LOGISTIC REGRESSION AND LDA

PREPARED BY
MURALIDHARAN N

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.

Loading all the necessary library for the model building.

Now, reading the head and tail of the dataset to check whether data has been properly fed

HEAD OF THE DATA

	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

	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

SHAPE OF THE DATA (872, 8)

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
```

- No null values in the dataset,
- We have integer and object data

DATA DESCRIBE

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Unnamed: 0	872.0	NaN	NaN	NaN	436.500000	251.869014	1.0	218.75	436.5	654.25	872.0
Holliday_Package	872	2	no	471	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Salary	872.0	NaN	NaN	NaN	47729.172018	23418.668531	1322.0	35324.00	41903.5	53469.50	236961.0
age	872.0	NaN	NaN	NaN	39.955275	10.551675	20.0	32.00	39.0	48.00	62.0
educ	872.0	NaN	NaN	NaN	9.307339	3.036259	1.0	8.00	9.0	12.00	21.0
no_young_children	872.0	NaN	NaN	NaN	0.311927	0.612870	0.0	0.00	0.0	0.00	3.0
no_older_children	872.0	NaN	NaN	NaN	0.982798	1.086786	0.0	0.00	1.0	2.00	6.0
foreign	872	2	no	656	NaN	NaN	NaN	NaN	NaN	NaN	NaN

We have integer and continuous data,

Holiday package is our target variable

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

Null value 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
```

check for duplicates in data

```
: dups = df.duplicated()
  print('Number of duplicate rows = %d' % (dups.sum()))
```

```
Number of duplicate rows = 0
```

Unique values in the categorical data

HOLLIDAY_PACKAGE: 2
Yes   401

No 471
Name: Holliday Package, dtype: int64

FOREIGN : 2
Yes 216
No 656
Name: foreign, dtype: int64

Percentage of target :

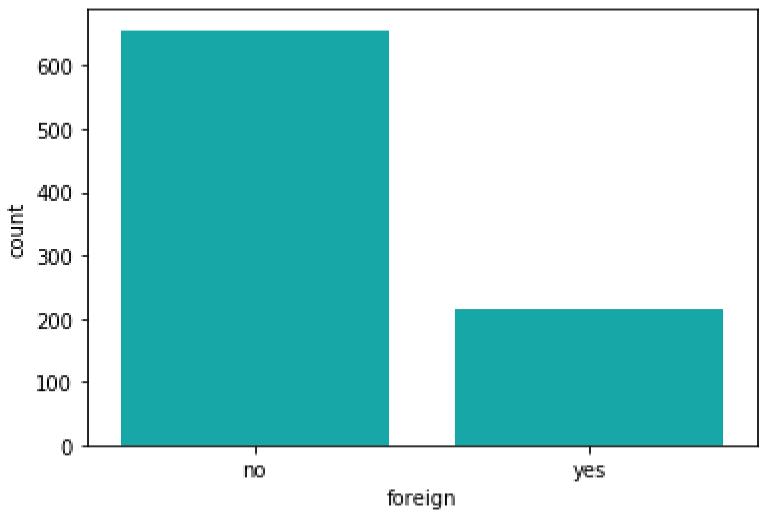
```
df.Holliday_Package.value_counts(1)
```

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

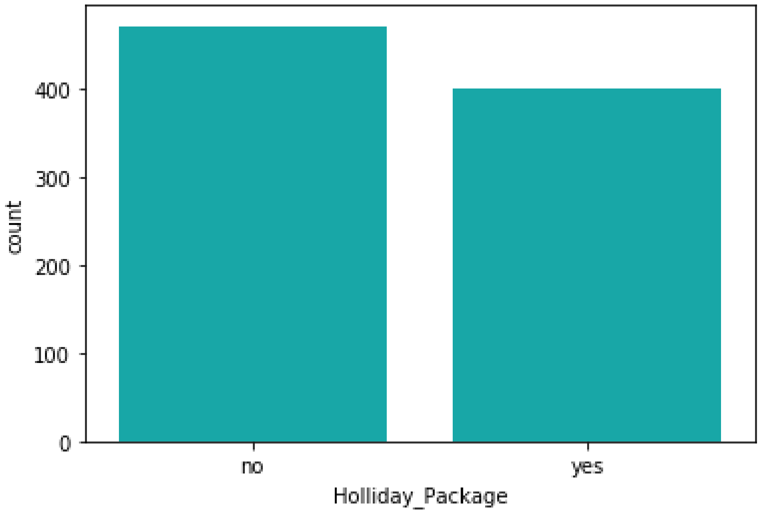
This split indicates that 45% of employees are interested in the holiday package.

CATEGORICAL UNIVARIATE ANALYSIS

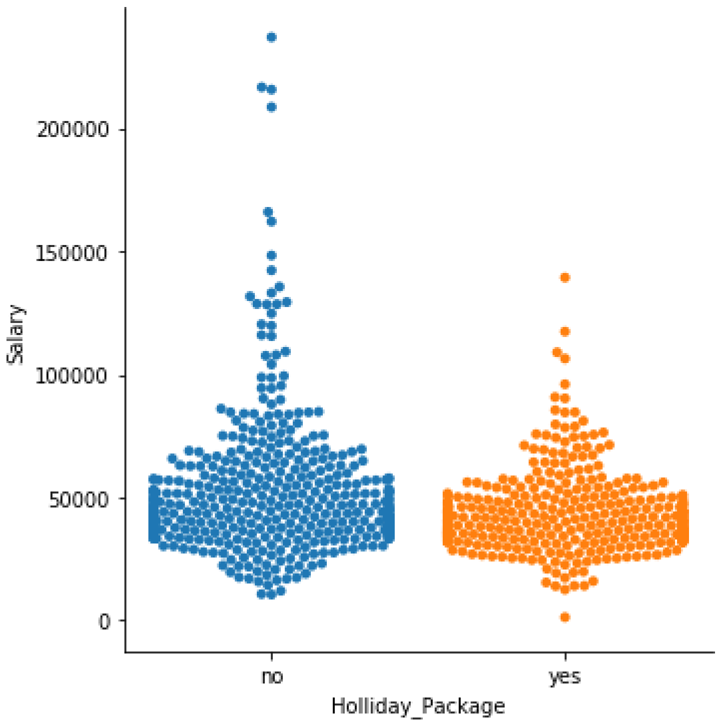
FOREIGN



HOLIDAY PACKAGE

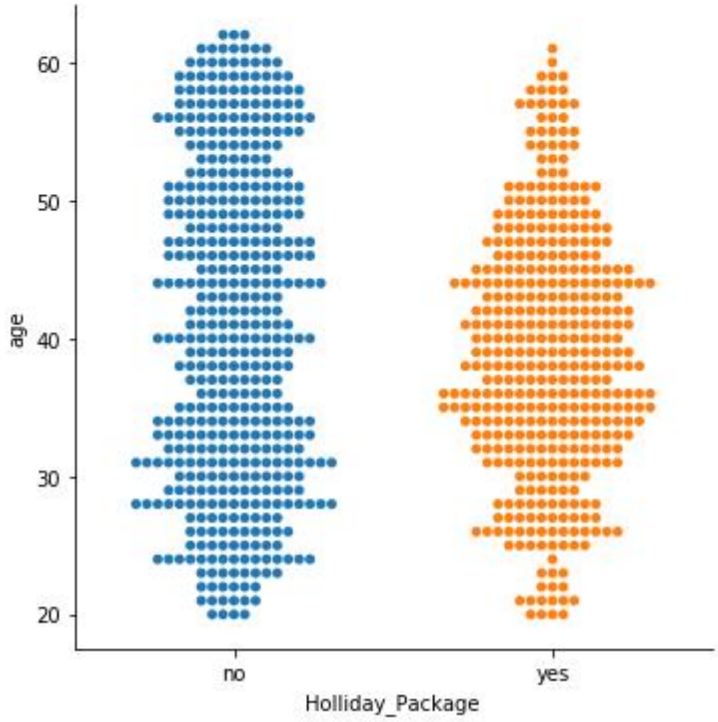


HOLIDAY PACKAGE VS SALARY

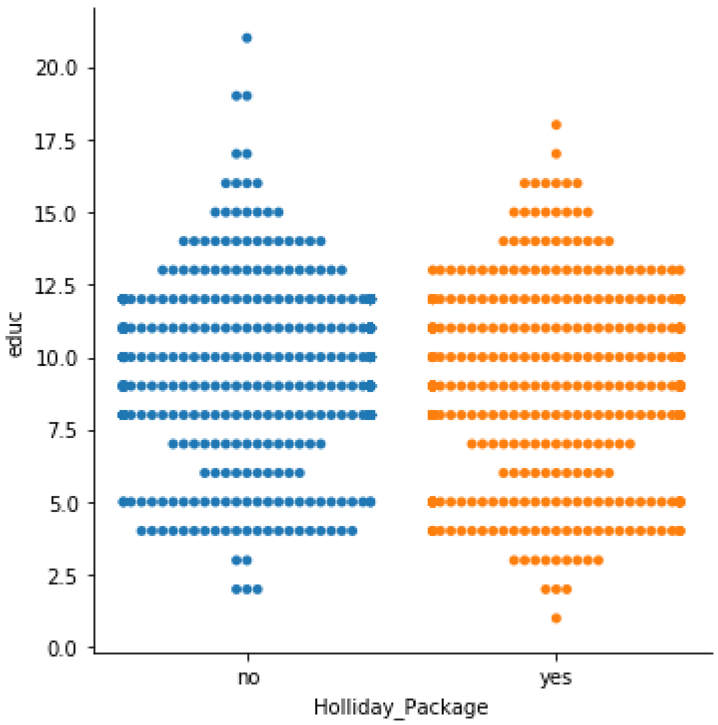


We can see employee below salary 150000 have always opted for holiday package

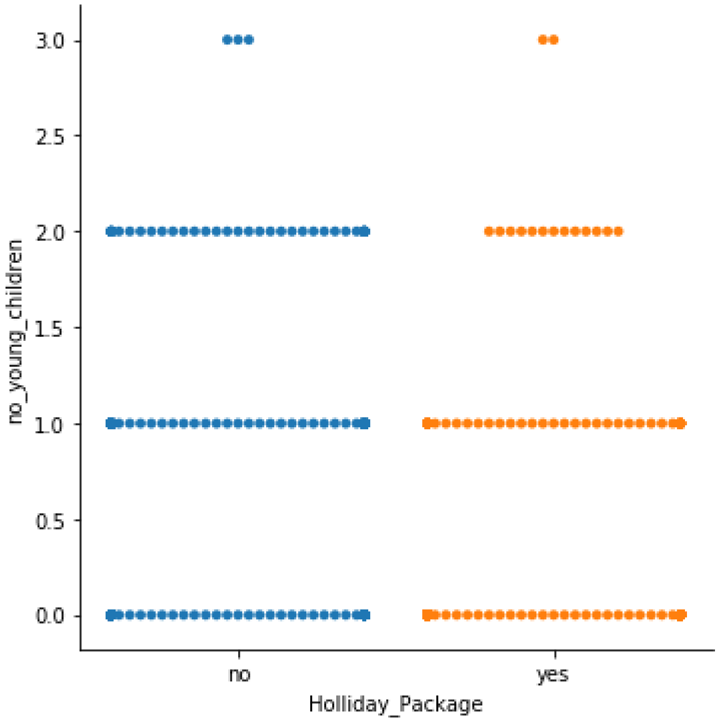
HOLIDAY PACKAGE VS AGE



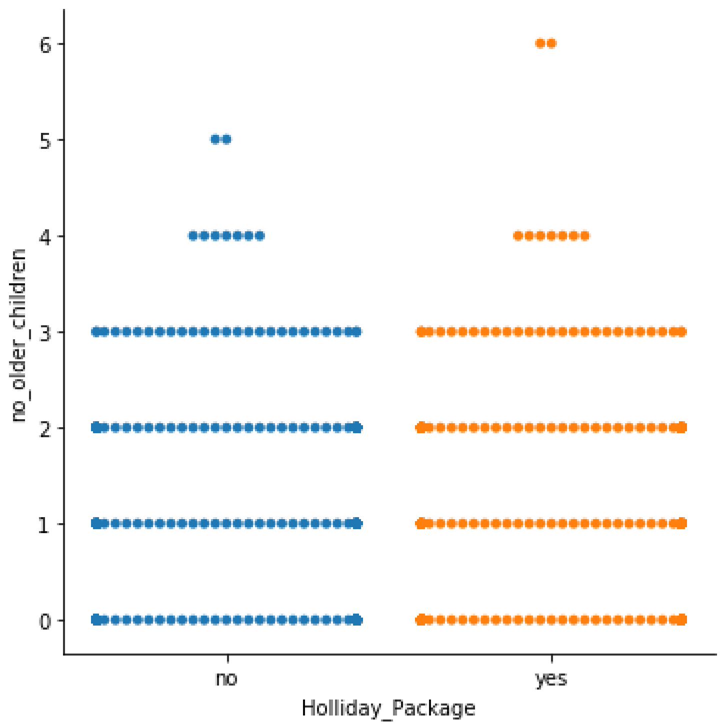
HOLIDAY PACKAGE VS EDUC



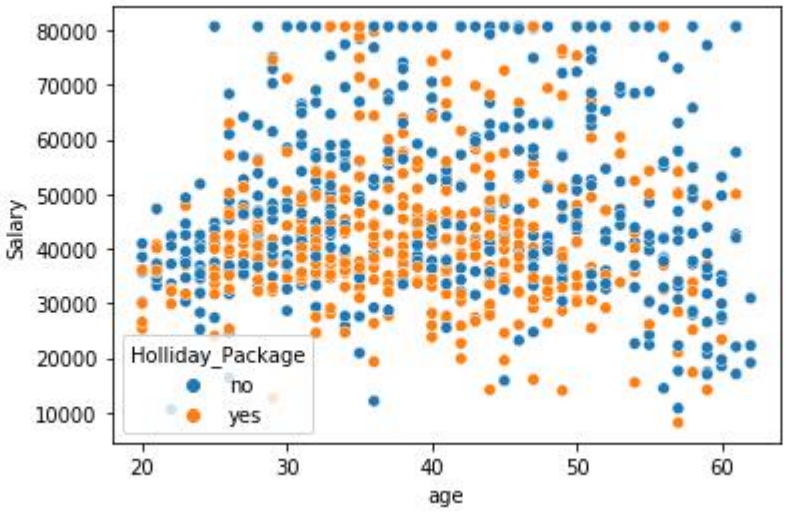
HOLIDAY PACKAGE VS YOUNG CHILDREN

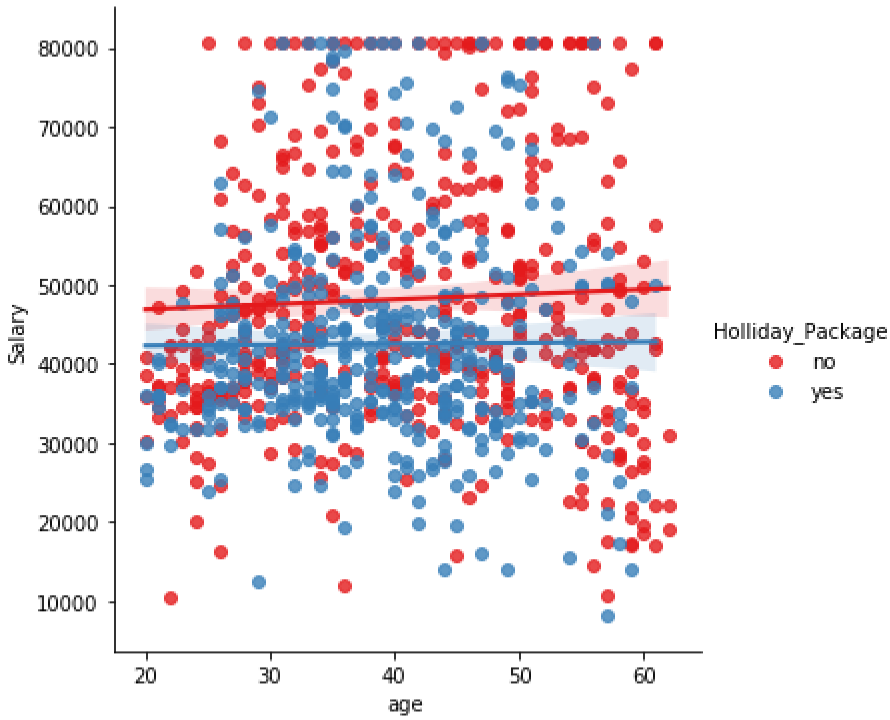


HOLIDAY PACKAGE VS OLDER CHILDREN



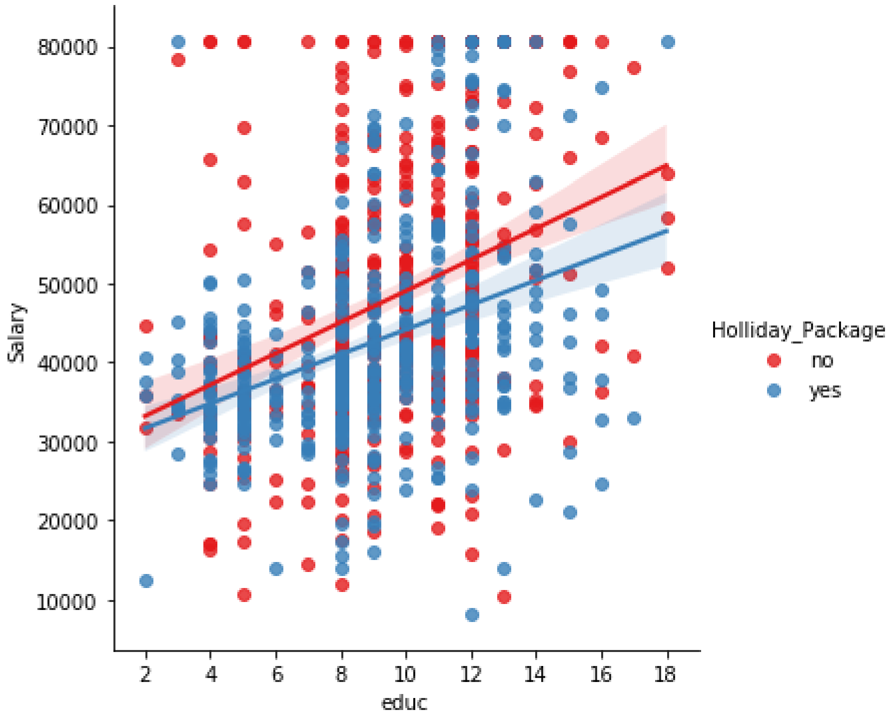
AGE VS SALARY VS HOLIDAY PACKAGE

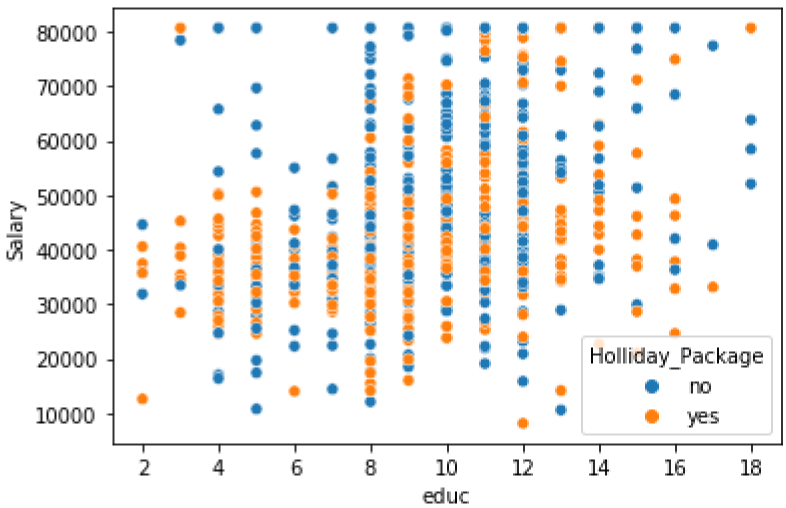




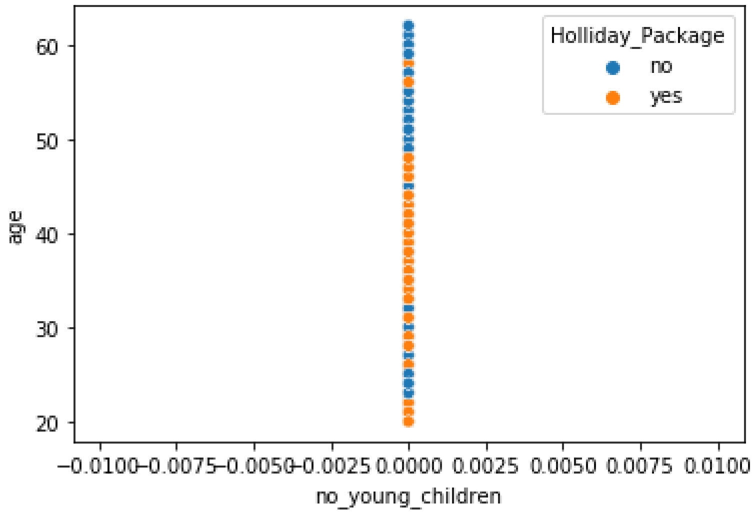
Employee age over 50 to 60 have seems to be not taking the holiday package, whereas in the age 30 to 50 and salary less than 50000 people have opted more for holiday package.

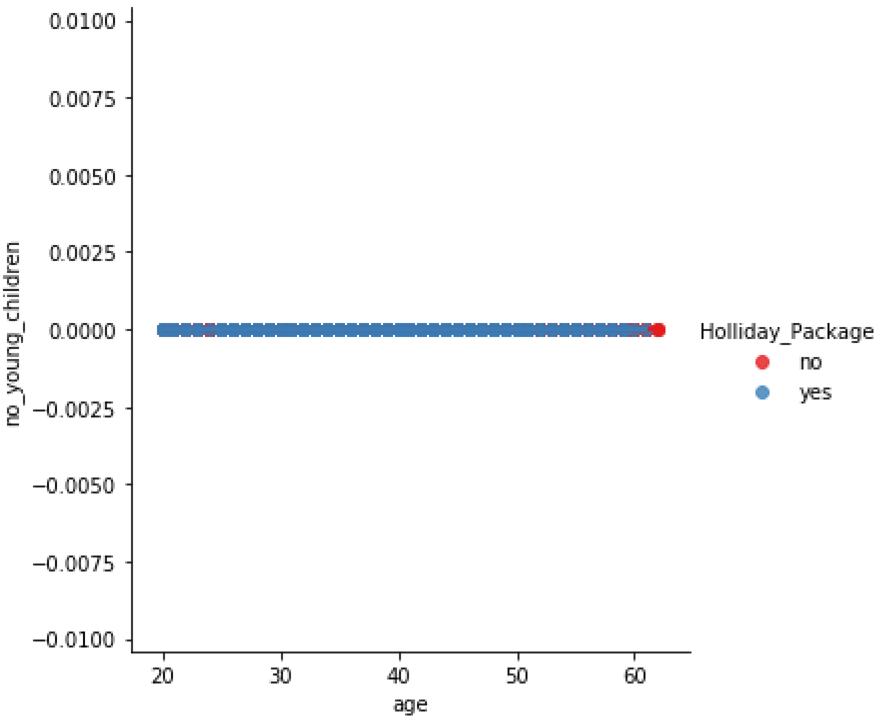
EDUC VS SALARY VS HOLIDAY PACKAGE



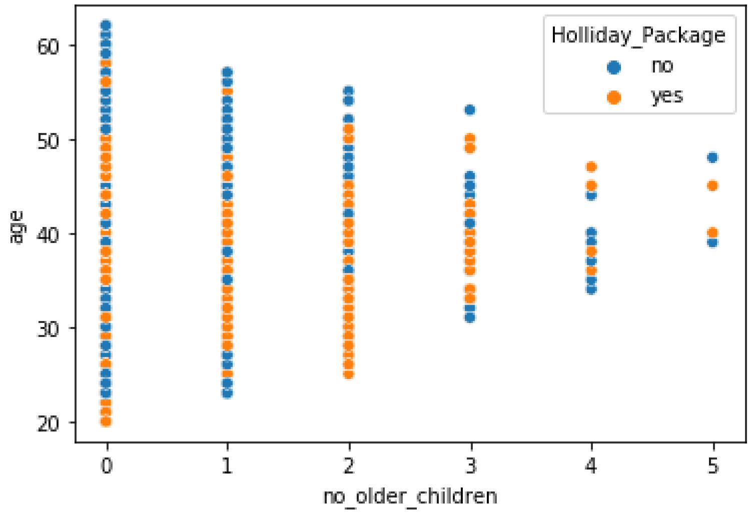


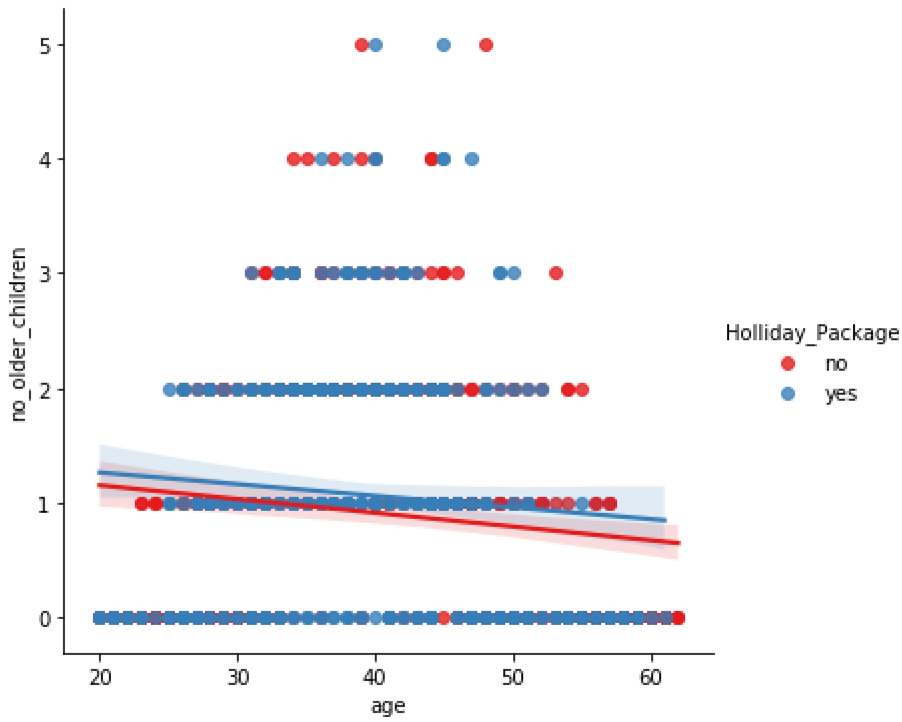
YOUNG CHILDREN VS AGE VS HOLIDAY PACKAGE





OLDER CHILDREN VS AGE VS HOLIDAY_PACKAGE

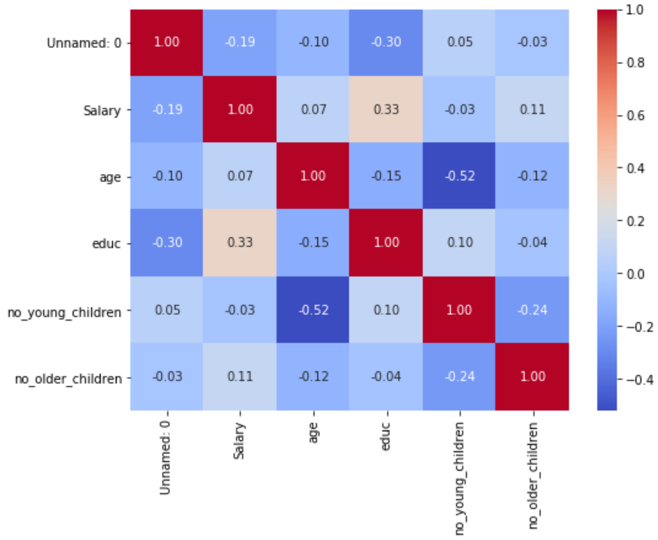




BIVARITE ANALYIS

DATA DISTRIBUTION

There is no correlation between the data, the data seems to be normal. There is no huge difference in the data distribution among the holiday package, I don’t see any clear two different distribution in the data.

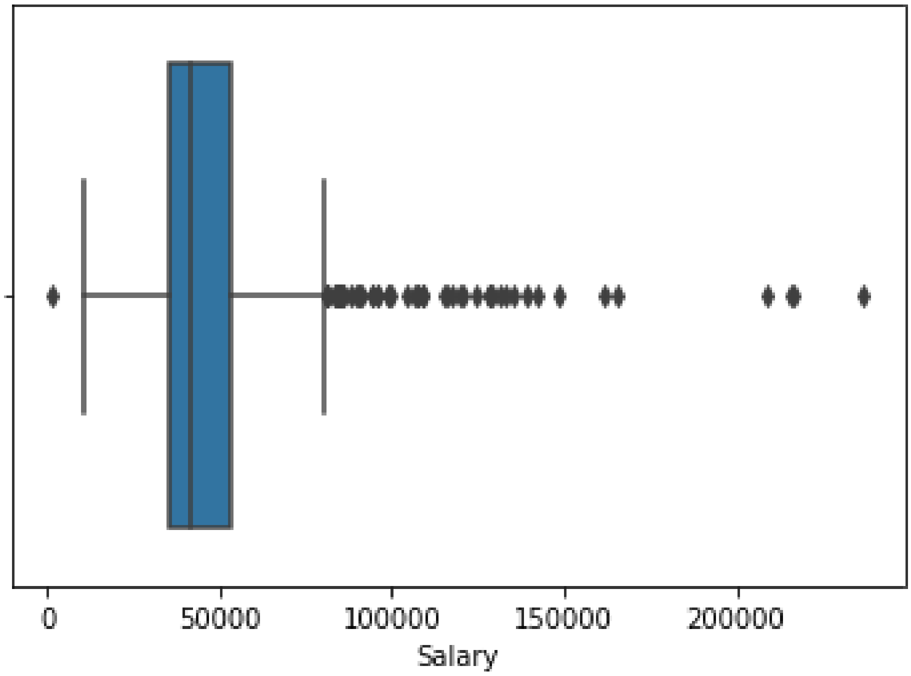


No multi collinearity in the data

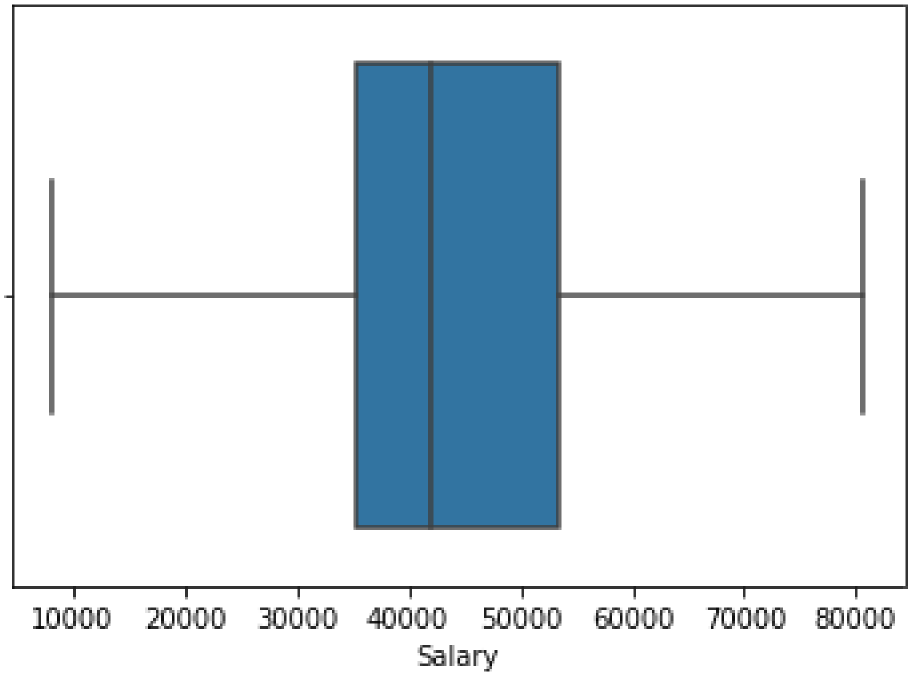
TREATING OUTLIERS

BEFORE OUTLIER TREATMENT

we have outliers in the dataset, as LDA works based on numerical computation treating outliers will help perform the model better.



AFTER OUTLIER TREATMENT



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

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).

ENCODING CATEGORICAL VARIABLE

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

```
data.head()
```

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

The encoding helps the logistic regression model predict better results

Train/ Test split

```
# Copy all the predictor variables into X dataframe
X = data.drop('Holliday_Package_yes', axis=1)

# Copy target into the y dataframe.
y = data['Holliday_Package_yes']
```

```
# Split X and y into training and test set in 70:30 ratio
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30 , random_state=1,stratify=y)
```

GRID SEARCH METHOD:

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

```
grid={'penalty':['l1','l2','none'],
      'solver':['lbfgs', 'liblinear'],
      'tol':[0.0001,0.000001]}
```

```
model = LogisticRegression(max_iter=100000,n_jobs=2)
```

```
grid_search = GridSearchCV(estimator = model, param_grid = grid, cv = 3,n_jobs=-1,scoring='f1')
```

```
print(grid_search.best_params_,'\n')
print(grid_search.best_estimator_)
```

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

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

Tolerance and penalty has been found using grid search method

Predicting the training data,

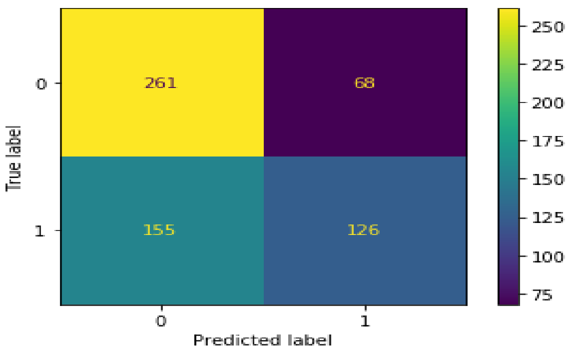
```
# Prediction on the training set

ytrain_predict = best_model.predict(X_train)
ytest_predict = best_model.predict(X_test)
```

```
## Getting the probabilities on the test set

ytest_predict_prob=best_model.predict_proba(X_test)
pd.DataFrame(ytest_predict_prob).head()
```

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

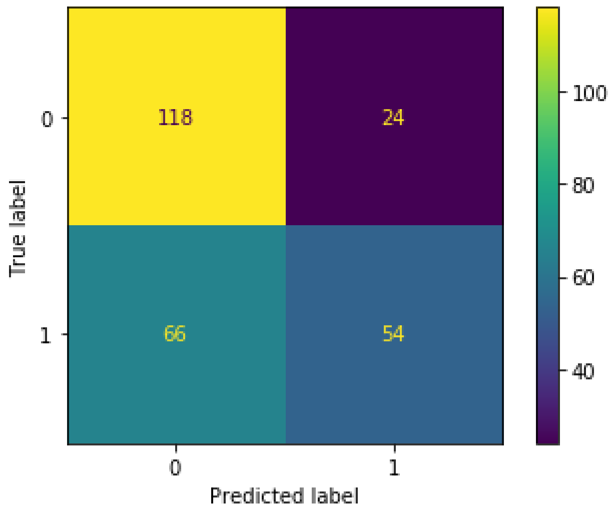
CONFUSION MATRIX TRAIN DATA

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



CONFUSION MATRIX FOR TEST DATA

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



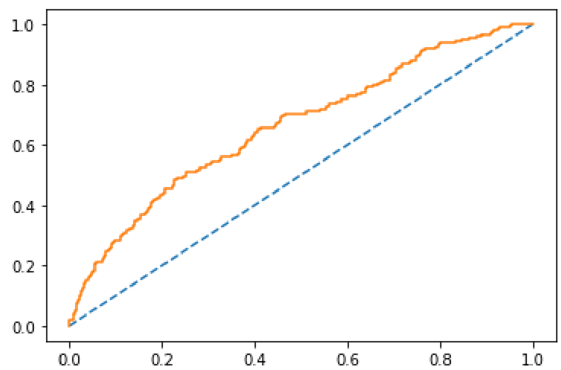
ACCURACY

```
# Accuracy - Training Data
best_model.score(X_train, y_train)
```

```
0.6344262295081967
```

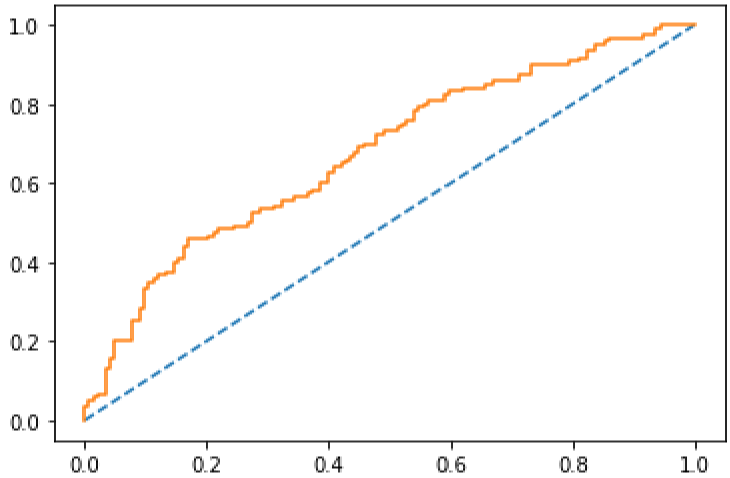
AUC, ROC CURVE FOR TRAIN DATA

```
AUC: 0.661
```



AUC, ROC CURVE FOR TEST DATA

```
AUC: 0.661
```



LDA

```
#Build LDA Model
clf = LinearDiscriminantAnalysis()
model=clf.fit(X_train,Y_train)
```

```
# Training Data Class Prediction with a cut-off value of 0.5
pred_class_train = model.predict(X_train)

# Test Data Class Prediction with a cut-off value of 0.5
pred_class_test = model.predict(X_test)
```

PREDICTING THE VARIBALE

```
# Training Data Probability Prediction
pred_prob_train = model.predict_proba(X_train)

# Test Data Probability Prediction
pred_prob_test = model.predict_proba(X_test)
```

MODEL SCORE

```
model.score(X_train,Y_train)
```

```
0.6327868852459017
```

CLASSFICATION REPORT TRAIN DATA

```
print(classification_report(Y_train, pred_class_train))
```

```
              precision    recall  f1-score   support

           0       0.62      0.80      0.70       329
           1       0.65      0.44      0.52       281

    accuracy                           0.63       610
   macro avg       0.64      0.62      0.61       610
weighted avg       0.64      0.63      0.62       610
```

```
confusion_matrix(Y_train, pred_class_train)
```

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

MODEL SCORE

```
model.score(X_test,Y_test)
```

```
0.6564885496183206
```

CLASSIFICATION REPORT TEST DATA

```
print(classification_report(Y_test, pred_class_test))
```

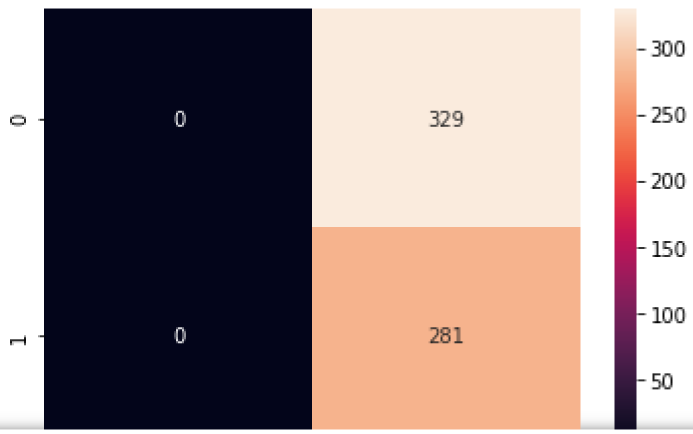
```
              precision    recall  f1-score   support

           0       0.64      0.83      0.72       142
           1       0.69      0.45      0.55       120

    accuracy                           0.66       262
   macro avg       0.67      0.64      0.63       262
weighted avg       0.66      0.66      0.64       262
```

```
confusion_matrix(Y_test, pred_class_test)
```

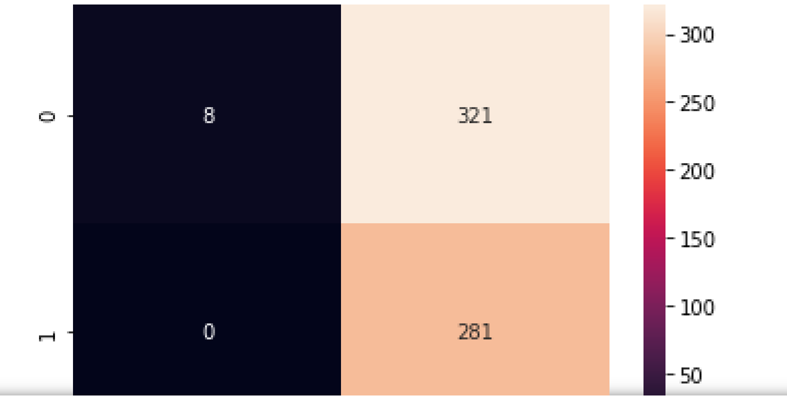
```
array([[118,  24],
       [ 66,  54]], dtype=int64)
```

CHANGING THE CUTT OFF VALUE TO CHECK OPTIMAL VALUE THAT GIVES BETTER ACCURACY AND F1 SCORE

```
0.1

Accuracy Score 0.4607
F1 Score 0.6308

Confusion Matrix
```



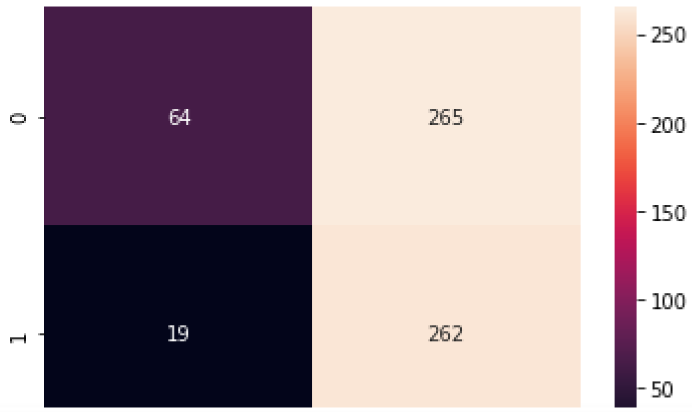
```
0.2

Accuracy Score 0.4738
F1 Score 0.6365

Confusion Matrix
```



0.3

Accuracy Score 0.5344
F1 Score 0.6485

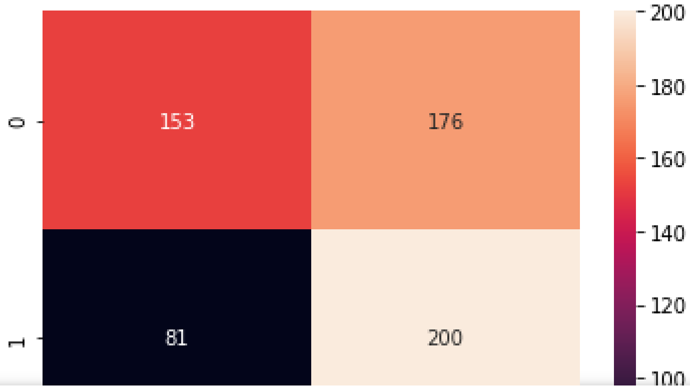
Confusion Matrix



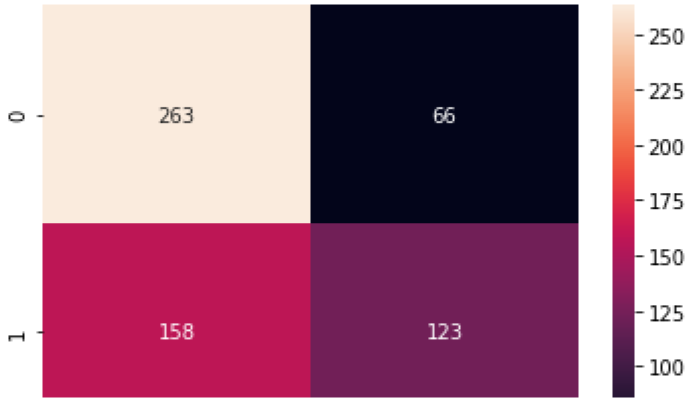
0.4

Accuracy Score 0.5787
F1 Score 0.6088

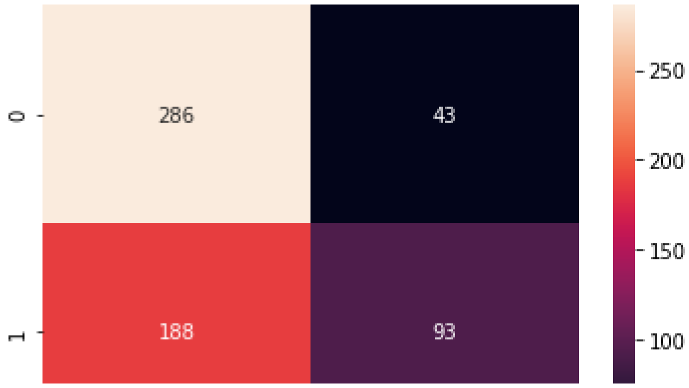
Confusion Matrix



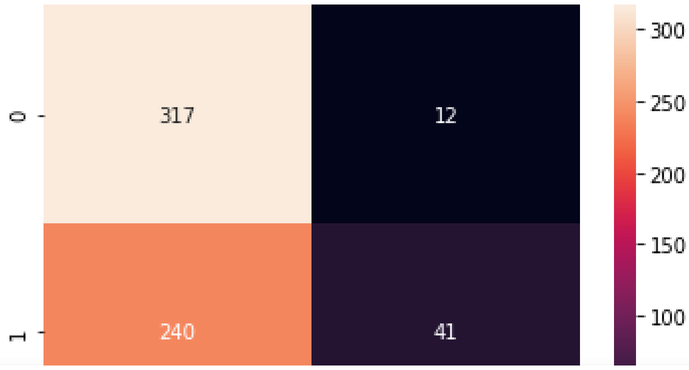
```
0.5

Accuracy Score 0.6328
F1 Score 0.5234

Confusion Matrix
```



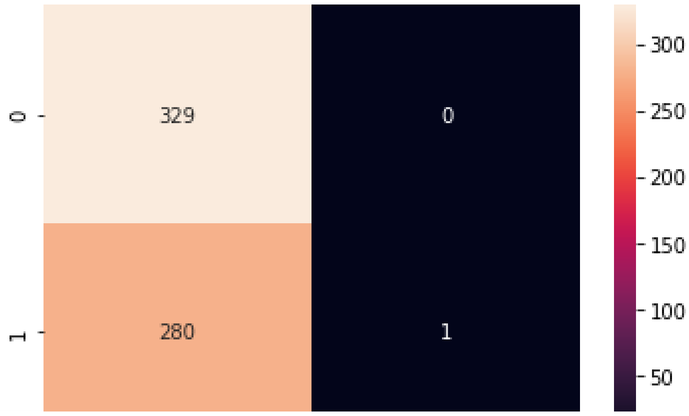

```
0.6

Accuracy Score 0.6213
F1 Score 0.446

Confusion Matrix
```



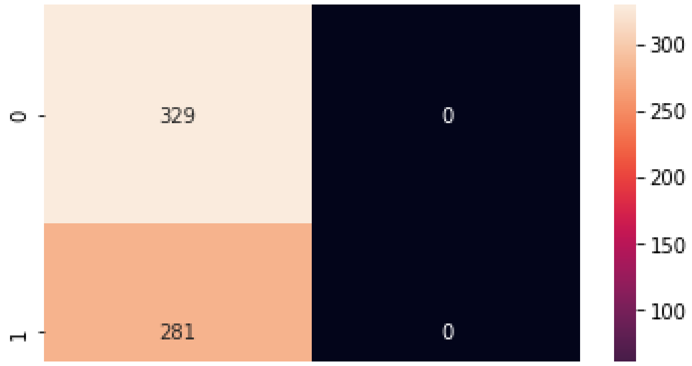
```
0.7

Accuracy Score 0.5869
F1 Score 0.2455

Confusion Matrix
```




```
0.8

Accuracy Score 0.541
F1 Score 0.0071

Confusion Matrix
```

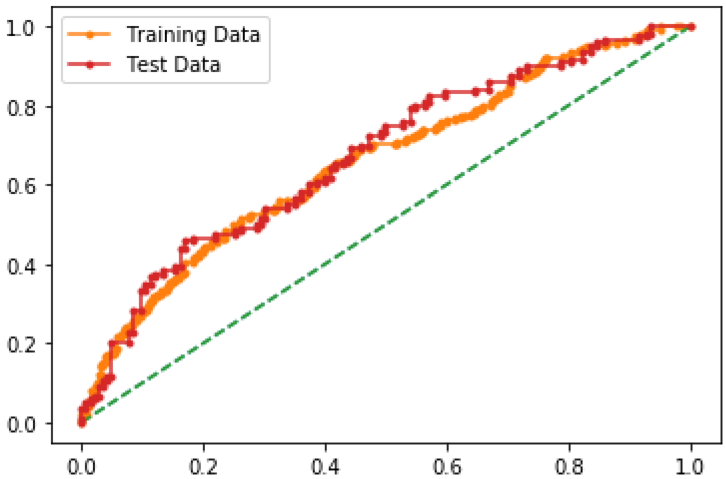


```
0.9

Accuracy Score 0.5393
F1 Score 0.0

Confusion Matrix
```



AUC AND ROC CURVE

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



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

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

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

We had a business problem where we need predict whether an employee would opt for a holiday package or not, for this problem we had done predictions both logistic regression and linear discriminant analysis. Since both are results are same.
The EDA analysis clearly indicates certain criteria where we could find people aged above 50 are not interested much in holiday packages.
So this is one of the we find aged people not opting for holiday packages.
People ranging from the age 30 to 50 generally opt for holiday packages.
Employee age over 50 to 60 have seems to be not taking the holiday package, whereas in the age 30 to 50 and salary less than 50000 people have opted more for holiday package.

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

Recommendations

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

THE END