

PREDICTIVE MODELLING PROJECT

M.K. SUGANTHE RAMYA

Case study 1:

**Predicting the price for the
stone using linear regression
model**

Table of contents

 **Project objective**

 **Assumptions**

 **Exploratory data analysis**

- **Summary of the dataset**

- **Bivariate analysis**

 **Converting object data type into categorical**

 **Splitting the data into train and test data**

- **Dimensions on the train and test data**

 **Model building: Linear Regression**

 **Linear regression Model Prediction**

 **Model evaluation**

 **Conclusion**

 **Recommendation**

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.

Dataset for Problem 1: [cubic zirconia.csv](#)

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 best and J the worst.
Clarity	cubic zirconia Clarity refers to the absence of the Inclusions and Blemishes. (In order from Best to Worst, FL = flawless, I3= level 3 inclusions) FL, IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1, I2, I3
Depth	The Height of a 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.

Project Objective:

The Objective of the report is to explore the dataset “[cubic zirconia.csv](#)” in Python (JUPYTER NOTEBOOK) and generate insights about the dataset. This exploration report will consist of the following:

- ✚ Importing the dataset in jupyter notebook.
- ✚ Understanding the structure of dataset.
- ✚ Exploratory Data analysis
- ✚ Graphical exploration
- ✚ Prediction using linear Regression model
- ✚ Insights from the dataset

Assumptions:

Simple linear regression analysis is a technique to find the association between two variables. The two variables involved are a dependent variable which response to the change and the independent variable. Here we are going to predict the price for the stone on the bases of the details given in the dataset using linear regression model which will helps to distinguish between higher and lower profitable stones

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

Exploratory Data Analysis:

cubic_zirconia.csv dataset

	carat	cut	color	clarity	depth	table	x	y	z	price
0	0.3	Ideal	E	SI1	62.1	58	4.27	4.29	2.66	499
1	0.33	Premium	G	IF	60.8	58	4.42	4.46	2.7	984
2	0.9	Very Good	E	VVS2	62.2	60	6.04	6.12	3.78	6289
3	0.42	Ideal	F	VS1	61.6	56	4.82	4.8	2.96	1082
4	0.31	Ideal	F	VVS1	60.4	59	4.35	4.43	2.65	779
5	1.02	Ideal	D	VS2	61.5	56	6.46	6.49	3.99	9502

Information on dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26967 entries, 0 to 26966
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   carat       26967 non-null   float64
1   cut         26967 non-null   object
2   color       26967 non-null   object
3   clarity     26967 non-null   object
4   depth       26270 non-null   float64
5   table       26967 non-null   float64
6   x           26967 non-null   float64
7   y           26967 non-null   float64
8   z           26967 non-null   float64
9   price       26967 non-null   int64
dtypes: float64(6), int64(1), object(3)
memory usage: 2.1+ MB
```

Inference

- The column "Unnamed : 0" is removed from the dataset before proceeding further as its insignificant for the analysis.
- There are 26967 rows and 10 columns
- There are null values in column "depth" and other columns have no null values.
- There are 6 float , 3 object and 1 integer datatypes

Summary of the dataset:

	carat	cut	color	clarity	depth	table	x	y	z	price
count	26967	26967	26967	26967	26270	26967	26967	26967	26967	26967
unique	#N/A	5	7	8	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
top	#N/A	Ideal	G	SI1	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
freq	#N/A	10816	5661	6571	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
mean	0.798375	#N/A	#N/A	#N/A	61.74515	57.45608	5.729854	5.733569	3.538057	3939.518
std	0.477745	#N/A	#N/A	#N/A	1.41286	2.232068	1.128516	1.166058	0.720624	4024.865
min	0.2	#N/A	#N/A	#N/A	50.8	49	0	0	0	326
25%	0.4	#N/A	#N/A	#N/A	61	56	4.71	4.71	2.9	945
50%	0.7	#N/A	#N/A	#N/A	61.8	57	5.69	5.71	3.52	2375
75%	1.05	#N/A	#N/A	#N/A	62.5	59	6.55	6.54	4.04	5360
max	4.5	#N/A	#N/A	#N/A	73.6	79	10.23	58.9	31.8	18818

Inference

- Min value of "x", "y", "z" is zero this indicates that there are faulty values in data that represents a dimensionless or 2-dimensional diamonds. So, we need to filter out those as it clearly faulty data points.

Carat:

- Minimum Carat weight of the cubic zirconia is 0.200000
- Minimum Carat weight of the cubic zirconia is 4.500000

depth:

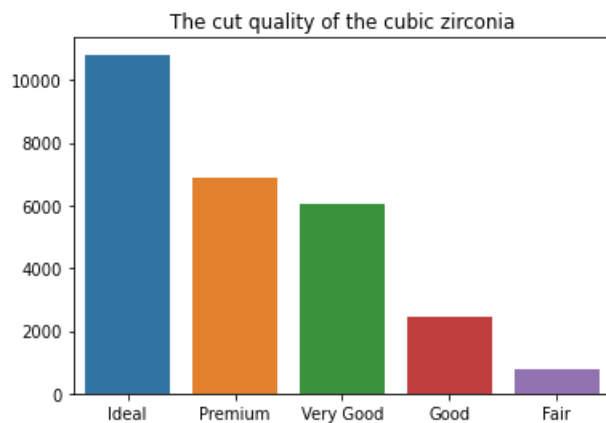
- The minimum Height of a cubic zirconia: 50.8
- Average Height of a cubic zirconia:61.745

price

- price (326--18,818). This is the target column containing tags for the features.
- There are 5 categories in cut variable
- There are 7 categories in colour variable
- There are 8 categories in clarity variable

Unique values for categorical variables

The cut quality of the cubic zirconia



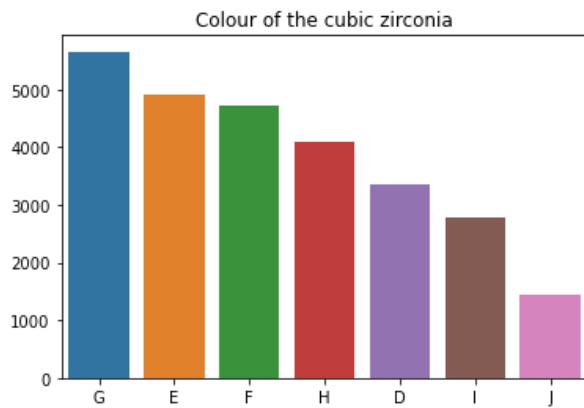
Ideal	0.401083
Premium	0.255831
Very Good	0.223607
Good	0.090518
Fair	0.028961

Name: cut, dtype: float64

Inference

- In determining the quality of the cut, the gemstone grader evaluates the cutter's skill in the fashioning of the gemstone.
- The more precise the gemstone is cut, the more captivating the diamond is to the eye.
- Nearly 40% of gemstone in this dataset is Ideal which is the highest grader cut
- Premium grader cut gemstone is around 25%
- Around .02% of gemstone in this dataset is Fair which is very lower grade cut.

Colour of the cubic zirconia



G 0.209923

E 0.182334

F 0.175362

H 0.152112

D 0.124003

I 0.102755

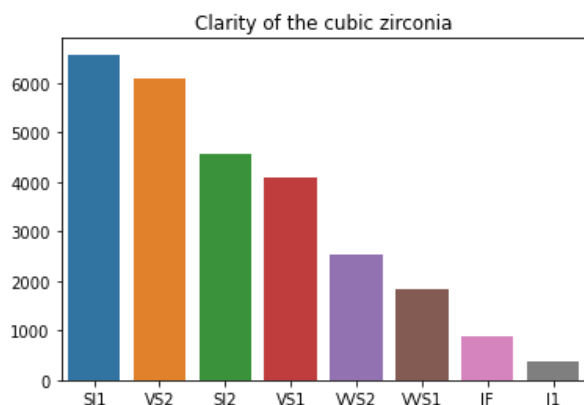
J 0.053510

Name: color, dtype: float64

Inference

- Color, from J (worst) to D (best) The colour of gem-quality gemstone occurs in many hues. In the range from colourless to light yellow or light brown.
- Colourless gemstone are the rarest. Other natural colours (blue, red, pink for example) are known as "fancy," and their colour grading is different than from white colorless gemstones
- Only 12.4% of "D" (colorless) which is considered as best color of gemstone
- Only 5% of "J" which is considered as worst color of gemstone

Clarity of the cubic zirconia



SI1 0.243668

VS2 0.226165

SI2 0.169652

VS1 0.151778

VVS2 0.093855

VVS1 0.068194

IF 0.033152

I1 0.013535

Name: clarity, dtype: float64

Inference

- Clarity of the cubic zirconia("SI1") Slightly Included category: 24.3%
- Clarity of the cubic zirconia("IF") internally flawless category :3.3%

Duplicates in dataset

Number of duplicate rows = 34

Inference

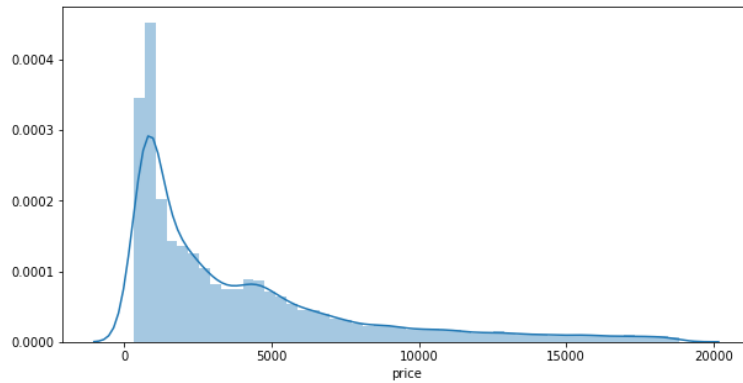
These 34 rows are not exactly pure duplicates (two to three columns are always having different values), but for the purpose of this project, we drop the duplicates.

Before dropping duplicates (26967, 10)

After dropping duplicates (26933, 10)

1.2. Perform Univariate and Bivariate Analysis.

Price Distribution

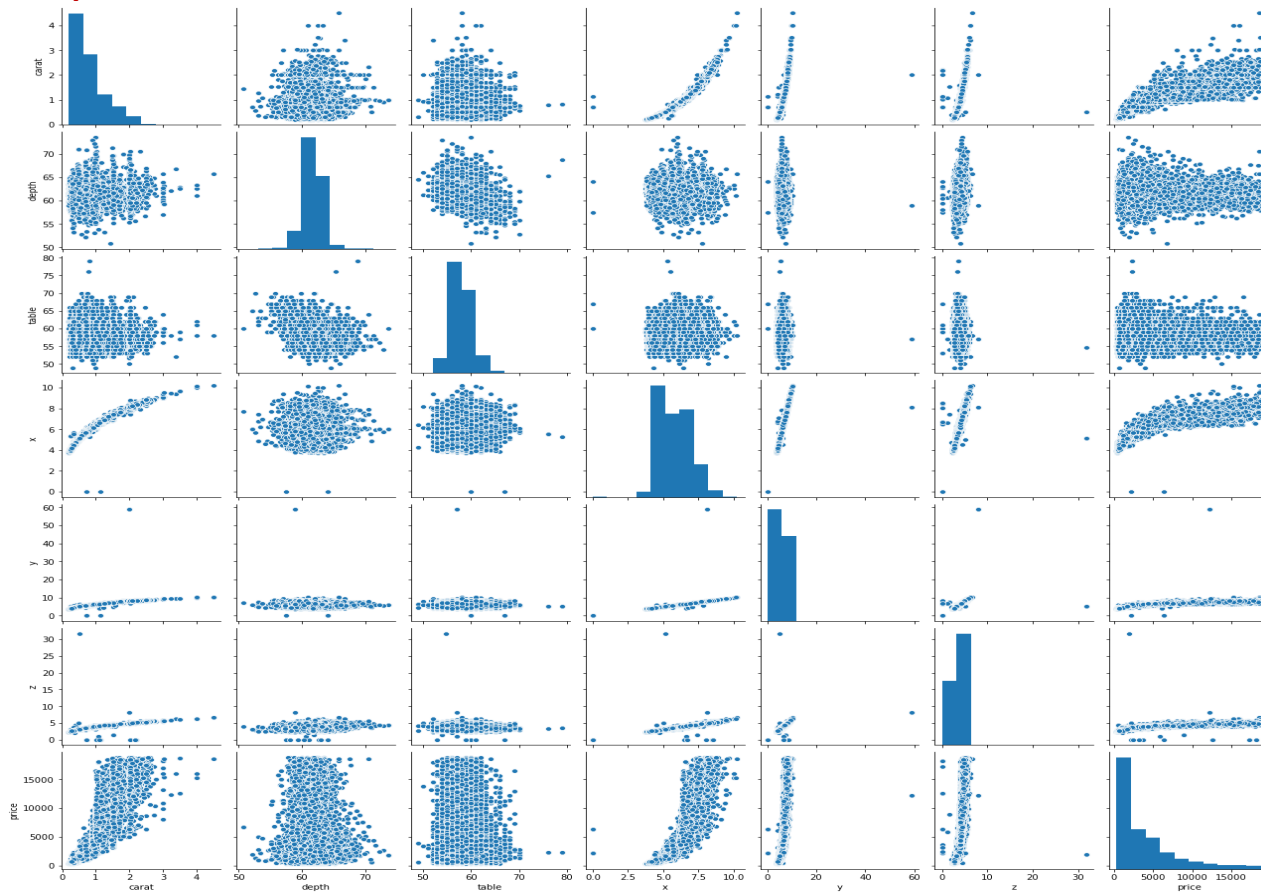


Inference

- Distribution of "price" variable left skewed

Bivariate Analysis:

Pair plot between variables:

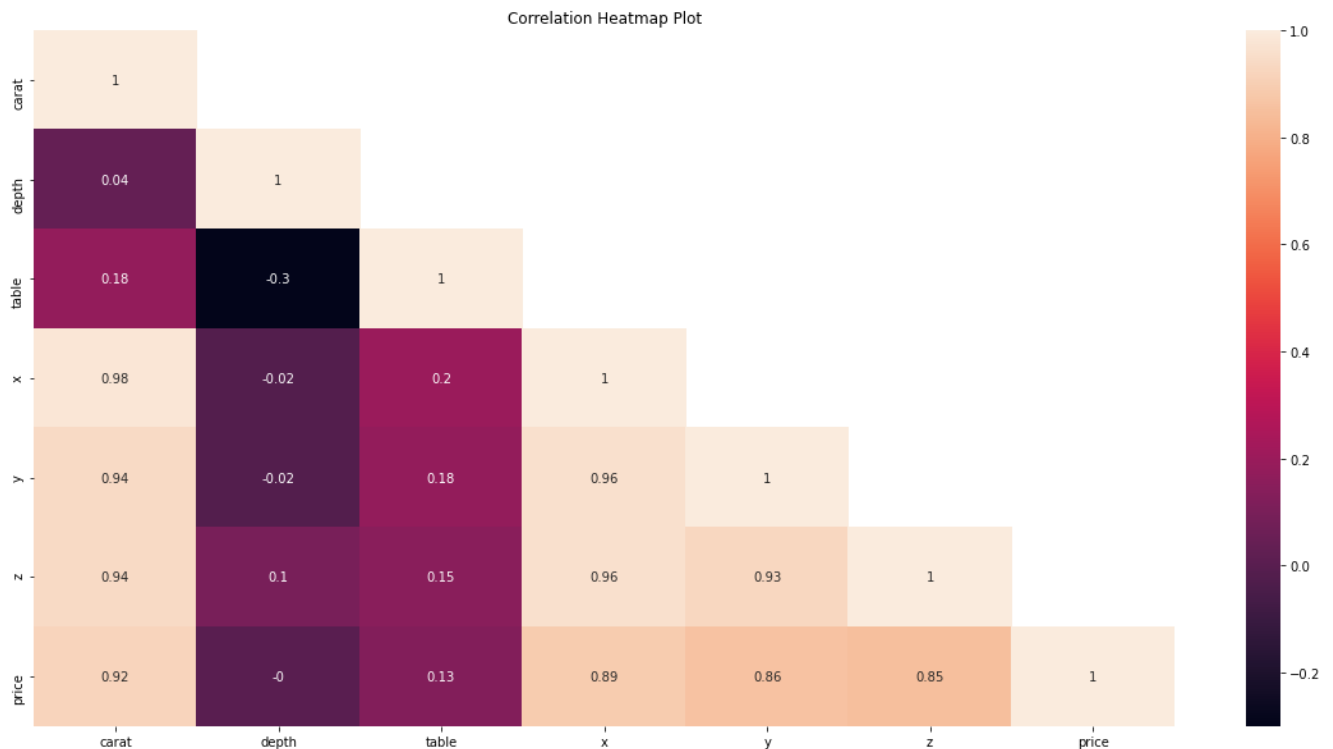


Inference

- Price and carat have linear relationship. other variables are either slight or no linear relationship seen
- Looks like only "x" variable is normally distributed
- Other variables are skewed

Correlation Heatmap:

Correlation Heatmap Plot

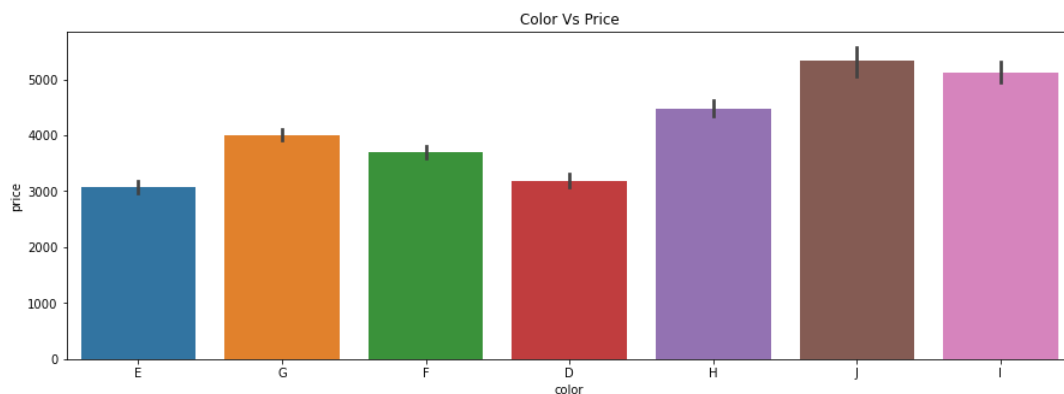


Inference

- The variables "x", "y", "z" is highly correlated to "carat" variable
- The target variables are highly correlated (98%) with "carat" variable which shows us carat variable is particularly important feature
- "x", "y" and "z" variables are highly correlated with each other
- Negative correlation is seen between table and depth (-0.3)

Bivariate Analysis

Color Vs Price



Inference

Although "J" variety is worst color, it is sold more
Best color "D" is best color, is sold less

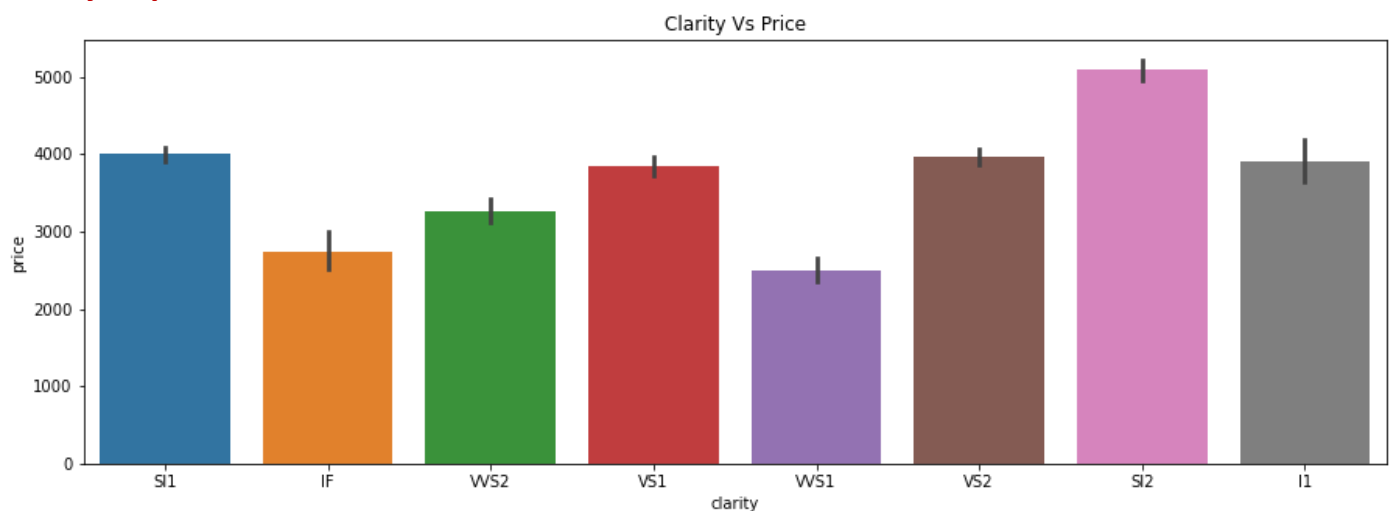
Cut Vs Price



Inference

- "Premium" and "Fair" category is highly sold diamond
- "Ideal" is high grade diamond is sold less

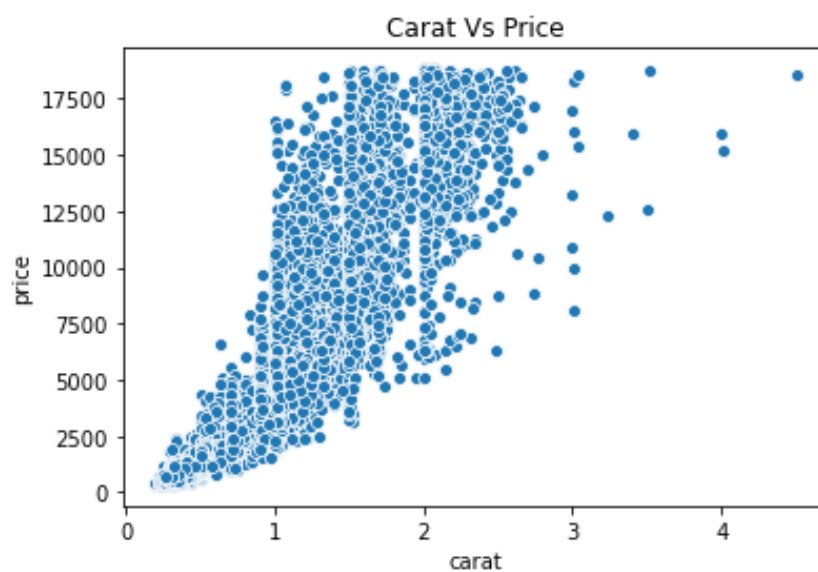
Clarity vs price



Inference

- SI2 clarity diamond is frequently sold diamond

Scatterplot - Carat Vs Price



Inference

There may be linear relationship between Price and carat

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?

check for null values

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

After null values treatment

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

Inference

- There are null values in depth variable, we need to treat the null values. After treating, null values are filled with mean of depth variables

Checking for columns with values as 0

```
[]
[]
[]
[]
[]
[]
[]
[5821, 17506]
[5821, 17506]
[5821, 6034, 10827, 12498, 12689, 17506, 18194, 23758]
[]
```

Inference

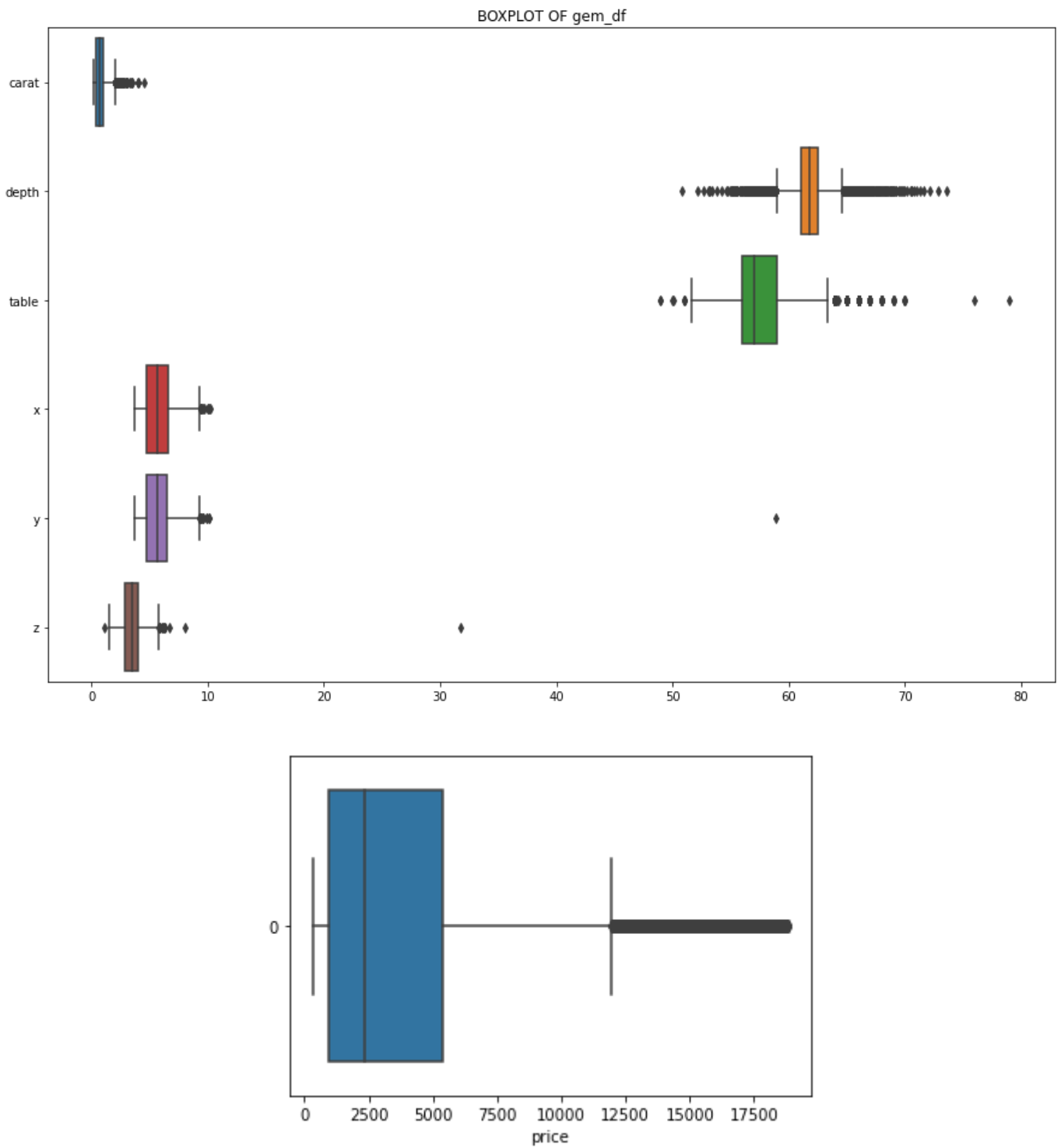
- Min value of "x", "y", "z" is zero this indicates that there are faulty values in data that represents a dimensionless or 2-dimensional diamonds. So, we need to filter out those as it clearly faulty data points.
- we replace "0" with np.nan

	carat	depth	table	x	y	z	price
count	26925	26925	26925	26925	26925	26925	26925
mean	0.797821	61.74557	57.45531	5.729385	5.733152	3.53882	3936.25
std	0.477085	1.39343	2.231327	1.126081	1.16382	0.717483	4020.983
min	0.2	50.8	49	3.73	3.71	1.07	326
25%	0.4	61.1	56	4.71	4.71	2.9	945
50%	0.7	61.8	57	5.69	5.7	3.52	2373
75%	1.05	62.5	59	6.55	6.54	4.04	5353
max	4.5	73.6	79	10.23	58.9	31.8	18818

- After treating 0 with np.nan, There are no zeros in the dataset

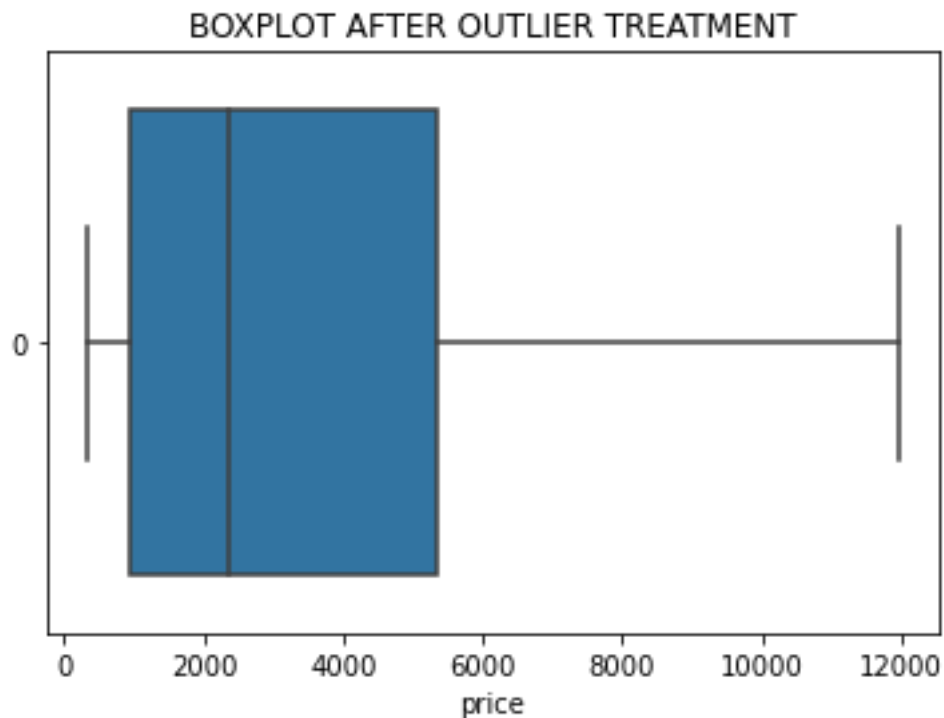
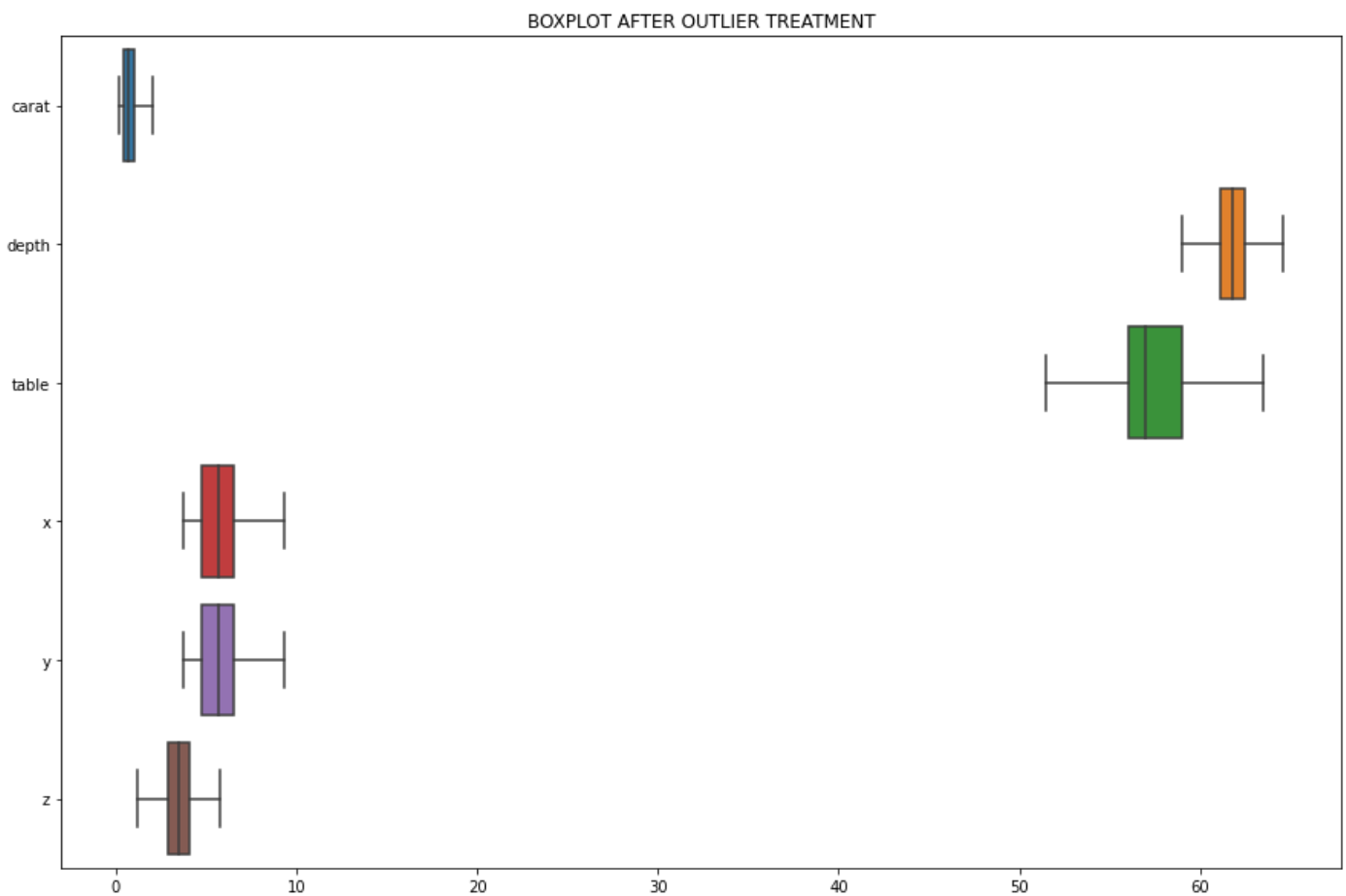
Boxplot

Outliers before treatment



- There are outliers in all the variables

BOXPLOT AFTER OUTLIER TREATMENT



➤ After the outlier treatment, all the outliers are removed

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.

Encode the data (having string values) for Modelling.

	carat	depth	table	x	y	z	price	clarity_IF	clarity_SI1	clarity_SI2	...	cut_Good	cut_Ideal	cut_Premium	cut_verygood	color_E	color_F	color_G	color_H	color_I	color_J
0	0.3	62.1	58	4.27	4.29	2.66	499	0	1	0	...	0	1	0	0	1	0	0	0	0	0
1	0.33	60.8	58	4.42	4.46	2.7	984	1	0	0	...	0	0	1	0	0	0	1	0	0	0
2	0.9	62.2	60	6.04	6.12	3.78	6289	0	0	0	...	0	0	0	1	1	0	0	0	0	0
3	0.42	61.6	56	4.82	4.8	2.96	1082	0	0	0	...	0	1	0	0	0	1	0	0	0	0
4	0.31	60.4	59	4.35	4.43	2.65	779	0	0	0	...	0	1	0	0	0	1	0	0	0	0

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 26925 entries, 0 to 26966
Data columns (total 24 columns):
#   Column                Non-Null Count  Dtype
---  -
0   carat                 26925 non-null  float64
1   depth                 26925 non-null  float64
2   table                 26925 non-null  float64
3   x                     26925 non-null  float64
4   y                     26925 non-null  float64
5   z                     26925 non-null  float64
6   price                 26925 non-null  float64
7   clarity_IF            26925 non-null  uint8
8   clarity_SI1           26925 non-null  uint8
9   clarity_SI2           26925 non-null  uint8
10  clarity_VS1           26925 non-null  uint8
11  clarity_VS2           26925 non-null  uint8
12  clarity_VVS1          26925 non-null  uint8
13  clarity_VVS2          26925 non-null  uint8
14  cut_Good              26925 non-null  uint8
15  cut_Ideal             26925 non-null  uint8
16  cut_Premium           26925 non-null  uint8
17  cut_verygood          26925 non-null  uint8
18  color_E               26925 non-null  uint8
19  color_F               26925 non-null  uint8
20  color_G               26925 non-null  uint8
21  color_H               26925 non-null  uint8
22  color_I               26925 non-null  uint8
23  color_J               26925 non-null  uint8
dtypes: float64(7), uint8(17)
memory usage: 3.3 MB
```

Inference:

- `get_dummies`: Convert categorical variable into dummy/indicator variables. Dummy coding is mainly used for including nominal and ordinal variables in linear regression analysis. Since such variables do not have a fixed unit of measurement, assuming a linear relation between them and an outcome variable does not make sense.
- Using `get_dummies`, all categorical variables are converted into Dummy-coded data
- Here, `get_dummies` are used, because of ordinal variables in categorical
- After conversion, all variables are now numerical variables

DataFrame Scaling

	carat	depth	table	x	y	z	price	clarity_IF	clarity_SI1	clarity_SI2	...	cut_Good	cut_Ideal	cut_Premium	cut_verygood	color_E	color_F	color_G	color_H	color_I	color_J
0	-1.067382	0.287935	0.261968	-1.29653	-1.289659	-1.261558	-0.933395	-0.184999	1.761227	-0.451601	...	-0.315251	1.221434	-0.585856	-0.53703	2.115896	-0.461166	-0.515335	-0.423276	-0.338298	-0.237705
1	-1.002446	-0.779219	0.261968	-1.163253	-1.13753	-1.20406	-0.793477	5.405447	-0.567786	-0.451601	...	-0.315251	-0.81871	1.706903	-0.53703	-0.472613	-0.461166	1.940486	-0.423276	-0.338298	-0.237705
2	0.231349	0.370024	1.189326	0.276134	0.347964	0.348406	0.73696	-0.184999	-0.567786	-0.451601	...	-0.315251	-0.81871	-0.585856	1.862095	2.115896	-0.461166	-0.515335	-0.423276	-0.338298	-0.237705
3	-0.807636	-0.122509	-0.66539	-0.807849	-0.833272	-0.830318	-0.765205	-0.184999	-0.567786	-0.451601	...	-0.315251	1.221434	-0.585856	-0.53703	-0.472613	2.168417	-0.515335	-0.423276	-0.338298	-0.237705
4	-1.045737	-1.107574	0.725647	-1.225449	-1.164377	-1.275933	-0.852618	-0.184999	-0.567786	-0.451601	...	-0.315251	1.221434	-0.585856	-0.53703	-0.472613	2.168417	-0.515335	-0.423276	-0.338298	-0.237705

Inference

- Z-score is a variation of scaling that represents the number of standard deviations away from the mean.
- z-score to ensure your feature distributions have mean = 0 and std = 1.
- Scaling done to whole dataset using z-score scaler because carat, depth, table and price has different scales .x, y, z variables have single digit unit
- This will affect the model. Scaling will make dataset are not biases and all data are at same level. so, it is important to scale the data

Train-Test Split

- Separating independent (train) and dependent (test)variables for the linear regression model
- X = independent (train) variables
- Y = dependent (test)variables
- The training set for the independent variables: (18847, 23)
- The training set for the dependent variable: (18847, 1)
- The test set for the independent variables: (8078, 23)
- The test set for the dependent variable: (8078, 1)
- spilting the dataset into train and test set to build Linear regression model (70:30)
- X_train :70% of data randomly chosen from the 23 columns. These are training independent variables
- X_test :30% of data randomly chosen from the 23 columns. These are test independent variables
- # y_train :70% of data randomly chosen from the "price" column. These are training dependent variables
- # y_test :30% of data randomly chosen from the "price" columns. These are test dependent variables

Linear Model from Sklearn library

Model building: Linear Regression

Linear regression is a linear model, e.g., a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).

Here, two types of library are used

1. Sklearn learn

2. statsmodels

Fit the model to the training set

We now fit our model to the linear regression model by training the model with our independent variable and dependent variables.

The coefficients for each of the independent attributes

```
The coefficient for carat is 1.2262124015885945
The coefficient for depth is 0.004353241032529015
The coefficient for table is -0.014362465653953445
The coefficient for x is -0.382284181088136
The coefficient for y is 0.3489222184258426
The coefficient for z is -0.12853588593174087
The coefficient for clarity_IF is 0.20616926599894056
The coefficient for clarity_SI1 is 0.31411709427489776
The coefficient for clarity_SI2 is 0.185279232333912
The coefficient for clarity_VS1 is 0.3472721324455984
The coefficient for clarity_VS2 is 0.3708294790623183
The coefficient for clarity_VVS1 is 0.27486279036869943
The coefficient for clarity_VVS2 is 0.3170706673107511
The coefficient for cut_Good is 0.03203937066436818
The coefficient for cut_Ideal is 0.08906993313935459
The coefficient for cut_Premium is 0.07532878397570261
The coefficient for cut_verygood is 0.060411638656746244
The coefficient for color_E is -0.02105022143782283
The coefficient for color_F is -0.02536843505101097
The coefficient for color_G is -0.0482904959618329
The coefficient for color_H is -0.08610959234164053
The coefficient for color_I is -0.11648250106508748
The coefficient for color_J is -0.120833333476361
```

The coefficients for each of the independent attributes

- The sign of a regression coefficient tells you whether there is a positive or negative correlation between each independent variable the dependent variable.
- A positive coefficient indicates that as the value of the independent variable increases, the mean of the dependent variable also tends to increase.
- A negative coefficient suggests that as the independent variable increases, the dependent variable tends to decrease.

- The coefficient for carat is 1.22 which shows "carat" variable is high positive correlation with "price" (dependent variables)
- If Carat of diamond increases, the price of diamond increases
- There is more negative correlation between "price " and x, z variables .so, x(length) and z(height) increases, price of diamond decreases

Score on training and test set:

Model score - R2 or coeff of determinant

$$R^2 = 1 - \text{RSS} / \text{TSS} = \text{RegErr} / \text{TSS}$$

- The coefficient of determination R^2 of the prediction on Train set 0.9404719027464119
- The coefficient of determination R^2 of the prediction on Test set 0.9416169664411842

Inference

- The most common interpretation of r-squared is how well the regression model fits the observed data.
- Accuracy of both training and test dataset is 94% and it shows that there is no overfitting of the data
- r-squared of 94% reveals that 94% of the data fit the regression model. Generally, a higher
- r-squared indicates a better fit for the model.

RMSE on Training data

The Root Mean Square Error (RMSE) of the model is for training set is 0.24341465472
The Root Mean Square Error (RMSE) of the model is for testing set is 0.24293360896

Inference

- The RMSE is the square root of the variance of the residuals.
- As the square root of a variance, RMSE can be interpreted as the standard deviation of the unexplained variance and has the useful property of being in the same units as the response variable.
- Lower values of RMSE indicate better fit.
- Here, The Root Mean Square Error (RMSE) of the model is for training and testing set is 24%
- RMSE value ≥ 0.5 reflects the poor ability of the model to accurately predict the data.
- Since both, the Training and Testing data both have RMSE value lesser than 0.5 we can clearly see that the model built is good to go

Multi-collinearity using VIF

```
carat ---> 33.36253335577607
depth ---> 4.56787352371706
table ---> 1.8068714798640795
x ---> 467.49864261811786
y ---> 474.49691271700016
z ---> 240.90385842104675
clarity_IF ---> 3.5704318660243244
clarity_SI1 ---> 15.004036090719572
clarity_SI2 ---> 11.56856412272262
clarity_VS1 ---> 10.913390573779143
clarity_VS2 ---> 14.39844247861095
clarity_VVS1 ---> 6.054298242493809
clarity_VVS2 ---> 7.6653943644166835
cut_Good ---> 4.097082396915314
cut_Ideal ---> 11.357092991336216
```

Inference

- A variance inflation factor (VIF) detects multicollinearity in regression analysis. Multicollinearity is when there's correlation between predictors (i.e., independent variables) in a model; its presence can adversely affect your regression results.
- The VIF estimates how much the variance of a regression coefficient is inflated due to multicollinearity in the model.
- table, clarity_IF, cut_Good, depth are moderately correlated
- Others have huge VIF values which shows that there is a heavy influence on multicollinearity which needs to be treated.

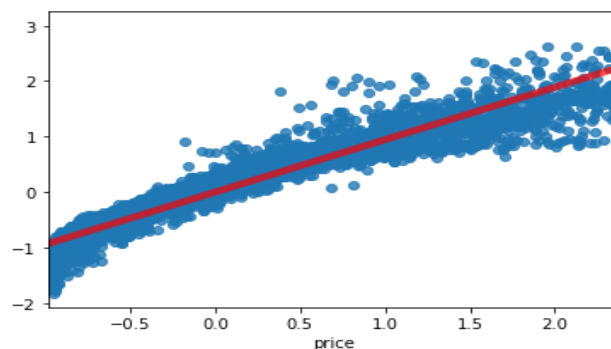
Intercept for the model

The intercept for our model is -0.00014727

Inference

- The intercept (often labelled as constant) is the point where the function crosses the y-axis.
- The negative intercept shows where the linear model predicts price (y) would be when subs (x) are 0.

Scatter Plot (Actual and Predicted Values):



- Since this is regression model, plot between predicted y value vs actual y values of the test data
- A good model prediction will be close to actual leading to high R and R2 values
- The above scatter plot is based on the scaled data and so range for both the Predicted and the actual Prices is between -2 to +2. T
- The X- axis is the "Actual" Price of the test data and the Y axis is the "Predicted" values of the test data.
- Because of the huge volumes of test data, the scatter plot seems to be too crowded which we can be corrected by decreasing the test data amount considerably

Linear Regression using statsmodels

Linear models with independently and identically distributed errors, and for errors with heteroscedasticity or autocorrelation. This module allows estimation by ordinary least squares (OLS), weighted least squares (WLS), generalized least squares (GLS), and feasible generalized least squares with autocorrelated AR(p) errors.

OLS : ORDINARY LEAST SQ

```
Intercept      -0.000134
clarity_IF      0.206186
clarity_SI1     0.314415
clarity_SI2     0.185483
clarity_VS1     0.347412
clarity_VS2     0.371054
clarity_VVS1    0.274940
clarity_VVS2    0.317189
cut_Good        0.032349
cut_Ideal       0.088552
cut_Premium     0.074859
cut_verygood    0.060283
color_E         -0.021054
color_F         -0.025340
color_G         -0.048240
color_H         -0.086060
color_I         -0.116402
color_J         -0.120803
carat           1.227296
table          -0.014927
x               -0.391184
y               0.331633
z              -0.103267
dtype: float64
```

OLS summary

OLS Regression Results

```
=====
Dep. Variable:          price      R-squared:          0.940
Model:                  OLS        Adj. R-squared:       0.940
Method:                 Least Squares    F-statistic:       1.352e+04
Date:                  Sun, 07 Mar 2021    Prob (F-statistic): 0.00
Time:                  11:52:16      Log-Likelihood:    -112.83
No. Observations:      18847        AIC:              271.7
Df Residuals:          18824        BIC:              452.1
Df Model:               22
Covariance Type:       nonrobust
=====
```

```
=====
              coef      std err          t      P>|t|      [0.025      0.975]
-----
Intercept    -0.0001      0.002     -0.076      0.940     -0.004      0.003
clarity_IF    0.2062      0.003     61.560      0.000      0.200      0.213
clarity_SI1   0.3144      0.007     45.704      0.000      0.301      0.328
clarity_SI2   0.1855      0.006     30.685      0.000      0.174      0.197
clarity_VS1   0.3474      0.006     59.122      0.000      0.336      0.359
clarity_VS2   0.3711      0.007     54.989      0.000      0.358      0.384
clarity_VVS1  0.2749      0.004     62.782      0.000      0.266      0.284
clarity_VVS2  0.3172      0.005     64.423      0.000      0.308      0.327
cut_Good       0.0323      0.004      8.904      0.000      0.025      0.039
cut_Ideal      0.0886      0.006     14.654      0.000      0.077      0.100
cut_Premium    0.0749      0.005     14.534      0.000      0.065      0.085
cut_verygood   0.0603      0.005     11.875      0.000      0.050      0.070
color_E        -0.0211      0.003     -8.320      0.000     -0.026     -0.016
color_F        -0.0253      0.003    -10.035      0.000     -0.030     -0.020
color_G        -0.0482      0.003    -18.250      0.000     -0.053     -0.043
color_H        -0.0861      0.002    -34.631      0.000     -0.091     -0.081
color_I        -0.1164      0.002    -49.701      0.000     -0.121     -0.112
color_J        -0.1208      0.002    -56.808      0.000     -0.125     -0.117
carat         1.2273      0.010    119.461      0.000      1.207      1.247
table         -0.0149      0.002     -6.387      0.000     -0.020     -0.010
x             -0.3912      0.044     -8.957      0.000     -0.477     -0.306
y              0.3316      0.042      7.882      0.000      0.249      0.414
z            -0.1033      0.015     -6.717      0.000     -0.133     -0.073
=====
```

```
=====
Omnibus:          4646.699    Durbin-Watson:          2.002
Prob(Omnibus):    0.000      Jarque-Bera (JB):       17384.030
Skew:             1.198      Prob(JB):               0.00
Kurtosis:         7.049      Cond. No.:              70.4
=====
```

Warnings:

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

Inference

- Adjusted. R-squared reflects the fit of the model. R-squared values range from 0 to 1, where a higher value generally indicates a better fit, assuming certain conditions are met.
- Adj. R-squared: 0.940 indicates good fit
- coefficient represents the change in the output Y due to a change of one unit in the unemployment rate (everything else held constant)
- carat is highly correlated with y(price)
- std err reflects the level of accuracy of the coefficients. Most of variables has lower levels.

- The lower it is, the higher is the level of accuracy
- $P > |t|$ is your p-value. A p-value of less than 0.05 is statistically significant
- expect depth, all other variables have p-value of less than 0.05 is considered to be statistically significant
- Confidence Interval for dataset is $[0.025 \quad 0.975]$ which represents the range in which our coefficients are likely to fall (with a likelihood of 95%)
- const coefficient is your Y-intercept. It means that if both the Interest_Rate and Unemployment_Rate coefficients are zero, then the expected output (i.e., the Y) would be equal to the const coefficient.
- Interest_Rate coefficient represents the change in the output Y due to a change of one unit in the interest rate (everything else held constant)

Residual's check:

```
11971    1.334660
3294     0.440258
25427    2.735538
709      2.369388
8010     1.536468
...
17455    0.198454
26170    0.175127
22115    -1.150356
2275     -0.923435
25166    0.382826
Length: 8078, dtype: float64
```

Inference

A residual is the vertical distance between a data point and the regression line. Each data point has one residual. They are positive if they are above the regression line and negative if they are below the regression line. If the regression line passes through the point, the residual at that point is zero.

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

Best attributes of this dataset

```
(-0.0) * Intercept + (0.21) * clarity_IF + (0.31) * clarity_SI1 + (0.19) * clarity_SI2
+ (0.35) * clarity_VS1 + (0.37) * clarity_VS2 + (0.27) * clarity_VVS1 + (0.32) * clarit
y_VVS2 + (0.03) * cut_Good + (0.09) * cut_Ideal + (0.08) * cut_Premium + (0.06) * cut_v
erygood + (-0.02) * color_E + (-0.03) * color_F + (-0.05) * color_G + (-0.09) * color_H
+ (-0.12) * color_I + (-0.12) * color_J + (1.23) * carat + (0.0) * depth + (-0.01) * ta
ble + (-0.38) * x + (0.35) * y + (-0.13) * z +
```

Inference

The best attributes of linear equation that is produced by the model for the target variable "Price".

From the above results we can say that the five most important attributes based on order are:

1. Carat
2. Y

3. Clarity

4. Z

Inference

- Based on the best attributes we can decide the price of the gemstone
- If the carat of the gemstone increases price also increases. same goes with y (width) and clarity of the gemstone, if we increase them price of the gemstone automatically increases
- x (length) of the gemstone increases price of the gemstone will decrease

Prediction of the Prices:

Prediction without scaling

	Predicted	Actual
0	8363.4849	8758
1	5257.5979	4718
2	13212.767	11965
3	11949.745	11965
4	9060.6521	8165
...
8073	4415.0042	4642
8074	4347.8779	4038
8075	251.26996	613
8076	527.79209	844
8077	5065.836	5198

Prediction with scaling

	Predicted	Actual
0	1.335427	1.44924
1	0.439411	0.283743
2	2.734394	2.374426
3	2.370025	2.374426
4	1.536552	1.278166
...
8073	0.196332	0.261818
8074	0.176966	0.08757
8075	-1.14984	-0.900507
8076	-0.925088	-0.833866
8077	0.38409	0.422218

Inference

The above Predicted results shows somewhat similar results to that of the actual prices which says that the model built is a good one.

Conclusions

Cubic Zirconia is a colourless, synthetic gemstone made of the cubic crystalline form of zirconium oxide. Often called as an inexpensive diamond or look like diamond. It is significantly cheaper than diamonds.

Like Diamonds, "4C" (Cut, Color, Clarity and Carat) play vital role in deciding the price of the gemstone. After Analysing and predicting using linear Regression in Cubic_Zirconia dataset, it seems Carat is a particularly important attribute to decide the price of the gemstone.

If the Carat increases, the price of the gemstone increases. Other attributes are Clarity, Y: Width, Z: Height. The X: Length of the gemstone increases, Price of the gemstone decreases

Recommendation

I recommend the company to concentrate mainly on carat while deciding the price of the Cubic Zirconia gemstone. So, Carat helps us to differentiate the higher and lower profitable stones. This will improve the profit of the company.

**CASE STUDY 2:
PREDICTION USING LOGISTIC
REGRESSION AND LINEAR
DISCRIMANT ANALYSIS**

Table of contents

 **Project objective**

 **Assumptions**

 **Exploratory data analysis**

- **Summary of the dataset**

- **Bivariate analysis**

 **Converting object data type into categorical**

 **Splitting the data into train and test data**

- **Dimensions on the train and test data**

 **Model building: Logistic Regression and Linear Discriminant analysis**

 **Linear regression Model Prediction**

 **Model evaluation**

 **Conclusion**

 **Recommendation**

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.

Dataset for Problem 2: Holiday_Package.csv

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

Project Objective:

The Objective of the report is to explore the dataset “**Holiday_Package.csv**” in Python (JUPYTER NOTEBOOK) and generate insights about the dataset. This exploration report will consist of the following:

-  Importing the dataset in jupyter notebook.
-  Understanding the structure of dataset.
-  Exploratory Data analysis
-  Graphical exploration
-  Prediction using Logistic Regression and Linear Discriminant Analysis (LDA)
-  Insights from the dataset

Assumptions:

Logistic Regression is used when the dependent variable(target) is categorical.

Types of Logistic Regression

1. Binary Logistic Regression

The categorical response has only two 2 possible outcomes. Example: Spam or Not

2. Multinomial Logistic Regression

Three or more categories without ordering. Example: Predicting which food is preferred more (Veg, Non-Veg, Vegan)

3. Ordinal Logistic Regression

Three or more categories with ordering. Example: Movie rating from 1 to 5

Here, we are using Binary Logistic Regression where dependent variable is categorical variable with yes/no

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

	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

Information on dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 872 entries, 0 to 871
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Holliday_Package      872 non-null    object
1   Salary                872 non-null    int64
2   age                  872 non-null    int64
3   educ                 872 non-null    int64
4   no_young_children     872 non-null    int64
5   no_older_children     872 non-null    int64
6   foreign               872 non-null    object
dtypes: int64(5), object(2)
memory usage: 47.8+ KB
```

Inference

- The column “Unnamed : 0” is removed from the dataset before proceeding further as its insignificant for the analysis.
- There are 872 rows and 7 columns
- There are no null values
- There are 5 integer and 2 object variables

Summary of the dataset

	Holliday_Packag e	Salary	age	educ	no_young_childre n	no_older_childre n	foreign
count	872	872	872	872	872	872	872
unique	2	#N/A	#N/A	#N/A	#N/A	#N/A	2
top	no	#N/A	#N/A	#N/A	#N/A	#N/A	no
freq	471	#N/A	#N/A	#N/A	#N/A	#N/A	656
mean	#N/A	47729.17 2	39.95527 5	9.30733 9	0.311927	0.982798	#N/A
std	#N/A	23418.66 9	10.55167 5	3.03625 9	0.61287	1.086786	#N/A
min	#N/A	1322	20	1	0	0	#N/A
25%	#N/A	35324	32	8	0	0	#N/A
50%	#N/A	41903.5	39	9	0	1	#N/A
75%	#N/A	53469.5	48	12	0	2	#N/A
max	#N/A	236961	62	21	3	6	#N/A

Inference:

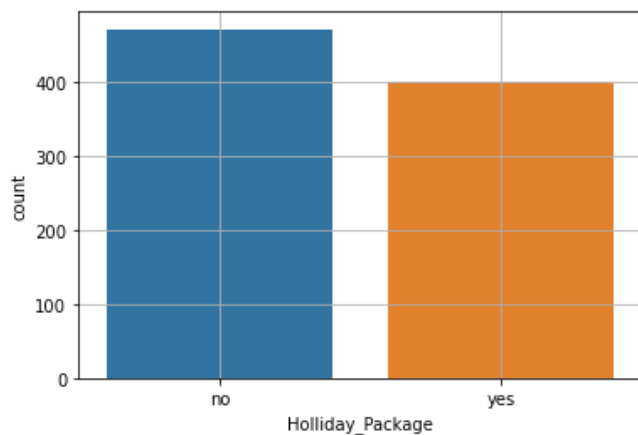
- Holliday_Package: There are 2 unique values
- Employees of a company who not opted(no) is 471
- Foreign: There are 2 unique values
- 656 employees are not foreigner
- highest salary paid:236961
- lowest paid salary:1322
- Minimum age of the empl0yee :20
- Maximum age of the employee: 62
- Average Years of formal education:9.3 years

Check for null values and Duplicates:

There are no null values and duplicates in this dataset

Unique counts of all Objects

Holliday_Package



Inference:

- Employees who opted for Holiday Package:401 (45.9%)
- Employees who not opted for Holiday Package:471 (54%)

Foreign

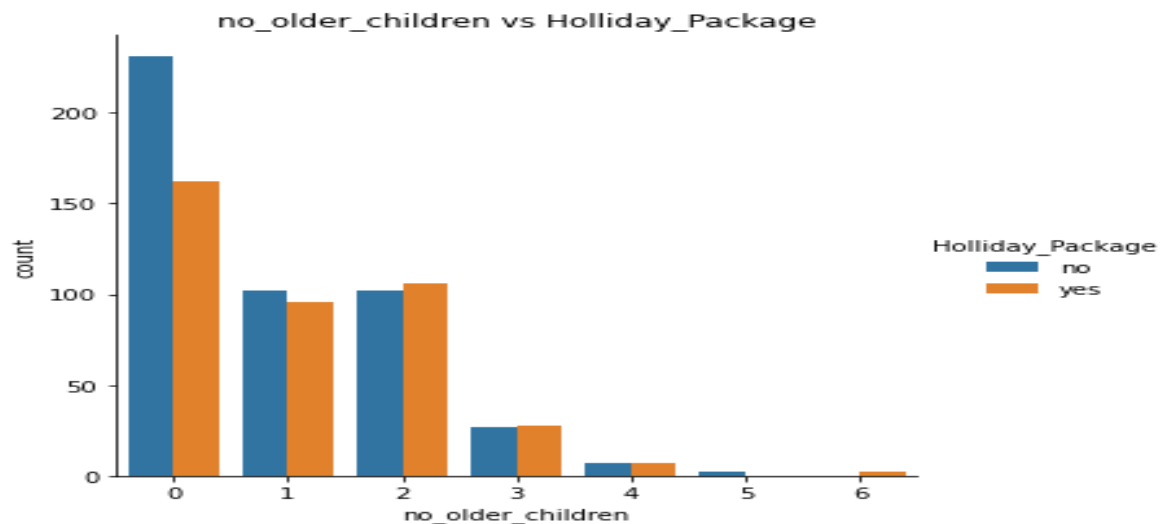
```
no      656
yes      216
Name: foreign, dtype: int64
```

Inference

- foreign employers:216
- non-foreign employers:656

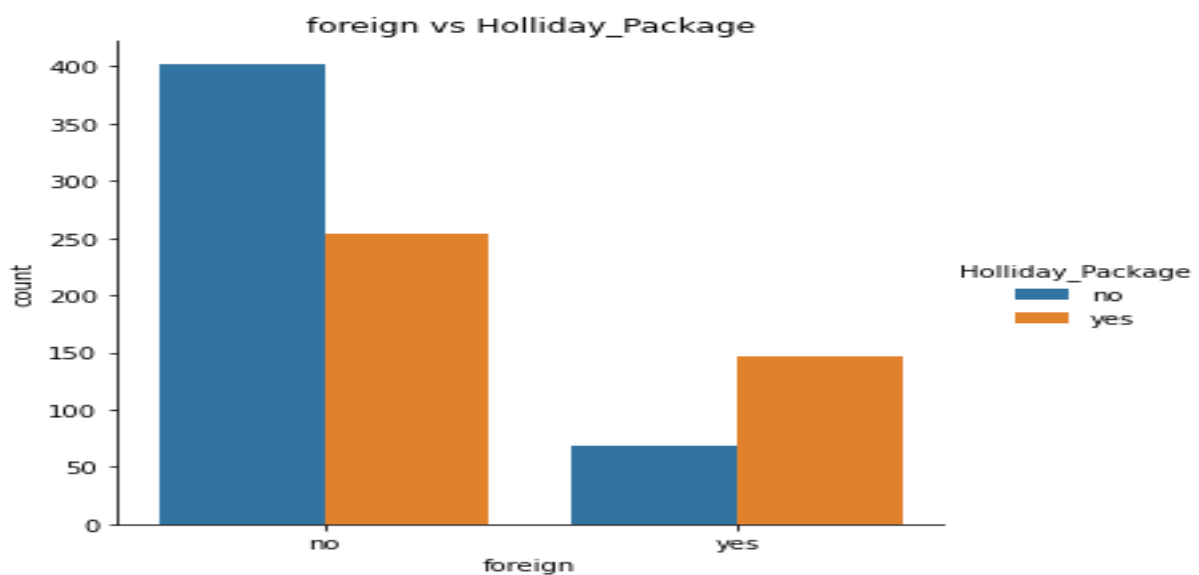
Bi-variate analysis

no_older_children vs Holliday_Package



Some Employees with no kids opted for Holiday_package but most of employees with no kids did not opt the Holiday package

foreign vs Holiday_Package



Most of foreign employees are not opted for holiday package

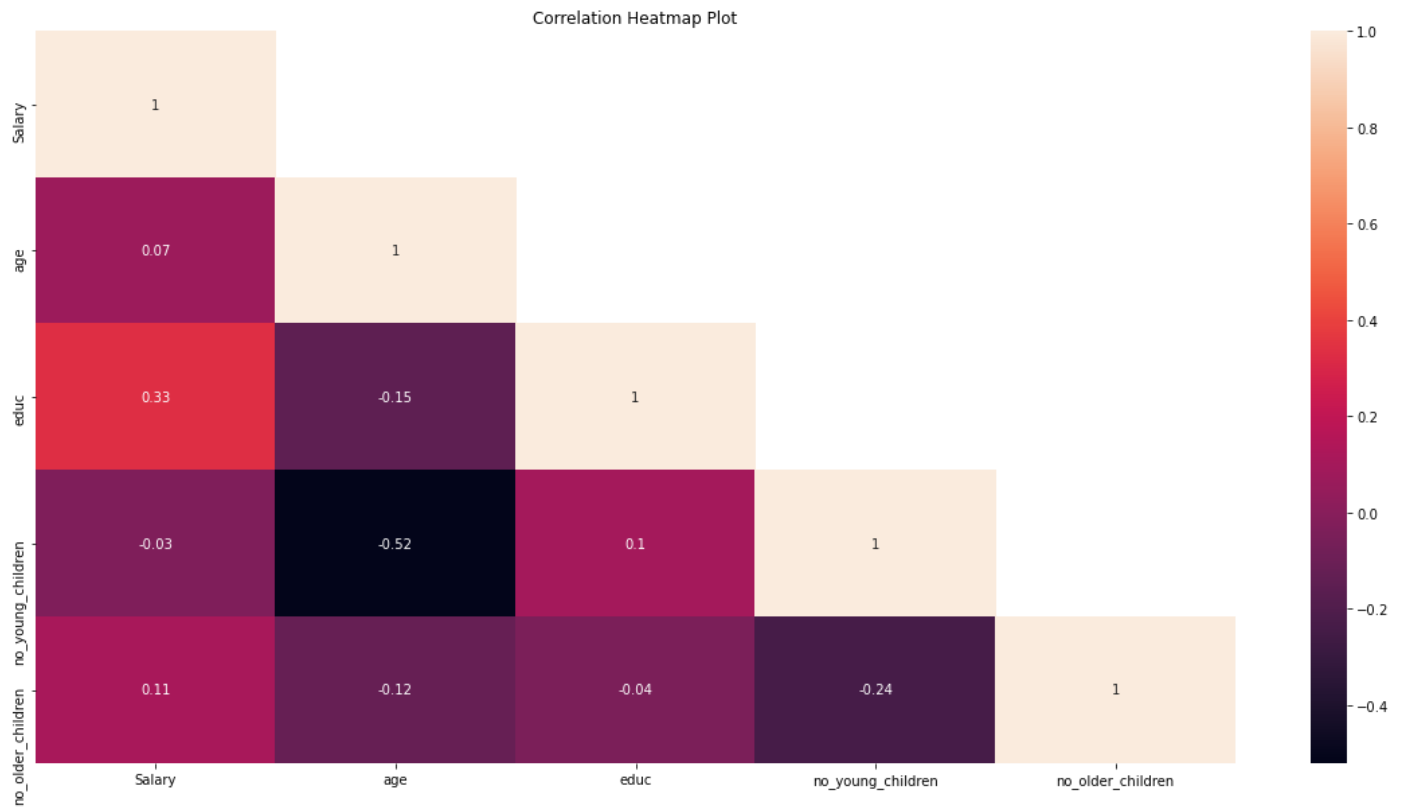
Pair plot between variables:



Inference

- Some of the attributes look like they may have an exponential distribution
- Salary should probably have a normal distribution, the constraints on the data collection may have skewed the distribution.
- age and salary are correlated with each other
- educ and salary are correlated with each other
- There is no obvious relationship between no younger children and holiday package

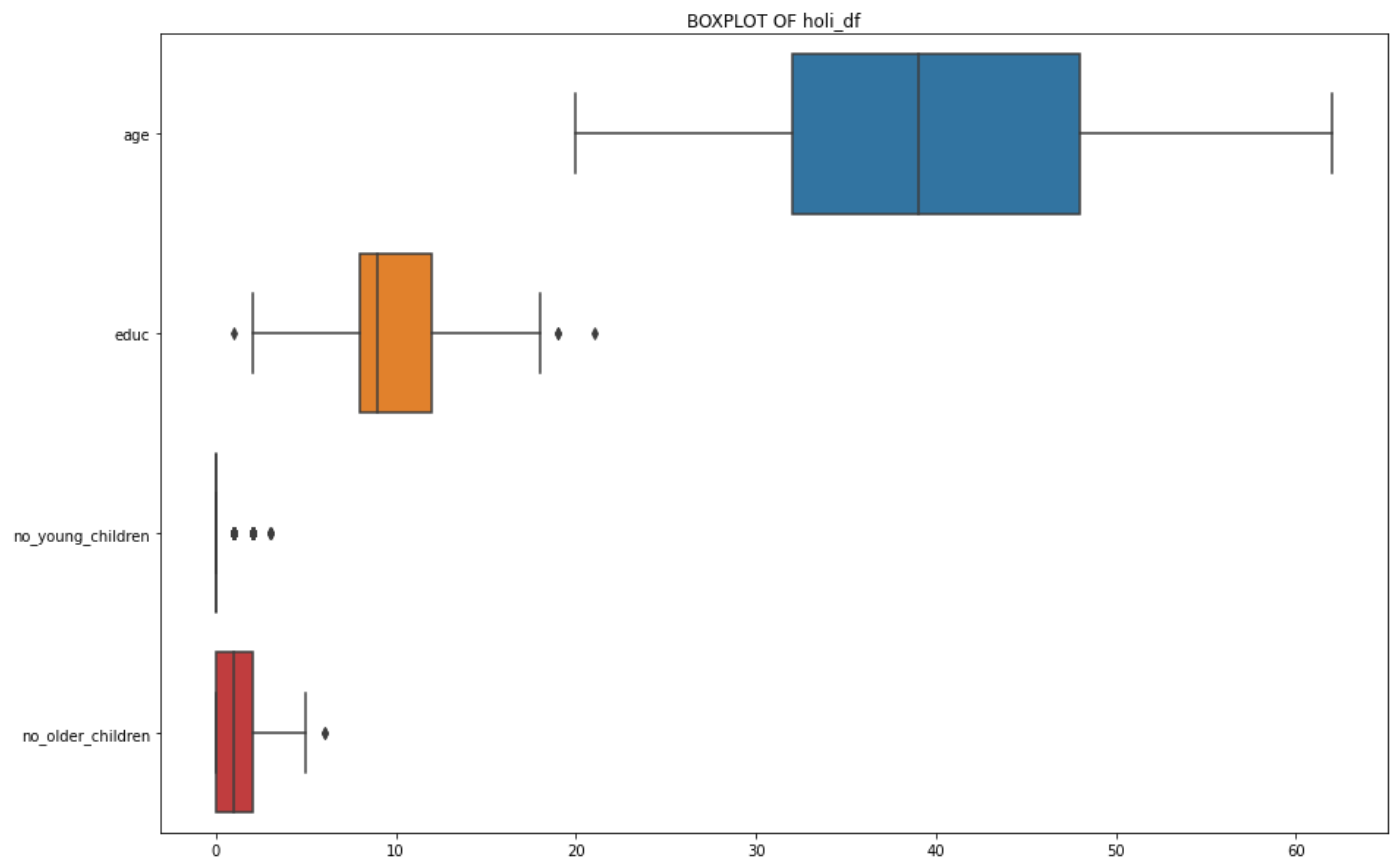
Correlation Heatmap



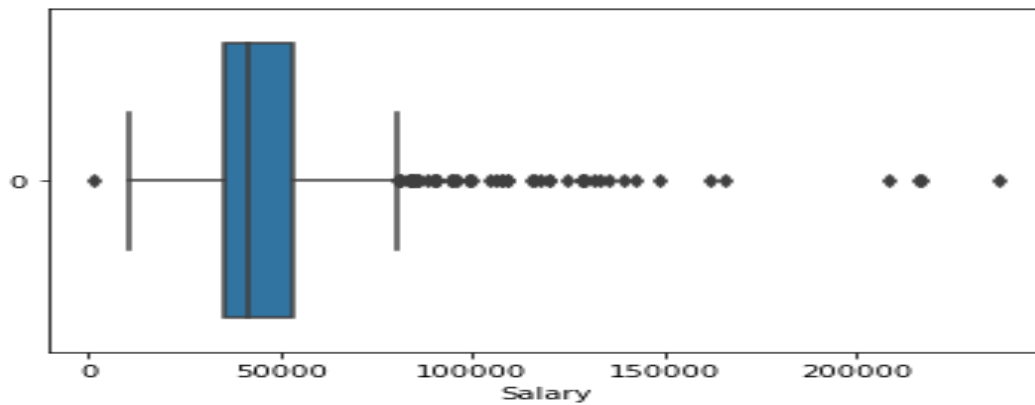
Inference

- Positive correlation between educ and salary
- Moderate negative correlation seen between age and no_young_children

Boxplot of holi_df



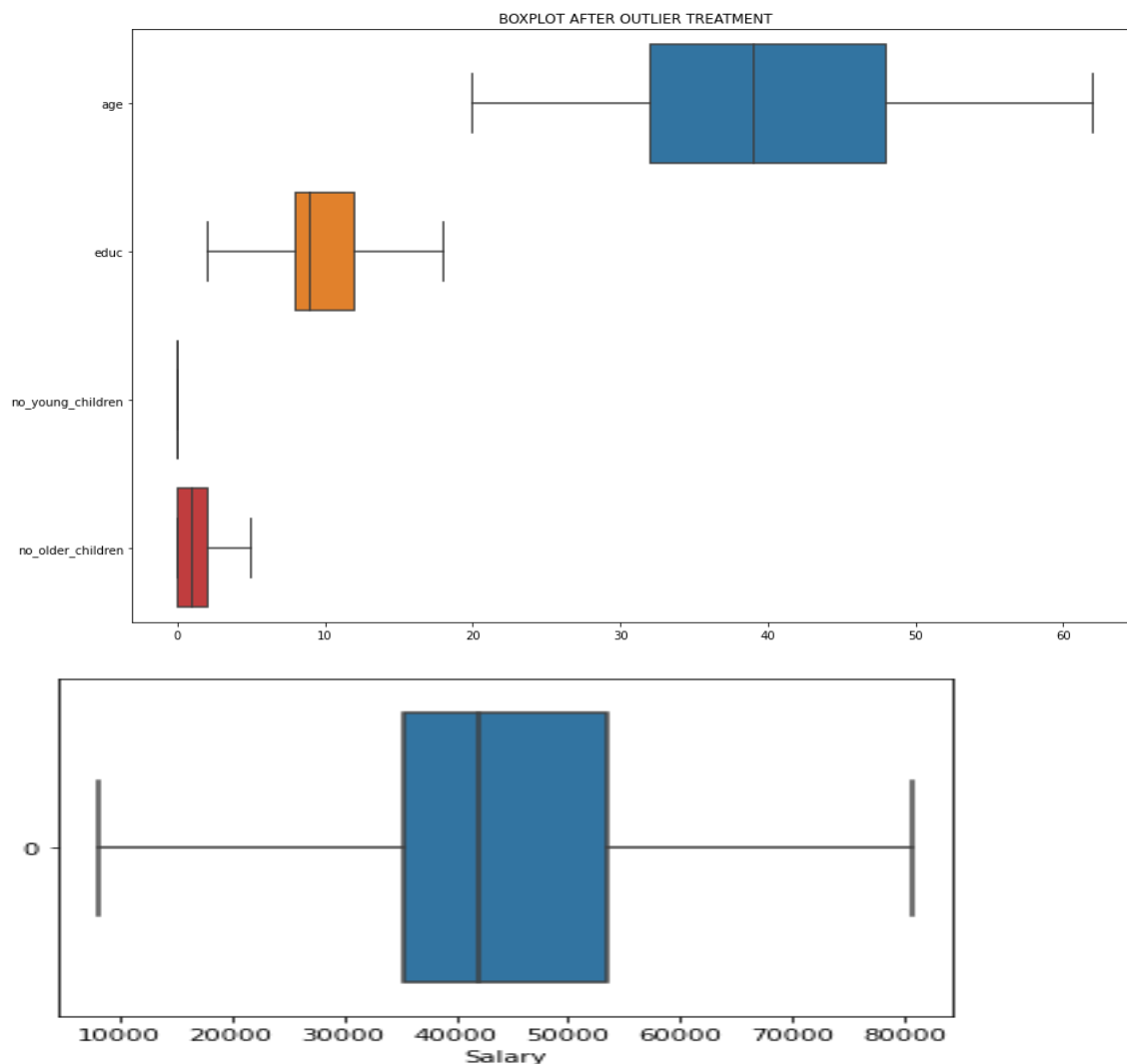
Boxplot of Salary



Inference

Except age, all other variables have Outliers

BOXPLOT AFTER OUTLIER TREATMENT



Inference

After outlier treatment, all outliers are removed

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).

Encode the data (having string values) for Modelling

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

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

Inference

- Codes are an array of integers which are the positions of the actual values in the categories array.
- Here Holliday_Package and foreign are categorical variables are now converted into integers using codes
- Now all the variables in the dataset are numeric variables

Train-Test Split

- Separating independent (train) and dependent (test) variables for the logistic regression model
- X = independent (train) variables
- Y = dependent (test) variables
- The training set for the independent variables: (610, 6)
- The training set for the dependent variable: (610, 1)
- The test set for the independent variables: (262, 6)
- The test set for the dependent variable: (262, 1)

Inference

- Splitting the dataset into train and test set to build Logistic regression and LDA model (70:30)
- X_train :70% of data randomly chosen from the 6 columns. These are training independent variables
- X_test :30% of data randomly chosen from the 6 columns. These are test independent variables
- y_train :70% of data randomly chosen from the "Holliday_Package" column. These are training dependent variables
- y_test :30% of data randomly chosen from the "Holliday_Package" columns. These are test independent variables

Logistic Regression Model

Logistic regression is a fundamental classification technique. It belongs to the group of linear classifiers and is somewhat like polynomial and linear regression.

It is the go-to method for binary classification problems (problems with two class values).

Two libraries are used

- 1.sklearn
- 2.statsmodel

Fit the Logistic Regression model

```
LogisticRegression()
```

Inference

- We now fit our model to the logistic regression model by training the model with our independent variable and dependent variables.
- At this point, you have the classification model defined.

Applying GridSearchCV for Logistic Regression

The probabilities on the training set

Column	0	1
0	0.703245	0.296755
1	0.292506	0.707494
2	0.736856	0.263144
3	0.674156	0.325844
4	0.506643	0.493357

The probabilities on the test set

Column	0	1
0	0.696807	0.303193
1	0.332213	0.667787
2	0.620128	0.379872
3	0.686886	0.313114
4	0.354964	0.645036

Accuracy - Training Data :64%

Accuracy – Test Data :63%

Inference

- Using GridsearchCV, we input various parameters like 'max_iter', 'penalty', 'solver', 'tol' which will helps us to find best grid for prediction of the better model
- max_iter is an integer (100 by default) that defines the maximum number of iterations by the solver during model fitting.
- solver is a string ('liblinear' by default) that decides what solver to use for fitting the model. Other options are 'newton-cg', 'lbfgs', 'sag', and 'saga'.
- penalty is a string ('l2' by default) that decides whether there is regularization and which approach to use. Other options are 'l1', 'elasticnet', and 'none'.
- bestgrid:{'max_iter': 10000, 'penalty': 'none', 'solver': 'newton-cg', 'tol': 0.0001}
- Accuracy score of training data:64%
- Accuracy score of training data:62.9%

Logistic Regression with Statsmodel

Statsmodels is a Python module which provides various functions for estimating different statistical models and performing statistical tests

first, we define the set of dependent(y) and independent(X) variables. If the dependent variable is in non-numeric form, it is first converted to numeric using encoding

.concatenate X(independent) and y (variables) into a single dataframe for logistic regression statsmodel

```
Optimization terminated successfully.
      Current function value: 0.641276
      Iterations 5
Intercept          -0.052700
Salary             -0.000019
age                -0.008975
educ               0.066395
no_older_children  0.186668
foreign            1.331779
dtype: float64
```

Inference

Statsmodels provides a Logit () function for performing logistic regression. The Logit () function accepts y and X as parameters and returns the Logit object. The model is then fitted to the data.

In the output, Iterations refer to the number of times the model iterates over the data, trying to optimise the model. the maximum number of iterations performed is 5, after which the optimisation fails.

Logit Regression Summary Table

Logit Regression Results						
=====						
Dep. Variable:	Holliday_Package		No. Observations:		610	
Model:	Logit		Df Residuals:		604	
Method:	MLE		Df Model:		5	
Date:	Mon, 08 Mar 2021		Pseudo R-squ.:		0.07166	
Time:	23:37:29		Log-Likelihood:		-391.18	
converged:	True		LL-Null:		-421.37	
Covariance Type:	nonrobust		LLR p-value:		1.010e-11	
=====						
	coef	std err	z	P> z	[0.025	0.975]

Intercept	-0.0527	0.578	-0.091	0.927	-1.185	1.080
Salary	-1.853e-05	6.22e-06	-2.982	0.003	-3.07e-05	-6.35e-06
age	-0.0090	0.009	-1.053	0.292	-0.026	0.008
educ	0.0664	0.035	1.904	0.057	-0.002	0.135
no_older_children	0.1867	0.081	2.308	0.021	0.028	0.345
foreign	1.3318	0.236	5.635	0.000	0.869	1.795

Inference

- The summary table above, gives us a descriptive summary about the regression results.
- foreign have coefficient of 1.3318 which shows foreign is important independent variable feature
- This means that for a one-unit increase in foreign we expect a 1.3318 increase in the log-odds of the dependent variable holiday_package, holding all other independent variables constant.
- Std. Err. – These are the standard errors associated with the coefficients. age have extremely low std.err
- foreign and salary having p-value <0.05. they are statistically significant.

Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a dimensionality reduction technique which is commonly used for the supervised classification problems.

It is used for modelling differences in groups i.e., separating two or more classes. It is used to project the features in higher dimension space into a lower dimension space. Library used in LDA is sklearn

Build LDA Model

We now fit our model to the LinearDiscriminantAnalysis Algorithm by training the model with our independent variable and dependent variables.

- At this point, the classification model defined.
- Training Data Class Prediction with a cut-off value of 0.5
- Test Data Class Prediction with a cut-off value of 0.5

Prediction for the training and test data

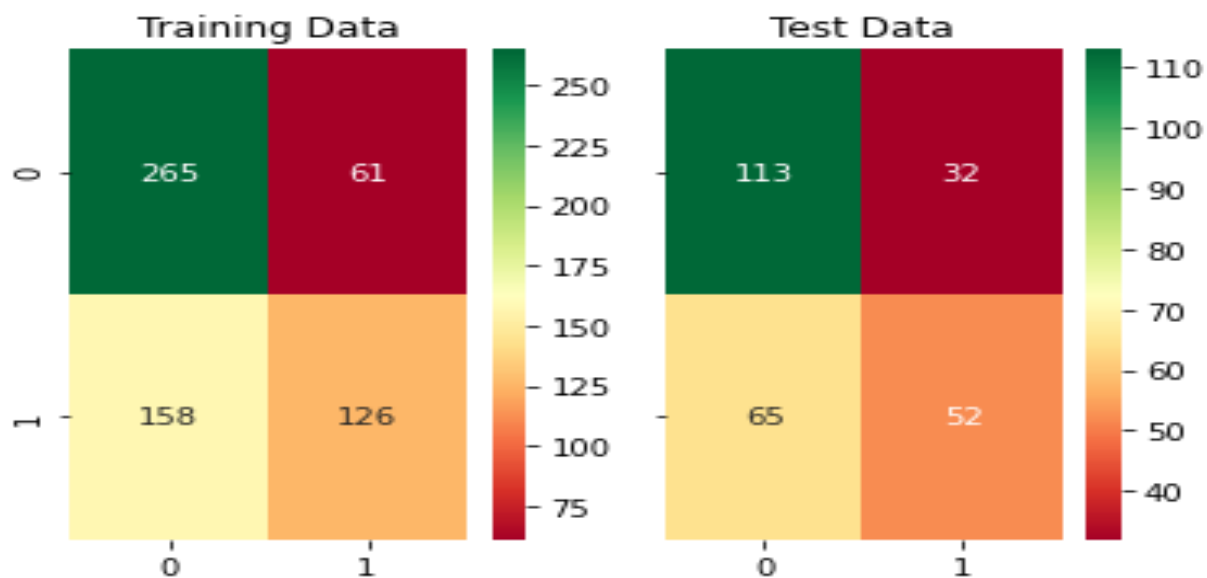
Training Accuracy:64%

Testing Accuracy:63%

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.

Logistic regression

Confusion matrix on the training and test data



Inference:

Training data:

True Negative: 265

False Positive : 61

False Negative: 158

True Positive : 126

Test data:

True Negative: 113

False Positive: 32

False Negative: 65

True Positive: 52

Classification Report of training and test data

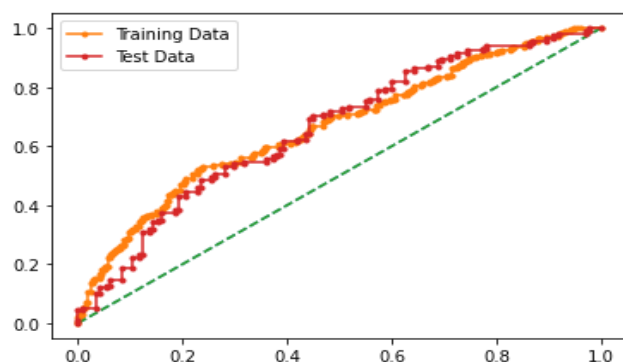
Training data

	precision	recall	f1-score	support
0	0.63	0.81	0.71	326
1	0.67	0.44	0.54	284
accuracy			0.64	610
macro avg	0.65	0.63	0.62	610
weighted avg	0.65	0.64	0.63	610

Test data

	precision	recall	f1-score	support
0	0.63	0.78	0.70	145
1	0.62	0.44	0.52	117
accuracy			0.63	262
macro avg	0.63	0.61	0.61	262
weighted avg	0.63	0.63	0.62	262

AUC and ROC for the training data



Inference

Logistic regression

Train Data:

- AUC: 66.7%
- Accuracy: 64%
- precision: 67%
- recall: 44%
- f1: 54%

Test Data:

- AUC: 66.1%
- Accuracy: 63%
- precision: 63%
- recall: 44%
- f1: 52%
- Training and Test set results are almost similar, this proves no overfitting or underfitting
- The Precision and Recall metrics also almost similar between training and test set.

The intercept for the model is : [-0.05269843]

Inference

The intercept is negative corresponds to that the estimated probability of the response is less than 50% when all model covariates equal zero.

The coefficients for each of the independent attributes

The coefficient for Salary is -1.8534315667025707e-05

The coefficient for age is -0.008975293705519546

The coefficient for educ is 0.06639467564981394

The coefficient for no_young_children is 0.0

The coefficient for no_older_children is 0.18666757986416826

The coefficient for foreign is 1.3317783895558775

Inference

- The coefficients for each of the independent attributes
- The sign of a regression coefficient tells you whether there is a positive or negative correlation between each independent variable the dependent variable.
- A positive coefficient indicates that as the value of the independent variable increases, the mean of the dependent variable also tends to increase.
- A negative coefficient suggests that as the independent variable increases, the dependent variable tends to decrease.
- The coefficient for foreign is 1.33 which shows "foreign" variable is high positive correlation with "holiday_package" (dependent variables)
- If more foreign employees opted for holiday package, no of employees opted for holiday_package will increase
- age, Salary, No_young_children have negative coefficient

Feature importance

Feature: 0, Score: -0.00002

Feature: 1, Score: -0.00898

Feature: 2, Score: 0.06639

Feature: 3, Score: 0.00000

Feature: 4, Score: 0.18667

Feature: 5, Score: 1.33178

Feature: 0 = Salary

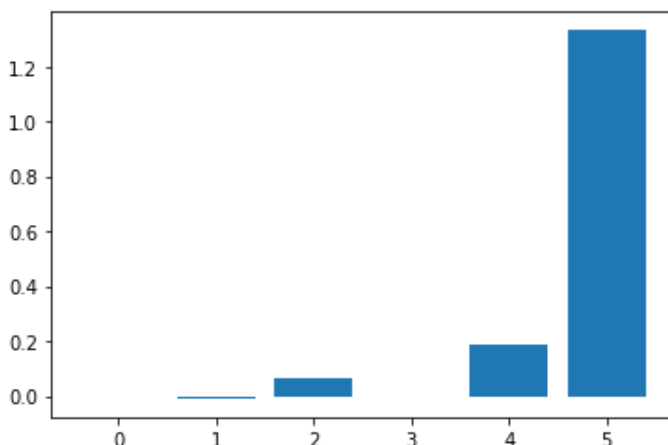
Feature: 1 = age

Feature: 2 = educ

Feature: 3 = no_young_children

Feature: 4 = no_older_children

Feature: 5 = foreign



Inference

- foreign (+ve) – 84 % increase per unit
- educ (+ve) – 4.2 % increase per unit
- age (-ve) – .569 % decrease per unit
- no_older_children (+ve) -11.84 % decrease per unit

Variance inflation factor

```
Salary ---> 10.691328770445477
age ---> 7.883717090792149
educ ---> 9.289879244969368
no_young_children ---> nan
no_older_children ---> 1.8287587912872858
foreign ---> 1.3123489821120458
```

Inference:

foreign and no older children are important variable

no_young_children have no value

LDA

Intercept

The intercept of the model is [-0.07340127]

Inference

- The negative intercept shows where the linear model predicts price (y) would be when subs (x) are 0.

The coefficients for each of the independent attributes

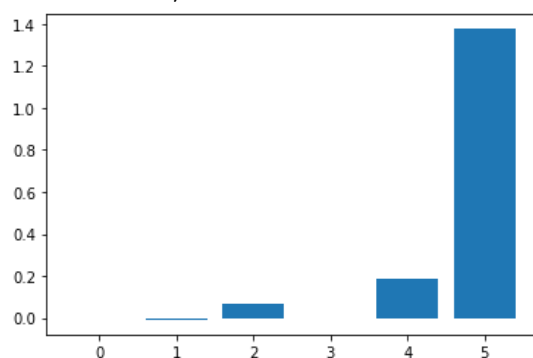
```
The coefficient for Salary is -1.8319623552908965e-05
The coefficient for age is -0.009003817204418316
The coefficient for educ is 0.0651380279296207
The coefficient for no_young_children is -1.3864459965067275e-16
The coefficient for no_older_children is 0.1874710395626884
The coefficient for foreign is 1.3765145540963208
```

Inference

- foreign and no older children are important variable
- The coefficient for foreign is 1.376 which shows "foreign" variable is high positive correlation with "holiday_package" (dependent variables)
- If more foreign employees opted for holiday package, no of employees opted for holiday_package will increase
- There is more negative correlation between "holiday package" and salary, age variables
- age, Salary, No_young_children have negative coefficient

Summarize feature importance

Feature: 0, Score: -0.00002	Feature: 0 = Salary
Feature: 1, Score: -0.00900	Feature: 1 = age
Feature: 2, Score: 0.06514	Feature: 2 = educ
Feature: 3, Score: 0.00000	Feature: 3 = no_young_children
Feature: 4, Score: 0.18747	Feature: 4 = no_older_children
Feature: 5, Score: 1.37651	Feature: 5 = foreign

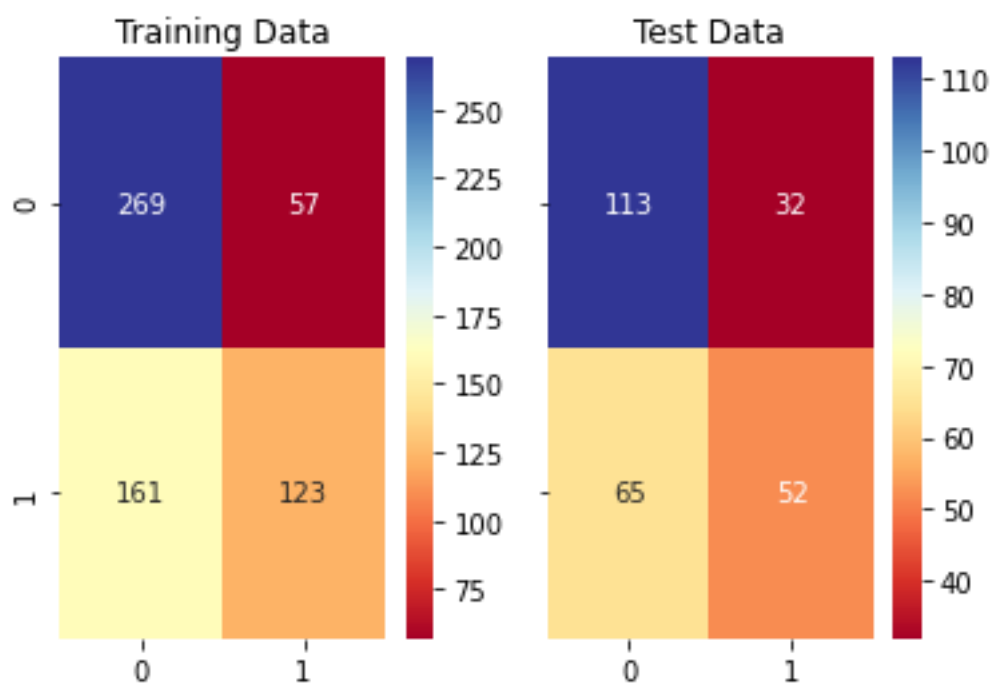


Inference

- foreign (+ve) – 84 % increase per unit
- educ (+ve) – 4.02 % increase per unit
- age (-ve) – .55 % decrease per unit
- no_older_children (+ve) -11.57 % decrease per unit

LDA

Confusion matrix on the training and test data



Inference

Training data:

True Negative: 269

False Positive: 57

False Negative: 161

True Positive: 123

Test data:

True Negative: 113

False Positive: 32

False Negative: 65

True Positive: 52

Classification Report of training and test data

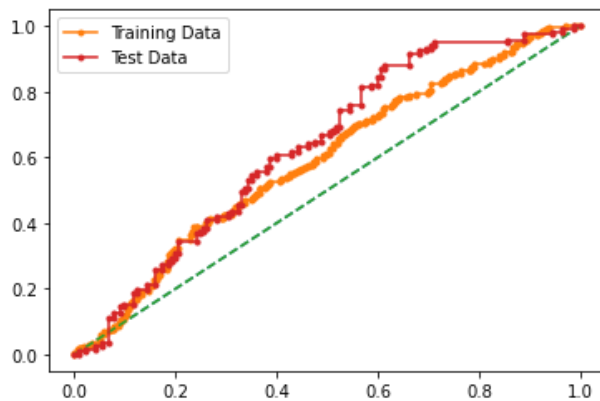
Training data

precision	recall	f1-score	support		
	0	0.63	0.83	0.71	326
	1	0.68	0.43	0.53	284
accuracy				0.64	610
macro avg		0.65	0.63	0.62	610
weighted avg		0.65	0.64	0.63	610

Test data

	precision	recall	f1-score	support
0	0.63	0.78	0.70	145
1	0.62	0.44	0.52	117
accuracy			0.63	262
macro avg	0.63	0.61	0.61	262
weighted avg	0.63	0.63	0.62	262

AUC and ROC for the training data



Inference

Train Data:

- AUC: 59.1%
- Accuracy: 64%
- Precision : 68%
- recall : 43%
- f1 :53%

Test Data:

- AUC: 63.3%
 - Accuracy: 63%
 - precision :63%
 - recall : 44%
 - f1 : 52%
-
- Training and Test set results are almost similar, This proves no overfitting or underfitting
 - The Precision and Recall metrics also almost similar between training and test set.

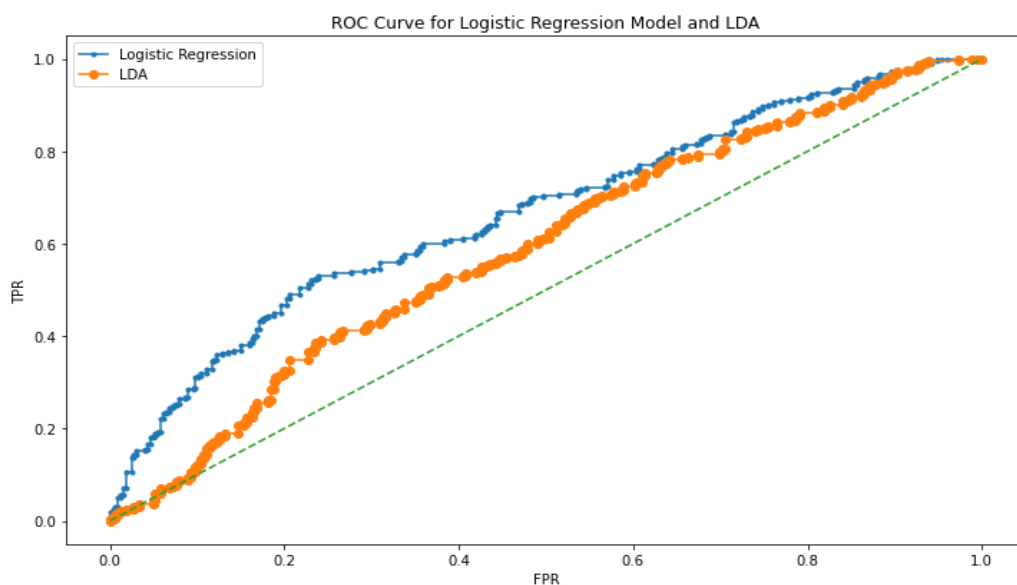
Final Model: Compare all the Logistic Regression and Linear Discriminant Analysis model

	LR Train	LR Test	LDA Train	LDA Test
Accuracy	0.641	0.63	0.643	0.63
AUC	0.667	0.661	0.591	0.633
Recall	0.44	0.44	0.43	0.44
Precision	0.67	0.62	0.68	0.62
F1 Score	0.54	0.52	0.53	0.52

ROC Curve for the 2 models on the Training data

Area under the curve for Logistic Regression Model is 0.666864685042772

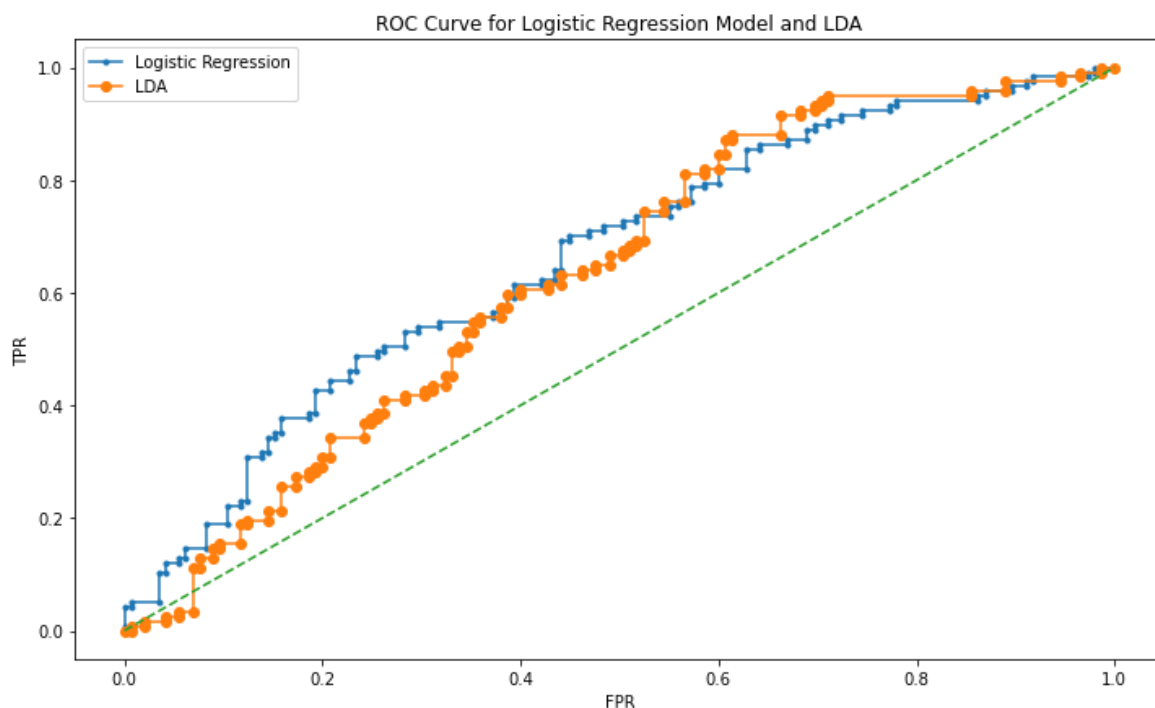
Area under the curve for LDA is 0.590976842650998



ROC Curve for the 2 models on the Test data

Area under the curve for Logistic Regression Model is 0.6610079575596816

Area under the curve for LDA is 0.6329501915708813



Inference

- Both Logistic Regression and LDA models performed at same level
- Train and test score is 64% and 63% respectively for both models
- AUC of Train and test in Logistic Regression is 66% and AUC of Train and test in LDA is 59% and 63% respectively
- f1 score of Train and test in Logistic Regression is 54% and 52% respectively
- Accuracy, Precision, recall for test data are almost in line with training data in both models. This indicates no overfitting or underfitting in the model
- We need to collect more data to build a good model.

2.4 Inference: Basis on these predictions, what are the insights and recommendations. Please explain and summarise the various steps performed in this project. There should be proper business interpretation and actionable insights present.

Conclusion:

- After Logistic regression and LDA Prediction in the given Holiday_Package dataset. We found that foreign is an important factor. Although Foreign employers are less, they opt for Holiday Package more.
- The coefficient for foreign is 1.33 in Logistic Regression and 1.37 in LDA which shows "foreign" variable have high positive correlation with "holiday_package" (dependent variables). A positive coefficient indicates that as the value of the independent variable increases, the mean of the dependent variable also tends to increase
- we can less importance to age because it has negative coefficient which shows if the company focus on employees age or salary, the mean of the dependent variable also tends to decrease. employees with increase in number of age prefer not to take the Holiday Package decreases. Small percentage the employees who are young prefer to opt for the package
- educ variable we can see an average importance given to that feature but in the positive trend. employees with increase in number of years of formal education opt for Holiday_Package
- no_older_children variable show least importance in a positive way. Employees with older children prefer to opt the Holiday_package
- Both Logistic Regression and LDA models performed at same level
- Train and test score is 64% and 63% respectively for both models
- Accuracy, f1, Precision, recall for test data are almost in line with training data in both models. This indicates no overfitting or underfitting in the model

Recommendations:

- The company should introduce new and interesting schemes like Holiday package based on job classification and length of service
- Through unique family friendly programs
- Rewards based package
- These things will attract more employees to opt the Holiday_Package
- Prediction Models needs to be trained with huge volumes of documents/transactions to cover all possible scenarios.

- In machine learning, Right data source and quality of data used to train predictive models is equally important as the quantity
- In our dataset, there are not enough to train the data, so it created class imbalance problem during the prediction. so company needs to collect more data to get the best results

Problem 2 Summary:

- **EDA:** All the basic EDA along with the univariate and the Bivariate analysis were performed and analysed including the descriptive statistics and null value check. Outliers were removed
- **Scaling and Encoding:** Scaling was not performed as per the requirement. The data (having string values) was encoded for Modelling.
- **Train and Test Splitting:** The data was split into train and test (70:30).
- **Model:** Logistic Regression (sklearn and statsmodel) and LDA (linear discriminant analysis) were applied.
- **Performance Metrics:** The performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model were checked properly. Both the models were compared, and inferences were written on which model is the best one.
- Inference on Basis on these predictions along with the business insights and recommendations were provided.

Reference

- <https://www.researchgate.net/publication/242579096> An Introduction to Logistic Regression Analysis and Reporting
- [https://datajobs.com/data-science-repo/Logistic-Regression-\[Peng-et-al\].pdf](https://datajobs.com/data-science-repo/Logistic-Regression-[Peng-et-al].pdf)
- <https://www.diamonds.pro/education/cubic-zirconia-vs-diamond/>
- <https://www.statsmodels.org/stable/examples/notebooks/generated/ols.html>
- <https://medium.com/devcareers/simple-linear-regression-and-multiple-linear-regression-analysis-with-statsmodel-library-in-python-a3292657ef87>
- <https://www.geeksforgeeks.org/ml-linear-discriminant-analysis/>