

# **PREDICTIVE MODELING PROJECT**

## **Data Analysis Report**

**Prepared By**

**JAI GOUTHAM**

**Submitted on**

**17-01-2021**

## **PROJECT OBJECTIVE**

### **Problem 1: Linear Regression**

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

**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.

**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? Justify

**1.3** Encode the data (having string values) for Modelling. Data Split: Split the data into test and train (70:30). Apply Linear regression. Performance Metrics: Check the performance of Predictions on Train and Test sets using R2, RMSE.

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

### **Problem 2: Logistic Regression and LDA**

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

**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.

**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).

**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.

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

## **PROBLEM 1: LINEAR REGRESSION**

**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.

### **Data Insights and EDA:**

The dataset: "cubic\_zircona.csv" which contains data of 26967 rows and 11 variables namely as follows:

#### **Data Dictionary:**

Variable Name	Description
Carat	Carat weight of the cubic zirconia.
Cut	Describe the cut quality of the cubic zirconia. Quality is increasing order Fair, Good, Very Good, Premium, Ideal.
Color	Colour of the cubic zirconia. Width D being the best and J the worst.
Clarity	cubic zirconia Clarity refers to the absence of the Inclusions and Blemishes. (In order from Best to Worst, FL = flawless, I3= level 3 inclusions) FL, IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1, I2, I3
Depth	The Height of a 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.

### Description of dataset:

The below table which shows the count, average, maximum, minimum values for the variables like carat, Depth, Table, X,Y,Z of the diamonds.

	count	mean	std	min	25%	50%	75%	max
carat	26933.0	0.798010	0.477237	0.2	0.40	0.70	1.05	4.50
depth	26933.0	61.745282	1.393848	50.8	61.10	61.80	62.50	73.60
table	26933.0	57.455950	2.232156	49.0	56.00	57.00	59.00	79.00
x	26933.0	5.729346	1.127367	0.0	4.71	5.69	6.55	10.23
y	26933.0	5.733102	1.165037	0.0	4.71	5.70	6.54	58.90
z	26933.0	3.537769	0.719964	0.0	2.90	3.52	4.04	31.80
price	26933.0	3937.526120	4022.551862	326.0	945.00	2375.00	5356.00	18818.00

### Checking for Missing Values:

The dataset does contain 697 missing/Null values in depth variable, and the null values are imputed using median values.

```
1 data_df.isnull().sum()
carat      0
cut        0
color      0
clarity    0
depth     697
table      0
x          0
y          0
z          0
price      0
dtype: int64
```

### Checking for Duplicate Values:

The dataset does contain 34 duplicated entry in the dataset and it is dropped.

```
dups = data_df.duplicated()
print('Number of duplicate rows = %d' % (dups.sum()))
#df[dups]

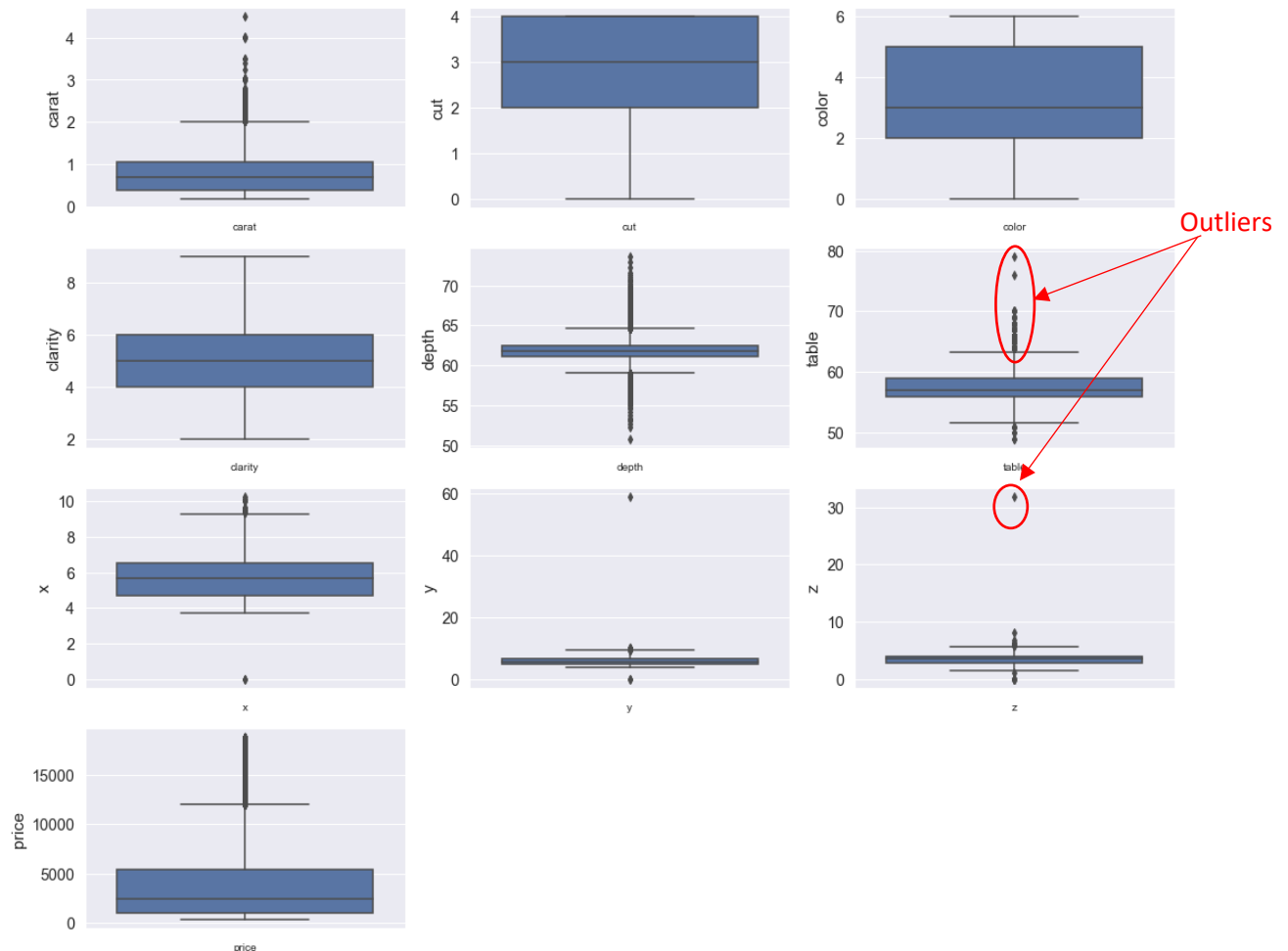
print('Before',df.shape)
data_df.drop_duplicates(inplace=True)
print('After',df.shape)
```

```
Number of duplicate rows = 34
Before (26967, 10)
After (26967, 10)
Number of duplicate rows = 0
```

## Univariate and Multivariate Analysis:

### Boxplot:

Using the Boxplot in a dataset we can able to find the outliers, spread of values, median, range, etc., (outliers are the extreme values present in the dataset)

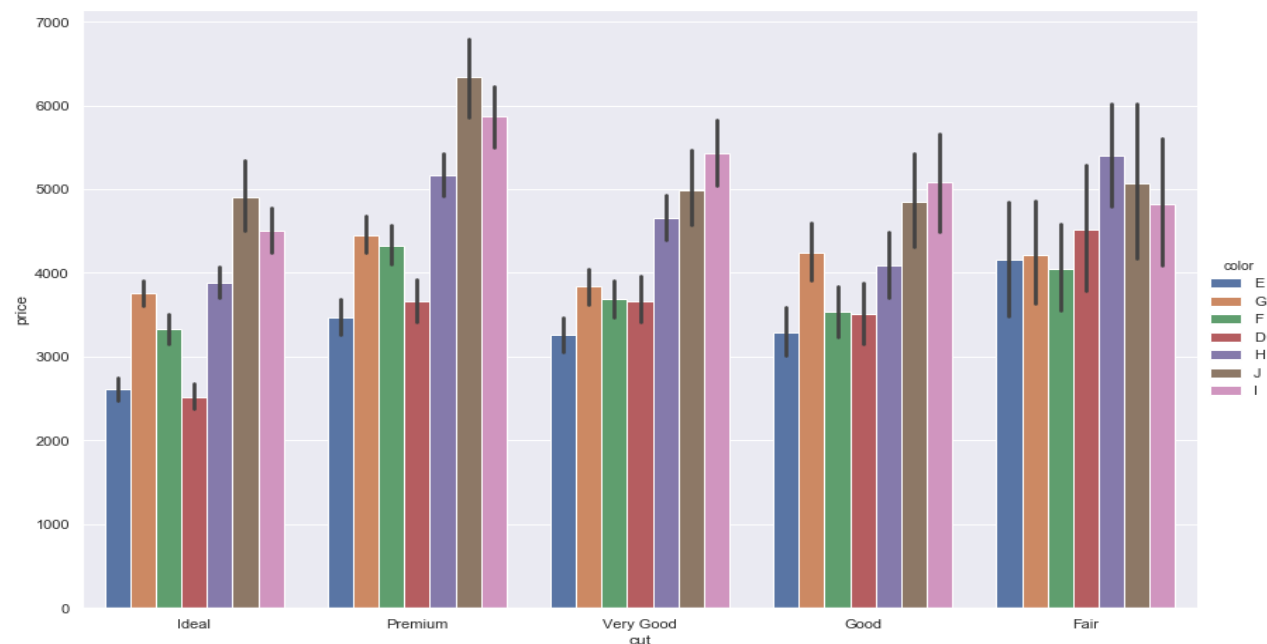
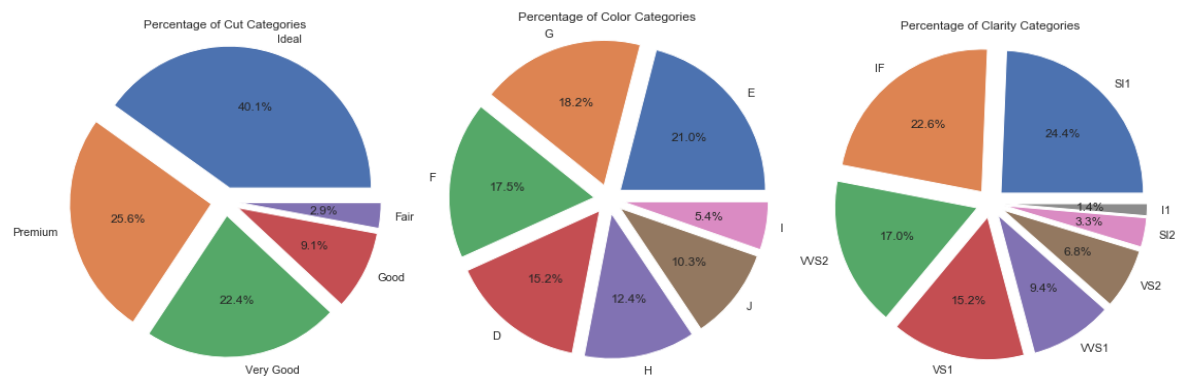
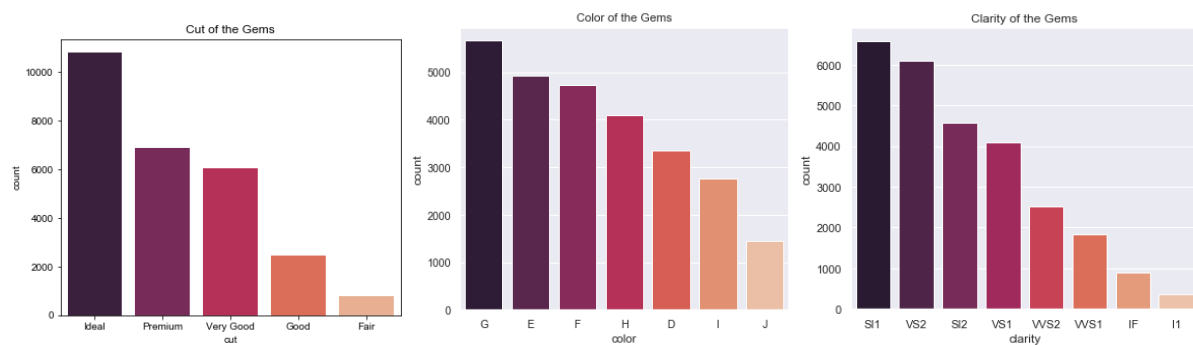


### Inference from boxplot:

From the above Boxplots for all the variables, we can conclude that most of the variables has outliers. There are many outliers present in the dataset, so significantly it may impact on our dataset, so we need to do outlier treatment.

- ❖ The customers in the dataset have higher spending capacity
- ❖ Most of the customers are making advance payments
- ❖ The probability of the customers for making full payment is more (86%-89%)
- ❖ Majority of the customers are maintaining the current balance in higher side.
- ❖ The maximum spending of customers in single shopping is high.

## countplot:

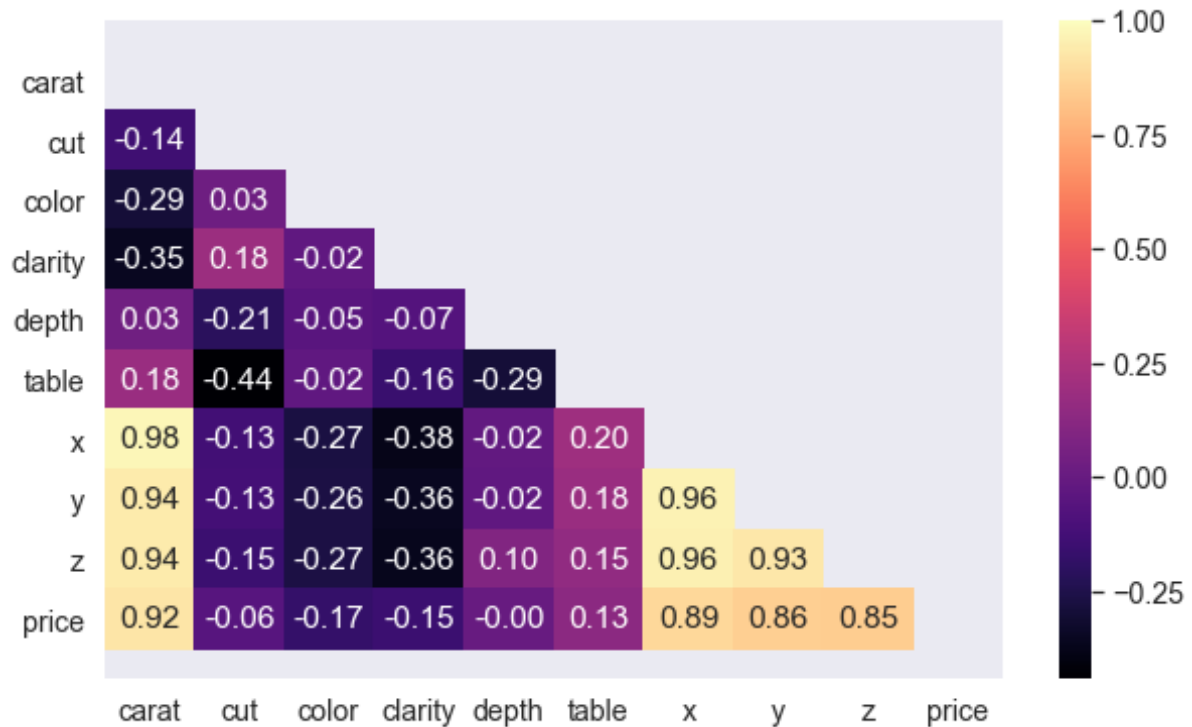


## Inference from bar plot:

- ❖ The Ideal cut-colour(D) which has **3K low price**, compare to Fair cut-colour(H)
- ❖ The Premium cut-colour(J) diamond has **highest Price**.
- ❖ The Ideal cut segment has **lowest price** when compare to other segments.
- ❖ Clarity SI2, Ideal cut, colour(G) has more counts, compare to other segments.
- ❖ Clarity I1, Fair cut, colour(J) has less counts, compare to other segments.

## Heat Map

The Heat Map shows the relationship between different variables in our dataset. This graph can help us to check for any correlations between different variables.

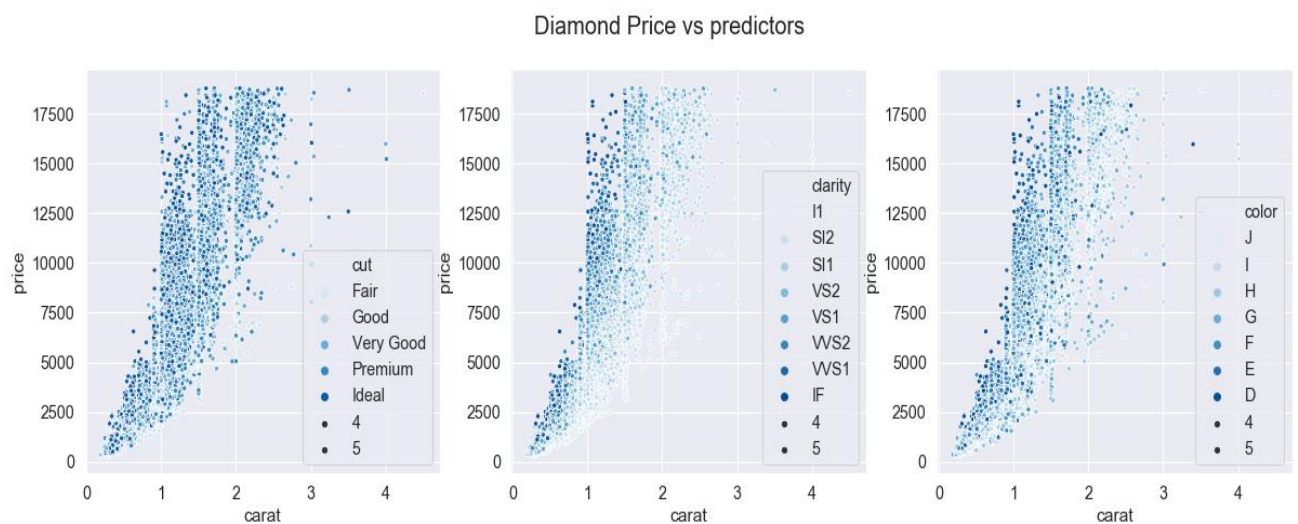


## Inference from heatmap:

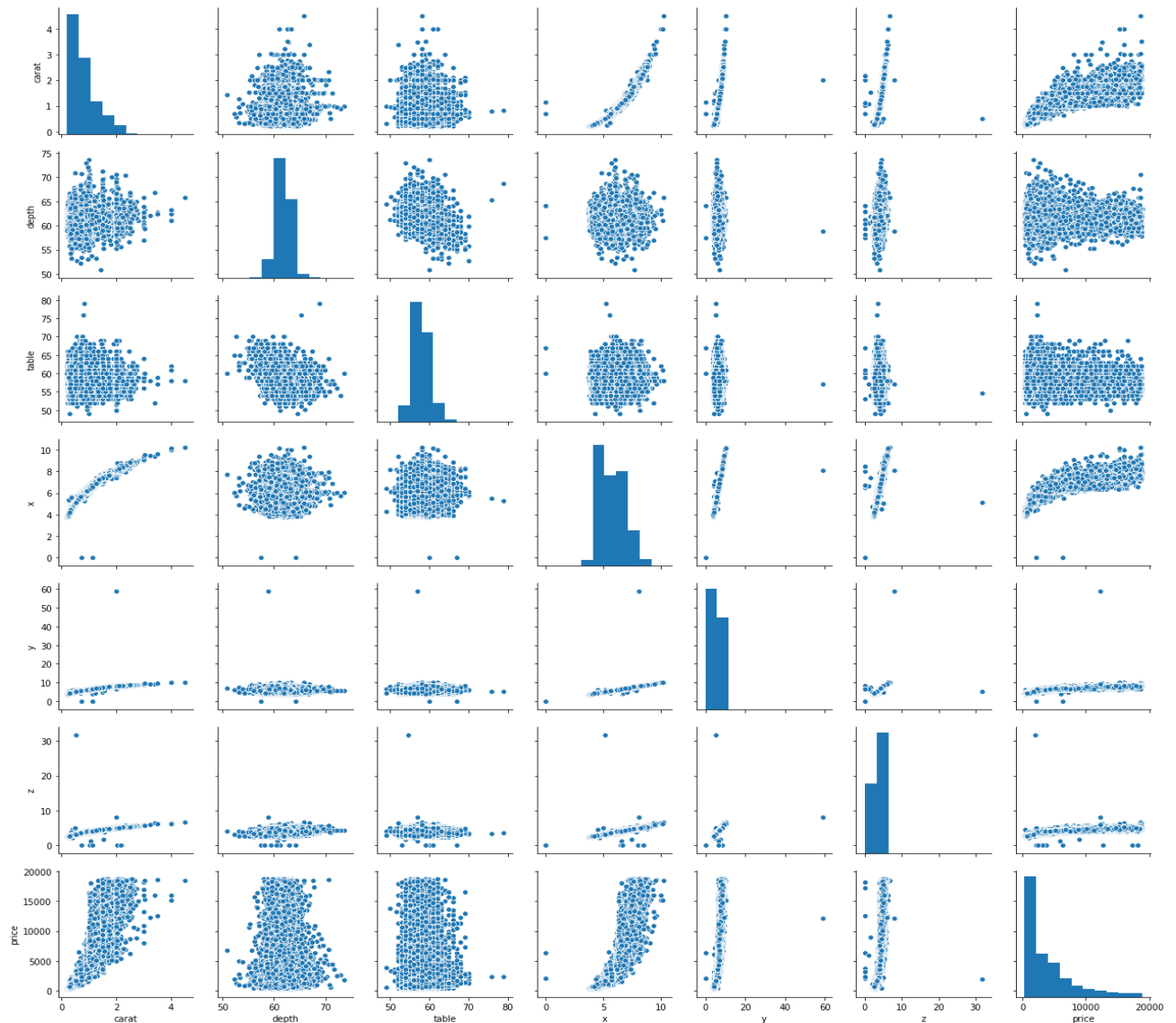
We can see that there is a high positive correlation between the variables.

- ❖ The variables like Price, x, y, z is highly positive correlated with Carat.
- ❖ The variables like clarity, colour is negatively correlated with Carat

## ScatterPlot:



## Pairplot:



## Inference from Univariate and multivariate analysis:

- Approximately 75% of the cubic zirconia stones weight between 0.20 and 1.05 carats.
- The depth of majority of cubic zirconia ranges between 60 and 62mm.
- Majority of the stones have Table value between 56 and 60.
- The average length of majority of zirconia stones lies between 4-7mm.
- The width of almost 75% of the stones ranges 3-10mm with maximum value of 58mm.
- The average height ranges between 3-6mm.
- The price being our target variable displays a right skewed graph with approximately 75% of the stones costing within the range of 945 to 5360 with the remaining percentage to be the premium stones costing more than 10,000.
- As per the above graphs, Ideal cut, SI1 clarity and colour E is found in abundance in our dataset.



**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? Justify

```
1 data_df.isnull().sum()

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

```
1 # Replacing the nan values in depth column with mean value.
2 data_df.depth.replace( to_replace=np.nan,
3     value=data_df.depth.mean(),
4     inplace=True)
```

The dataset contains 697 null values in depth variable, and the values imputed with mean of their respective variable. The below table shows the zero values present in the z variable which has been replaced by mean of the respective columns.

	carat	cut	color	clarity	depth	table	x	y	z	price
5821	0.71	1.0	4.0	3.0	64.1	60.0	0.00	0.00	0.0	2130
6034	2.02	3.0	2.0	5.0	62.7	53.0	8.02	7.95	0.0	18207
10827	2.20	3.0	2.0	4.0	61.2	59.0	8.42	8.37	0.0	17265
12498	2.18	3.0	2.0	3.0	59.4	61.0	8.49	8.45	0.0	12631
12689	1.10	3.0	3.0	3.0	63.0	59.0	6.50	6.47	0.0	3696
17506	1.14	0.0	3.0	6.0	57.5	67.0	0.00	0.00	0.0	6381
18194	1.01	3.0	2.0	2.0	58.1	59.0	6.66	6.60	0.0	3167
23758	1.12	3.0	3.0	2.0	60.4	59.0	6.71	6.67	0.0	2383

### The Coefficient and intercept before scaling:

The coefficient for carat is 8854.793190267937  
 The coefficient for cut is 111.12983692619733  
 The coefficient for color is 277.6809260988709  
 The coefficient for clarity is 438.8413337017509  
 The coefficient for depth is 33.676323624533865  
 The coefficient for table is -12.517224548596108  
 The coefficient for x is -1174.1941772944256  
 The coefficient for y is 1376.6155979335215  
 The coefficient for z is -927.9868637533705

```
1 reg_model.intercept_

-6010.66788430174
```

### The Coefficient and intercept after scaling:

```
The coefficient for carat is 1.6232591768620688
The coefficient for cut is 0.012704180932410249
The coefficient for color is -0.12078069680902284
The coefficient for clarity is 0.1185091398305296
The coefficient for depth is -0.02688362074722267
The coefficient for table is -0.052491498734117084
The coefficient for x is -0.8449759932129506
The coefficient for y is 0.5224085316210725
The coefficient for z is -0.3194063291081761
```

The intercept for our model is  $-7.329286690089521e-16$

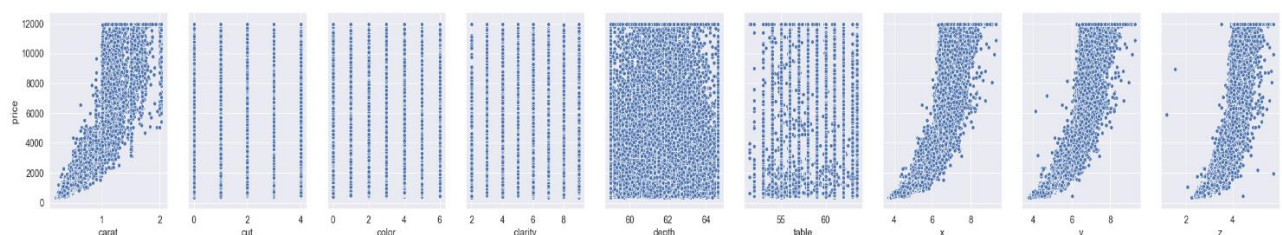
### Inference from Above analysis:

- Model score - R2 or coefficient of determinant was 90% and did not increase after scaling.
- Intercept becomes zero when we standardize variable. Z-score scaling does not much improve the accuracy score but removes the intercept
- Centring of data using Z-Score and building the model has no impact on accuracy score but coefficient will change because of scaled data
- Hence concluding that scaling is not required for this case.

## **Assumptions for Linear Regression:**

### 1. Linearity

- Linear regression needs the relationship between the independent and dependent variables to be linear. Let's use a pair plot to check the relation of independent variables with the Price variable
- visualize the relationship between the features and the response using scatterplots



By looking at the plots we can see that with the Price vs (carat, X,Y,Z) it forms an linear shape but the other variables which seems hardly have no specific shape. So it shows that a linear regression model may not best model for it. A linear model might not be able to *efficiently* explain the data in terms of variability, prediction accuracy etc.

Now rest of the assumptions require us to perform the regression before we can even check for them. So let's perform regression on it.

## 2.Mean of Residuals

Residuals as we know are the differences between the true value and the predicted value. One of the assumptions of linear regression is that the mean of the residuals should be zero. So let's find out.

```
1 residuals = y_train.values-y_pred
2 mean_residuals = np.mean(residuals)
3 print("Mean of Residuals {}".format(mean_residuals))
```

Mean of Residuals -1.9944915371611916e-12

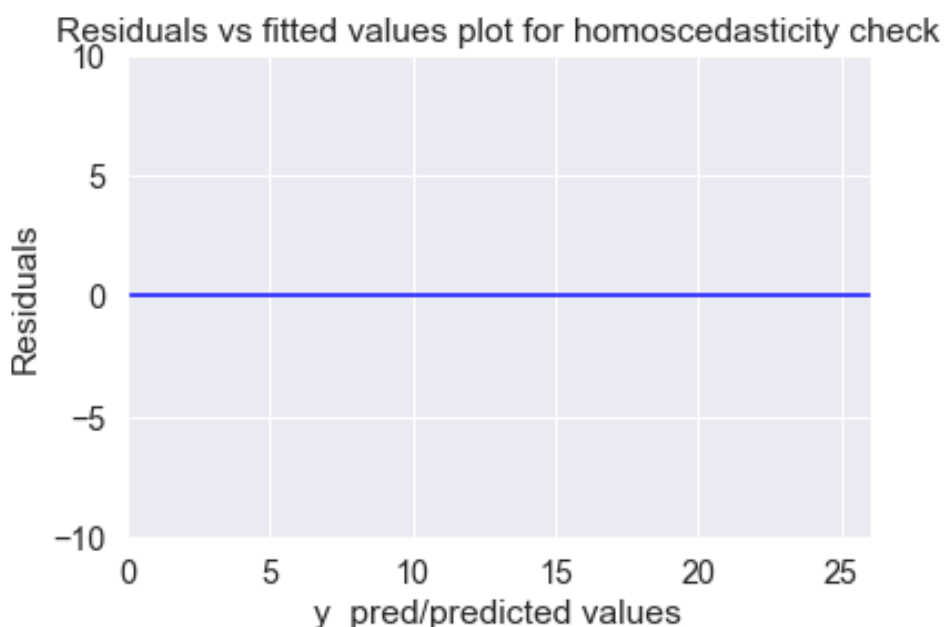
The mean of residuals is close to zero.

## 3.Check for homoscedasticity

Homoscedasticity means that the residuals have equal or almost equal variance across the regression line. By plotting the error terms with predicted terms we can check that there should not be any pattern in the error terms.

### **Detecting heteroscedasticity!**

Graphical Method: Firstly do the regression analysis and then plot the error terms against the predicted values ( $\hat{Y}_i$ ). If there is a definite pattern (like linear or quadratic or funnel shaped) obtained from the scatter plot then heteroscedasticity is present.



Visually there is no Patterns found in the Scatter plot, so the heteroscedasticity is not found.

## Goldfeld Quandt Test

Checking heteroscedasticity : Using Goldfeld Quandt we test for heteroscedasticity.

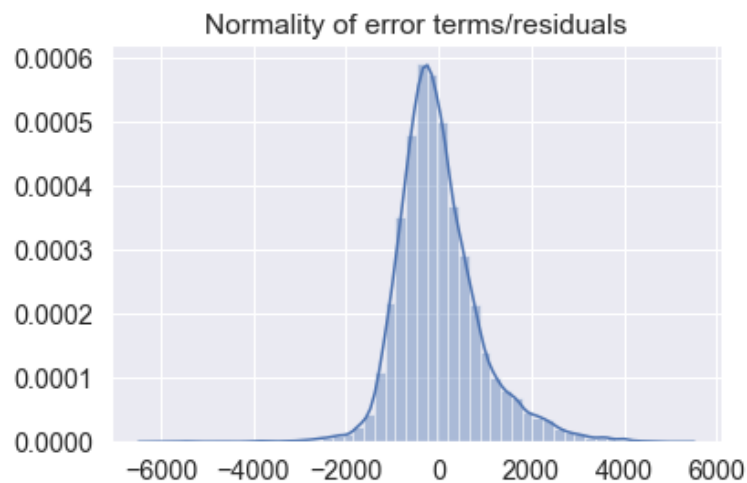
- Null Hypothesis: Error terms are homoscedastic
- Alternative Hypothesis: Error terms are heteroscedastic.

```
1 import statsmodels.stats.api as sms
2 from statsmodels.compat import lzip
3 name = ['F statistic', 'p-value']
4 test = sms.het_goldfeldquandt(residuals, X_train)
5 lzip(name, test)
```

```
[('F statistic', 0.9726086328552119), ('p-value', 0.9111008601114475)]
```

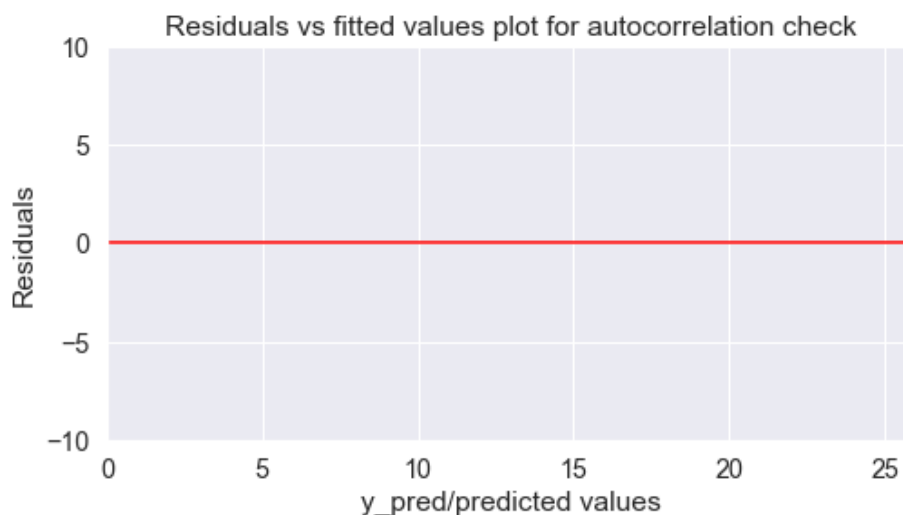
Since p value is more than 0.05 in Goldfeld Quandt Test, we can't reject it's null hypothesis that error terms are homoscedastic. Good.

## 4.Check for normality of the residuals



The error terms/residuals is normally distributed from the above graph.

## 5.Check for auto correlations



When the residuals are autocorrelated, it means that the current value is dependent of the previous (historic) values and that there is a definite unexplained pattern in the Y variable that shows up in the error terms. Though it is more evident in time series data. In plain terms autocorrelation takes place when there's a pattern in the rows of the data

There should not be autocorrelation in the data so the error terms should not form any pattern.

## 6. Check for multicollinearity

In regression, multicollinearity refers to the extent to which independent variables are correlated. Multicollinearity affects the coefficients and p-values, but it does not influence the predictions, precision of the predictions, and the goodness-of-fit statistics. If your primary goal is to make predictions, and you don't need to understand the role of each independent variable, you don't need to reduce severe multicollinearity.



The multicollinearity can be removed by aping PCA, a dimension reduction technique which reduces the dimension and it creates the variable with no linear relationship with the independent variables.

**The variable creation technique which also helps in increase the reduction of multicollinearity. In this a new variable can be created like  $\text{Volume} = X \cdot Y \cdot Z$  variables so that the new variables can be formed which reduces the correlation between them.**

**1.3 Encode the data (having string values) for Modelling. Data Split: Split the data into test and train (70:30). Apply Linear regression. Performance Metrics: Check the performance of Predictions on Train and Test sets using R2, RMSE.**

From the below table the categorical data is encoded with numerical data, the label encoding has done due to the ordinal data present in Cut, Colour and Clarity variables.

```
1 Cut={'Fair':0,'Good':1,'Very Good':2,'Premium':3,'Ideal':4}
2 Color={'D':6,'E':5,'F':4,'G':3,'H':2,'I':1,'J':0}
3 Clarity={'IF':9,'VVS1':8,'VVS2':7,'VS1':6,'VS2':5,'SI1':4,'SI2':3,'I1':2}
```

```
1 #Converting Object column to numerical column
2 data_df.replace({'cut':Cut},inplace=True)
3 data_df.replace({'color':Color},inplace=True)
4 data_df.replace({'clarity':Clarity},inplace=True)
```

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

```
1 data_df.head()
```

	carat	cut	color	clarity	depth	table	x	y	z	price
0	0.30	4	5	4	62.1	58.0	4.27	4.29	2.66	499
1	0.33	3	3	9	60.8	58.0	4.42	4.46	2.70	984
2	0.90	2	5	7	62.2	60.0	6.04	6.12	3.78	6289
3	0.42	4	4	6	61.6	56.0	4.82	4.80	2.96	1082
4	0.31	4	4	8	60.4	59.0	4.35	4.43	2.65	779

```
1 # Separating INDEPENDENT AND DEPENDENT VARIABLE
2
3 X=data_df.drop('price',axis=1)
4 y=data_df.price
```

Splitting the dataset and dropping the dependent variable price for X.

```

1 from sklearn.model_selection import train_test_split
2 X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.30,random_state=1)

```

## Linear Regression Model

```

1 reg_model = LinearRegression()
2 reg_model.fit(X_train, y_train)
3 y_pred=reg_model.predict(X_train)

```

The coefficient for carat is 8854.793190267937  
 The coefficient for cut is 111.12983692619733  
 The coefficient for color is 277.6809260988709  
 The coefficient for clarity is 438.8413337017509  
 The coefficient for depth is 33.676323624533865  
 The coefficient for table is -12.517224548596108  
 The coefficient for x is -1174.1941772944256  
 The coefficient for y is 1376.6155979335215  
 The coefficient for z is -927.9868637533705

```
1 reg_model.intercept_
```

-6010.66788430174

```

1 #R-square on train dataset
2 reg_model.score(X_train,y_train)

```

0.9312964828856415

```

1 #R-square on test dataset
2 reg_model.score(X_test,y_test)

```

0.9313549570011984

```
1 print("R squared: {}".format(r2_score(y_true=y_train,y_pred=y_pred)))
```

R squared: 0.9312964828856415

The Performance metrics for the Linear regression was given below

```

('Mean Absolute Error: 662.6894296293423',
 'Mean Square Error: 828449.3713730423',
 'r2 score: 0.9313549570011984')

```

```

1 #RMSE on train dataset
2 predicted_train=reg_model.fit(X_train,y_train).predict(X_train)
3 mse=metrics.mean_squared_error(y_train,predicted_train)
4 rmse=np.sqrt(mse)
5 rmse

```

908.3897871803956

```

1 #RMSE on test dataset
2 predicted_train=reg_model.fit(X_test,y_test).predict(X_test)
3 mse1=metrics.mean_squared_error(y_test,predicted_train)
4 rmse1=np.sqrt(mse1)
5 rmse1

```

909.2258190308854

**Training & Test model score is almost same approximately equal to 93% so our model is in Right fitting zone.**

## Linear Regression using statsmodel

```

1 # concat X and Y in single dataframe
2 train=pd.concat([X_train,y_train],axis=1)
3 test=pd.concat([X_test,y_test],axis=1)

```

```
1 formula='price ~ carat + cut + color + clarity + depth + table + x + y + z'
```

```
1 import statsmodels.formula.api as smf
```

```

1 lml=smf.ols(data=train,formula=formula).fit()
2 lml.params

```

```

Intercept    -5683.078177
carat         8883.022072
cut           110.286689
color         278.250529
clarity       440.209865
depth         29.187620
table        -12.878704
x            -1244.635360
y             1414.711025
z            -891.509163
dtype: float64

```



```
1 print(lml.summary())
```

```

                                OLS Regression Results
=====
Dep. Variable:                  price    R-squared:                  0.931
Model:                            OLS    Adj. R-squared:              0.931
Method:                 Least Squares    F-statistic:                2.838e+04
Date:                Thu, 14 Jan 2021    Prob (F-statistic):          0.00
Time:                18:08:49            Log-Likelihood:             -1.5517e+05
No. Observations:          18853        AIC:                        3.104e+05
Df Residuals:              18843        BIC:                        3.104e+05
Df Model:                    9
Covariance Type:            nonrobust
=====
               coef      std err          t      P>|t|      [0.025      0.975]
-----
Intercept    -5683.0782     798.143     -7.120     0.000    -7247.509    -4118.647
carat         8883.0221      82.435    107.758     0.000     8721.443     9044.601
cut           110.2867       7.347     15.012     0.000       95.887     124.687
color         278.2505       4.108     67.737     0.000      270.199     286.302
clarity       440.2099       4.458     98.747     0.000      431.472     448.948
depth         29.1876      11.002       2.653     0.008       7.623     50.752
table        -12.8787       3.904     -3.299     0.001     -20.531     -5.226
x            -1244.6354     123.427    -10.084     0.000    -1486.563    -1002.708
y             1414.7110     121.629     11.631     0.000     1176.307     1653.115
z             -891.5092     136.647     -6.524     0.000    -1159.350     -623.668
=====
Omnibus:                 2744.027    Durbin-Watson:              1.989
Prob(Omnibus):            0.000    Jarque-Bera (JB):           9116.497
Skew:                     0.738    Prob(JB):                    0.00
Kurtosis:                 6.070    Cond. No.                   1.04e+04
=====

```

**Note: Check the Python File for the Algorithms and Graphs.**

	Algorithms	R2_Scores	Mean_square_error	Mean_absolute_error
0	Linear Regression	0.931355	8.284494e+05	662.689430
1	Lasso Regression	0.922174	9.392545e+05	654.095324
2	AdaBoost Regression	0.936743	7.634237e+05	670.201163
3	Ridge Regression	0.843874	1.884224e+06	975.080468
4	RandomForest Regression	0.931355	1.555166e+05	212.125947
5	KNeighbours Regression	0.960671	4.746443e+05	412.298787

```

1 for i,j in np.array(lml.params.reset_index()):
2     print('{} * {} +'.format(round(j,2),i),end=' ')

```

```

(-5683.08) * Intercept + (8883.02) * carat + (110.29) * cut + (278.25) * color + (440.21) * clarity + (29.19) * depth + (-12.8
8) * table + (-1244.64) * x + (1414.71) * y + (-891.51) * z +

```

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

The final Linear Regression equation is

$$\text{price} = b_0 + b_1 * \text{carat} + b_2 * \text{cut} + b_3 * \text{color} + b_4 * \text{clarity} + b_5 * \text{depth} + b_6 * \text{table} + b_7 * x + b_8 * y + b_9 * z$$

$$\text{price} = (-5180.55) * \text{Intercept} + (8863.62) * \text{carat} + (107.12) * \text{cut} + (271.91) * \text{colour} + (433.46) * \text{clarity} + (24.51) * \text{depth} + (-14.97) * \text{table} + (-1172.05) * x + (1308.01) * y + (-842.55) * z$$

When the carat increases by 1 unit, the diamond price increases by 8863 units, keeping all other predictors constant. Similarly, when the colour increases by 1 unit, the diamond price increases by 272 units, keeping all other predictors constant.

There are also some negative co-efficient values, for instance, table has its corresponding co-efficient as -1172. This implies, when the table is expressed, the price decreases by -1172 units, keeping all other predictors constant.

### Inference and Business Recommendation:

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.

Important attributes in the order of priority

1. Carat (Carat weight of the cubic zirconia) – (8863.62 units) Positive coefficient
2. clarity (Colour of the cubic zirconia) – (433.46 units) Positive coefficient
3. y (Width of the cubic zirconia in mm) – (1308.01 units) Positive coefficient
4. x (Length of the cubic zirconia in mm.) - (-1172.12 units) Negative coefficient
5. z (Height of the cubic zirconia in mm) – (-842.55) Negative coefficient

Our business recommendation is to focus on optimising the combination of increased Carat weight of the cubic zirconia (since for every 1 unit of increase, the price gets increased to 8863.62 units) as much as possible concurrently reducing the length of the cubic zirconia (for every reduction of 1 unit of length the price gets decreased by 1172.12 units) with comparative stress on carat weight rather than length.

The business study also needs to focus on optimising increased carat weight of the cubic zirconia concurrently increasing the width of the cubic zirconia; as there is high correlation of carats with that of the width and the price is also effectively influenced by raise in width of the cubic zirconia (for every unit of width the price gets increased by 1308.01 units).

It is also noticed that length and height of the cubic zirconia too has negative correlation with the carat weight variable. Therefore, the business study also needs to have an effective optimisation technique to optimize the height and length of the cubic zirconia concurrently increasing the carat weight of the cubic zirconia. (for every unit of height and length, the price gets decreased by 842.55 units and 1172.12 units).

It needs to be recommended to the business that table attribute of the cubic zirconia do have some influence on the price variable however **not so significant to focus as much time and resource as in the above cases**. It is also to be noted that the price gets decreased only by 14.97 units for every unit of change in depth and increase of 24.51 units for table variable.

For higher profit share, the company's spread out of rise in price should be fixed in accordance with the rise in carat weight & width of the cubic zirconia taking into consideration the reduced height & length simultaneously.

Similarly, for higher profit share, the spread out of reduced-price fixation must be in accordance with reduced carat weight & width of the cubic zirconia taking into account the need for fixing lesser price on cubic zirconia of increased height and length.

## **PROBLEM 2: Logistic Regression and LDA**

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

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.

### **Data Insights and EDA:**

The dataset provided to us is stored as "Holiday\_Package.csv" which contains data of 872 employees and 7 variables namely:

Variable Name	Description
Holiday_Package	Opted for Holiday Package yes/no?
Salary	Employee salary
age	Age in years
educ	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

### **Description of dataset:**

The below table which shows the count, average, maximum, minimum values for the variables like Salary, Age, Education, Number of young children, no of older children.

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

### Checking for Missing Values:

The dataset does not contain any Null values. The total no of missing /Null values is 0.

```
1 data_df.isna().sum()
Holliday_Package    0
Salary              0
age                 0
educ                0
no_young_children   0
no_older_children   0
foreign             0
dtype: int64
```

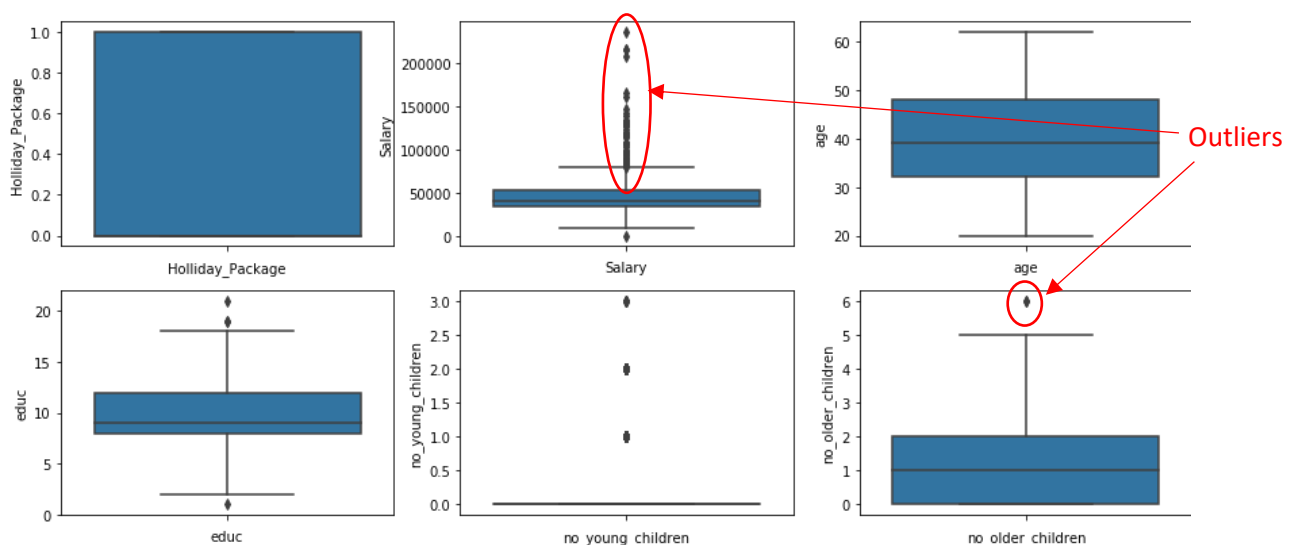
### Checking for Duplicate Values:

```
1 # Are there any duplicates ?
2 dups = data_df.duplicated()
3 print('Number of duplicate rows = %d' % (dups.sum()))
4 #df[dups]
5
6 print('Before',data_df.shape)
7 data_df.drop_duplicates(inplace=True)
8
9
10 dups = data_df.duplicated()
11 print('Number of duplicate rows = %d' % (dups.sum()))
```

```
Number of duplicate rows = 0
Before (872, 7)
Number of duplicate rows = 0
```

### Univariate Analysis:

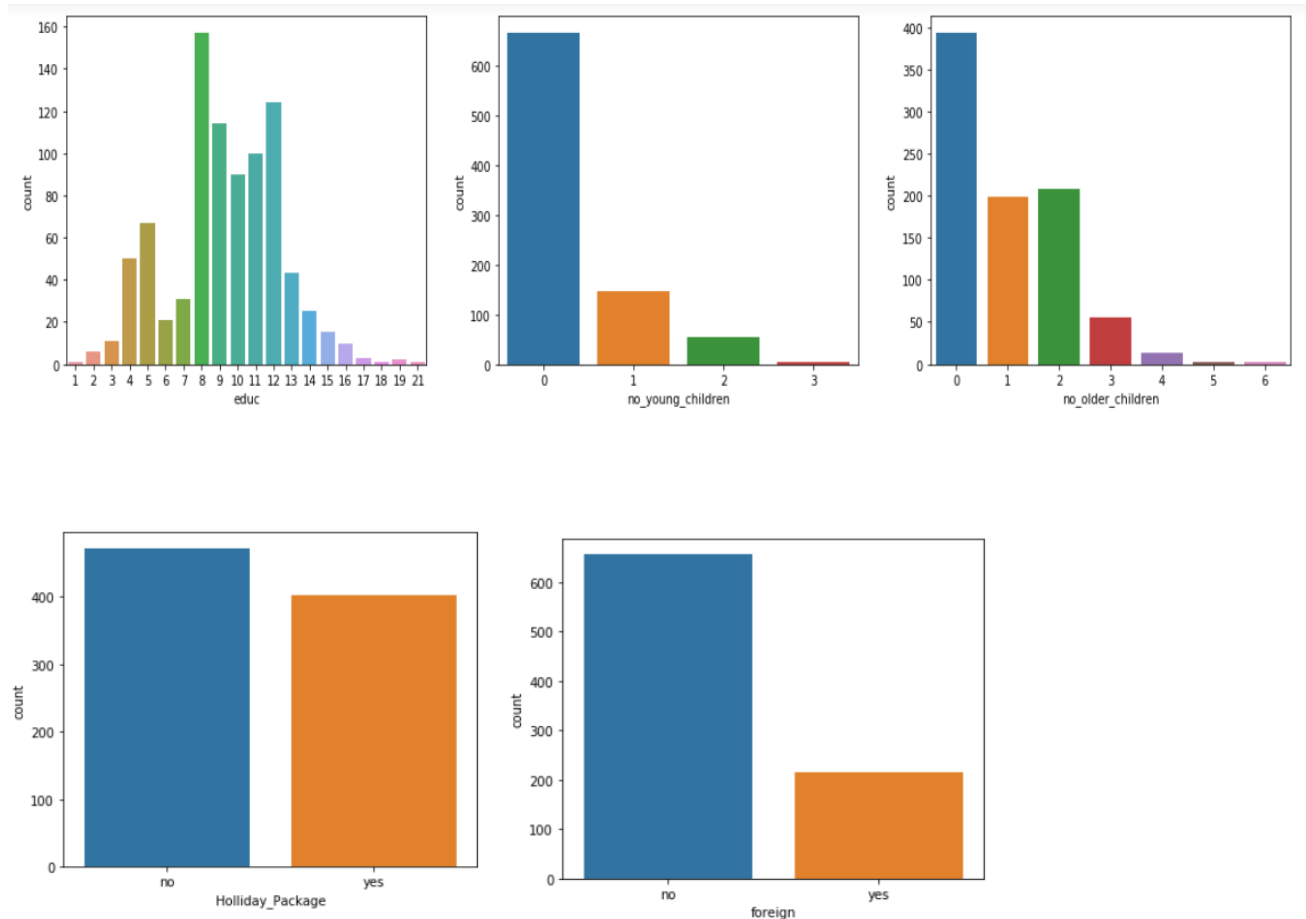
Using the Boxplot in a dataset we can able to find the outliers, spread of values, median, range, etc., (outliers are the extreme values present in the dataset)



### Inference from boxplot:

From the above Boxplots for the variables like Salary, Education, no of young children no of older children has outliers present in it. So we need to do outlier treatment for the given dataset.

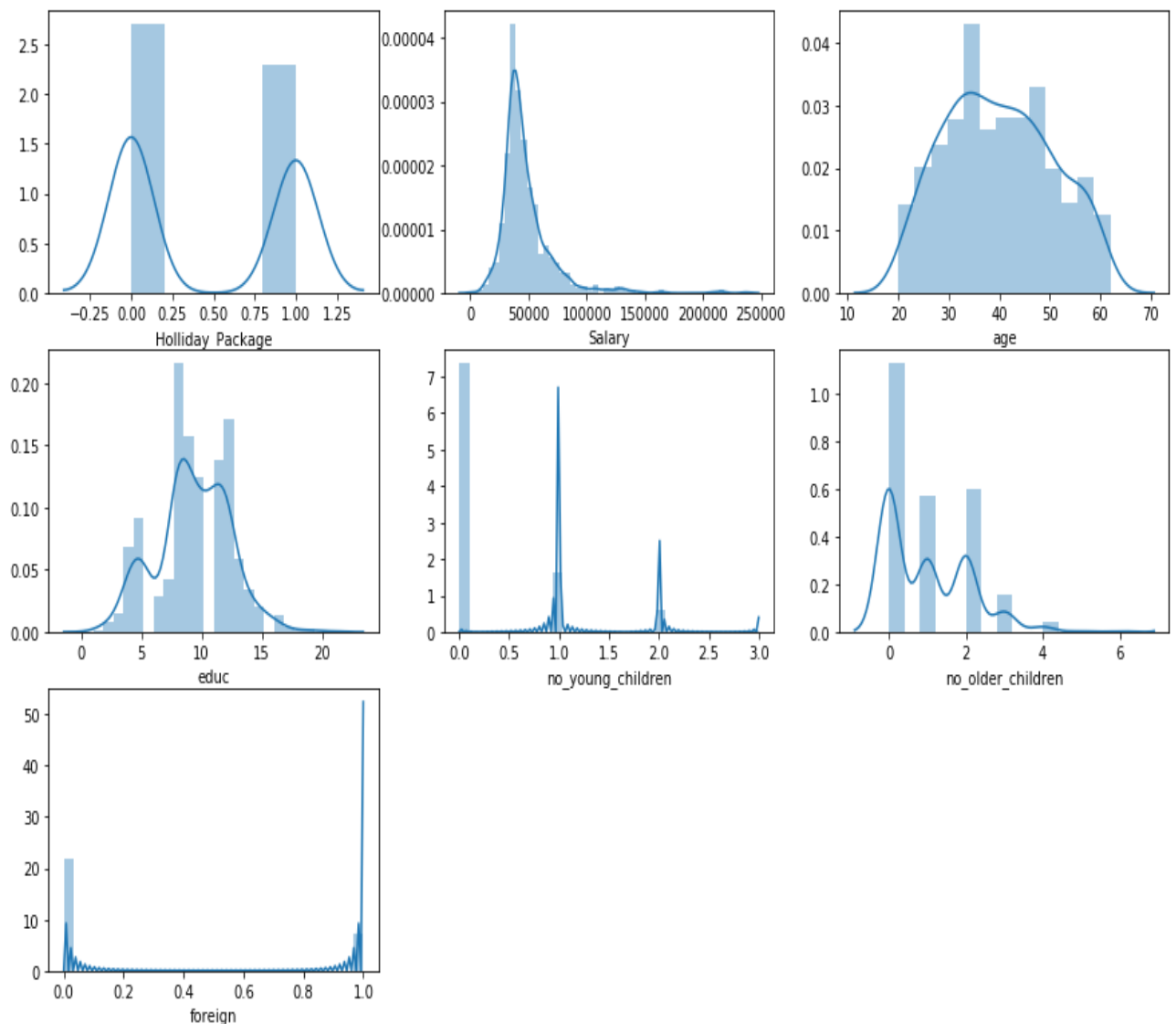
### CountPlot:



### Inference from count plot:

- ❖ Approximately 75% of the employees has education ranging between class 8 to 12.
- ❖ Approximately 75% of the employees has “0” young children.
- ❖ Approximately 45% of the employees has “0” older children.
- ❖ The average years of education of majority of employees range between 3-15 years.
- ❖ Majority of the employees have no children younger than 7 years of age.
- ❖ Majority of the employees have on an average 0-2 children older than 7 years of age.
- ❖ Majority of the employees are not foreigners.
- ❖ The Holliday Package variable is our dependent variable with 471 employees not opting for holiday package and only 401 employees who have opted for the holiday package.

## Histogram:



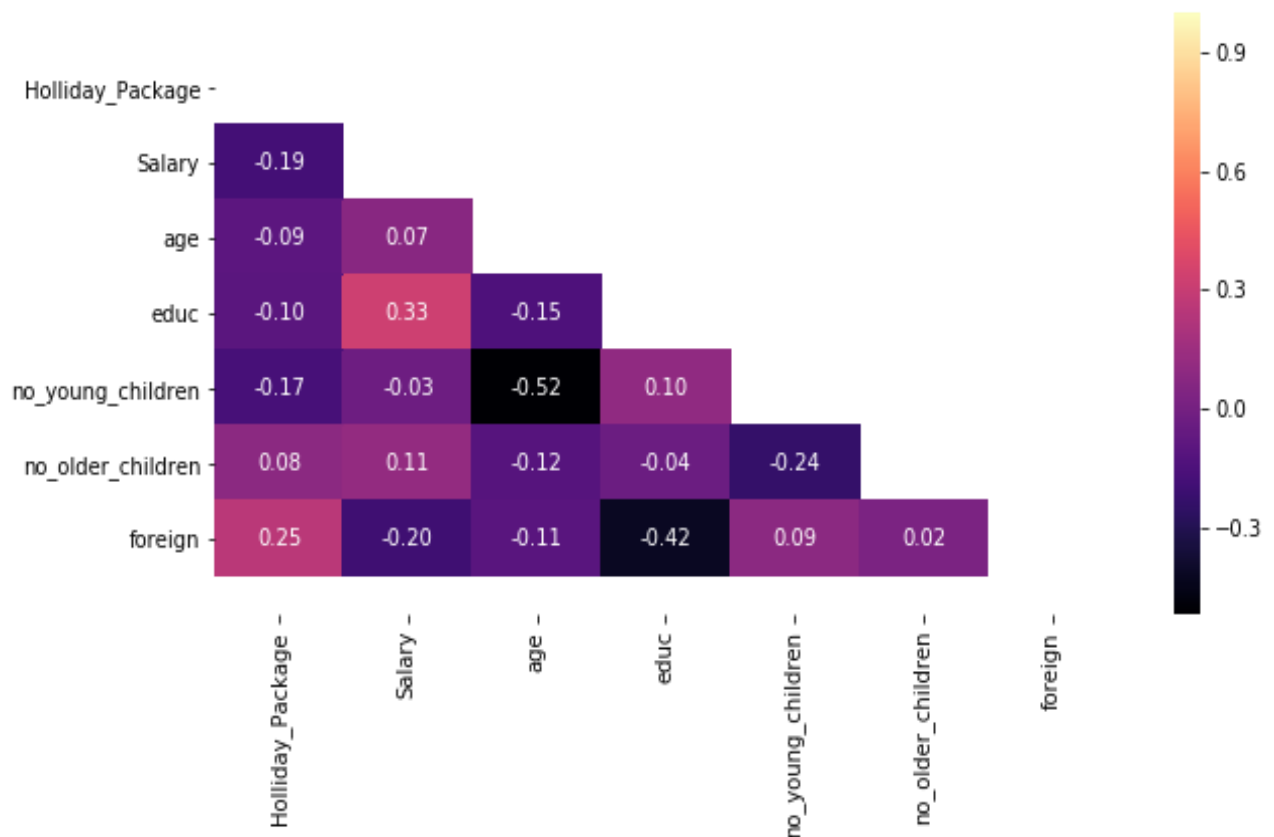
## Inference from histogram:

- ❖ Approximately 75% of the employees have salary ranging between 35,000 and 55,000.
- ❖ The age of majority of employees ranges between 20-60 years.
- ❖ The average years of education of majority of employees range between 3-15 years.
- ❖ Majority of the employees have no children younger than 7 years of age.
- ❖ Majority of the employees have on an average 0-2 children older than 7 years of age.
- ❖ The Holliday Package variable is our dependent variable with 471 employees not opting for holiday package and only 401 employees who have opted for the holiday package.

## Multivariate Analysis

### Heat Map

The Heat Map shows the relationship between different variables in our dataset. This graph can help us to check for any correlations between different variables.



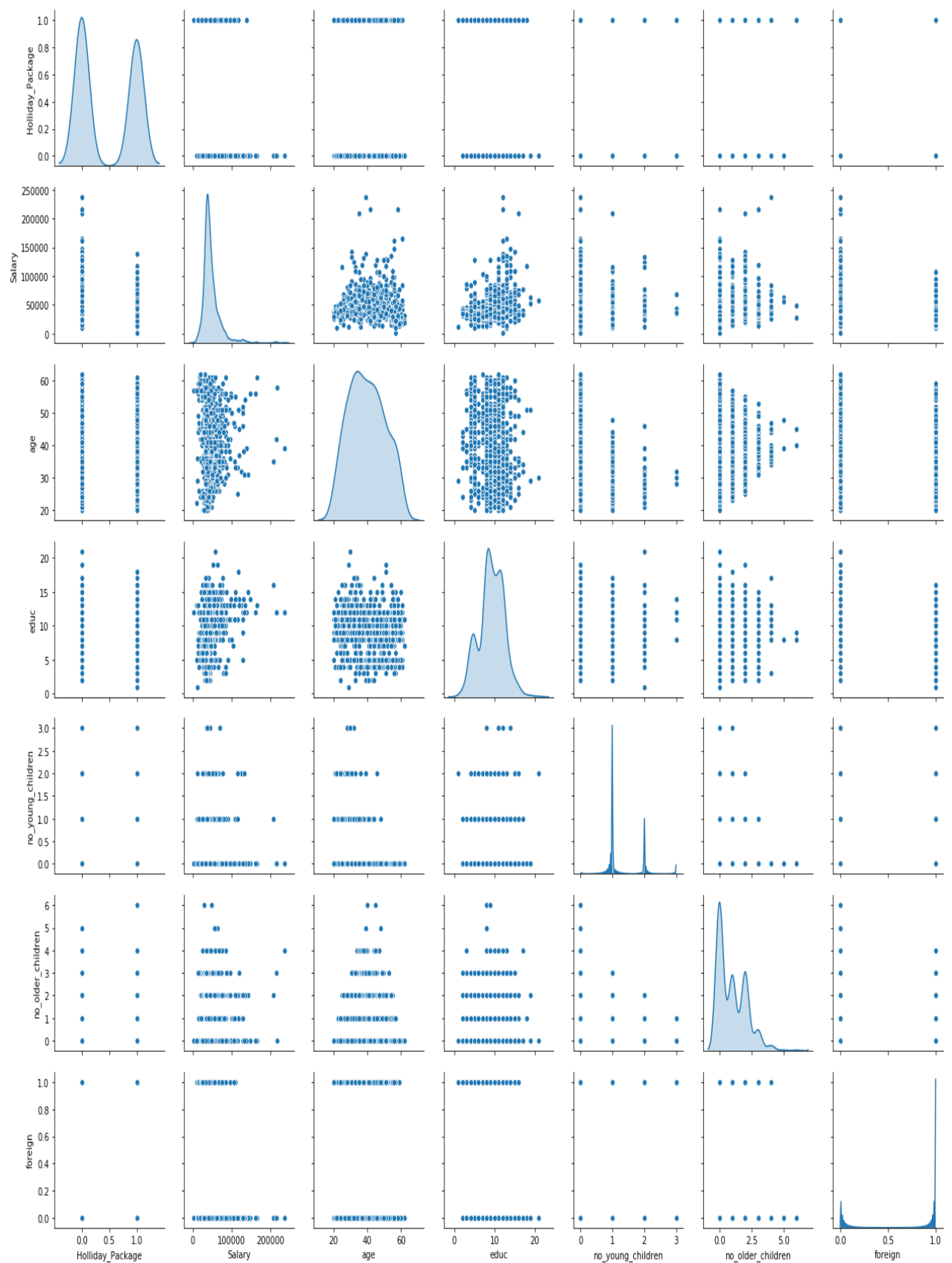
### Inference from heatmap:

We can see that there is **no** highly positive and negative correlation between the variables.

- ❖ The variables like Education and Salary are positive correlated.
- ❖ The variables like No of young children and Age are negatively correlated.
- ❖ The variables like Education and foreign are negatively correlated.
- ❖ Overall from the dataset there is no perfect strong correlation between the variables



## Pairplot



**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).**

```
1 # Seperating INDEPENDENT AND DEPENDENT VARIABLE
2
3 X=data_df.drop('Holliday_Package',axis=1)
4 y=data_df.pop('Holliday_Package')
5
6 y.value_counts()

0.0    471
1.0    401
Name: Holliday_Package, dtype: int64
```

### Splitting Dataset in Train and Test Data (70:30)

For building the models we will now have to split the dataset into Training and Testing data with the ratio of 70:30.

### Logistic Regression

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.

It is a technique to analyse a data-set which has a dependent variable and one or more independent variables to predict the outcome in a binary variable, meaning it will have only two outcomes.

The dependent variable is categorical in nature. Dependent variable is also referred as target variable and the independent variables are called the predictors.

Logistic regression is a special case of linear regression where we only predict the outcome in a categorical variable. It predicts the probability of the event using the log function.

We use the Sigmoid function/curve to predict the categorical value. The threshold value decides the outcome(win/lose).

Logistic Regression equation:

$$p = 1 / 1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 \dots + \beta_n X_n)}$$

Using the Train Dataset(X\_train) we will be creating a Logistic regression model and then further testing the model on Test Dataset(X\_test)

For creating the Logistic Regression, the package “Logistic Regression” is imported from sklearn library.

Using the GridSearchCV package from sklearn.model\_selection we will identify the best parameters to build a logistic regression namely, model. Hence, doing a few iterations with the values we got the best parameters to build the grid search Model which are as follows

```
1 X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.30,random_state=0)

1 grid={'penalty':['l2','none'],
2       'solver':['saga','lbfgs'],
3       'tol':[0.0001,0.00001]}
4
5 model = LogisticRegression(max_iter=10000,n_jobs=2)
6
7 grid_search = GridSearchCV(estimator = model, param_grid = grid, cv =5 ,n_jobs=-1,scoring='f1')

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

```
GridSearchCV(cv=5, estimator=LogisticRegression(max_iter=10000, n_jobs=2),
             n_jobs=-1,
             param_grid={'penalty': ['l2', 'none'], 'solver': ['saga', 'lbfgs'],
                          'tol': [0.0001, 1e-05]},
             scoring='f1')
```

BEST PARAMETER:

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

{'penalty': 'l2', 'solver': 'lbfgs', 'tol': 0.0001}

LogisticRegression(max_iter=10000, n_jobs=2)
```

## LINEAR DISCRIMINANT ANALYSIS(LDA)

Linear Discriminant Analysis or Normal Discriminant Analysis or Discriminant Function Analysis is a dimensionality reduction technique which is commonly used for the supervised classification problems. It is used for modelling differences in groups i.e. separating two or more classes. It is used to project the features in higher dimension space into a lower dimension space.

### LDA MODEL

Using Sklearn's Discriminant Analysis package import LinearDiscriminantAnalysis. We will fit our model into the train dataset.

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

```

1 # Training Data Class Prediction with a cut-off value of 0.5
2 pred_class_train = model.predict(X_train)
3
4 # Test Data Class Prediction with a cut-off value of 0.5
5 pred_class_test = model.predict(X_test)

```

```

1 # Training Data Probability Prediction
2 pred_prob_train = model.predict_proba(X_train)
3
4 # Test Data Probability Prediction
5 pred_prob_test = model.predict_proba(X_test)
6

```

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

### **Logistic Regression Classification Report for training and testing dataset:**

```

1 print(classification_report(y_train,ytrain_predict))
2 print('\n')
3 print(classification_report(y_test,ytest_predict))

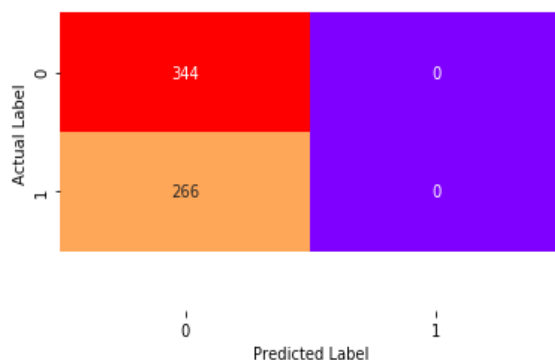
```

	precision	recall	f1-score	support
0.0	0.56	1.00	0.72	344
1.0	0.00	0.00	0.00	266
accuracy			0.56	610
macro avg	0.28	0.50	0.36	610
weighted avg	0.32	0.56	0.41	610

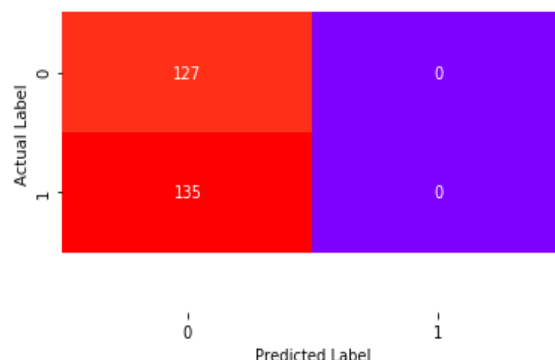
	precision	recall	f1-score	support
0.0	0.48	1.00	0.65	127
1.0	0.00	0.00	0.00	135
accuracy			0.48	262
macro avg	0.24	0.50	0.33	262
weighted avg	0.23	0.48	0.32	262

### **Confusion Matrix**

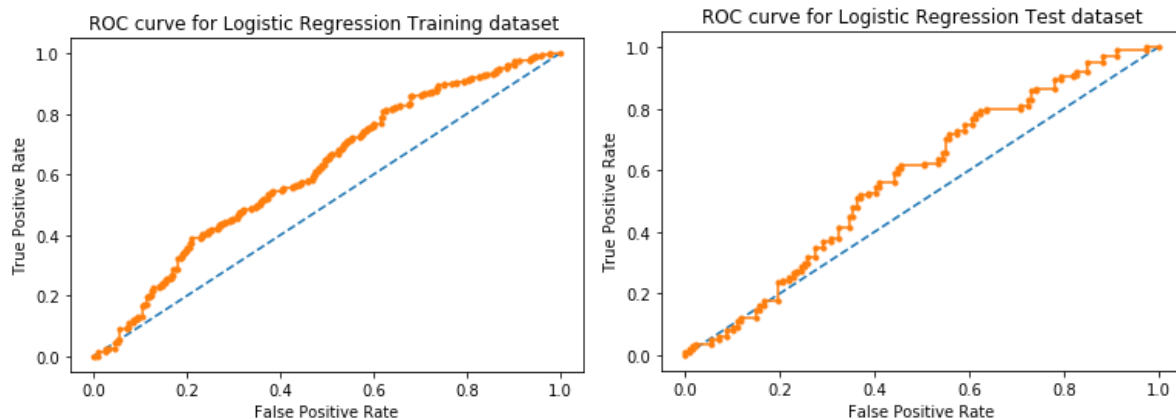
Confusion Matrix for Y\_Train Dataset



Confusion Matrix for Y\_Test Dataset



## ROC AUC Score and ROC Curve



## Model Score

- The Logistic Regression Model Training dataset AUC score is 0.615
- The Logistic Regression Model Testing dataset AUC score is 0.577

## Linear Discriminant Analysis:

The classification report, confusion matrix, AUC-ROC score for LDA model is described below

## Classification Report

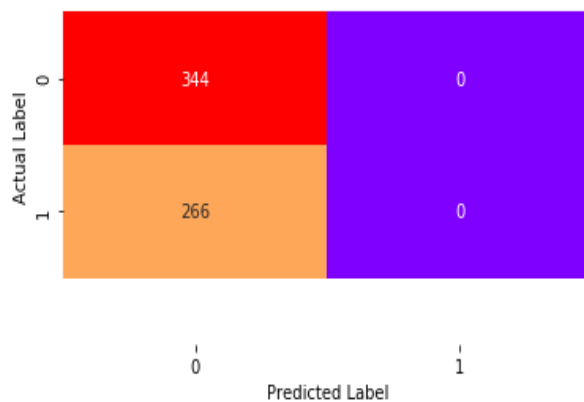
```
1 print(classification_report(y_train,ytrain_predict))
2 print('\n')
3 print(classification_report(y_test,ytest_predict))
```

	precision	recall	f1-score	support
0.0	0.56	1.00	0.72	344
1.0	0.00	0.00	0.00	266
accuracy			0.56	610
macro avg	0.28	0.50	0.36	610
weighted avg	0.32	0.56	0.41	610

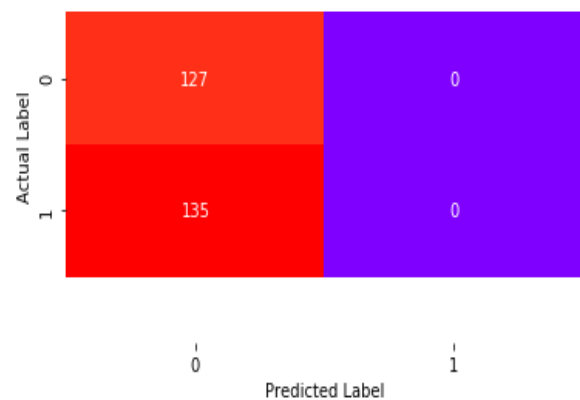
	precision	recall	f1-score	support
0.0	0.48	1.00	0.65	127
1.0	0.00	0.00	0.00	135
accuracy			0.48	262
macro avg	0.24	0.50	0.33	262
weighted avg	0.23	0.48	0.32	262

## Confusion Matrix

Confusion Matrix for Y\_Train Dataset

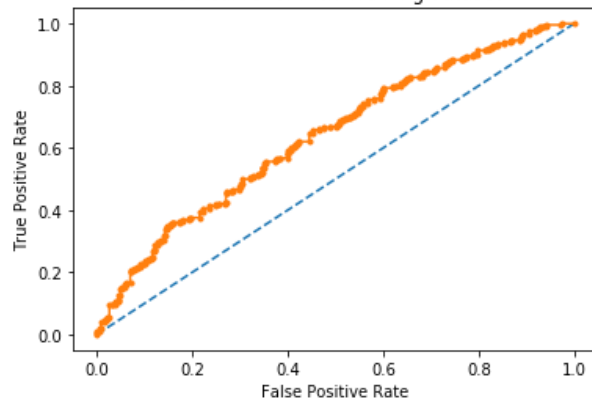


Confusion Matrix for Y\_Test Dataset

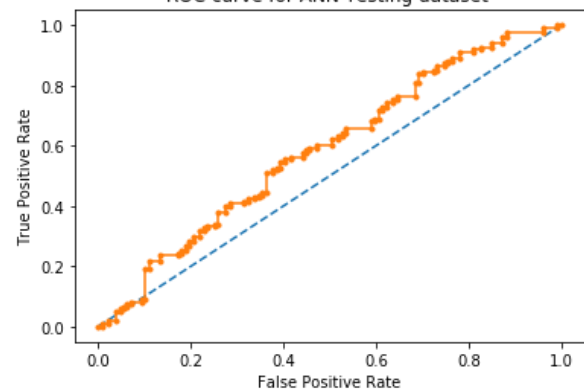


## AUC ROC Score and ROC Curve

ROC curve for ANN Training dataset



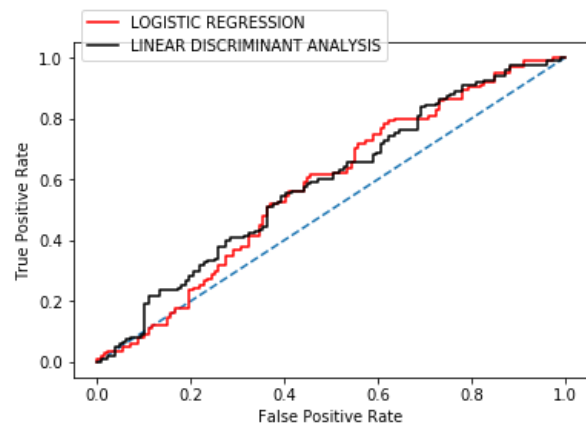
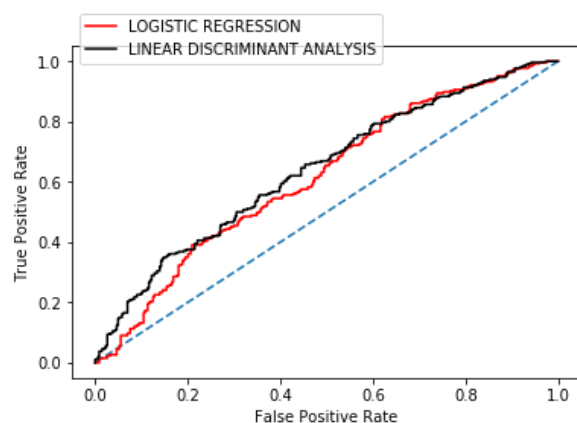
ROC curve for ANN Testing dataset



## Model Score

- The LDA model Training dataset AUC score is 0.640
- The LDA model Testing dataset AUC score is 0.587

## Comparision of AUC ROC Score and ROC Curve



### Choosing Correct cut-off point for getting Better F1 score and Accuracy:

```
1 #Predicting the classes on the test data
2
3 data_pred_custom_cutoff=[]
4 for i in range(0,len(pred_prob_test[:,1])):
5     if np.array(pred_prob_test[:,1])[i]>0.4:
6         a=1
7     else:
8         a=0
9     data_pred_custom_cutoff.append(a)

1 confusion_matrix(y_test,ytest_predict)
2 ax=sns.heatmap(confusion_matrix(y_test,data_pred_custom_cutoff)
3 bottom, top = ax.get_ylim()
4 ax.set_ylim(bottom + 0.5, top - 0.5)
5 plt.xlabel('Predicted Label')
6 plt.ylabel('Actual Label')
7 plt.title('Confusion Matrix for Y_Test Dataset')
8 plt.show()
```

### Classification Report

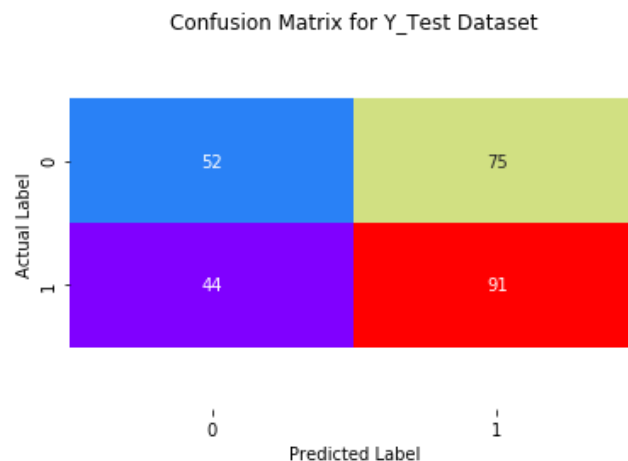
Classification Report of the default cut-off test data:

	precision	recall	f1-score	support
0.0	0.52	0.72	0.61	127
1.0	0.59	0.38	0.46	135
accuracy			0.55	262
macro avg	0.56	0.55	0.53	262
weighted avg	0.56	0.55	0.53	262

Classification Report of the custom cut-off test data:

	precision	recall	f1-score	support
0.0	0.54	0.41	0.47	127
1.0	0.55	0.67	0.60	135
accuracy			0.55	262
macro avg	0.54	0.54	0.54	262
weighted avg	0.55	0.55	0.54	262

## Confusion Matrix



Since we are building a model to predict whether the employee is opted for tour package or not. For our purposes, we will be more interested in correctly classifying 1 (employee opted for tour package) than 0 (employee is not opted for tour package). From the above model, we looking at the Accuracy, Recall, F1score and AUC score for the training and testing data and we are looking especially in predicting **Class 1**. Comparing of all the performance evaluators for the three models are given in the following table. We are using Accuracy, Precision, Recall, F1 Score and AUC Score for our evaluation.

<u>Model</u>	<u>Accuracy</u>	<u>Precision</u>	<u>Recall</u>	<u>F1 Score</u>	<u>AUC Score</u>
<b><u>Logistic Regression</u></b>					
<i>Train Data</i>	0.56	0	0	0	<b>0.615</b>
<i>Test Data</i>	0.48	0	0	0	<b>0.577</b>
<b><u>LDA at 0.5 probability</u></b>					
<i>Train Data</i>	0.55	0	0	0	<b>0.640</b>
<i>Test Data</i>	0.55	0.59	0.38	0.46	<b>0.587</b>
<b><u>LDA with custom probability</u></b>					
<i>Test Data</i>	<b>0.55</b>	<b>0.55</b>	<b>0.67</b>	<b>0.60</b>	<b>0.641</b>

From the above table, comparing the model performance evaluators for the models it is quite clear that the **Linear Discriminate Analysis Model** is performing well as compared to the other as it has **high Recall, Precision, F1 score**.



## 2.5 Inference: Based on the whole Analysis, what are the business insights and recommendations





The case given to us is about a tour and travels company trying to sell holiday packages to some companies' employees so as to increase their profits. The information given to us i.e. their Salary, age, education, no. of children etc. help us to determine whether or not these factors help in determining the behaviour of these employees in opting the holiday package. So as to increase their business and profits we have built two models namely, Logistic Regression and Linear Discriminant Analysis (LDA).

As per the case study given to us, the major conclusions are as follows:

- Logistic Regression Model and LDA are able to predict the employees behaviour with only 53% and 64% accuracy respectively. Although both of these are not good models for our predictions but in this scenario we can take LDA model into consideration.
- As per the Performance Metrics computed above for all the models it can be evidently seen that the best model for our prediction is LDA model over Logistic Regression Model as all the performance metrics are comparatively high than Logistic Regression Model.
- The employees opting for the holiday package are correctly identified by approximately 56% of the times by LDA model whereas Logistic Regression Model is predicting these types of employees only 9% of the times which is definitely a huge difference in our results.

So as to increase the profits for the company and to maximise sales of the holiday package it is concluded that LDA is favoured as it maximises the chances for discriminating between the two classes i.e. between the ones opting for the package from the ones not opting for it.

As a tour and travel agency they should focus on the following employee because they have more chance to opted for a tour package, they follow:

-  The employee aged between 25 to 48 years old.
-  The employees are from same country
-  The employee having a smaller number of older children
-  The employees have 4 to 18 years of education.