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.

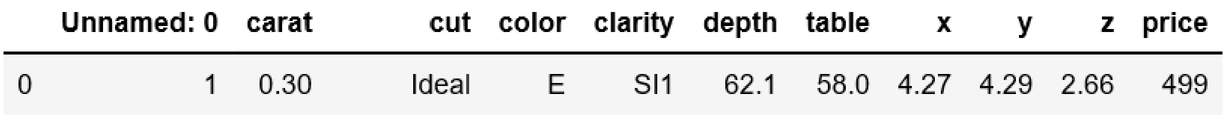
1.1. Read the data and do exploratory data analysis. Describe the data briefly. (Check the null values, Data types, shape, EDA). Perform Univariate and Bivariate Analysis.

The required libraries are loaded into Jupyter notebook, and the file is read into it subsequently.

a) **DATA TYPES**: Using the info () and dataframe.shape features, the datatypes are checked – it is found that most of the columns are numerical with either float or integer types. Cut, Color and Clarity columns are object data types or and appear to be strings/characters.

b) **SHAPE** :The number of rows was found to be 26967 with 11 columns. After going through the data, some columns are dropped.

c) **DROPPING UNWANTED COLUMNS**:



	Unnamed: 0	carat	cut	color	clarity	depth	table	x	y	z	price
0	1	0.30	Ideal	E	SI1	62.1	58.0	4.27	4.29	2.66	499

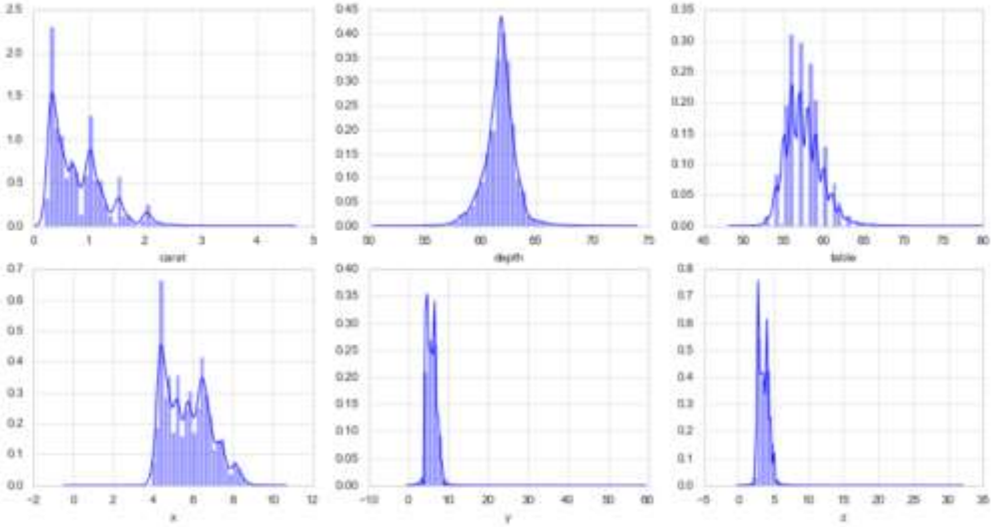
From a general glance at the first few data rows, it appears that the column “Unnamed: 0” may not be useful and is dropped.

d) **CHECK FOR DUPLICATES** : Duplicates are checked for and none are to be found

e) **NULL VALUE**: Null values are checked for using isnull(). It is found that the depth column has 627 NaN/null values. These values are later imputed.

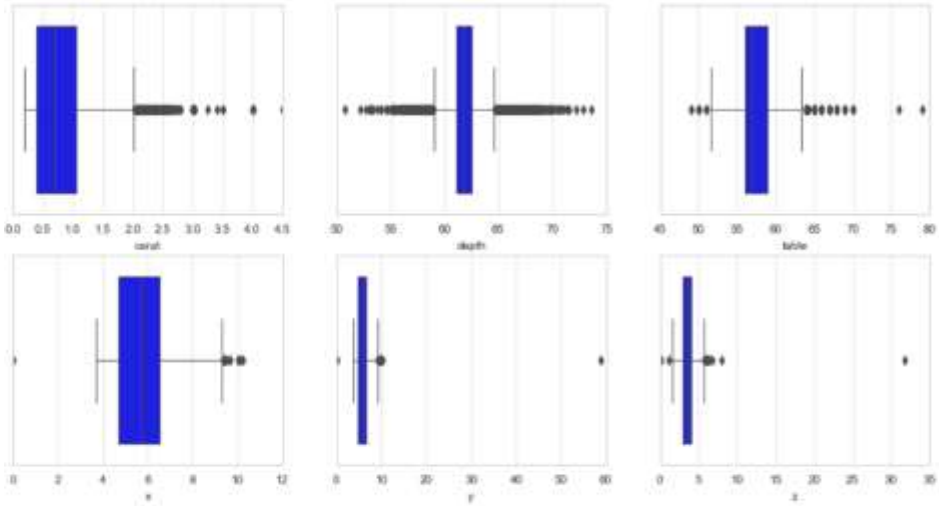
f) **Univariate and Bivariate Analyses**:

CUT, COLOR and CLARITY are categorical values. Hence, they can be ignored for the time being while plotting distribution plots.

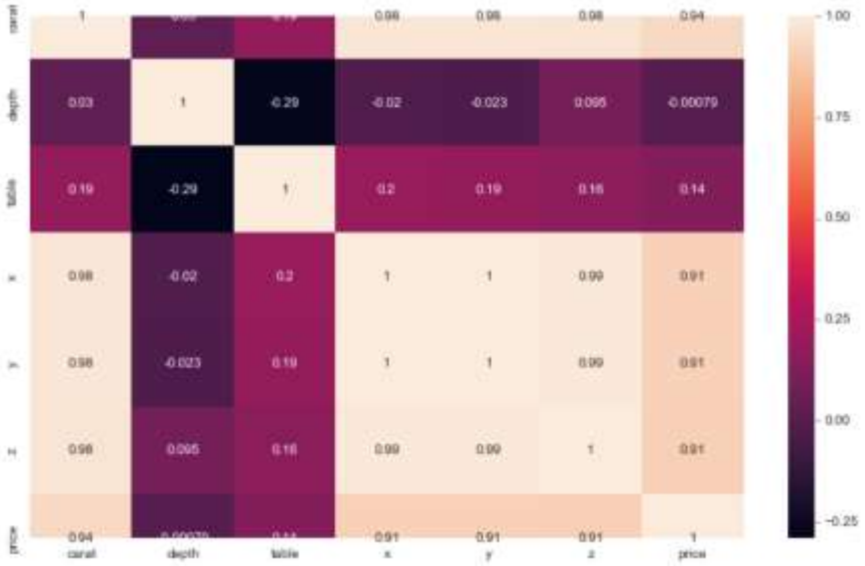


For the rest of the variables, it is found that there is large amount of skewness and it's possible that it may be caused due to outliers.

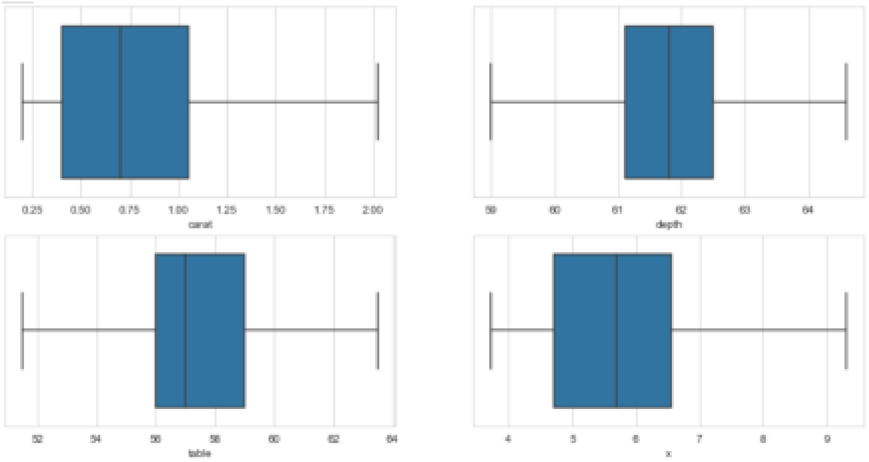
Outliers are checked for.



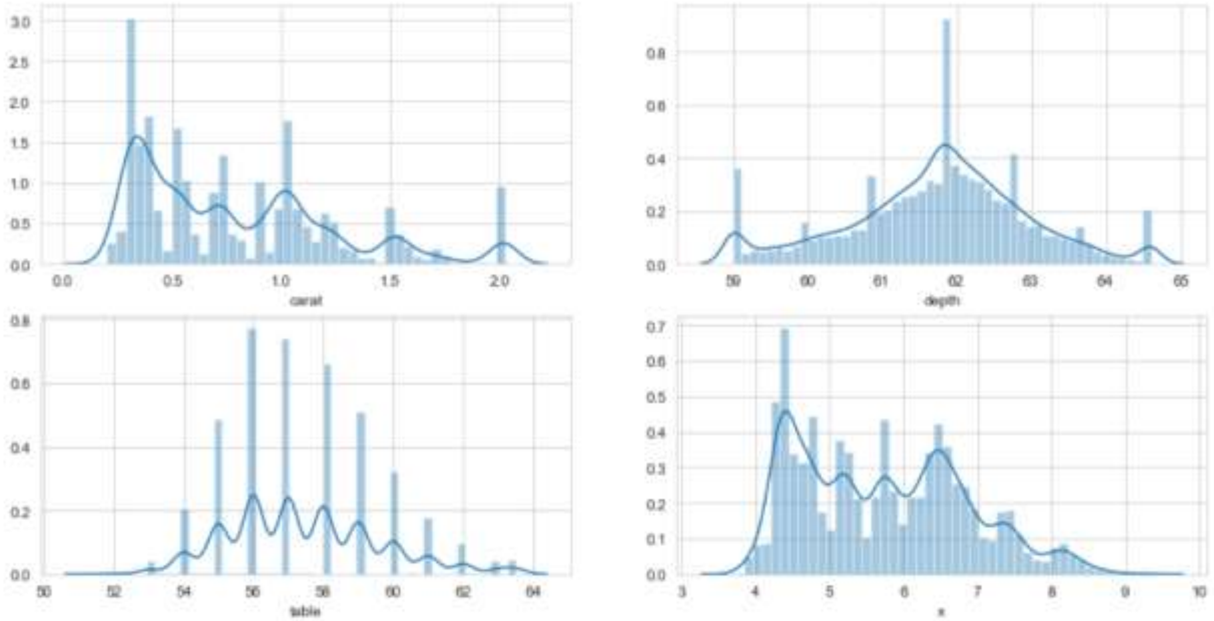
A heatmap is drawn to understand correlation between values. It is found that **<u>X ,Y, Z show multicollinearity with correlation value of 0.99 (X,Z) and 1 (X,Y)</u>**. Hence we can **drop one of the variables in each case (Y and Z)**, and we are left only with X out of the 3 dimensions.



Outliers are checked for and values which are higher than the upper and lower limits are brought to the limits through imputation. The new boxplot looks like the figure below.



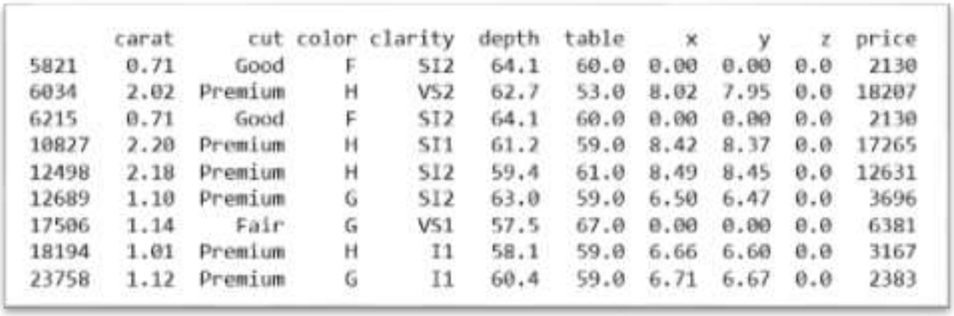
For the same data, a distplot showing the distribution and histogram is plotted to understand the distribution of the data with outliers removed



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?

As mentioned earlier in the document, the isnull() or isna() function is used to get the null or NaN values. (). It is found that the depth column has 627 NaN/null values .These are replaced or simply imputed with the median value. These NaN values do not have any specific meaning and need to be fixed (imputed with median in this case).

The zero value items are then checked for. It is found that few rows have zero values in the X,Y,Z variables. Please note that this process was done initially, before dropping Y and Z for being multicollinear with X, and hence show up on the figure below.



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

These are unrealistic cases as geometric dimensions of the stone cannot be zero. Therefore, there are two options to proceed

a) Remove the lines with zero geometric values
b) Impute them with a suitable number (median or mean value of the column).

Since these gem-stones are of high value and we do not want this imputation to influence the solution, we can go ahead and drop the values.We take this decision also because the number of data rows/points dropped are less than 1% of the overall data.

In our case, zero values may be outliers and it can also be taken care of automatically during imputation to remove outliers, if one does not want to delete these rows from the analysis. The best decision can be taken based on understanding of industry knowledge.

SCALING:

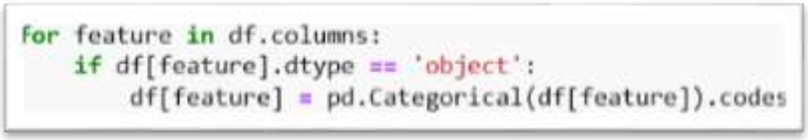
Linear regression models generally do not need scaling inspite of the variables being in different units. The coefficients are estimated such that they convert the units of each explanatory variable into the units of the response variable appropriately.

In regression analysis, you do not need to center or standardize your data for multiple regression. Standardization of the independent variables may be carried out when the model contains polynomial terms to model curvature or interaction terms.

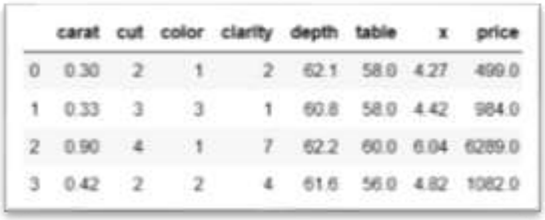
In our case therefore, we do not scale or standardize.

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 Rsquare, RMSE.

The data having string values is encoded using the ‘Categorical’ feature in Pandas.

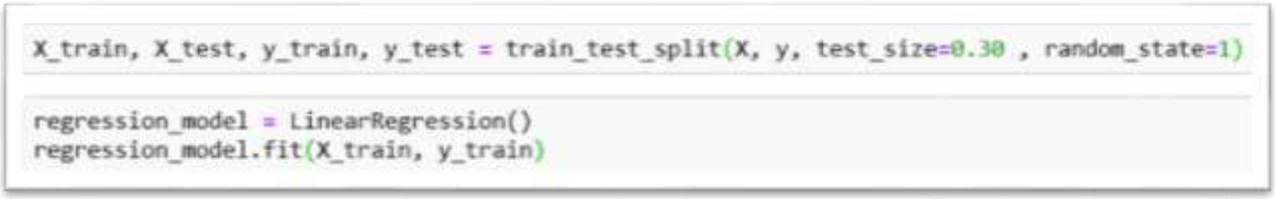


```
for feature in df.columns:
    if df[feature].dtype == 'object':
        df[feature] = pd.Categorical(df[feature]).codes
```



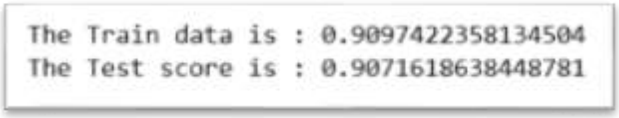
	carat	cut	color	clarity	depth	table	x	price
0	0.30	2	1	2	62.1	58.0	4.27	499.0
1	0.33	3	3	1	60.8	58.0	4.42	984.0
2	0.90	4	1	7	62.2	60.0	6.04	6289.0
3	0.42	2	2	4	61.6	56.0	4.82	1082.0

The train-test split was performed in the 70:30 ratio and the linear regression model was applied to fit the training data. The test data is then taken and applied to the fitted model to see what the response/prediction is for the test data.



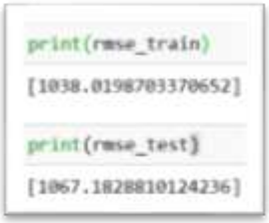
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30 , random_state=1)

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

Coefficients and the intercept can then be derived and printed. The train and test scores are checked for to see if the fit is agreeable.



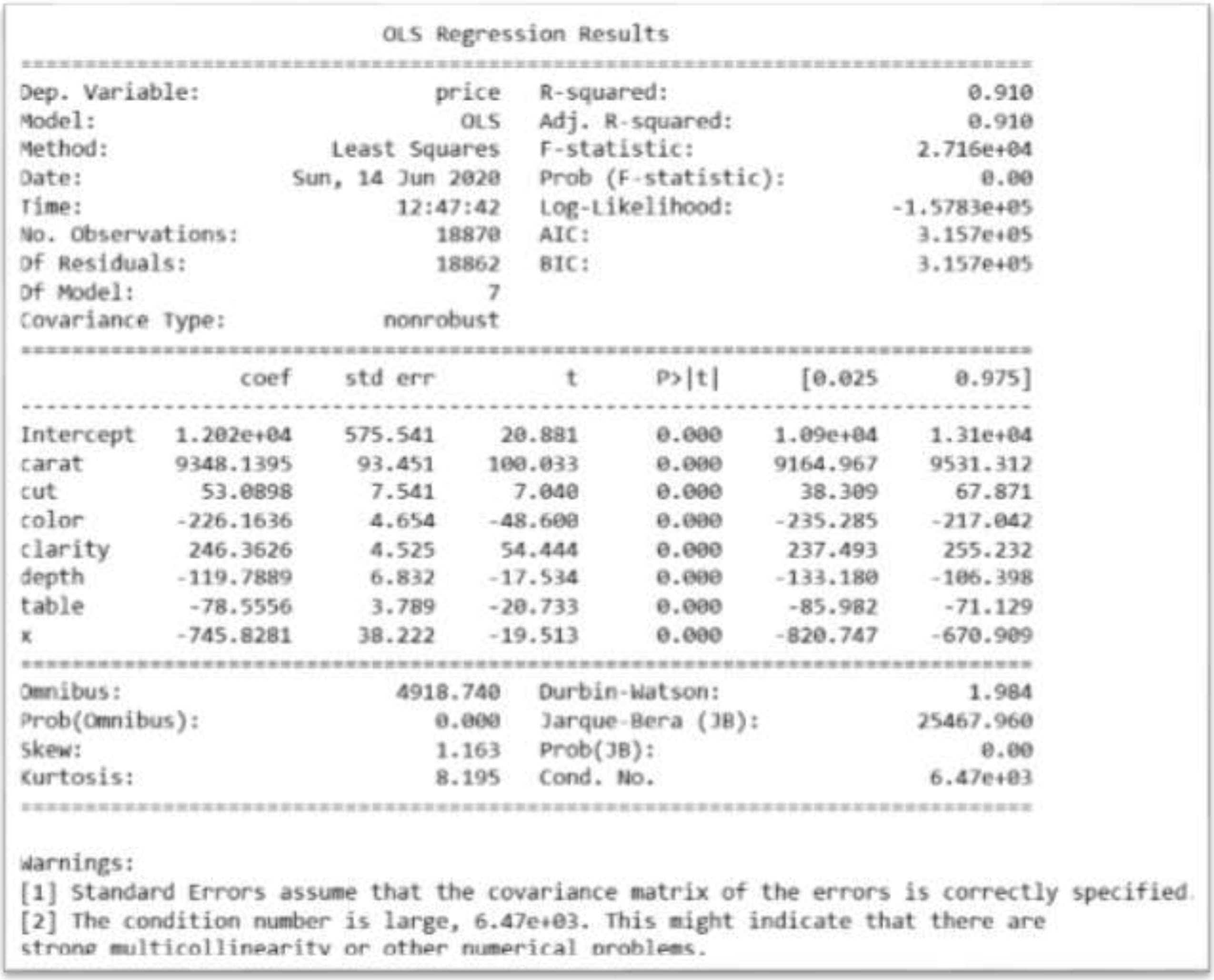
```
The Train data is : 0.9097422358134504
The Test score is : 0.9071618638448781
```

The Root mean square error for train and test data are calculated and printed.



```
print(rmse_train)

[1038.0198703370652]

print(rmse_test)

[1067.1828810124236]
```

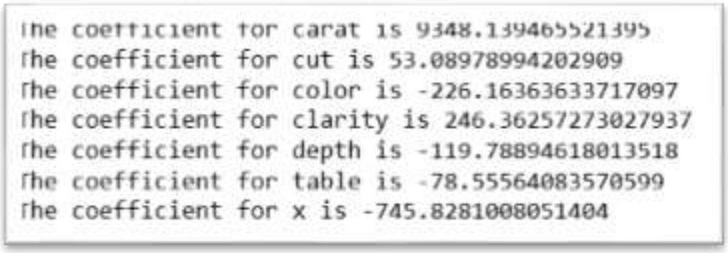
The same can be done using a Stats model library instead of an sklearn approach, and a summary can be printed out :



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.910
Model:                            OLS   Adj. R-squared:                  0.910
Method:                 Least Squares   F-statistic:                 2.716e+04
Date:                Sun, 14 Jun 2020   Prob (F-statistic):               0.00
Time:                        12:47:42   Log-Likelihood:            -1.5783e+05
No. Observations:               18870   AIC:                         3.157e+05
Df Residuals:                   18862   BIC:                         3.157e+05
Df Model:                           7
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   1.202e+04    575.541     20.881      0.000    1.09e+04    1.31e+04
carat       9348.1395     93.451    100.033      0.000    9164.967    9531.312
cut           53.0898      7.541      7.040      0.000      38.309      67.871
color       -226.1636      4.654    -48.600      0.000    -235.285    -217.042
clarity      246.3626      4.525     54.444      0.000     237.493     255.232
depth       -119.7889      6.832    -17.534      0.000    -133.180    -106.398
table        -78.5556      3.789    -20.733      0.000     -85.982     -71.129
x           -745.8281     38.222    -19.513      0.000    -820.747    -670.909
==============================================================================
Omnibus:                     4918.740   Durbin-Watson:                   1.984
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            25467.960
Skew:                           1.163   Prob(JB):                         0.00
Kurtosis:                       8.195   Cond. No.                     6.47e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.47e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

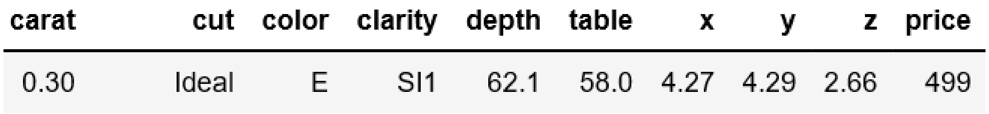
1.4 Inference: Basis on these predictions, what are the business insights and recommendations.

From the coefficients that are obtained as part of the study, one can understand the effect of changing independent variables (such as carar,cut,x e.t.c) on the dependent variable (price).



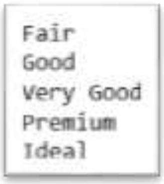
```
The coefficient for carat is 9348.139465521395
The coefficient for cut is 53.08978994202909
The coefficient for color is -226.16363633717097
The coefficient for clarity is 246.36257273027937
The coefficient for depth is -119.78894618013518
The coefficient for table is -78.55564083570599
The coefficient for x is -745.8281008051404
```

The variable which affects the price the most seems to be carat. Higher the number of carats, more the price.



carat	cut	color	clarity	depth	table	x	y	z	price
0.30	Ideal	E	SI1	62.1	58.0	4.27	4.29	2.66	499

For the variables which are categorical such as **cut, color and clarity**, multiplication of coefficients or values represent switching from one category to another. For example, in ‘cut’, by increasing the independent associated variable by 53 times, one would be able to switch to the next best ‘cut’ category.



```
Fair
Good
Very Good
Premium
Ideal
```

Therefore, better the cut, higher the price.

For **Color**, the negative value indicated that prices of different colors can be obtained by subtracting the coefficient value (without changing other terms).

A similar logic is applied for **clarity**.

For other continuous variables like **depth, table and x**, the logic is that the price varies as a factor of the coefficient. For depth, table and clarity, the coefficients are negative. Therefore, reducing the variables associated with these can increase the price. In general, higher the absolute value of the coefficient, the more it can change (or effects) the price.

Based on the absolute value of coefficients, the **five most important factors** affecting price of the gemstone is

a) Carat
b) X (and Y,Z since they are similar)
c) Clarity
d) Color
e) Depth

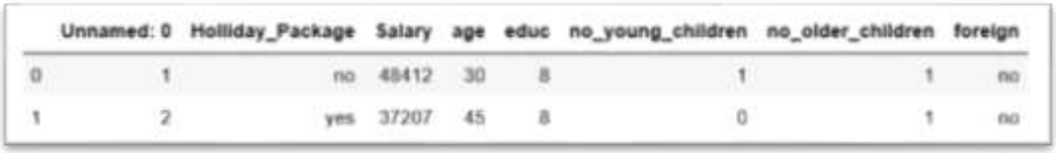
Therefore, on the basis of understand the importance of each variable through their coefficients, the business can make an educated decision on choice of gemstones that will yield the best profit.

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.

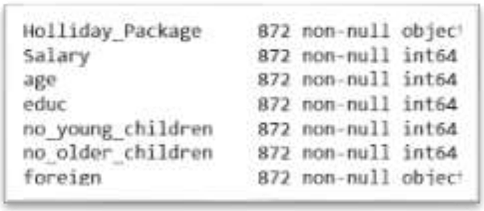
The data is read and is glanced through to get a quick idea of what it looks like.



	Unnamed: 0	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
0	1	no	48412	30	8	1	1	no
1	2	yes	37207	45	8	0	1	no

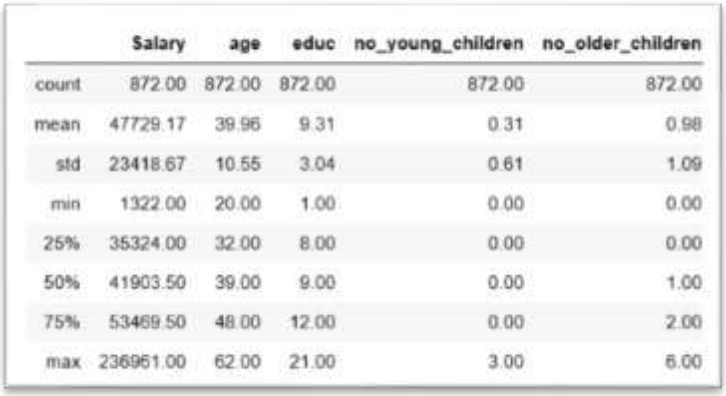
The column named “Unnamed: 0” is not useful and it is dropped.

The data types of the columns are then checked for :



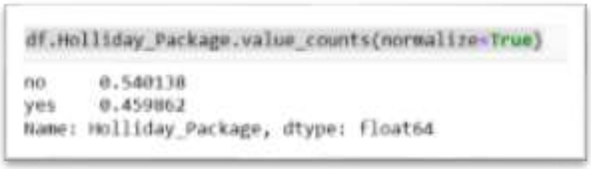
```
Holliday_Package     872 non-null object
Salary               872 non-null int64
age                  872 non-null int64
educ                 872 non-null int64
no_young_children    872 non-null int64
no_older_children    872 non-null int64
foreign              872 non-null object
```

Using the Pandas describe() function, the data is viewed from statistical viewpoint.



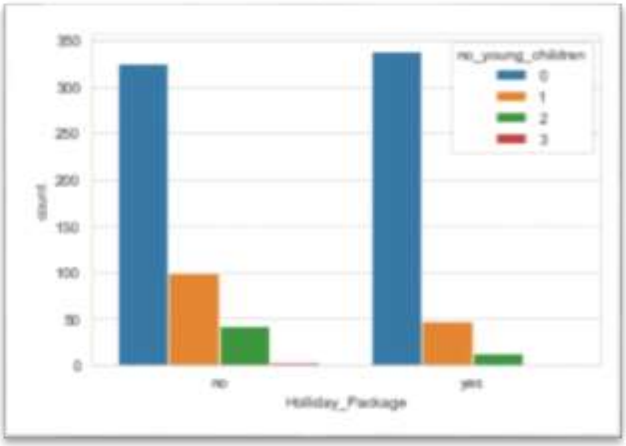
	Salary	age	educ	no_young_children	no_older_children
count	872.00	872.00	872.00	872.00	872.00
mean	47729.17	39.96	9.31	0.31	0.98
std	23418.67	10.55	3.04	0.61	1.09
min	1322.00	20.00	1.00	0.00	0.00
25%	35324.00	32.00	8.00	0.00	0.00
50%	41903.50	39.00	9.00	0.00	1.00
75%	53469.50	48.00	12.00	0.00	2.00
max	236961.00	62.00	21.00	3.00	6.00

Further, checks are made on how many people amongst all in the data have taken holiday packages. It was found that 46% of the total have taken up holiday packages.

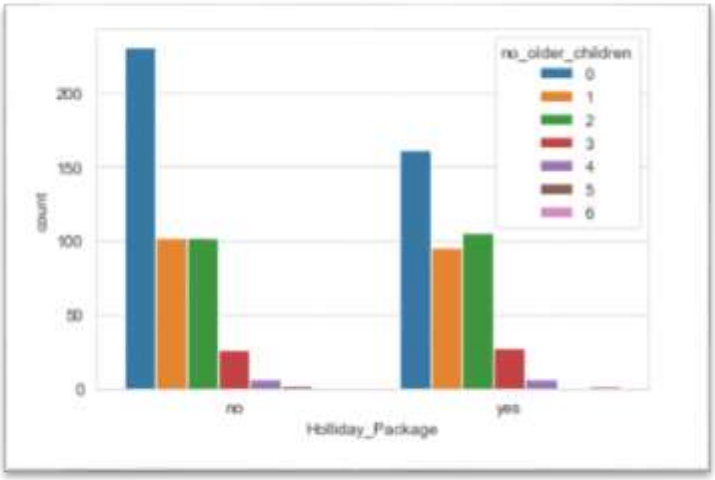


```
df.Holliday_Package.value_counts(normalize=True)

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

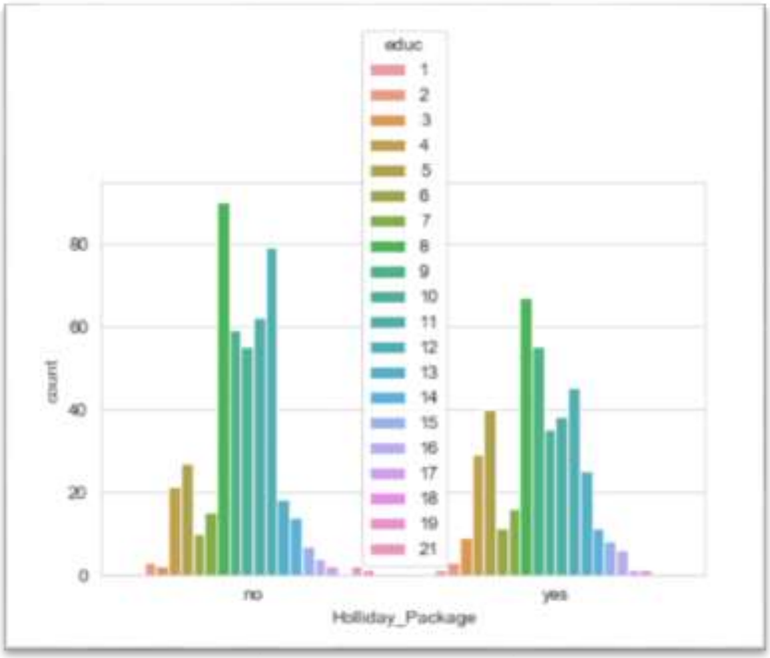
We then try to check if the number of young children a person has an effect on their decision to take a holiday by plotting a countplot such as the one shown below. It is observed that people with few or no children have a higher probability of taking up a holiday package.



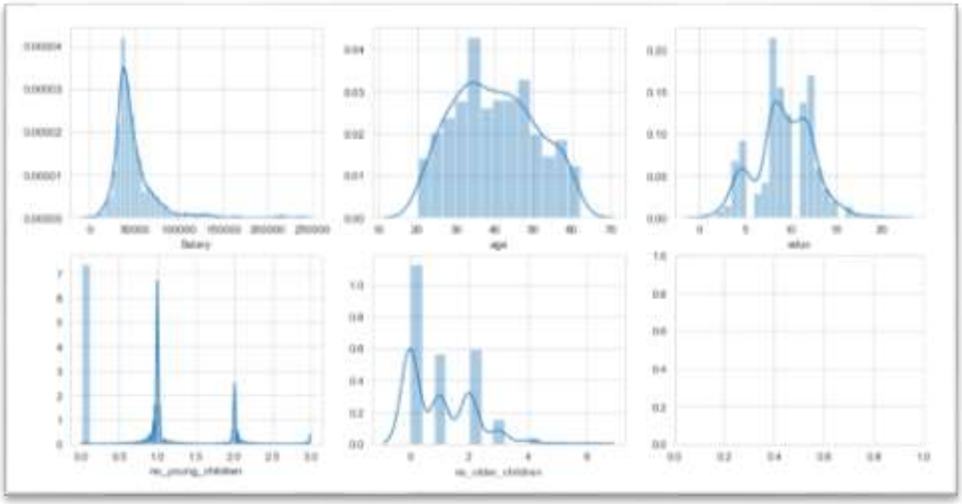
Similarly, we can check if the number of older children a person has an effect on their decision to take a holiday. Here, we observe that people with no older children tend to be the group that takes the most number of holidays.



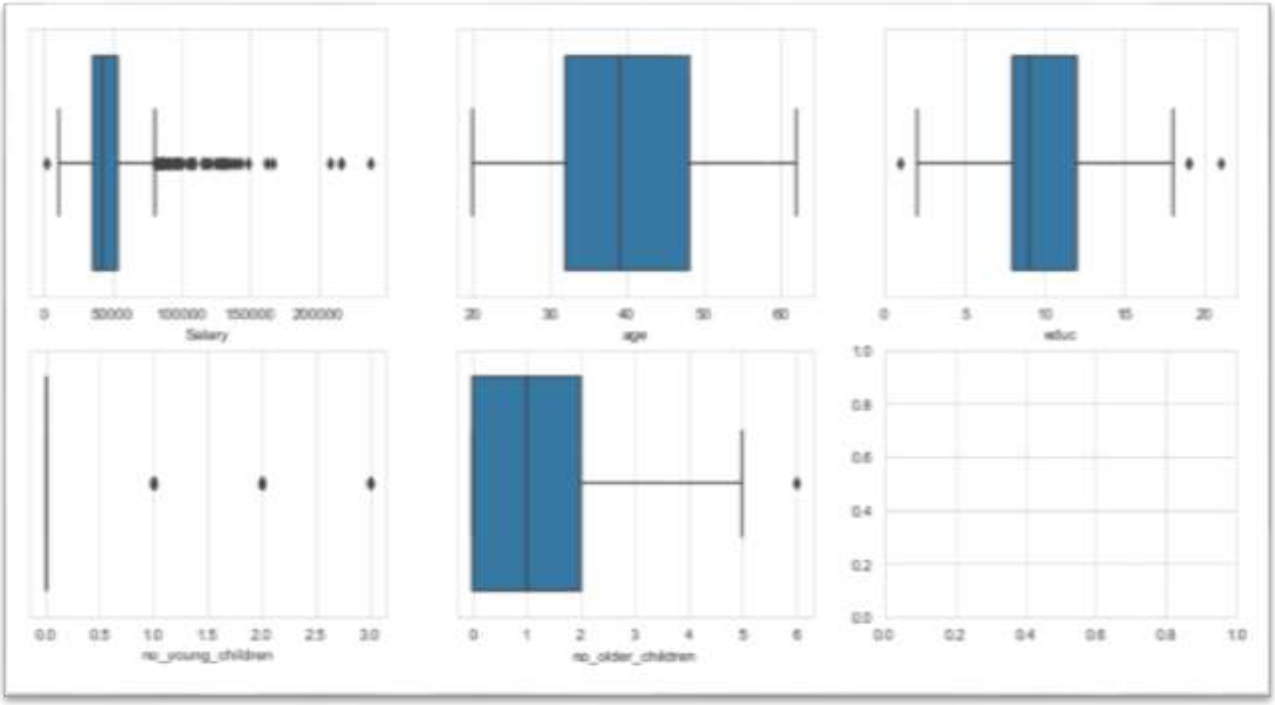
It is also observed that people within 8-12 years of education take the most number of holidays compared to the rest.



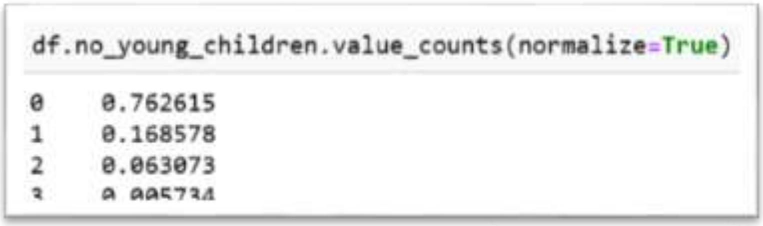
We then create distribution plots to view the distribution of data for each variable and how normal they are. Some skewness is observed in data related to the number of children and this is acceptable. It will be checked later whether grouping can be done for data with higher number of children.



Outlier checking is done and the continuous variables (salary, age, edu) are treated (imputed) by moving the outliers to the lower and upper limits (Q1-1.5 IQR, Q3 + 1.5 IQR) of the box plot shown. This kind of treatment is done because of lack of points with values in the outlier region. More data in this region would make the data more Gaussian/Normal.



It is observed that people with younger children get flagged as outliers. This is because the number of people with zero younger children as a proportion of the set studied is large (76%).



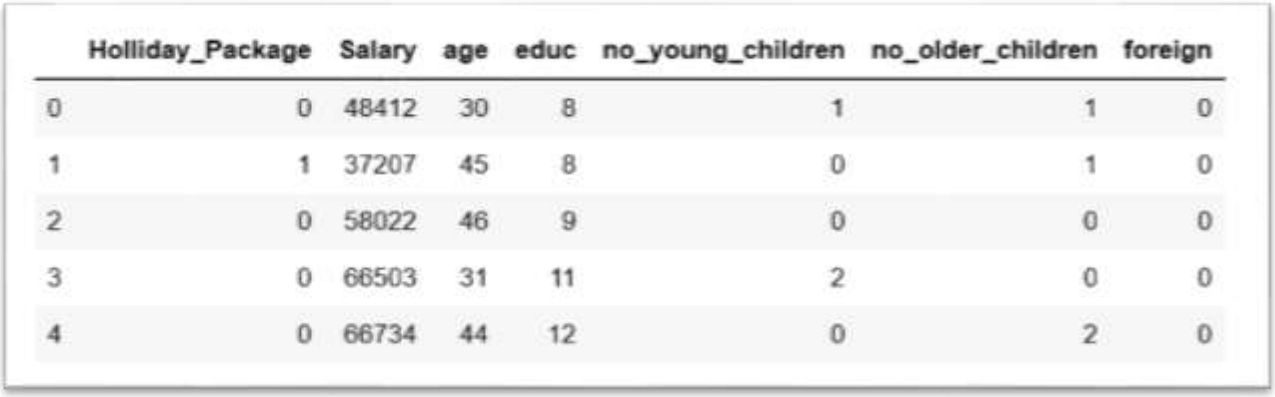
```
df.no_young_children.value_counts(normalize=True)
```

```
0    0.762615
1    0.168578
2    0.063073
3    0.005734
```

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).

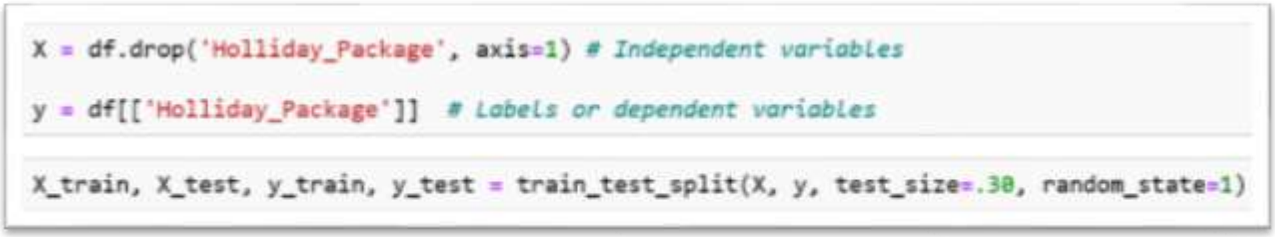
Data encoding is done to the columns “Holliday_Package” and “Foreign”.

Encoding is done with 0 for “no” and 1 for “yes”.



	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
0	0	48412	30	8	1	1	0
1	1	37207	45	8	0	1	0
2	0	58022	46	9	0	0	0
3	0	66503	31	11	2	0	0
4	0	66734	44	12	0	2	0

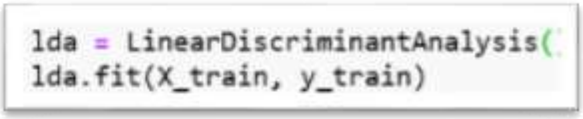
Data is then split into labels(y) and independent variables(X). The data for independent variables is further split into train and test data using a split of 70:30.



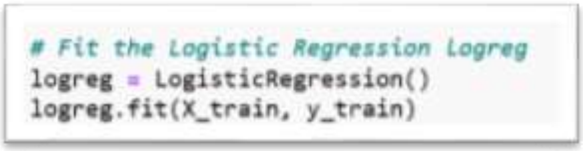
```
X = df.drop('Holliday_Package', axis=1) # Independent variables

y = df[['Holliday_Package']]  # Labels or dependent variables
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.30, random_state=1)
```

LDA and Logistic regression is performed on the data which has been split to train and test the model. The results are reviewed further.



```
lda = LinearDiscriminantAnalysis(
lda.fit(X_train, y_train)
```



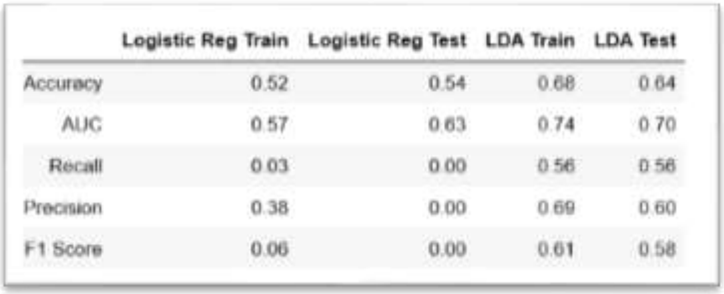
```
# Fit the Logistic Regression Logreg
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
```

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

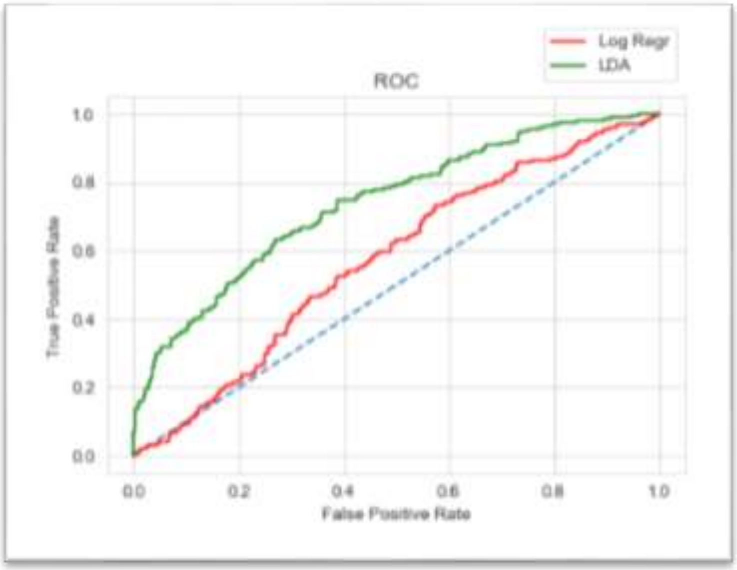
INITIAL MODEL :

We see outliers in "no_of_young_children" and "no_of_older_children". Here the number of people with zero children are higher and hence this set skews the data. We choose not to treat it as this represents a real scenario. More data that would help normalize the data could make the predictions better.

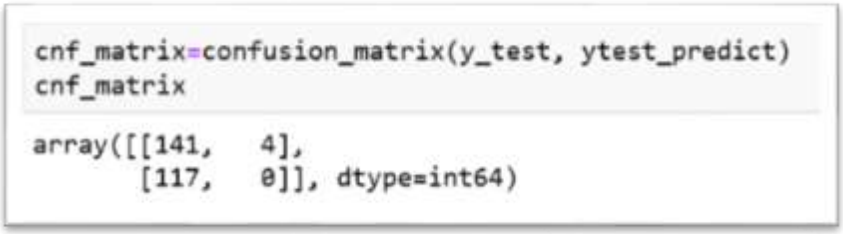
We get the following results :



	Logistic Reg Train	Logistic Reg Test	LDA Train	LDA Test
Accuracy	0.52	0.54	0.68	0.64
AUC	0.57	0.63	0.74	0.70
Recall	0.03	0.00	0.56	0.56
Precision	0.38	0.00	0.69	0.60
F1 Score	0.06	0.00	0.61	0.58

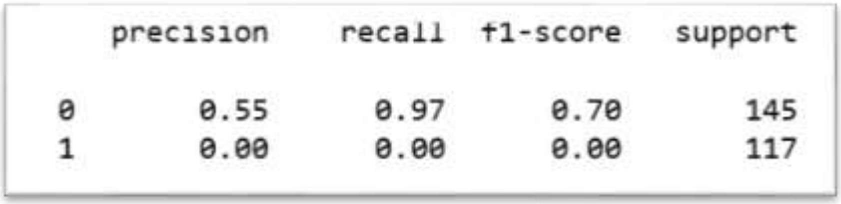


When comparing these two models, we can conclude based on Accuracy, AUC and other parameters that LDA is much better compared to logistic regression in this case.



```
cnf_matrix=confusion_matrix(y_test, ytest_predict)
cnf_matrix
```

```
array([[141,   4],
       [117,   0]], dtype=int64)
```



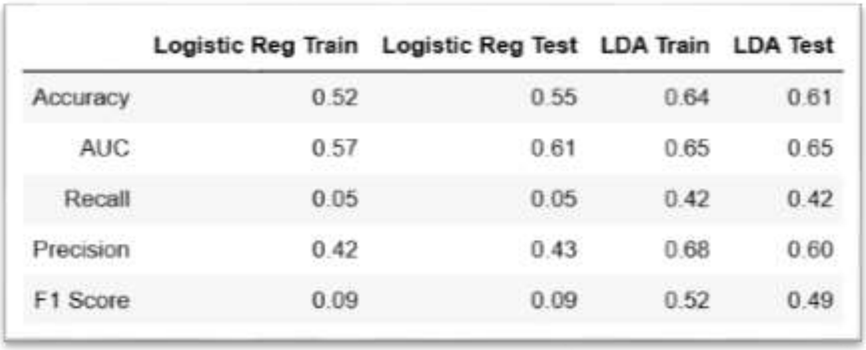
```
      precision    recall  f1-score   support

0          0.55      0.97      0.70       145
1          0.00      0.00      0.00       117
```

It is seen from the confusion matrix and classification report that the recall, precision and f1 scores are low in logistic regression especially for ‘1’s or positives.

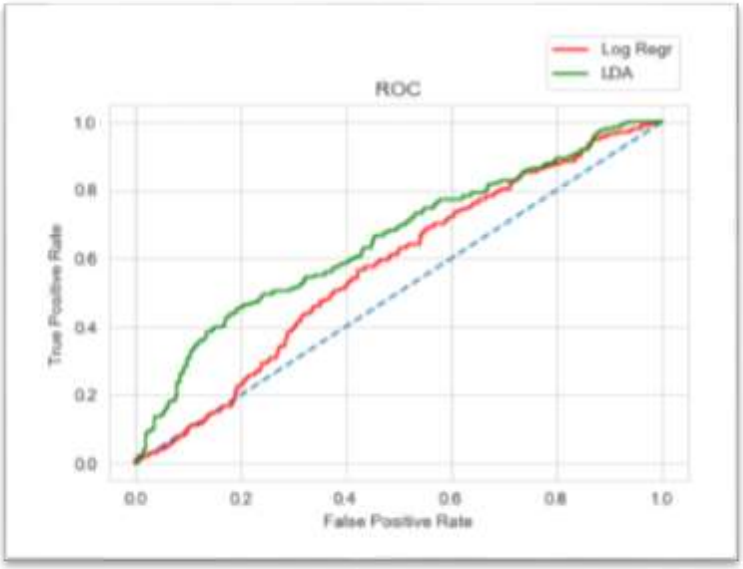
This can be improved by getting more data that correspond to this case, which will help improve the prediction of ‘1’s or positives.

Further, using feature selection techniques, scaling and grouping data to make the results can be further improved.

To test this, we remove the features ‘no_young_children’ and ‘no_older_children’. We observe that using selected features improves the efficiency of the logistic regression method.



	Logistic Reg Train	Logistic Reg Test	LDA Train	LDA Test
Accuracy	0.52	0.55	0.64	0.61
AUC	0.57	0.61	0.65	0.65
Recall	0.05	0.05	0.42	0.42
Precision	0.42	0.43	0.68	0.60
F1 Score	0.09	0.09	0.52	0.49



Therefore, we can try backward and forward feature elimination to improve the score. However, this is a lengthy process and beyond the scope of this exercise.

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

On the basis of these predictions, we find that LDA is more suitable for this dataset. Better accuracy can be obtained by feature reduction or having more relevant data which provides more positives (as observed from classification report).

With better tuning of the logistic regression case along with more data, we can get better results.

Some features, which are not important such as salary can be removed and the model can be run again to get results.

This would give an idea as to which variables are most important in affecting the decision of a person to purchase a ticket.