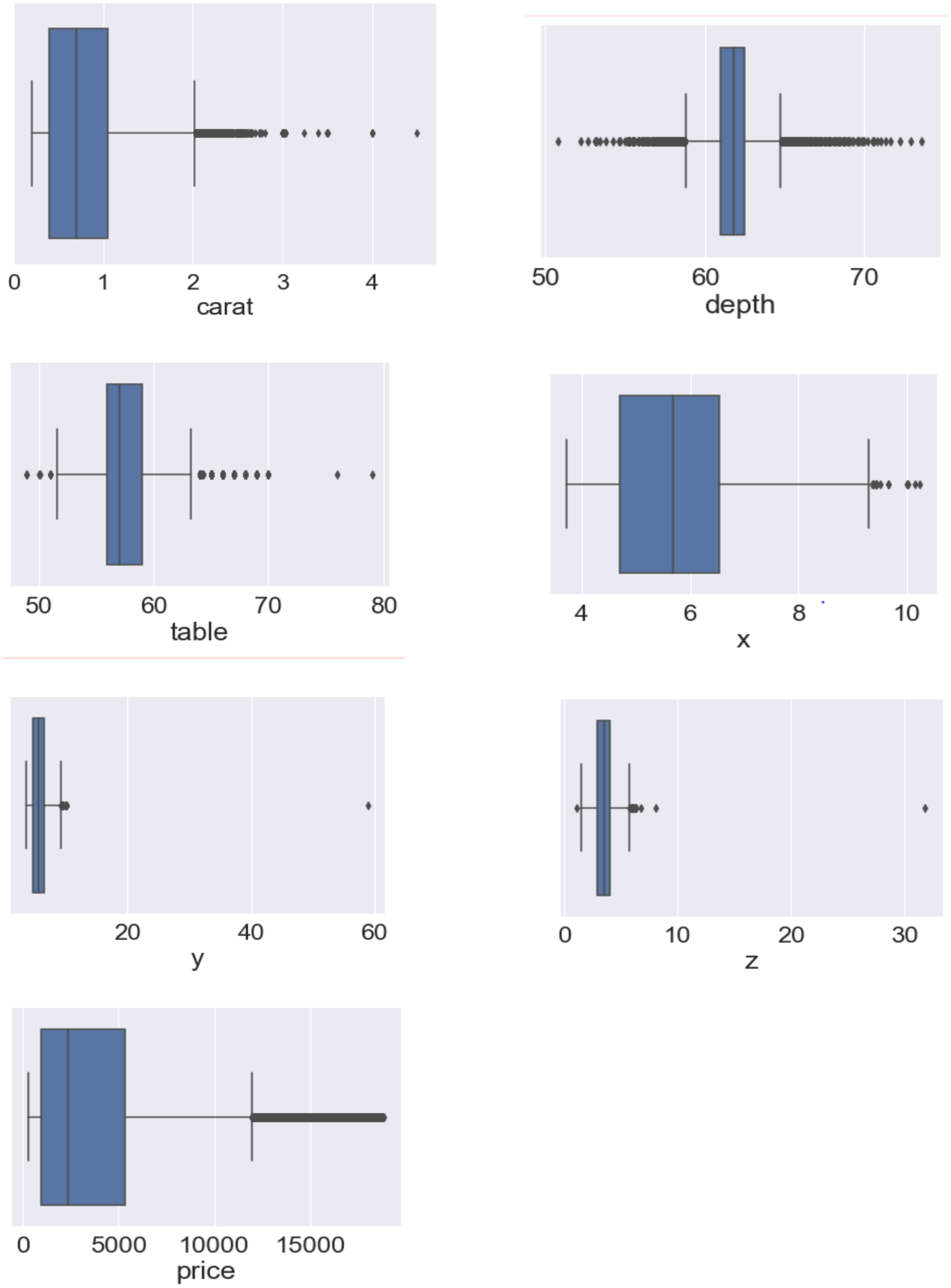


Figure no: 8– Before outliers treatment of the dataset variables

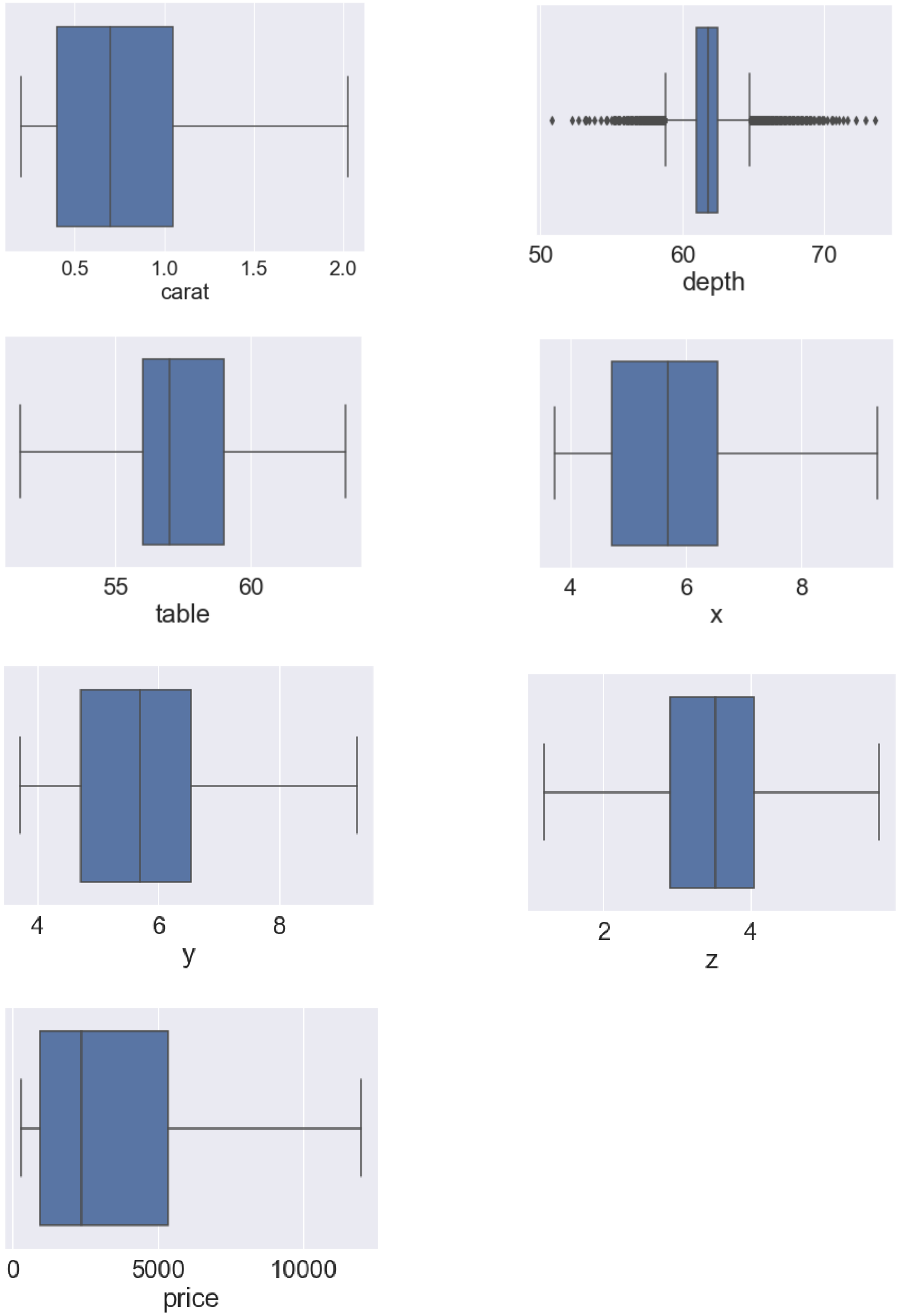


Figure no: 9– After outliers treatment of the dataset variables

EDA-Step 4: Univariate Analysis.

We can see the distribution of the variable using histplot function as follows,

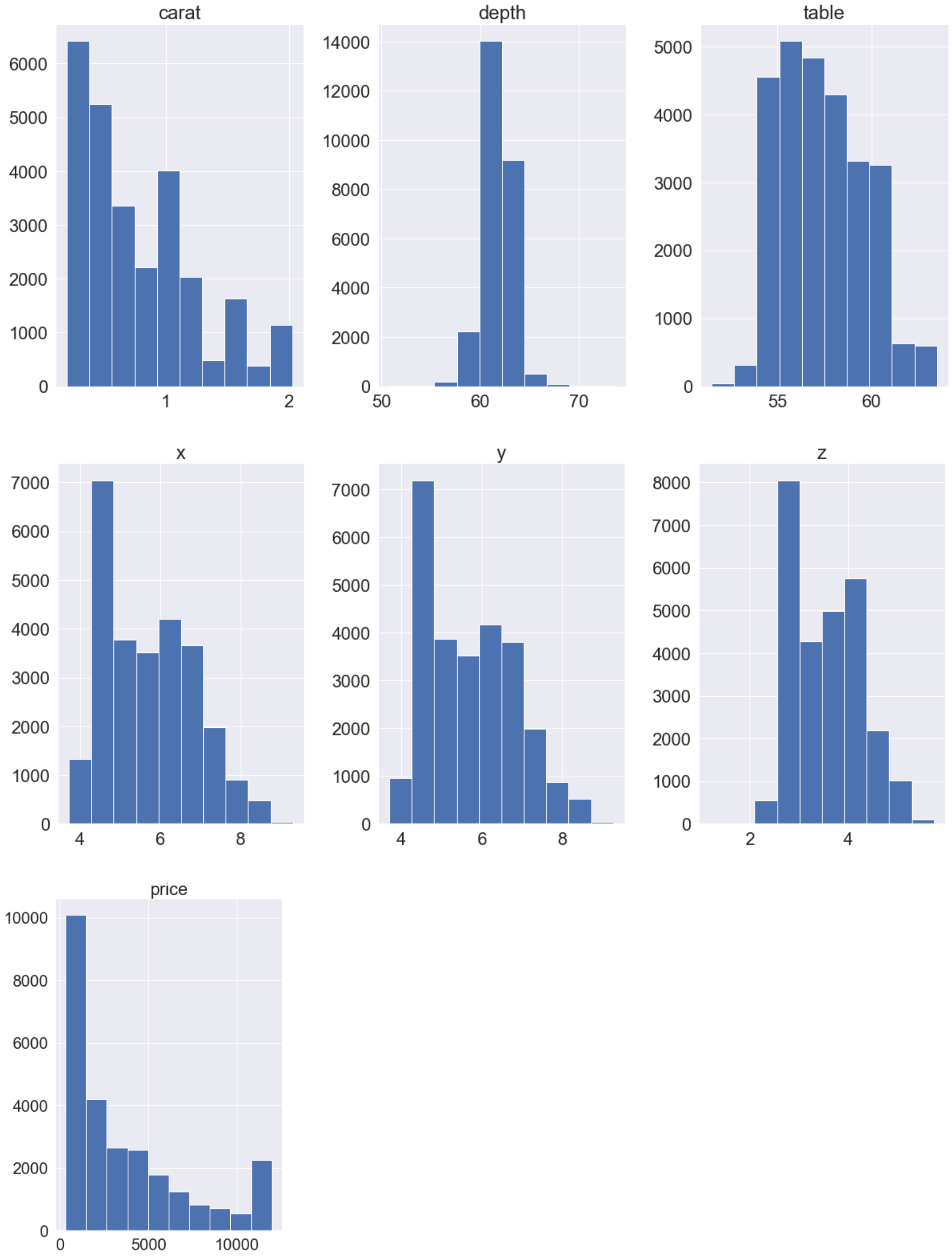


Figure no: 10– Distribution view histplot of the dataset variables

The measure of skeweness of attribute.

```
carat     0.917214
depth    -0.025042
table     0.480476
x         0.397696
y         0.394060
z         0.394819
price     1.157121
dtype: float64
```

Inference from the Oberservation-4

- There is significant amount of outlier present in some variable.
- We can see that the distribution of some quantitative features like "carat" and the target feature "price" are heavily "right-skewed".

EDA-Step 5 : Bivariate Analysis

Pairplot view between dataset variables

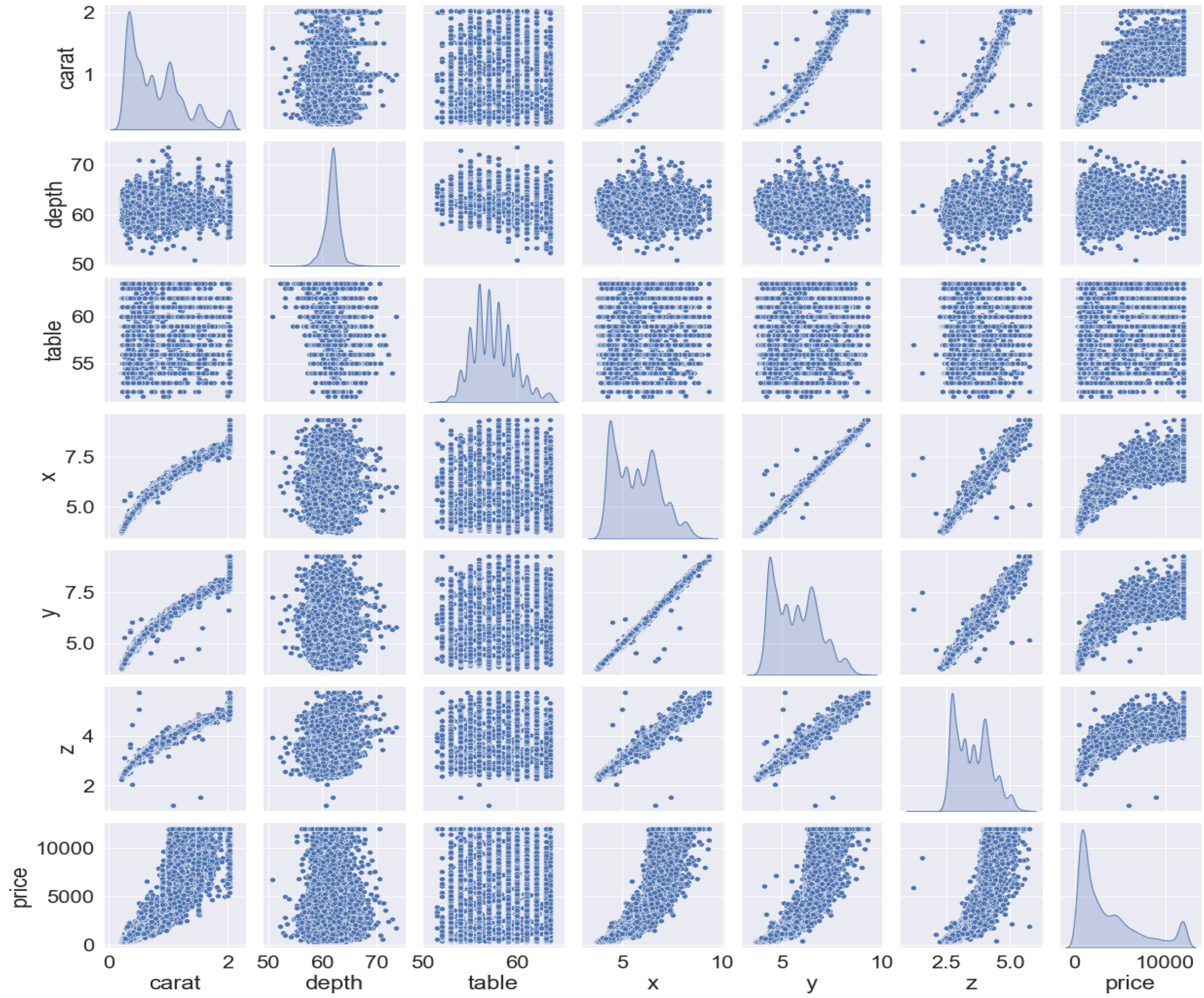


Figure no: 11– Pairplot of the dataset variables

Heatmap correlation view of the dataset

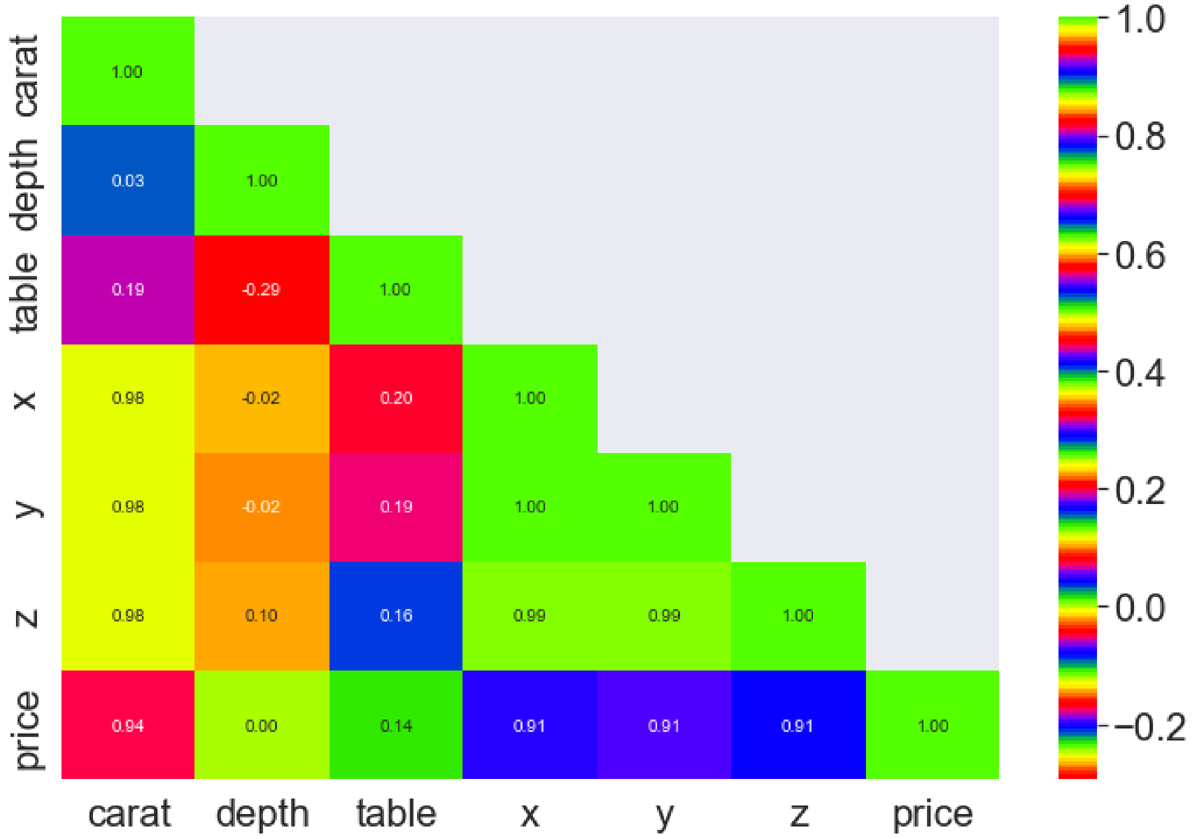


Figure no: 12– heatmap of the dataset variables

How each feature affects the price of diamonds

```
price     1.000000
carat     0.936765
y         0.914838
x         0.913409
z         0.908599
table     0.137915
depth     0.000313
Name: price, dtype: float64
```

Inference from the Oberservation-5

- It can be inferred that most features correlate with the price of Diamond.
- The notable exception is "depth" which has a negligible correlation (<1%).

EDA for Categorical variable.

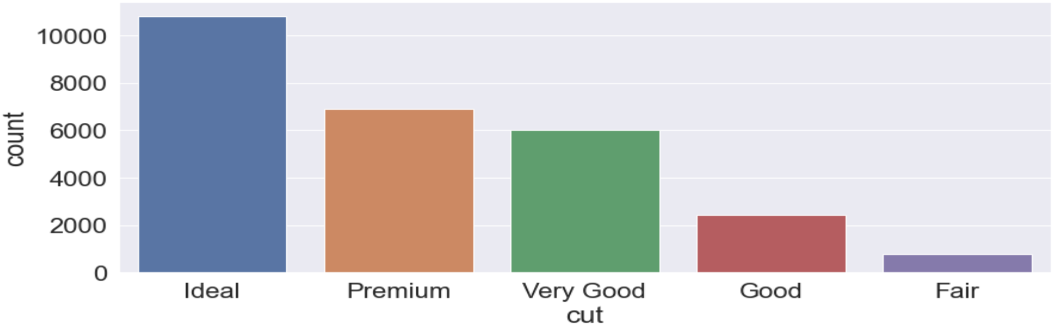


Figure no: 13– Catplot view of cut Vs count

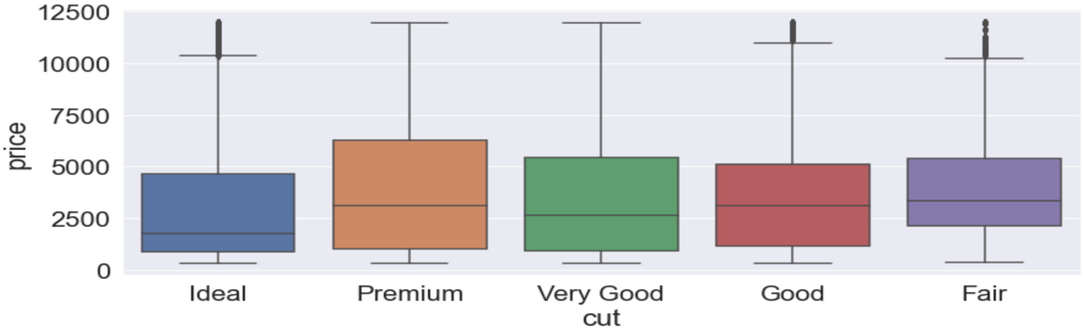


Figure no: 14– Catplot view of cut Vs price

Observation on 'CUT':

- The Premium Cut on Diamonds are the most Expensive, followed by Very Good Cut.

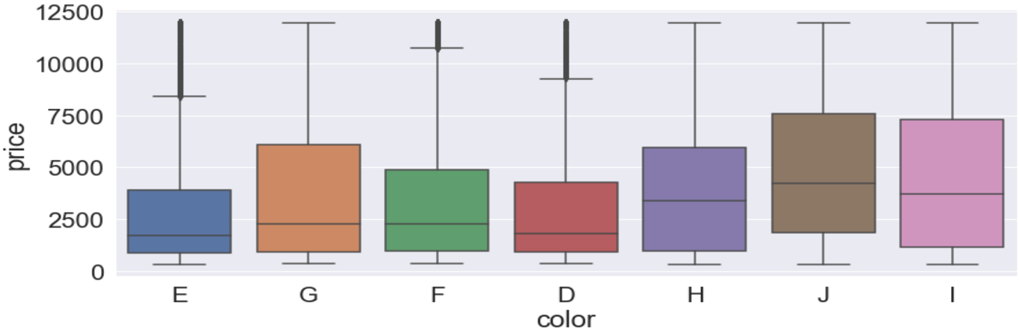


Figure no: 15– Catplot view of color Vs price

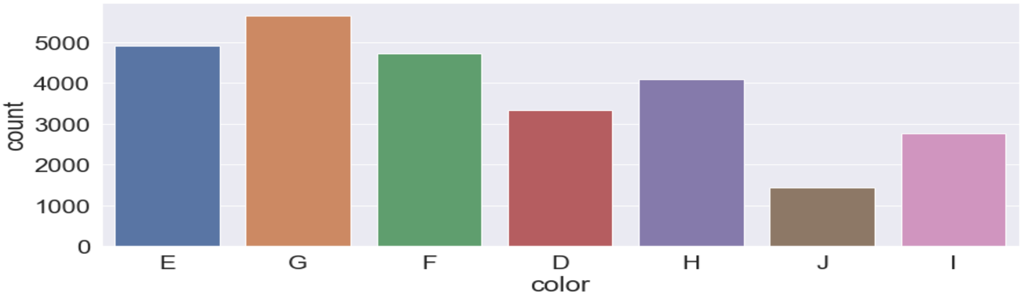


Figure no: 16– Catplot view of color Vs count

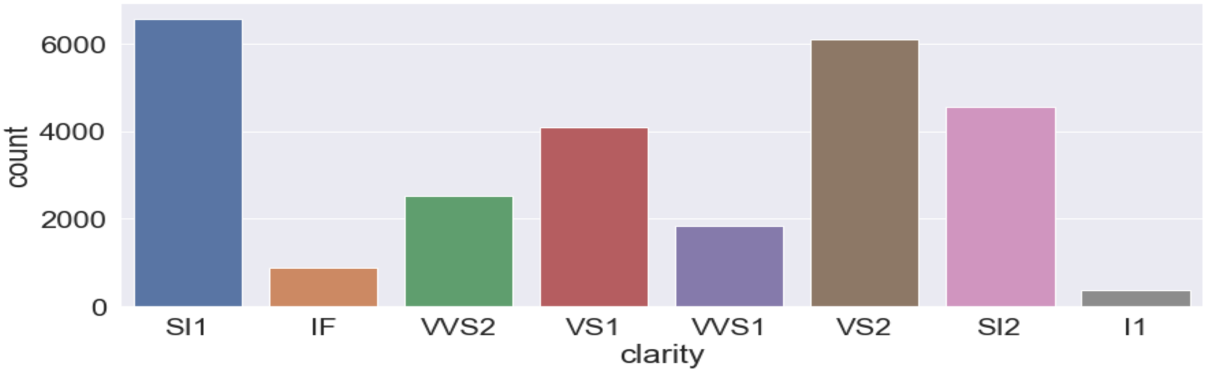


Figure no: 17– Catplot view of clarity Vs count

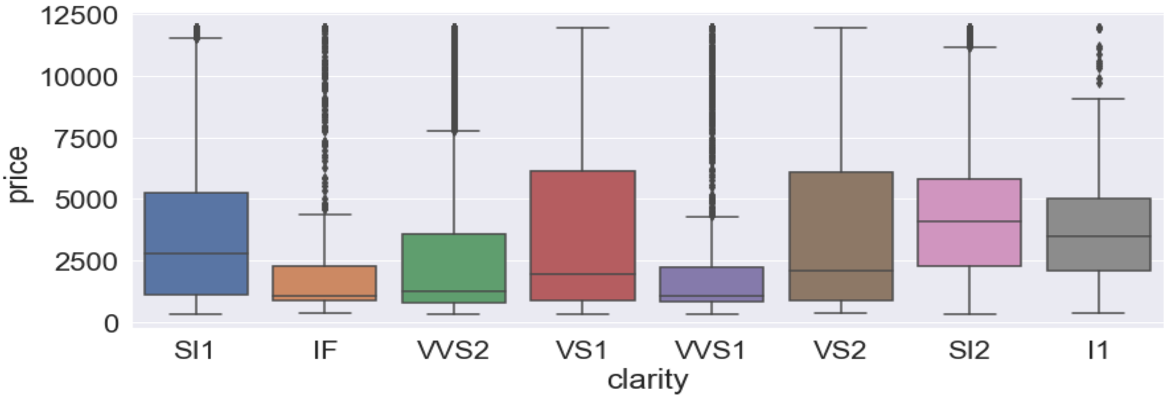


Figure no: 18– Catplot view of clarity Vs price

Observation on 'clarity':

- The Diamonds clarity with VS1 & VS2 are the most Expensive

The inferences drawn from the above Exploratory Data analysis:

Observation-1:

- 'Price' is the target variable while all others are the predictors.
- The data set contains 26967 row, 11 column.
- In the given data set there are 2 Integer type features,6 Float type features. 3 Object type features. Where 'price' is the target variable and all other are predector variable.
- The first column is an index ("Unnamed: 0")as this only serial no, we can remove it.

Observation-2:

- On the given data set the the mean and median values does not have much differenc.
- We can observe Min value of "x", "y", "z" are zero this indicates that they are faulty values. As we know dimensionless or 2-dimensional diamonds are not possible. So we have filter out those as it clearly faulty data entries.
- There are three object data type 'cut', 'color' and 'clarity'.

Observation-3:

- We can observe there are 697 missing value in the depth column.
- There are some duplicate row present. (33 duplicate rows out of 26958). Which is nearly 0.12 % of the total data.
- So on this case we have dropped the duplicated row.

Observation-4:

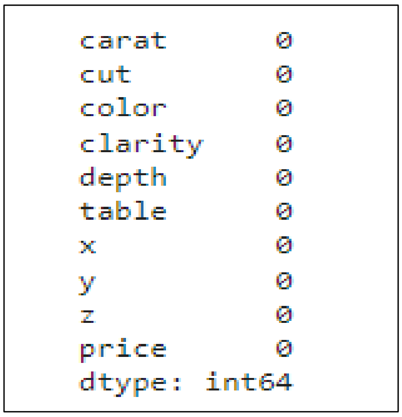
- There are significant amount of outlier present in some variable, the features with data point that are far from the rest of dataset which will affect the outcome of our regression model.
- So we have treat the outlier. We can see that the distribution of some quantitative features like "carat" and the target feature "price" are heavily "right-skewed".

Observation-5:

- It looks like most features do correlate with the price of Diamond.
- The notable exception is "depth" which has a negligible correlation (~1%).
- Observation on 'CUT': The Premium Cut on Diamonds are the most Expensive, followed by Very Good Cut.

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.

Imputing the missing values



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

- Fill the median value where the missing values
- Checking for the values which are equal to zero.
- In Qs.1.1 we have already check for 'Zero' value. And we can observe there are some amount of 'Zero' value present on the data set on variable 'x', 'y','z'.
- This indicates that they are faulty values.
- As we know dimensionless or 2-dimensional diamonds are not possible. So we have filter out those as it clearly faulty data entries.

Table No: 1 – Missing value counts after imputing the missing values treatment

Do you think scaling is necessary in this case?

- Scaling or standardizing the features around the centre and 0 with a standard deviation of 1 is important when we compare measurements that have different units. Variables that are measured at different scales do not contribute equally to the analysis and might end up creating a bias.

- For example, A variable that ranges between 0 and 1000 will outweigh a variable that ranges between 0 and 1. Using these variables without standardization will give the variable with the larger range weight of 1000 in the analysis. Transforming the data to comparable scales can prevent this problem.

- In this data set we can see the all the variable are in different scale i.e. price are in 1000s unit and depth and table are in 100s unit, and carat is in 10s. So it's necessary to scale or standardise the data to allow each variable to be compared on a common scale. With data measured in different "units" or on different scales (as here with different means and variances) this is an important data processing step if the results are to be meaningful or not dominated by the variables that have large variances.

But is scaling necessary in this case?

- No, it is not necessary, we'll get an equivalent solution whether we apply some kind of linear scaling or not. But recommended for regression techniques as well because it would help gradient descent to converge fast and reach the global minima. When number of features becomes large, it helps is running model quickly else the starting point would be very far from minima, if the scaling is not done in pre-processing.

- For now we will process the model without scaling and later we will check the output with scaled data of regression model output.

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.

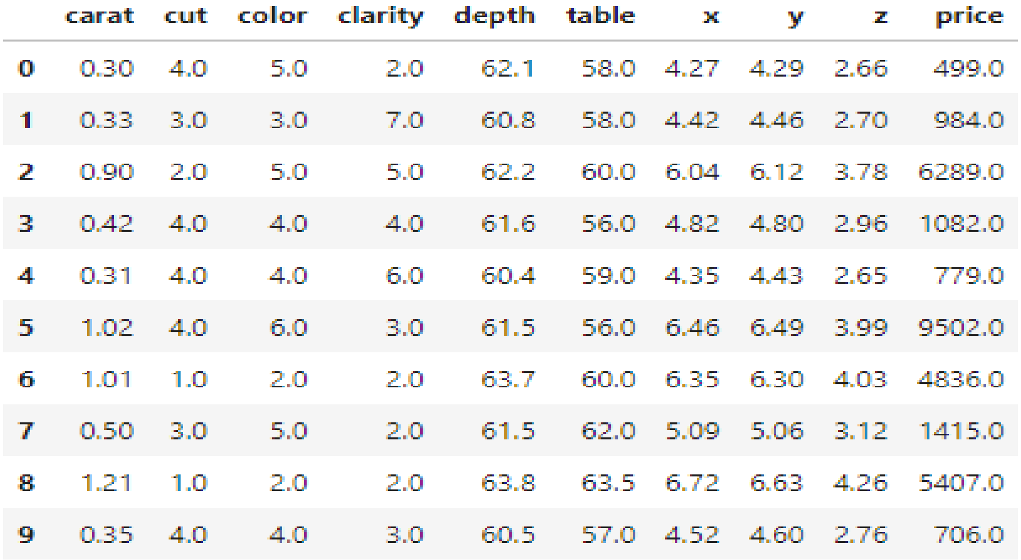
Encoding the string values as follows,

Column 'cut':
Fair-0, Good-1, Very Good-2, Premium-3, Ideal-4

Column 'color':
D-6, E-5, F-4, G-3, H-2, I-1, J-0

Column 'clarity':
IF-7, VVS1-6, VVS2-5, VS1-4, VS2-3, SI1-2, SI2-1, I1-0

Head of the dataset after Encoding:

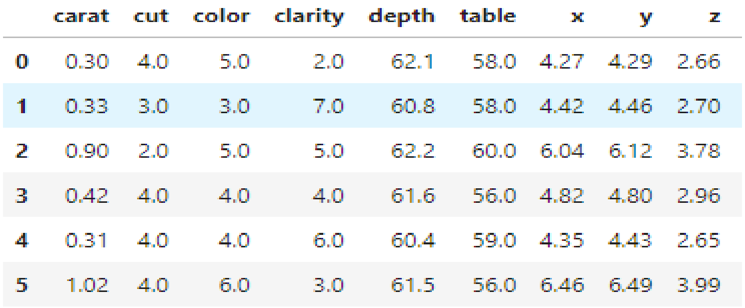


	carat	cut	color	clarity	depth	table	x	y	z	price
0	0.30	4.0	5.0	2.0	62.1	58.0	4.27	4.29	2.66	499.0
1	0.33	3.0	3.0	7.0	60.8	58.0	4.42	4.46	2.70	984.0
2	0.90	2.0	5.0	5.0	62.2	60.0	6.04	6.12	3.78	6289.0
3	0.42	4.0	4.0	4.0	61.6	56.0	4.82	4.80	2.96	1082.0
4	0.31	4.0	4.0	6.0	60.4	59.0	4.35	4.43	2.65	779.0
5	1.02	4.0	6.0	3.0	61.5	56.0	6.46	6.49	3.99	9502.0
6	1.01	1.0	2.0	2.0	63.7	60.0	6.35	6.30	4.03	4836.0
7	0.50	3.0	5.0	2.0	61.5	62.0	5.09	5.06	3.12	1415.0
8	1.21	1.0	2.0	2.0	63.8	63.5	6.72	6.63	4.26	5407.0
9	0.35	4.0	4.0	3.0	60.5	57.0	4.52	4.60	2.76	706.0

Figure no: 19– Head of dataset after encoding process

Data Split:

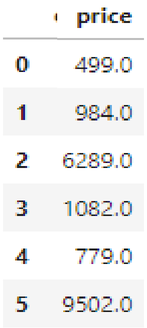
A) Independent Variable (X):



	carat	cut	color	clarity	depth	table	x	y	z
0	0.30	4.0	5.0	2.0	62.1	58.0	4.27	4.29	2.66
1	0.33	3.0	3.0	7.0	60.8	58.0	4.42	4.46	2.70
2	0.90	2.0	5.0	5.0	62.2	60.0	6.04	6.12	3.78
3	0.42	4.0	4.0	4.0	61.6	56.0	4.82	4.80	2.96
4	0.31	4.0	4.0	6.0	60.4	59.0	4.35	4.43	2.65
5	1.02	4.0	6.0	3.0	61.5	56.0	6.46	6.49	3.99

Figure no: 20– Independent variable (X) dataset

B) Target Variable (y)

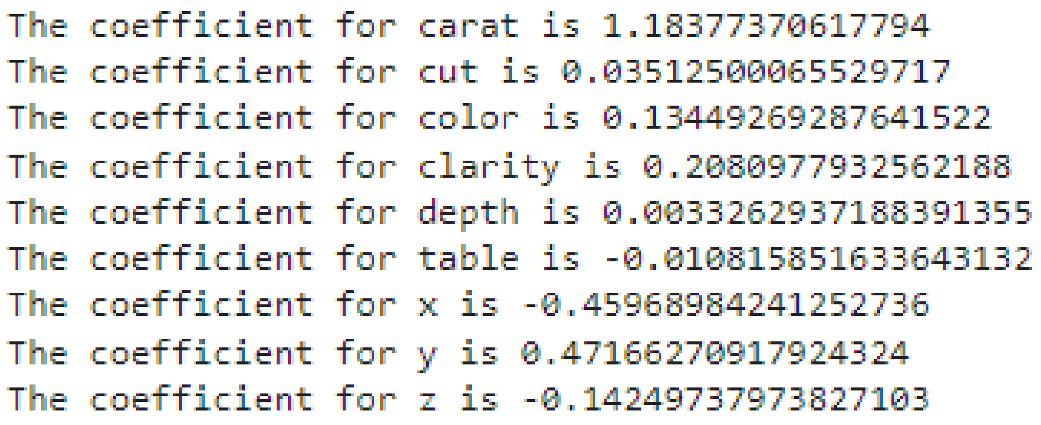


	price
0	499.0
1	984.0
2	6289.0
3	1082.0
4	779.0
5	9502.0

Figure no: 21– Target variable (y) dataset

C) Linear regression model

Successfully built the model for linear regression using LinearRegression() function.



```
The coefficient for carat is 1.18377370617794
The coefficient for cut is 0.03512500065529717
The coefficient for color is 0.13449269287641522
The coefficient for clarity is 0.2080977932562188
The coefficient for depth is 0.0033262937188391355
The coefficient for table is -0.010815851633643132
The coefficient for x is -0.45968984241252736
The coefficient for y is 0.47166270917924324
The coefficient for z is -0.14249737973827103
```

Figure no: 22– Coefficient of variables

Observation-1:

- Y=mx +c (m= m1, m2, m3, m4, m5, m6, m7, m8 and m9) here 9 different co-efficient will learn along with the intercept which is "c" from the model.
- From the above coefficients for each of the independent attributes we can conclude
- The one unit increase in carat increases price by 8901.941.
- The one unit increase in cut increases price by 109.188.
- The one unit increase in color increases price by 272.921.
- The one unit increase in clarity increases price by 436.441.
- The one unit increase in y increases price by 1464.827.
- The one unit increase in depth increases price by 8.236,
- But The one unit increase in table decreases price by -17.345,
- The one unit increase in x decreases price by -1417.908,
- The one unit increase in z decreases price by -711.225.

Check the intercept for the model:

- The intercept for our model is -3171.950447307687

Observation-2:

- The intercept (often labelled the constant) is the expected mean value of Y when all X=0. If X never equals 0, then the intercept has no intrinsic meaning.

- The intercept for our model is -3171.950447307667. In present case when the other predictor variable are zero i.e. like carat, cut, color, clarity all are zero then the C=-3172. (Y = m1X1 + m2X2+ ….. + mnXn + C + e) that means price is -3172. Which is meaningless. We can do Z score or scaling the data and make it nearly zero.

Model score - R2 or co-eff of determinant:

- The regression model score – 0.931543712584074

Observation-3:

- R-square is the percentage of the response variable variation that is explained by a linear model. Or:
- R-square = Explained variation / Total variation
- R-squared is always between 0 and 100%: 0% indicates that the model explains none of the variability of the response data around its mean.100% indicates that the model explains all the variability of the response data around its mean. In this regression model we can see the R-square value on Training and Test data respectively 0.9311935886926559 and 0.931543712584074.

RMSE on Training data - 907.1312415459143

RMSE on testing data - 911.8447345328436

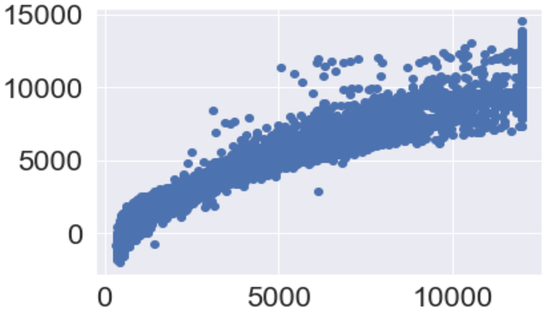


Figure no: 23– Scatter plot the predicted y value vs actual y values for the test data

Observation-4:

- we can see that the is a linear plot, very strong correlation between the predicted y and actual y. But there are lots of spread. That indicated some kind noise present on the data set i.e. unexplained variances on the output.

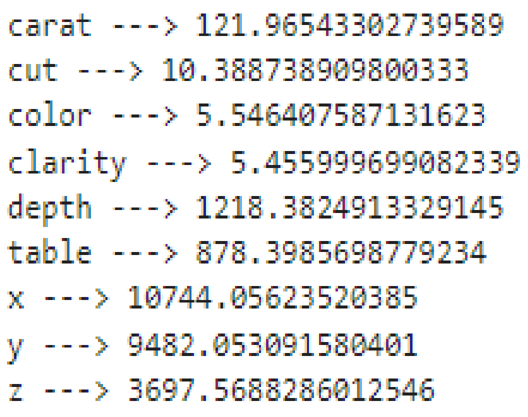
Linear regression Performance Metrics:

- intercept for the model: -3171.950447307667
- R square on training data: 0.9311935886926559
- R square on testing data: 0.931543712584074
- RMSE on Training data: 907.1312415459143
- RMSE on Testing data: 911.8447345328436
- As the training data & testing data score are almost inline, we can conclude this model is a Right-Fit Model.

Observation-5:

- Now we can observe by applying z score the intercept became -5.87961525130473e-16.
- Earlier it was -3171.950447307667. The co-efficient has changed, the bias became nearly zero but the overall accuracy still same.

Checking of Multi-collinearity using VIF



```
carat ---> 121.96543302739589
cut ---> 10.38873890980033
color ---> 5.546407587131623
clarity ---> 5.455999699082339
depth ---> 1218.3824913329145
table ---> 878.3985698779234
x ---> 10744.05623520385
y ---> 9482.053091580401
z ---> 3697.5688286012546
```

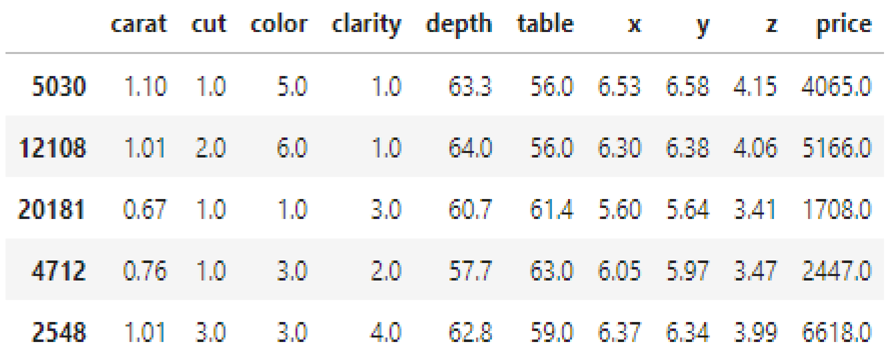
Figure no: 24– Multicollinearity presence in the data

- We can observe there are very strong multi collinearity present in the data set. Ideally it should be within 1 to 5.

- We are exploring the Linear Regression using stats models as we are interested in some more statistical metrics of the model.

Linear Regression using statsmodels:

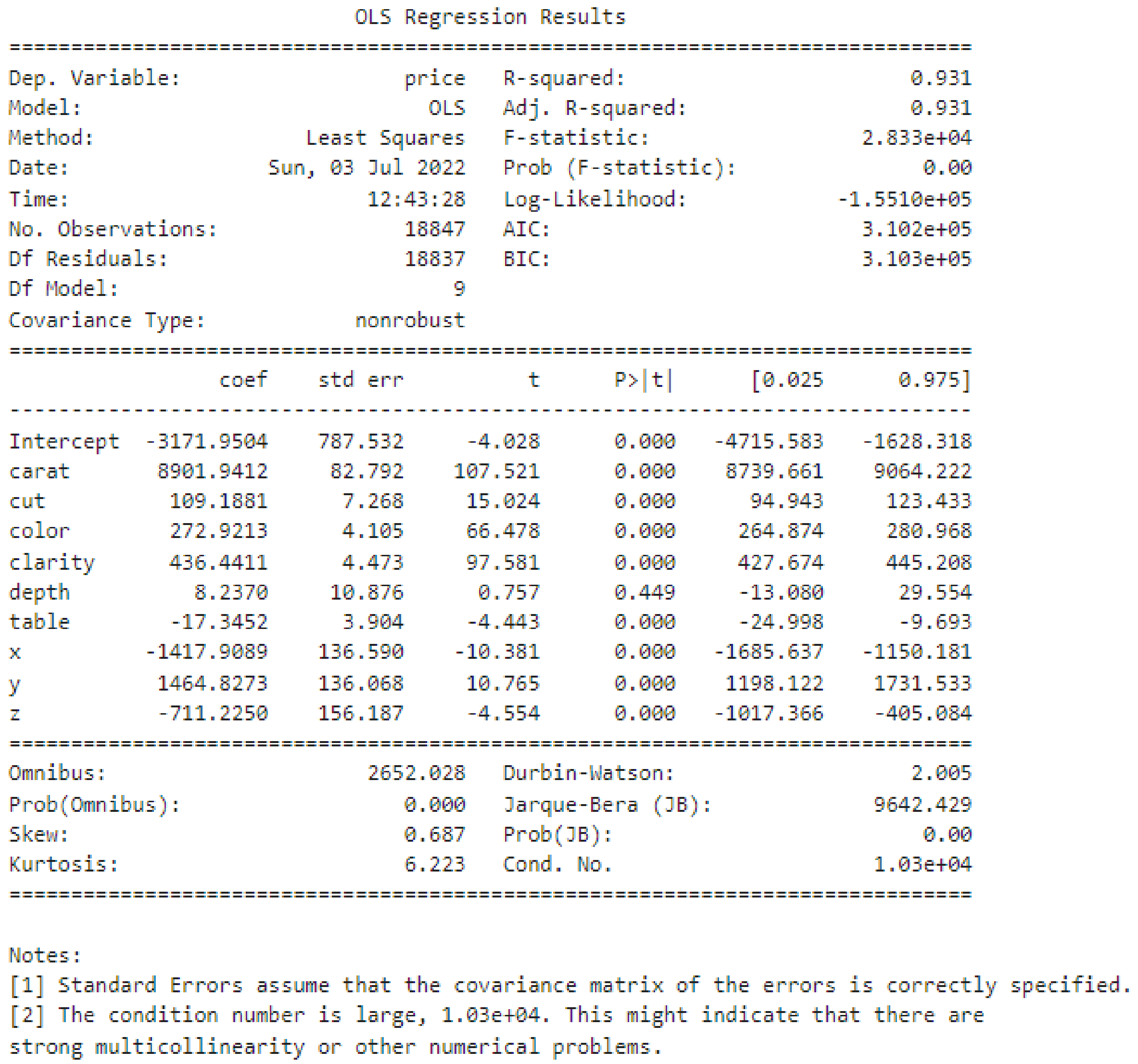
Concatenate X and y into a single dataframe



	carat	cut	color	clarity	depth	table	x	y	z	price
5030	1.10	1.0	5.0	1.0	63.3	56.0	6.53	6.58	4.15	4065.0
12108	1.01	2.0	6.0	1.0	64.0	56.0	6.30	6.38	4.06	5166.0
20181	0.67	1.0	1.0	3.0	60.7	61.4	5.60	5.64	3.41	1708.0
4712	0.76	1.0	3.0	2.0	57.7	63.0	6.05	5.97	3.47	2447.0
2548	1.01	3.0	3.0	4.0	62.8	59.0	6.37	6.34	3.99	6618.0

Figure no: 25– Concatenate X and y into a single dataframe

Inferential statistics:



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.931
Model:                            OLS   Adj. R-squared:                  0.931
Method:                 Least Squares   F-statistic:                 2.833e+04
Date:                Sun, 03 Jul 2022   Prob (F-statistic):               0.00
Time:                        12:43:28   Log-Likelihood:            -1.5510e+05
No. Observations:               18847   AIC:                         3.102e+05
Df Residuals:                   18837   BIC:                         3.103e+05
Df Model:                           9
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept  -3171.9504    787.532     -4.028      0.000   -4715.583   -1628.318
carat       8901.9412     82.792    107.521      0.000    8739.661    9064.222
cut          109.1881      7.268     15.024      0.000      94.943     123.433
color        272.9213      4.105     66.478      0.000     264.874     280.968
clarity      436.4411      4.473     97.581      0.000     427.674     445.208
depth          8.2370     10.876      0.757      0.449     -13.080      29.554
table        -17.3452      3.904     -4.443      0.000     -24.998      -9.693
x          -1417.9089    136.590    -10.381      0.000   -1685.637   -1150.181
y           1464.8273    136.068     10.765      0.000    1198.122    1731.533
z           -711.2250    156.187     -4.554      0.000   -1017.366    -405.084
==============================================================================
Omnibus:                     2652.028   Durbin-Watson:                   2.005
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             9642.429
Skew:                           0.687   Prob(JB):                         0.00
Kurtosis:                       6.223   Cond. No.                     1.03e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.03e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Figure no: 26– OLS Regression Results

Observation-6:

- Assuming null hypothesis is true, i.e. there is no relationship between this variable with price. From that universe we have drawn the sample and on this sample we have found this co-efficient for the variable shown above.

- Now we can ask what is the probability of finding this co-efficient in this drawn sample if in the real world the co-efficient is zero. As we see here the overall P value is less than alpha, so rejecting H0 and accepting Ha that atleast 1 regression co-efficient is not '0'. Here all regression co-efficient are not '0'.

- For an example: we can see the p value is showing 0.449 for 'depth' variable, which is much higher than 0.05. That means this dimension is useless. So we can say that the attribute which are having p value greater than 0.05 are poor predictor for price.

The final Linear Regression equation is following

price = b0 + b1 carat[T.True] + b2 cut + b3 color + b4 clarity+ b5 depth + b6 table + b7 x + b8 y + b9 *z True

price = (-3171.95) Intercept + (8901.94) carat + (109.19) cut + (272.92) color + (436.44) clarity + (8.24) depth + (-17.35) table + (-1417.91)) x + (1464.83) y + (-711.23) z _True

- When carat increases by 1 unit, diamond price increases by 8901.94 units, keeping all other predictors constant.
- When cut increases by 1 unit, diamond price increases by 109.19 units, keeping all other predictors constant.
- When color increases by 1 unit, diamond price increases by 272.92 units, keeping all other predictors constant.
- When clarity increases by 1 unit, diamond price increases by 436.44 units, keeping all other predictors constant.
- When y increases by 1 unit, diamond price increases by 1464.83 units, keeping all other predictors constant.
- As per model these five attributes that are most important attributes 'Carat', 'Cut', 'color','clarity' and width i.e. 'y' for predicting the price.

- There are also some negative co-efficient values, for instance, corresponding co-efficient (-1417.91) for 'x',(-711.23) for z and (-17.35) for table This implies, these are inversely proportional with diamond price.

Observation-7:

- On the given data set we can see the 'X' i.e. Length of the cubic zirconia in mm. having negative co-efficient. And the p value is less than 0.05, so can conclude that as higher the length of the stone is a lower profitable stones.

- Similarly for the 'z' variable having negative co-efficient i.e. -711.23. And the p value is less than 0.05, so we can conclude that as higher the 'z' of the stone is a lower profitable stones.

- Also we can see the 'y' width in mm having positive co-efficient. And the p value is less than 0.05, so we can conclude that higher the width of the stone is a higher profitable stones.

- Finally we can conclude that best 5 attributes that are most important are 'Carat', 'Cut', 'color','clarity' and width i.e. 'y' for predicting the price.

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

Inference:

- we can see that the from the linear plot, very strong correlation between the predicted y and actual y. But there are lots of spread. That indicates some kind noise present on the data set i.e. unexplained variances on the output.

Linear regression Performance Metrics:

- intercept for the model: -3171.950447307667
- R square on training data: 0.9311935886926559
- R square on testing data: 0.931543712584074
- RMSE on Training data: 907.1312415459143
- RMSE on Testing data: 911.8447345328436
- As the training data & testing data score are almost inline, we can conclude this model is a Right-Fit Model.

Impact of scaling:

- Now we can observe by applying z score the intercept became -5.87961525130473e-16. Earlier it was -3171.950447307667. The co-efficient has changed, the bias became nearly zero but the overall accuracy still same.

Multi collinearity:

- We can observe there are very strong multi collinearity present in the data set.

From statsmodels:

- We can see R-squared: 0.931 and Adj. R-squared: 0.931 are same. The overall P value is less than alpha.

- Finally we can conclude that Best 5 attributes that are most important are 'Carat', 'Cut', 'color','clarity' and width i.e. 'y' for predicting the price.

- When 'carat' increases by 1 unit, diamond price increases by 8901.94 units, keeping all other predictors constant.

- When 'cut' increases by 1 unit, diamond price increases by 109.19 units, keeping all other predictors constant.
- When 'color' increases by 1 unit, diamond price increases by 272.92 units, keeping all other predictors constant.
- When 'clarity' increases by 1 unit, diamond price increases by 436.44 units, keeping all other predictors constant.
- When 'y' increases by 1 unit, diamond price increases by 1464.83 units, keeping all other predictors constant.

- We can see the p value is showing 0.449 for depth variable, which is much greater than 0.05. That means this attribute is useless.
- There are also some negative co-efficient values, we can see the 'X' i.e. Length of the cubic zirconia in mm. having negative co-efficient -1417.9089. And the p value is less than 0.05, so can conclude that as higher the length of the stone is a lower profitable stones.
- Similarly for the 'z' variable having negative co-efficient i.e. -711.23. And the p value is less than 0.05, so we can conclude that as higher the 'z' of the stone is a lower profitable stones.

Recommendations:

- The Gem Stones company should consider the features 'Carat', 'Cut', 'color','clarity' and width i.e 'y' as most important for predicting the price. To distinguish between higher profitable stones and lower profitable stones so as to have better profit share.
- As we can see from the model Higher the width ('y') of the stone is higher the price.
- So the stones having higher width ('y') should consider in higher profitable stones.
- The 'Premium Cut' on Diamonds are the most Expensive, followed by 'Very Good' Cut, these should consider in higher profitable stones.
- The Diamonds clarity with 'VS1' &'VS2' are the most Expensive. So these two category also consider in higher profitable stones.
- As we see for 'X' i.e. Length of the stone, higher the length of the stone is lower the price.
- So higher the Length ('x') of the stone are lower is the profitability.
- Higher the 'z' i.e. Height of the stone is, lower the price. This is because if a Diamond's Height is too large Diamond will become 'Dark' in appearance because it will no longer return an Attractive amount of light. That is why
- Stones with higher 'z' is also are lower in profitability.

End of Problem-1

Problem 2: Logistic Regression and LDA

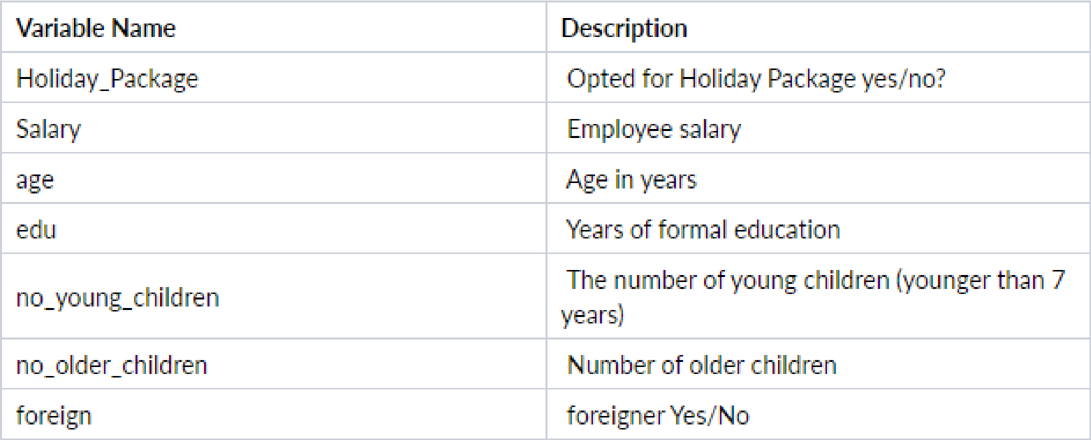
Problem statement:

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

Domain:

Tour and Travel

Data Dictionary:

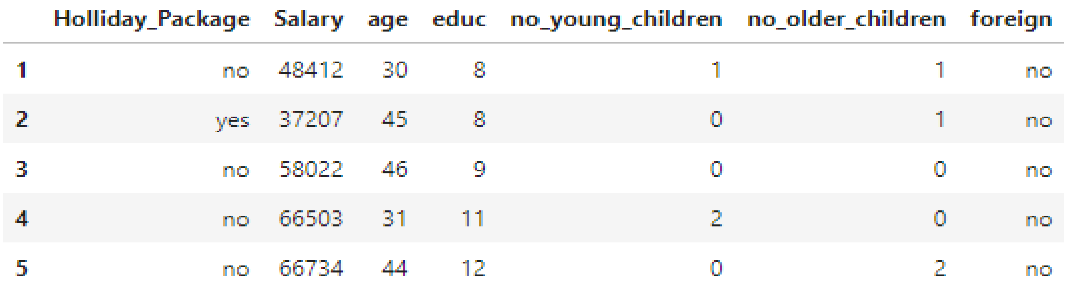


Variable Name	Description
Holiday_Package	Opted for Holiday Package yes/no?
Salary	Employee salary
age	Age in years
edu	Years of formal education
no_young_children	The number of young children (younger than 7 years)
no_older_children	Number of older children
foreign	foreigner Yes/No

Figure no: 27– Data dictionary of 'Holiday_Package' Dataset

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.

Read the dataset in to the notebook

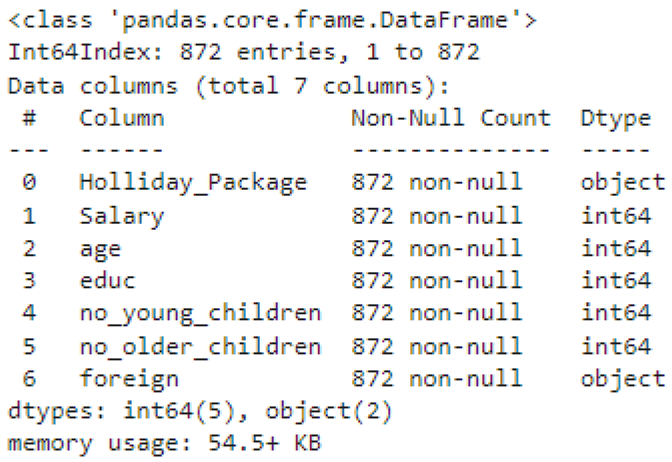


	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
1	no	48412	30	8	1	1	no
2	yes	37207	45	8	0	1	no
3	no	58022	46	9	0	0	no
4	no	66503	31	11	2	0	no
5	no	66734	44	12	0	2	no

Figure no: 28– Head of 'Holiday_Package' Dataset

Exploratory Data Analysis:

Checking the data types /information



```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 872 entries, 1 to 872
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: 54.5+ KB
```

Figure no: 29– Information of ‘Holiday_Package’ Dataset

Inference from the information of the data:

- The data set contains 872 observations of data and 7 features.
- Since non null count is same in every column variable except depth, there appears no null data.

Shape of the Dataset:

Rows – 872
Columns – 7

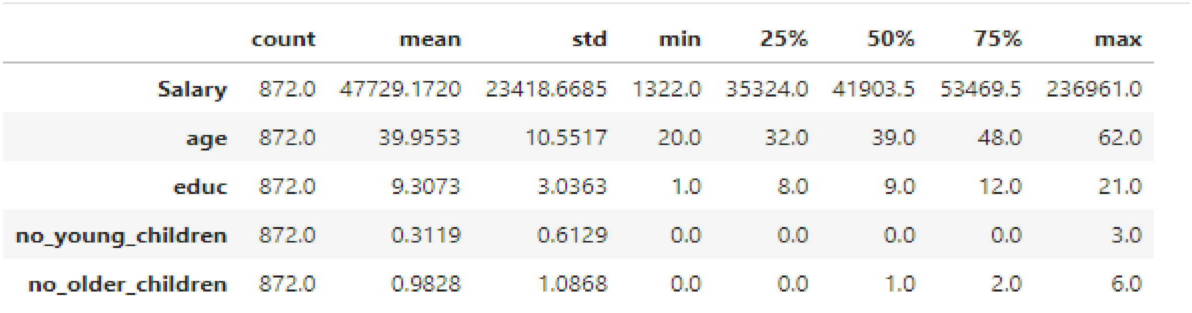
Checking for duplicates:

Number of duplicate rows = 0

Inference from the checking the duplicates:

- No duplicated data is present

Univariate Analysis:



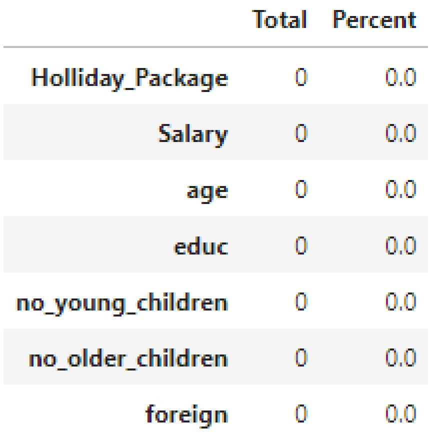
	count	mean	std	min	25%	50%	75%	max
Salary	872.0	47729.1720	23418.6685	1322.0	35324.0	41903.5	53469.5	236961.0
age	872.0	39.9553	10.5517	20.0	32.0	39.0	48.0	62.0
educ	872.0	9.3073	3.0363	1.0	8.0	9.0	12.0	21.0
no_young_children	872.0	0.3119	0.6129	0.0	0.0	0.0	0.0	3.0
no_older_children	872.0	0.9828	1.0868	0.0	0.0	1.0	2.0	6.0

Figure no: 30– Description of the dataset

From summary, we can see that:-

- Max salary (236K) is very high as compared to mean (47K) and median (42K). Hence it contains outlier
- Mean and median of age are approximately similar 39-40. It doesn't contains outlier.
- Education middle 50% of data lies in between 8 to 12 range with few outliers.
- Most employees have no of young children as 0.
- Most of the employees have 1 child who is older than 7 years
- All the columns are positively skewed except education

Checking for missing values:



	Total	Percent
Holliday_Package	0	0.0
Salary	0	0.0
age	0	0.0
educ	0	0.0
no_young_children	0	0.0
no_older_children	0	0.0
foreign	0	0.0

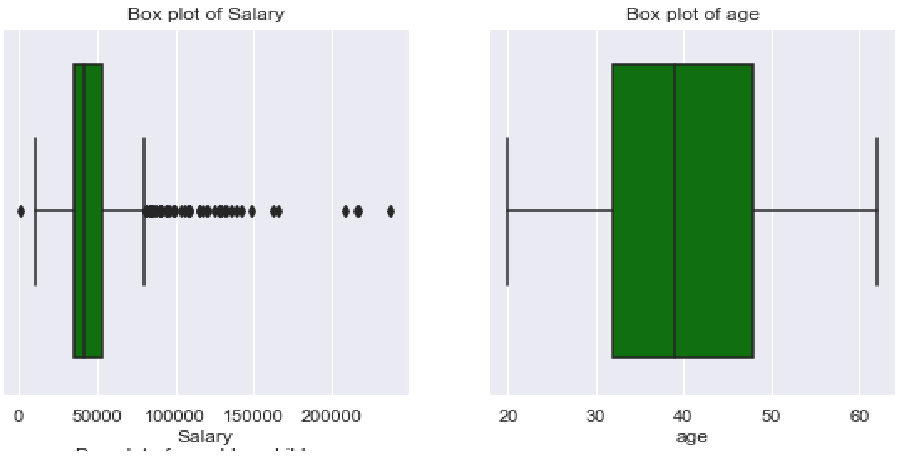
Figure no: 31– Missing values check result

Inference from the checking for missing values:

- We can confirm that there are no NULL values in the data

Check for Outliers:

Boxplot:



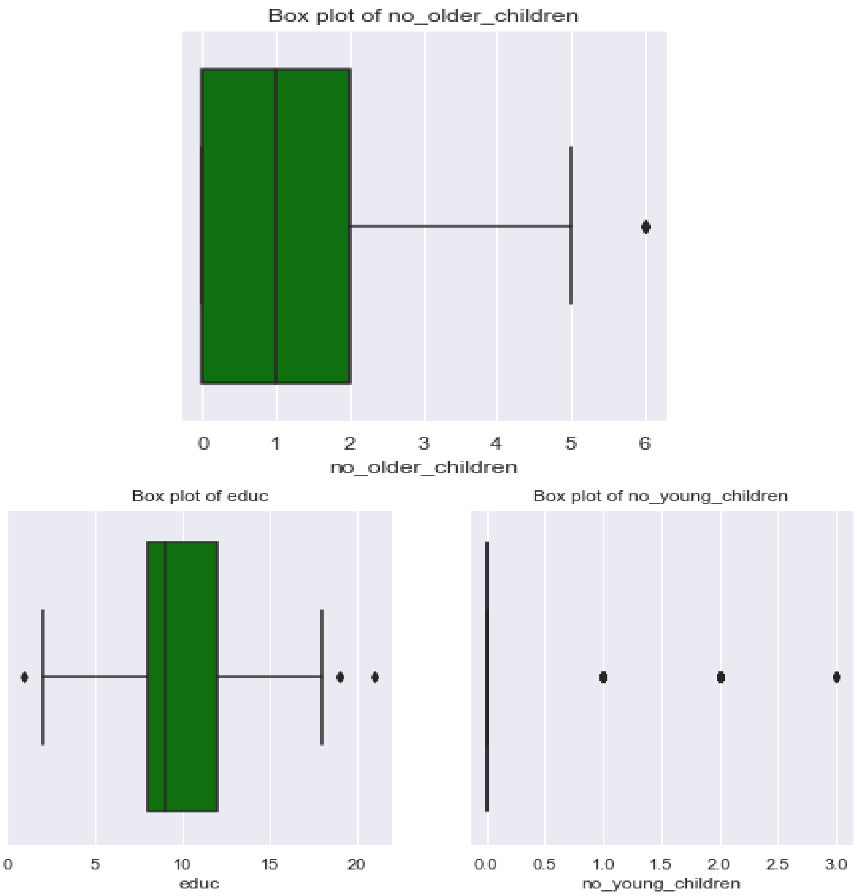
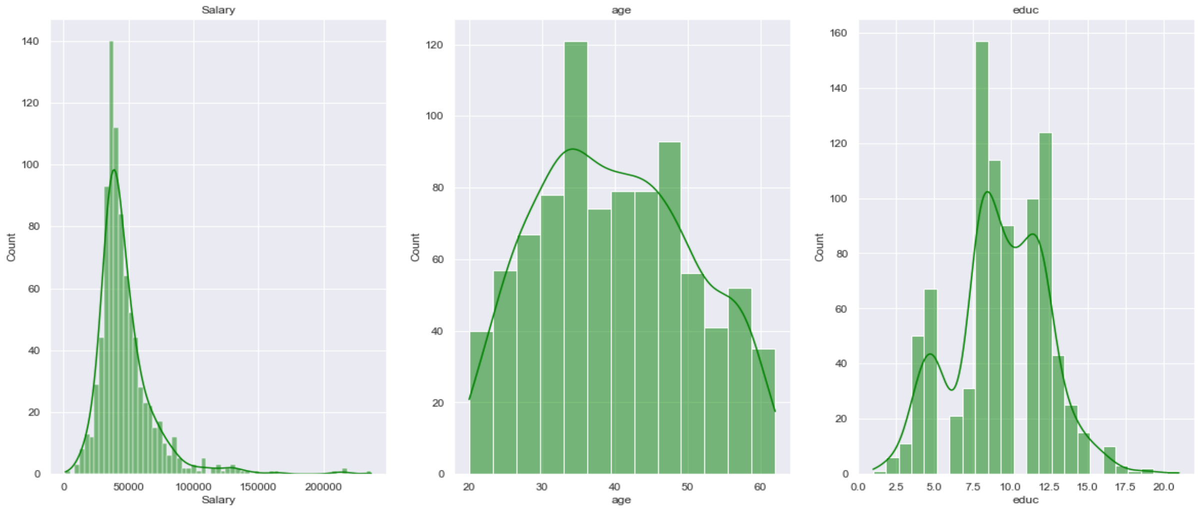


Figure no: 32– Boxplot view of dataset variables for outliers checking result

Inference from the check for outliers:

- As evident from above box plot, there are many outliers in salary column.
- Education, no of young children and old children columns have very few outliers which we can ignore

Histograms



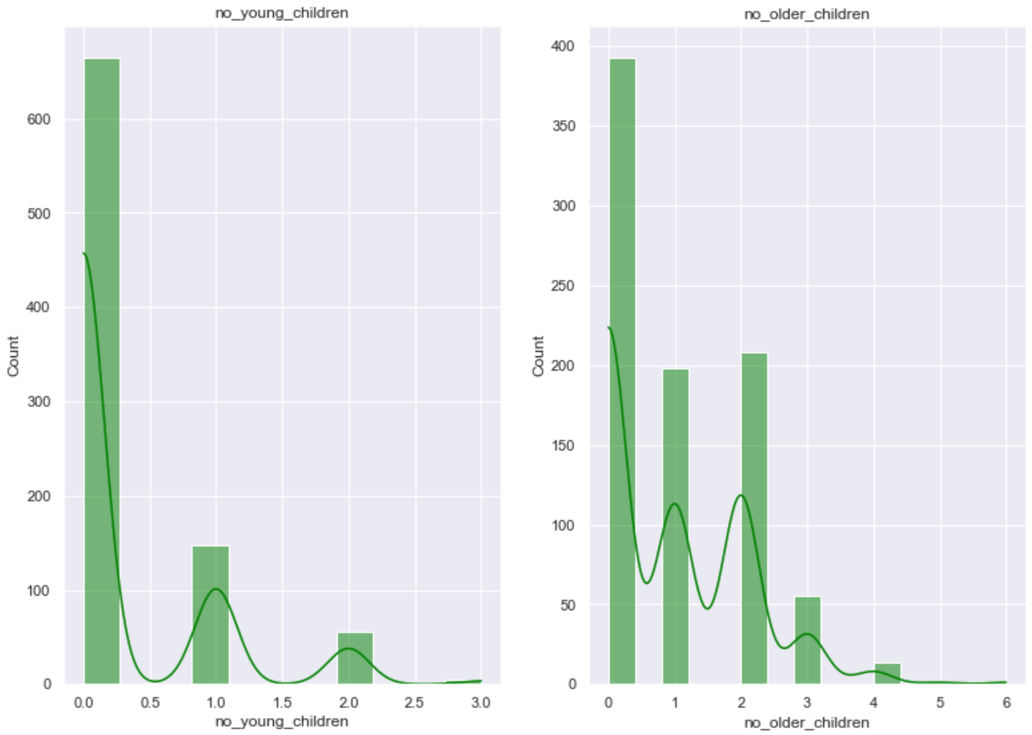


Figure no: 33– Histplot view of dataset variables for distribution checking

From histograms we can see that,

- Salary range is 0-100000 for most of the employees. However few employees are getting more salary causing skewness
- Age appears to be normally distributed
- Around 650 employees out of 872 have their young children as 0.
- Around 380 out of 872 employees have no of older children as 0
- Education middle 50% of data lies in between 8 to 12 range with few outliers.

Correcting Spelling error in Column names

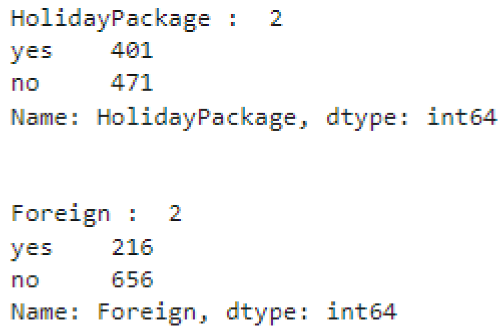
Before correction:

```
Index(['Holliday_Package', 'Salary', 'age', 'educ', 'no_young_
children',
       'no_older_children', 'foreign'],
      dtype='object')
```

After correction:

```
Index(['HolidayPackage', 'Salary', 'Age', 'Educ', 'No_young_ch
ildren',
       'No_older_children', 'Foreign'],
      dtype='object')
```

Checking the unique values for categorical variables,



```
HolidayPackage :  2
yes    401
no     471
Name: HolidayPackage, dtype: int64


Foreign :  2
yes    216
no     656
Name: Foreign, dtype: int64
```

Figure no: 34– Result of unique values for categorical variables

Bi-variate, and multivariate analysis:

Swarm Plots:

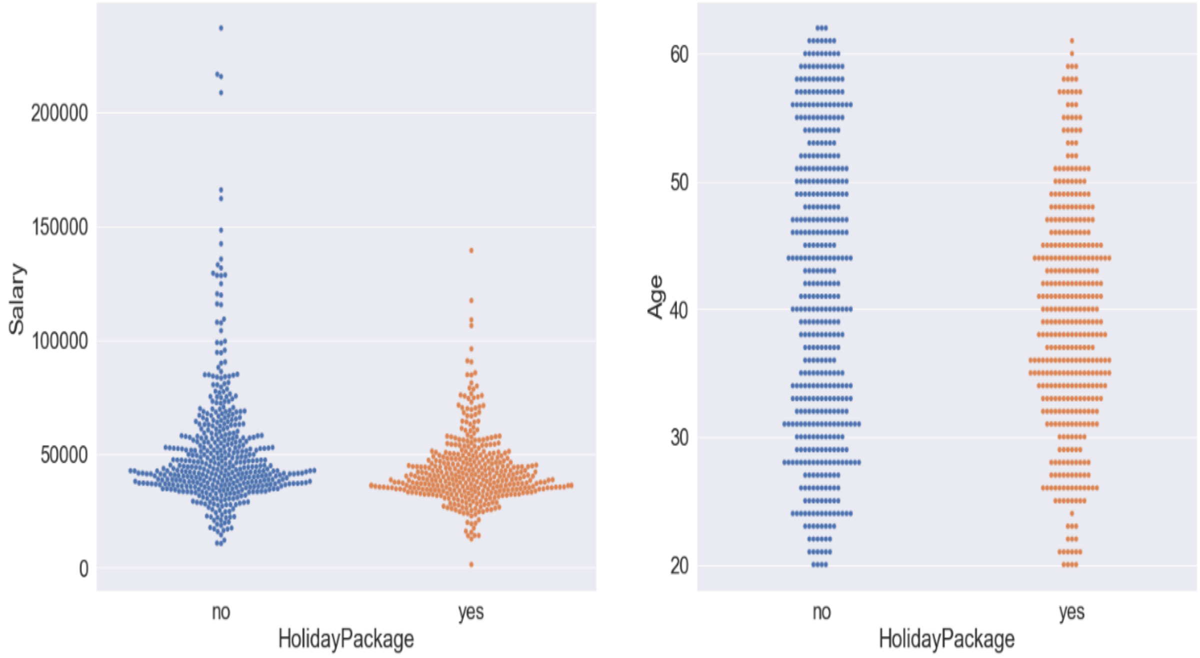


Figure no: 35– Swarmplot of HolidayPackage respect to Salary and Age

Inference from the swarm plot:

- We can see that as:- As Salary increases to the max value, employees count increases for the not opting for the holiday package.
- As Age increases beyond 50 level, less employees opt for the holiday package

Count Plot:

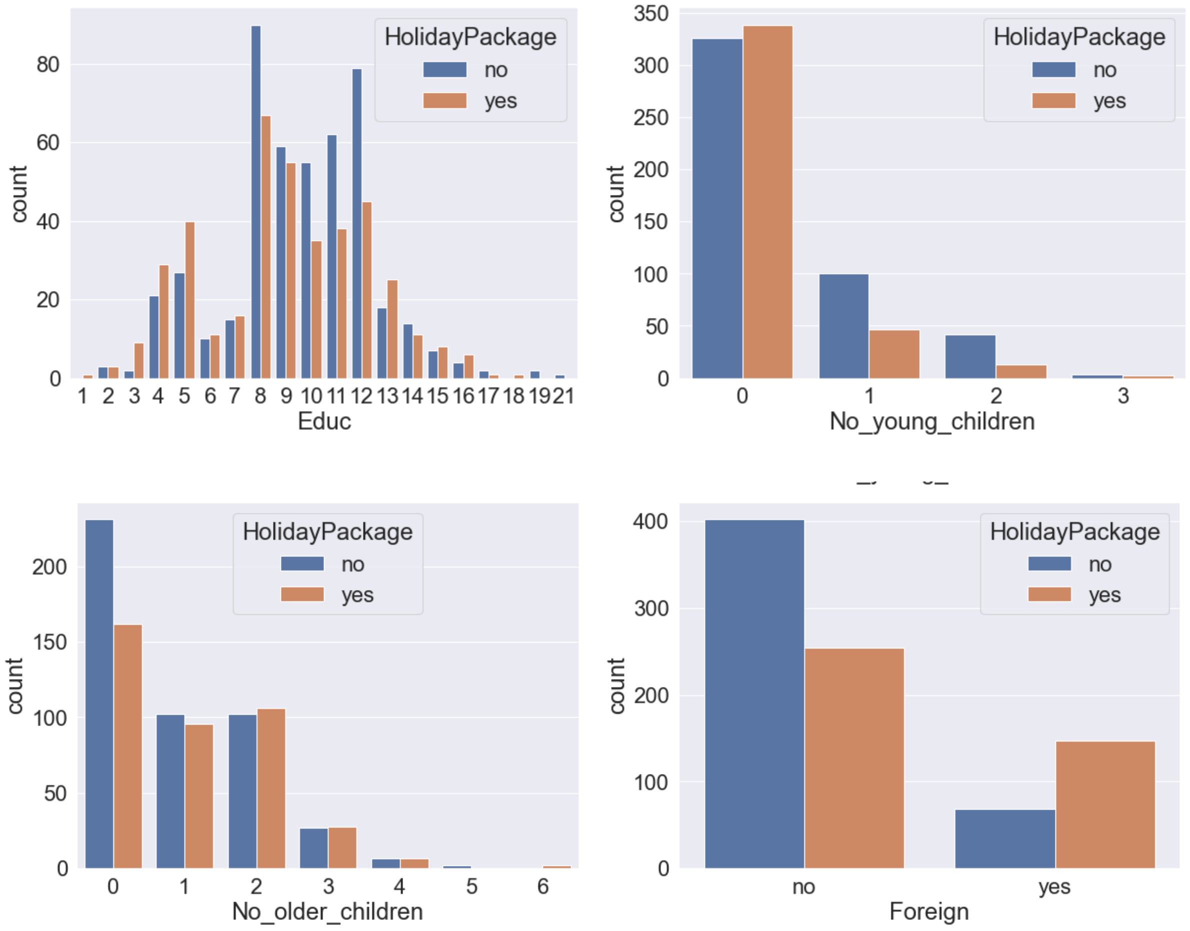
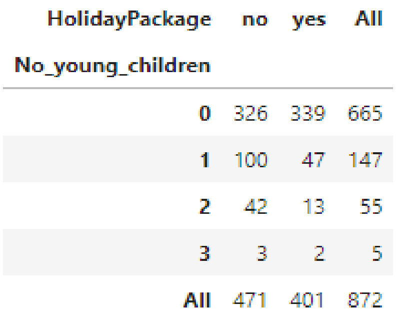


Figure no: 36– Count Plot view of dataset variables

Inference from the Count plot:

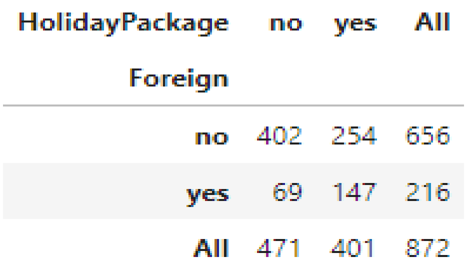
- More employees opt for Tours if their education level is 3,4,5,6,7,13,14,15,16
- Employees don't opt for tours if they have young child
- Older children count doesn't appears to have much impact on tour opted by employees or not
- Foreiger employees tends to opt more for the tour



HolidayPackage	no	yes	All
No_young_children			
0	326	339	665
1	100	47	147
2	42	13	55
3	3	2	5
All	471	401	872

We can see that around 24% of employees have one or more young child. Out of these employees, 70% ((100+42+3)/(147+55+5)) are not opting for tours.

Figure no: 37– Crosstab of HolidayPackage Vs No_young_children

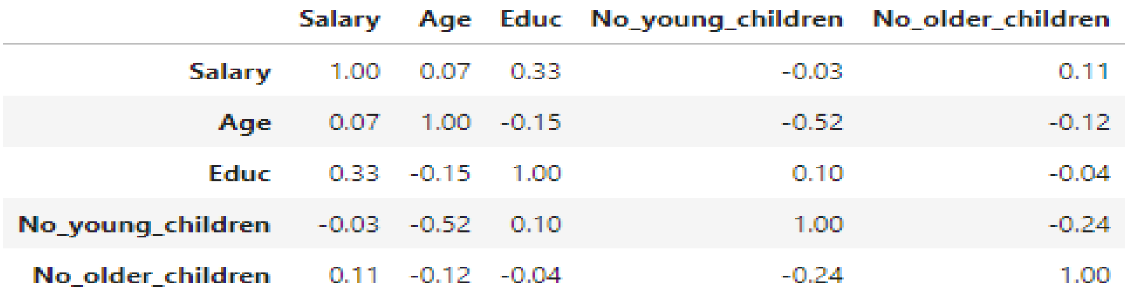


HolidayPackage	no	yes	All
Foreign			
no	402	254	656
yes	69	147	216
All	471	401	872

As per the data, we can say that 68% of foreign employees are opting for the tour packages.

Figure no: 38– Crosstab of HolidayPackage Vs Foreign

Correlation matrix:



	Salary	Age	Educ	No_young_children	No_older_children
Salary	1.00	0.07	0.33	-0.03	0.11
Age	0.07	1.00	-0.15	-0.52	-0.12
Educ	0.33	-0.15	1.00	0.10	-0.04
No_young_children	-0.03	-0.52	0.10	1.00	-0.24
No_older_children	0.11	-0.12	-0.04	-0.24	1.00

Figure no: 39– Correlation matrix of the dataset

Heat Map:

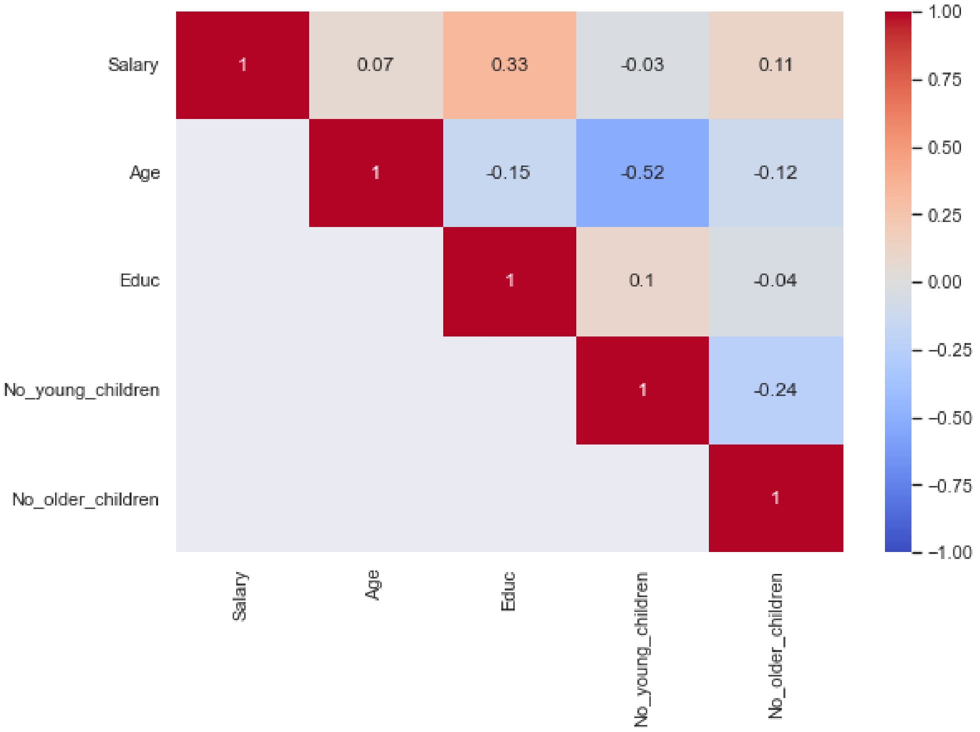


Figure no: 40– Heatmap view of the dataset

Inference from the Heat map:

- We can see in heatmap & correlation matrix that Salary has correlation with educ.
- Age is negatively correlated with No_young_children

Pairplot:

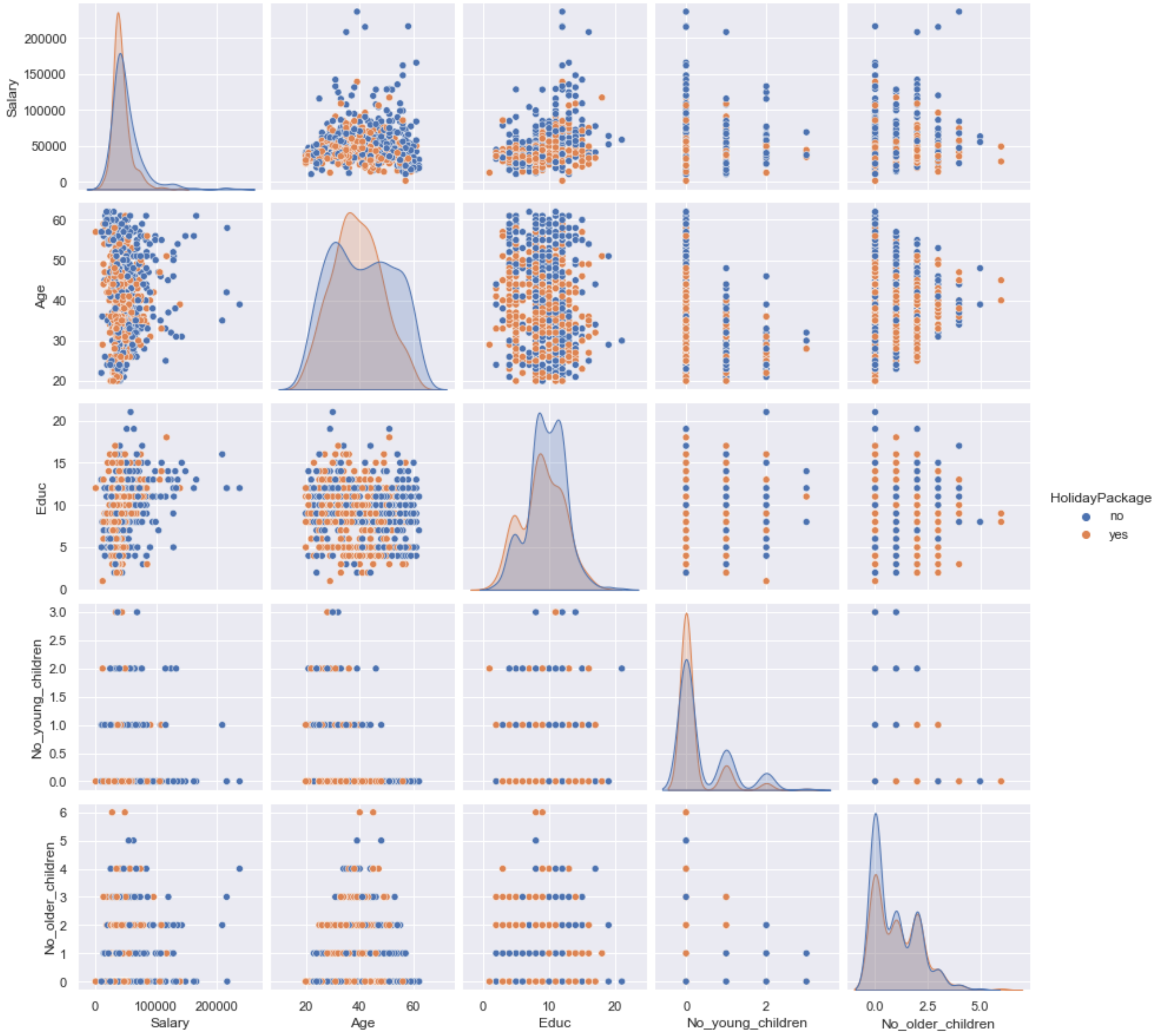


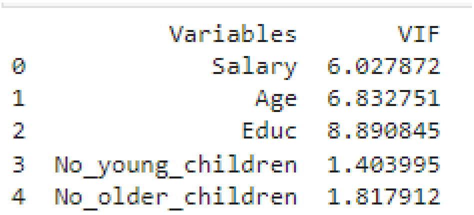
Figure no: 41– Pairplot view of the dataset

Inference from the Pairplot:

- As depicted in heat map of correlation matrix, we can see that no of young children negatively correlated with age.

- Salary is slightly correlated with Educ

VIF Checking for Multicollinearity



```
              Variables       VIF
0                Salary  6.027872
1                   Age  6.832751
2                  Educ  8.890845
3     No_young_children  1.403995
4     No_older_children  1.817912
```

Figure no: 42– VIF Multicollinearity checking result

Inference from the CIF Checking for Multicollinearity:

- We can see that VIF is greater than 5 for salary, age and education.
- However its value is less than 10. So dataset has some multicollinearity

Outlier treatment (flooring and capping)

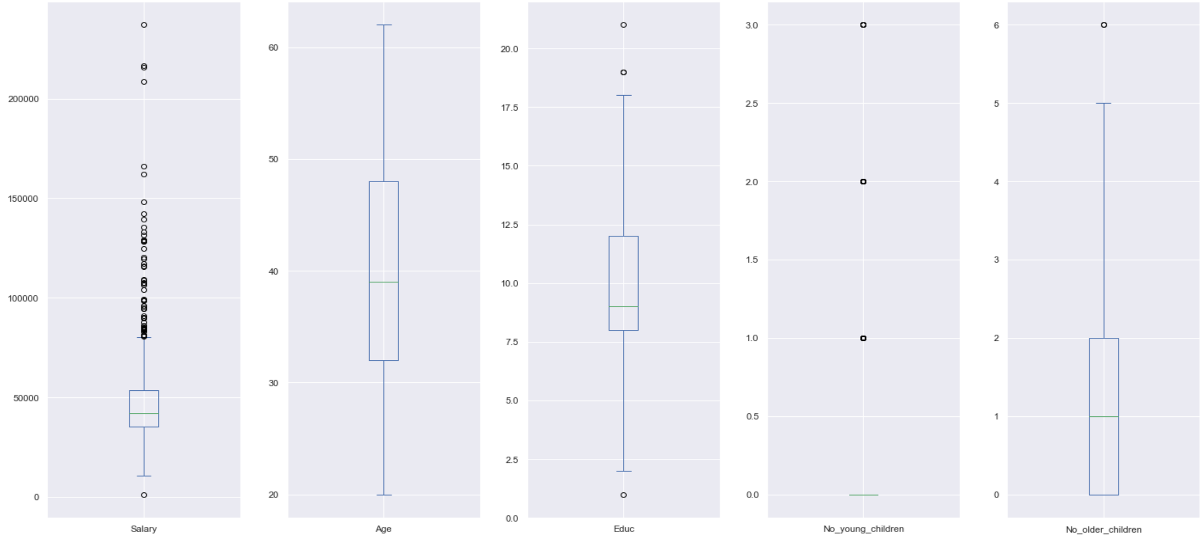


Figure no: 43– Outlier treatment in before stage

We are only doing outlier treatment for salary attributes as other columns have very less outliers and that are near lower and upper ranges

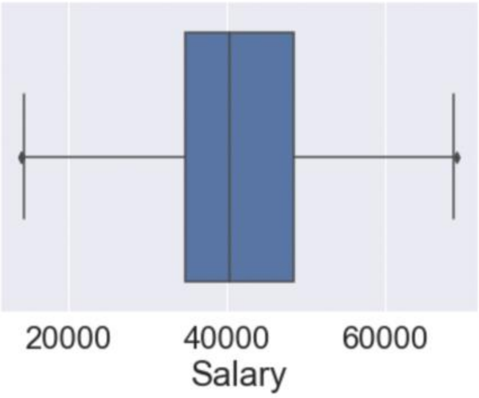


Figure no: 44– Outlier treatment of salary attribute

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).

Scaling:

- Scaling is not required for the logistic regression model. Hence it's not performed here

Encoding object data to Numerical:

- Since we have only two categorical inputs are there in the dataset, we have to converted them to 0 and 1.

'yes' value changed to 1

'no' value changed to 0

Train-Test Split:

- After extracting the Target column, data set is split into 70:30 ratio
- I.e. 70% of input observations into train dataset for building the model and 30% observation into test dataset for testing and validating the model.

X_train.shape - (570, 6)

X_test.shape - (245, 6)

y_train.shape - (570, 1)

y_test.shape - (245, 1)

Dataset (df2.shape) - (815, 7)

- We can see that data has been split into train (70%) and test (30%) successfully.

Logistic Regression Model:

Initially we built the logistic regression model using sklearn as per following hyper parameters,

solver='newton-cg',
max_iter=10000,
penalty='none',
verbose=**True**,
n_jobs=2

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.

Model Evaluation:

Accuracy score for train and test data se as shown below,

- Model score for training dataset 0.6526315789473685
- Model score for testing dataset 0.6857142857142857

AUC and ROC for the training data & test data:

- AUC for Train dataset: 0.722
- AUC for test dataset: 0.722

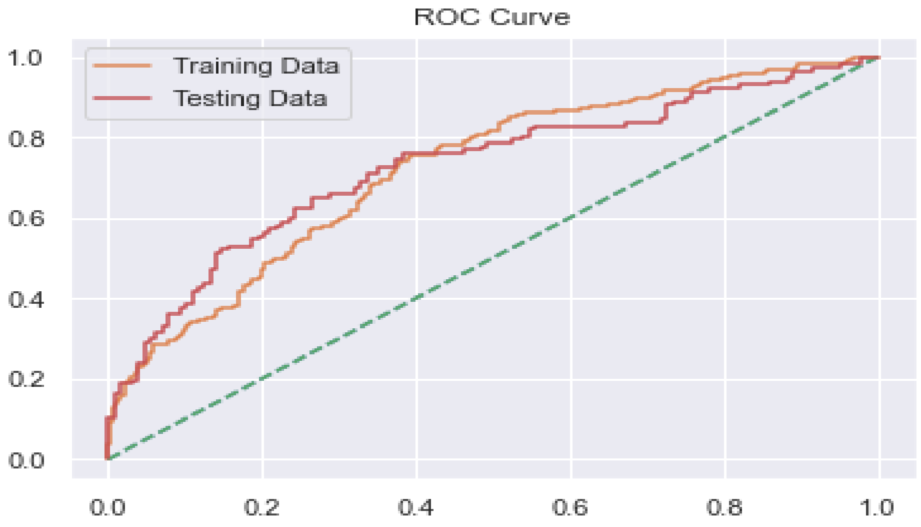


Figure no: 45– AUC and ROC for the training data & test data

Confusion Matrix for the training data and testing data:

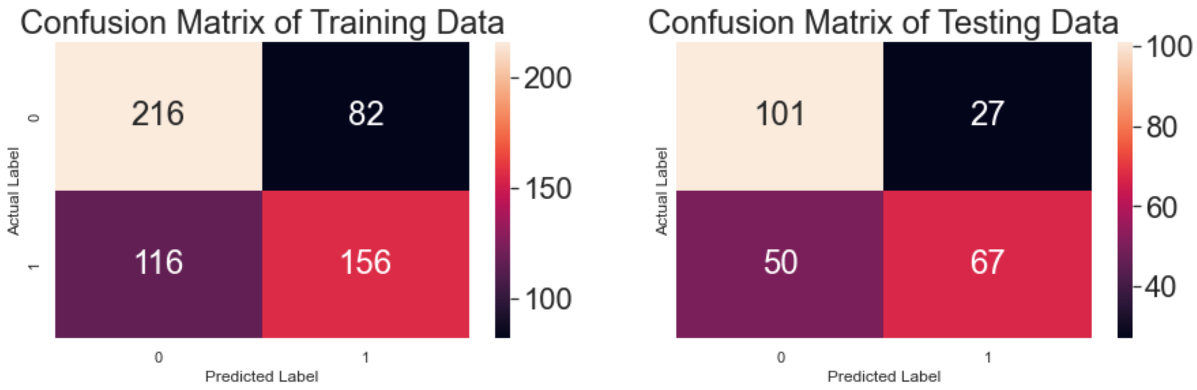
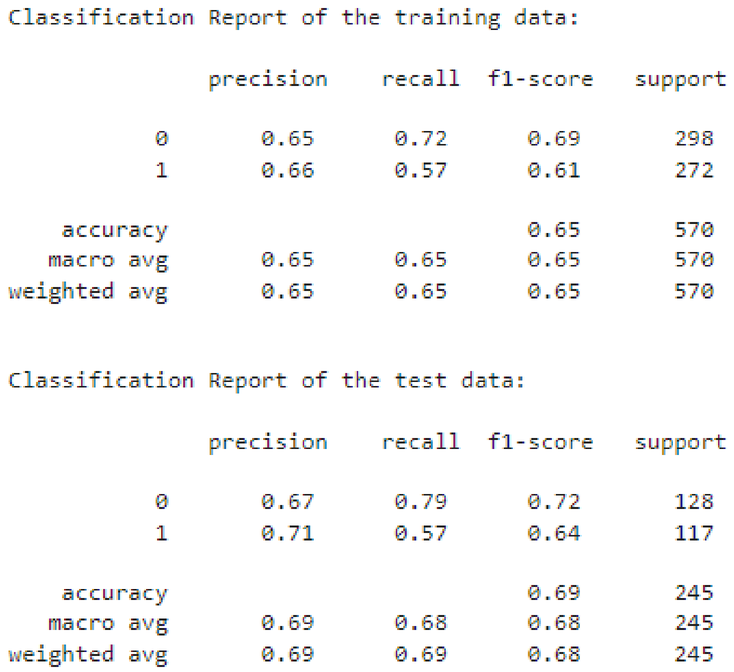


Figure no: 46– Confusion Matrix for the training data and testing data

Training Data and Test Data Classification Report Comparison:



```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.65      0.72      0.69       298
           1       0.66      0.57      0.61       272

    accuracy                           0.65       570
   macro avg       0.65      0.65      0.65       570
weighted avg       0.65      0.65      0.65       570


Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.67      0.79      0.72       128
           1       0.71      0.57      0.64       117

    accuracy                           0.69       245
   macro avg       0.69      0.68      0.68       245
weighted avg       0.69      0.69      0.68       245
```

Figure no: 47– Training Data and Test Data Classification Report

Applying GridSearchCV for Logistic Regression:

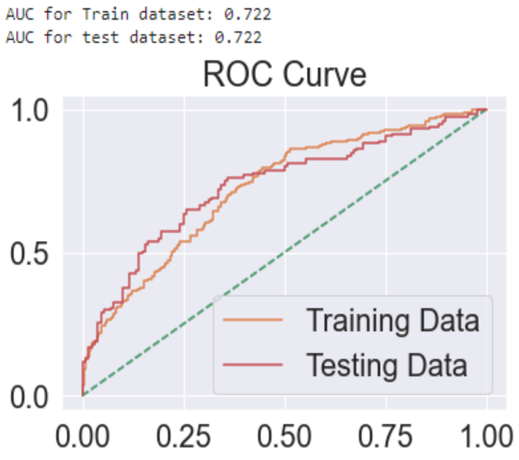
Showing best parameters for the grid search

{'penalty': 'l1', 'solver': 'liblinear', 'tol': 1e-05}

Model Evaluation:

Model score for training dataset 0.6491228070175439
Model score for training dataset 0.6857142857142857

AUC and ROC for the training data & test data:



```
AUC for Train dataset: 0.722
AUC for test dataset: 0.722
```

Figure no: 48– Training Data and Test Data Classification Report

Confusion Matrix for the training data and testing data:

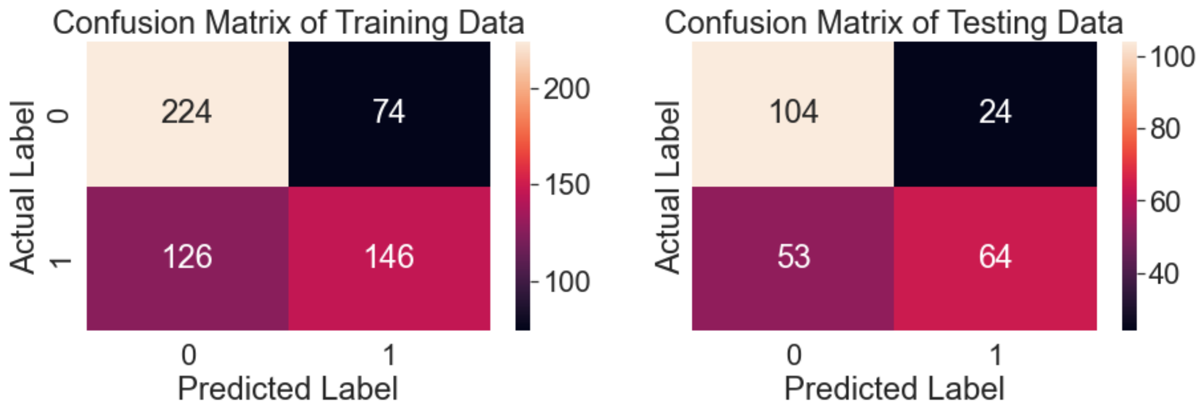
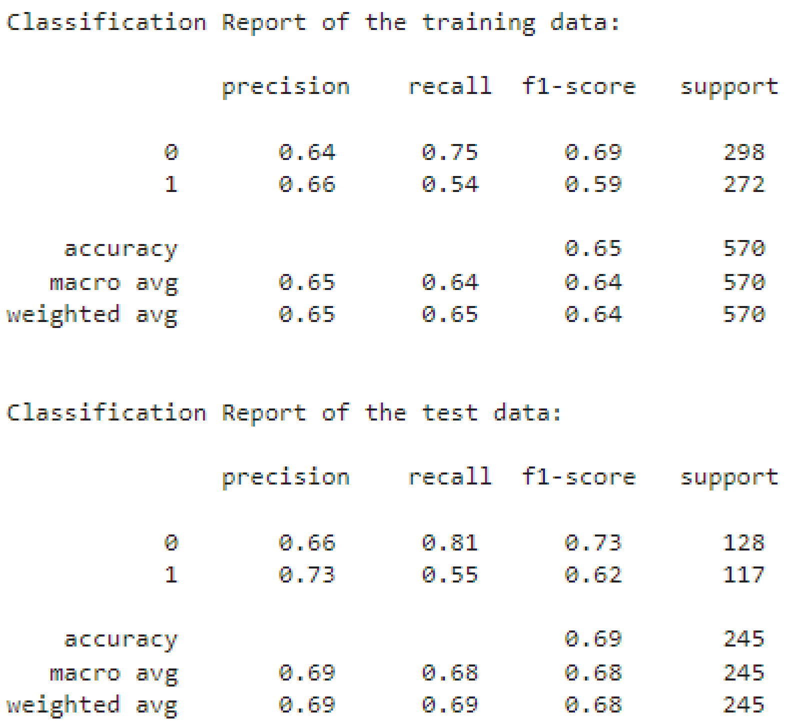


Figure no: 49– Confusion Matrix for the training data and testing data

Training Data and Test Data Classification Report Comparison:

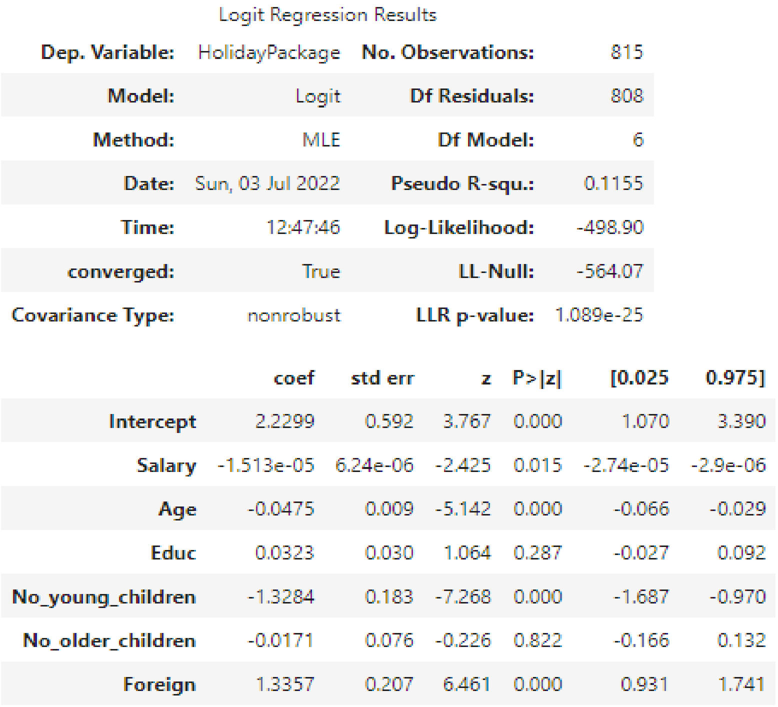


```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.64      0.75      0.69       298
           1       0.66      0.54      0.59       272

    accuracy                           0.65       570
   macro avg       0.65      0.64      0.64       570
weighted avg       0.65      0.65      0.64       570


Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.66      0.81      0.73       128
           1       0.73      0.55      0.62       117

    accuracy                           0.69       245
   macro avg       0.69      0.68      0.68       245
weighted avg       0.69      0.69      0.68       245
```

Figure no: 50– Training Data and Test Data Classification Report

Getting the equation:

- Using statsmodel, we can find the equation of log odds and we can find which coefficient has the more weightage in deciding the target response variable



Logit Regression Results

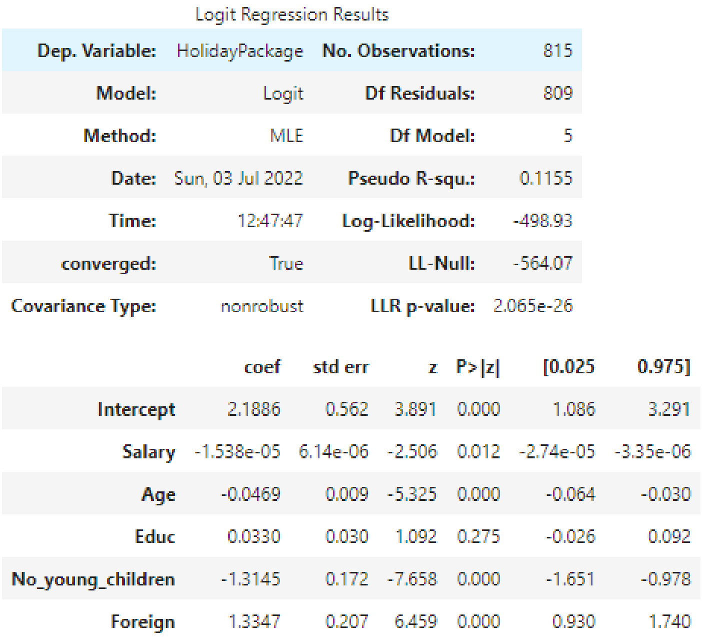
Dep. Variable:	HolidayPackage	**No. Observations:**	815
Model:	Logit	**Df Residuals:**	808
Method:	MLE	**Df Model:**	6
Date:	Sun, 03 Jul 2022	**Pseudo R-squ.:**	0.1155
Time:	12:47:46	**Log-Likelihood:**	-498.90
converged:	True	**LL-Null:**	-564.07
Covariance Type:	nonrobust	**LLR p-value:**	1.089e-25

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	2.2299	0.592	3.767	0.000	1.070	3.390
Salary	-1.513e-05	6.24e-06	-2.425	0.015	-2.74e-05	-2.9e-06
Age	-0.0475	0.009	-5.142	0.000	-0.066	-0.029
Educ	0.0323	0.030	1.064	0.287	-0.027	0.092
No_young_children	-1.3284	0.183	-7.268	0.000	-1.687	-0.970
No_older_children	-0.0171	0.076	-0.226	0.822	-0.166	0.132
Foreign	1.3357	0.207	6.461	0.000	0.931	1.741

Figure no: 51– Logit Regression Results

- We can see that the p value of No_older_children is the highest (.822) and it is greator than 0.05.

- Hence it confirms that No_older_children attribute has no impact on dependent variable HolidayPackage



Logit Regression Results

Dep. Variable:	HolidayPackage	**No. Observations:**	815
Model:	Logit	**Df Residuals:**	809
Method:	MLE	**Df Model:**	5
Date:	Sun, 03 Jul 2022	**Pseudo R-squ.:**	0.1155
Time:	12:47:47	**Log-Likelihood:**	-498.93
converged:	True	**LL-Null:**	-564.07
Covariance Type:	nonrobust	**LLR p-value:**	2.065e-26

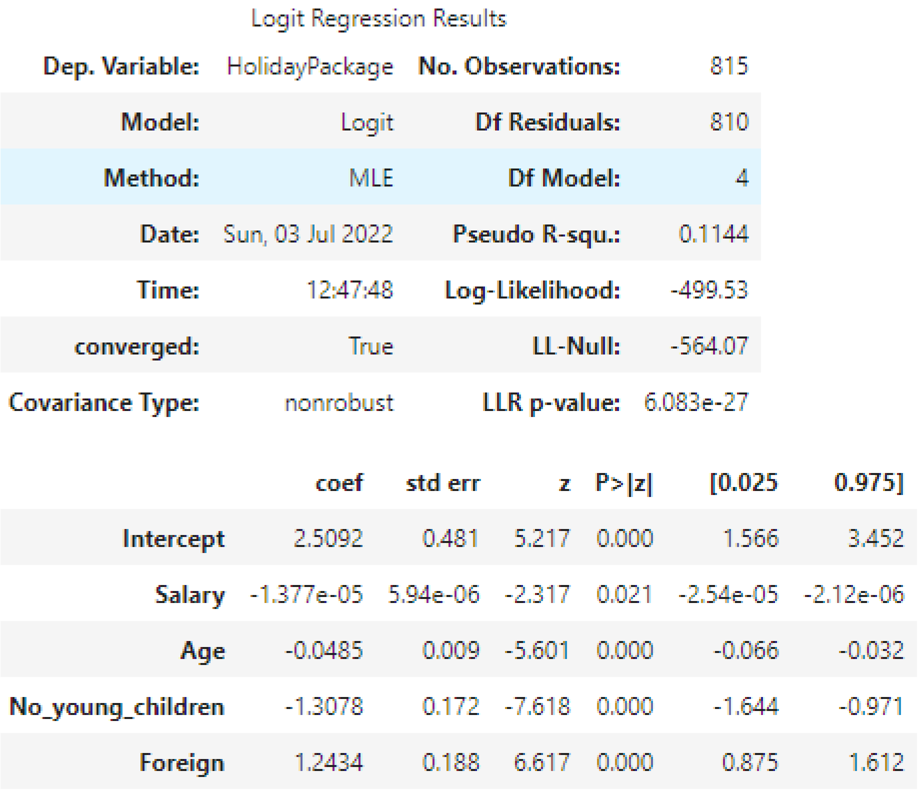
	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	2.1886	0.562	3.891	0.000	1.086	3.291
Salary	-1.538e-05	6.14e-06	-2.506	0.012	-2.74e-05	-3.35e-06
Age	-0.0469	0.009	-5.325	0.000	-0.064	-0.030
Educ	0.0330	0.030	1.092	0.275	-0.026	0.092
No_young_children	-1.3145	0.172	-7.658	0.000	-1.651	-0.978
Foreign	1.3347	0.207	6.459	0.000	0.930	1.740

Figure no: 52– Logit Regression Results

- We can see that the p value of Educ is the highest (.275) and it is greator than 0.05.

- Hence it confirms that Educ attribute has no impact on dependent variable HolidayPackage

Removing these 2 columns, we run the model again to get the following report,



Logit Regression Results

Dep. Variable:	HolidayPackage	**No. Observations:**	815
Model:	Logit	**Df Residuals:**	810
Method:	MLE	**Df Model:**	4
Date:	Sun, 03 Jul 2022	**Pseudo R-squ.:**	0.1144
Time:	12:47:48	**Log-Likelihood:**	-499.53
converged:	True	**LL-Null:**	-564.07
Covariance Type:	nonrobust	**LLR p-value:**	6.083e-27

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	2.5092	0.481	5.217	0.000	1.566	3.452
Salary	-1.377e-05	5.94e-06	-2.317	0.021	-2.54e-05	-2.12e-06
Age	-0.0485	0.009	-5.601	0.000	-0.066	-0.032
No_young_children	-1.3078	0.172	-7.618	0.000	-1.644	-0.971
Foreign	1.2434	0.188	6.617	0.000	0.875	1.612

Figure no: 53– Logit Regression Results

- Now all p values are less than 0.05. Hence all these attributes and their coefficients have importance in deciding the target variable HolidayPackage.
- Also we can see that coefficients value is highest for No_young_children followed by foreign, Age and salary
- Salary coefficient value is very low i.e. -00001377. So its impact is almost 0 on dependent variable

Logistics Regression Conclusion:

Train Data:

- AUC: 72%
- Accuracy: 65%
- Precision: 66%
- f1-Score: 59%
- Recall: 54%

Test Data:

- AUC: 72%
- Accuracy: 69%
- Precision: 73%
- f1-Score: 62%
- Recall: 55%

Table No: 2– Logistics Regression Conclusion

Train and Test dataset have similar statistics, hence model is giving similar result for test and train data set...

- With accuracy of 69% and recall rate of 55%, model is only able to predict 55% of total tours which were actually claimed as claimed.
- Precision is 73% of test data which means, out of total employees predicted by model as opt for tour, 73% employees actually opted for the tour
- F1-score is the harmonic mean of precision and recall, it takes into the effect of both the scores and this value is low if any of these 2 value is low.
- Since we are building a model to predict if whether employee will opt for tour or not, for practical purposes, we will be more interested in correctly classifying 1 (employees opting for tour) than 0(employees not opting for tour).
- If a employee not opting for tour is incorrectly predicted to be "opted for tour" by the model, then the impact on cost for the travel company would be bare minimum. But if an employee opted for tour is incorrectly predicted to be not opted by the model, then the cost impact would be very high for the tour and travel company. Its a loss of potential lead for the company. Hence recall rate (actual data point identified as True by model) is very important in this scenario.

As Recall rate of test dataset is very poor around 52% thus this doesn't looks good enough for classification Logistic regression equation is as shown below :-

- Log (odd) = (2.51) + (-0.0) Salary + (-0.05) Age + (-1.31) No_young_children + (1.24) Foreign
- We can see that salary coefficient is very small, this it can be removed. So our equation would become :-
- Log (odd) = (2.51) + (-0.05) Age + (-1.31) No_young_children + (1.214) * Foreign
- Most important attribute here is No of young children followed by Foreign and age

LDA Model:

We have built the model using LinearDiscriminantAnalysis method

Model Evaluation as follows:

Training Data and Test Data Confusion Matrix Comparison:

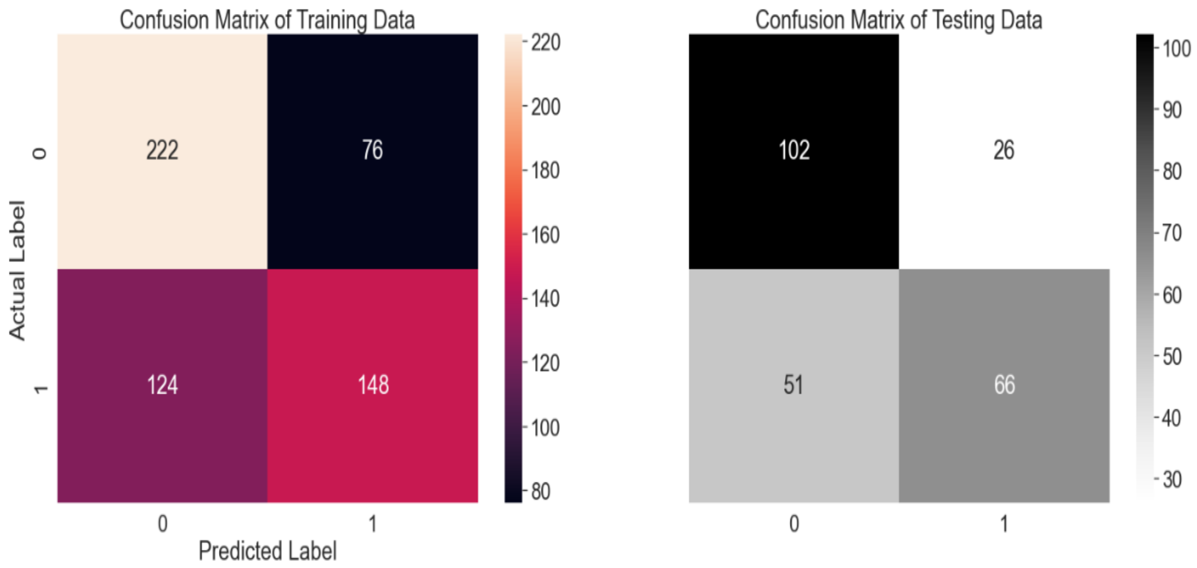
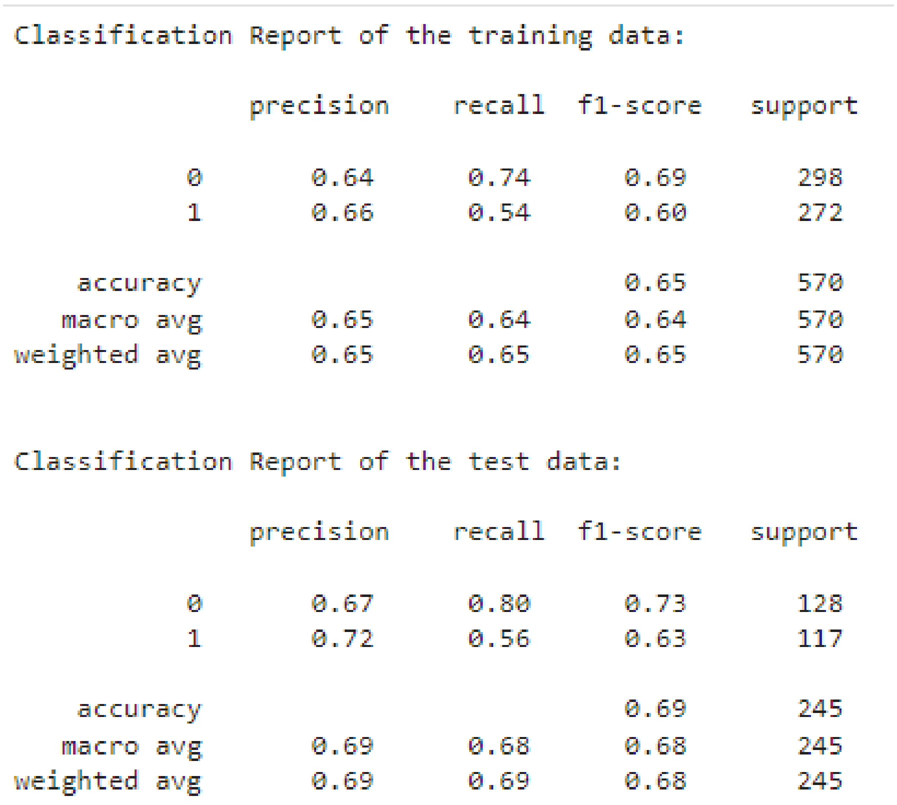


Figure no: 54– Training Data and Test Data Confusion Matrix Comparison

Training Data and Test Data Classification Report Comparison:



```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.64      0.74      0.69       298
           1       0.66      0.54      0.60       272

    accuracy                           0.65       570
   macro avg       0.65      0.64      0.64       570
weighted avg       0.65      0.65      0.65       570


Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.67      0.80      0.73       128
           1       0.72      0.56      0.63       117

    accuracy                           0.69       245
   macro avg       0.69      0.68      0.68       245
weighted avg       0.69      0.69      0.68       245
```

Figure no: 55– Classification Report of the training data

Probability prediction for the training and test data:

AUC and ROC for the training and testing data

- AUC for the Training Data: 0.721
- AUC for the Test Data: 0.730

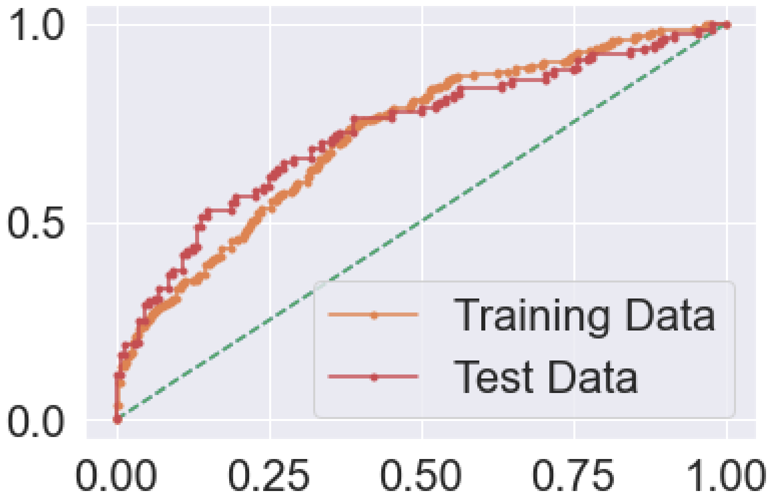


Figure no: 56– AUC and ROC for the training and testing data

LDA Conclusion:

Train Data:

- AUC: 72%
- Accuracy: 65%
- Precision: 66%
- f1-Score: 60%
- Recall: 54%

Test Data:

- AUC: 73%
- Accuracy: 69%
- Precision: 72%
- f1-Score: 63%
- Recall: 56%

Table No: 3– LDA Conclusion

Train and Test dataset have similar statistics, hence model is giving similar result for test and train data set…

- With accuracy of 73% and recall rate of 56%, model is only able to predict 54% of total tours which were actually claimed as claimed.

- Precision is 72% of test data which means, out of total employees predicted by model as opt for tour, 72% employees actually opted for the tour

- F1-score is the harmonic mean of precision and recall, it takes into the effect of both the scores and this value is low if any of these 2 value is low.

- Since we are building a model to predict if whether employee will opt for tour or not, for practical purposes, we will be more interested in correctly classifying 1 (employees opting for tour) than 0(employees not opting for tour).

- If a employee not opting for tour is incorrectly predicted to be "opted for tour" by the model, then the impact on cost for the travel company would be bare minimum. But if an employee opted for tour is incorrectly predicted to be not opted by the model, then the cost impact would be very high for the tour and travel company. It's a loss of potential lead for the company. Hence recall rate (actual data point identified as True by model) is very important in this scenario.

- As Recall rate of test dataset is very poor around 56% thus this doesn't looks good enough for classification

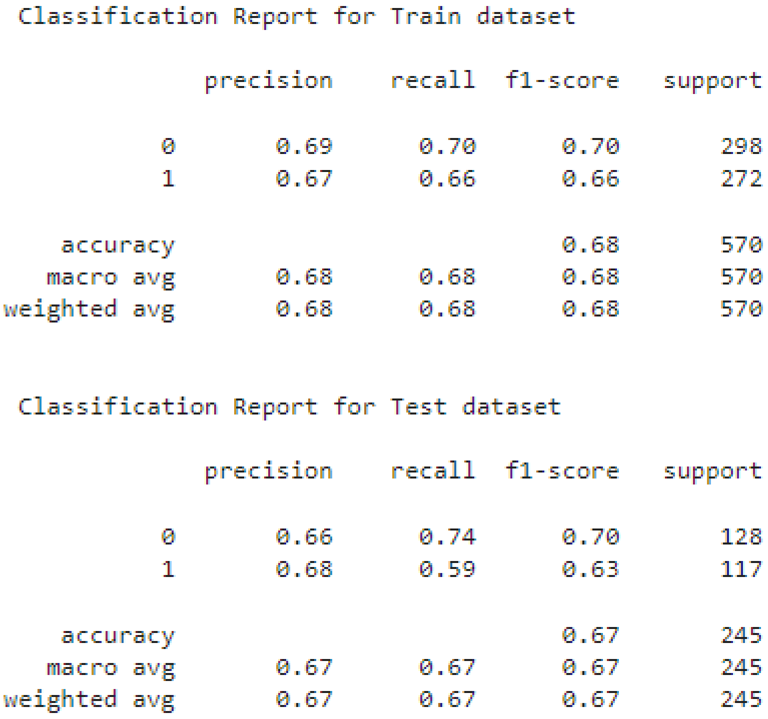
Running other Classification models:

Train and test dataframe have been scaled now

Make 4 models using ANN, Decision Tree, Random Forest, and Linear Regression

- Check Train and Test AUC
- Check Train and Test Scores

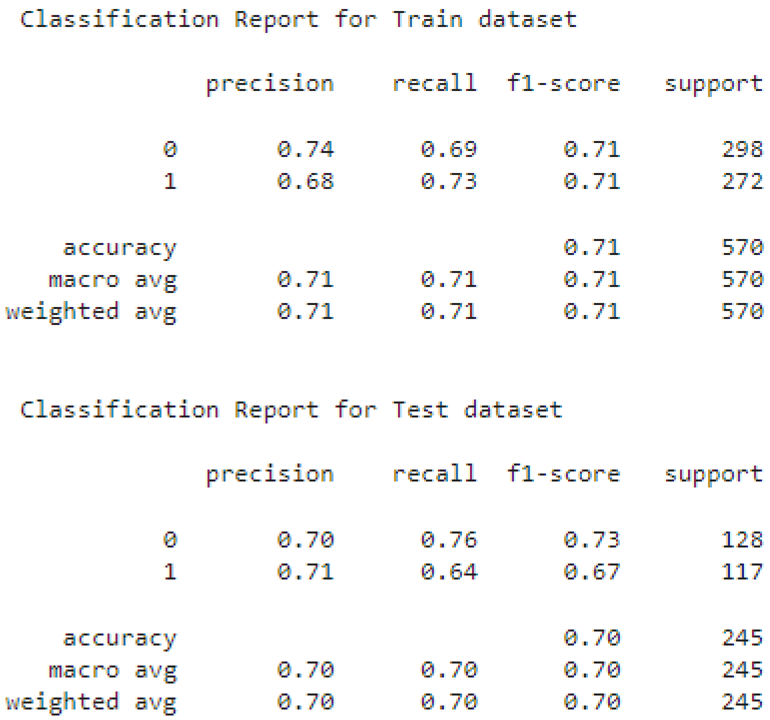
Running Grid search for Decision Tree:



```
 Classification Report for Train dataset

              precision    recall  f1-score   support

           0       0.69      0.70      0.70       298
           1       0.67      0.66      0.66       272

    accuracy                           0.68       570
   macro avg       0.68      0.68      0.68       570
weighted avg       0.68      0.68      0.68       570


 Classification Report for Test dataset

              precision    recall  f1-score   support

           0       0.66      0.74      0.70       128
           1       0.68      0.59      0.63       117

    accuracy                           0.67       245
   macro avg       0.67      0.67      0.67       245
weighted avg       0.67      0.67      0.67       245
```

Figure no: 57– Classification Report for Decision Tree

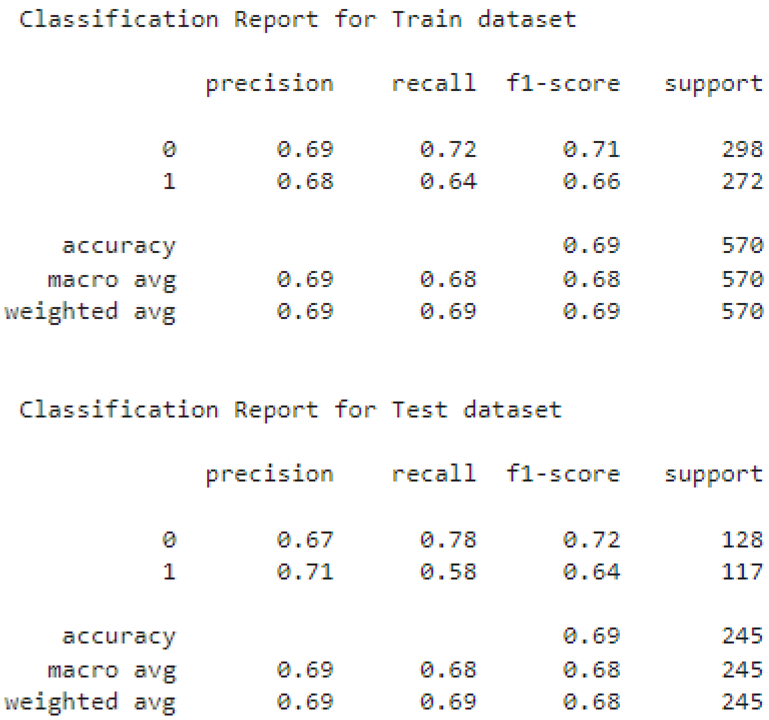
Running Grid search for Random Forest:



```
 Classification Report for Train dataset

              precision    recall  f1-score   support

           0       0.74      0.69      0.71       298
           1       0.68      0.73      0.71       272

    accuracy                           0.71       570
   macro avg       0.71      0.71      0.71       570
weighted avg       0.71      0.71      0.71       570


 Classification Report for Test dataset

              precision    recall  f1-score   support

           0       0.70      0.76      0.73       128
           1       0.71      0.64      0.67       117

    accuracy                           0.70       245
   macro avg       0.70      0.70      0.70       245
weighted avg       0.70      0.70      0.70       245
```

Figure no: 58– Classification Report for Random Forest

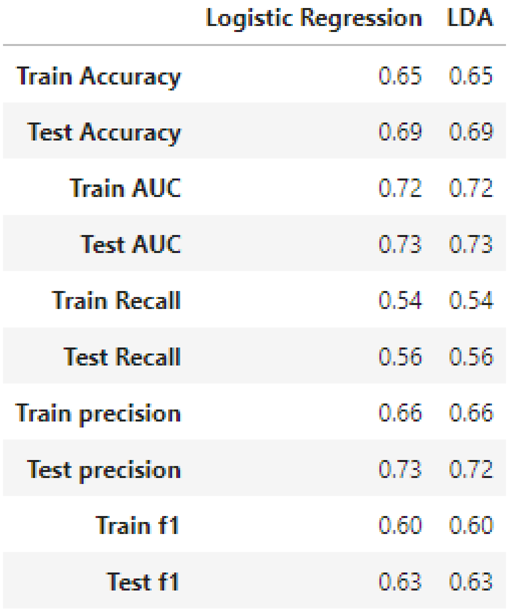
Running Grid search for ANN:



```
 Classification Report for Train dataset

              precision    recall  f1-score   support

           0       0.69      0.72      0.71       298
           1       0.68      0.64      0.66       272

    accuracy                           0.69       570
   macro avg       0.69      0.68      0.68       570
weighted avg       0.69      0.69      0.69       570


 Classification Report for Test dataset

              precision    recall  f1-score   support

           0       0.67      0.78      0.72       128
           1       0.71      0.58      0.64       117

    accuracy                           0.69       245
   macro avg       0.69      0.68      0.68       245
weighted avg       0.69      0.69      0.68       245
```

Figure no: 59– Classification Report for ANN

Comparing Logistic Regression Vs LDA:



	Logistic Regression	LDA
Train Accuracy	0.65	0.65
Test Accuracy	0.69	0.69
Train AUC	0.72	0.72
Test AUC	0.73	0.73
Train Recall	0.54	0.54
Test Recall	0.56	0.56
Train precision	0.66	0.66
Test precision	0.73	0.72
Train f1	0.60	0.60
Test f1	0.63	0.63

Figure no: 60– Comparing Logistic Regression Vs LDA

AUC and ROC for the training data and test data:

- AUC for LogR is: 0.73
- AUC for LDA is: 0.73
- AUC for DT is: 0.73
- AUC for RF is: 0.74

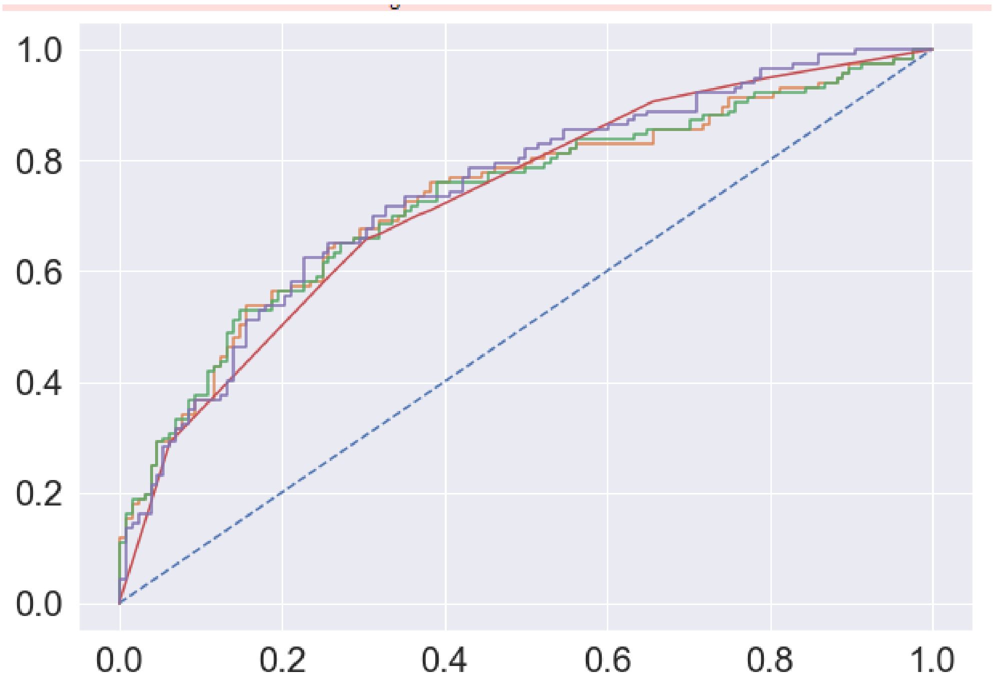
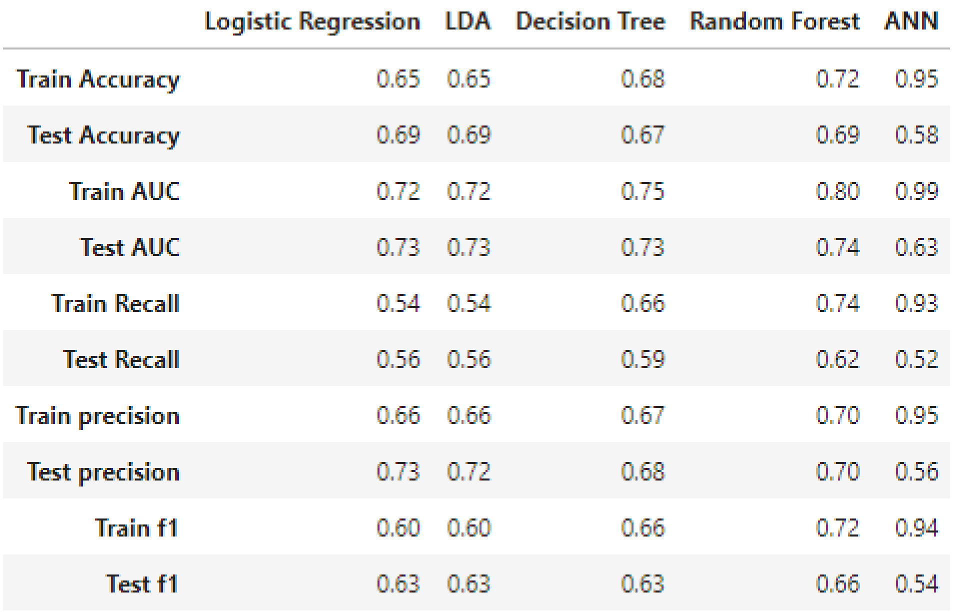


Figure no: 61– AUC and ROC for the training data and test data

Summary of the performance metrics of all the models is as shown below,



	Logistic Regression	LDA	Decision Tree	Random Forest	ANN
Train Accuracy	0.65	0.65	0.68	0.72	0.95
Test Accuracy	0.69	0.69	0.67	0.69	0.58
Train AUC	0.72	0.72	0.75	0.80	0.99
Test AUC	0.73	0.73	0.73	0.74	0.63
Train Recall	0.54	0.54	0.66	0.74	0.93
Test Recall	0.56	0.56	0.59	0.62	0.52
Train precision	0.66	0.66	0.67	0.70	0.95
Test precision	0.73	0.72	0.68	0.70	0.56
Train f1	0.60	0.60	0.66	0.72	0.94
Test f1	0.63	0.63	0.63	0.66	0.54

Figure no: 62– Summary of the performance metrics of all the models

- On comparing all the models, it looks like that no model is over-fitting/under fitting.
- All models test and train score are comparable and within 5-6% range.
- We can see that all models are giving similar results with not much of difference in accuracy.
- Random Forest and Artificial Neural Network gives better f1 score and better recall rate as compared to the logistics/LDA
- Among all these models we will go for Artificial Neuro Network MLP classifier as its test f1 score and test accuracy is the highest. </b>

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

Business Recommendations:-

- We have run five different models (Logistic Regression/ Linear Discriminant Analysis/ CART/ Random Forest/ Artificial Neuro Network) for predicting whether an employee is opting for holiday package or not.
- Based on the reports and analysis done it was found that all models were not good enough for classification as accuracy coming out is 66%.

So our recommendation to the business is as shown below:-

- In order to further improve the predictive model results for finding the employees which will opt for tour in future more accurately, more data sample is required for analysis.
- Current model is useful to predict when tours are not getting claimed with more than 70% accuracy.
- Most important attribute here is No of young children of an employee followed by Foreigner column and lastly the age.

- As seen in EDA, 70% of foreign employees are opting for the tour packages. So the travel company should make dedicated tours for these foreigners keeping in mind which places/areas that these foreigners would like to travel. If these customers are satisfied then when they will travel back to their country they will refer more of their friends/family members for tours. This way company can retain and increase its customer base.

- Currently 18% of employees have 1 or more young child. It was found during EDA that out of these employees 82% are not opting for the tours. So the travel company should make dedicated tour for the employees who have young child and provide them with some extra child care benefits (like play area for child, child food, medical facilities etc.) so as to lure these employees.

- Old age employees (age greater than 50) opt less for the tours. So company can provide dedicated tour plans for old aged senior employees.

- As per the analysis, Salary of the employees is not an important attribute in deciding that whether employee will opt for tour or not. So the travel company should not focus on salary of the employee. May be salary can decide which tour (economical or lavish) that particular employee will be interested in but he/she will opt for some tour irrespective of his/her salary.

End of Problem-2