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:Yes/No

Questions:

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.

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).

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.

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.

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

Import the data set and import all the necessary libraries

Read the dataset.

```
# Read the Dataset and store it in dataframe
df=pd.read_csv('Holiday_Package.csv')
```

Head of the data : First 5 rows

	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

Tail of the data : last 5 rows

	Unnamed: 0	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
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

Data Info:

```
<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
```

Holliday_Package and foreign variables are of object type, we will assign codes to it and then feed this data to Model. Unnamed 0 is unwanted variable, which does not play any role here, Total we have 872 rows 8 columns in given dataset. No Null Values in the Data.

Description of Data:

	Unnamed: 0	Salary	age	educ	no_young_children	no_older_children
count	872.000000	872.000000	872.000000	872.000000	872.000000	872.000000
mean	436.500000	47729.172018	39.955275	9.307339	0.311927	0.982798
std	251.869014	23418.668531	10.551675	3.036259	0.612870	1.086786
min	1.000000	1322.000000	20.000000	1.000000	0.000000	0.000000
25%	218.750000	35324.000000	32.000000	8.000000	0.000000	0.000000
50%	436.500000	41903.500000	39.000000	9.000000	0.000000	1.000000
75%	654.250000	53469.500000	48.000000	12.000000	0.000000	2.000000
max	872.000000	236961.000000	62.000000	21.000000	3.000000	6.000000

Salary is continuous variable, Whereas age, educ and number young children are having integers. Two Variables 'Holiday Package' and 'Employee went to Foreign or Not' are Categorical Variables. Here Holiday Package is Target Variable/Independent variable.

Lets drop the unwanted column : Unnamed: 0

Lets see the Head of Data After Removal of Unnamed column

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

Lets check for Null Values,

```
# Null values check
df.isnull().sum()
```

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

No Null Values can be seen in the data.

lets Check whether duplicates are present in the Dataset or not.

```
Number of duplicate rows = 0
(872, 7)
```

No Duplicates seen in the Dataset

Check for Percentage of Independent Target variable: Yes/No Result.

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

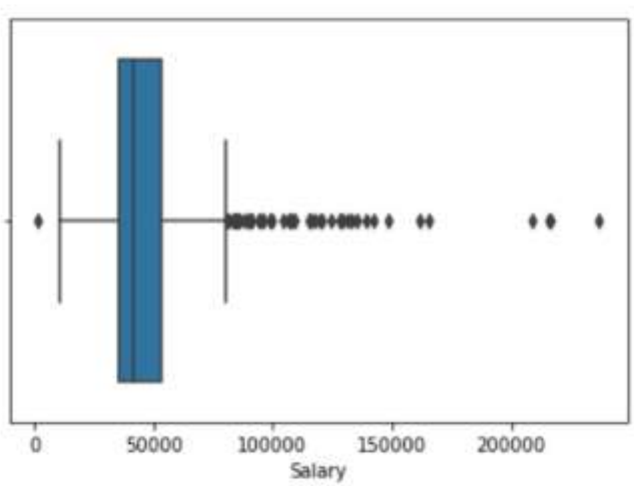
Output/Target Variable seems to be stable as we have almost same number of Records With Yes/No Output.54 % No and 46% Yes.

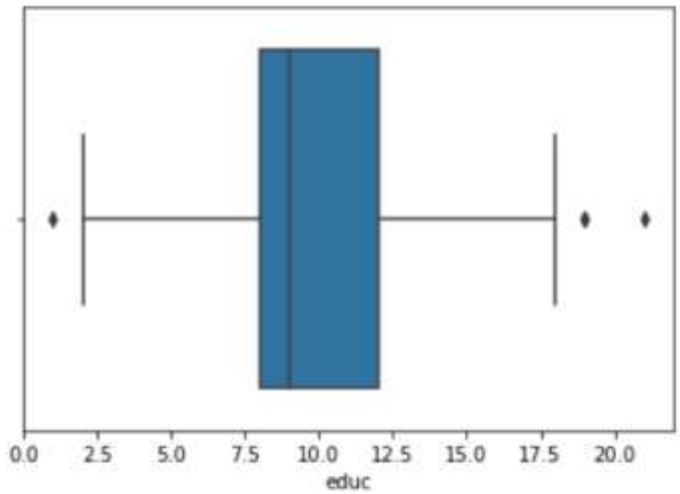
Lets Encode the Nominal Categorical Variables. Two Categorical variables: Holliday_Package and foreign.

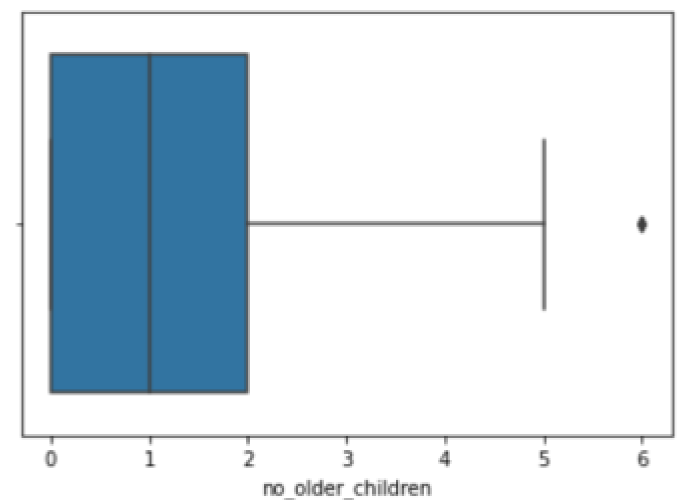
```
Holliday_Package
no     471
yes    401
Name: Holliday_Package, dtype: int64


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

Lets plot the boxplot to get the idea of outliers. We Can see only Salary variable has outliers, which is also continuous variable.Other Variables are either Categorical or Integer, which has valid values.

Check for Outliers:







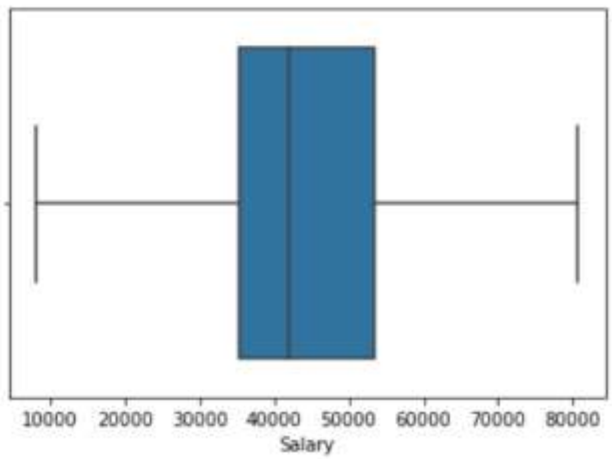
'Salary'-> It has Outliers, We will remove it ahead. 'age'->No Outliers in this varibale. 'educ'-> It seems this variable has 3 Outiers, which seems valid. 'no_young_children','no_older_children' have valid Outliers. so we will keep it as it is.

We will treat outliers only for Salary variable here

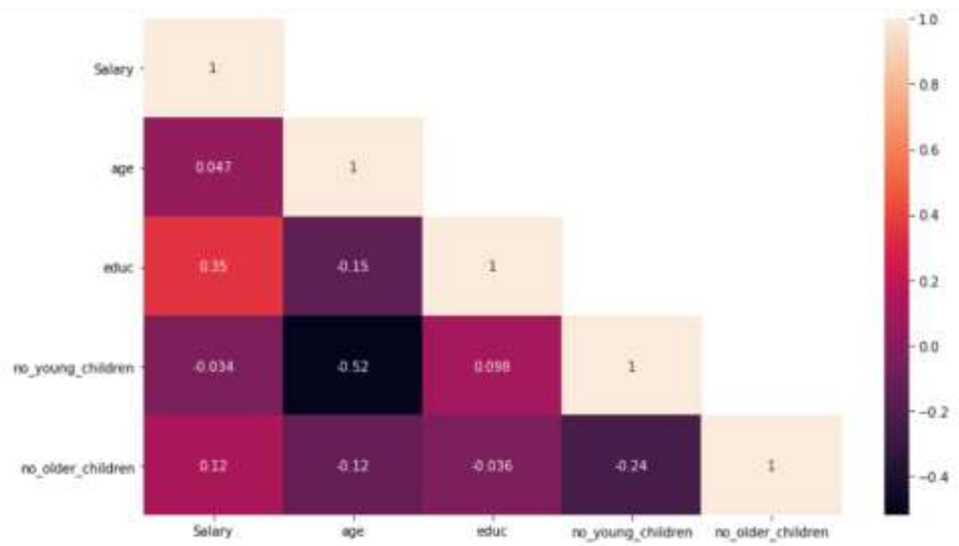
```
lr,ur=remove_outlier(df_data['Salary'])
print('Lower Range :',lr,'\nUpper Range :',ur)
df_data['Salary']=np.where(df_data['Salary']>ur,ur,df_data['Salary'])
df_data['Salary']=np.where(df_data['Salary']<lr,lr,df_data['Salary'])
```

```
Lower Range : 8105.75
Upper Range : 80687.75
```

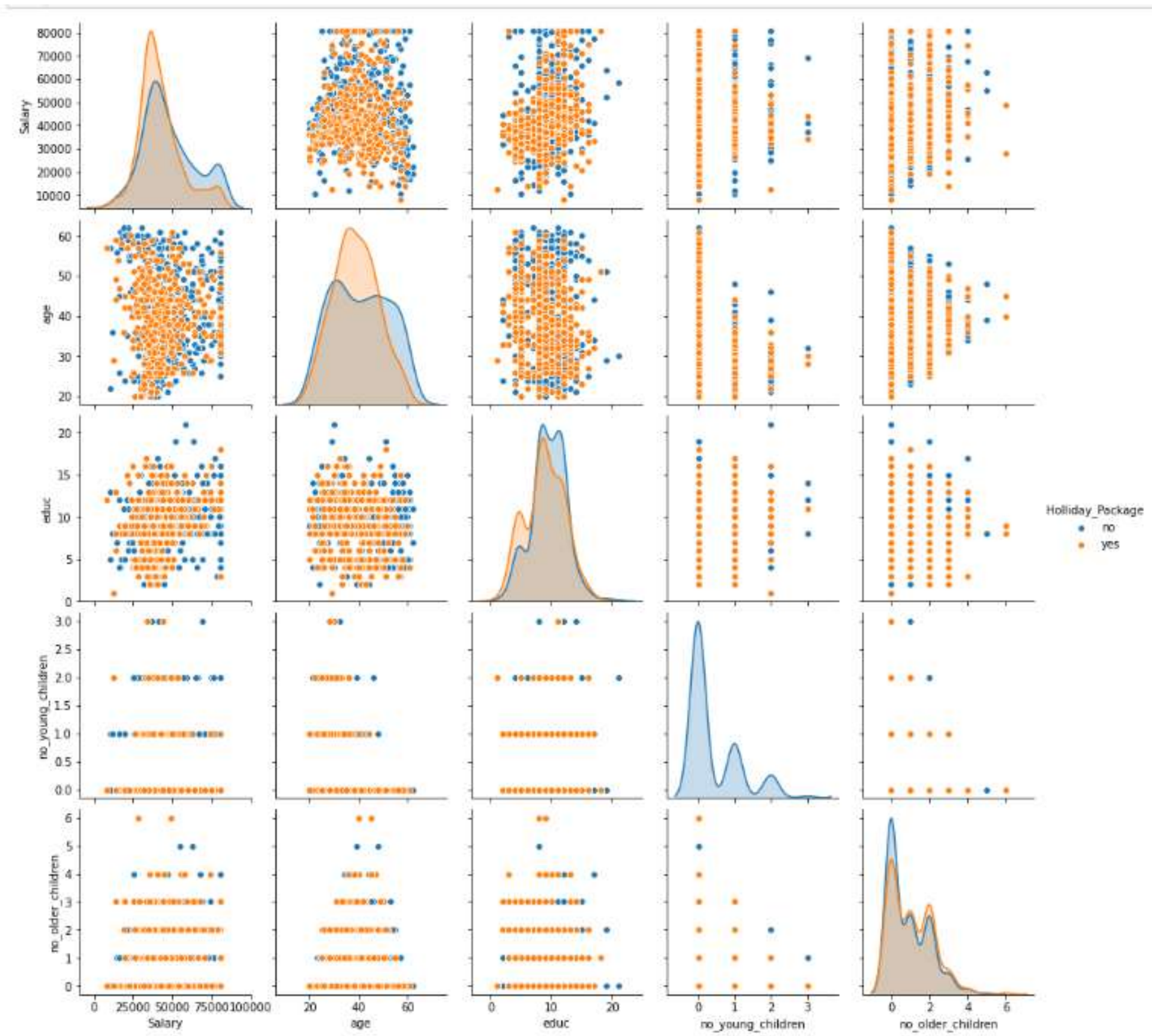
After Outlier Removal Salary Variable can be seen like this.



To see the Multicollinearity, Lets plot the heatmap:



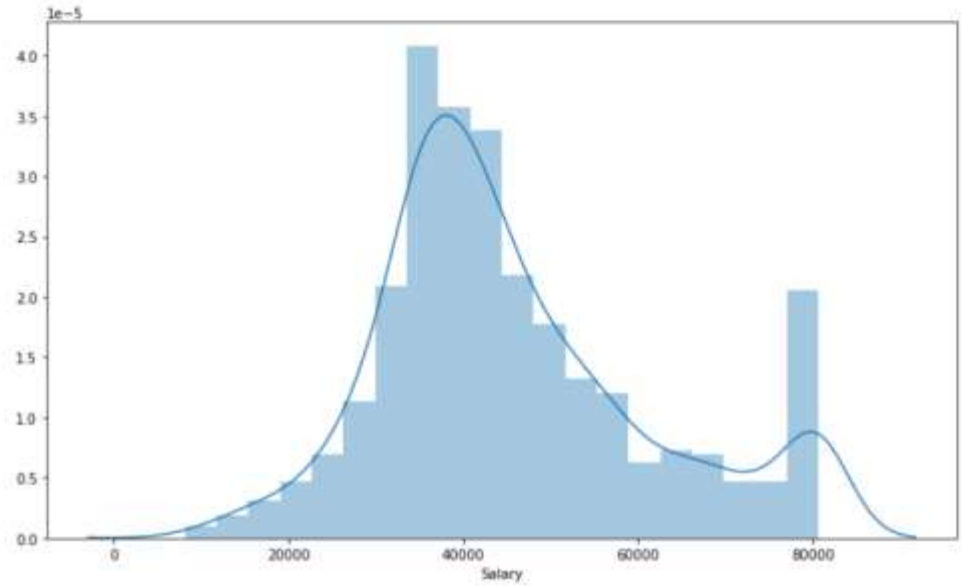
Lets plot the pairplot to see the relation between all variables.



We can see that there is no issue of Multicollinearity, No dependent Variables show Strong Correlation amongst them.Education And Salary has some Relation between them.

Lets do some Univariate and Multivariate Analysis and try to get some hidden patterns.

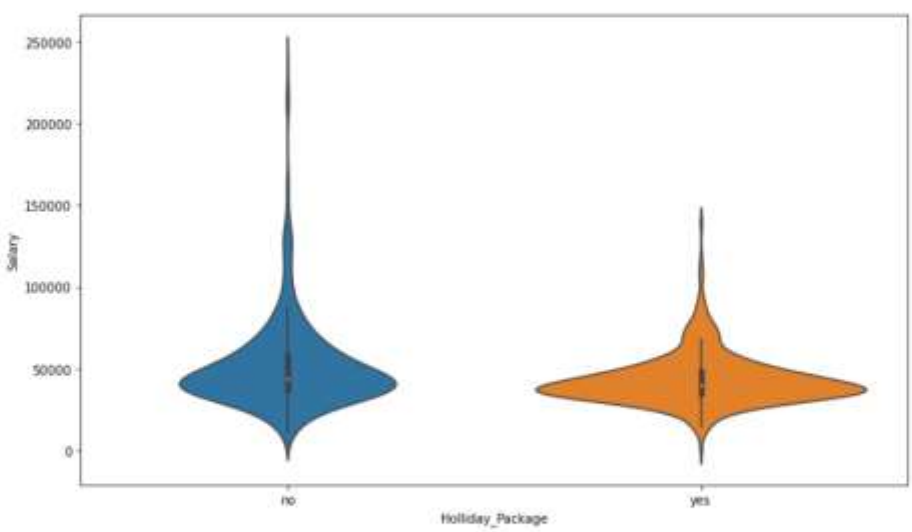
This is the Distribution for Salary Variable.



The data is normally distributed almost equally around mean value, so we can say it is similar to Normal Distribution.

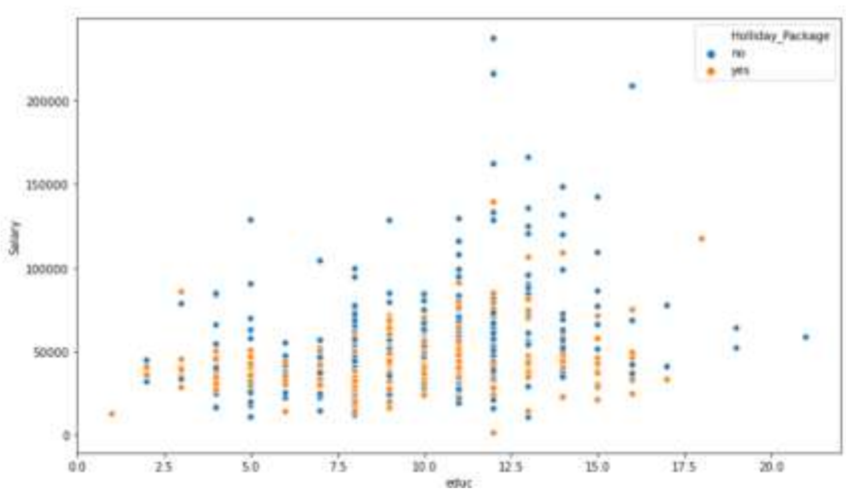
Lets do Bivariate Analysis:

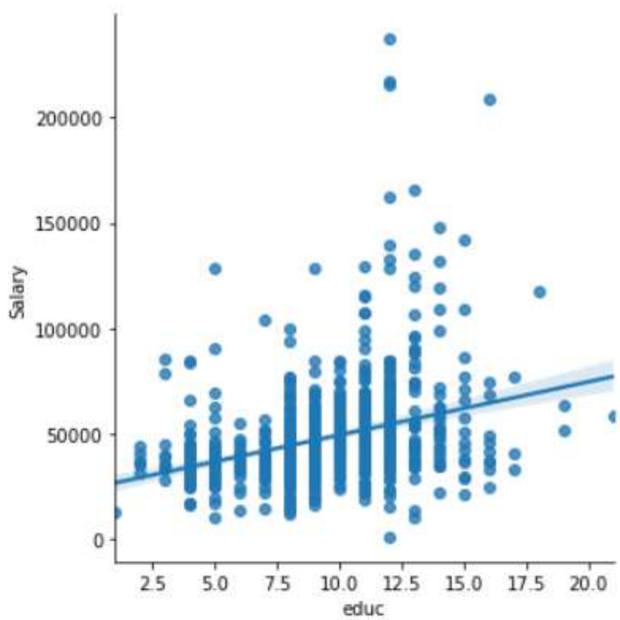
1.Salary vs Holiday_Package,



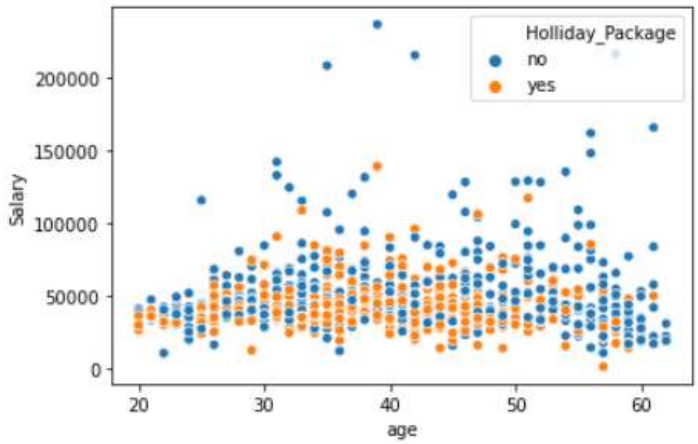
Salary above 1,50,000 have always not opted for Holiday Package

2.Salary vs Education ,

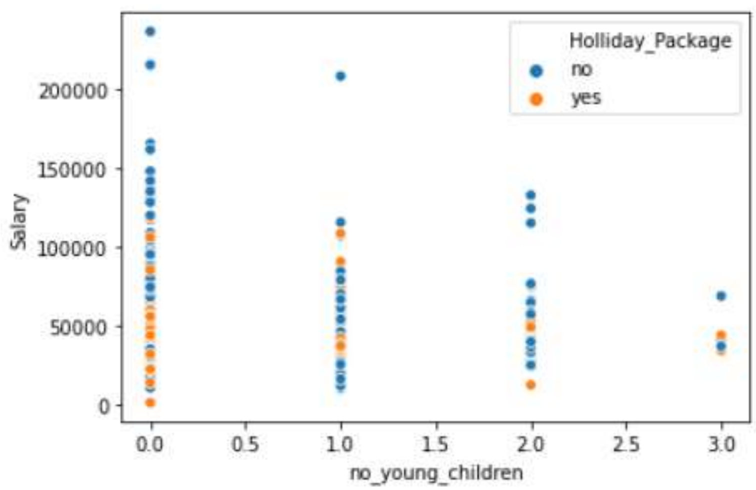




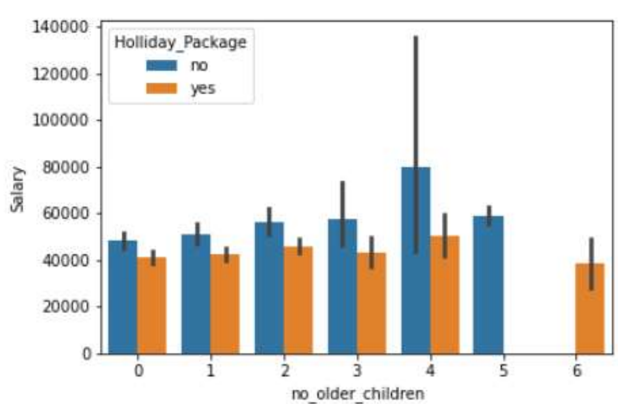
3.Salary vs Age



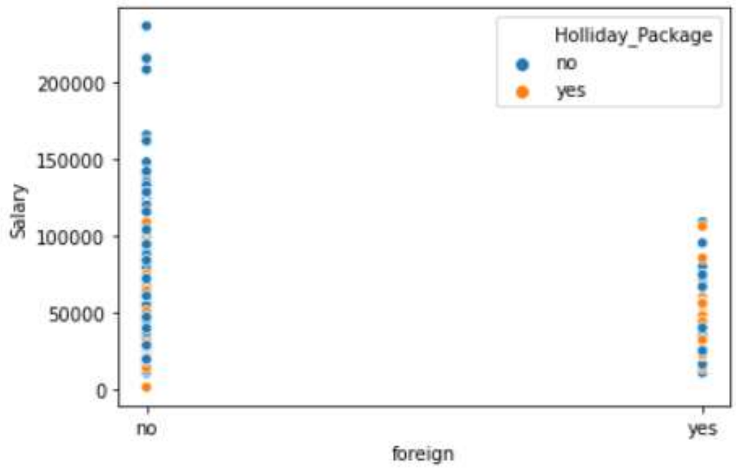
4.No of Younge Children vs Holiday Package,



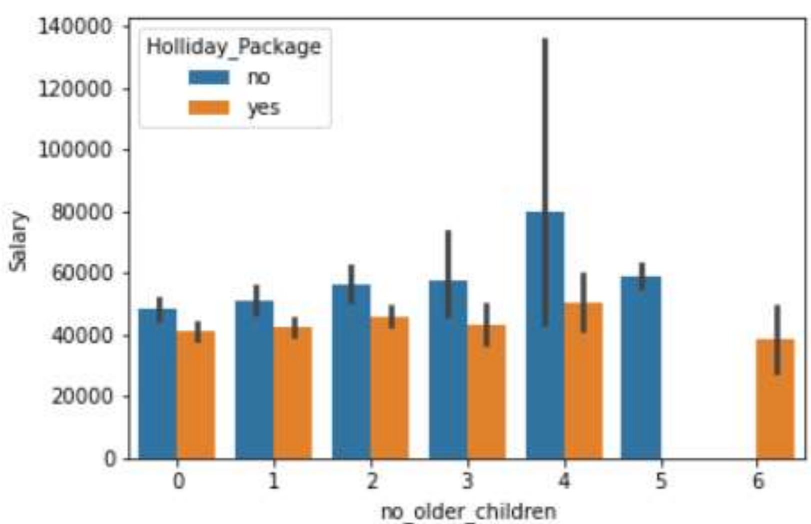
5.No of Older Children vs Salary



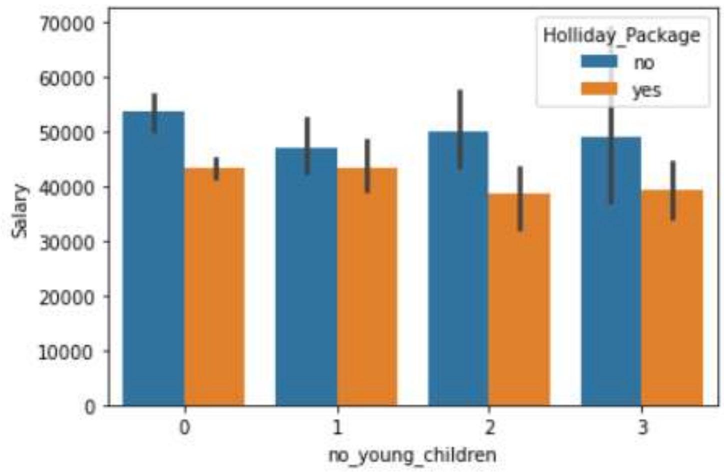
5.Foreign vs Salary



6. Salary vs no. of Older Children



No_young_children vs Salary



Most of The Employees get the Salary between 10,000 to 1,00,000. and they are the ones who choosing Holiday_Packages. It can be seen that Salary is High if No of Education years is High.

Very few Datapoints can suggest that Salary is Increasing with Age. Age group 20-60 Shows Almost Similar pattern in terms of Salary.Employee age over 50 to 60 have seems to be not taking the holiday package.

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).

Converting all objects to categorical codes, Encoding the data –

```
feature: Holliday_Package
[no, yes]
Categories (2, object): [no, yes]
[0 1]


feature: foreign
[no, yes]
Categories (2, object): [no, yes]
[0 1]
```

Two variables are there : Holiday Package and Foreign

After Encoding the data:

```
1  df_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 872 entries, 0 to 871
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Holiday_Package    872 non-null    int8
 1   Salary             872 non-null    float64
 2   age                872 non-null    int64
 3   Education          872 non-null    int64
 4   no_young_children  872 non-null    int64
 5   no_older_children  872 non-null    int64
 6   Foreigner          872 non-null    int8
dtypes: float64(1), int64(4), int8(2)
memory usage: 35.9 KB
```

Salary and Foreigner variables data type as float from object.

Head of the Data:

	Holliday_Package	Salary	age	educ	no_young_children	no_older_children	foreign
0	0	48412.0	30	8	1	1	0
1	1	37207.0	45	8	0	1	0
2	0	58022.0	46	9	0	0	0
3	0	66503.0	31	11	2	0	0
4	0	66734.0	44	12	0	2	0

Clean the Column Names:

```
## Cleaning the coulmn names
df_data.columns = df_data.columns.str.replace('Holliday_Package', 'Holiday_Package')
df_data.columns = df_data.columns.str.replace('age', 'age')
df_data.columns = df_data.columns.str.replace('educ', 'Education')
df_data.columns = df_data.columns.str.replace('foreign', 'Foreigner')
```

Holliday_Package as Holiday_Package

educ as education

foreign as Foreigner

VIF Values:

```
Salary VIF =  1.2
age VIF =  1.56
Education VIF =  1.41
no_young_children VIF =  1.57
no_older_children VIF =  1.19
Foreigner VIF =  1.27
```

VIF is less than 4, Hence all Dependent Variables dont show much multicollinearity.

Train and Test Split:

All the Independent variables stored in X while Dependent Variable is stored in y.

```
# Train Test Split
# Copy all the predictor variables into X dataframe
X = df_data.drop('Holiday_Package', axis=1)

# Copy target into the y dataframe.
y = df_data['Holiday_Package']
```

Split X and y into training and test set in 70:30 ratio

Y train value Count or Distribution percentage.

```
y_train_lr.value_counts(1)
```

```
0    0.539344
1    0.460656
Name: Holiday_Package, dtype: float64
```

Y test value Count or Distribution percentage.

```
y_test_lr.value_counts(1)
```

```
0    0.541985
1    0.458015
Name: Holiday_Package, dtype: float64
```

Apply Logistic Regression, fitting the Logistic Regression model.

```
# Fit the Logistic Regression model
model = LogisticRegression(solver='newton-cg',max_iter=10000,penalty='none',verbose=True,n_jobs=2)
model.fit(X_train_lr, y_train_lr)
```

```
[Parallel(n_jobs=2)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=2)]: Done   1 out of   1 | elapsed:   32.5s finished
```

```
LogisticRegression(max_iter=10000, n_jobs=2, penalty='none', solver='newton-cg',
                   verbose=True)
```

Predicting on Training and Test dataset and then Getting the Predicted Classes and Probs.

	0	1
0	0.677845	0.322155
1	0.534493	0.465507
2	0.691845	0.308155
3	0.487745	0.512255
4	0.571939	0.428061

Apply Linear Discriminant Analysis Algorithm:

Split X and y into training and test set in 70:30 ratio

```
# Split X and y into training and test set in 70:30 ratio
X_train_lda, X_test_lda, y_train_lda, y_test_lda = train_test_split(X, y, test_size=0.30 , random_state=1,stratify=y)
```

Build the Model

```
#Build LDA Model
clf = LinearDiscriminantAnalysis()
model=clf.fit(X_train_lda,y_train_lda)
```

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.

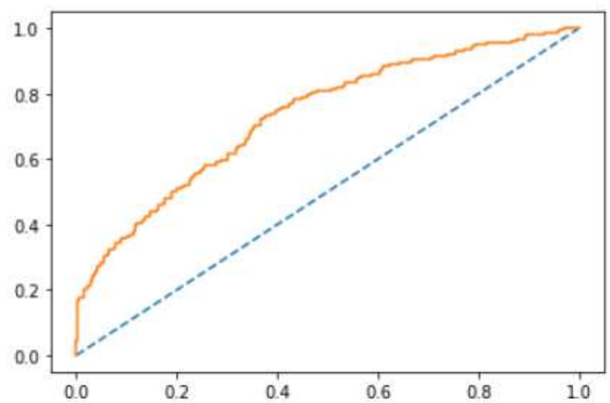
Performance Metrics Linear Regression:

<u>On Train Data</u>

Accuracy of the Training data

AUC and ROC for the train data

AUC: 0.714

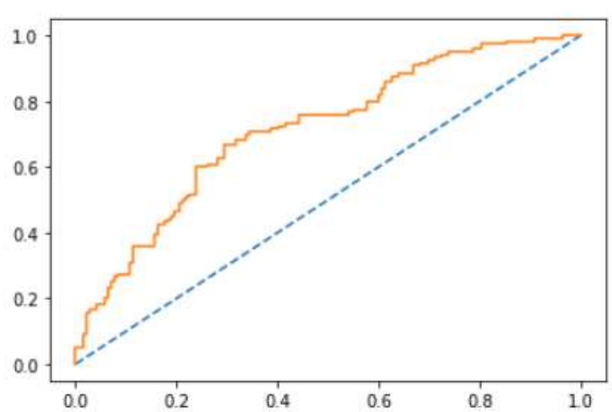


<u>On Test Data</u>

Accuracy of the test data

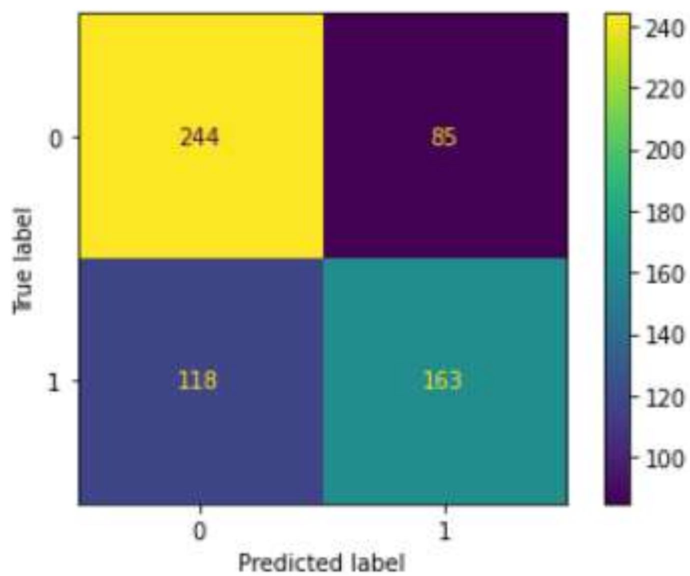
AUC and ROC for the test data

AUC: 0.715



Confusion Matrix:

For Train Data:



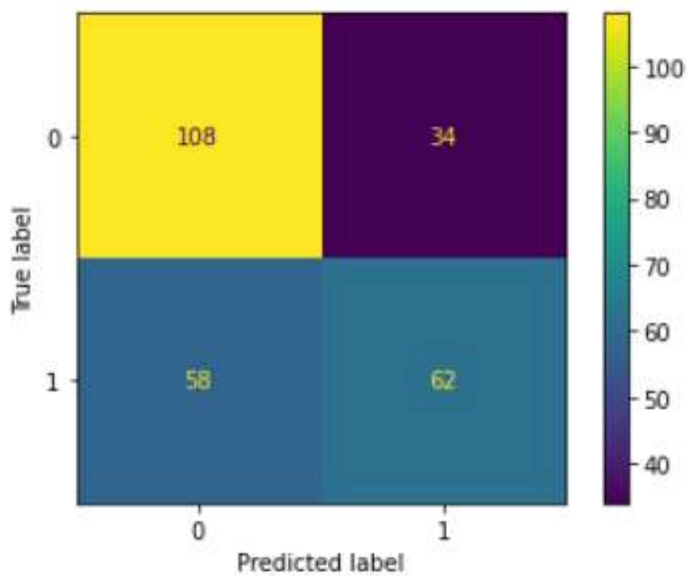
Classification Report for Train Data:

```
print(classification_report(y_train_lr, ytrain_predict_lr))
```

```
              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
```

```
lr_train_precision  0.66
lr_train_recall  0.58
lr_train_f1  0.62
```

Confusion Matrix for test data:



Test Data Classification report for Linear Regression

	precision	recall	f1-score	support
0	0.65	0.76	0.70	142
1	0.65	0.52	0.57	120
accuracy			0.65	262
macro avg	0.65	0.64	0.64	262
weighted avg	0.65	0.65	0.64	262

```
lr_test_precision  0.65
lr_test_recall  0.52
lr_test_f1  0.57
```

A

```python
# Applying GridSearchCV for Logistic Regression
grid={'penalty':['l2','none'],
      'solver':['newton-cg','lbfgs','liblinear'],
      'tol':[0.0001,0.00001]}
```

Applying Grid Search to improve the model:

```
# Applying GridSearchCV for Logistic Regression
grid={'penalty':['l2','none'],
      'solver':['newton-cg','lbfgs','liblinear'],
      'tol':[0.0001,0.00001]}
```

We have selected three Solvers: newton-cg, lbfgs, liblinear.

We are finding out the best Combination here.

Set the model as Logistic Regression with 10,000 iterations

```
model = LogisticRegression(max_iter=10000,n_jobs=2)
```

Applying and fitting a Grid search,

```
GridSearchCV(cv=3, estimator=LogisticRegression(max_iter=10000, n_jobs=2),
             n_jobs=-1,
             param_grid={'penalty': ['l2', 'none'],
                         'solver': ['newton-cg', 'lbfgs', 'liblinear'],
                         'tol': [0.0001, 1e-05]},
             scoring='f1')
```

We Received the best Solver is newton-cg, with tol: 0.0001

```
{'penalty': 'none', 'solver': 'newton-cg', 'tol': 0.0001}

LogisticRegression(max_iter=10000, n_jobs=2, penalty='none', solver='newton-cg')
```

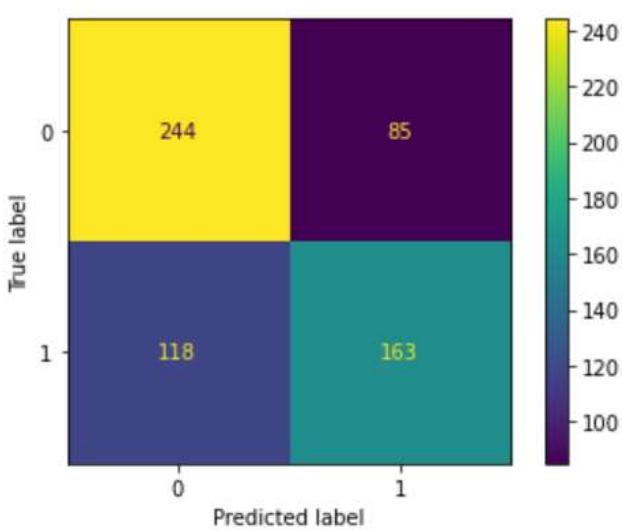
Getting The Probabilities of Getting 0 and 1.

	0	1
0	0.677845	0.322155
1	0.534493	0.465507
2	0.691845	0.308155
3	0.487745	0.512255
4	0.571939	0.428061

We are Interested Whether Employee will select Holiday Package or not. So we need good Probability of getting 1.

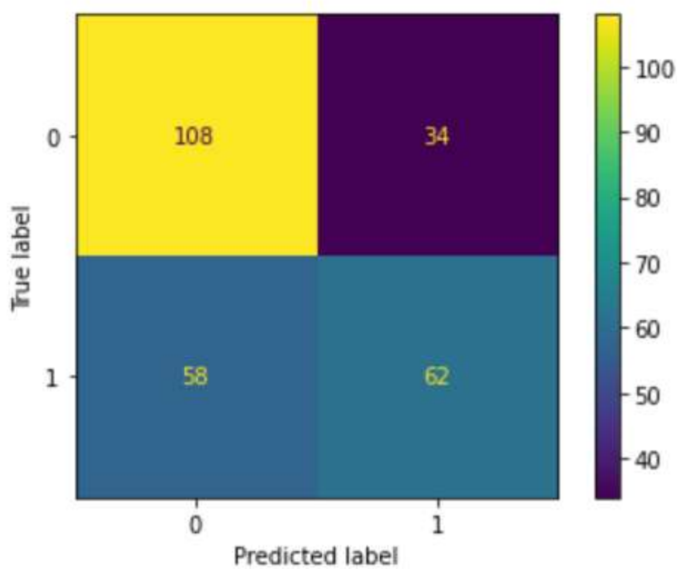
Confusion matrix and Classification Report on the training data:

```
              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
```



Confusion matrix and Classification Report on the test data

	precision	recall	f1-score	support
0	0.65	0.76	0.70	142
1	0.65	0.52	0.57	120
accuracy			0.65	262
macro avg	0.65	0.64	0.64	262
weighted avg	0.65	0.65	0.64	262



Model Summary:

Dep. Variable:	Holiday_Package	**No. Observations:**	872
Model:	Logit	**Df Residuals:**	865
Method:	MLE	**Df Model:**	6
Date:	Sun, 11 Apr 2021	**Pseudo R-squ.:**	0.1244
Time:	21:06:51	**Log-Likelihood:**	-526.78
converged:	True	**LL-Null:**	-601.61
Covariance Type:	nonrobust	**LLR p-value:**	9.138e-30

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	2.5432	0.559	4.550	0.000	1.448	3.639
Salary	-2.088e-05	5.26e-06	-3.970	0.000	-3.12e-05	-1.06e-05
age	-0.0496	0.009	-5.491	0.000	-0.067	-0.032
Education	0.0342	0.029	1.172	0.241	-0.023	0.091
no_young_children	-1.3287	0.180	-7.386	0.000	-1.681	-0.976
no_older_children	-0.0251	0.074	-0.341	0.733	-0.169	0.119
Foreigner	1.3037	0.200	6.519	0.000	0.912	1.696

Performance Metrics for Linear Discriminant Analysis:

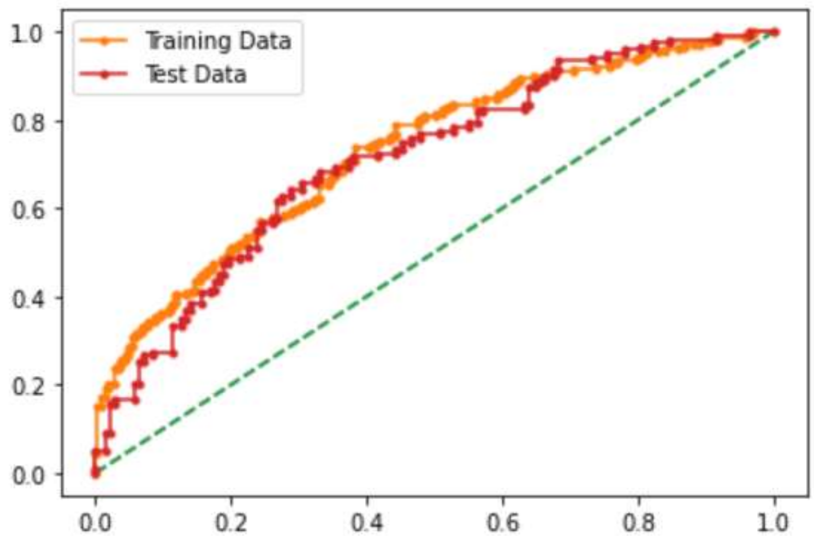
Training Data and Test Data Confusion Matrix Comparison

Training Data and Test Data Classification Report Comparison

```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.67      0.74      0.70       329
           1       0.65      0.57      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


Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.65      0.76      0.70       142
           1       0.65      0.52      0.57       120

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

AUC and ROC for Test Data:

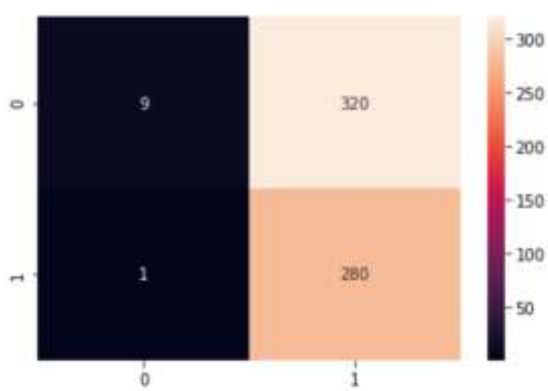
```
AUC for the Training Data: 0.731
AUC for the Test Data: 0.714
```



Calculating Accuracy, F1 Score and Confusion Matrix:

Cutoff: 0.1

```
Accuracy Score 0.4738
F1 Score 0.6356

Confusion Matrix
```



Cutoff: 0.2

```
Accuracy Score 0.523
F1 Score 0.6498

Confusion Matrix
```

Cutoff: 0.3

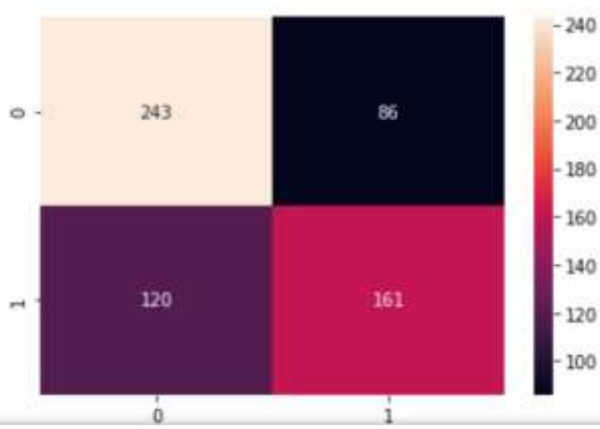
```
Accuracy Score 0.6066
F1 Score 0.6774

Confusion Matrix
```

Cutoff : 0.4

```
Accuracy Score 0.6623
F1 Score 0.6677

Confusion Matrix
```

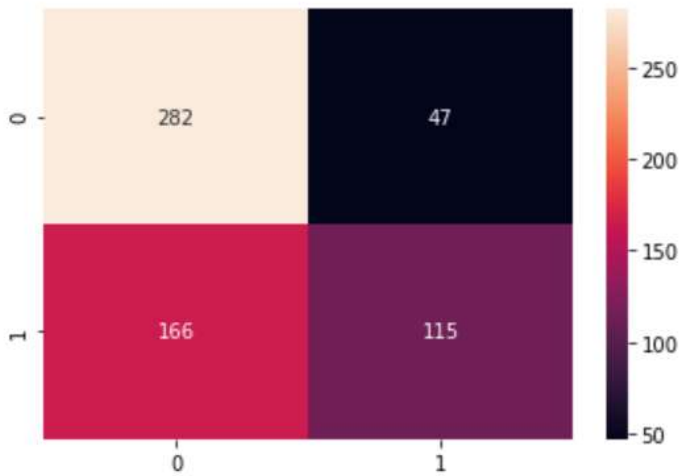
Cutoff 0.5

```
Accuracy Score 0.6623
F1 Score 0.6098

Confusion Matrix
```



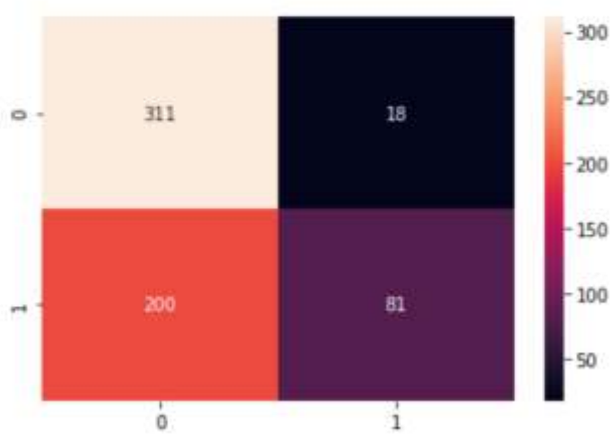
Cutoff : 0.6

```
Accuracy Score 0.6508
F1 Score 0.5192

Confusion Matrix
```



Cutoff : 0.7

```
Accuracy Score 0.6426
F1 Score 0.4263

Confusion Matrix
```

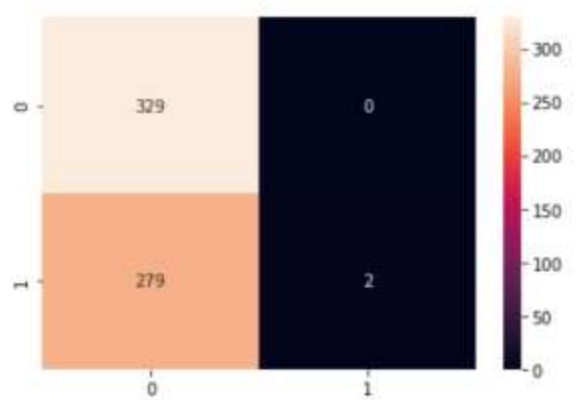


Cutoff : 0.8

```
Accuracy Score 0.5902
F1 Score 0.2038

Confusion Matrix
```

Cutoff : 0.9

```
Accuracy Score 0.5426
F1 Score 0.0141

Confusion Matrix
```

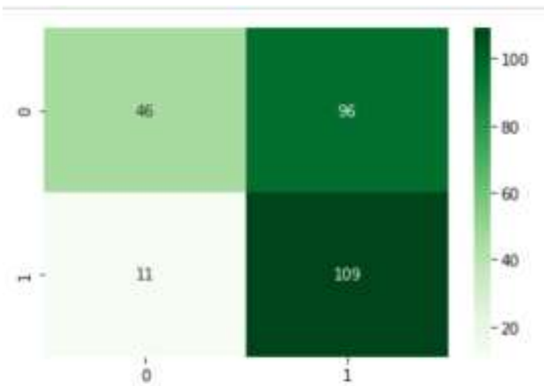


Overall Result :

```
Cutoff: 0.1      Accuracy: 0.4738 recall: 0.9964 precision: 0.4667 F1 Score: 0.6356
Cutoff: 0.2      Accuracy: 0.523 recall: 0.9609 precision: 0.4909 F1 Score: 0.6498
Cutoff: 0.3      Accuracy: 0.6066 recall: 0.8968 precision: 0.5443 F1 Score: 0.6774
Cutoff: 0.4      Accuracy: 0.6623 recall: 0.7367 precision: 0.6106 F1 Score: 0.6677
Cutoff: 0.5      Accuracy: 0.6623 recall: 0.573 precision: 0.6518 F1 Score: 0.6098
Cutoff: 0.6      Accuracy: 0.6508 recall: 0.4093 precision: 0.7099 F1 Score: 0.5192
Cutoff: 0.7      Accuracy: 0.6426 recall: 0.2883 precision: 0.8182 F1 Score: 0.4263
Cutoff: 0.8      Accuracy: 0.5902 recall: 0.1139 precision: 0.9697 F1 Score: 0.2038
Cutoff: 0.9      Accuracy: 0.5426 recall: 0.0071 precision: 1.0 F1 Score: 0.0141
```

Here we will select 0.3 as It has maximum F1 score, Precision, Accuracy and recall are good for this cutoff.

Confusion Matrix for Test Data:



Classification Report Custom and Default:

Default:

Classification Report of the default cut-off test data:

	precision	recall	f1-score	support
0	0.65	0.76	0.70	142
1	0.65	0.52	0.57	120
accuracy			0.65	262
macro avg	0.65	0.64	0.64	262
weighted avg	0.65	0.65	0.64	262

Classification Report of the custom cut-off test data:

	precision	recall	f1-score	support
0	0.81	0.32	0.46	142
1	0.53	0.91	0.67	120
accuracy			0.59	262
macro avg	0.67	0.62	0.57	262
weighted avg	0.68	0.59	0.56	262

Overall Performance:

```
lda_test_precision  0.53
lda_test_recall  0.91
lda_test_f1  0.67
```

Final Comparision:

	LR Train	LR Test	LDA Train	LDA Test
Accuracy	0.67	0.65	0.66	0.65
AUC	0.73	0.71	0.73	0.71
Recall	0.58	0.52	0.90	0.91
Precision	0.66	0.65	0.54	0.53
F1 Score	0.62	0.57	0.68	0.67

If we compare the above results, Accuracy and AUC are almost same for both models. In case of Recall and Precision both Models have performed good and better vice versa, so this can be little nullified. Now F1 score remains a strong deciding Factor, but we can see that both are having almost similar value. but LDA seens to be on brighter side in comparision with Logistic Regression.Generally Linear Discriminant Analysis Performs better if Target variable is Categorical.

2.4 Inference: Basis on these predictions, what are the insights and recommendations. Please explain and summarize the various steps performed in this project. There should be proper business interpretation and actionable insights present.

Summary:

In this Case study we have to decide strategy or plan by looking at 872 employees data. We have to provide a plan to improve the count of Holiday Packages and earn more profit.

Steps Performed and Insights:

To Achive this first we did Exploratory Data Analysis, to get the hidden patterns within variables.and we found some Interesting insights in this step.

Salary is continuous variable,Whereas age, educ and number young children are having integers. Two Variables 'Holiday Package' and 'Employee went to Foreign or Not' are Categorical Variables. Here Holiday Package is Target Variable/Independent variable.

'Salary' is independent variable and it has Outliers, We have removed it.Other variables has outliers but we have not removed it, as it seems valid.

Most of The Employees get the Salary between 10,000 to 1,00,000. and they are the ones who choosing Holiday_Packages. It can be seen that Salary is High if No of Education years is High.

Very few Datapoints can suggest that Salary is Increasing with Age. Age gro up 20-60 Shows Almost Similar pattern in terms of Salary.Employee age over 50 to 60 have seems to be not taking the holiday package.

After EDA, We have split the data and Applied two algorithms 1.Logistic Regression 2.Linear Discriminant Analyis

Both Algorithms shown almost similar Results after performing metric checks.

Salary, Education and age seems to be deciding factors. here it can be seen that person who went to Foreign opted for Holiday Package.

Recommendations:

The one who are Earning above 1,50,000 are not choosing Packages, we need further data to find why. or on present data we can say that by providing some better tour plans, foreign trips ,If they are busy with their work we can provide Some uninterrupted Connection/Internet/Connectivity plans, so that we can convince them to select package.

The Person who are above 50 needs to be Targeted,as they are not choosing Holiday Packages. We might need to change the plans, Or promotional offers, Couple discounts, Some additional Security. Or we can convince this age group by giving a company of more people from same age group, so that they will not feel alone or missed.

We have to Provide the better plans to the ones who has Older children and more convinient trips to the once who has children in range of 0-7 years, to increase Sales.