

Predictive Modelling Project Report

Deepti Yadav
July'22
Date : 31/07/2022

Table of Contents

Problem 1 : Linear Regression	4
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.	5
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.	19
1.3 Encode the data (having string values) for Modelling. Split the data into train and test (70:30). Apply Linear regression using scikit learn. Perform checks for significant variables using appropriate method from statsmodel. Create multiple models and check the performance of Predictions on Train and Test sets using Rsquare, RMSE & Adj Rsquare. Compare these models and select the best one with appropriate reasoning.	22
1.4 Inference: Basis on these predictions, what are the business insights and recommendations. Please explain and summarize the various steps performed in this project. There should be proper business interpretation and actionable insights present.	30
Problem 2 : Logistic Regression and LDA.....	33
2.1 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.	33
2.2 Do they 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).....	48
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.	50
2.4 Inference: Basis on these predictions, what are the insights and recommendations.	62

List of Figures

Figure 1 - Numerical/continuous variable after outlier treatment	13
Figure 2 -Count plot for categorical variables	14
Figure 3 -Box plot for categorical variables	15
Figure 4 - Pairplot.....	17
Figure 5 – Heat Map	17
Figure 6 – Point plot for categorical variables and price.....	21
Figure 7 – Scaler plot actual test price v/s predicted price	26
Figure 8 – Boxplot for salary before outlier treatment	39
Figure 9 – Boxplot for salary after outlier treatment	39
Figure 10 -Count plot for object type variables.....	40
Figure 11 -Count plot for numeric type variables	41
Figure 12 - Histogram for numerical variables	42
Figure 13 - Pairplot	43
Figure 14 – Heat Map	44
Figure 15 - Salary v/s HolidayPackage	45
Figure 16 - Age v/s HolidayPackage.....	45
Figure 17 - Educ v/s HolidayPackage	46
Figure 18 - No_young_children v/s HolidayPackage	46
Figure 19 - No_older_children v/s HolidayPackage.....	47
Figure 20 - – AUC and ROC for the training data (LR).....	50
Figure 21 – AUC and ROC for the test data (LR)	51
Figure 22 – Confusion Matrix for the training data (LR).....	51
Figure 23 – Confusion Matrix for test data (LR)	52
Figure 24 – AUC and ROC for the training & testing data (LDA).....	53
Figure 25 – Confusion Matrix for the training & testing data (LDA).....	53
Figure 26 – Confusion Matrix with custom cut off	59
Figure 27 – Confusion Matrix with custom cut off for Test data	59
Figure 28 – Confusion Matrix with custom cut off for Train data	60

List of Tables

Table 1- Dataset Description	5
Table 2 - Dataset Information.....	6
Table 3- Dataset Description (after dropping Unnamed column).....	6
Table 4 - Dataset Information (after dropping Unnamed column)	6
Table 5 – Five point summary (numerical variables).....	7
Table 6 – Five point summary (categorical variables)	7
Table 7 – Records with x, y, z =0	7
Table 8 – Checking for duplicates.....	8
Table 9 - Missing values Check	9
Table 10 – Univariate Analysis.....	10
Table 11 – Skewness Analysis	11
Table 12 – Correlation Values.....	18

Table 13 - Missing values Check after imputing	19
Table 14 – Value counts for categorical variables	19
Table 15 – Ordinal values for categorical variables.....	21
Table 16 – Data set with Ordinal values for categorical variables	22
Table 17 – Value count of Ordinal values for categorical variables	23
Table 18 – Data info with Ordinal values for categorical variables.....	23
Table 19 – OLS Regression summary	29
Table 20- Dataset Description	33
Table 21 - Dataset Information.....	34
Table 22- Dataset Description (after dropping Unnamed column).....	34
Table 23 - Dataset Information (after dropping & renaming Unnamed column)	35
Table 24 – Five point summary (numerical variables).....	35
Table 25 – Five point summary (categorical variables)	35
Table 26 - Missing values Check	36
Table 27 – Univariate Analysis.....	37
Table 28 – Skewness Analysis	38
Table 29 – Correlation Values.....	44
Table 30 – Encoding for categorical variables	48
Table 31 – Classification report for training data (LR)	52
Table 32 – Classification report for test data (LR)	52
Table 33 – Classification report for training data (LDA)	54
Table 34 – Classification report for test data (LDA).....	54
Table 35 – Classification report for test data with default and custom cut-off.....	60
Table 36 – Classification report for train data with default and custom cut-off.....	61
Table 37 – Comparison of all model	61

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.

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 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, SI1, SI2, I1
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.

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.

Table 1- Dataset Description

	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
5	6	1.02	Ideal	D	VS2	61.5	56.0	6.46	6.49	3.99	9502
6	7	1.01	Good	H	SI1	63.7	60.0	6.35	6.30	4.03	4836
7	8	0.50	Premium	E	SI1	61.5	62.0	5.09	5.06	3.12	1415
8	9	1.21	Good	H	SI1	63.8	64.0	6.72	6.63	4.26	5407
9	10	0.35	Ideal	F	VS2	60.5	57.0	4.52	4.60	2.76	706

	Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
26957	26958	2.09	Premium	H	SI2	60.6	59.0	8.27	8.22	5.00	17805
26958	26959	1.37	Premium	E	SI2	61.0	57.0	7.25	7.19	4.40	6751
26959	26960	1.05	Very Good	E	SI2	63.2	59.0	6.43	6.36	4.04	4281
26960	26961	1.10	Very Good	D	SI2	NaN	63.0	6.76	6.69	3.94	4361
26961	26962	0.25	Premium	F	VVS2	62.0	59.0	4.04	3.99	2.49	740
26962	26963	1.11	Premium	G	SI1	62.3	58.0	6.61	6.52	4.09	5408
26963	26964	0.33	Ideal	H	IF	61.9	55.0	4.44	4.42	2.74	1114
26964	26965	0.51	Premium	E	VS2	61.7	58.0	5.12	5.15	3.17	1656
26965	26966	0.27	Very Good	F	VVS2	61.8	56.0	4.19	4.20	2.60	682
26966	26967	1.25	Premium	J	SI1	62.0	58.0	6.90	6.88	4.27	5166

The dataset has Unnamed column which we are going to drop as it contains serial numbers.

Table 2 - Dataset Information

```
<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
```

Table 3- Dataset Description (after dropping Unnamed column)

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

Table 4 - Dataset Information (after dropping Unnamed column)

```
<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
```

Table 5 – Five point summary (numerical variables)

	count	mean	std	min	25%	50%	75%	max
carat	26967.0	0.798375	0.477745	0.2	0.40	0.70	1.05	4.50
depth	26270.0	61.745147	1.412860	50.8	61.00	61.80	62.50	73.60
table	26967.0	57.456080	2.232068	49.0	56.00	57.00	59.00	79.00
x	26967.0	5.729854	1.128516	0.0	4.71	5.69	6.55	10.23
y	26967.0	5.733569	1.166058	0.0	4.71	5.71	6.54	58.90
z	26967.0	3.538057	0.720624	0.0	2.90	3.52	4.04	31.80
price	26967.0	3939.518115	4024.864666	326.0	945.00	2375.00	5360.00	18818.00

Table 6 – Five point summary (categorical variables)

	cut	color	clarity
count	26967	26967	26967
unique	5	7	8
top	Ideal	G	SI1
freq	10816	5661	6571

Variables x, y & z have '0' as minimum values which is not possible as these variables are length, width & height of cubic zirconium. We need to either replace the zeros with some values or we can also drop these rows if numbers of rows are not significant. Let's find out the rows with x, y, z = 0

Table 7 – Records with x, y, z =0

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

from above we can see that only 9 entries out of 26967 (0.12 %) are 0 so we can drop these rows.

Performing EDA : We will follow the below mentioned steps to perform EDA

Step 1 :Checking & Removing duplicates, if any

Step 2: Checking a Missing value.

Step 3: Univariate Analysis with Outlier treatment

Step 4: Bivariate Analysis.

EDA-Step 1 :Checking & Removing duplicates, if any

Number of duplicate rows = 33

Table 8 – Checking for duplicates

	carat	cut	color	clarity	depth	table	x	y	z	price
4756	0.35	Premium	J	VS1	62.4	58.0	5.67	5.64	3.53	949
8144	0.33	Ideal	G	VS1	62.1	55.0	4.46	4.43	2.76	854
8919	1.52	Good	E	I1	57.3	58.0	7.53	7.42	4.28	3105
9818	0.35	Ideal	F	VS2	61.4	54.0	4.58	4.54	2.80	906
10473	0.79	Ideal	G	SI1	62.3	57.0	5.90	5.85	3.66	2898
10500	1.00	Premium	F	VVS2	60.6	54.0	6.56	6.52	3.96	8924
12894	1.21	Premium	D	SI2	62.5	57.0	6.79	6.71	4.22	6505
13547	0.43	Ideal	G	VS1	61.9	55.0	4.84	4.86	3.00	943
13783	0.79	Ideal	G	SI1	62.3	57.0	5.90	5.85	3.66	2898
14389	0.60	Premium	D	SI2	62.0	57.0	5.43	5.35	3.34	1196
14410	1.00	Very Good	D	SI1	63.1	56.0	6.34	6.30	3.99	5645
15798	0.90	Very Good	I	VS2	58.4	62.0	6.29	6.35	3.69	3334
16852	0.79	Ideal	G	SI1	62.3	57.0	5.90	5.85	3.66	2898
17263	1.04	Premium	I	SI2	62.0	57.0	6.53	6.47	4.03	3774
18025	1.51	Good	I	SI1	63.8	57.0	7.21	7.18	4.59	6046
18777	0.32	Premium	H	VS2	60.6	58.0	4.47	4.44	2.70	648
18837	1.01	Premium	H	VS1	61.2	61.0	6.44	6.41	3.93	5294
19731	0.30	Good	J	VS1	63.4	57.0	4.23	4.26	2.69	394
19877	2.01	Premium	I	VS2	60.3	62.0	8.13	8.08	4.89	15939
20301	0.30	Ideal	H	SI1	62.2	57.0	4.26	4.29	2.66	450
20760	1.80	Ideal	H	VS1	62.3	56.0	7.79	7.76	4.84	15105
22322	2.05	Premium	I	SI2	62.0	58.0	8.13	8.08	5.02	9850
22488	2.42	Premium	J	VS2	61.3	59.0	8.61	8.58	5.27	17168
22583	0.33	Ideal	F	IF	61.2	56.0	4.47	4.49	2.74	1240
23458	2.66	Good	H	SI2	63.8	57.0	8.71	8.65	5.54	16239

Looking at the duplicate rows we can conclude that they are not duplicate at all row, so we decide not to drop them.

EDA-Step 2: Checking Missing value.

Table 9 - Missing values Check

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

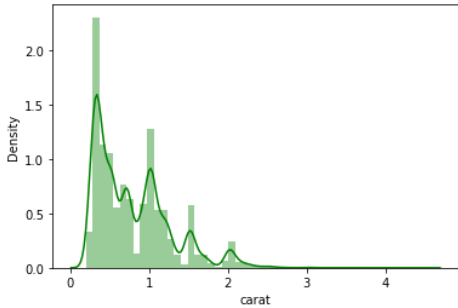
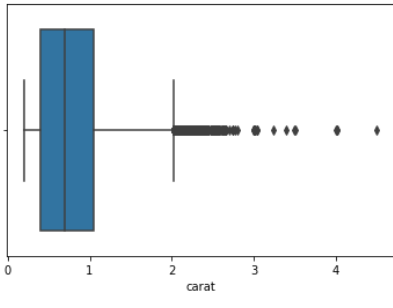
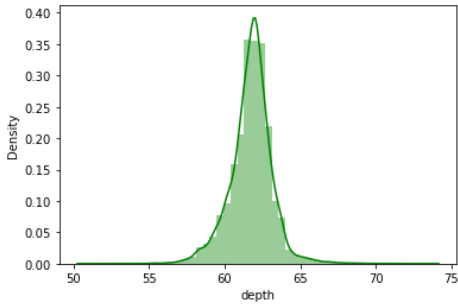
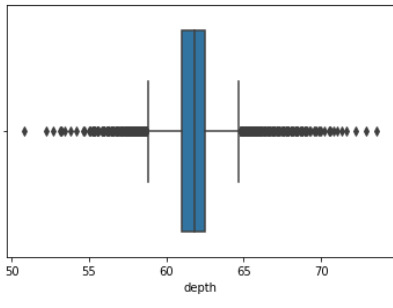
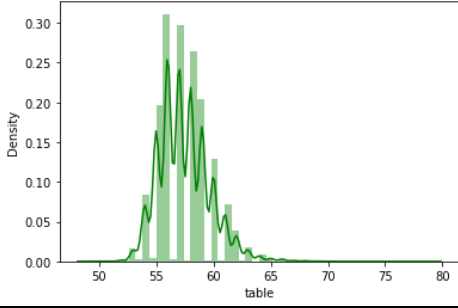
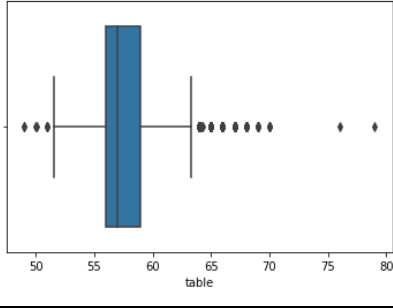
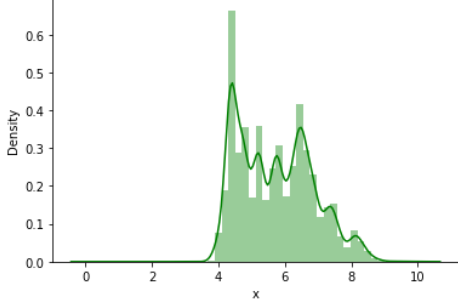
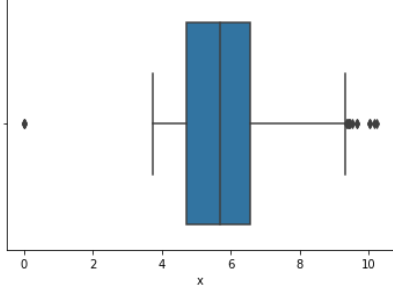
Observations:

- There are 10 variables (1 variable unnamed dropped) and 26967 records.
- There are 26958 records after dropping some rows for x, y & z.
- The variables 'carat', 'depth', 'table', 'x', 'y', 'z', are numeric type.
- The variables 'cut', 'color' and 'clarity' are object type.
- The variable 'Price' is target variable and others are predictor variables
- Depth has total of 697 missing values
- There are 34 duplicate rows but looking at the row for every column entry, it is found that all columns are not duplicate. So, dropping duplicate rows is not recommended.

EDA Step 3 :Univariate Analysis with Outlier treatment

- Check five-point summary for continuous variables
- Check distribution of variables
- Check outliers
-

Table 10 – Univariate Analysis

Description	Distribution plot	Boxplot
<p>Description of carat</p> <hr/> <p>count 26967.000000 mean 0.798375 std 0.477745 min 0.200000 25% 0.400000 50% 0.700000 75% 1.050000 max 4.500000 Name: carat, dtype: float64 Distribution of carat</p> <hr/>		
<p>Description of depth</p> <hr/> <p>count 26270.000000 mean 61.745147 std 1.412860 min 50.800000 25% 61.000000 50% 61.800000 75% 62.500000 max 73.600000 Name: depth, dtype: float64 Distribution of depth</p> <hr/>		
<p>Description of table</p> <hr/> <p>count 26967.000000 mean 57.456080 std 2.232068 min 49.000000 25% 56.000000 50% 57.000000 75% 59.000000 max 79.000000 Name: table, dtype: float64 Distribution of table</p> <hr/>		
<p>Description of x</p> <hr/> <p>count 26967.000000 mean 5.729854 std 1.128516 min 0.000000 25% 4.710000 50% 5.690000 75% 6.550000 max 10.230000 Name: x, dtype: float64 Distribution of x</p> <hr/>		

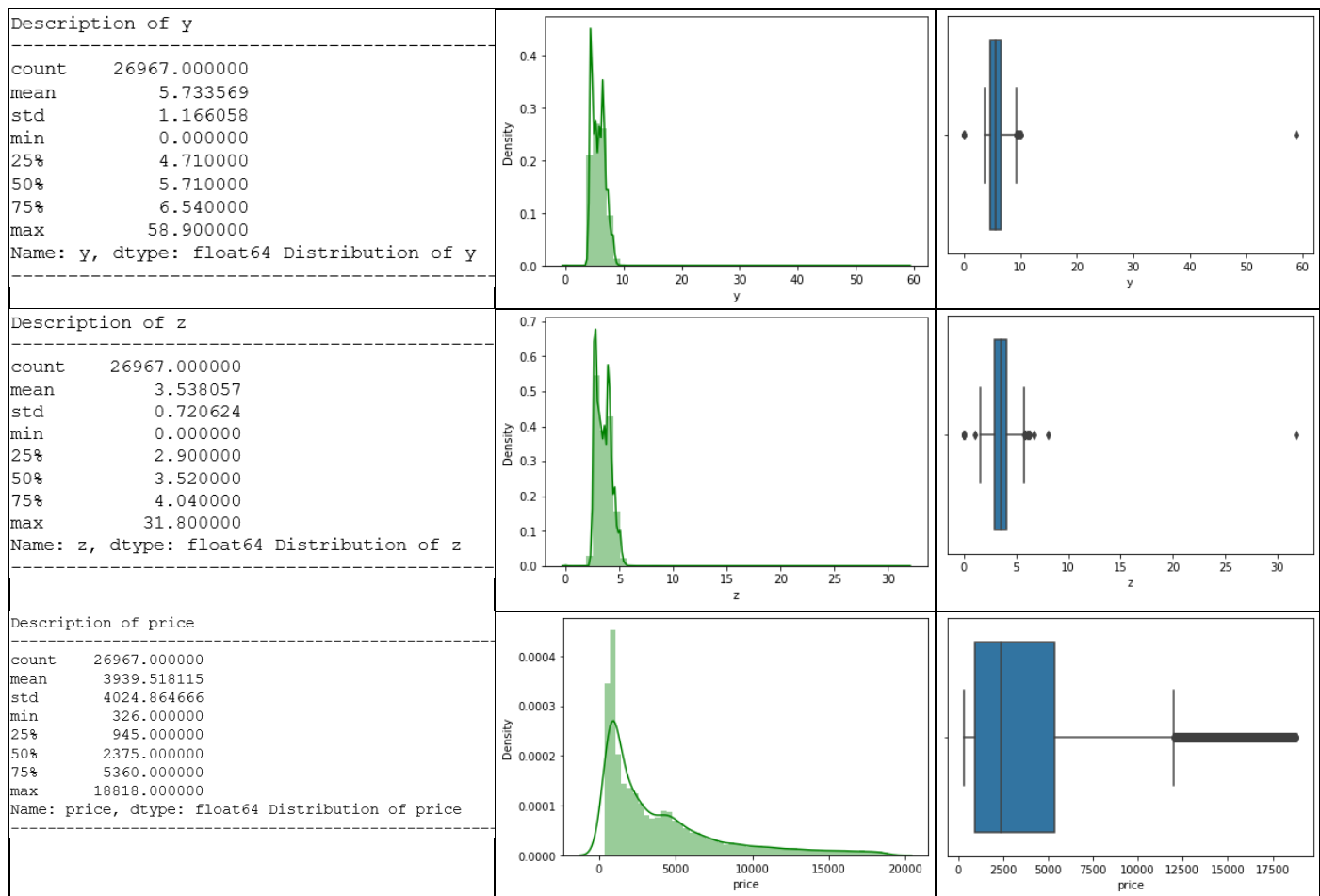


Table 11 – Skewness Analysis

```

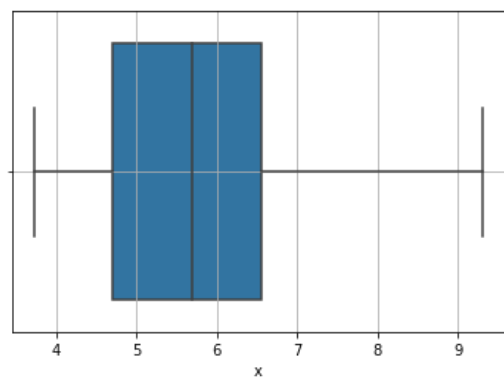
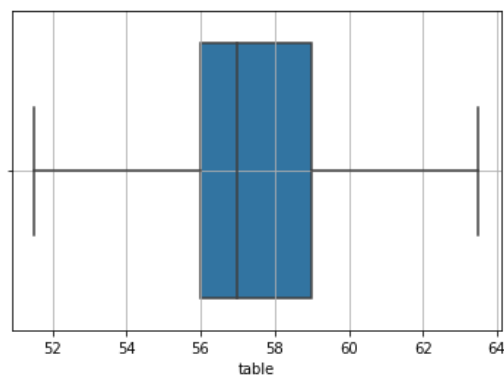
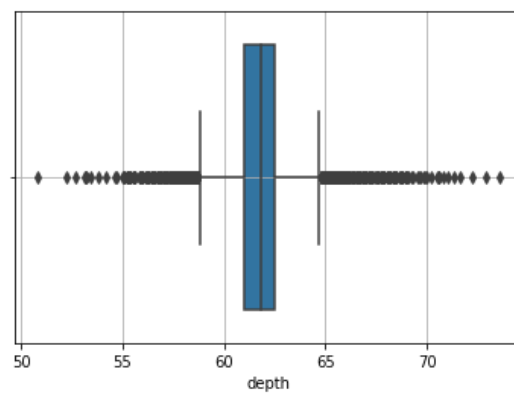
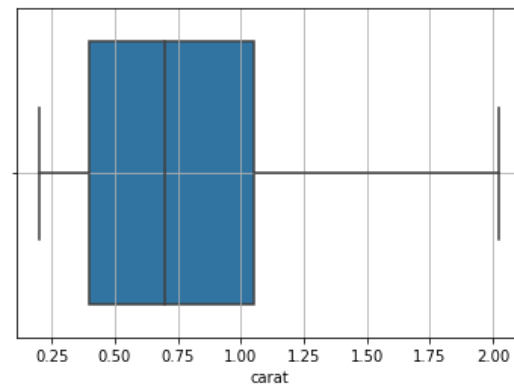
carat    1.116530
depth    -0.027571
table     0.764957
x         0.402531
y         3.879939
z         2.634182
price     1.618432
dtype: float64

```

Observations:

- All the variables except depth (left skewed) are not normally distributed and right skewed.
- Outliers are present in all the numerical/continuous variables. There is significant amount of outlier present in some variable, the features with datapoint that are far from the rest of dataset which will affect the outcome of our regression model. So, we have treated the outlier.
- Looking at the modes in distributed, there could be some clusters present in the variables.

Removing Outliers



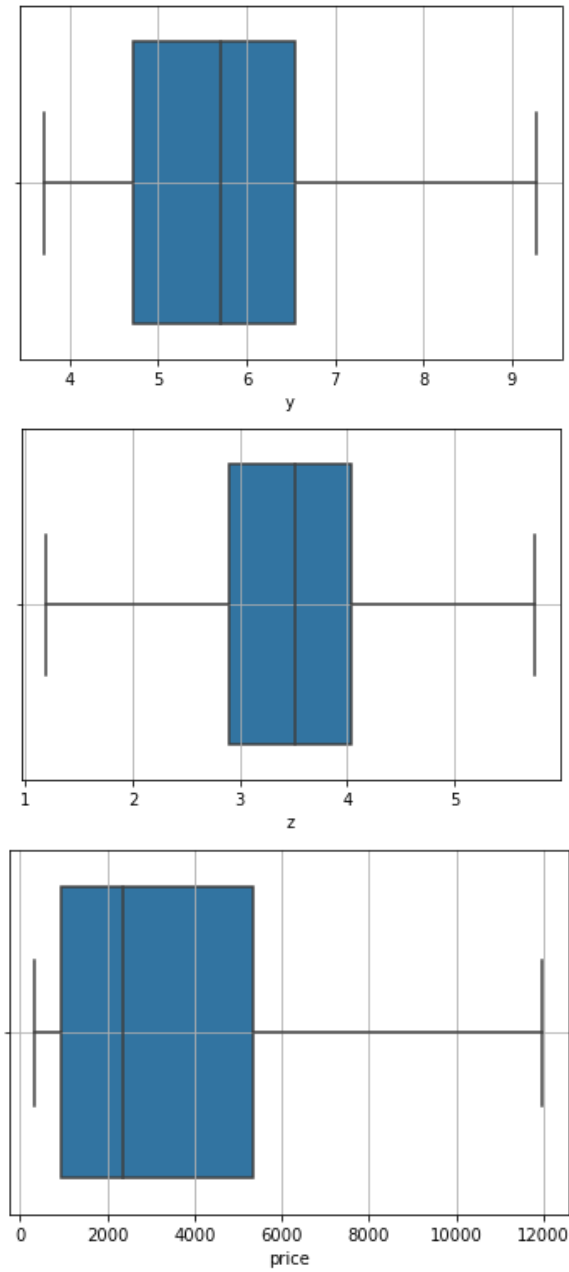


Figure 1 - Numerical/continuous variable after outlier treatment

Except depth outliers have been treated

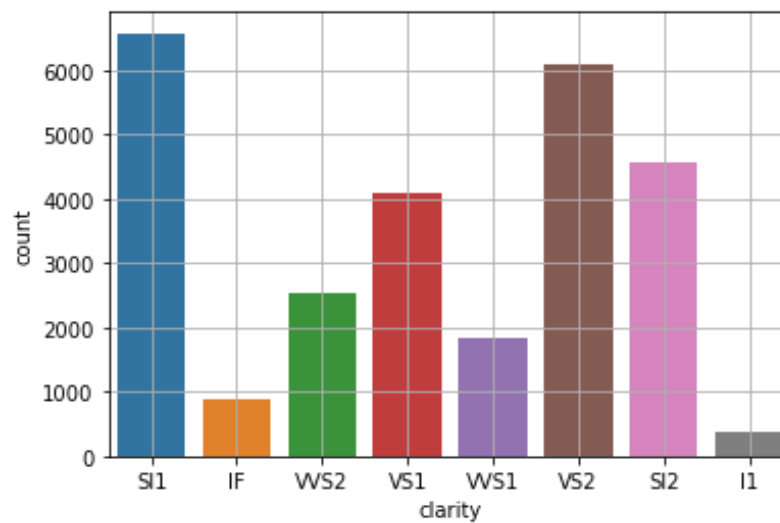
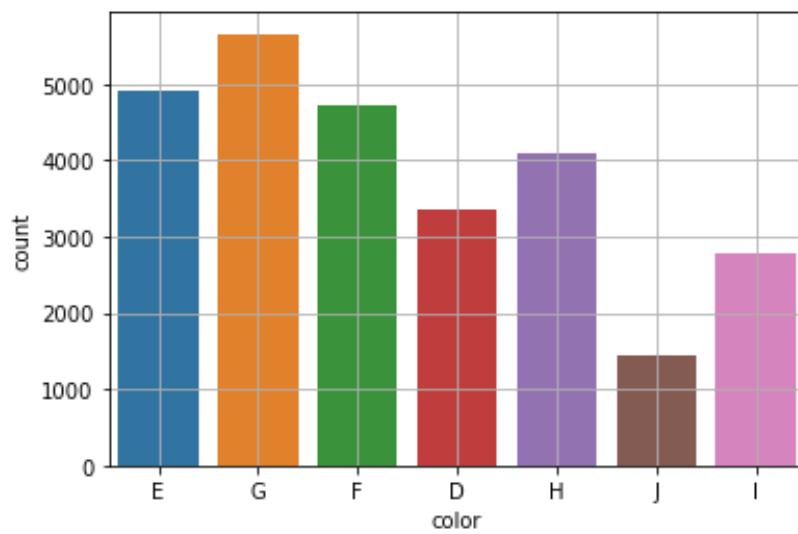
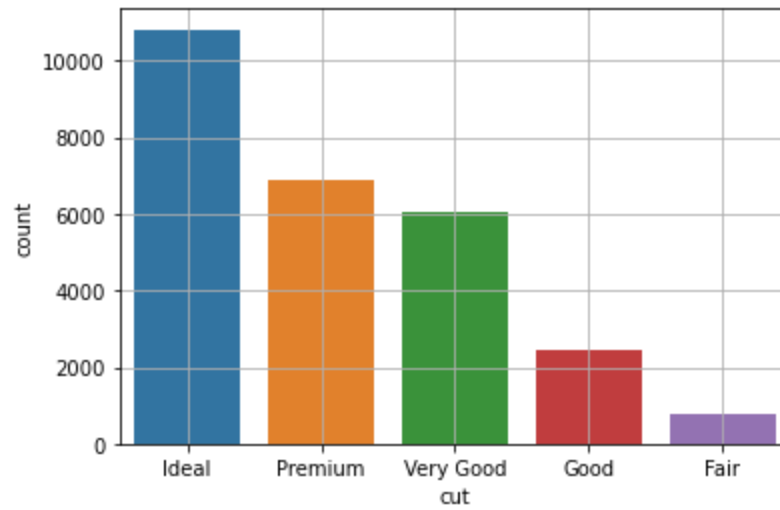


Figure 2 -Count plot for categorical variables

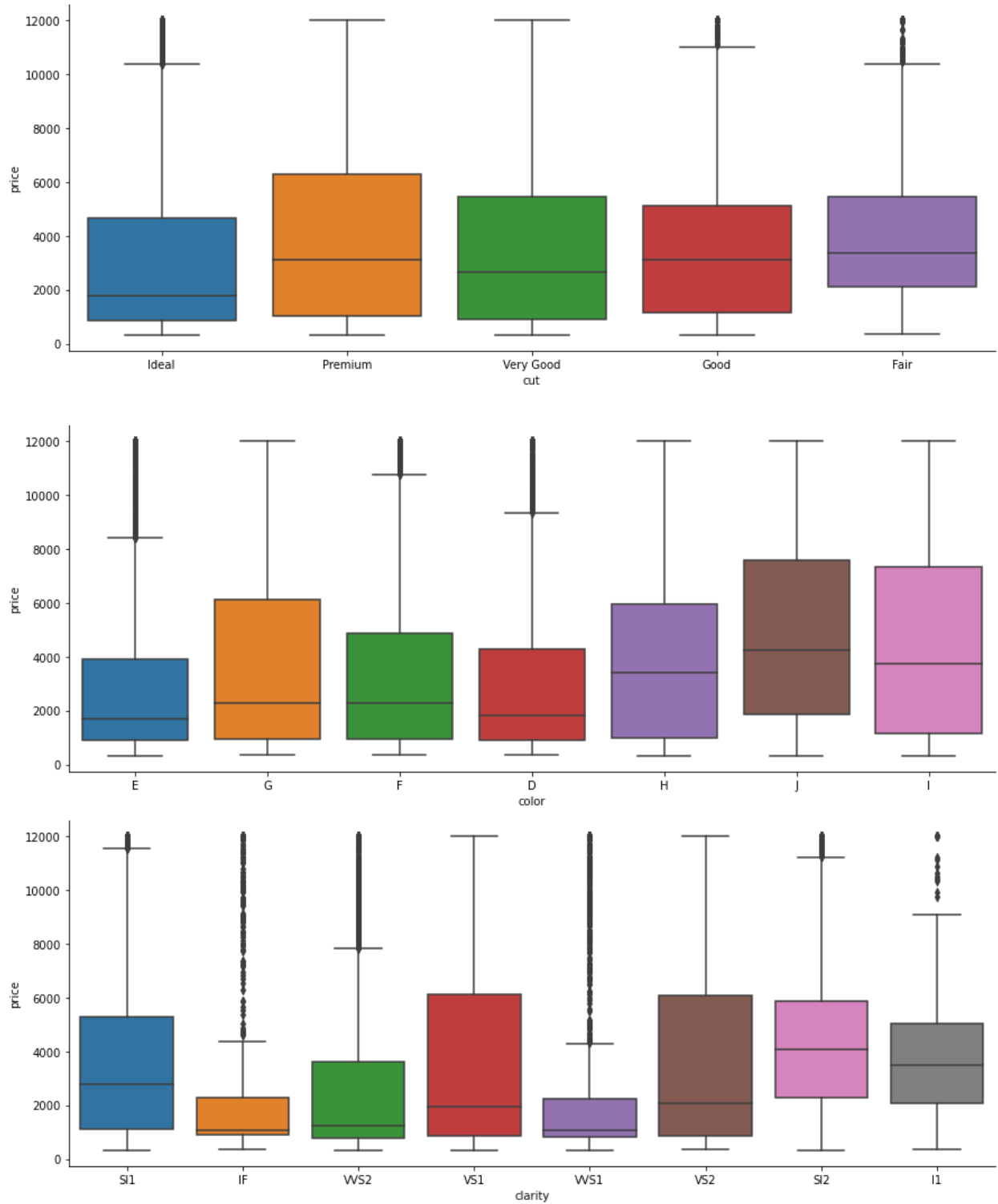


Figure 3 -Box plot for categorical variables

The independent attributes have different units and scales of measurement

Observations:

- Maximum no of zirconium is of Ideal cut while zirconium with fair cut is least.
- Maximum no of zirconium is of G color while zirconium with J color is least.
- Maximum no of zirconium is of SI1 clarity while zirconium with I1 clarity is least.

EDA- Step 4: Multivariate Analysis

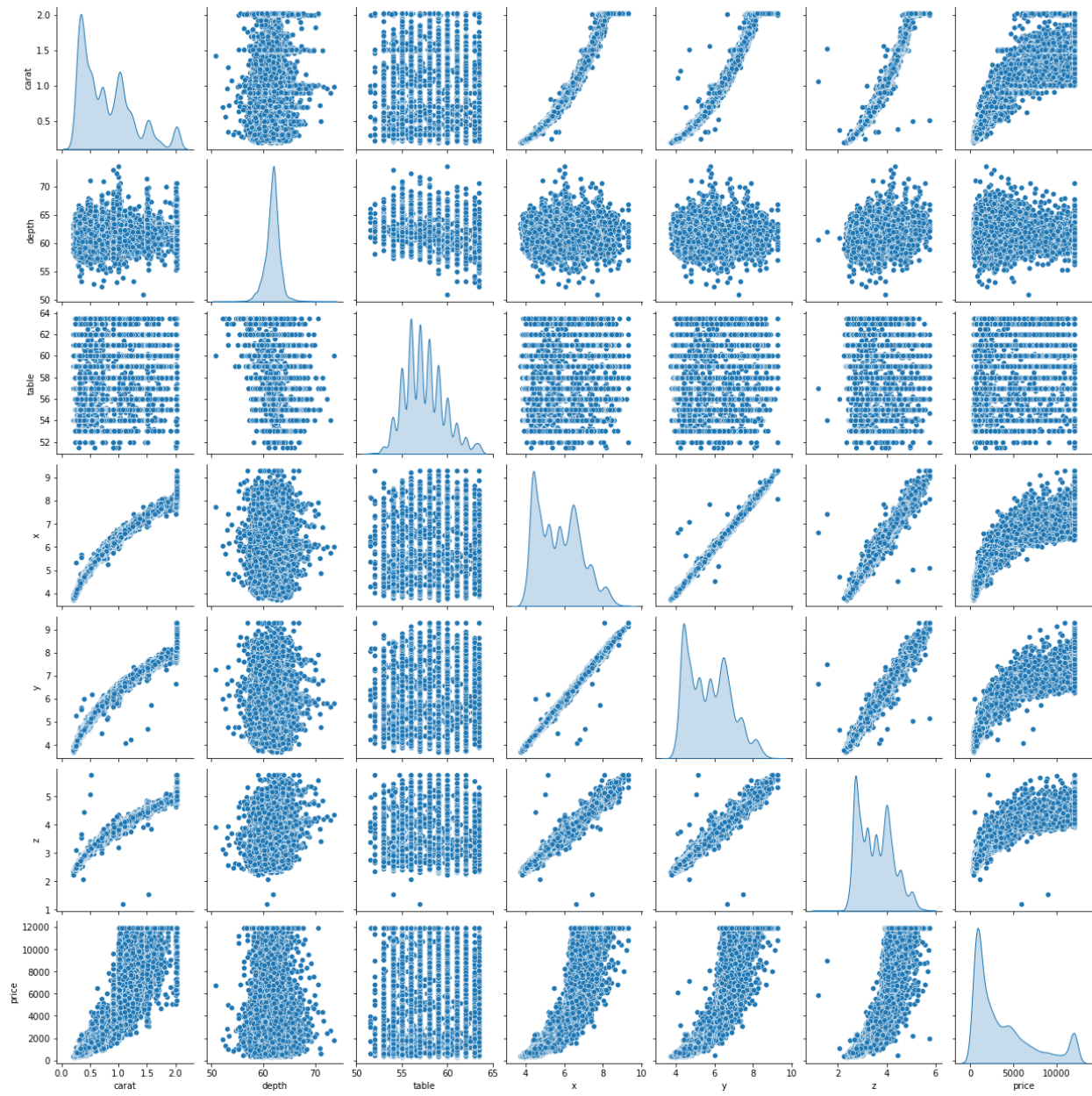


Figure 4 - Pairplot

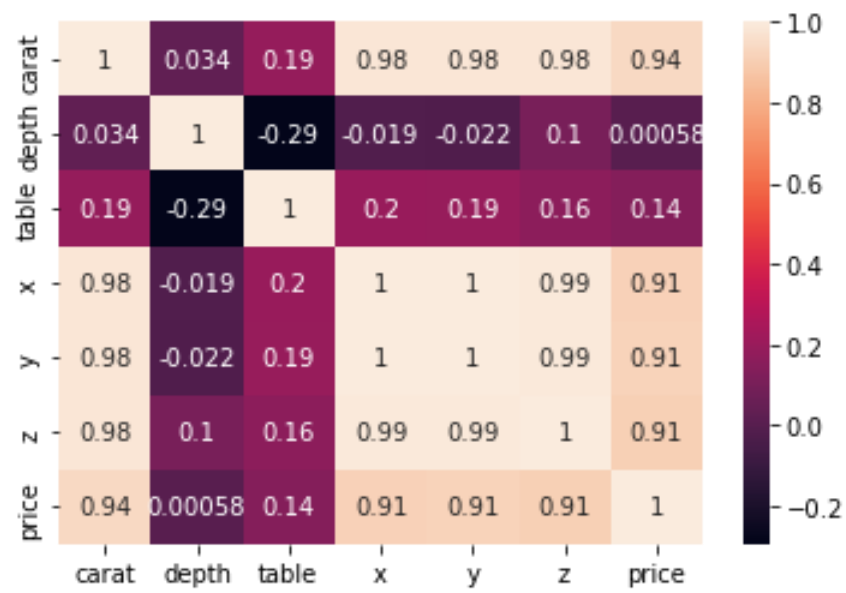


Figure 5 – Heat Map

Table 12 – Correlation Values

correlation		
y	x	0.998492
z	x	0.990898
	y	0.990535
x	carat	0.982882
	y	0.981964
carat	z	0.980877
	y	0.914793
price	carat	0.936743
	y	0.913356
x	price	0.913356
price	z	0.908588
	depth	0.294291
table	depth	0.294291
	x	0.199932
carat	y	0.194311
	table	0.187400
table	z	0.160748
price	table	0.138027
	depth	0.101463
depth	carat	0.033644
	y	0.022265
price	x	0.018593
	depth	0.000585

Observation

- Depth have almost zero correlation with price, so it is a weak predictor.
- Table is also a weak predictor, but we will check the coefficient and p value in further sections.
- There is strong correlation between independent variables x, y, z, & carat which means presence of multicollinearity.
- There is strong correlation between price and predictors carat, x, y & z hence they are good predictor
- Independent variables table & x, y, z, carat have weak correlation between each other which is good for model.
- Independent variables depth & x, y, z, carat have weak correlation between each other which is good for model.

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.

Imputing missing values

As we have seen that only depth has missing values which is a continuous variable. We also found outliers in depth so we will computer the missing values with median.

Table 13 - Missing values Check after imputing

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

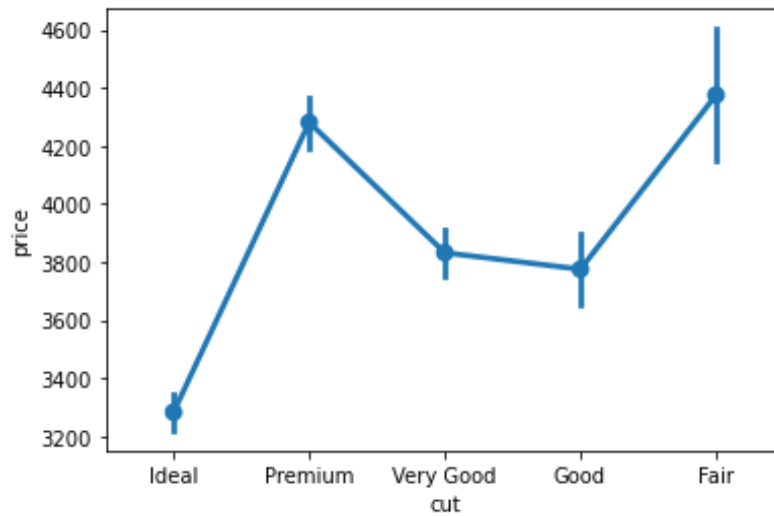
From above we can see that there are no missing values after imputing them

In Qs.1.1 we have already check for 'Zero' value for variable 'x', 'y', 'z' which indicated that they are faulty values. As we know dimensionless or 2-dimensional diamonds are not possible. So, we have filter out those as it clearly faulty data entries. We have imputed them.

Table 14 – Value counts for categorical variables

```
Ideal      10816
Premium    6893
Very Good  6030
Good       2439
Fair       780
Name: cut, dtype: int64
```

```
SI1      6570
VS2      6098
SI2      4571
VS1      4092
VVS2      2531
VVS1     1839
IF        894
I1        363
Name: clarity, dtype: int64
G        5658
E        4917
F        4727
H        4098
D        3344
I        2771
J        1443
Name: color, dtype: int64
```



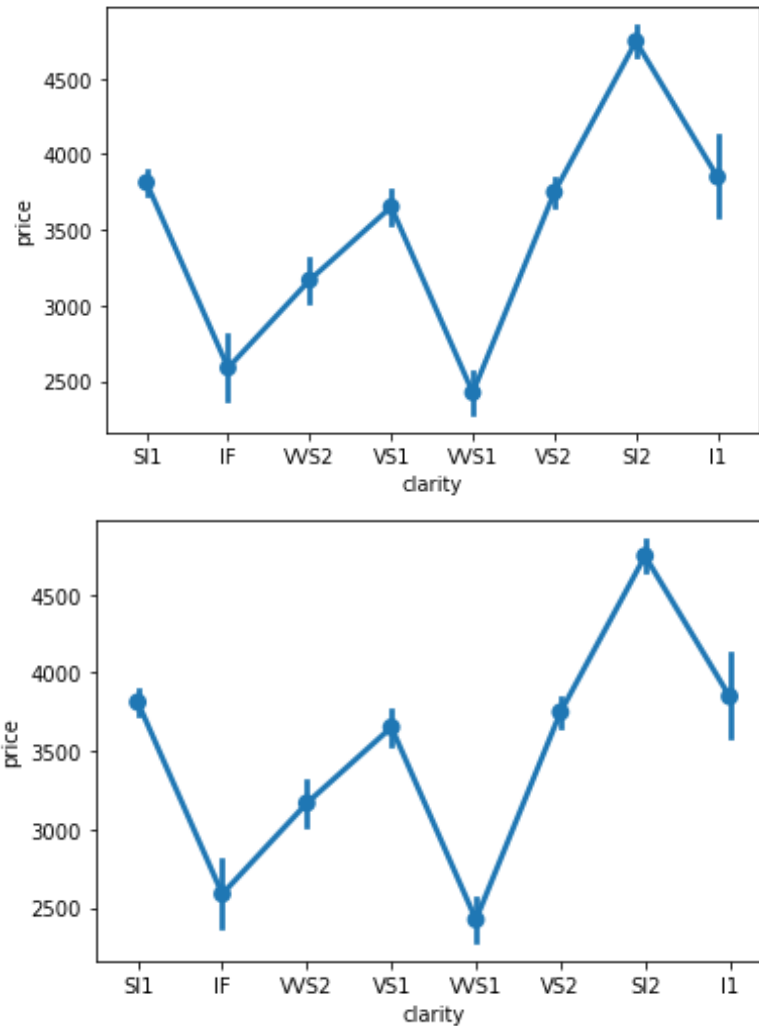


Figure 6 – Point plot for categorical variables and price

In model chosen here is by combining ordinal variable subcategories. As seen in figure 6, which is point plots for different categorical variables with respect to price. The categories which have similar price and are neighbors in respect of the quality are clubbed together as shown below

Table 15 – Ordinal values for categorical variables

Category	subcategory	Ordinal values
Cut	Fair	0
	Good	1
	Very Good	2
	Premium	2
	Ideal	3

Color	D	3
	E	3
	F	2
	G	2
	H	1
	I	0
	J	0
Clarity	IF	4
	VVS1	4
	VVS2	3
	VS1	2
	VS2	2
	SI1	2
	SI2	1
	I1	0

1.3 Encode the data (having string values) for Modelling. Split the data into train and test (70:30). Apply Linear regression using scikit learn. Perform checks for significant variables using appropriate method from statsmodel. Create multiple models and check the performance of Predictions on Train and Test sets using Rsquare, RMSE & Adj Rsquare. Compare these models and select the best one with appropriate reasoning.

As we have seen that cut, clarity and color are object, so we must convert them into categorical. As these are ordinal values, we encode the variables in a way that will highlight the rank order that is mentioned in the problem statement.

Table 16 – Data set with Ordinal values for categorical variables

	carat	cut	color	clarity	depth	table	x	y	z	price
0	0.30	3	3	2	62.1	58.0	4.27	4.29	2.66	499.0
1	0.33	2	2	4	60.8	58.0	4.42	4.46	2.70	984.0
2	0.90	2	3	3	62.2	60.0	6.04	6.12	3.78	6289.0
3	0.42	3	2	2	61.6	56.0	4.82	4.80	2.96	1082.0
4	0.31	3	2	4	60.4	59.0	4.35	4.43	2.65	779.0

Table 17 – Value count of Ordinal values for categorical variables

```
CUT : 4
0      780
1     2439
3     10816
2     12923
Name: cut, dtype: int64
```

```
COLOR : 4
1      4098
0      4214
3      8261
2     10385
Name: color, dtype: int64
```

```
CLARITY : 5
0        363
3       2531
4       2733
1       4571
2      16760
Name: clarity, dtype: int64
```

Table 18 – Data info with Ordinal values for categorical variables

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 26958 entries, 0 to 26966
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   carat        26958 non-null  float64
1   cut          26958 non-null  int64
2   color        26958 non-null  int64
3   clarity      26958 non-null  int64
4   depth        26958 non-null  float64
5   table        26958 non-null  float64
6   x            26958 non-null  float64
7   y            26958 non-null  float64
8   z            26958 non-null  float64
9   price        26958 non-null  float64
dtypes: float64(7), int64(3)
memory usage: 3.3 MB
```

Now all variables are numeric, and we can now proceed with model buliding

Splitting data into training and test set in 30% test data

```
X_train (18870, 9)
X_test (8088, 9)
y_train (18870, 1)
y_test (8088, 1)
Total Obs 26958
```

Linear Regression Model using scikit learn

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

```
▼ LinearRegression
LinearRegression()
```

```
The coefficient for carat is 9116.76188303201
The coefficient for cut is 168.85031927677127
The coefficient for color is 430.65947159563933
The coefficient for clarity is 755.8531686184117
The coefficient for depth is -1.0837125878658247
The coefficient for table is -20.73534747619374
The coefficient for x is -1013.6570242786354
The coefficient for y is 1056.0377146915669
The coefficient for z is -893.4699106842887
```

$Y = mx + c$ ($m = m_1, m_2, m_3 \dots m_9$) here 9 different co-efficient will align with the intercept which is "c" from the model.

From the above coefficients for each of the independent attributes we can conclude

The one unit increase in carat increases price by 9116.76.

The one unit increase in cut increases price by 168.85.

The one unit increase in color increases price by 430.66.

The one unit increase in clarity increases price by 755.85.

The one unit increase in y increases price by 1056.04

The one unit increase in depth decreases price by -1.08,

The one unit increase in table decreases price by -20.73

The one unit increase in x decreases price by -1013.66,

The one unit increase in z decreases price by -893.47.

The intercept for our model is -2077.02

```
# R square on training data  
regression_model.score(X_train, y_train)
```

0.9275605988850442

```
# R square on testing data  
# Model score - R2 or coeff of determinant  
# R^2=1-RSS / TSS = RegErr / TSS  
regression_model.score(X_test, y_test)
```

0.9225449139305559

```
#RMSE on Training data  
predicted_train=regression_model.fit(X_train, y_train).predict(X_train)  
np.sqrt(metrics.mean_squared_error(y_train,predicted_train))
```

929.9311746321381

```
#RMSE on Testing data  
predicted_test=regression_model.fit(X_train, y_train).predict(X_test)  
np.sqrt(metrics.mean_squared_error(y_test,predicted_test))
```

974.7665420050732

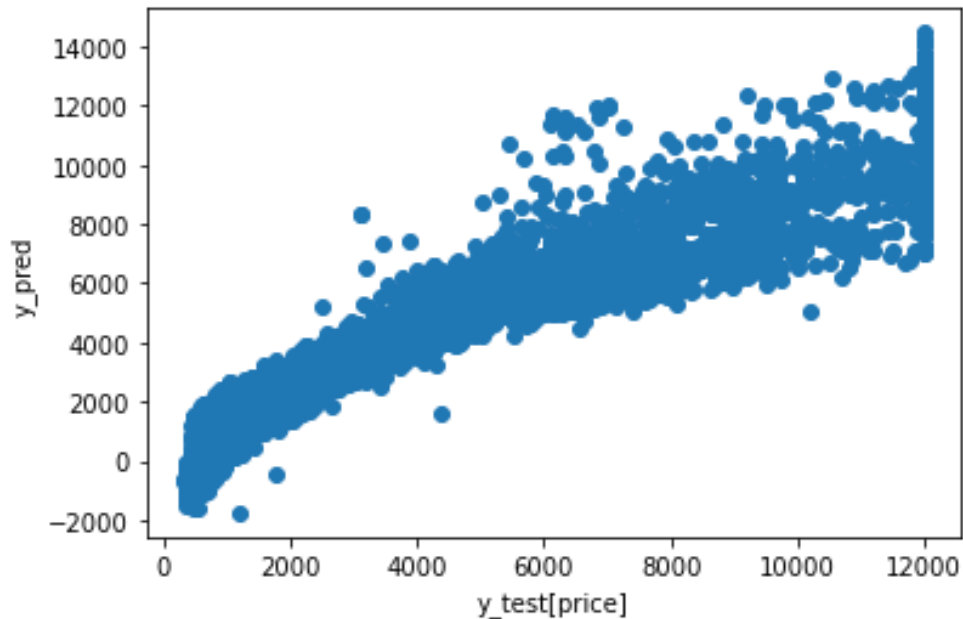


Figure 7 – Scaler plot actual test price v/s predicted price

we can see that the is a linear plot, very strong correlation between the predicted y and actual y. But there are lots of spread. That indicated some kind noise present on the data set i.e. Unexplained variances on the output.

R-square is the percentage of the response variable variation that is explained by a linear model.

R-square = Explained variation / Total variation

R-squared is always between 0 and 100%. 0% indicates that the model explains none of the variability of the response data around its mean. 100% indicates that the model explains all the variability of the response data around its mean. In this regression model we can see the R-square value on Training and Test data respectively 0.927 and 0.923.

RMSE on Training data: 929.93

RMSE on Testing data: 974.77

As the training data & testing data score are almost inline, we can conclude this model is a Right-Fit Model.

The independent attributes have different units and scales of measurement
It is always a good practice to scale all the dimensions using z scores or some other method to address the problem of different scales

Scaling or standardizing the features around the center and 0 with a standard deviation of 1 is important when we compare measurements that have different units. Variables that are measured at different scales do not contribute equally to the analysis and might end up creating a bias.

For example, A variable that ranges between 0 and 1000 will outweigh a variable that ranges between 0 and 1. Using these variables without standardization will give the variable with the larger range weight of 1000 in the analysis. Transforming the data to comparable scales can prevent this problem.

In this data set we can see the all the variable are in different scale i.e., prices are in 1000s unit and depth and table are in 100s unit, and carat is in 10s. So, it's necessary to scale or standardize the data to allow each variable to be compared on a common scale. With data measured in different "units" or on different scales (as here with different means and variances) this is an important data processing step if the results are to be meaningful or not dominated by the variables that have large variances.

But is scaling necessary in this case?

No, it is not necessary, we'll get an equivalent solution whether we apply linear scaling or not. But recommended for regression techniques as well because it would help gradient descent to converge fast and reach the global minima. When number of features becomes large, it helps in running model quickly else the starting point would be very far from minima, if the scaling is not done in preprocessing.

For now, we will process the with scaling and check the output with scaled data of regression model output.

```
The coefficient for carat is 1.2130676087280998
The coefficient for cut is 0.03593146787987769
The coefficient for color is 0.12819490444927809
The coefficient for clarity is 0.18433941202235374
The coefficient for depth is -0.0004328008119489717
The coefficient for table is -0.012885380988889755
The coefficient for x is -0.3290473056912069
The coefficient for y is 0.34037981320184413
The coefficient for z is -0.17933779191549232
```

The intercept for our model is -8.34093657044169e-17

```
regression_model.score(X_train_scaled, y_train_scaled)

0.9275605988850442
```

```
regression_model.score(X_test_scaled, y_test_scaled)

0.9226391269334447
```

Now we can observe by applying z score the intercept became $-8.34093657044169e-17$ Which is almost zero. Earlier it was -2077.02 . The co-efficient has changed, the bias became nearly zero **but the overall accuracy still same.**

VIF to check multicollinearity

```
carat ---> 122.02802455329707
cut ---> 14.543151811484508
color ---> 4.677205239193653
clarity ---> 8.681296622855571
depth ---> 1282.4795746977334
table ---> 917.9055641527887
x ---> 10414.26841167036
y ---> 9856.462752435658
z ---> 3752.1226365311113
```

We can observe there are very strong multi collinearity present in the data set. Ideally it should be within 1 to 5.

We are exploring the Linear Regression using stats models as we are interested in some more statistical metrics of the model.

R^2 is not a reliable metric as it always increases with addition of more attributes even if the attributes have no influence on the predicted variable.

Instead, we use adjusted R^2 which removes the statistical chance that improves R^2 Scikit does not provide a facility for adjusted R^2 ... so we use stats model, a library that gives results like what you obtain in R language

This library expects the X and Y to be given in one single data frame.

Linear Regression using statsmodels

expr= 'price ~ carat + cut + color + clarity + depth + table + x + y + z'

```
import statsmodels.formula.api as smf
lml = smf.ols(formula= expr, data = data_train).fit()
lml.params
```

```
Intercept    -2077.018671
carat         9116.761883
cut           168.850319
color         430.659472
clarity       755.853169
depth         -1.083713
table        -20.735347
x            -1013.657024
y             1056.037715
z            -893.469911
dtype: float64
```

Table 19 – OLS Regression summary

OLS Regression Results						
=====						
Dep. Variable:	price		R-squared:	0.928		
Model:	OLS		Adj. R-squared:	0.928		
Method:	Least Squares		F-statistic:	2.683e+04		
Date:	Sun, 31 Jul 2022		Prob (F-statistic):	0.00		
Time:	02:15:44		Log-Likelihood:	-1.5575e+05		
No. Observations:	18870		AIC:	3.115e+05		
Df Residuals:	18860		BIC:	3.116e+05		
Df Model:	9					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]

Intercept	-2077.0187	823.753	-2.521	0.012	-3691.648	-462.389
carat	9116.7619	84.580	107.789	0.000	8950.977	9282.546
cut	168.8503	11.561	14.606	0.000	146.190	191.510
color	430.6595	6.940	62.054	0.000	417.056	444.263
clarity	755.8532	8.839	85.512	0.000	738.528	773.179
depth	-1.0837	11.441	-0.095	0.925	-23.509	21.341
table	-20.7353	4.087	-5.073	0.000	-28.746	-12.724
x	-1013.6570	121.226	-8.362	0.000	-1251.272	-776.042
y	1056.0377	124.355	8.492	0.000	812.291	1299.785
z	-893.4699	167.083	-5.347	0.000	-1220.968	-565.971
=====						
Omnibus:	3096.000		Durbin-Watson:	1.987		
Prob(Omnibus):	0.000		Jarque-Bera (JB):	9473.053		
Skew:	0.855		Prob(JB):	0.00		
Kurtosis:	6.021		Cond. No.	1.05e+04		
=====						

Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 1.05e+04. This might indicate that there are strong multicollinearity or other numerical problems.

Hypothesis Testing for Linear Regression

The big question one needs to ask is, are these coefficients really reflecting the relation between the targets variable and the independent variable or are they by statistical chance.

To establish the reliability of the coefficients, we need hypothesis testing

The Null hypothesis (H_0) claims there is no relation between price and any of the variables. That means the coefficient is 0 in the universe

Assuming H_0 to be true, what is the probability of finding the coefficients found in the sample if the sample is drawn from that universe in which H_0 is true

At 95% confidence level

- if the p value is $< .05$, we reject the H_0 i.e., probability of finding these coefficients in sample if they are 0 in the universe is very low

- If p value is $\geq .05$, we do not have sufficient evidence in the data to reject the H_0 and hence we do not reject H_0 . We believe H_0 is likely to be true in the universe

P is the conditional probability given H_0 is true

From statsmodel summary we can see the p value is showing 0.925 for 'depth' variable, which is much higher than 0.05. That means this dimension is useless. So we can say that the attribute which are having p value greater than 0.05 are poor predictor for price

Overall, model P value is lower than 0.05 which means model is reliable after eliminating the useless attributes that is depth

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

The study was performed by Cleaning and preprocessing the data. Then Exploratory Data Analysis (EDA) was performed to get some insight into data. We build a linear regression with 'sklearn' as well as 'statsmodels'. Different techniques like scaling, different combinations of variables encoding techniques were used to check and improve the performance. We checked for multicollinearity with vif and found that multicollinearity is present which is strong. We also checked which variables have no significant correlation with target value and found depth variable to be the one.

Then Evaluate was evaluated by checking the performance of Predictions on Train and Test sets using Rsquare, RMSE & Adj Rsquare. We saw that the accuracy, Rsquare, RMSE of model with 'sklearn' is same as 'statsmodels'.

Scaling : co-efficient has changed, the bias became nearly zero but the overall accuracy still same

Inference:

The final Linear Regression equation is

$$\text{price} = (-2077.02) * \text{Intercept} + (9116.76) * \text{carat} + (168.85) * \text{cut} + (430.66) * \text{color} + (755.85) * \text{clarity} + (-1.08) * \text{depth} + (-20.74) * \text{table} + (-1013.66) * x + (1056.04) * y + (-893.47) * z +$$

From the above coefficients for each of the independent attributes we can conclude

The one unit increase in carat increases price by 9116.76.

The one unit increase in cut increases price by 168.85.

The one unit increase in color increases price by 430.66.

The one unit increase in clarity increases price by 755.85.

The one unit increase in y increases price by 1056.04

The one unit increase in depth decreases price by -1.08,

The one unit increase in table decreases price by -20.73

The one unit increase in x decreases price by -1013.66,

The one unit increase in z decreases price by -893.47.

There negative co-efficient values implies that variables inversely proportional with diamond price.

92.7% of the variation in the price is explained by the predictors in the model for train set

92.2% of the variation in the price is explained by the predictors in the model for test set

R-squared:0.933 and Adj. R-squared: 0.933 are same. The overall P value is less than alpha.

As the training data & testing data score are almost inline, we can conclude this model is a Right-Fit Model.

Finally, we can conclude that Best 5 attributes that are most important are 'Carat', 'Cut', 'color', 'clarity' and width i.e. 'y' for predicting the price.

Recommendations:

- 1) The Gemstone company should consider the features 'carat', 'cut', 'color', 'clarity' and width i.e., 'y' as most important for predicting the price.
- 2) As we can see from the model Higher the width('y') of the stone is higher the price. So, the stones having higher width('y') should consider in higher profitable stones.
- 3) The 'Premium Cut' on Diamonds are the most Expensive, followed by 'Very Good' Cut, these should consider in higher profitable stones.
- 4) The Diamonds clarity with 'VS1' & 'VS2' are the most Expensive. So, these two categories also consider in higher profitable stones.
- 5) As we see for 'x' i.e., Length of the stone, higher the length of the stone is lower the price. The p value is less than 0.05, so can conclude that as higher the length of the stone is a lower profitable stone.
- 6) Similarly, for the 'z' variable having negative co-efficient i.e., -905.38. And the p value is less than 0.05, so we can conclude that as higher the 'z' of the stone is a lower profitable stone.
- 7) Also, we can see the 'y' width in mm having positive co-efficient. And the p value is less than 0.05, so we can conclude that higher the width of the stone is a higher profitable stone.

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.

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

2.1 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.

Table 20- Dataset Description

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

	Unnamed: 0	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
867	868	no	40030	24	4	2	1	yes
868	869	yes	32137	48	8	0	0	yes
869	870	no	25178	24	6	2	0	yes
870	871	yes	55958	41	10	0	1	yes
871	872	no	74659	51	10	0	0	yes

The dataset has Unnamed column which we are going to drop as it contains serial numbers.

Table 21 - Dataset Information

```
<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
```

Table 22- Dataset Description (after dropping Unnamed column)

	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

Table 23 - Dataset Information (after dropping & renaming Unnamed column)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 872 entries, 0 to 871
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   HolidayPackage         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
```

Observation:

- In the above table we can see that column names have been corrected/Renamed.
- There are no missing values. 5 variables are numeric and remaining categorical.
- Categorical variables are not in encoded format.
-

Table 24 – Five point summary (numerical variables)

	count	mean	std	min	25%	50%	75%	max
Salary	872.0	47729.172018	23418.668531	1322.0	35324.0	41903.5	53469.5	236961.0
Age	872.0	39.955275	10.551675	20.0	32.0	39.0	48.0	62.0
Educ	872.0	9.307339	3.036259	1.0	8.0	9.0	12.0	21.0
No_young_children	872.0	0.311927	0.612870	0.0	0.0	0.0	0.0	3.0
No_older_children	872.0	0.982798	1.086786	0.0	0.0	1.0	2.0	6.0

Table 25 – Five-point summary (categorical variables)

	HolidayPackage	foreign
count	872	872
unique	2	2
top	no	no
freq	471	656

Performing EDA : We will follow the below mentioned steps to perform EDA

Step 1 :Checking & Removing duplicates, if any

Step 2: Checking a Missing value.

Step 3: Univariate Analysis with Outlier treatment

Step 4: Bivariate Analysis.

EDA-Step 1 :Checking & Removing duplicates, if any

Number of duplicate rows = 0

(872, 7)

No duplicates have been found

EDA-Step 2: Checking Missing value.

Table 26 - Missing values Check

HolidayPackage	0
Salary	0
Age	0
Educ	0
No_young_children	0
No_older_children	0
foreign	0
dtype: int64	

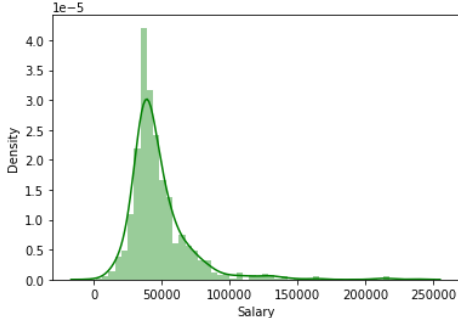
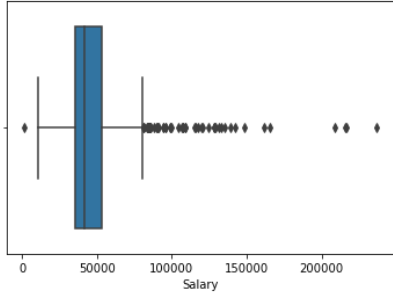
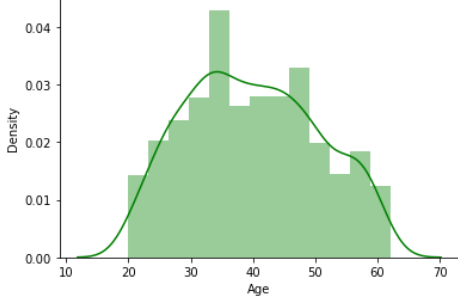
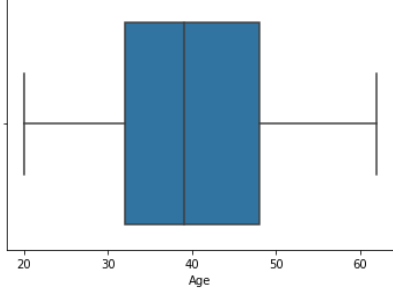
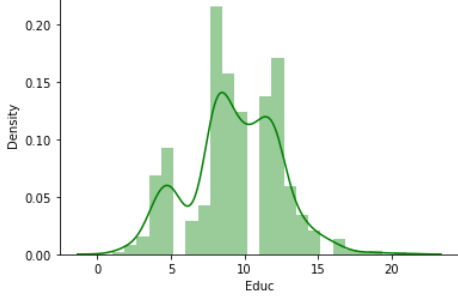
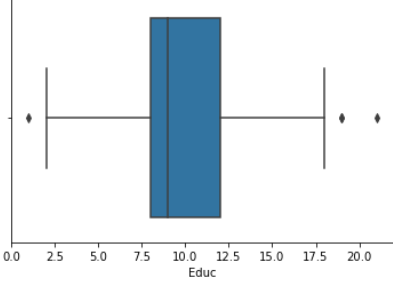
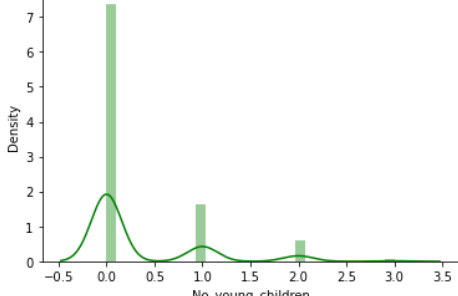
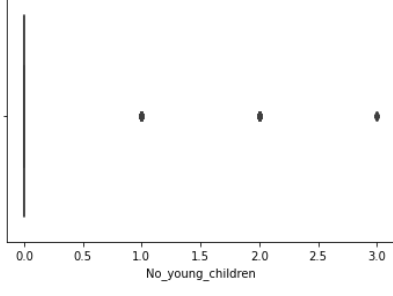
Observation:

- There are 7 variables (1 variable unnamed dropped) and 872 records.
- The variables 'Salary', 'Age', 'Educ', 'No_young_children', 'No_older_children', are numeric type .
- The variables 'HolidayPackage' and 'foreign' are object type.
- The variable 'HolidayPackage' is target variable and others are predictor variables
- No missing values and duplicate rows found.

EDA Step 3 :Univariate Analysis with Outlier treatment

- Check five-point summary for continuous variables
- Check distribution of variables
- Check outliers
-

Table 27 – Univariate Analysis

Description	Distribution plot	Boxplot
Description of Salary <hr/> count 872.000000 mean 47729.172018 std 23418.668531 min 1322.000000 25% 35324.000000 50% 41903.500000 75% 53469.500000 max 236961.000000 Name: Salary, dtype: float64 Distribution of Salary		
Description of Age <hr/> count 872.000000 mean 39.955275 std 10.551675 min 20.000000 25% 32.000000 50% 39.000000 75% 48.000000 max 62.000000 Name: Age, dtype: float64 Distribution of Age		
Description of Educ <hr/> count 872.000000 mean 9.307339 std 3.036259 min 1.000000 25% 8.000000 50% 9.000000 75% 12.000000 max 21.000000 Name: Educ, dtype: float64 Distribution of Educ		
Description of No_young_children <hr/> count 872.000000 mean 0.311927 std 0.612870 min 0.000000 25% 0.000000 50% 0.000000 75% 0.000000 max 3.000000 Name: No_young_children, dtype: float64 Distribution of No_young_children		

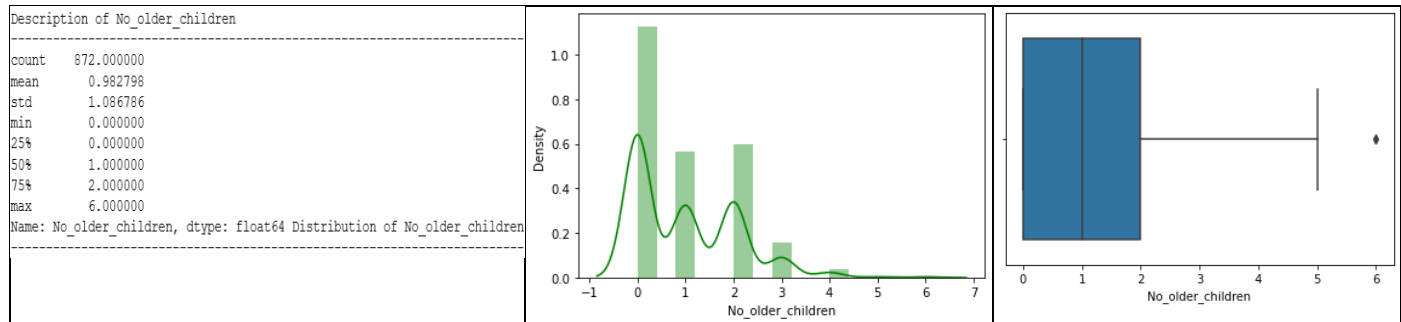


Table 28 – Skewness Analysis

Salary	3.103216
Age	0.146412
Educ	-0.045501
No_young_children	1.946515
No_older_children	0.953951
dtype:	float64

Observations:

- All the variables except Educ (left skewed) are right skewed.
- Outliers are present in salary and education
- Looking at the modes in distributed, there could be some clusters present in the variables.

Removing Outliers

We are only doing outlier treatment for Salary attribute as other columns have very less outliers and that are near lower and upper ranges

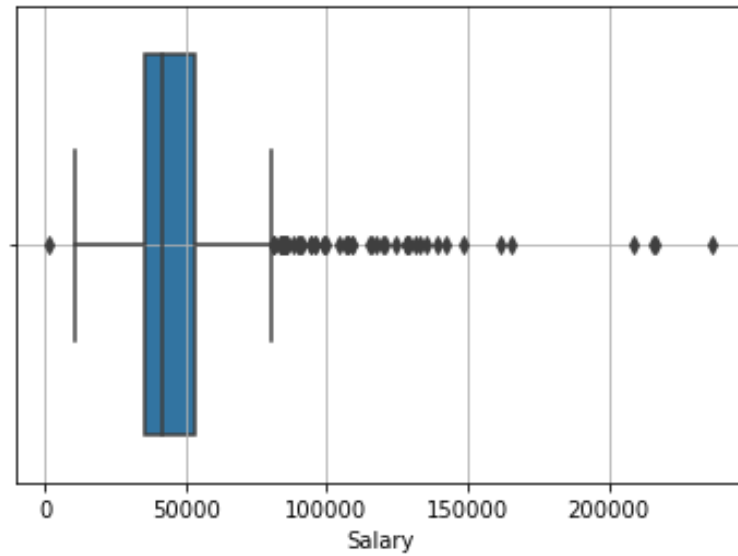


Figure 8 – Boxplot for salary before outlier treatment

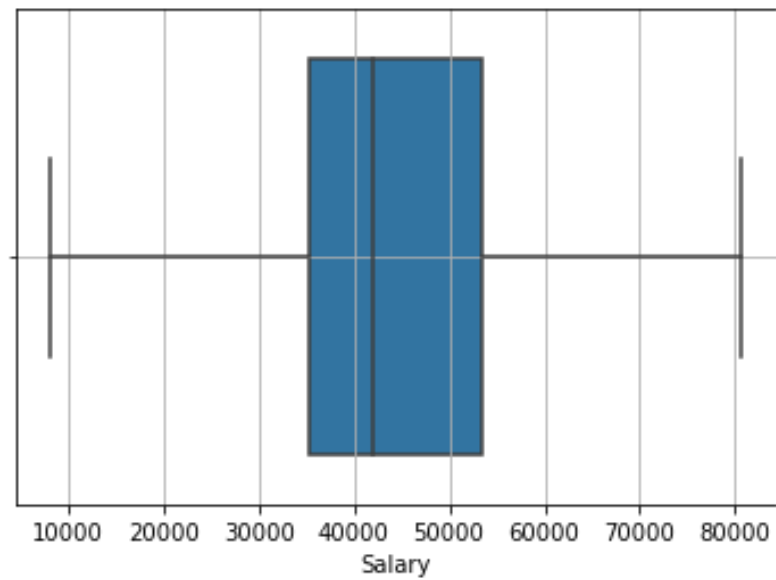


Figure 9 – Boxplot for salary after outlier treatment

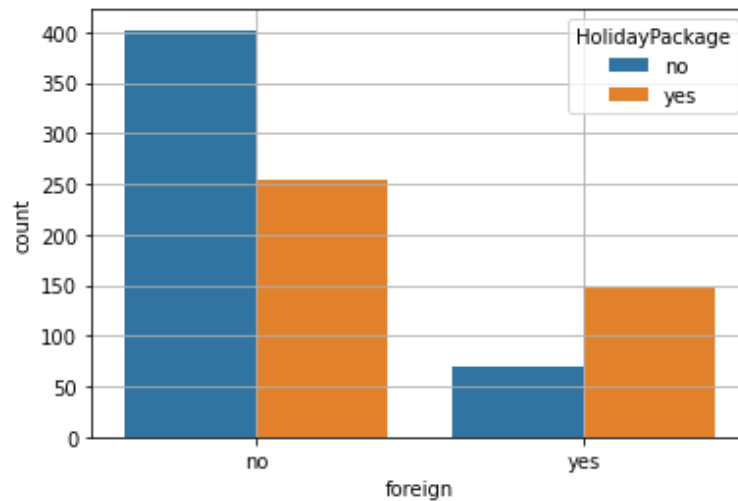
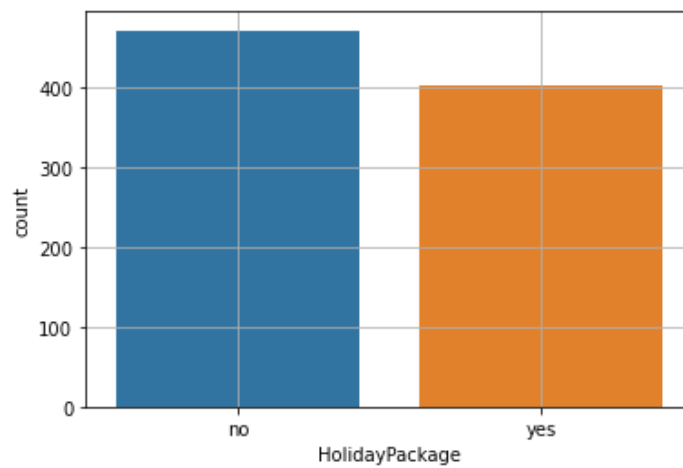


Figure 10 -Count plot for object type variables

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

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

Data looks balanced for the target variable which is Holliday_Package

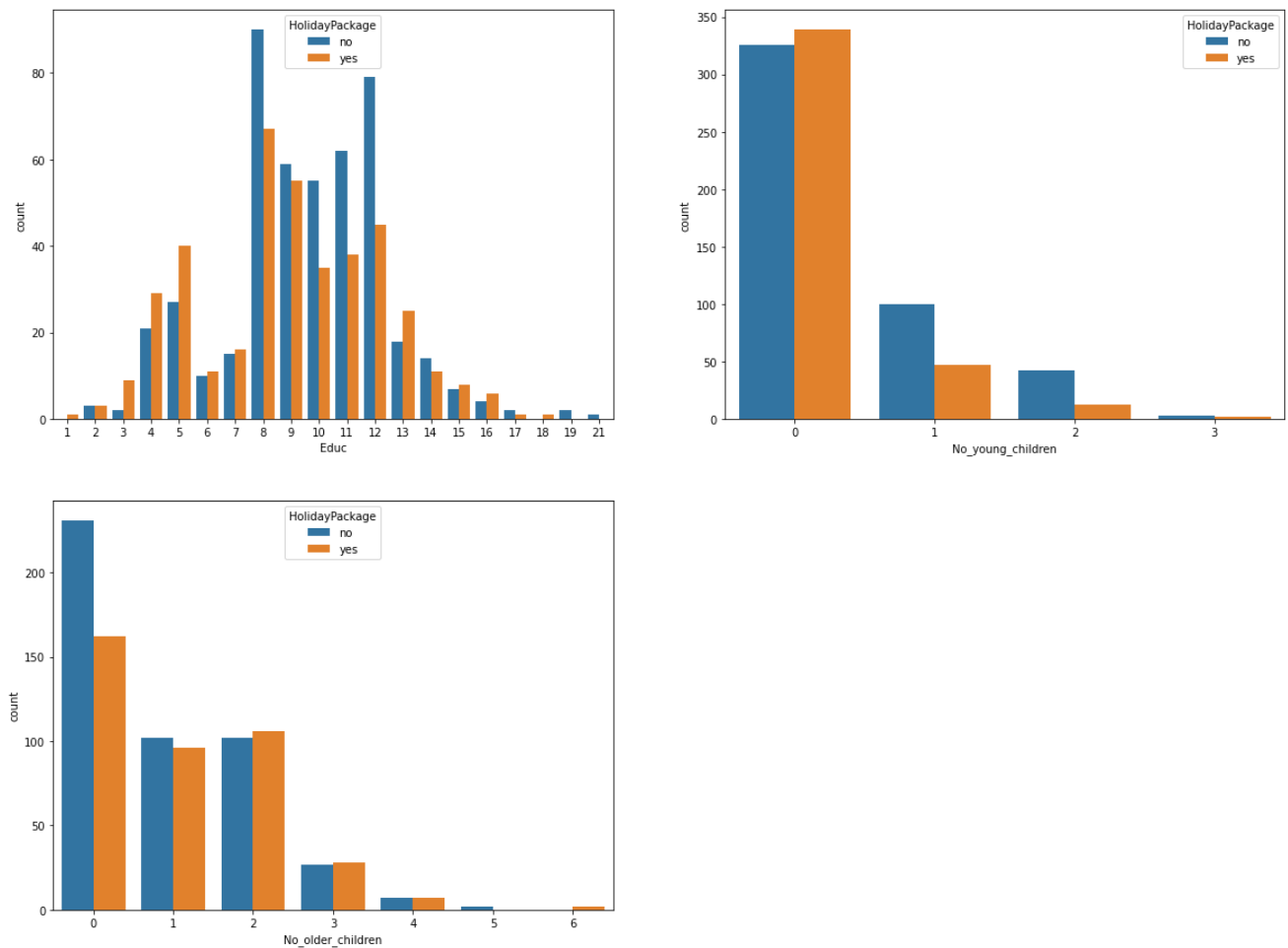


Figure 11 -Count plot for numeric type variables

Observations

- More Employees opt for Tours if their education level is 3,4,5,6,7,13,14,15,16
- Employees don't opt for tours if they have young child
- Older children count doesn't appear to have much impact on tour opted by employees or not
- Foreigner employees tends to opt more for the tour

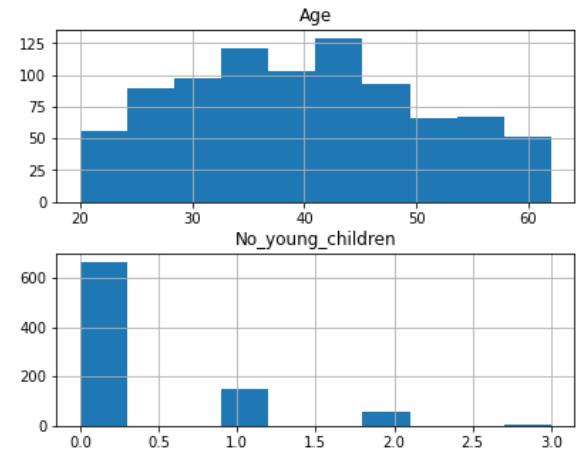
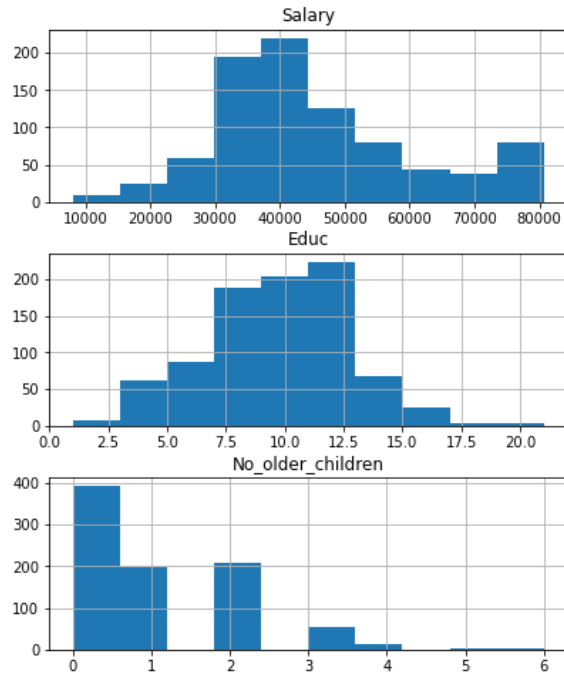


Figure 12 - Histogram for numerical variables

EDA- Step 4: Multivariate Analysis

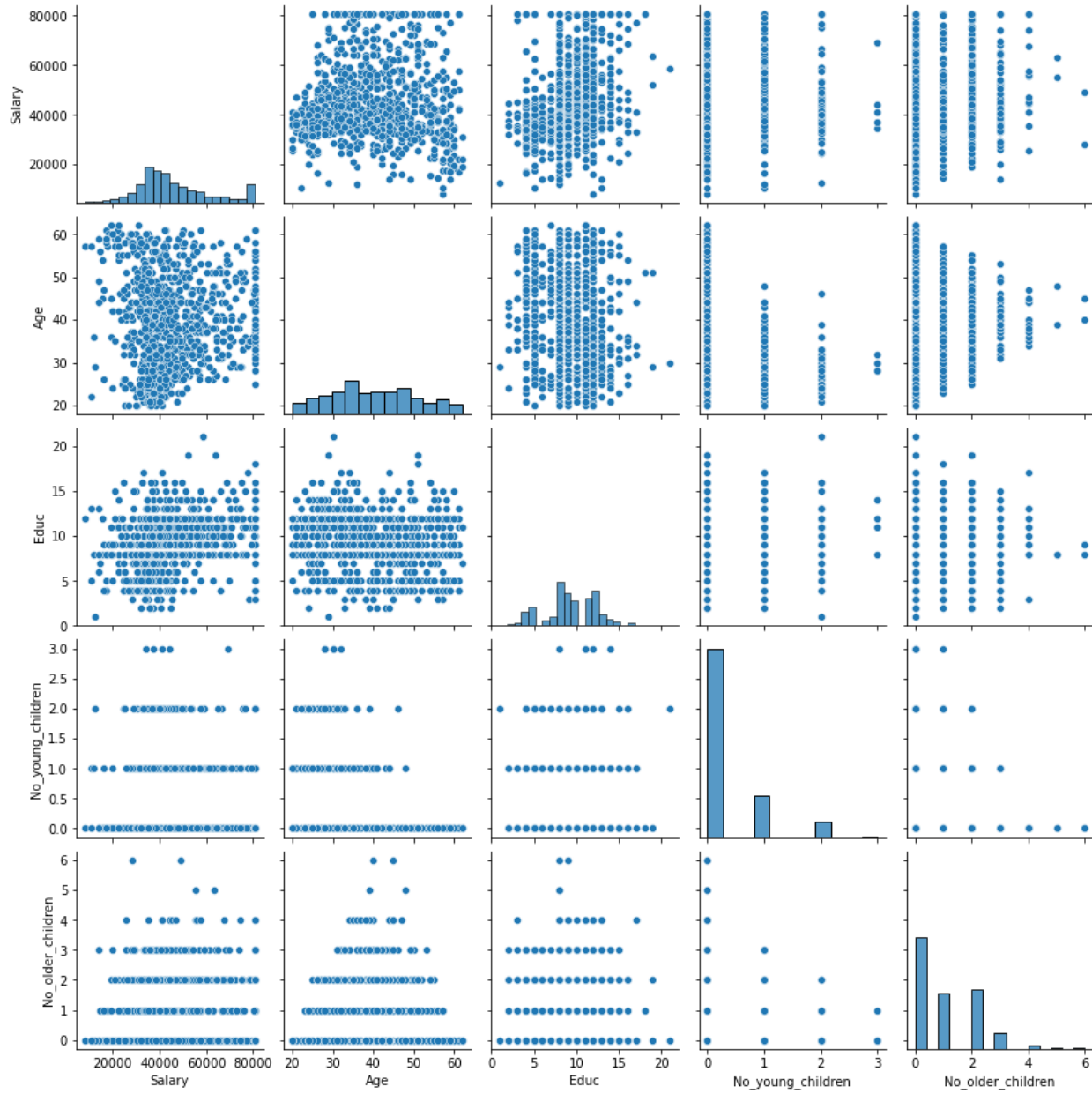


Figure 13 - Pairplot

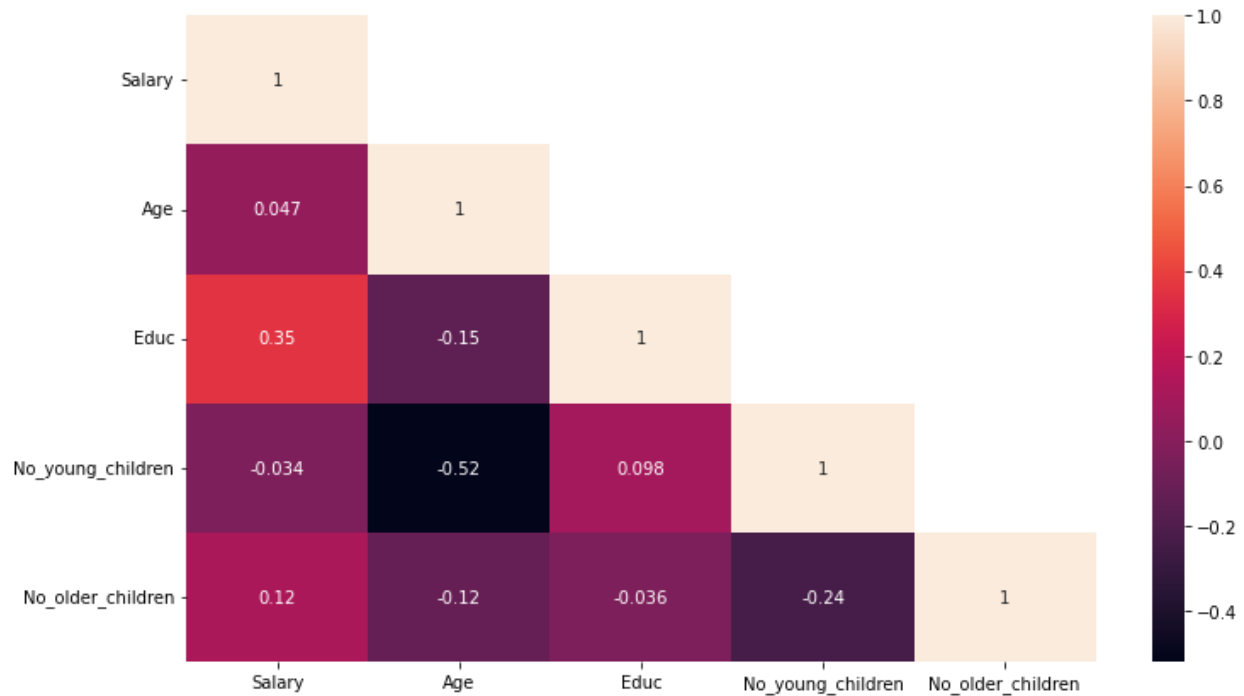


Figure 14 – Heat Map

Table 29 – Correlation Values

			correlation
No_young_children	Age		0.519093
Educ	Salary		0.352726
No_older_children	No_young_children		0.238428
Age	Educ		0.149294
No_older_children	Salary		0.121993
Age	No_older_children		0.116205
No_young_children	Educ		0.098350
Age	Salary		0.047029
No_older_children	Educ		0.036321
No_young_children	Salary		0.034360

There is some correlation between Age & No_young_children as well as Educ & Salary

Bi-Variate Analysis with Target variable

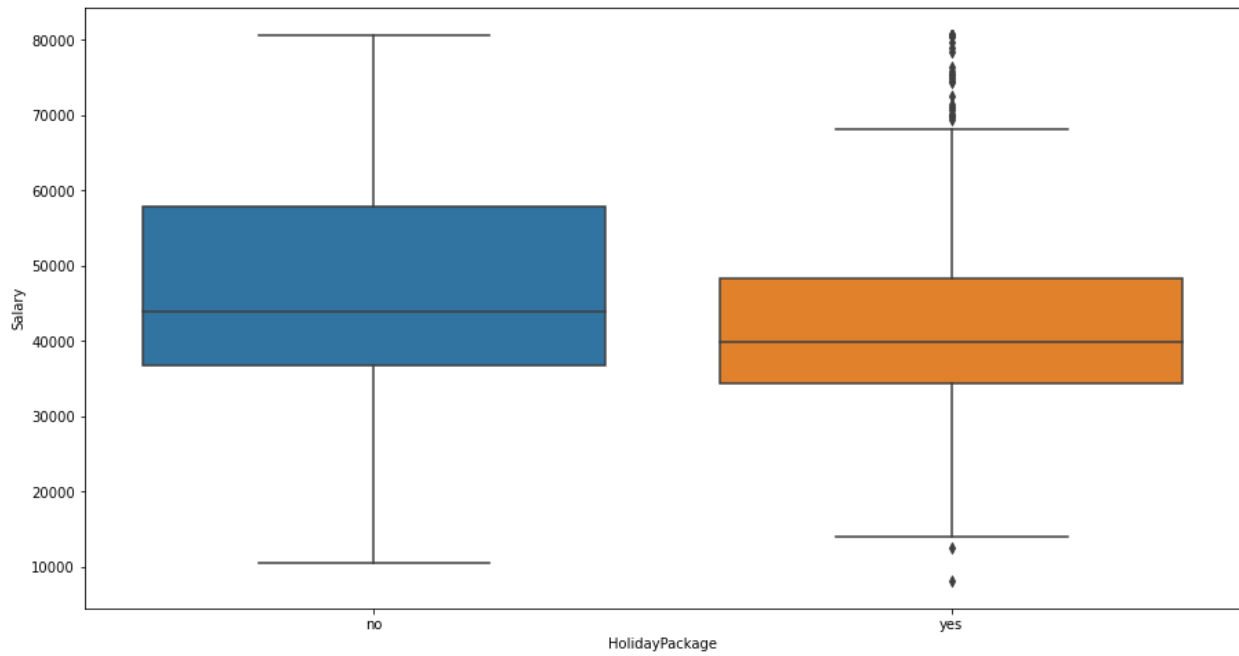


Figure 15 - Salary v/s HolidayPackage

Employees with salary greater than 50000 are less opting for holiday package.

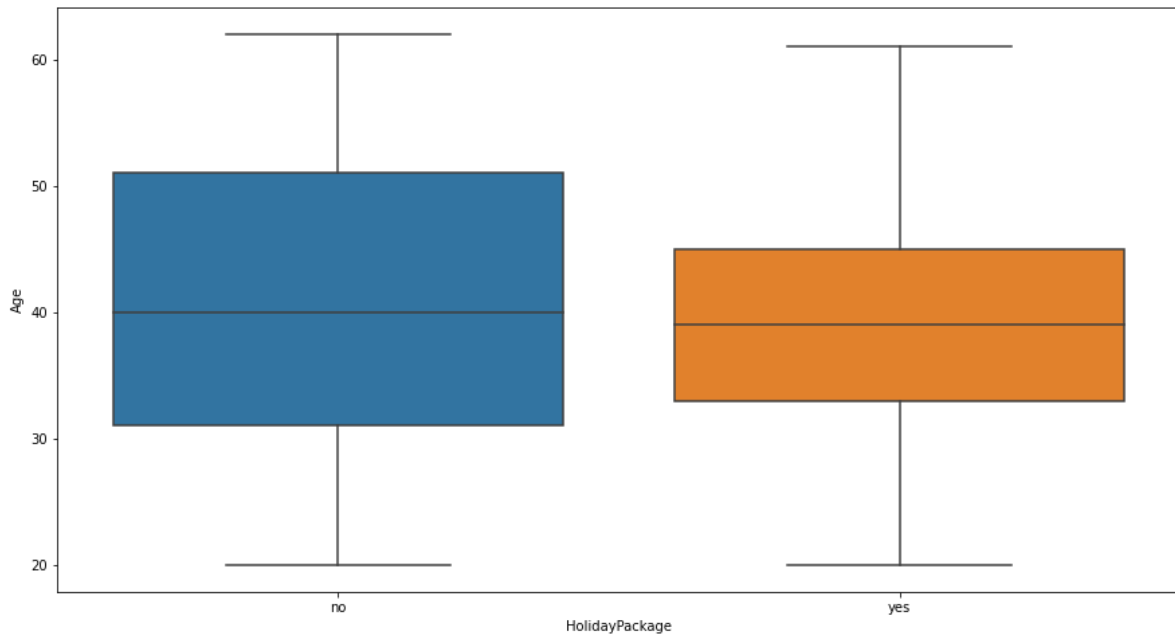


Figure 16 - Age v/s HolidayPackage

Employees with age less than 50 are more opting for holiday package.

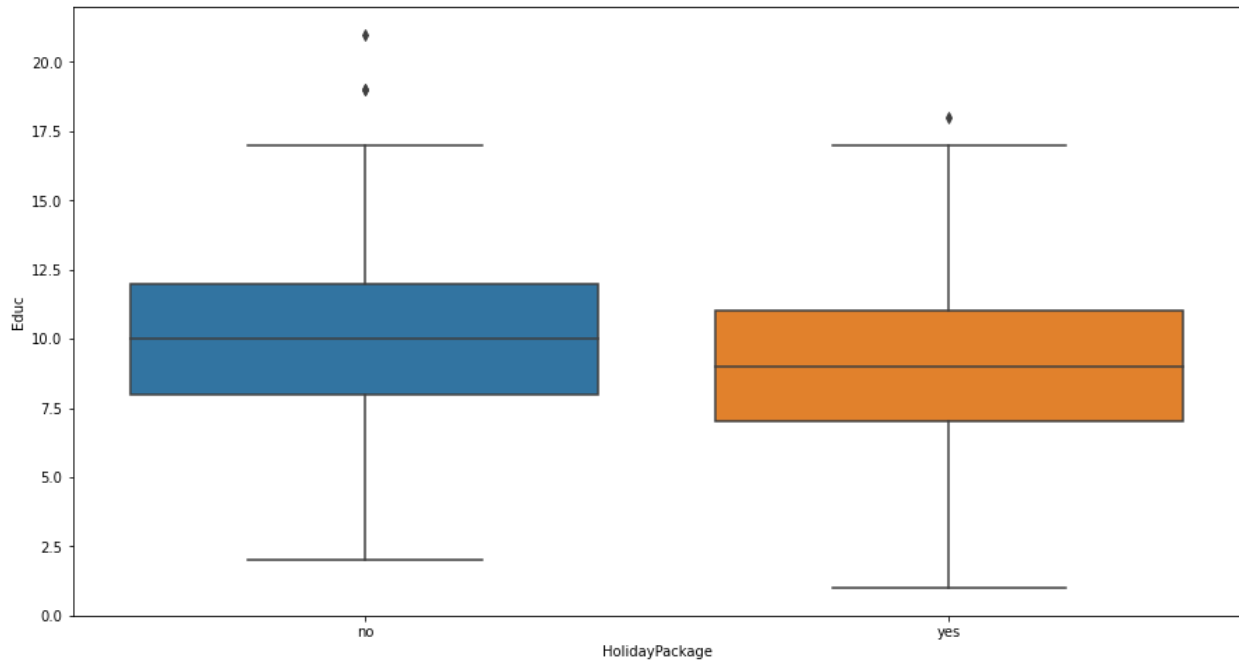


Figure 17 - Educ v/s HolidayPackage

From education point of view, almost same pattern can be seen for yes and no. Employees lies with between 7.5 years to 12.5 year of education

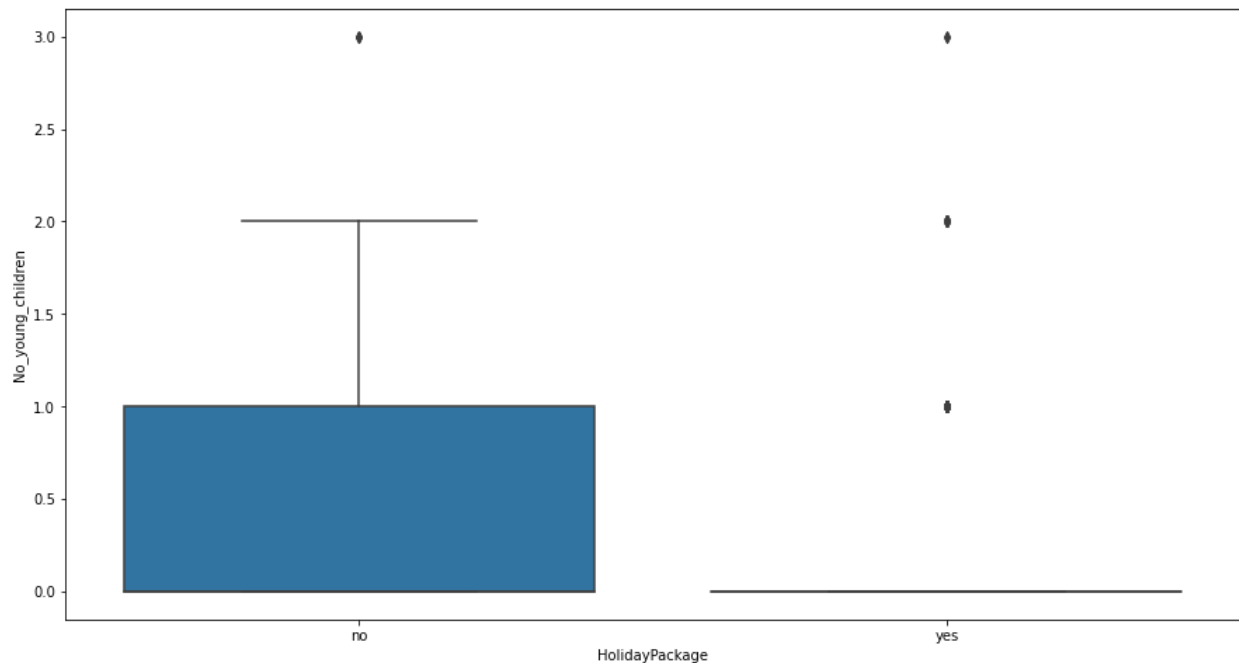


Figure 18 - No_young_children v/s HolidayPackage

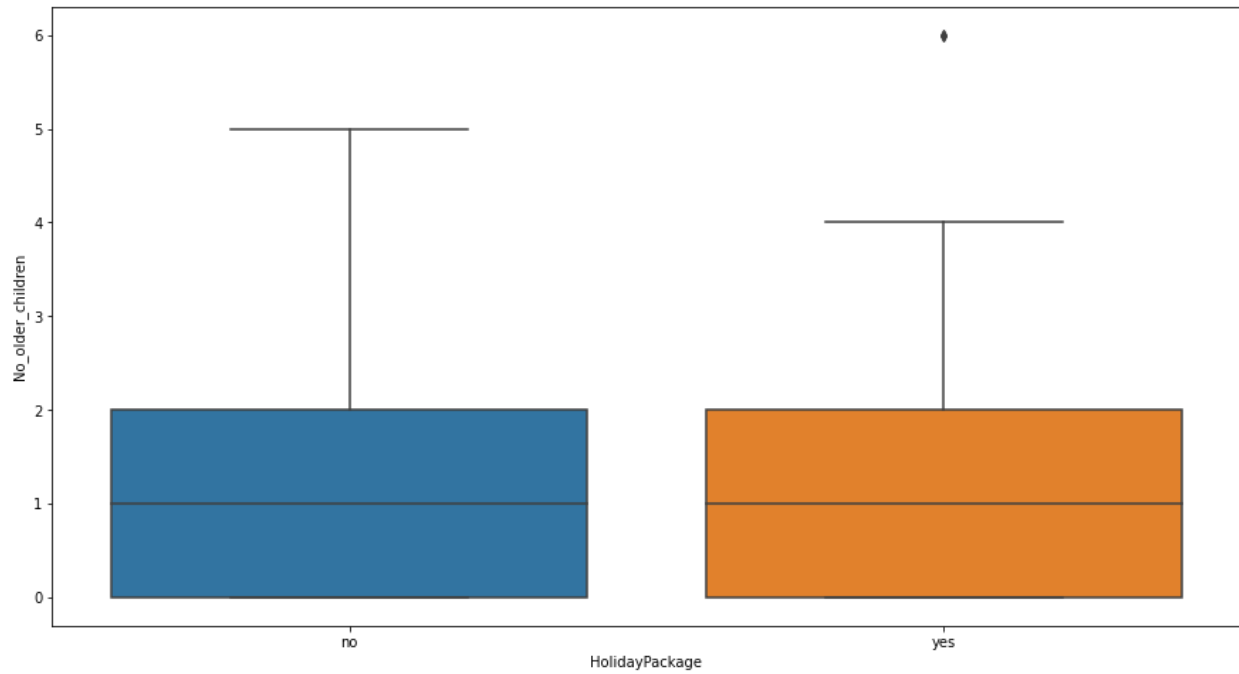


Figure 19 - No_older_children v/s HolidayPackage

Proportion in the Target classes

```
no      0.540138
yes     0.459862
Name: HolidayPackage, dtype: float64
```

45.9% of employees are opting for holiday package.

2.2 Do they 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).

Converting the 'HolidayPackage' into numeric by using the Label Encoder functionality inside sklearn.

Converting foreign variables as dummy variables

Table 30 – Encoding for categorical variables

	HolidayPackage	Salary	Age	Educ	No_young_children	No_older_children	foreign_yes
0	0	48412.0	30	8	1	1	0
1	1	37207.0	45	8	0	1	0
2	0	58022.0	46	9	0	0	0
3	0	66503.0	31	11	2	0	0
4	0	66734.0	44	12	0	2	0

	HolidayPackage	Salary	Age	Educ	No_young_children	No_older_children	foreign_yes
867	0	40030.0	24	4	2	1	1
868	1	32137.0	48	8	0	0	1
869	0	25178.0	24	6	2	0	1
870	1	55958.0	41	10	0	1	1
871	0	74659.0	51	10	0	0	1

Now all variables are numeric and we can now proceed with model building

Splitting data into training and test set in 30% test data

```
X_train (610, 6)
X_test (262, 6)
y_train (610,)
y_test (262,)
Total Obs 872
```

```
y_train.value_counts(1)
```

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

```
y_test.value_counts(1)
```

```
0    0.541985
1    0.458015
Name: HolidayPackage, dtype: float64
```

Logistic Regression Model

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

Applying GridSearchCV for Logistic Regression

```
GridSearchCV(cv=3, estimator=LogisticRegression(max_iter=10000, n_jobs=2),
             n_jobs=-1,
             param_grid={'penalty': ['l2', 'none', 'l1'],
                         'solver': ['liblinear', 'lbfgs'],
                         'tol': [0.0001, 1e-05]},
             scoring='f1')
  └─ estimator: LogisticRegression
      └─ LogisticRegression
```

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

```
{'penalty': 'l2', 'solver': 'liblinear', 'tol': 1e-05}
```

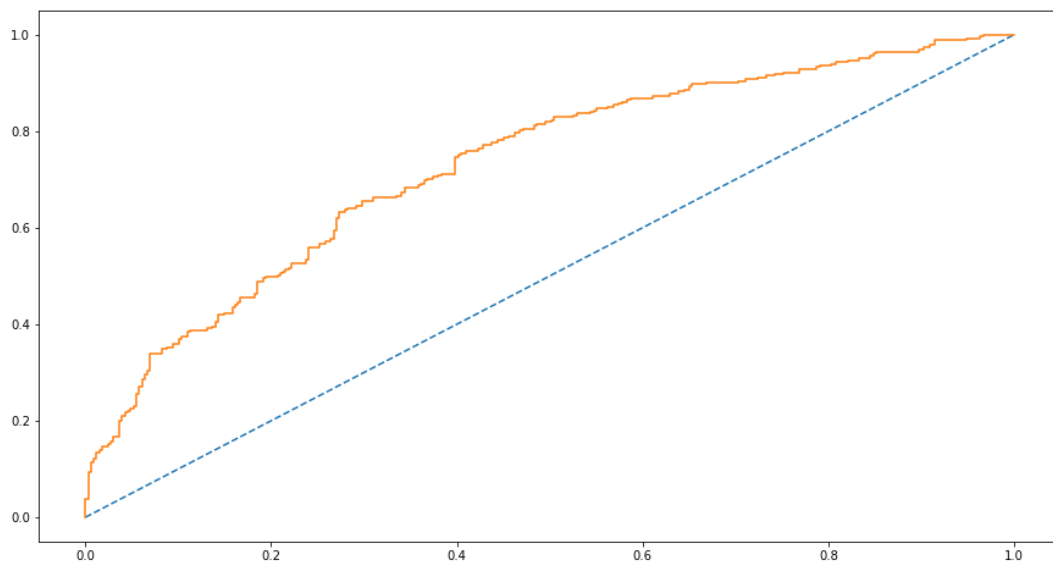
```
LogisticRegression(max_iter=10000, n_jobs=2, solver='liblinear', tol=1e-05)
```

Linear Discriminant Analysis

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

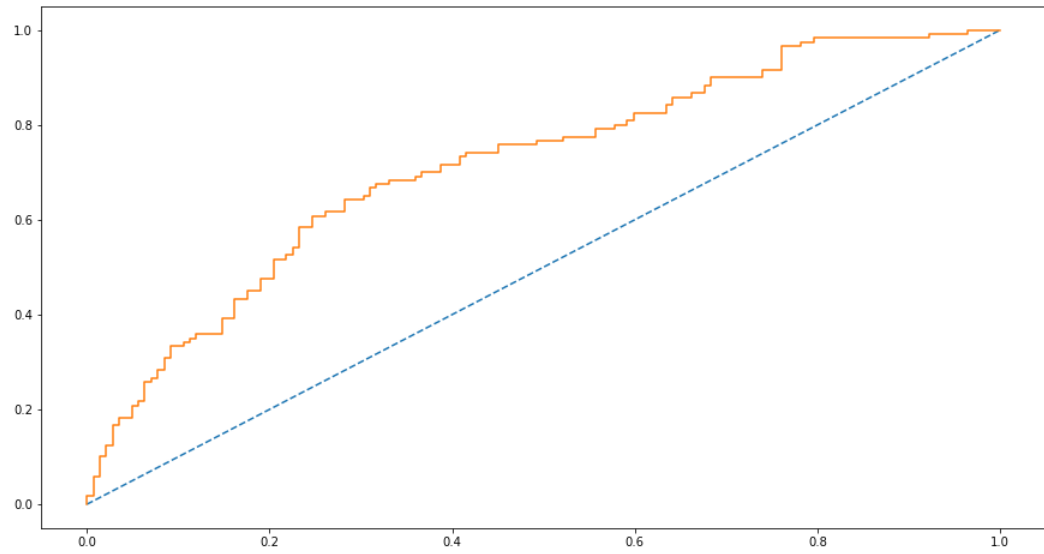
2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model Final Model: Compare Both the models and write inference which model is best/optimized.

Logistic Regression model (LR)



AUC: 0.729

Figure 20 - – AUC and ROC for the training data (LR)



AUC: 0.729

Figure 21 – AUC and ROC for the test data (LR)

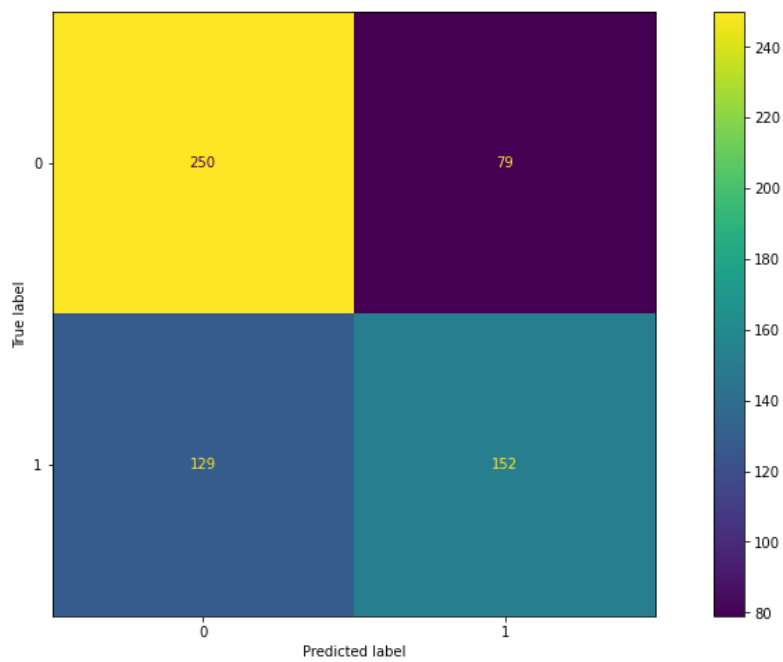


Figure 22 – Confusion Matrix for the training data (LR)

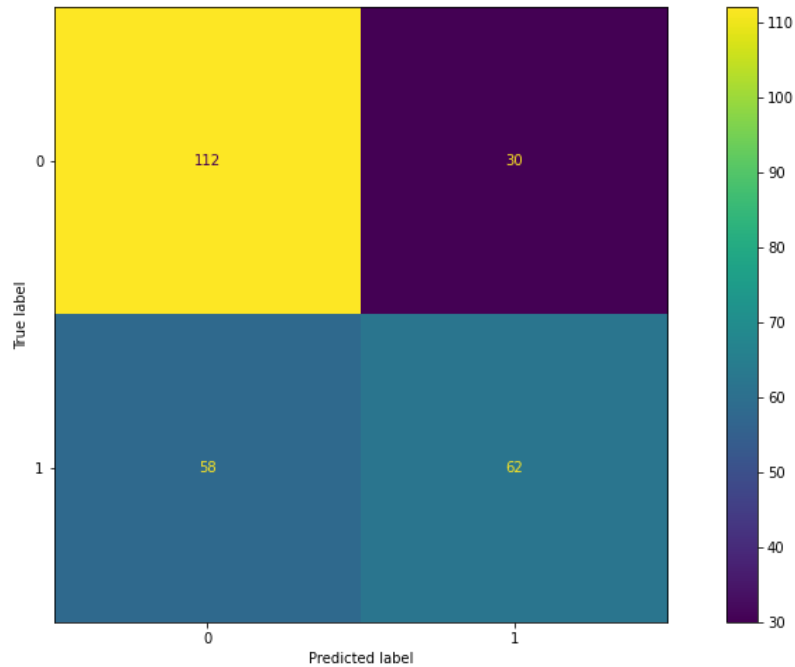


Figure 23 – Confusion Matrix for test data (LR)

Table 31 – Classification report for training data (LR)

	precision	recall	f1-score	support
0	0.66	0.76	0.71	329
1	0.66	0.54	0.59	281
accuracy			0.66	610
macro avg	0.66	0.65	0.65	610
weighted avg	0.66	0.66	0.65	610

Table 32 – Classification report for test data (LR)

	precision	recall	f1-score	support
0	0.66	0.79	0.72	142
1	0.67	0.52	0.58	120
accuracy			0.66	262
macro avg	0.67	0.65	0.65	262
weighted avg	0.67	0.66	0.66	262

Linear Discriminant model (LDA)

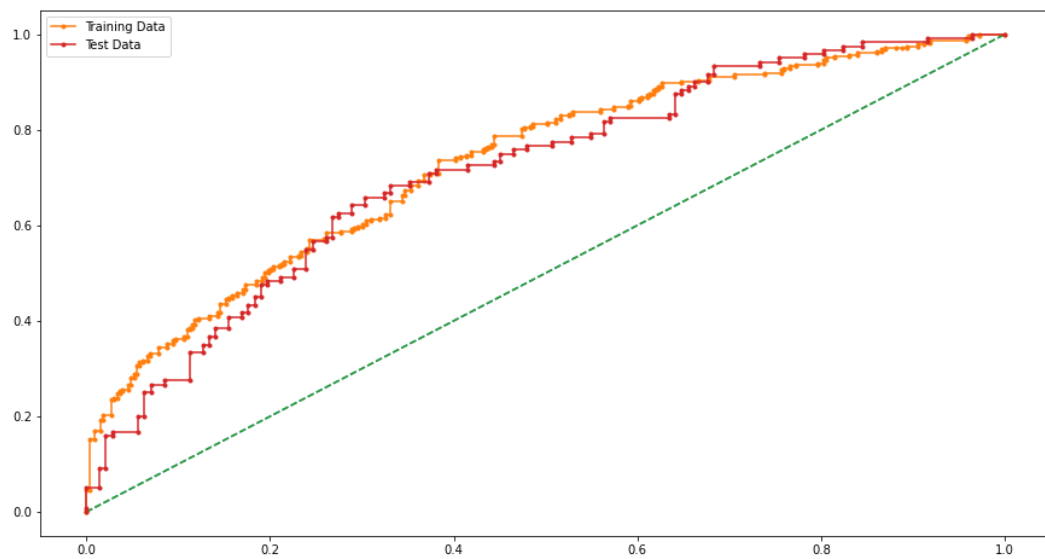


Figure 24 – AUC and ROC for the training & testing data (LDA)

AUC for the Training Data: 0.731

AUC for the Test Data: 0.714

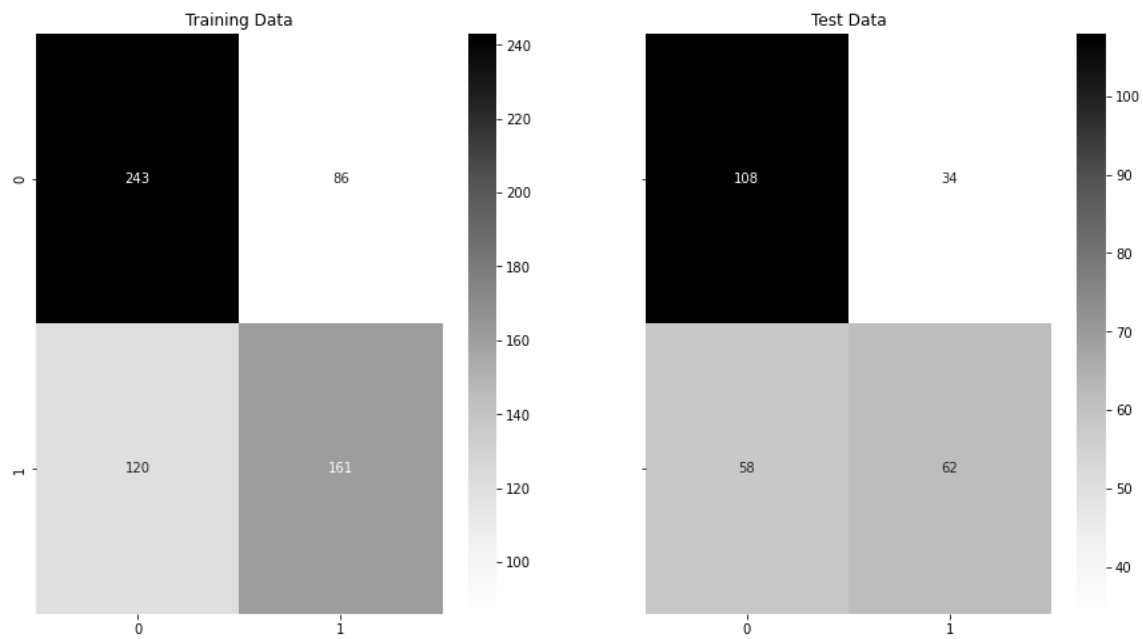


Figure 25 – Confusion Matrix for the training & testing data (LDA)

Table 33 – Classification report for training data (LDA)

Classification Report of the training data:

	precision	recall	f1-score	support
0	0.67	0.74	0.70	329
1	0.65	0.57	0.61	281
accuracy			0.66	610
macro avg	0.66	0.66	0.66	610
weighted avg	0.66	0.66	0.66	610

Table 34 – Classification report for test data (LDA)

Classification Report of the test data:

	precision	recall	f1-score	support
0	0.65	0.76	0.70	142
1	0.65	0.52	0.57	120
accuracy			0.65	262
macro avg	0.65	0.64	0.64	262
weighted avg	0.65	0.65	0.64	262

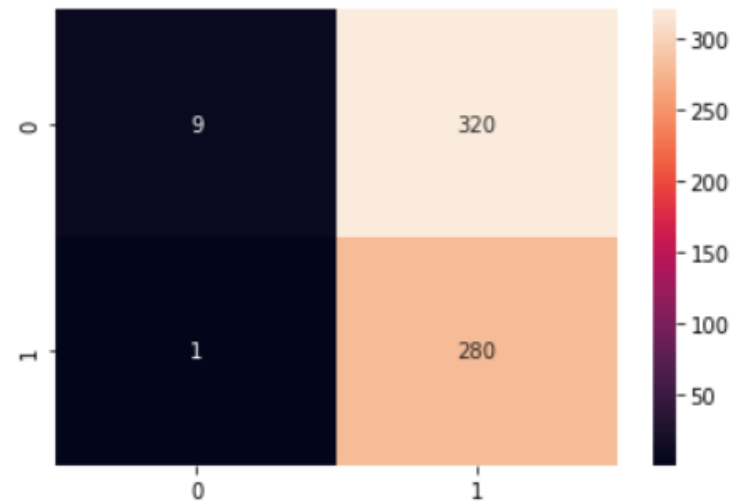
Change the cut-off values for maximum accuracy

0.1

Accuracy Score 0.4738

F1 Score 0.6356

Confusion Matrix

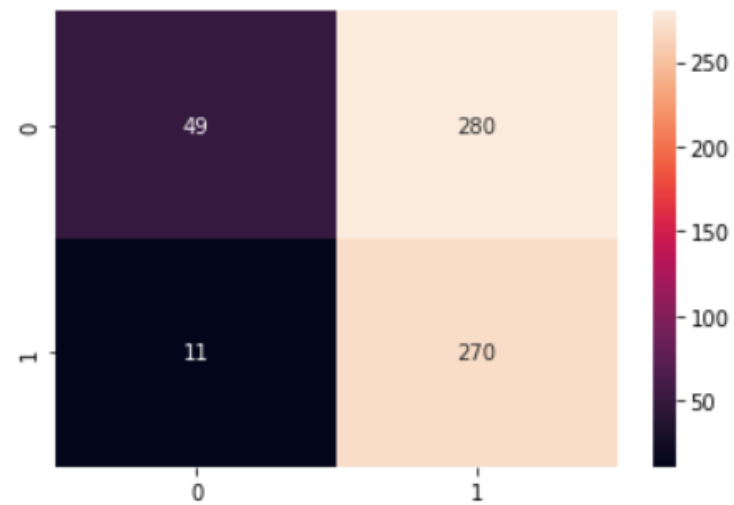


0.2

Accuracy Score 0.523

F1 Score 0.6498

Confusion Matrix

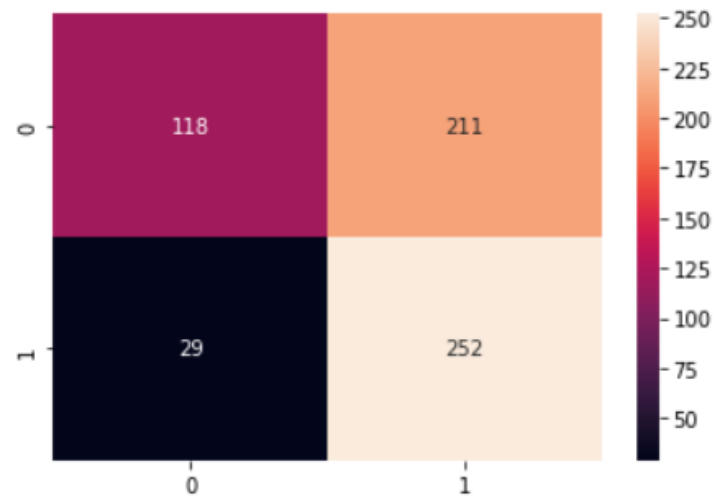


0.3

Accuracy Score 0.6066

F1 Score 0.6774

Confusion Matrix

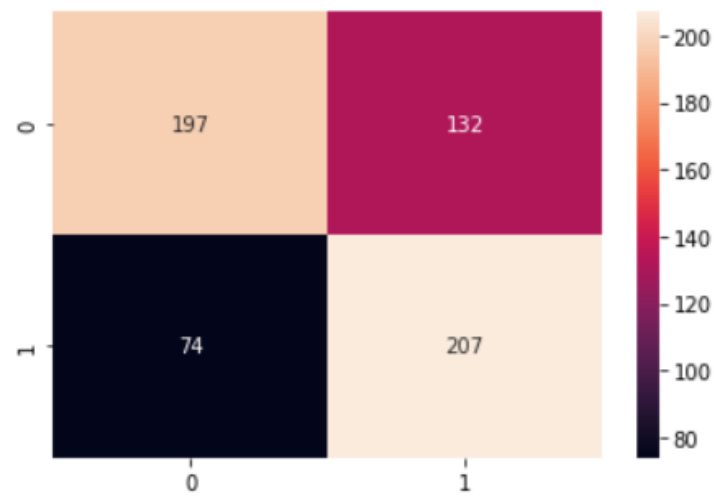


0.4

Accuracy Score 0.6623

F1 Score 0.6677

Confusion Matrix

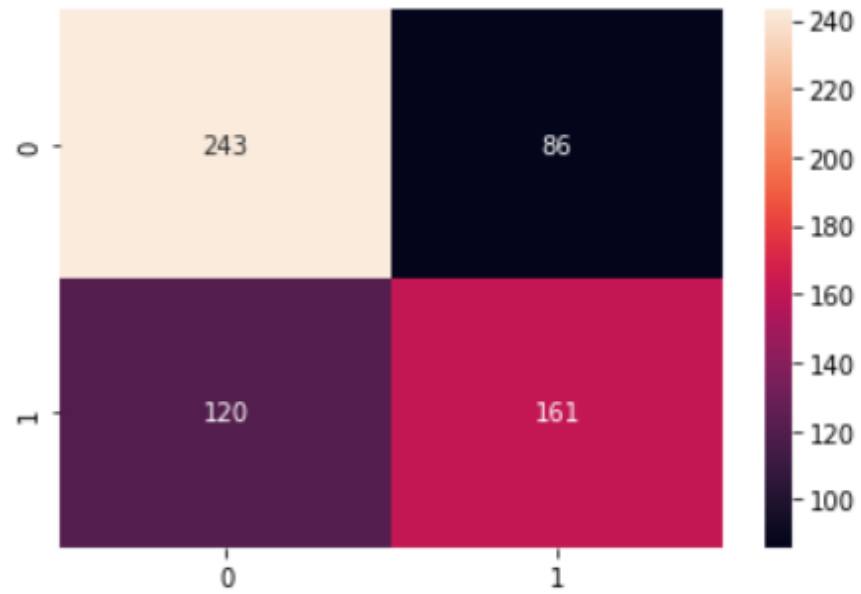


0.5

Accuracy Score 0.6623

F1 Score 0.6098

Confusion Matrix

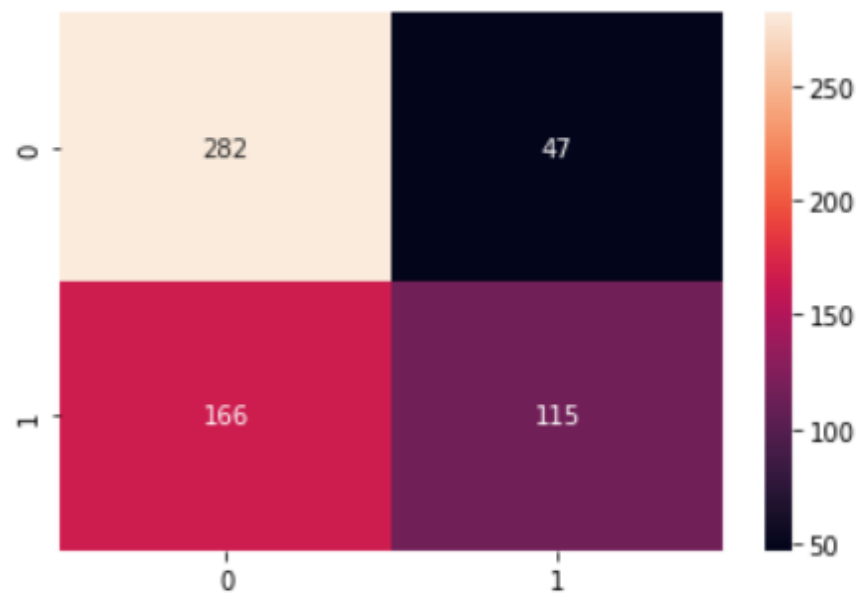


0.6

Accuracy Score 0.6508

F1 Score 0.5192

Confusion Matrix

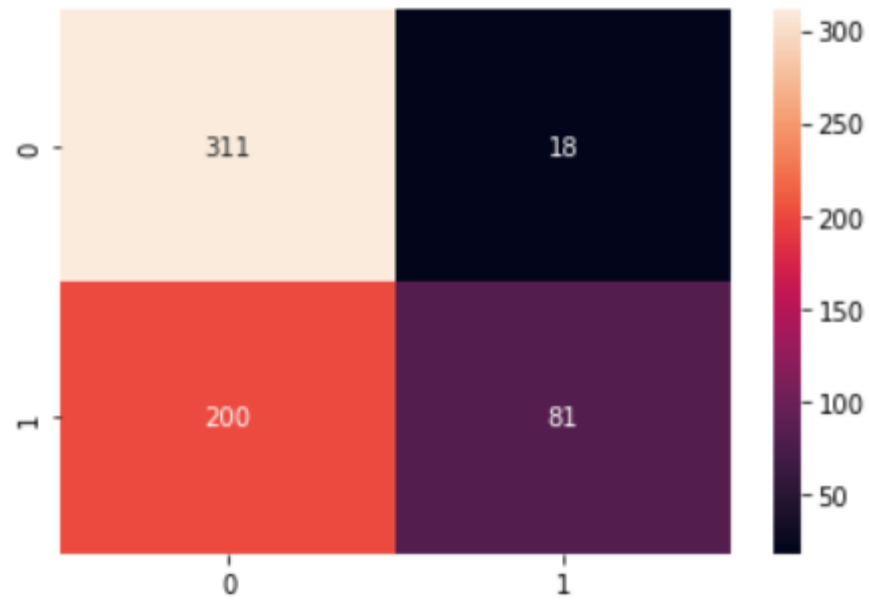


0.7

Accuracy Score 0.6426

F1 Score 0.4263

Confusion Matrix

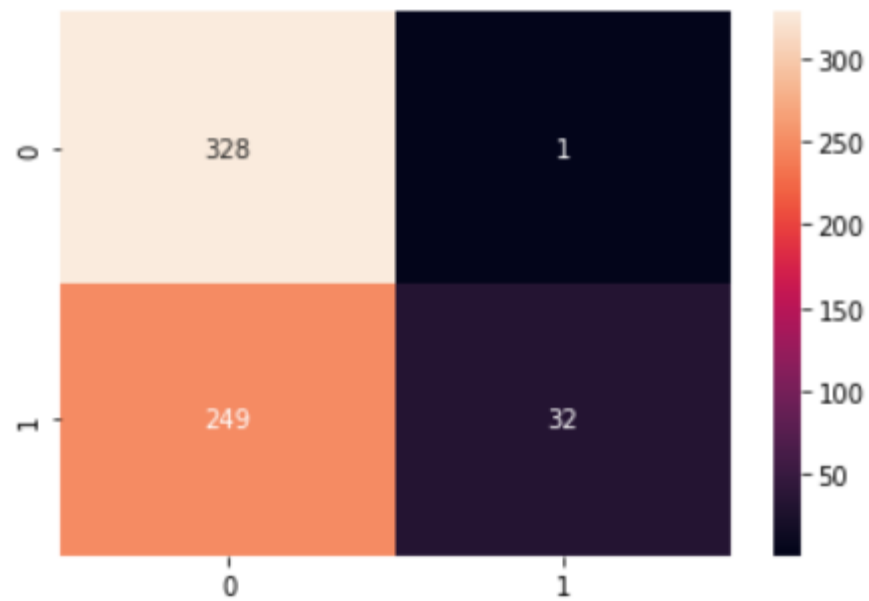


0.8

Accuracy Score 0.5902

F1 Score 0.2038

Confusion Matrix



0.9

Accuracy Score 0.5426

F1 Score 0.0141

Confusion Matrix

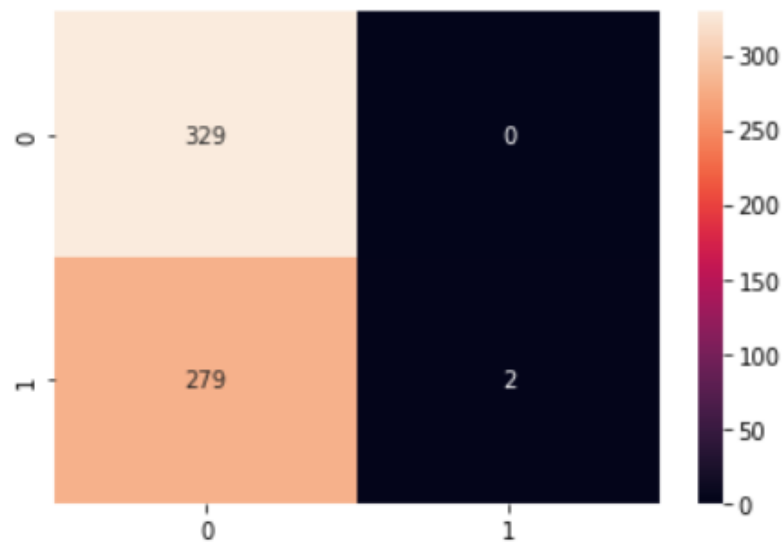


Figure 26 – Confusion Matrix with custom cut off

0.4 cut-off gives us the best 'f1-score' and accuracy score. Let us evaluate the predictions of the test data using these cut-off values.

Predicting the classes on the test data for cutoff = 0.4

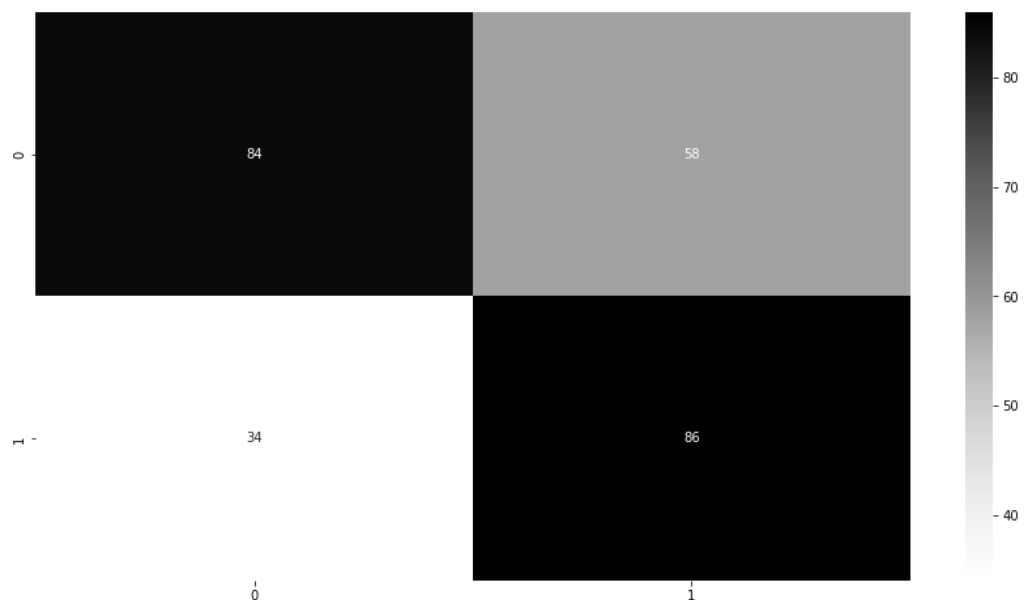


Figure 27 – Confusion Matrix with custom cut off for Test data

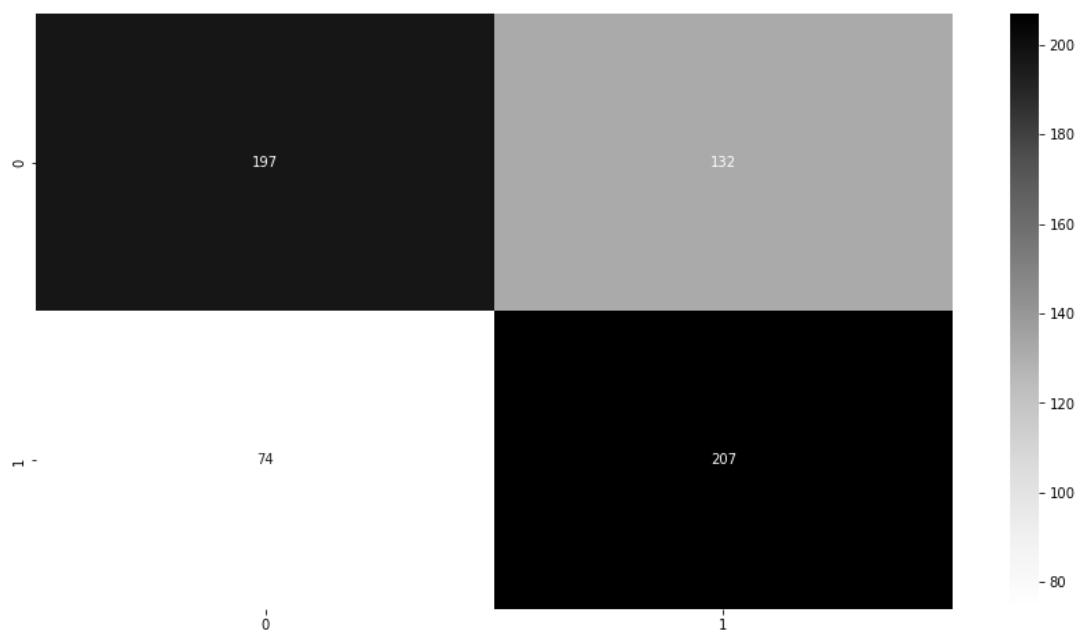


Figure 28 – Confusion Matrix with custom cut off for Train data

Table 35 – Classification report for test data with default and custom cut-off

Classification Report of the default cut-off test data:

	precision	recall	f1-score	support
0	0.65	0.76	0.70	142
1	0.65	0.52	0.57	120
accuracy			0.65	262
macro avg	0.65	0.64	0.64	262
weighted avg	0.65	0.65	0.64	262

Classification Report of the custom cut-off test data:

	precision	recall	f1-score	support
0	0.71	0.59	0.65	142
1	0.60	0.72	0.65	120
accuracy			0.65	262
macro avg	0.65	0.65	0.65	262
weighted avg	0.66	0.65	0.65	262

Table 36 – Classification report for train data with default and custom cut-off

Classification Report of the default cut-off train data:

	precision	recall	f1-score	support
0	0.67	0.74	0.70	329
1	0.65	0.57	0.61	281
accuracy			0.66	610
macro avg	0.66	0.66	0.66	610
weighted avg	0.66	0.66	0.66	610

Classification Report of the custom cut-off train data:

	precision	recall	f1-score	support
0	0.73	0.60	0.66	329
1	0.61	0.74	0.67	281
accuracy			0.66	610
macro avg	0.67	0.67	0.66	610
weighted avg	0.67	0.66	0.66	610

Since we are building a model to predict if whether employee will opt for tour or not, for practical purposes, we will be more interested in correctly classifying 1 (employees opting for tour) than 0(employees not opting for tour).Below is comparison of all models.

Table 37 – Comparison of all model

	LR Train	LR Test	LDA Train With default cutoff	LDA Test With default cutoff	LDA Train with cutoff 0.4	LDA Test with cutoff 0.4
Accuracy	0.66	0.66	0.66	0.65	0.66	0.65
AUC	0.729	0.729	0.731	0.714	0.731	0.714
Recall	0.54	0.52	0.57	0.52	0.74	0.72
Precision	0.66	0.67	0.65	0.65	0.61	0.60
F1 Score	0.59	0.58	0.61	0.57	0.67	0.65

Form above we can see that for LDA, there is improvement in Recall and F score which means LDA perform better.

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

For the business problem of holiday package, two model were analysed i.e. Logistic Regression & Linear Discriminant Analysis for the predictions. These two models were evaluated on training and testing datasets and model performance were analysed.

The Accuracy, Precision and F1 score was computed using classification report. The confusion matrix, AUC_ROC score and ROC plot was computed and compared for different models.

Train and Test dataset have similar statistics; hence model is giving similar result for test and train data set.

With maximum accuracy of 65% and recall rate of 72% for test data model is only able to predict 72% of total tours which were actually claimed as claimed.

F1-score is the harmonic mean of precision and recall, it takes into the effect of both the scores and this value is low if any of these 2 values is low.

Both the models have similar results for Accuracy and precision while for LDA, there is improvement in Recall and F score which means overall LDA perform better.

Based on LDA

For predicting Holiday package = yes (Label 1):

- Precision (60%) – 60% of employees predicted are actually opting for holiday package of all employees predicted to opt for holiday package
- Recall (72%) – Out of all the employees actually opting for holiday package, 72% of employees have been predicted correctly

For predicting Holiday package = no (Label 0):

- Precision (71%) – 71% of employees predicted are actually opting for holiday package of all employees predicted to opt for holiday package .
- Recall (59%) – Out of all the employees actually opting for holiday package, 59% of employees have been predicted correctly .

Since we are building a model to predict if whether employee will opt for tour or not, for practical purposes, we will be more interested in correctly classifying 1 (employees opting for tour) than 0(employees not opting for tour).

Overall accuracy of the model – 65 % of total predictions are correct

Accuracy, AUC, Precision and Recall for test data is almost in line with training data.

This proves no overfitting or underfitting has happened, and overall, the model is a good model for classification

Recommendation & Insights:

- Employees over the age of 50 seems to be not taking holiday packages are compared to younger employees.
- Employees with salary less than 50000 are opting for holiday package.
- 45% employees are taking holiday packages.
- If employee is foreigner and employee not having young children, chances of opting for Holiday Package is good.
- A survey to understand good destination for people above 50 years may help to attract them to take holiday packages
- Targeting employees with younger children should be avoided as conversion rate seems to be less.