Priyadharshini K

Project Report - Predictive Modelling course

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.

	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

Univariate Analysis:

Summary Statistics of the dataset

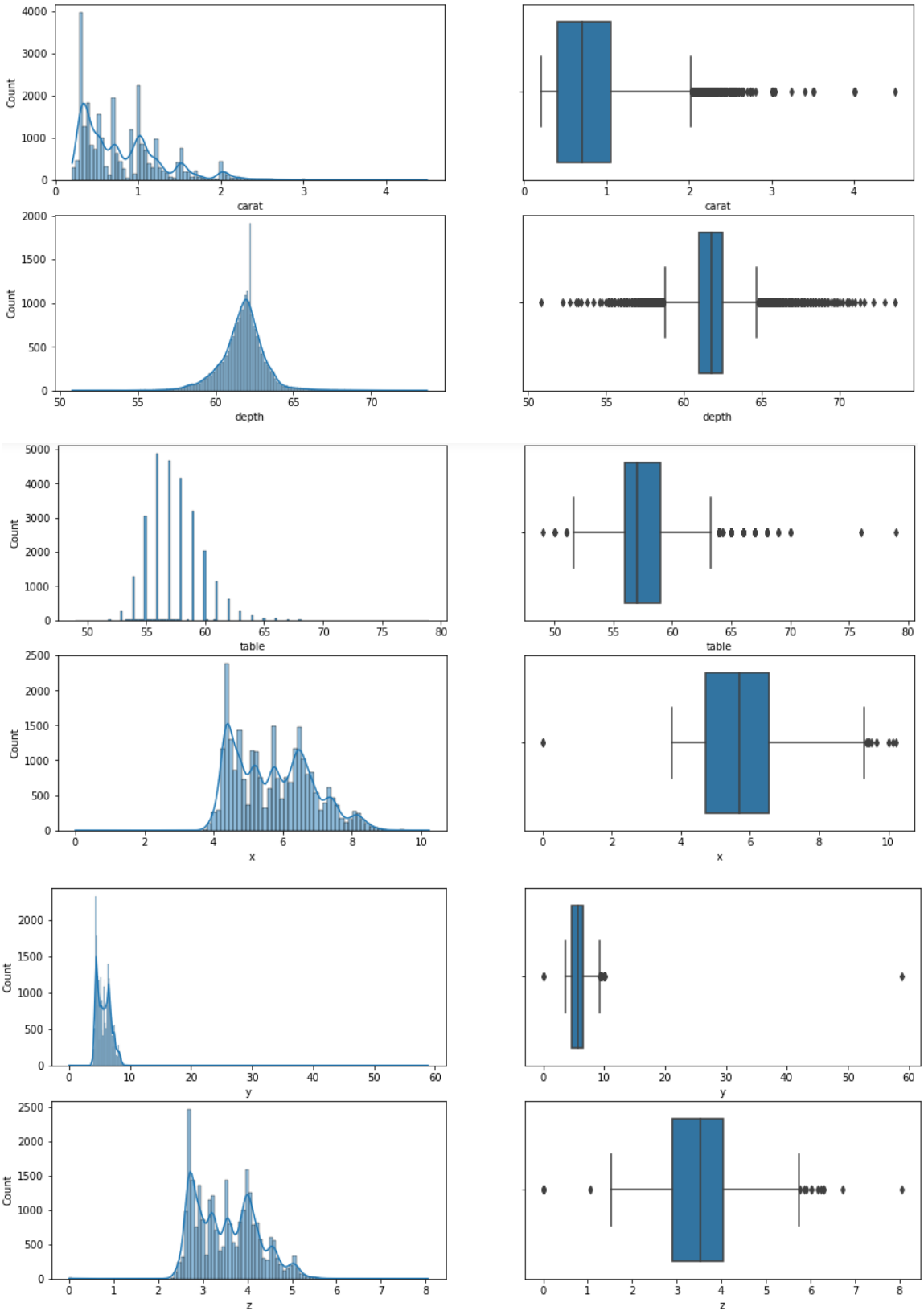
	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Unnamed: 0	26967	NaN	NaN	NaN	13484	7784.85	1	6742.5	13484	20225.5	26967
carat	26967	NaN	NaN	NaN	0.798375	0.477745	0.2	0.4	0.7	1.05	4.5
cut	26967	5	Ideal	10816	NaN	NaN	NaN	NaN	NaN	NaN	NaN
color	26967	7	G	5661	NaN	NaN	NaN	NaN	NaN	NaN	NaN
clarity	26967	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

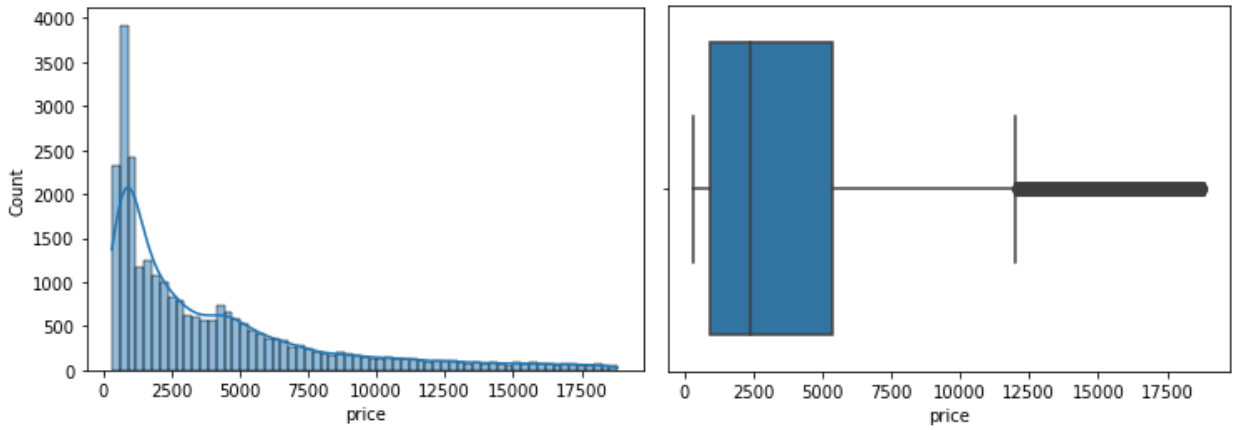
Inferences:

- The dataset has a total of 10 dependent and 1 independent variables. The dataset has both categorical and continuous data.
- Shape of the dataset: (26270, 11)
- Null values are present in the dataset
- No duplicate values are present in the dataset
- Object Datatype variables – cut, color, clarity

- Int Datatype variables – Unnamed: 0, price
- Float Datatype variables – carat, depth, table, x, y, z

BOX PLOT & DISTRIBUTION PLOT





Skewness:

Skewness assesses the extent to which a variable's distribution is symmetrical.

```
Skewness of carat :  1.117688359567716

Skewness of depth :  -0.028616421040645105

Skewness of table :  0.7686814656119176

Skewness of x :  0.38731881697027615

Skewness of y :  3.93929233731628

Skewness of z :  0.3648073463317343

Skewness of price :  1.6199469874625374
```

Kurtosis:

Kurtosis is a measure of whether the distribution is too peaked (a very narrow distribution with most of the responses in the center).

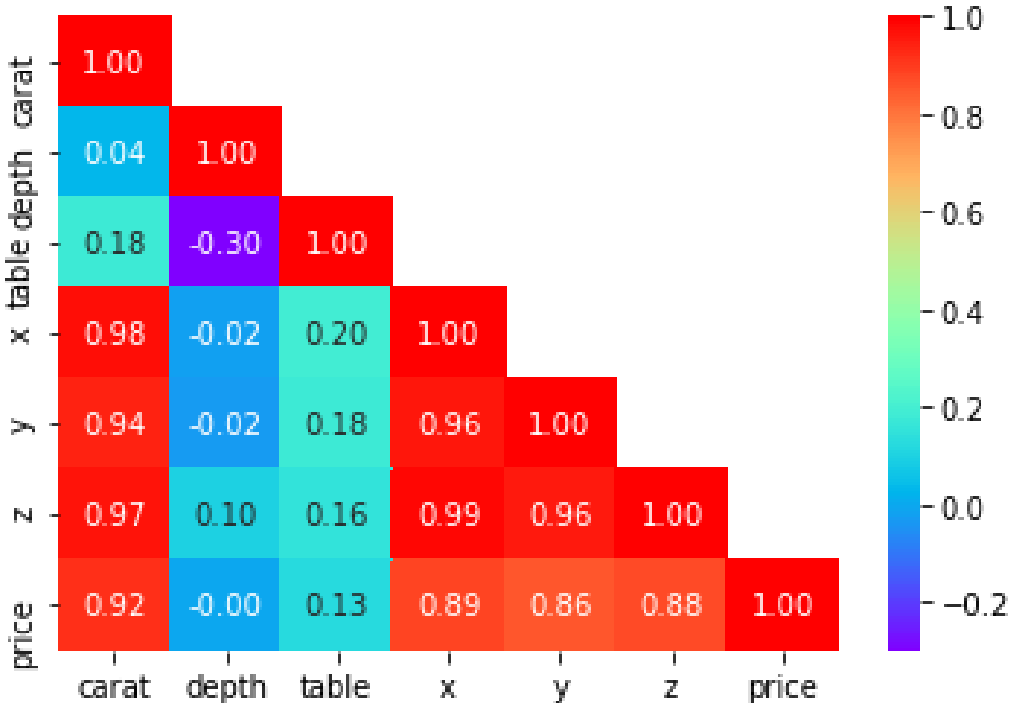
```
Kurtosis of carat :  1.2281660266810714

Kurtosis of depth :  3.6735028620292898

Kurtosis of table :  1.5829639923577359

Kurtosis of x :  -0.6542651883364003

Kurtosis of y :  163.3630736178029

Kurtosis of z :  -0.4539599096795288

Kurtosis of price :  2.1569146818207496
```

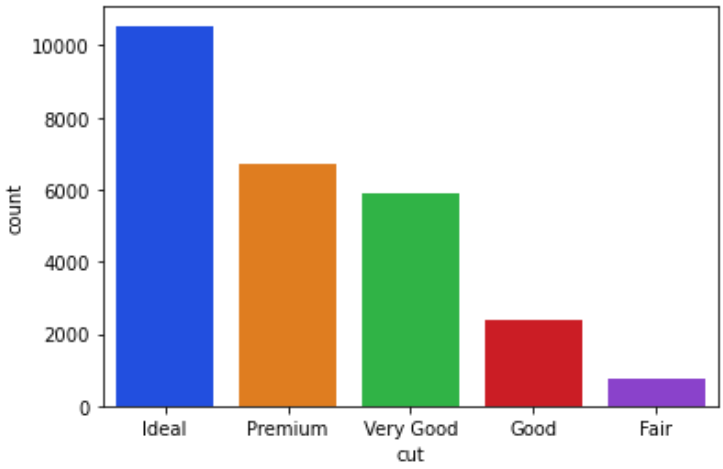
Inferences on Univariate Analysis:

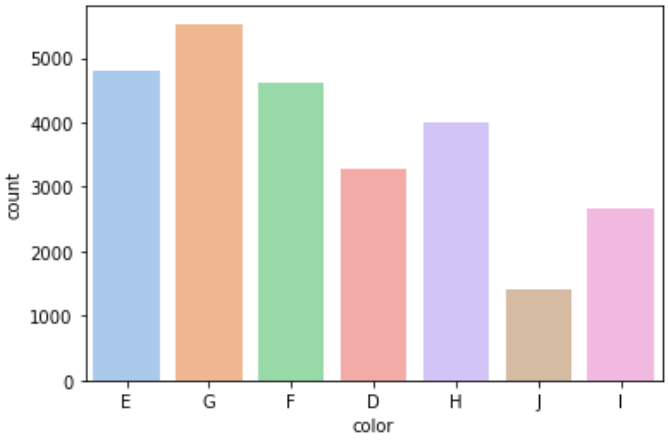
- The dataset has 3 Object datatypes, 6 float and 2 int64 variables and label encoding has to be performed to categorical variables to perform regression.
- Data is Normally distributed with slight skew for almost all features
- Outliers are present in all the variables and presence of outliers to be treated
- From the box plots, we can conclude that all the variables have outliers. The outliers were treated and were made equal to the whisker values i.e., Q1 - 1.5*IQR or Q3 + 1.5*IQR whichever is nearer.
- Positive values (carat,depth,table,price,y) of kurtosis indicate that a distribution is peaked and possess thick tails.
- If the kurtosis is less than zero, then the distribution is light tailed (x,z).
- An extreme positive kurtosis indicates a distribution where more of the values are located in the tails of the distribution rather than around the mean.
- Skewness of depth, x, z is between -0.5 to 0.5, indicates that the data is symmetrical.
- Skewness of carat, y and price is more than 1, indicates that the data is highly skewed to right.
- Skewness of table is between 0.5 and 1, the data are moderately skewed.

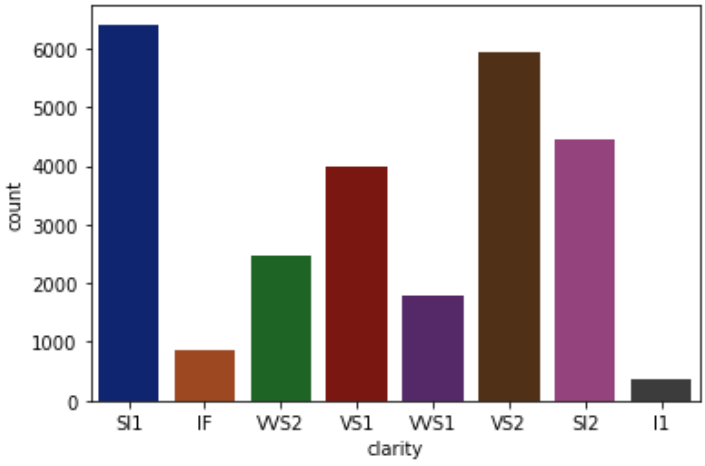
Multi-variate Analysis:

X, Y, Z, Price & Carat has good co-relation ; table and depth has poor co-relation







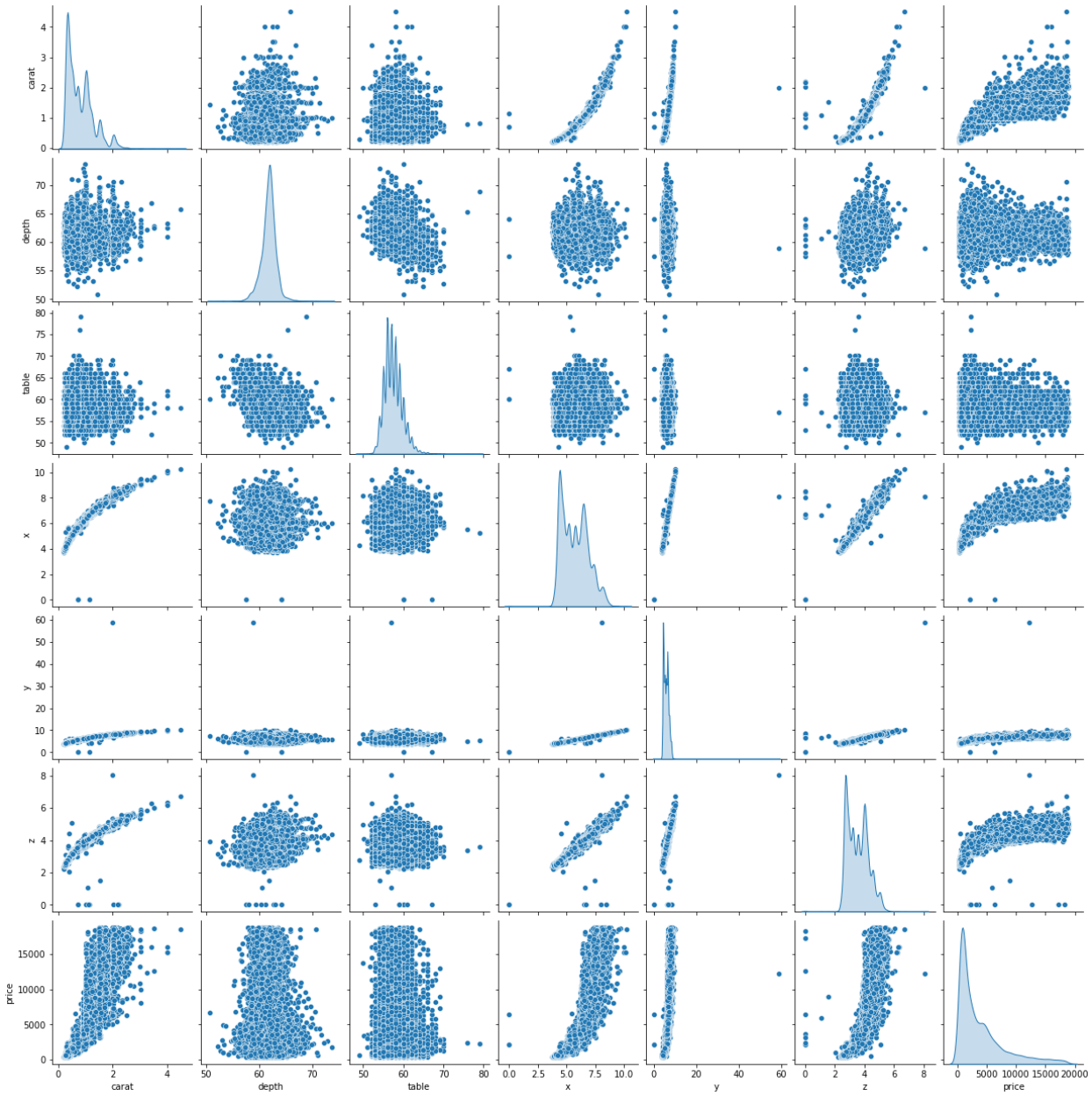


The count of 'Ideal' category in 'Cut' variable is the maximum and 'Fair' category count is minimum.

The count of 'G' category in 'Color' variable is the maximum and 'J' category count in 'Color' variable is minimum.

The count of 'SI1' category in 'Clarity' variable is the maximum and 'I1' category count in 'Clarity' variable is minimum.

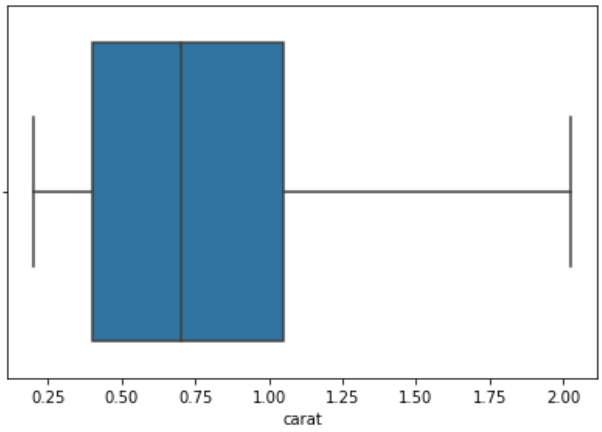
Pair Plot:

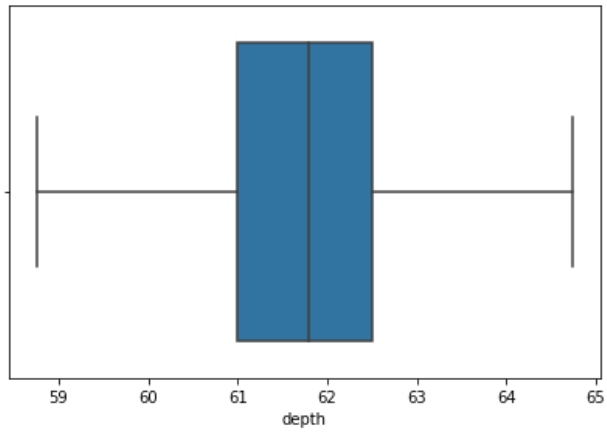


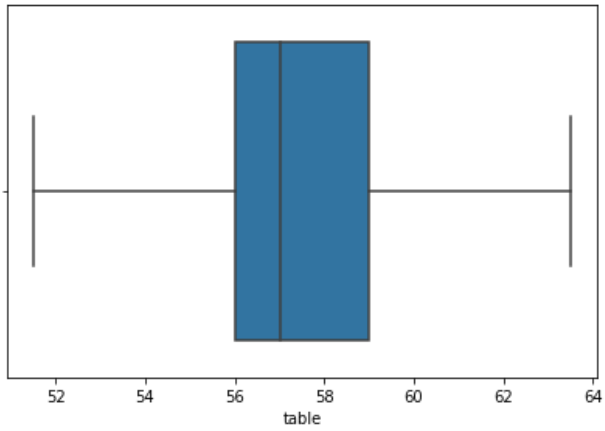
Inferences:

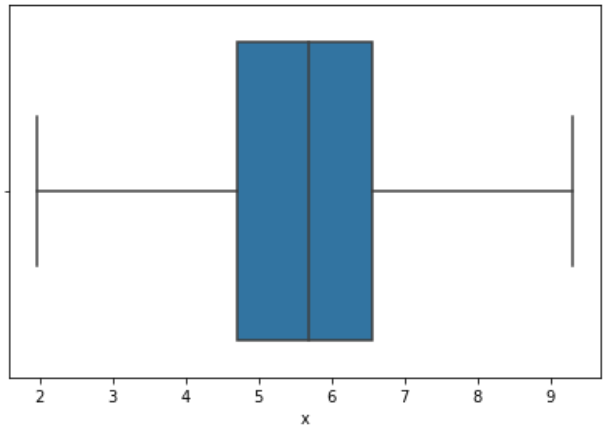
- X (0.98), Y (0.94), Z (0.97), Price (0.92) have good co-relation with Carat; table (0.18) and depth (0.04) have poor co-relation with Carat.
- X has good co-relation with Y (0.96) and Z (0.99)
- Carat has almost linear relationship with X & Z and similarly, X & Z have a linear relationship
- Variable Y has a constant value irrespective of increase in values in Carat, depth, table, Z and price

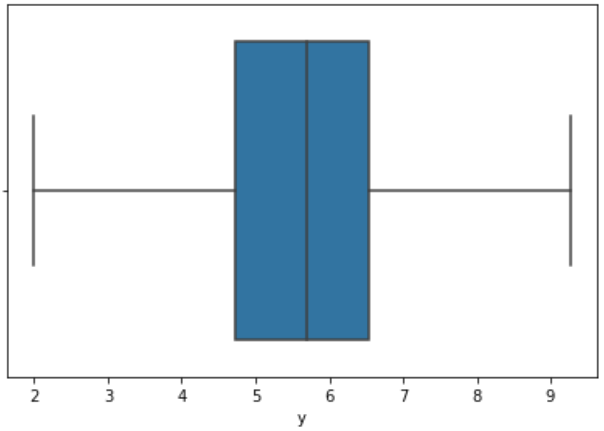
After Outlier Treatment:

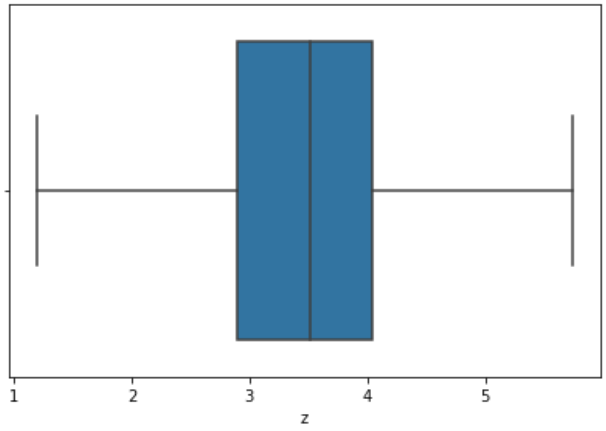












1.2 Impute null values if present, also check for the values which are equal to zero. Do they have any meaning, or do we need to change them or drop them? Do you think scaling is necessary in this case?

```
# null value check
print('Number of Null value Rows = %d{}' .format(df.isna().sum()))
```

```
Number of Null value Rows = %dUnnamed: 0      0
carat            0
cut              0
color            0
clarity          0
depth          697
table            0
x                0
y                0
z                0
price            0
dtype: int64
```

Except for the 'depth' variable other variables do not have null value.

Depth variable has about only 2.5% of missing data. Hence it is better to drop the null value rows.

Scaling is not necessary for performing Linear Regression. Linear Regression algorithm almost gives similar model performance for data with and without scaling and hence scaling is not required.

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.

Encoding:

One hot encoding is done to the categorical variables in the dataset using get_dummies function after which 11 columns in the dataset becomes 25 columns

```
df = pd.get_dummies(df, columns=['cut','color','clarity'],drop_first=True)
```

```
df.head()
```

	x	y	z	price	cut_Good	cut_Ideal	...	color_H	color_I	color_J	clarity_IF	clarity_SI1	clarity_SI2	clarity_VS1	clarity_VS2	clarity_VVS1	clarity_VVS2
27	4.29	2.66	499.0	0	1	...	0	0	0	0	1	0	0	0	0	0
42	4.46	2.70	984.0	0	0	...	0	0	0	1	0	0	0	0	0	0
04	6.12	3.78	6289.0	0	0	...	0	0	0	0	0	0	0	0	0	1
82	4.80	2.96	1082.0	0	1	...	0	0	0	0	0	0	1	0	0	0
35	4.43	2.65	779.0	0	1	...	0	0	0	0	0	0	0	0	1	0

Splitting the data into train and test set in 70:30 ratio:

```
# Split X and y into training and test set in 70:30 ratio
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3 , random_state=1)
```

Applying Linear Regression model:

```
## Linear Regression Model

regression_model = LinearRegression()
regression_model.fit(X_train, y_train)
```

Performance of predictions on Train and Test sets:

R-square

R-squared is the percentage of the dependent variable variation that a linear model explains.

R-squared is a goodness-of-fit measure for linear regression models. This statistic indicates the percentage of the variance in the dependent variable that the independent variables explain collectively.

```
# R square on training data
print('R square on training data is {}'.format(regression_model.score(X_train, y_train)))
```

```
R square on training data is 0.9405460602987309
```

94% of the variation in the Price is explained by the predictors in the model for train set.

```
# R square on test data
print('R square on test data is {}'.format(regression_model.score(X_test, y_test)))
```

```
R square on test data is 0.9403861399640809
```

94% of the variation in the Price is explained by the predictors in the model for test set as well.

Inference:

R-square is almost same on the train set and test set. The model accuracy is good.

RMSE:

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are. RMSE is a measure of how spread out these residuals are.

```
#RMSE on Training data
predicted_train=regression_model.fit(X_train, y_train).predict(X_train)
print('RMSE on training data is {}'.format(np.sqrt(metrics.mean_squared_error(y_train,predicted_train))))
```

```
RMSE on training data is 846.8383643580527
```

```
#RMSE on Testing data
predicted_test=regression_model.fit(X_train, y_train).predict(X_test)
print('RMSE on test data is {}'.format(np.sqrt(metrics.mean_squared_error(y_test,predicted_test))))
```

```
RMSE on test data is 844.4398851317088
```

Inference:

RMSE values between the Training data and Testing data are the same. The model is a good fit model.

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

The final Linear Regression equation is¶

price = b0 + b1 * carat + b2 * depth + b3 * table + b4 * x + b5 * y + b6 * z + b7 * cut_Good + b8 * cut_Ideal + b9 * cut_Premium + b10 * cut_Very_Good + b11 * color_E + b12 * color_F + b13 * color_G + b14 * color_H + b15 * color_I + b16 * color_J + b17 * clarity_IF + b18 * clarity_SI1 + b19 * clarity_SI2 + b20 * clarity_VS1 + b21 * clarity_VS2 + b22 * clarity_VVS1 + b23 * clarity_VVS2

price = (-2591.8) * Intercept + (9060.41) * carat + (-2.79) * depth + (-21.23) * table + (-1121.06) * x + (915.82) * y + (-367.38) * z + (425.69) * cut_Good + (648.86) * cut_Ideal + (635.21) * cut_Premium + (539.18) * cut_Very_Good + (-201.82) * color_E + (-270.31) * color_F + (-430.96) * color_G + (-869.21) * color_H + (-1334.07) * color_I + (-1899.66) * color_J + (4150.61) * clarity_IF + (2655.11) * clarity_SI1 + (1832.65) * clarity_SI2 + (3466.69) * clarity_VS1 + (3173.75) * clarity_VS2 + (3900.67) * clarity_VVS1 + (3869.1) * clarity_VVS2

Keeping all the other predictors constant,

- For a unit change in the variable 'carat', price increases by 9060.41 and for a unit change in depth, the price decreases by -21.23 and for a unit change in x the price decreases by 1121. 06 and for a unit change in y the price increases by 915.82.
- When the cut is 'Good', the price increases maximum of 648.86 compared with Premium, Ideal and very good.
- The price value decreases by a minimum value for color 'J' followed by color 'I', 'H', 'G', 'F' and so on.
- Clarity 'IF' fetches the maximum increase in price (4150.61) for a unit change in its value followed by clarity_VS2
- When depth increases by 1 unit, price decreases by 2.79 keeping other predictors constant.
- Hence business recommendation to increase profitability to Gem Stones co ltd, is as follows.

Higher profitable stone is that which has higher values carat, y, Ideal cut with color E and clarity IF with lower values for depth table, x, z, color J.

Lower profitable stone on the other side has lower values for carat, y and so on.

Multicollinearity occurs when there are two or more independent variables in a multiple regression model, which have a high correlation among themselves. When some features are highly correlated, we might have difficulty in distinguishing between their individual effects on the dependent variable. Multicollinearity can be detected using various techniques, one such technique being the **Variance Inflation Factor**(**VIF**).

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
vif = [variance_inflation_factor(df.values , ix) for ix in range( df.shape [1])]
i=0
for column in df.columns:
    if (i < 12):
        print (column ,"------>", vif[i])
    i=i+1
```

```
Unnamed: 0 ------> 4.004704518013347
carat ------> 211.71276246801057
depth ------> 1097.8009012547943
table ------> 874.6650574338905
x ------> 11750.812039214312
y ------> 11268.65390447232
z ------> 2104.5584415980056
price ------> 36.31018038962774
cut_Good ------> 4.482543666519543
cut_Ideal ------> 17.615587313909376
cut_Premium ------> 10.769371828516562
cut_Very Good ------> 9.938612073263485
```

X and Y has very high values of VIF which shows that X and Y are multi collinear and considering both the variables together leads to a model with multicollinearity. X influences Y

```
# Let us explore the coefficients for each of the independent attributes

for idx, col_name in enumerate(X_train.columns):
    print("The coefficient for {} is {}".format(col_name, regression_model.coef_[0][idx]))
```

```
The coefficient for Unnamed: 0 is -0.00048148441026179254
The coefficient for carat is 9040.40666355195
The coefficient for depth is -6.619651396943221
The coefficient for table is -22.590244908160226
The coefficient for x is -1144.7319625875748
The coefficient for y is 921.0243423173067
The coefficient for z is -324.6621284750098
The coefficient for cut_Good is 415.9357310688661
The coefficient for cut_Ideal is 632.3617780090086
The coefficient for cut_Premium is 623.8505219640028
The coefficient for cut_Very Good is 523.3642372466684
The coefficient for color_E is -201.57366272153732
The coefficient for color_F is -270.72695093214844
The coefficient for color_G is -438.98745837727137
The coefficient for color_H is -874.2764131288081
The coefficient for color_I is -1341.20895996173
The coefficient for color_J is -1901.4941682600345
The coefficient for clarity_IF is 4133.514555197883
The coefficient for clarity_SI1 is 2630.7307270008755
The coefficient for clarity_SI2 is 1815.751478675133
The coefficient for clarity_VS1 is 3439.5115137454
The coefficient for clarity_VS2 is 3156.114148493433
The coefficient for clarity_VVS1 is 3879.812520760072
The coefficient for clarity_VVS2 is 3849.8496277755776
```

The best 5 attributes which are most important are determined by checking the co-efficient of determination.

Carat, Clarity, Color, X and Y

The least important variables are Cut, Z, Table and Depth

OLS Regression Results

Dep. Variable:	price	R-squared:	0.941
Model:	OLS	Adj. R-squared:	0.940
Method:	Least Squares	F-statistic:	1.263e+04
Date:	Sun, 16 May 2021	Prob (F-statistic):	0.00
Time:	15:35:37	Log-Likelihood:	-1.5006e+05
No. Observations:	18389	AIC:	3.002e+05
Df Residuals:	18365	BIC:	3.004e+05
Df Model:	23		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2271.6411	694.160	-3.273	0.001	-3632.259	-911.023
carat	9040.7971	75.776	119.310	0.000	8892.270	9189.325
depth	-6.6010	9.090	-0.726	0.468	-24.419	11.217
table	-22.5845	3.933	-5.742	0.000	-30.294	-14.875
x	-1145.1765	119.810	-9.558	0.000	-1380.015	-910.338
y	921.5581	118.584	7.771	0.000	689.123	1153.994
z	-325.0526	104.845	-3.100	0.002	-530.558	-119.547
cut_Good	416.0637	43.238	9.623	0.000	331.312	500.815
cut_Ideal	632.4493	42.349	14.934	0.000	549.442	715.457
cut_Premium	623.8267	40.704	15.326	0.000	544.043	703.611
cut_Very_Good	523.3740	41.518	12.606	0.000	441.994	604.754
color_E	-201.6446	23.013	-8.762	0.000	-246.752	-156.537
color_F	-270.7880	23.420	-11.562	0.000	-316.694	-224.882
color_G	-438.9193	22.751	-19.293	0.000	-483.513	-394.326
color_H	-874.3584	24.315	-35.960	0.000	-922.018	-826.699
color_I	-1341.3380	27.181	-49.349	0.000	-1394.614	-1288.061
color_J	-1901.3161	33.124	-57.400	0.000	-1966.242	-1836.390
clarity_IF	4133.7828	65.865	62.762	0.000	4004.682	4262.883
clarity_SI1	2630.9601	56.471	46.590	0.000	2520.273	2741.648
clarity_SI2	1815.8678	56.777	31.983	0.000	1704.580	1927.155
clarity_VS1	3439.6606	57.638	59.677	0.000	3326.685	3552.636
clarity_VS2	3156.2997	56.747	55.620	0.000	3045.070	3267.529
clarity_VVS1	3879.9243	61.019	63.585	0.000	3760.321	3999.527
clarity_VVS2	3850.1479	59.322	64.903	0.000	3733.872	3966.424

Omnibus:	4645.442	Durbin-Watson:	2.000
Prob(Omnibus):	0.000	Jarque-Bera (JB):	16866.303
Skew:	1.238	Prob(JB):	0.00
Kurtosis:	6.985	Cond. No.	9.48e+03

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.

```
data=pd.read_csv('Holiday_Package.csv')
```

```
data.head()
```

	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

Descriptive Statistics:

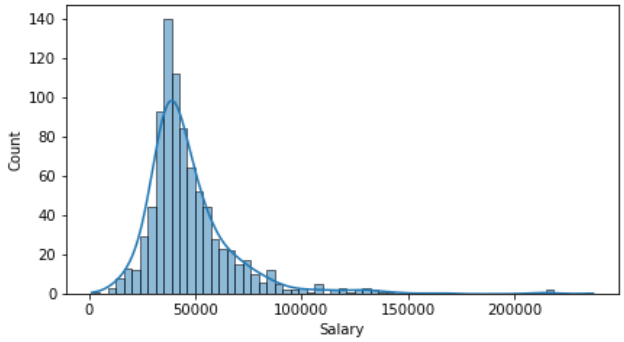
```
data.describe(include='all').T
```

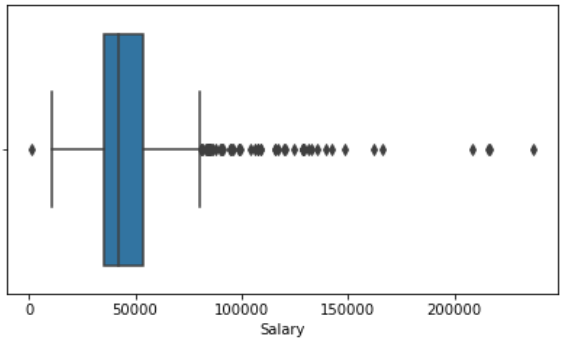
	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Unnamed: 0	872	NaN	NaN	NaN	436.5	251.869	1	218.75	436.5	654.25	872
Holliday_Package	872	2	no	471	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Salary	872	NaN	NaN	NaN	47729.2	23418.7	1322	35324	41903.5	53469.5	236961
age	872	NaN	NaN	NaN	39.9553	10.5517	20	32	39	48	62
educ	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

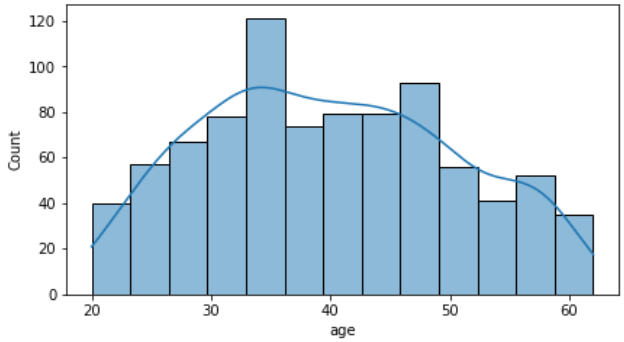
Inferences:

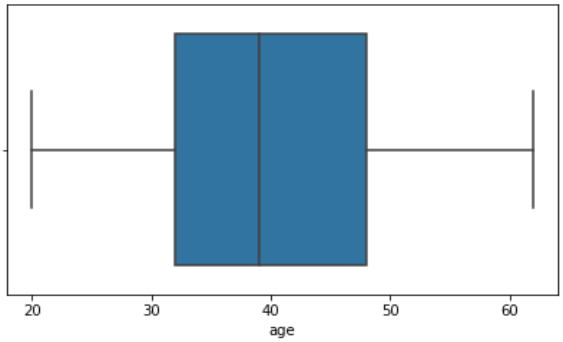
- The dataset has a total of 7 dependent and 1 independent variables. The dataset has both categorical and continuous data.
- Shape of the dataset: (872, 8)
- Null values are NOT present in the dataset
- No duplicate values are present in the dataset
- Object Datatype variables – foreign, Holiday_Package
- Int Datatype variables – 'Unnamed: 0', 'Salary', 'age', 'educ', 'no_young_children', 'no_older_children'

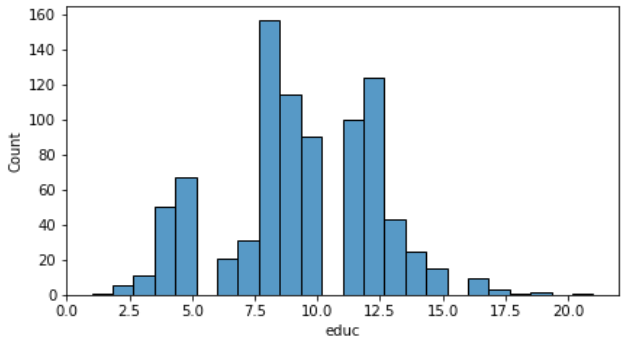
BOX PLOT & DISTRIBUTION PLOT

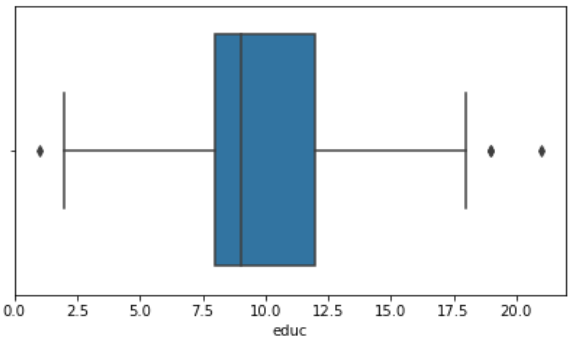


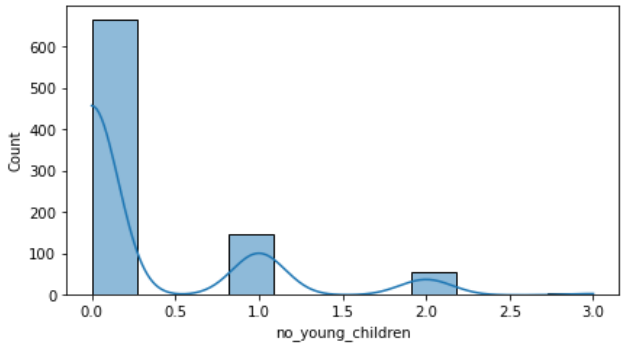


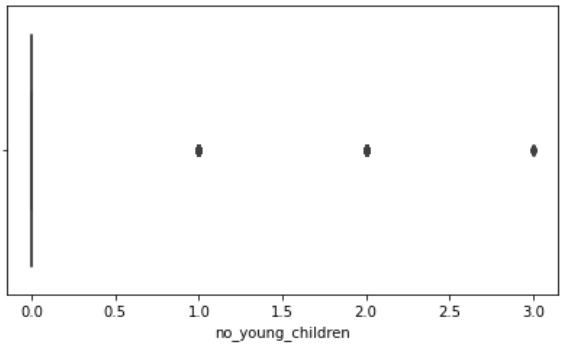


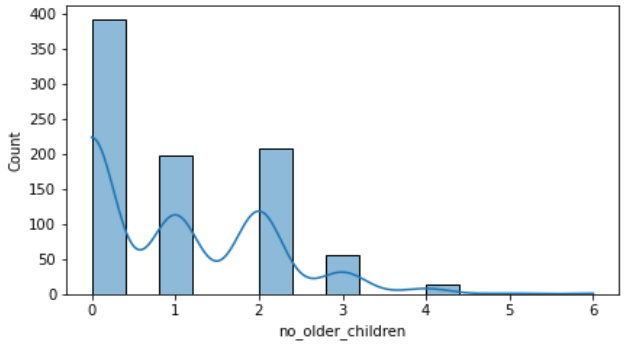


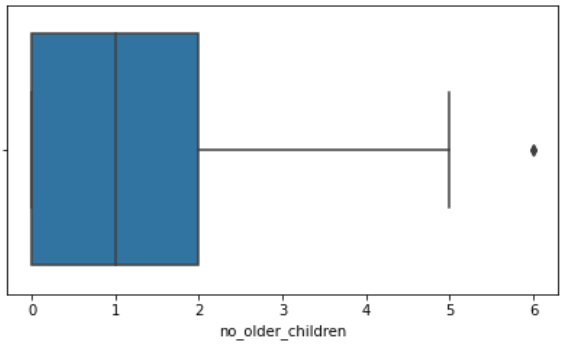












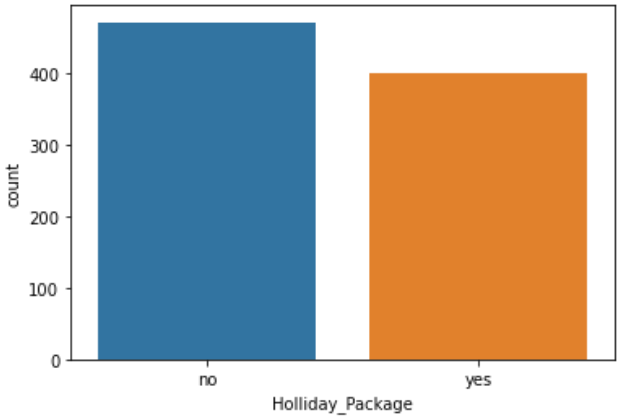
Inferences from Univariate Analysis

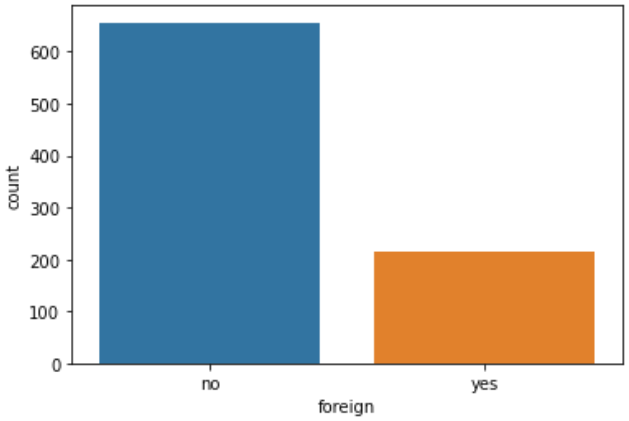
'Salary' variable is almost a normal distribution skewed towards left.

'Age' variable is normally distributed.

'Education', 'No_young_children', 'No_older_children' does not for a rgular band of continuos distribution.

All variables 'Salary', 'Education', 'No_young_children', 'No_older_children'except 'Age' has outliers present.






```
data["no_older_children"].value_counts(normalize=True)
```

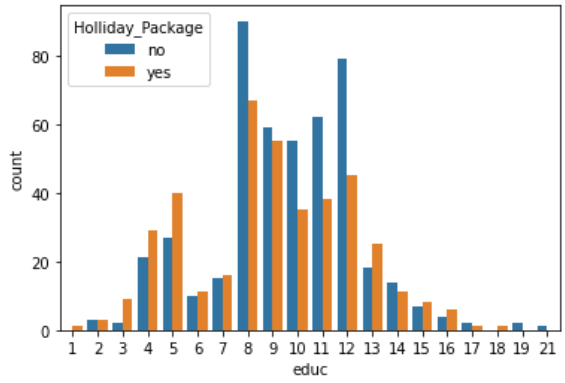
```
0    0.450688
2    0.238532
1    0.227064
3    0.063073
4    0.016055
6    0.002294
5    0.002294
Name: no_older_children, dtype: float64
```

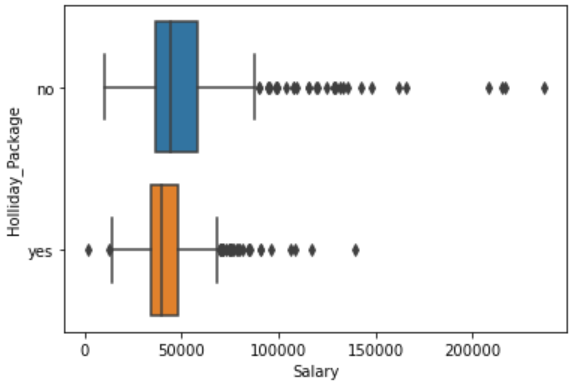
```
data['no_young_children'].value_counts(normalize=True)
```

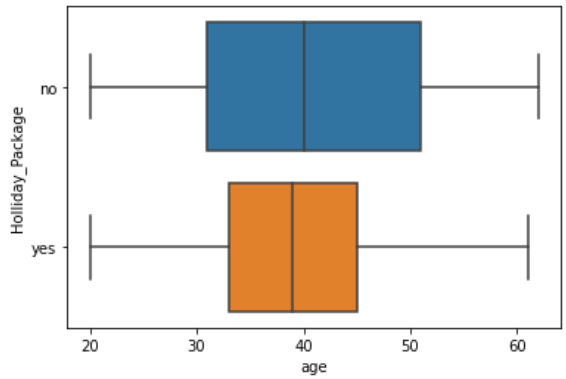
```
0    0.762615
1    0.168578
2    0.063073
3    0.005734
Name: no_young_children, dtype: float64
```

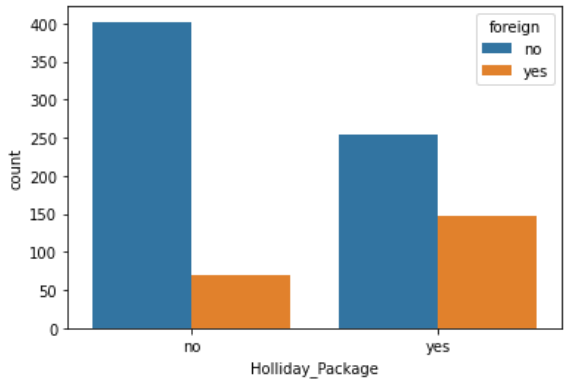
- Amongst the group of people who have kids below age group of 7 years, 76% of the population have no kid followed by 16% with 1 kid and 6% with 2 kids. Only 0.5 % of people have 3 kids or more.
- Amongst the group of people who have kids above the age group of 7 years, 45% of the population have no kid followed by 23% with 2 kids and 22% with 1 kid.
- Of the total population, 45% of the people have opted for Holiday package and 54% have not opted the package.
- Of the total population, about 25% of the population are foreigners and the rest 75% are local

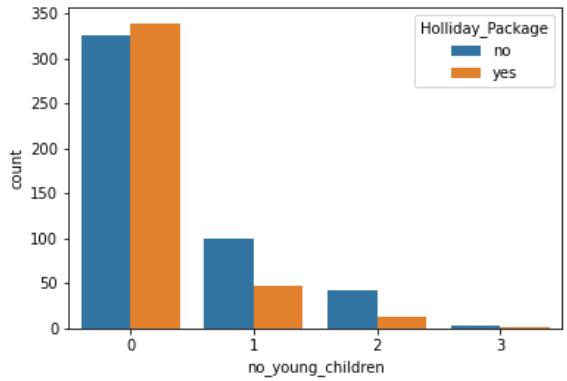
Bivariate Analysis

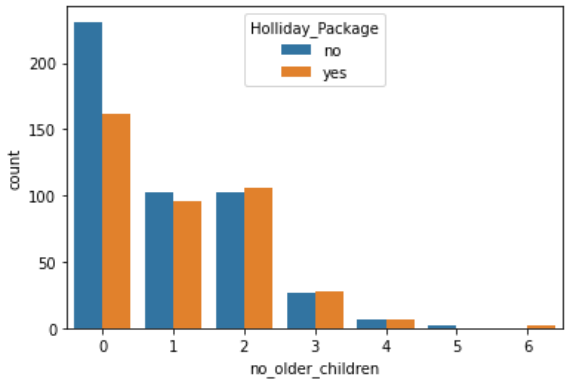












1. People with about 8, 9 years of formal education have opted for Holiday Package.

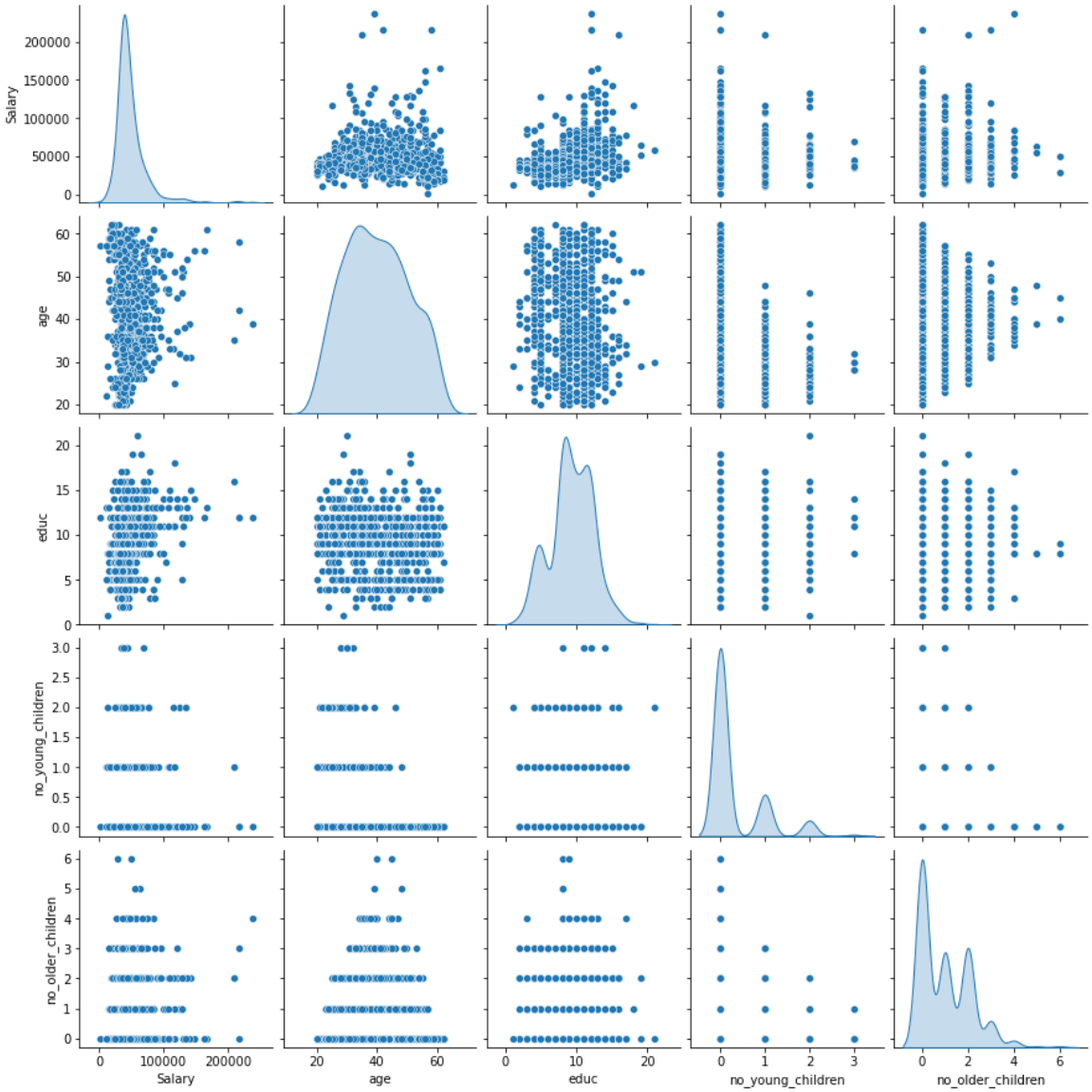
2. Amongst the group of people who have Salary bands in the range of about 40,000 to 50,000 have opted for the Holiday package and generally people above the band of 50,000 has not opted the package though there are some exceptions to this statement.

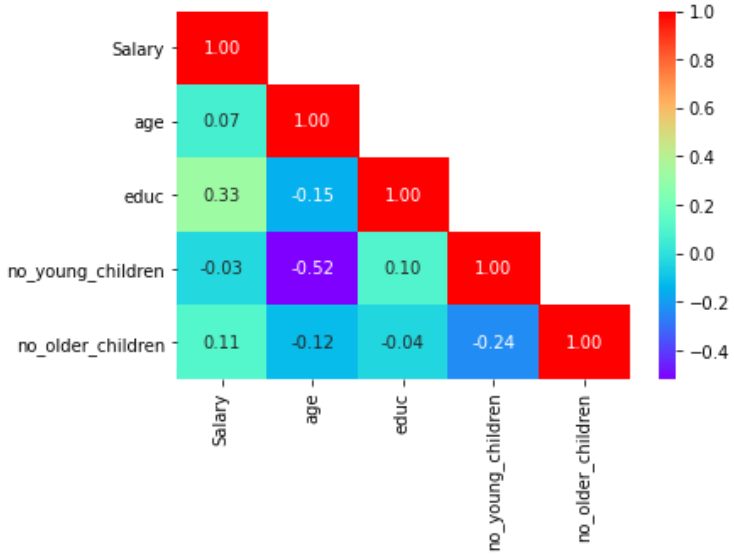
3. People in the age group range of 35 years to 45 years have opted for Holiday Package; People above the age group of 45 years have not opted for Holiday Package though there are some exceptions to it

4. Amongst the Foreigners population, the number of people who have opted of Holiday package is more than those who have not opted for it.

5. People who have no young children (below the age group of 7 years) have opted the Holiday package followed by people with 1 kid and 2 kids.
6. People who have no children (above the age group of 7 years) have opted the Holiday package followed by 2 kids and 1 kid.

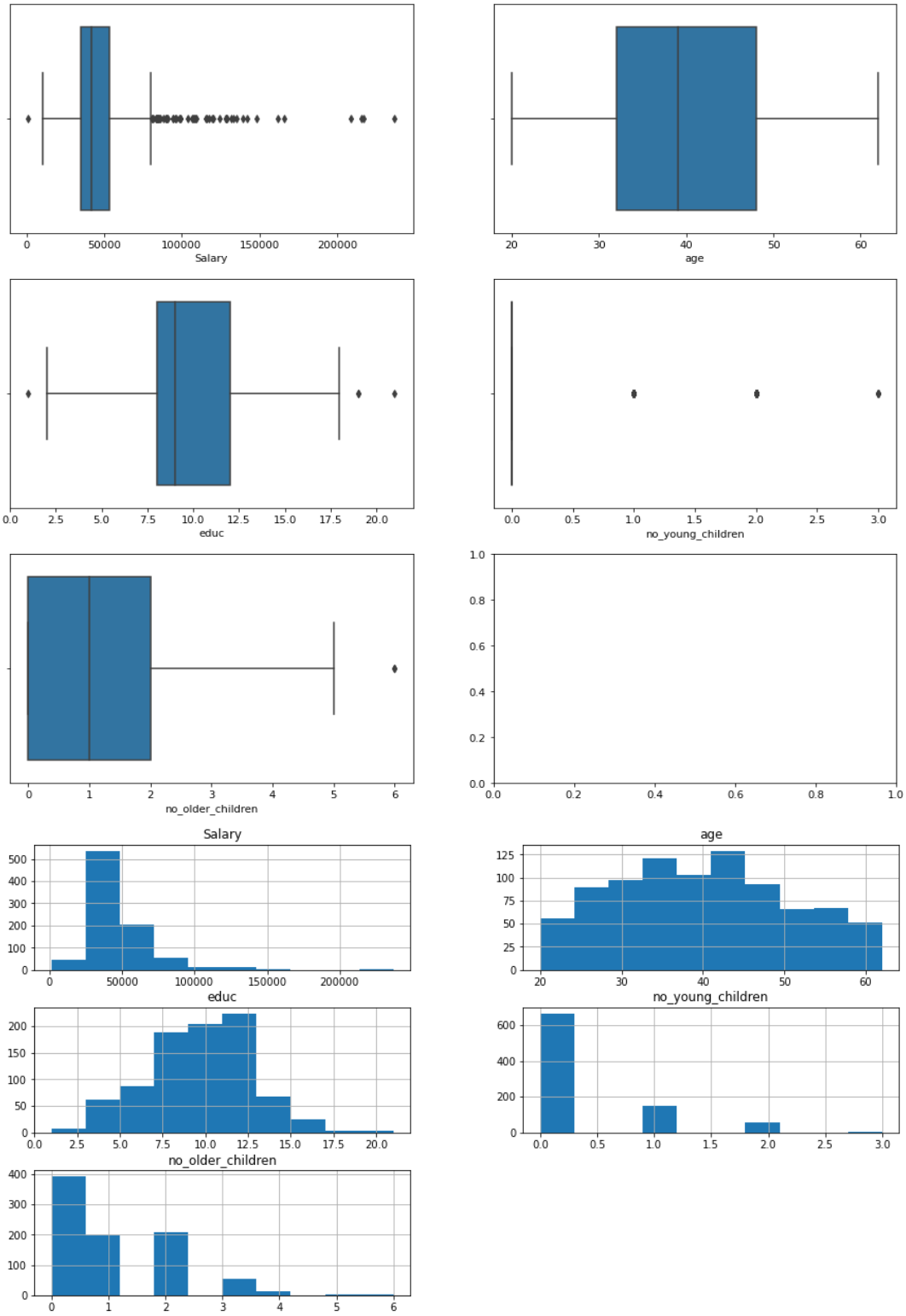
Pair plot:





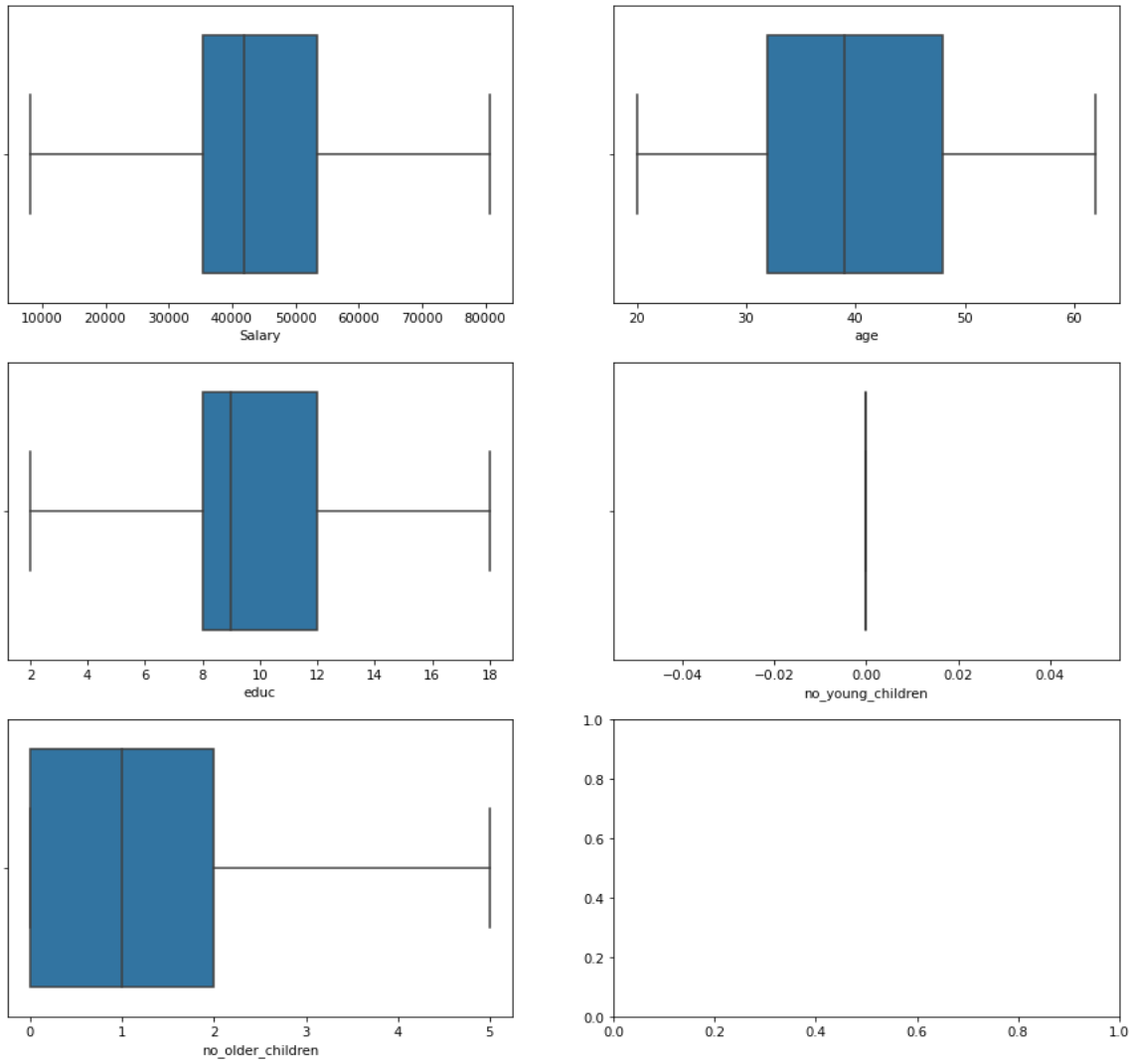
None of the variables have co-relation with each other.

Box Plot with Outliers



- Salary, age, educ variable is almost normally distributed and educ is left skewed
- No_young_Children, no_older_children variable is not continuously distributed

Box Plot without Outliers



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).

Label Encoding:

```
## Converting the 'Holiday Package' Variable into numeric by using the LabelEncoder functionality inside sklearn.
from sklearn.preprocessing import LabelEncoder

## Defining a Label Encoder object instance
LE = LabelEncoder()
```

```
## Applying the created Label Encoder object for the target class
## Assigning the 0 to people who have NOT opted for Holiday Package and 1 to people who have opted for Holiday Package

data['Holliday_Package'] = LE.fit_transform(data['Holliday_Package'])
data.head()
```

	Unnamed: 0	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
0	1.0	0	48412.0	30.0	8.0	0.0	1.0	no
1	2.0	1	37207.0	45.0	8.0	0.0	1.0	no
2	3.0	0	58022.0	46.0	9.0	0.0	0.0	no
3	4.0	0	66503.0	31.0	11.0	0.0	0.0	no
4	5.0	0	66734.0	44.0	12.0	0.0	2.0	no

Converting categorical to dummy variables

```
data= pd.get_dummies(data,columns=['foreign'], drop_first=True)
```

```
data.head()
```

	Unnamed: 0	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign_yes
0	1.0	0	48412.0	30.0	8.0	0.0	1.0	0
1	2.0	1	37207.0	45.0	8.0	0.0	1.0	0
2	3.0	0	58022.0	46.0	9.0	0.0	0.0	0
3	4.0	0	66503.0	31.0	11.0	0.0	0.0	0
4	5.0	0	66734.0	44.0	12.0	0.0	2.0	0

Train Test Split

```
# Copy all the predictor variables into X dataframe
X = data.drop('Holliday_Package', axis=1)

# Copy target into the y dataframe.
y = data[['Holliday_Package']]
```

```
# Split X and y into training and test set in 70:30 ratio
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30 , random_state=1,stratify=data['Holliday_Package'])
```

Number of rows and columns of the training set for the independent variables: (610, 7)
Number of rows and columns of the training set for the dependent variable: (610, 1)
Number of rows and columns of the test set for the independent variables: (262, 7)
Number of rows and columns of the test set for the dependent variable: (262, 1)

Logistic Regression Model:

```
# Fit the model on original data i.e. before upsampling
model = LogisticRegression()
model.fit(X_train, y_train)
#Predicting on Training and Test dataset
ytrain_predict = model.predict(X_train)
ytest_predict = model.predict(X_test)
model_score = model.score(X_test, y_test)
```

Accuracy of the training data:

```
# Accuracy - Training Data
print('Logistic Regression Model Accuracy score of the Training data set {}'.format(model.score(X_train, y_train)))
```

```
Logistic Regression Model Accuracy score of the Training data set 0.6344262295081967
```

Accuracy of the test data:

```
#Accuracy - Test Data
print('Logistic Regression Model Accuracy score of the Test data set {}'.format(model.score(X_test, y_test)))
```

```
Logistic Regression Model Accuracy score of the Test data set 0.6564885496183206
```

LDA – LINEAR DISCRIMINANT ANALYSIS

```
#Build LDA Model
clf = LinearDiscriminantAnalysis()
model=clf.fit(X_train,y_train)
```

```
# Training Data Class Prediction with a cut-off value of 0.5
pred_class_train = model.predict(X_train)

# Test Data Class Prediction with a cut-off value of 0.5
pred_class_test = model.predict(X_test)
```

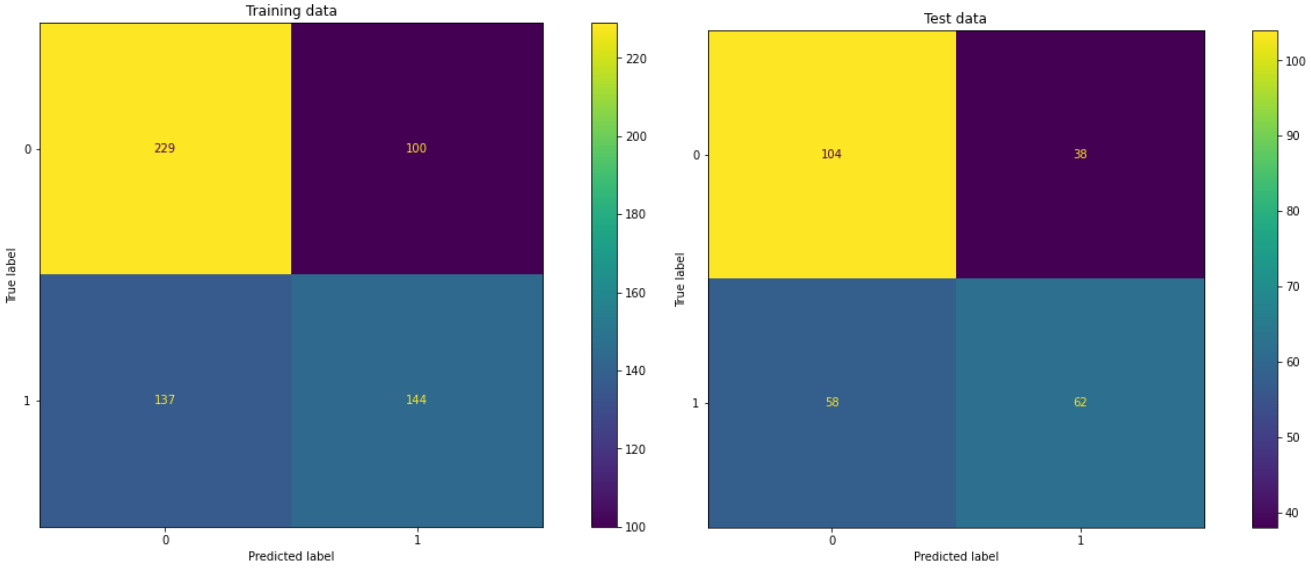
2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model Final Model: Compare Both the models and write inference which model is best/optimized.

Classification Report:

```
Classification Report of the Logistic Regression model training data:

              precision    recall  f1-score   support

           0       0.63      0.70      0.66       329
           1       0.59      0.51      0.55       281

    accuracy                           0.61       610
   macro avg       0.61      0.60      0.60       610
weighted avg       0.61      0.61      0.61       610


Classification Report of the Logistic Regression model test data:

              precision    recall  f1-score   support

           0       0.64      0.73      0.68       142
           1       0.62      0.52      0.56       120

    accuracy                           0.63       262
   macro avg       0.63      0.62      0.62       262
weighted avg       0.63      0.63      0.63       262
```

Confusion Matrix of Logistic Regression model:

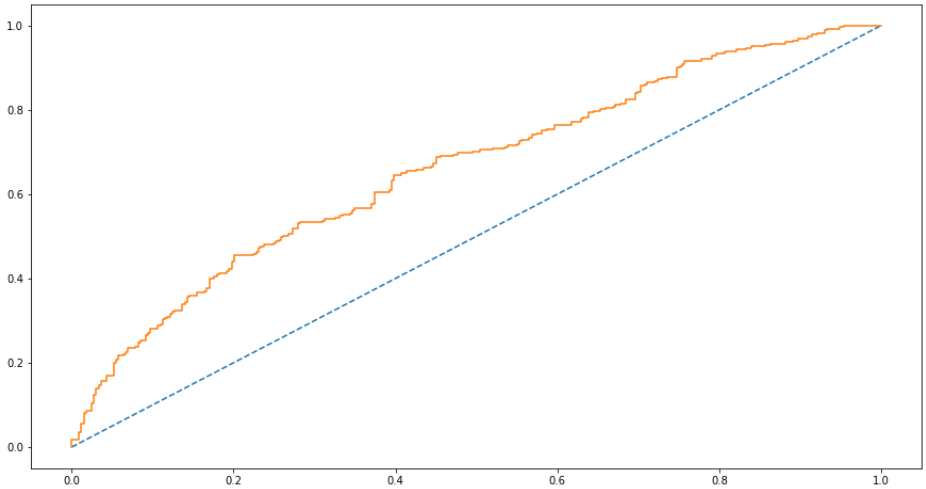
```
Confusion matrix of the Logistic Regression training data
[[229 100]
[137 144]]

Confusion matrix of the Logistic Regression test data
[[104  38]
[ 58  62]]
```



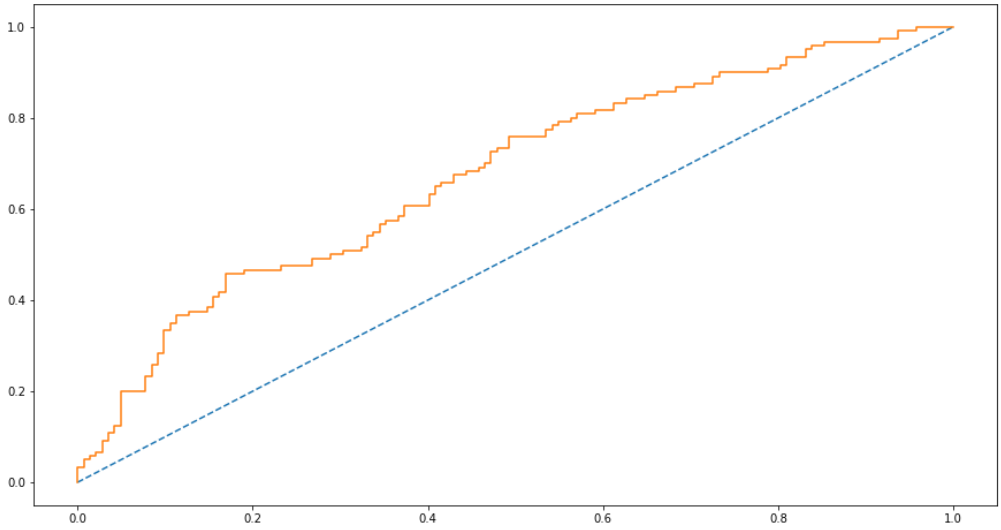
Accuracy score of the training data and test data in the Logistic Regression model score is similar. Hence it is a good fit model.

AUC OF TRAINING DATA:

AUC score of Training Data: 0.661

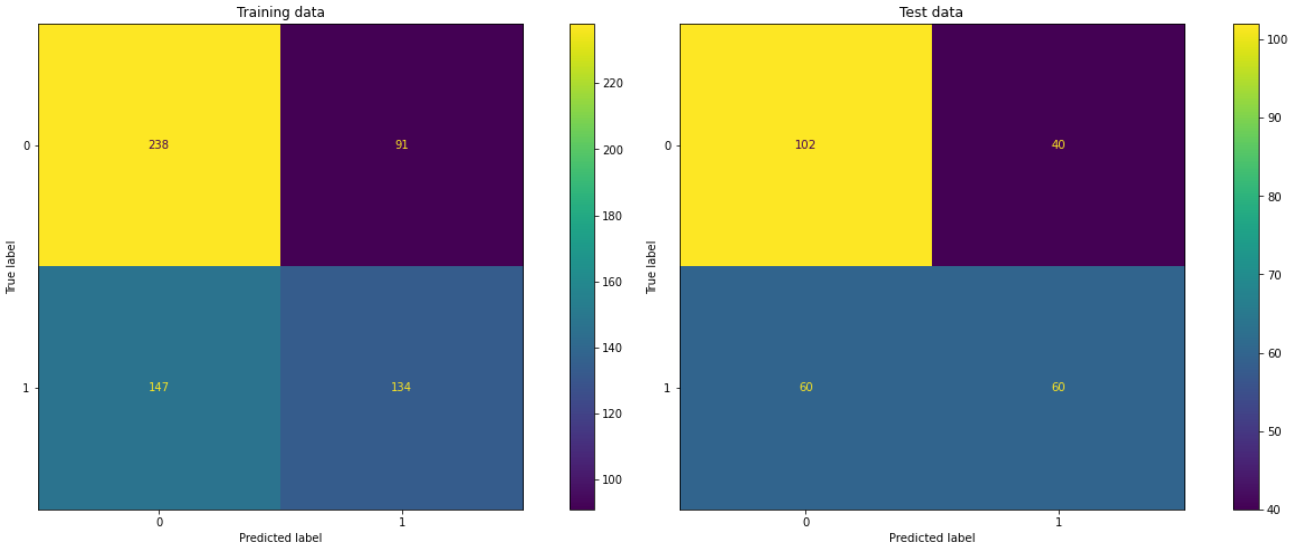


AUC score of Test data: 0.661



AUC score of Training data and Test data is SAME, hence it a good fit model.

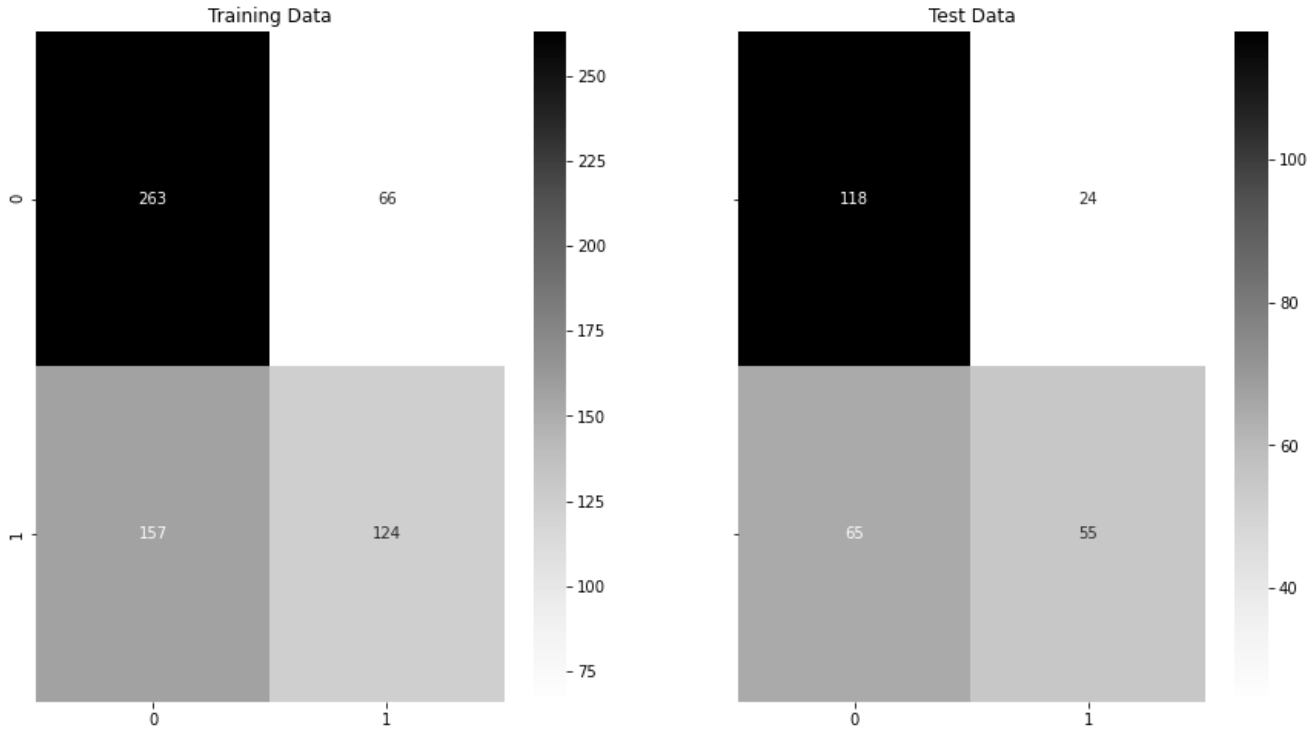
After applying GridSearch CV for Logistic regression model to find the best parameters for tuning, here is the updated confusion matrix



LDA – LINEAR DISCRIMINANT ANALYSIS

```
#Build LDA Model
clf = LinearDiscriminantAnalysis()
model=clf.fit(X_train,y_train)
```

```
# Training Data Class Prediction with a cut-off value of 0.5
pred_class_train = model.predict(X_train)

# Test Data Class Prediction with a cut-off value of 0.5
pred_class_test = model.predict(X_test)
```



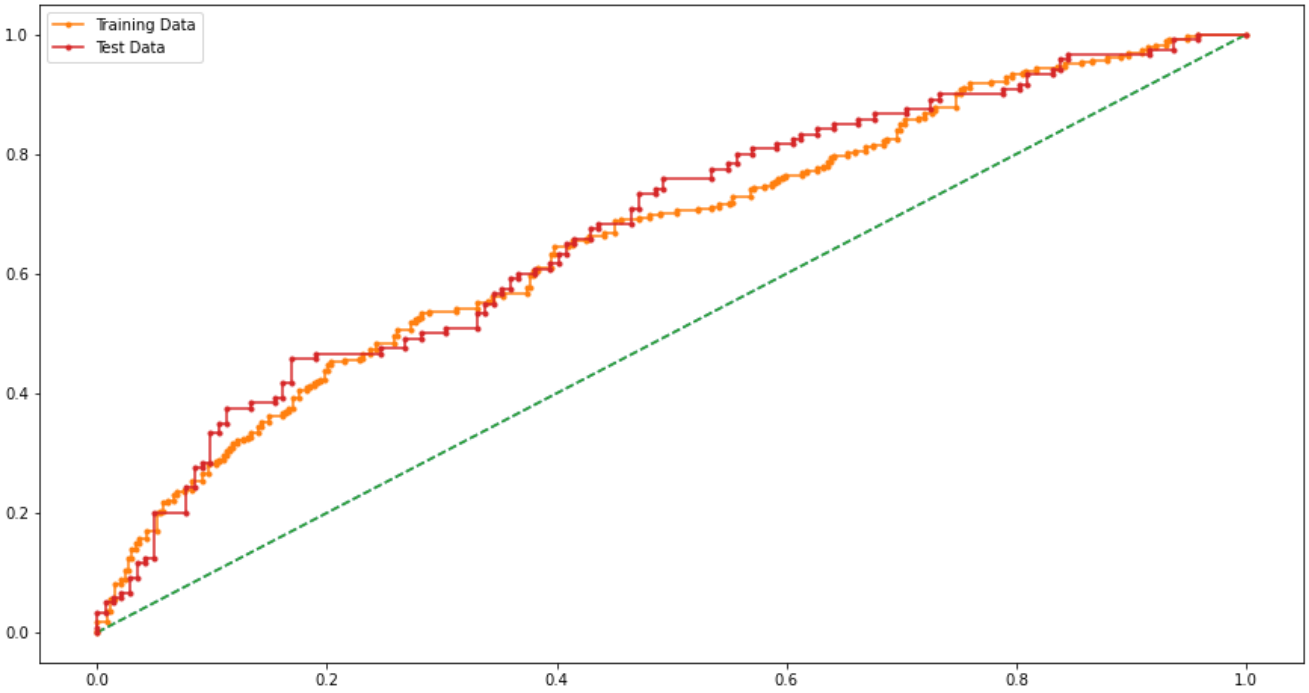
Classification Report of the training data:

	precision	recall	f1-score	support
0	0.63	0.80	0.70	329
1	0.65	0.44	0.53	281
accuracy			0.63	610
macro avg	0.64	0.62	0.61	610
weighted avg	0.64	0.63	0.62	610

Classification Report of the test data:

	precision	recall	f1-score	support
0	0.64	0.83	0.73	142
1	0.70	0.46	0.55	120
accuracy			0.66	262
macro avg	0.67	0.64	0.64	262
weighted avg	0.67	0.66	0.65	262

AUC for the Training Data of LDA model: 0.661
AUC for the Test Data of LDA model: 0.675



Comparing the classification report, confusion matrix, accuracy score and AUC_ROC curve, LDA model is better compared to Logistic regression model.
The test data set follows training data set and LDA model is best optimized.

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

```
# Fit the Logistic Regression model
model = LogisticRegression(solver='newton-cg',max_iter=10000,penalty='none',verbose=True,n_jobs=2)
Class=model.fit(X_train, y_train)
```

```
C:\Users\kpriyadh\AppData\Roaming\Python\Python38\site-packages\sklearn\utils\validation.py:73: Dat
-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), f
  return f(**kwargs)
[Parallel(n_jobs=2)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=2)]: Done   1 out of   1 | elapsed:    1.8s finished
```

```
print(Class.coef_, Class.intercept_)
```

```
[[-8.51719917e-05 -1.92038239e-05 -1.44599906e-02  4.98279814e-02
   0.00000000e+00  1.96884560e-01  1.13652931e+00]] [0.37355537]
```

```
Unnamed: 0 -> -8.51719917e-05
salary -> -1.92038239e-05
age -> -1.44599906e-02
educ -> 4.98279814e-02
no_young_children ->  0.00000000e+00
no_older_children -> 1.96884560e-01
foreign_yes -> 1.13652931e+00
```

The important factors, the company should focus on employees to sell their packages is as follows:

- Amongst the total foreigner population more persons have opted for the holiday package, so immediate target customer / first preference for selling holiday packages goes to Foreigner group, also as the co-efficient for foreigner_yes group is positive.

- Next group of focus customers are those who have no kids, special offers / discount coupons to holiday packages can be given to the ‘Couple’ only group to attract more revenue.

- Next group is employees who have kids in the age group of above 7 years also shares a pie. Employees with 2 kids are greater in population than with 1 kid, so special arrangements for kids can be made by the travel agency like ‘free combo cots’, ‘kids play area’ coupons etc.

- Age of the employee must be prioritized next post which education and salary forms the precedence.