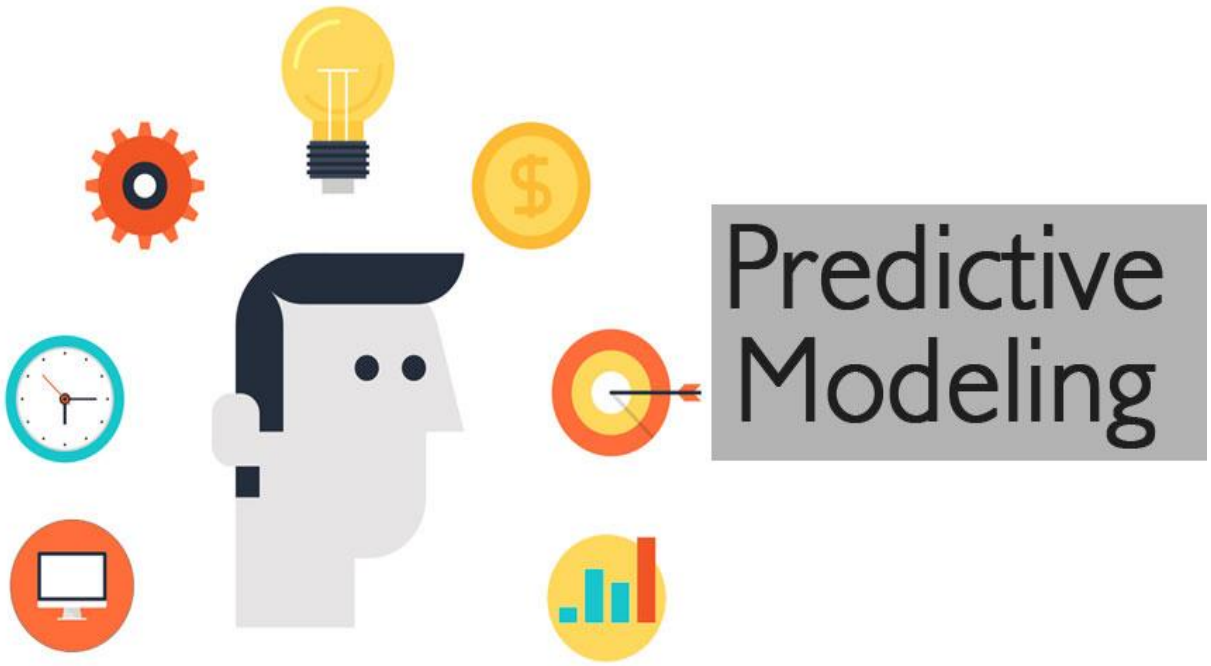
PARAKANDLA. MANISHA, PGP-DSBA ONLINE, 19-12-2021.



- Linear Regression
- Logistic Regression
- Linear Discriminant Analysis(LDA)

Contents

List of tables:

List of Figures:

Problem 1: Linear Regression

Executive Summary:

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.

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.

Dataset for Problem 1: cubic_zirconia.csv

INTRODUCTION:

The dataset consists of 26967 rows and 11 columns. In this problem we perform linear regression analysis for 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. To perform the analysis, we need to import all the required libraries and initial descriptive data analysis to be done.

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.

SUMMARY OF THE DATA:

	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
...	...	...	...	...	...	...	...	...	...	...	...
26962	26963	1.11	Premium	G	SI1	62.3	58.0	6.61	6.52	4.09	5408
26963	26964	0.33	Ideal	H	IF	61.9	55.0	4.44	4.42	2.74	1114
26964	26965	0.51	Premium	E	VS2	61.7	58.0	5.12	5.15	3.17	1656
26965	26966	0.27	Very Good	F	VVS2	61.8	56.0	4.19	4.20	2.60	682
26966	26967	1.25	Premium	J	SI1	62.0	58.0	6.90	6.88	4.27	5166

26967 rows × 11 columns

Table 1.1 Summary of gem stone data

→The dataset is having 26967 rows and 11 columns.

→I have renamed the Unnamed: 0 column as index and set the index column as index of data.

→The info function gives us the brief understanding about the features and its datatypes.

→The shape of the dataset is (26967, 10).

→ The feature depth has 697 null values, rest other features have no null values. Datatypes are correctly assigned that shows data doesn't have garbage values as well. The data consist

s of 11 features having 6 float datatype, 2 int datatype and 3 object datatype features.The d ata consists 26967 entries.2.58% null values are present in the data.

EXPLORATORY DATA ANALYSIS:

Description of the data:

The describe () method is used for **calculating some statistical data** like percentile, mean and std of the numerical values in the dataset. It analyses both numeric and object series and also the column sets of mixed data types.

	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

Table1.2 Description of the data

→Unique values of categorical variables:

```
CUT :  5
Fair           781
Good          2441
Very Good     6030
Premium       6899
Ideal        10816
Name: cut, dtype: int64


COLOR :  7
J    1443
I    2771
D    3344
H    4102
F    4729
E    4917
G    5661
Name: color, dtype: int64


CLARITY :  8
I1        365
IF        894
VVS1     1839
VVS2     2531
VS1      4093
SI2      4575
VS2      6099
SI1      6571
Name: clarity, dtype: int64
```

→ nunique () function **return number of unique elements in the object**. It returns a scalar value which is the count of all the unique values in the Index. By default, the NaN values are not included in the count.

```
carat     257
depth     169
table     112
x         531
y         526
z         356
price    8742
dtype: int64
```

UNIVARIATE AND BIVARIATE ANALYSIS:

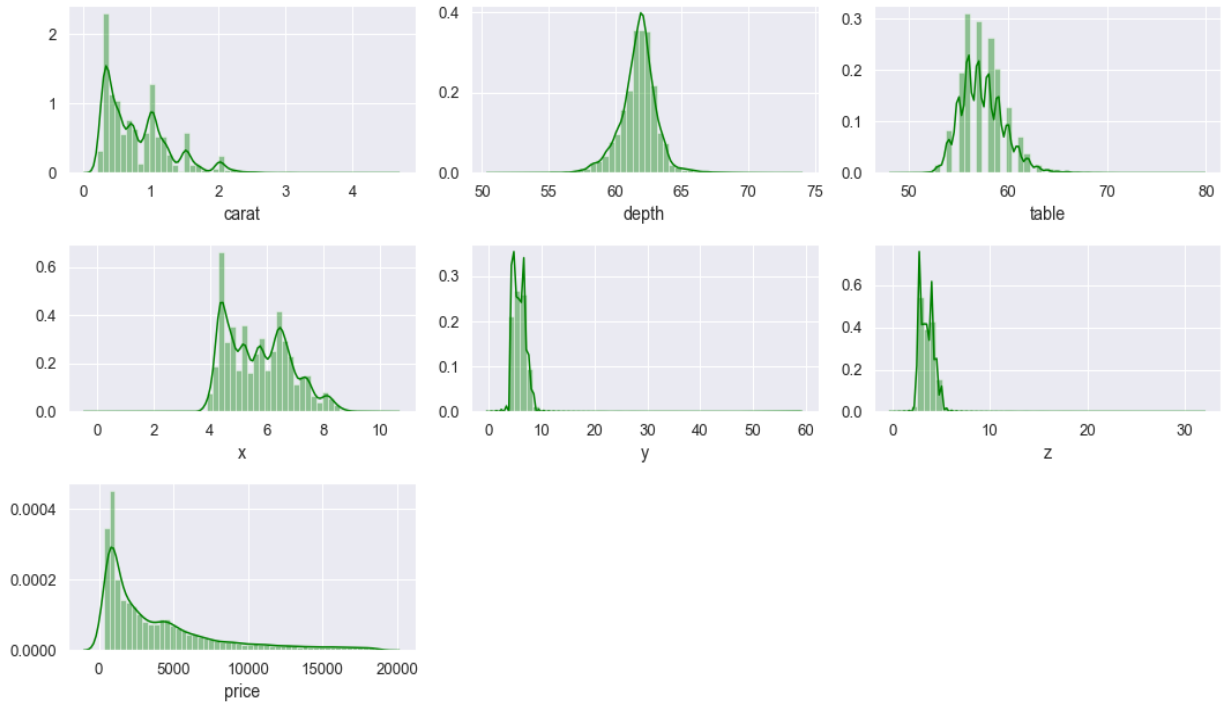


Fig1. Distribution Plot

There are multiple clusters evident in Carat, table, X, Y and Z as multiple peaks are present.

Most of the continuous features are normal but have multiple peaks while Y and Z seems to be not normal.

Only depth and Price don't have multiple clusters and are normally distributed and price is slightly right skewed.

Boxplot:

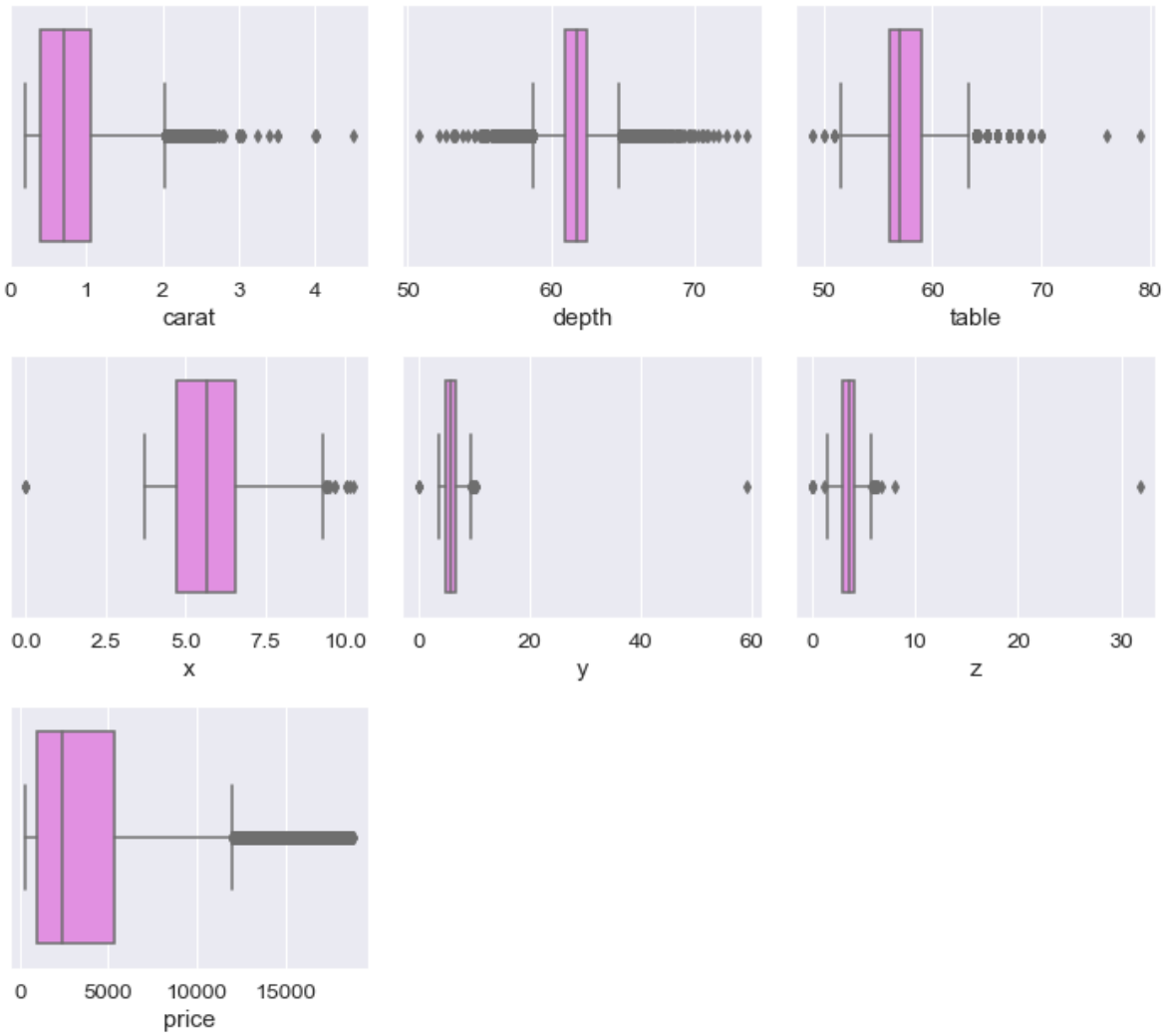


Fig2 Boxplot showing Outliers

All the continuous variables have outliers.

Since the outliers are continuous so they seem to be genuine thus treating them may distort the outcome of regression.

Skewness:

```
carat    1.116481
depth   -0.028618
table    0.765758
x        0.387986
y        3.850189
z        2.568257
price    1.618550
dtype: float64
```

Probplot:

Generates a probability plot of sample data against the quantiles of a specified theoretical distribution (the normal distribution by default). **probplot** optionally calculates a best-fit line for the data and plots the results using Matplotlib or a given plot function.

```
((array([-4.04913901, -3.83653887, -3.72035344, ...,  3.72035344,
          3.83653887,  4.04913901]),
  array([ 0.  ,  0.  ,  0.  , ...,  6.72,  8.06, 31.8 ])),
 (0.6878855999754362, 3.5380572551637193, 0.9544486377952857))
```

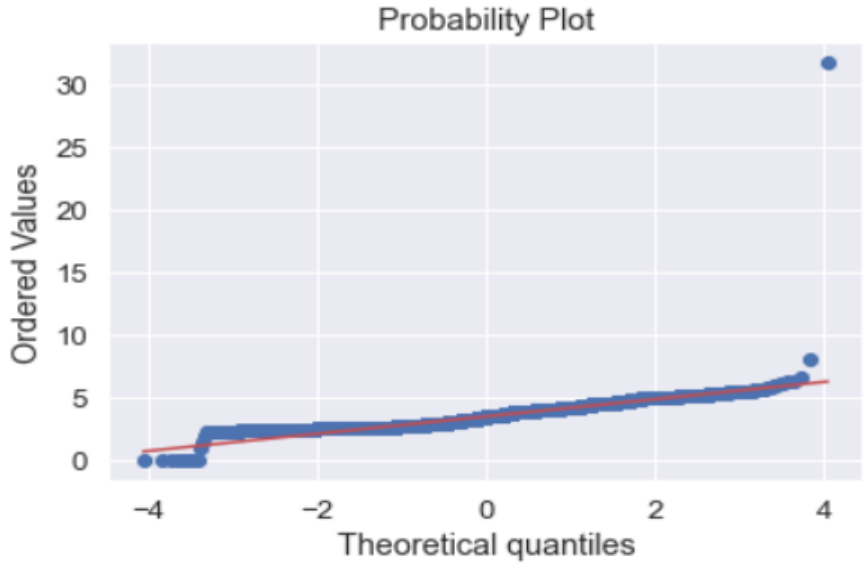


Fig3. Probability Plot

→Boxplot colour, cut, clarity Vs Price is as shown below:

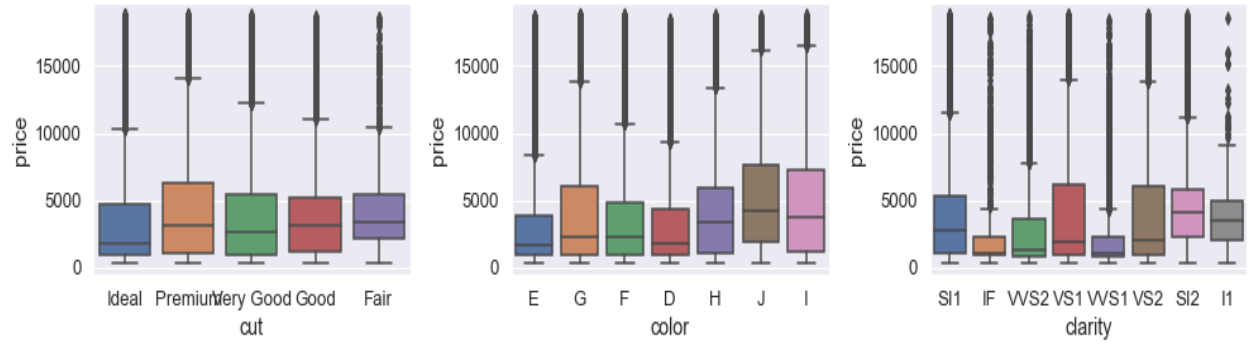


Fig4. Boxplot for colour, cut, clarity Vs Price

H, I, J coloured diamonds are costlier ones.

SI2 and I1 clarity level diamonds are expensive.

Multivariate Analysis:

Pairplot:

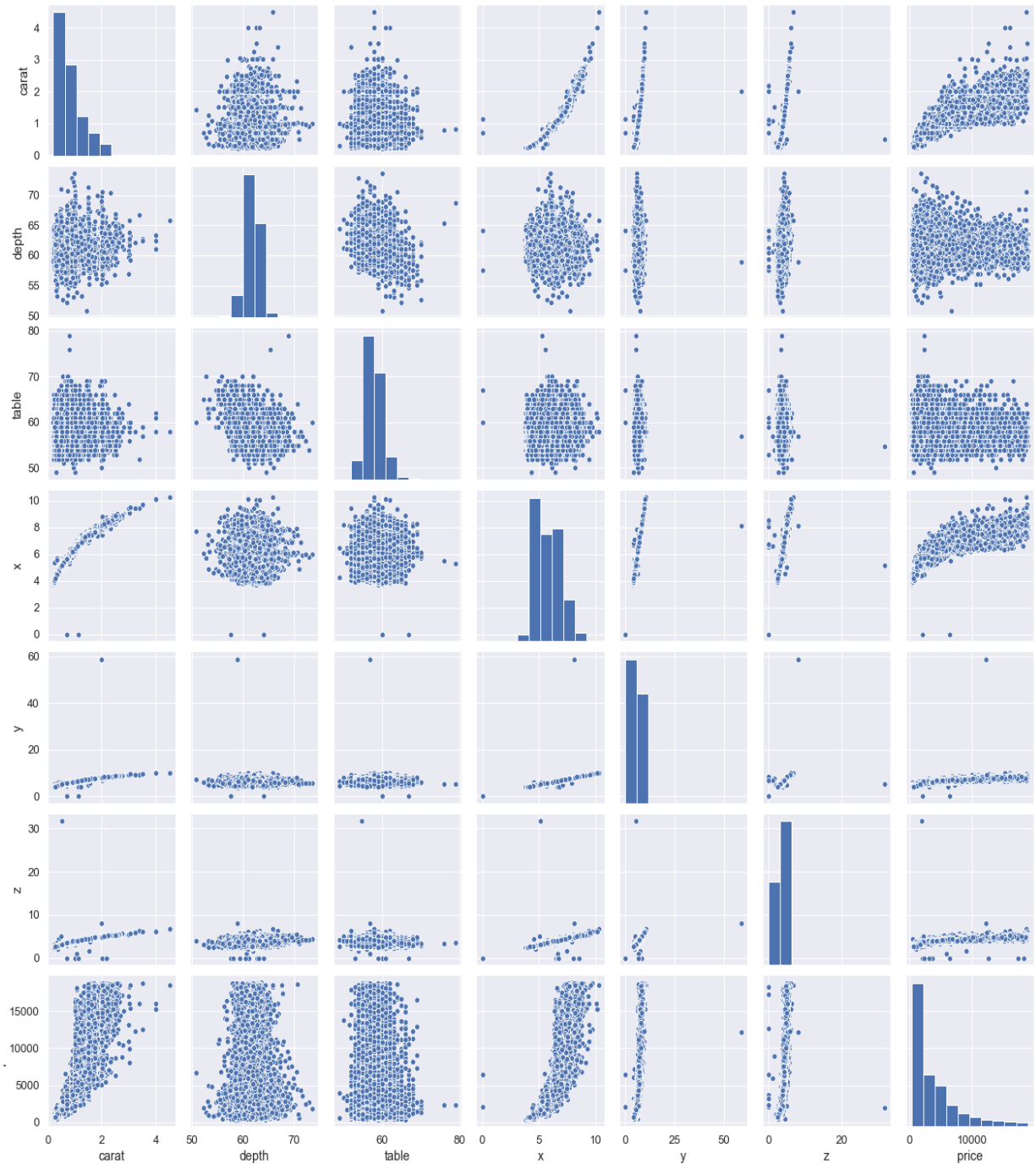


Fig5. Pairplot

Price has high correlation with carat, X, Y, Z.

X, Y, Z also have high correlation with each other as well.

It seems that Price is highly dependent over X, Y and Z

Heatmap:

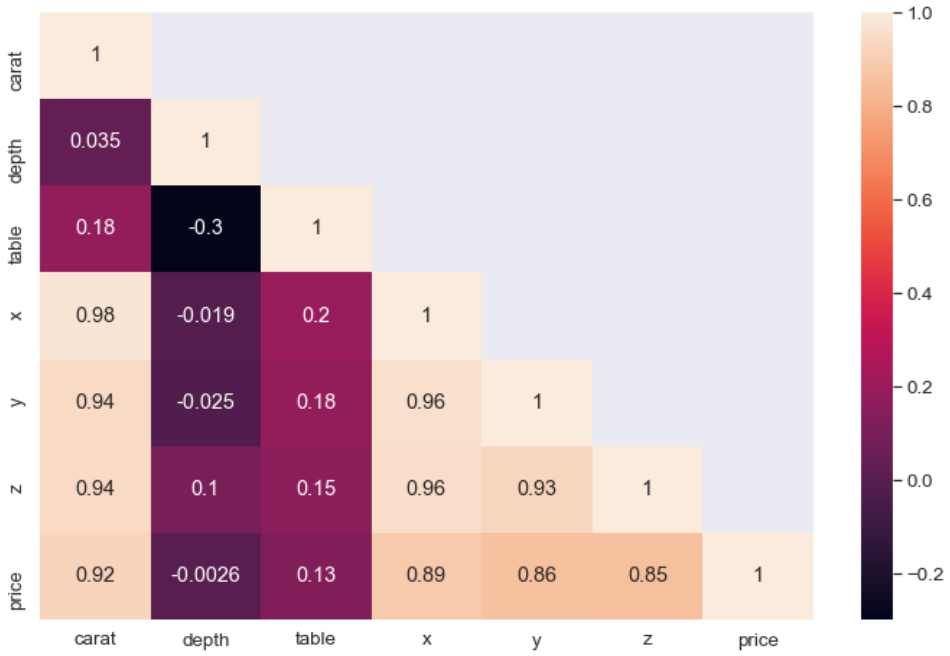


Fig6. Heatmap

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 ordinal variables and take actions accordingly. Explain why you are combining these sub levels with appropriate reasoning.

→There are null values present in the data, I have imputed the null values with median.

→Combining clarity category:

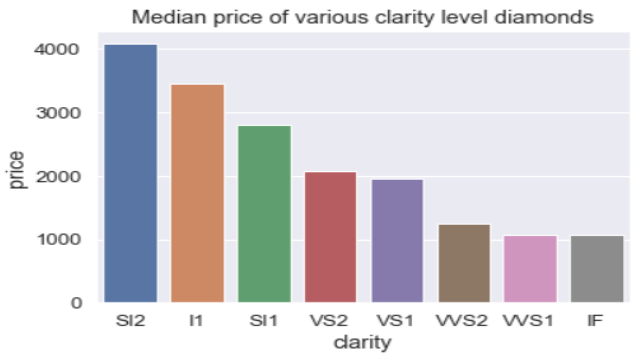


Fig7. Median price of clarity level diamonds

Here in median price of various clarity level diamonds, we can see SI2 is the most expensive vs IF is the least expensive one.

we can categorise the various levels of clarity in 3 categories low, medium and high based on median price 1000-2000, 2000-3000 and more than 3000 respectively.

IF, WS1, WS2- low clarity diamonds

VS1, VS2, and SI1 - medium clarity diamonds

I1, SI2- high clarity diamonds.

→Combining colour Category:

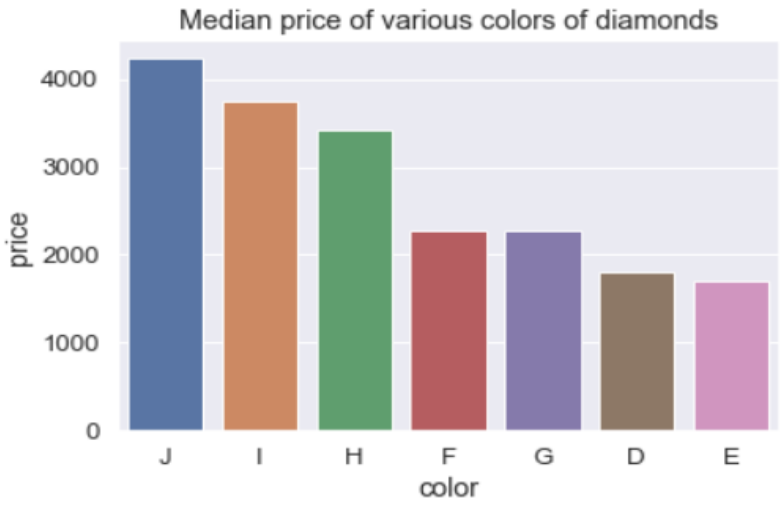


Fig8.Median price of various colours of diamonds

Here, in median price of various colours of diamonds we can see J is the most expensive while E is the least expensive one whereas the data dictionary confirms D being the worst colour for diamonds.

we can categorise the various colours in 4 categories low, medium, high and premium based on median price 1000-2000, 2000-3000, 3000-4000 and respectively.

D and E - low quality.

F and G - medium qulity color.

H and I - high quality color.

J - premium quality color.

→Combining cut category:

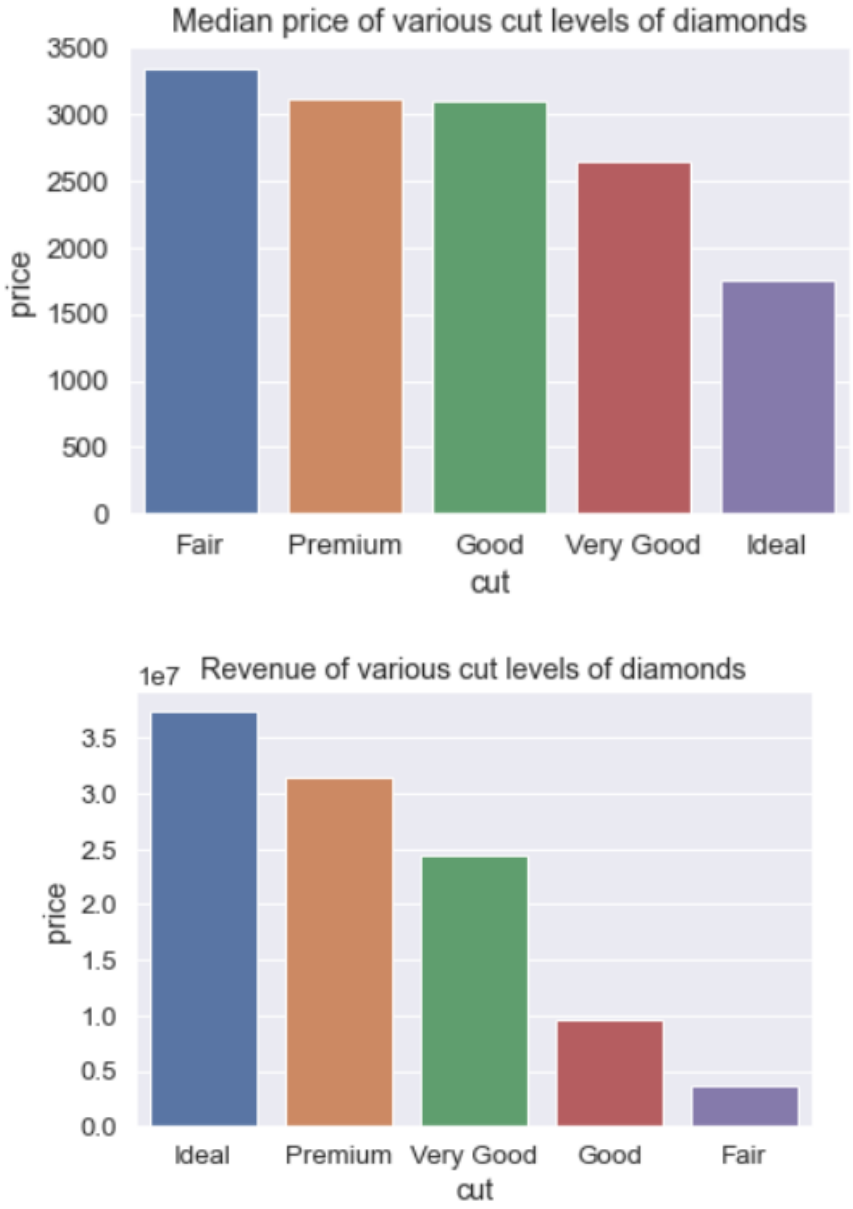


Fig9.Median price & Revenue of cut levels of diamonds

Here in median price of various cut levels of diamonds we can see Fair cut diamond is the most expensive while ideal cut is the least expensive one.

The data dictionary confirms Ideal being the best cut while the decreasing order for quality of cut is Ideal, Premium, Very Good, Good and Fair but the same is not confirmed by the data as it shows the decreasing order of cut as Fair, Premium, Good, Very Good and Ideal.

Though this order is not confirmed by the analysis but will follow as given by the business stake holder.

Classifying the data & encoding null values:

Clarity:

→ low_clarity for ['IF', 'VVS1', 'VVS2'].

→ medium_clarity for ['VS1', 'VS2', 'SI1'].

→ high_clarity for ['I1', 'SI2']

Color:

['E', 'D'] → low_quality

['G', 'F']→ medium_quality

['I', 'H'] →high_quality

J→ premium_quality

Cut:

['Fair', 'Good', 'Very Good'] → average_cut

['Premium', 'Ideal'] → precision_cut

We have combined the sublevels for effective classification of data for proceeding with the modelling process and also helps in encoding.

Sample showing the classification:

index	carat	cut	color	clarity	depth	table	x	y	z	price
1	0.30	precision_cut	low_quality	medium_clarity	62.1	58.0	4.27	4.29	2.66	499
2	0.33	precision_cut	medium_quality	low_clarity	60.8	58.0	4.42	4.46	2.70	984
3	0.90	average_cut	low_quality	low_clarity	62.2	60.0	6.04	6.12	3.78	6289
4	0.42	precision_cut	medium_quality	medium_clarity	61.6	56.0	4.82	4.80	2.96	1082
5	0.31	precision_cut	medium_quality	low_clarity	60.4	59.0	4.35	4.43	2.65	779
6	1.02	precision_cut	low_quality	medium_clarity	61.5	56.0	6.46	6.49	3.99	9502
7	1.01	average_cut	high_quality	medium_clarity	63.7	60.0	6.35	6.30	4.03	4836
8	0.50	precision_cut	low_quality	medium_clarity	61.5	62.0	5.09	5.06	3.12	1415
9	1.21	average_cut	high_quality	medium_clarity	63.8	64.0	6.72	6.63	4.26	5407
10	0.35	precision_cut	medium_quality	medium_clarity	60.5	57.0	4.52	4.60	2.76	706

Table1.3 Sample showing the classification

After this, we have done encoding the null values. Checking the data once with info () .

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 26967 entries, 1 to 26967
Data columns (total 10 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   carat    26967 non-null  float64
 1   cut      26967 non-null  int64
 2   color    26967 non-null  int64
 3   clarity  26967 non-null  int64
 4   depth    26270 non-null  float64
 5   table    26967 non-null  float64
 6   x        26967 non-null  float64
 7   y        26967 non-null  float64
 8   z        26967 non-null  float64
 9   price    26967 non-null  int64
dtypes: float64(6), int64(4)
memory usage: 3.3 MB
```

We can observe that there are null values present in the depth column. I have imputed the null values with median.

→Checking the Zeroes in Predictor variable:

Series ([], Name: carat, dtype: float64)

index
3 0
7 0
9 0
13 0
15 0
..
26955 0
26956 0
26960 0
26961 0
26966 0
Name: cut, Length: 9252, dtype: int64

index
1 0
3 0
6 0
8 0
11 0
..
26957 0
26959 0
26960 0

```
26961    0
26965    0
Name: color, Length: 8261, dtype: int64

index
2       0
3       0
5       0
21      0
32      0
       ..
26952    0
26955    0
26962    0
26964    0
26966    0
Name: clarity, Length: 5264, dtype: int64

Series ([], Name: depth, dtype: float64)

Series ([], Name: table, dtype: float64)

index
5822     0.0
6216     0.0
17507    0.0
Name: x, dtype: float64

index
5822     0.0
6216     0.0
17507    0.0
Name: y, dtype: float64

index
5822     0.0
6035     0.0
6216     0.0
10828    0.0
12499    0.0
12690    0.0
17507    0.0
18195    0.0
23759    0.0
Name: z, dtype: float64

Series ([], Name: price, dtype: int64)
```

For x==0:

index	carat	cut	color	clarity	depth	table	x	y	z	price
5822	0.71	0	1	2	64.1	60.0	0.0	0.0	0.0	2130
6216	0.71	0	1	2	64.1	60.0	0.0	0.0	0.0	2130
17507	1.14	0	1	1	57.5	67.0	0.0	0.0	0.0	6381

For y==0:

index	carat	cut	color	clarity	depth	table	x	y	z	price
5822	0.71	0	1	2	64.1	60.0	0.0	0.0	0.0	2130
6216	0.71	0	1	2	64.1	60.0	0.0	0.0	0.0	2130
17507	1.14	0	1	1	57.5	67.0	0.0	0.0	0.0	6381

For z==0: (found the zeroes in the predictor variable)

index	carat	cut	color	clarity	depth	table	x	y	z	price
5822	0.71	0	1	2	64.1	60.0	0.00	0.00	0.0	2130
6035	2.02	1	2	1	62.7	53.0	8.02	7.95	0.0	18207
6216	0.71	0	1	2	64.1	60.0	0.00	0.00	0.0	2130
10828	2.20	1	2	1	61.2	59.0	8.42	8.37	0.0	17265
12499	2.18	1	2	2	59.4	61.0	8.49	8.45	0.0	12631
12690	1.10	1	1	2	63.0	59.0	6.50	6.47	0.0	3696
17507	1.14	0	1	1	57.5	67.0	0.00	0.00	0.0	6381
18195	1.01	1	2	2	58.1	59.0	6.66	6.60	0.0	3167
23759	1.12	1	1	2	60.4	59.0	6.71	6.67	0.0	2383

we can see that there are 9 rows having 0 in x, y, z columns, it is sure that these are garbage value which we should remove.

Now, we have 26958 rows and 10 columns.

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.

Linear Regression:

The term "regression" generally refers to predicting a real number. The term "linear" in the name “linear regression” refers to the fact that the method models data with linear combination of the explanatory variables. A linear combination is an expression where one or more variables are scaled by a constant factor and added together.
In the case of simplest linear regression with a single explanatory variable,
the linear combination used in linear regression can be expressed as:
Dependent variable value = (weight ∗independent variable) + constant.
It is the straight line in the scatter plot of the variables

I have already encoded the data, proceeding with the modelling process.

→Split the data into train and test (70:30).

→Shape of train & test data ((18870, 9), (8088, 9)).

→Shape of train_labels and test_labels ((18870, 1), (8088, 1)).

→Fit the Linear regression model with the function LinearRegression ().

→ R-squared value for training data: 0.9063616902769321

→ R-squared value of testing data: 0.8985176726004975

→ RMSE value for test data: 1308.808548059965

→ RMSE value for train data: 1219.7322693531626

→ Intercept – array ([14449.49151293])

Coefficient of relation - Pearson's coefficient p(x,y) = Cov(x,y) / (stnd Dev (x) X stnd Dev (y))

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2}\sqrt{\sum(y_i - \bar{y})^2}}$$

	Coeff
carat	11188.770183
cut	160.599585
color	-588.092345
clarity	-1229.145524
depth	-132.805692
table	-58.465831
x	-1064.761268
y	14.445782
z	-37.764764

Ols_model:

Split the data into train & test (70:30).

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

y and z variables seem to have insignification for determining the price as p value is very high for the alpha level 0.05% so we can conclude that we fail to reject null hypothesis that says y and z have no relation with price.

Thus, we can remove y and z variables and make new model and check the validations and various parameters.

Overall summary of the model:

Model:	OLS	Adj. R-squared:	0.906
Dependent Variable:	price	AIC:	321765.7729
Date:	2021-12-19 17:29	BIC:	321844.2262
No. Observations:	18870	Log-Likelihood:	-1.6087e+05
Df Model:	9	F-statistic:	2.028e+04
Df Residuals:	18860	Prob (F-statistic):	0.00
R-squared:	0.906	Scale:	1.4885e+06

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	14449.4915	660.3446	21.8817	0.0000	13155.1569	15743.8262
carat	11188.7702	93.0459	120.2500	0.0000	11006.3918	11371.1486
cut	160.5996	20.3344	7.8979	0.0000	120.7423	200.4568
color	-588.0923	10.6485	-55.2276	0.0000	-608.9644	-567.2203
clarity	-1229.1455	15.7531	-78.0259	0.0000	-1260.0229	-1198.2681
depth	-132.8057	7.7006	-17.2462	0.0000	-147.8995	-117.7119
table	-58.4658	4.5423	-12.8713	0.0000	-67.3692	-49.5625
x	-1064.7613	50.5904	-21.0467	0.0000	-1163.9229	-965.5996
y	14.4458	23.9131	0.6041	0.5458	-32.4260	61.3176
z	-37.7648	41.9097	-0.9011	0.3675	-119.9115	44.3819

Omnibus:	4294.619	Durbin-Watson:	1.985
Prob(Omnibus):	0.000	Jarque-Bera (JB):	226564.551
Skew:	0.058	Prob(JB):	0.000
Kurtosis:	19.975	Condition No.:	6322

```
R-squared value of ols_model: 0.9063616902769321

R-Squared adj value of ols_model: 0.906317006035813

RMSE for train: 1219.7322693531626

RMSE for test: 1308.8085480599657
```

Fig10. Predicted Vs True price for Training

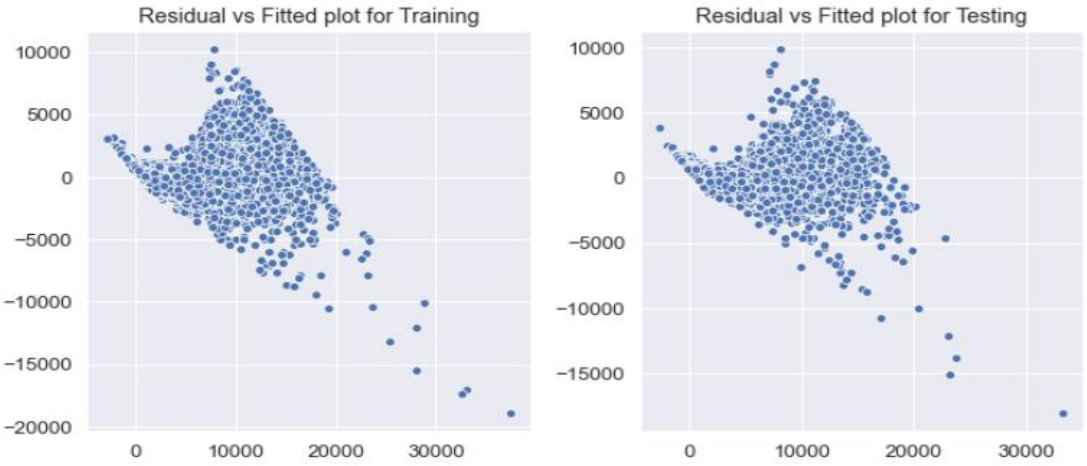


Fig 11. Residual vs fitted plot for training and testing

	residual	y_train_pred_mod2	y_train_true
residual	1.000000e+00	2.399818e-16	0.306004
y_train_pred_mod2	2.399818e-16	1.000000e+00	0.952030
y_train_true	3.060038e-01	9.520303e-01	1.000000

```
index
7599      2790.822838
8883       246.246691
22764     1880.600033
6644      1150.149597
18702    10916.208747
             ...
10960     4348.959030
17296     2856.232494
5193     14414.768401
12177     5735.268906
236       7986.575374
Length: 18870, dtype: float64
```

Model3:

formula2 = 'price~ carat + cut + color + clarity + depth + table + x'

summary of the model:

Model:	OLS	Adj. R-squared:	0.906
Dependent Variable:	price	AIC:	321762.8329
Date:	2021-12-19 18:42	BIC:	321825.5955
No. Observations:	18870	Log-Likelihood:	-1.6087e+05
Df Model:	7	F-statistic:	2.608e+04
Df Residuals:	18862	Prob (F-statistic):	0.00
R-squared:	0.906	Scale:	1.4885e+06

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	14589.7742	645.4241	22.6049	0.0000	13324.6850	15854.8633
carat	11188.9108	92.9930	120.3199	0.0000	11006.6362	11371.1854
cut	160.7643	20.3027	7.9184	0.0000	120.9692	200.5594
color	-587.9987	10.6478	-55.2225	0.0000	-608.8693	-567.1280
clarity	-1229.1348	15.7526	-78.0274	0.0000	-1260.0114	-1198.2583
depth	-135.1039	7.3254	-18.4433	0.0000	-149.4623	-120.7455
table	-58.4440	4.5358	-12.8850	0.0000	-67.3346	-49.5534
x	-1073.6261	39.4144	-27.2394	0.0000	-1150.8819	-996.3703

Omnibus:	4295.559	Durbin-Watson:	1.985
Prob(Omnibus):	0.000	Jarque-Bera (JB):	226852.353
Skew:	0.057	Prob(JB):	0.000
Kurtosis:	19.986	Condition No.:	6159

```
RMSE for train: 1219.7665290022803

RMSE for test: 1308.884412435523
```

Correlation:

	y_train_pred_mod3	y_train_true	residual
y_train_pred_mod3	1.000000e+00	0.952028	6.017934e-16
y_train_true	9.520275e-01	1.000000	-3.060124e-01
residual	6.017934e-16	-0.306012	1.000000e+00

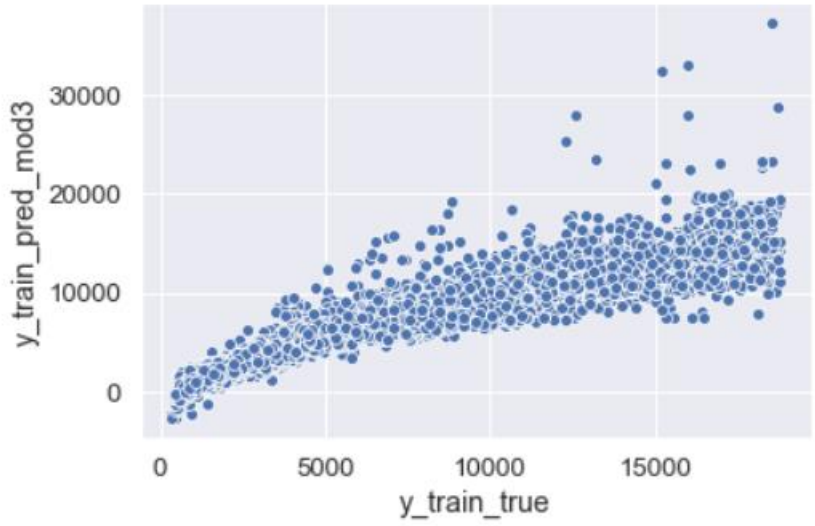


Fig12. Y_train_pred vs y_train_true

Ridge Model:

→Split the data into train and test (70:30)

```
Ridge(alpha=10, max_iter=100, tol=0.0001)
```

Coeff:

	coeff
carat	10556.087958
cut	160.984541
color	-579.808511
clarity	-1232.858644
depth	-121.059094
table	-57.601171
x	-805.095273
y	16.150455
z	-38.386730

```
Model train score: 0.9061319024436579

Model test score: 0.8982318891737651

RMSE train: 1221.2279600363613

RMSE test: 1310.6501141820684
```

From the analysis, I consider OLS model as the best fit model.

```
price = (14449.49) * Intercept + (11188.77) * carat + (160.6) * cut + (-588.09)
* color + (-1229.15) * clarity + (-132.81) * depth + (-58.47) * table + (-1064.
76) * x + (14.45) * y + (-37.76) * z +
```

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.

We had a business problem to predict the price of the stone and provide insights for the company on the profits on different prize slots. From the EDA analysis we could understand the cut, ideal cut had number profits to the company. The colours H, I, J have bought profits for the company. In clarity if we could see there were no flawless stones and they were no profits coming from I1, I2, I3 stones. The ideal, premium and very good types of cut were bringing profits where as fair and good are not bringing profits.

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

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

For better accuracy dropping depth column in iteration for better results.

Recommendations

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

Problem 2: Logistic Regression and LDA

EXECUTIVE SUMMARY:

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.

Dataset for Problem 2: Holiday_Package.csv

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

INTRODUCTION:

The dataset is having 872 rows and 8 columns. We need to perform Logistic regression and Linear discriminant analysis (LDA) for predicting whether an employee will opt for the holiday package or not on the basis of the information given in dataset. Before, performing the model we need to import all the necessary libraries and perform the exploratory data analysis.

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.

→I have dropped the Unnamed:0 column because it is not necessary.

→The shape of the dataset is (872,8)

→There are no null values present in the dataset.

→We have integer and object datatypes.

→We can use info () function for brief understanding of the features in the dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 872 entries, 0 to 871
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Holliday_Package   872 non-null    object
 1   Salary             872 non-null    int64
 2   age                872 non-null    int64
 3   educ               872 non-null    int64
 4   no_young_children  872 non-null    int64
 5   no_older_children  872 non-null    int64
 6   foreign            872 non-null    object
dtypes: int64(5), object(2)
memory usage: 47.8+ KB
```

SAMPLE OF DATASET:

	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
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
...	...	...	...	...	...	...	...	...
867	868	no	40030	24	4	2	1	yes
868	869	yes	32137	48	8	0	0	yes
869	870	no	25178	24	6	2	0	yes
870	871	yes	55958	41	10	0	1	yes
871	872	no	74659	51	10	0	0	yes

872 rows × 8 columns

Table 1.4 Sample of Holiday data

HEAD OF THE DATA:

	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

Table 1.5 Head of the data

TAIL OF THE DATA:

	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
867	no	40030	24	4	2	1	yes
868	yes	32137	48	8	0	0	yes
869	no	25178	24	6	2	0	yes
870	yes	55958	41	10	0	1	yes
871	no	74659	51	10	0	0	yes

Table 1.6 Tail of data

Description of data:

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

Table 1.7 Description of Holiday data

Unique values of categorical variables:

```
HOLLIDAY_PACKAGE :  2
yes    401
no     471
Name: Holliday_Package, dtype: int64


FOREIGN :  2
yes    216
no     656
Name: foreign, dtype: int64
```

We have integer and continuous data,

Holiday package is our target variable

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

Percentage of target variable:

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

Univariate & Bivariate Analysis:

Univariate analysis is **the simplest form of analysing data**. It takes data, summarizes that data and finds patterns in the data.

Like other forms of statistics, it can be inferential or descriptive. It helps us to understand the distribution of data in the dataset.

With univariate analysis we can find patterns and summarize the data.

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 analysed simultaneously with other variables.

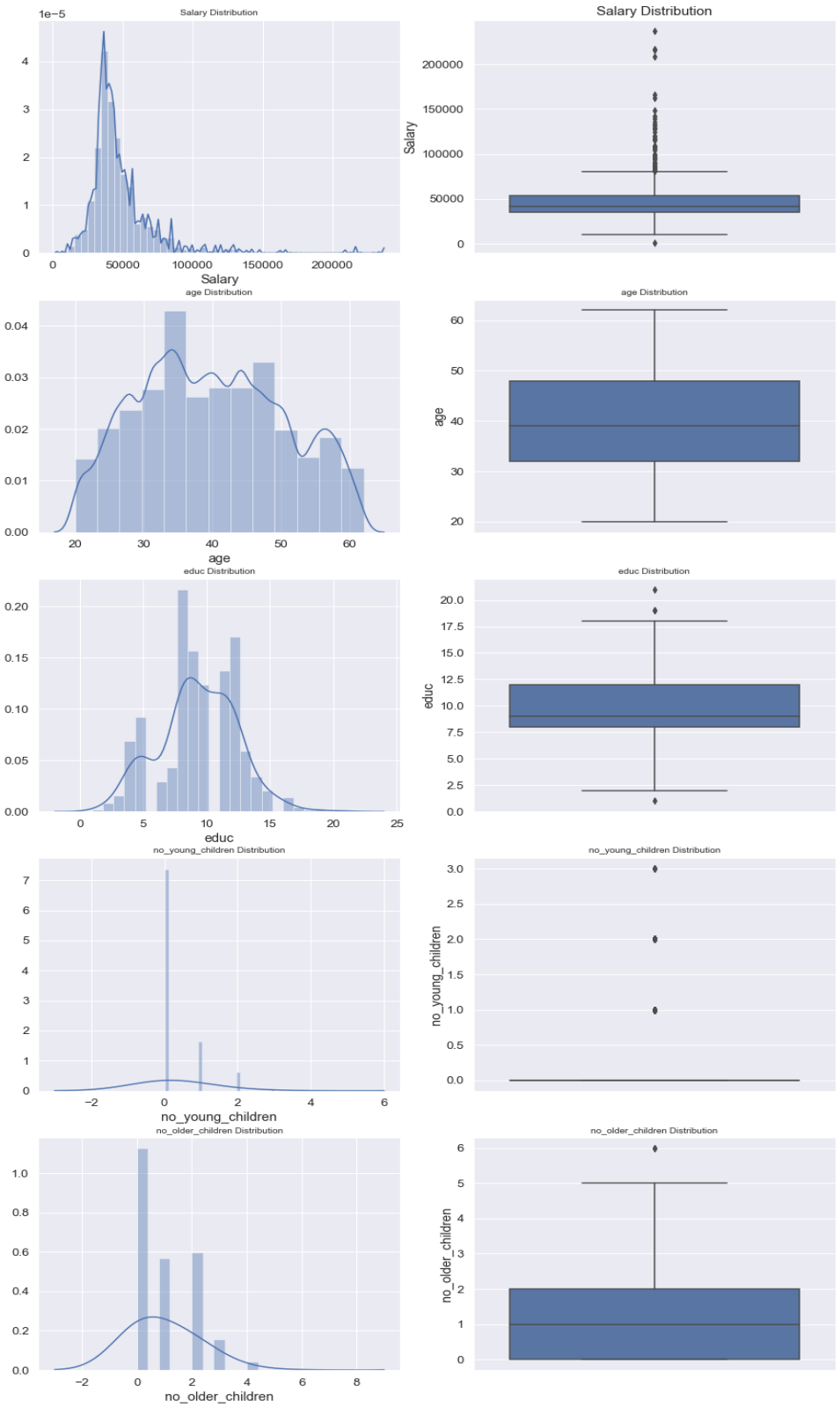


Fig13. Univariate Analysis for Holiday data

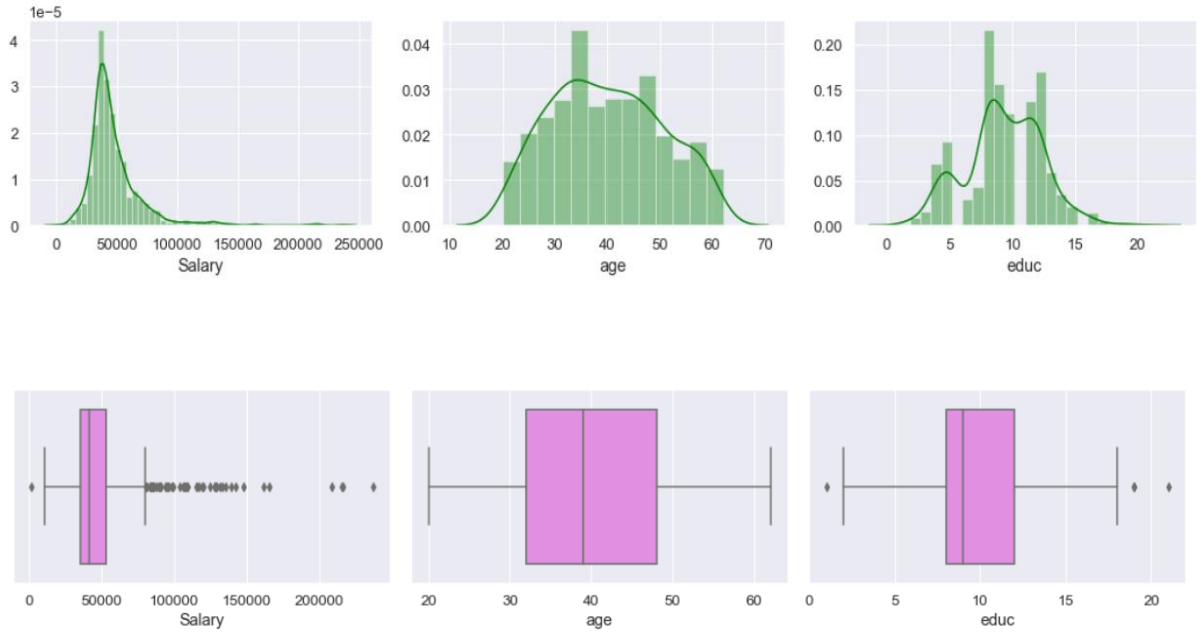


Fig14. Distribution and Boxplot for continuous variables

Skewness:

Salary	3.103216
age	0.146412
educ	-0.045501
no_young_children	1.946515
no_older_children	0.953951
dtype: float64	

Categorical Univariate Analysis:

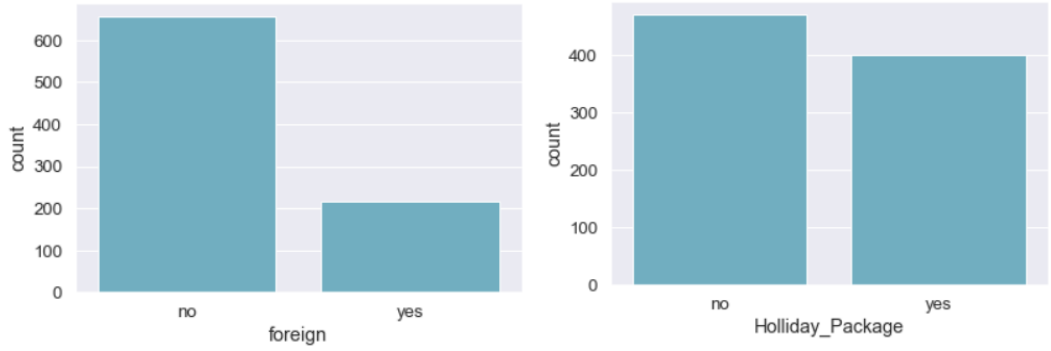
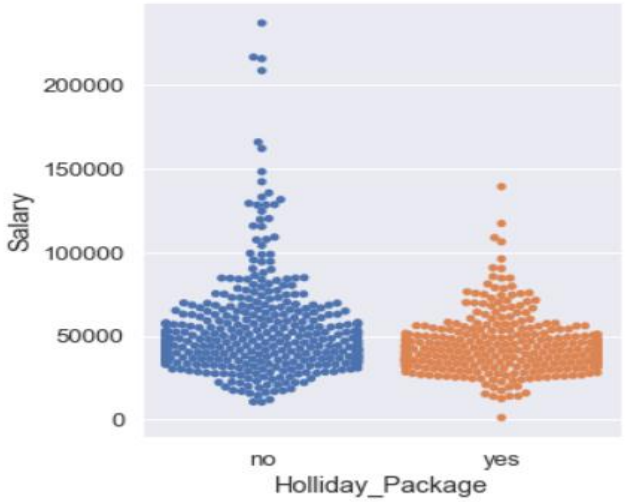


Fig15. Foreign vs Holiday_Package

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

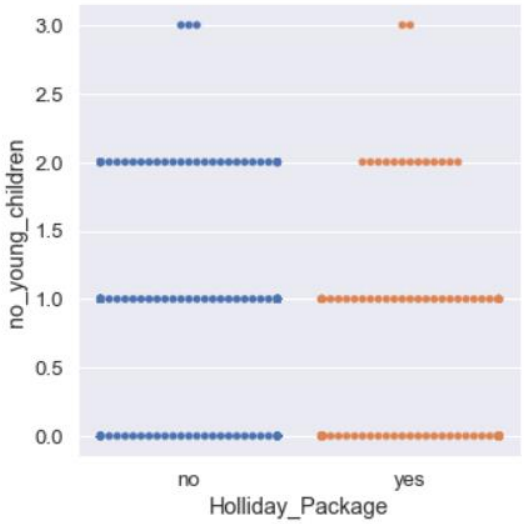
HOLIDAY PACKAGE VS SALARY:



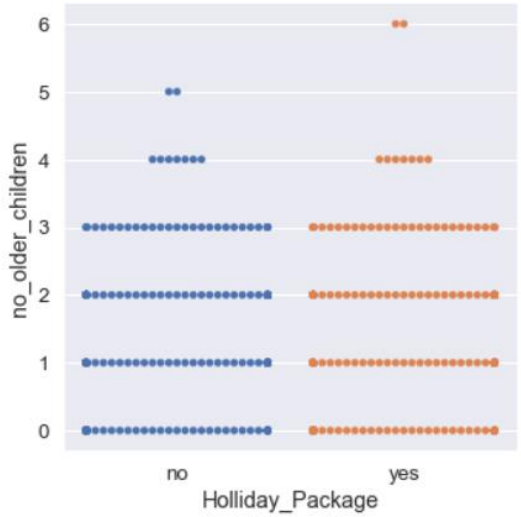
HOLIDAY PACKAGE VS AGE:

HOLIDAY PACKAGE VS EDUC:

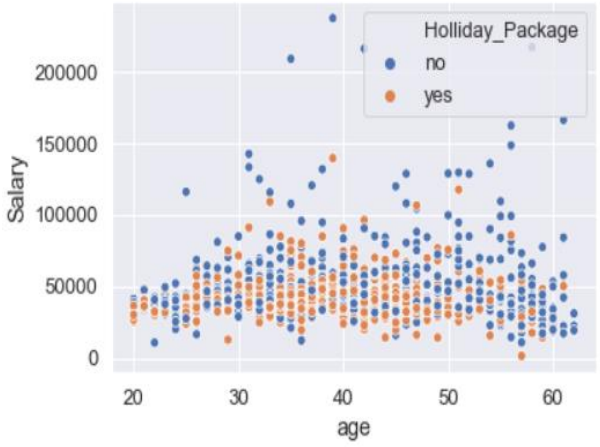
HOLIDAY PACKAGE VS YOUNG CHILDREN:

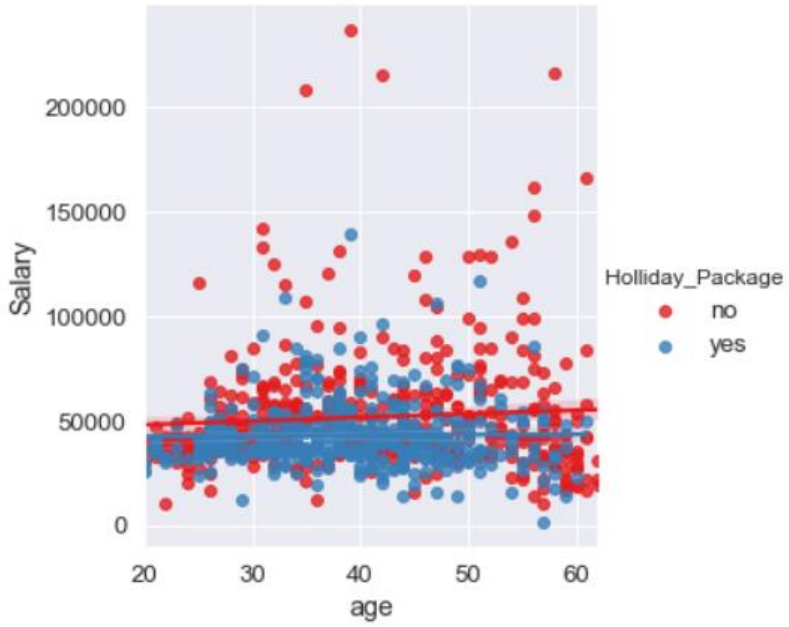


HOLIDAY PACKAGE VS OLDER CHILDREN:



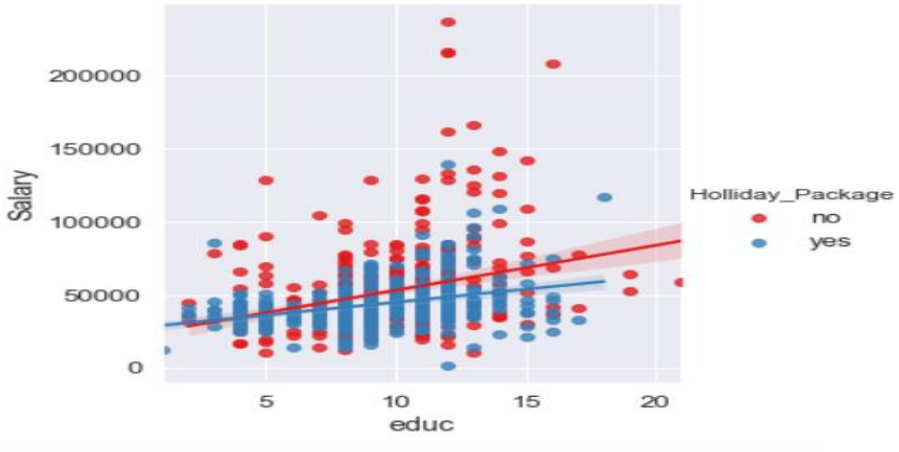
AGE VS SALARY VS HOLIDAY PACKAGE:

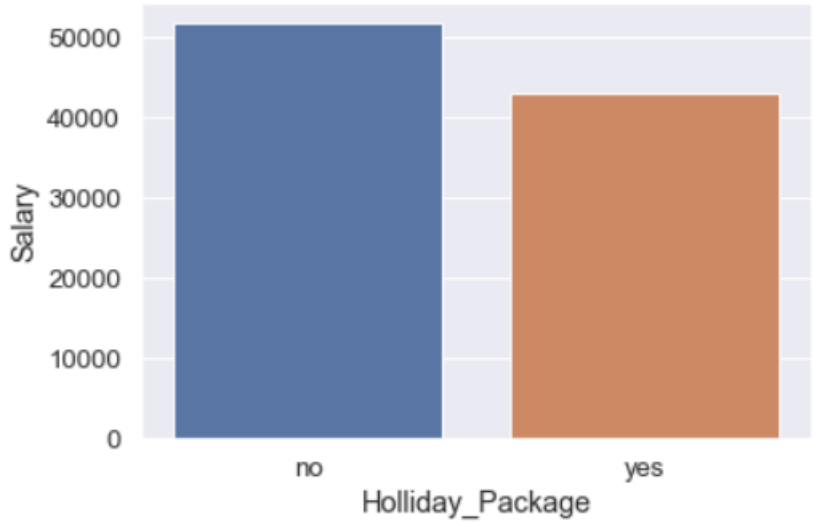




Employee age over 50 to 60 have seems to be not taking the holiday package, whereas in the age 30 to 50 and salary less than 50000 people have opted more for holiday package.

EDUS VS SALARY VS HOLIDAY PACKAGE:





PERCENTAGE OF TARGET VARIABLE:

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

Heatmap:

No multi collinearity in the data

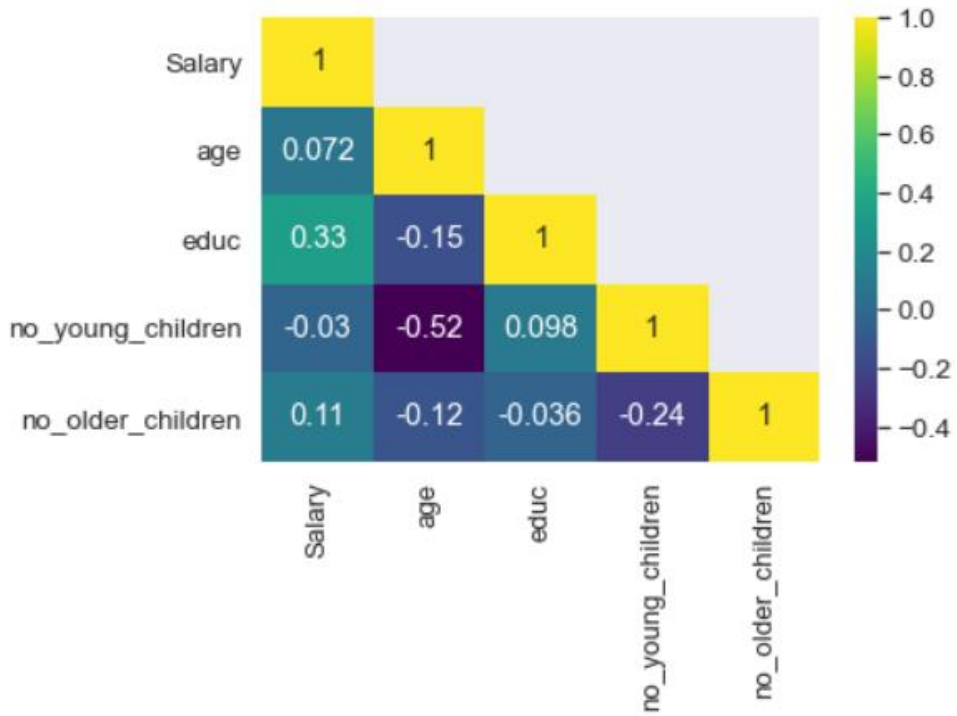


Fig16.Heatmap

Pairplot:

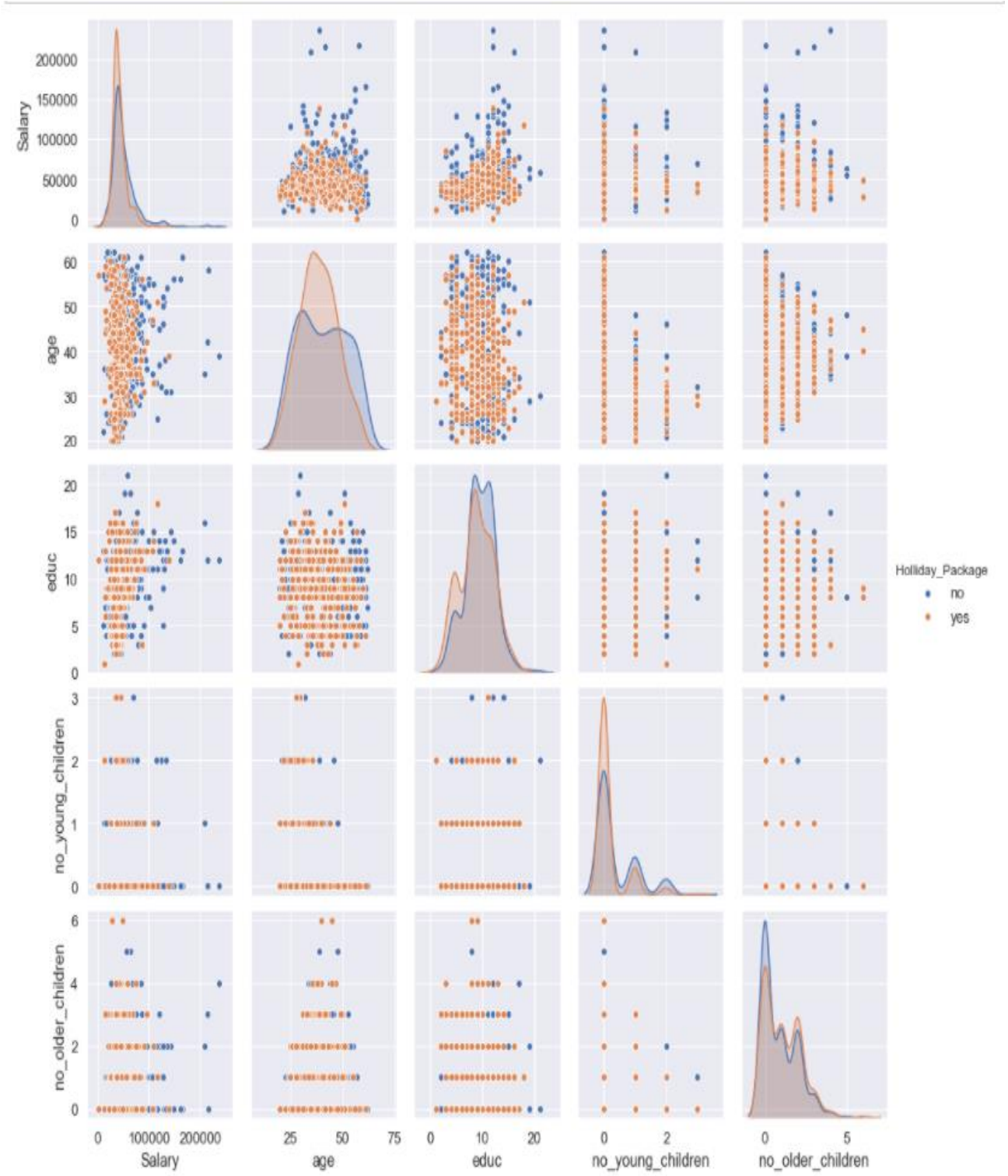


Fig17.Pairplot

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, I don't see any clear two different distribution in the data.

Outlier Treatment:

No outliers in the data, all outliers have been treated.

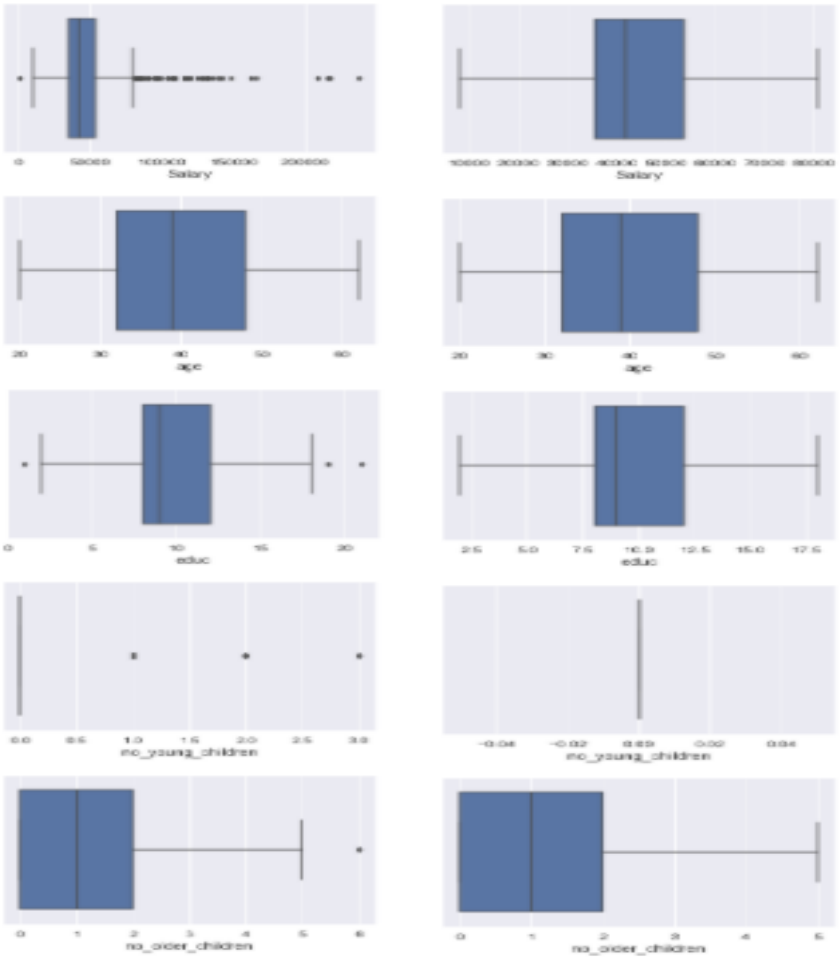


Fig18. Outlier tratment

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).

→Scaling is not necessary for this dataset.

→One hot Encoding should be done for the data before performing the data.

→The encoding helps the logistic regression model predict better results

→Let us look into the sample of encoded data:

	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

Table1.8 Sample of Encoded data

→Split the data into train and test (70:30)

→Fit the LogisticRegression() model

→GRID SEARCH METHOD:

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

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

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

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

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.

PERFORMANCE METRICS:

The grid search method gives, liblinear solver which is suitable for small datasets.

Tolerance and penalty have been found using grid search method

Predicting the training data,

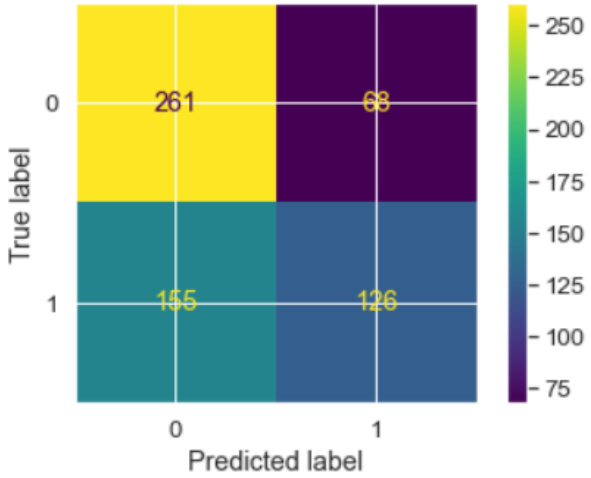
```
array([1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
       1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1,
       0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0,
       1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0,
       1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0,
       1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1,
       1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1,
       0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
       1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1,
       0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1,
       0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0], dtype=uint8)
```

Getting the probabilities on the test set:

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

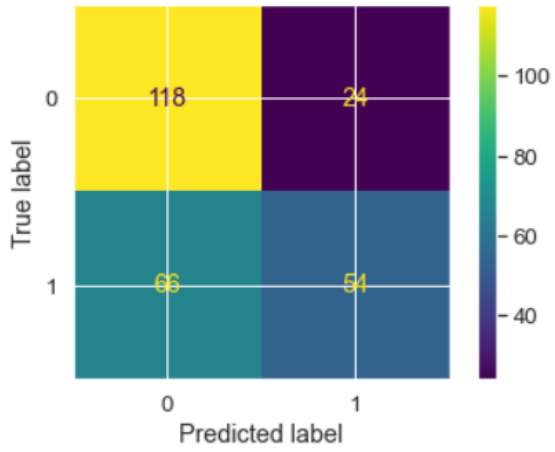
CONFUSION MATRIX ON TRAIN DATA:

	precision	recall	f1-score	support
0	0.63	0.79	0.70	329
1	0.65	0.45	0.53	281
accuracy			0.63	610
macro avg	0.64	0.62	0.62	610
weighted avg	0.64	0.63	0.62	610



CONFUSION MATRIX ON TEST DATA:

	precision	recall	f1-score	support
0	0.64	0.83	0.72	142
1	0.69	0.45	0.55	120
accuracy			0.66	262
macro avg	0.67	0.64	0.63	262
weighted avg	0.66	0.66	0.64	262



ACCURACY – train data → 0.6344262295081967

AUC, ROC CURVE FOR TRAIN DATA:

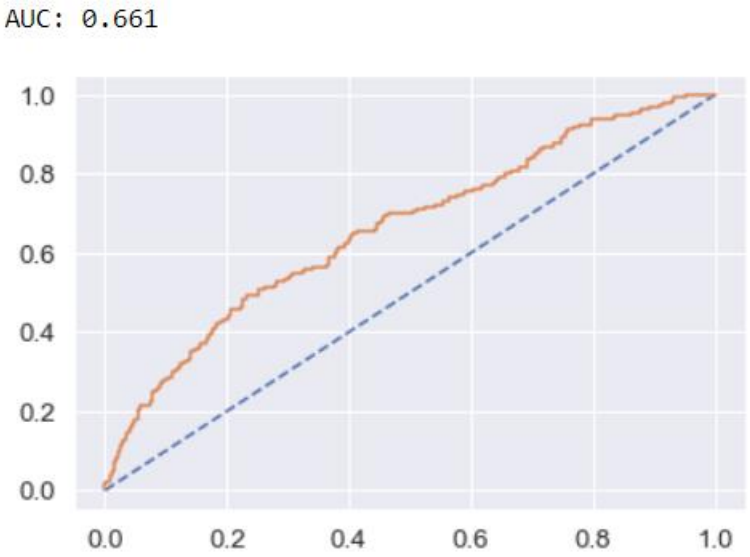


Fig19. Auc, Roc curve for train data

ACCURACY – test data → 0.6564885496183206

AUC, ROC CURVE FOR TEST DATA:

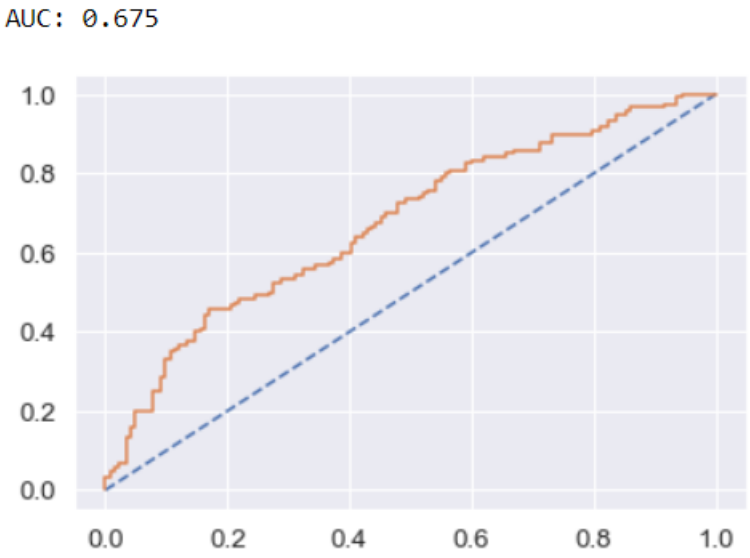


Fig20. Auc, Roc curve for test data

```
lr_train_precision  0.65
lr_train_recall  0.45
lr_train_f1  0.53
```

```
lr_test_precision  0.69
lr_test_recall  0.45
lr_test_f1  0.55
```

LINEAR DISCRIMINANT ANALYSIS:

→Build LDA model using LinearDiscriminantAnalysis ().

→Training data class prediction with a cut-off value of 0.5

→Test data class prediction with a cut-off value of 0.5

→PREDICTING THE VARIABLE:

Training data probability prediction.

Test data class prediction.

```
Model train accuracy: 0.6327868852459017


Model test accuracy: 0.6564885496183206
```

Classification report for train data:

	precision	recall	f1-score	support
0	0.62	0.80	0.70	329
1	0.65	0.44	0.52	281
accuracy			0.63	610
macro avg	0.64	0.62	0.61	610
weighted avg	0.64	0.63	0.62	610

Confusion Matrix of train data:

```
array([[263,  66],
       [158, 123]], dtype=int64)
```

Classification report of test data:

	precision	recall	f1-score	support
0	0.64	0.83	0.72	142
1	0.69	0.45	0.55	120
accuracy			0.66	262
macro avg	0.67	0.64	0.63	262
weighted avg	0.66	0.66	0.64	262

Confusion Matrix of test data:

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

AUC AND ROC CURVE:

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

Fig21. AUC &ROC curve for Training and testing data

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

Comparing both these models, we find both results are same, but LDA works better when there is category target variable.

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.

We had a business problem where we need predict whether an employee would opt for a holiday package or not, for this problem we had done predictions both logistic regression and linear discriminant analysis. Since both are results are same.
The EDA analysis clearly indicates certain criteria where we could find people aged above 50 are not interested much in holiday packages.
Hence, this is one of the we find aged people not opting for holiday packages.

People ranging from the age 30 to 50 generally opt for holiday packages. 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.

The important factors deciding the predictions are salary, age and educ.

Recommendations:

→To improve holiday packages over the age above 50 we can provide religious destination places.
→For people earning more than 150000 we can provide vacation holiday packages.
→For employee having more than number of older children we can provide packages in holiday vacation places.

THE END!