PROJECT REPORT - PREDICTIVE MODELING

By prakash v Mahadole

Table of Content

Problem 1 : Linear Regression

Statement:

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

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

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

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

Problem 2 : Logistic Regression and LDA

Statement:

Q 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.

Q 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).

Q 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.

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

Please explain and summarise the various steps performed in this project. There should be proper business interpretation and actionable insights present.

Quality of Business Report

Problem 1: Linear Regression

Problem Statement:

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:

- Carat: Carat weight of the cubic zirconia.
- Cut: Describe the cut quality of the cubic zirconia. Quality is increasing order Fair, Good, Very Good, Premium, Ideal.
- Color: Colour of the cubic zirconia.With D being the best and J the worst.
- Clarity: cubic zirconia Clarity refers to the absence of the Inclusions and Blemishes. (In order from Best to Worst, FL = flawless, I1= level 1 inclusion) IF, VVS1, VVS2, VS1, VS2, Sl1, Sl2, 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. ### Q 1.1. Read the data and do exploratory data analysis. Describe the data briefly. (Check the null values, Data types, shape, EDA). Perform Univariate and Bivariate Analysis.

Loading all the necessary library for the model building. Now, reading the head and tail of the dataset to check whether data has been

Loading all the necessary library for the model building. Now, reading the head and tail of the dataset to check whether data has been properly fed.

Head of the data

Out[3]:

	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

Tail of the data

Out[4]:

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

Checking shape of the data

(26967,11)

Checking Data info

```
<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
```

As we have 11 columns with 3 object, 6 float and 2 int data types in the data.

Check for null values

```
Out[7]: 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
```

we have total of 697 null values in the Depth variable.

Data Description

Out[10]:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Unnamed: 0	26967	NaN	NaN	NaN	13484	7784.85	1	6742.5	13484	20225.5	26967
carat	26967	NaN	NaN	NaN	0.798375	0.477745	0.2	0.4	0.7	1.05	4.5
cut	26967	5	Ideal	10816	NaN	NaN	NaN	NaN	NaN	NaN	NaN
color	26967	7	G	5661	NaN	NaN	NaN	NaN	NaN	NaN	NaN
clarity	26967	8	SI1	6571	NaN	NaN	NaN	NaN	NaN	NaN	NaN
depth	26270	NaN	NaN	NaN	61.7451	1.41286	50.8	61	61.8	62.5	73.6
table	26967	NaN	NaN	NaN	57.4561	2.23207	49	56	57	59	79
x	26967	NaN	NaN	NaN	5.72985	1.12852	0	4.71	5.69	6.55	10.23
y	26967	NaN	NaN	NaN	5.73357	1.16606	0	4.71	5.71	6.54	58.9
z	26967	NaN	NaN	NaN	3.53806	0.720624	0	2.9	3.52	4.04	31.8
price	26967	NaN	NaN	NaN	3939.52	4024.86	326	945	2375	5360	18818

Observation:

- Based on summary descriptive, the data looks good.
- we see most of the variable have mean/median nearly equal.

We have both categorical and continuous data,

For categorical data we have cut, colour and clarity

For continuous data we have carat, depth, table, x. y, z and price

Price will be target variable

Checking duplicate data

There is no duplicate rows in data

Getting unique values of all the categorical variables

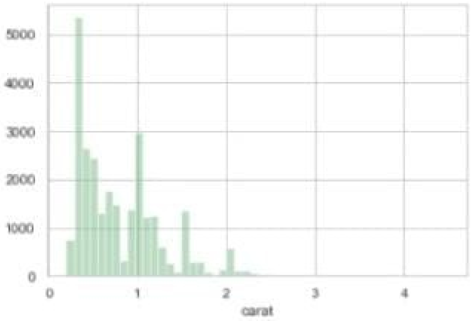
```
cut :  5
Fair          781
Good         2441
Very Good    6030
Premium      6899
Ideal       10816
Name: cut, dtype: int64

color : 7
J    1443
I    2771
D    3344
H    4102
F    4729
E    4917
G    5661
Name: color, dtype: int64

clarity : 8
I1       365
IF       894
VVS1    1839
VVS2    2531
VS1     4093
SI2     4575
VS2     6099
SI1     6571
Name: clarity, dtype: int64
```


As we can see we have all the unique values of all the categorical variables.

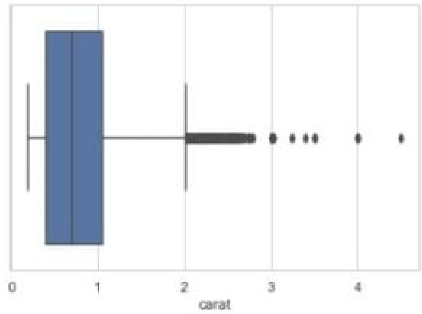
Univariate/Bivariate Analysis

Distribution of Carat:

```
Description of carat
...........................................................
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
...........................................................
```




```
Boxplot of carat
...........................................................
```



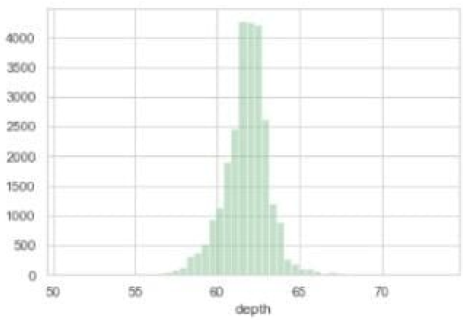
Observation:

Distribution of carat:

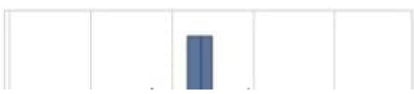
The distribution of data in carat seems to positively skewed, as there are multiple peaks points in the distribution there could multimode and the box plot of carat seems to have large number of outliers. In the range of 0 to 1 where majority of data lies.

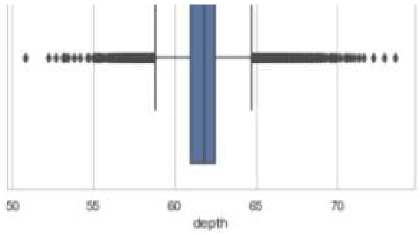
Destribution of Depth:

```
Description of depth
...........................................................
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
...........................................................
```




```
Boxplot of depth
...........................................................
```





Distribution of Depth:

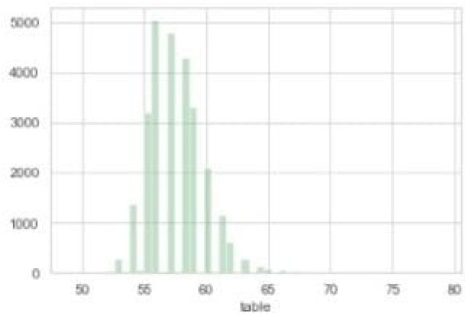
The distribution of depth seems to be normal distribution,

The depth ranges from 55 to 65.

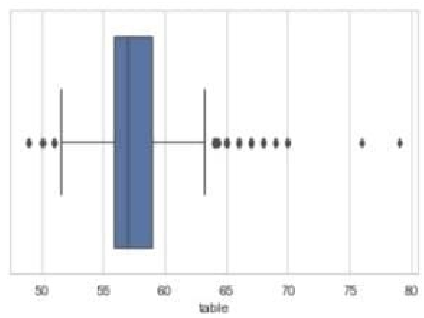
The box plot of the depth distribution holds many outliers.

Distribution of table:

```
Description of table
..........................................................
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
..........................................................
```




```
Boxplot of table
..........................................................
```



Distribution of table:

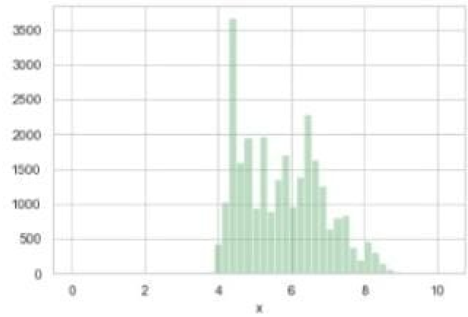
The distribution of table also seems to be positively skewed.

The box plot of table has outliers.

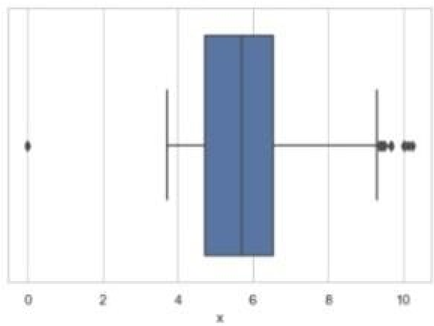
The data distribution where there is maximum distribution is between 55 to 65.

Distribution of x:

```
Description of x
..........................................................
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
..........................................................
```




```
Boxplot of x
..........................................................
```



Distribution of x:

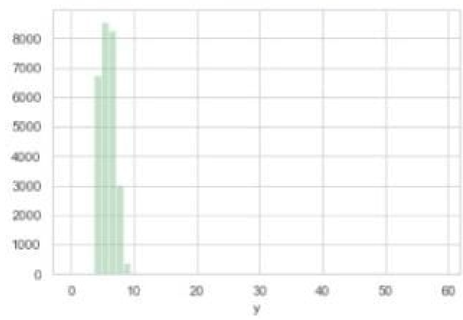
The distribution of x (Length of the cubic zirconia in mm.) is positively skewed.

The box plot of the data consists of many outliers.

The distribution rages from 4 to 8.

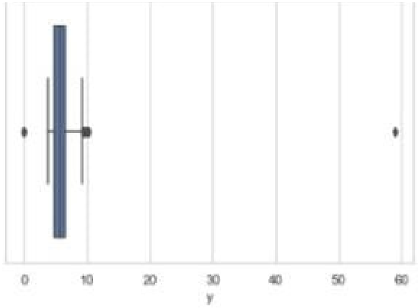
Distribution of y:

```
Description of y
..........................................................
count    26967.000000
mean         5.733569
std          1.166058
min          0.000000
25%          4.710000
50%          5.710000
75%          6.540000
max         58.900000
Name: y, dtype: float64
Distribution of y
..........................................................
```



y

```
Boxplot of y
..........................................................
```



Distribution of y:

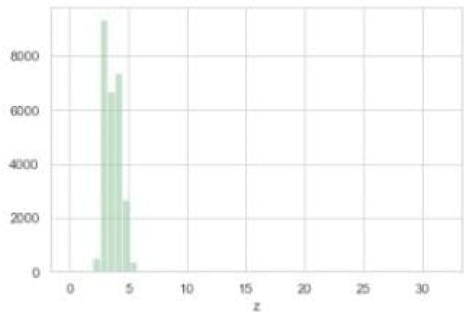
The distribution of Y (Width of the cubic zirconia in mm.) is positively skewed.

The box plot also consists of outliers.

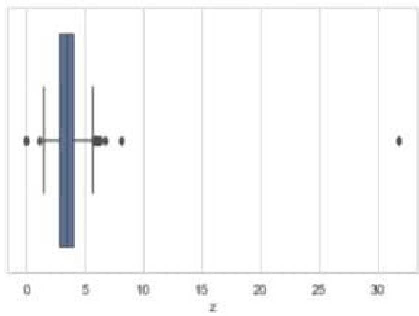
The distribution too much positively skewed. The skewness may be due to the diamonds are always made in specific shape. There might not be too much sizes in the market.

Distribution of z:

```
Description of z
.............................................................
count    26967.000000
mean         3.538057
std          0.720624
min          0.000000
25%          2.900000
50%          3.520000
75%          4.040000
max         31.800000
Name: z, dtype: float64
Distribution of z
.............................................................
```




```
Boxplot of z
.............................................................
```



Distribution of z:

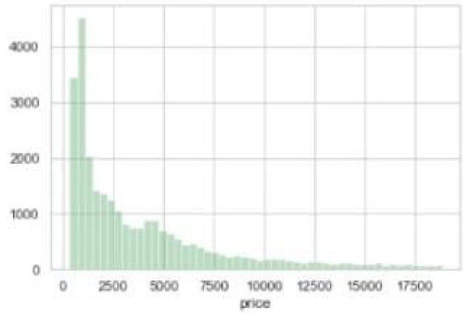
The distribution of z (Height of the cubic zirconia in mm.) is positively skewed.

The box plot also consists of outliers.

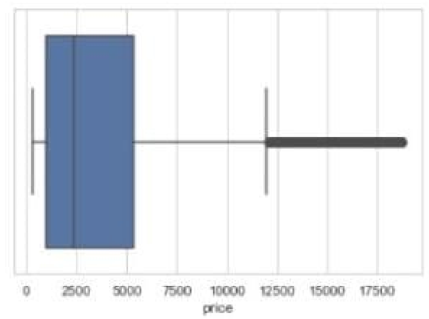
The distribution too much positively skewed. The skewness may be due to the diamonds are always made in specific shape. There might not be too much sizes in the market.

Distribution of price:

```
Description of price
...........................................................
count    26967.000000
mean      3939.518115
std       4024.864666
min        326.000000
25%        945.000000
50%       2375.000000
75%       5360.000000
max      18818.000000
Name: price, dtype: float64
Distribution of price
...........................................................
```




```
Boxplot of price
...........................................................
```



Distribution of Price:

The price has seems to be positively skewed. The skew is positive.

The price has outliers in the data.

The price distribution is from rs 100 to 8000.

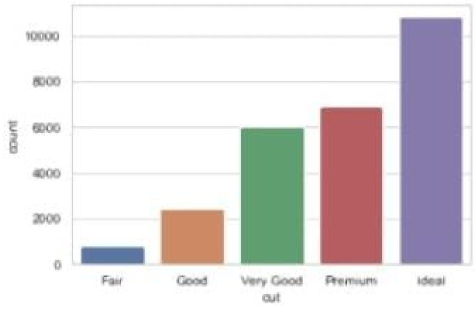
Checking Skewness

```
Out[16]: y        3.850189
         z        2.568257
         price    1.618550
         carat    1.116481
         table    0.765758
         x        0.387986
         depth   -0.028618
         dtype: float64
```

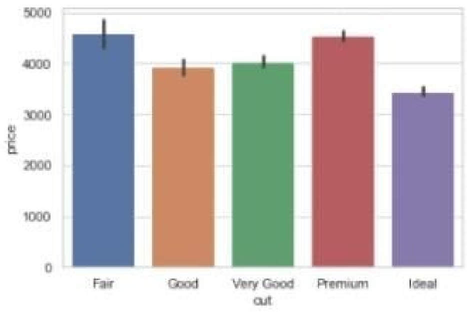
Bivariate Analysis

Categorical Variables

Cut:

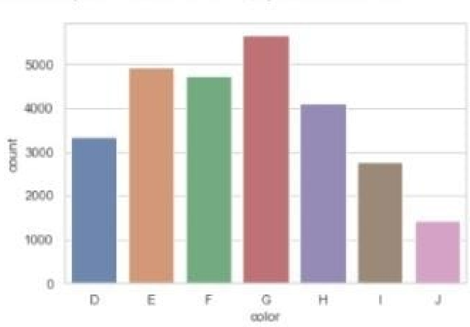


The most preferred cut seems to be ideal cut for diamonds.

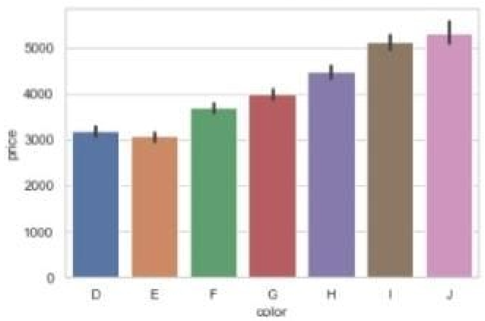


The reason for the most preferred cut ideal is because those diamonds are priced lower than other cuts.

COLOR:

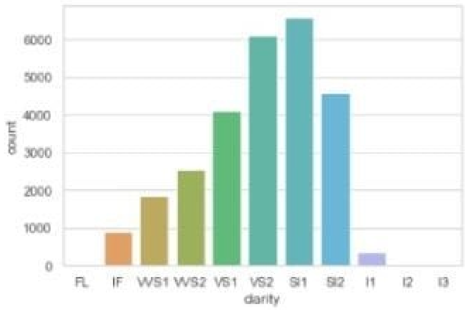


We have 7 colours in the data, The G seems to be the preferred colour.

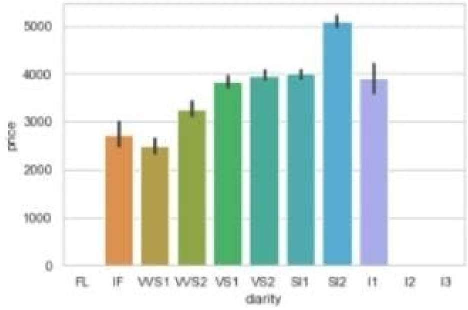


We see the G is priced in the middle of the seven colours, whereas J being the worst colour price seems too high.

CLARITY:

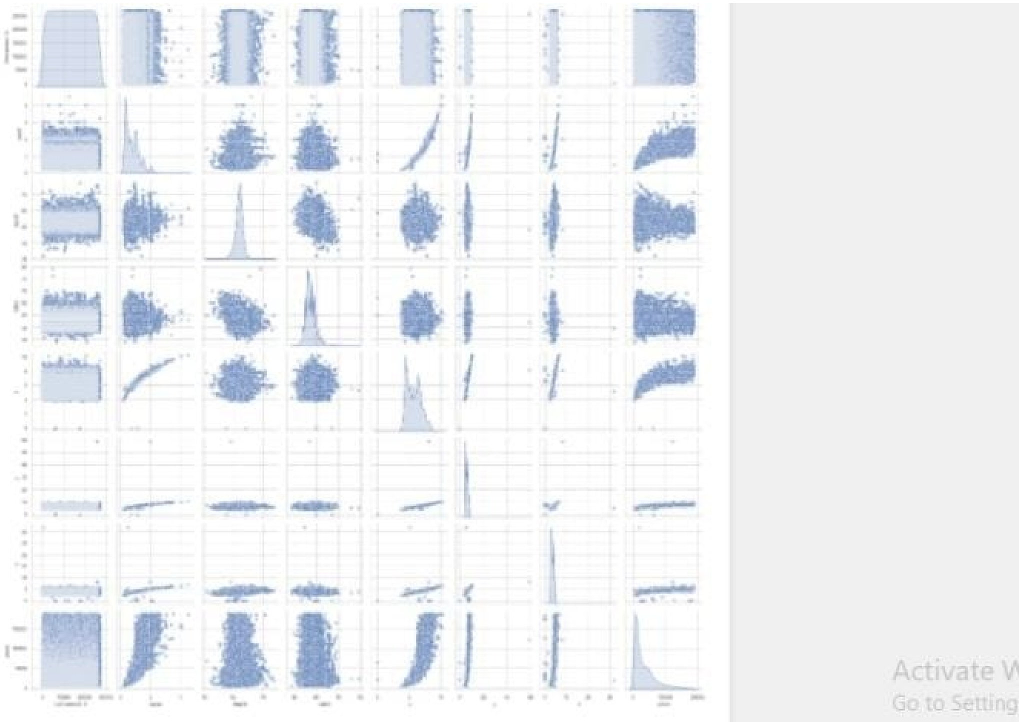


The clarity VS2 seems to be preferred by people.



The data has No FL diamonds, from this we can clearly understand the flawless diamonds are not bringing any profits to the store.

Pairplot:



Correlation Matrix:

This matrix clearly shows the presence of multi collinearity in the dataset.

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

Yes we have Null values in depth, since depth being continuous variable mean or median imputation can be done. The percentage of Null values is less than 5%, we can also drop these if we want. After median imputation, we don't have any null values in the datase

```
Out[27]: 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
```

After median imputation, we don't have any null values in the dataset.

Checking for values which are equal to zero

Out[29]:

	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

We have certain rows having values zero, the x, y, z are the dimensions of a diamond so this can't take into model. As there are very less rows.

We can drop these rows as don't have any meaning in model building.

Scaling:

Scaling can be useful to reduce or check the multi collinearity in the data, so if scaling is not applied I find the VIF – variance inflation factor values very high.

Which indicates presence of multi collinearity.

These values are calculated after building the model of linear regression. To understand the multi collinearity in the model.

The scaling had no impact in model score or coefficients of attributes nor the intercept.

Out[34]:

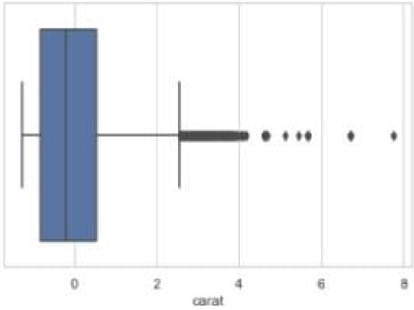
	Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
0	-1.731904	-1.043125	Ideal	E	SI1	0.253399	0.244112	-1.295920	-1.240065	-1.224865	-0.854851
1	-1.731776	-0.980310	Premium	G	IF	-0.679158	0.244112	-1.162787	-1.094057	-1.169142	-0.734303
2	-1.731647	0.213173	Very Good	E	VVS2	0.325134	1.140496	0.275049	0.331668	0.335404	0.584271
3	-1.731519	-0.791865	Ideal	F	VS1	-0.105277	-0.652273	-0.807766	-0.802041	-0.806936	-0.709945
4	-1.731390	-1.022187	Ideal	F	VVS1	-0.966099	0.692304	-1.224916	-1.119823	-1.238796	-0.785257

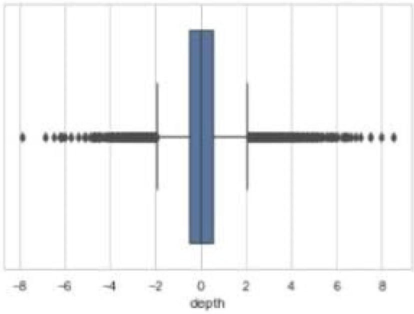
Checking data head after scaling.

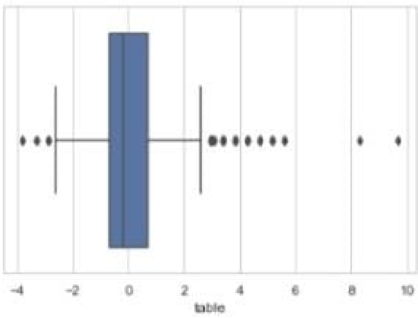
VIF Values after Scaling:

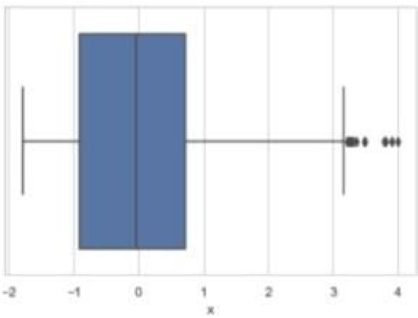
```
carat ---> 33.35086119845924
depth ---> 4.573918951598579
table ---> 1.7728852812619
x ---> 463.5542785436457
y ---> 462.769821646584
z ---> 238.65819968687333
cut_Good ---> 3.6096181949437143
cut_Ideal ---> 14.34812508118844
cut_Premium ---> 8.623414379121153
cut_Very Good ---> 7.848451571723688
color_E ---> 2.371070464762613
```

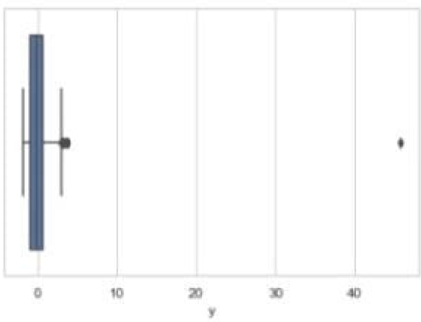
Checking Outliers in the data before outlier Treatment

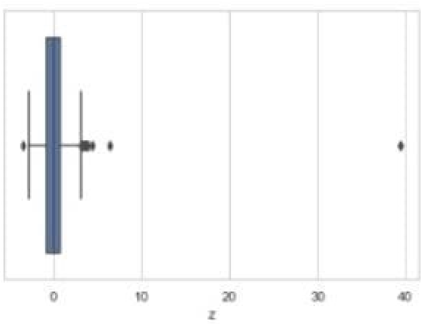

0 2 4 6 8
carat

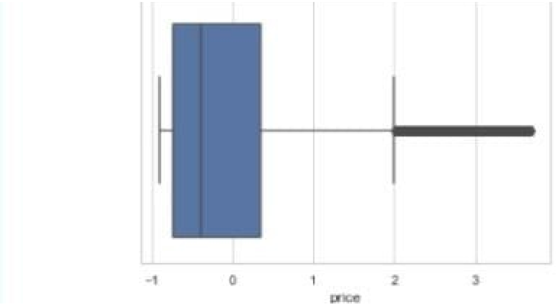

-8 -6 -4 -2 0 2 4 6 8
depth


-4 -2 0 2 4 6 8 10
table

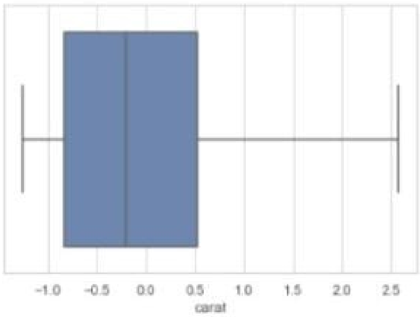

-2 -1 0 1 2 3 4
x

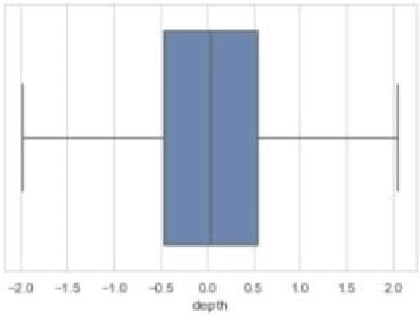

0 10 20 30 40
y

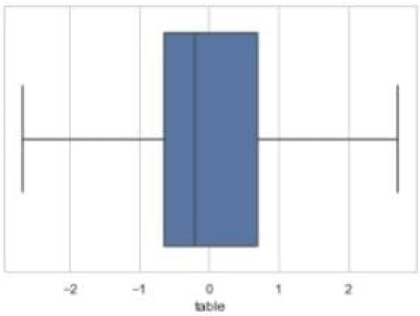

0 10 20 30 40
z

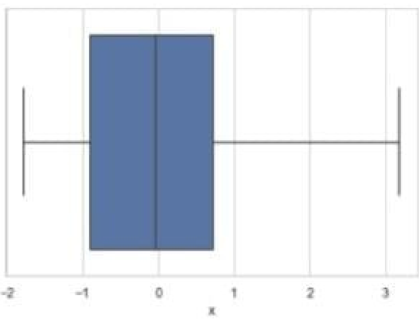


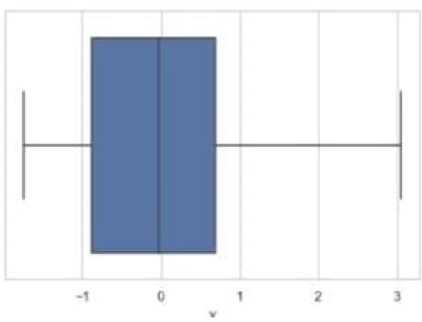
After Treating Outlier

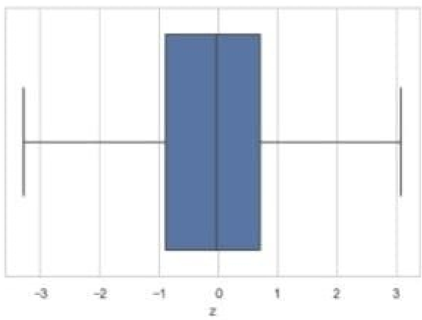


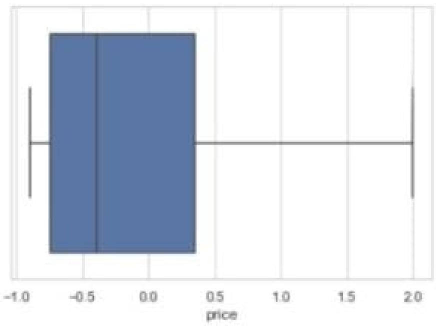












As we can see the ouliers are successfuly removed

Q uestion1.3 Encode the data (having string values) for Modelling. Data Split: Split the data into train and test (70:30). Apply Linear regression. Performance Metrics: Check the performance of Predictions on Train and Test sets using Rsquare, RMSE.

ENCODING THE STRING VALUES

GET DUMMIES

Data head after Converting Categorical variables into Dummy variables in data

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

5 rows × 25 columns

Data Columns after Converting Categorical variables into Dummy variables in data

```
Out[42]: Index(['Unnamed: 0', 'carat', 'depth', 'table', 'x', 'y', 'z', 'price',
       'cut_Good', 'cut_Ideal', 'cut_Premium', 'cut_Very Good', 'color_E',
       'color_F', 'color_G', 'color_H', 'color_I', 'color_J', 'clarity_IF',
       'clarity_SI1', 'clarity_SI2', 'clarity_VS1', 'clarity_VS2',
       'clarity_VVS1', 'clarity_VVS2'],
      dtype='object')
```

Dummies have been encoded.

Linear regression model does not take categorical values so that we have encoded categorical values to integer for better results.

DROPING UNWANTED COLUMNS

droping 'Unnamed:0' column as it is of no use in data set

```
Out[44]: Index(['carat', 'depth', 'table', 'x', 'y', 'z', 'price', 'cut_Good',
       'cut_Ideal', 'cut_Premium', 'cut_Very Good', 'color_E', 'color_F',
       'color_G', 'color_H', 'color_I', 'color_J', 'clarity_IF', 'clarity_SI1',
       'clarity_SI2', 'clarity_VS1', 'clarity_VS2', 'clarity_VVS1',
       'clarity_VVS2'],
      dtype='object')
```

Doing train test split and creating Linear Regression model

Doing train test split and creating Linear Regression model

The coefficients for each of the independent attributes

```
The coefficient for carat is 1.1009417847804501
The coefficient for depth is 0.005605143445570377
The coefficient for table is -0.013319500386804035
The coefficient for x is -0.30504349819633475
The coefficient for y is 0.30391448957926553
The coefficient for z is -0.13916571567987943
The coefficient for cut_Good is 0.09403402912977911
The coefficient for cut_Ideal is 0.1523107462056746
The coefficient for cut_Premium is 0.14852774839849378
The coefficient for cut_Very Good is 0.12583881878452705
The coefficient for color_E is -0.04705442233369822
The coefficient for color_F is -0.06268437439142825
The coefficient for color_G is -0.10072161838356786
The coefficient for color_H is -0.20767313311661612
The coefficient for color_I is -0.3239541927462737
The coefficient for color_J is -0.46858930275015803
The coefficient for clarity_IF is 0.9997691394634902
The coefficient for clarity_SI1 is 0.6389785818271332
The coefficient for clarity_SI2 is 0.42959662348315514
The coefficient for clarity_VS1 is 0.8380875826737564
The coefficient for clarity_VS2 is 0.7660244466083613
The coefficient for clarity_VVS1 is 0.9420769630114072
The coefficient for clarity_VVS2 is 0.9313670288415696
```


R squre on training data

0.9419557931252712

R square on testing data

0.9381643998102491

Checking RMSE Value on training and testing data

0.20690072466418796

RMSE on Testing data

0.21647817772382869

VIF values

```
carat ---> 33.35086119845924
depth ---> 4.573918951598579
table ---> 1.7728852812619
x ---> 463.5542785436457
y ---> 462.769821646584
z ---> 238.65819968687333
cut_Good ---> 3.6096181949437143
cut_Ideal ---> 14.34812508118844
cut_Premium ---> 8.623414379121153
cut_Very Good ---> 7.848451571723688
color_E ---> 2.371070464762613
```

We still find we have multi collinearity in the dataset, to drop these values to Lower level we can drop columns after doing stats model.

From stats model we can understand the features that do not contribute to the Model.

We can remove those features after that the Vif Values will be reduced.

Ideal value of VIF is less tha 5%.

BEST PARAMS SUMMARY

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.942
Model:                            OLS   Adj. R-squared:                  0.942
Method:                 Least Squares   F-statistic:                 1.330e+04
Date:                Sun, 01 Aug 2021   Prob (F-statistic):               0.00
Time:                        12:47:27   Log-Likelihood:                 2954.6
No. Observations:               18870   AIC:                            -5861.
Df Residuals:                   18846   BIC:                            -5673.
Df Model:                          23
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
Intercept        -0.7568      0.016    -46.999      0.000      -0.788      -0.725
carat             1.1009      0.009    121.892      0.000       1.083       1.119
depth             0.0056      0.004      1.525      0.127      -0.002       0.013
table            -0.0133      0.002     -6.356      0.000      -0.017      -0.009
x                -0.3050      0.032     -9.531      0.000      -0.368      -0.242
y                 0.3039      0.034      8.934      0.000       0.237       0.371
z                -0.1392      0.024     -5.742      0.000      -0.187      -0.092
cut_Good          0.0940      0.011      8.755      0.000       0.073       0.115
cut_Ideal         0.1523      0.010     14.581      0.000       0.132       0.173
cut_Premium       0.1485      0.010     14.785      0.000       0.129       0.168
cut_Very_Good     0.1258      0.010     12.269      0.000       0.106       0.146
color_E          -0.0471      0.006     -8.429      0.000      -0.058      -0.036
color_F          -0.0627      0.006    -11.075      0.000      -0.074      -0.052
color_G          -0.1007      0.006    -18.258      0.000      -0.112      -0.090
color_H          -0.2077      0.006    -35.323      0.000      -0.219      -0.196
color_I          -0.3240      0.007    -49.521      0.000      -0.337      -0.311
color_J          -0.4686      0.008    -58.186      0.000      -0.484      -0.453
clarity_IF        0.9998      0.016     62.524      0.000       0.968       1.031
clarity_SI1       0.6390      0.014     46.643      0.000       0.612       0.666
clarity_SI2       0.4296      0.014     31.177      0.000       0.403       0.457
clarity_VS1       0.8381      0.014     59.986      0.000       0.811       0.865
clarity_VS2       0.7660      0.014     55.618      0.000       0.739       0.793
clarity_VVS1      0.9421      0.015     63.630      0.000       0.913       0.971
clarity_VVS2      0.9314      0.014     64.730      0.000       0.903       0.960
==============================================================================
Omnibus:                     4696.785   Durbin-Watson:                   1.994
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            17654.853
Skew:                           1.208   Prob(JB):                         0.00
Kurtosis:                       7.076   Cond. No.                         57.0
```

To ideally bring down the values to lower levels we can drop one of the variable that is highly correlated.

Dropping variables would bring down the multi collinearity level down.

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

We had a business problem to predict the price of the stone and provide insights for the company on the profits on different prize slots.

From the EDA analysis we could understand the cut, ideal cut had number profits to the company. The colours H, I, J have bought profits for the company.

In clarity if we could see there were no flawless stones and there were no profits coming from I1, I2, I3 stones. The ideal, premium and very good types of cut were bringing profits where as fair and good are not bringing profits.

The predictions were able to capture 95% variations in the price and it is explained by the predictors in the training set.

Using stats model if we cam run the model again we can have P values and coefficients which will give us better understanding of the relationship, so that values more 0.05 we can drop those variables and re run the model again for better results.

For better accuracy drop depth column in iteration for better results.

The equation,

(-0.76) *Intercept* + *(1.1)* carat + (-0.01) *table* + *(-0.32)* x + (0.28) *y* + *(-0.11)* z + (0.1) *cut_Good* + *(0.15)* 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 + (-0.32) *color_I* + *(-0.47)* color_J + (1.0) *clarity_IF* + *(0.64)* clarity_SI1 + (0.43) *clarity_SI2* + *(0.84)* clarity_VS1 + (0.77) *clarity_VS2* + *(0.94)* clarity_VVS1 + (0.93) * clarity_VVS2 +

Recommendations

1. The ideal, premium, very good cut types are the one which are bringing profits so that we could use marketing for these to bring in more profits.
2. The clarity of the diamond is the next important attributes the more the clear is the stone the profits are more

The best attributes are:

Carat,

Y the diameter of the stone,

clarity_IF,

clarity_SI1,

clarity_SI2,

clarity_VS1,

clarity_VS2,

clarity_VVS1,

clarity_VVS2

Problem 2: Logistic Regression and LDA

Problem Statement:

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:

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

Question 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.

Loading all the necessary library for the model building.

Now, reading the head and tail of the dataset to check whether data has been properly fed

HEAD OF THE DATA

Out[75]:

	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

TAIL OF THE DATA

Out[76]:

	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

Shape of the data

(872, 8)

Checking data info

```
<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
```

observation :

- 8 variables and 872 records.
- No missing record based on initial analysis.
- two object variables and six numaric variables.
- variable "Unnamed: 0" seems useless variable.

Getting data Discription

Out[79]:

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	872.0	436.500000	251.869014	1.0	218.75	436.5	654.25	872.0
Salary	872.0	47729.172018	23418.668531	1322.0	35324.00	41903.5	53469.50	236961.0
age	872.0	39.955275	10.551675	20.0	32.00	39.0	48.00	62.0
educ	872.0	9.307339	3.036259	1.0	8.00	9.0	12.00	21.0
no_young_children	872.0	0.311927	0.612870	0.0	0.00	0.0	0.00	3.0
no_older_children	872.0	0.982798	1.086786	0.0	0.00	1.0	2.00	6.0

Observation:

- Based on summary descriptive, the data looks good.
- We see for most of the variable,mean/median are nearly equal.
- Std Deviation is high for Salary variable.

Checking Null Values

```
Out[80]: Unnamed: 0           0
         Holliday_Package     0
         Salary               0
         age                  0
         educ                 0
         no_young_children    0
         no_older_children    0
         foreign              0
         dtype: int64
```

No null values found in the data.

Checking for Duplicate Data

No duplicate data found.

Getting unique values of all the categorical variables

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

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

Holliday_Package variable have two categories Yes/no with 401 yes and 471 no also variable foreign have two categories Yes/no with 216 yes and 656 no.

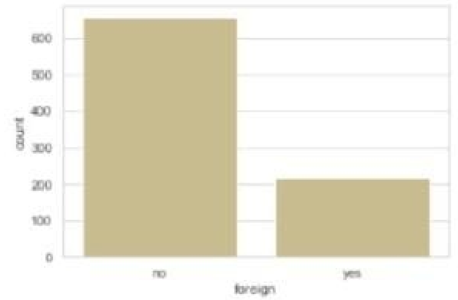
Checking skewness

```
Out[89]: Salary               3.103216
         no_young_children    1.946515
         no_older_children    0.953951
         age                  0.146412
         Unnamed: 0           0.000000
         educ                -0.045501
         dtype: float64
```

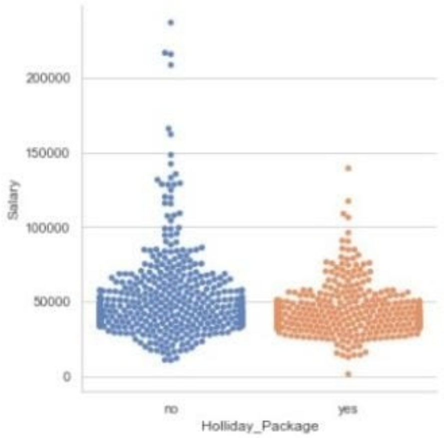
CATEGORICAL UNIVARIATE ANALYSIS

Holliday Package

Foreign



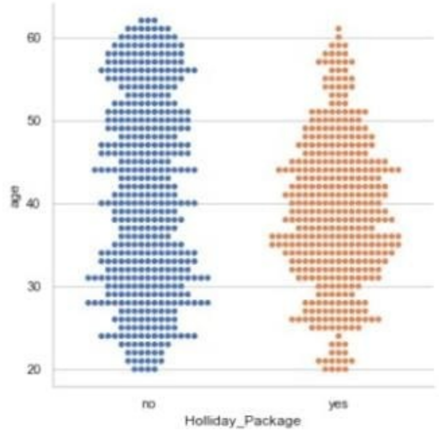
HOLIDAY PACKAGE VS SALARY



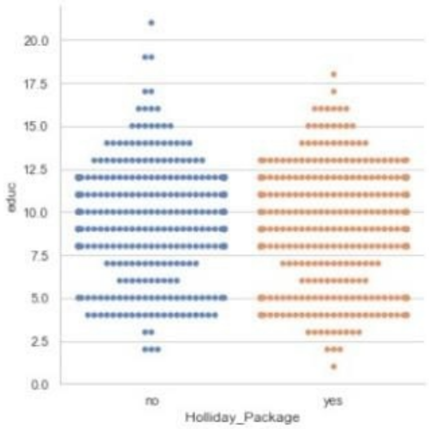
Observation:

We can see employees below salary 150000 have always opted for holiday package.

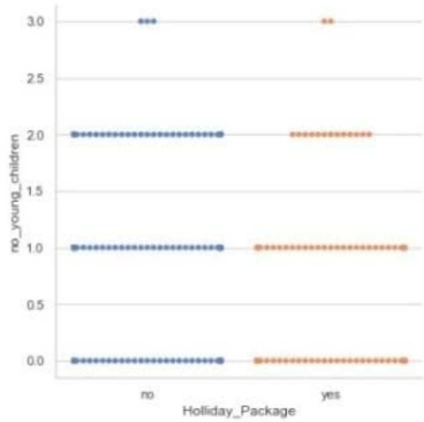
HOLIDAY PACKAGE VS AGE



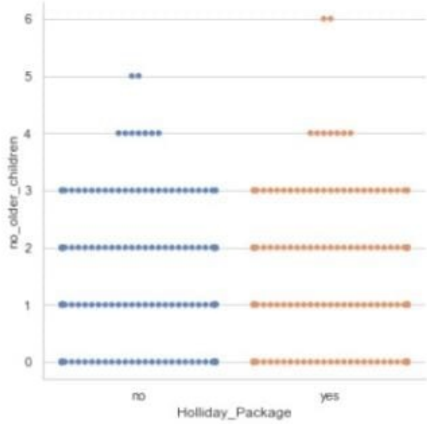
HOLIDAY PACKAGE VS EDUC



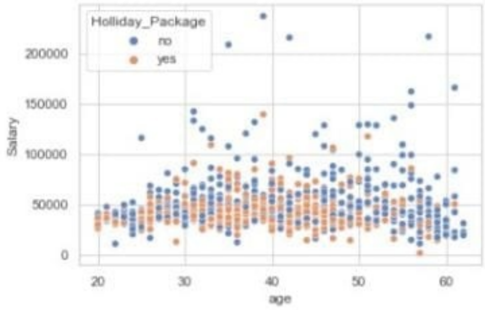
HOLIDAY PACKAGE VS YOUNG CHILDREN



HOLIDAY PACKAGE VS OLDER CHILDREN



AGE VS SALARY VS HOLIDAY PACKAGE

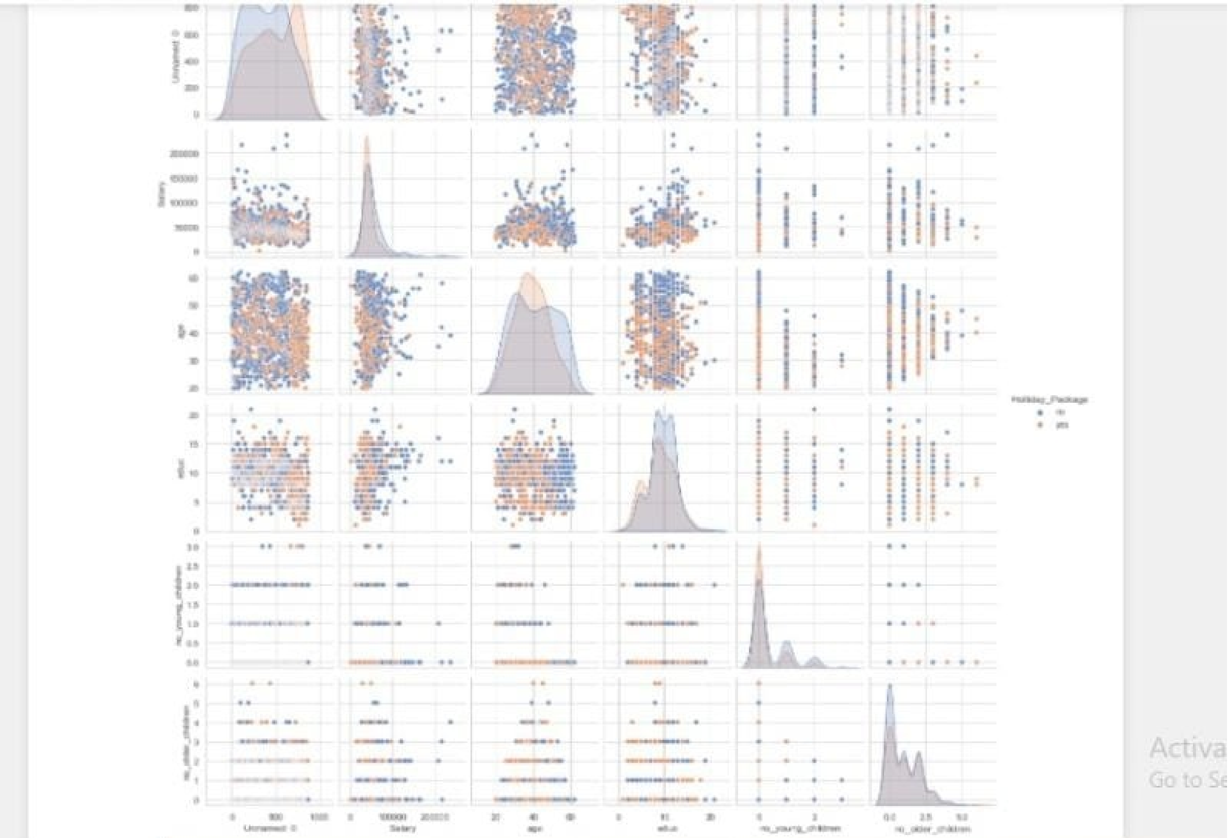


Obseravtion:

Employee age over 50 to 60 have seems to be not taking the holiday package, whereas in the age 30 to 50 and salary less than 50000 people have opted more for holiday package

Bivariate Analysis

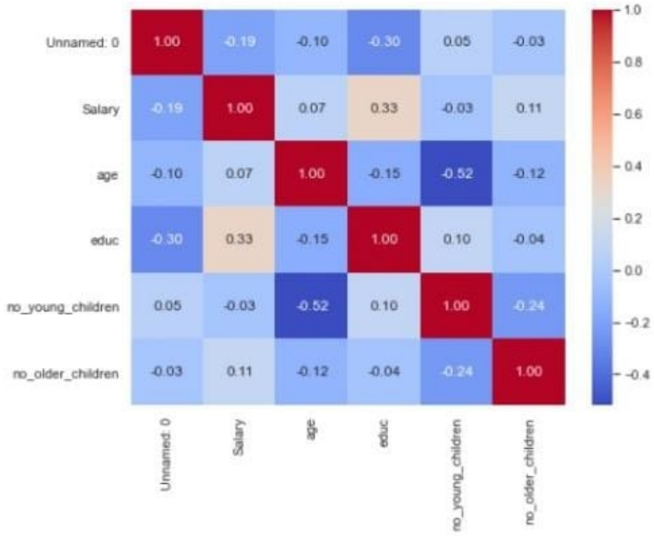
Data Distribution



There is no correlation between the data, the data seems to be normal.

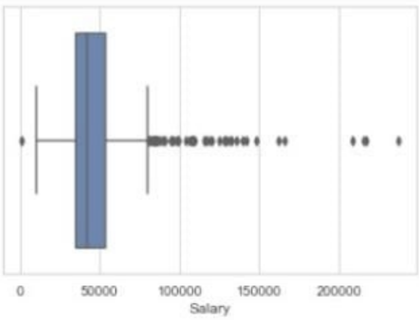
There is no huge difference in the data distribution among the holiday package, I don’t see any clear two different distribution in the data

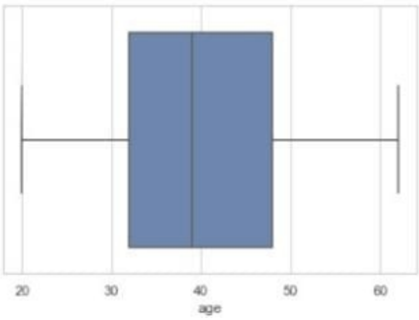
Correlation Matrix



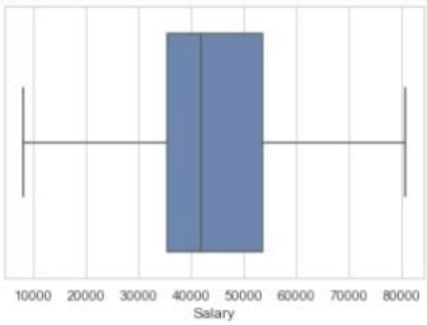
No multi collinearity in the data

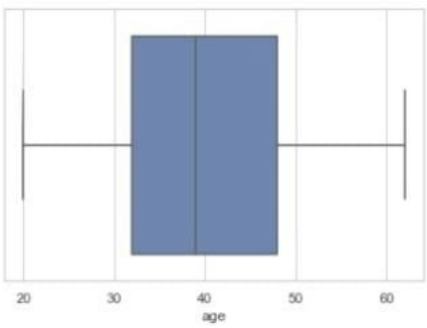
Checking Outliers before Outlier Treatment





Boxplots after Outlier Treatment





Observation :

No outliers in the data, all the outliers have been treated.

Question 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).

Converting categorical variable in to dummy variable in data1

Data head after Converting Categorical variables into Dummy variables in data

Out[108]:

	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

The encoding helps the logistic regression model predict better results

Data Columns

```
Out[109]: Index(['Salary', 'age', 'educ', 'no_young_children', 'no_older_children',
       'Holliday_Package_yes', 'foreign_yes'],
      dtype='object')
```

Doing Train / Test split and fitting model on the data

LogisticRegression(max_iter=10000, n_jobs=2, penalty='none', solver='newton-cg',

```
                   verbose=True)
```

Getting probabilties on test set

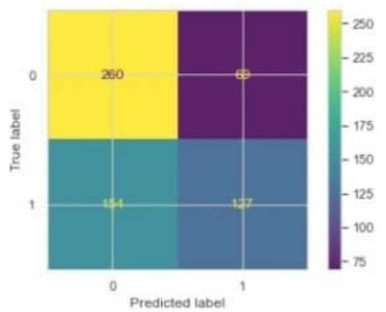
Out[116]:

	0	1
0	0.640764	0.359236
1	0.569909	0.430091
2	0.655265	0.344735
3	0.564147	0.435853
4	0.538869	0.461131

Classification Report and Confusion Matrix on training data

```
              precision    recall  f1-score   support

           0       0.63      0.79      0.70       329
           1       0.65      0.45      0.53       281

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

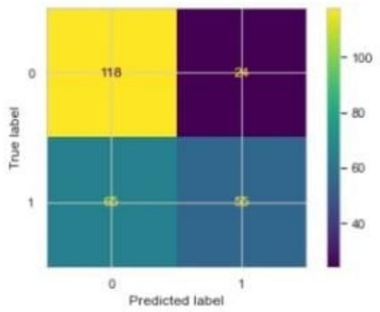
Out[118]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x27d71d40460>



Classification Report and Confusion Matrix on testing data

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

```
Out[119]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x27d71d6e580>
```

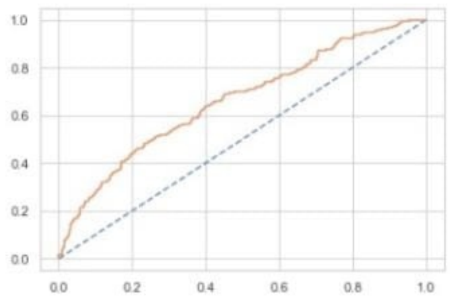


Accuracy - training data

0.6344262295081967

AUC and ROC for the training data

AUC: 0.661

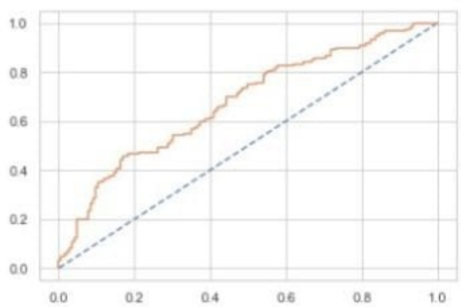


Accuracy - test data

0.6603053435114504

AUC and ROC for the test data

AUC: 0.675



BUILDING LDA MODEL

Training Data Class Prediction with a cut-off value of 0.5

Test Data Class Prediction with a cut-off value of 0.5

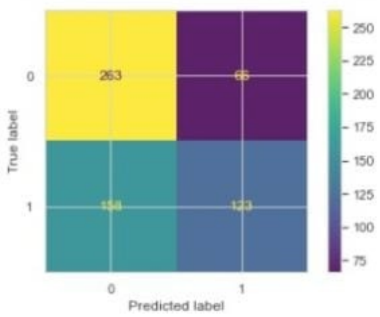
Checking train model Score on train data

0.6327868852459017

Creating Confusion Matrix on train data

array([[263, 66],

```
[158, 123]], dtype=int64)
```



Classification report train data

	precision	recall	f1-score	support
0	0.62	0.80	0.70	329
1	0.65	0.44	0.52	281
accuracy			0.63	610
macro avg	0.64	0.62	0.61	610
weighted avg	0.64	0.63	0.62	610

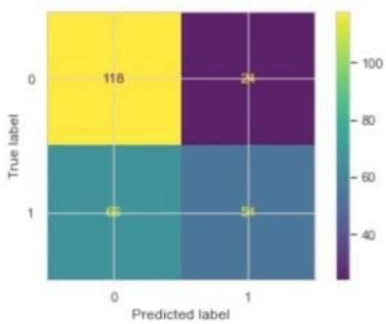
Checking test model Score on test data

0.6564885496183206

Creating Confusion matrix on test data

array([[118, 24],

```
[ 66,  54]], dtype=int64)
```

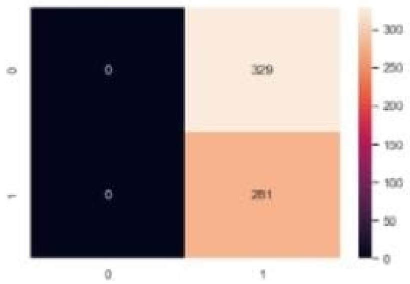


Classification report test data

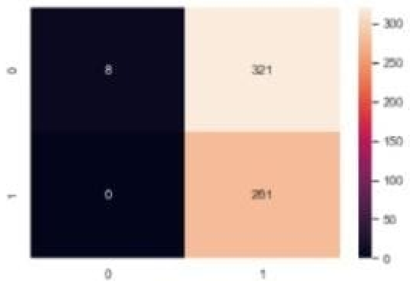
	precision	recall	f1-score	support
0	0.64	0.83	0.72	142
1	0.69	0.45	0.55	120
accuracy			0.66	262
macro avg	0.67	0.64	0.63	262
weighted avg	0.66	0.66	0.64	262

Changing the cutt off value to check optimal value that gives better Accuracy and F1 Score.

```
Accuracy Score 0.4607
F1 Score 0.6308

Confusion Matrix
```




```
0.2

Accuracy Score 0.4738
F1 Score 0.6365

Confusion Matrix
```



AUC AND ROC curve for training and testing data

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

Out[151]:

	LR Train	LR Test	LDA Train	LDA Test
Accuracy	0.63	0.66	0.63	0.66
AUC	0.66	0.67	0.66	0.68
Recall	0.45	0.46	0.44	0.45
Precision	0.65	0.70	0.65	0.69
F1 Score	0.53	0.55	0.52	0.55

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

Please explain and summarise the various steps performed in this project. There should be proper business interpretation and actionable insights present.

We had a business problem where we need predict whether an employee would opt for a holiday package or not, for this problem we had done predictions both logistic regression and linear discriminant analysis. Since both results are same.

The EDA analysis clearly indicates certain criteria where we could find people age above 50 are not interested much in holiday packages.

So, we found aged people are not opting for holiday packages.

People ranging from the age 30 to 50 generally opt for holiday packages.

Employee age over 50 to 60 have seems to be not taking the holiday package, whereas in the age 30 to 50 and salary less than 50000 people have opted more for holiday package.

The important factors deciding the predictions are salary, age and educ.

Recommendations :

1. To improve holiday packages over the age above 50 we can provide religious destination places.
2. For people earning more than 150000 we can provide vacation holiday packages.
3. For employee having more than number of older children we can provide packages in holiday vacation places.

In [ ]: