PREDICTIVE MODELING

BUSINESS REPORT

THAKUR ARUN SINGH

JUNE 2021

This Business Report shall provide detailed explanation of how we approached each problem given in the assignment. It shall also provide relative resolution and explanation with regards to the problems

CONTENTS

Problem 1:..... 2

Problem 1.1 2

Problem 1.2 13

Problem 1.3 20

Problem 1.4 23

Problem 2:..... 24

Problem 2.1 24

Problem 2.2 38

Problem 2.3 42

Problem 2.4 48

Problem 1:

You are hired by a company Gem Stones co ltd, which is a cubic zirconia manufacturer. You are provided with the dataset containing the prices and other attributes of almost 27,000 cubic zirconia (which is an inexpensive diamond alternative with many of the same qualities as a diamond). The company is earning different profits on different prize slots. You have to help the company in predicting the price for the stone on the bases of the details given in the dataset so it can distinguish between higher profitable stones and lower profitable stones so as to have better profit share. Also, provide them with the best 5 attributes that are most important.

PROBLEM 1.1

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.

Resolution:

Describing the data:

- First we import all the necessary libraries in Python, and then import the data file which is 'cubic_zirconia'. Once we import the file we confirm whether the data has been uploaded correctly or not using 'head' function. Using this function we can view the data and all the columns and headers whether they are aligning correctly or not.
- Then using the 'shape' function we can understand how many row and columns are there in our data set.
- To check the data type of all the columns and also to check the null values, 'info' function. Has been used.
- To see the detail description of the data such as, Count, Mean, Median, Min, Max, Standard Deviations etc,

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	26967	13484	7784.847	1	6742.5	13484	20225.5	26967
carat	26967	0.798375	0.477745	0.2	0.4	0.7	1.05	4.5
depth	26270	61.74515	1.41286	50.8	61	61.8	62.5	73.6
table	26967	57.45608	2.232068	49	56	57	59	79
x	26967	5.729854	1.128516	0	4.71	5.69	6.55	10.23
y	26967	5.733569	1.166058	0	4.71	5.71	6.54	58.9
z	26967	3.538057	0.720624	0	2.9	3.52	4.04	31.8
price	26967	3939.518	4024.865	326	945	2375	5360	18818

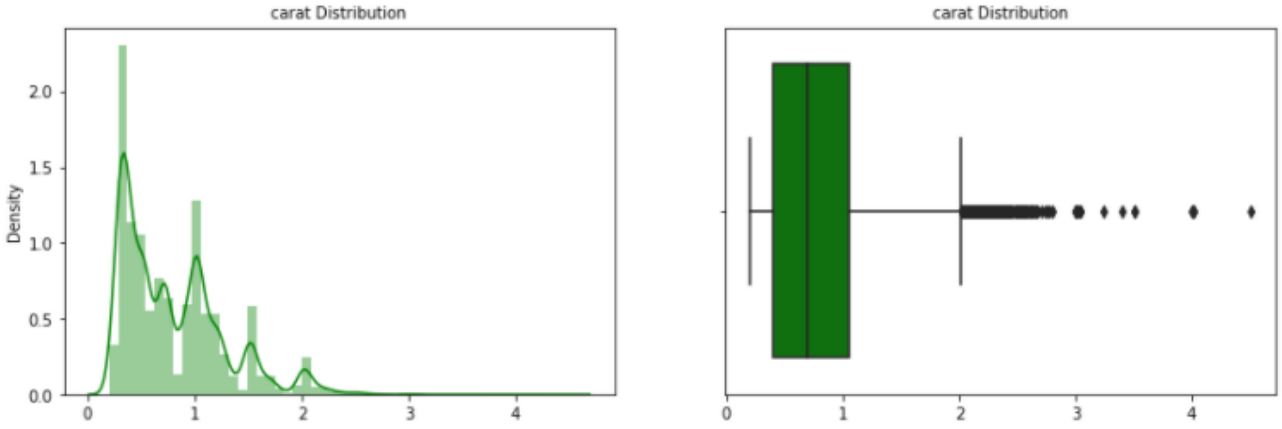
- Using the 'isnull' function, one can understand if there are any null values in the data set. And we do not have any null values in the existing data set.
- Using the 'dups' function we check for the duplicates and there were no duplicate values.
- We have both categorical and continuous data. For categorical data, we have we have cut, color and clarity. For continuous data, we have carat, depth, table and price.

- We also identified the unique values in categorical data.

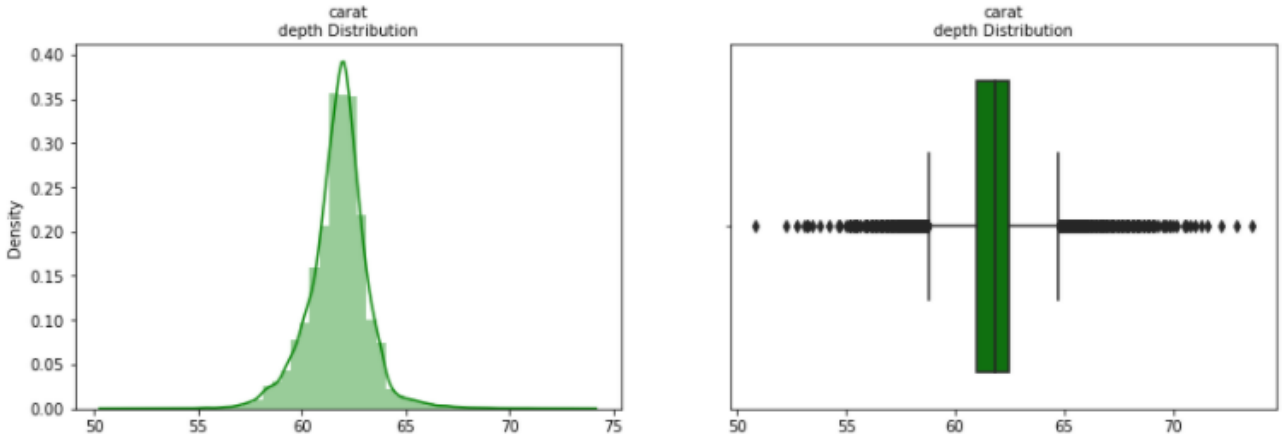
After reviewing the data thoroughly, and based on the above analysis we can say that, we have seven variables, Mean and Median values are almost equal, and Standard deviation for ‘Spending’ is higher than other variables. There are no duplicates in the data set.

Exploratory data analysis

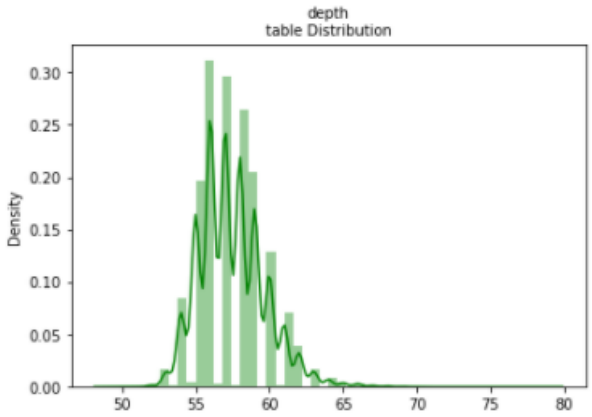
Univariate and multivariate analysis

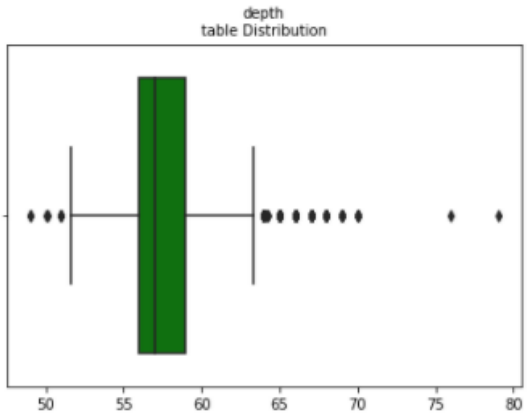


The distribution of data seems to be positively skewed as there are multiple peak points in the distribution there could be multimode and the box plot of carat seems to have large number of outliers .In the range of 0 to 1, where majority of data resides.

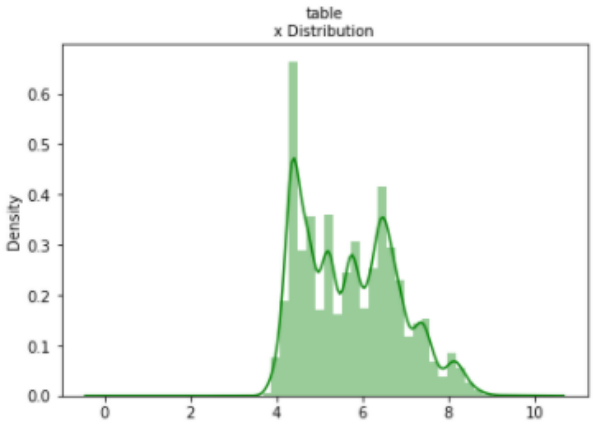


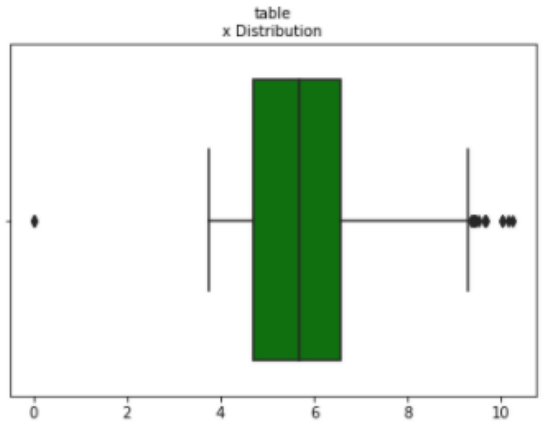
The above dist plot shows the normal distribution of data from 55 – 65. Boxplot shows that there are lot of outliers.



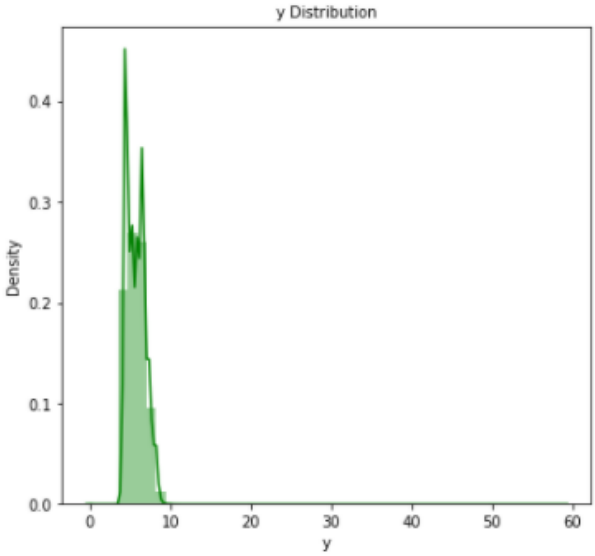


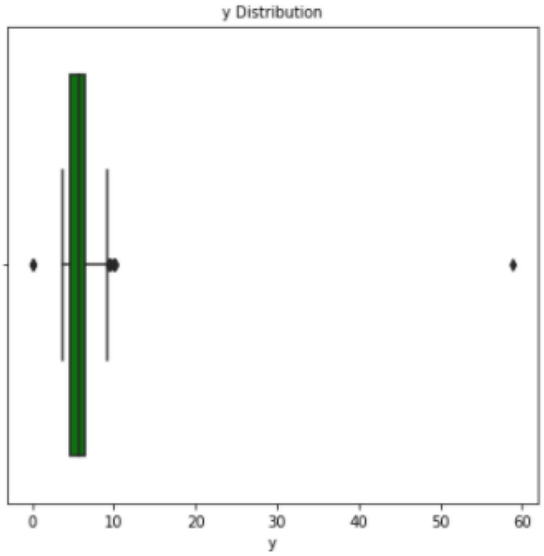
The above dist plot shows the distribution of data from 55 – 65 and is positively skewed. Boxplot shows that there are a few outliers.



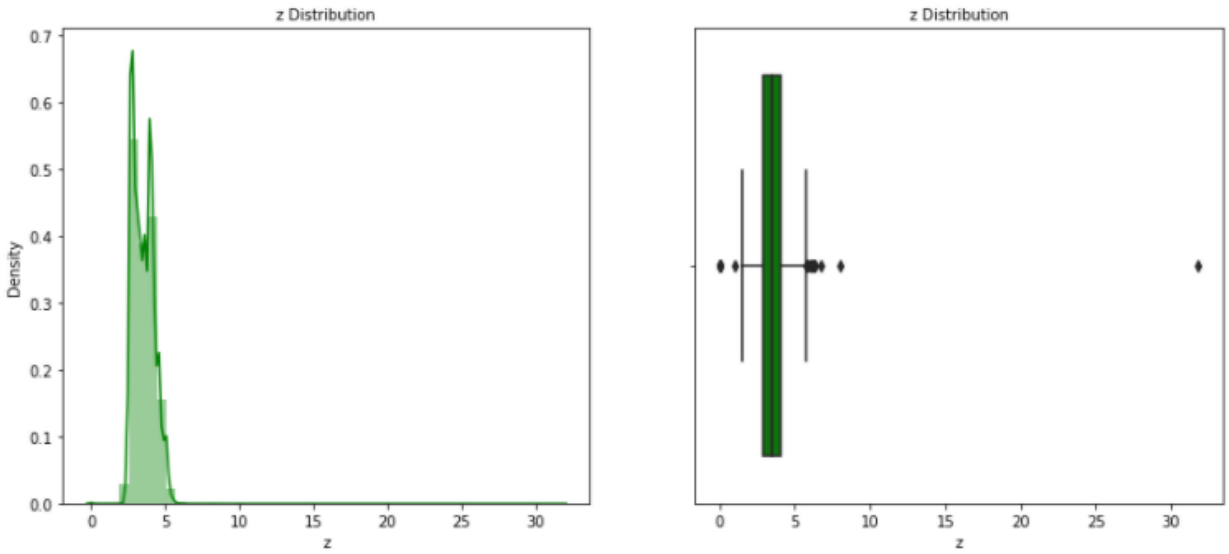


The above dist plot shows the distribution of data from 4 – 8 and is positively skewed. Boxplot shows that there are lot of outliers.

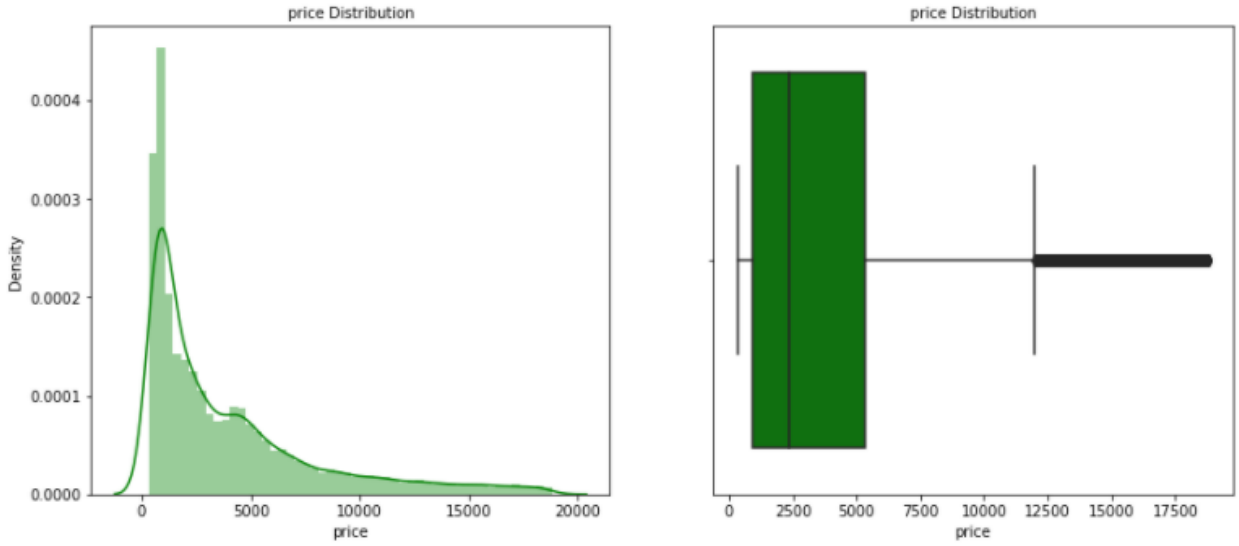




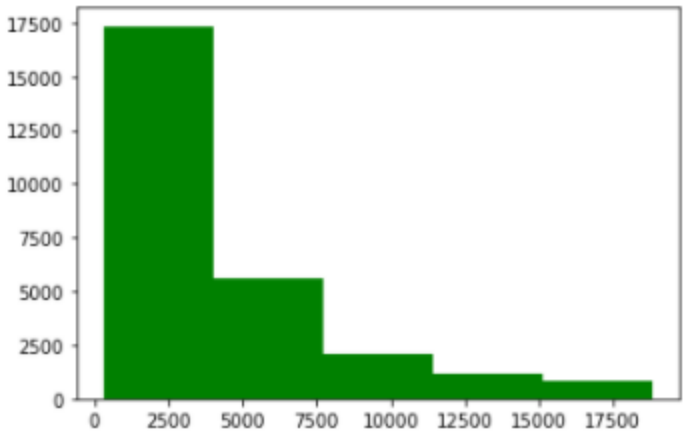
The above dist plot shows the distribution of data from 5 – 10 and is positively skewed. Boxplot shows that there are few outliers. The distribution is too much positively skewed. The skewness maybe due to diamonds are always made in specific shape. There might not be too many sizes in the market.



The above dist plot shows the distribution of data from 2 – 5 and is positively skewed. Boxplot shows that there are a few outliers. The skewness may be due to diamonds that are made in specific shape. There may not be too many sizes in the market.



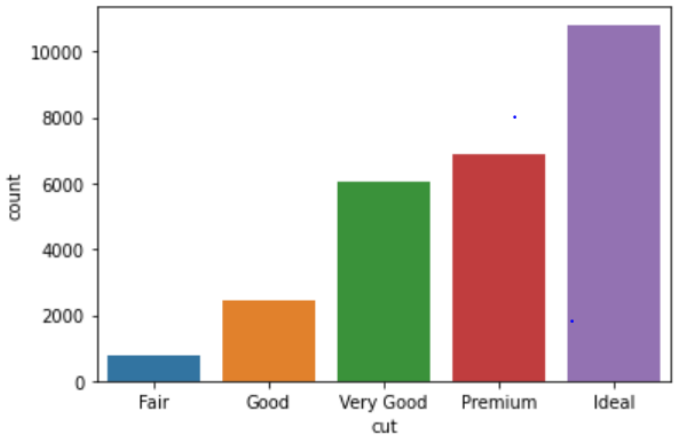
The above dist plot shows the distribution of data from 100 – 8000 and is positively skewed. Boxplot shows that there are a lot of outliers.



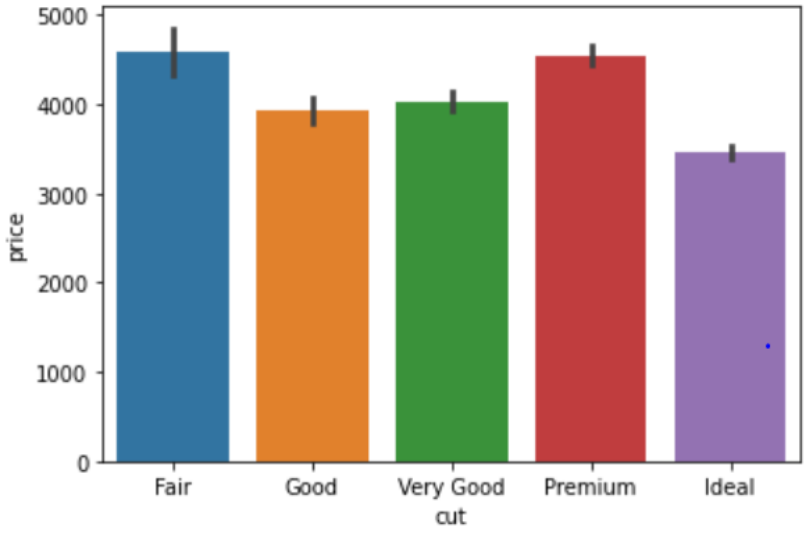
The above histogram shows the price.

Bi – Variate Analysis:

Quality is increasing order fair, good , very good , premium, ideal.

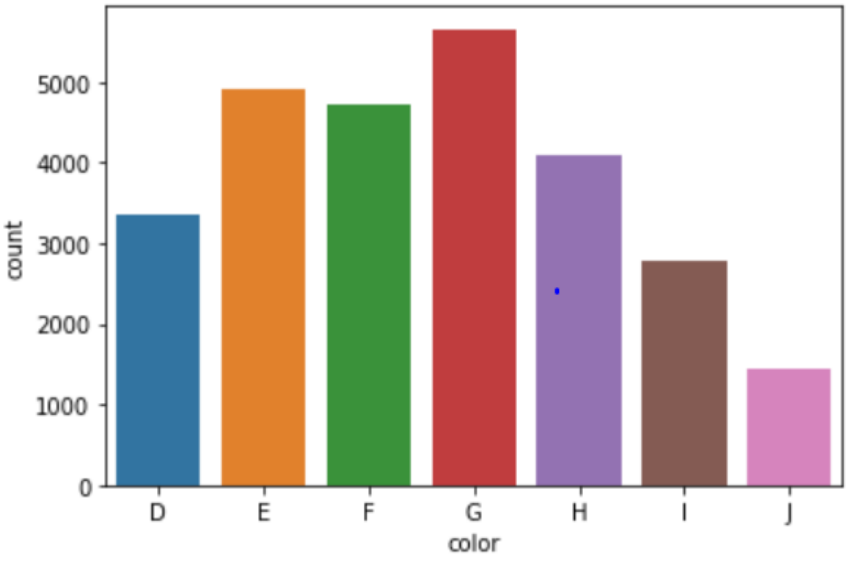


The most preferred cut seems to be ideal for diamonds



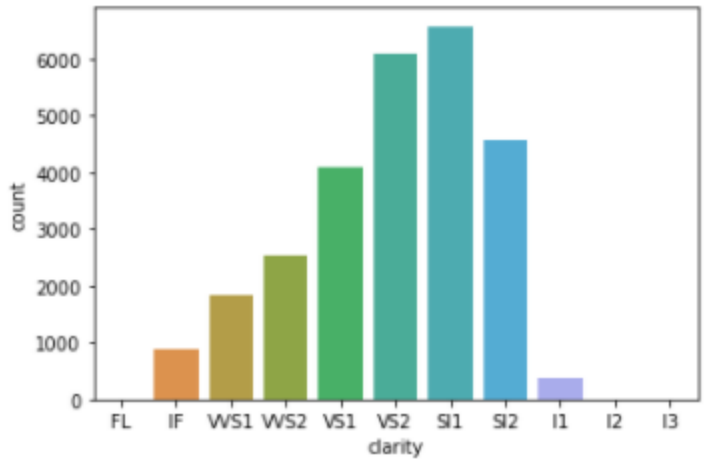
The reason for the most preferred cut ideal is because those diamonds are priced lower than the other cuts.

'D' being the best and J the worst.

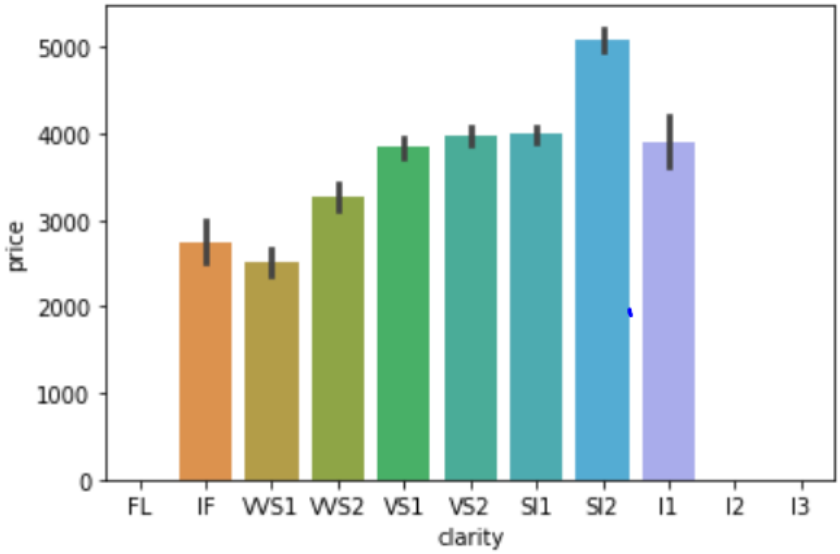


We see the G is priced in the middle of the seven colours, where J being the worst colour price seems to be high.

Best to worst, FL-> flawless, I3-> level 3 inclusions)FL,IF,VVS1,VVS2,VS1, VS2,SI1,SI2,I1,I2,I3



the clarity VS2 seems to be preferred by people.

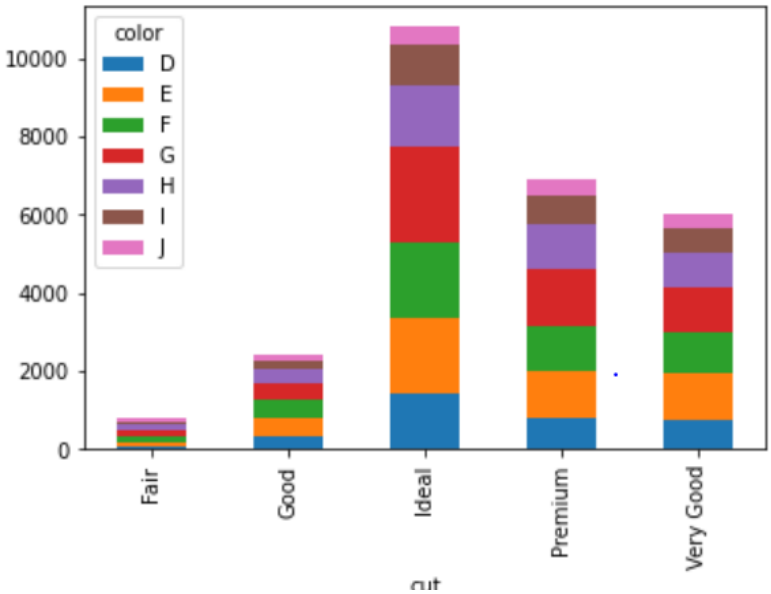


The clarity VS2 seems to be preferred by people.

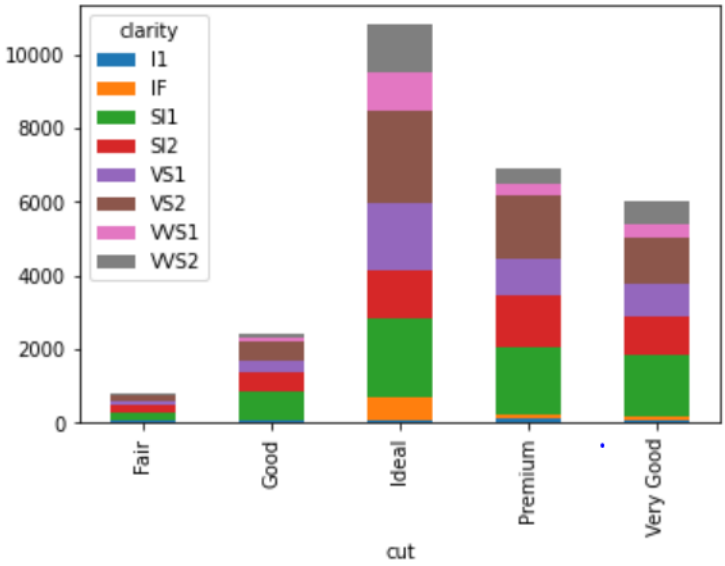
The data has no FL diamonds from this we can determine the flawless diamonds are not bringing any profits to the store.

Relationship between categorical variables.

Cut and color:

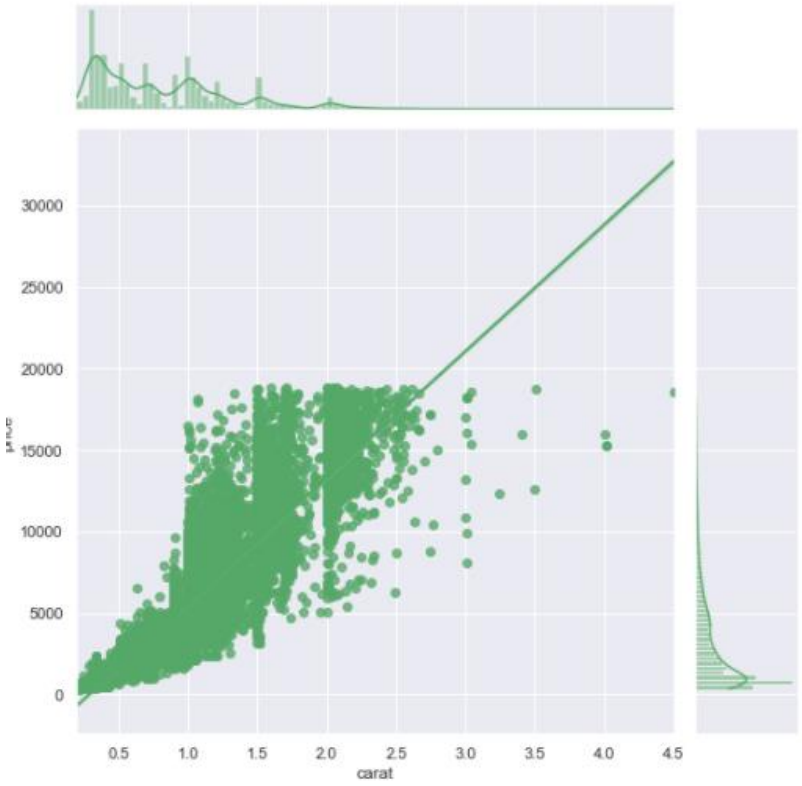


Cut and Clarity

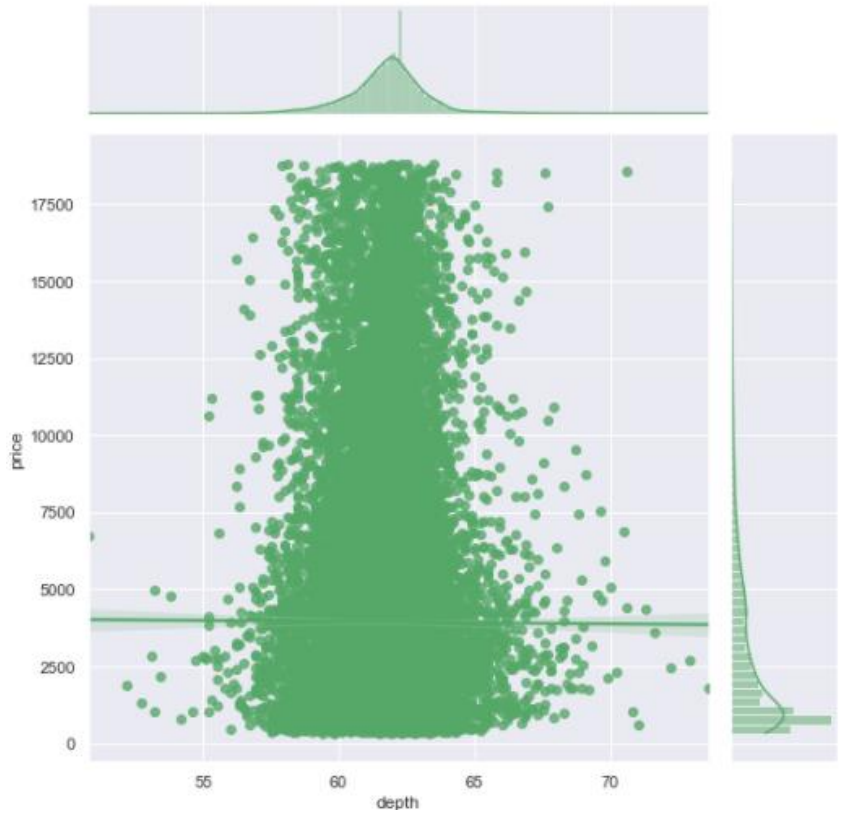


Correlation:

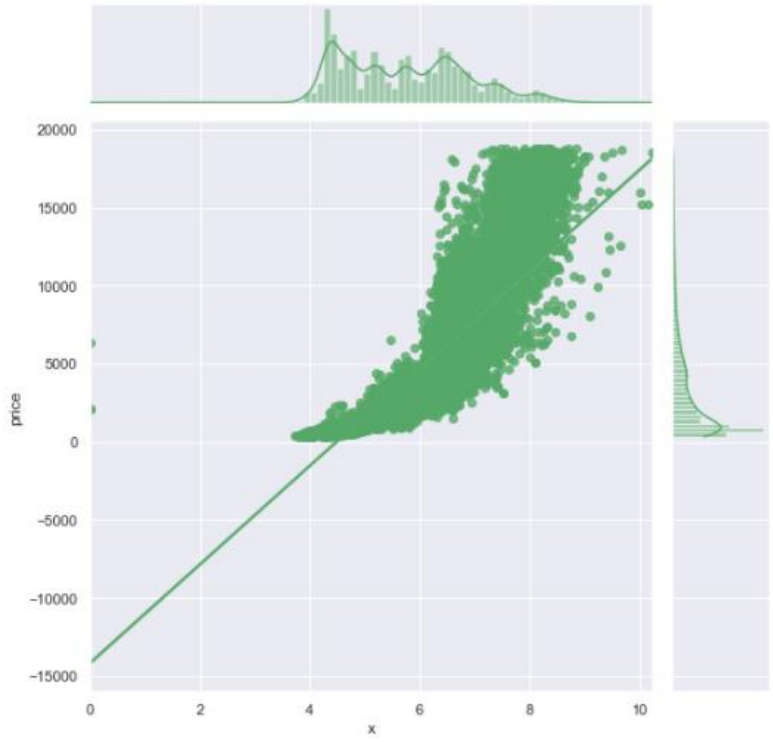
CARAT V/S Price:



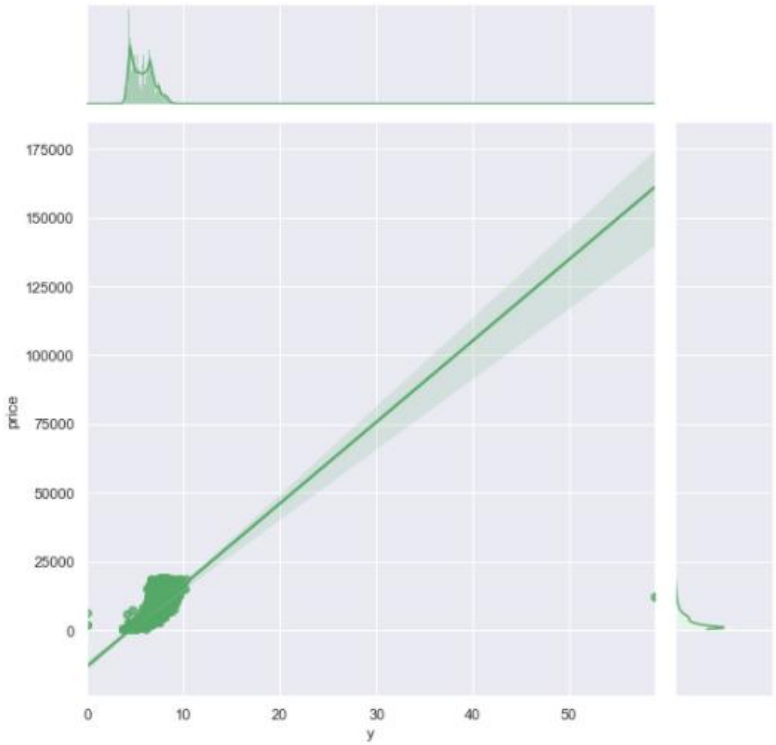
Depth V/S Price



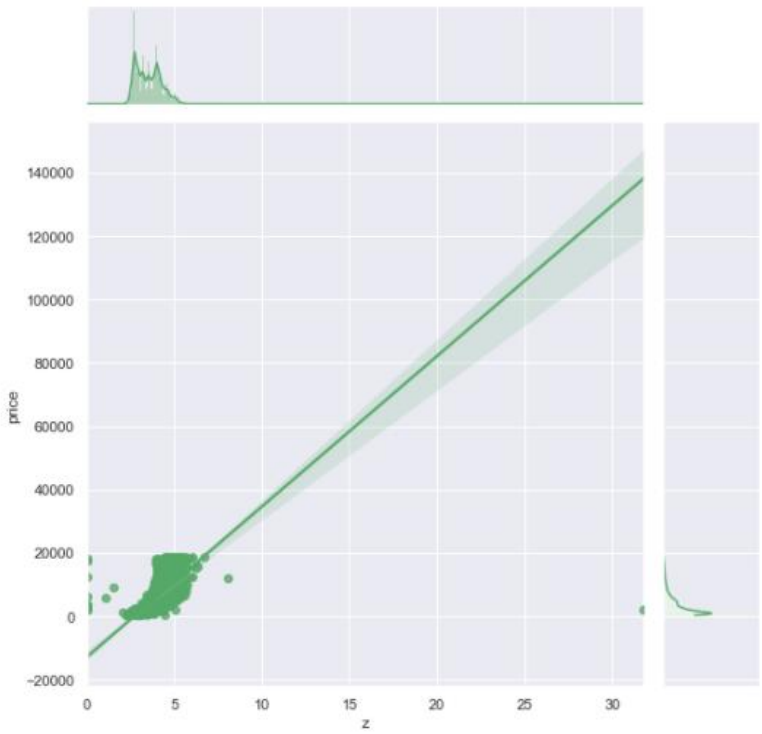
X V/S Price



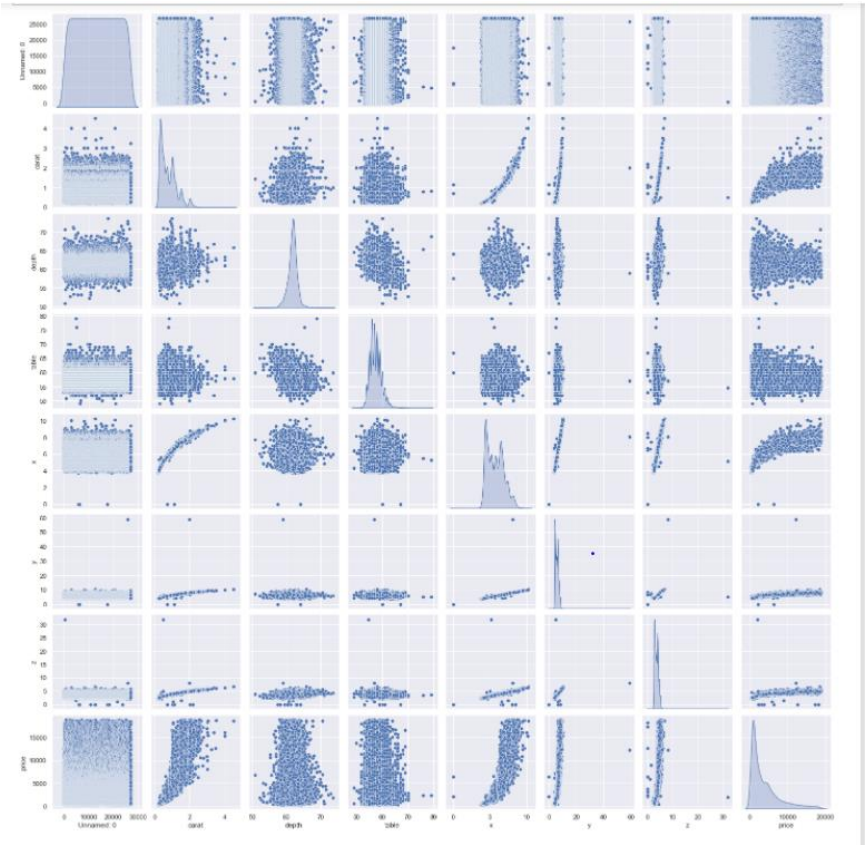
Y V/S Price



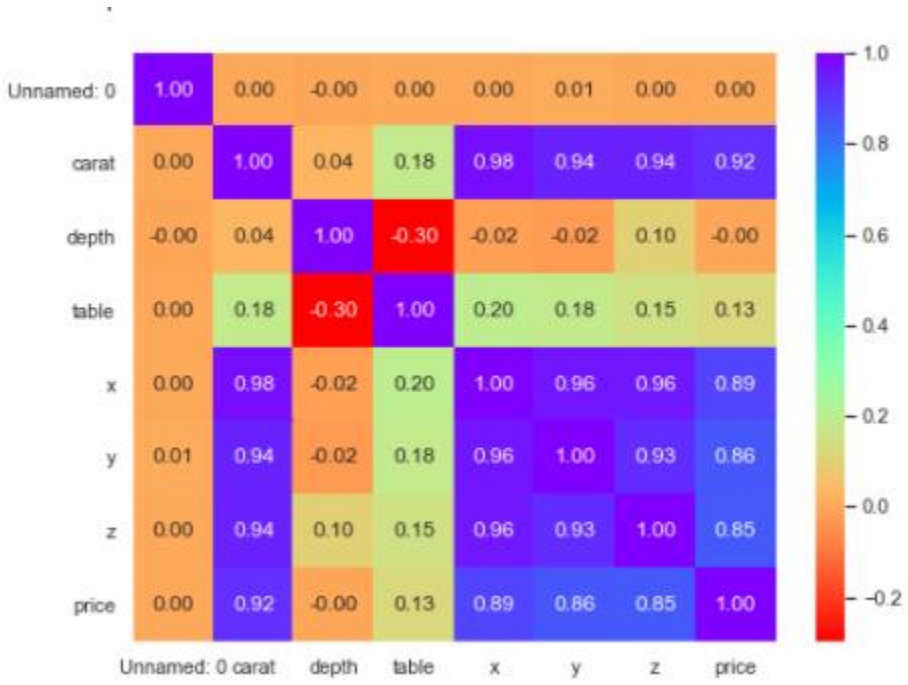
Z V/S Price



Data Distribution



Correlation Matrix



The matrix shows the presence of multi-collinearity in the dataset.

From both the analysis we can see that there is strong positive correlation between all the variables.

PROBLEM 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?

Resolution:

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

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

As per the above screen shot, there are null values since depth being continuous, carriable 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 dataset.

```
5821   x    0
       y    0
       z    0
6034   z    0
6215   x    0
       y    0
       z    0
10827  z    0
12498  z    0
12689  z    0
17506  x    0
       y    0
       z    0
18194  z    0
23758  z    0
dtype: object
```

Checking value that is 0

	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 which has zero the x,y and z are the dimensions of a diamond of a diamond so this can't take into model as these are very less rows.

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

Scaling:

Scaling can be reduced to check the multi-collinearity in the data.so if scaling is not applied we find the VIF- variance inflation factor values which are high, which indicates the presence of multi-collinearity.

These values are calculated after building the model of linear regression. To understand the multi-collieearity in the model. The scaling has no impact in model score or coeffients of neither attributes nor the intercept.

Before scaling:

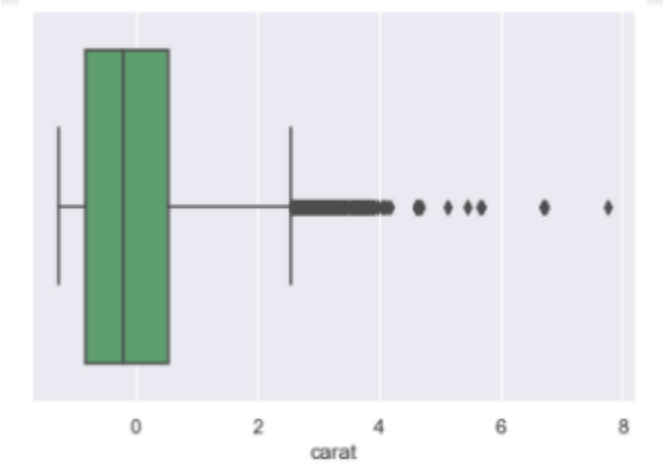
```
carat ---> 124.32595405062301
depth ---> 1407.6352441517224
table ---> 1002.8676766903022
x ---> 12004.212489729716
y ---> 11533.491914672948
z ---> 3442.374035538099
cut_Good ---> 4.5067464355335405
cut_Ideal ---> 18.17410430875144
cut_Premium ---> 10.884031423492264
cut_Very Good ---> 10.062010659328736
color_E ---> 2.479875675651354
```

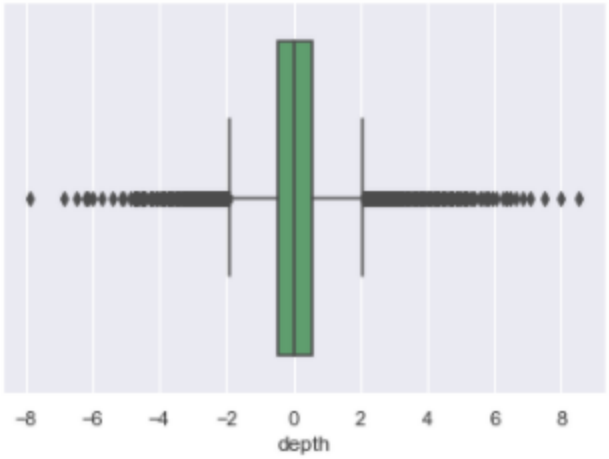
After Scaling

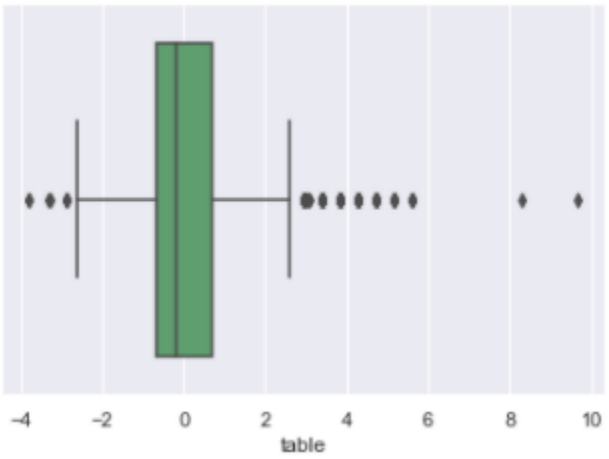
```
carat ---> 33.35287649550623
depth ---> 4.574003842337535
table ---> 1.7722022611198975
x ---> 463.94494858728734
y ---> 463.08309600508517
z ---> 238.6002431605187
cut_Good ---> 3.6104961328079184
cut_Ideal ---> 14.347409690217962
cut_Premium ---> 8.623207030351887
cut_Very Good ---> 7.852218650260111
color_E ---> 2.371053795458172
```

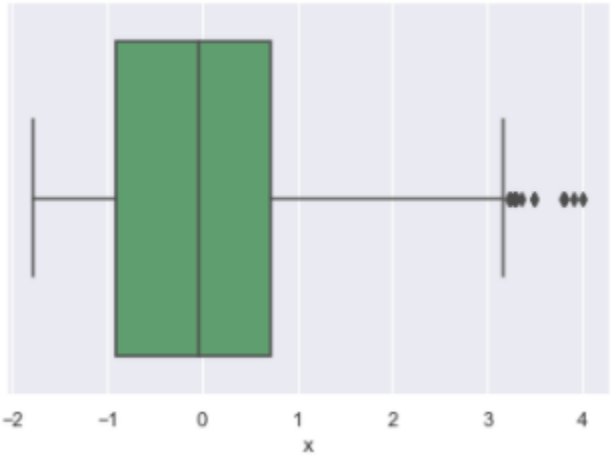
Checking for outliers in the data.

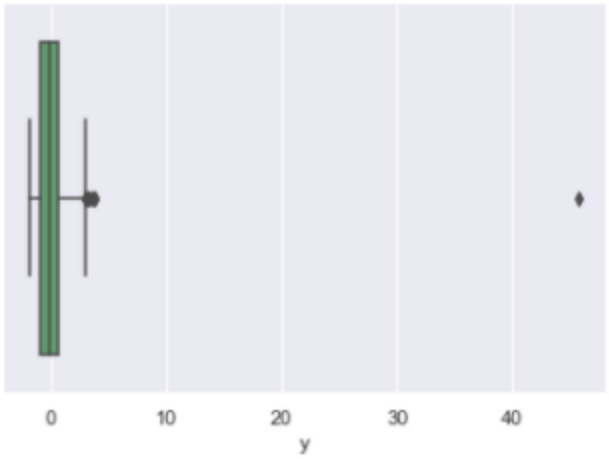
Before treating outliers

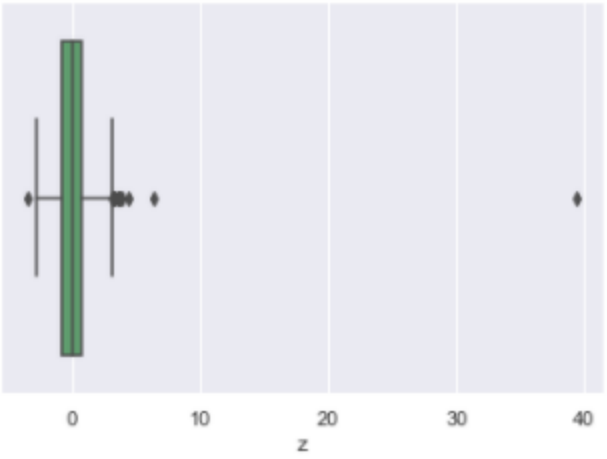


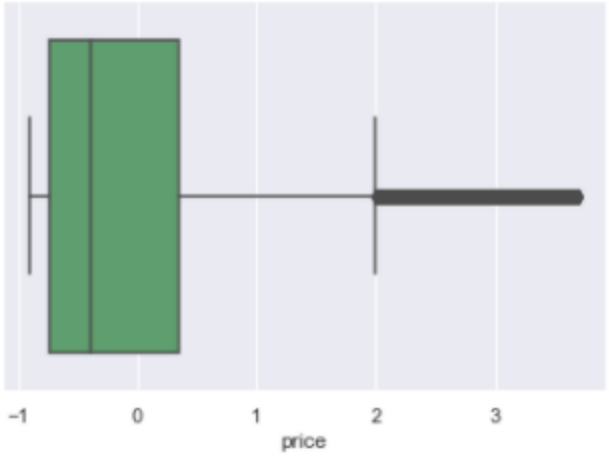

−8
−6
−4
−2
0
2
4
6
8
depth


−4
−2
0
2
4
6
8
10
table

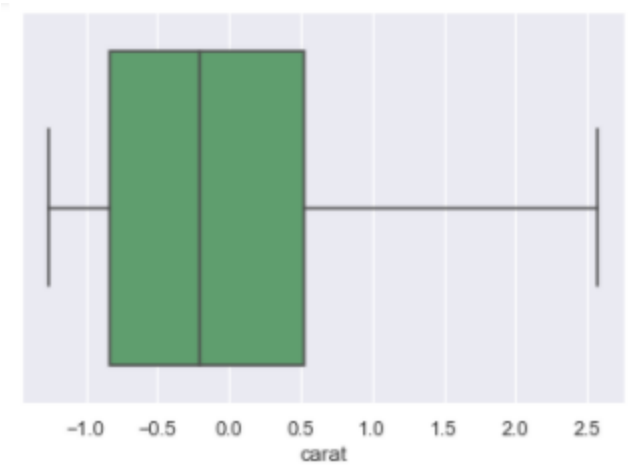

−2
−1
0
1
2
3
4
x

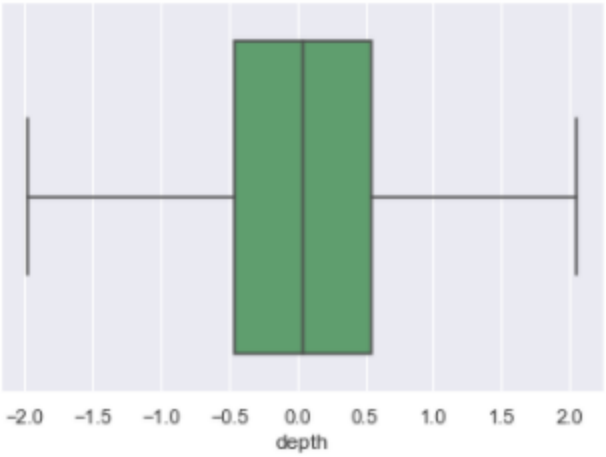


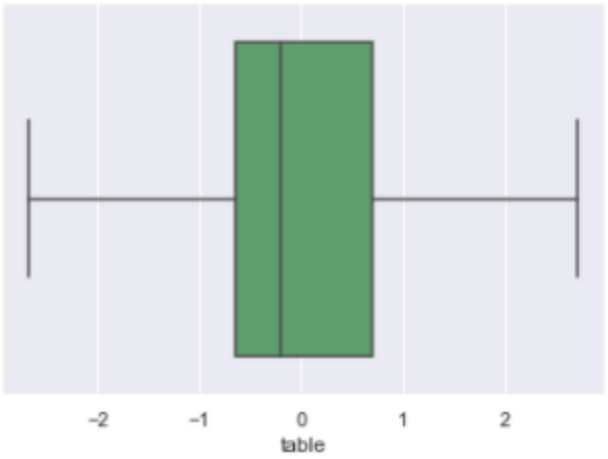


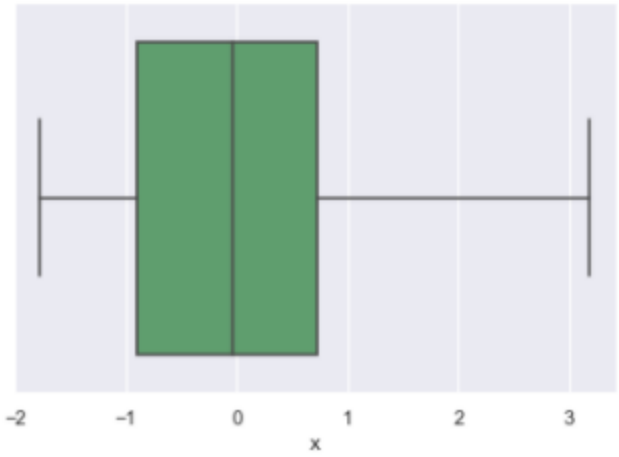


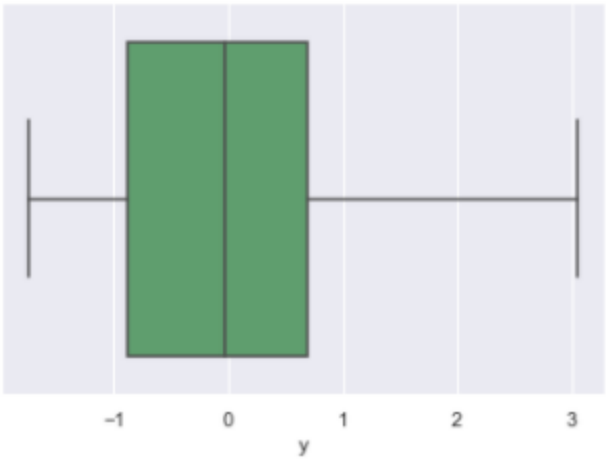
After Treating Outliers

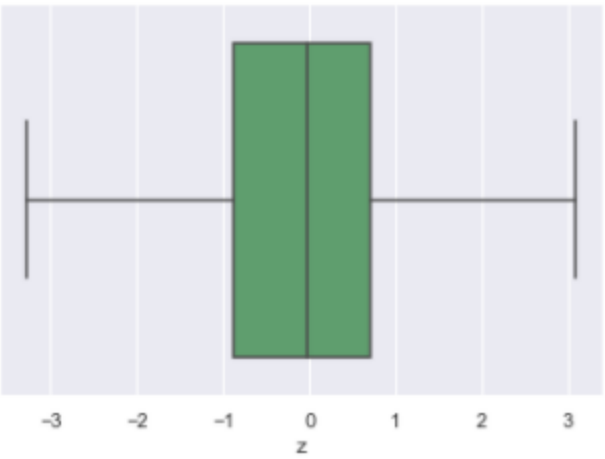

−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
carat

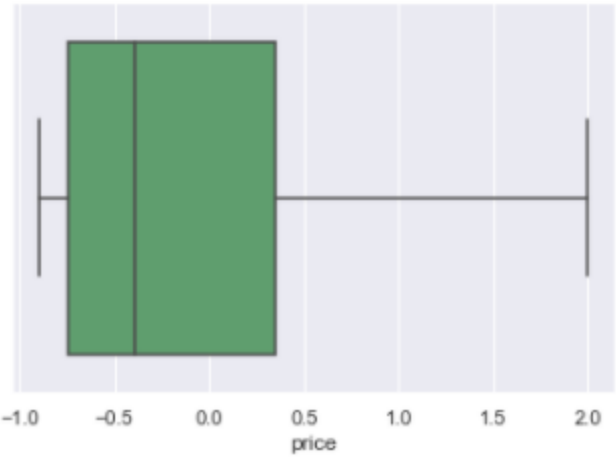

−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
depth


−2
−1
0
1
2
table

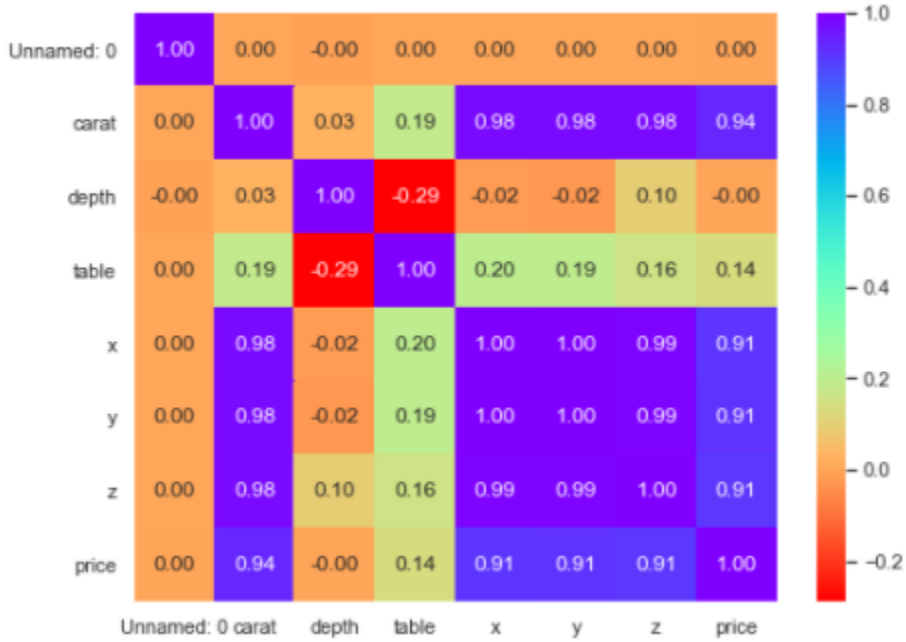

−2
−1
0
1
2
3
x


−1
0
1
2
3
y


−3
−2
−1
0
1
2
3
z



Correlation



PROBLEM 1.3

Encode the data (having string values) for Modeling. 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.

Resolution:

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

	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	

We drop all the unwanted columns and then create Linear regression model.

```
The coefficient for carat is 1.1009417847804506
The coefficient for depth is 0.005605143445570335
The coefficient for table is -0.013319500386804291
The coefficient for x is -0.3050434981963357
The coefficient for y is 0.3039144895792674
The coefficient for z is -0.13916571567987907
The coefficient for cut_Good is 0.09403402912977867
The coefficient for cut_Ideal is 0.15231074620567378
The coefficient for cut_Premium is 0.14852774839849311
The coefficient for cut_Very Good is 0.12583881878452674
The coefficient for color_E is -0.047054422333698755
The coefficient for color_F is -0.06268437439142906
The coefficient for color_G is -0.10072161838356847
The coefficient for color_H is -0.20767313311661695
The coefficient for color_I is -0.3239541927462749
The coefficient for color_J is -0.46858930275015964
The coefficient for clarity_IF is 0.999769139463492
The coefficient for clarity_SI1 is 0.6389785818271342
The coefficient for clarity_SI2 is 0.42959662348315625
The coefficient for clarity_VS1 is 0.8380875826737572
The coefficient for clarity_VS2 is 0.7660244466083622
The coefficient for clarity_VVS1 is 0.9420769630114083
The coefficient for clarity_VVS2 is 0.9313670288415702
```

```
# check the intercept for the model

intercept = regression_model.intercept_[0]

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

```
The intercept for our model is -0.7567627863049385
```

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

```
0.9419557931252712
```

```
# R square on testing data
regression_model.score(X_test, y_test)
```

```
0.9381643998102491
```

```
#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))
```

```
0.20690072466418796
```

```
#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))
```

```
0.21647817772382874
```

VIF Values

```
carat ---> 33.35086119845924
depth ---> 4.573918951598577
table ---> 1.7728852812618978
x ---> 463.5542785436457
y ---> 462.769821646584
z ---> 238.65819968687333
cut_Good ---> 3.609618194943713
cut_Ideal ---> 14.34812508118844
cut_Premium ---> 8.623414379121153
cut_Very Good ---> 7.848451571723688
color_E ---> 2.371070464762613
```

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

From the stat model, we can understand the feature that do not contribute to the model. We can remove the features after the VIF values be reduced. Ideal value of VIF is less than 5%

STATSMODEL

Best param 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:	Fri, 18 Jun 2021	Prob (F-statistic):	0.00
Time:	23:03:06	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

Best param summary after dropping the depth variable:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.942
Model:                            OLS   Adj. R-squared:                  0.942
Method:                 Least Squares   F-statistic:                 1.390e+04
Date:                Fri, 18 Jun 2021   Prob (F-statistic):               0.00
Time:                        23:04:08   Log-Likelihood:                 2953.5
No. Observations:               18870   AIC:                            -5861.
Df Residuals:                   18847   BIC:                            -5680.
Df Model:                          22
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-0.7567	0.016	-46.991	0.000	-0.788	-0.725
carat	1.1020	0.009	122.331	0.000	1.084	1.120
table	-0.0139	0.002	-6.770	0.000	-0.018	-0.010
x	-0.3156	0.031	-10.101	0.000	-0.377	-0.254
y	0.2834	0.031	9.069	0.000	0.222	0.345
z	-0.1088	0.014	-7.883	0.000	-0.136	-0.082
cut_Good	0.0951	0.011	8.876	0.000	0.074	0.116
cut_Ideal	0.1512	0.010	14.508	0.000	0.131	0.172
cut_Premium	0.1474	0.010	14.711	0.000	0.128	0.167
cut_Very_Good	0.1255	0.010	12.239	0.000	0.105	0.146
color_E	-0.0471	0.006	-8.439	0.000	-0.058	-0.036
color_F	-0.0627	0.006	-11.082	0.000	-0.074	-0.052
color_G	-0.1007	0.006	-18.246	0.000	-0.111	-0.090
color_H	-0.2076	0.006	-35.306	0.000	-0.219	-0.196
color_I	-0.3237	0.007	-49.497	0.000	-0.337	-0.311
color_J	-0.4684	0.008	-58.169	0.000	-0.484	-0.453
clarity_IF	1.0000	0.016	62.544	0.000	0.969	1.031
clarity_SI1	0.6398	0.014	46.738	0.000	0.613	0.667
clarity_SI2	0.4302	0.014	31.232	0.000	0.403	0.457
clarity_VS1	0.8386	0.014	60.042	0.000	0.811	0.866
clarity_VS2	0.7667	0.014	55.691	0.000	0.740	0.794
clarity_VVS1	0.9424	0.015	63.655	0.000	0.913	0.971
clarity_VVS2	0.9319	0.014	64.784	0.000	0.904	0.960

```
==============================================================================
Omnibus:                     4699.504   Durbin-Watson:                   1.994
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            17704.272
Skew:                           1.208   Prob(JB):                         0.00
Kurtosis:                       7.084   Cond. No.                         56.5
==============================================================================
```

To ideally bring down the values to lower levels we can drop one of the variables that are highly correlated.
Dropping variables would bring down the multi-collinearity level down.

PROBLEM 1.4

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

Resolution:

To predict the price of the stone and provide insights and provide insights for the company on the profits on different prize slots.

- From the EDA we could understand the cut; ideal cut had number profits to the company. The colors H, I, J have bought profits to the company.
- In clarity, if we could see there were no flawless stones and no profits coming from L1, L2, and L3 stones.
- The ideal, premium and very good types of cut were bringing profits where as fair are not bringing profits.

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

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

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

****

Problem 2:

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

PROBLEM 2.1

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.

Resolution:

- First we import all the necessary libraries in Python, and then import the data file which is 'Holiday_Package'. Once we import the file we confirm whether the data has been uploaded correctly or not using 'head' function. Using this function we can view the data and all the columns and headers whether they are aligning correctly or not.
- Then using the 'shape' function we can understand how many row and columns are there in our data set.
- To check the data type of all the columns and also to check the null values, 'info' functions. Has been used.

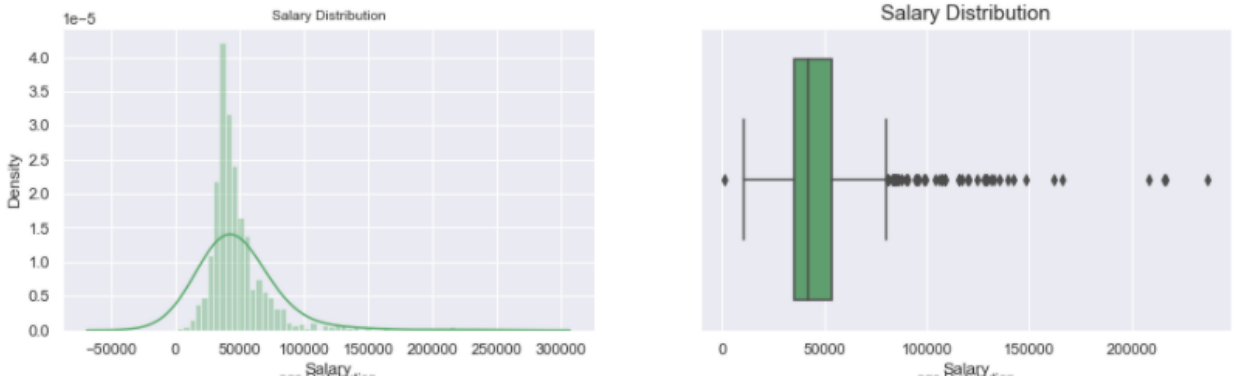
- To see the detail description of the data such as, Count, Mean, Median, Min, Max, Standard Deviations etc,

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	872	436.5	251.869	1	218.75	436.5	654.25	872
Salary	872	47729.17	23418.67	1322	35324	41903.5	53469.5	236961
age	872	39.95528	10.55168	20	32	39	48	62
educ	872	9.307339	3.036259	1	8	9	12	21
no_young_children	872	0.311927	0.61287	0	0	0	0	3
no_older_children	872	0.982798	1.086786	0	0	1	2	6

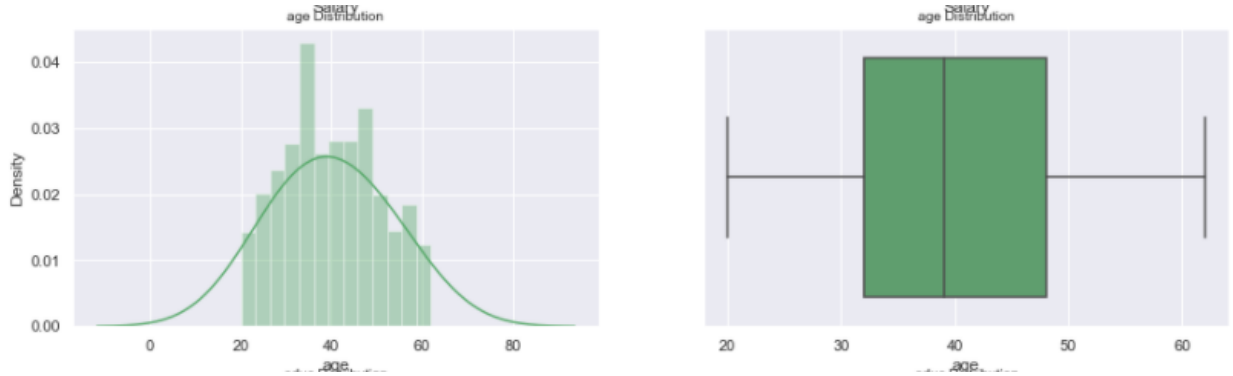
- Using the ‘isnull’ function, one can understand if there are any null values in the data set. And we do not have any null values in the existing data set.
- Using the ‘dups’ function we check for the duplicates and there were few duplicate values which are noted.
- Using the ‘drop_duplicates’ function, we can exclude the duplicate values. Then check for the data.

The split indicates that 45% of the employees are interested in the holiday package

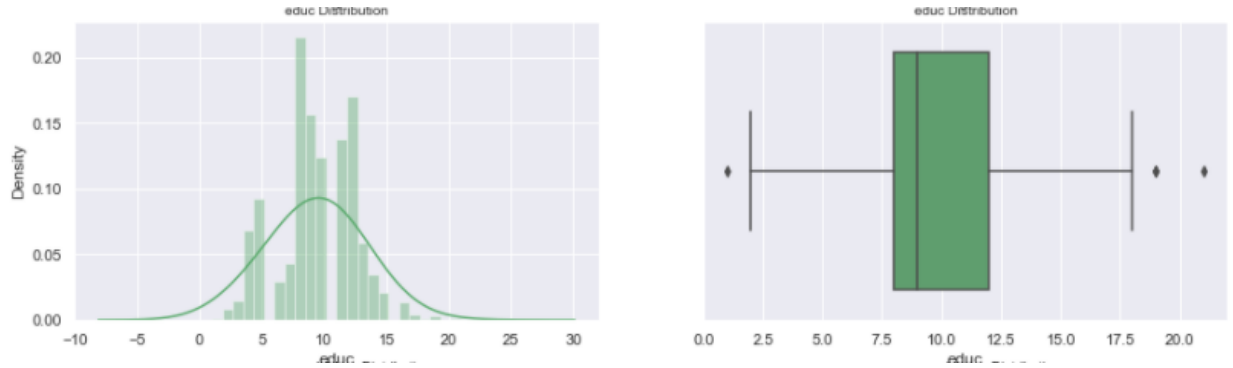
Categorical univariate analysis:



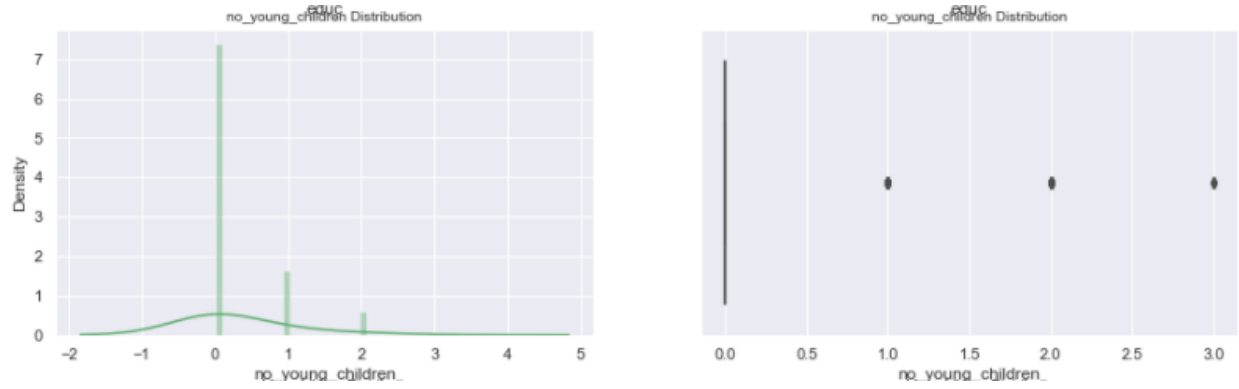
The above dist plot shows the distribution of data from 1000 – 150000 and is positively skewed. Boxplot shows that there are lot of outliers. Majority of the distribution lies in the range of 0 – 100000.



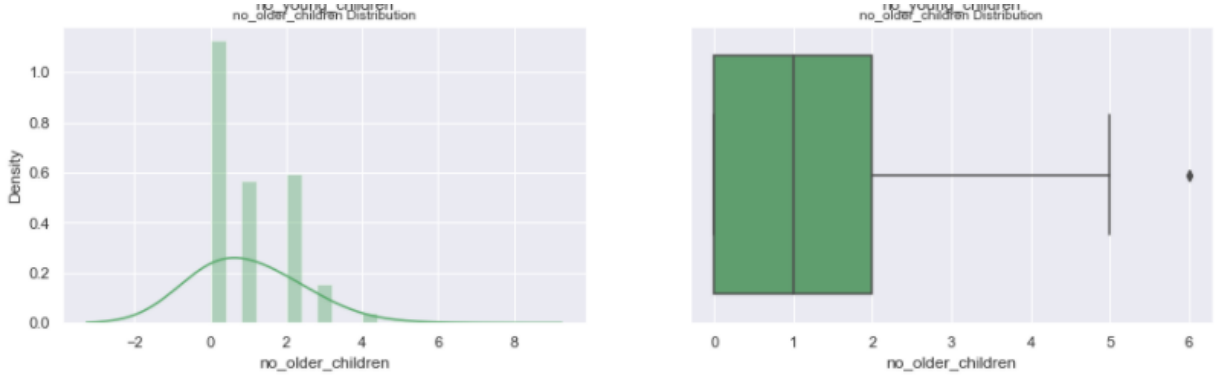
The above dist plot shows the distribution of data from 20 – 60 and is positively skewed. Boxplot shows that there are no outliers.



The above dist plot shows the distribution of data from 0 – 20 and is positively skewed. Boxplot shows that there are a few outliers.

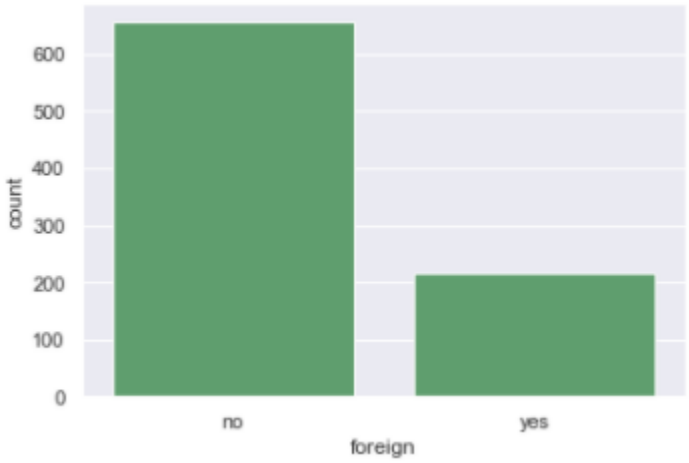


The above dist plot shows the distribution of data from 0 – 2 and is positively skewed. Boxplot shows that there are a few outliers.



The above dist plot shows the distribution of data from 0 – 4 and is positively skewed. Boxplot shows that there are a few outliers.

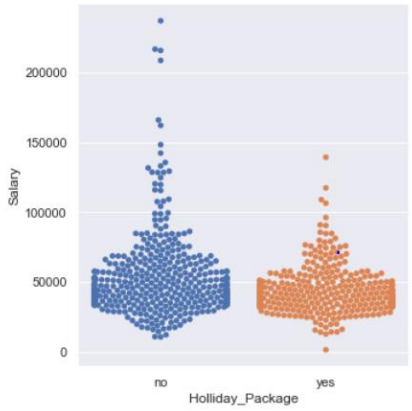
Categorical variables



The distribution of the foreign ‘no’ shows maximum frequency.

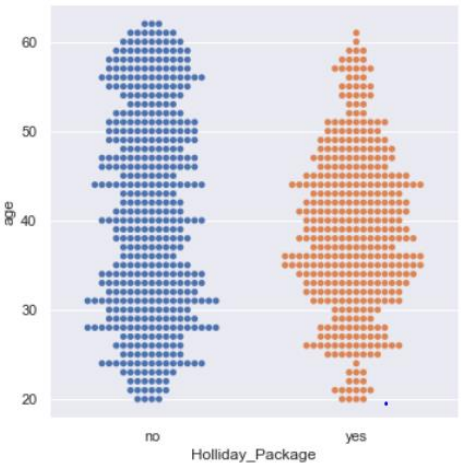
The distribution of the Holiday Package ‘no’ shows maximum frequency.

Holiday Package V/S Salary

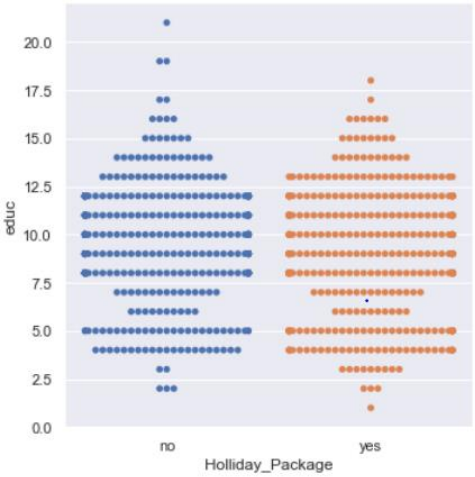


We can see employee below salary 150000 have always opted for holiday package

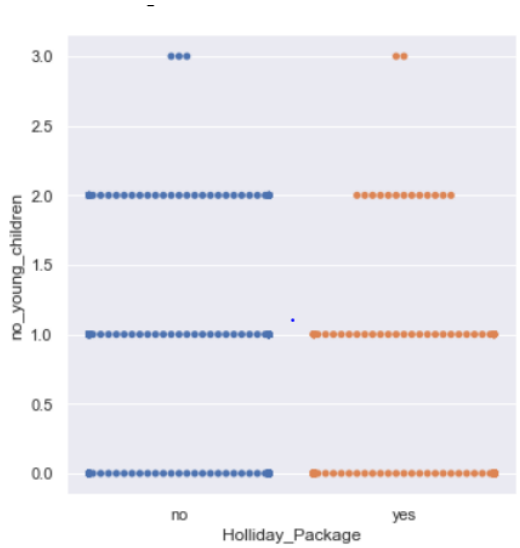
Holiday Package V/S Age



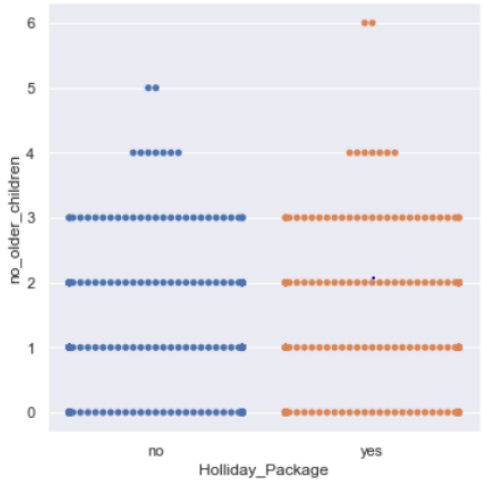
Holiday Package V/S EDUC



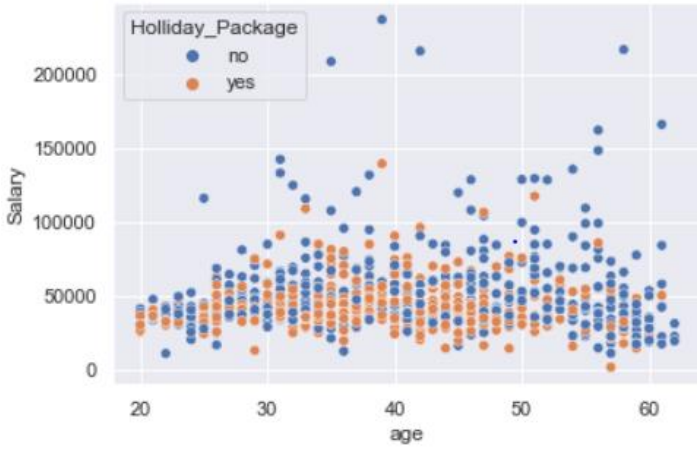
Holiday Package V/S no. of Young Children

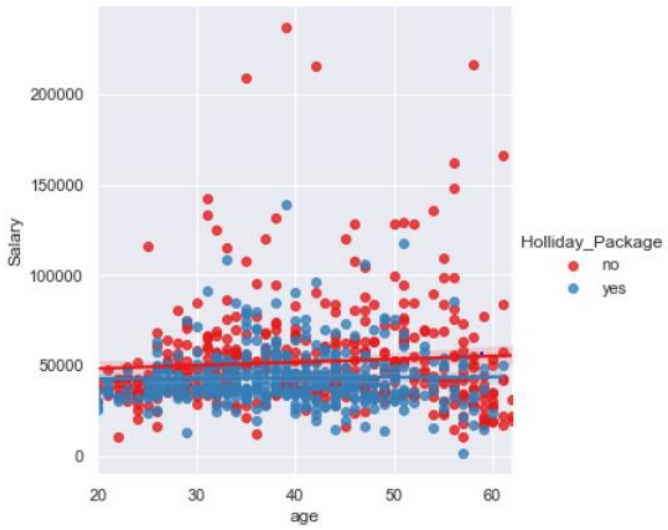


Holiday Package V/S no. of Older Children



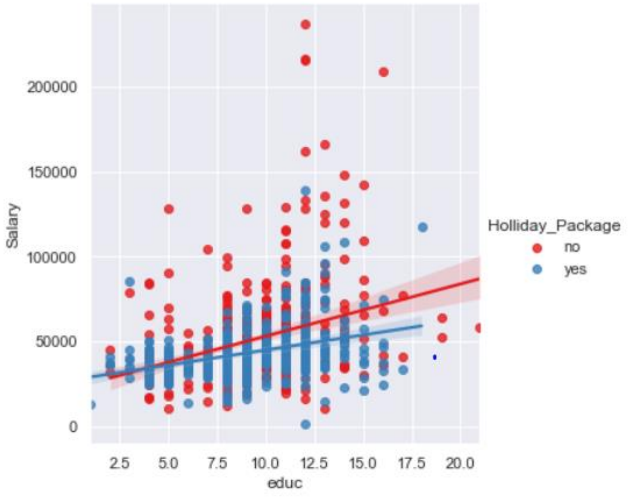
Holiday Package V/S Age V/S Salary

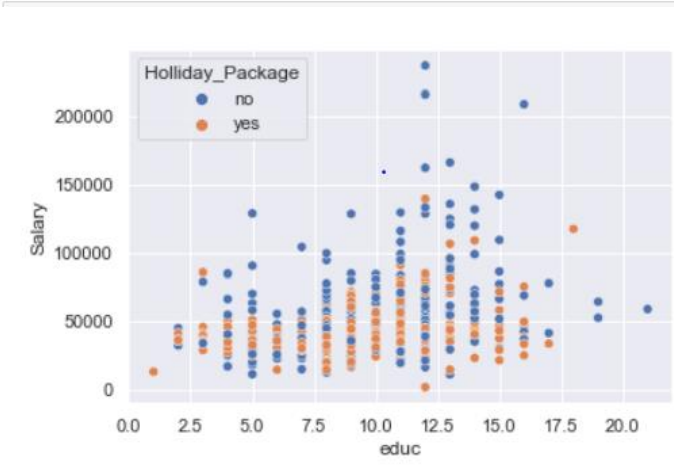




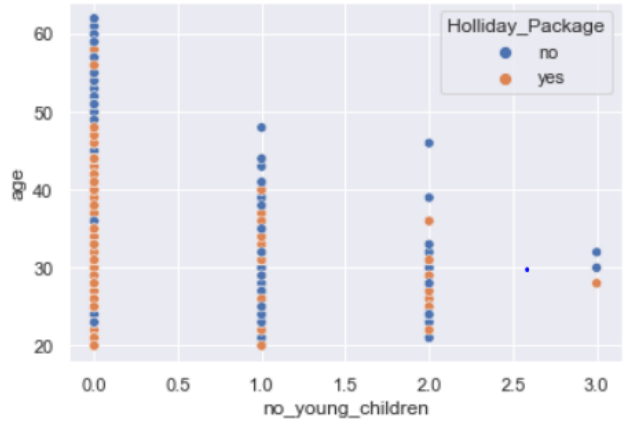
Employee age over 50 to 60 have seems not taking the holiday package. Whereas in the age 30 to 50 group salary less than 50000 people have opted for holiday package.

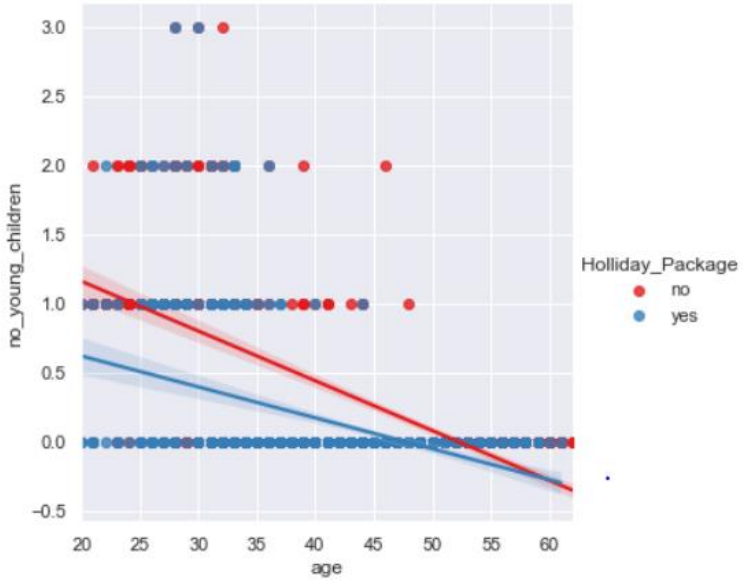
Holiday Package V/S EDUC V/S Salary



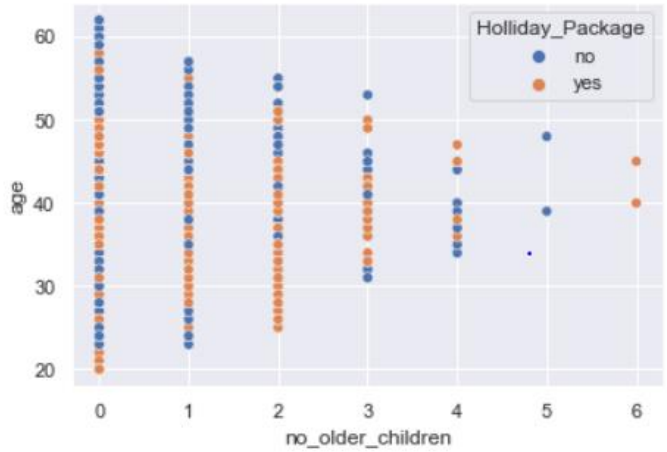


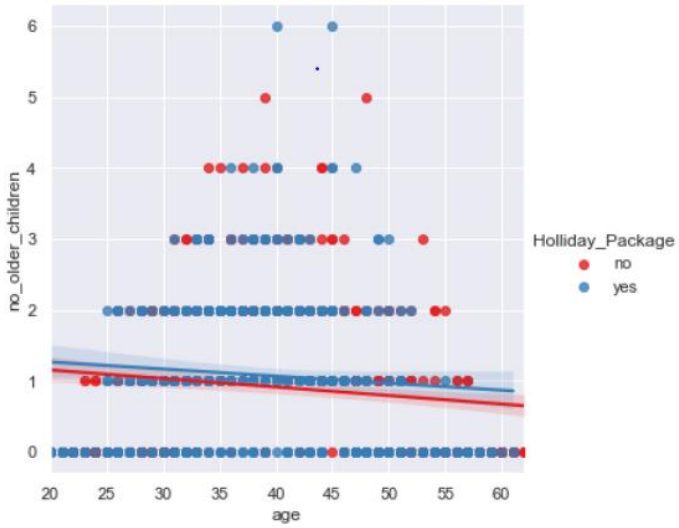
Holiday Package V/S Age V/S Young Children





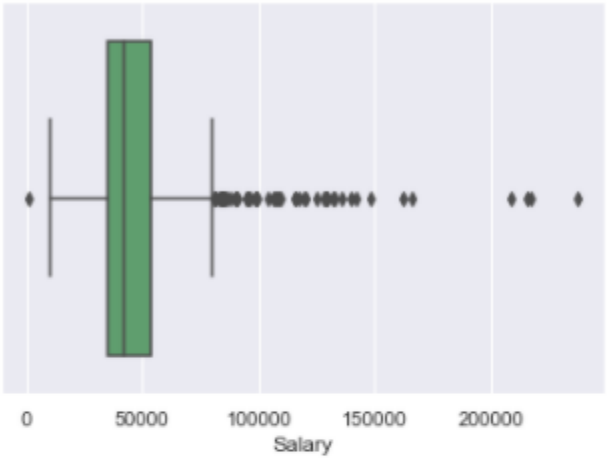
Holiday Package V/S Age V/S Older Children

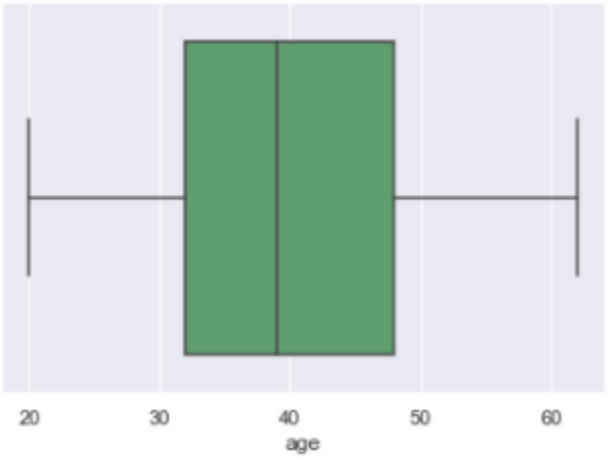


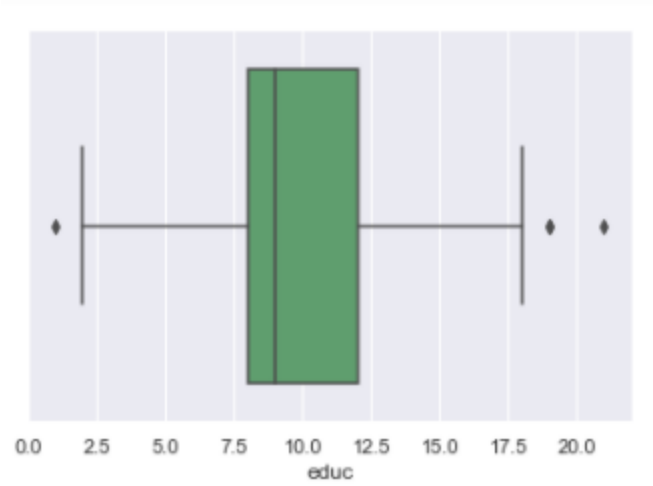


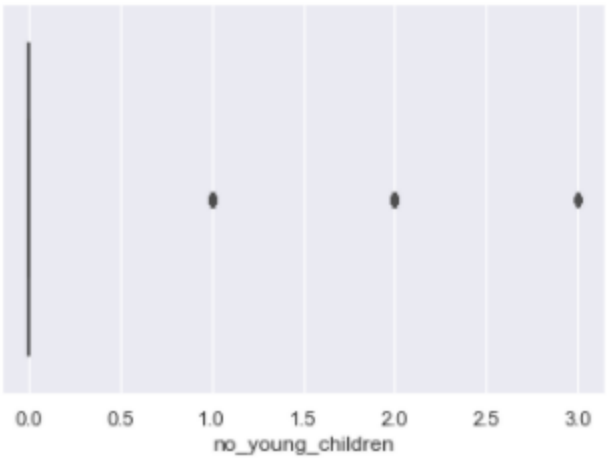
Performing and identifying outliers for Salary ,age , educ , no_young_children, no_older_children

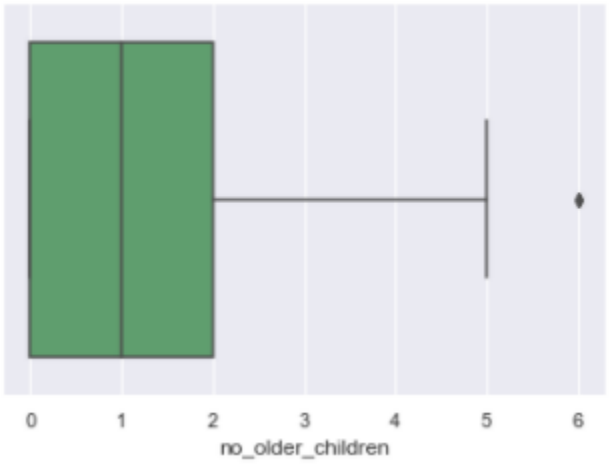
Salary and Age



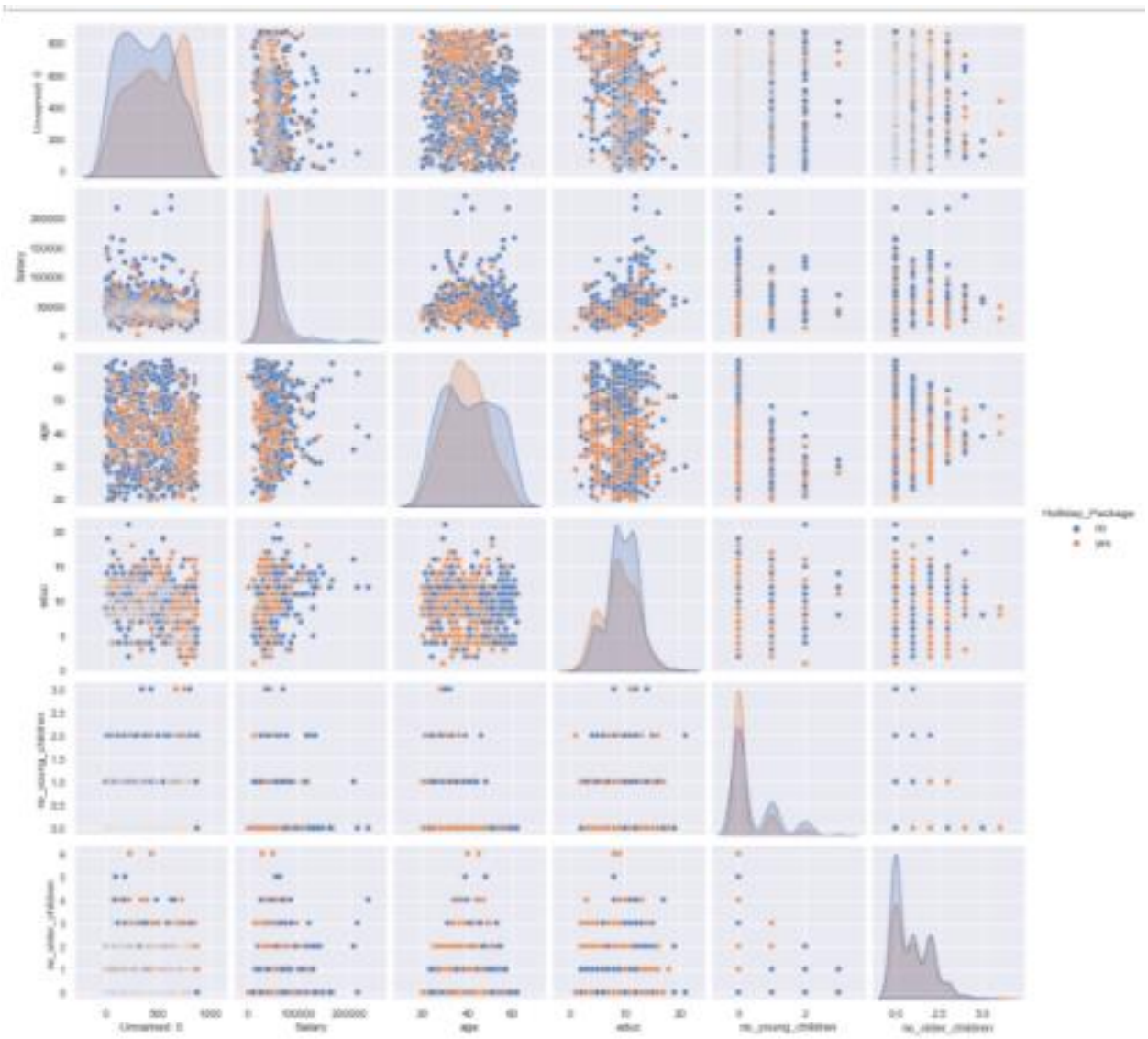



0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
educ

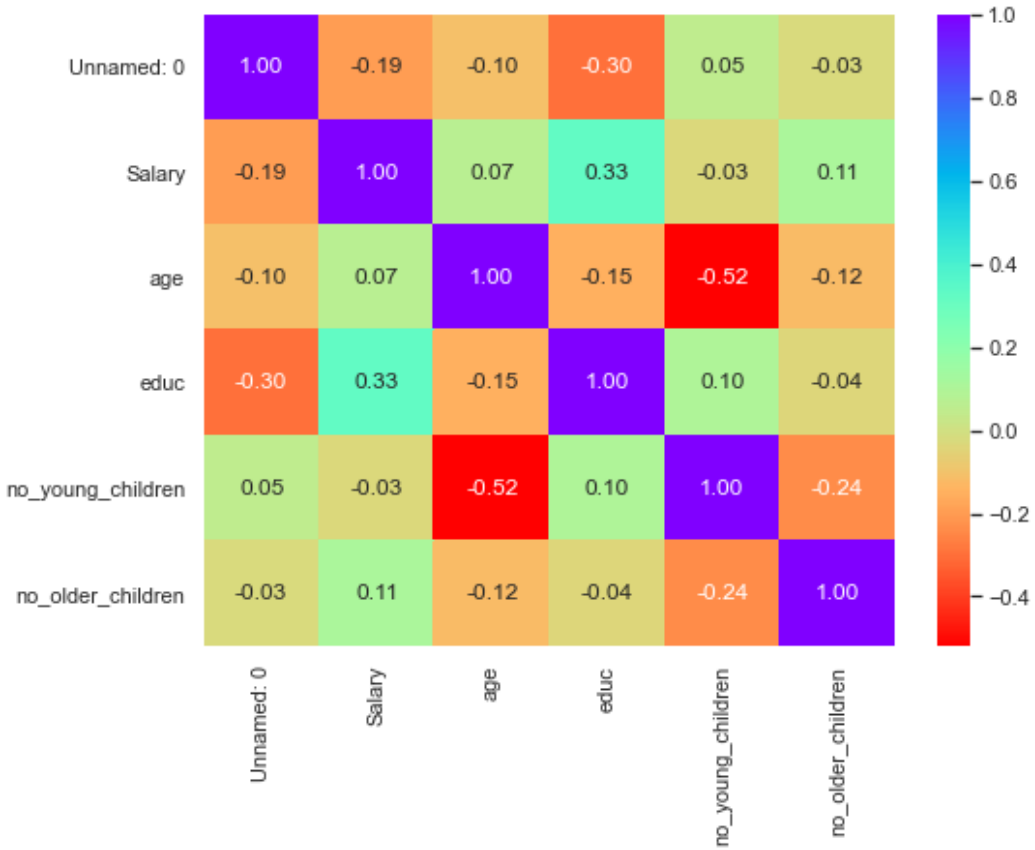

0.0
0.5
1.0
1.5
2.0
2.5
3.0
no_young_children


0
1
2
3
4
5
6
no_older_children

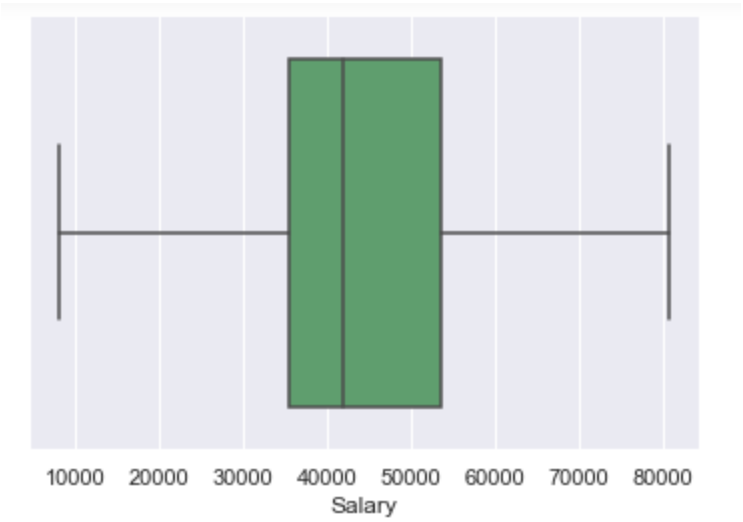
Bi Variate analysis data distribution

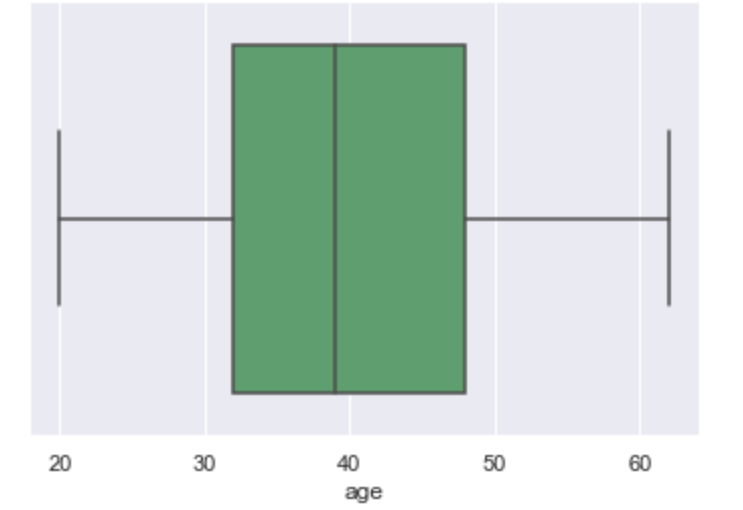


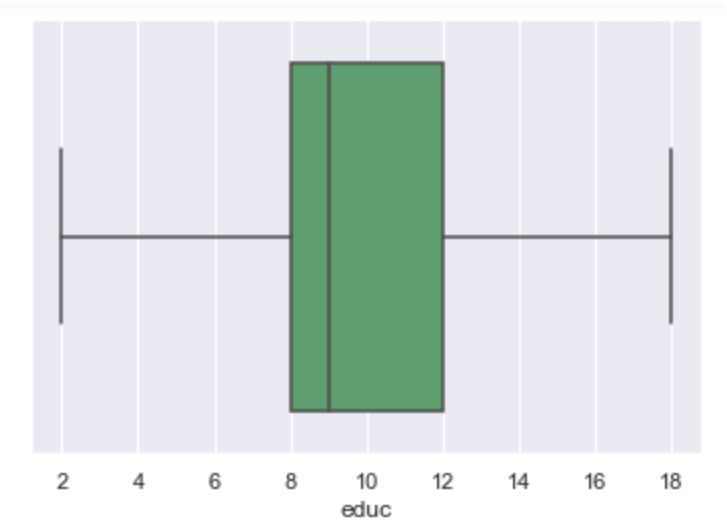
There is no correlation between the data; the data seems to be normal. There is no huge difference in the data distribution among the holiday package.

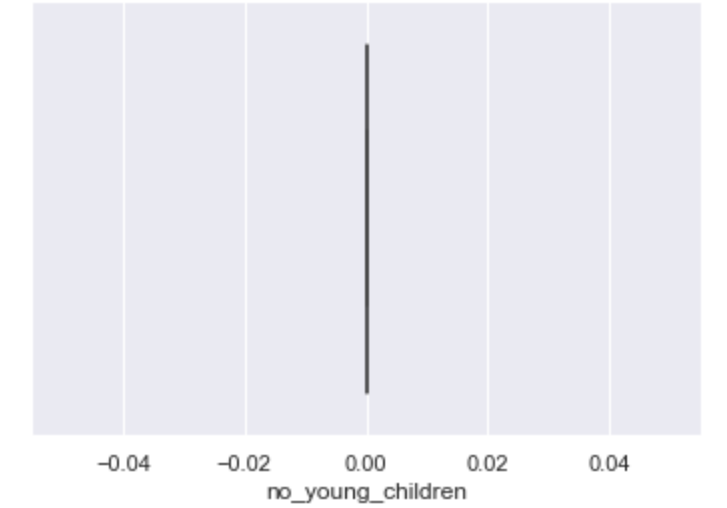


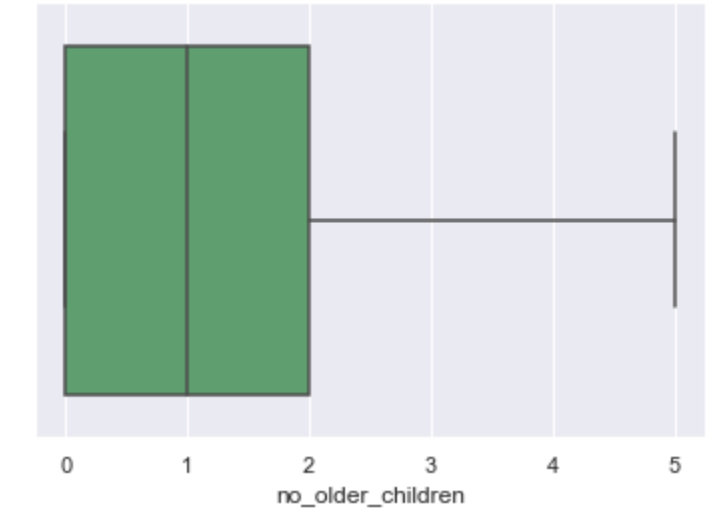
After Outlier Treatment


10000
20000
30000
40000
50000
60000
70000
80000
Salary


20
30
40
50
60
age


2
4
6
8
10
12
14
16
18
educ


−0.04
−0.02
0.00
0.02
0.04
no_young_children


0
1
2
3
4
5
no_older_children

PROBLEM 2.2

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

Resolution:

For training and testing purpose we are splitting the dataset into train and test data in the ratio 70:30.

We have divided the dataset into train and test.

Encoding the categorical variables:

```
data = pd.get_dummies(df2, columns=['Holliday_Package','foreign'], drop_first = True)
```

```
data.head()
```

	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

\

Train and Test Split

```
# Copy all the predictor variables into X dataframe
X = data.drop('Holliday_Package_yes', axis=1)

# Copy target into the y dataframe.
y = data['Holliday_Package_yes']
```

```
# Split X and y into training and test set in 70:30 ratio
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30 , random_state=1,stratify=y)
```

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

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

Grid search method:

The grid search method is used for logistic regression to find the optimal solving and the parameters for solving:

```
### Applying GridSearchCV for Logistic Regression
```

```
grid={'penalty':['l1','l2','none'],
      'solver':['lbfgs', 'liblinear'],
      'tol':[0.0001,0.000001]}
```

```
model = LogisticRegression(max_iter=100000,n_jobs=2)
```

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

```
grid_search.fit(X_train, y_train)
```

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

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

```
{'penalty': 'l2', 'solver': 'liblinear', 'tol': 1e-06}

LogisticRegression(max_iter=100000, n_jobs=2, solver='liblinear', tol=1e-06)
```

The grid search method gives,linear solver which is suitable for small datasets.tolerance and penalty has been found using grid search method

Predicting the training data:

```
# Prediction on the training set

ytrain_predict = best_model.predict(X_train)
ytest_predict = best_model.predict(X_test)
```

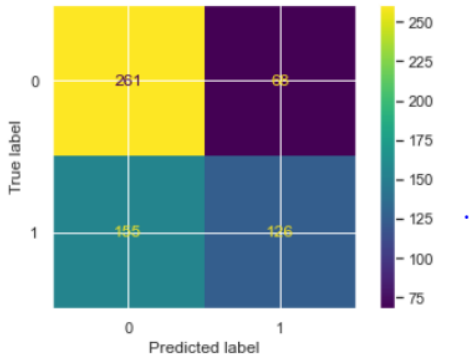
```
## Getting the probabilities on the test set

ytest_predict_prob=best_model.predict_proba(X_test)
pd.DataFrame(ytest_predict_prob).head()
```

	0	1
0	0.636523	0.363477
1	0.576651	0.423349
2	0.650835	0.349165
3	0.568064	0.431936
4	0.536356	0.463644

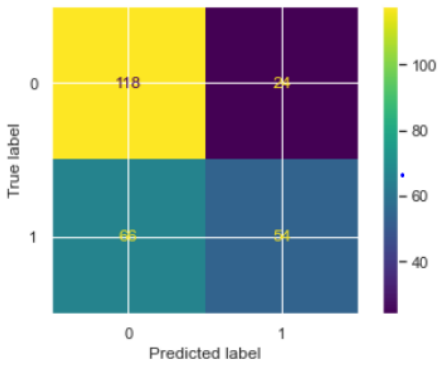
Confusion Matrix Data

	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



Confusion Matrix 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



Accuracy

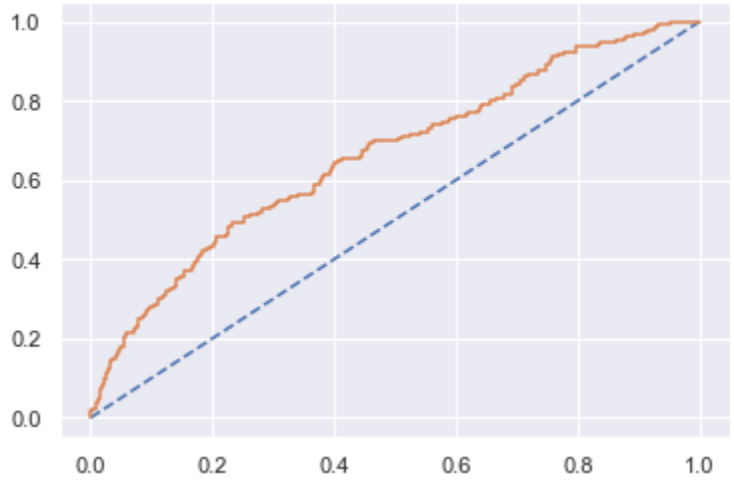
```
#Accuracy - Training Data

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

```
0.6344262295081967
```

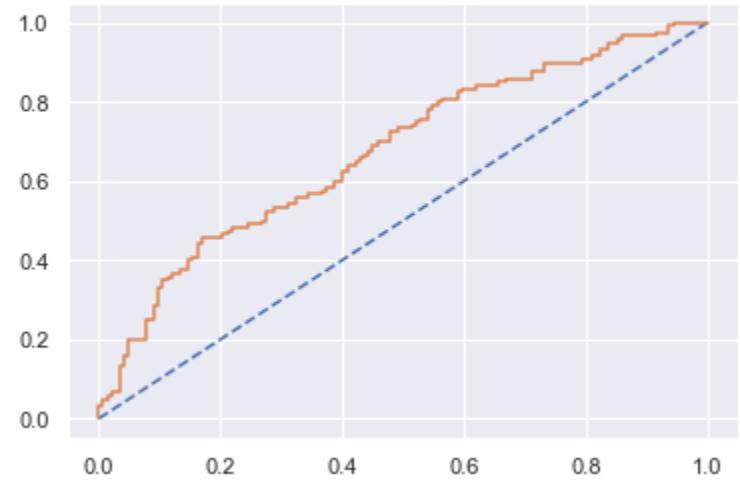
AUC and ROC for the training data

```
AUC: 0.661
```



AUC and ROC Curve from the Test data

```
AUC: 0.675
```



LDA –

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

```
# Training Data Class Prediction with a cut-off value of 0.5
pred_class_train = model.predict(X_train)

# Test Data Class Prediction with a cut-off value of 0.5
pred_class_test = model.predict(X_test)
```

Predicting the variable –

```
# Training Data Class Prediction with a cut-off value of 0.5
pred_class_train = model.predict(X_train)

# Test Data Class Prediction with a cut-off value of 0.5
pred_class_test = model.predict(X_test)
```

PROBLEM 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.

Resolution:

Model Score:

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

```
0.6327868852459017
```

Classification report for the test data:

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

```
              precision    recall  f1-score   support

           0       0.64      0.83      0.72       142
           1       0.69      0.45      0.55       120

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

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

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

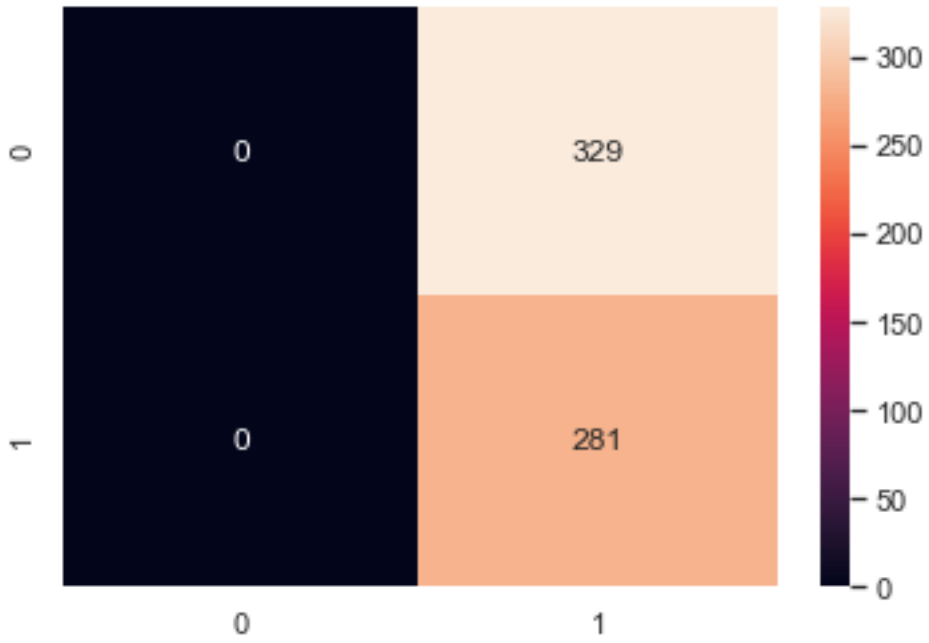
Changing the cut off value to check optimal value that gives better accuracy and f1 score

0.1

Accuracy Score 0.4607

F1 Score 0.6308

Confusion Matrix

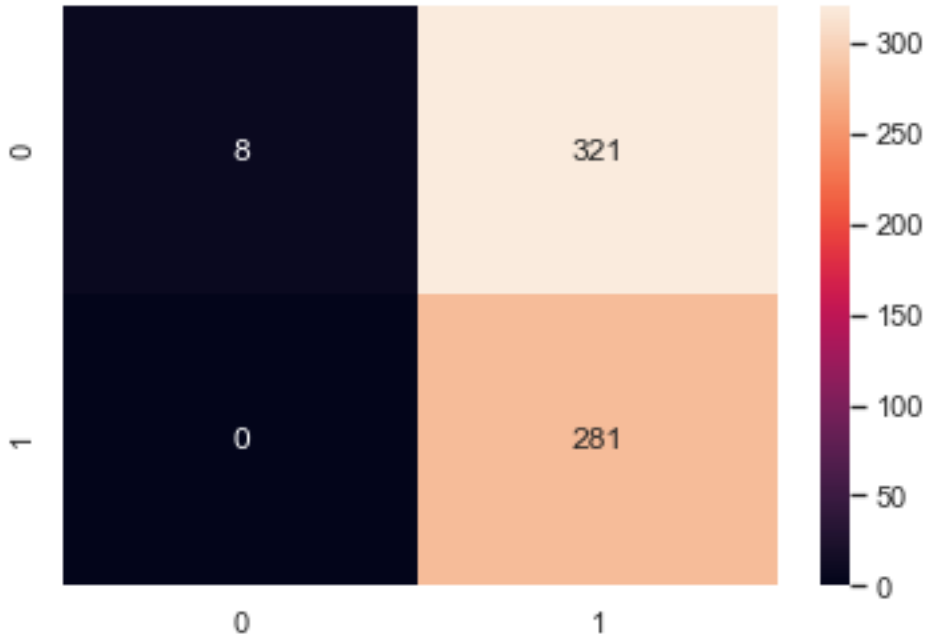


0.2

Accuracy Score 0.4738

F1 Score 0.6365

Confusion Matrix

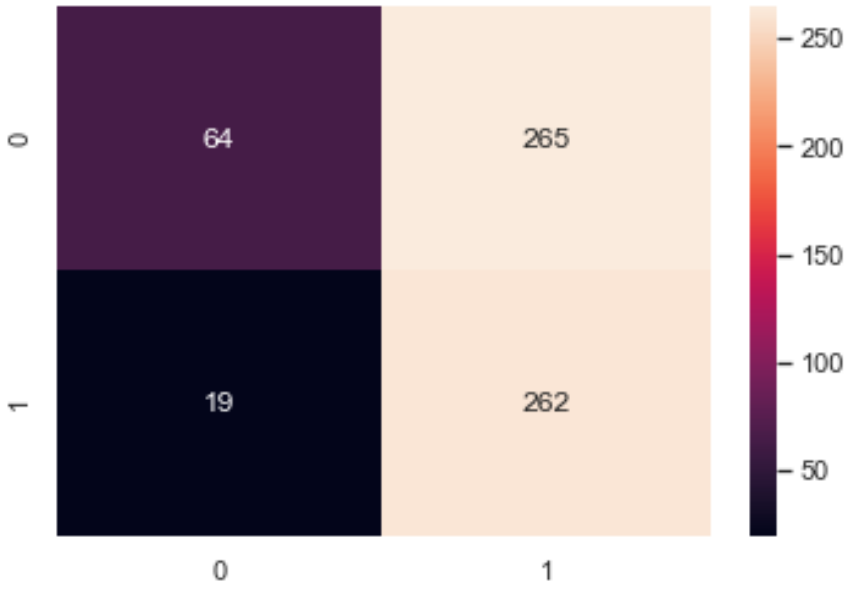


0.3

Accuracy Score 0.5344

F1 Score 0.6485

Confusion Matrix

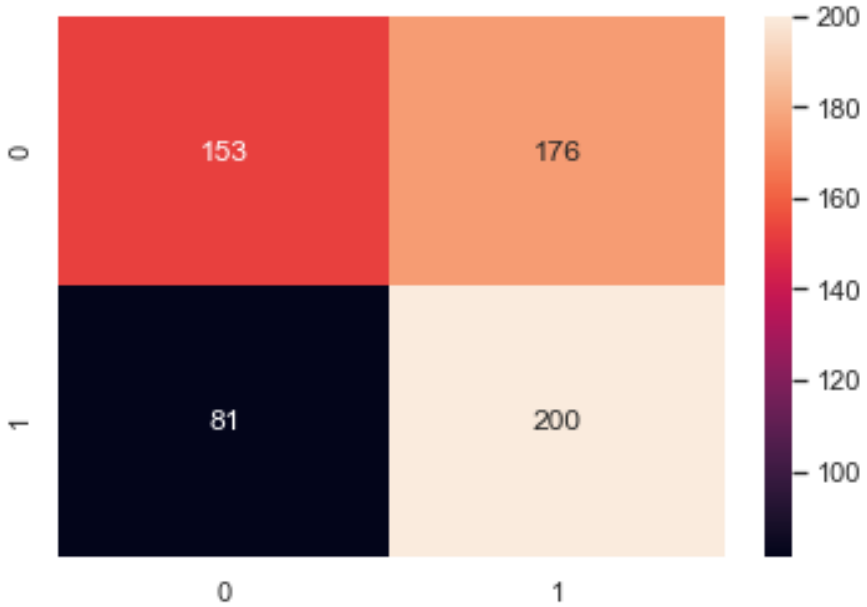


0.4

Accuracy Score 0.5787

F1 Score 0.6088

Confusion Matrix

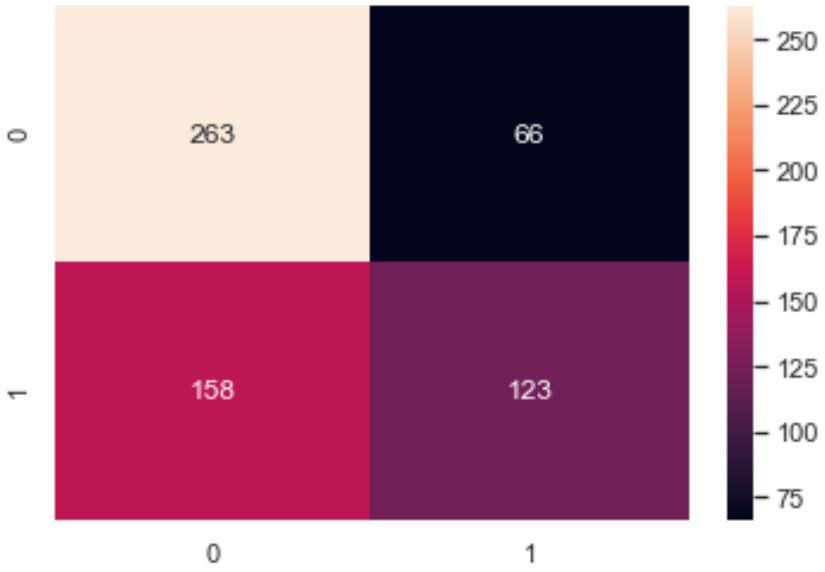


0.5

Accuracy Score 0.6328

F1 Score 0.5234

Confusion Matrix

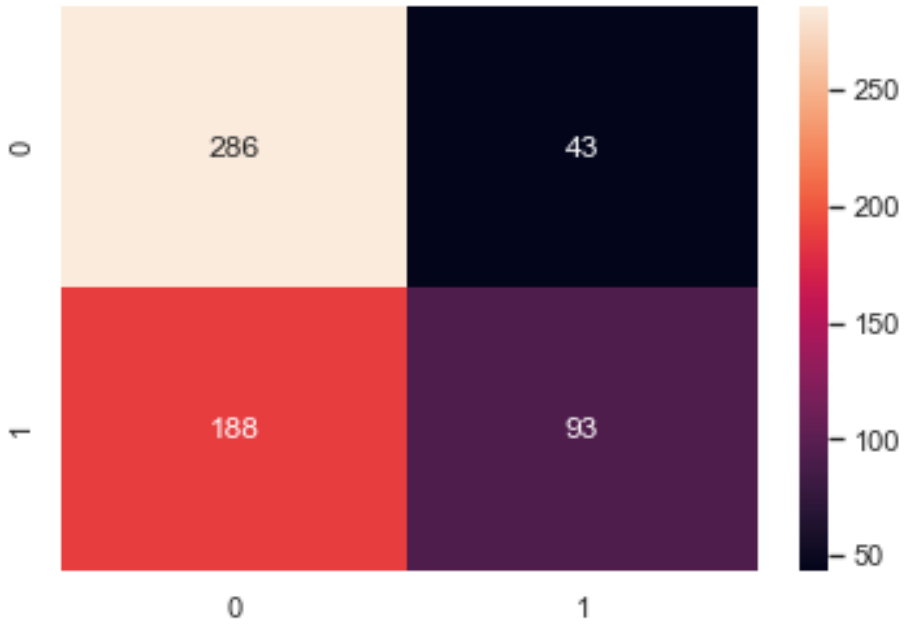


0.6

Accuracy Score 0.6213

F1 Score 0.446

Confusion Matrix

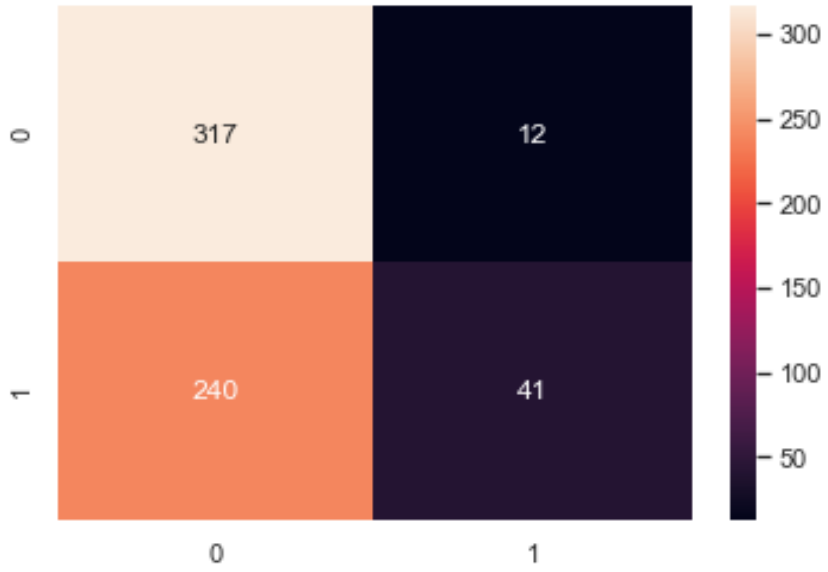


0.7

Accuracy Score 0.5869

F1 Score 0.2455

Confusion Matrix

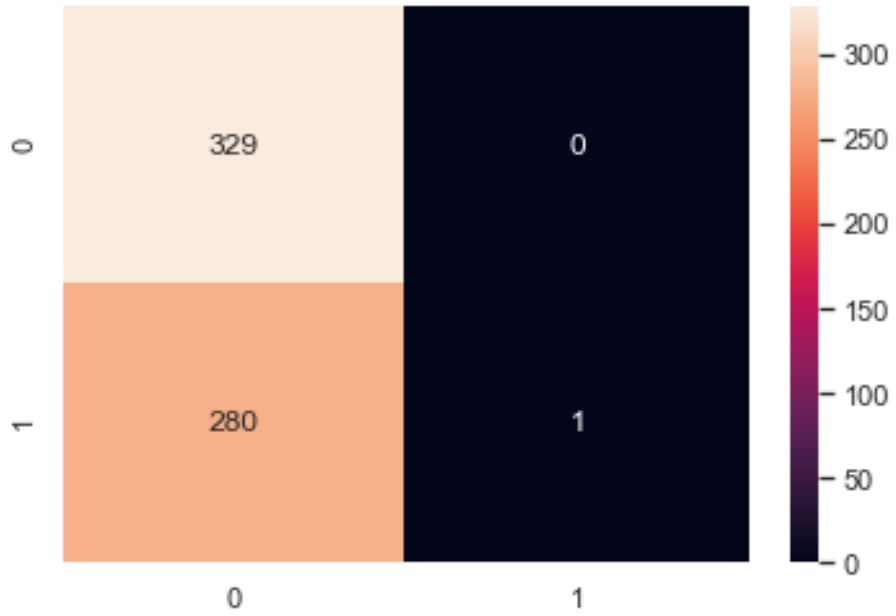


0.8

Accuracy Score 0.541

F1 Score 0.0071

Confusion Matrix

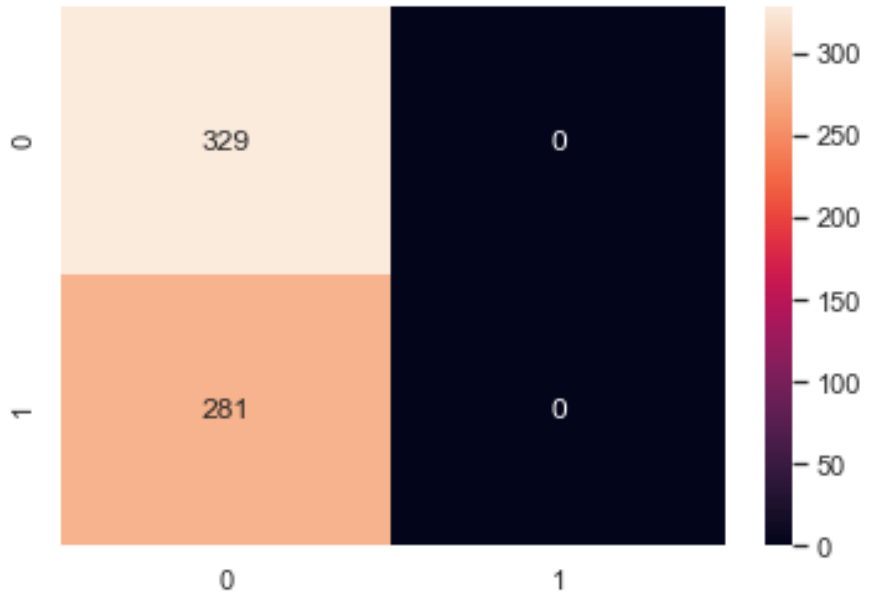


0.9

Accuracy Score 0.5393

F1 Score 0.0

Confusion Matrix



AUC and ROC curve for the train data:

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

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

Comparing the above models, the results are pretty much the same, but LDA works better when there is categorical variable.

PROBLEM 2.4

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

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

Resolution:

We had a business report where we need to predict whether an employee would opt for a holiday or not, in order to arrive at this prediction both logistic regression and LDA, since both the results are pretty much the same.

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

Employees ranging from 30 to 50 generally opt for holiday packages, while in the age bracket of 30 to 50 and salary less than 50k, people have opted for more holiday packages.

The important factors for predictions are salary, age and educ.

Recommendations:

- To improve holiday packages above 50 , we can provide religious packages.
- For people earning more than 150k , we can provide vacation packages.

The End

Thakur Arun Singh

***************************^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^*************************