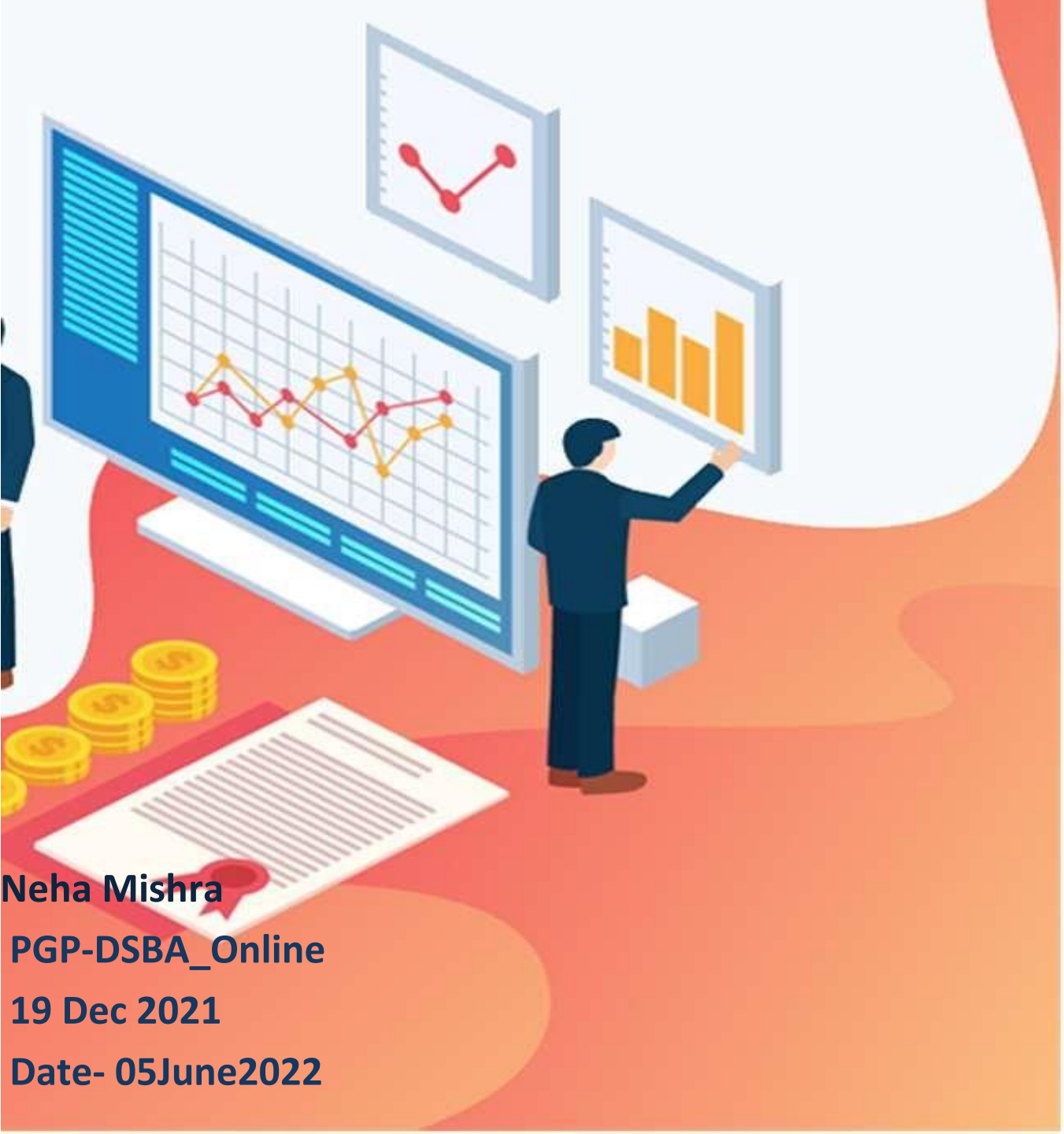


Business Report

On Cubic Zebronica & Holiday Package Dataset

By Linear Regression

Neha Mishra
PGP-DSBA_Online
19 Dec 2021
Date- 05June2022

Table of contents

List of Tables

List of Figures

Executive summary

Problem 1: Linear Regression

We are hired by a company Gem Stones co ltd, which is a cubic zirconia manufacturer 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. We 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, we have to provide them with the best 5 attributes that are most important.

Problem 2: Logistic Regression and LDA

We are hired by a tour and travel agency which deals in selling holiday packages. We are provided details of 872 employees of a company. Among these employees, some opted for the package and some didn't. We 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.

Z : Height of the cubic zirconia in mm.

Data Dictionary of the Dataset

Variable Name	Description
Carat	Carat weight of the cubic zirconia.
Cut	Describe the cut quality of the cubic zirconia. Quality is increasing order Fair, Good, Very Good, Premium, Ideal.
Color	Colour of the cubic zirconia.With D being the worst and J the best.
Clarity	Clarity refers to the absence of the Inclusions and Blemishes. (In order from Worst to Best in terms of avg price) IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1
Depth	The Height of cubic zirconia, measured from the Culet to the table, divided by its average Girdle Diameter.
Table	The Width of the cubic zirconia's Table expressed as a Percentage of its Average Diameter.
Price	the Price of the cubic zirconia.
X	Length of the cubic zirconia in mm.
Y	Width of the cubic zirconia in mm.
Z	Height of the cubic zirconia in mm.

Variable Name	Description
Holiday_Package	Opted for Holiday Package yes/no?
Salary	Employee salary
age	Age in years
edu	Years of formal education
no_young_children	The number of young children (younger than 7 years)
no_older_children	Number of older children
foreign	foreigner Yes/No

Z : Height of the cubic zirconia in mm.

Sample of dataset

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

Table no.1: Sample of Dataset

Exploratory Data Analysis

Let us check the types of variables and missing value in the data frame

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 26967 entries, 1 to 26967
Data columns (total 10 columns):
 #    Column    Non-Null Count   Dtype
---   ------    --------------   -----
 0    carat     26967 non-null   float64
 1    cut       26967 non-null   object
 2    color     26967 non-null   object
 3    clarity   26967 non-null   object
 4    depth     26270 non-null   float64
 5    table     26967 non-null   float64
 6    x         26967 non-null   float64
 7    y         26967 non-null   float64
 8    z         26967 non-null   float64
 9    price     26967 non-null   int64
dtypes: float64(6), int64(1), object(3)
memory usage: 2.3+ MB
```

Table no.2 : data type and non-null value

Data Description

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

Table no 3 : Data Description

Here we 26967nos. of row in each column and having missing values in depth columns.

carat : continuous values from 0.2 to 4.50
depth : continuous values from 50.8 to 73.60
table : continuous values from 49.0 to 79.0
x : continuous values from 0 to 10.23
y : continuous values from 0 to 58.90
z : continuous values from 0 to 31.80
price : continuous values from 326.0 to 18818.0

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

1. carat

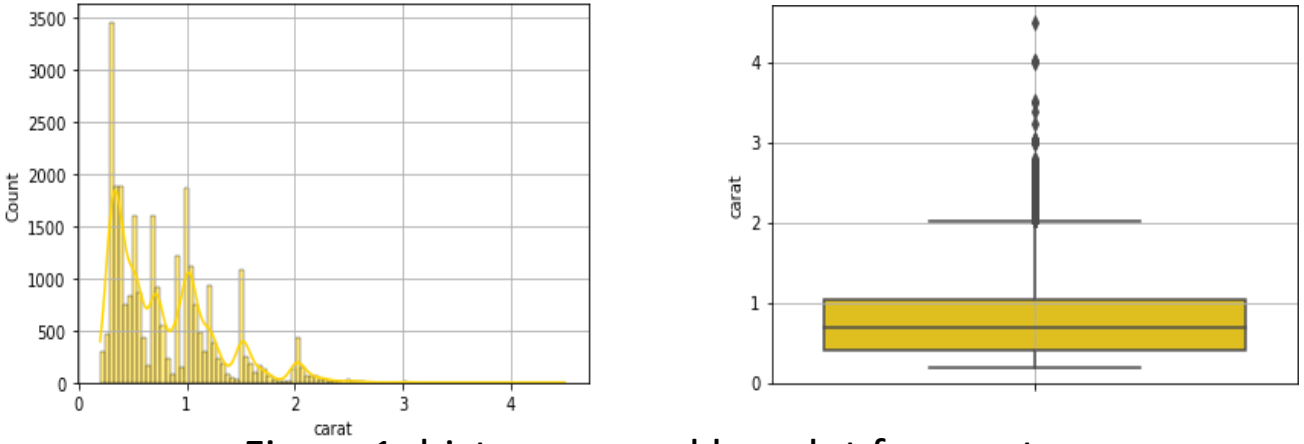


Fig no 1: histogram and boxplot for carat

From above histogram and boxplot, the data point are Left skewed and having outliers in the column.

2. depth

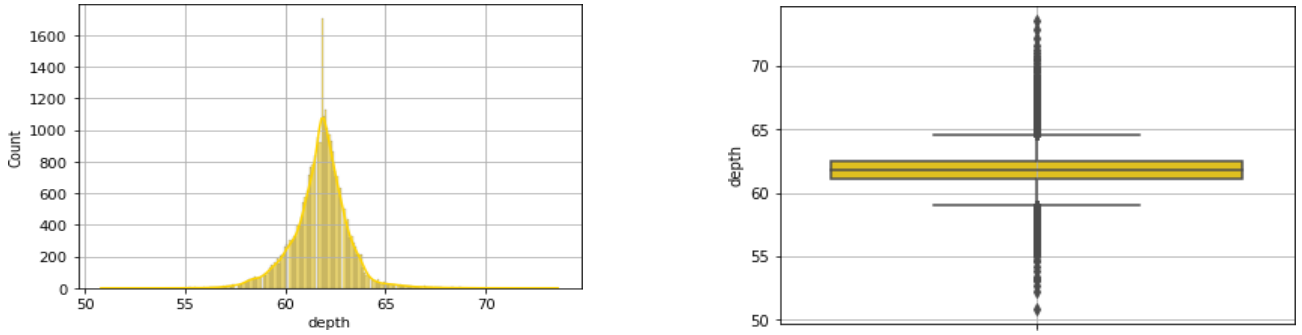
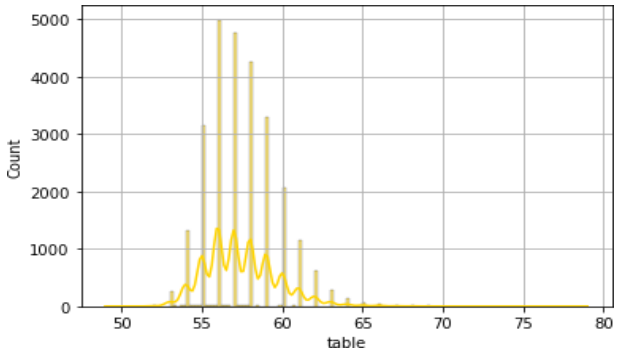


Fig no 2: histogram and boxplot for depth

From above histogram and boxplot, the data point are normally distributed and having outliers in the column.

3.Probability of table



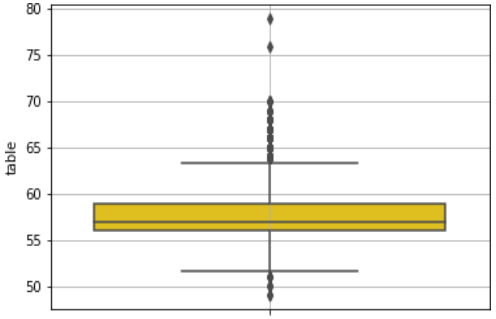
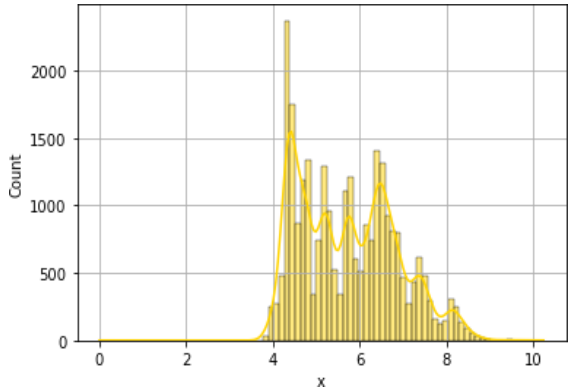


Fig no 3: histogram and boxplot for table

From above histogram and boxplot, the data point are normally distributed and having outliers in the column.

4. X



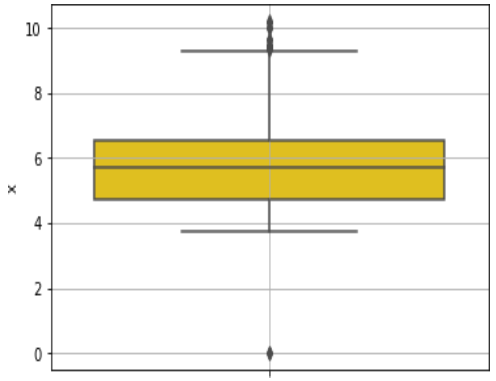


Fig no 4: histogram and boxplot for X

From above histogram and boxplot, the data point are normally distributed and having outliers in the column.

5. Y

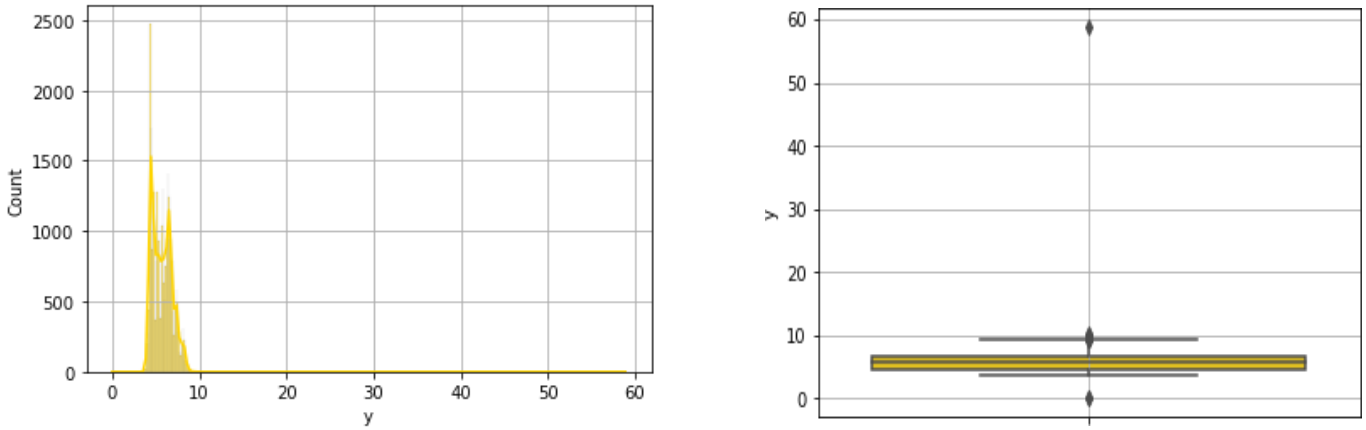


Fig no 5: histogram and boxplot Y

From above histogram and boxplot, the data point are left skewed and having outliers in the column.

6. Z

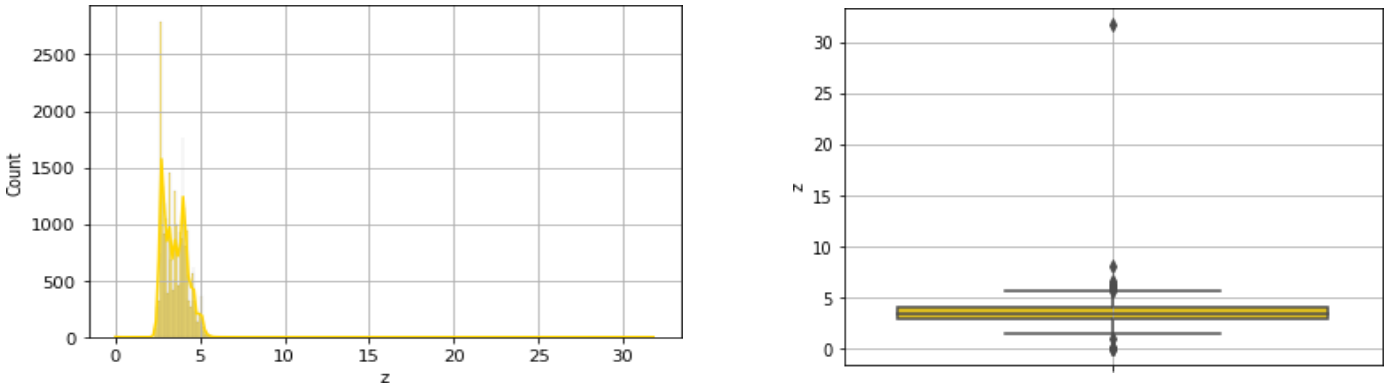


Fig no 6: histogram and boxplot for Z

From above histogram and boxplot, the data point are left skewed and having outliers in the column.

7. price

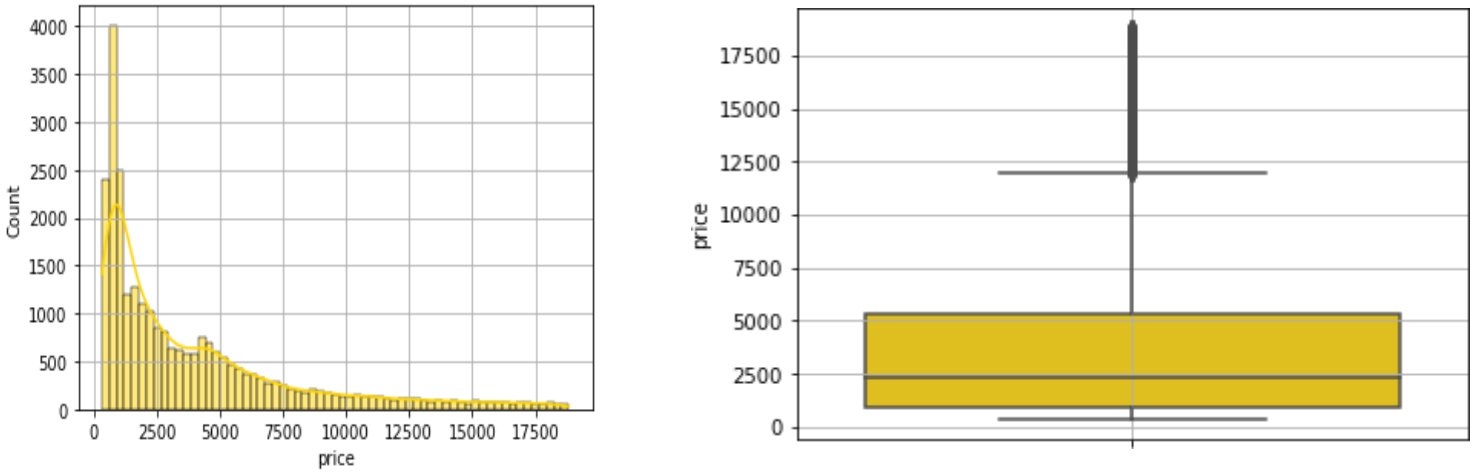


Fig no 7: histogram and boxplot Price

From above histogram and boxplot, the data point are left skewed and having outliers in the column.

Check for duplicates

So, we have 34 duplicates row in our dataset and we will drop it.

Check for null values

We have 697 null value in depth column . Yes we have outlier in depth column we will replace it by median.

Bivariate Analysis

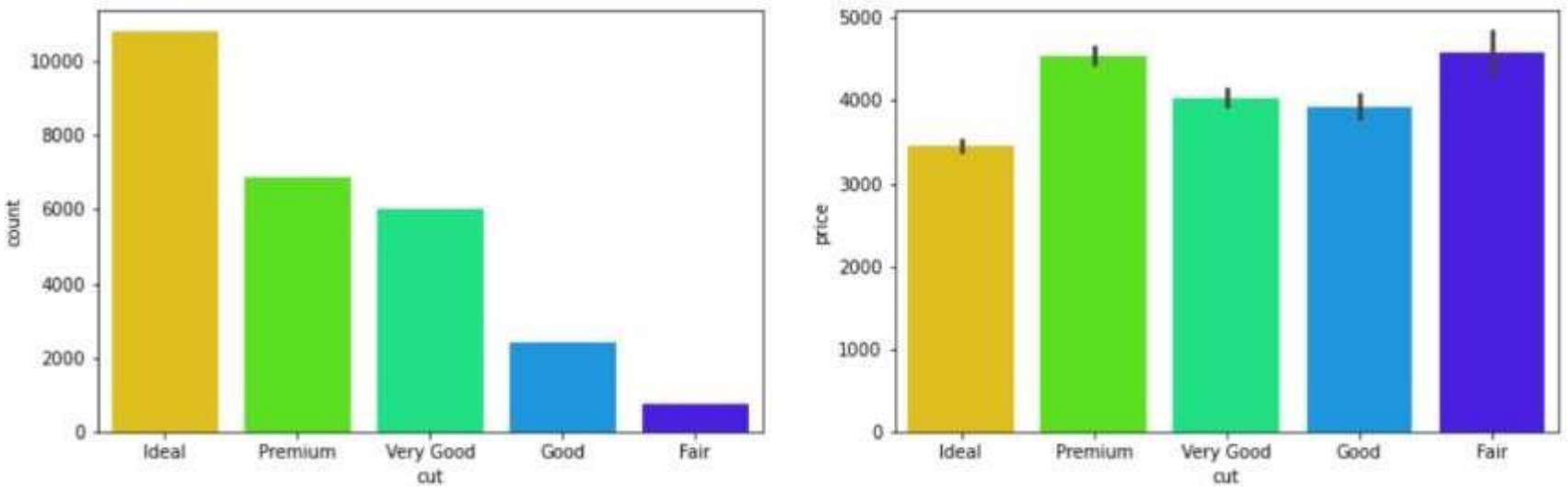


Fig no 8: count and bar Plot for cut

From the above count plot, we can say that ideal cut are more preferable and fair cut are less preferred.

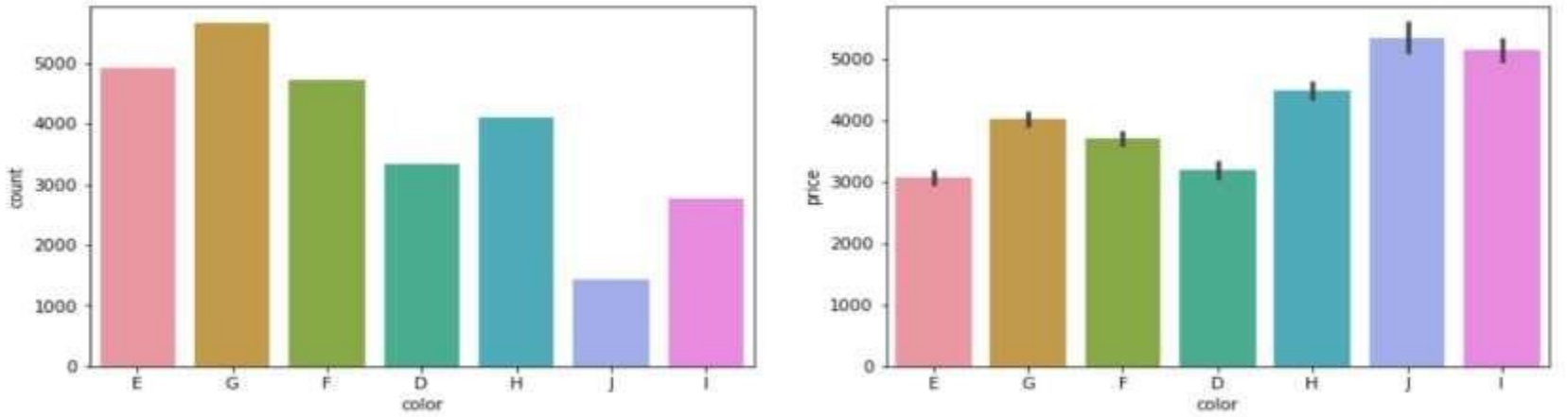


Fig no 9: count and bar Plot for color

From the count plot and bar plot, J type having less in count having higher in price. Similarly for I type color having less in count but having higher price after J type.
G type having lots of selling but having less price.
So, we can easily conclude that worst the color grade less the price and vice versa.

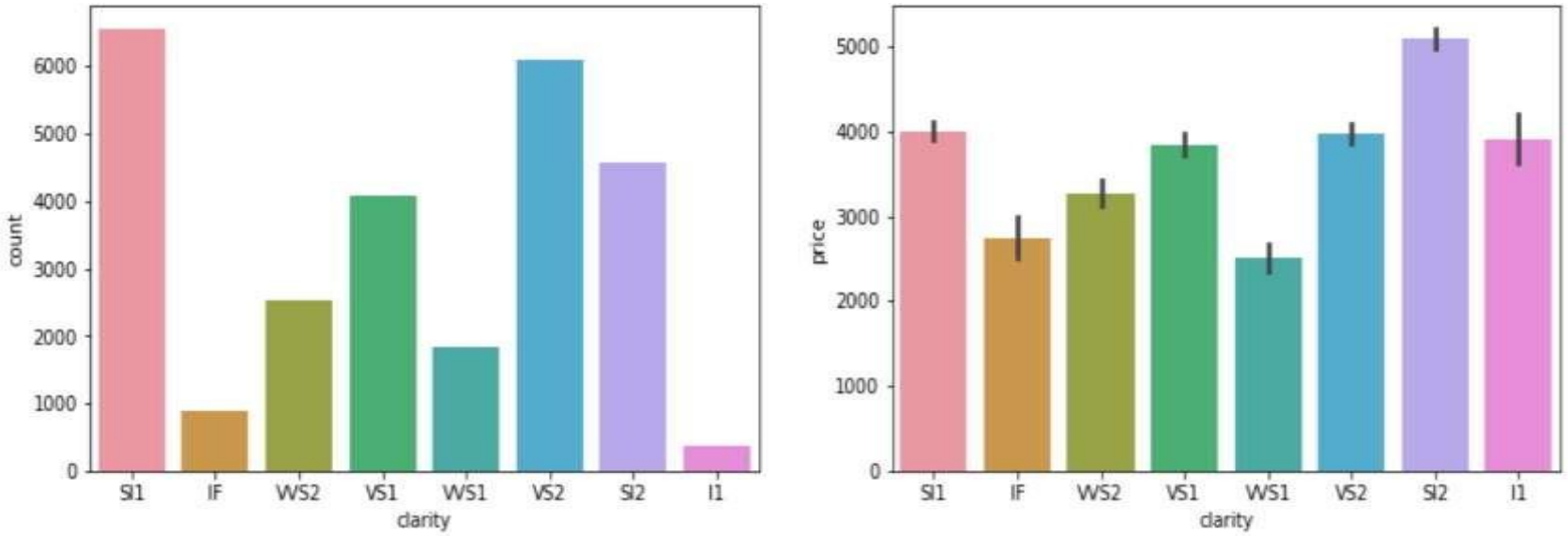


Fig no 10: count and bar Plot for clarity

From the count plot and bar plot, SL1 having more selling but less price,L1 having less in count but having higher price. SL2 showing moderate result in term of selling and price

Multivariate Analysis

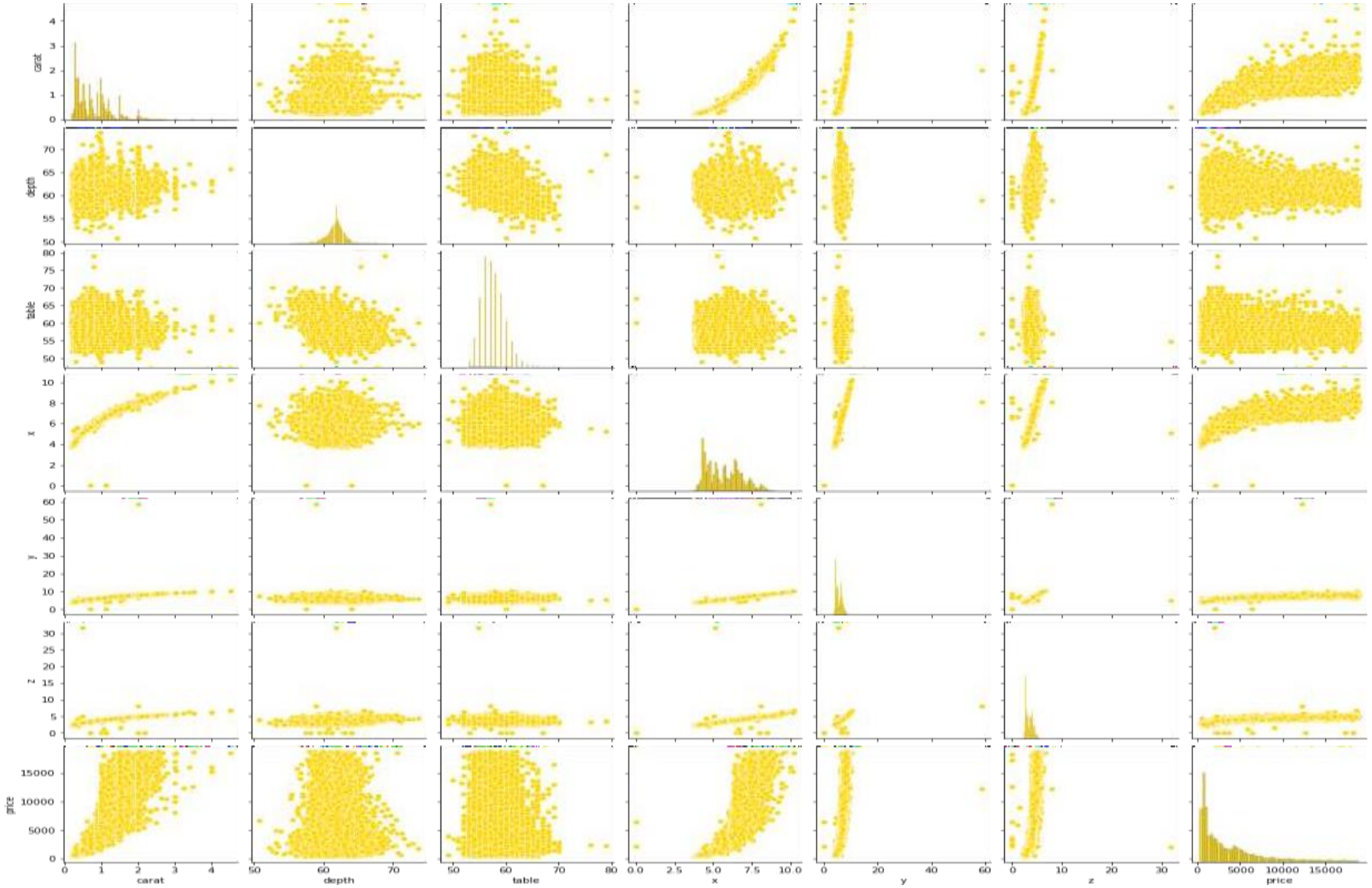


Fig no 11: Pair Plot

Correlation Plot

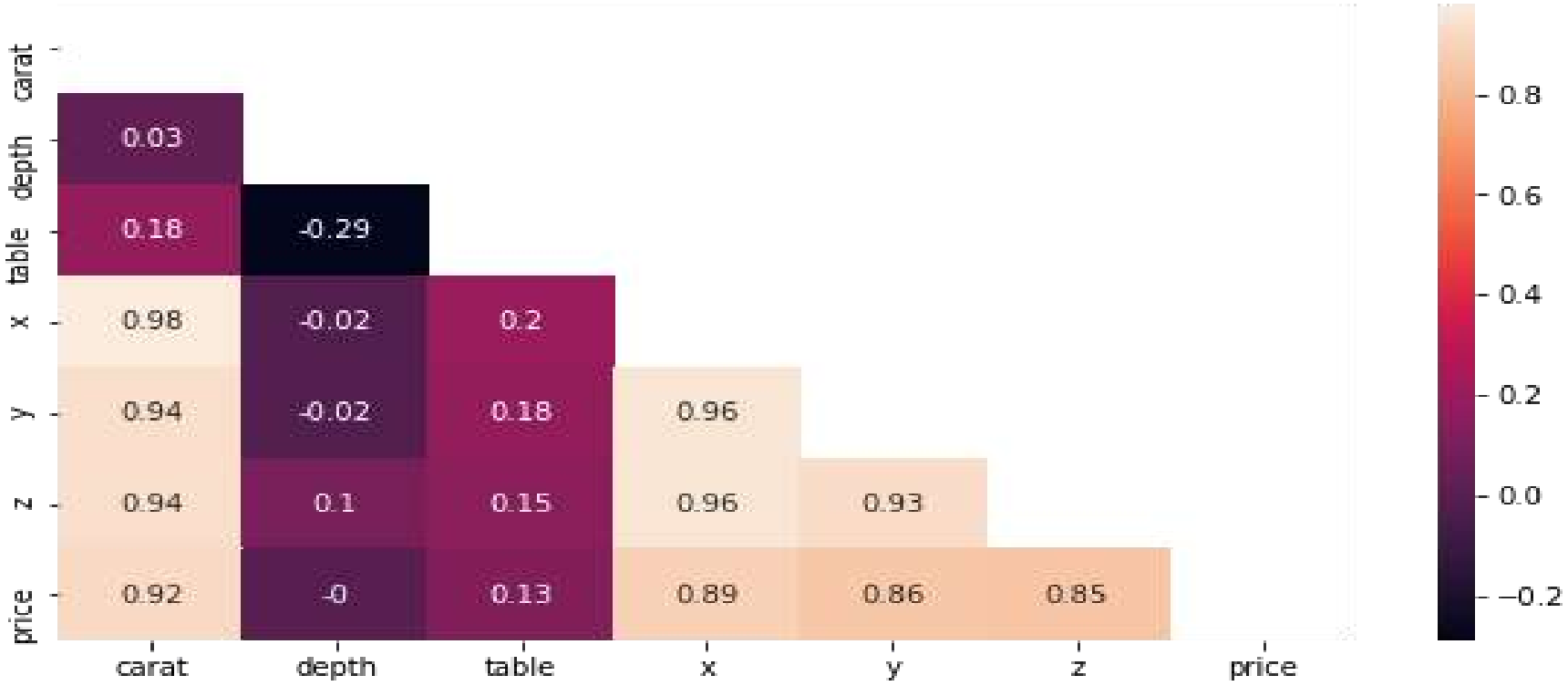


Fig no 12 : Correlation Plot

From the above correlation plot,

We can say that, price is highly correlate with carat, x, y and z.

Even we can say that x, y and z are highly correlated with each other.

Carat also having highly correlated with x, y and z columns.

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

Checking for null values

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

Table no. 4: checking of null before and after treatment

After checking the null attribute, we found null value in depth column. As we know there are outlier in the column by univariate analysis so we will impute it by median.
After treatment null value are 0 in the whole Dataset.

Checking for zeroes

We have 8 rows having zeroes. These rows don't make any sense in our model that's why we will drop these rows.

Checking of sub categories in the columns

Ideal	10805
Premium	6886
Very Good	6027
Good	2435
Fair	780

Name: cut, dtype: int64

G	5653
E	4916
F	4723
H	4095
D	3341
I	2765
J	1440

Name: color, dtype: int64

SI1	6565
VS2	6093
SI2	4564
VS1	4087
VVS2	2530
VVS1	1839
IF	891
I1	364

Name: clarity, dtype: int64

Table no. 5: Sub-Categories of cut , color and clarity

For cut column,

We have ample amount of data in all sub categories and they are providing significant information about cut of the cubic so we will kept as it is.

For color column,

Even color column also have significant amount of row and providing valued information. So we not merge the sub-categories.

For clarity column,

Also the clarity column signify variation of clarity. That's why we will go with no change

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

Converting string to number:

We have 3 column having categorical data. So, we will get dummies for these columns with the the help pd.get_dummies command.

Now, we have converted categorical data into numeric. Currently we have 27 columns in our dataset.

Now, we will create "x" (independent variable) by dropping price column through our dataset and "y" (dependent variable)

With the help of sklearn package we will split our data into test and train in the ratio 70:30.

```
# Copy all the predictor variables into x dataframe. Since 'price' is dependent variable drop it
x = df.drop('price', axis=1)

# Copy the 'mpg' column alone into the y dataframe. This is the dependent variable
y = df[['price']]
```

```
# 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)
```

Importing and fitting the data into the Linear regression model

```
# invoke the LinearRegression function and find the bestfit model on training data

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

```
LinearRegression()
```

The coefficient for the variable in the dataset

```
The coefficient for carat is 9879.749064706852
The coefficient for depth is 6.64572596773207
The coefficient for table is -16.839616995702595
The coefficient for x is -1006.5848626597977
The coefficient for y is 507.9045607303988
The coefficient for z is -783.0114291254704
The coefficient for cut_Fair is -311.1785238207959
The coefficient for cut_Good is -15.387551841476752
The coefficient for cut_Ideal is 148.86595592633105
The coefficient for cut_Premium is 114.62223466036333
The coefficient for cut_Very Good is 63.07788507556897
The coefficient for color_D is 436.7856701360562
The coefficient for color_E is 298.26078579899297
The coefficient for color_F is 239.10432396715856
The coefficient for color_G is 134.62680278844272
The coefficient for color_H is -88.72238439285871
The coefficient for color_I is -375.9290202705343
The coefficient for color_J is -644.1261780272552
The coefficient for clarity_I1 is -1759.3077080526045
The coefficient for clarity_IF is 779.3607309049739
The coefficient for clarity_SI1 is -163.48485581128665
The coefficient for clarity_SI2 is -640.184775080114
The coefficient for clarity_VS1 is 372.2201490243507
The coefficient for clarity_VS2 is 200.22824452805784
The coefficient for clarity_VVS1 is 619.4209995066794
The coefficient for clarity_VVS2 is 591.7472149799588
```

```
# Let us 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 1751.0743953364308
```

Score of our model

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

```
0.9545358701420875
```

Statistics report

```
import statsmodels.formula.api as smf
lm1 = smf.ols(formula= 'price ~ carat+depth+table+x+y+z', data = data_train).fit()
lm1.params
```

```
Intercept     7820.504293
carat        10143.749380
depth          -26.529408
table          -54.508893
x            -2104.752663
y             1689.011762
z            -1437.314343
dtype: float64
```

OLS Regression Results

Dep. Variable:	price	R-squared:	0.915
Model:	OLS	Adj. R-squared:	0.915
Method:	Least Squares	F-statistic:	3.372e+04
Date:	Mon, 13 Dec 2021	Prob (F-statistic):	0.00
Time:	17:15:11	Log-Likelihood:	-1.5131e+05
No. Observations:	18847	AIC:	3.026e+05
Df Residuals:	18840	BIC:	3.027e+05
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	7820.5043	641.529	12.190	0.000	6563.050	9077.958
carat	1.014e+04	134.252	75.557	0.000	9880.603	1.04e+04
depth	-26.5294	8.613	-3.080	0.002	-43.412	-9.646
table	-54.5089	3.060	-17.815	0.000	-60.506	-48.511
x	-2104.7527	120.554	-17.459	0.000	-2341.049	-1868.457
y	1689.0118	114.633	14.734	0.000	1464.321	1913.703
z	-1437.3143	110.723	-12.981	0.000	-1654.342	-1220.287

Omnibus:	1755.176	Durbin-Watson:	1.987
Prob(Omnibus):	0.000	Jarque-Bera (JB):	9112.314
Skew:	0.304	Prob(JB):	0.00
Kurtosis:	6.352	Cond. No.	1.02e+04

Table no 6: OLS Regression Results

```
# Let us check the sum of squared errors by predicting value of y for test cases and
# subtracting from the actual y for the test cases

mse = np.mean((regression_model.predict(x_test)-y_test)**2)
```

```
# underroot of mean_sq_error is standard deviation i.e. avg variance between predicted and actual

import math

math.sqrt(mse)
```

```
546.4932963492643
```

```
# Model score - R2 or coeff of determinant
# R^2=1-RSS / TSS

regression_model.score(x_test, y_test)
```

```
0.9546467224179124
```

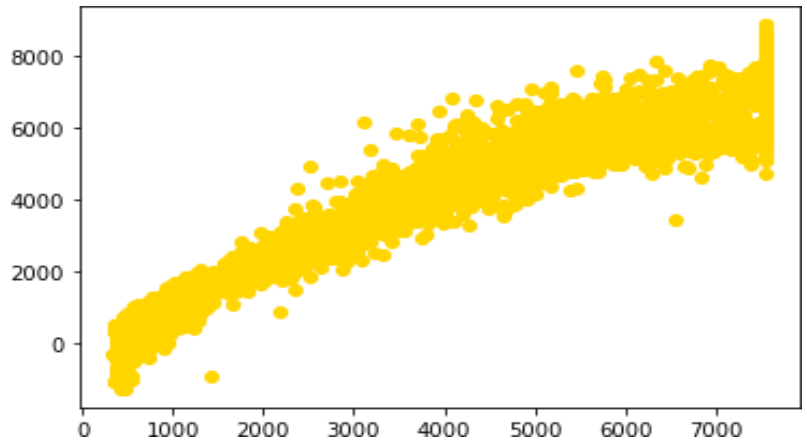


Fig. no.13 : Predicted Regression model

Let us check for after scaling is there any impact on score

```
from scipy.stats import zscore

x_train_scaled  = x_train.apply(zscore)
x_test_scaled = x_test.apply(zscore)
y_train_scaled = y_train.apply(zscore)
y_test_scaled = y_test.apply(zscore)
```

```
The coefficient for carat is 1.4620749863912152
The coefficient for depth is 0.0028925267902885966
The coefficient for table is -0.012715665910179374
The coefficient for x is -0.4160883572690479
The coefficient for y is 0.20848127302000327
The coefficient for z is -0.19998616361283875
The coefficient for cut_Fair is -0.02555371238609221
The coefficient for cut_Good is -0.011062669465593413
The coefficient for cut_Ideal is 0.012861363166163096
The coefficient for cut_Premium is 0.005549132542145552
The coefficient for cut_Very Good is -0.003137087569508546
The coefficient for color_D is 0.04567059107297998
The coefficient for color_E is 0.032649901082723004
The coefficient for color_F is 0.02327162019928337
The coefficient for color_G is 0.008168663764500315
The coefficient for color_H is -0.024348021506458074
The coefficient for color_I is -0.05484951000702485
The coefficient for color_J is -0.06428090776571553
The coefficient for clarity_I1 is -0.08256493320349179
The coefficient for clarity_IF is 0.05086541089526199
The coefficient for clarity_SI1 is -0.038690566125514816
The coefficient for clarity_SI2 is -0.10415568661288031
The coefficient for clarity_VS1 is 0.04336897593826926
The coefficient for clarity_VS2 is 0.02219586761736806
The coefficient for clarity_VVS1 is 0.05462564928868137
The coefficient for clarity_VVS2 is 0.06041322859274245
```

```
intercept = regression_model.intercept_[0]

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

```
The intercept for our model is 5.596392682296588e-16
```

```
# Model score - R2 or coeff of determinant
# R^2=1-RSS / TSS

regression_model.score(x_test_scaled, y_test_scaled)
```

```
0.9546665332957563
```

```
import math

math.sqrt(mse)
```

```
0.21291657216910992
```

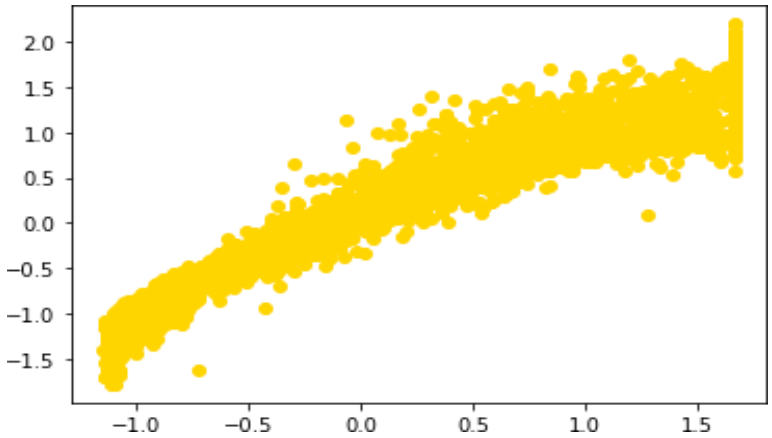


Fig. no.14 : Predicted Regression model after scaled

variance_inflation_factor

```
i=0
for column in x.columns:
    if i < 11:
        print (column ,"--->",  vif[i])
        i = i+1
```

```
carat ---> 89.76155021601936
depth ---> 3.361565126924102
table ---> 1.820130688961126
x ---> 485.94173750348426
y ---> 442.19216182623285
z ---> 202.7219307011127
cut_Fair ---> inf
cut_Good ---> inf
cut_Ideal ---> inf
cut_Premium ---> inf
cut_Very Good ---> inf
```

Let us compare with lasso

	model	best_score	best_params
0	linear_regression	0.954884	{'normalize': False}
1	lasso	0.954589	{'alpha': 1, 'selection': 'cyclic'}

So, more or less all model providing similar result we can go any one of them. we will go with linear_regression model.

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

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

Business insights :

1) So after the study of the dataset we can say that, ideal , premium and very good cut provides a better business having better price along the other.
2) From the scatter plot between price and carat, as the carat weight increase the price increase
3) L1 type of clarity having more price than any other clarity type.
4) From the pairplot, depth and table column not having major impact on the price of the cubic. We can drop these column for price prediction.

Our model works with 95% of score. We test with it lasso, unscaled and scaled linear regression model.

Recommendations :

Carat , cut, clarity, x(length of cubic zebronica in mm) and y(width of cubic zebronica in mm) are most impactful factor to predict and increase the price of the cubic zebronica.

Sample of dataset

	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
1	no	48412	30	8	1	1	no
2	yes	37207	45	8	0	1	no
3	no	58022	46	9	0	0	no
4	no	66503	31	11	2	0	no
5	no	66734	44	12	0	2	no

Table no.1 : Sample of Dataset

Exploratory Data Analysis

Let us check the types of variables and missing value in the data frame

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 872 entries, 1 to 872
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Holliday_Package   872 non-null    object
 1   Salary             872 non-null    int64
 2   age                872 non-null    int64
 3   educ               872 non-null    int64
 4   no_young_children  872 non-null    int64
 5   no_older_children  872 non-null    int64
 6   foreign            872 non-null    object
dtypes: int64(5), object(2)
memory usage: 54.5+ KB
```


Table no.2 : data type and non-null value

Data Description

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Holliday_Package	872	2	no	471	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Salary	872.0	NaN	NaN	NaN	47729.172018	23418.668531	1322.0	35324.0	41903.5	53469.5	236961.0
age	872.0	NaN	NaN	NaN	39.955275	10.551675	20.0	32.0	39.0	48.0	62.0
educ	872.0	NaN	NaN	NaN	9.307339	3.036259	1.0	8.0	9.0	12.0	21.0
no_young_children	872.0	NaN	NaN	NaN	0.311927	0.61287	0.0	0.0	0.0	0.0	3.0
no_older_children	872.0	NaN	NaN	NaN	0.982798	1.086786	0.0	0.0	1.0	2.0	6.0
foreign	872	2	no	656	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table no 3 : Data DescriptionHere we 872 nos. of row in

each column.

Salary : continuous values from 1322.0 to 236961.0
Age : continuous values from 20 to 62
educ : categorical values from 1 to 21
No_young_children: categorical values from 0, 1, 2 and 3
No_older_children : categorical values from 0 to 6 foreign
: categorical values ‘no' and ‘yes'
Hollidat_Package : categorical values ‘no' and ‘yes'

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

We will first go univariate analysis for each continuous column in the dataset.

1. Salary

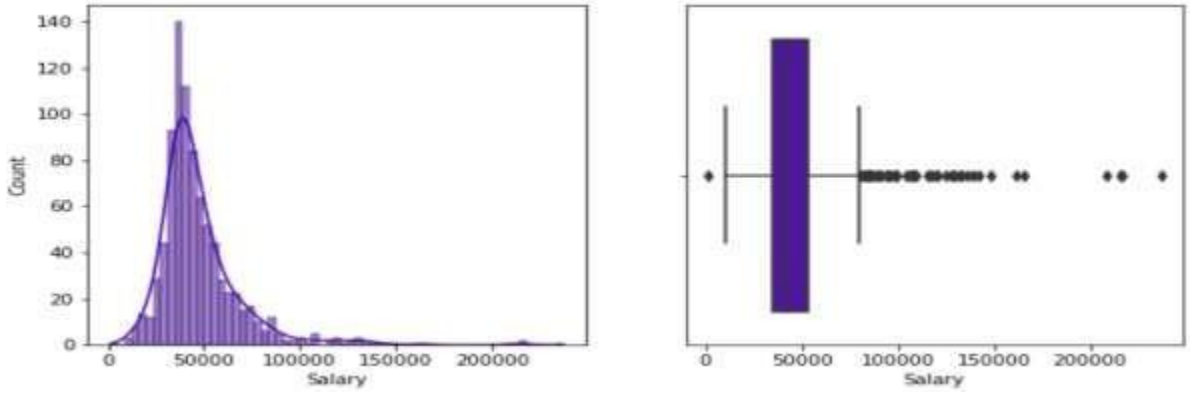


Fig no 1: histogram and boxplot for Salary

From above histogram and boxplot, the data point are normally distributed and having outliersin the column.

2. Age

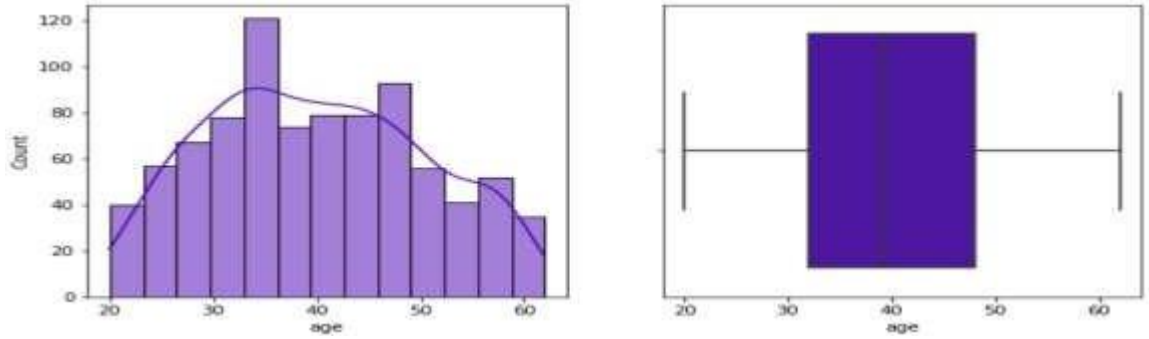


Fig no 2: histogram and boxplot for Age

From above histogram and boxplot, the data point are normally distributed and having no outliers in the column.

Bivariate Analysis

3.Educ

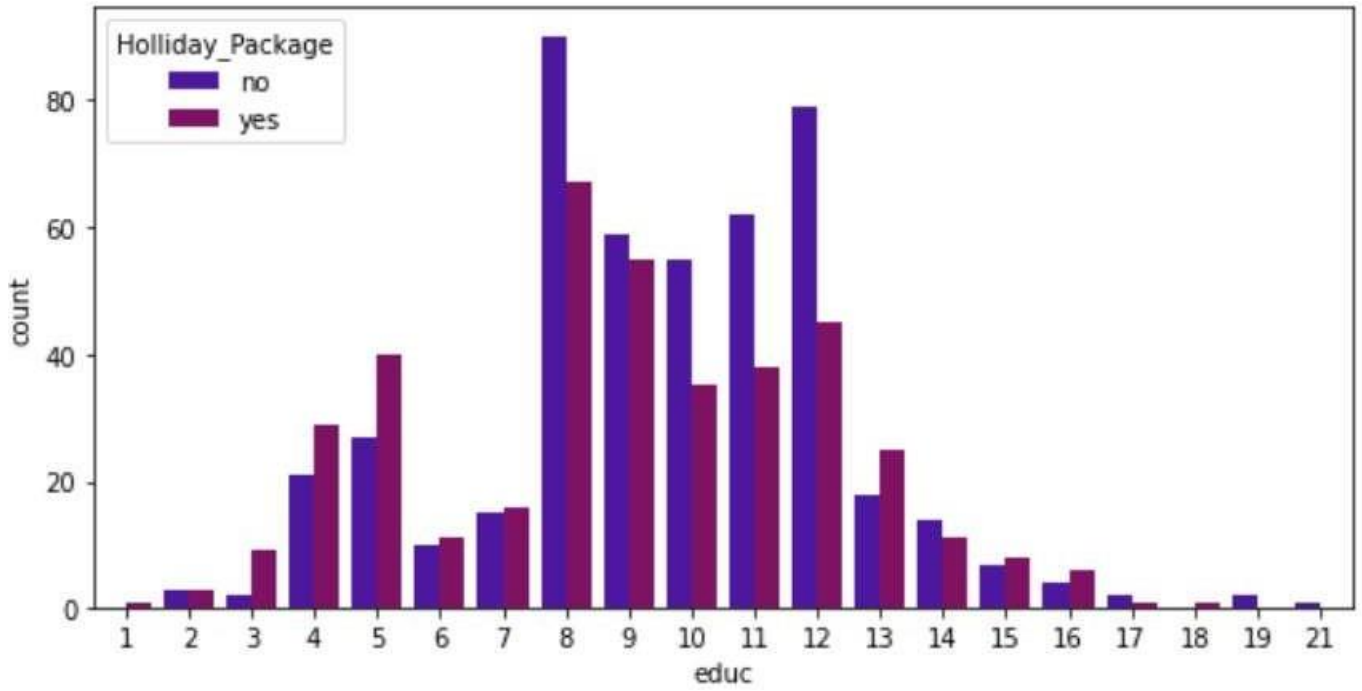


Fig no 3: count plot for Educ column

From above countplot , we are getting less information about whose is going to holiday. Those who has education 8 t0 12 years they moderately preferring no over yes but providing attractive offer they can be convince for the holiday.

3. no_young_children

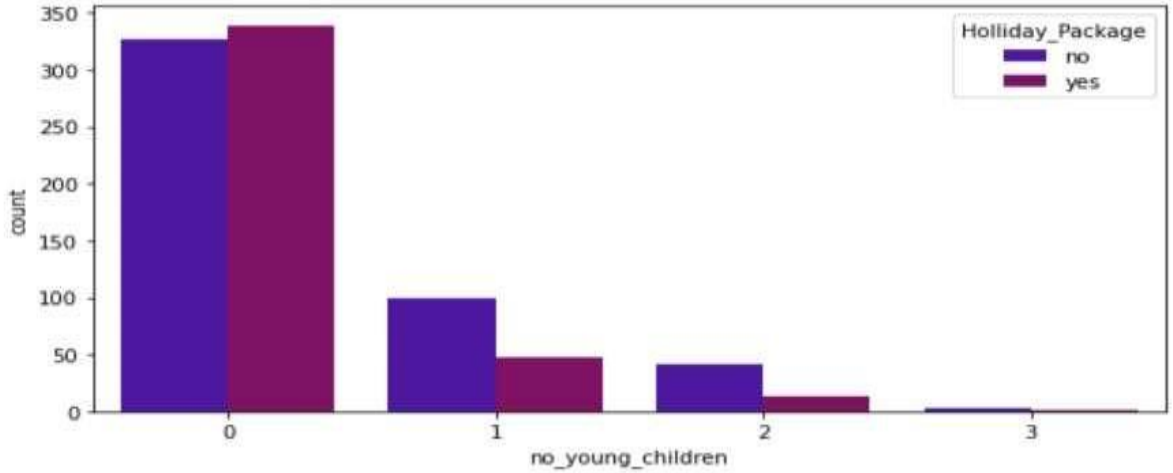


Fig no 4: Count plot for no_young_children column

From the above plot we have good amount of data having no young child half of thempreferring for holiday. Those whose are not preferring can given more discount to participate.

4. no_older_children

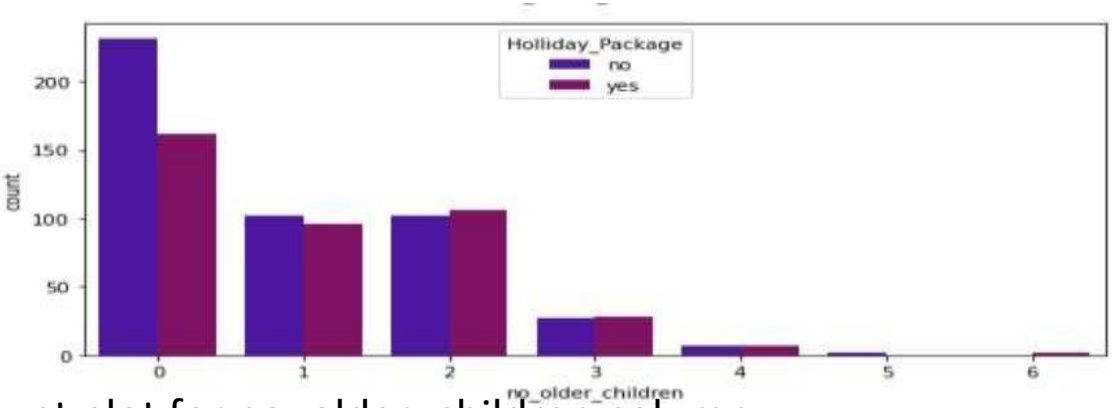


Fig no 5: Count plot for no_older_children column

From the above plot we have good amount of data having 0, 1, 2 older child half of them preferring for holiday. Those whose are not preferring can given more discount to participate.

Multivariate Analysis

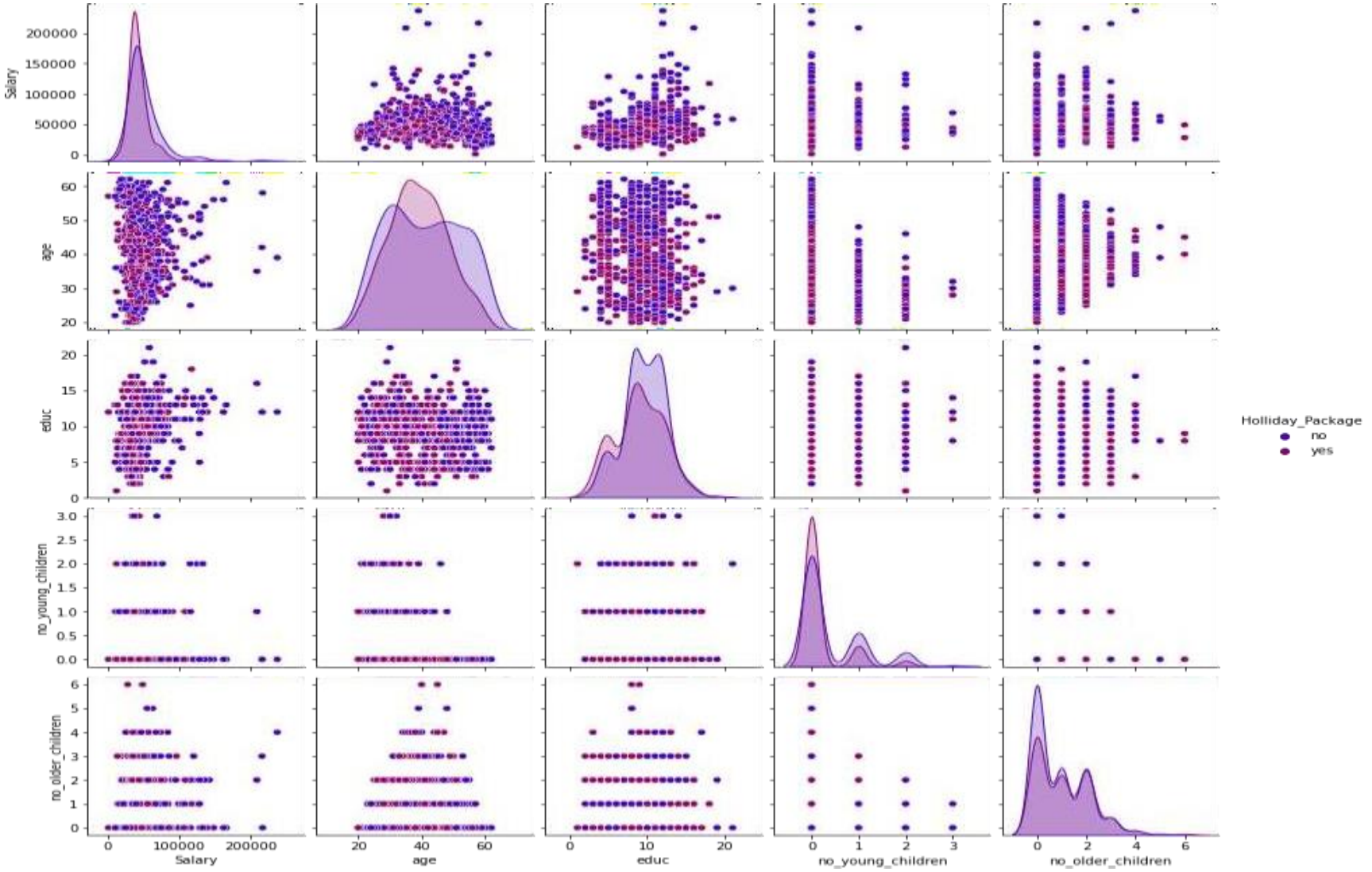


Fig no 6: Pair Plot

Correlation Plot

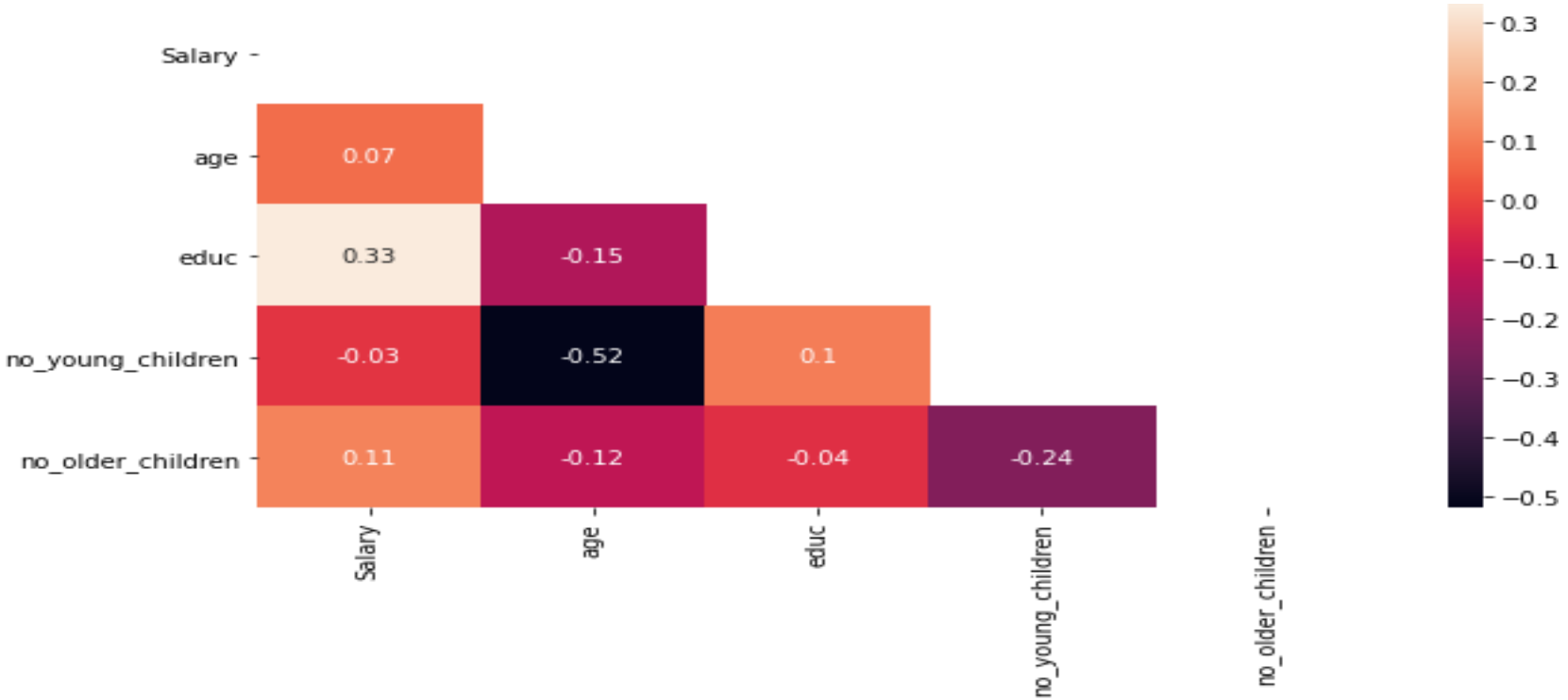


Fig no 7 : Correlation Plot

From the above correlation plot and pair plot,
Continuous columns showing less correlation with each other. No_young_children are negatively correlated with age.
From pair plot , having ager group between 30 to 50 preferring holidays and for new customer with the same age group can be attracted by good marketing and offer.

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)

Firstly we will remove outlier from our salary columns.

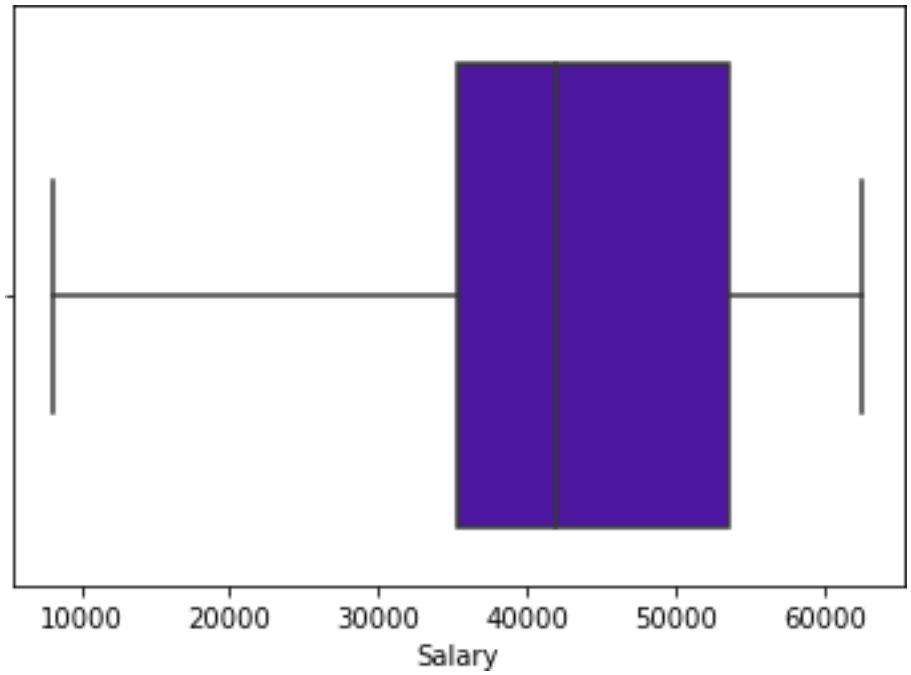


Fig no 8 : box Plot for salary column

Now, we will create dummies for foreign columns and impute 0 and 1 in place of no and yes in holiday package column respectively.

	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign_no	foreign_yes
1	0	48412.00	30	8	1	1	1	0
2	1	37207.00	45	8	0	1	1	0
3	0	58022.00	46	9	0	0	1	0
4	0	62542.25	31	11	2	0	1	0
5	0	62542.25	44	12	0	2	1	0

Fig no 9 : Sample of model data

We first need to create x(dependent) and y(independent) variable to split the data

```
# Copy all the predictor variables into x dataframe. Since 'Holliday_Package' is dependent variable drop it
x = df.drop('Holliday_Package', axis=1)

# Copy the 'Holliday_Package' column alone into the y dataframe. This is the dependent variable
y = df[['Holliday_Package']]
```

```
# 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)
```

Now, successfully split the data in the ratio 70:30

Now we will go for gridsearchCV and sufflesplit for implimentation of hyper parameter tuning on the Dataset.

```
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import ShuffleSplit
```

After implimentation of hyperpameter in Logistic regression and linear discriminant analysiswe have get the result for training and testing set.

For Training Set

	model	best_score	best_params
0	LogisticRegression	0.665574	{'penalty': 'l1', 'solver': 'liblinear', 'tol'...
1	LinearDiscriminantAnalysis	0.663388	{'n_components': 0, 'solver': 'svd', 'tol': 0....

Our model work with the score of 67% for Logistic Regression and 66% for Linear Discriminant Analysis

For Testing Set

	model	best_score	best_params
0	LogisticRegression	0.640506	{'penalty': 'l2', 'solver': 'newton-cg', 'tol'...
1	LinearDiscriminantAnalysis	0.643038	{'n_components': 0, 'solver': 'svd', 'tol': 0....

Our model work with the score of 64% for Logistic Regression and 64% for Linear Discriminant Analysis

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/optimize

Now we have get the best parameter for our Logistic regression model we will calculate other parameter accordingly

```
# checking for LogisticRegression
model = LogisticRegression(penalty='l2',tol=0.00001,solver='liblinear')
model.fit(x_train, y_train)
```

```
LogisticRegression(solver='liblinear', tol=1e-05)
```

Confusion matrix for train data

```
confusion_matrix(y_train,y_pred_train)
```

```
array([[252,  74],
       [125, 159]], dtype=int64)
```

Confusion matrix for test data

```
confusion_matrix(y_test,y_pred_test)
```

```
array([[104,  41],
       [ 50,  67]], dtype=int64)
```

Classification Report for Train data sample

	precision	recall	f1-score	support
0	0.67	0.77	0.72	326
1	0.68	0.56	0.62	284
accuracy			0.67	610
macro avg	0.68	0.67	0.67	610
weighted avg	0.67	0.67	0.67	610

Fig no 10 : Classification Report on Train data for Logistic Regression

So we can clearly understand that model performing 67% of accuracy but having 56% of recall score for yes kind of holiday_package on train data.

Classification Report Test data sample

	precision	recall	f1-score	support
0	0.68	0.72	0.70	145
1	0.62	0.57	0.60	117
accuracy			0.65	262
macro avg	0.65	0.64	0.65	262
weighted avg	0.65	0.65	0.65	262

Fig no 11 : Classification Report on Test data for Logistic Regression

So we can clearly understand that model performing 65% of accuracy but having 57% of recall score for yes kind of holiday_package on test data.

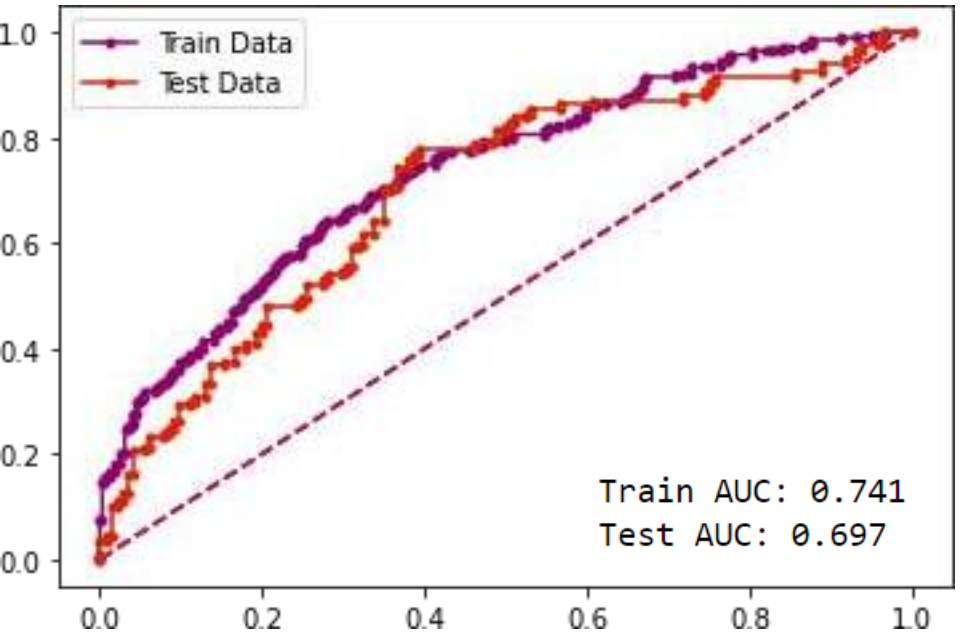


Fig no 12 : ROC_AUC_Curve for Logistic Regression model

From the above figure of ROC_AUC_Curve, AUC score for both train and test model are quite near so we can say that our model is robust.

Now we have get the best parameter for our Linear Discriminant Analysis model we willcalculate other parameter accordingly

```
clf = LinearDiscriminantAnalysis()
model=clf.fit(x_train,y_train)
model
```

```
LinearDiscriminantAnalysis()
```

Confusion matrix for train data

```
confusion_matrix(y_train,y_pred_train)
```

```
array([[254,  72],
       [125, 159]], dtype=int64)
```

Confusion matrix for test data

```
confusion_matrix(y_test,y_pred_test)
```

```
array([[100,  45],
       [ 51,  66]], dtype=int64)
```

Classification Report for Train data sample

	precision	recall	f1-score	support
0	0.67	0.78	0.72	326
1	0.69	0.56	0.62	284
accuracy			0.68	610
macro avg	0.68	0.67	0.67	610
weighted avg	0.68	0.68	0.67	610

Fig no 13 : Classification Report on Train data for Linear Discriminant Analysis

So we can clearly understand that model performing 68% of accuracy but having 56% of recall score for yes kind of holiday_package on train data.

Classification Report Test data sample

	precision	recall	f1-score	support
0	0.66	0.69	0.68	145
1	0.59	0.56	0.58	117
accuracy			0.63	262
macro avg	0.63	0.63	0.63	262
weighted avg	0.63	0.63	0.63	262

Fig no 14 : Classification Report on Test data for Linear Discriminant Analysis

So we can clearly understand that model performing 63% of accuracy but having 56% of recall score for yes kind of holiday_package on test data.

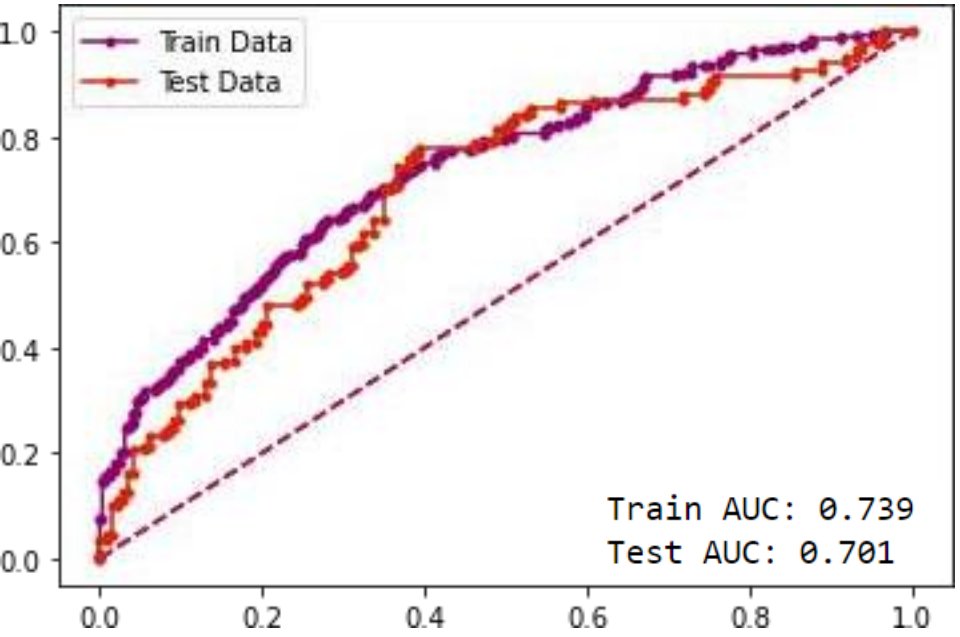


Fig no 12 : ROC_AUC_Curve for Logistic Regression model

From the above figure of ROC_AUC_Curve, AUC score for both train and test model are quite near so we can say that our model is robust.

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

From the analysis, performing Logistic Regression and Linear Discriminent analysis both the model giving the similar results in term of accuracy. We impute the catagorical column and performing EDA and we conclude results . We build model on train data and tested. We get therobust result.

Recommendation:

1)The age group between 30 to 50 preferring holidays . rest of the customer no taking that much of interest in holiday. By various attractive offers one can try this age group of customer for holidays.

2)The no_young_children and no_older_children column having 0 count can be attracted more for holidays already half of them are signing. By finding the appropriate reason we can attract more from this group of customer.

3)From income group 35000 to 50000 are too excited about holidays. By good marketing these customer flow can be increase to seek holiday package.

The End.....!