



Predictive Modeling Project

2 0 2 1

Date: 19 December 2021

Great Learning

Authored by: ANIMESH HALDER

Content

Problem 1: Linear Regression	5
Introduction	5
Data Dictionary	5
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.	5
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.	7
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 statsmodel. 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.	10
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.	15
Problem 2: Logistic Regression and Linear Discriminant Analysis	17
Introduction	17
Data Dictionary	17
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.	17
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).	23
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.	24
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.	26

List of Tables:

Table 1.1a:	First five rows of the dataset used in the present analysis. The dataset consists of seven columns.	6
Table 2.1b:	Last five rows of the dataset used in the present analysis.	6
Table 1.2	The statistical summary of the dataset used in the present analysis.	7
Table 1.3	The counts of the data and their variations present in the dataset.	7
Table 1.4	The amount of skewness of the continuous variables.	10
Table 1.5a	The correlation between the two categorical variables 'color' and 'cut'.	14
Table 1.5b	The correlation between the two categorical variables 'clarity' and 'cut'.	14
Table 1.5c	The correlation between the two categorical variables 'clarity' and 'color'.	15
Table 1.6	Screenshot of the table formed with the first 5 rows of the dataset after one-hot encoding.	17
Table 1.7	The index and the obtained coefficient using <i>regression_model</i> .	18
Table 1.8	Regression result estimated by OLS (ordinary least squares)	19
Table 1.9	Comparative analysis of the performance parameters	20
Table 1.10	Regression result estimated by OLS (ordinary least squares) after dropping 'depth'.	21
Table 2.1a	First five rows of the dataset used in the present analysis. The dataset consists of seven columns.	23
Table 2.1b	Last five rows of the dataset used in the present analysis.	24
Table 2.2	The statistical summary of the dataset used in the present analysis.	24
Table 2.3	The counts of the data and their variations present in the dataset.	25
Table 2.4	Screenshot of the table formed with the first 5 rows of the dataset after one-hot encoding.	29
Table 2.5	Split of the dataset and trace the performance of the split.	30
Table 2.6	Performance of Predictions on Train & Test sets.	30
Table 2.7	Confusion matrix, precision, recall, and F1 score provides better insights	32

List of Figures:

Figure 1.1a	Screenshot of the distplot and boxplot distribution for the variable 'carat'.	8
Figure 1.1b	Screenshot of the distplot and boxplot distribution for the variable 'depth'.	8
Figure 1.1c	Screenshot of the distplot and boxplot distribution for the variable 'table'.	8
Figure 1.1d	Screenshot of the distplot and boxplot distribution for the variable 'x'.	9
Figure 1.1e	Screenshot of the distplot and boxplot distribution for the variable 'y'.	9
Figure 1.1f	Screenshot of the distplot and boxplot distribution for the variable 'z'.	10
Figure 1.1g	Screenshot of the distplot and boxplot distribution for the variable 'price'.	10
Figure 1.2a	Distribution of variable 'cut'.	11
Figure 1.2b	Distribution of variable 'color'.	11
Figure 1.2c	Distribution of variable 'clarity'.	11
Figure 1.3	The pairplot for the given Dataset. It shows the interrelationship of all the variables available.	12
Figure 1.4	Heatmap for the given dataset, shows the multi-collinearity of all the variables.	13
Figure 1.5	The interrelationship of the categorical variables are represented.	15
Figure 1.6	The interrelationship of the categorical variables are represented.	15
Figure 1.7	Heatmap for the given dataset, shows the similar agreement before the outlier treatment.	16
Figure 1.8	The plot showing the change of predicted value with the actual one.	20
Figure 2.1a	Screenshot of the distplot, boxplot, and swarmplot distribution for the variable 'salary'.	25
Figure 2.1b	Screenshot of the distplot and boxplot distribution for the variable 'age'.	26
Figure 2.1c	Screenshot of the distplot and boxplot distribution for the variable 'educ'.	26
Figure 2.1d	Screenshot of the distplot and boxplot distribution for the variable 'no_young_children'.	26
Figure 2.1e	Screenshot of the distplot and boxplot distribution for the variable 'no_older_children'.	27
Figure 2.2	The count of the employees opted holiday package and select foreign trip.	27
Figure 2.3	The pairplot for the given Dataset. It shows the interrelationship of all the variables available.	28
Figure 2.4	Heatmap for the given dataset, shows the no collinearity of all the variables.	28
Figure 2.5	Variables in the dataset before and after outlier treatment.	29

Problem 1: Linear Regression

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. The aim of the present problem is to predict 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. The data dictionary is given.

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. 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.

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.

The given dataset is in .csv format. The dataset is loaded using '`pd.read_csv("")`' command. A variable is assigned to denote the loaded dataset. The `head()` and `tail()` functions are used to check the first and last five rows of the loaded dataset as shown in Table 1.1a-b.

Table 3.1a: First five rows of the dataset used in the present analysis. The dataset consists of seven 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
1	2	0.33	Premium	G	IF	60.8	58.0	4.42	4.46	2.70	984
2	3	0.90	Very Good	E	VVS2	62.2	60.0	6.04	6.12	3.78	6289
3	4	0.42	Ideal	F	VS1	61.6	56.0	4.82	4.80	2.96	1082
4	5	0.31	Ideal	F	VVS1	60.4	59.0	4.35	4.43	2.65	779

Table 1.1b: Last five rows of the dataset used in the present analysis.

	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

The shape and the detailed information about the column and row numbers, the data types, memory used, are obtained in the initial steps of the data analysis. The statistical summary of the dataset is illustrated in Table 1.2.

Insights:

1. The dataset appears to be flawless.
2. The dataset consists of both categorical and continuous entries.
3. The dataset consists of 26967 rows and 10 columns after dropping the column named as '**Unnamed: 0**'.
4. The 'info' of the dataset indicates the variables are float64 (6), int64 (1) and object (3) type. The number within the braces indicates the quantity of the data of the particular type present in the dataset.
5. The categorical data are **cut**, **colour** and **clarity** while **carat**, **depth**, **table**, **x**, **y**, **z** and **price** are continuous data.
6. **Price** is the target variable.
7. There are 697 null values and 34 duplicate entries in the dataset.
8. The price of the cubic zirconia varies from 326 to 18818 units.
9. The weight of the zirconia varies from 0.2 to 4.5 carat.

Table 1.4: The statistical summary of the dataset used in the present analysis.

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
carat	26967.0	NaN	NaN	NaN	0.798375	0.477745	0.2	0.4	0.7	1.05	4.5
cut	26967	5	Ideal	10816	NaN	NaN	NaN	NaN	NaN	NaN	NaN
color	26967	7	G	5661	NaN	NaN	NaN	NaN	NaN	NaN	NaN
clarity	26967	8	SI1	6571	NaN	NaN	NaN	NaN	NaN	NaN	NaN
depth	26270.0	NaN	NaN	NaN	61.745147	1.41286	50.8	61.0	61.8	62.5	73.6
table	26967.0	NaN	NaN	NaN	57.45608	2.232068	49.0	56.0	57.0	59.0	79.0
x	26967.0	NaN	NaN	NaN	5.729854	1.128516	0.0	4.71	5.69	6.55	10.23
y	26967.0	NaN	NaN	NaN	5.733569	1.166058	0.0	4.71	5.71	6.54	58.9
z	26967.0	NaN	NaN	NaN	3.538057	0.720624	0.0	2.9	3.52	4.04	31.8
price	26967.0	NaN	NaN	NaN	3939.518115	4024.864666	326.0	945.0	2375.0	5360.0	18818.0

The data have the following number of unique variations as describe in Table 1.3. Referring to the given data dictionary, 1443 zirconia are of best color available, where the number of worst colored zirconia available is 3344.

Table 1.5: The counts of the data and their variations present in the dataset.

a. Name: cut, dtype: int64		b. Name: color, dtype: int64	
Fair	781	J	1443
Good	2441	I	2771
Very Good	6030	D	3344
Premium	6899	H	4102
Ideal	10816	F	4729
		E	4917
		G	5661
c. Name: clarity, dtype: int64		d. dtype: int64	
I1	365	carat	257
IF	894	depth	169
VVS1	1839	table	112
VVS2	2531	x	531
VS1	4093	y	526
SI2	4575	z	356
VS2	6099	price	8742
SI1	6571		

Exploratory Data Analysis (Univariate / Bivariate) helps to understand the distribution of data in the dataset. The distribution of the continuous variables are evaluated by both distplot and boxplot. The distplot represents data distribution of a variable against the density distribution, whereas the boxplot is a standardized way of displaying the distribution of data based on a five number summary and reveals the presence of outliers.

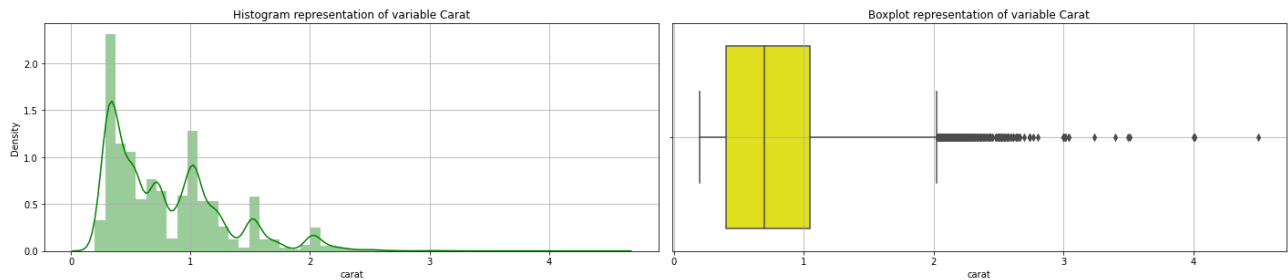


Figure 1.1a: Screenshot of the distplot and boxplot distribution for the variable 'carat'.

Insights:

1. The distplot shows the distribution of data from 0 to 1.
2. The distribution of data in 'carat' seems to positively skew.
3. There could be chance of multi modes in the dataset.
4. The box plot of 'carat' shows multiple outliers.

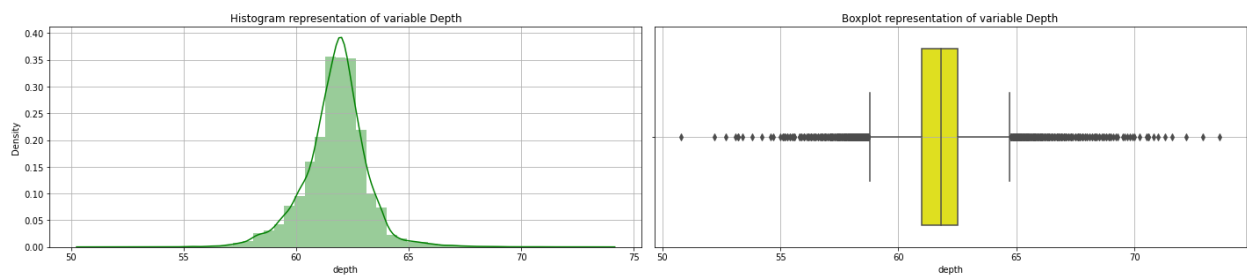


Figure 1.1b: Screenshot of the distplot and boxplot distribution for the variable 'depth'.

Insights:

1. The distplot shows the distribution of data from 55 to 65.
2. The distribution of 'depth' seems to be normal distribution.
3. The box plot of 'depth' shows multiple outliers.

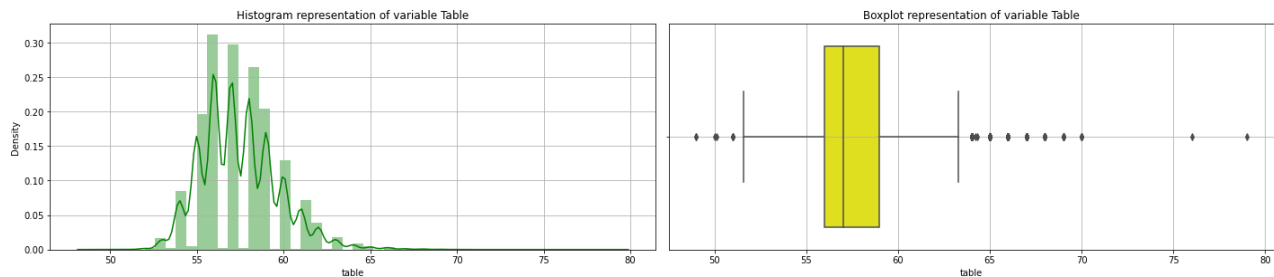


Figure 1.1c: Screenshot of the distplot and boxplot distribution for the variable 'table'.

Insights:

1. The distplot shows the maximum distribution varies in between 55 to 65.
2. The distribution of data in 'table' seems to positively skew.
3. There could be chance of multi modes in the dataset.
4. The box plot of 'table' shows outliers.

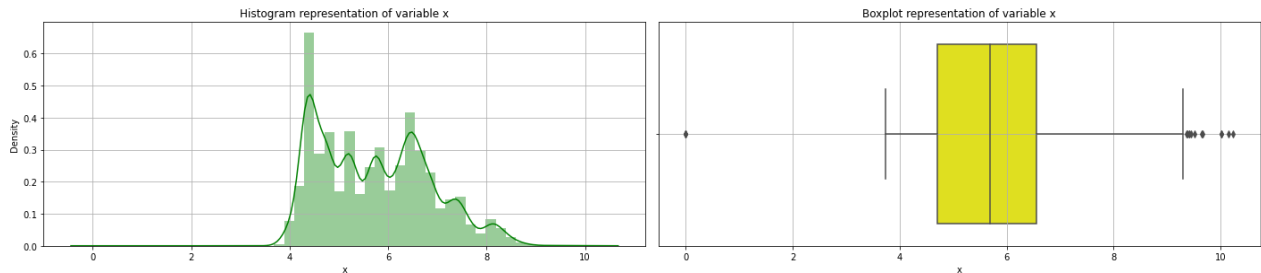


Figure 1.1d: Screenshot of the distplot and boxplot distribution for the variable 'x (Length of the cubic zirconia in mm)'.

Insights:

1. The distplot shows the distribution of data from 4 to 8.
2. The distribution of data in 'x or the length of the cubic zirconia in mm' seems to positively skew.
3. There could be chance of multi modes in the dataset.
4. The box plot of 'x' shows numbers of outlier.

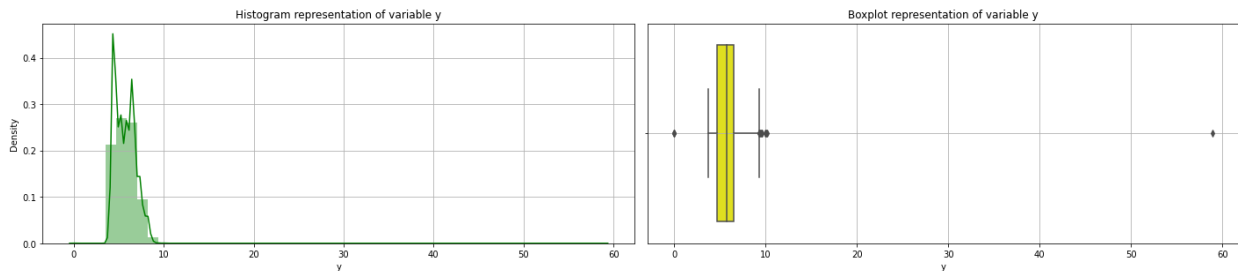


Figure 1.1e: Screenshot of the distplot and boxplot distribution for the variable 'y (Width of the cubic zirconia in mm)'.

Insights:

1. The distplot shows the distribution of data from 5 to 9.
2. The distribution of data in 'y or the width of the cubic zirconia in mm' seems too extremely positive skew. This might be because of less amount of sizes available in the market.
3. There could be chance of multi modes in the dataset.
4. The box plot of 'y' shows a few outliers.

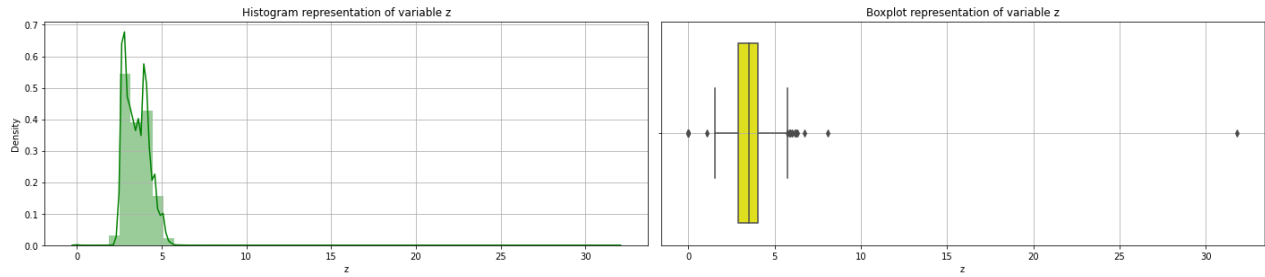


Figure 1.1f: Screenshot of the distplot and boxplot distribution for the variable 'z (Height of the cubic zirconia in mm)'.

Insights:

1. The distplot shows the distribution of data from 2.5 to 5.
2. The distribution of data in 'z or the height of the cubic zirconia in mm' seems extremely positive skew.
3. There could be chance of multi modes in the dataset.

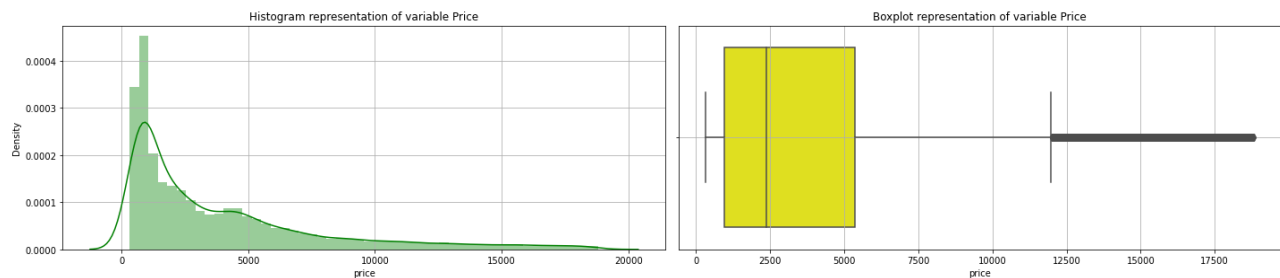


Figure 1.1g: Screenshot of the distplot and boxplot distribution for the variable 'price'.

Insights:

1. The distplot shows the distribution of data from 100 to 9000.
2. The distribution of data in 'carat' seems to positively skew.
3. The box plot of 'price' shows multiple outliers.

The degree of skewness is evaluated using the '*skew()*' command, which in turn reveals the amount of skewness present in the dataset as shown in the Table 1.4.

Table 1.6: The amount of skewness of the continuous variables.

Continuous variables	Skewness
carat	1.116481
depth	-0.028618
table	0.765758
x or height of the cubic zirconia in mm	0.387986
y or height of the cubic zirconia in mm	3.850189
z height of the cubic zirconia in mm	2.568257
price	1.618550

Bivariate analysis is appropriate to examine the distribution of the categorical variables. The categorical variables present in the dataset are 'cut', 'color', and 'clarity'.

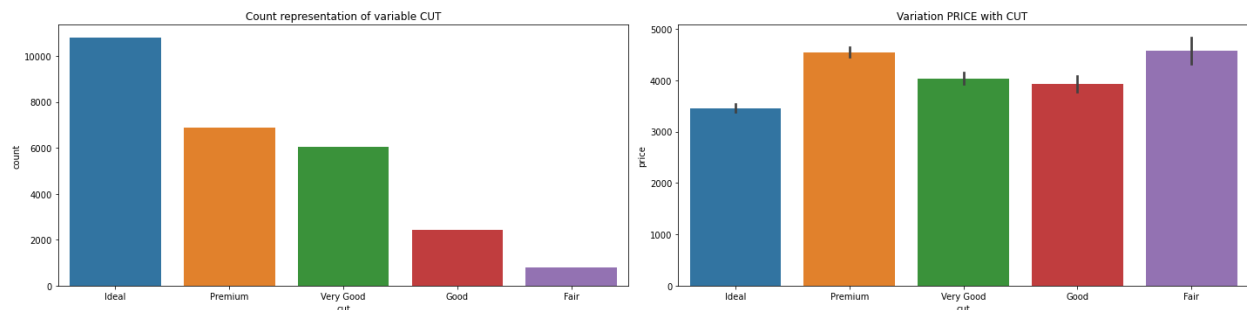


Figure 1.2a: Distribution of variable 'cut'.

Insights:

1. Quality is increasing order Fair, Good, Very Good, Premium, Ideal.
2. The most preferred cut seems to be 'ideal' for zirconia because those diamonds are priced lower than other cuts.

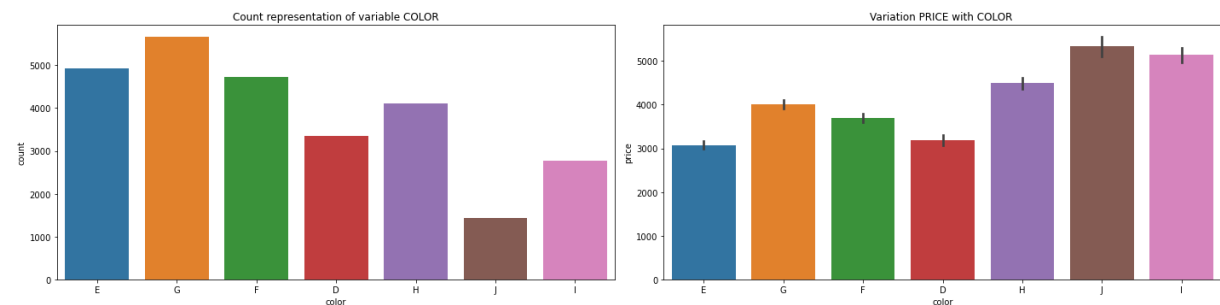


Figure 1.2b: Distribution of variable 'color'.

Insights:

1. 'D' being the best and 'J' the worst as per the data dictionary.
2. Among the available 7 colors in the data, 'G' seems to be the preferred colour.
3. 'D' is priced in the middle of the seven colors, whereas 'J' being the colour that priced too high.

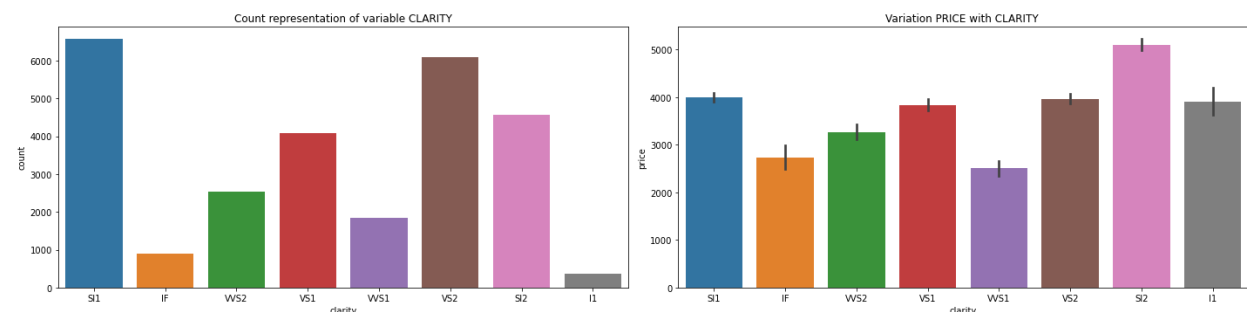


Figure 1.2c: Distribution of variable 'clarity'.

Insights:

1. As per the data dictionary, 'clarity' classified in order from worst to best in terms of avg. price. IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1.
2. The 'clarity' SI2 seems to be most preferred category and seems to be most costlier.

Multivariate analysis is a Statistical procedure for analysis of data involving more than one type of measurement or observation. It may also mean solving problems where more than one dependent variable is analyzed simultaneously with other variables. For multivariate analysis pairplot (Figure 1.3) and heatmap (Figure 1.4) are two useful methods applied here.

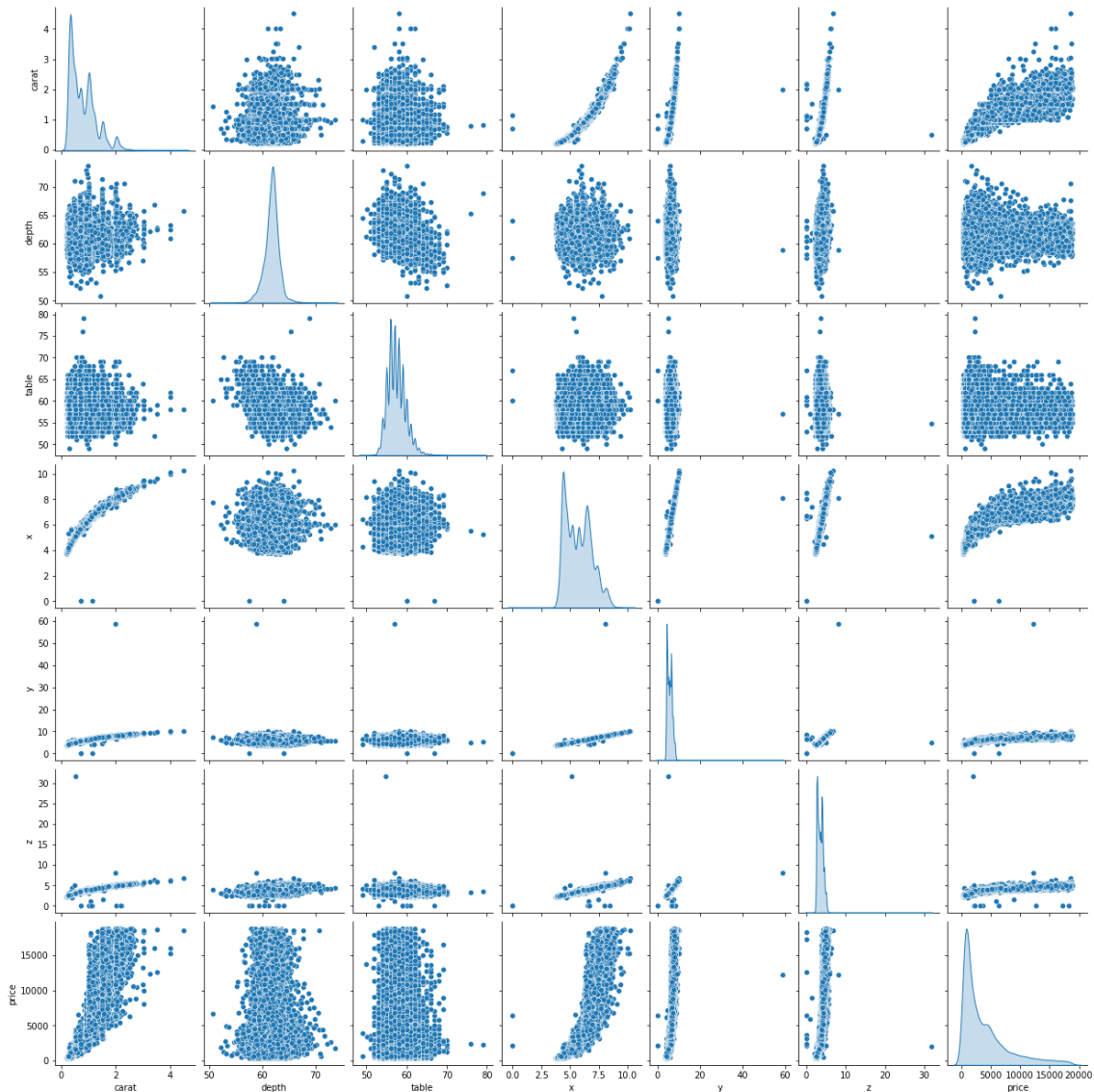


Figure 1.3: The pairplot for the given Dataset. It shows the interrelationship of all the variables available.

Insights:

1. Strong positive correlation between the variables:

- a. carat & x or the length of the cubic zirconia in mm.
 - b. carat & y or the width of the cubic zirconia in mm.
 - c. carat & z or the height of the cubic zirconia in mm.
 - d. carat & price
 - e. x & y
 - f. x & z
 - g. y & z
2. Good positive correlation between the variables:
 - a. price & x
 - b. price & y
 - c. price & z
 3. The price increase with the weight (carat) of the zirconia. The length, width and height of a zirconia are equally proportional. Any change of the dimension linearly change the price of the gem.
 4. The price increase with the dimension of the zirconia.

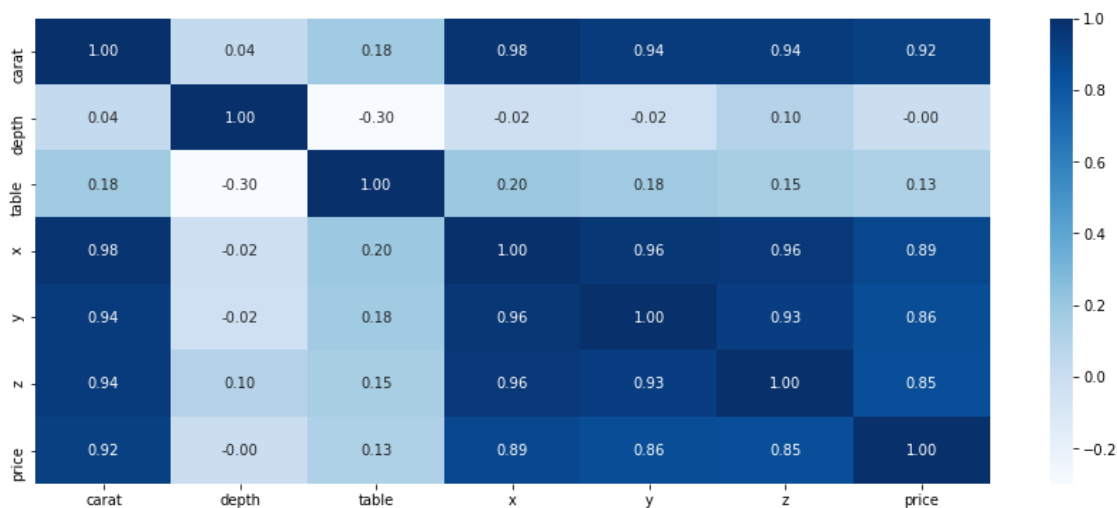


Figure 1.4: Heatmap for the given dataset, shows the multi-collinearity of all the variables.

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.

The total null values arrested is 697 under 'depth' variable which is about 2.5% of the total value. Though those null entries can be dropped, but in this work, those entries are imputed by the median value of the said variable, to keep the dataset shape fixed.

Investigation revealed the presence of zeros in the dataset under the variables x, y and z, which represent the dimensions, length, width and height of a cubic zirconia in mm. It is obvious, those said parameters never be zero, and somehow the information were mistakenly registered in the

dataset. Modification of the zero value, may change the entire flavor of the dataset, hence those entries are dropped from the dataset.

In the present dataset three categorical attributes are given, 'cut', 'color', and 'clarity'. Each variables have their own categories, as per the data dictionary.

'cut' : The cut quality of the cubic zirconia 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.

Though the categories are in order from worst to best, but the spacing between the values may not be the same across the levels of the variables. Scores like 1, 2, 3 and 4 can be assigned to these levels and draw the interrelationship to justify the price of the gem varies with the features listed in the categorical variables.

In the present work assigning number to the three categorical variables would cause inconsistent outcome. The interrelation between the categorical variables are tabulated in Table 1.5a-c.

Table 1.7a: The correlation between the two categorical variables 'color' and 'cut'.

color	D	E	F	G	H	I	J
cut							
Fair	74	100	148	147	150	94	68
Good	311	491	454	419	352	253	161
Ideal	1409	1966	1893	2470	1552	1073	453
Premium	808	1174	1167	1471	1161	711	407
Very Good	742	1186	1067	1154	887	640	354

Table 1.5b: The correlation between the two categorical variables 'clarity' and 'cut'.

clarity	I1	IF	SI1	SI2	VS1	VS2	VVS1	VVS2
cut								
Fair	89	4	193	225	93	129	10	38
Good	51	30	765	530	331	491	100	143
Ideal	74	613	2150	1324	1784	2528	1036	1307
Premium	108	115	1809	1449	998	1697	307	416
Very Good	43	132	1654	1047	887	1254	386	627

Table 1.5c: The correlation between the two categorical variables ‘clarity’ and ‘color’.

clarity	I1	IF	SI1	SI2	VS1	VS2	VVS1	VVS2
color								
D	25	38	1040	671	369	804	121	276
E	54	87	1249	849	625	1202	342	509
F	67	183	1088	751	672	1107	360	499
G	67	342	1001	778	1077	1205	507	681
H	81	149	1081	795	595	803	288	306
I	48	69	725	469	480	603	183	194
J	21	26	386	258	274	374	38	66

Figure 1.5, ‘G’ colored zirconia with ‘ideal’ cut is available most. The chance of ‘G’ colored with VS2 clarity is high. The best color ‘D’ with ‘premium’ cut with ‘I1’ clarity is very rare; only 25 in 26967 zirconia.

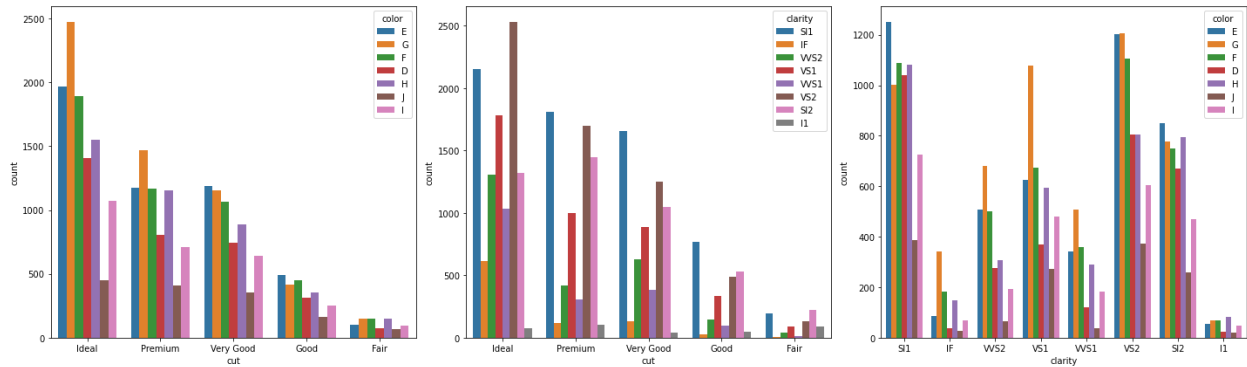


Figure 1.5: The interrelationship of the categorical variables are represented.

Outliers increase the variability in data, which decreases statistical power. Consequently, excluding outliers can cause results to become statistically significant. The continuous variables present in the dataset are affected by large number of outliers, so get the statistically significant result outliers are treated (Figure 1.6) in the present work.

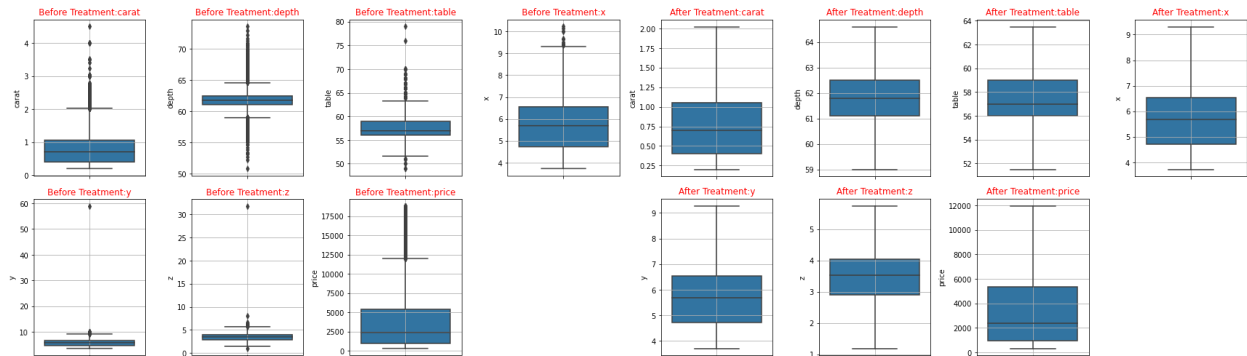


Figure 1.6: Variables in the dataset before and after outlier treatment.

Moreover, the correlation agreement between the variables remain unaltered after outlier treatment (Figure 1.7).

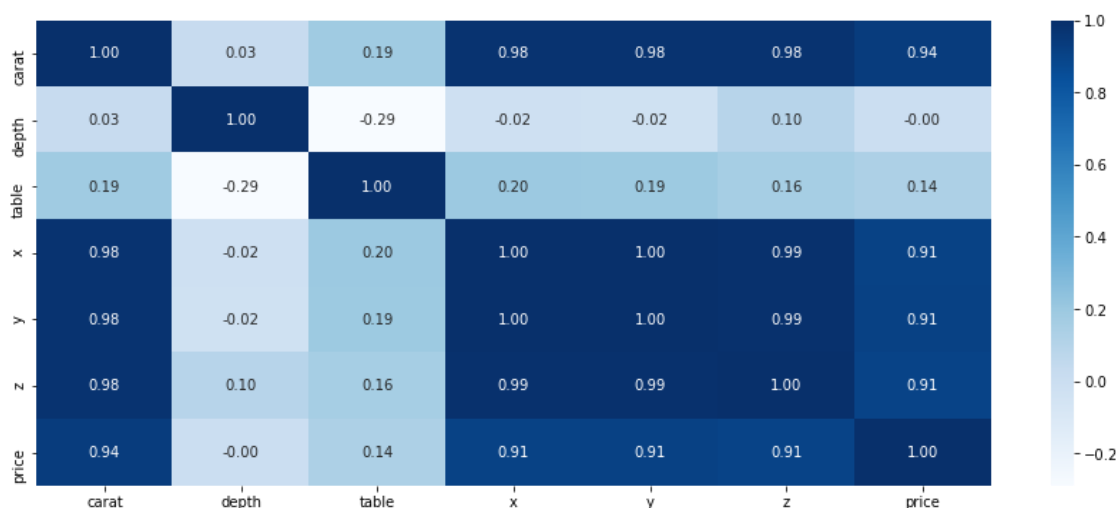


Figure 1.7: Heatmap for the given dataset, shows the similar agreement before the outlier treatment.

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 statsmodel. 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.

The dataset consists of three categorical variables, 'cut', 'color', and 'clarity' with numbers of categories. To improve predictions those categories are needed to be configured into numerical data. One-hot encoding is essentially the representation of categorical variables as binary vectors. These categorical values are first mapped to integer values. Each integer value is then represented as a binary vector that is all 0s except the index of the integer which is marked as 1.

In this context, dummy variable is used to segregate between different sub-groups of the data for better regression analysis. The following code is used to get the dummy variable.

```
data_df = pd.get_dummies(df, columns=['cut', 'color', 'clarity'], drop_first=True)
```

Here, *data_df* is the variable to which the dummy variable is assigned. *pd.get_dummies* is the command to create dummy variables in python. *drop_first = True* command is used in reducing the extra column created during dummy variable creation, which in turn reduces the correlations created among dummy variables.

Table 1.6 illustrates the modified dataset, after one-hot encoding. Using *drop_first = True* the column size reduced to 24 from 27, which improve the computation speed, reduce the correlation complexity, and the memory size.

Table 1.6: Screenshot of the table formed with the first 5 rows of the dataset after one-hot encoding.

	carat	depth	table	x	y	z	price	cut_Good	cut_Ideal	cut_Premium	...	color_H	color_I	color_J	clarity_IF	clarity_SI1	clarity_SI2	clarity_VS1
0	0.30	62.1	58.0	4.27	4.29	2.66	499.0	0	1	0	...	0	0	0	0	1	0	0
1	0.33	60.8	58.0	4.42	4.46	2.70	984.0	0	0	1	...	0	0	0	1	0	0	0
2	0.90	62.2	60.0	6.04	6.12	3.78	6289.0	0	0	0	...	0	0	0	0	0	0	0
3	0.42	61.6	56.0	4.82	4.80	2.96	1082.0	0	1	0	...	0	0	0	0	0	0	1
4	0.31	60.4	59.0	4.35	4.43	2.65	779.0	0	1	0	...	0	0	0	0	0	0	0

5 rows × 24 columns

The index of the newly formed encoded dataset are like:

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

The scaling has no impact in model score or coefficients of attributes nor the intercept. It helps to reduce the high inflation factors. Thus before splitting the dataset processed by *StandardScaler()* function. To split the data into train and test set in 70:30 ratio, the following commands are suitable:

```
X = data_df.drop('price', axis=1)
y = data_df[['price']]
```

For training and testing purpose the dataset can be split into train and test data in the ratio 70:30 following the '*X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.30, random_state = 1)*' command.

The dimensions of the train and test data are obtained as follows:

The dimension of X_train: (18870, 23)

The dimension of X_test: (8088, 23)

The dimension of y_train: (18870, 1)

The dimension of y_test: (8088, 1)

Application of Scikit-learn library to find best fit Linear Regression model on training data the following code used.

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

The coefficients for each of the independent attributes for the model (Table 1.7), obtained using the following instruction:

```
for idx, col_name in enumerate(X_train.columns):
    print("The coefficient for {} is {}".format(col_name, regression_model.coef_[0][idx]))
```

Table 1.7: The index and the obtained coefficient using *regression_model*.

Index	Coefficient without scaling	Coefficient with scaling
carat	9274.42	1.1
depth	16.18	0.01
table	-24.02	-0.01
x	-1089.27	-0.31
y	1050.17	0.3
z	-780	-0.14
cut_Good	378.33	0.09
cut_Ideal	612.79	0.15
cut_Premium	597.57	0.15
cut_Very Good	506.29	0.13
color_E	-189.31	-0.05
color_F	-252.2	-0.06
color_G	-405.23	-0.1
color_H	-835.53	-0.21
color_I	-1303.36	-0.32
color_J	-1885.27	-0.47
clarity_IF	4022.36	1
clarity_SI1	2570.79	0.64
clarity_SI2	1728.39	0.43
clarity_VS1	3371.87	0.84
clarity_VS2	3081.93	0.77
clarity_VVS1	3790.24	0.94
clarity_VVS2	3747.16	0.93

R^2 is an important indicator to know the goodness-of-fit measure for linear regression models. It measures the strength of the relationship between the model and the dependent variable.

Linear regression using '*Statsmodel*' estimation by ordinary least squares:

R^2 is not a reliable metric as it always increases with addition of more attributes even if the attributes have no influence on the predicted variable. Whereas adjusted R^2 which removes the statistical chance that improves R^2 which '*Scikit*' does not provide a facility for adjusted R^2 . The '*Statsmodel*', a library is used to get adjusted R^2 for building a Linear Regression model.

The inferential statistics obtained shown in Table 1.8

In hypothesis testing, first the null hypothesis is formulated. The null hypothesis claims there is no relationship between the dependent ('price') and the independent ('carat', 'depth', 'table', 'x', 'y', 'z', 'price', 'cut_Good', 'cut_Ideal', 'cut_Premium', 'cut_Very Good', 'color_E', 'color_F', 'color_G', 'color_H', 'color_I', 'color_J', 'clarity_IF', 'clarity_SI1', 'clarity_SI2', 'clarity_VS1', 'clarity_VS2', 'clarity_VVS1', and 'clarity_VVS2') attributes. Thus the coefficients between the 'price' and other independent attributes are zero in the universe.

Table 1.8: Regression result estimated by OLS (ordinary least squares)

OLS Regression result						
=====						
Dep. Variable:	price	R-squared:	0.942			
Model:	OLS	Adj. R-squared:	0.942			
Method:	Least Squares	F-statistic:	1.330e+04			
Date:	Fri, 17 Dec 2021	Prob (F-statistic):	0.00			
Time:	23:04:57	Log-Likelihood:	2954.6			
No. Observations:	18870	AIC:	-5861.			
Df Residuals:	18846	BIC:	-5673.			
Df Model:	23					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]

Intercept	-0.7568	0.016	-46.999	0.000	-0.788	-0.725
carat	1.1009	0.009	121.892	0.000	1.083	1.119
depth	0.0056	0.004	1.525	0.127	-0.002	0.013
table	-0.0133	0.002	-6.356	0.000	-0.017	-0.009
x	-0.3050	0.032	-9.531	0.000	-0.368	-0.242
y	0.3039	0.034	8.934	0.000	0.237	0.371
z	-0.1392	0.024	-5.742	0.000	-0.187	-0.092
cut_Good	0.0940	0.011	8.755	0.000	0.073	0.115
cut_Ideal	0.1523	0.010	14.581	0.000	0.132	0.173
cut_Premium	0.1485	0.010	14.785	0.000	0.129	0.168
cut_Very_Good	0.1258	0.010	12.269	0.000	0.106	0.146
color_E	-0.0471	0.006	-8.429	0.000	-0.058	-0.036
color_F	-0.0627	0.006	-11.075	0.000	-0.074	-0.052
color_G	-0.1007	0.006	-18.258	0.000	-0.112	-0.090
color_H	-0.2077	0.006	-35.323	0.000	-0.219	-0.196
color_I	-0.3240	0.007	-49.521	0.000	-0.337	-0.311
color_J	-0.4686	0.008	-58.186	0.000	-0.484	-0.453
clarity_IF	0.9998	0.016	62.524	0.000	0.968	1.031
clarity_SI1	0.6390	0.014	46.643	0.000	0.612	0.666
clarity_SI2	0.4296	0.014	31.177	0.000	0.403	0.457
clarity_VS1	0.8381	0.014	59.986	0.000	0.811	0.865
clarity_VS2	0.7660	0.014	55.618	0.000	0.739	0.793
clarity_VVS1	0.9421	0.015	63.630	0.000	0.913	0.971
clarity_VVS2	0.9314	0.014	64.730	0.000	0.903	0.960
=====						
Omnibus:	4696.785	Durbin-Watson:	1.994			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	17654.853			
Skew:	1.208	Prob(JB):	0.00			
Kurtosis:	7.076	Cond. No.	57.0			
=====						

Assuming the null hypothesis is true, then probability of the sample drawn from the universe to real world to build a model having coefficient given in the 'coef' column in Table 1.8, have to determine. That probably is expressed in 'P>|t|' column. Analysis the attribute, the 'depth' is related to the 'price' with coefficient value 0.0056 and the corresponding p-Value is 0.127 which is much higher than 0.05. So, the attribute 'depth' is useless as coefficient appear by chance. Hence statistically the coefficient obtained from the attribute 'depth' is unreliable and should be dropped in making in making robust model.

The visual representation (Figure 1.8) informs how good the model is by plotting actual versus predicted value. There is a strong correlation between the two variables. There is lots of spread

in the data points especially after 1.5 of the actual value. This spread indicate the presence of some kind of noise in the dataset.

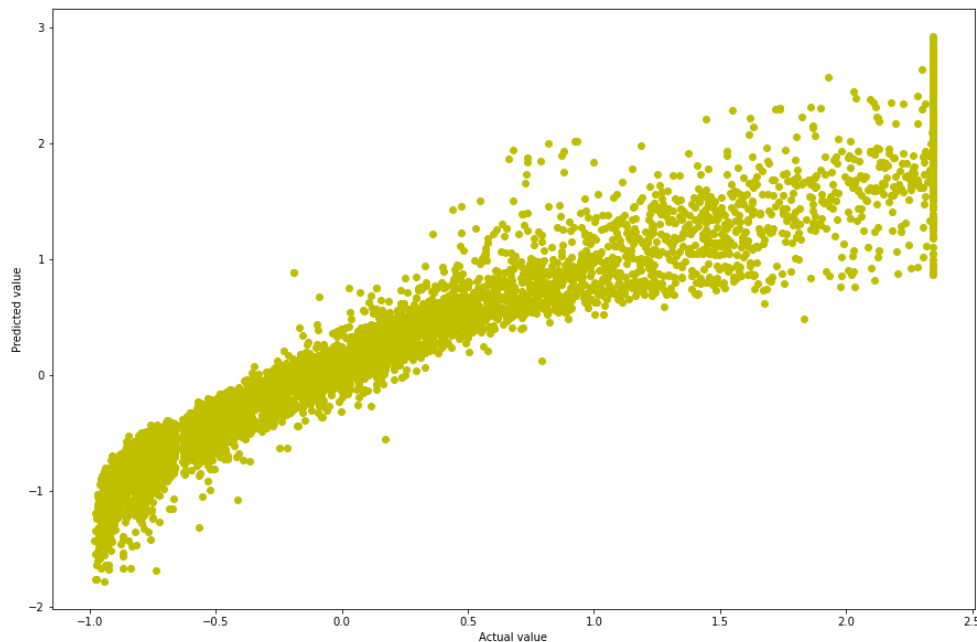


Figure 1.8: The plot showing the change of predicted value with the actual one.

To improve the Linear Regression model using the '*Stastmodel*', the '*depth*' attribute needs to be dropped. For that reason the test and train dataset are scaled using '*zscore*'. All the zscored-columns are equivalent as there is no difference between them in terms of scale.

After re-building the Linear Regression model using the '*fit*' command on the scaled data, the coefficients and the intercept are evaluated. The intercept for the model is $3.473986556018693e^{-17}$ which is almost zero for all the practical purposes. Thus the use of the '*zscale*' does not contribute anything except reducing the intercept value.

The performance parameters of model are tabulated in Table 1.9. According to the table, both the functions offer very stable model. According to the thumb rule, RMSE values between 0.2 and 0.5 shows that the model can relatively predict the data accurately.

Table 1.9: Comparative analysis of the performance parameters

Function	Model score - R2		RMSE		Adj. R2
	Train Data	Test Data	Train Data	Test Data	
Sciket-learn	0.942	0.938	0.206	0.216	NA
Statsmodel- Iteration I	0.941	0.938	0.477	0.491	0.942
Statsmodel- Iteration II	0.942	0.938	0.241	0.248	0.942

Dropping '*depth*' the inferential statistics obtained shown in Table 1.10

The model now improved as no attributes having p-Value greater than 0.05. The intercept is close to zero.

Table 1.10: Regression result estimated by OLS (ordinary least squares) after dropping 'depth'.

OLS Regression result						
=====						
Dep. Variable:	price	R-squared:	0.942			
Model:	OLS	Adj. R-squared:	0.942			
Method:	Least Squares	F-statistic:	1.390e+04			
Date:	Fri, 17 Dec 2021	Prob (F-statistic):	0.00			
Time:	23:04:58	Log-Likelihood:	80.669			
No. Observations:	18870	AIC:	-115.3			
Df Residuals:	18847	BIC:	65.10			
Df Model:	22					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]

Intercept	6.765e-17	0.002	3.85e-14	1.000	-0.003	0.003
carat	1.2352	0.010	122.331	0.000	1.215	1.255
table	-0.0156	0.002	-6.770	0.000	-0.020	-0.011
x	-0.3658	0.036	-10.101	0.000	-0.437	-0.295
y	0.3157	0.035	9.069	0.000	0.247	0.384
z	-0.1224	0.016	-7.883	0.000	-0.153	-0.092
cut_Good	0.0319	0.004	8.876	0.000	0.025	0.039
cut_Ideal	0.0863	0.006	14.508	0.000	0.075	0.098
cut_Premium	0.0744	0.005	14.711	0.000	0.064	0.084
cut_Very_Good	0.0613	0.005	12.239	0.000	0.051	0.071
color_E	-0.0211	0.003	-8.439	0.000	-0.026	-0.016
color_F	-0.0277	0.002	-11.082	0.000	-0.033	-0.023
color_G	-0.0478	0.003	-18.246	0.000	-0.053	-0.043
color_H	-0.0872	0.002	-35.306	0.000	-0.092	-0.082
color_I	-0.1148	0.002	-49.497	0.000	-0.119	-0.110
color_J	-0.1222	0.002	-58.169	0.000	-0.126	-0.118
clarity_IF	0.2090	0.003	62.544	0.000	0.202	0.216
clarity_SI1	0.3200	0.007	46.738	0.000	0.307	0.333
clarity_SI2	0.1875	0.006	31.232	0.000	0.176	0.199
clarity_VS1	0.3517	0.006	60.042	0.000	0.340	0.363
clarity_VS2	0.3729	0.007	55.691	0.000	0.360	0.386
clarity_VVS1	0.2757	0.004	63.655	0.000	0.267	0.284
clarity_VVS2	0.3179	0.005	64.784	0.000	0.308	0.328
=====						
Omnibus:	4699.504	Durbin-Watson:	1.994			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	17704.272			
Skew:	1.208	Prob(JB):	0.00			
Kurtosis:	7.084	Cond. No.	58.7			
=====						

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.)

Predicting the price of the stone and providing insights on profits from different prize slots are the most important objectives of this problem. The colors H, I, and J generated profits for the company. The ideal, premium and very good types of cuts bring profits, while fair and good do

not. Approximately 95% of the variation in the price could be captured by the predictions in the training set.

The stats model can be re-run in order to get p values and coefficients and a better understanding of the relationship. If the p value is more than 0.05, then we can drop those variables and run the model again for better results. Dropping the depth column in iteration will lead to better results. As a result of the model, the following equation provides the final relationship.

$$\begin{aligned} \text{Price} = & (0.0) * \text{Intercept} + 1.24 * \text{carat} - 0.02 * \text{table} - 0.37 * x + 0.32 * y - 0.12 * z + 0.03 * \text{cut_Good} \\ & + 0.09 * \text{cut_Ideal} + 0.07 * \text{cut_Premium} + 0.06 * \text{cut_Very_Good} - 0.02 * \text{color_E} - 0.03 * \text{color_F} + - \\ & 0.05 * \text{color_G} + (-0.09) * \text{color_H} - 0.11 * \text{color_I} - 0.12 * \text{color_J} + 0.21 * \text{clarity_IF} + 0.32 * \text{clarity_SI1} \\ & + 0.19 * \text{clarity_SI2} + 0.35 * \text{clarity_VS1} + (0.37) * \text{clarity_VS2} + 0.28 * \text{clarity_VVS1} + \\ & 0.32 * \text{clarity_VVS2} \end{aligned}$$

Recommendations

1. The premium, very good cut types are the ones bringing profits, so we can use marketing to increase profits.
2. A diamond's clarity is the next important attribute. The clearer the diamond is, the more profits it will generate.

Problem 2: Logistic Regression and LDA

A tour and travel agency which deals in selling holiday packages. The details of 872 employees of a company are given. Among these employees, some opted for the package and some didn't. The objective of the problem is 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. The data dictionary is given.

Data Dictionary:

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

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 *head()* and *tail()* functions are used to check the first and last five rows of the loaded dataset as shown in the Table 2.1a-b.

Table 2.1a: First five rows of the dataset used in the present analysis. The dataset consists of seven columns.

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

Table 2.1b: Last five rows of the dataset used in the present analysis.

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

The shape of the data and the detailed information about column and row numbers, data types, and memory used is obtained during the initial steps of data analysis. A summary of the dataset is shown in Table 2.2.

Table 2.2: The statistical summary of the dataset used in the present analysis.

	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

Insights:

1. The dataset appears to be flawless.
2. The dataset consists of both categorical and continuous entries.
3. The dataset consists of 872 rows and 7 columns after dropping the column named as 'Unnamed: 0'.
4. The 'info' of the dataset indicates the variables are, int64 (5) and object (2) type. The number within the braces indicates the quantity of the data of the particular type present in the dataset.
5. The categorical data are **Holliday_Package**, and **foreign** while **Salary**, **age**, **educ**, **no_young_children**, and **no_older_children** are continuous data.
6. **Holliday_Package** is the target variable.
7. The dataset is free from any null and duplicate entries.
8. The Salary varies from 1322.0 to 236961.0 units.
9. The age of the employee varies from 20 to 62.

The data have the following number of unique variations as describe in Table 2.3. Referring to the given data dictionary, 1443 zirconia are of best color available, where the number of worst colored zirconia available is 3344.

Table 2.3: The counts of the data and their variations present in the dataset.

a. Name: Holliday_Package, dtype: int64		b. Name: foreign, dtype: int64	
yes	401	yes	216
no	471	no	656
c. dtype: int64			
Salary	864		
age	43		
educ	20		
no_young_children	4		
no_older_children	7		

Salary, age, educ and number young children, number older children of employee have the went to foreign, these are the attributes needs to cross examine and help the company predict weather the person will opt for holiday package or not.

Univariate / Bivariate analysis helps to understand the distribution of data in the dataset. The chance of the employee to opt the holiday package to go foreign trip, depends upon the continuous attributes as illustrated in Figure 2.1a-e.

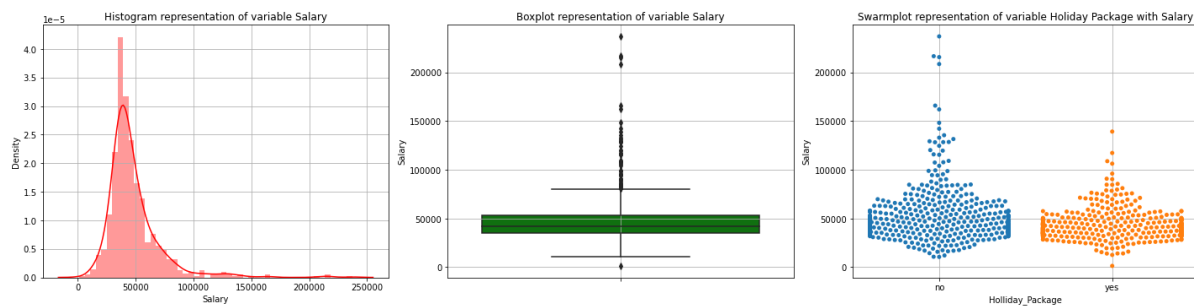


Figure 2.1a: Screenshot of the distplot, boxplot, and swarmplot distribution for the variable 'salary'.

Insights:

1. The distplot shows the distribution of data from 2000 to 200000.
2. The distribution of data in 'Salary' seems normal.
3. The box plot of 'Salary' shows multiple outliers.

4. Employee below salary 150000 have always opted for holiday package.

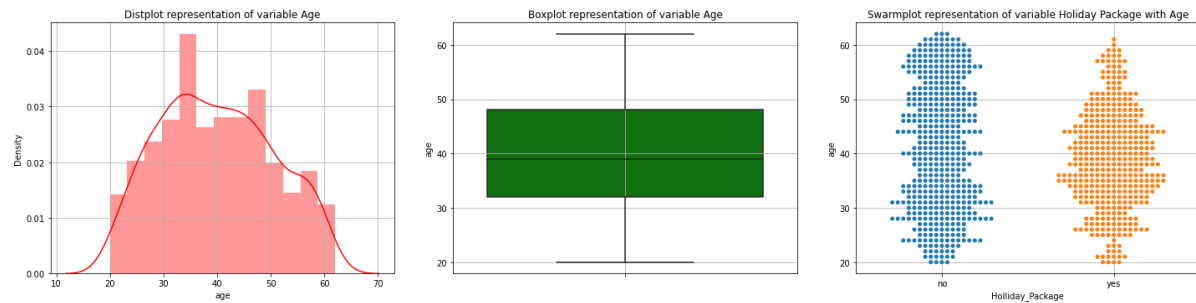


Figure 2.1b: Screenshot of the distplot and boxplot distribution for the variable 'age'.

Insights:

1. The distplot shows the distribution of data from 20 to 65.
2. The distribution of data in 'age' seems normal.
3. The box plot of 'age' shows zero outlier.
4. Employee age over 50 to 60 have seems to be not taking the holiday package, whereas in the age 30 to 50 and salary less than 50000 people have opted more for holiday package.

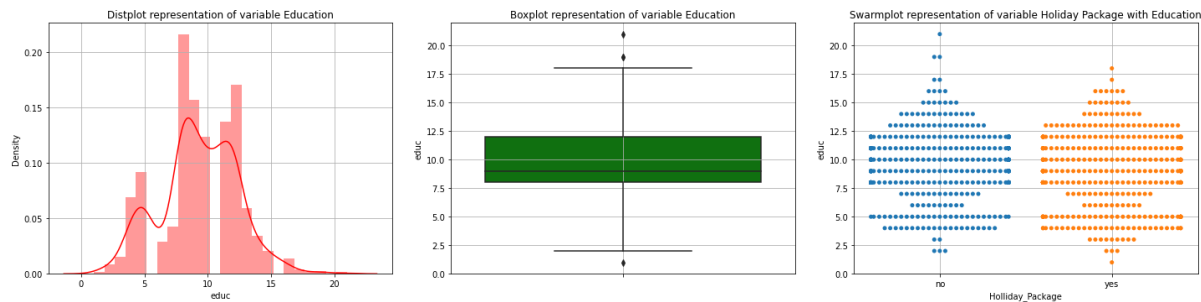


Figure 2.1c: Screenshot of the distplot and boxplot distribution for the variable 'educ'.

Insights:

1. The distplot shows the distribution of data from 2 to 16, though some values are absent.
2. The distribution of data in 'educ' seems normal with multiple peaks.
3. The box plot of 'educ' shows few outliers.
4. Education is not a bar to opt the holiday package.

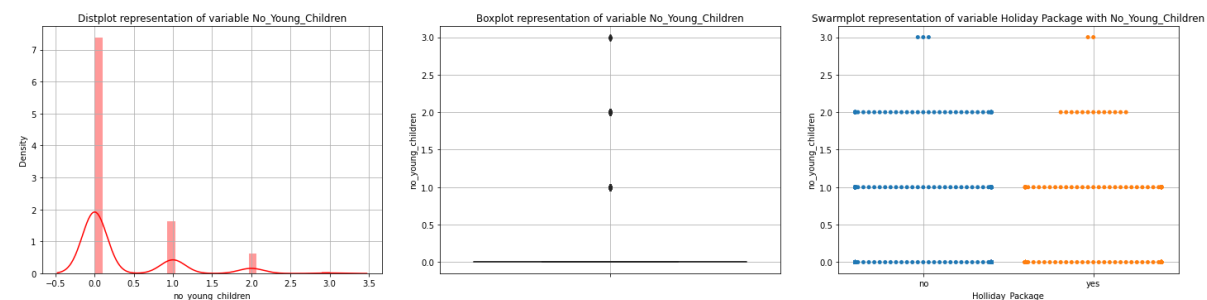


Figure 2.1d: Screenshot of the distplot and boxplot distribution for the variable 'no_young_children'.

Insights:

1. Data points are very few to form a proper distribution.
2. Having two young children reduce the probability to opt holiday package.

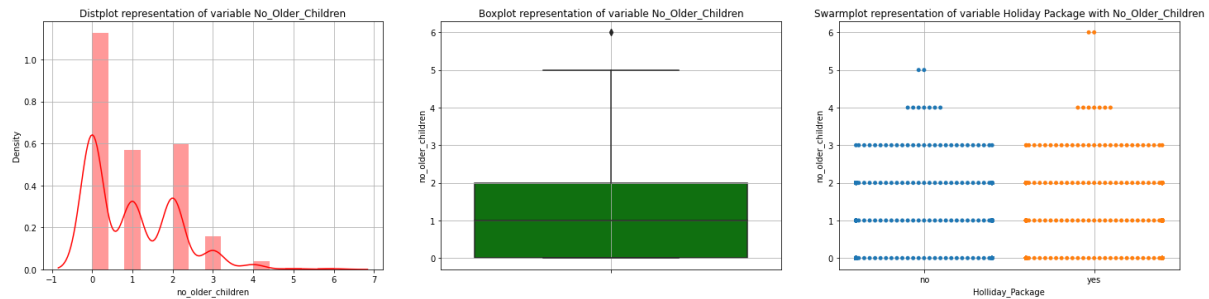


Figure 2.1e: Screenshot of the distplot and boxplot distribution for the variable 'no_older_children'.

Insights:

1. The distplot shows the irregular distribution since the data points are very few.
2. Almost zero outlier.
3. Employee having older children more than 3 loose the interest to opt the holiday package.

Bivariate analysis is appropriate to examine the distribution of the categorical variables (Figure 2.2).

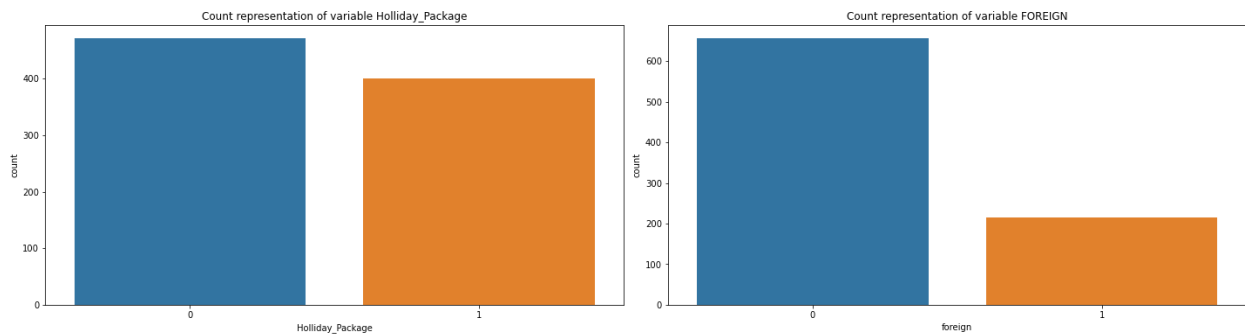


Figure 2.2: The count of the employees opted holiday package and select foreign trip.

Insights:

1. 46 percentage of employee opt the holiday package while 54 percentage do not.
2. 25 percentage of employee opt the foreign trip while 75 percentage of employee do not opt foreign trip.

Multivariate analysis is a Statistical procedure for analysis of data involving more than one type of measurement or observation. It may also mean solving problems where more than one dependent variable is analyzed simultaneously with other variables. For multivariate analysis pairplot (Figure 2.3) and heatmap (Figure 2.4) are two useful methods applied here.

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

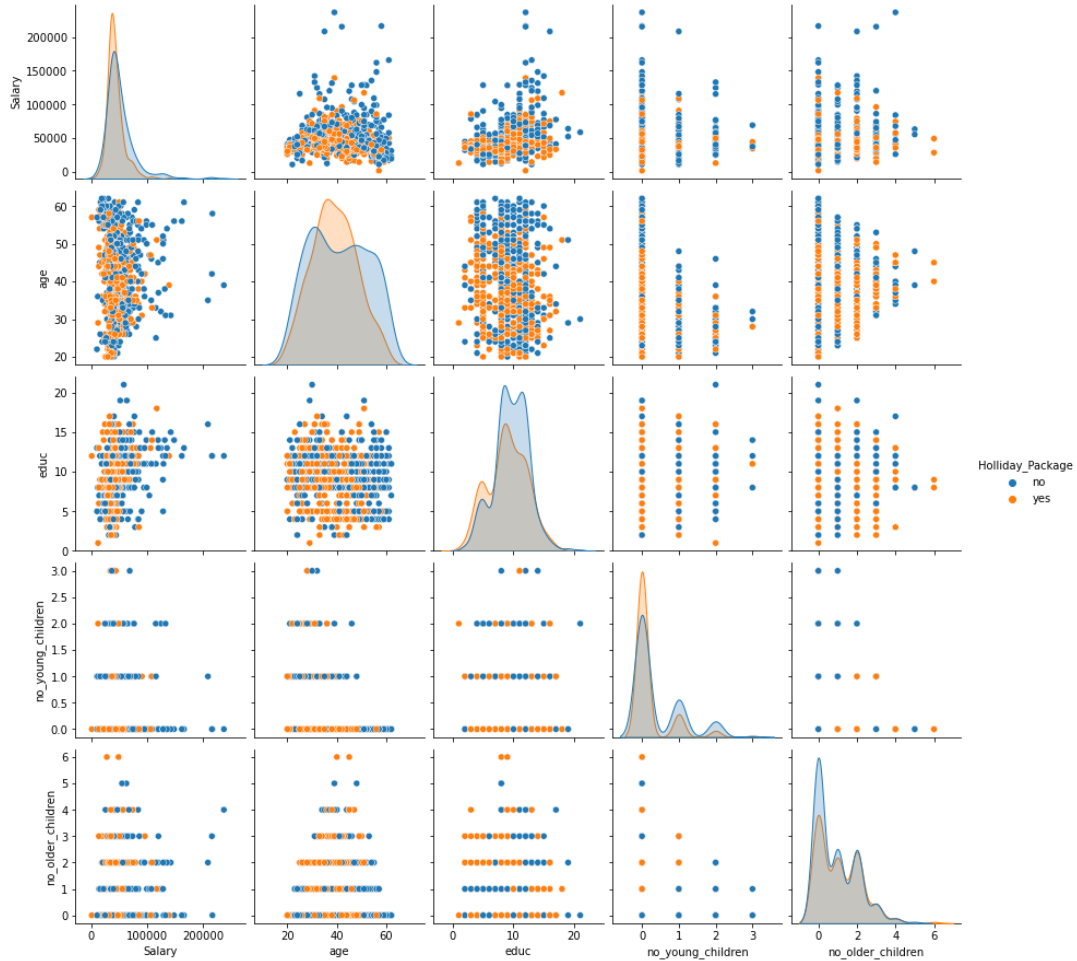


Figure 2.3: The pairplot for the given Dataset. It shows the interrelationship of all the variables available.

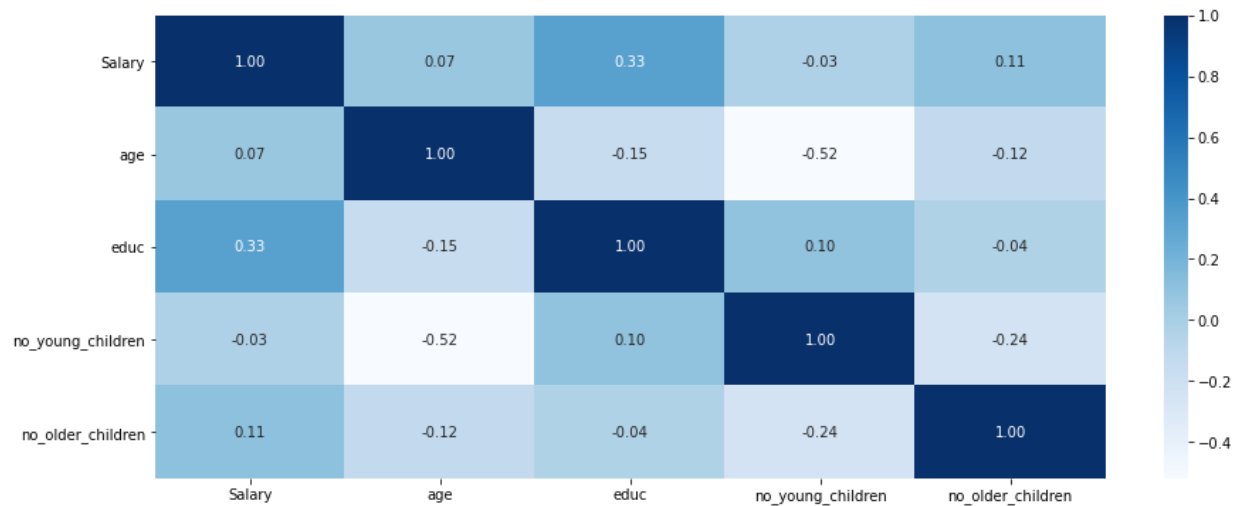


Figure 2.4: Heatmap for the given dataset, shows the no collinearity of all the variables.

No multi collinearity found in the dataset.

Outliers increase the variability in data, which decreases statistical power. LDA works based on numerical computation treating outliers (Figure 2.6) will help perform the model better.

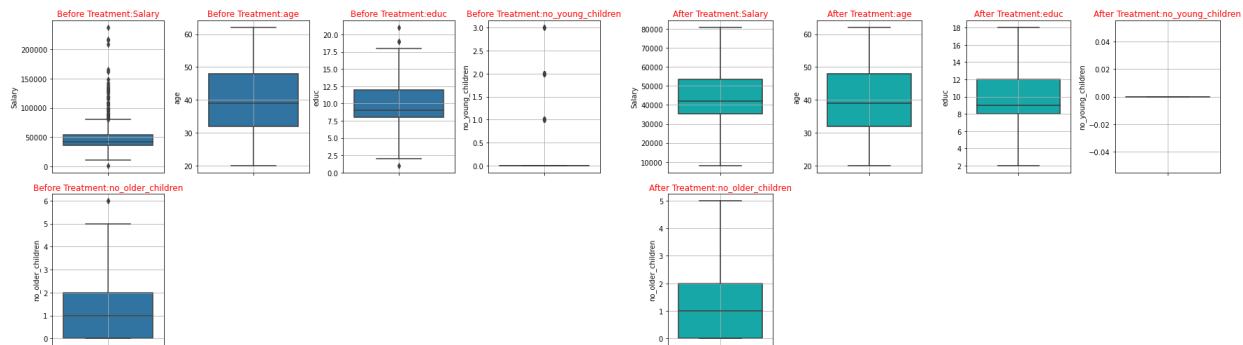


Figure 2.6: Variables in the dataset before and after outlier treatment.

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).

The dataset consists of two categorical variables, 'Holiday package', and 'foreign' with two categories. To improve predictions those categories are needed to be configured into numerical data. For this reason one hot encoding is used.

In this regard, dummy variable is used to segregate between different sub-groups of the data for better regression analysis. The following code is used to get the dummy variable:

```
data2 = pd.get_dummies(df2, columns=['Holliday_Package', 'foreign'], drop_first = True)
```

Here, *data2* is the variable to which the dummy variable is assigned and Table 2.4 illustrates the updated dataset.

Table 2.4: Screenshot of the table formed with the first 5 rows of the dataset after one-hot encoding.

	Salary	age	educ	no_young_children	no_older_children	Holliday_Package_yes	foreign_yes
0	48412.0	30.0	8.0	0.0	1.0	0	0
1	37207.0	45.0	8.0	0.0	1.0	1	0
2	58022.0	46.0	9.0	0.0	0.0	0	0
3	66503.0	31.0	11.0	0.0	0.0	0	0
4	66734.0	44.0	12.0	0.0	2.0	0	0

The index of the newly formed encoded dataset are like: `Index(['Salary', 'age', 'educ', 'no_young_children', 'no_older_children', 'Holliday_Package_yes', 'foreign_yes'], dtype='object')`

To split the data into train and test set in 70:30 ratio, the following commands are suitable:

```
XX = data2.drop('Holliday_Package_yes', axis=1)
```



```
yy = data2['Holliday_Package_yes']
```

For training and testing purpose the dataset can be split into train and test data in the ratio 70:30 following the '`Xx_train, Xx_test, yy_train, yy_test = train_test_split(XX, yy, test_size=0.30 , random_state=1)`' command.

The counts for 0 and 1 for the y value both in train and test set are given in Table 2.5. The value obtained from the table results the perfect split.

Table 2.5: Split of the dataset and trace the performance of the split.

value	yy_train	yy_test
0	0.534426	0.553435
1	0.465574	0.446565

Forming a logistic regression model to get the optimal solution of the business problem, grid search method is used to find the parameters required for the same. The grid assigns the variable like `{'penalty': ['l1','l2','none'], 'solver':['lbfgs','liblinear'],'tol':[0.0001,0.000001]}`. The best parameters obtained as: `{'penalty': 'l1', 'solver': 'liblinear', 'tol': 1e-06}` and the best estimator for fitting the logistic regression model is, `LogisticRegression(max_iter=10000, n_jobs=2, penalty='l1', solver='liblinear', tol=1e-06)`. The grid search method gives, 'liblinear' solver which is suitable for small datasets.

`LinearDiscriminantAnalysis()` function is used to build the LDA model as given in the commands,

```
clf = LinearDiscriminantAnalysis()
```

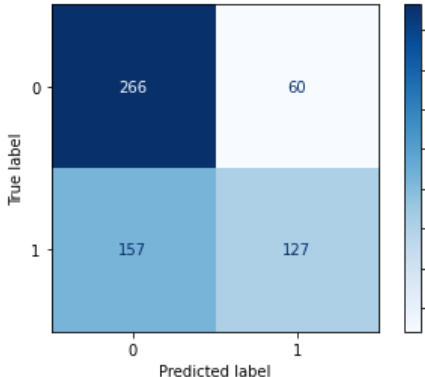
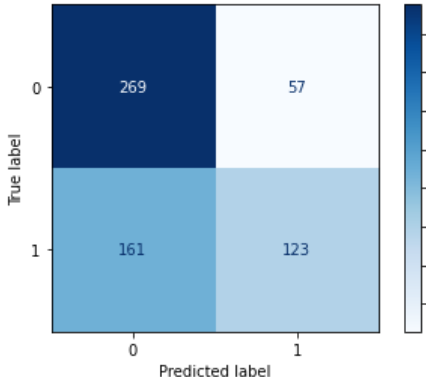
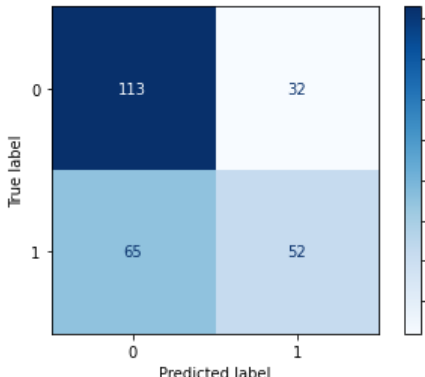
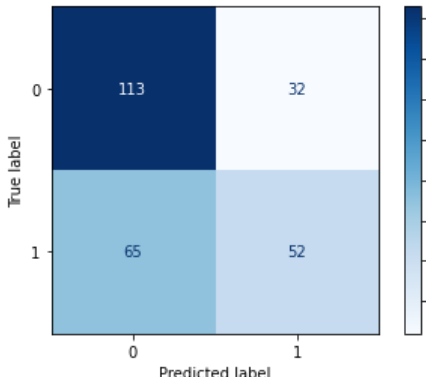
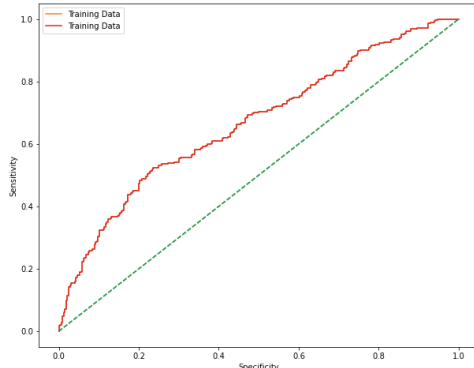
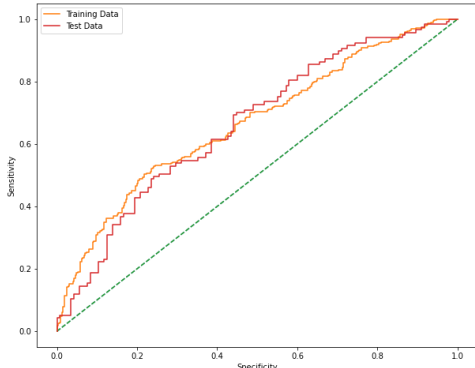
```
model=clf.fit(Xx_train,yy_train)
```

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.

The performance metrics of the two models are illustrated in Table 2.6 based on features like accuracy, confusion matrix, ROC plot, area under curve value and classification report.

Table 2.6: Performance of Predictions on Train & Test sets.

Model		Logistic Regression Analysis	Linear Discriminant Analysis
Feature			
Accuracy	Train data	0.644	0.642
	Test data	0.629	0.629
AUC Score	Train data	0.666	0.667
	Test data	0.662	0.662

Model		Logistic Regression Analysis					Linear Discriminant Analysis																																																																
Feature																																																																							
Classification Report	Train data	Training set: <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.63</td><td>0.82</td><td>0.71</td><td>326</td></tr><tr><td>1</td><td>0.68</td><td>0.45</td><td>0.54</td><td>284</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.64</td><td>610</td></tr><tr><td>macro avg</td><td>0.65</td><td>0.63</td><td>0.62</td><td>610</td></tr><tr><td>weighted avg</td><td>0.65</td><td>0.64</td><td>0.63</td><td>610</td></tr></tbody></table>						precision	recall	f1-score	support	0	0.63	0.82	0.71	326	1	0.68	0.45	0.54	284	accuracy			0.64	610	macro avg	0.65	0.63	0.62	610	weighted avg	0.65	0.64	0.63	610	Training set: <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.63</td><td>0.83</td><td>0.71</td><td>326</td></tr><tr><td>1</td><td>0.68</td><td>0.43</td><td>0.53</td><td>284</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.64</td><td>610</td></tr><tr><td>macro avg</td><td>0.65</td><td>0.63</td><td>0.62</td><td>610</td></tr><tr><td>weighted avg</td><td>0.65</td><td>0.64</td><td>0.63</td><td>610</td></tr></tbody></table>						precision	recall	f1-score	support	0	0.63	0.83	0.71	326	1	0.68	0.43	0.53	284	accuracy			0.64	610	macro avg	0.65	0.63	0.62	610	weighted avg	0.65	0.64	0.63	610
		precision	recall	f1-score	support																																																																		
0	0.63	0.82	0.71	326																																																																			
1	0.68	0.45	0.54	284																																																																			
accuracy			0.64	610																																																																			
macro avg	0.65	0.63	0.62	610																																																																			
weighted avg	0.65	0.64	0.63	610																																																																			
	precision	recall	f1-score	support																																																																			
0	0.63	0.83	0.71	326																																																																			
1	0.68	0.43	0.53	284																																																																			
accuracy			0.64	610																																																																			
macro avg	0.65	0.63	0.62	610																																																																			
weighted avg	0.65	0.64	0.63	610																																																																			
Test data	Test set: <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.63</td><td>0.78</td><td>0.70</td><td>145</td></tr><tr><td>1</td><td>0.62</td><td>0.44</td><td>0.52</td><td>117</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.63</td><td>262</td></tr><tr><td>macro avg</td><td>0.63</td><td>0.61</td><td>0.61</td><td>262</td></tr><tr><td>weighted avg</td><td>0.63</td><td>0.63</td><td>0.62</td><td>262</td></tr></tbody></table>						precision	recall	f1-score	support	0	0.63	0.78	0.70	145	1	0.62	0.44	0.52	117	accuracy			0.63	262	macro avg	0.63	0.61	0.61	262	weighted avg	0.63	0.63	0.62	262	Test set: <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.63</td><td>0.78</td><td>0.70</td><td>145</td></tr><tr><td>1</td><td>0.62</td><td>0.44</td><td>0.52</td><td>117</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.63</td><td>262</td></tr><tr><td>macro avg</td><td>0.63</td><td>0.61</td><td>0.61</td><td>262</td></tr><tr><td>weighted avg</td><td>0.63</td><td>0.63</td><td>0.62</td><td>262</td></tr></tbody></table>						precision	recall	f1-score	support	0	0.63	0.78	0.70	145	1	0.62	0.44	0.52	117	accuracy			0.63	262	macro avg	0.63	0.61	0.61	262	weighted avg	0.63	0.63	0.62	262	
	precision	recall	f1-score	support																																																																			
0	0.63	0.78	0.70	145																																																																			
1	0.62	0.44	0.52	117																																																																			
accuracy			0.63	262																																																																			
macro avg	0.63	0.61	0.61	262																																																																			
weighted avg	0.63	0.63	0.62	262																																																																			
	precision	recall	f1-score	support																																																																			
0	0.63	0.78	0.70	145																																																																			
1	0.62	0.44	0.52	117																																																																			
accuracy			0.63	262																																																																			
macro avg	0.63	0.61	0.61	262																																																																			
weighted avg	0.63	0.63	0.62	262																																																																			
Confusion Matrix	Train data																																																																						
	Test data																																																																						
ROC																																																																							

The confusion matrix, precision, recall, and F1 score gives better intuition of prediction results as compared to accuracy. It is actually the combination of four possibilities. In this present context, those possibilities are,

- (i) Employee opts the holiday package and the agency also predict the same: TP
- (ii) Employee opts the holiday package and the agency does not predict the same: FN
- (iii) Employee does not opt the holiday package and the agency does not predict the same: FP
- (iv) Employee does not opt the holiday package and the agency also predict the same: TN

The number of correct and incorrect predictions are summarized with count values and broken down by each class. The summary of the confusion matrix (Table 2.7) is reflected in classification report as shown below.

Table 2.7: Confusion matrix, precision, recall, and F1 score provides better insights

	LR Train	LR Test	LDA Train	LDA Test
Accuracy	0.64	0.63	0.64	0.63
Recall	0.45	0.44	0.43	0.44
Precision	0.68	0.62	0.68	0.62
F1 Score	0.54	0.52	0.53	0.52

In generally for any stabilized working model, the values of TPR (true positive rate) and TNR (true negative rate) should be high, and FPR (false positive rate) and FNR (false negative rate) should be as low as possible.

The recall or sensitivity term is associated with the true positive and false negative value (type II error, which is fatal). In present case, both LA and LDA show almost same recall value for both training and test dataset.

Precision is another important term which deals with true positive and false positive (type I error). Here also both LA and LDA models offer almost same precision value for both training and test dataset.

The F1 score deals with both false positive and false negatives into account. The highest possible value of an F1-score is 1.0, indicating perfect precision and recall, and the lowest possible value is 0, if either the precision or the recall is zero. Both LA and LDA model show almost same score for training and test dataset.

Analyzing both logistic regression and linear discriminant models, both results are same, but LDA works better when the target attribute is categorical type.

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.)

The objective of the problem is to predict whether or not an employee will choose to take a holiday package. Both logistic regression and linear discriminant analysis are used to address this problem. The results are the same in both cases.

In the EDA analysis, certain criteria clearly indicate that people over 50 are less likely to be interested in holiday packages. Thus, this is one of the reasons that older people don't opt for holiday packages. People between the ages of 30 and 50 tend to opt for holiday packages.

The employees over 50 to 60 are not taking the holiday package, whereas the employees under 50000 and 30 to 50 are taking it. Salary, age, and education are the most important factors in determining predictions.

Recommendations

1. Holiday packages over the age of 50 can be improved with religious destinations.
2. Vacation packages are available for people earning more than 150000.
3. Employees with more than one older child may be able to take advantage of holiday vacation packages.