Great Learning

PREDICTIVE MODELING

ASHWIN KUMAR AG
PGDSBA – NOV'2020

TABLE OF CONTENTS

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.

PROBLEM 1: Linear Regression

APPROACH:

The Main purpose for carrying out the case study is to predict the prices of (Diamonds : Cubic Zirconia) based on several factors given in the dataset . Arrive at a business plan based on predictions to increase the profitability.

Based on the several factors which influence the quality of stone such length,width,cut etc; we aim to build a linear regression model inorder to predict the prices which is our main objective.

The Dataset provided is stored as "cubic_zirconia.csv" which contains data of 26967 cubic zirconia stones and 11 variables namely Carat,Cut,Colour,Clarity,Depth,Table,Price,X,Y,Z.

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.

IMPORTING THE REQUIRED PACKAGES:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
import matplotlib.pyplot as plt
import matplotlib.style
import scipy.stats as stats
from warnings import filterwarnings
filterwarnings('ignore')
```

EXPLORATORY DATA ANALYSIS:

The Dataset has 26,967 rows & 10 columns.

The Dataset has 10 variables namely : carat,cut,colour,clarity,depth,table,length,width,height and price.

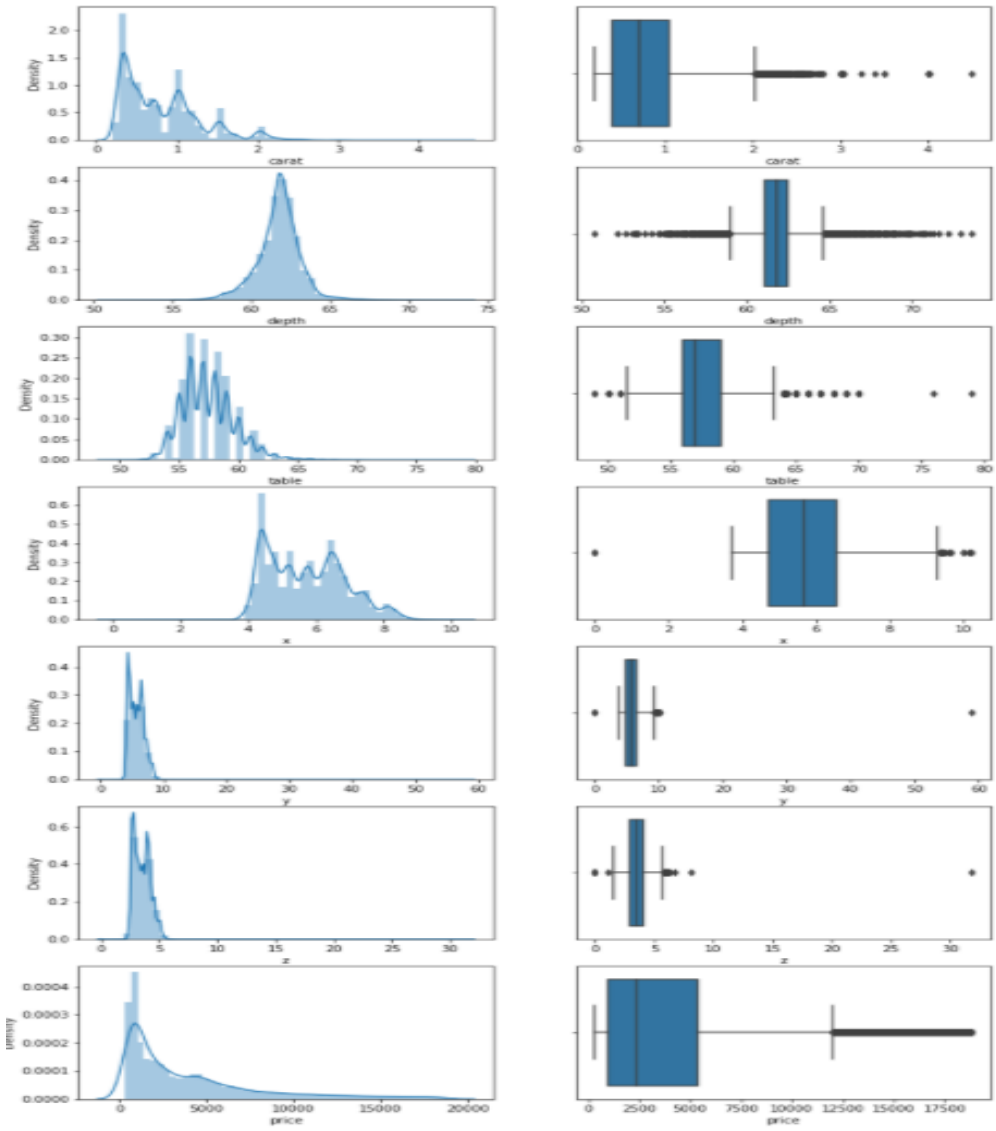
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26967 entries, 0 to 26966
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  26967 non-null  int64
 1   carat       26967 non-null  float64
 2   cut         26967 non-null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

SUMMARY OF THE DATA:

```
df.describe(include='all')
```

	Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
count	26967.000000	26967.000000	26967	26967	26967	26270.000000	26967.000000	26967.000000	26967.000000	26967.000000	26967.000000
unique	NaN	NaN	5	7	8	NaN	NaN	NaN	NaN	NaN	NaN
top	NaN	NaN	Ideal	G	SI1	NaN	NaN	NaN	NaN	NaN	NaN
freq	NaN	NaN	10816	5661	6571	NaN	NaN	NaN	NaN	NaN	NaN
mean	13484.000000	0.798375	NaN	NaN	NaN	61.745147	57.456080	5.729854	5.733569	3.538057	3939.518115
std	7784.846691	0.477745	NaN	NaN	NaN	1.412860	2.232068	1.128516	1.166058	0.720624	4024.864666
min	1.000000	0.200000	NaN	NaN	NaN	50.800000	49.000000	0.000000	0.000000	0.000000	326.000000
25%	6742.500000	0.400000	NaN	NaN	NaN	61.000000	56.000000	4.710000	4.710000	2.900000	945.000000
50%	13484.000000	0.700000	NaN	NaN	NaN	61.800000	57.000000	5.690000	5.710000	3.520000	2375.000000
75%	20225.500000	1.050000	NaN	NaN	NaN	62.500000	59.000000	6.550000	6.540000	4.040000	5360.000000
max	26967.000000	4.500000	NaN	NaN	NaN	73.600000	79.000000	10.230000	58.900000	31.800000	18818.000000

UNIVARIATE ANALYSIS:



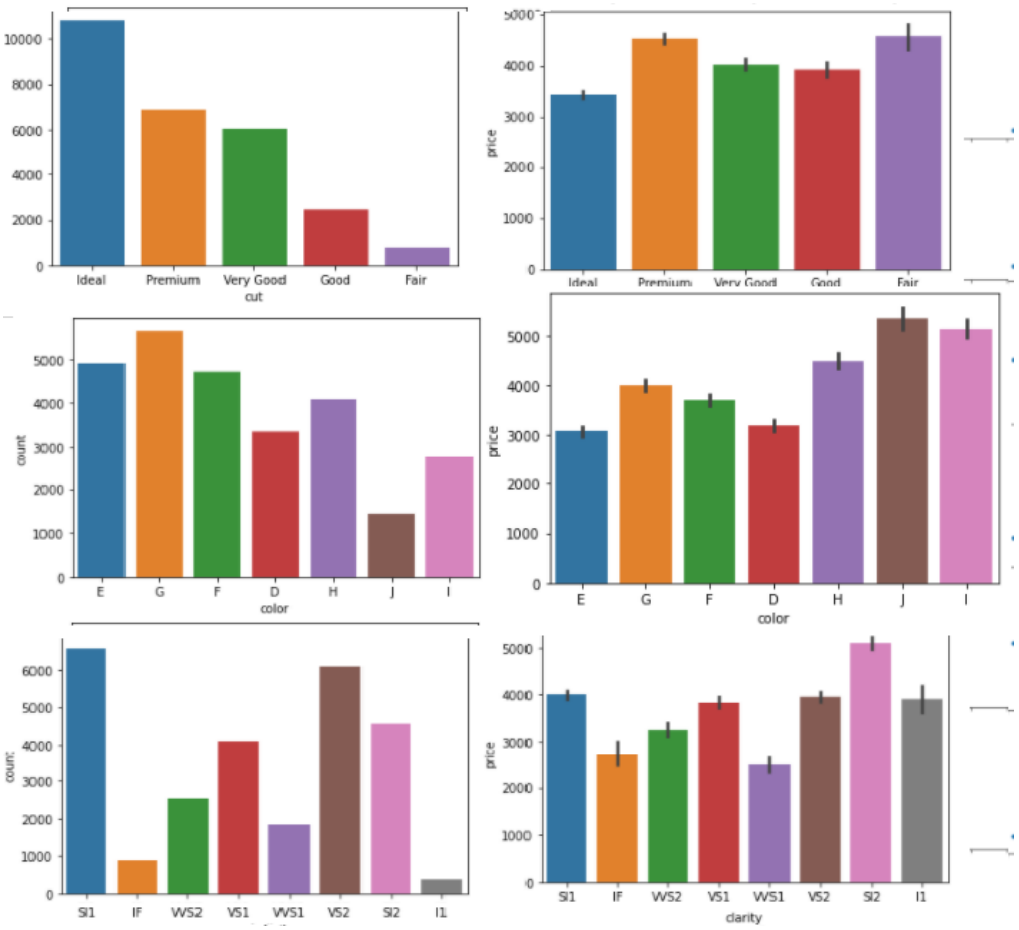
From the above plots,the observations are as follows:

Carat,Table,X,Y,Z Variables are right skewed.

Price being the target variable displays a right skewed graph

The Data has extreme outliers which be handled in further steps for analysis.

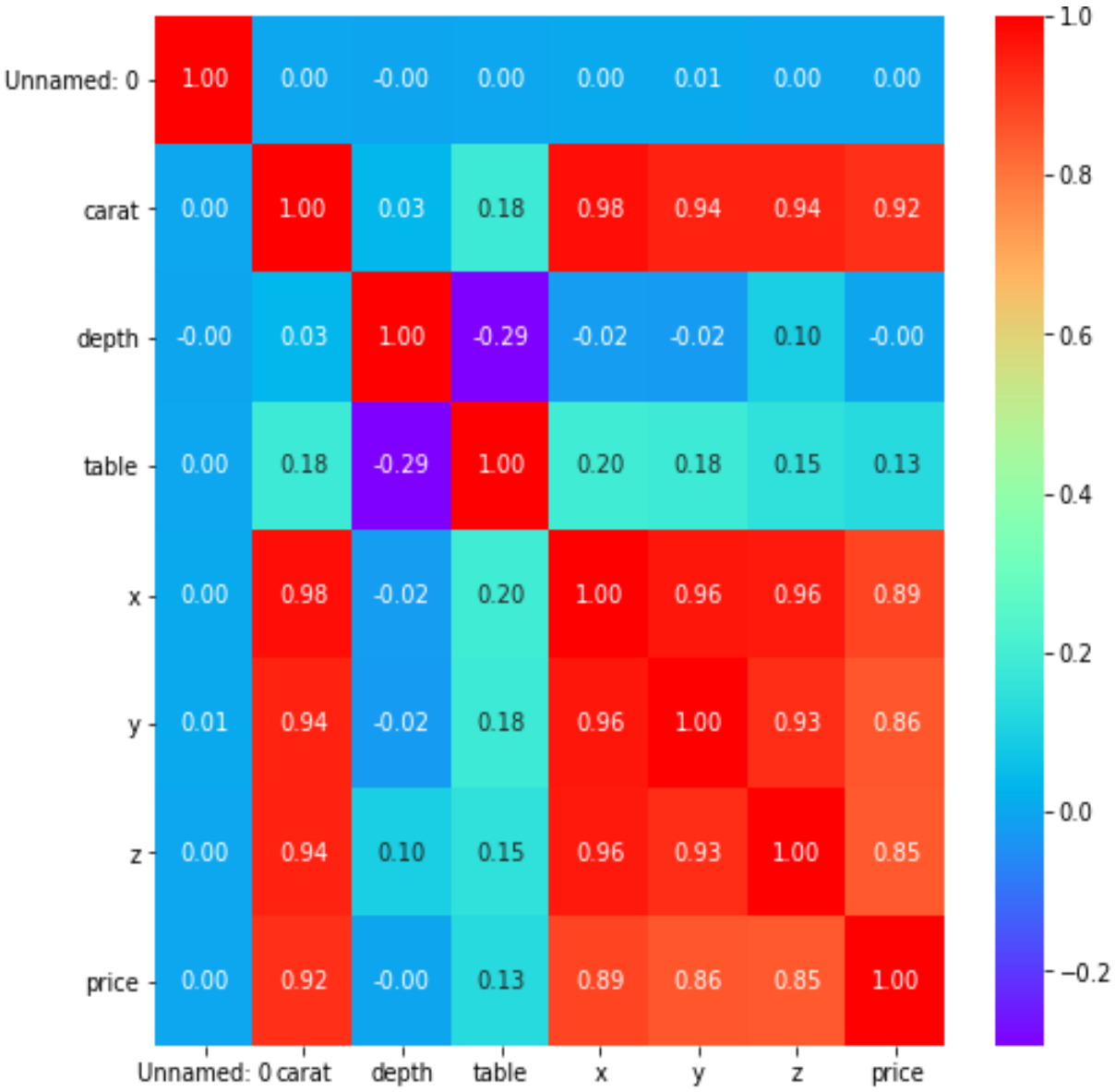
CATEGORICAL VARIABLES ANALYSIS:

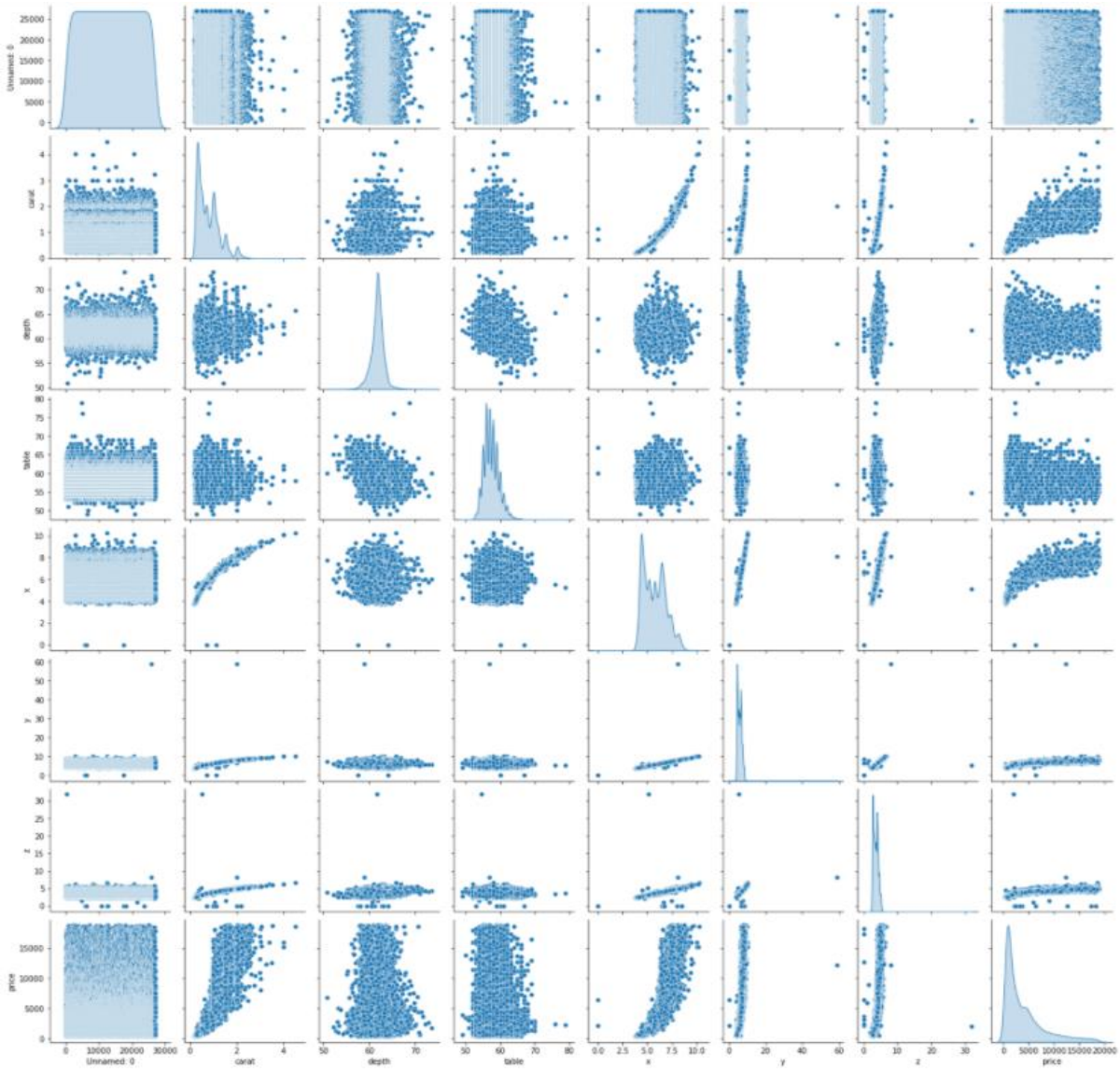


- *As per the above graph,Ideal Cut,SI1 clarity & colour G is found in abundance in the dataset.*
- *The Categorical Variables do not have influence on the price as there are higher prices in all the categories*

MULTIVARIATE ANALYSIS:

CORRELATION





Variable Carat has high correlation with price and also the variables x (length),y(width),z (height).

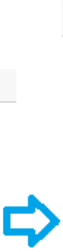
X(Length),Y(Width)&Z(Height) has high correlation between themselves & also with Price Variable.

Depth Variable does not have correlation with the other variables.Thus,with the pairplot we could see the overall trends across the variables

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?

First step,is to convert Categorical Variables into Dummy Variables. Few variables such as colour,cut & clarity are categorical variables which is converted into dummy variables for our further analysis.

```
df.isnull().sum()

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



```
df.isnull().sum()

Unnamed: 0      0
carat           0
depth           0
table           0
x               0
y               0
z               0
price           0
cut_Good        0
cut_Ideal       0
cut_Premium     0
cut_Very Good   0
color_E         0
color_F         0
color_G         0
color_H         0
color_I         0
color_J         0
clarity_IF      0
clarity_SI1     0
clarity_SI2     0
clarity_VS1     0
clarity_VS2     0
clarity_VVS1    0
clarity_VVS2    0
dtype: int64
```

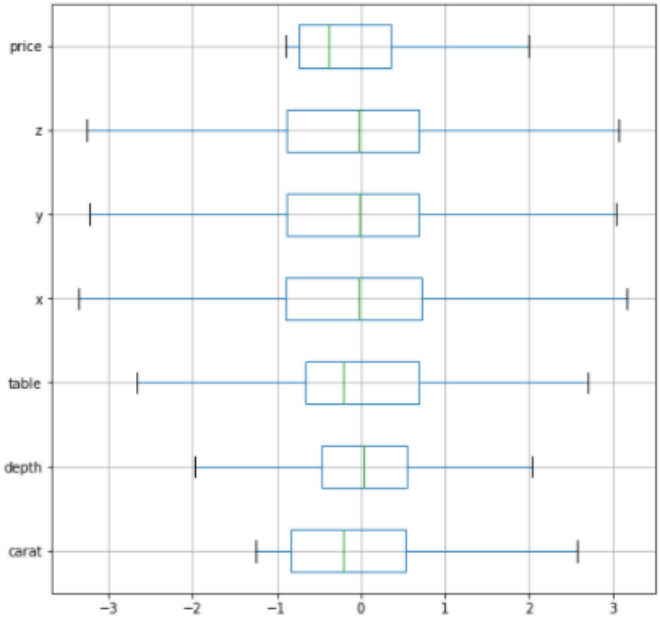
It was observed that there are 697 missing values in variable depth.Since,depth being a continuous variable mean imputation is done.Thus,instead of dropping the missing values ,we impute these values with its mean value for carrying out further analysis.

SCALING:

Scaling is a very useful tool to reduce the multi-collinearity of the data.It is an essential pre-processing step to convert variables with different range of measurements into a single scale.

We also have categorical variables ,which we will encode in the further steps.

OUTLIER TREATMENT:



We have remove & eliminated the outliers in the dataset for further analysis.

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

***LINEAR REGRESSION** is a linear model that assumes a linear relationship between the dependent or output variable(y) and one or more independent or input variables(X).*

ENCODING THE DATA HAVING STRING VALUES:

We have three Categorical values in the dataset : Cut,Color & Clarity.Since,Linear Regression Model does not take categorical values ; we have encoded into integer values for building the model.

We have dropped unwanted column named Unnamed: 0 which is not necessary for the analysis.

	carat	depth	table	x	y	z	price	cut_Good	cut_Ideal	cut_Premium	...	color_H	color_I	color_J	clarity_IF	cla
0	-1.043201	0.254475	0.243689	-1.293628	-1.238014	-1.218491	-0.854832	0	1	0	...	0	0	0	0	
1	-0.980405	-0.677789	0.243689	-1.160708	-1.092221	-1.162983	-0.734329	0	0	1	...	0	0	0	1	
2	0.212721	0.326187	1.139736	0.274832	0.331406	0.335747	0.583753	0	0	0	...	0	0	0	0	
3	-0.792017	-0.104088	-0.652358	-0.806254	-0.800635	-0.802177	-0.709979	0	1	0	...	0	0	0	0	
4	-1.022269	-0.964639	0.691712	-1.222737	-1.117949	-1.232368	-0.785263	0	1	0	...	0	0	0	0	

5 rows × 24 columns

The data was further split into 70% Training & 30% Testing data & Linear Regression Model was applied.

```
X = df1.drop('price',axis=1)
y=df1[['price']]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30 , random_state=1)
```

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

```
LinearRegression()
```

We will further fit our model into the Train dataset for further analysis.

*The **Co-efficients** for all the variables are derived:*

```
The coefficient for carat is 1.086780148215858
The coefficient for depth is -0.005875950399465979
The coefficient for table is -0.012555714989848907
The coefficient for x is -0.3687294527403408
The coefficient for y is 0.2637778479690415
The coefficient for z is -0.019439444968085223
The coefficient for cut_Good is 0.09402956253566346
The coefficient for cut_Ideal is 0.15620925180550174
The coefficient for cut_Premium is 0.1505867277544861
The coefficient for cut_Very Good is 0.12532462386457213
The coefficient for color_E is -0.048421354641000246
The coefficient for color_F is -0.06721648039435818
The coefficient for color_G is -0.10483306249085562
The coefficient for color_H is -0.2097241317021764
The coefficient for color_I is -0.329127039110375
The coefficient for color_J is -0.4759660793355477
The coefficient for clarity_IF is 1.0079796651227293
The coefficient for clarity_SI1 is 0.648718817170941
The coefficient for clarity_SI2 is 0.44630258377667476
The coefficient for clarity_VS1 is 0.850392434737886
The coefficient for clarity_VS2 is 0.7829215834318555
The coefficient for clarity_VVS1 is 0.9587533800432638
The coefficient for clarity_VVS2 is 0.9505610675317564
```

The Intercept of the model is also derived:

```
intercept = regression_model.intercept_[0]

print("The intercept for our model is {}".format(intercept))
```

```
The intercept for our model is -0.76943051543636
```

PERFORMANCE METRICS:

To evaluate our Linear Regression Model, we will take two measures namely, R square and RMSE which we will compute for both train and test datasets.

R-Square & RMSE on Train Data

```
regression_model.score(X_train, y_train)
```

```
0.9408511725362039
```

```
predicted_train=regression_model.fit(X_train, y_train).predict(X_train)
np.sqrt(metrics.mean_squared_error(y_train,predicted_train))
```

```
0.2101860221069641
```

R-Square & RMSE on Test Data

```
regression_model.score(X_test, y_test)
```

```
0.940351720878591
```

```
predicted_test=regression_model.fit(X_train, y_train).predict(X_test)
np.sqrt(metrics.mean_squared_error(y_test,predicted_test))
```

```
0.20954324304877975
```

In both the train & test datasets of the model we could observe R-Square value being 94% & has worked well.

VIF :

```
carat ---> 31.78217228374219
depth ---> 2.832265730043202
table ---> 1.7722917489040528
x ---> 437.0773361411652
y ---> 428.2789368306612
z ---> 105.82753801456637
cut_Good ---> 3.589224353795378
cut_Ideal ---> 14.297271436530734
cut_Premium ---> 8.606284212865514
cut_Very Good ---> 7.818947700289891
color_E ---> 2.368845047250475
```

Now that we have a normalised training data set,we use Variance inflation factor to observe the multicollinearity in the set of multiple regression variables.There seems to be high collinearity for several features.By conducting stats model we can see which features shall be removed so that the VIF is reduced .

STATS MODEL:

```
data_train = pd.concat([X_train, y_train], axis=1)
data_test=pd.concat([X_test,y_test],axis=1)
data_train.head()
```

	carat	depth	table	x	y	z	cut_Good	cut_Ideal	cut_Premium	cut_Very Good	...	color_I	color_J	clarity_IF
11687	-0.812949	0.397900	-0.652358	-0.850560	-0.860667	-0.802177	0	1	0	0	...	1	0	0
9728	1.908216	0.756463	-0.204334	1.639481	1.557783	1.681828	0	1	0	0	...	0	1	0
1936	-0.980405	0.039337	2.035782	-1.178431	-1.100797	-1.107474	1	0	0	0	...	0	0	0
26220	-0.205920	0.756463	-0.204334	-0.106206	-0.063093	0.002696	0	0	0	1	...	0	0	0
18445	-0.205920	0.254475	-0.652358	-0.053038	-0.020213	-0.011181	0	1	0	0	...	0	0	0

5 rows × 24 columns

LM1 - Linear Regression model score applied on the dataset removing outliers, removing duplicates,scaling, imputing null values, encoding the categorical values the model has

the following scores:

```
print(lm1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.941
Model:                            OLS   Adj. R-squared:                  0.941
Method:                 Least Squares   F-statistic:                 1.304e+04
Date:                Mon, 17 May 2021   Prob (F-statistic):               0.00
Time:                        13:48:36   Log-Likelihood:                2658.2
No. Observations:               18876   AIC:                            -5268.
Df Residuals:                   18852   BIC:                            -5080.
Df Model:                          23
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.7694      0.016    -47.005      0.000      -0.802      -0.737
carat          1.0868      0.009    122.034      0.000       1.069       1.104
depth         -0.0059      0.003     -2.082      0.037      -0.011      -0.000
table         -0.0126      0.002     -5.871      0.000      -0.017      -0.008
x             -0.3687      0.037     -9.927      0.000      -0.442      -0.296
y              0.2638      0.039      6.780      0.000       0.188       0.340
z             -0.0194      0.015     -1.326      0.185      -0.048       0.009
cut_Good       0.0940      0.011      8.722      0.000       0.073       0.115
cut_Ideal      0.1562      0.010     14.906      0.000       0.136       0.177
cut_Premium    0.1506      0.010     14.962      0.000       0.131       0.170
cut_Very_Good  0.1253      0.010     12.142      0.000       0.105       0.146
color_E       -0.0484      0.006     -8.572      0.000      -0.059      -0.037
color_F       -0.0672      0.006    -11.802      0.000      -0.078      -0.056
color_G       -0.1048      0.006    -18.839      0.000      -0.116      -0.094
color_H       -0.2097      0.006    -35.303      0.000      -0.221      -0.198
color_I       -0.3291      0.007    -49.847      0.000      -0.342      -0.316
color_J       -0.4760      0.008    -58.068      0.000      -0.492      -0.460
clarity_IF     1.0080      0.016     61.701      0.000       0.976       1.040
clarity_SI1    0.6487      0.014     46.203      0.000       0.621       0.676
clarity_SI2    0.4463      0.014     31.595      0.000       0.419       0.474
clarity_VS1    0.8504      0.014     59.373      0.000       0.822       0.878
clarity_VS2    0.7829      0.014     55.466      0.000       0.755       0.811
clarity_VVS1   0.9588      0.015     63.144      0.000       0.929       0.989
clarity_VVS2   0.9506      0.015     64.350      0.000       0.922       0.980
==============================================================================
Omnibus:                     4696.973   Durbin-Watson:                   1.982
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            17423.441
Skew:                           1.212   Prob(JB):                         0.00
Kurtosis:                       7.034   Cond. No.                         67.6
==============================================================================

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

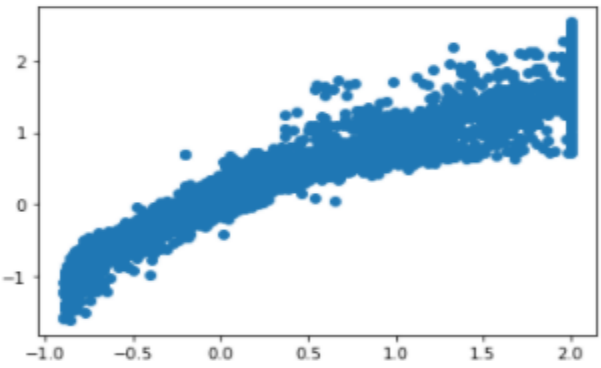
STATS MODEL SCORES & PLOT

```
mse
```

```
0.044178163889149344
```

```
np.sqrt(lm1.mse_resid)
```

```
0.21031977078235542
```



We need to perform feature selection to identify the most significant independent variables. We reduce the features by eliminating a single feature with iterations until the p-values are <= 0.05.We observe P values >0.05 in depth & z feature , so we can eliminate these & reiterate the model.

LM2 - We perform Linear Regression by dropping the Depth & z values where the P Value > 0.05 & the results are as follows:

OLS Regression Results

Dep. Variable:	price	R-squared:	0.941
Model:	OLS	Adj. R-squared:	0.941
Method:	Least Squares	F-statistic:	1.427e+04
Date:	Mon, 17 May 2021	Prob (F-statistic):	0.00
Time:	13:51:03	Log-Likelihood:	2649.6
No. Observations:	18876	AIC:	-5255.
Df Residuals:	18854	BIC:	-5083.
Df Model:	21		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-0.7815	0.016	-48.486	0.000	-0.813	-0.750
carat	1.0775	0.009	125.160	0.000	1.061	1.094
table	-0.0092	0.002	-4.679	0.000	-0.013	-0.005
x	-0.3867	0.037	-10.540	0.000	-0.459	-0.315
y	0.2724	0.038	7.262	0.000	0.199	0.346
cut_Good	0.0987	0.011	9.216	0.000	0.078	0.120
cut_Ideal	0.1687	0.010	16.811	0.000	0.149	0.188
cut_Premium	0.1621	0.010	16.760	0.000	0.143	0.181
cut_Very_Good	0.1340	0.010	13.251	0.000	0.114	0.154
color_E	-0.0485	0.006	-8.578	0.000	-0.060	-0.037
color_F	-0.0673	0.006	-11.808	0.000	-0.078	-0.056
color_G	-0.1054	0.006	-18.933	0.000	-0.116	-0.094
color_H	-0.2102	0.006	-35.378	0.000	-0.222	-0.199
color_I	-0.3297	0.007	-49.929	0.000	-0.343	-0.317
color_J	-0.4767	0.008	-58.144	0.000	-0.493	-0.461
clarity_IF	1.0120	0.016	62.026	0.000	0.980	1.044
clarity_SI1	0.6500	0.014	46.286	0.000	0.622	0.677
clarity_SI2	0.4480	0.014	31.715	0.000	0.420	0.476
clarity_VS1	0.8527	0.014	59.557	0.000	0.825	0.881
clarity_VS2	0.7848	0.014	55.604	0.000	0.757	0.812
clarity_VVS1	0.9618	0.015	63.391	0.000	0.932	0.992
clarity_VVS2	0.9532	0.015	64.567	0.000	0.924	0.982

Omnibus:	4664.566	Durbin-Watson:	1.982
Prob(Omnibus):	0.000	Jarque-Bera (JB):	17047.203
Skew:	1.209	Prob(JB):	0.00
Kurtosis:	6.979	Cond. No.	58.7

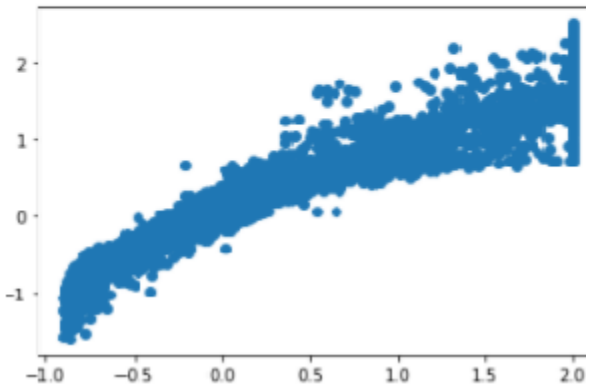
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
mse
```

```
0.04421859621953349
```

```
np.sqrt(lm2.mse_resid)
```

```
0.21040483164909268
```



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

The Linear Regression Equation for our model is as follows:

INFERENCE:

Comparing the Model Scores from the Linear Regression models , LM2 sees to be an optimum model.

From the Analysis,we can get a clear picture that on basis of the cut,Ideal Cut had a significant number of turnover to the company.The Colours H,I,J are also comparatively superior over the other colours & have provided profits.We could see that the dimesional features had strong positive correlation among themselves & with the price feature.

From our result,the computed R-Square is 94% for the model which means that 94% of the variance of the target variable price is explained by the predictors (independent variables) in the training data set.

Also,for better accuracies we have dropped the columns depth & z as the had p values more than 0.05 for better results.

The final Linear Regression equation is as follows:

```
for i,j in np.array(lm2.params.reset_index()):
    print('({}) * {} +'.format(round(j,2),i),end=' ')
```

```
(-0.78) * Intercept + (1.08) * carat + (-0.01) * table + (-0.39) * x + (0.27) * y + (0.1) * cut_Good + (0.17) * cut_Ideal + (0.
16) * cut_Premium + (0.13) * cut_Very_Good + (-0.05) * color_E + (-0.07) * color_F + (-0.11) * color_G + (-0.21) * color_H + (-
0.33) * color_I + (-0.48) * color_J + (1.01) * clarity_IF + (0.65) * clarity_SI1 + (0.45) * clarity_SI2 + (0.85) * clarity_VS1
+ (0.78) * clarity_VS2 + (0.96) * clarity_VVS1 + (0.95) * clarity_VVS2 +
```

- ***Carat is the highest co-efficient in the predicting the price, change in 1 unit of the carat will have an impact on the price 1.08 times, henceforth it is the most important deciding factor for the price of the diamond.***
- ***The Top Attributes reasonable & influencing the price are : Carat, clarity_IF ,color J,clarity_SI1 ,clarity_SI2 ,clarity_VS1, clarity_VS2 ,clarity_VVS1 ,clarity_VVS2.***
- ***The company should focus on key strengths & develop marketing strategies to promote diamond's carat and clarity & also be competitive about their prices. More Carat & clear the stone the profits are more. The cuts of the diamond are also a significant factor in the prices***

Problem 2 : You are hired by a tour and travel agency which deals in selling holiday packages. You are provided details of 872 employees of a company. Among these employees, some opted for the package and some didn't. You have to help the company in predicting whether an employee will opt for the package or not on the basis of the information given in the data set. Also, find out the important factors on the basis of which the company will focus on particular employees to sell their packages.

PROBLEM 2: LOGISTIC REGRESSION AND LDA

APPROACH :

The Main aim of conducting this case study is to predict whether the employee will choose a holiday package or not based on the data given to us.The idea is to give the company a competetive knowledge & insights to take business decisions for selling the holiday packages effectively based on the predictions.

The Objective is to build Logistic Regression Model & LDA as per the data given to us; predict if the employee opts for a package by comparing both the models performance metrics accordingly.

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.

IMPORTING THE REQUIRED PACKAGES:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import sklearn
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import roc_auc_score,roc_curve,classification_report,confusion_matrix,plot_confusion_matrix
from sklearn import metrics
from sklearn import metrics,model_selection
from sklearn.preprocessing import scale
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
import warnings
warnings.filterwarnings("ignore")
```

IMPORTING THE DATASET:

```
df = pd.read_csv("Holiday_Package.csv")
df.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

The Dataset ha 7 Variables – Holiday_Package,Salary,Age,Education,No_young_children,No_older_children & Foreign.

The Dataset has 872 entries with 7 Columns.5 Integer & 2 Object Type

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

There were no duplicated & null values found in the dataset.

df.duplicated().sum()	df.isnull().sum().sum()
0	0

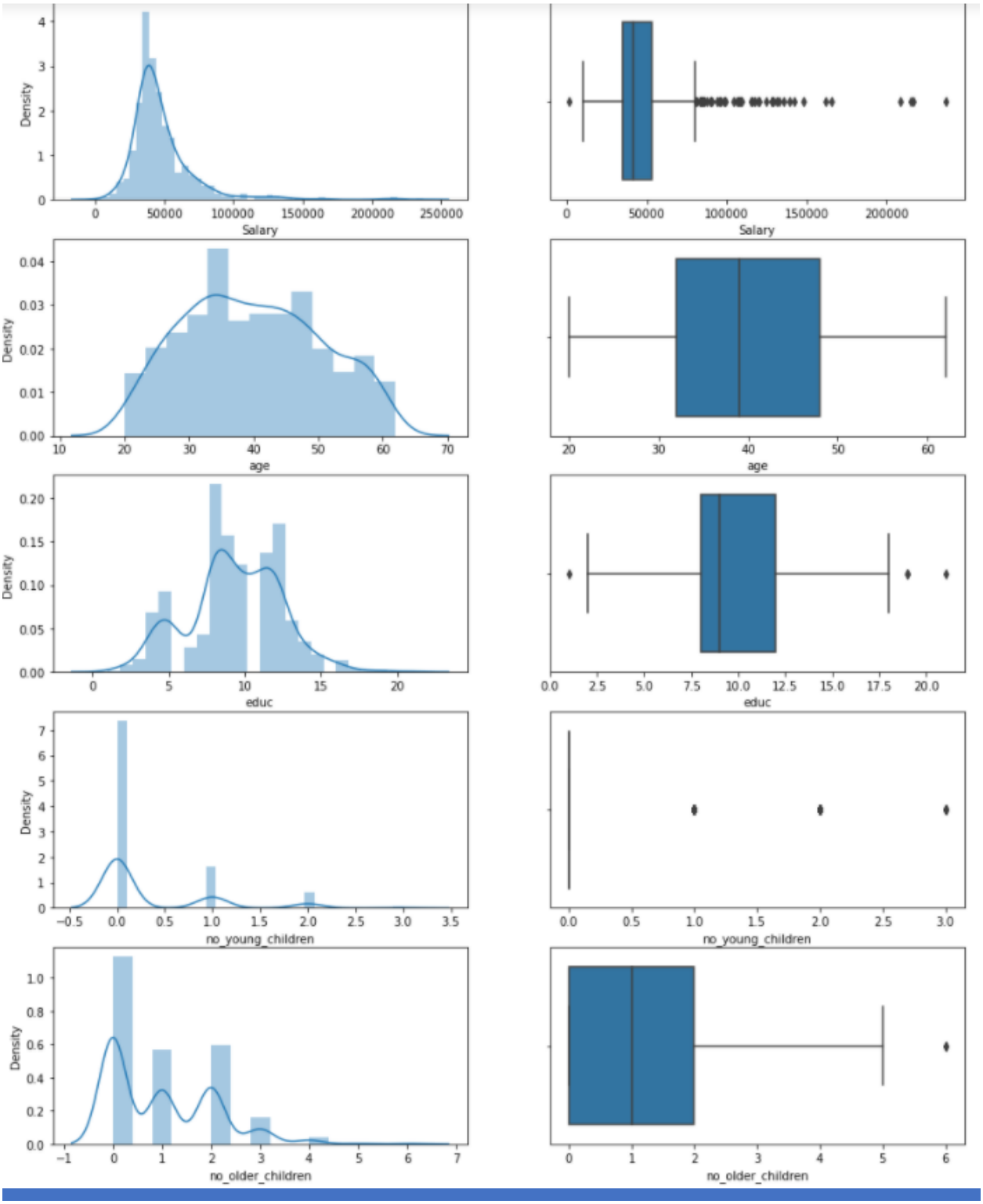
DATA SUMMARY:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
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

- ***Holliday_Package – is our Target Variable.***
- ***Salary, age, educ variables are numerical or continuous variables.***
- ***Holliday_Package and foreign are binary variables which we will be converted into categorical codes for model building.***
- ***We will drop the first column 'Unnamed: 0' column as this is not important for our study.***

EXPLORATORY ANALYSIS:

UNIVARIATE ANALYSIS:



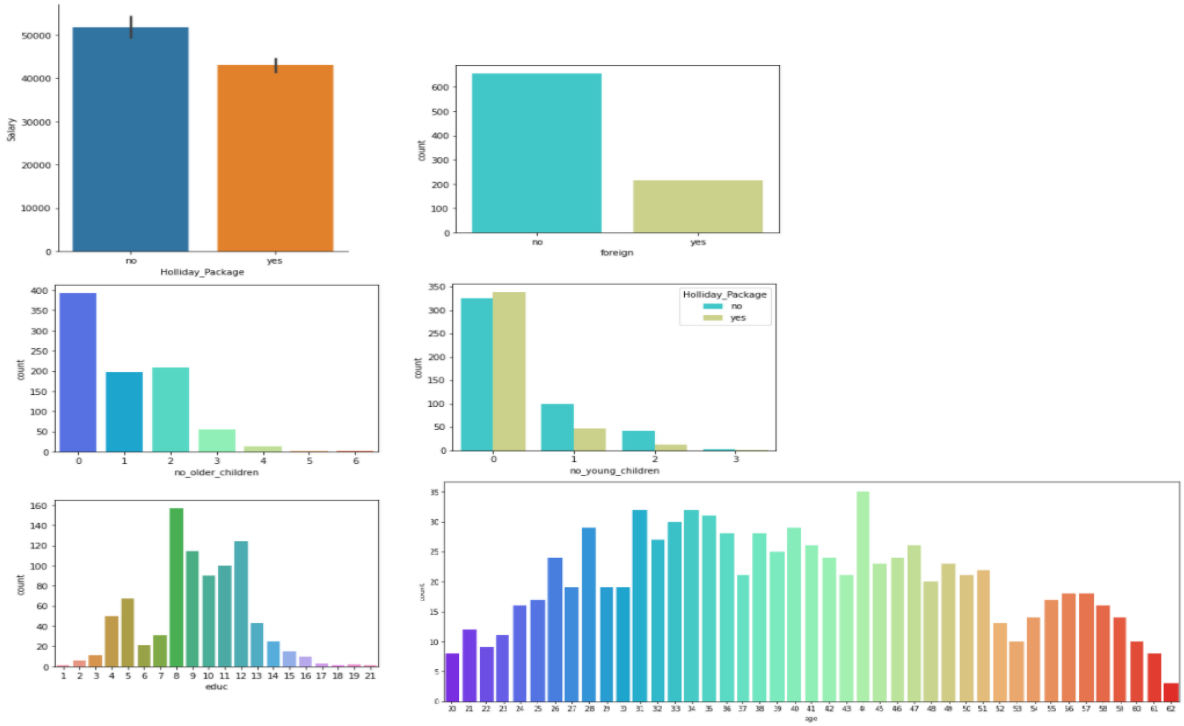
The above graph indicates the variables have outliers except the age variable.

```
df.skew()
```

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

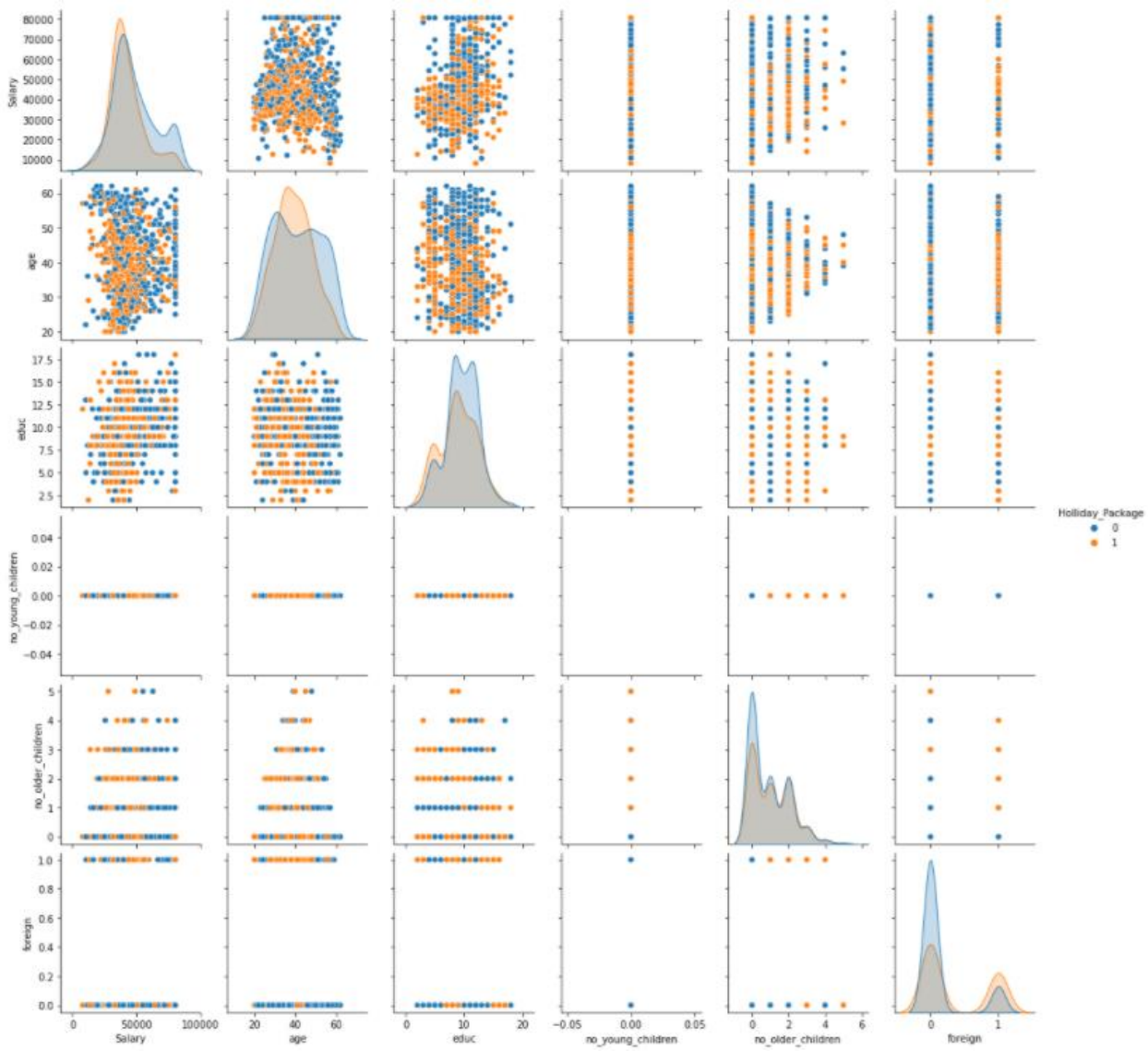
Age & Education is distributed quite uniformly;Salary is highly skewed.Salary Variable has a lot of extreme outliers present.

CATEGORICAL UNIVARIATE ANALYSIS:



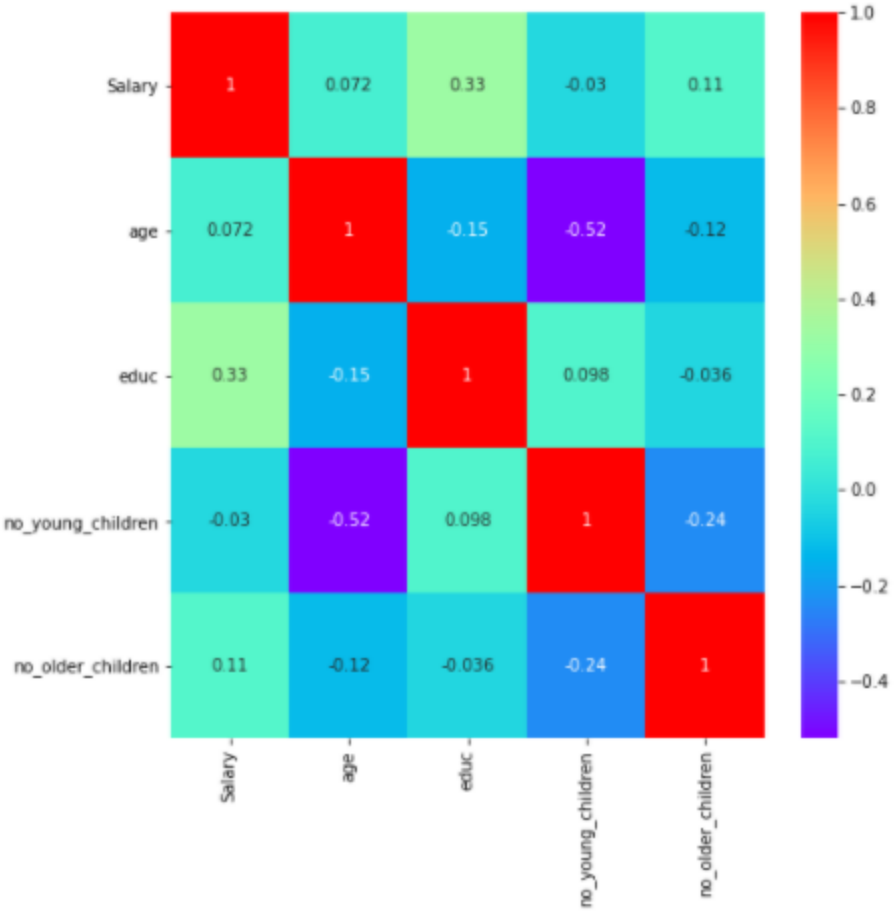
- *There are very minimum number of foreigners;majority of employees are not foreigners*
- *Majority of the employees have no children younger than 7 years of age*
- *Employees with no_older_children are more*
- *Majority of employees range between 25-50 years.*
- *Avg.years of education of majority of the employees ranges from 3-15 years.*
- *The Data states that only 45% of the employees chose holiday packages,rest 54% have not opted.*

MULTIVARIATE ANALYSIS:



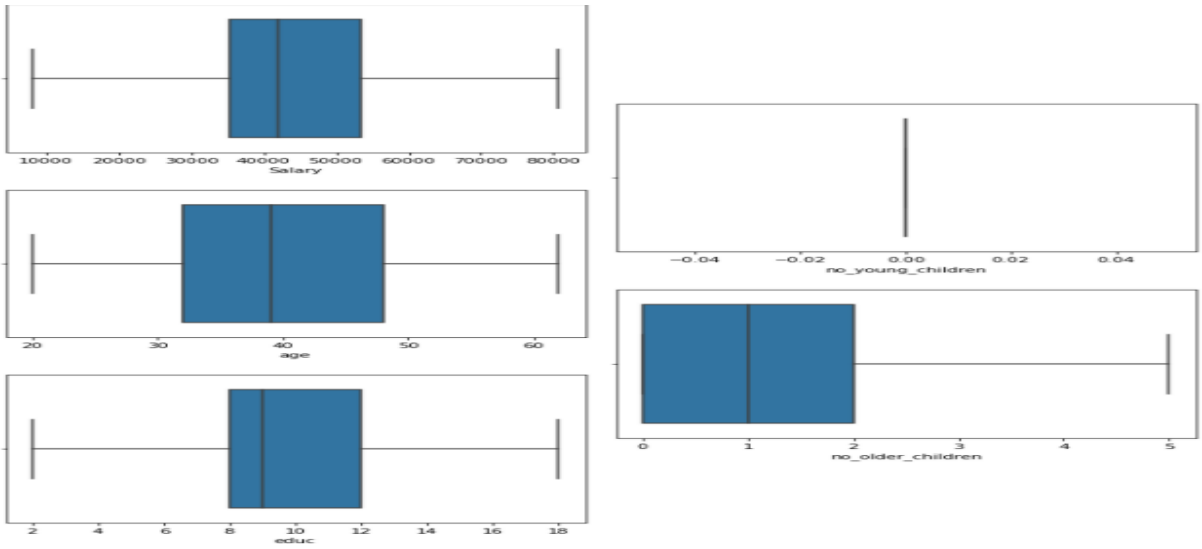
There is minimal correlation.Very Minimal correlation between Education & Salary.

Overall the data looks good.



As per the Heat Map, we can conclude that the variables are not highly correlated to each other.

OUTLIER TREATMENT:



We have treated the outliers in our data. The next step is to convert the categorical variables into numerical variables for Model Building.

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).

LOGISTIC REGRESSION

Logistic regression is a special case of linear regression where we only predict the outcome in a categorical variable. It predicts the probability of the event using the log function

Converting Categorical Variables into Numerical Variables for Model Building.

```
df2.info()
```

```
<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    float64
 2   age                872 non-null    float64
 3   educ               872 non-null    float64
 4   no_young_children  872 non-null    float64
 5   no_older_children  872 non-null    float64
 6   foreign            872 non-null    object 
dtypes: float64(5), object(2)
memory usage: 47.8+ KB
```

Splitting the dataset into Train and Test Data (70:30):

We will split the data into train and test data set in 70:30 ratio

We then extract the target column to separate vector for training & test set.We drop the Holiday Package column and assign it to X & pop the Holiday package column to vector Y.

```
X = df2.drop('Holliday_Package_yes',axis=1)
y = df2.pop('Holliday_Package_yes')

X.head()
```

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

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30 , random_state=1,stratify=y)
```

```
y_train.value_counts(normalize=True)
```

```
0    0.539344
1    0.460656
Name: Holliday_Package_yes, dtype: float64
```

We observe that 0s are 54% & 1s are 46%.The data seems to be balanced for the model building

LOGISTIC REGRESSION MODEL:

We used Logistic Regression Model with the following parameters

```
grid={'penalty':['l1','l2','none'],
      'solver':['liblinear','lbfgs'],
      'tol':[0.0001,0.00001]}
```

```
model = LogisticRegression(solver='newton-cg',max_iter=10000,penalty='none',verbose=True,n_jobs=2)
```

```
grid_search = GridSearchCV(estimator = model, param_grid = grid, cv = 3,n_jobs=-1,scoring='f1')
```

The Following were the best outcome of parameter grid.

```
print(grid_search.best_params_,'\n')
print(grid_search.best_estimator_)
```

```
{'penalty': 'l1', 'solver': 'liblinear', 'tol': 0.0001}

LogisticRegression(max_iter=10000, n_jobs=2, penalty='l1', solver='liblinear',
                   verbose=True)
```

LINEAR DISCRIMINANT ANALYSIS:

```
clf = LinearDiscriminantAnalysis()
model=clf.fit(X_train,Y_train)
model
```

```
LinearDiscriminantAnalysis()
```

```
pred_class_train = model.predict(X_train)
pred_class_test = model.predict(X_test)
```

We have fit our model into our train dataset using the above command & training the data for probability prediction.

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

PERFORMANCE METRICS: LOGISTIC REGRESSION MODEL

ACCURACY OF THE MODEL:

```
lr_train = best_model.score(X_train, y_train)
lr_train
```

```
0.6278688524590164
```

```
lr_test = best_model.score(X_test, y_test)
lr_test
```

```
0.6564885496183206
```

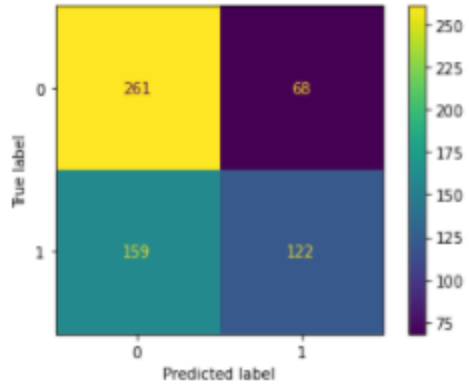
We have calculated the best accuracy scores on our train & test data of our model.

Classification Report & Confusion Matrix:

On Train Dataset:

```
plot_confusion_matrix(best_model,X_train,y_train)
print(classification_report(y_train, train_predict),'\n');
```

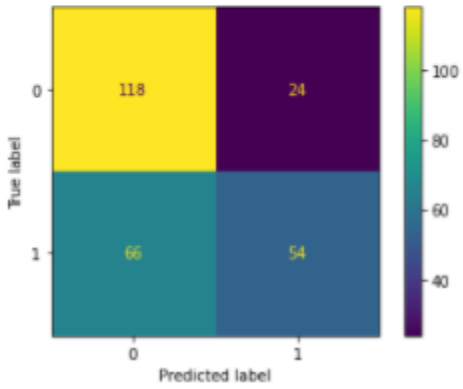
```
              precision    recall  f1-score   support

           0       0.62      0.79      0.70       329
           1       0.64      0.43      0.52       281

    accuracy                           0.63       610
   macro avg       0.63      0.61      0.61       610
weighted avg       0.63      0.63      0.61       610
```



On Test Dataset:

```
plot_confusion_matrix(best_model,X_test,y_test)
print(classification_report(y_test, test_predict),'\n');
```

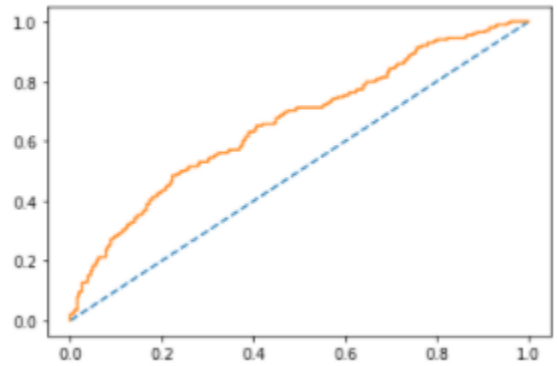
```
              precision    recall  f1-score   support

           0       0.64      0.83      0.72       142
           1       0.69      0.45      0.55       120

    accuracy                           0.66       262
   macro avg       0.67      0.64      0.63       262
weighted avg       0.66      0.66      0.64       262
```



The Accuracy of the model on train data is 63% ; accuracy of test data is 65%.The above Logistic model based on the accuracy & other critical values seems ok, however will build LDA to compare the scores .

AUC,ROC CURVE FOR TRAIN DATA:

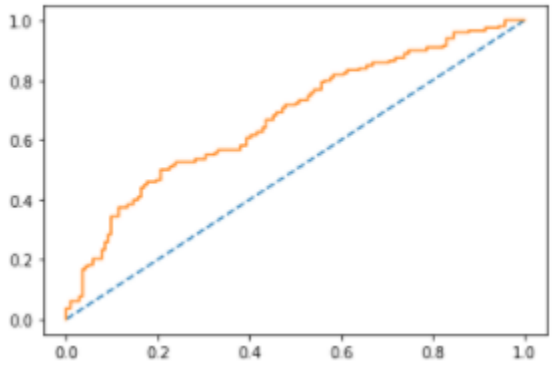
AUC: 0.662



The Area Under the Curve for Train Dataset is 66.2%

AUC,ROC CURVE FOR TEST DATA:

AUC: 0.675



The Area Under the Curve for Test Dataset is 67.5%

PERFORMANCE METRICS OF LDA MODEL

ACCURACY OF THE MODEL:

```
lda_train = model.score(X_train,Y_train)
lda_train
```

```
0.6327868852459017
```

```
lda_test = model.score(X_test,Y_test)
lda_test
```

```
0.6564885496183206
```

The accuracy of Train Data set is 63.21% and on test dataset is 65.12%.

Classification Report & Confusion Matrix:

On Train Dataset:

```
print(classification_report(Y_train, pred_class_train))
```

```
              precision    recall  f1-score   support

           0       0.62      0.80      0.70       329
           1       0.65      0.44      0.52       281

    accuracy                           0.63       610
   macro avg       0.64      0.62      0.61       610
weighted avg       0.64      0.63      0.62       610
```

```
confusion_matrix(Y_train, pred_class_train)
```

```
array([[263,  66],
       [158, 123]], dtype=int64)
```

On Test Dataset:

```
print(classification_report(Y_test, pred_class_test))
```

```
              precision    recall  f1-score   support

           0       0.64      0.83      0.72       142
           1       0.69      0.45      0.55       120

    accuracy                           0.66       262
   macro avg       0.67      0.64      0.63       262
weighted avg       0.66      0.66      0.64       262
```

```
confusion_matrix(Y_test, pred_class_test)
```

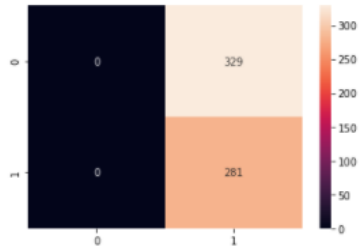
```
array([[118,  24],
       [ 66,  54]], dtype=int64)
```

CUT-OFF VALUES FOR MAX ACCURACY:

0.1

Accuracy Score 0.4607
F1 Score 0.6308

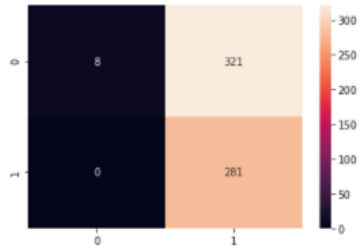
Confusion Matrix



0.2

Accuracy Score 0.4738
F1 Score 0.6365

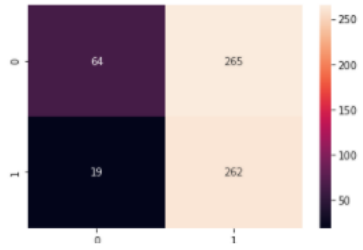
Confusion Matrix



0.3

Accuracy Score 0.5344
F1 Score 0.6485

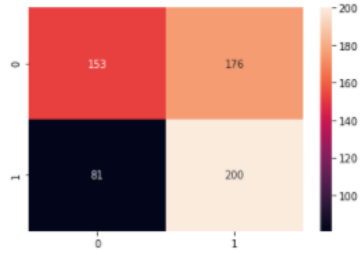
Confusion Matrix



0.4

Accuracy Score 0.5787
F1 Score 0.6088

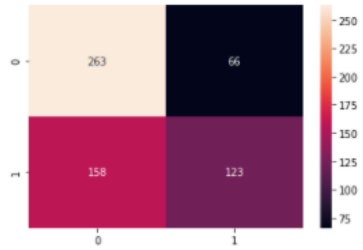
Confusion Matrix



0.5

Accuracy Score 0.6328
F1 Score 0.5234

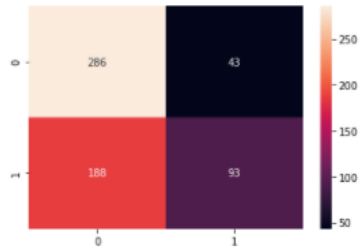
Confusion Matrix



0.6

Accuracy Score 0.6213
F1 Score 0.446

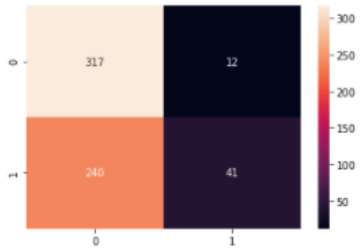
Confusion Matrix



0.7

Accuracy Score 0.5869
F1 Score 0.2455

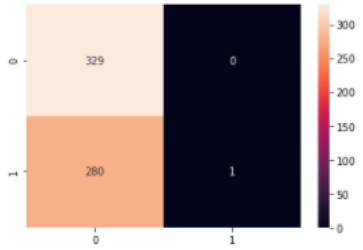
Confusion Matrix



0.8

Accuracy Score 0.541
F1 Score 0.0071

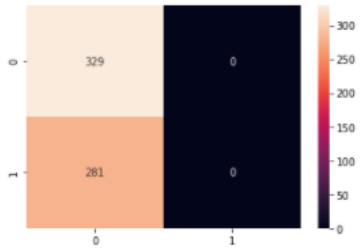
Confusion Matrix



0.9

Accuracy Score 0.5393
F1 Score 0.0

Confusion Matrix



We observe from the plot that 0.5 & 0.6 gives a better accuracy than rest of the cut-off values. 0.3 cut-off values gives the best gives us the best 'f1-score'. Henceforth,we will take the cut-off as 0.3 to get the optimum 'f1' score.

```
data_pred_custom_cutoff=[]
for i in range(0,len(pred_prob_test[:,1])):
    if np.array(pred_prob_test[:,1])[i]>0.3:
        a=1
    else:
        a=0
    data_pred_custom_cutoff.append(a)
```

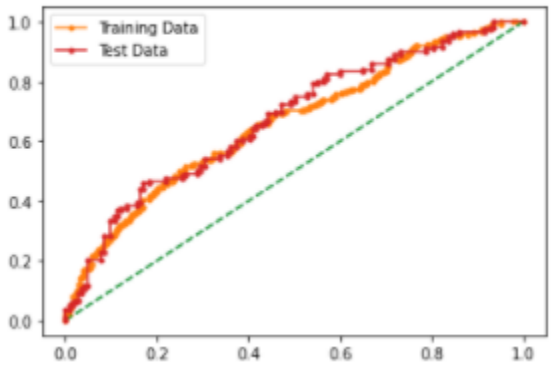
```
'Classification Report of the default cut-off test data:\n\n',metrics.classification_report(Y_test,pred_class_test),'\n\n\n')
'Classification Report of the custom cut-off test data:\n\n',metrics.classification_report(Y_test,data_pred_custom_cutoff),'\n'
```

```
Classification Report of the default cut-off test data:

              precision    recall  f1-score   support

           0       0.64      0.83      0.72       142
           1       0.69      0.45      0.55       120

    accuracy                           0.66       262
   macro avg       0.67      0.64      0.63       262
weighted avg       0.66      0.66      0.64       262



Classification Report of the custom cut-off test data:

              precision    recall  f1-score   support

           0       0.72      0.20      0.31       142
           1       0.49      0.91      0.64       120

    accuracy                           0.52       262
   macro avg       0.60      0.55      0.47       262
weighted avg       0.61      0.52      0.46       262
```

We have used the the custom cut-off values as 0.3 & derived the above results.Cut-off value 0.1 gives better scores.

COMPUTING AUC,ROC CURVE FOR TRAIN & TEST DATA:

```
AUC for the Training Data: 0.661
AUC for the Test Data: 0.675
```



The Area Under the Curve for Train Dataset is 66.1% & Train Dataset is 67.5%.

COMPARISON OF PERFORMANCE METRICS FOR OUR MODELS:

	Logistic Train	Logistic Test	LDA Train	LDA Test
Accuracy	0.63	0.66	0.63	0.66
AUC	0.66	0.67	0.66	0.67
Recall	0.43	0.45	0.44	0.45
Precision	0.64	0.69	0.65	0.69
F1 Score	0.52	0.55	0.52	0.55

We have done evaluations and have derived the outcomes of Confusion Matrix,Area Under Curve%,Accuracy%,Recall,Precision & F1 Score for the Train & Test of our model built using Logistic Regression & Linear Discriminant Analysis.Both,these models have almost similar scores.LDA could be preferred in classification cases & does better in the presence of categorical variables.Hence,we would prefer to explore with LDA.

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

Based on the information provided to us I.e Salary,age,education,no.of.children,foreigners the company seeks to determine the behaviour of the employees in opting for the holiday package.So,inorder to scale up their business we have observed the influence of the categorical variables using plots & also built two models Logistic Regression & LDA.

INFERENCE:

1. *As for the Model Prediction,Both Models gave similar results.LDA could be preferred over Logistic Regression model as it works better in the presence of categorical variables.*
2. *The EDA signifies that the employees aged above 50 are not interested as much in the holiday packages.Employees whose age group (25-50) are the ones who were much willing to go for a holiday package. Foreign Employees opt for holiday packages more than native employees.Salaried employees less than 50000 have opted for holiday packages more than employees whose salary more than 150000.*
3. *The Important Factors influencing the predictions are Salary,Education & Age.*
4. *No.of Older Children also impacts whether the employee would opt for a holiday package,so management can provide better packages,holiday destinations which would scale up the likeliness of an employee opting for a holiday package.*
5. *Based on these data factors, management could expand the package options, Offers,seasonal destinations,Targeting non-participanting employee groups For ex: Employee aging +50 could be given bonus offers ;presenting religious destinations*

THANK YOU