Problem 2: Logistic Regression and LDA

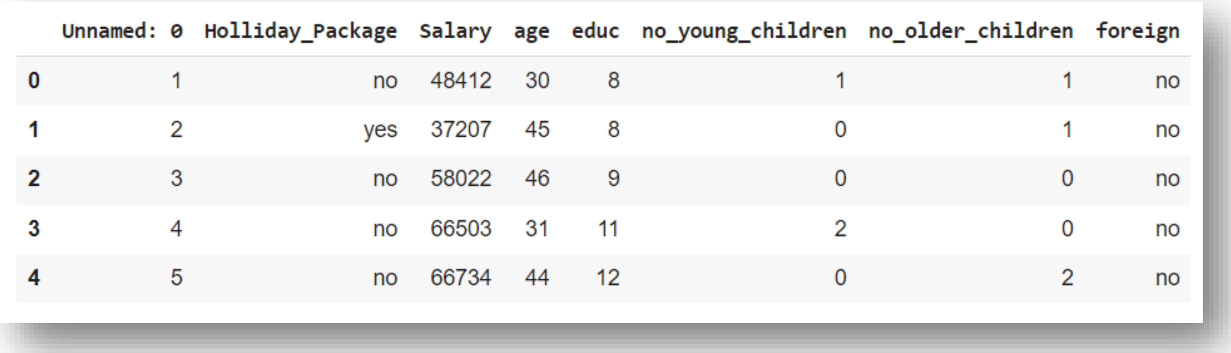
You are hired by a tour and travel agency which deals in selling holiday packages. You are provided details of 872 employees of a company. Among these employees, some opted for the package and some didn't. You have to help the company in predicting whether an employee will opt for the package or not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

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.

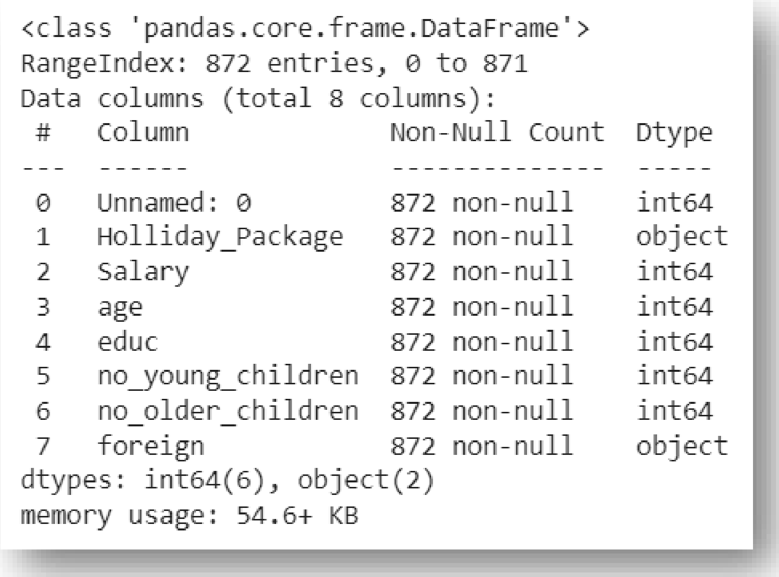
Table 1: Head of the dataset showing the first 5 records



	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

The dataset was loaded and the head of the dataset was checked. Table 1 shows the first 5 records of the dataset. From this table, we can see the different variables or columns of the dataset.

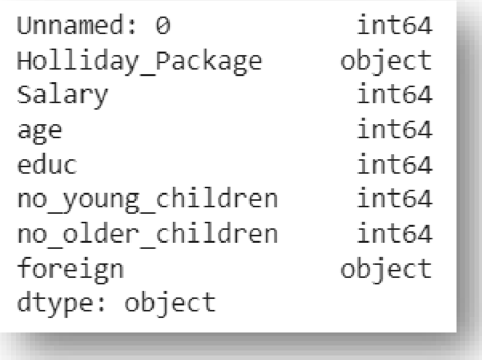
Table 2: Information of the dataset



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 872 entries, 0 to 871
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Unnamed: 0         872 non-null    int64 
 1   Holliday_Package   872 non-null    object
 2   Salary             872 non-null    int64 
 3   age                872 non-null    int64 
 4   educ               872 non-null    int64 
 5   no_young_children  872 non-null    int64 
 6   no_older_children  872 non-null    int64 
 7   foreign            872 non-null    object
dtypes: int64(6), object(2)
memory usage: 54.6+ KB
```

The dataset has 8 columns and 872 records which is seen in Table 2. There are 872 non-null records in all the 8 columns meaning there are no missing records based on this initial analysis that was done.

Table 3: Data type of the columns in the dataset

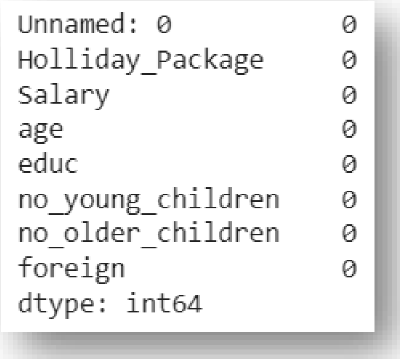


```
Unnamed: 0            int64
Holliday_Package     object
Salary                int64
age                   int64
educ                  int64
no_young_children     int64
no_older_children     int64
foreign              object
dtype: object
```

From Table 3**Error! Reference source not found.**, it is seen that the variable 'Holliday_Package' and 'foreign' is of object data type and integer for all the other variables. 'Unnamed: 0' is the serial number. Therefore, it is not considered as an independent variable. Hence there are 6 independent variables and one target variable – 'Holliday_Package'.

The shape of the data is (872, 8) meaning the dataset has 872 rows and 8 columns.

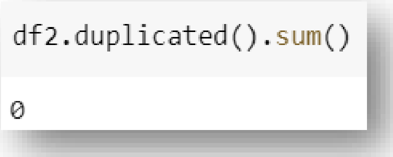
Table 4: Missing value of the columns in the dataset



```
Unnamed: 0          0
Holliday_Package    0
Salary              0
age                 0
educ                0
no_young_children   0
no_older_children   0
foreign             0
dtype: int64
```

The dataset was further checked for missing values and it is seen from Table 4 that there are no missing values in the dataset.

The 'Unnamed: 0' is basically the serial number and hence it is dropped as it may interfere with the exploratory data analysis. The dataset is now checked for duplicates.

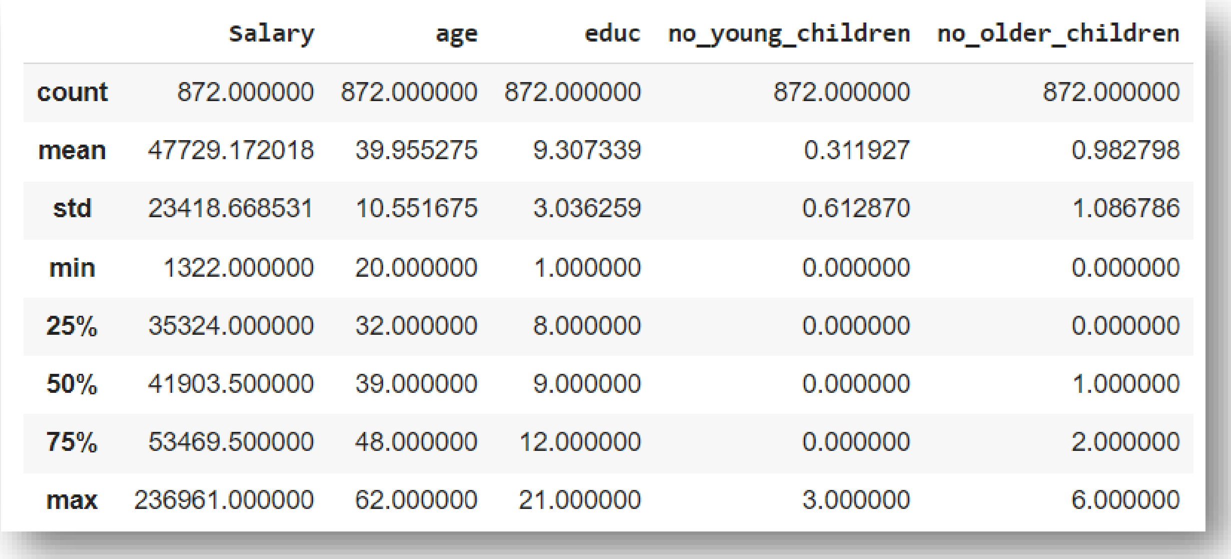


```
df2.duplicated().sum()

0
```

Figure 1: Duplicates in the dataset

The dataset is checked for duplicate values and it was found that there are no duplicates as seen in Figure 1.

Table 5: Description of the dataset



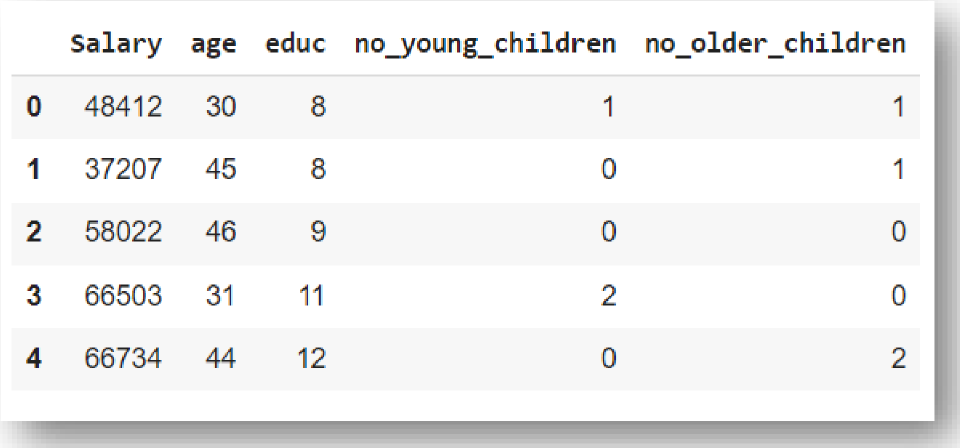
	Salary	age	educ	no_young_children	no_older_children
count	872.000000	872.000000	872.000000	872.000000	872.000000
mean	47729.172018	39.955275	9.307339	0.311927	0.982798
std	23418.668531	10.551675	3.036259	0.612870	1.086786
min	1322.000000	20.000000	1.000000	0.000000	0.000000
25%	35324.000000	32.000000	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

Table 5 shows the description or the summary of the numerical columns in the dataset after dropping the ‘Unnamed: 0’ column. The ‘Unnamed: 0’ is basically the serial number and hence it is dropped as it may interfere with the exploratory data analysis. By looking at Table 5, we are able to deduce that ‘Salary’ has the highest mean value while ‘no_young_children’ has the lowest mean value. ‘Salary’ has the highest standard deviation value while ‘no_young_children’ has the lowest standard deviation value. This is probably because salary and the number of young children is two different measurements. ‘Salary’ is the employee salary while ‘no_young_children’ is the number of young children (younger than 7 years).

Univariate Analysis:

Numerical variables:

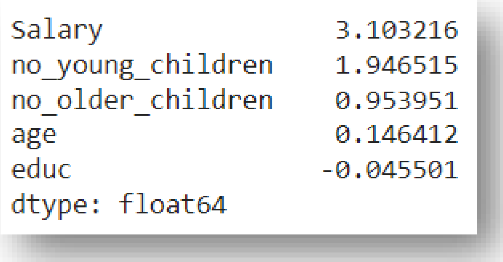
Table 6: Head of the numerical dataset showing the first 5 records



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

Table 6 shows the first 5 records of the numerical dataset. The numerical dataset was used to calculate the skewness, plot the univariate distribution and the boxplot.

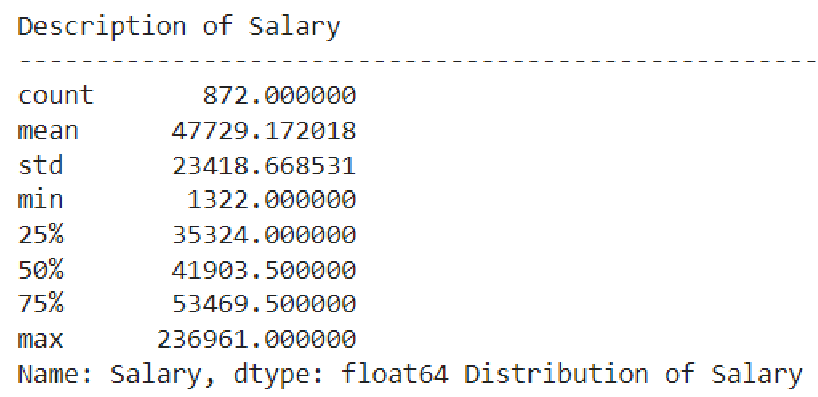
Table 7: Skewness of the variables of the dataset



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

1. **Salary:**

Table 8: Description of 'Salary'



```
Description of Salary
----------------------------------------------------
count         872.000000
mean        47729.172018
std         23418.668531
min          1322.000000
25%         35324.000000
50%         41903.500000
75%         53469.500000
max        236961.000000
Name: Salary, dtype: float64 Distribution of Salary
```

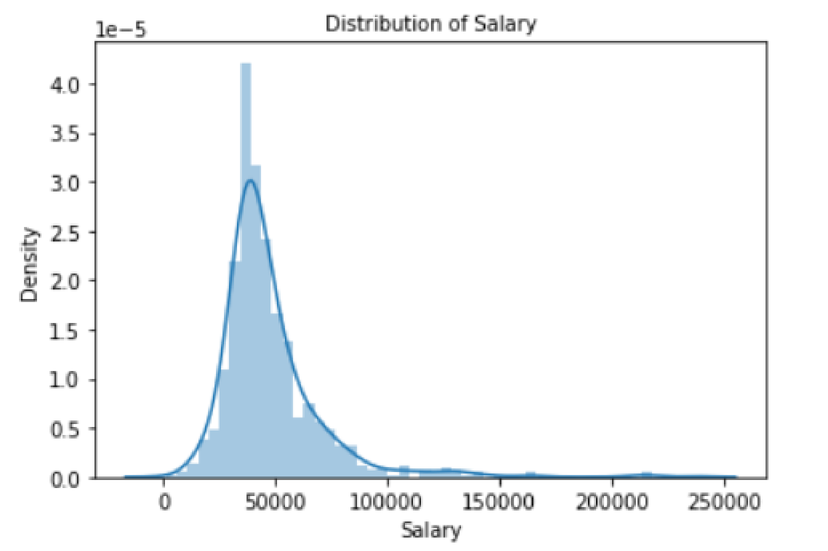


Figure 2: Univariate distribution of 'Salary'

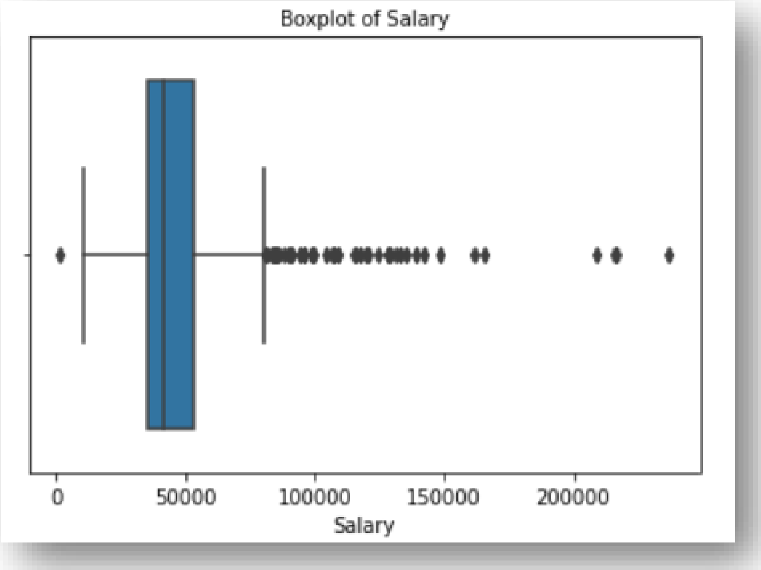


Figure 3: Boxplot showing the distribution of 'Salary'

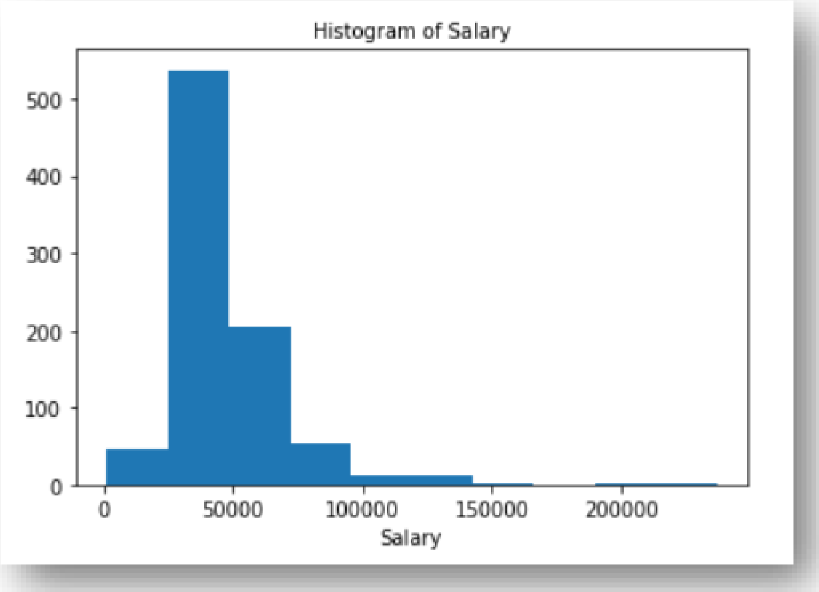
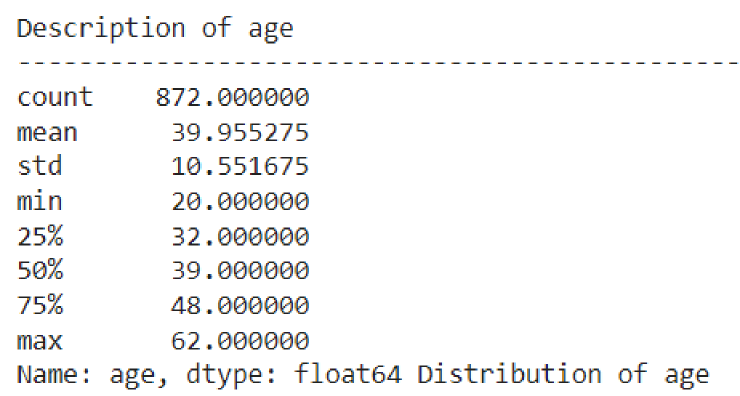


Figure 4: Histogram of 'Salary'

Univariate analysis of ‘Salary’ is done to understand the patterns and distribution of the data. From Figure 3, we can see that the Box plot of ‘Salary’ variable has many outliers. The distribution of the data is moderately right skewed which is seen in Figure 2. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘Salary’ variable is 3.103216. From Table 8, it is seen that the mean of the data is 47729.172018 meaning the average employee salary is 47729.17. The minimum salary is 1322 and the maximum salary is 236961.

2. Age:

Table 9: Description of 'Age'



```
Description of age
------------------------------------------------
count    872.000000
mean      39.955275
std       10.551675
min       20.000000
25%       32.000000
50%       39.000000
75%       48.000000
max       62.000000
Name: age, dtype: float64 Distribution of age
```

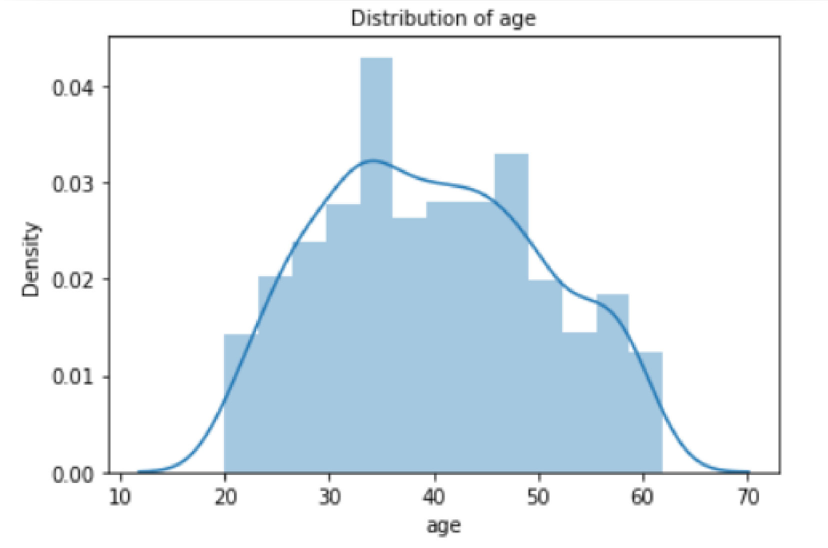


Figure 5: Univariate distribution of 'age'

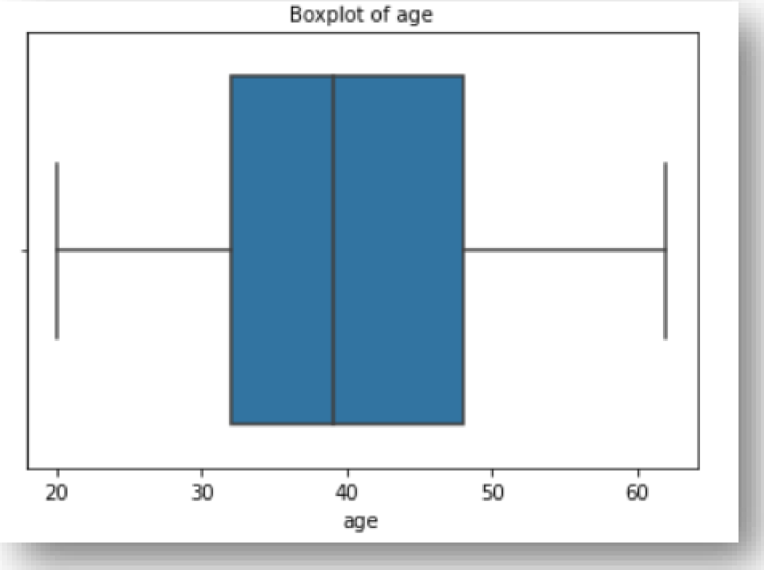


Figure 6: Boxplot showing the distribution of 'age'

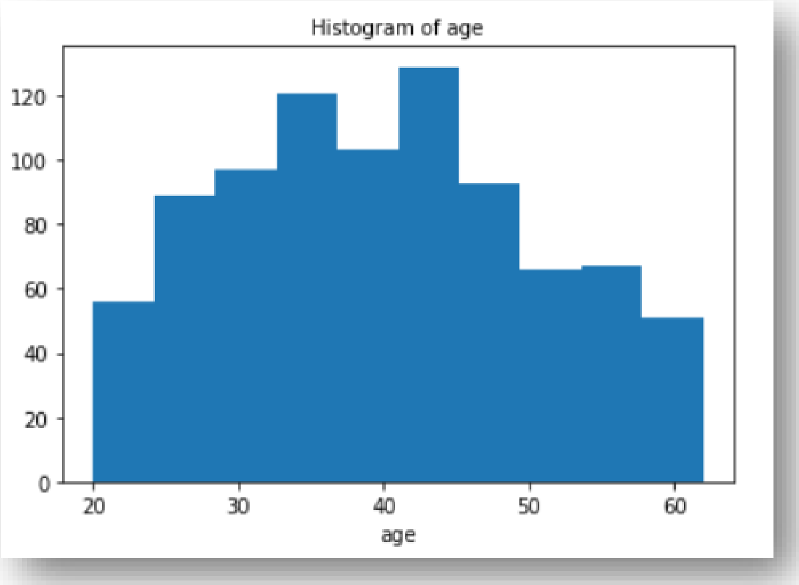
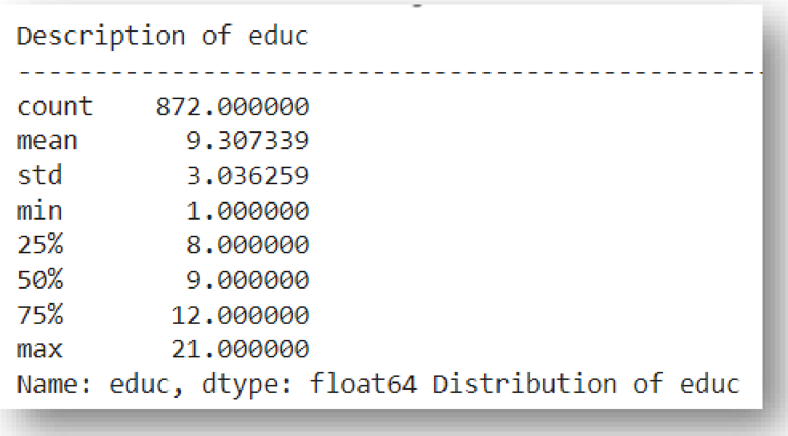


Figure 7: Histogram of 'age'

Univariate analysis of ‘age’ is done to understand the patterns and distribution of the data. From Figure 6, we can see that the Box plot of ‘age’ variable has no outliers. The distribution of the data is moderately right skewed which is seen in Figure 5. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘age’ variable is 0.146412. From Table 9, it is seen that the mean of the data is 39.955275 meaning the average age of the employee is 39.9 years. The minimum age is 20 and the maximum age is 62.

3. Educ:

Table 10: Description of 'educ'



```
Description of educ
-------------------------------------------------
count    872.000000
mean       9.307339
std        3.036259
min        1.000000
25%        8.000000
50%        9.000000
75%       12.000000
max       21.000000
Name: educ, dtype: float64 Distribution of educ
```

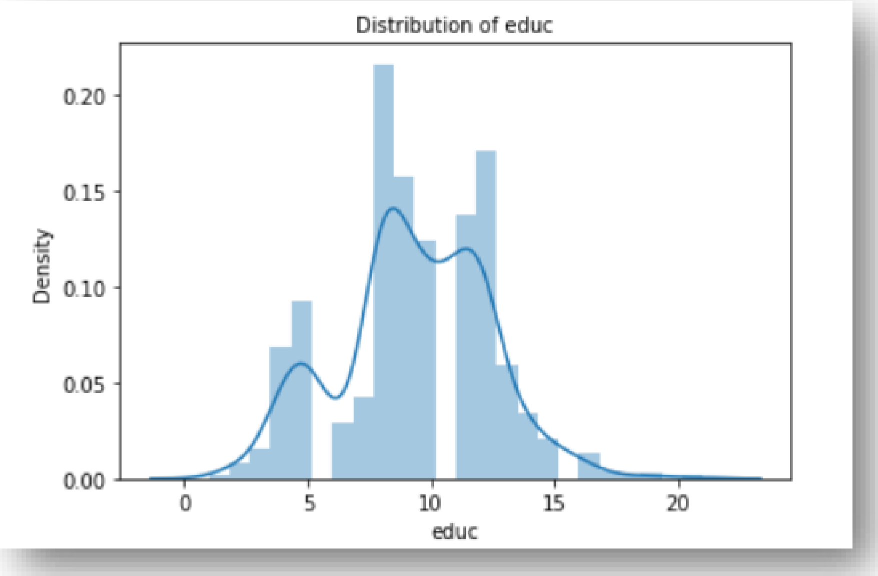


Figure 8: Univariate distribution of 'educ'

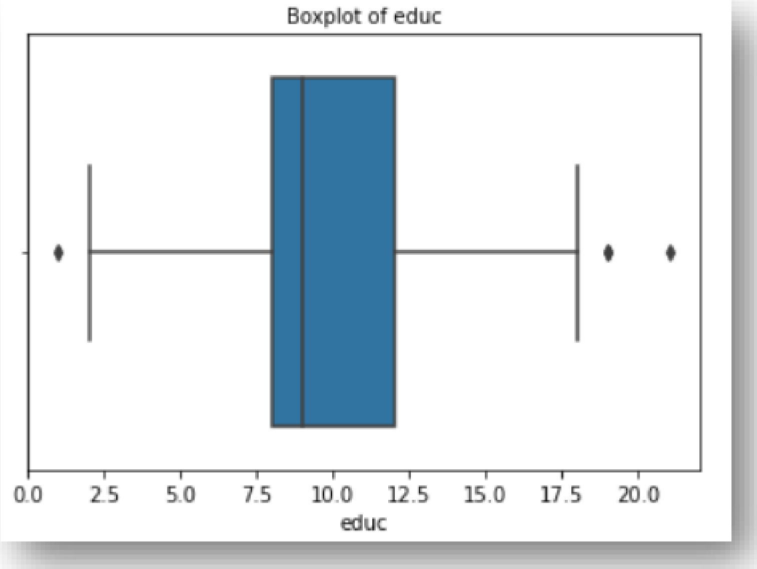


Figure 9: Boxplot showing the distribution of 'educ'

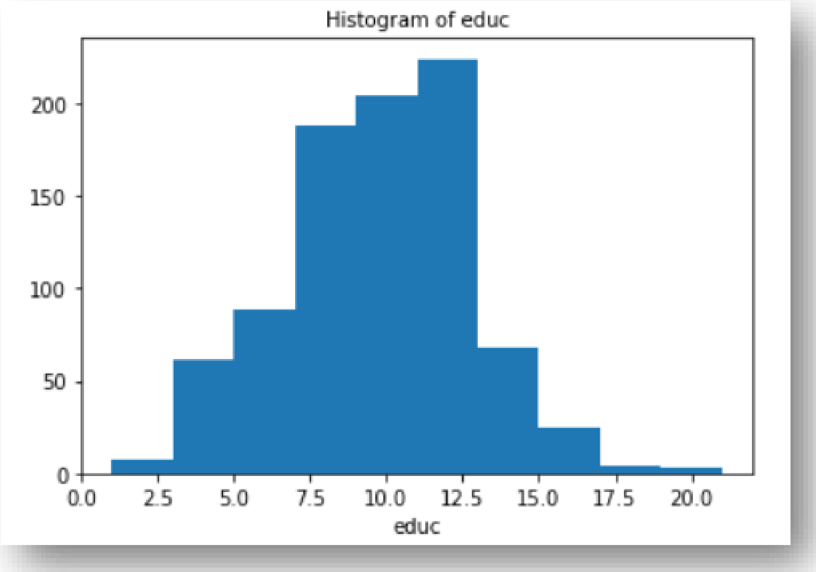
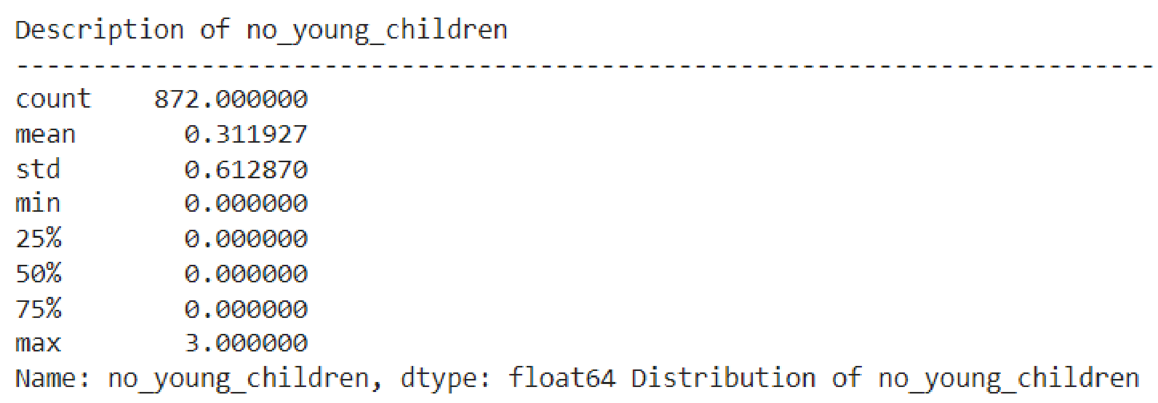


Figure 10: Histogram of 'educ'

Univariate analysis of ‘educ’ is done to understand the patterns and distribution of the data. From Figure 9, we can see that the Box plot of ‘educ’ variable has few outliers. The distribution of the data is moderately left skewed which is seen in Figure 8. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘educ’ variable is -0.045501. We can also observe from Figure 8 that the distribution of ‘educ’ variable has multiple modes. From Table 10, it is seen that the mean of the data is 9.307339 meaning the average years of formal education is 9.30 years. The minimum number of years of formal education is 1 and the maximum is 21.

4. no_young_children:

Table 11: Description of 'no_young_children'



```
Description of no_young_children
-----------------------------------------------------------------------------
count     872.000000
mean        0.311927
std         0.612870
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max         3.000000
Name: no_young_children, dtype: float64 Distribution of no_young_children
```

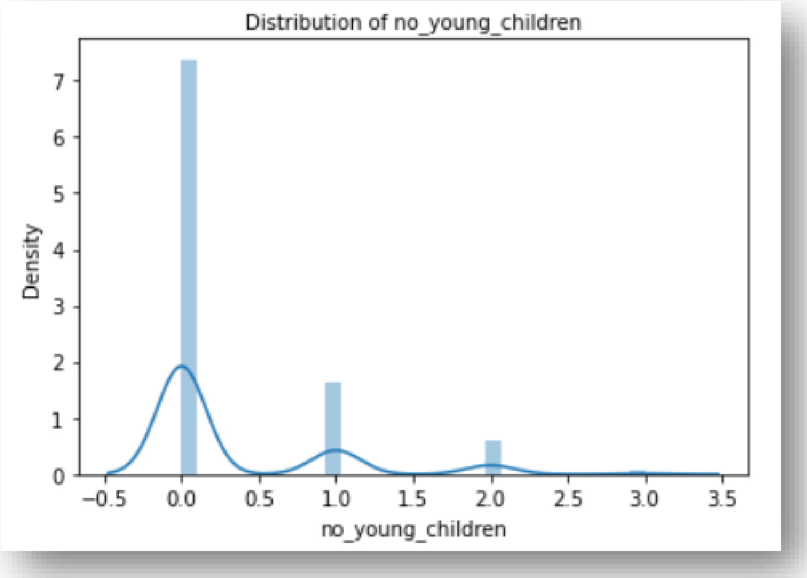


Figure 11: Univariate distribution of 'no_young_children'

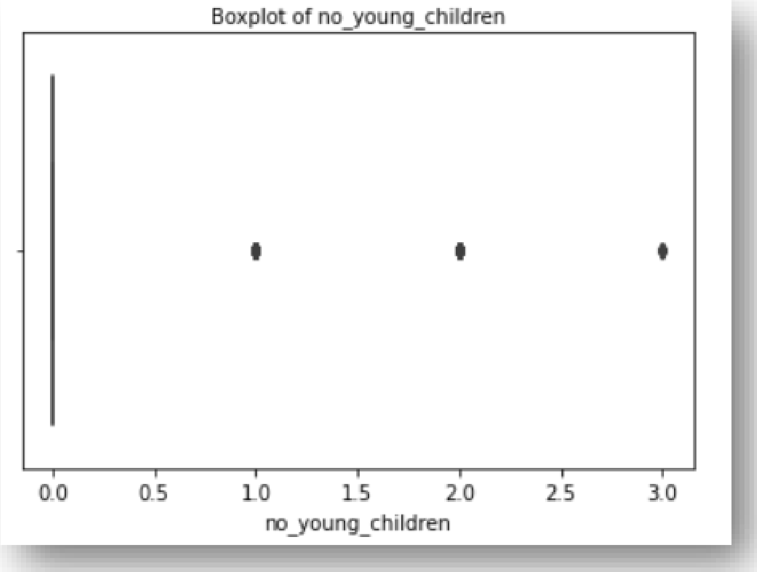


Figure 12: Boxplot showing the distribution of 'no_young_children'

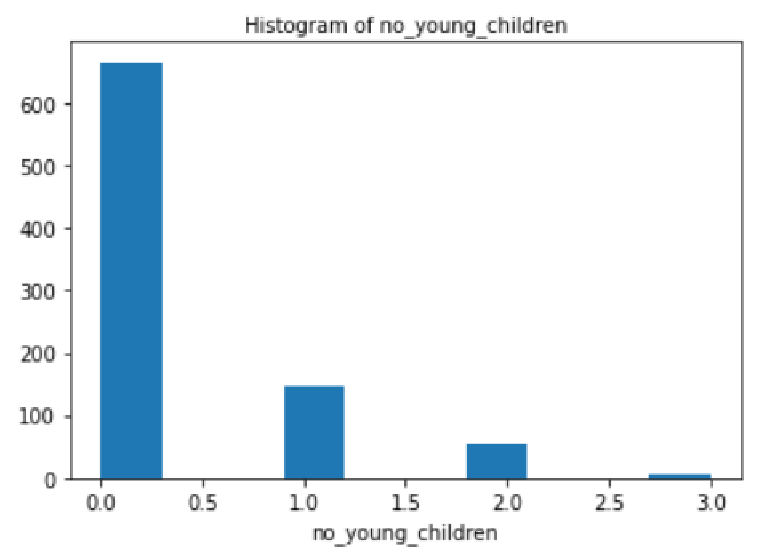
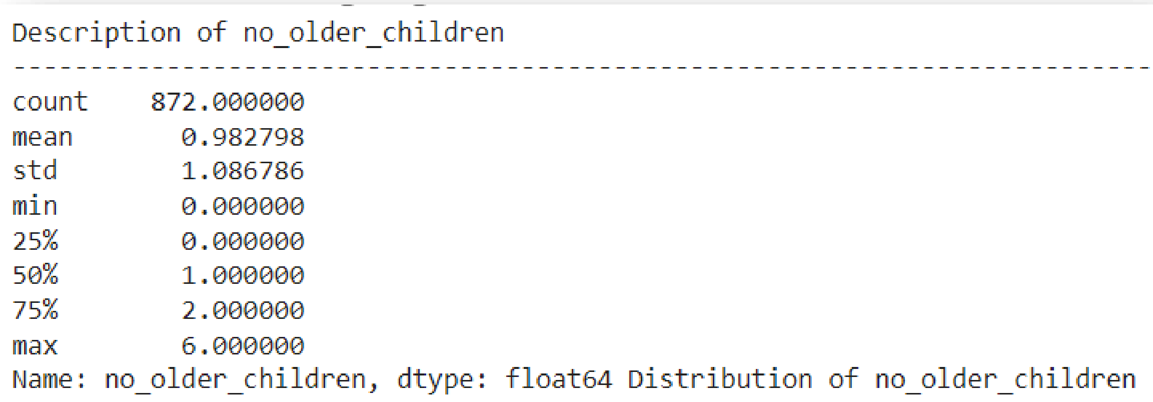


Figure 13: Histogram of 'no_young_children'

Univariate analysis of ‘no_young_children’ is done to understand the patterns and distribution of the data. From Figure 12, we can see that the Box plot of ‘no_young_children’ variable has few outliers. The distribution of the data is moderately right skewed which is seen in Figure 11. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘no_young_children’ variable is 1.946515. We can also observe from Figure 11 that the distribution of ‘no_young_children’ variable has multiple modes. From Table 11, it is seen that the mean of the data is 0.311927 meaning the average number of young children (younger than 7 years) is 0.31. The minimum number of young children is 0 and the maximum is 3.

5. no_older_children:

Table 12: Description of 'no_older_children'



```
Description of no_older_children
----------------------------------------------------------------------------
count    872.000000
mean       0.982798
std        1.086786
min        0.000000
25%        0.000000
50%        1.000000
75%        2.000000
max        6.000000
Name: no_older_children, dtype: float64 Distribution of no_older_children
```

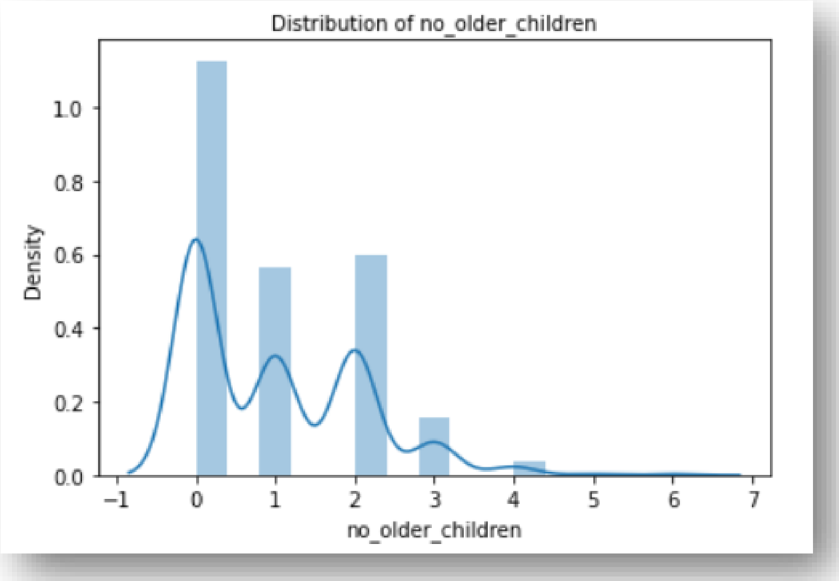


Figure 14: Univariate distribution of 'no_older_children'

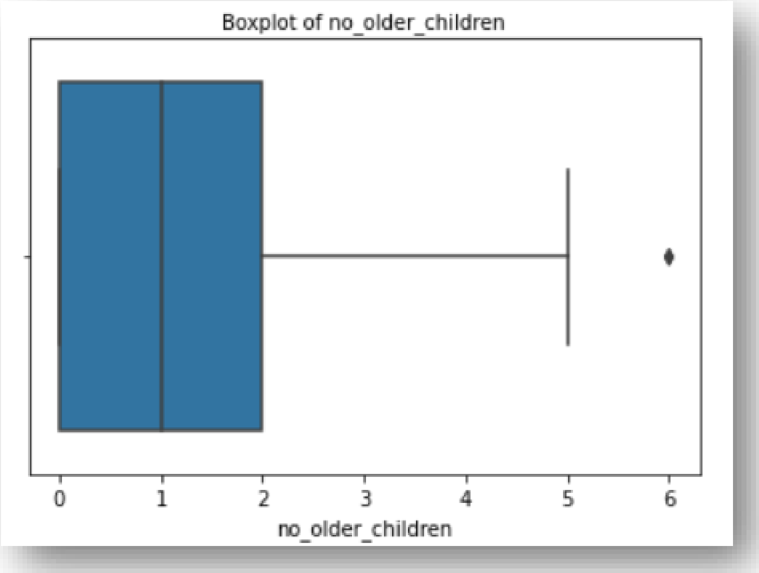


Figure 15: Boxplot showing the distribution of 'no_older_children'

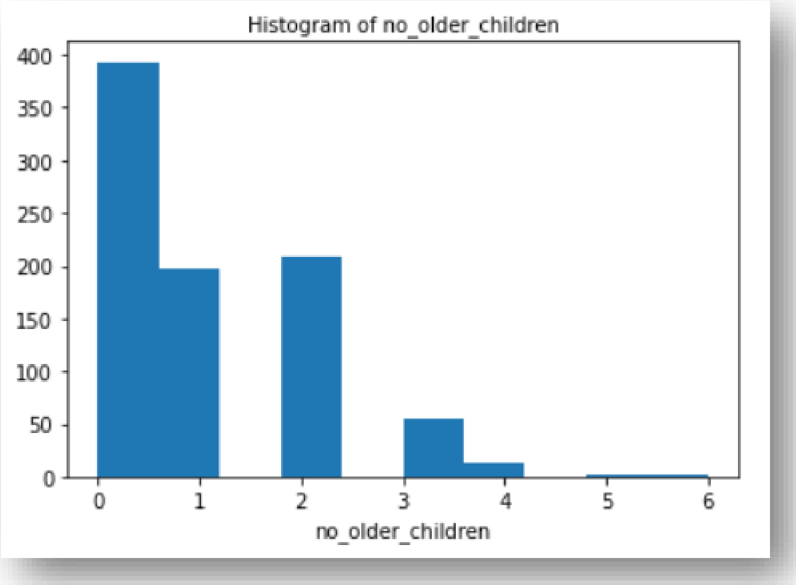


Figure 16: Histogram of 'no_older_children'

Univariate analysis of ‘no_older_children’ is done to understand the patterns and distribution of the data. From Figure 15,we can see that the Box plot of ‘no_older_children’ variable has few outliers. The distribution of the data is moderately right skewed which is seen in Figure 14. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘no_older_children’ variable is `0.953951`. We can also observe from Figure 14 that the distribution of ‘no_older_children’ variable has multiple modes. From Table 12, it is seen that the mean of the data is `0.982798` meaning the average number of older children is 0.98. The minimum number of older children is 0 and the maximum is 6.

Categorical variables:

6. **Holliday_Package:**

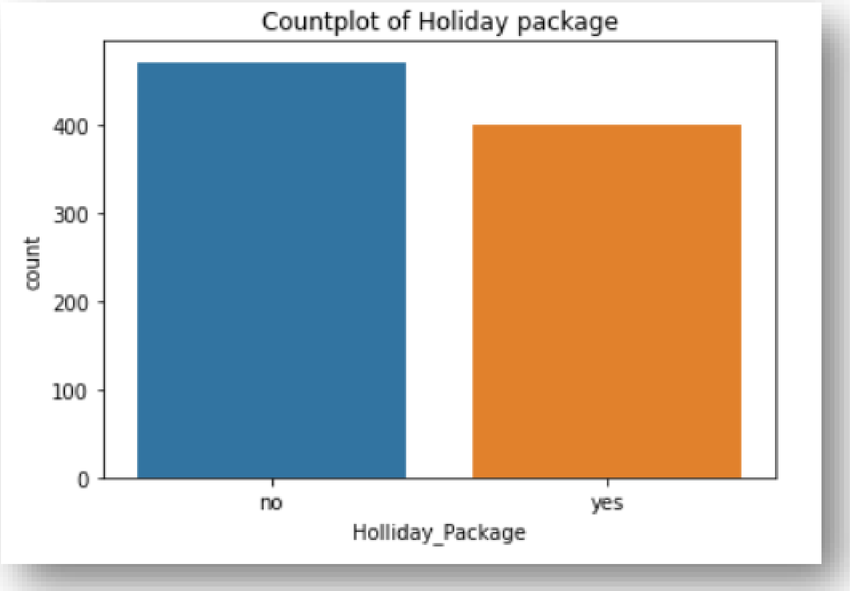


Figure 17: Countplot of 'Holiday_Package'

Univariate analysis of 'Holliday_Package' is done to understand the patterns and distribution of the data. From Figure 17, we can interpret that majority of the employees have not opted for the holiday package.

7. **Foreign:**

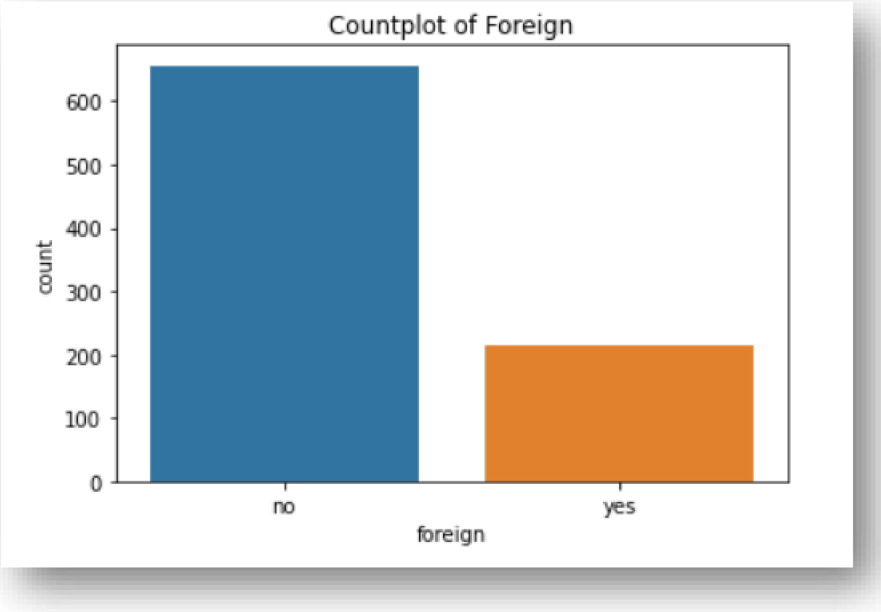


Figure 18: Countplot of 'foreign'

Univariate analysis of ‘foreign’ is done to understand the patterns and distribution of the data. From Figure 18, we can interpret that majority of the employees are not foreigners.

Bivariate Analysis:

Numerical variables:

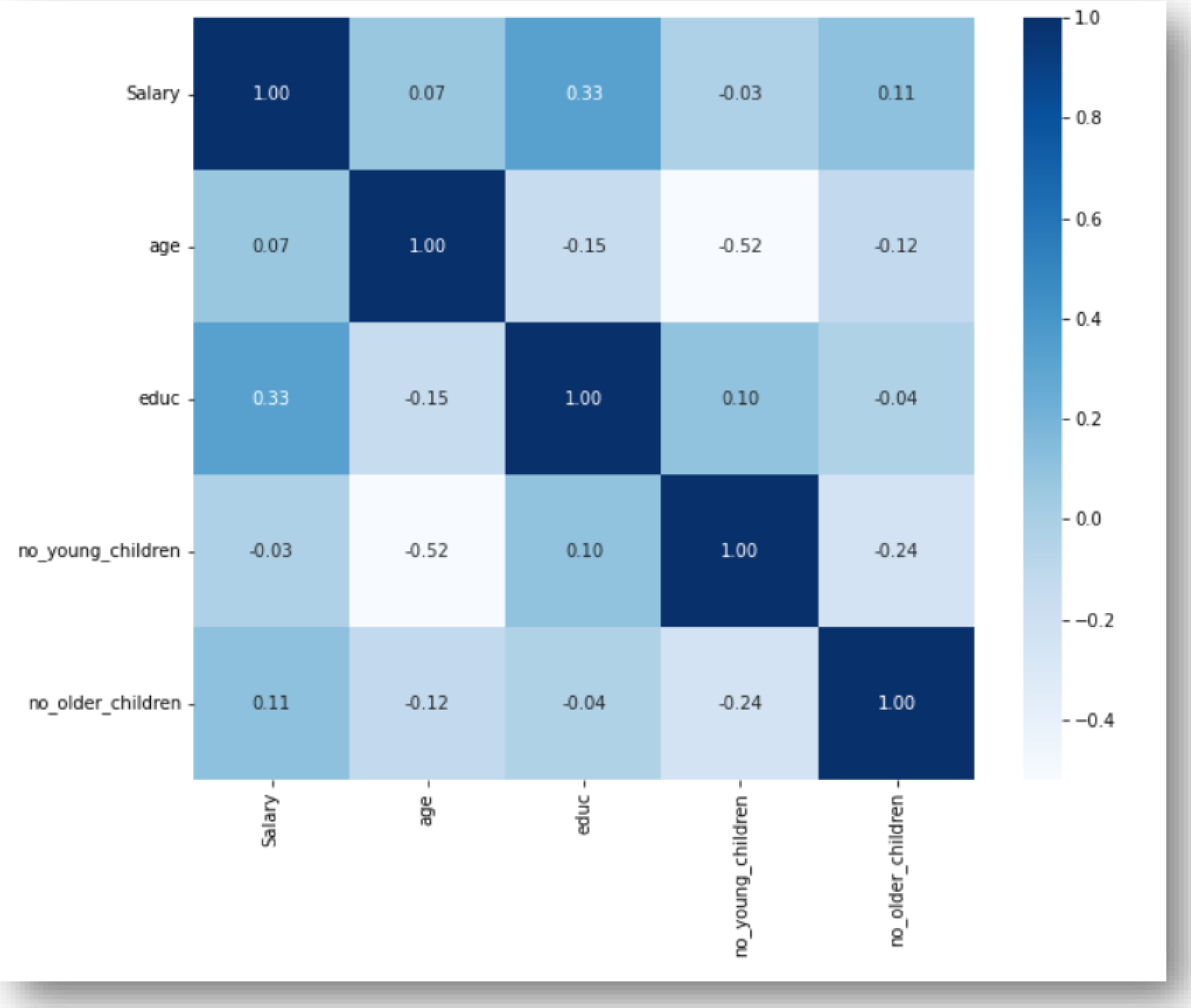


Figure 19: Heat map showing the bivariate analysis of the dataset

Bivariate analysis is done using the help of a heat map. A heat map is used to understand the correlation between two numerical values in a dataset. Figure 19 shows the heatmap of the dataset.

Observations:

- There is no strong correlation between any variables in the dataset
- Salary and education have a moderate correlation
- Salary and no_older_children also have a moderate correlation

Categorical variables:

1. **Foreign:**

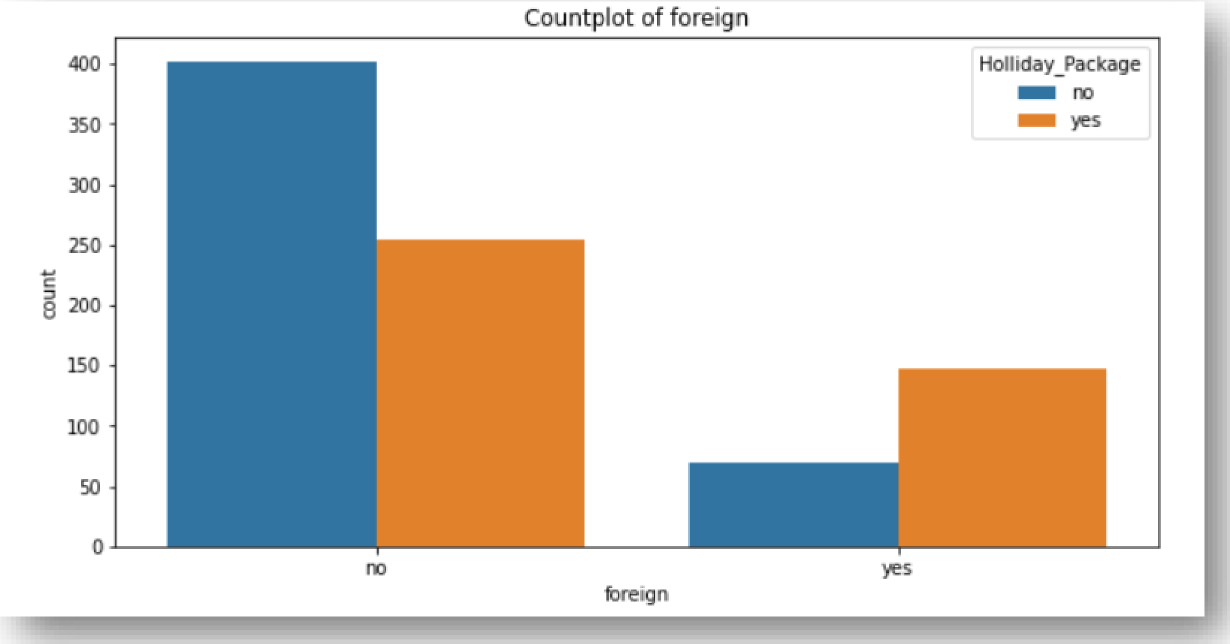


Figure 20: Countplot of 'foreign' and 'Holliday_Package'

From the **Error! Reference source not found.**, we can interpret that majority of the employees are n ot foreigners and this is also true for employees who have not opted for the holiday package.

Multivariate Analysis:

Numerical variables:

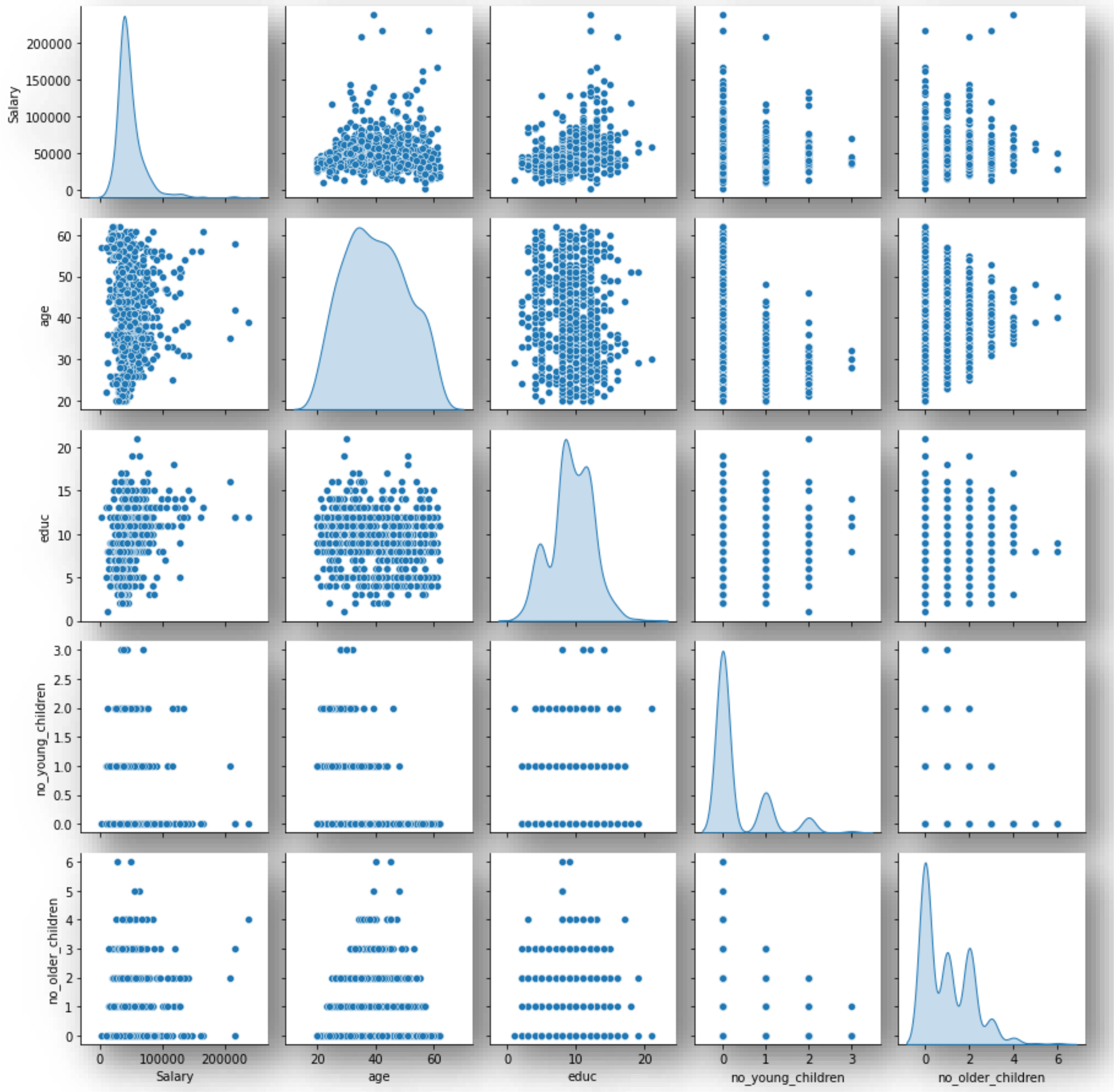


Figure 21: Pair plot showing the multivariate analysis of the numerical variables in the dataset

Multivariate analysis is done using the help of a pair plot to understand the relationship between all the numerical values in the dataset. Pair plot can be used to compare all the variables with each other to understand the patterns or trends in the dataset. Figure 21 shows the pair plot of the dataset.

Observations:

From Figure 19 and Figure 21we can observe that there are no strong correlations between the numerical variables in the dataset.

Categorical variables:

1. **Foreign:**

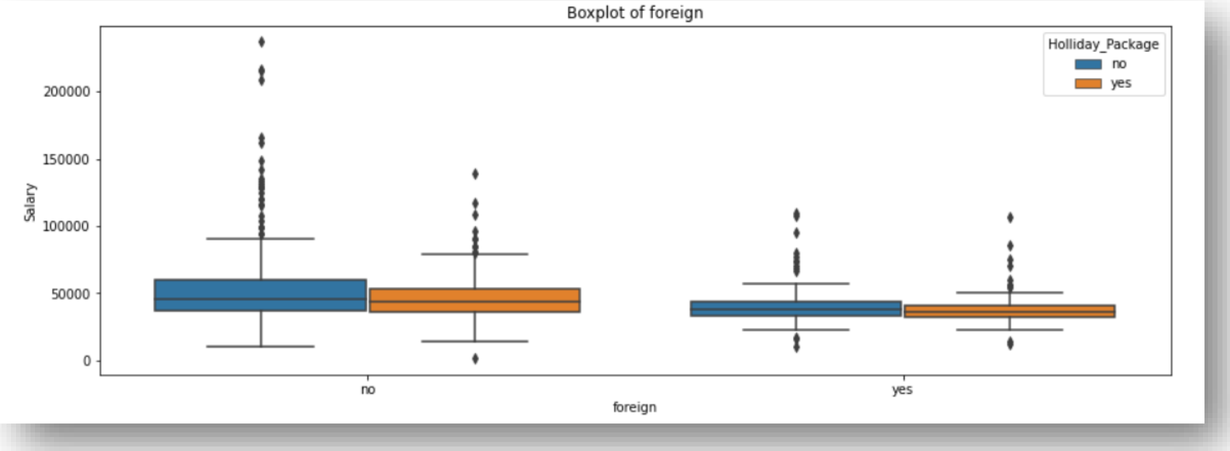
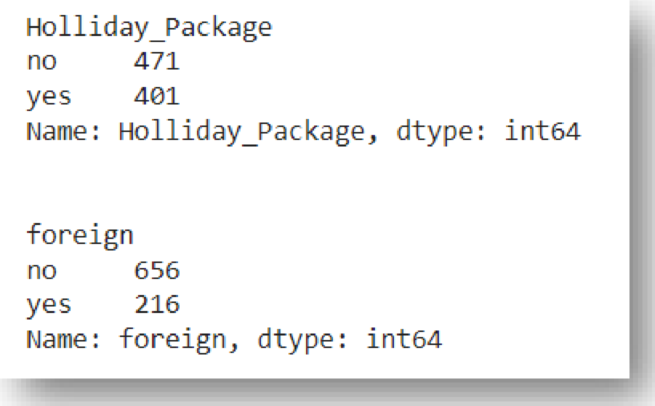


Figure 22: Boxplot of 'foreign'

From Figure 22, we can see that the Box plot of ‘foreign’ variable has many outliers.

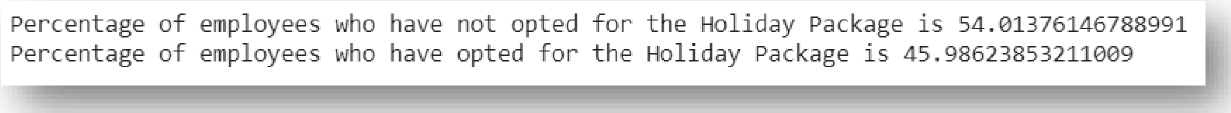
2.2 Do not scale the data. Encode the data (having string values) for Modelling. Data Split: Split the data into train and test (70:30). Apply Logistic Regression and LDA (linear discriminant analysis).



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


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

Figure 23: Value counts of the categorical variables in the dataset

There are 2 categorical variables in this dataset – ‘Holliday_Package’ and ‘foreign’. Figure 23 shows the different categorical variables and the value counts for each of the types in the different categories.



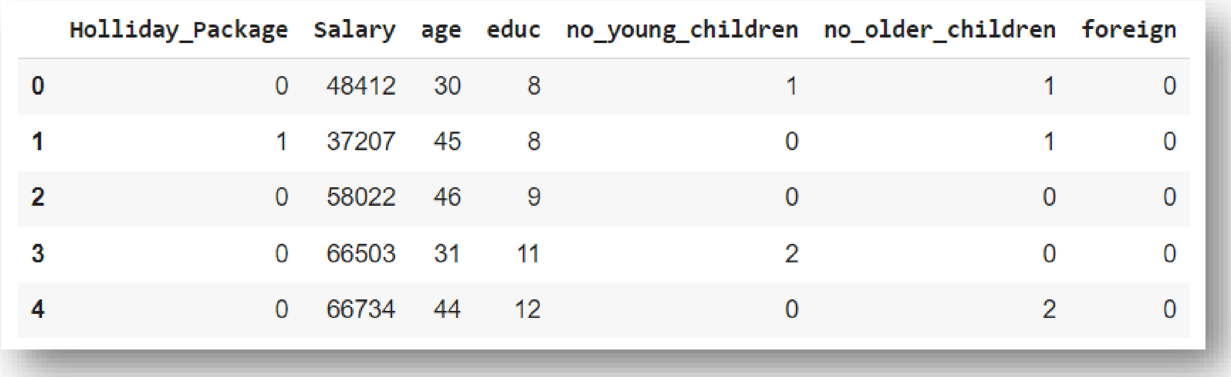
```
Percentage of employees who have not opted for the Holiday Package is 54.01376146788991
Percentage of employees who have opted for the Holiday Package is 45.98623853211009
```

Figure 24: Split of dependent variable data

From Figure 24 it is seen that the data split is not good. The data split is 54.0% for the employees who have not opted for the holiday package and 45.9% for the employees who have opted for the holiday package. Therefore, it is likely that the model will give poor results based on this data.

Logistic Regression and Linear Discriminant Analysis requires all the columns to be numerical. Hence these categorical columns have to be changed into numerical columns. **Label encoding method** can be used to convert the categorical columns to numerical columns.

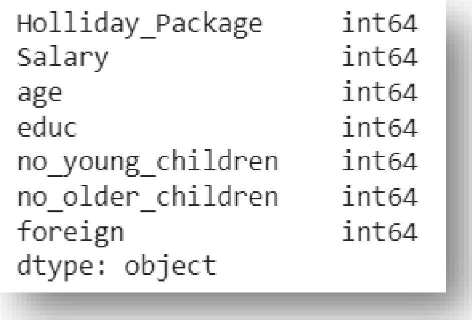
Table 13: Head of the dataset after doing label encoding



	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

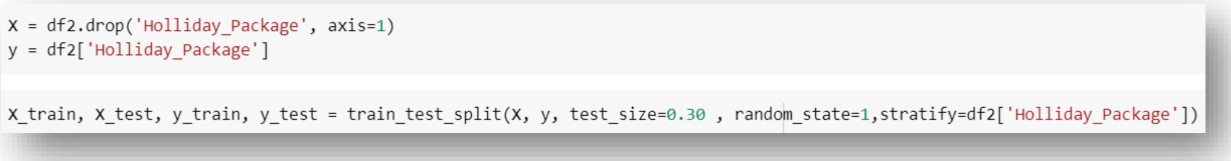
Table 13 shows the head of the dataset after performing label encoding.

Table 14: Data type after label encoding



```
Holliday_Package     int64
Salary               int64
age                  int64
educ                 int64
no_young_children    int64
no_older_children    int64
foreign              int64
dtype: object
```

Table 14 shows that all the variables are now numerical variables.

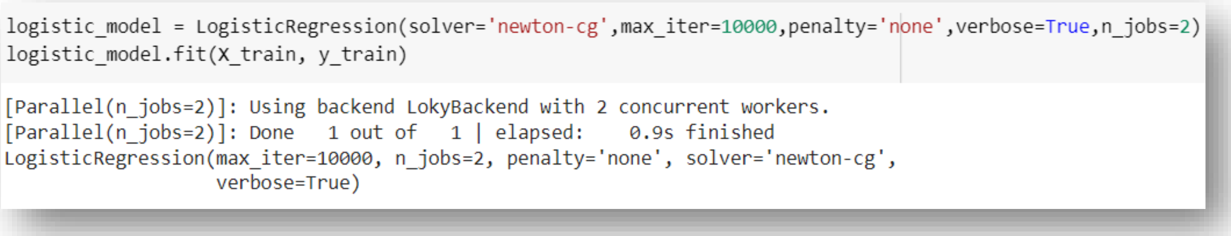


```
X = df2.drop('Holliday_Package', axis=1)
y = df2['Holliday_Package']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30 , random_state=1,stratify=df2['Holliday_Package'])
```

Figure 25: Splitting the data into train and test

The data is first split into train and test data as shown in Figure 25 before building the logistic regression model and linear discriminant analysis model. There is no fixed rule for separation training and testing data sets. Most of the researchers use **70:30 ratio** for separation data sets. The same ratio was used in this dataset to split the data into train and test. The random state was set to be 1.

Logistic Regression

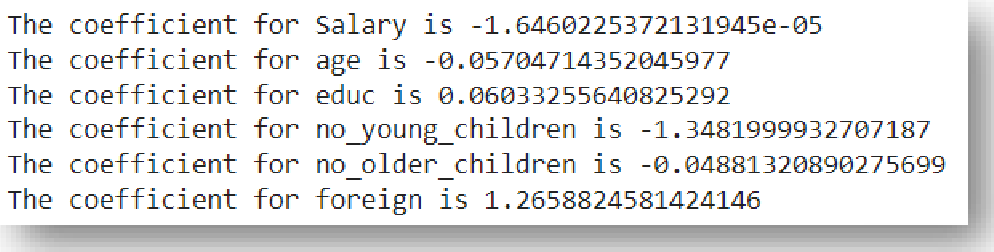


```
logistic_model = LogisticRegression(solver='newton-cg',max_iter=10000,penalty='none',verbose=True,n_jobs=2)
logistic_model.fit(X_train, y_train)

[Parallel(n_jobs=2)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=2)]: Done   1 out of   1 | elapsed:    0.9s finished
LogisticRegression(max_iter=10000, n_jobs=2, penalty='none', solver='newton-cg',
                   verbose=True)
```

Figure 26: Parameters for Logistic Regression Model

Logistic Regression model is supervised learning algorithm which can be used for classification type of problems. It establishes relation between dependent class variable and independent variables using regression. Values were passed for the different parameters as shown in Figure 26.

- The **‘solver’** is the algorithm used in the process of backpropagation to calculate the weights of the coefficients in the logistic regression model. The ‘solver’ used is **‘newton-cg’** solver.
- The **‘max_iter’** parameter is the maximum number of iterations to update the synaptic weights of neurons. The ‘max_iter’ value used is **10000**.
- Penalized logistic regression imposes a penalty to the logistic model for having too many variables. The **‘penalty’** is **‘none’** meaning no penalty is added.

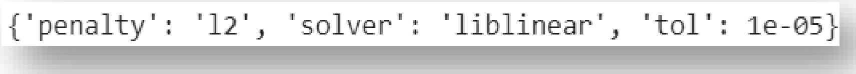


```
The coefficient for Salary is -1.6460225372131945e-05
The coefficient for age is -0.05704714352045977
The coefficient for educ is 0.06033255640825292
The coefficient for no_young_children is -1.3481999932707187
The coefficient for no_older_children is -0.04881320890275699
The coefficient for foreign is 1.2658824581424146
```

Figure 27: Coefficient values for each column - Logistic Regression

There are 6 attributes in this dataset and hence there are 6 coefficients. For every one-unit change in x (the different independent variables), y (the dependent variable) changes m (the coefficient value) times. The coefficient values for each column are shown in Figure 27.

Logistic Regression using GridSearchCV

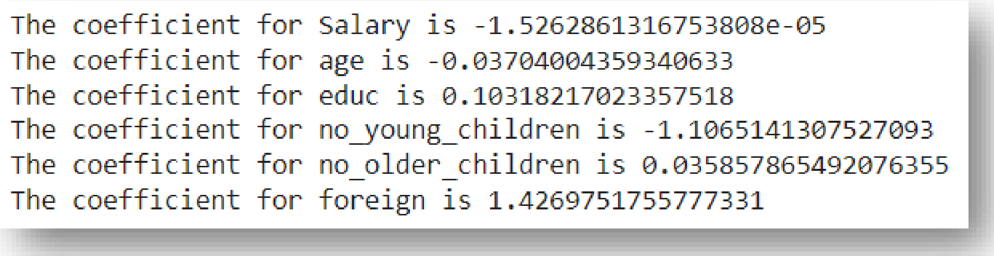


```
{'penalty': 'l2', 'solver': 'liblinear', 'tol': 1e-05}
```

Figure 28: Best parameters for Logistic Regression Model using GridSearchCV

Logistic Regression model was performed using the GridSeachCV. Many combinations of parameters were tried using Grid Search Cross Validation or the Grid Search CV function. Multiple values were passed for the different parameters. Figure 28 shows the best parameters for logistic regression model.

- The **'solver'** is the algorithm used in the process of backpropagation to calculate the weights of the coefficients in the logistic regression model. The optimum 'solver' was found to be **'liblinear'** solver.
- Penalized logistic regression imposes a penalty to the logistic model for having too many variables. The **'penalty'** is **'l2'.**
- The **'tol'** parameter is the threshold level. The lower the threshold, higher the accuracy and lesser the number of times the model will execute and vice versa. The optimum threshold value was found to be **'0.00001'**.

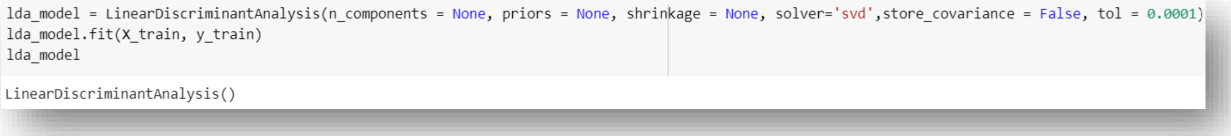


```
The coefficient for Salary is -1.5262861316753808e-05
The coefficient for age is -0.03704004359340633
The coefficient for educ is 0.10318217023357518
The coefficient for no_young_children is -1.1065141307527093
The coefficient for no_older_children is 0.035857865492076355
The coefficient for foreign is 1.4269751755777331
```

Figure 29: Coefficient values for each column - Logistic Regression using GridSearchCV

There are 6 attributes in this dataset and hence there are 6 coefficients. For every one-unit change in x (the different independent variables), y (the dependent variable) changes m (the coefficient value) times. The coefficient values for each column are shown in Figure 29.

Linear Discriminant Analysis

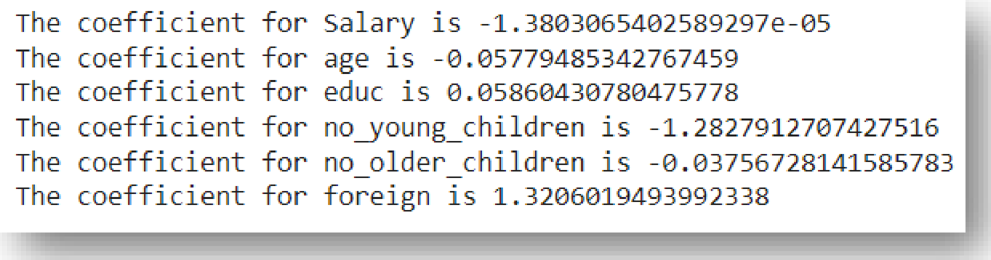


```
lda_model = LinearDiscriminantAnalysis(n_components = None, priors = None, shrinkage = None, solver='svd',store_covariance = False, tol = 0.0001)
lda_model.fit(X_train, y_train)
lda_model

LinearDiscriminantAnalysis()
```

Figure 30: Parameters for Linear Discriminant Analysis Model

Linear Discriminant Analysis (LDA) uses linear combinations of independent variables to predict the class in the response variable of a given observation. LDA assumes that the independent variables (p) are normally distributed and there is equal variance / covariance for the classes. LDA is popular, because it can be used for both classification and dimensionality reduction. Values were passed for the different parameters as shown in Figure 30.

- The **'solver'** is the algorithm used in the process of backpropagation to calculate the weights of the coefficients in the logistic regression model. The 'solver' used is **'svd'** solver.
- The '**n_components'** or the number of components is for dimensionality reduction. The 'n_components' is **'None'** meaning it will be set to minimum (n_classes - 1, n_features).
- The **'prior'** is the data proportion as prior probability is 54.0% and 45.9% as shown in Figure 24. 'prior' is **'None'** meaning the prior probability 54.0% and 45.9% is used.
- The **'tol'** parameter is the threshold level. The lower the threshold, higher the accuracy and lesser the number of times the model will execute and vice versa. The threshold value used was **'0.0001'**.



```
The coefficient for Salary is -1.3803065402589297e-05
The coefficient for age is -0.05779485342767459
The coefficient for educ is 0.05860430780475778
The coefficient for no_young_children is -1.2827912707427516
The coefficient for no_older_children is -0.03756728141585783
The coefficient for foreign is 1.3206019493992338
```

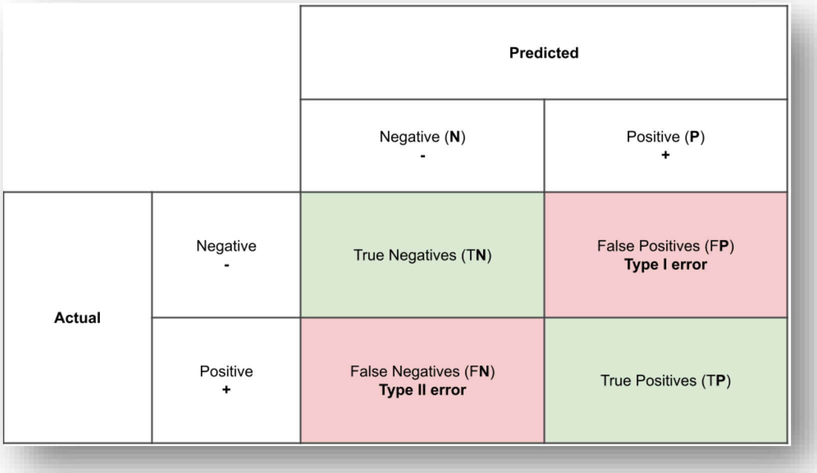
Figure 31: Coefficient values for each column - Linear Discriminant Analysis

There are 6 attributes in this dataset and hence there are 6 coefficients. For every one-unit change in x (the different independent variables), y (the dependent variable) changes m (the coefficient value) times. The coefficient values for each column are shown in Figure 31.

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.

Model Performance Metrics:

Confusion matrix:

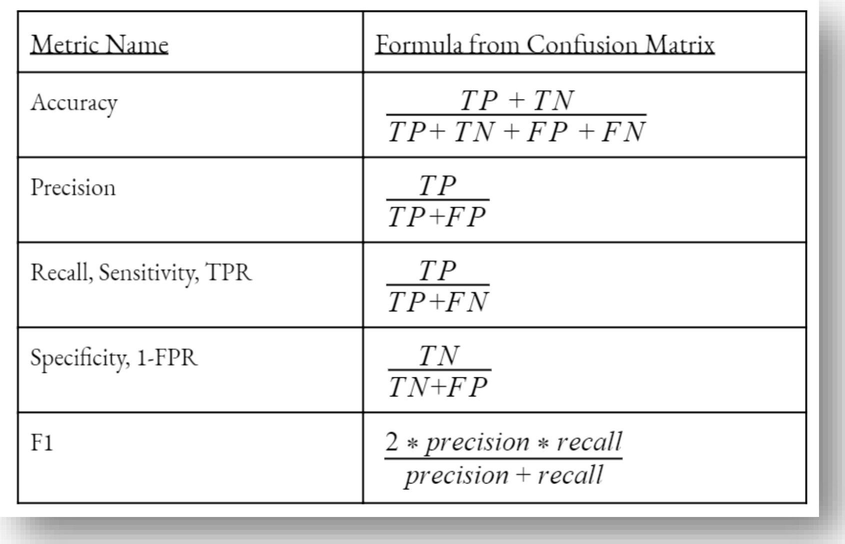


		Predicted	
		Negative (**N**) **-**	Positive (**P**) **+**
Actual	Negative **-**	True Negatives (**TN**)	False Positives (**FP**) **Type I error**
	Positive **+**	False Negatives (**FN**) **Type II error**	True Positives (**TP**)

Figure 32: Confusion Matrix

A confusion matrix is a 2x2 tabular structure reflecting the performance of the model in four blocks. Figure 32 shows how a confusion matrix will look like. True Positive (TP) and True Negative (TN) are correct predictions. False Positive (FP) and False Negative (FN) are incorrect predictions.

Table 15: Confusion Matrix formulas



Metric Name	Formula from Confusion Matrix
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$
Recall, Sensitivity, TPR	$\frac{TP}{TP + FN}$
Specificity, 1-FPR	$\frac{TN}{TN + FP}$
F1	$\frac{2 * precision * recall}{precision + recall}$

- **Accuracy** – it is a measure of how accurately or cleanly the model classifies the data points. Lesser the false predictions, more the accuracy. In a classification problem, the best score is **100%** accuracy.
- **Precision** – it is a measure of how many among the points identified as positive by the model, are really positive. A precision score of 1.0 means that all the points are identified as positive by the model. **1.0** is a perfect precision score. However, it does not indicate about the number of observations that were not labelled correctly.
- **Recall or sensitivity** – is the ratio of correctly predicted positive observations to the all observations in actual class. A perfect recall score is **1.0** but a recall score above 0.5 is considered as a good recall score. However, the recall score does not indicate about how many observations are incorrectly predicted.
- **Specificity** – is a measure of how many of the actual negative data points are identified as negative by the model.
- **F1** – it is the measure of the model's accuracy on a dataset. The F-score is a way of combining the precision and recall of the model and it is defined as the harmonic mean of the model's precision and recall. An F1 score is considered perfect when it's **1**, while the model is a total failure when the score is 0.

All these can be calculated from the confusion matrix by using the formulas given in Table 15.

Classification report:

A Classification report is used to measure the quality of predictions from a classification algorithm. The classification report shows the main classification metrics and their scores. The metrics are precision, recall, f1-score and accuracy for the actual and predicted data. The metrics are calculated by using true and false positives, true and false negatives from the confusion matrix.

ROC Curve:

Receiver Operating Characteristics (ROC) Curve is a technique for visualizing classifier performance. It is a graph between true positive (TP) rate and false positive (FP) rate.

$$\text{TP rate} = \frac{TP}{total\, positive}$$

$$\text{FP rate} = \frac{FP}{total\, negative}$$

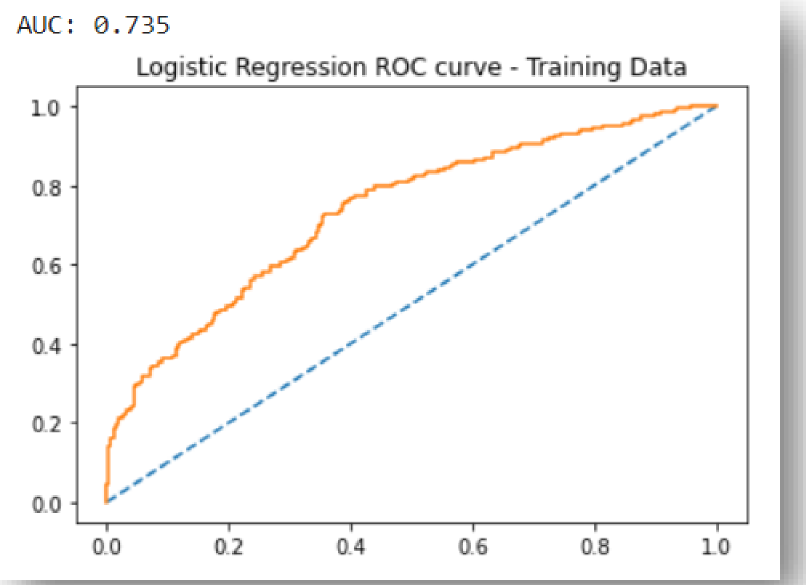
ROC graph is a trade-off between benefits (TP) and costs (FP). The steeper the ROC Curve, the stronger the model will be and vice versa.

ROC_AUC score:

Area under the ROC Curve (AUC) is the measure of the area under the ROC Curve. The ROC_AUC score gives us the value of the area under the ROC Curve. The larger the area under the curve, the better the model.

Logistic Regression

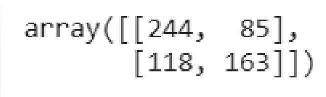
Training Data



```
AUC: 0.735
```

Figure 33: Logistic Regression ROC Curve for Training Data

The ROC_AUC score for the training data was calculated to be 0.735. The ROC curve was plotted and is shown in Figure 33.



```
array([[244,  85],
       [118, 163]])
```

Figure 34: Logistic Regression Confusion Matrix for Training Data

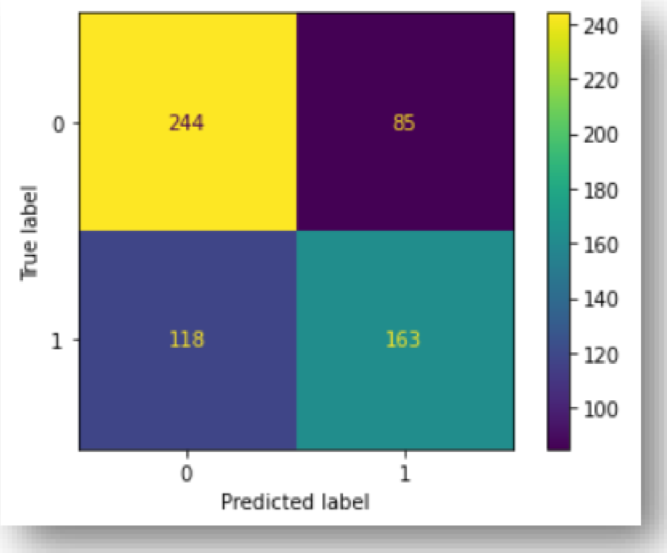
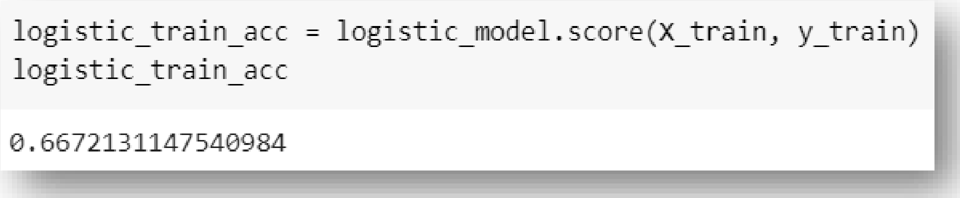


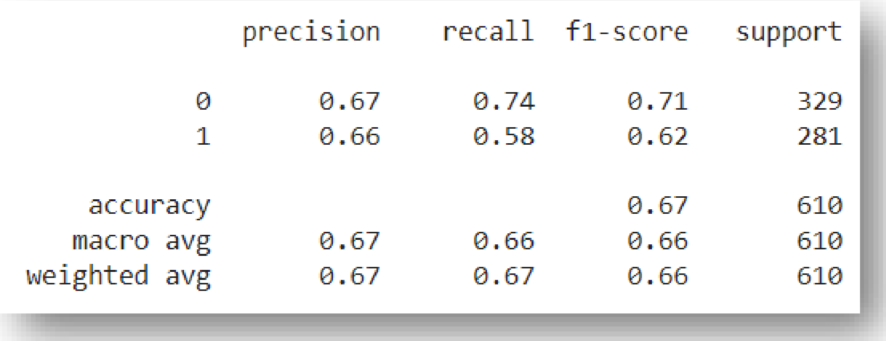
Figure 35: Plot of Logistic Regression Confusion Matrix for Training Data



```
logistic_train_acc = logistic_model.score(X_train, y_train)
logistic_train_acc

0.6672131147540984
```

Figure 36: Logistic Regression Accuracy score for Training Data

The confusion matrix was calculated and shown in Figure 34 and Figure 35. The accuracy score for the training data was found to be 0.67 which is shown in Figure 36.



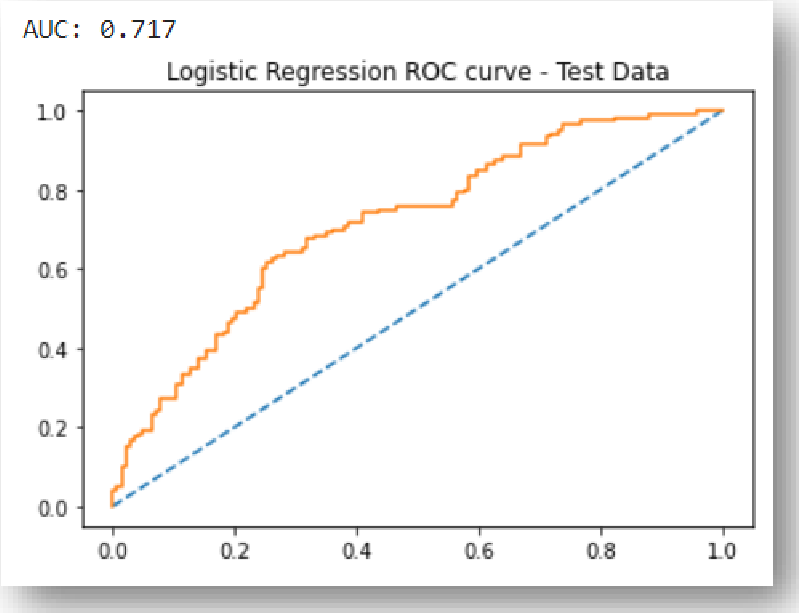
```
              precision    recall  f1-score   support

           0       0.67      0.74      0.71       329
           1       0.66      0.58      0.62       281

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

Figure 37: Logistic Regression Classification Report for Training Data

The classification report for the logistic regression model with the scores for the different model performance measures is calculated and shown in Figure 37. From this figure we can see that the precision of the model is 0.66 means 66% of the data points identified as positive by the model, are really positive. The f1-score is 0.62 means the model is 62% accurate on this data set. Both these

scores are low. The model has 67% accuracy. The recall score is 0.58 which means 58% of the positive observations are correctly predicted.

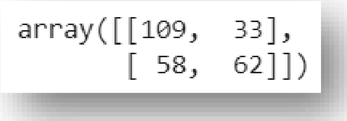
Test Data



```
AUC: 0.717
```

Figure 38: Logistic Regression ROC Curve for Test Data

The ROC_AUC score for the test data was calculated to be 0.717. The ROC curve was plotted and is shown in Figure 38.



```
array([[109,  33],
       [ 58,  62]])
```

Figure 39: Logistic Regression Confusion Matrix for Test Data

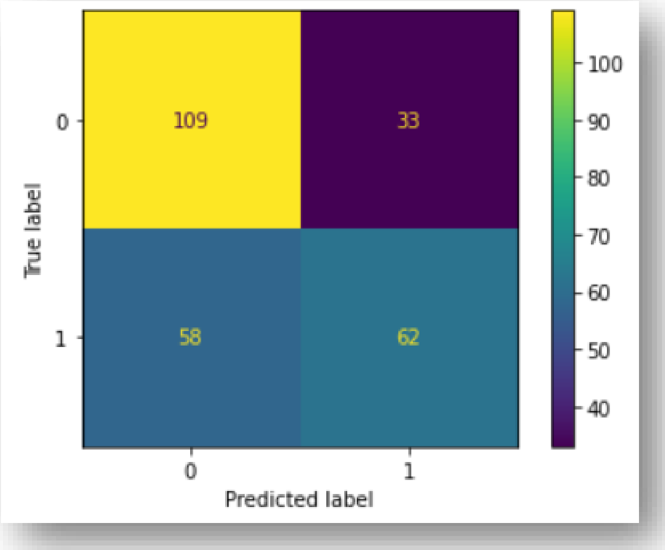
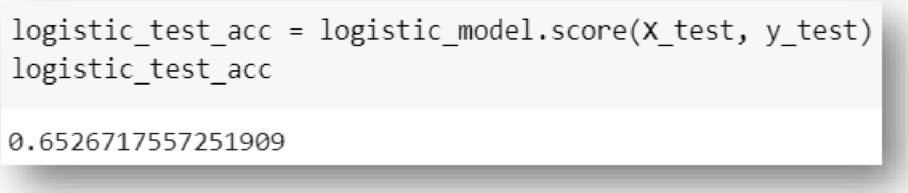


Figure 40: Plot of Logistic Regression Confusion Matrix for Test Data

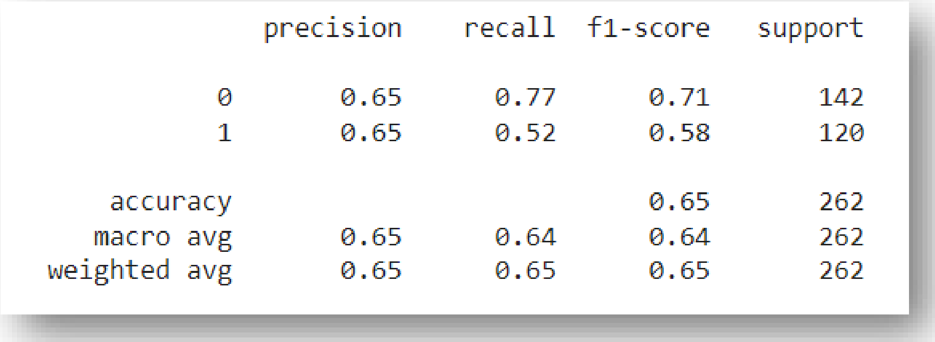


```
logistic_test_acc = logistic_model.score(X_test, y_test)
logistic_test_acc
```

```
0.6526717557251909
```

Figure 41: Logistic Regression Accuracy score for Test Data

The confusion matrix was calculated and shown in Figure 39 and Figure 40. The accuracy score for the test data was found to be 0.65 which is shown in Figure 41.



```
              precision    recall  f1-score   support

           0       0.65      0.77      0.71       142
           1       0.65      0.52      0.58       120

    accuracy                           0.65       262
   macro avg       0.65      0.64      0.64       262
weighted avg       0.65      0.65      0.65       262
```

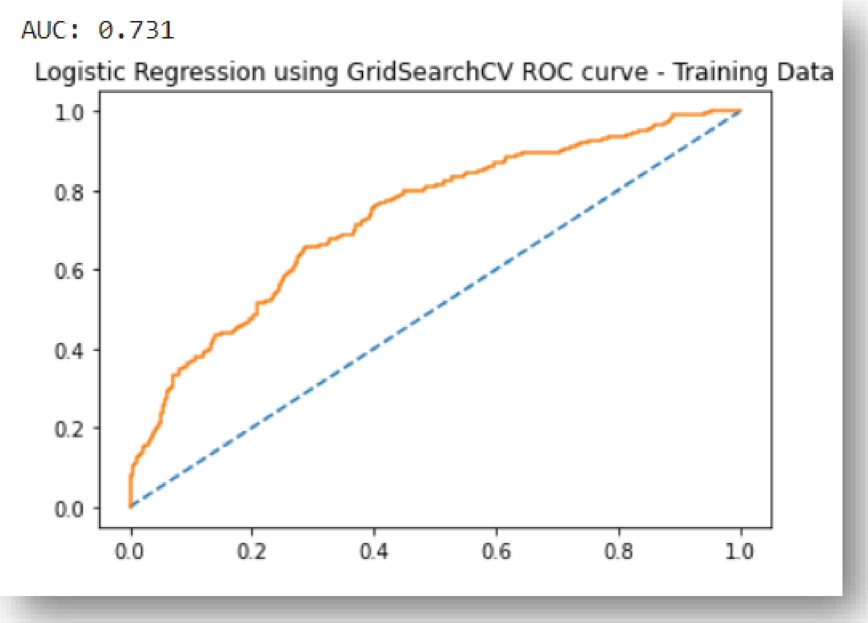
Figure 42: Logistic Regression Classification Report for Test Data

The classification report for the logistic regression model with the scores for the different model performance measures is calculated and shown in Figure 42. From this figure we can see that the precision of the model is 0.65 means 65% of the data points identified as positive by the model, are really positive. The f1-score is 0.58 means the model is 58% accurate on this data set. Both these

scores are low. The model has 65% accuracy. The recall score is 0.52 which means 52% of the positive observations are correctly predicted.

Logistic Regression using GridSearchCV

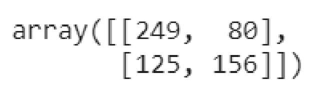
Training Data



```
AUC: 0.731
```

Figure 43: Logistic Regression using GridSearchCV ROC Curve for Training Data

The ROC_AUC score for the training data was calculated to be 0.731. The ROC curve was plotted and is shown in Figure 43.



```
array([[249,  80],
       [125, 156]])
```

Figure 44: Logistic Regression using GridSearchCV Confusion Matrix for Training Data

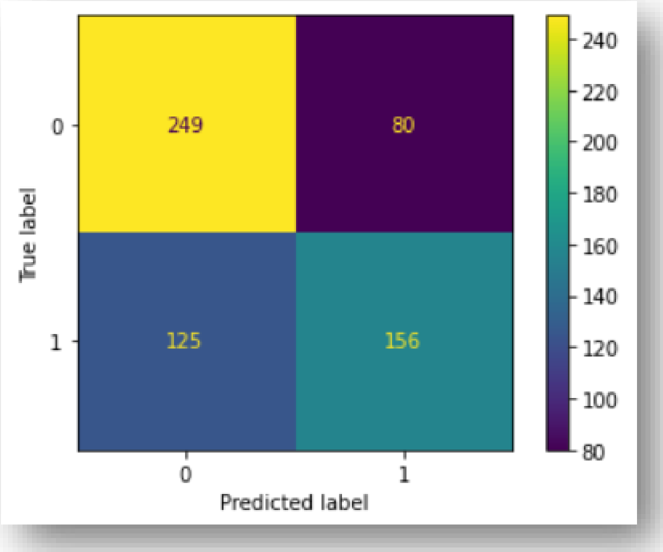
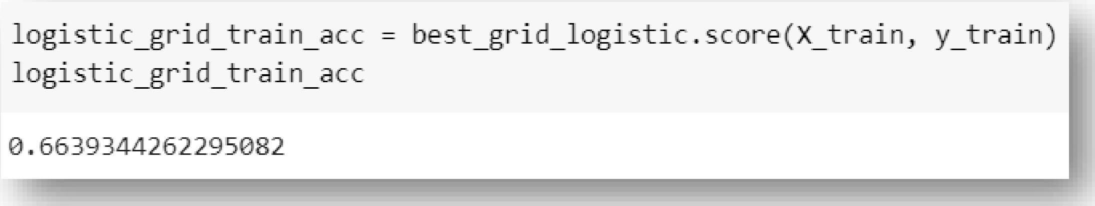


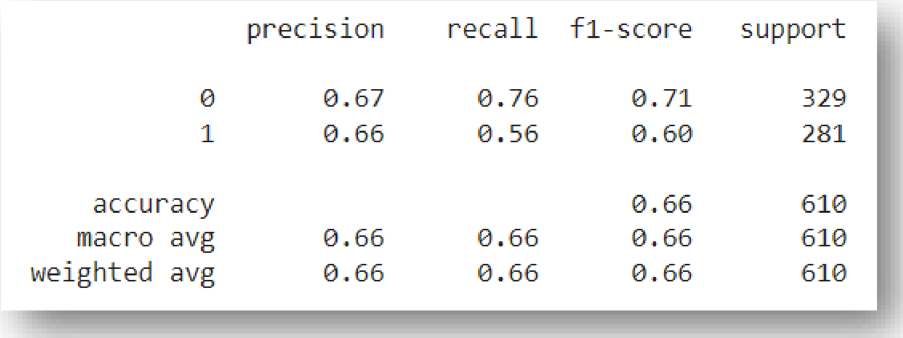
Figure 45: Plot of Logistic Regression using GridSearchCV Confusion Matrix for Training Data



```
logistic_grid_train_acc = best_grid_logistic.score(X_train, y_train)
logistic_grid_train_acc

0.6639344262295082
```

Figure 46: Logistic Regression using GridSearchCV Accuracy score for Training Data

The confusion matrix was calculated and shown in Figure 44 and Figure 45. The accuracy score for the training data was found to be 0.66 which is shown in Figure 46.



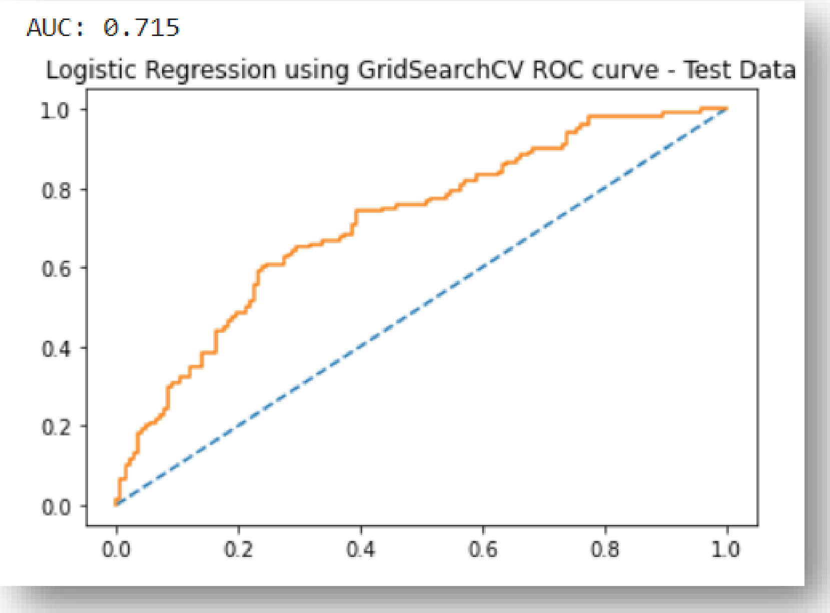
```
              precision    recall  f1-score   support

           0       0.67      0.76      0.71       329
           1       0.66      0.56      0.60       281

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

Figure 47: Logistic Regression using GridSearchCV Classification Report for Training Data

The classification report for the logistic regression model using GridSearchCV with the scores for the different model performance measures is calculated and shown in Figure 47. From this figure we can see that the precision of the model is 0.66 means 66% of the data points identified as positive by the model, are really positive. The f1-score is 0.60 means the model is 60% accurate on this data set.

Both these scores are low. The model has 66% accuracy. The recall score is 0.56 which means 56% of the positive observations are correctly predicted.

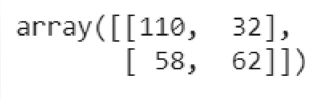
Test Data



```
AUC: 0.715
```

Figure 48: Logistic Regression using GridSearchCV ROC Curve for Test Data

The ROC_AUC score for the test data was calculated to be 0.715. The ROC curve was plotted and is shown in Figure 48.



```
array([[110,  32],
       [ 58,  62]])
```

Figure 49: Logistic Regression using GridSearchCV Confusion Matrix for Test Data

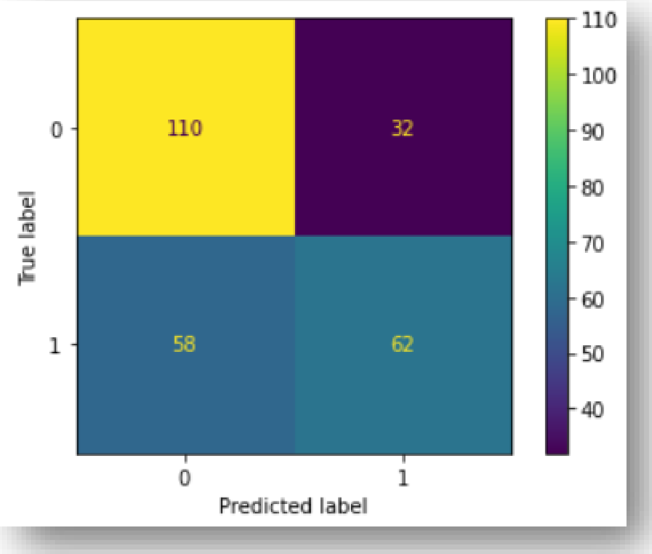
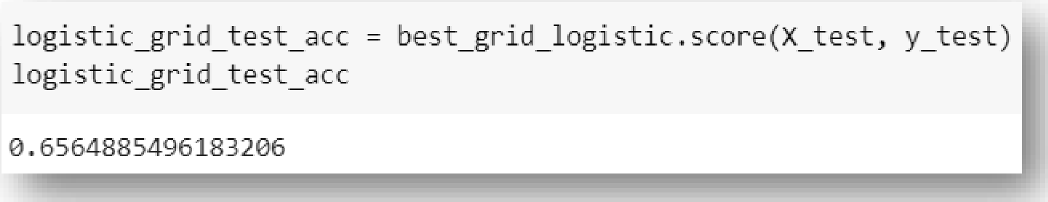


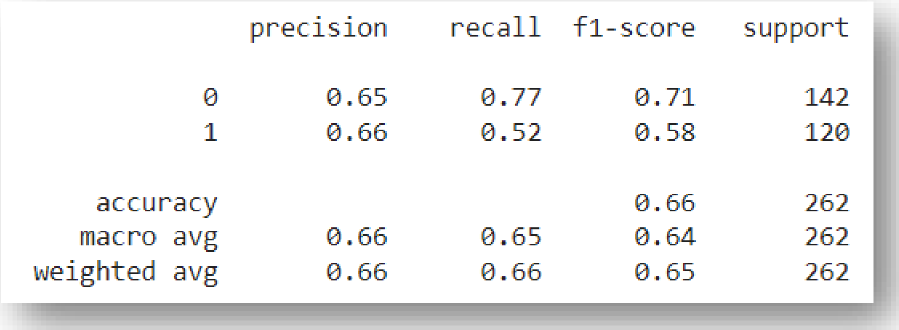
Figure 50: Plot of Logistic Regression using GridSearchCV Confusion Matrix for Test Data



```
logistic_grid_test_acc = best_grid_logistic.score(X_test, y_test)
logistic_grid_test_acc

0.6564885496183206
```

Figure 51: Logistic Regression using GridSearchCV Accuracy score for Test Data

The confusion matrix was calculated and shown in Figure 49 and Figure 50. The accuracy score for the test data was found to be 0.66 which is shown in Figure 51.



```
              precision    recall  f1-score   support

           0       0.65      0.77      0.71       142
           1       0.66      0.52      0.58       120

    accuracy                           0.66       262
   macro avg       0.66      0.65      0.64       262
weighted avg       0.66      0.66      0.65       262
```

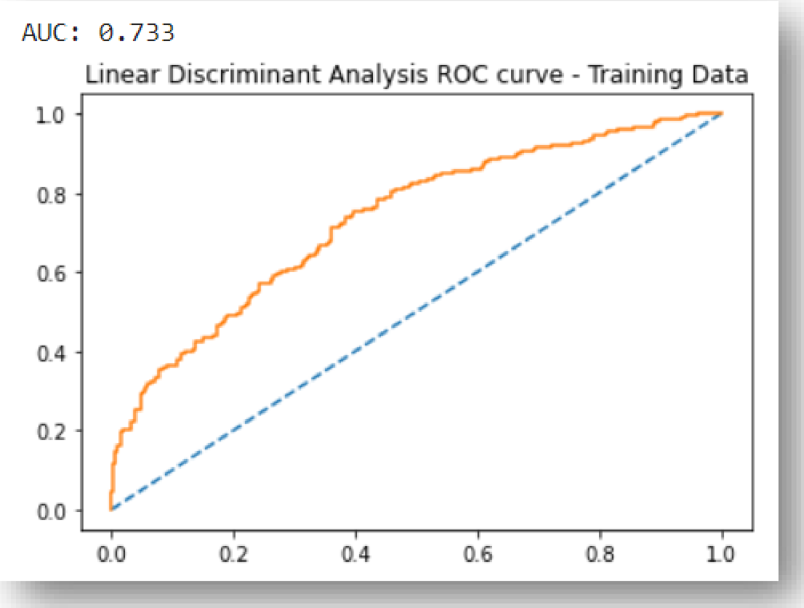
Figure 52: Logistic Regression using GridSearchCV Classification Report for Test Data

The classification report for the logistic regression model using GridSearchCV with the scores for the different model performance measures is calculated and shown in Figure 52. From this figure we can see that the precision of the model is 0.66 means 66% of the data points identified as positive by the model, are really positive. The f1-score is 0.58 means the model is 58% accurate on this data set.

Both these scores are low. The model has 66% accuracy. The recall score is 0.52 which means 52% of the positive observations are correctly predicted.

Linear Discriminant Analysis

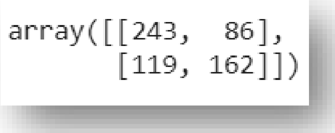
Training Data



```
AUC: 0.733
```

Figure 53: Linear Discriminant Analysis ROC Curve for Training Data

The ROC_AUC score for the training data was calculated to be 0.733. The ROC curve was plotted and is shown in Figure 53.



```
array([[243,  86],
       [119, 162]])
```

Figure 54: Linear Discriminant Analysis Confusion Matrix for Training Data

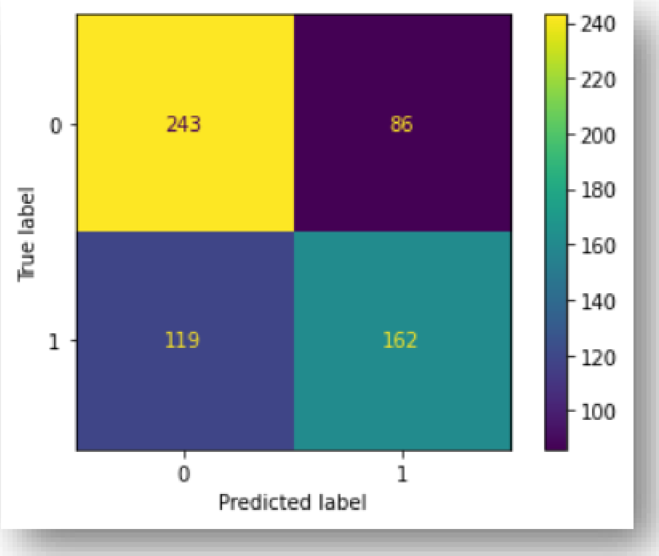
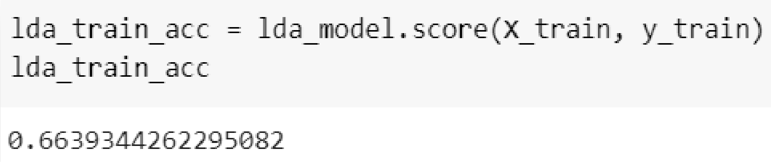


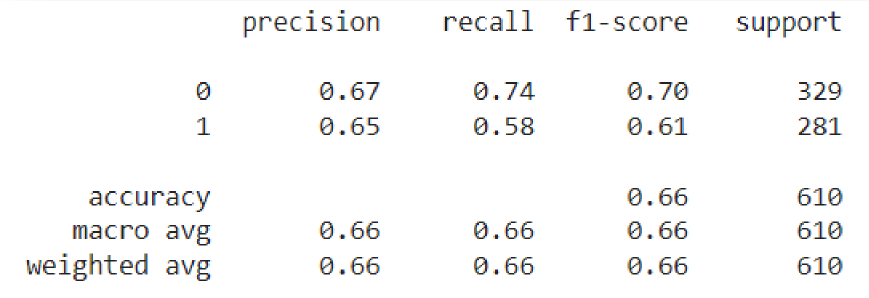
Figure 55: Plot of Linear Discriminant Analysis Confusion Matrix for Training Data



```
lda_train_acc = lda_model.score(X_train, y_train)
lda_train_acc

0.6639344262295082
```

Figure 56: Linear Discriminant Analysis Accuracy score for Training Data

The confusion matrix was calculated and shown in Figure 54 and Figure 55. The accuracy score for the training data was found to be 0.66 which is shown in Figure 56.



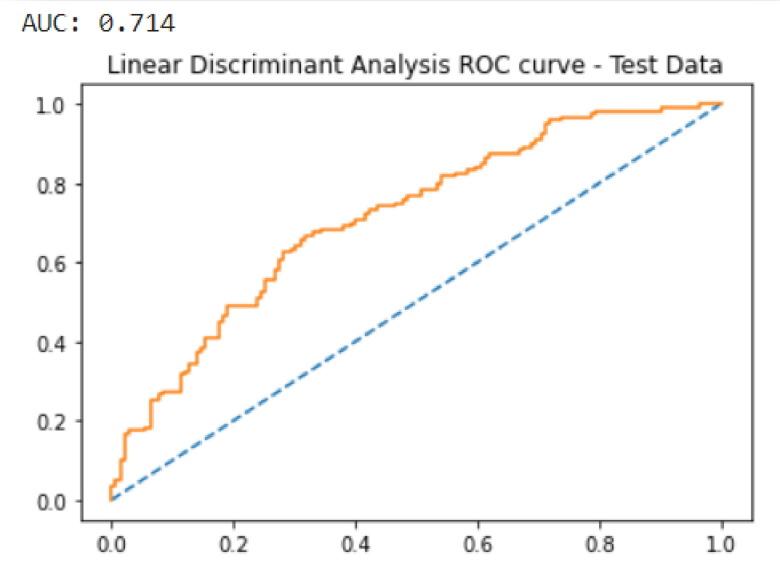
```
              precision    recall  f1-score   support

           0       0.67      0.74      0.70       329
           1       0.65      0.58      0.61       281

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

Figure 57: Linear Discriminant Analysis Classification Report for Training Data

The classification report for the linear discriminant analysis model with the scores for the different model performance measures is calculated and shown in Figure 57. From this figure we can see that the precision of the model is 0.65 means 65% of the data points identified as positive by the model, are really positive. The f1-score is 0.61 means the model is 61% accurate on this data set. Both these

scores are low. The model has 66% accuracy. The recall score is 0.58 which means 58% of the positive observations are correctly predicted.

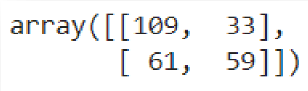
Test Data



```
AUC: 0.714
```

Figure 58: Linear Discriminant Analysis ROC Curve for Test Data

The ROC_AUC score for the test data was calculated to be 0.714. The ROC curve was plotted and is shown in Figure 58.



```
array([[109,  33],
       [ 61,  59]])
```

Figure 59: Linear Discriminant Analysis Confusion Matrix for Test Data

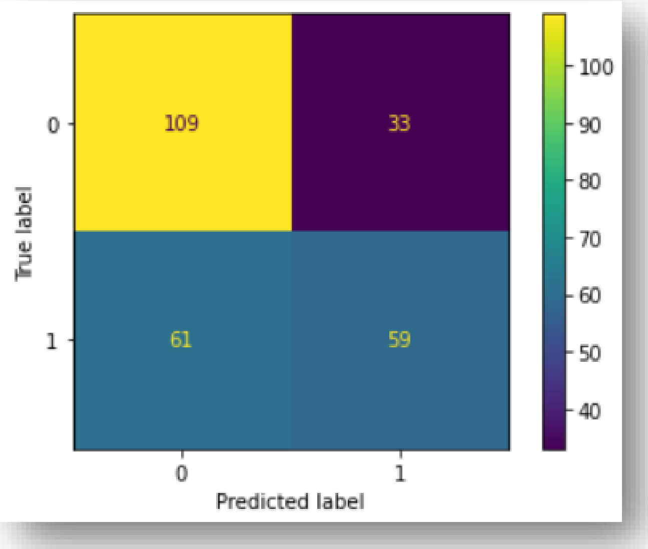
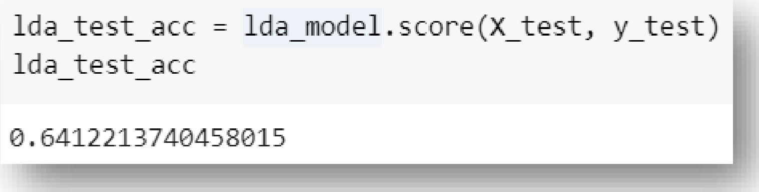


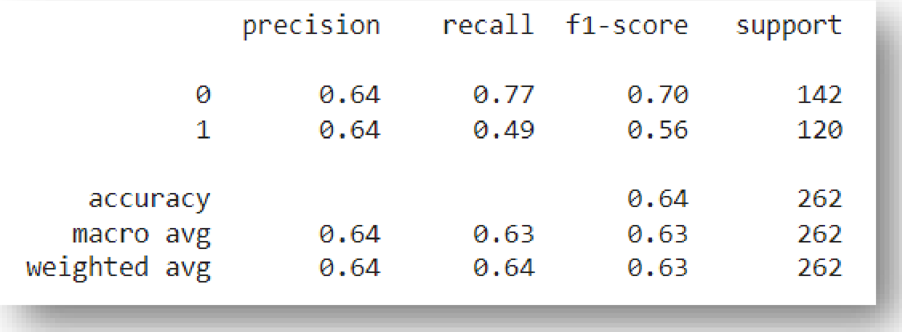
Figure 60: Plot of Linear Discriminant Analysis Confusion Matrix for Test Data



```
lda_test_acc = lda_model.score(X_test, y_test)
lda_test_acc

0.6412213740458015
```

Figure 61: Linear Discriminant Analysis Accuracy score for Test Data

The confusion matrix was calculated and shown in Figure 59 and Figure 60. The accuracy score for the test data was found to be 0.64 which is shown in Figure 61.



```
              precision    recall  f1-score   support

           0       0.64      0.77      0.70       142
           1       0.64      0.49      0.56       120

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

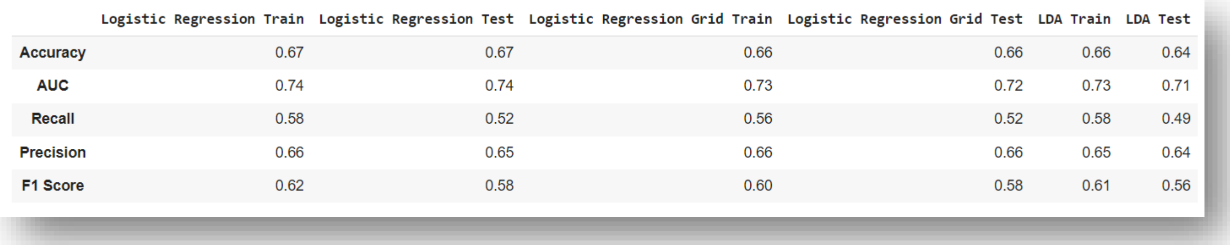
Figure 62: Linear Discriminant Analysis Classification Report for Test Data

The classification report for the linear discriminant analysis model with the scores for the different model performance measures is calculated and shown in Figure 62. From this figure we can see that the precision of the model is 0.64 means 64% of the data points identified as positive by the model, are really positive. The f1-score is 0.56 means the model is 56% accurate on this data set. Both these

scores are low. The model has 64% accuracy. The recall score is 0.49 which means 49% of the positive observations are correctly predicted.

A comparison of all the models was done to understand which model is the best suited for our case study. The model performance measures of all the models were tabulated and it is shown in Table 16.

Table 16: Comparison of all the models



	Logistic Regression Train	Logistic Regression Test	Logistic Regression Grid Train	Logistic Regression Grid Test	LDA Train	LDA Test
Accuracy	0.67	0.67	0.66	0.66	0.66	0.64
AUC	0.74	0.74	0.73	0.72	0.73	0.71
Recall	0.58	0.52	0.56	0.52	0.58	0.49
Precision	0.66	0.65	0.66	0.66	0.65	0.64
F1 Score	0.62	0.58	0.60	0.58	0.61	0.56

From Table 16, we are able to understand that the ROC_AUC score for the logistic regression model is the highest compared to the other two models. The larger the area under the curve, the better the model. This is also true in case of accuracy where the score is highest in the logistic regression model for both training and test data compared to the other models.

The dataset has outliers in ‘salary’. Logistic regression model is more robust predictor in case of outliers. Therefore, it is recommended to use Logistic regression model.

Moreover, the precision and f1 score of the logistic regression train data is the highest. The recall score for the logistic regression train data is the highest (0.58) but low for the test data (0.52). However, since all the other model performance measures are better for the **logistic regression model** compared to the other models, it is chosen as the **optimized model** for our problem.

This is further analysed with the help of the ROC Curve.

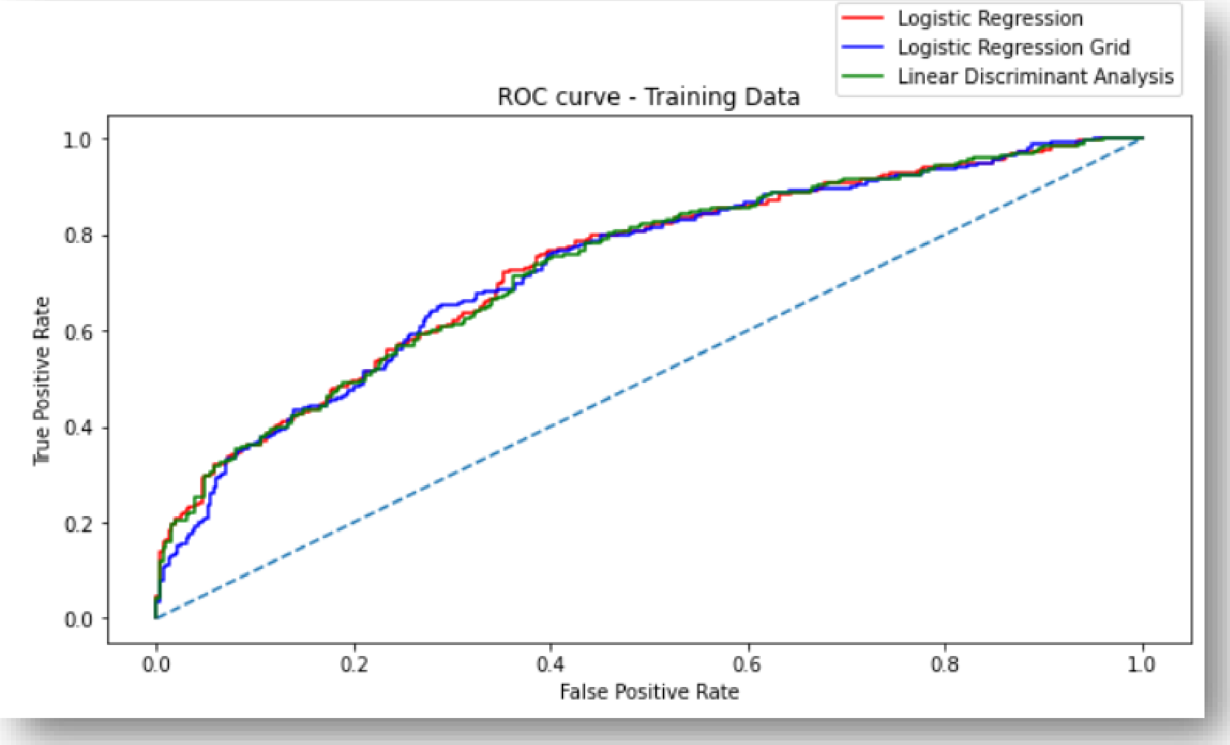


Figure 63: ROC Curve for all models - Training Data

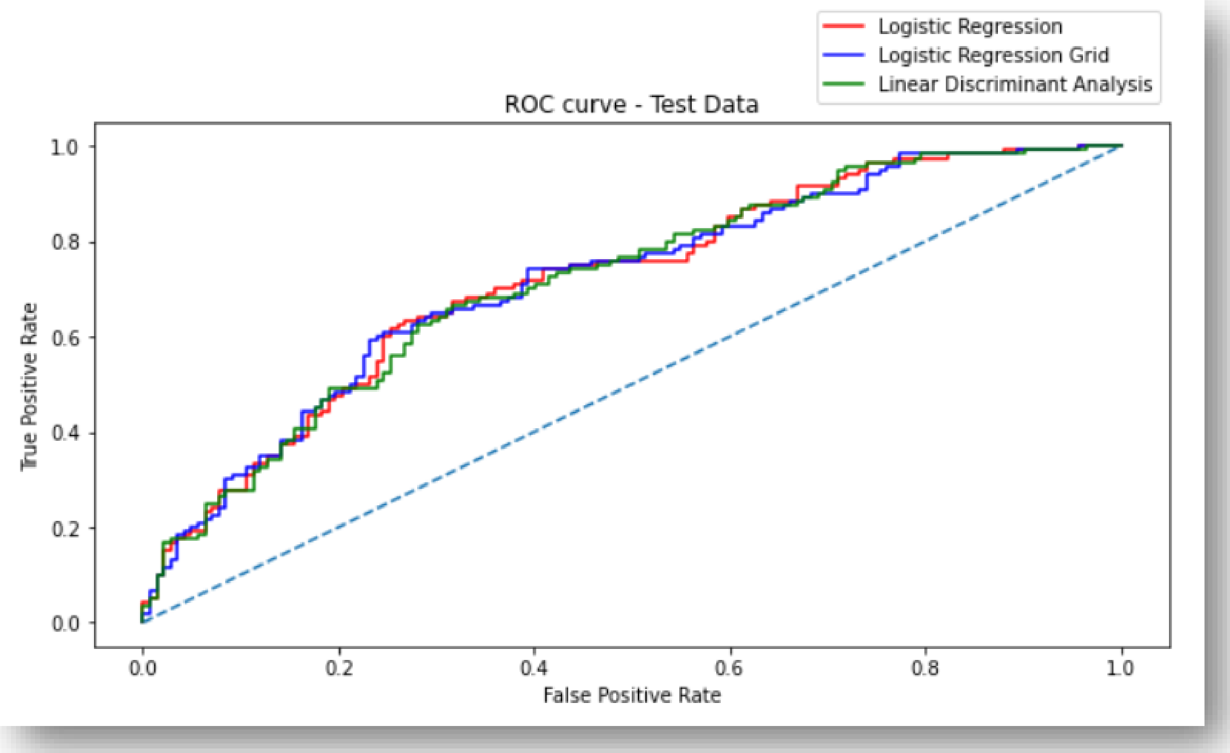


Figure 64: ROC Curve for all models - Test Data

From Figure 63 and Figure 64, we are able to see that the Logistic Regression model is the most optimum for our case study as the curve for the Logistic Regression model model is the steepest. The steeper the ROC Curve, the stronger the model.

Therefore, **Logistic Regression model** is selected as the best model for our problem as it has better accuracy, precision, recall and f1 score compared to logistic regression using GridSearchCV and linear discriminant analysis model.

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 recommendations for this problem are that more real time unstructured data and past data should be collected.

If the employee is a foreigner and the employee do not have young children, the chances of opting for holiday package is good.

Special offer can be designed to domestic employees to opt for holiday package. It can be seen that many high salary employees are not opting for holiday package. Therefore, the company can focus on high salary employees to sell holiday package. The company can also offer Holiday Packages as incentives or bonus to the employees with the best performance for that year. This will be a way of promoting holiday packages and also benefit the company by making it employee friendly and a comfortable place to work.

We observe that employees having older children are not opting for holiday package. This maybe because the number of holidays reduces when the children grow up. There maybe more tests, assignments for the children and so they might not want to go on a holiday. In such cases, discounts can be offered to the employees whose children have scored more than 90 marks. This might improve the chances of opting for a holiday package.

Age of the employee is not a material in opting for holiday package.

It can be observed from coefficient values from all the models in Figure 27, Figure 29 and Figure 31 that opting for holiday package has strong negative relation with number of young children. Holiday packages can be modified to make infant and young children friendly to attract more employees having young children. Family packages can be introduced. Special discounts can be given to family packages to attract employees having children.