Business Report

SMDM Project Business Report DSBA

Sanjay Srinivasan
PGP-DSBA Online
JULY' 21 Batch
Date: 19-12-2021

INDEX

List Of Tables

List Of Figures

Problem - 1

Summary

The data is gathered from the company Gem Stones co ltd, which deals in distinguish between higher profitable stones and lower profitable stones to make better profitable stones. 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.

Introduction

The purpose of this exercise is to explore the dataset and make the price predictions for the diamonds, based on the higher and lower profitable stones.

Data Description

Carat	Carat weight of the cubic zirconia.
Cut	Describe the cut quality of the cubic zirconia. Quality is increasing order Fair, Good, Very Good, Premium, Ideal.
Color	Colour of the cubic zirconia.With D being the worst and J the best.
Clarity	Clarity refers to the absence of the Inclusions and Blemishes. (In order from Worst to Best in terms of avg price) IF, VVS1, VVS2, VS1, VS2, Sl1, Sl2, l1
Depth	The Height of cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter.
Table	The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter.
Price	The Price of the cubic zirconia.
X	Length of the cubic zirconia in mm.
Y	Width of the cubic zirconia in mm.
Z	Height of the cubic zirconia in mm.

Sample of the dataset:

	Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
0	1	0.30	Ideal	E	SI1	62.1	58.0	4.27	4.29	2.66	499
1	2	0.33	Premium	G	IF	60.8	58.0	4.42	4.46	2.70	984
2	3	0.90	Very Good	E	VVS2	62.2	60.0	6.04	6.12	3.78	6289
3	4	0.42	Ideal	F	VS1	61.6	56.0	4.82	4.80	2.96	1082
4	5	0.31	Ideal	F	VVS1	60.4	59.0	4.35	4.43	2.65	779

Table 1.1 Dataset Sample

Exploratory Data Analysis

Let us check the types of variables in the data frame.

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

Table- 1.2. Datatypes of the variable

There are total 26967 rows and 11 columns in the dataset. 6 columns are of float64 type , 3 columns are object and 2 columns are int64

Check for missing values in the dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26967 entries, 0 to 26966
Data columns (total 11 columns):
Unnamed: 0    26967 non-null int64
carat         26967 non-null float64
cut           26967 non-null object
color         26967 non-null object
clarity       26967 non-null object
depth         26270 non-null float64
table         26967 non-null float64
x             26967 non-null float64
y             26967 non-null float64
z             26967 non-null float64
price         26967 non-null int64
dtypes: float64(6), int64(2), object(3)
memory usage: 2.3+ MB
```

Table- 1.3. Check null values

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

Uni-Variate Analysis:

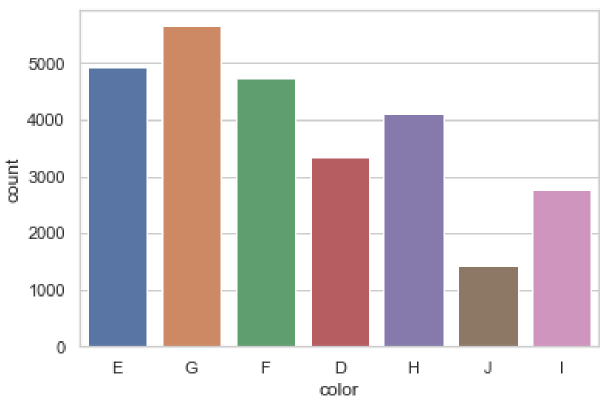


Fig – 1.1 diamond colour count

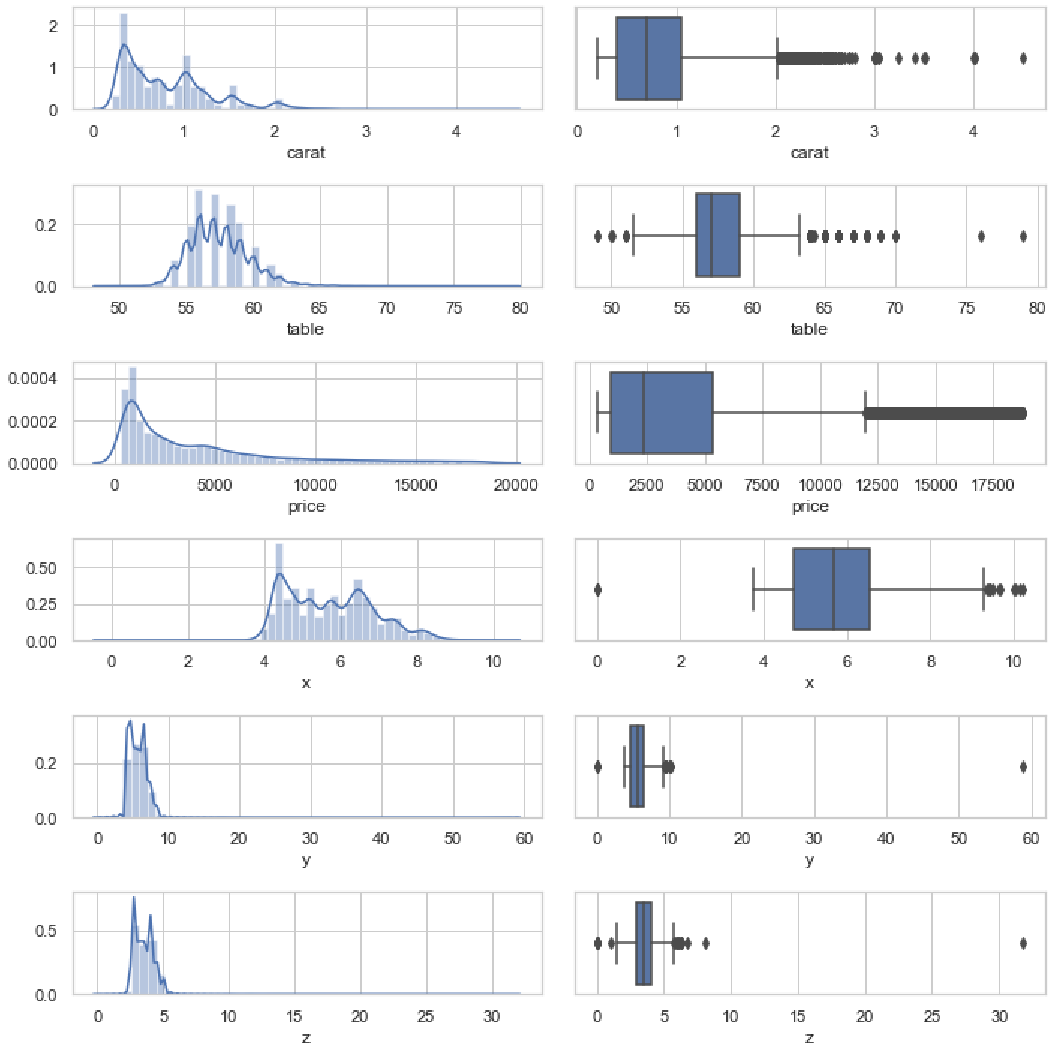


Fig – 1.2 Univariate Analysis

From the above chart (displot and boxplot), there are outliers present in the data.

<u>Bi – variate Analysis:</u>

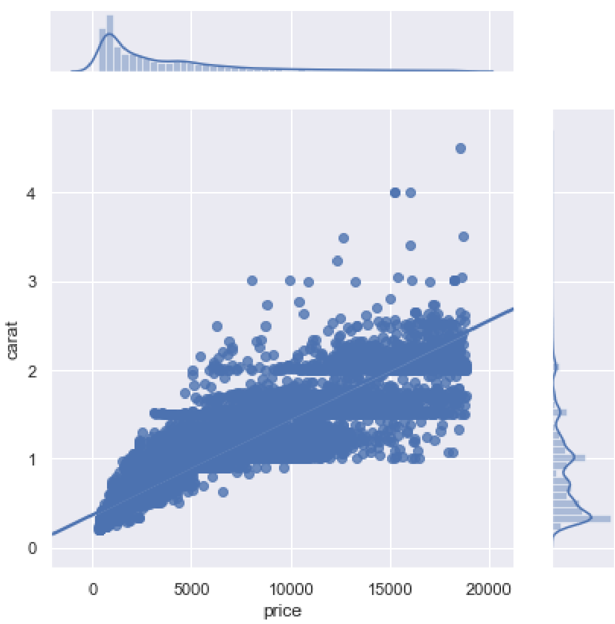


Fig – 1.3 Jointplot for price vs. carat using Bivariate Analysis

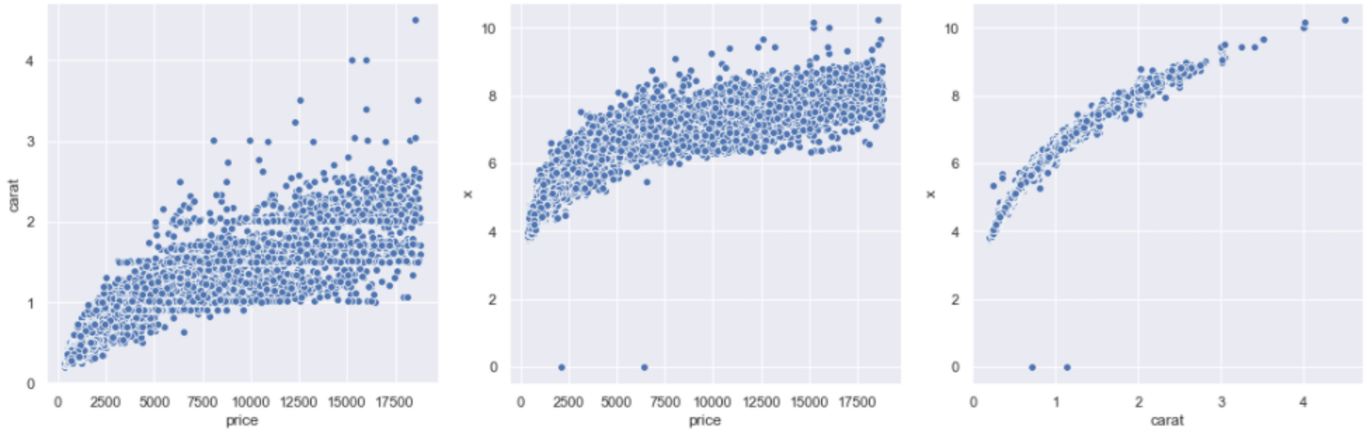


Fig – 1.4 Bivariate Analysis

From the scatterplot, we can infer that as the 'price' and ‘carat’ increases, the ‘carat’, 'x' increases.

<u>Multi – variate Analysis:</u>

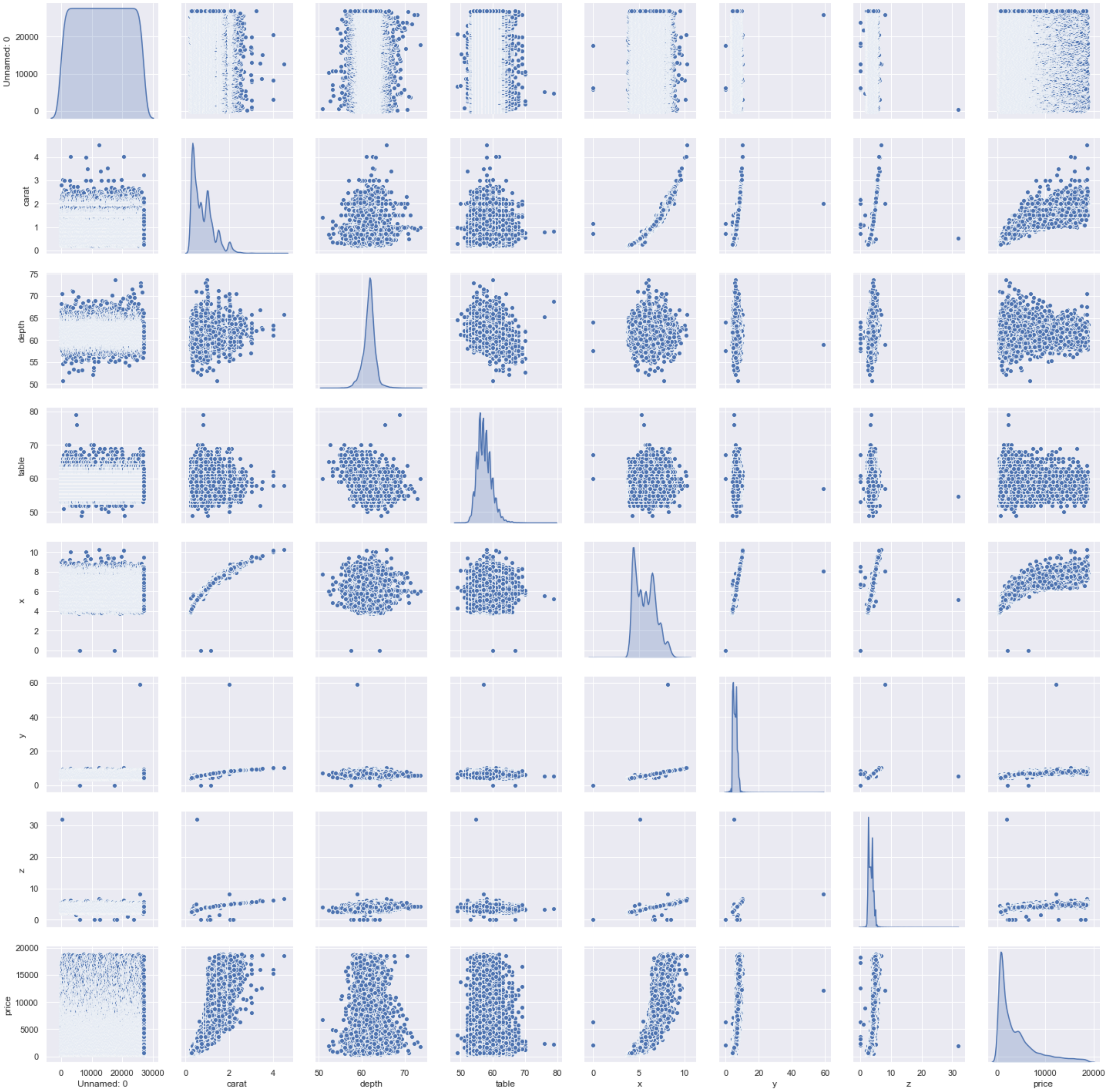


Fig – 1.5 Multivariate analysis of pairplot

	Unnamed: 0	carat	depth	table	x	y	z	price
Unnamed: 0	1.000000	0.003490	-0.001588	0.003817	0.004626	0.006844	0.001681	0.002650
carat	0.003490	1.000000	0.035364	0.181685	0.976368	0.941071	0.940640	0.922416
depth	-0.001588	0.035364	1.000000	-0.298011	-0.018715	-0.024735	0.101624	-0.002569
table	0.003817	0.181685	-0.298011	1.000000	0.196206	0.182346	0.148944	0.126942
x	0.004626	0.976368	-0.018715	0.196206	1.000000	0.962715	0.956606	0.886247
y	0.006844	0.941071	-0.024735	0.182346	0.962715	1.000000	0.928923	0.856243
z	0.001681	0.940640	0.101624	0.148944	0.956606	0.928923	1.000000	0.850536
price	0.002650	0.922416	-0.002569	0.126942	0.886247	0.856243	0.850536	1.000000

Fig – 1.6 Multivariate analysis for correlation

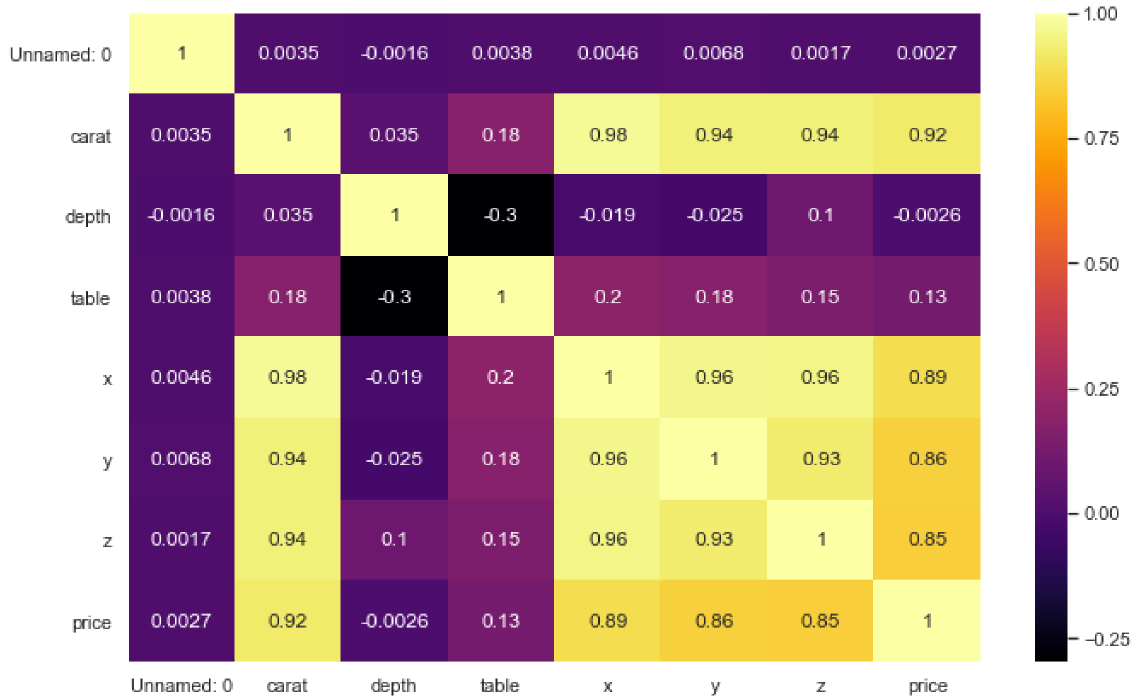


Fig – 1.7 Multivariate analysis of plotting correlation in heatmap

1.2 Impute null values if present, also check for the values which are equal to zero. Do they have any meaning or do we need to change them or drop them? Check for the possibility of combining the sub levels of a ordinal variables and take actions accordingly. Explain why you are combining these sub levels with appropriate reasoning.

```
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
```

Fig – 1.8 Null values count

	Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
5821	5822	0.71	Good	F	SI2	64.1	60.0	0.00	0.00	0.0	2130
6034	6035	2.02	Premium	H	VS2	62.7	53.0	8.02	7.95	0.0	18207
6215	6216	0.71	Good	F	SI2	64.1	60.0	0.00	0.00	0.0	2130
10827	10828	2.20	Premium	H	SI1	61.2	59.0	8.42	8.37	0.0	17265
12498	12499	2.18	Premium	H	SI2	59.4	61.0	8.49	8.45	0.0	12631
12689	12690	1.10	Premium	G	SI2	63.0	59.0	6.50	6.47	0.0	3696
17506	17507	1.14	Fair	G	VS1	57.5	67.0	0.00	0.00	0.0	6381
18194	18195	1.01	Premium	H	I1	58.1	59.0	6.66	6.60	0.0	3167
23758	23759	1.12	Premium	G	I1	60.4	59.0	6.71	6.67	0.0	2383

Table – 1.4 Null values in x,y,z variable

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

Table – 1.5 After null values treatment

Before Scaling and treating outliers:

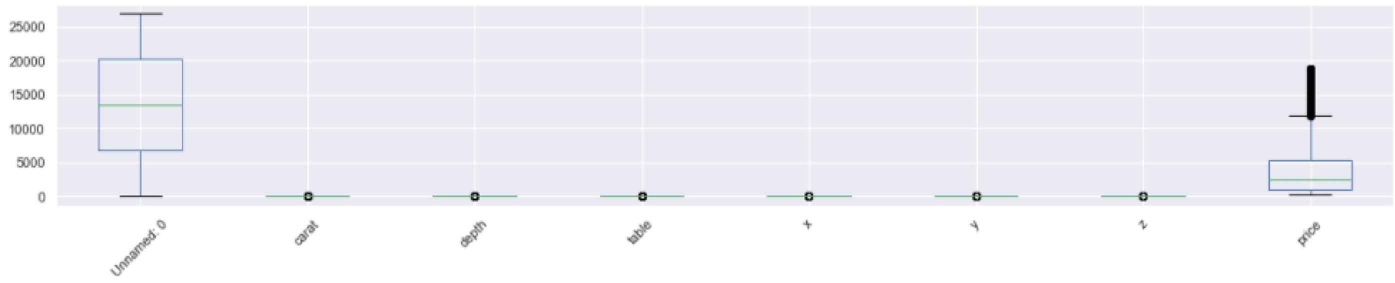


Fig – 1.9 Before Scaling

After Scaling:

	Unnamed: 0	carat	depth	table	x	y	z	price
0	-1.731904	-1.043125	0.253399	0.244112	-1.295920	-1.240065	-1.224865	-0.854851
1	-1.731776	-0.980310	-0.679158	0.244112	-1.162787	-1.094057	-1.169142	-0.734303
2	-1.731647	0.213173	0.325134	1.140496	0.275049	0.331668	0.335404	0.584271
3	-1.731519	-0.791865	-0.105277	-0.652273	-0.807766	-0.802041	-0.806936	-0.709945
4	-1.731390	-1.022187	-0.966099	0.692304	-1.224916	-1.119823	-1.238796	-0.785257

Table – 1.6 After scaling

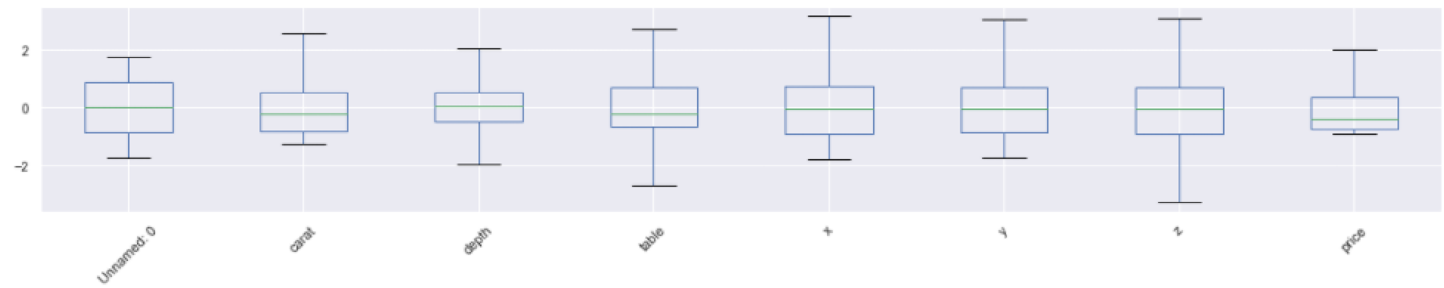


Fig – 1.10 After Scaling

Yes, Scaling needs to be done as the values of the variables are different. price, carat, x, y, z, depth, table are in different values and this may get more weightage. The plot of the data prior and after scaling. Scaling will have all the values in the relative same range. I have used z-score to standardised the data to relative same scale -3 to +3

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

Encoding the data:

carat	depth	table	x	y	z	price	cut_Good	cut_Ideal	...	color_H	color_I	color_J	clarity_IF	clarity_SI1	clarity_SI2
-1.043125	0.253399	0.244112	-1.295920	-1.240065	-1.224865	-0.854851	0	1	...	0	0	0	0	1	0
-0.980310	-0.679158	0.244112	-1.162787	-1.094057	-1.169142	-0.734303	0	0	...	0	0	0	1	0	0
0.213173	0.325134	1.140496	0.275049	0.331668	0.335404	0.584271	0	0	...	0	0	0	0	0	0
-0.791865	-0.105277	-0.652273	-0.807766	-0.802041	-0.806936	-0.709945	0	1	...	0	0	0	0	0	0
-1.022187	-0.966099	0.692304	-1.224916	-1.119823	-1.238796	-0.785257	0	1	...	0	0	0	0	0	0

Table – 1.7 Sample Encoded data

From the above dataframe, we can infer that the data are encoded using get dummies encoding method.

Dataframe after separating the Target variable:

	carat	depth	table	x	y	z	cut_Good	cut_Ideal	cut_Premium	cut_Very Good	...	color_H	color_I	color_J	clarity_IF	cla
0	-1.043125	0.253399	0.244112	-1.295920	-1.240065	-1.224865	0	1	0	0	...	0	0	0	0	
1	-0.980310	-0.679158	0.244112	-1.162787	-1.094057	-1.169142	0	0	1	0	...	0	0	0	1	
2	0.213173	0.325134	1.140496	0.275049	0.331668	0.335404	0	0	0	1	...	0	0	0	0	
3	-0.791865	-0.105277	-0.652273	-0.807766	-0.802041	-0.806936	0	1	0	0	...	0	0	0	0	
4	-1.022187	-0.966099	0.692304	-1.224916	-1.119823	-1.238796	0	1	0	0	...	0	0	0	0	

Table – 1.8 sample Dataframe without the TARGET variable

	price
0	-0.854851
1	-0.734303
2	0.584271
3	-0.709945
4	-0.785257

Table – 1.9 sample Dataframe with the TARGET variable

LINEAR REGRESSION:

```
The coefficient for carat is 1.081627832522888
The coefficient for depth is 0.012467768513946348
The coefficient for table is -0.010375104168364378
The coefficient for x is -0.28583158930894487
The coefficient for y is 0.3338652177635207
The coefficient for z is -0.17104194176343152
The coefficient for cut_Good is 0.08632175838808641
The coefficient for cut_Ideal is 0.14741709168972442
The coefficient for cut_Premium is 0.13708966355071203
The coefficient for cut_Very Good is 0.11958512818375423
The coefficient for color_E is -0.056527359921143654
The coefficient for color_F is -0.0696128596080928
The coefficient for color_G is -0.11639282676700596
The coefficient for color_H is -0.21601184437275148
The coefficient for color_I is -0.33268780880422083
The coefficient for color_J is -0.47964981363888193
The coefficient for clarity_IF is 1.014064310227127
The coefficient for clarity_SI1 is 0.6466590629622312
The coefficient for clarity_SI2 is 0.43938535770675996
The coefficient for clarity_VS1 is 0.8505783926251638
The coefficient for clarity_VS2 is 0.7780939346329789
The coefficient for clarity_VVS1 is 0.9446069412003394
The coefficient for clarity_VVS2 is 0.9476542157836995
```

Fig – 1.11 Coefficient of independent variable

```
The intercept for our model is -0.7523485007042121
```

Fig – 1.12 Intercept of our model

```
R square value for training data  0.9404067032706919
```

Fig – 1.13 R – Square Value for Training data

```
R square value for testing data  0.9416664173652372
```

Fig – 1.14 R – Square Value for Testing data

```
Mean squared error for the training data is  0.20982119520296377
```

Fig – 1.15 Mean squared error for training data

```
Mean squared error for the testing data is  0.2098640889031238
```

Fig – 1.16 Mean squared error for Testing data

```
carat ---> 33.35086119845924
depth ---> 4.573918951598584
table ---> 1.772885281261897
x ---> 463.5542785436457
y ---> 462.769821646584
z ---> 238.65819968687333
cut_Good ---> 3.609618194943713
cut_Ideal ---> 14.34812508118844
cut_Premium ---> 8.623414379121153
cut_Very Good ---> 7.848451571723695
color_E ---> 2.371070464762613
```

Fig – 1.17 Variance Inflation Factor of our model

RIDGE REGRESSION:

```
The coefficient for the Ridge model: [[ 1.08097395  0.01236059 -0.01041468 -0.28263441  0.33056432 -0.17055804
   0.08692936  0.14804495  0.13757653  0.1203011  -0.05636345 -0.06942944
  -0.11614727 -0.21578484 -0.33237215 -0.47917899  1.00672437  0.63986318
   0.43266118  0.84368165  0.7712573   0.93753083  0.9406768 ]]
```

Fig – 1.18 Coefficient of independent variable

```
R square value for training data  0.9404058025987212
R square value for testing data  0.9416568212139137
```

Fig – 1.19 R – Square Value for Training and Testing data

```
The intercept for our model is [-0.7463866]
```

Fig – 1.20 Intercept of our model

```
Mean squared error for the training data is  0.2098227807785714
```

Fig – 1.21 Mean squared error for Training data

```
Mean squared error for the testing data is  0.20988135001282085
```

Fig – 1.22 Mean squared error for Testing data

ORDINARY LEAST SQUARE METHOD :

```
The Coefficient are


  Intercept        -0.781720
carat             1.086958
depth             0.008694
table            -0.011344
x                -0.303284
y                 0.318809
z                -0.142880
cut_Good          0.097282
cut_Ideal         0.156601
cut_Premium       0.151019
cut_Very_Good     0.129184
color_E          -0.049548
color_F          -0.063134
color_G          -0.104572
color_H          -0.210754
color_I          -0.328478
color_J          -0.472208
clarity_IF        1.016587
clarity_SI1       0.658588
clarity_SI2       0.454275
clarity_VS1       0.861128
clarity_VS2       0.789454
clarity_VVS1      0.966044
clarity_VVS2      0.960458
dtype: float64
```

Fig – 1.23 Coefficient of independent variable

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.941
Model:                            OLS   Adj. R-squared:                  0.941
Method:                 Least Squares   F-statistic:                 1.862e+04
Date:                Sat, 18 Dec 2021   Prob (F-statistic):               0.00
Time:                        21:39:21   Log-Likelihood:                 3852.4
No. Observations:               26958   AIC:                            -7657.
Df Residuals:                   26934   BIC:                            -7460.
Df Model:                          23
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
Intercept        -0.7817      0.014    -57.768      0.000      -0.808      -0.755
carat             1.0870      0.008    142.536      0.000       1.072       1.102
depth             0.0087      0.003      2.780      0.005       0.003       0.015
table            -0.0113      0.002     -6.383      0.000      -0.015      -0.008
x                -0.3033      0.028    -10.961      0.000      -0.358      -0.249
y                 0.3188      0.029     10.986      0.000       0.262       0.376
z                -0.1429      0.020     -6.980      0.000      -0.183      -0.103
cut_Good          0.0973      0.009     10.785      0.000       0.080       0.115
cut_Ideal         0.1566      0.009     17.821      0.000       0.139       0.174
cut_Premium       0.1510      0.008     17.909      0.000       0.134       0.168
cut_Very_Good     0.1292      0.009     14.981      0.000       0.112       0.146
color_E          -0.0495      0.005    -10.511      0.000      -0.059      -0.040
color_F          -0.0631      0.005    -13.219      0.000      -0.072      -0.054
color_G          -0.1046      0.005    -22.417      0.000      -0.114      -0.095
color_H          -0.2108      0.005    -42.348      0.000      -0.221      -0.201
color_I          -0.3285      0.006    -59.270      0.000      -0.339      -0.318
color_J          -0.4722      0.007    -69.454      0.000      -0.486      -0.459
clarity_IF        1.0166      0.013     75.358      0.000       0.990       1.043
clarity_SI1       0.6586      0.012     57.163      0.000       0.636       0.681
clarity_SI2       0.4543      0.012     39.232      0.000       0.432       0.477
clarity_VS1       0.8611      0.012     73.230      0.000       0.838       0.884
clarity_VS2       0.7895      0.012     68.162      0.000       0.767       0.812
clarity_VVS1      0.9660      0.012     77.536      0.000       0.942       0.990
clarity_VVS2      0.9605      0.012     79.250      0.000       0.937       0.984
==============================================================================
Omnibus:                     6631.652   Durbin-Watson:                   2.006
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            25175.638
Skew:                           1.191   Prob(JB):                         0.00
Kurtosis:                       7.091   Cond. No.                         58.0
==============================================================================

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

Fig – 1.24 OLS Summary

```
Mean squared error for the testing data is  0.20986408890312383
```

Fig – 1.25 OLS Summary Mean squared error

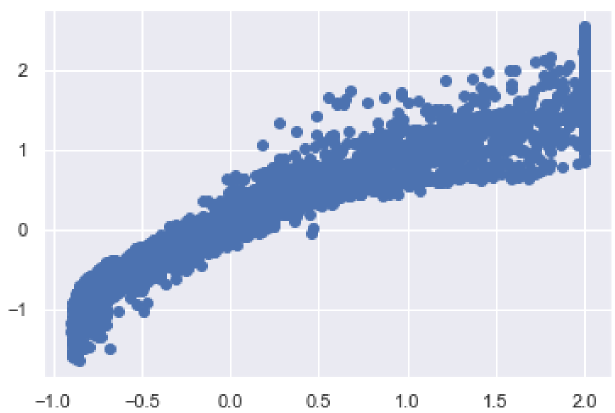


Fig – 1.26 predicted Output vs. testing data

```
price =
 (-0.78) * Intercept +  (1.09) * carat +  (0.01) * depth +  (-0.01) * table +  (-0.3) * x +  (0.32) * y +  (-0.14) * z +  (0.1)
* cut_Good +  (0.16) * cut_Ideal +  (0.15) * cut_Premium +  (0.13) * cut_Very_Good +  (-0.05) * color_E +  (-0.06) * color_F +
(-0.1) * color_G +  (-0.21) * color_H +  (-0.33) * color_I +  (-0.47) * color_J +  (1.02) * clarity_IF +  (0.66) * clarity_SI1
+  (0.45) * clarity_SI2 +  (0.86) * clarity_VS1 +  (0.79) * clarity_VS2 +  (0.97) * clarity_VVS1 +  (0.96) * clarity_VVS2 +
```

Fig – 1.27 predicted Output vs. testing data Linear equation

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

Based on the R2(R squared) value, The company making out the better profitable share by distinguishing the higher profitable stones and lower profitable stones so that the company can increase the price for higher and lower profitable stones to make higher profits.

The best 5 attributes that are most important from the coefficients are carat, clarity_IF, clarity_VVS2, clarity_VVS1, clarity_VS1. These are the best attributes that will increase the price of the diamond costliest.

Business Insights:

Train set:

rsquare: 0.94

adjusted square: 0.94

R - Square is 0.94 which tells the correlation between price of diamonds vs. different independent variable's explained by 94%

If we see the final model:

price = (-0.78) * Intercept + (1.09) * carat + (0.01) * depth + (-0.01) * table + (-0.3) * x + (0.32) * y + (-0.14) * z + (0.1) * cut_Good + (0.16) * cut_Ideal + (0.15) * cut_Premium + (0.13) * cut_Very_Good + (-0.05) * color_E + (-0.06) * color_F + (-0.1) * color_G + (-0.21) * color_H + (-0.33) * color_I + (-0.47) * color_J + (1.02) * clarity_IF + (0.66) * clarity_SI1 + (0.45) * clarity_SI2 + (0.86) * clarity_VS1 + (0.79) * clarity_VS2 + (0.97) * clarity_VVS1 + (0.96) * clarity_VVS2

Co-efficient of the Carat is highest most, which signifies if there is increase of one unit of carat there will increase of 1.09 in price.

Next most positive effecting independent variable is IF clarity type variable.

Most Negatively effecting parameter is J colour type diamonds ,means a loss of -0.47 will occur with decrease the price of one unit of J colour type diamonds.

Test set:

rsquare: 0.941

Adjusted square: 0.941

Finally, Our linear model is good as the r-square difference in train & test dataset is less than 5%.

Recommendations :

To Increase the price of the diamond, carat, clarity of the diamond needs to be increased so that the price of the diamond increase which in turn increases profits. The company can sell the diamonds and make higher profitable price from the stone with lower profitable price.

Problem – 2

Summary

The data is gathered from an tour and travel agency which deals in selling holiday packages to sell their packages to employees. 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.

Introduction

The purpose of this exercise is to sell tour and travel packages to their employees by predicting the employees would buy tour and travel package using Logistic regression and Linear discriminant analysis (LDA). This dataset consist of 872 rows and 8 columns,

Data Description

1. Holiday_Package: Opted for Holiday Package yes/no?
2. Salary: Employee salary
3. Age: Age in years
4. Edu: Years of formal education
5. No_young_children: The number of young children (younger than 7 years)
6. No_older_children: Number of older children
7. Foreign: foreigner Yes/No

Sample of the dataset:

	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

Table 2.1 Dataset Sample

Dataset has 8 variables with Tour and travel package. Based on the travel package, employees will buy the tour and travel packages.

Exploratory Data Analysis

Let us check the types of variables in the data frame.

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

Table 2.2 Datatypes of the variable

There are total 872 rows and 8 columns in the dataset. Out of 8, 2 column is of Object type and rest 6 are of integer data type.

Check for missing values in the dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 10 columns):
Age             3000 non-null int64
Agency_Code     3000 non-null object
Type            3000 non-null object
Claimed         3000 non-null object
Commision       3000 non-null float64
Channel         3000 non-null object
Duration        3000 non-null int64
Sales           3000 non-null float64
Product Name    3000 non-null object
Destination     3000 non-null object
dtypes: float64(2), int64(2), object(6)
memory usage: 234.5+ KB
```

Table 2.3 Check null values

From this, it is clear that there are no null values present in the dataset.

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

There are no null values present in the dataset

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

Table 2.4 Null values

The numbers of unique variables are taken from the categorical column.

```
HOLLIDAY_PACKAGE :  2
yes    401
no     471
Name: Holliday_Package, dtype: int64


FOREIGN :  2
yes    216
no     656
Name: foreign, dtype: int64
```

Fig – 2.1 Categorical value count

The numbers of duplicate values are taken from the dataset and duplicate records have been dropped from the dataset.

```
Number of duplicate rows = 0
```

Fig – 2.2 Number of duplicate rows

Univariate Analysis:

Univariate analysis is the simplest form of analysing data. Analyzing each variable in a detailed manner. There are outliers present in all variables except age.

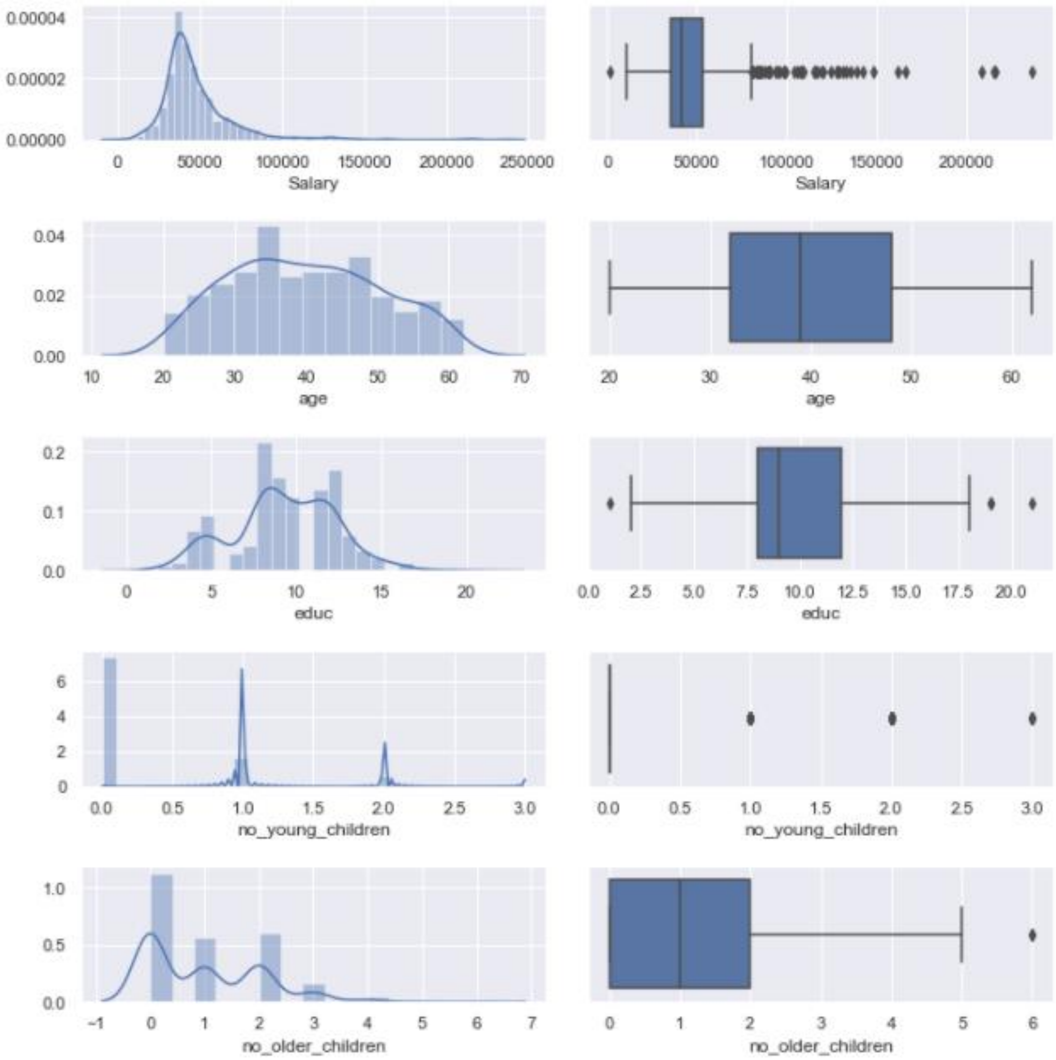


Fig – 2.3 Univariate analysis

The value counts from the Holiday Package variable are, 401 customers have opted for Holiday Package whereas 471 customers have not opted for Holiday Package.

```
no     471
yes    401
Name: Holliday_Package, dtype: int64
```

Fig – 2.4 Univariate analysis for Holiday package

The value counts from the Foreign variable are, 216 customers have opted for Foreign whereas 656 customers have not opted for Foreign.

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

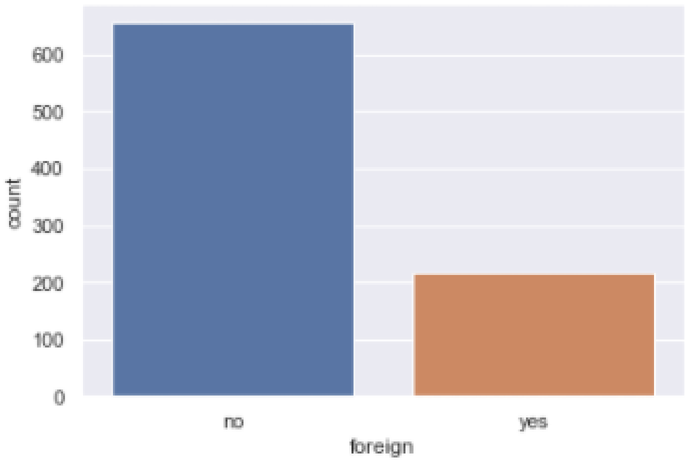


Fig – 2.5 Univariate analysis for Foreign

<u>Bivariate Analysis:</u>

Bivariate analysis is the simplest form of analysing data. Analyzing a single variable with another variable in detail.

<u>Insights:</u>

The Educ (Years of Education) is plotted against the salary. This Bi-variate plot shows the 20+ years of education earns 60000 whereas, education with 12 years, earns the salary greater than 200000.

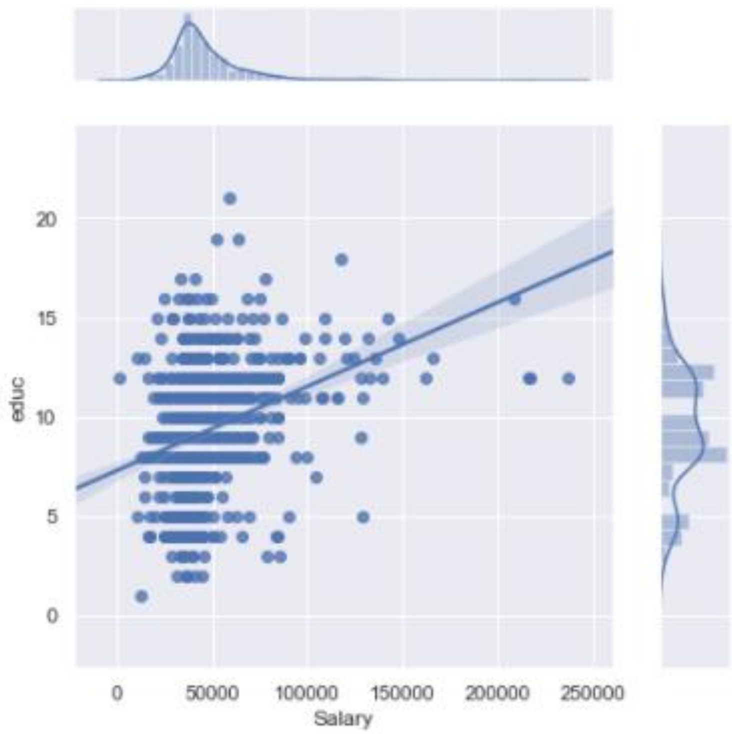


Fig – 2.6 Bivariate analysis Salary vs. Educ

The salary and age in the x-axis is plotted against education and age. The plot 1 denotes, Only few employees with 10-17 years of education earns salary greater than 200000. In 2^{nd} plot, more number of employees gets salary in the range of 50000 - 100000.

Fig – 2.7 Bivariate analysis

Multivariate Analysis:

Analysing the data with two or more variable.

Insights:

There is no strong correlation observed between few fields.

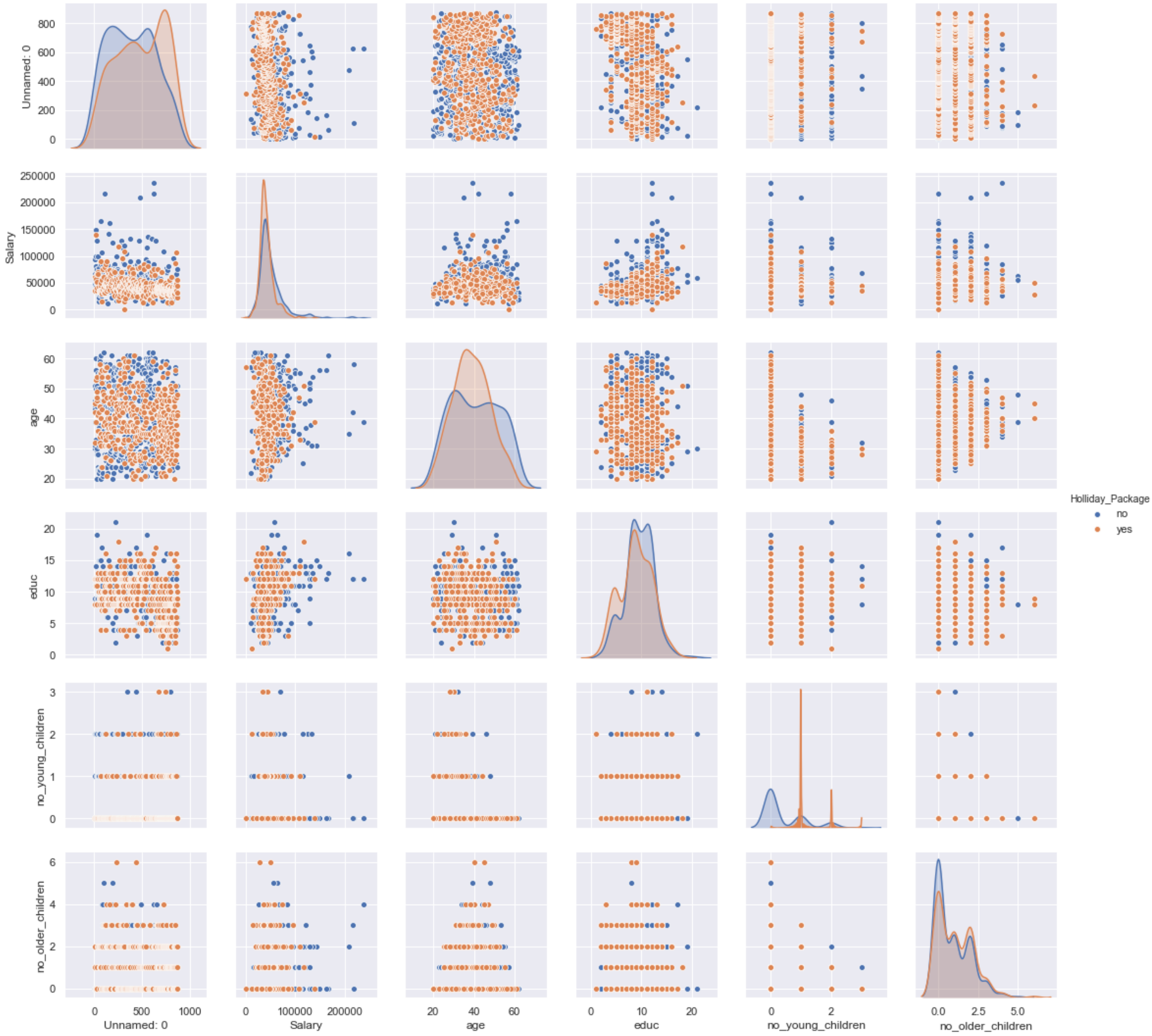


Fig – 2.8 Multivariate analysis of pairplot

Fig – 2.9 Multivariate analysis heatmap

Before Treating Outlier :

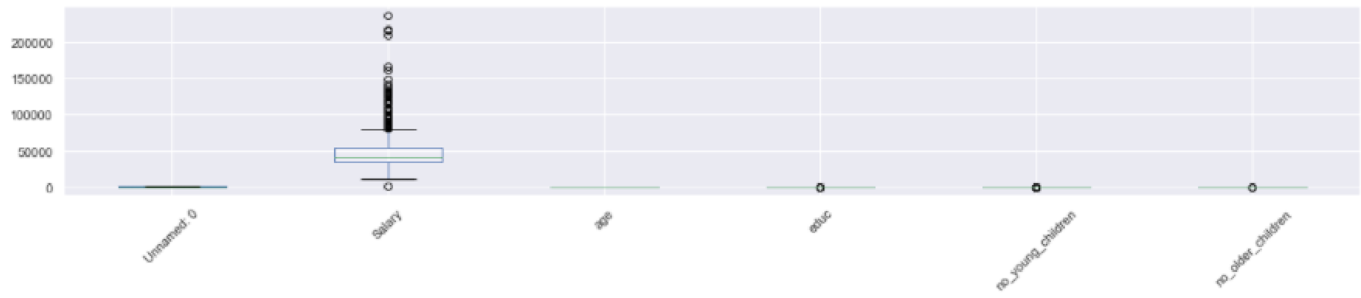


Fig – 2.10 Before Treating Outlier

Fig – 2.11 After Treating Outlier

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

Table – 2.5 Sample Dataset after dropping Unnamed :0 column.

2.2 Do not scale the data. Encode the data (having string values) for Modelling. Data Split: Split the data into train and test (70:30). Apply Logistic Regression and LDA (linear discriminant analysis).

The get dummies encoding method is performed for Holiday_package and Foreign column.

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

Table 2.6 Sample dataframe after Encoding

The Sample dataframe after removing the Target variable from the original dataframe. The dataframe are split into train and test data from the dataframe. The train data has 70% of the data and test data has 30% of the data from the dataframe.

	Salary	age	educ	no_young_children	no_older_children	foreign_yes
400	59692.00	43.0	11.0	0.0	2.0	0
234	22366.00	55.0	7.0	0.0	2.0	0
338	41582.00	36.0	9.0	0.0	3.0	0
71	35344.00	35.0	9.0	0.0	2.0	0
727	80687.75	55.0	15.0	0.0	0.0	1

Table 2.7 Train dataframe

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:

The Logistic Regression with the parameters using the grid search CV. The model is fitted into the Logistic Regression using the grid search CV.

```
GridSearchCV(cv=3, estimator=LogisticRegression(max_iter=100000, n_jobs=2),
             n_jobs=-1,
             param_grid={'penalty': ['l1', 'l2', 'none'],
                         'solver': ['lbfgs', 'liblinear'],
                         'tol': [0.0001, 1e-06]},
             scoring='f1')
```

Fig – 2.12 Parameters for GridsearchCV in Logistic Regression

The best parameters are identified from the decision tree algorithm by using the grid search CV.

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

Fig – 2.13 Best parameter for Logistic Regression

```
LogisticRegression(max_iter=100000, n_jobs=2, penalty='l1', solver='liblinear')
```

Fig – 2.14 Best estimator for Logistic Regression

The values are predicted from the train data.

```
array([0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0,
       0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0,
       1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1,
       0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0,
       0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1,
       0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0,
       1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0,
       0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0], dtype=uint8)
```

Fig – 2.15 Predicted values from the train dataset of Logistic Regression model

Confusion Matrix is obtained from the train data and test data using Logistic Regression.

```
array([[275,  54],
       [168, 113]], dtype=int64)
```

Fig 2.16 confusion matrix from Train data of Logistic Regression

```
array([[117,  25],
       [ 73,  47]], dtype=int64)
```

Fig 2.17 confusion matrix from test data of Logistic Regression

	precision	recall	f1-score	support
0	0.62	0.84	0.71	329
1	0.68	0.40	0.50	281
accuracy			0.64	610
macro avg	0.65	0.62	0.61	610
weighted avg	0.65	0.64	0.62	610

Fig 2.18 Classification Report from train data of Logistic Regression

	precision	recall	f1-score	support
0	0.62	0.82	0.70	142
1	0.65	0.39	0.49	120
accuracy			0.63	262
macro avg	0.63	0.61	0.60	262
weighted avg	0.63	0.63	0.61	262

Fig 2.19 Classification Report from test data of Logistic Regression

ROC curve (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters.

- True Positive Rate
- False Positive Rate

The probability of the Area under the ROC curve for the train data is 66.1%

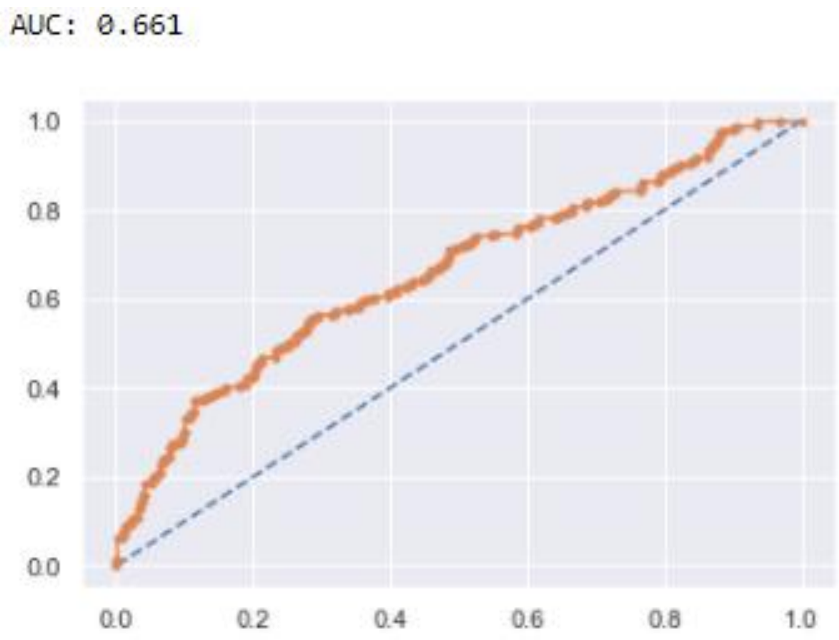


Fig 2.20 AUC and ROC curve train data of Logistic Regression

The probability of the Area under the ROC curve for the train data is 67.3%

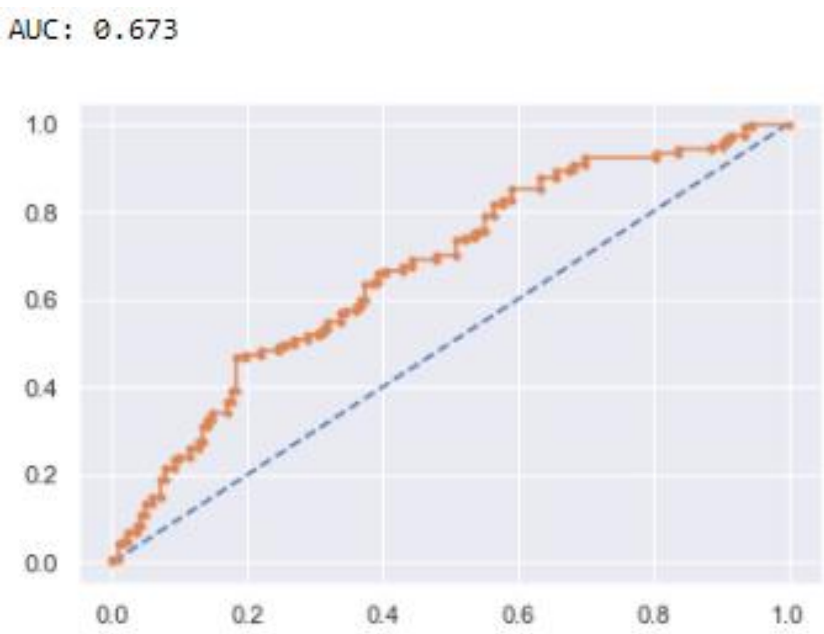


Fig 2.21 AUC and ROC curve test data of Logistic Regression

Linear Discriminant Analysis:

The values are predicted from the train data.

```
array([0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1,
       0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1,
       0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1,
       0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0,
       0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1,
       1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0,
       1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1,
       0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0], dtype=uint8)
```

Fig – 2.22 Predicted values from the train dataset of LDA

Confusion Matrix is obtained from the train data and test data using Random Forest Algorithm.

```
array([[268,  63],
       [155, 124]], dtype=int64)
```

Fig 2.23 confusion matrix from Train data of LDA Model

```
array([[117,  23],
       [ 72,  50]], dtype=int64)
```

Fig 2.24 confusion matrix from test data of LDA Model

	precision	recall	f1-score	support
0	0.63	0.81	0.71	331
1	0.66	0.44	0.53	279
accuracy			0.64	610
macro avg	0.65	0.63	0.62	610
weighted avg	0.65	0.64	0.63	610

Fig 2.25 Classification Report from train data of LDA Model

	precision	recall	f1-score	support
0	0.62	0.84	0.71	140
1	0.68	0.41	0.51	122
accuracy			0.64	262
macro avg	0.65	0.62	0.61	262
weighted avg	0.65	0.64	0.62	262

Fig 2.26 Classification Report from test data of LDA Model

ROC curve (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters.

- True Positive Rate
- False Positive Rate

The probability of the Area under the ROC curve for the train data is 66.9%.

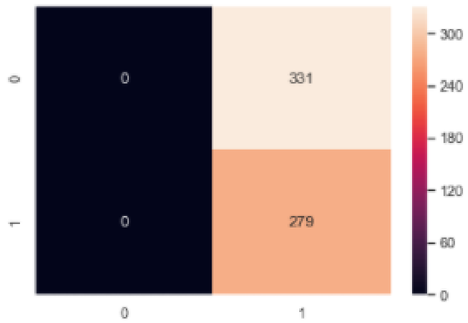
The probability of the Area under the ROC curve for the train data is 65.5%.

```
AUC for the Training Data: 0.669
AUC for the Test Data: 0.655
```

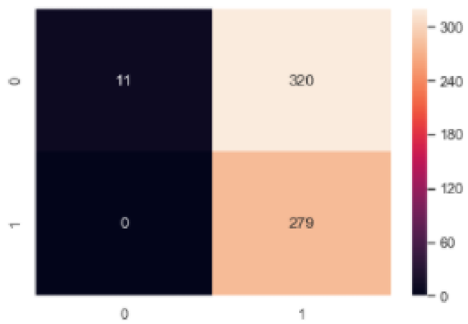
Fig 2.27 AUC and ROC curve train and test data of LDA model

Confusion Matrix, Accuracy and F1 score for different cut off value.

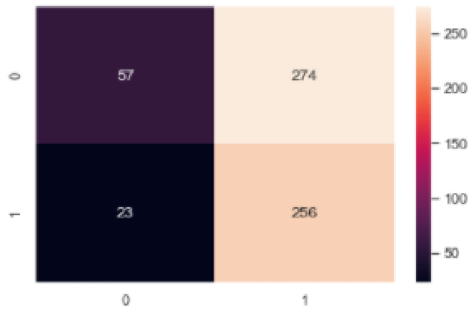
```
0.1

Accuracy Score 0.4574
F1 Score 0.6277

Confusion Matrix
```



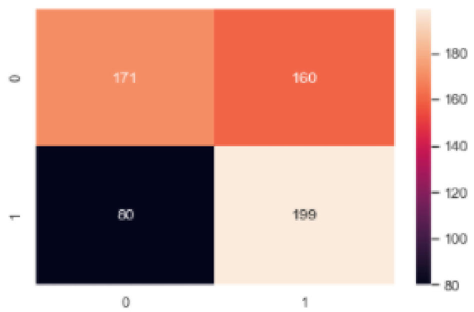

```
0.2

Accuracy Score 0.4754
F1 Score 0.6355

Confusion Matrix
```



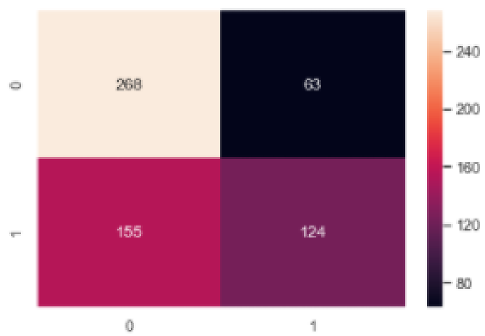
```
0.3

Accuracy Score 0.5131
F1 Score 0.6329

Confusion Matrix
```



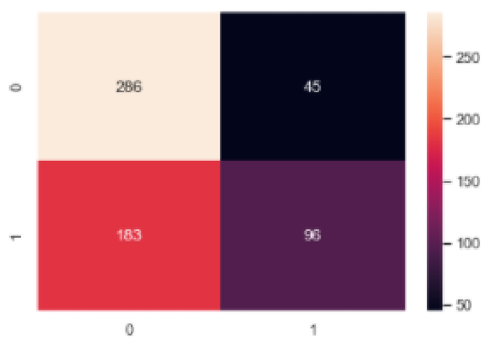

```
0.4

Accuracy Score 0.6066
F1 Score 0.6238

Confusion Matrix
```



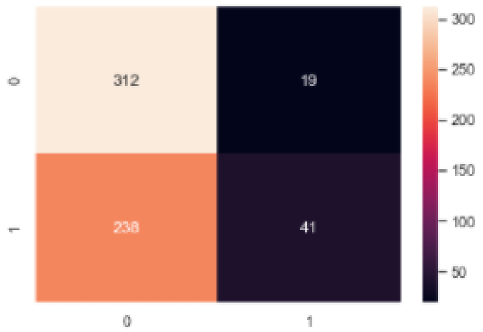

```
0.5

Accuracy Score 0.6426
F1 Score 0.5322

Confusion Matrix
```



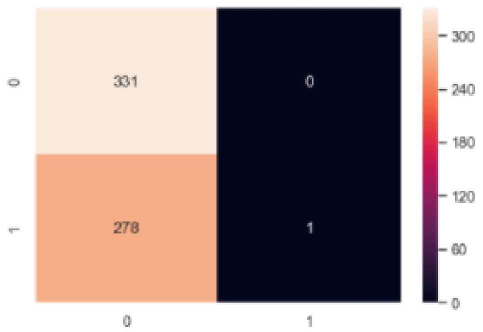

```
0.6

Accuracy Score 0.6262
F1 Score 0.4571

Confusion Matrix
```



```
0.7

Accuracy Score 0.5787
F1 Score 0.2419

Confusion Matrix
```




```
0.8

Accuracy Score 0.5443
F1 Score 0.0071

Confusion Matrix
```




```
0.9

Accuracy Score 0.5426
F1 Score 0.0

Confusion Matrix
```

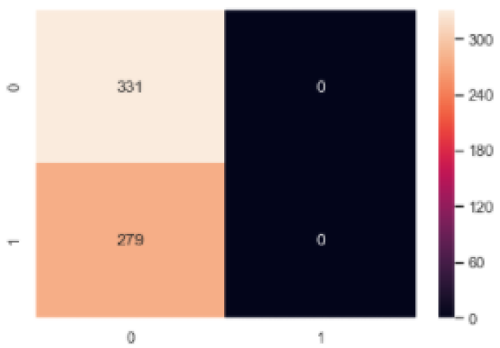


Fig 2.28 Accuracy,F1-Score and Confusion Matrix of LDA model at different cut off value

From this we can infer that, the better result comes at the cut off value of 0.2.

	LR Train	LR Test	LDA Train	LDA Test
Accuracy	0.64	0.63	0.64	0.64
AUC	0.68	0.67	0.67	0.65
Recall	0.68	0.65	0.66	0.68
Precision	0.50	0.49	0.53	0.51
F1 Score	0.40	0.39	0.44	0.41

Table 2.8 Comparing Logistic Regression and Linear Discriminant Analysis results

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

Insights:

Based on these inference from Logistic Regression(LR) and Linear Discriminant Analysis(LDA), LDA(Linear Discriminant Analysis) gives the better predictions and accurate results for both train and test data.

Recommendations:

To increase more holiday packages for the employee

- We can provide complimentary breakfast and dinner for the holiday package.
- Great deals like extra-day stay for the holiday package from the normal trip package.
- Travel package rewards for the employee performance