Project

Karthiheswar

Table of Contents

1 Project Objective

The objective of the report is to explore all the projects data set in Python and generate insights about the data set. This exploration report will consist of the following:

- Importing the dataset in Python
- Understanding the structure of dataset
- Checking null values and performing descriptive statistics
- Graphical exploration
- Univariate and Bivariate Analysis
- Encode the data for Modelling
- Applying Linear regression
- Predictions on Train and Test sets using Rsquare, RMSE
- Applying Logistic Regression and LDA
- Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and ROC_AUC score
- Insights from the dataset

2 Linear Regression on Cubic zirconia dataset for Gem Stones co ltd

2.1 Reading the data and exploratory data analysis

Reading the dataset (head)

	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

Exploratory data analysis

Describing the data

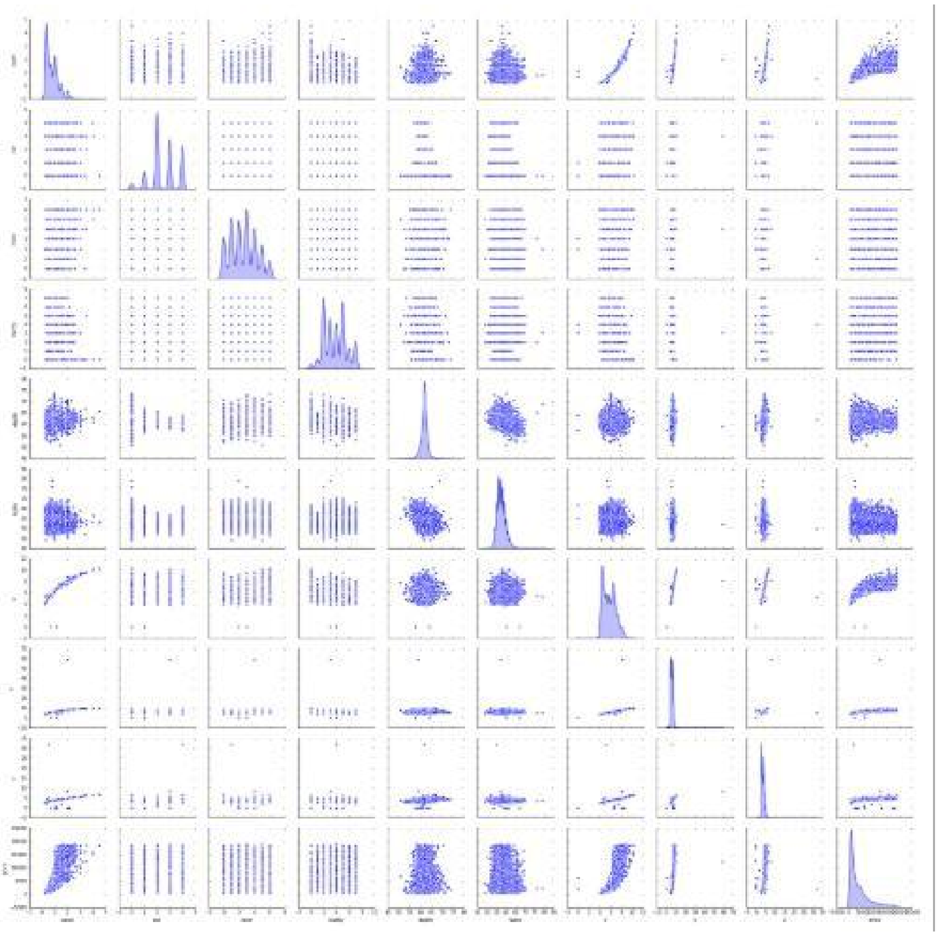
	carat	depth	table	x	y	z	price
count	26967.000000	26270.000000	26967.000000	26967.000000	26967.000000	26967.000000	26967.000000
mean	0.798375	61.745147	57.456080	5.729854	5.733569	3.538057	3939.518115
std	0.477745	1.412860	2.232068	1.128516	1.166058	0.720624	4024.864666
min	0.200000	50.800000	49.000000	0.000000	0.000000	0.000000	326.000000
25%	0.400000	61.000000	56.000000	4.710000	4.710000	2.900000	945.000000
50%	0.700000	61.800000	57.000000	5.690000	5.710000	3.520000	2375.000000
75%	1.050000	62.500000	59.000000	6.550000	6.540000	4.040000	5360.000000
max	4.500000	73.600000	79.000000	10.230000	58.900000	31.800000	18818.000000

S.no	Description	IQR values for all attributes
1	Carat	0.65
2	Depth	1.50
3	Table	3.00
4	x	1.84
5	y	1.83
6	z	1.14
7	Price	4415.00

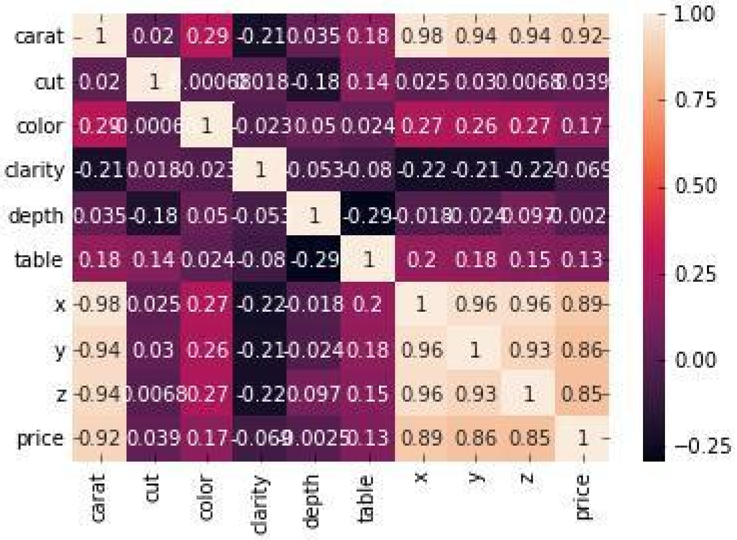
Covariance of each attribute against every other attribute

	carat	depth	table	x	y	z	price
carat	0.228241	0.023844	0.193742	0.526403	0.524250	0.323839	1.773678e+03
depth	0.023844	1.996174	-0.939265	-0.029813	-0.040760	0.100411	-1.459691e+01
table	0.193742	-0.939265	4.982127	0.494228	0.474596	0.239574	1.140420e+03
x	0.526403	-0.029813	0.494228	1.273549	1.266851	0.777946	4.025446e+03
y	0.524250	-0.040760	0.474596	1.266851	1.359690	0.780563	4.018538e+03
z	0.323839	0.100411	0.239574	0.777946	0.780563	0.519298	2.466906e+03
price	1773.677848	-14.596911	1140.419986	4025.446081	4018.537829	2466.905683	1.619954e+07

Scatter plot



Heatmap



Skeweness

S.no	Description	Skeweness of every attribute
1	Carat	1.11
2	Depth	-0.03
3	Table	0.76
4	x	0.38
5	y	3.85
6	z	2.56
7	Price	1.61

Checking the null values

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

Checking the Data types

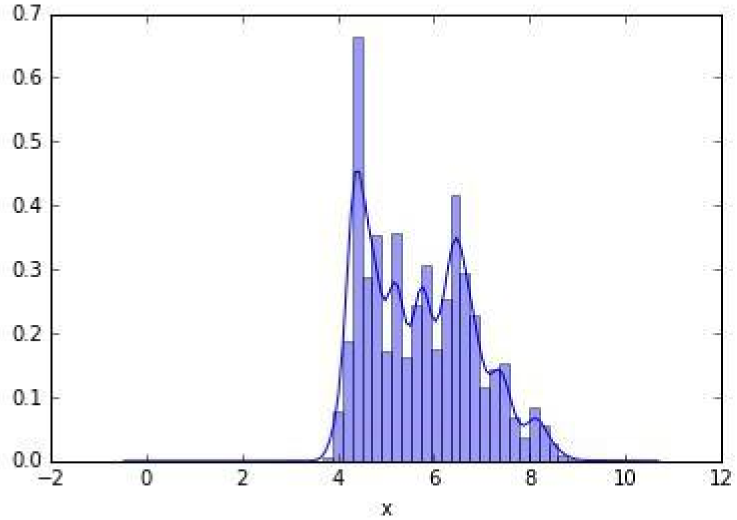
```
carat      float64
cut         object
color       object
clarity     object
depth      float64
table      float64
x          float64
y          float64
z          float64
price        int64
```

Checking the Shape

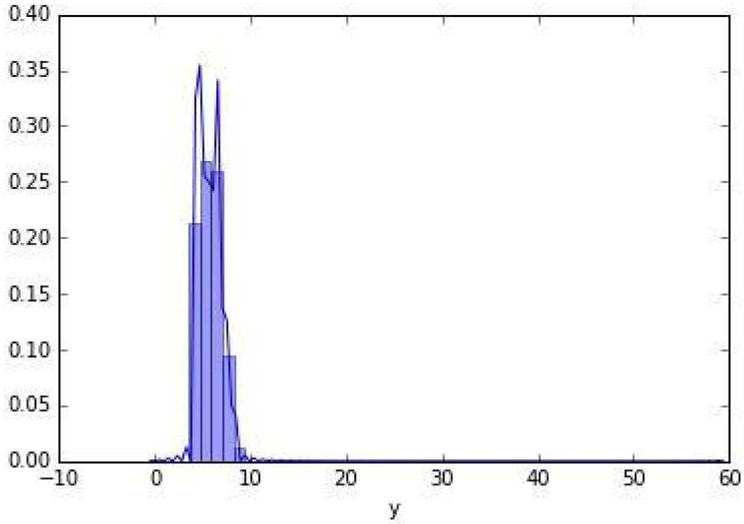
(26967, 10)

Univariate Analysis

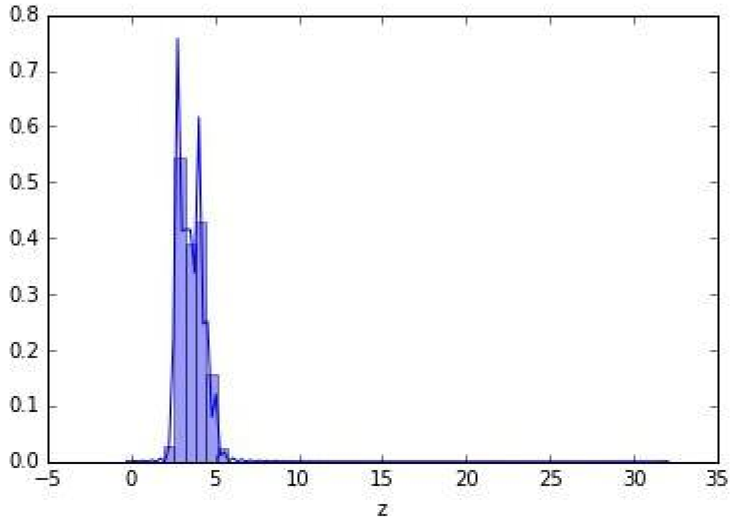
Variable- x



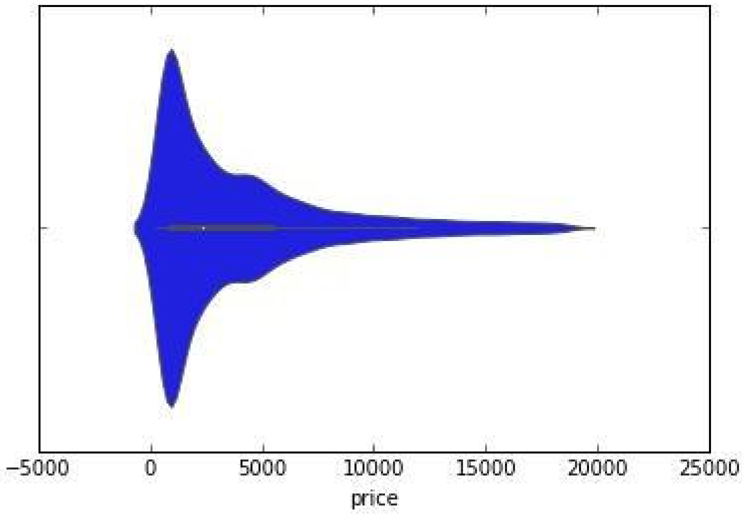
Variable- y



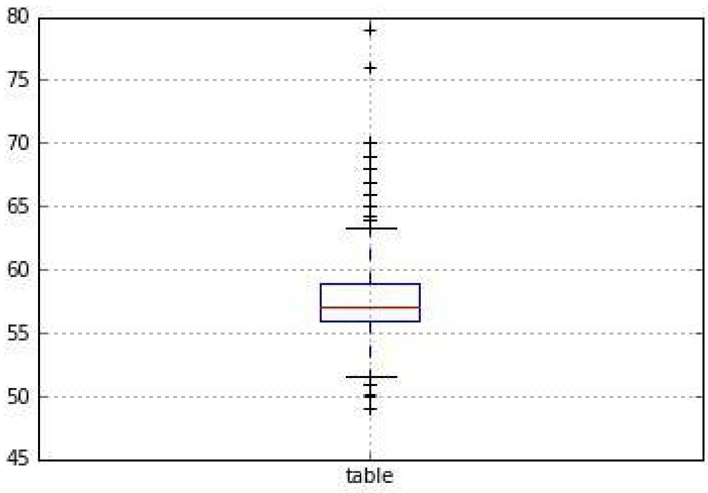
Variable- z



Variable- Price

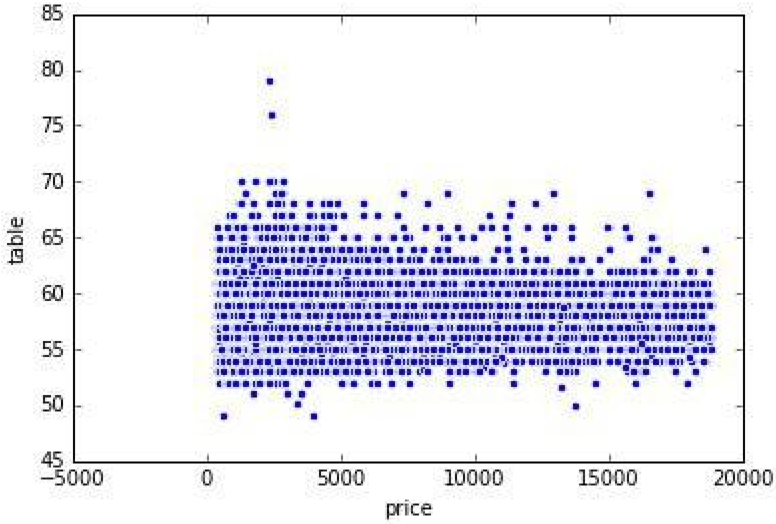


Variable- Table

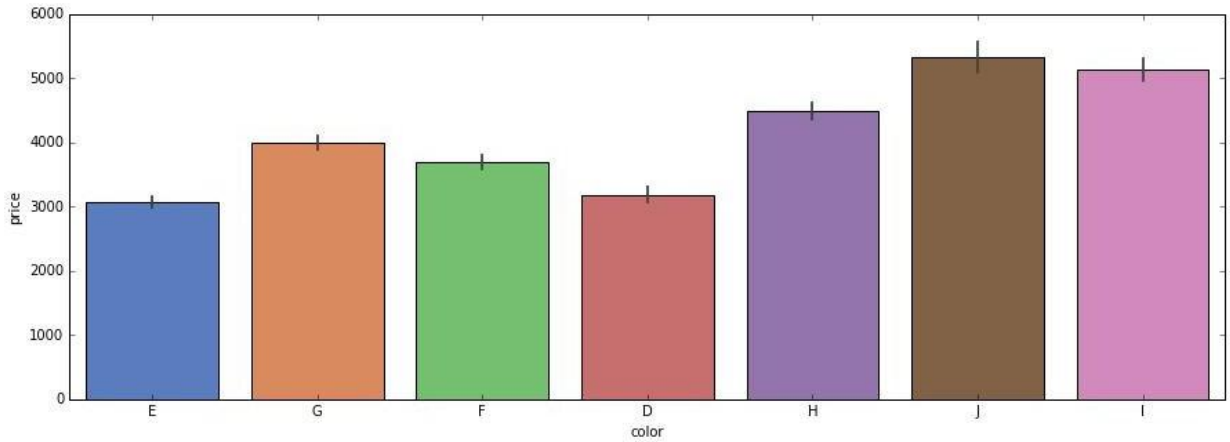


Bivariate Analysis

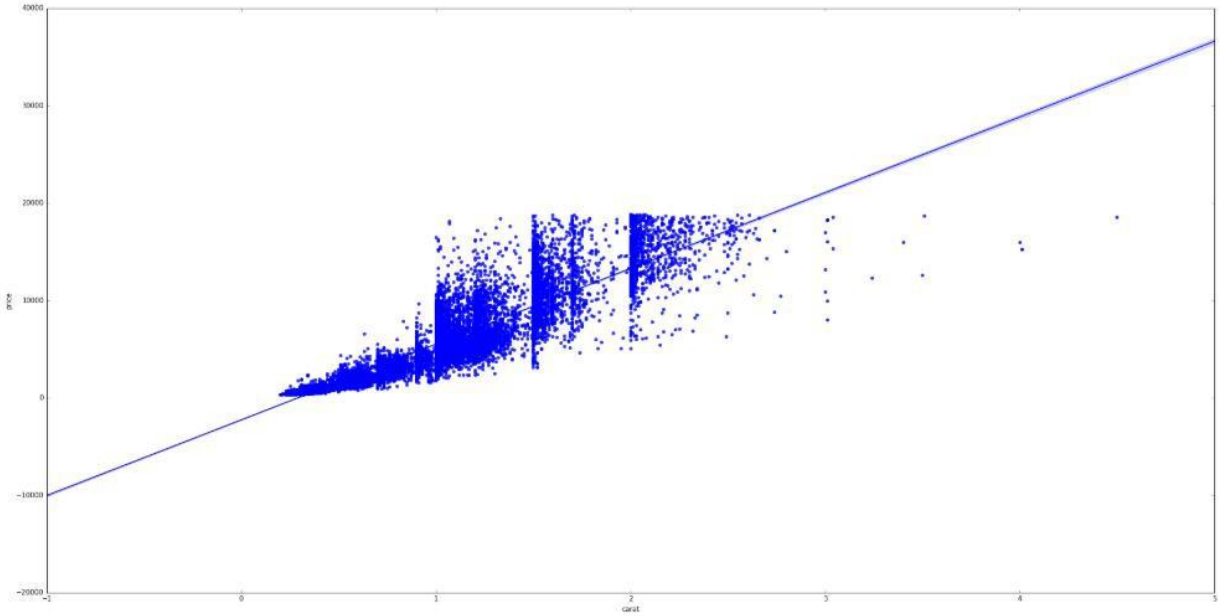
Variable- Table vs Price



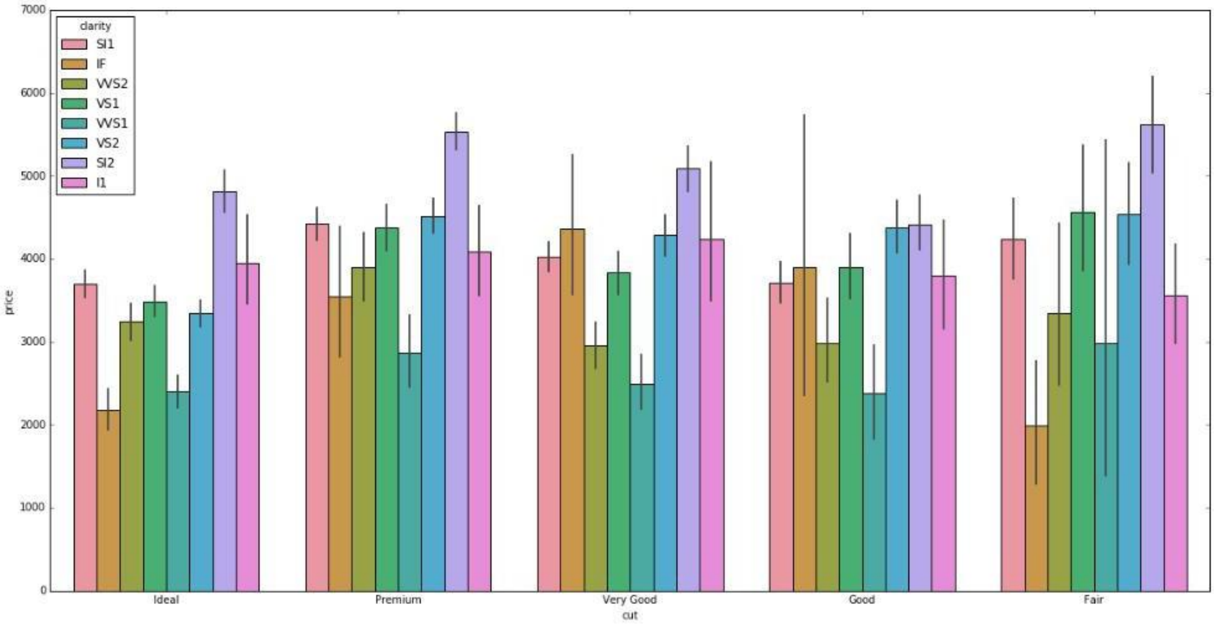
Variable- Color vs Price



Variable- Carat vs Price



Variable- Cut vs Price and Clarity



2.2 Imputing null values, checking the values equal to zero

Checking the values equal to zero

```
carat       0
cut         0
color       0
clarity     0
depth       0
table       0
x           3
y           3
z           9
price       0
```

There are zeros at x, y and z variable. As these variables denotes the length, width and height of the cubic zirconia, those values cannot be zero.

Imputing null values and the values equal to zero

The null values and values that are equal to zero imputed with their variable mean.

Scaling of data

Scaling is definitely necessary as each columns contains different range of variables and different parameters, it should be scaled. Also in data without scaling, the intercept value is too high which is not practically possible and model doesn't fit good. So scaling is necessary.

2.3 Applying Linear regression

Encoding the data

The dataset is encoded using label encoder. As one-hot encoding increases the no. of variables, this method is not used.

	carat	cut	color	clarity	depth	table	x	y	z	price
0	0.30	2	1	2	62.1	58.0	4.27	4.29	2.66	499
1	0.33	3	3	1	60.8	58.0	4.42	4.46	2.7	984
2	0.90	4	1	7	62.2	60.0	6.04	6.12	3.78	6289
3	0.42	2	2	4	61.6	56.0	4.82	4.8	2.96	1082
4	0.31	2	2	6	60.4	59.0	4.35	4.43	2.65	779

Data Split and Linear regression

The data is split into test and train (70:30) and then Linear regression is applied. The resulting coefficient of each variables and intercept value are

- The coefficient for carat is 1.2981884203932201
- The coefficient for cut is 0.014029789205391273
- The coefficient for color is -0.1154361779834659
- The coefficient for clarity is 0.12397605606404906
- The coefficient for depth is -0.05276359614332049
- The coefficient for table is -0.05371308482970911
- The coefficient for x is -0.30652181518825744
- The coefficient for y is 0.0003825747068687583
- The coefficient for z is -0.005671010725680481
- The intercept for our model is 2.683798169200023e-16

Predictions on Train and Test

Predictions on test data

```
array([[ 2.19430539],
       [ 1.34962685],
       [-0.8901628 ],
       ...,
       [-0.28540472],
       [ 1.17121855],
       [-0.48750799]])
```

Predictions on train data

```
array([[-0.71421742],
       [ 1.56764845],
       [-1.12649502],
       ...,
       [ 2.68859147],
       [ 1.3349705 ],
       [ 0.49290215]])
```

Rsquare value of test data is 0.8877

Rsquare value of train data is 0.8869

RMSE value of test data is 0.3349

RMSE value of train data is 0.3362

2.4 Inference

Almost many other variables doesn’t show any impact against price. Also x, y and z are correlated with each other, meanwhile x and carat are correlated. The variable- x shows comparatively better variation with price than other variables and it has highest coefficient of -0.306. After x, variable- clarity has higher positive coefficient with 0.123. So its better to concentrate on x and clarity to improve the profit on stones. Higher the length, lower the profit on stones and higher the clarity, higher the profit on stones. Also color, table and depth are some important attributes for the profit and loss on stones.

3 Logistic regression and LDA on Holiday_Package for tour and travel agency

3.1 Data Ingestion: Reading the dataset, performing descriptive statistics, null value condition check, Univariate and Bivariate Analysis and exploratory data analysis

Reading the dataset (head)

	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

Descriptive statistics

	Salary	age	educ	no_young_children	no_older_children
count	872.000000	872.000000	872.000000	872.000000	872.000000
mean	47729.172018	39.955275	9.307339	0.311927	0.982798
std	23418.668531	10.551675	3.036259	0.612870	1.086786
min	1322.000000	20.000000	1.000000	0.000000	0.000000
25%	35324.000000	32.000000	8.000000	0.000000	0.000000
50%	41903.500000	39.000000	9.000000	0.000000	1.000000
75%	53469.500000	48.000000	12.000000	0.000000	2.000000
max	236961.000000	62.000000	21.000000	3.000000	6.000000

Null value checks

S.no	Description	Null value condition
1	Holliday_Package	0
2	Salary	0
3	Age	0
4	Educ	0
5	No_young_children	0
6	No_older_children	0
7	Foreign	0

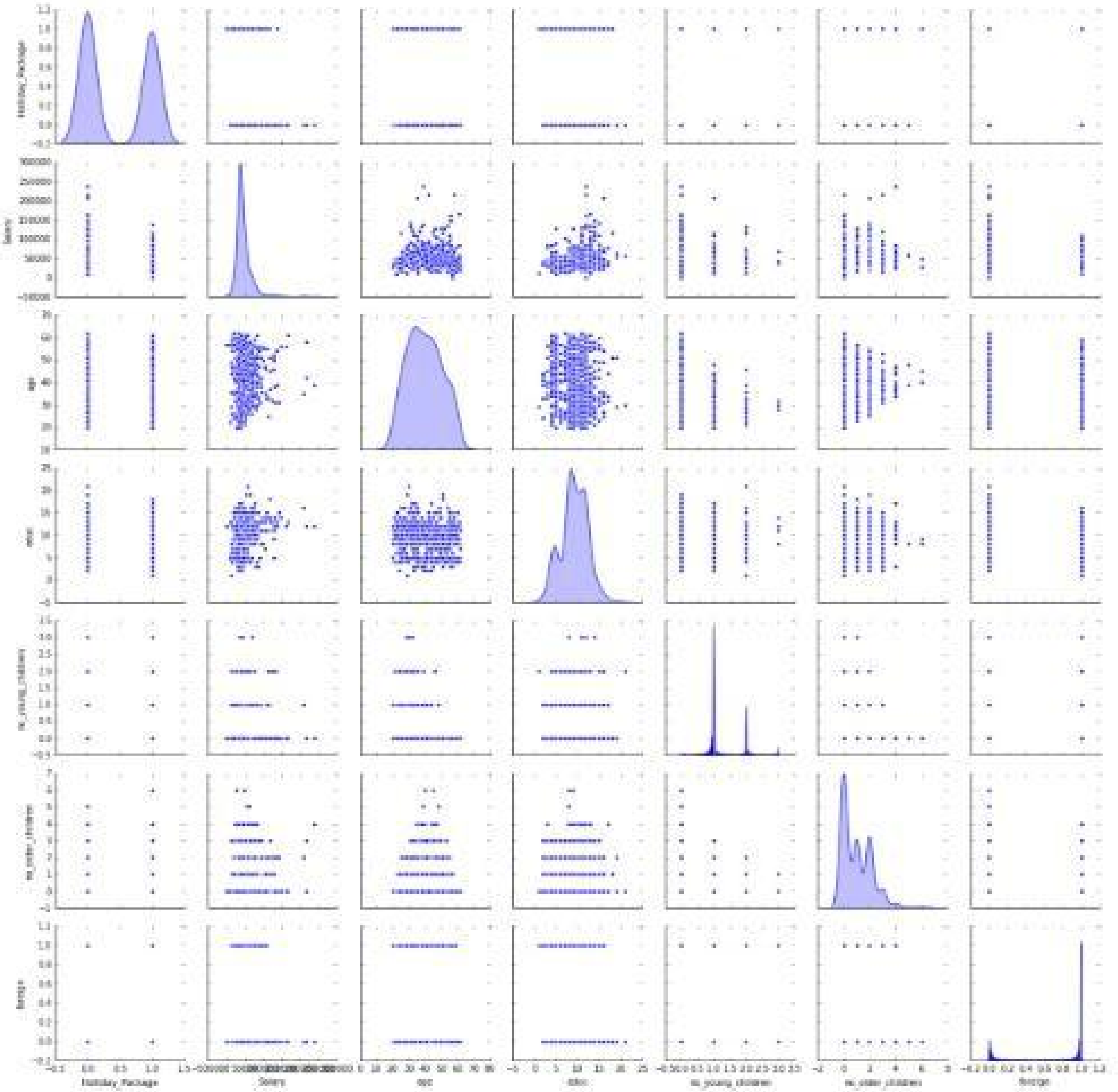
Skeweness

S.no	Description	Skeweness of every attribute
1	Holliday_Package	0.161
2	Salary	3.103
3	Age	0.146
4	Educ	-0.045
5	No_young_children	1.946
6	No_older_children	0.953
7	Foreign	1.170

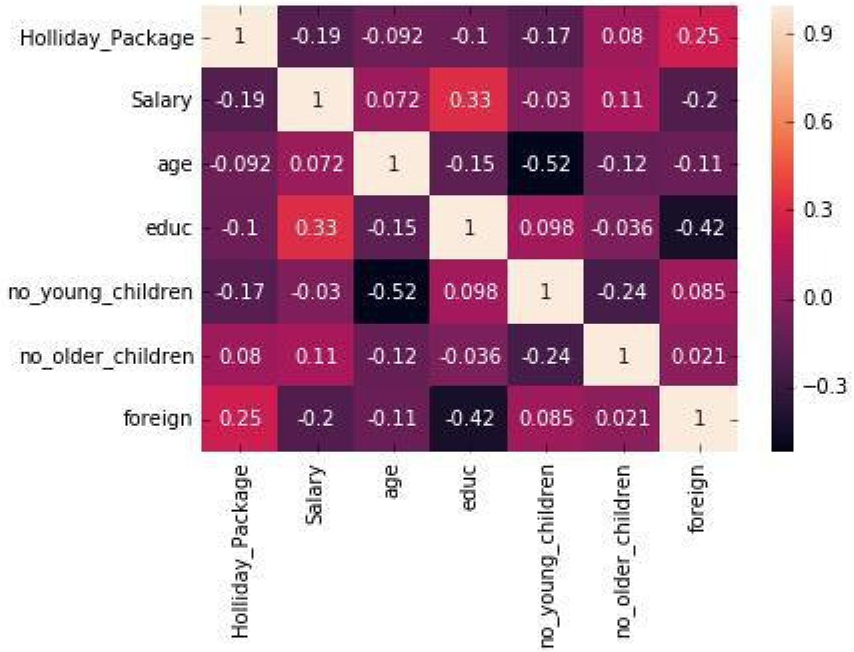
Covariance of each attribute against every other attribute

	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
Holliday_Package	0.248674	-2.168586e+03	-0.485724	-0.155273	-0.052908	0.043511	0.054730
Salary	-2168.585510	5.484340e+08	17719.779229	23218.662341	-425.752915	2895.613755	-2033.582269
age	-0.485724	1.771978e+04	111.337837	-4.783024	-3.356871	-1.332573	-0.488335
educ	-0.155273	2.321866e+04	-4.783024	9.218867	0.183012	-0.119851	-0.550385
no_young_children	-0.052908	-4.257529e+02	-3.356871	0.183012	0.375610	-0.158807	0.022530
no_older_children	0.043511	2.895614e+03	-1.332573	-0.119851	-0.158807	1.181104	0.010006
foreign	0.054730	-2.033582e+03	-0.488335	-0.550385	0.022530	0.010006	0.186562

Pairplot

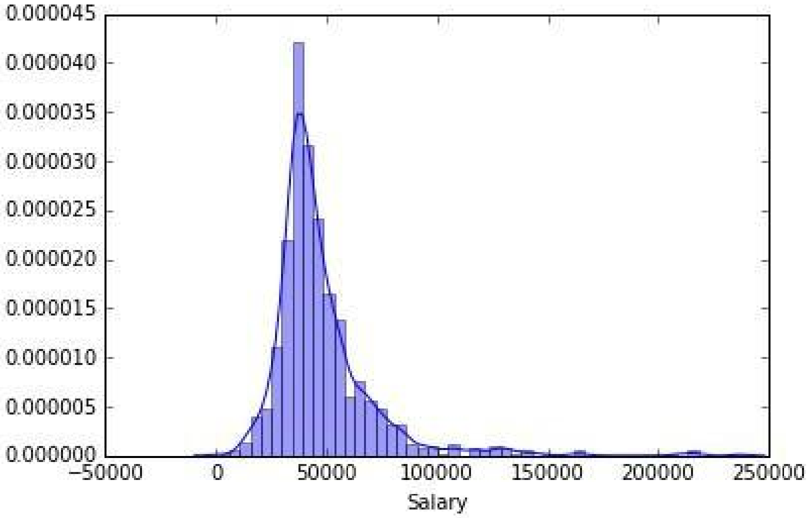


Heatmap

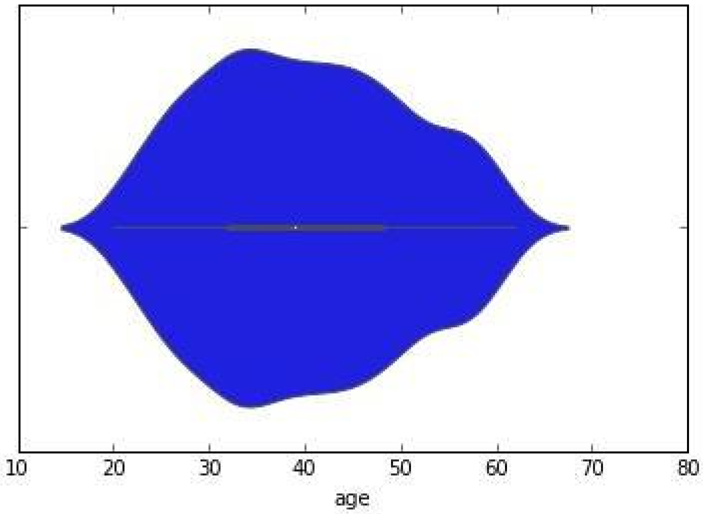


Univariate Analysis

Variable- Salary

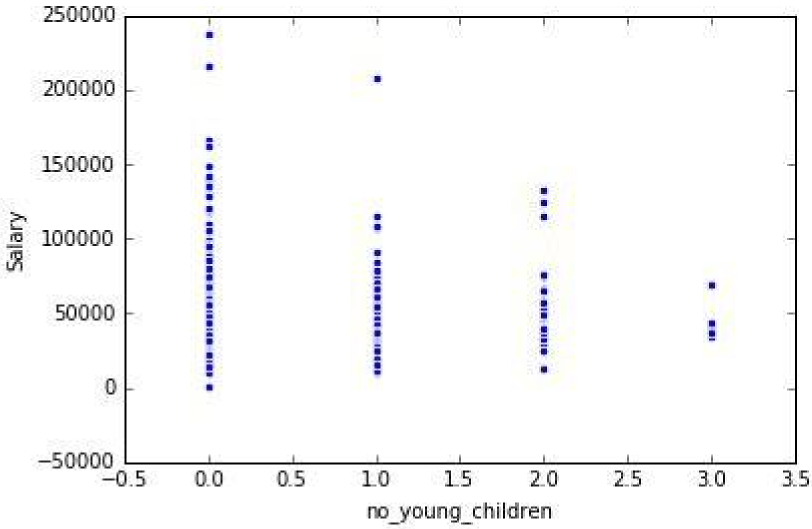


Variable- age

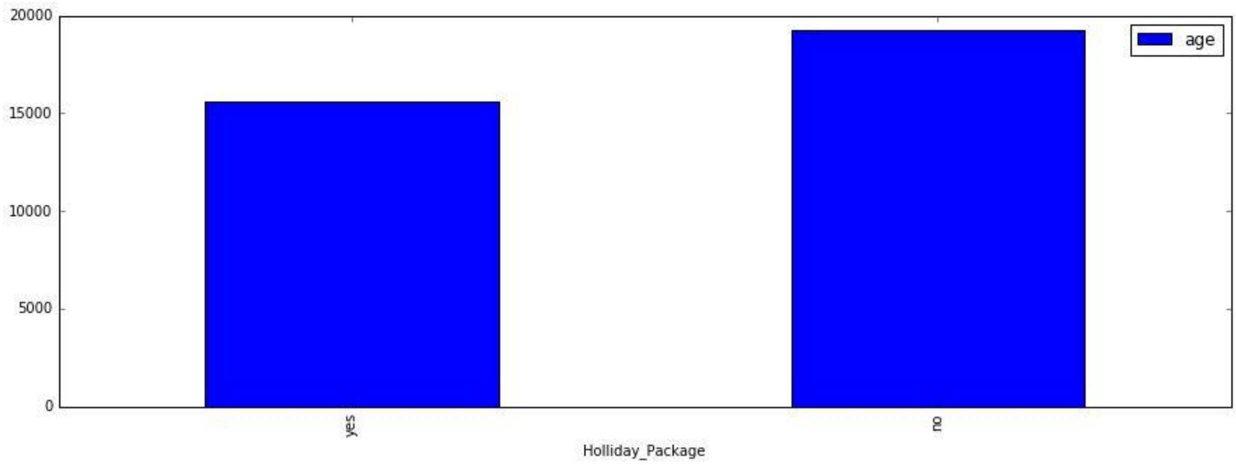


Bivariate Analysis

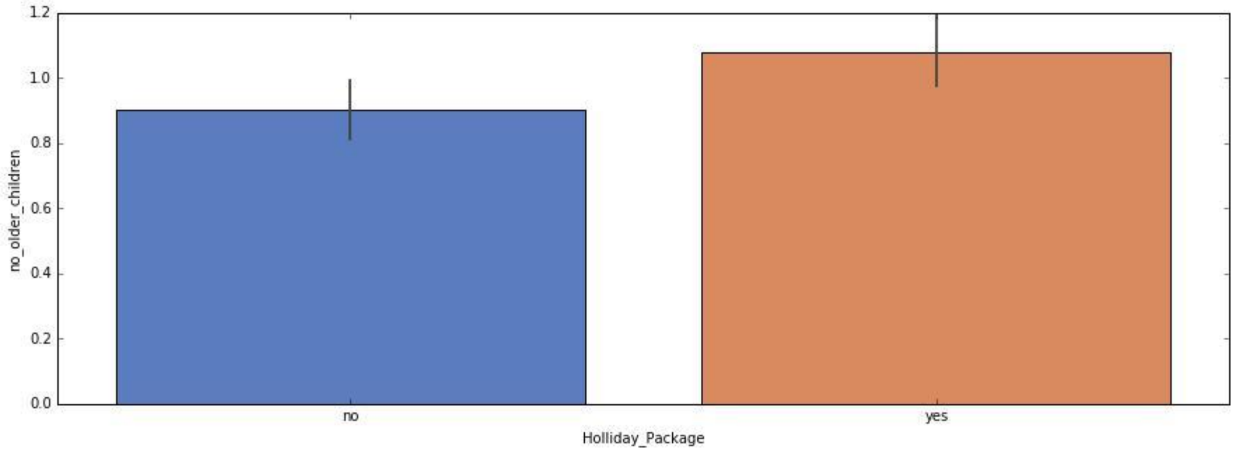
Variable- no_young_children vs Salary



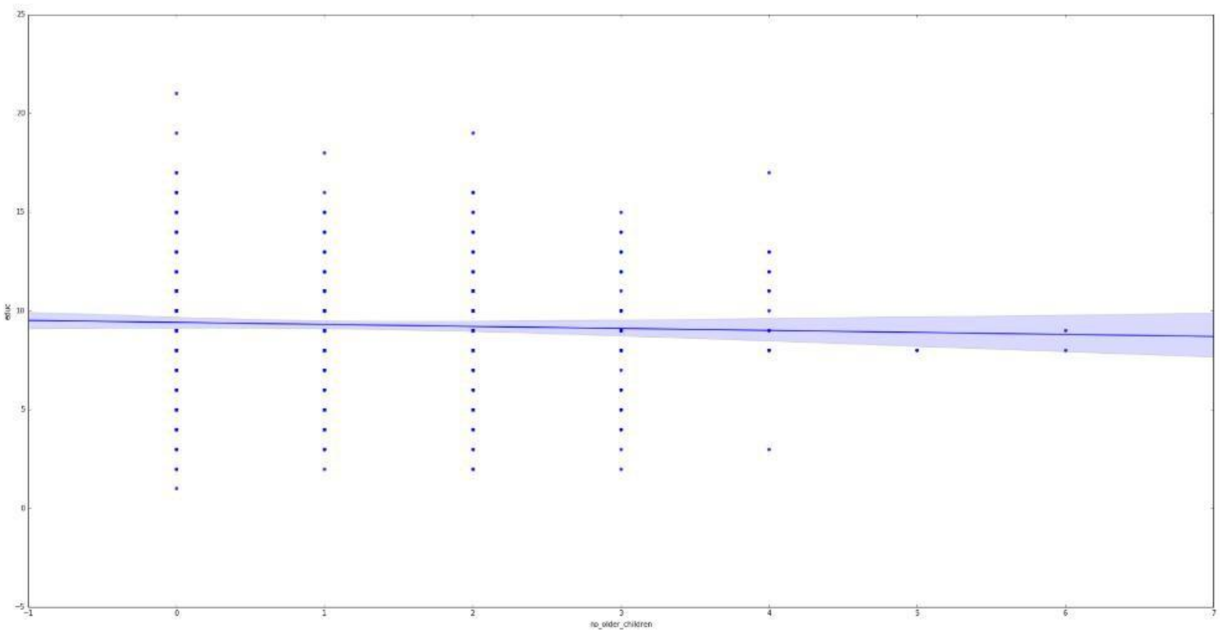
Variable- Holliday_Package vs age



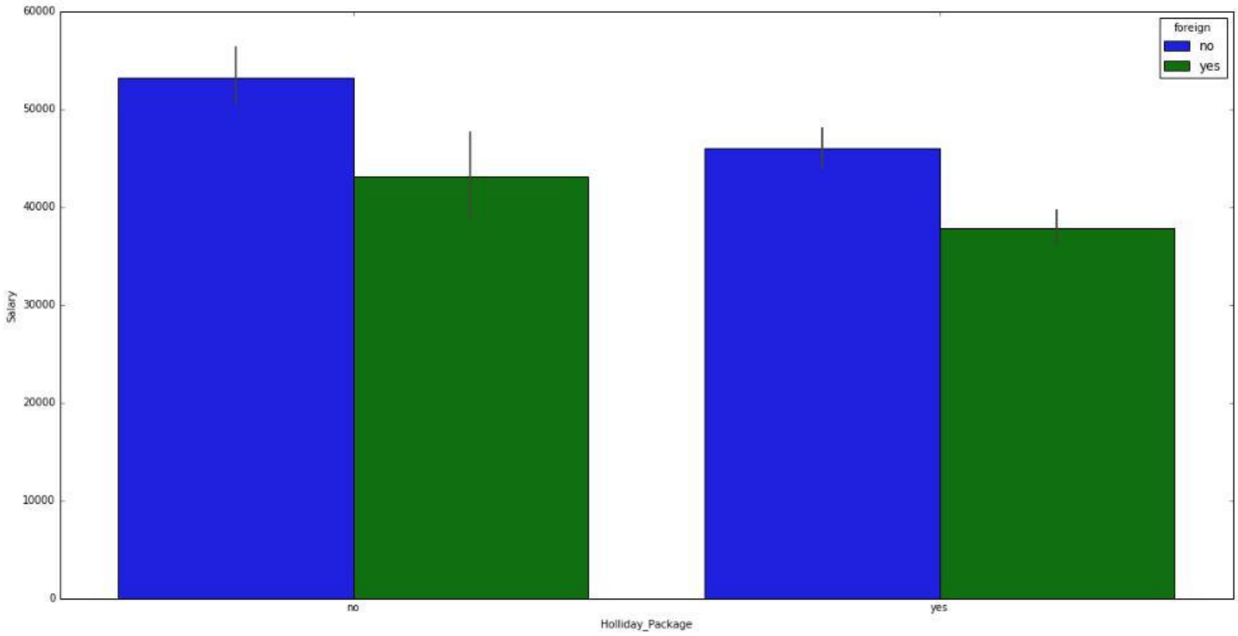
Variable- Holliday_Package vs no_older_children



Variable- no_older_children vs educ



Variable- Salary vs Holliday_Package and foreign



The given dataset is imported and there are no null values present. Every columns containing discrete variables are converted into continuous variables for purpose of model building.

3.2 Applying Logistic Regression and LDA (linear discriminant analysis)

Encoding the data

The dataset is encoded using label encoder. As one-hot encoding increases the no. of variables, this method is not used.

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

Data Split and Logistic regression

The data is split into test and train (70:30) and then Logistic regression is applied.

```
LogisticRegression(C=1.0, class_weight='balanced', dual=False,
        fit_intercept=True, intercept_scaling=1, l1_ratio=None,
        max_iter=100, multi_class='warn', n_jobs=None, penalty='l2',
        random_state=None, solver='warn', tol=0.0001, verbose=0,
        warm_start=False)
```

LDA

After splitting the data into test and train (70:30), LDA is applied.

```
LinearDiscriminantAnalysis(n_components=None, priors=None, shrinkage=None,
                           solver='svd', store_covariance=False, tol=0.0001)
```

3.3 Performance Metrics: Performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model

LOGISTIC REGRESSION

Predictions on train data

```
array([0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1,
       0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0,
       1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0,
       1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1,
       1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1,
       0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1,
       0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1,
       1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0,
       0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1,
       0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0,
       1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1,
       0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0,
       1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1,
       0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1,
       1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1,
       1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0,
       1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0])
```

Predictions on test data

```
array([0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1,
       1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0,
       1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1,
       1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0,
       1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
       1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0,
```

1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1,
0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0,
0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0,
0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0,
0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0])

Accuracy of train data

0.6540983606557377

Accuracy of test data

0.6335877862595419

Classification report and Confusion of train data

```
Confusion Matrix
 [[223 103]
 [108 176]]

Classification Report
              precision    recall  f1-score   support

           0       0.67      0.68      0.68       326
           1       0.63      0.62      0.63       284

    accuracy                           0.65       610
   macro avg       0.65      0.65      0.65       610
weighted avg       0.65      0.65      0.65       610
```

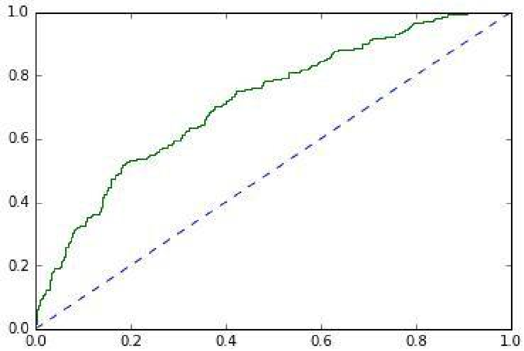
Classification report and Confusion of test data

```
Confusion Matrix
 [[96 49]
 [47 70]]

Classification Report
              precision    recall  f1-score   support

           0       0.67      0.66      0.67       145
           1       0.59      0.60      0.59       117

    accuracy                           0.63       262
   macro avg       0.63      0.63      0.63       262
weighted avg       0.63      0.63      0.63       262
```

AUC and ROC for the training data

```
AUC: 0.721
```

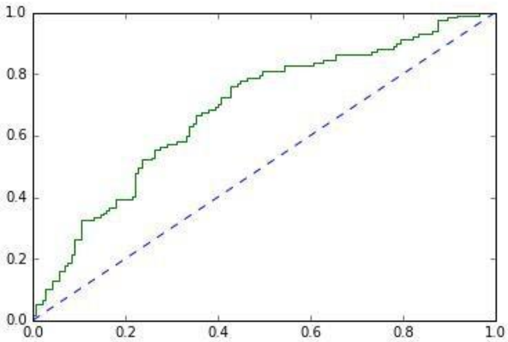
```
[<matplotlib.lines.Line2D at 0x32ef709630>]
```



AUC and ROC for the testing data

```
AUC: 0.687
```

```
[<matplotlib.lines.Line2D at 0x32ef75e400>]
```



LDA

Predictions on train data

```
array([0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0,
      0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1,
      0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
     1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0,
     0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
     1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
     0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
     0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1,
     0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0,
     0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1,
     0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0,
     1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1,
     0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
     1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0,
     1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0,
     1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1,
     0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0,
     0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1,
     0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0,
     1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0,
```

```
1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1,
0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0,
0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0,
1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0,
1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0])
```

Predictions on test data

```
array([0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1,
       1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0,
       0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1,
       1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0,
       1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
      1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0,
      0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
      0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
      1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0,
      0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0,
      0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1,
      0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0])
```

Accuracy of train data

0.6721311475409836

Accuracy of test data

0.6412213740458015

Classification report and Confusion of train data

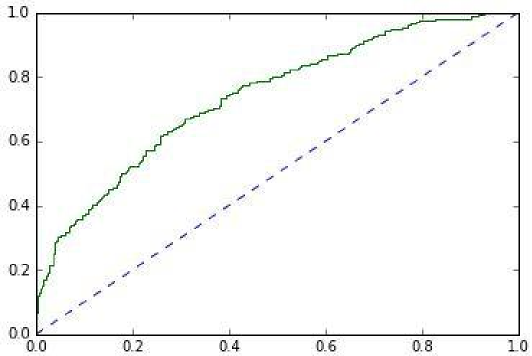
```
Confusion Matrix
 [[252  74]
 [126 158]]

Classification Report
              precision    recall  f1-score   support

           0       0.67      0.77      0.72       326
           1       0.68      0.56      0.61       284

    accuracy                           0.67       610
   macro avg       0.67      0.66      0.66       610
weighted avg       0.67      0.67      0.67       610
```

Classification report and Confusion of test data

```
Confusion Matrix
 [[103  42]
 [ 52  65]]

Classification Report
              precision    recall  f1-score   support

           0       0.66      0.71      0.69       145
           1       0.61      0.56      0.58       117

    accuracy                           0.64       262
   macro avg       0.64      0.63      0.63       262
weighted avg       0.64      0.64      0.64       262
```

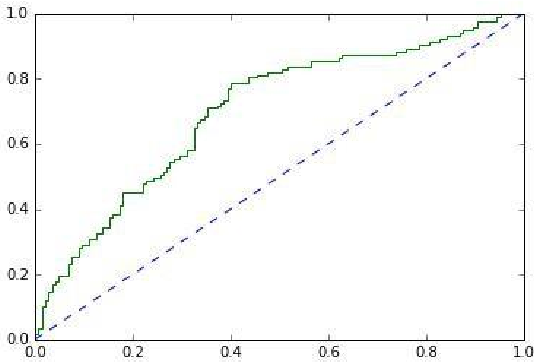
AUC and ROC for the training data

```
AUC: 0.742

[<matplotlib.lines.Line2D at 0x32ef7bfc88>]
```



AUC and ROC for the testing data

```
AUC: 0.703

[<matplotlib.lines.Line2D at 0x32ef822588>]
```



Comparison of both the models

<table>
<tr><th colspan="2">Comparison of classification report and confusion matrix for train data</th></tr>
<tr><td>Logistic Regression model</td><td></td></tr>
<tr><td>LDA model</td><td>Confusion Matrix
[[252 74]
[126 158]]

Classification Report
precision recall f1-score support

0 0.67 0.77 0.72 326
1 0.68 0.56 0.61 284

accuracy 0.67 610
macro avg 0.67 0.66 0.66 610
weighted avg 0.67 0.67 0.67 610</td></tr>
</table>

1. The LDA model predicts highest precision of 0.68 for opting the holiday package and precision of 0.67 for not opting the holiday package.
2. Logistic Regression tree model predicts same precision of 0.67 for not opting the holiday package but predicts least precision of 0.63 for opting the holiday package.
3. But however accuracy for LDA model is 0.67 which higher than Logistic Regression model which has 0.65.

3.4 Inference: Business insights and recommendations

The variable- foreign and salary shows comparatively better impact on holiday package. Employee with high foreigner have more opted for holiday package. Also employee with higher salary didn't opt for holiday package. Employee with no young children have opted for less holiday package.

4 Appendix A – Source Code

Karthiheswar_PredictiveModeling.ip