Project On Predictive Modelling

Problem 1: Linear Regression

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

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

The csv file was imported and converted into a data frame and first few records are being displayed.

	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

We have checked the info of the dataset and found out that variables are of float,int and object type

```
<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
```

The dataset is described below.

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Unnamed: 0	26967	NaN	NaN	NaN	13484	7784.85	1	6742.5	13484	20225.5	26967
carat	26967	NaN	NaN	NaN	0.798375	0.477745	0.2	0.4	0.7	1.05	4.5
cut	26967	5	Ideal	10816	NaN	NaN	NaN	NaN	NaN	NaN	NaN
color	26967	7	G	5661	NaN	NaN	NaN	NaN	NaN	NaN	NaN
clarity	26967	8	SI1	6571	NaN	NaN	NaN	NaN	NaN	NaN	NaN
depth	26270	NaN	NaN	NaN	61.7451	1.41286	50.8	61	61.8	62.5	73.6
table	26967	NaN	NaN	NaN	57.4561	2.23207	49	56	57	59	79
x	26967	NaN	NaN	NaN	5.72985	1.12852	0	4.71	5.69	6.55	10.23
y	26967	NaN	NaN	NaN	5.73357	1.16606	0	4.71	5.71	6.54	58.9
z	26967	NaN	NaN	NaN	3.53806	0.720624	0	2.9	3.52	4.04	31.8
price	26967	NaN	NaN	NaN	3939.52	4024.86	326	945	2375	5360	18818

There are null values in a column:depth

```
Out[30]: Unnamed: 0      0
         carat           0
         cut             0
         color           0
         clarity         0
         depth         697
         table           0
         x               0
         y               0
         z               0
         price           0
         dtype: int64
```

Since Unnamed column is not required, we would drop this column for now

Out[33]:

	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

```
data_df.duplicated().sum()
```

executed in 40ms, finished 20:25:02 2021-04-10

```
34
```

There were 34 duplicate rows which have been deleted.

There are columns having 0 values and it needs to be treated since length cannot be of 0 size.The values have been replaced by the mean values of the column.

```
carat       0
cut         0
color       0
clarity     0
depth       0
table       0
x           2
y           2
z           8
price       0
dtype: int64
```

Finding out Unique values in Object type columns.

Find out unique values in each categorical column

In [18]:
```
data_df['cut'].unique()
```
executed in 340ms, finished 13:20:51 2021-04-14

Out[18]:
```
array(['Ideal', 'Premium', 'Very Good', 'Good', 'Fair'], dtype=object)
```

In [19]:
```
data_df['color'].unique()
```
executed in 65ms, finished 13:20:52 2021-04-14

Out[19]:
```
array(['E', 'G', 'F', 'D', 'H', 'J', 'I'], dtype=object)
```

In [20]:
```
data_df['clarity'].unique()
```
executed in 82ms, finished 13:20:53 2021-04-14

Out[20]:
```
array(['SI1', 'IF', 'VVS2', 'VS1', 'VVS1', 'VS2', 'SI2', 'I1'],
      dtype=object)
```

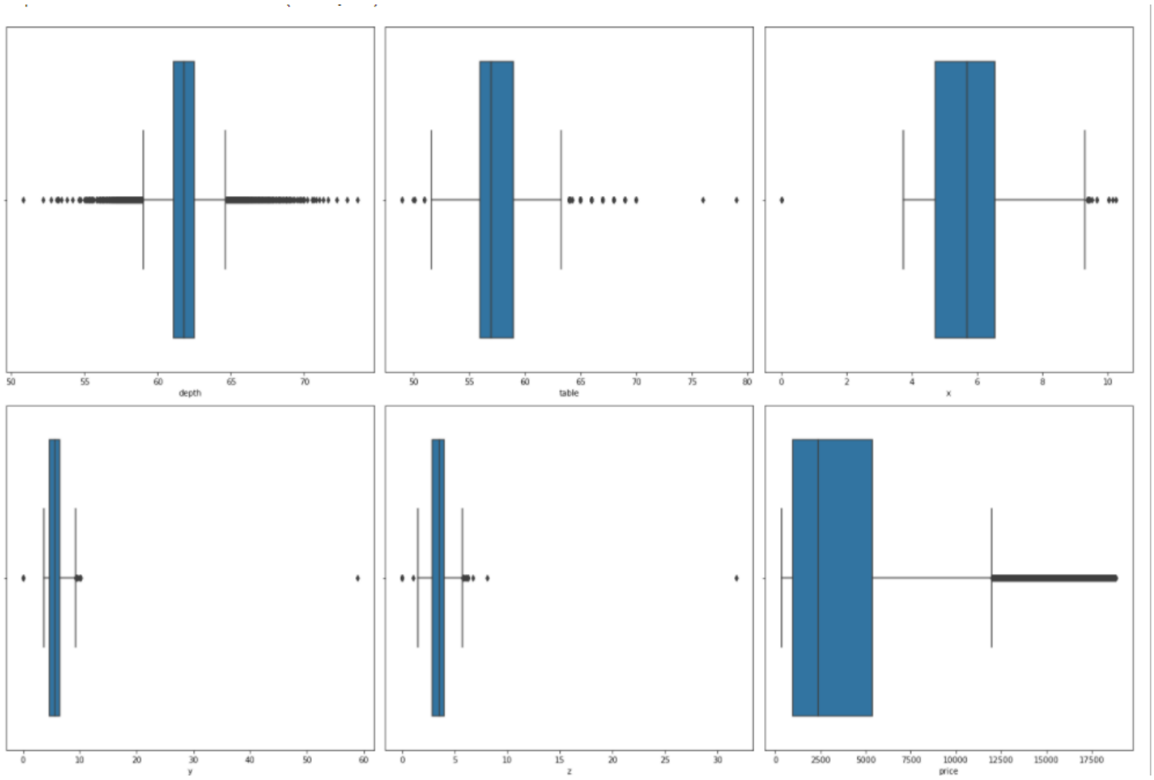
The depth column has null values, so we are replacing it with the median of the particular column.

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

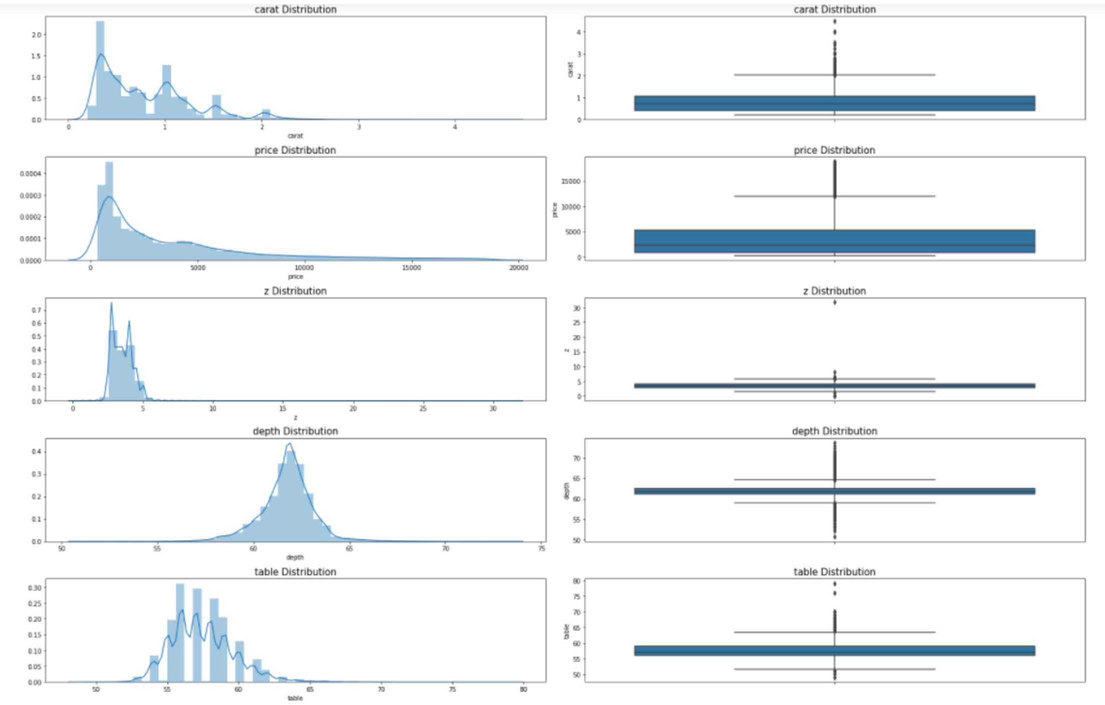
	carat	cut	color	clarity	depth	table	x	y	z	price
0	0.30	Ideal	E	SI1	62.1	58.0	4.27	4.29	2.66	499
1	0.33	Premium	G	IF	60.8	58.0	4.42	4.46	2.70	984
2	0.90	Very Good	E	VVS2	62.2	60.0	6.04	6.12	3.78	6289
3	0.42	Ideal	F	VS1	61.6	56.0	4.82	4.80	2.96	1082
4	0.31	Ideal	F	VVS1	60.4	59.0	4.35	4.43	2.65	779

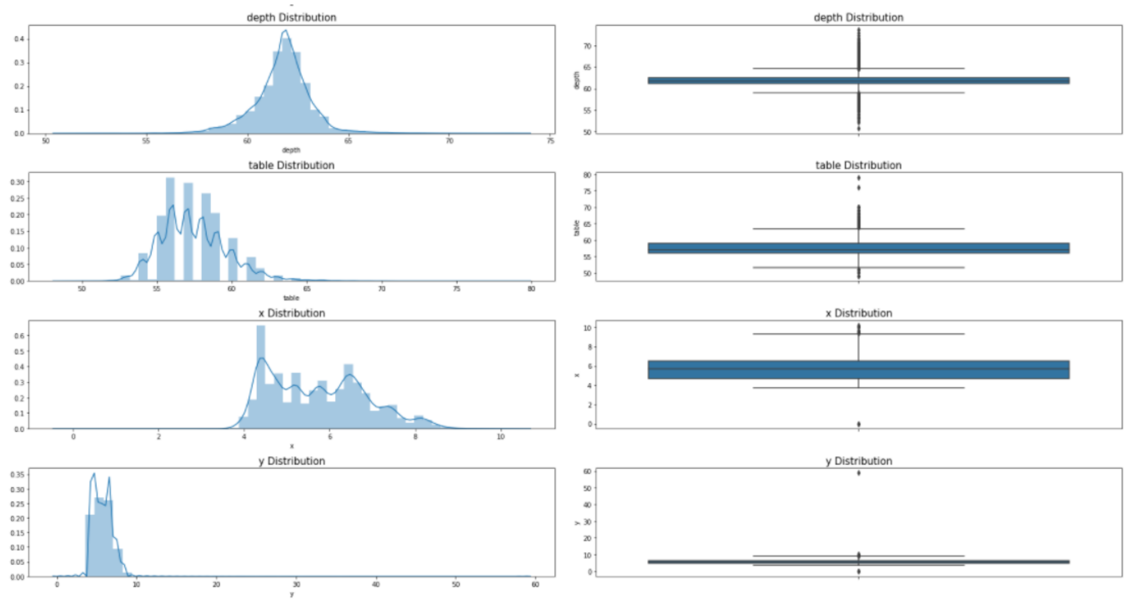
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 26933 entries, 0 to 26966
Data columns (total 10 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   carat    26933 non-null  float64
 1   cut      26933 non-null  object 
 2   color    26933 non-null  object 
 3   clarity  26933 non-null  object 
 4   depth    26933 non-null  float64
 5   table    26933 non-null  float64
 6   x        26933 non-null  float64
 7   y        26933 non-null  float64
 8   z        26933 non-null  float64
 9   price    26933 non-null  int64  
dtypes: float64(6), int64(1), object(3)
memory usage: 2.3+ MB
```

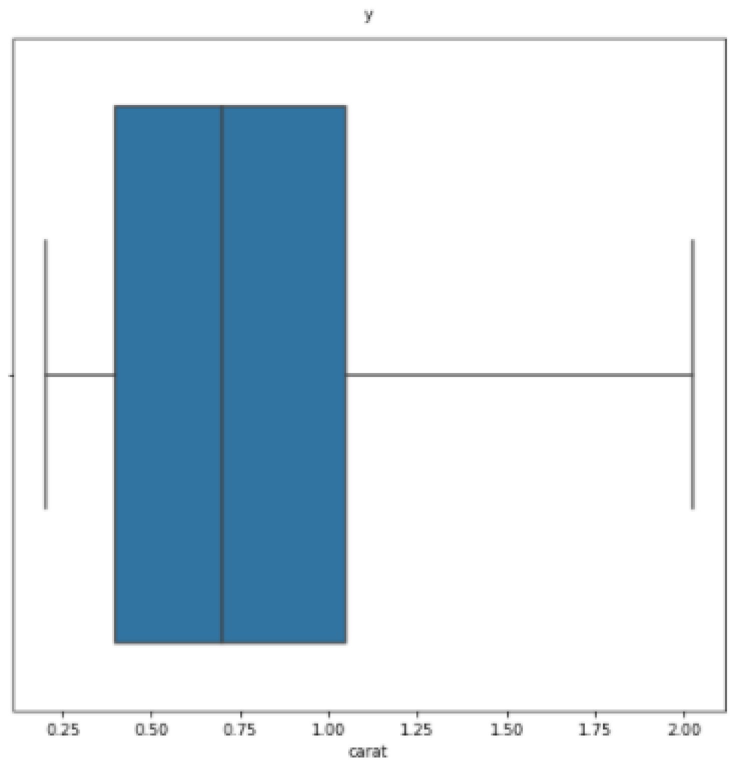
Univariate analysis:



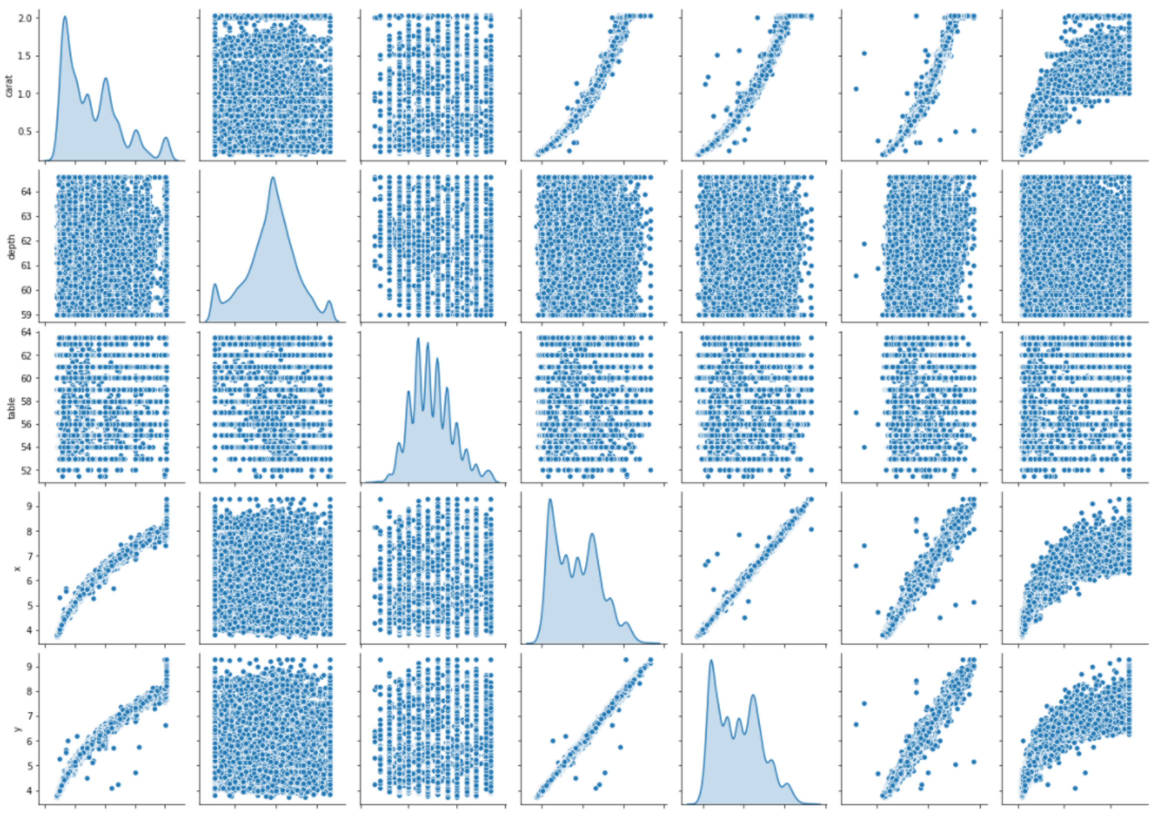
Depth,table,carat and price have significant outliers.
Almost all the variables have outliers.

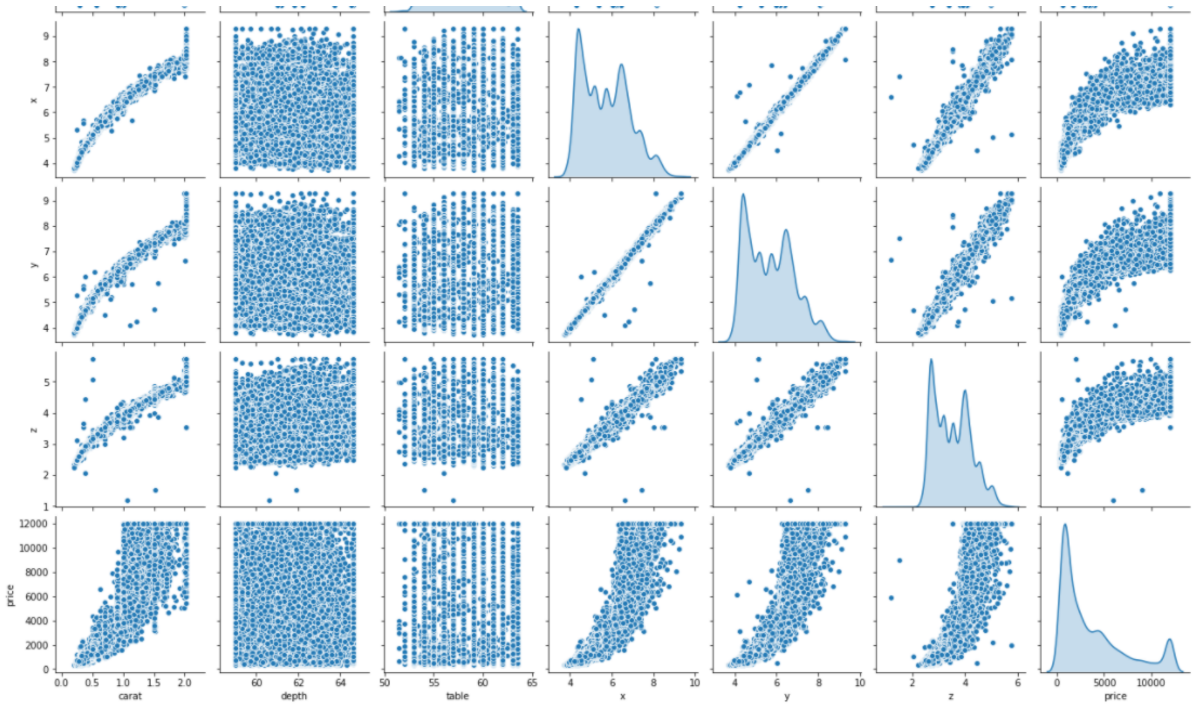

carat Distribution
carat
carat Distribution
carat
price Distribution
price
price Distribution
price
z Distribution
z
z Distribution
z
depth Distribution
depth
depth Distribution
depth
table Distribution
table
table Distribution
table

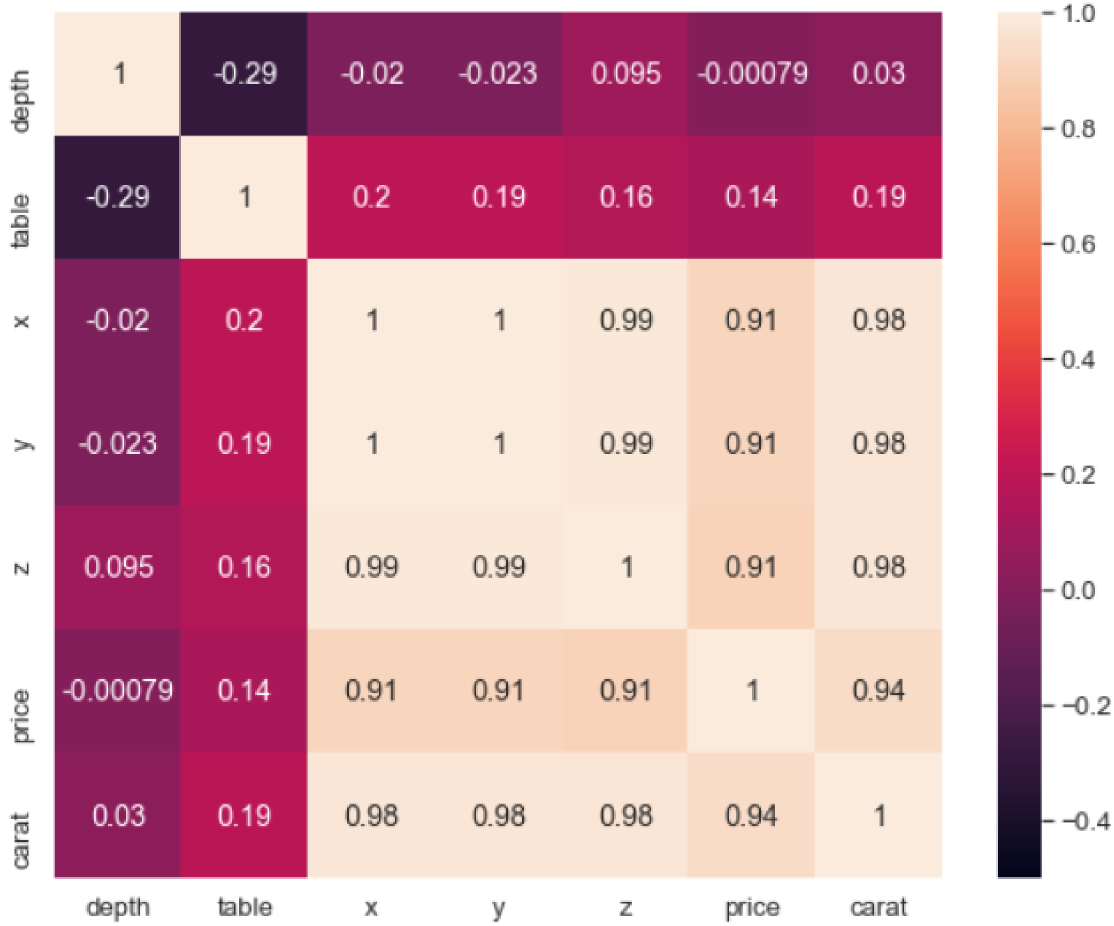

depth Distribution
depth
depth Distribution
depth
table Distribution
table
table Distribution
table
x Distribution
x
x Distribution
x
y Distribution
y
y Distribution
y



Multivariate Analysis







Apart from table and depth, all other variables are highly positively correlated

We can observe that there is a positive correlation of column x,y,z,carat with the price variable

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?

There are null values in a column:depth

```
Out[30]: Unnamed: 0      0
         carat           0
         cut             0
         color           0
         clarity         0
         depth         697
         table           0
         x               0
         y               0
         z               0
         price           0
         dtype: int64
```

Since Unnamed column is not required, we would drop this column for now

Out[33]:

	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

```
data_df.duplicated().sum()
```

executed in 40ms, finished 20:25:02 2021-04-10

```
34
```

There were 34 duplicate rows which have been deleted.

There are columns having 0 values and it needs to be treated since length cannot be of 0 size.The values have been replaced by the mean values of the column.

```
carat      0
cut        0
color      0
clarity    0
depth      0
table      0
x          2
y          2
z          8
price      0
dtype: int64
```

Outliers Treatment:

Shape after outlier treatment.
Outliers have been imputed with lower and higher range values.

```
lower_range
```

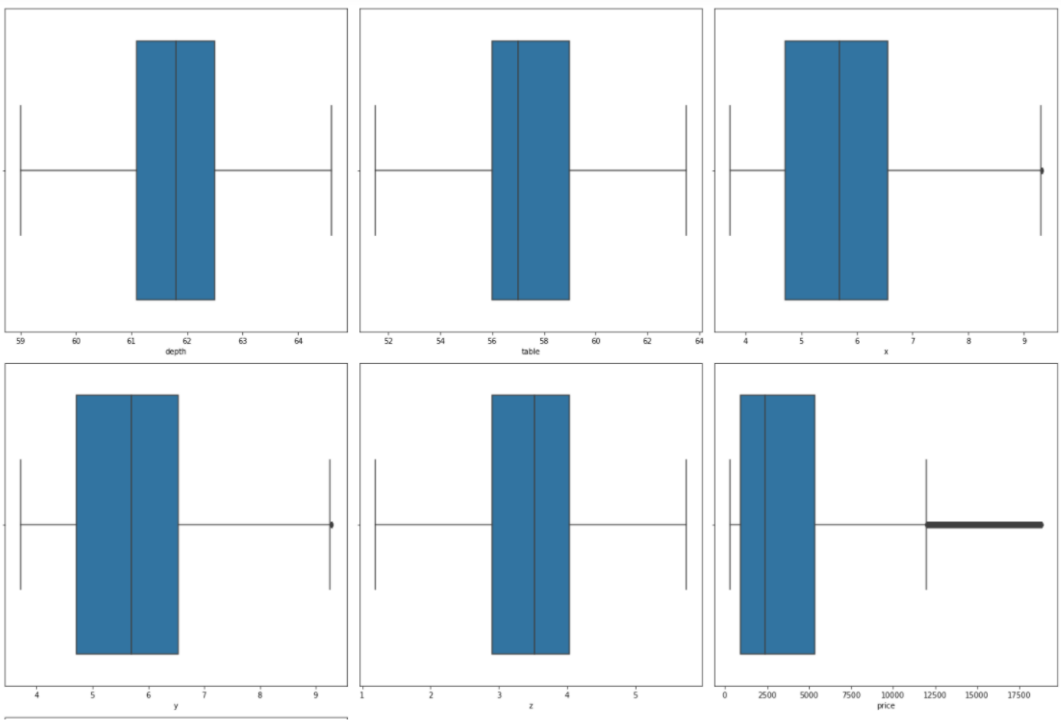
executed in 316ms, finished 13:45:01 2021-04-14

```
carat       -0.575
depth       59.000
table       51.500
x            1.950
y            1.990
z            1.190
price    -5671.500
dtype: float64
```

```
upper_range
```

executed in 21ms, finished 13:45:03 2021-04-14

```
carat         2.025
depth        64.600
table        63.500
x             9.310
y             9.270
z             5.750
price     11972.500
dtype: float64
```



The categorical variables were labelled and one hot encoding was performed for the column:Clarity

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

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

	carat	cut	color	depth	table	x	y	z	price	clarity_IF	clarity_SI1	clarity_SI2	clarity_VS1	clarity_VS2	clarity_VVS1	clarity_VVS2
0	0.30	4	5	62.1	58.0	4.27	4.29	2.66	499	0	1	0	0	0	0	0
1	0.33	3	3	60.8	58.0	4.42	4.46	2.70	984	1	0	0	0	0	0	0
2	0.90	2	5	62.2	60.0	6.04	6.12	3.78	6289	0	0	0	0	0	0	1
3	0.42	4	4	61.6	56.0	4.82	4.80	2.96	1082	0	0	0	1	0	0	0
4	0.31	4	4	60.4	59.0	4.35	4.43	2.65	779	0	0	0	0	0	1	0

Scaling is required in the dataset since all the variables are in different scales which have higher variances and can affect the model.

In linear Regression, Scaling is not mandatory since dependent variables are evaluated based on the coefficients of the independent variable so it does not have any impact.

<u>1.3. Encode the data (having string values) for Modelling. Data Split: Split the data into train and test (70:30). Apply Linear regression. Performance Metrics: Check the performance of Predictions on Train and Test sets using Rsquare, RMSE.</u>

The data set was separated from the dependent column and it was further divided into Training and testing sets with a ratio of 70:30 where 70% data were allocated to training and rest for Testing.

	carat	cut	color	depth	table	x	y	z	clarity_IF	clarity_SI1	clarity_SI2	clarity_VS1	clarity_VS2	clarity_VVS1	clarity_VVS2
0	0.30	4	5	62.1	58.0	4.27	4.29	2.66	0	1	0	0	0	0	0
1	0.33	3	3	60.8	58.0	4.42	4.46	2.70	1	0	0	0	0	0	0
2	0.90	2	5	62.2	60.0	6.04	6.12	3.78	0	0	0	0	0	0	1
3	0.42	4	4	61.6	56.0	4.82	4.80	2.96	0	0	0	1	0	0	0
4	0.31	4	4	60.4	59.0	4.35	4.43	2.65	0	0	0	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
26962	1.11	3	3	62.3	58.0	6.61	6.52	4.09	0	1	0	0	0	0	0
26963	0.33	4	2	61.9	55.0	4.44	4.42	2.74	1	0	0	0	0	0	0
26964	0.51	3	5	61.7	58.0	5.12	5.15	3.17	0	0	0	0	1	0	0
26965	0.27	2	4	61.8	56.0	4.19	4.20	2.60	0	0	0	0	0	0	1
26966	1.25	3	0	62.0	58.0	6.90	6.88	4.27	0	1	0	0	0	0	0

26933 rows × 15 columns

```
y
```

executed in 60ms, finished 16:30:14 2021-04-14

	price
0	499
1	984
2	6289
3	1082
4	779
...	...
26962	5408
26963	1114
26964	1656
26965	682
26966	5166

Initiating the Linear regression model and fitting the train and test data

The coefficient for carat is 13905.235430570028

The coefficient for cut is 121.04183106010112
The coefficient for color is 333.3525724863937
The coefficient for depth is -16.99480591702273
The coefficient for table is -30.19242433719361
The coefficient for x is -2216.615703498819
The coefficient for y is 1178.9055414638349
The coefficient for z is -1532.5229377355956
The coefficient for clarity_IF is 4743.671816845063
The coefficient for clarity_SI1 is 2973.871275710328
The coefficient for clarity_SI2 is 2052.022577830869
The coefficient for clarity_VS1 is 3911.730092688997
The coefficient for clarity_VS2 is 3575.16709014264
The coefficient for clarity_VVS1 is 4381.053337268161

The coefficient for clarity_VVS2 is 4327.667980181488

The intercept for our model is 4472.599152458577

The model score of our linear regression is

For Train set (R-Square) score is 0.922

For Test set (R-Square) score is 0.927

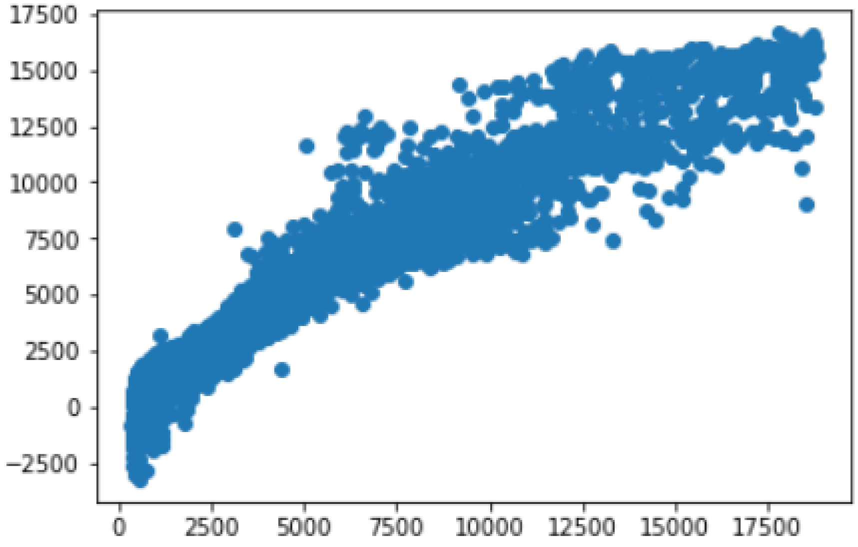
Root Mean Square Error (RMSE) - Root mean square error takes the difference for each observed and predicted value.

The RMSE of Train set for our model is 1117.9855875879305

For our model the RMSE for Test set is – 1114.81

Since this is regression, plot the predicted y value vs actual y values for the test data
A good model's prediction will be close to actual leading to high R and R2 values

```
<matplotlib.collections.PathCollection at 0x1c6e4324f10>
```



Stats model below are the OLS regression results
We can observe that p value is 0 for all the variables.

Dep. Variable:	price	R-squared:	0.925
Model:	OLS	Adj. R-squared:	0.925
Method:	Least Squares	F-statistic:	1.012e+04
Date:	Wed, 14 Apr 2021	Prob (F-statistic):	0.00
Time:	16:44:03	Log-Likelihood:	-1.5873e+05
No. Observations:	18853	AIC:	3.175e+05
Df Residuals:	18829	BIC:	3.177e+05
Df Model:	23		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	3998.8903	856.795	4.667	0.000	2319.495	5678.285
cut[T.1]	568.4359	57.239	9.931	0.000	456.243	680.629
cut[T.2]	700.2668	54.799	12.779	0.000	592.857	807.677
cut[T.3]	774.8570	53.554	14.469	0.000	669.886	879.828
cut[T.4]	850.3378	55.745	15.254	0.000	741.073	959.602
color[T.1]	792.9918	42.902	18.484	0.000	708.899	877.084
color[T.2]	1285.8018	40.741	31.560	0.000	1205.946	1365.658
color[T.3]	1809.9278	39.612	45.691	0.000	1732.285	1887.571
color[T.4]	2022.3290	40.606	49.804	0.000	1942.738	2101.920
color[T.5]	2131.6342	40.564	52.550	0.000	2052.126	2211.143
color[T.6]	2327.9712	42.771	54.429	0.000	2244.137	2411.806
carat	1.388e+04	98.623	140.711	0.000	1.37e+04	1.41e+04
depth	-59.7554	11.016	-5.424	0.000	-81.349	-38.162
table	-32.6734	4.974	-6.569	0.000	-42.423	-22.924
x	-2287.0372	155.985	-14.662	0.000	-2592.782	-1981.293
y	784.0762	156.152	5.021	0.000	478.005	1090.147
z	-744.1474	116.227	-6.403	0.000	-971.962	-516.333
clarity_IF	4633.0896	85.664	54.084	0.000	4465.180	4800.999
clarity_SI1	2959.6456	73.294	40.381	0.000	2815.983	3103.308
clarity_SI2	2029.3771	73.658	27.551	0.000	1885.001	2173.753
clarity_VS1	3864.8627	74.748	51.705	0.000	3718.349	4011.376
clarity_VS2	3551.8819	73.718	48.182	0.000	3407.389	3696.375
clarity_VVS1	4305.6257	78.950	54.536	0.000	4150.877	4460.375
clarity_VVS2	4263.6394	76.898	55.446	0.000	4112.913	4414.366

Omnibus:	6002.760	Durbin-Watson:	1.965
Prob(Omnibus):	0.000	Jarque-Bera (JB):	52613.683
Skew:	1.280	Prob(JB):	0.00
Kurtosis:	10.774	Cond. No.	9.14e+03

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

(3998.89) * Intercept + (568.44) * cut[T.1] + (700.27) * cut[T.2] + (774.86) * cut[T.3] + (850.34) * cut[T.4] + (792.99) * color[T.1] + (1285.8) * color[T.2] + (1809.93) * color[T.3] + (2022.33) * color[T.4] + (2131.63) * color[T.5] + (2327.97) * color[T.6] + (13877.37) * carat + (-59.76) * depth + (-32.67) * table + (-2287.04) * x + (784.08) * y + (-744.15) * z + (4633.09) * clarity_IF + (2959.65) * clarity_SI1 + (2029.38) * clarity_SI2 + (3864.86) * clarity_VS1 + (3551.88) * clarity_VS2 + (4305.63) * clarity_VVS1 + (4263.64) * clarity_VVS2 +

We can observe that the price is higher when the cut value is Ideal and price increases by 850.34 units.

Price increases by 2327.97 units where the color value is D
Price increases by 13877.37 units when the carat value increases by 1 unit.
When the clarity_VVS2 increases the price increases by 4263.64 units.

We have observed that cut,color and clarity are the highest factors which are driving the price. So company can focus on these parameters and think about the production.

Problem 2: Logistic Regression and LDA

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

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

Reading the data:

	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

Checking the datatype of variables

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

Description of the data

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Unnamed: 0	872	NaN	NaN	NaN	436.5	251.869	1	218.75	436.5	654.25	872
Holliday_Package	872	2	no	471	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Salary	872	NaN	NaN	NaN	47729.2	23418.7	1322	35324	41903.5	53469.5	236961
age	872	NaN	NaN	NaN	39.9553	10.5517	20	32	39	48	62
educ	872	NaN	NaN	NaN	9.30734	3.03626	1	8	9	12	21
no_young_children	872	NaN	NaN	NaN	0.311927	0.61287	0	0	0	0	3
no_older_children	872	NaN	NaN	NaN	0.982798	1.08679	0	0	1	2	6
foreign	872	2	no	656	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Checking Null values in the dataset:

No null value

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

Dropping Unnamed column since it is not required

	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

Checking Duplicacy in the data and there was none.

```
data_df.duplicated().sum()
```

executed in 353ms, finished 13:38:03 2021-04-12

```
0
```

Shape of the data
872 rows and 7 columns

```
data_df.shape
```

executed in 48ms, finished 20:15:44 2021-04-11

```
(872, 7)
```

Checking count of 0's in the column

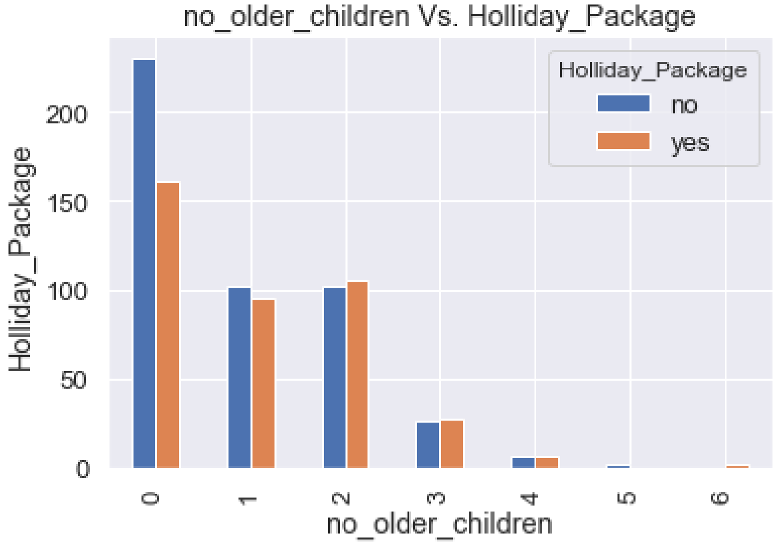
```
Holliday_Package         0
Salary                   0
age                      0
educ                     0
no_young_children      665
no_older_children      393
foreign                  0
dtype: int64
```

When we further checked the variables having 0's in the column and observed that the values are meaningful

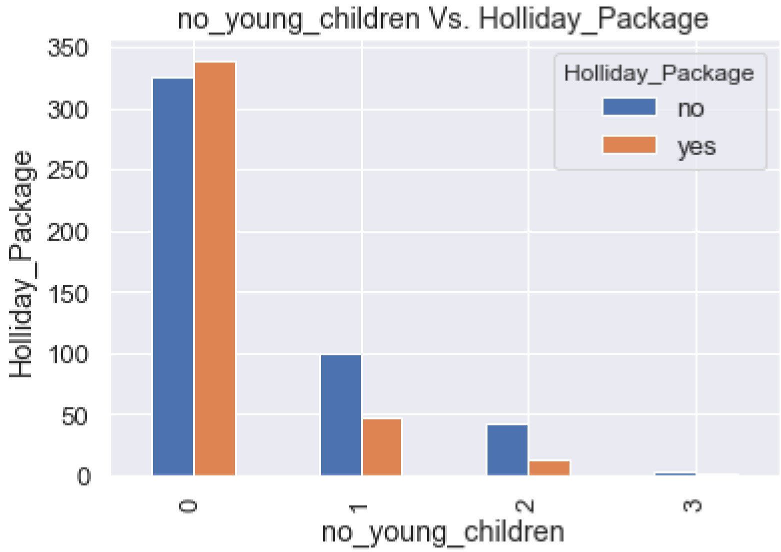
There are observation where no. of young children is 0

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

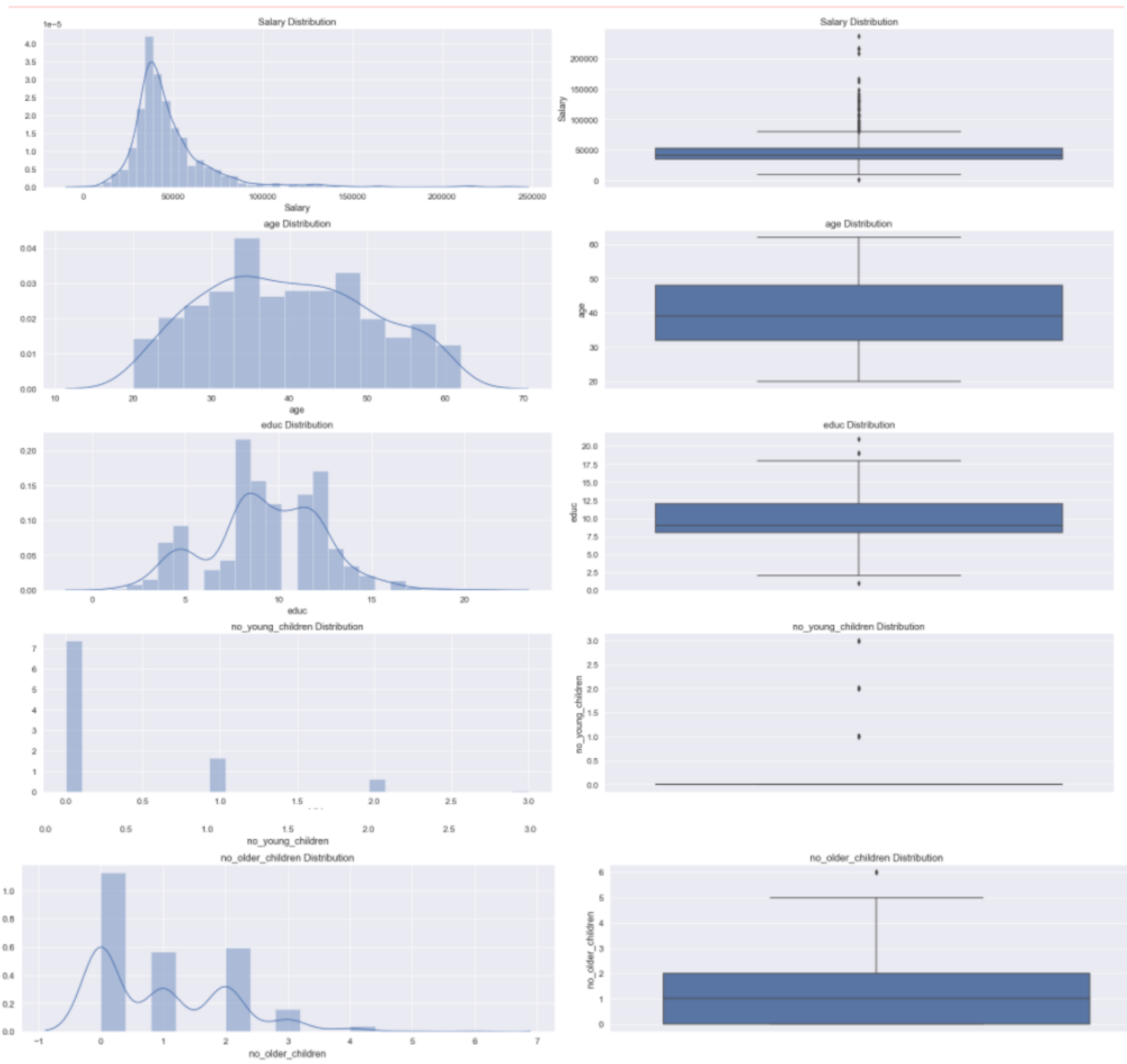
no_older_children	0	1	2	3	4	5	6	All
Holliday_Package								
no	231	102	102	27	7	2	0	471
yes	162	96	106	28	7	0	2	401
All	393	198	208	55	14	2	2	872



The value 0 present in the dataset is meaningful



Univariate Analysis:



Age coulmn is normally distributed and does not contain outliers whereas other variables has outliers

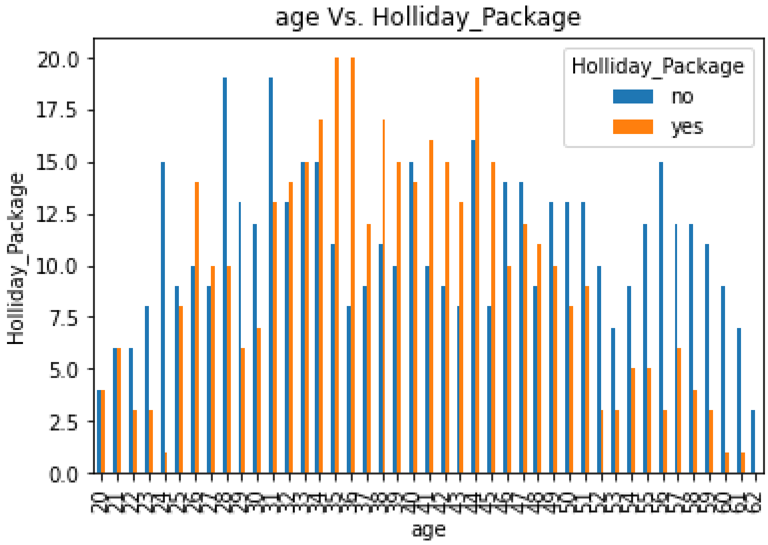
```
Propotion of yes is 45.98623853211009, Propotion of no is 54.01376146788991
```

Proportion of Employees didnt opt for Holiday package is 54.01

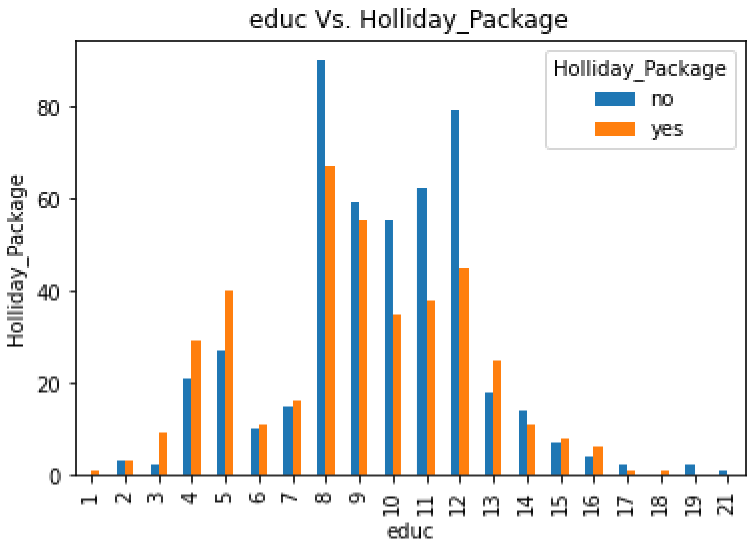
Crosstab comparison between Holiday package and foreign tour.
Majority of the holiday package costumes opted for Domestic travel

Holliday_Package	no	yes	All
foreign			
no	402	254	656
yes	69	147	216
All	471	401	872

Age and Holiday Package comparison.
Most of the travellers between age 32 till 45

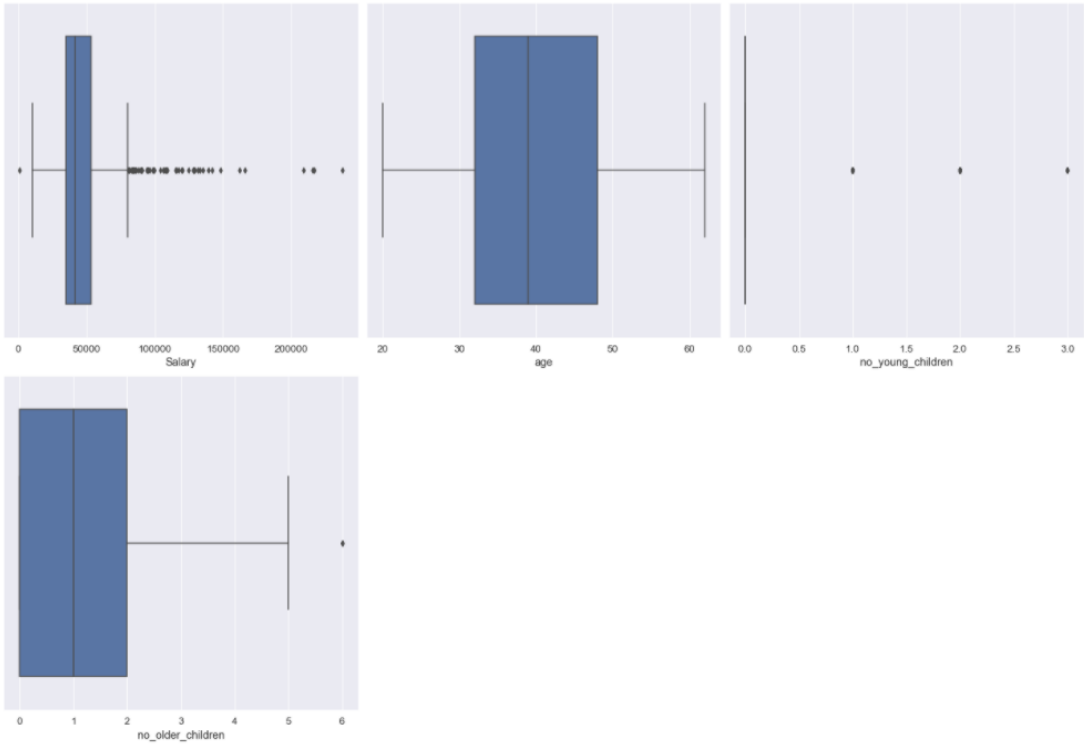


Highest travellers has years of education between 8 till 12

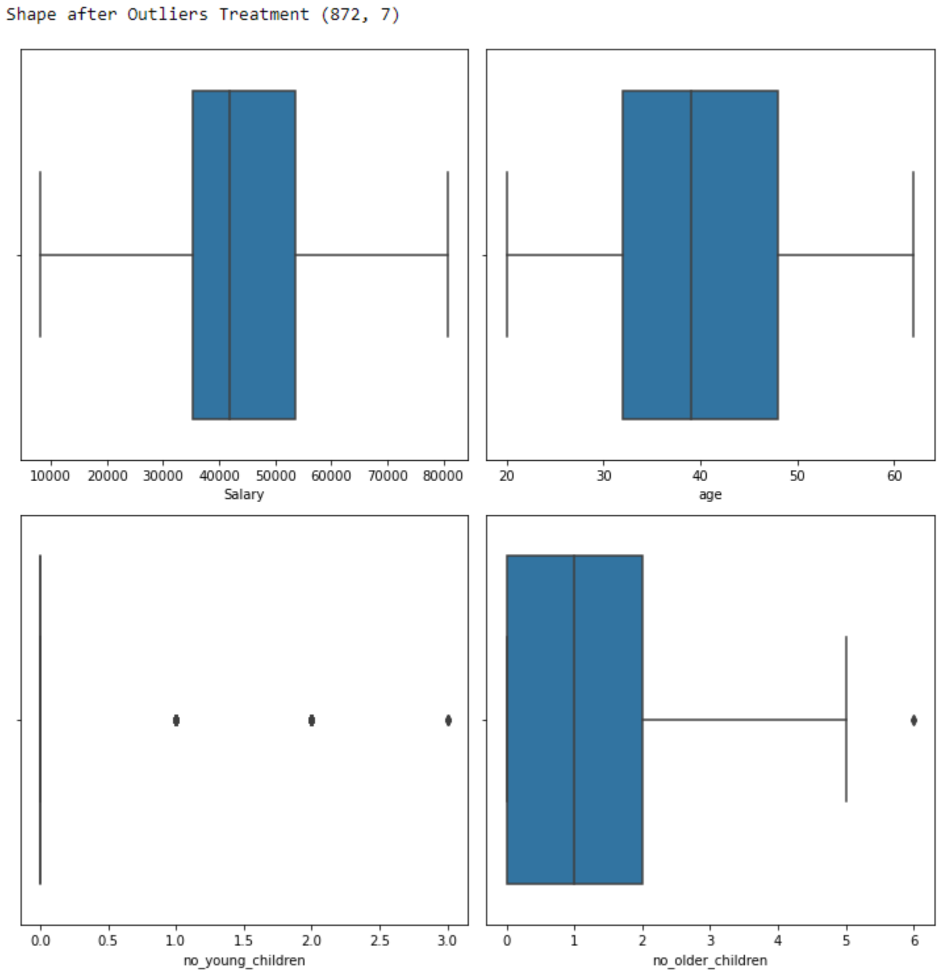


We can see from the below snapshot that salary has outlets

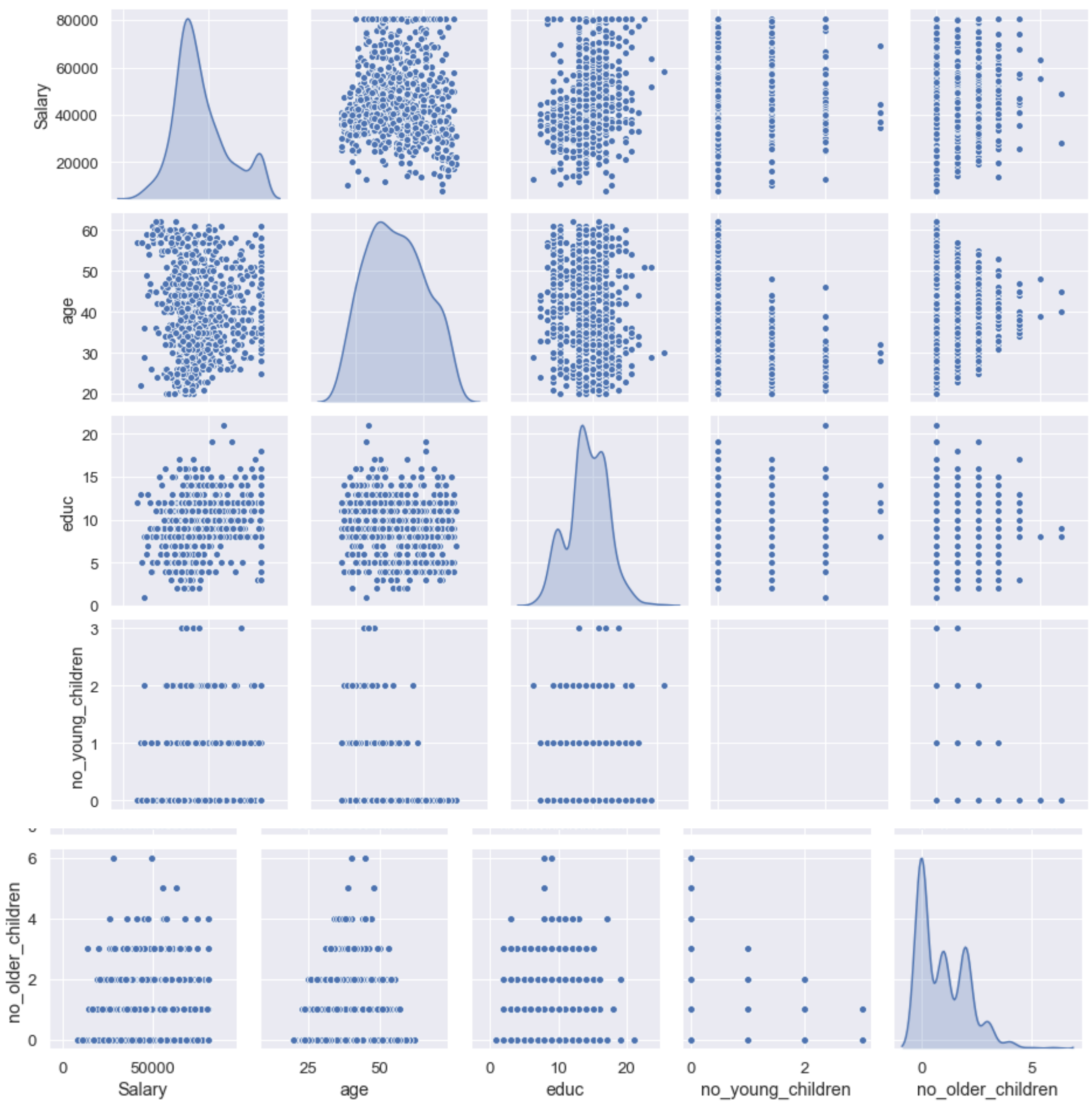
```
Shape before Outliers Treatment (872, 7)
```



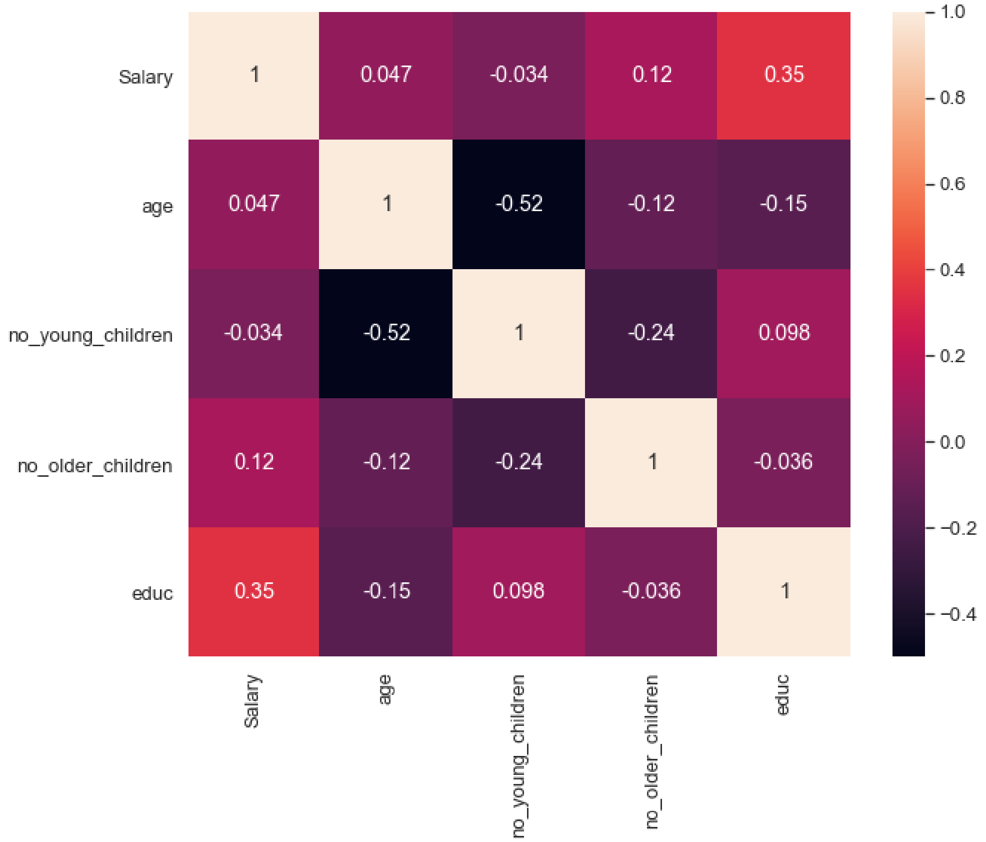
Shape after Outliers Treatment (872, 7)



Bivariate analysis:



Salary
age
educ
no_young_children
no_older_children



no_young_children is negatively correlated with age Education is positive correlated with Salary

We can observe that none of the variables has strong correlation, so multicollinearity is not an issue with this dataset.

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).

The column variables of Object data type were encoded (Holiday Package and foreign).

The data was splitted into 70%(training) and 30 % testing data with random_state=100

The holiday package column was dropped Since 'holiday package' is dependent variable

Linear regression function was invoked and fitted with the data.
Training and testing data class prediction was done after fitting with model with the dataset.

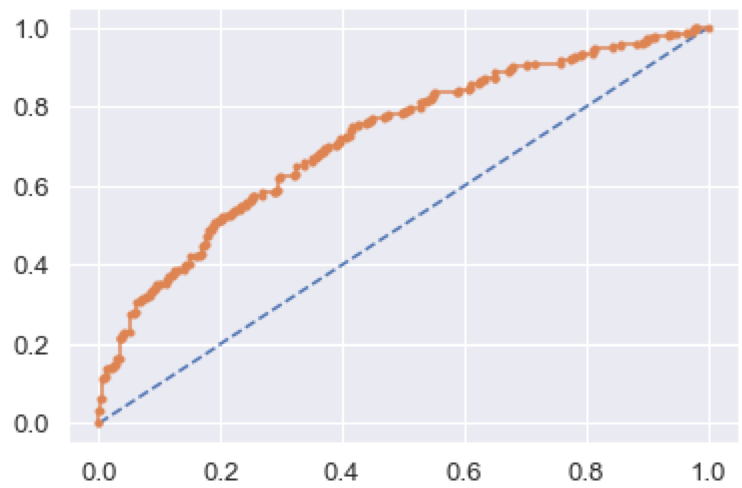
2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each

model Final Model: Compare Both the models and write inference which model is best/optimized.

Performance metrics of LinearDiscriminantAnalysis

```
AUC for the Training Data: 0.720

[<matplotlib.lines.Line2D at 0x23abddc90a0>]
```




```
AUC for the Test Data: 0.742
```

Confusion Matrix for Training dataset:
Accuracy:67%
Precision: 66%

	precision	recall	f1-score	support
0	0.67	0.77	0.71	332
1	0.66	0.55	0.60	278
accuracy			0.67	610
macro avg	0.67	0.66	0.66	610
weighted avg	0.67	0.67	0.66	610

Confusion Matrix for Testing dataset:
Accuracy:67%
Precision: 70%

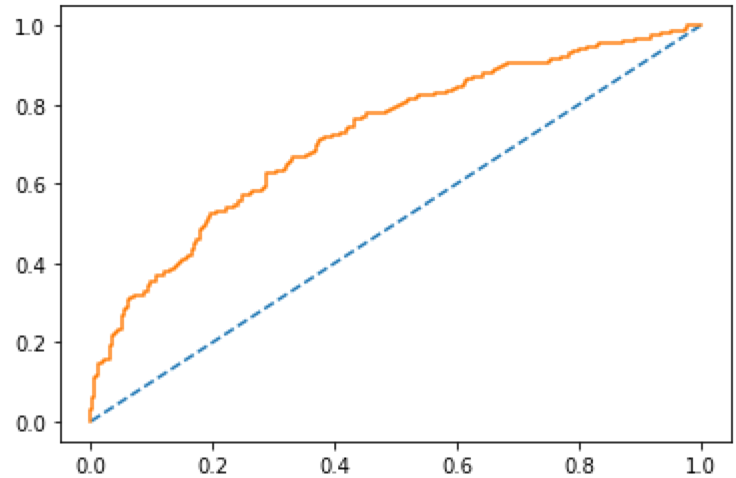
	precision	recall	f1-score	support
0	0.66	0.80	0.72	139
1	0.70	0.53	0.60	123
accuracy			0.67	262
macro avg	0.68	0.66	0.66	262
weighted avg	0.68	0.67	0.66	262

Model Score of Training Set:0.66
Model Score of Testing Set: 0.67

Performance metrics of Logistics Regression

Training Data:
Model Score:0.665
Accuracy:67%
Precision: 66%

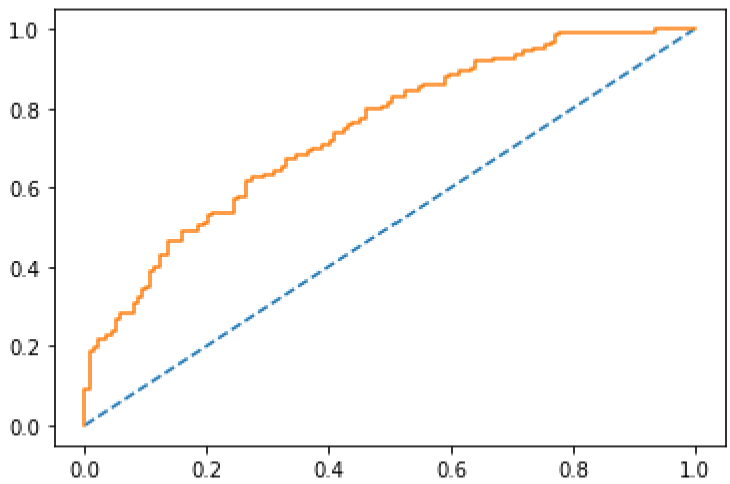
```
AUC: 0.721
```



Testing Data:
Model Score:0.67
Accuracy:67%
Precision: 67%

	precision	recall	f1-score	support
0	0.67	0.77	0.71	332
1	0.66	0.54	0.60	278
accuracy			0.67	610
macro avg	0.66	0.66	0.66	610
weighted avg	0.67	0.67	0.66	610

AUC: 0.743



	precision	recall	f1-score	support
0	0.66	0.79	0.72	139
1	0.69	0.54	0.61	123
accuracy			0.67	262
macro avg	0.68	0.66	0.66	262
weighted avg	0.68	0.67	0.67	262

Inference:
We can observe that both the models are performing equally without a significant difference as shown by comparison of Confusion matrix obtained by both the models.

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

1) **As per below analysis, we have observed that where no. of young and older children are 0, customers are not opting for Holiday Package.**

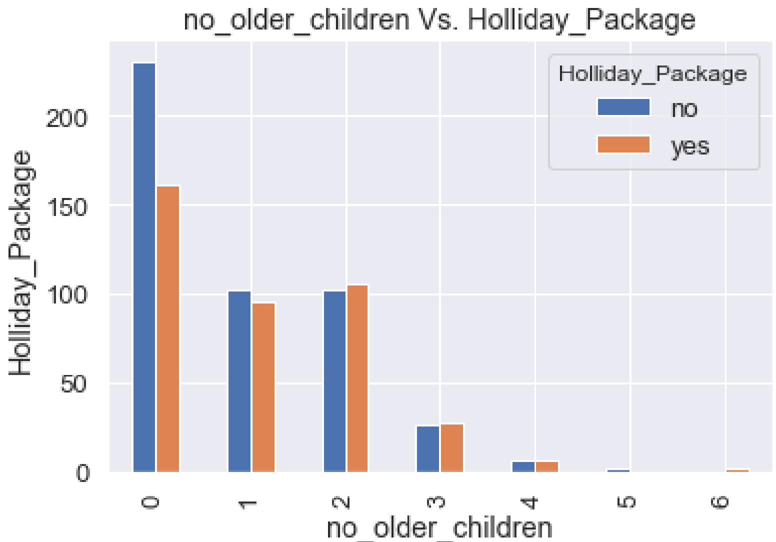
2) **It would mean that Holiday Package prices for these customer segment are not profitable, so the company should rethink in this direction and come up with a new package and consider the pricing issue.**

3)

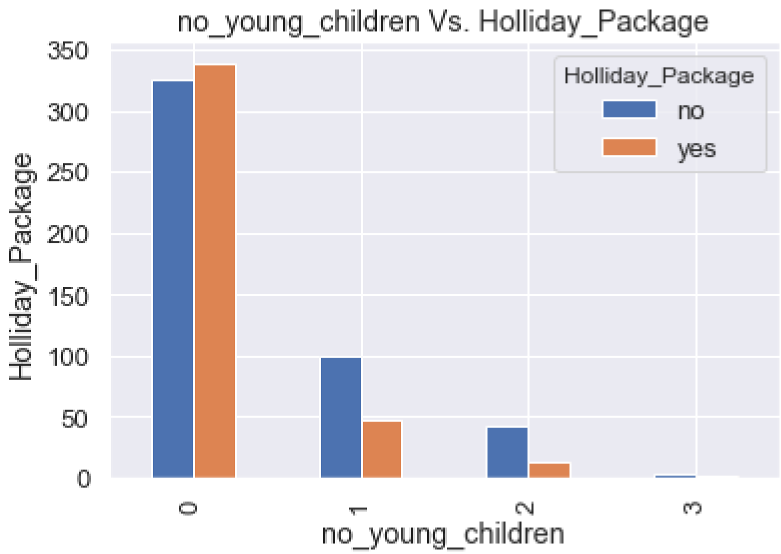
There are observation where no. of young children is 0

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

no_older_children Holliday_Package	0	1	2	3	4	5	6	All
no	231	102	102	27	7	2	0	471
yes	162	96	106	28	7	0	2	401
All	393	198	208	55	14	2	2	872

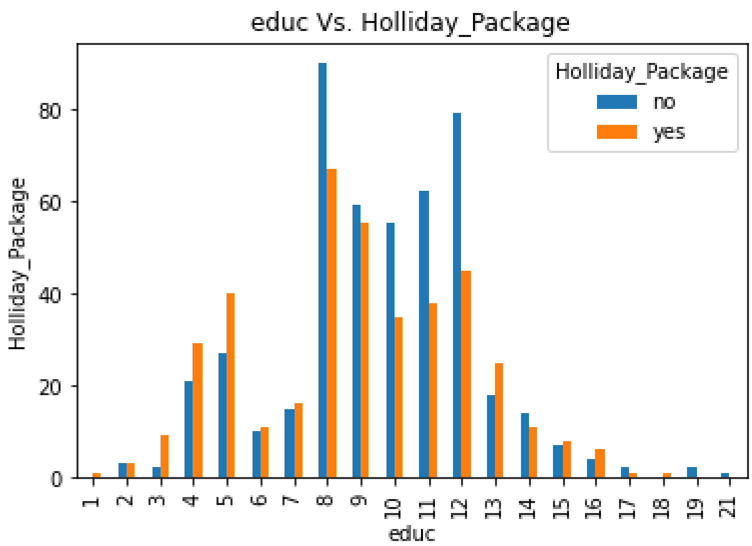


The value 0 present in the dataset is meaningful



Recommendation 2:

As per below analysis, we observe that more customers opting for packages have Education level between 8 till 12.
The company can do some more research on this behaviour and come up with a package to cover customers of other Education level range



Recommendation 3:

The company can come up with a package to target customers having higher salary. For e.g. Salary beyond 100000