CAPSTONE PROJECT BUSINESS REPORT

Tourism Package Adoption

JULY 11, 2021
FINAL BUSINESS REPORT-TOURISM
Srikanthpr.27@gmail.com

Contents

1. Introduction: Tourism Package Adoption

- Brief introduction about the problem statement and the need of solving it.

Business Problem:

- A reputed tourism company is planning to launch a long term travel package for the customers.
- Project manager have access to existing customer details and information collected from previous campaign. He wishes to analyse the behaviour of the customers to figure out which customer is going to purchase the long term travel package.

Need of study/project:

1. **EDA:**
   - Identify which customers will purchase long term travel package.
   - Understand the relationship between the product taken [Target variable] and the customer and sales attributes [Independent variables].
   - Understand the interaction between the customer demographics and the company sales marketing strategy.
   - Explore and visualize the dataset using central tendency and other statistical parameters.
2. **Model building Approach:**
   - Build various classification models to predict whether a customer will buy a tourism package or not.
   - Models such as linear based models, tree based models, distance based models or ensemble models can be built over for model prediction.
   - Company is interested in customers who purchased travel package in actual, want the model to predict the same. This will increase the sales and bring good revenue to the company.
   - Finally understand which are the most significant variables?
3. **Business Recommendations:**
   - Generate a set of insights and business recommendations to the business or client based on the analysis done which will help the business to increase their sales revenue.

2. EDA and Business Implication:

- Uni-variate / Bi-variate / Multi-variate analysis to understand relationship b/w variables.
- How your analysis is impacting the business?
- Both visual and non-visual understanding of the data.

Data Understanding:

Variable	Description
Customer Details	
CustomerID	Unique customer ID
ProdTaken	Product taken flag
Age	Age of customer
Gender	Gender of customer
MaritalStatus	Marital status of customer
Occupation	Occupation of customer
Designation	Designation of customer in current organization
Monthly Income	Gross monthly income of customer
Passport	Customer passport flag
OwnCar	Customers owns a car flag
CityTier	City tier
Customer Preference	
PreferredPropertyStar	Preferred hotel property rating by customer
PreferredLoginDevice	Preferred login device of customer in last month
NumberOfPersonVisited	Total number of person came with customer
NumberofChildrenVisited	Total number of children visit with customer
NumberofTrips	Average number of trip in a year by customer
Customer Interaction	
DurationOfPitch	Duration of pitch by a sales man to customer
NumberofFollowups	Total number of follow up has been done by sales person after sales pitch
ProductPitched	Product pitched by sales person
PitchSatisfactionScore	Sales pitch satisfactory score

Table-1: Data Description

- Import necessary packages/libraries and load the tourism dataset. Also copy entire dataset to another variable to avoid any changes to original dataset.

Transpose View of top 5 rows of Tourism dataset :

	0	1	2	3	4
CustomerID	200000	200001	200002	200003	200004
ProdTaken	1	0	1	0	0
Age	41	49	37	33	NaN
PreferredLoginDevice	Self Enquiry	Company Invited	Self Enquiry	Company Invited	Self Enquiry
CityTier	3	1	1	1	1
DurationOfPitch	6	14	8	9	8
Occupation	Salaried	Salaried	Free Lancer	Salaried	Small Business
Gender	Female	Male	Male	Female	Male
NumberOfPersonVisited	3	3	3	2	2
NumberOfFollowups	3	4	4	3	3
ProductPitched	Super Deluxe	Super Deluxe	Multi	Multi	Multi
PreferredPropertyStar	3	4	3	3	4
MaritalStatus	Single	Divorced	Single	Divorced	Divorced
NumberOfTrips	1	2	7	2	1
Passport	1	0	1	1	0
PitchSatisfactionScore	2	3	3	5	5
OwnCar	1	1	0	1	1
NumberOfChildrenVisited	0	2	0	1	0
Designation	Manager	Manager	Executive	Executive	Executive
MonthlyIncome	20993	20130	17090	17909	18468

Table-2: Data-Frame head with top 5 rows

- Tourism dataset has 4888 records/observations and 20 variables/features including the target variable.
- To understand the data-types, missing values of the variables, we look into info of the dataset.

Data-structure of the variables for tourism dataset :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4888 entries, 0 to 4887
Data columns (total 20 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   CustomerID               4888 non-null   int64
 1   ProdTaken                4888 non-null   int64
 2   Age                      4662 non-null   float64
 3   PreferredLoginDevice     4863 non-null   object
 4   CityTier                 4888 non-null   int64
 5   DurationOfPitch          4637 non-null   float64
 6   Occupation               4888 non-null   object
 7   Gender                   4888 non-null   object
 8   NumberOfPersonVisited    4888 non-null   int64
 9   NumberOfFollowups        4843 non-null   float64
 10  ProductPitched           4888 non-null   object
 11  PreferredPropertyStar    4862 non-null   float64
 12  MaritalStatus            4888 non-null   object
 13  NumberOfTrips            4748 non-null   float64
 14  Passport                 4888 non-null   int64
 15  PitchSatisfactionScore   4888 non-null   int64
 16  OwnCar                   4888 non-null   int64
 17  NumberOfChildrenVisited  4822 non-null   float64
 18  Designation              4888 non-null   object
 19  MonthlyIncome            4655 non-null   float64
dtypes: float64(7), int64(7), object(6)
memory usage: 763.9+ KB
```

Table-3a: Dataset Information before fixing Data-types

Out of 20, 7 columns are of float, 6 columns are of object and 7 are of integer data type.

- CustomerID is a unique identifier which can be dropped from the dataset before model building.
- ProdTaken is a target variables which is of binary category data-type need to change the data-type from integer to category data-type.
- 3 variables are of continuous numerical data and all of them have missing values.
  - Age, Monthly Income and Duration of pitch.
- 4 variables are of ordinal data and all of them need to change the data-type to category.
  - City tier: 3 Levels [1, 2, 3]
  - Preferred Property star: 3 Levels [3,4,5] – Missing values present
  - Pitch Satisfaction Score: 5 Levels [1,2,3,4,5]
  - Designation: 5 Levels [Executive, Manager, Senior Manager, AVP, VP]

- Number of Follow-ups and Number of trips variable is of continuous data-type needs to be converted to categorical data-type by creating bins using quantile method or manual method.
- Remaining 9 variables the data type needs to be changed to category data-type.

Data-structure of the variables for tourism dataset after fixing the data-type:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4888 entries, 0 to 4887
Data columns (total 20 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   CustomerID                4888 non-null   int64
 1   ProdTaken                 4888 non-null   category
 2   Age                       4662 non-null   float64
 3   PreferredLoginDevice      4863 non-null   category
 4   CityTier                  4888 non-null   category
 5   DurationOfPitch           4637 non-null   float64
 6   Occupation                4888 non-null   category
 7   Gender                    4888 non-null   category
 8   NumberOfPersonVisited     4888 non-null   category
 9   NumberOfFollowups         4843 non-null   category
 10  ProductPitched            4888 non-null   category
 11  PreferredPropertyStar     4862 non-null   category
 12  MaritalStatus             4888 non-null   category
 13  NumberOfTrips             4748 non-null   category
 14  Passport                  4888 non-null   category
 15  PitchSatisfactionScore    4888 non-null   category
 16  OwnCar                    4888 non-null   category
 17  NumberOfChildrenVisited   4822 non-null   category
 18  Designation               4888 non-null   category
 19  MonthlyIncome             4655 non-null   float64
dtypes: category(16), float64(3), int64(1)
memory usage: 231.5 KB
```

Table-3b: Dataset Information after fixing Data-types

- Tourism dataset do not have any duplicate records in it.
- Tourism dataset has few missing values present need to impute missing values before building models.

The number of Null values in Tourism dataset:

```
CustomerID                 0
ProdTaken                  0
Age                      226
PreferredLoginDevice      25
CityTier                   0
DurationOfPitch          251
Occupation                 0
Gender                     0
NumberOfPersonVisited      0
NumberOfFollowups         45
ProductPitched             0
PreferredPropertyStar     26
MaritalStatus              0
NumberOfTrips            140
Passport                   0
PitchSatisfactionScore     0
OwnCar                     0
NumberOfChildrenVisited   66
Designation                0
MonthlyIncome            233
dtype: int64
```

Table-4: Missing Values

- Monthly Income and Duration of Pitch contains few outliers need to take care or understand before building models.

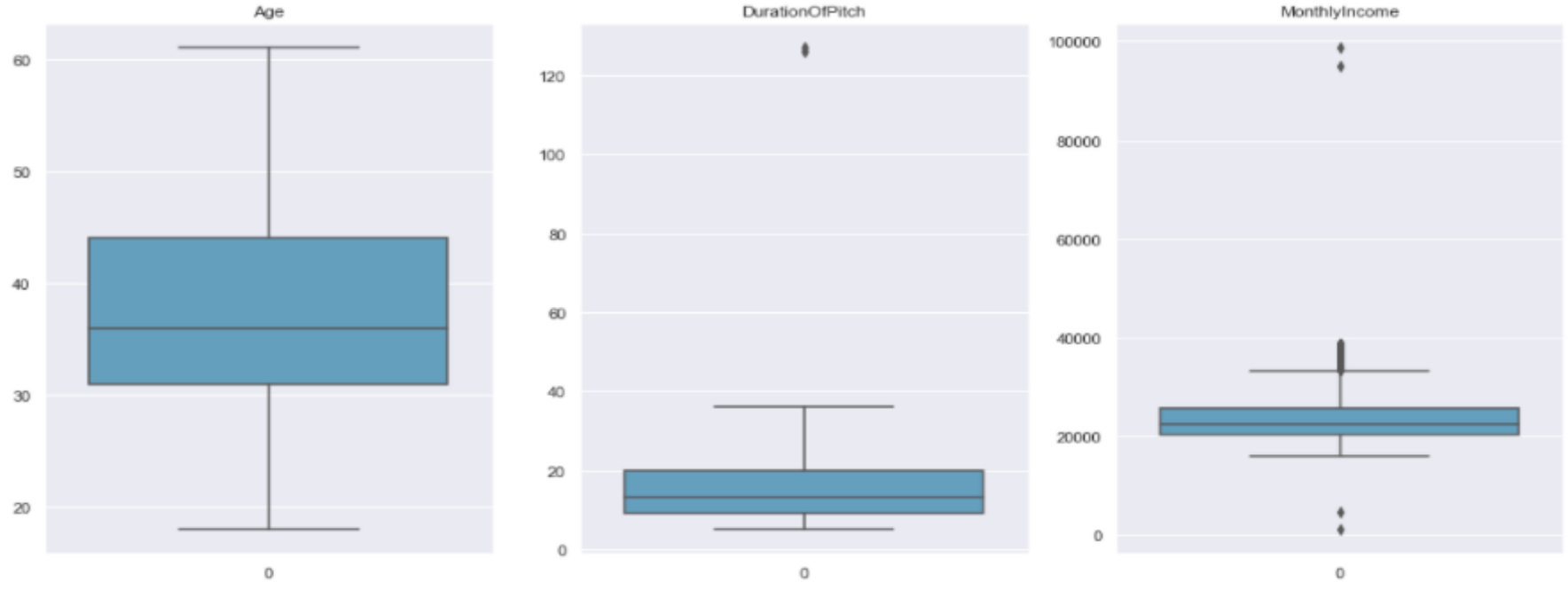


Figure-1: Outlier Detection

Exploratory Data Analysis:

1. Descriptive Analysis:

The Summary statistics of continuous variables before Data-cleaning:

	Age	DurationOfPitch	MonthlyIncome
count	4662.00	4637.00	4655.00
mean	37.62	15.49	23619.85
std	9.32	8.52	5380.70
min	18.00	5.00	1000.00
25%	31.00	9.00	20346.00
50%	36.00	13.00	22347.00
75%	44.00	20.00	25571.00
max	61.00	127.00	98678.00

The Summary statistics of continuous variables after Outlier and Missing value treatment:

	Age	DurationOfPitch	MonthlyIncome
count	4888.00	4888.00	4888.00
mean	37.44	15.32	23538.66
std	9.15	8.01	5026.86
min	18.00	5.00	16009.00
25%	31.00	9.00	20486.75
50%	36.00	13.00	22347.00
75%	43.00	19.00	25407.75
max	61.00	36.00	38677.00

The Summary statistics of categorical variables before Data-cleaning:

	count	unique	top	freq
ProdTaken	4888	2	0	3968
PreferredLoginDevice	4863	2	Self Enquiry	3444
CityTier	4888	3	1	3190
Occupation	4888	4	Salaried	2368
Gender	4888	3	Male	2916
NumberOfPersonVisited	4888	5	3	2402
NumberOfFollowups	4843	2	Max	2972
ProductPitched	4888	5	Multi	1842
PreferredPropertyStar	4862	3	3	2993
MaritalStatus	4888	4	Married	2340
NumberOfTrips	4748	3	Min	2084
Passport	4888	2	0	3466
PitchSatisfactionScore	4888	5	3	1478
OwnCar	4888	2	1	3032
NumberOfChildrenVisited	4822	4	1	2080
Designation	4888	5	Executive	1842

The Summary statistics of categorical variables after Missing value treatment:

	count	unique	top	freq
ProdTaken	4888	2	0	3968
PreferredLoginDevice	4888	2	Self Enquiry	3469
CityTier	4888	3	1	3190
Occupation	4888	4	Salaried	2368
Gender	4888	2	Male	2916
NumberOfPersonVisited	4888	5	3	2402
NumberOfFollowups	4888	2	Max	3017
ProductPitched	4888	5	Multi	1842
PreferredPropertyStar	4888	3	3	3019
MaritalStatus	4888	4	Married	2340
NumberOfTrips	4888	3	Min	2224
Passport	4888	2	0	3466
PitchSatisfactionScore	4888	5	3	1478
OwnCar	4888	2	1	3032
NumberOfChildrenVisited	4888	4	1	2080
Designation	4888	5	Executive	1842

Table-5: Statistical Description using the describe () method before and after Outlier and Missing Values Treatment

- **'ProdTaken'** is our target variable. Majority of the customers have not taken the travel package.

Customer-Details:

- Majority of the customer belongs to 31-44 Age group. Age is normally distributed data.
- Most of the customers are male and are married.
- Most of the customer's occupation type are salaried and their designation is executive.
- Monthly Income vary from 1000 to 98678. Variance is too high. Need to deal with Outliers.
- Most of the customers do not have a passport.
- Most of the customers have their own car.
- Most of the customers are from City-Tier 1.

Customer-Preference:

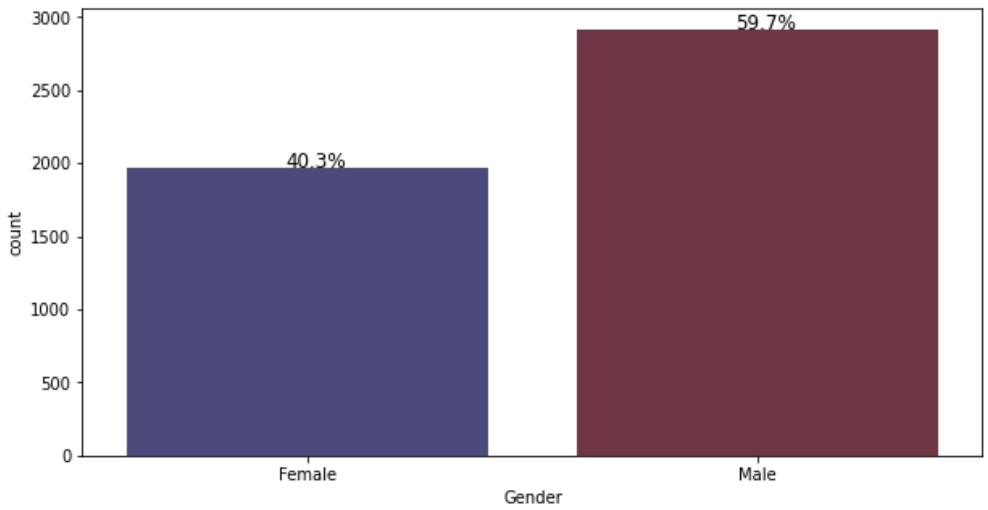
- Most of the customer preferred 3 star rating property.
- Most of the customers came into company contact by doing a self-Enquiry and number of person visited along with customer is 3, also the number of children visited along with customer is 1.
- Average number of trips in a year by most of the customers is minimum that is either 1 or 2.

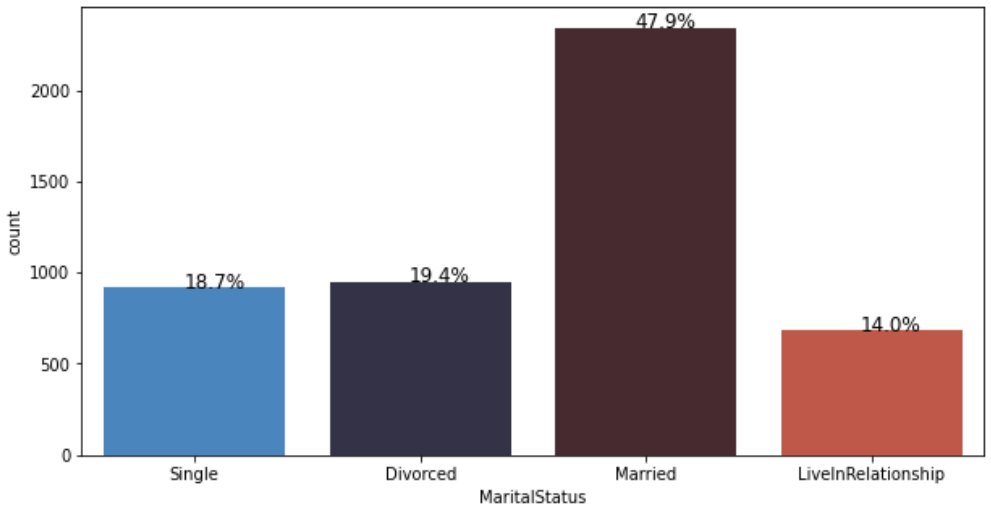
Customer-Interaction:

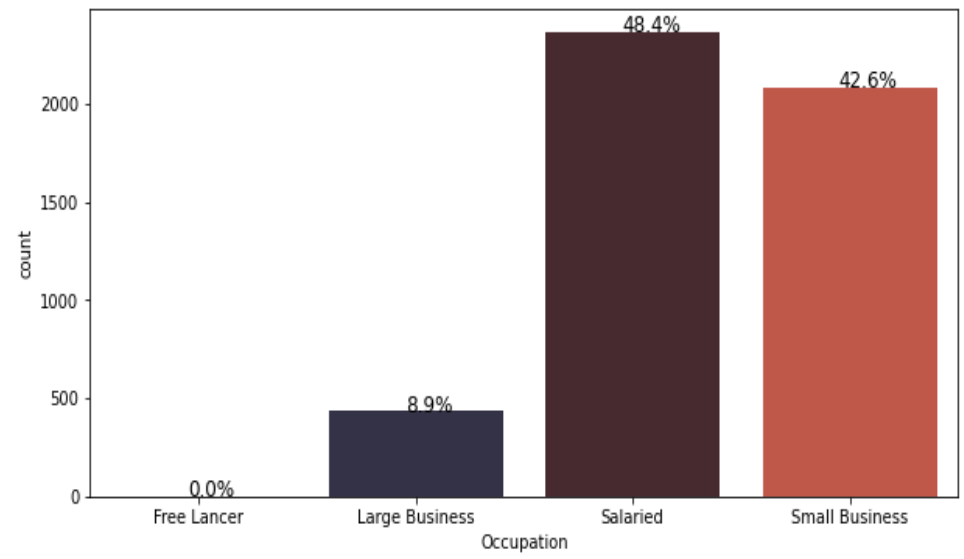
- Maximum number of follow-ups have been made by sales person to most of the customers.
- Product 'Multi' is pitched by sales person most of the times.
- Sales pitch satisfactory score for most of the sales person is 3.
- Duration of Sales pitch to majority of the customer is average. (10 - 20 minutes). Need to deal with few Outlier.

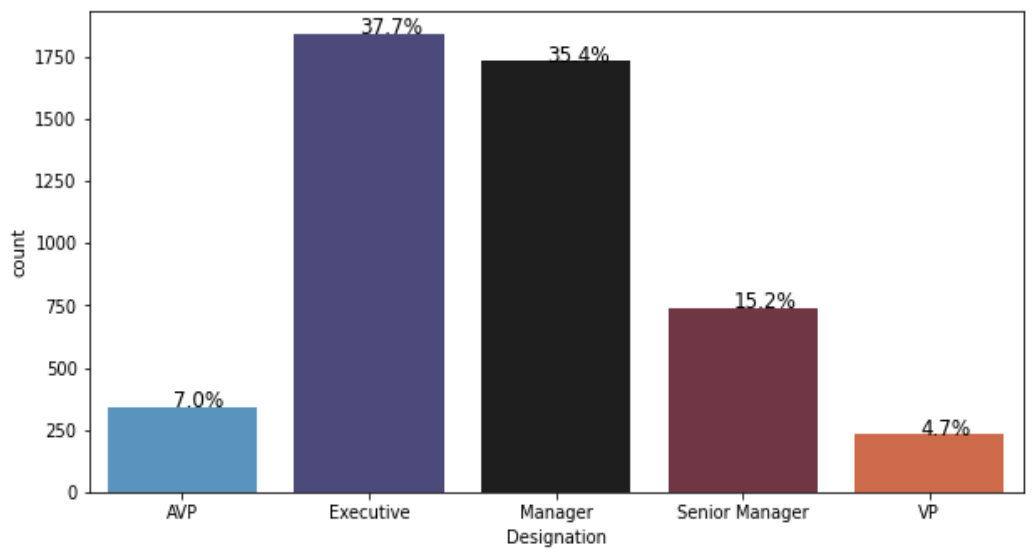
2. Univariate Analysis:

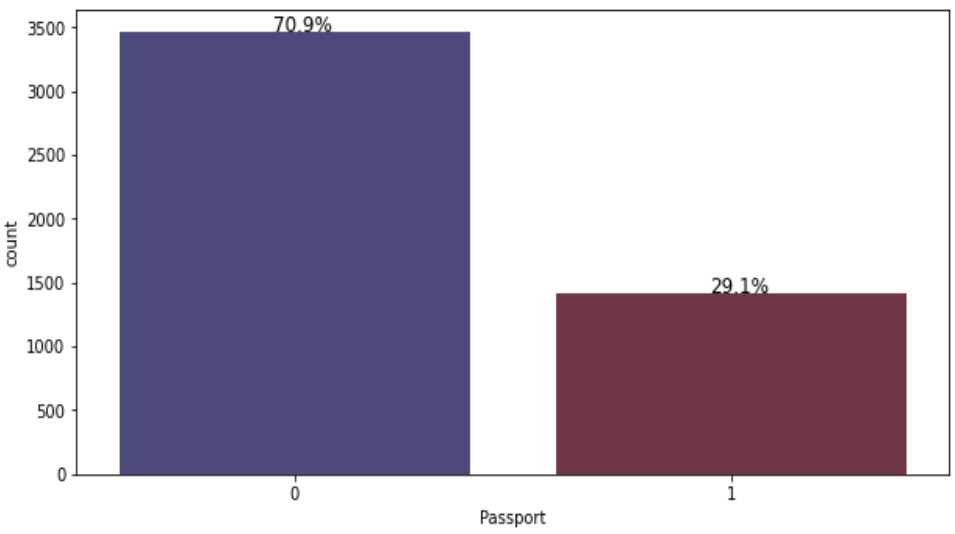
Univariate Analysis on Customer Details:

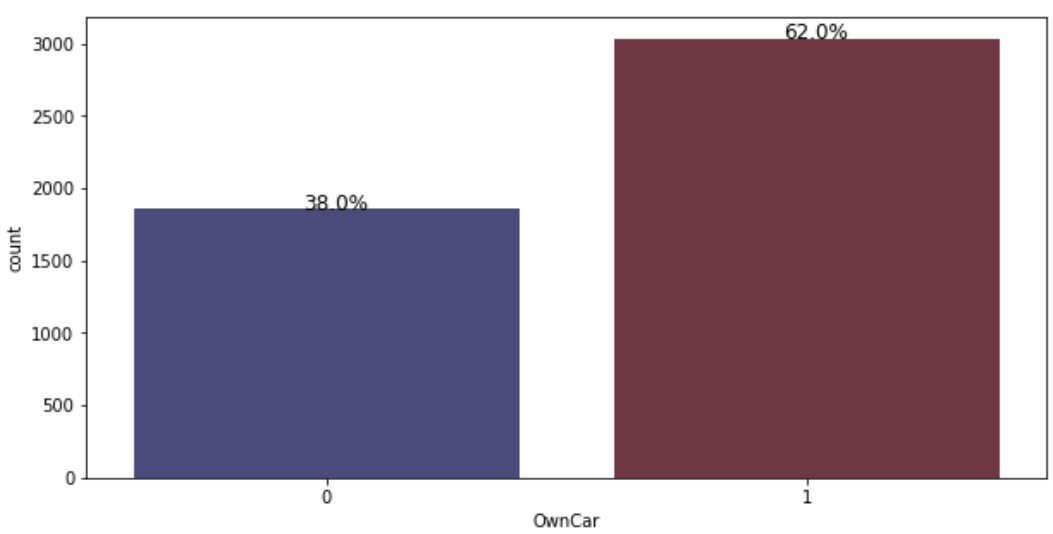












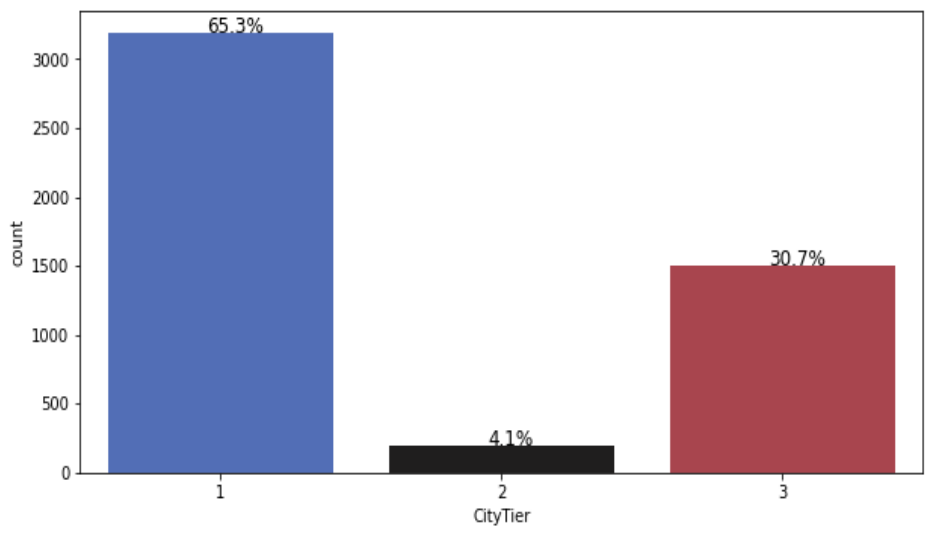


Figure-2a: Count plots of Categorical Variables

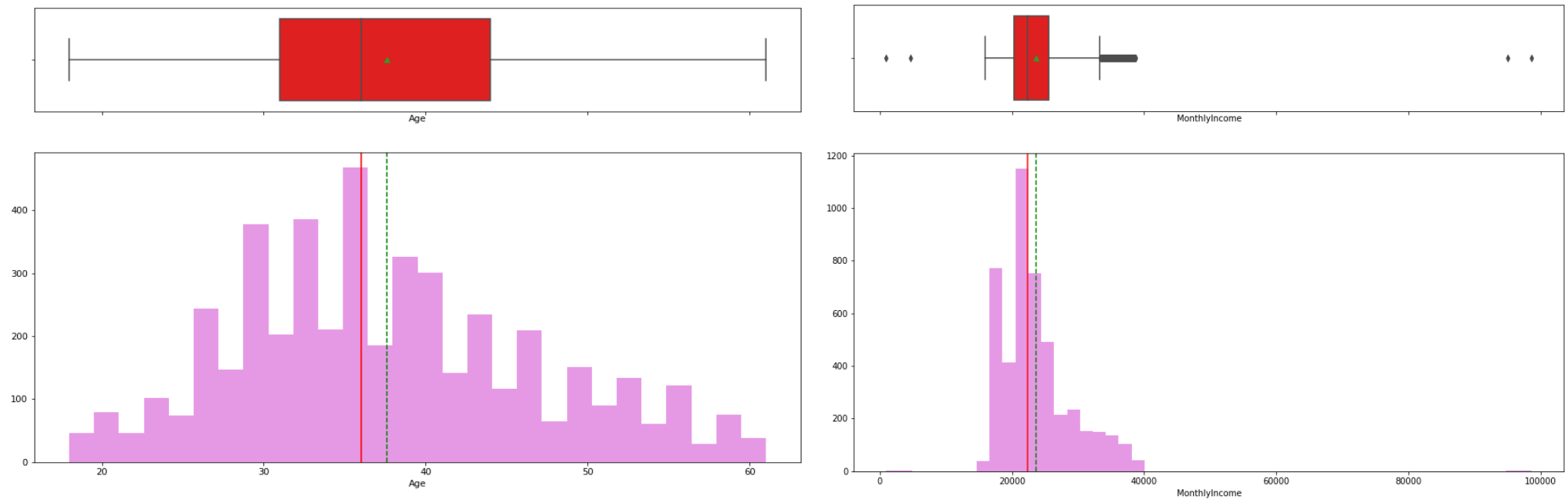


Figure-2b: Boxplot and Histogram plot on Age and Monthly Income Variable

Univariate Analysis on Customer Preference:

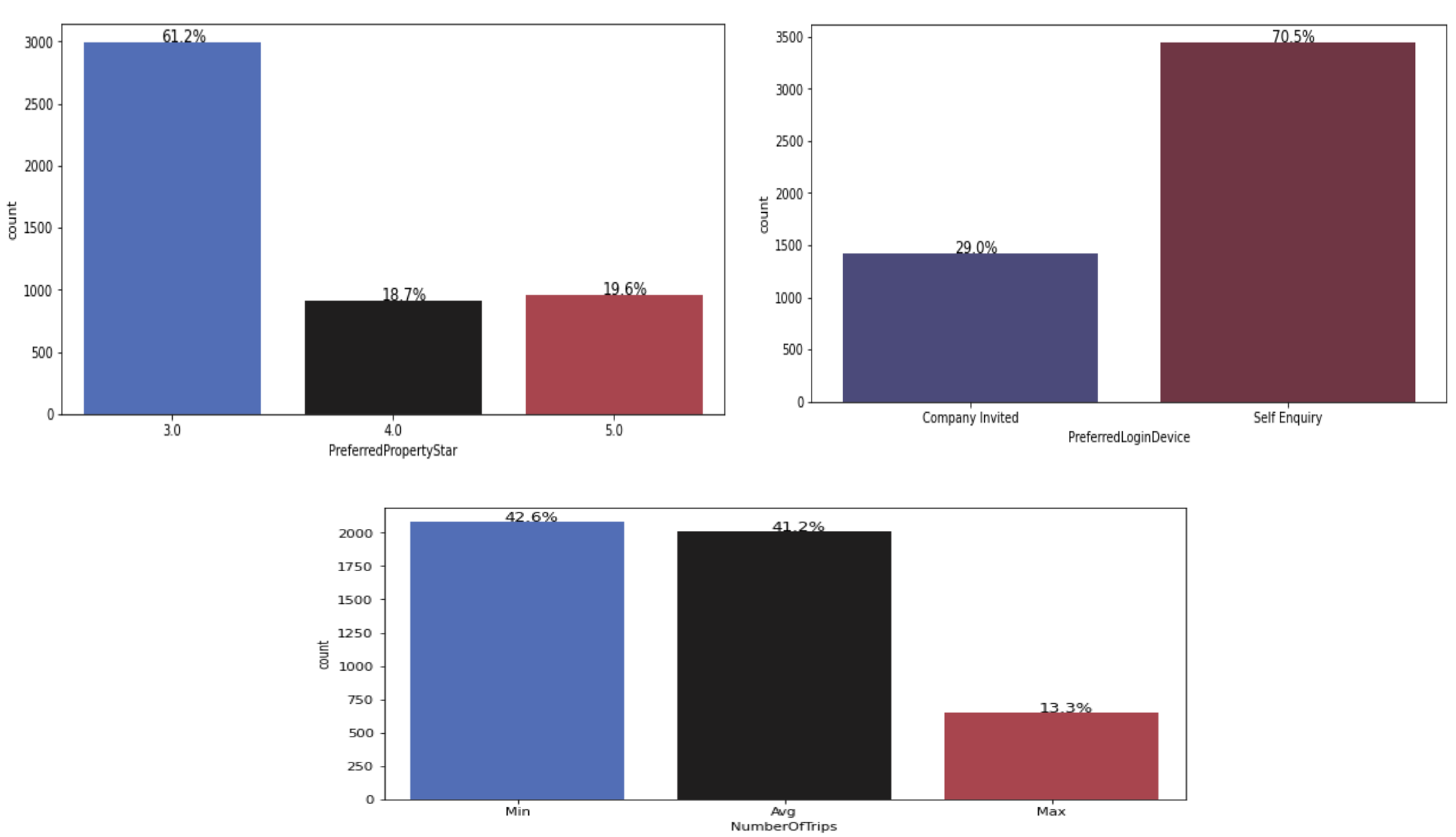


Figure-2c: Count plots of Categorical Variables

Univariate Analysis on Customer Interaction:

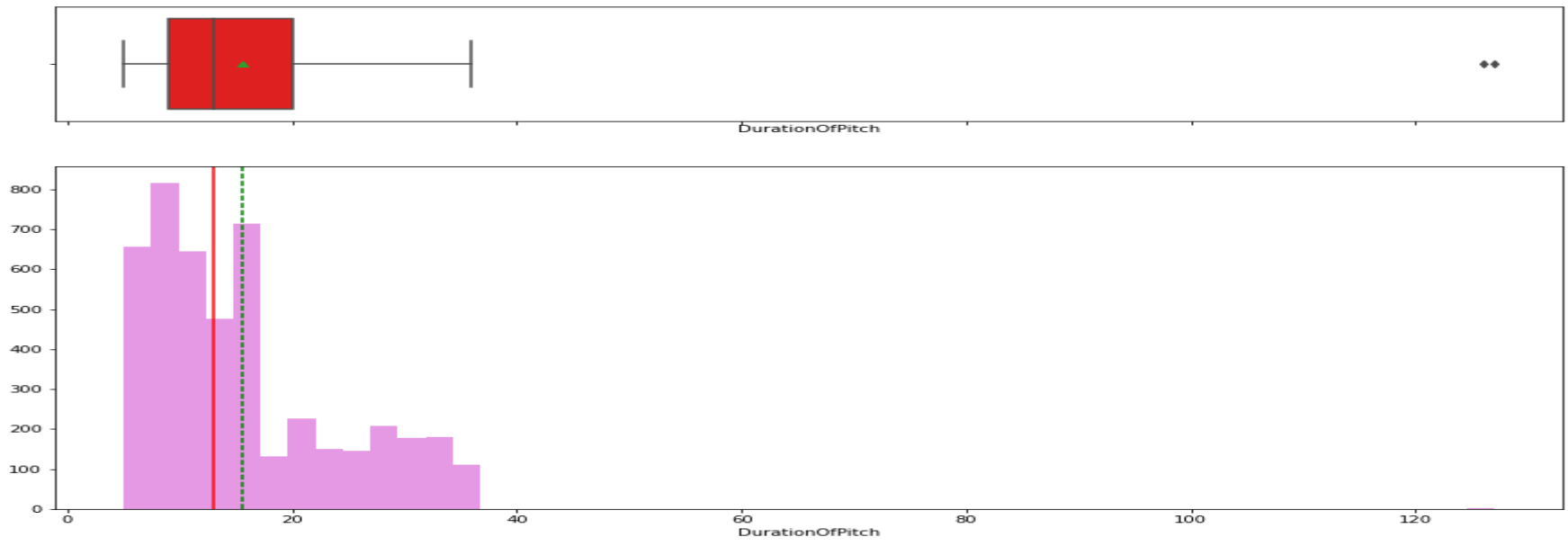
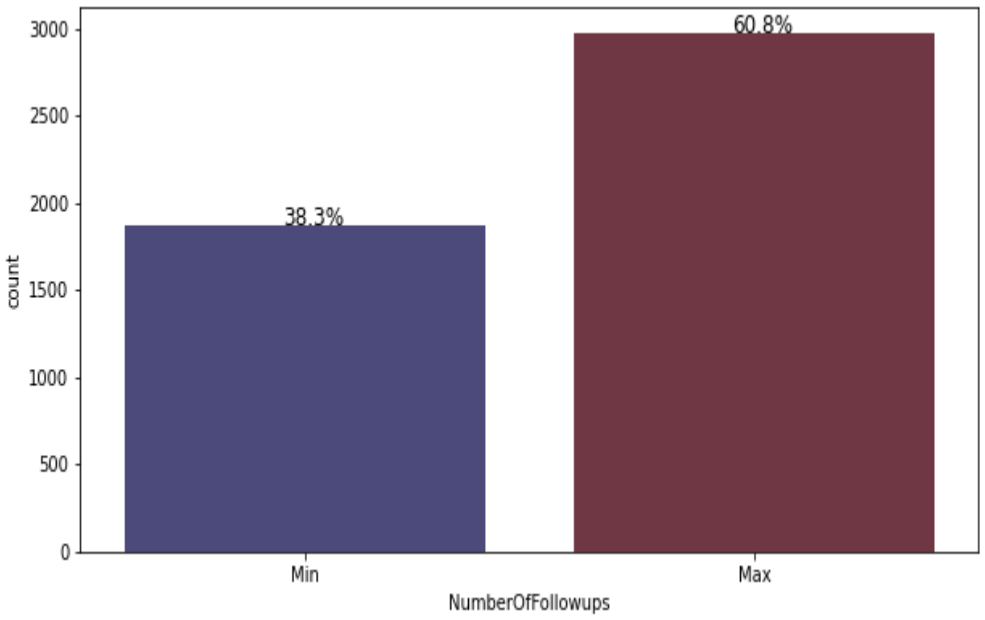
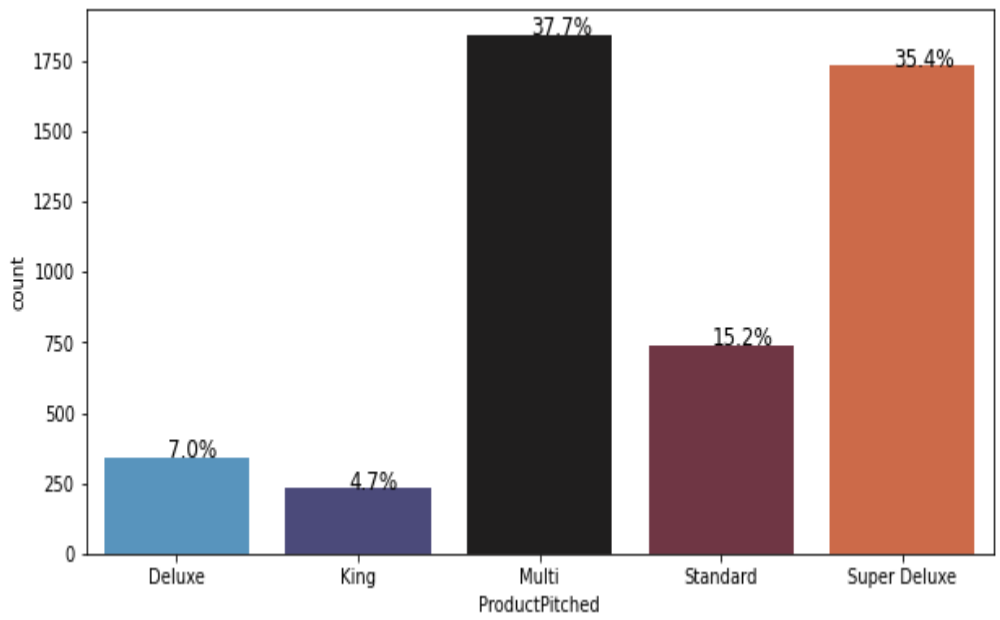


Figure-2d: Boxplot and Histogram plot on Duration Of Pitch Variable





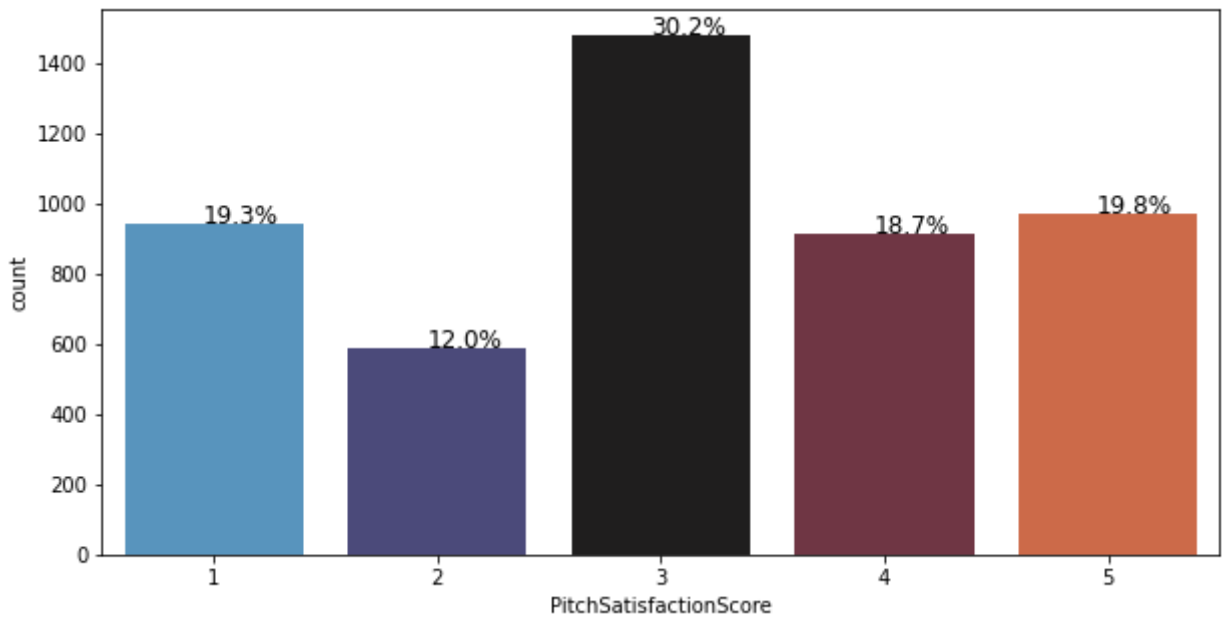


Figure-2e: Count plots of Categorical Variables

Summary of Univariate Analysis:

Customer Details:

- Age variable is normally distributed.
- About 60% of the customers are male and 40% are Females.
- About 48% of the customers are married.
- Majority of the customers that is 48.4% are Salaried. About 42.6% are small business owners and about 9% are large business owners. For Free Lancer we have only 2 customers.
- Majority of customer's designation are Executives followed by Manager's and Senior Manager's.
- Monthly Income for majority of the customer falls under the income range (15,000 to 40000). Presence of outliers on both the extreme ends. Need to deal with outliers because those are possible values.
- 71% of them do not have passport and 62% of the customers have their own car.
- Most of the customers are from city tier 1 followed by 3 and very small percentage is from tier 2.

Customer-Preference:

- Majority of the customers preferred to stay at 3-star rating's property.
- 70% of them self-enquired about the package.
- Minimum no.of trips (1 or 2) were taken by majority of the customers.

Customer-Interaction:

- Duration of Sales pitch is skewed towards left need to deal with extreme right values.
- Max number of follow-ups been made by the sales team which is 4 and above for majority of the customers.
- Multi is the most pitched product followed by Super-deluxe.
- Most of the customers gave score 3 for sales pitch.

3. Bivariate Analysis:

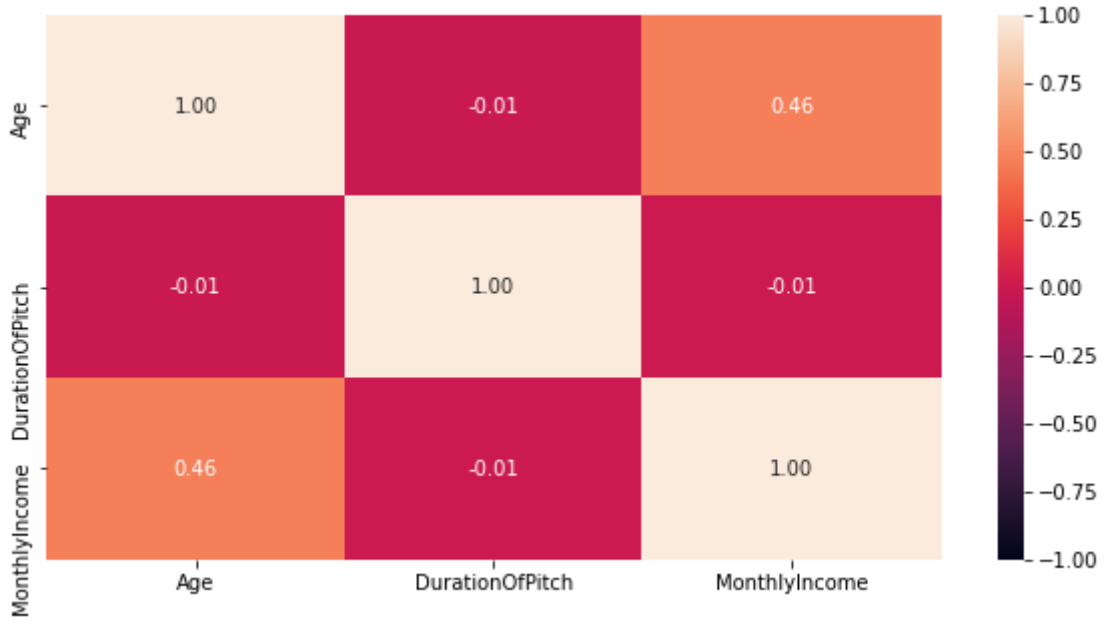
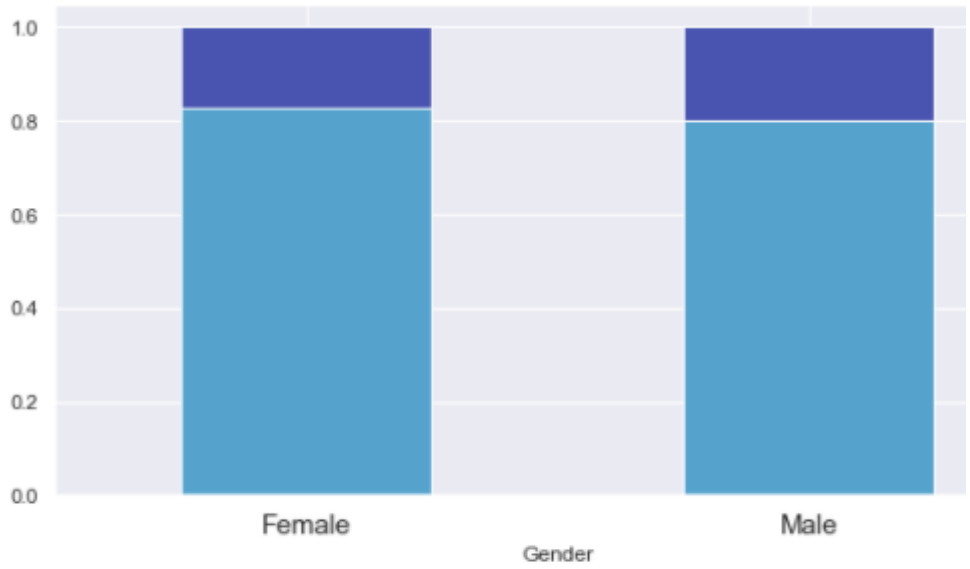


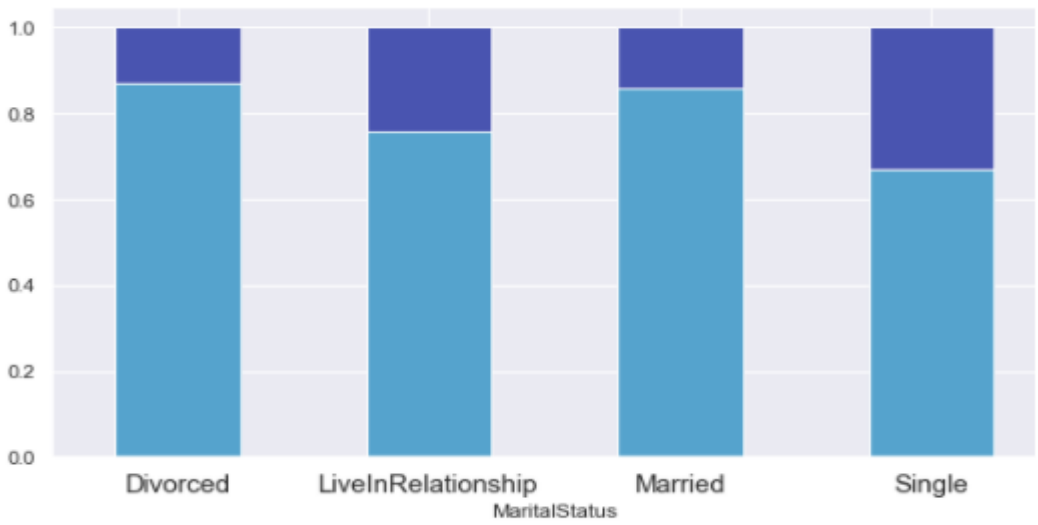
Figure-3a: Heat-Map of Continuous Variables

- Age variable is correlated with Monthly Income which is expected.

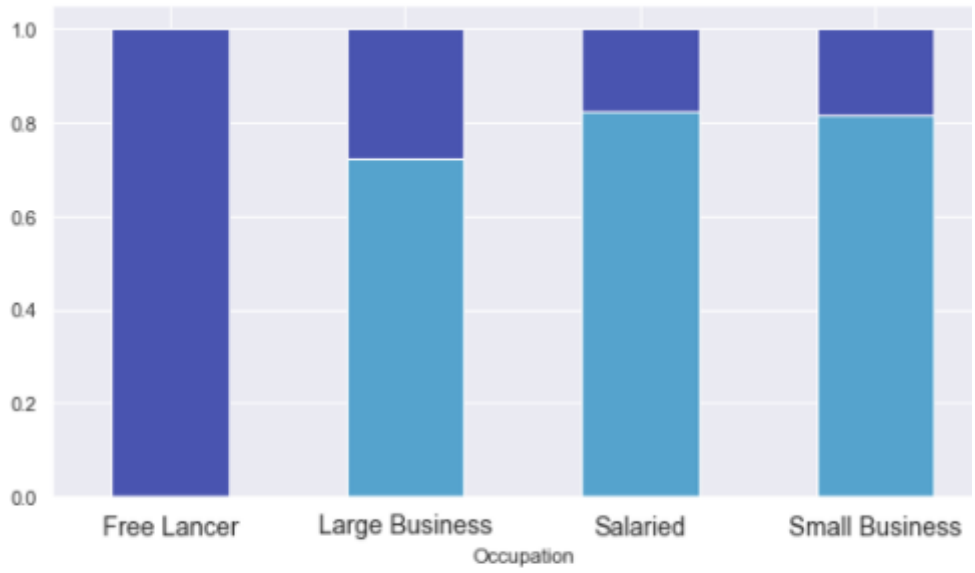
Bivariate Analysis on Customer Details V/s ProdTaken:



ProdTaken	0	1	All
Gender			
Female	1630	342	1972
Male	2338	578	2916
All	3968	920	4888

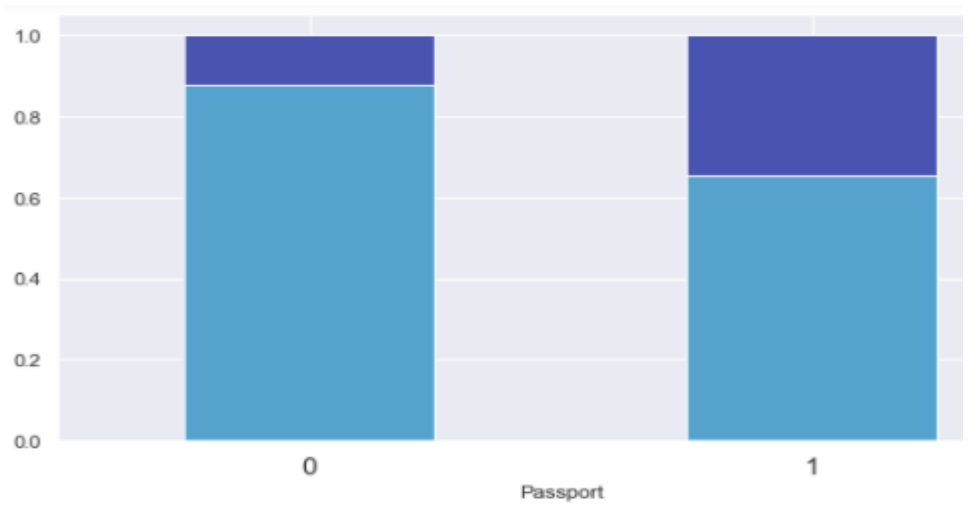


ProdTaken	0	1	All
MaritalStatus			
Divorced	826	124	950
LiveInRelationship	516	166	682
Married	2014	326	2340
Single	612	304	916
All	3968	920	4888

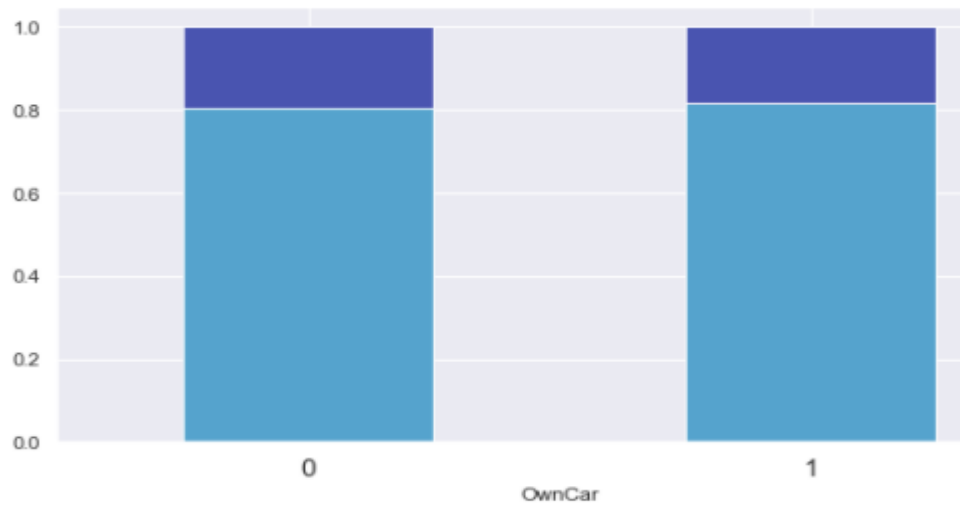


ProdTaken	0	1	All
Occupation			
Free Lancer	0	2	2
Large Business	314	120	434
Salaried	1954	414	2368
Small Business	1700	384	2084
All	3968	920	4888

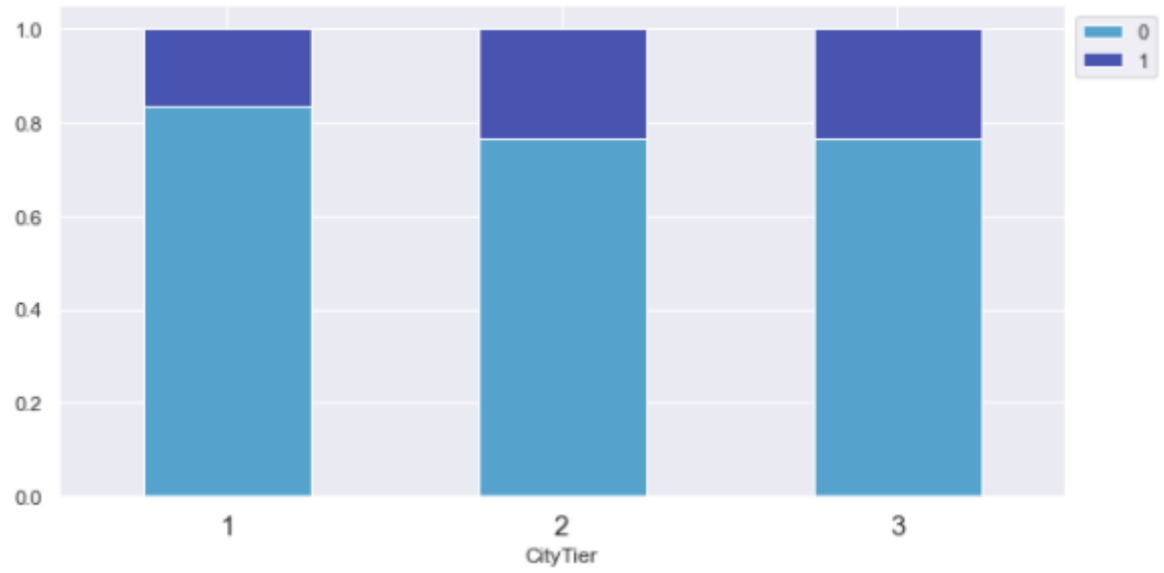
ProdTaken	0	1	All
Designation			
AVP	322	20	342
Executive	1290	552	1842
Manager	1528	204	1732
Senior Manager	618	124	742
VP	210	20	230
All	3968	920	4888



ProdTaken	0	1	All
Passport			
0	3040	426	3466
1	928	494	1422
All	3968	920	4888



ProdTaken	0	1	All
OwnCar			
0	1496	360	1856
1	2472	560	3032
All	3968	920	4888



ProdTaken	0	1	All
CityTier			
1	2670	520	3190
2	152	46	198
3	1146	354	1500
All	3968	920	4888

Figure-3b: Stack Plot of Categorical Variables [Customer-Details] V/s ProdTaken

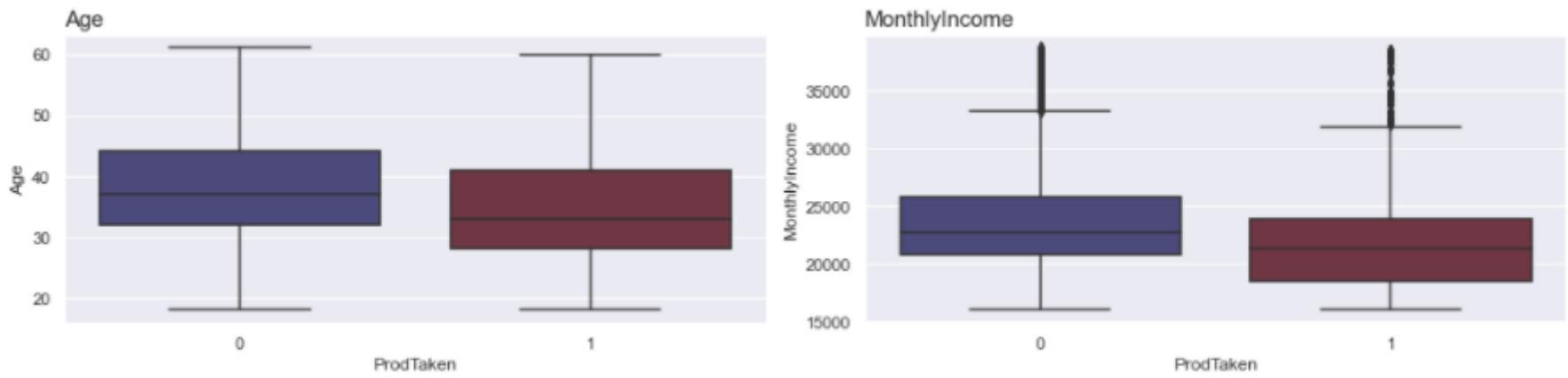
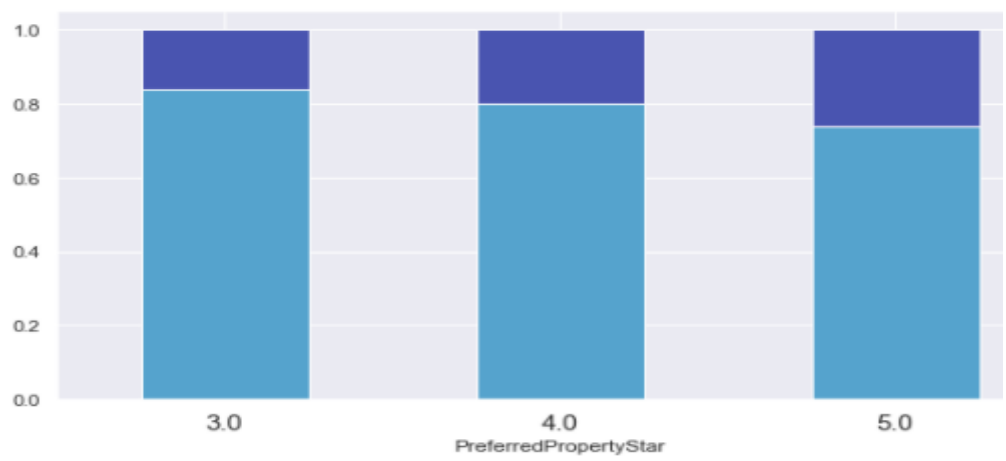


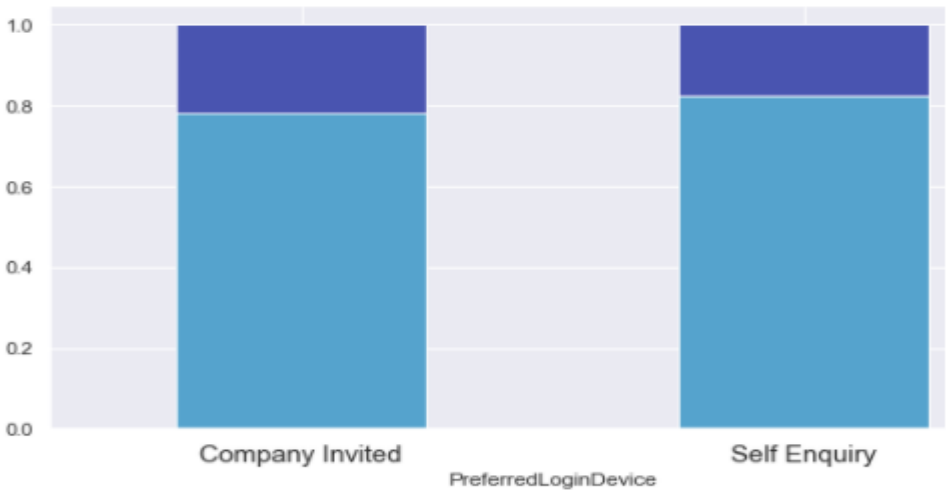
Figure-3c: Box-Plot of Numerical Variables [Customer-Details] V/s ProdTaken

---

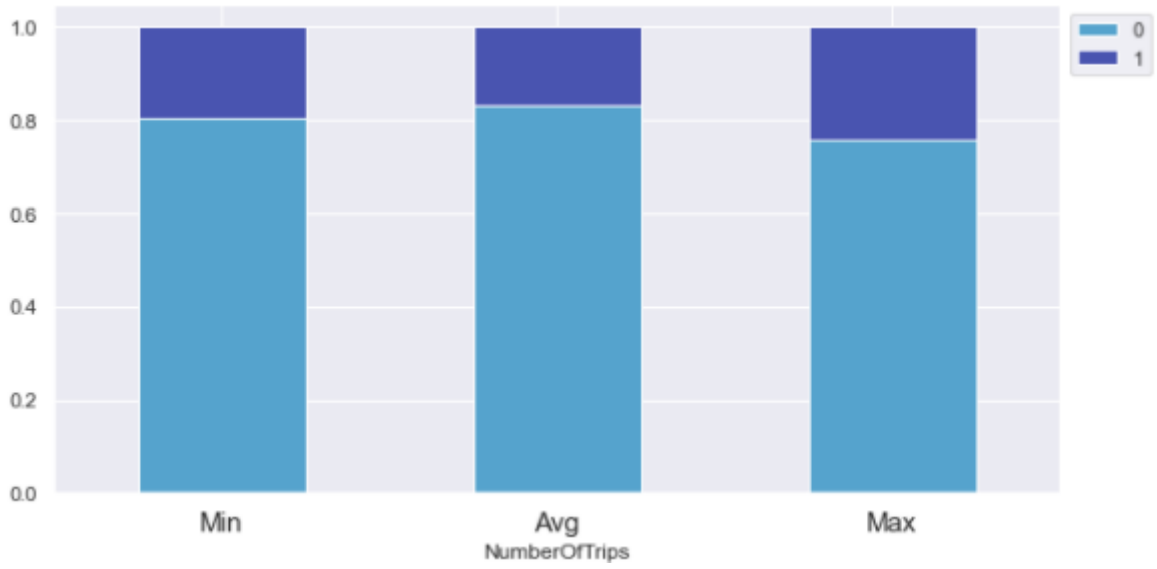
Bivariate Analysis on Customer Preference V/s ProdTaken:



ProdTaken	0	1	All
PreferredPropertyStar			
3.0	2511	482	2993
4.0	731	182	913
5.0	706	250	956
All	3948	914	4862



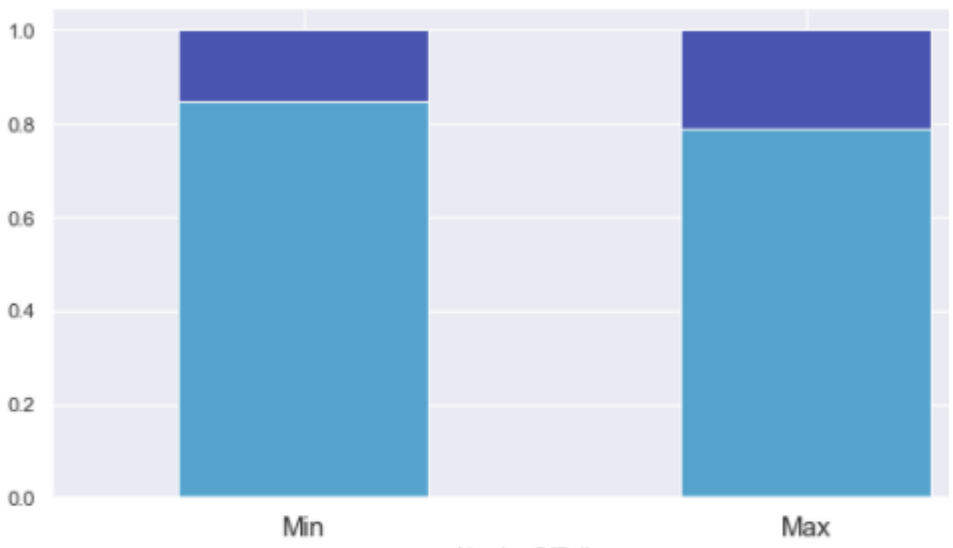
ProdTaken	0	1	All
PreferredLoginDevice			
Company Invited	1109	310	1419
Self Enquiry	2837	607	3444
All	3946	917	4863



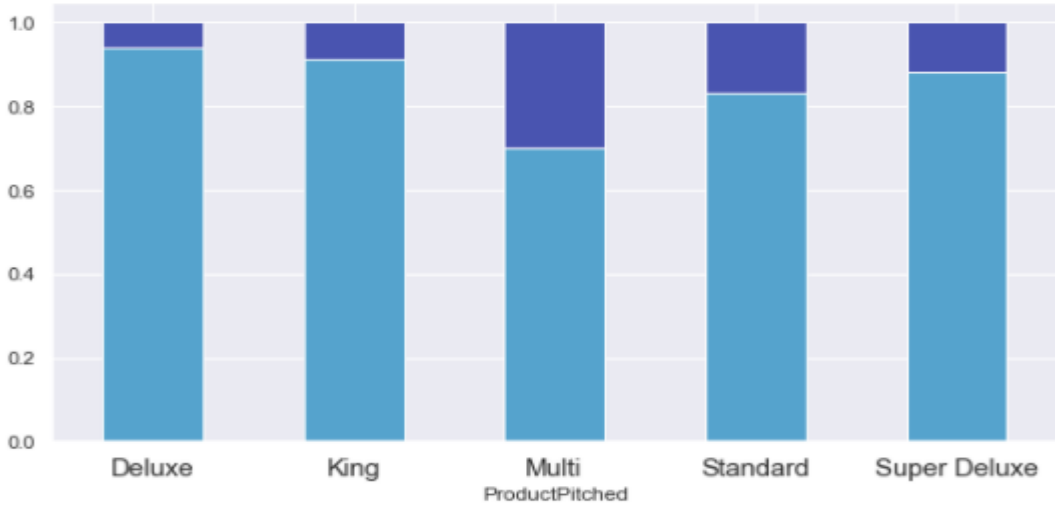
ProdTaken	0	1	All
NumberOfTrips			
Min	1673	411	2084
Avg	1675	340	2015
Max	492	157	649
All	3840	908	4748

Figure-3d: Stack Plot of Categorical Variables [Customer-Preference] V/s ProdTaken

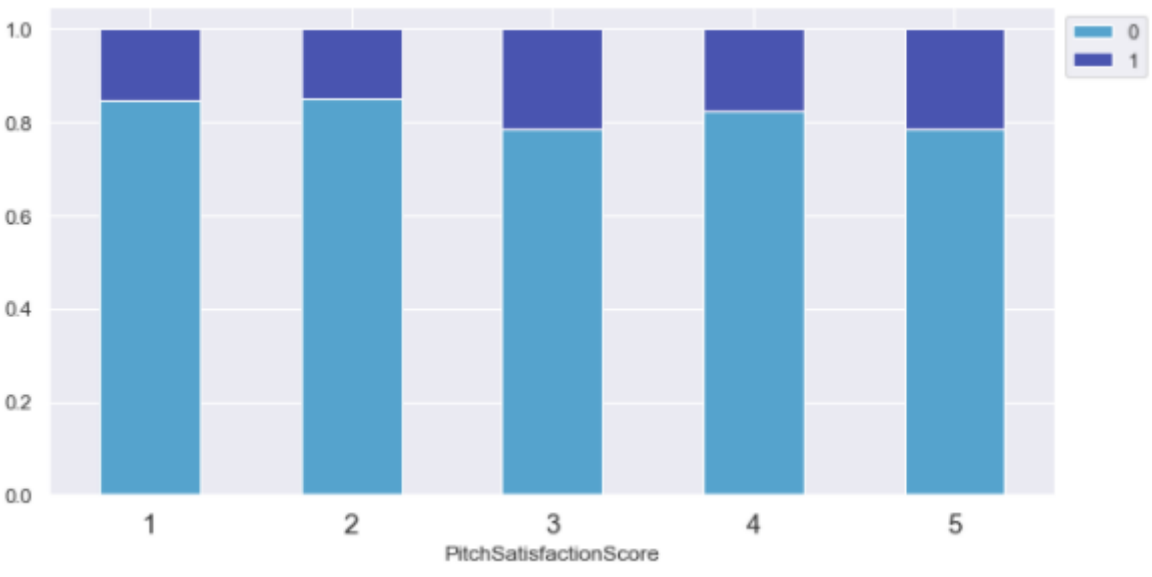
Bivariate Analysis on Customer Interaction V/s ProdTaken:



ProdTaken	0	1	All
NumberOfFollowups			
Min	1583	288	1871
Max	2348	624	2972
All	3931	912	4843



ProdTaken	0	1	All
ProductPitched			
Deluxe	322	20	342
King	210	20	230
Multi	1290	552	1842
Standard	618	124	742
Super Deluxe	1528	204	1732
All	3968	920	4888



ProdTaken	0	1	All
PitchSatisfactionScore			
1	798	144	942
2	498	88	586
3	1162	316	1478
4	750	162	912
5	760	210	970
All	3968	920	4888

Figure-3e: Stack Plot of Categorical Variables [Customer-Interaction] V/s ProdTaken

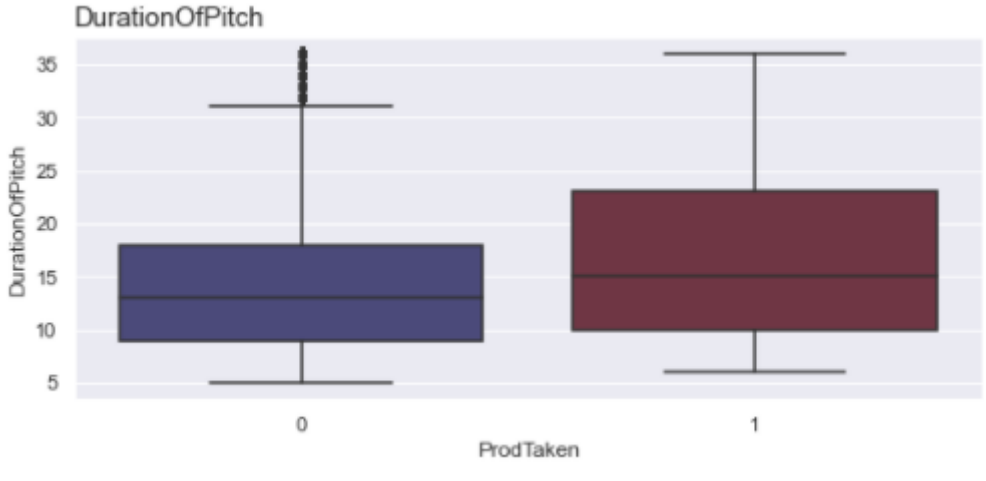


Figure-3f: Box-Plot of Numerical Variables [Customer-Interaction] V/s ProdTaken

Summary of Bivariate Analysis:

Customer Demographics v/s ProdTaken:

- Tourism package has been taken mostly by customers who are of less than 40 years of age.
- There is not much distinction among Male and Female customers who have purchased the travel package.
- Customers who are single and LiveInRelationship people has purchased the travel package more than married or divorced.
- We can see that all of the customers who are Free Lancers(though we have only 2 such customers in this dataset) have purchased the product.Also those with Large Business have purchased the product.
- Customers who are in executive positions have purchased the travel package more followed by those in senior manager position.
- There isn't a much difference in the monthly income of customers who have and have not taken the package.
- There is not much difference in purchasing travel package between customers with and without a car.
- Customers who have a passport have higher chance of purchasing the travel package than those without one.
- Customers from city tier 2 and 3 have mostly purchased the travel package.

Customer Preference v/s ProdTaken:

- Chances of purchasing product increases has the property rating increases.Customers who prefer 5-star property purchased travel package more.
- Most of the customers who took the travel package are those who were invited by the company to take tourism package .
- There is no difference in the purchasing travel package for customers who had 0,1,2,3 children visiting with them.
- We can see that those with 2,3,4 persons travelling with the customer has more chance of buying the travel package.Those who came with 1 and 5 persons did not buy any travel package.
- Customer who prefers to travel min(1 or 2) trips per year and max(>6) trips per year has higher chances of buying tourism package.

Customer Interaction v/s ProdTaken:

- Package has been purchased when there is higher duration of pitch by salesman to the customer.
- The package was purchased by those customers to whom the Multi product was pitched, followed by standard product.
- The chance of purchasing increased as the number of followups went up.With max (>4) showing good chance of the customer purchasing the package.
- Customers who have given a PitchSatisfactionScore of 3 or 5 have purchased the travel package more.

4. Multivariate Analysis:

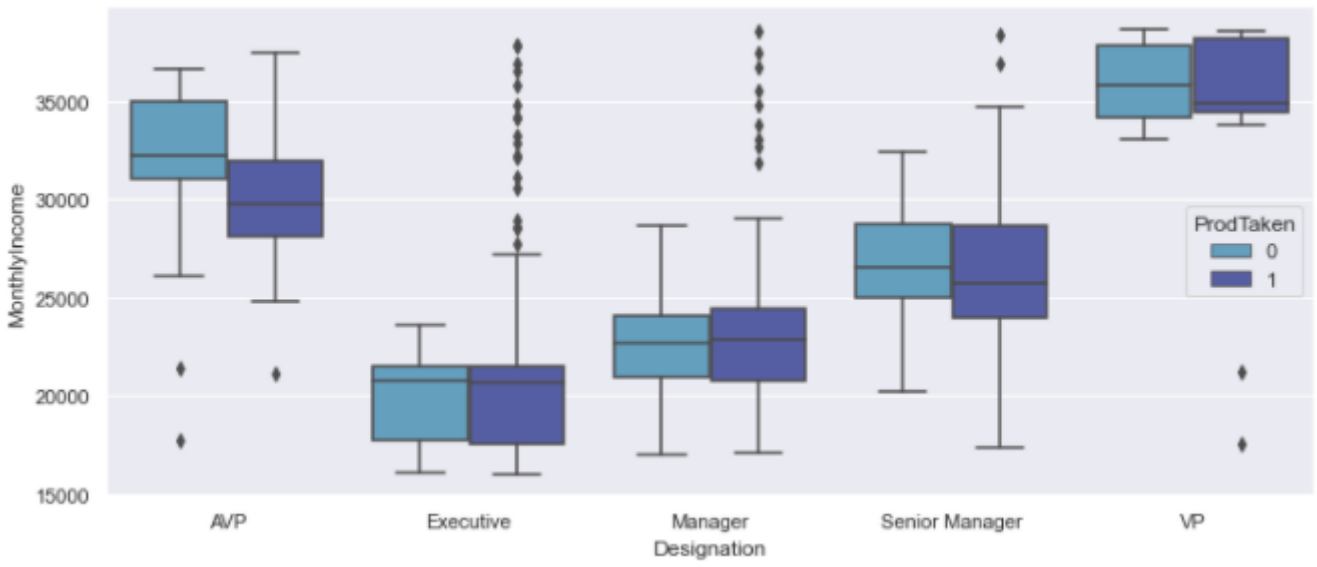


Figure-4a: Multivariate Analysis: Monthly Income V/s Designation hue=ProdTaken

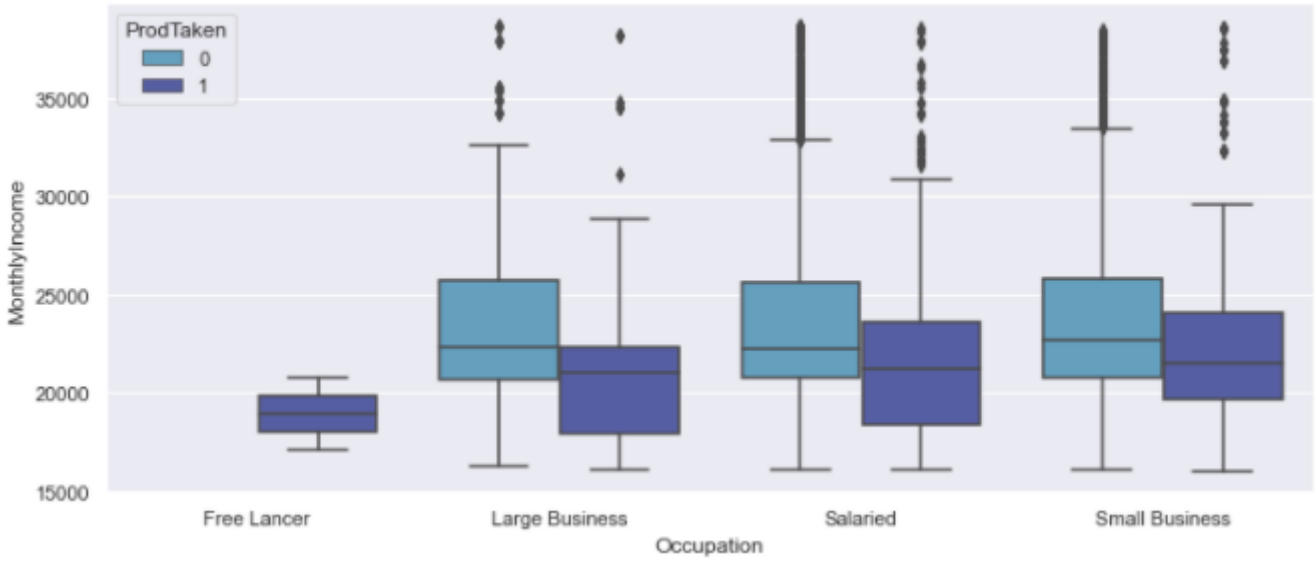


Figure-4b: Multivariate Analysis: Monthly Income V/s Occupation hue=ProdTaken

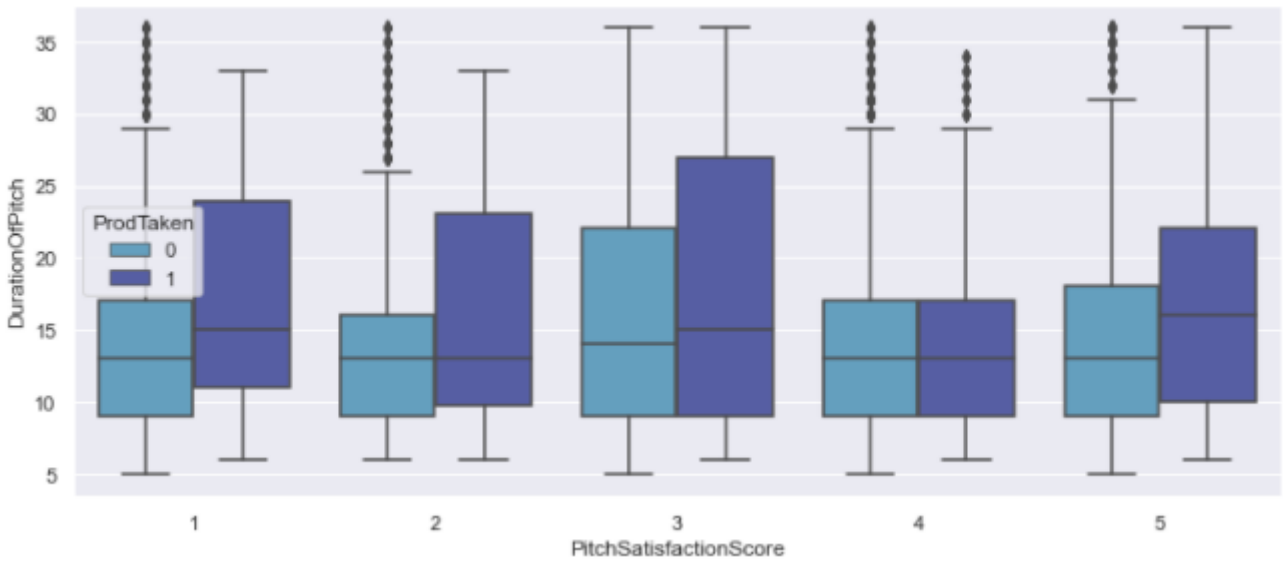


Figure-4c: Multivariate Analysis: DurationOfPitch V/s PitchSatisfactionScore hue=ProdTaken

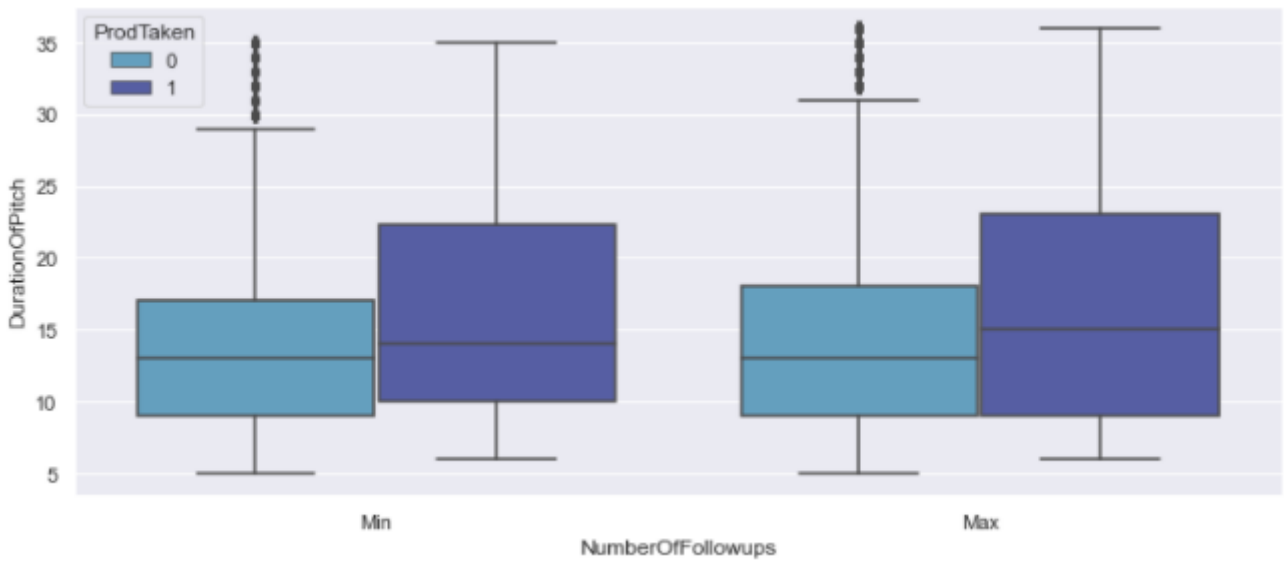


Figure-4d: Multivariate Analysis: DurationOfPitch V/s NumberofFollowups hue=ProdTaken

Summary of Multivariate Analysis:

- Monthly Income is high whose designation is AVP and VP.
- As the designation position changes from low level to high level, monthly income also increases.Hence there is a correlation between Monthly Income and Designation.
- Buying pattern for VP and AVP most of the customers did not prefer to buy.
- Majority of the customers across different levels of occupation falls under the monthly income range less than 25000 and most of them did not prefer to buy.
- Pitch satisfactory score with 1 and 5 might be data glitch majority of the customers might assume one being highest and also few might believe 5 being highest hence both seems to have higher chances of buying the package. This could be due to confusion created in ratings.
- With the increase in duration of sales pitch and with the increase in number of follow-ups the chances of buying tourism package increases.

3. Data Cleaning and Pre-processing:

- Approach used for identifying and treating missing values and outlier treatment (and why)
- Need for variable transformation (if any)
- Variables removed or added and why (if any)

1. Variable Transformation [Data-Cleaning]:

Gender Column:

- We can combine 'Fe Male' category with the 'Female' category.
- We have 155 Observations for 'Fe Male' category. We will replace all of them to 'Female' category.

```
GENDER :  3
Fe Male     155
Female     1817
Male       2916
Name: Gender, dtype: int64
```

```
Male      2916
Female    1972
Name: Gender, dtype: int64
```

Table-6a: Before and after Variable transformation

MaritalStatus Column:

- Assuming Unmarried category is not married but having a partner –Live in relationship.
- We will change the category name from 'Unmarried' to 'LiveInRelationship'.
- We have 682 observations found under unmarried category. Replace all of them to 'LiveInRelationship'.

```
MARITALSTATUS :  4
Divorced      950
Married      2340
Single        916
Unmarried     682
Name: MaritalStatus, dtype: int64
```

```
Married              2340
Divorced              950
Single                916
LiveInRelationship    682
Name: MaritalStatus, dtype: int64
```

Table-6b: Before and after Variable transformation

2. Outlier Treatment [Data-Cleaning]:

a. DurationOfPitch:

	ProdTaken	DurationOfPitch
1434	0	126.0
3878	0	127.0

- We can see couple of values 127 and 126 but those are possible values because the duration can get extended due to several reasons.
- For this particular case study, our class of interest is customers who has purchased product from the company. Even after sales pitch duration of 127 and 126, customers did not purchase the product.
- We will replace those values to DurationOfPitch median value.

b. Monthly Income:

	ProdTaken	MonthlyIncome
2482	0	98678.0
38	0	95000.0

	ProdTaken	MonthlyIncome
142	0	1000.0
2586	0	4678.0

- Monthly Income of extreme high values: 98678.0 and 95000.0 and extreme low values: 1000.0 and 4678.0 is an outliers and this can be a possible value.
- These customers have not taken product, since our class of interest is product taken we will convert this values to median of Monthly Income.

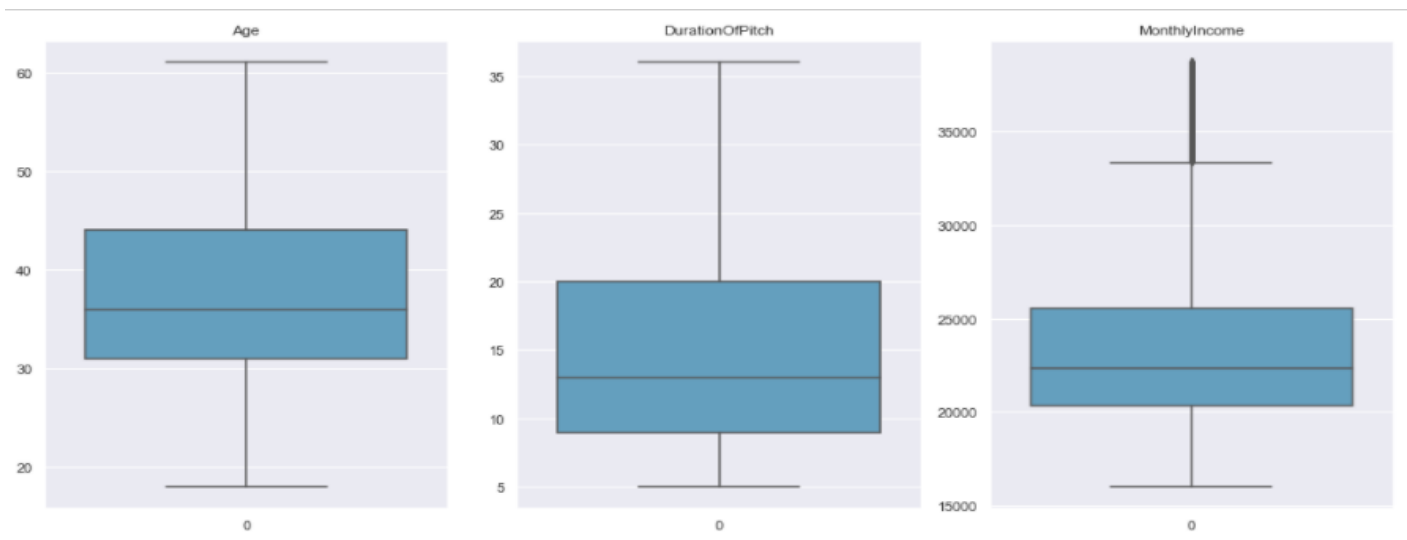
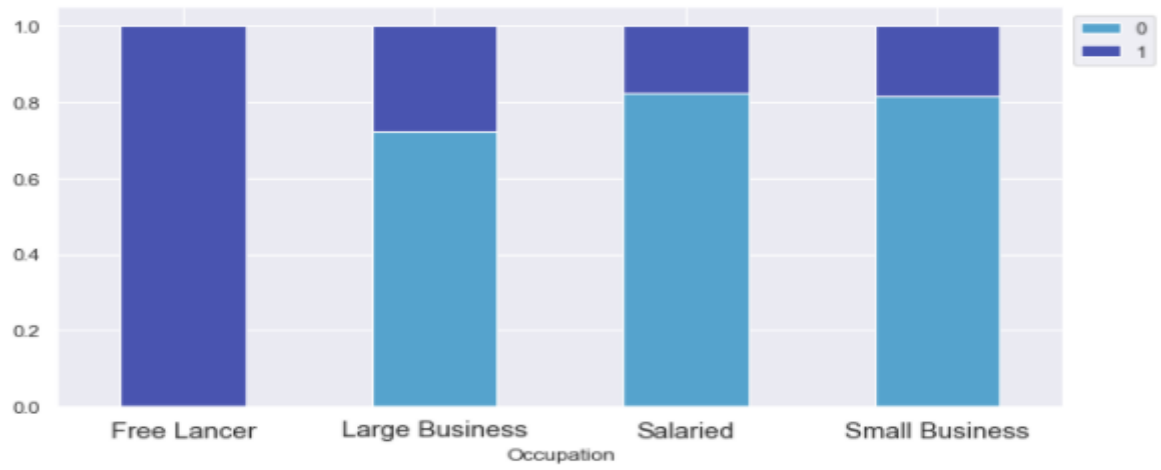


Figure-5a: Boxplot of Numerical Variables after Outlier Treatment

- Age and DurationOfPitch do not have any Outliers.
- Monthly Income has few outliers even after Outlier treatment and we found 341 observation were above the upper whisker and these values are possible values. As of now we keep it has it is and do not treat them.

Outlier Detection for Categorical columns:

c. Occupation:



ProdTaken	0	1	All
Occupation			
Free Lancer	0	2	2
Large Business	314	120	434
Salaried	1954	414	2368
Small Business	1700	384	2084
All	3968	920	4888

Figure-5b: Outlier Detection of Categorical Variable [Occupation]

- Free Lancer has only 2 values and both the customers have purchased the product, hence we will not treat this has an outlier.

3. Missing Value Treatment [Data-Cleaning]:

ProdTaken	0
Age	226
PreferredLoginDevice	25
CityTier	0
DurationOfPitch	251
Occupation	0
Gender	0
NumberOfPersonVisited	0
NumberOfFollowups	45
ProductPitched	0
PreferredPropertyStar	26
MaritalStatus	0
NumberOfTrips	140
Passport	0
PitchSatisfactionScore	0
OwnCar	0
NumberOfChildrenVisited	66
Designation	0
MonthlyIncome	233
dtype: int64	

ProdTaken	0
Age	0
PreferredLoginDevice	0
CityTier	0
DurationOfPitch	0
Occupation	0
Gender	0
NumberOfPersonVisited	0
NumberOfFollowups	0
ProductPitched	0
PreferredPropertyStar	0
MaritalStatus	0
NumberOfTrips	0
Passport	0
PitchSatisfactionScore	0
OwnCar	0
NumberOfChildrenVisited	0
Designation	0
MonthlyIncome	0
dtype: int64	

Table-7: Before and after Missing Value Treatment

- **Missing Values Identified:**
  - Customer Details: Age and Monthly Income.
  - Customer Preference: PreferredLoginDevice, PreferredPropertyStar, NumberofChildrenVisited and Number of trips
  - Customer Interaction: Duration of pitch and NumberofFollowups.
- **Missing Values Treatment:**

i. Age Column:

For Age Column, missing value imputation done one by one, by taking median value of age for male and female by grouping Designation and Gender.

	0	1	2	3	4	5	6	7	8	9
Designation	AVP	AVP	Executive	Executive	Manager	Manager	Senior Manager	Senior Manager	VP	VP
Gender	Female	Male	Female	Male	Female	Male	Female	Male	Female	Male
Age	48	49	32	32	37	36	38	39	50	47

ii. Monthly Income Column:

For Monthly Income, missing value imputation done one by one, by grouping occupation and then by taking median of monthly Income.

	0	1	2	3
Occupation	Free Lancer	Large Business	Salaried	Small Business
MonthlyIncome	18929	21881	22130	22689

iii. Duration of Pitch Column:

For Duration of pitch, missing value imputation done by using median value.

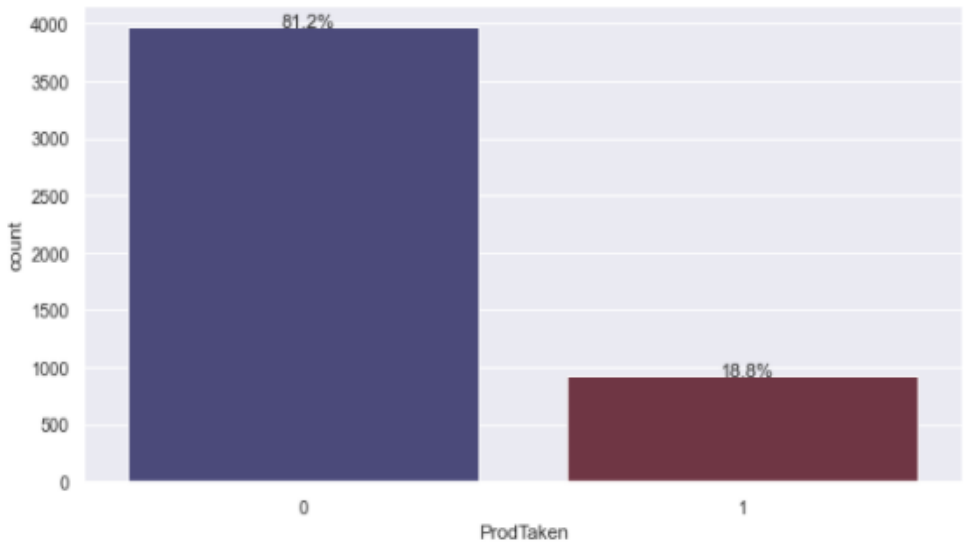
iv. Number of Children visited Column:

For NumberofChildrenVisited, assuming Children visited is missing because no children accompanied these customers so we will fill the missing values with 0.

v. Remaining Column:

For PreferredLoginDevice, PreferredPropertyStar, Number of trips and NumberofFollowups, missing value imputation done by using mode (highest occurring value) of the feature.

4. Data Imbalance:



- Distribution of Target variable is 81:19.
- Most of the customers did not buy tourism package.
- Since we have a significant imbalance in the distribution of the target classes, before building model we will use stratified sampling to ensure that relative class frequencies are approximately preserved in train and test sets.
- For that we will use the `stratify` parameter in the train_test_split function.

5. Scaling:

Scaling of variables is optional for linear models such as logistic regression, LDA and tree based models such as Decision tree, Random Forest. However, scaling is a necessity when using distance based models such KNN and SVM, because we do not want variables with greater numerical values to be given more importance by the model.

6. Encoding:

For the following object variables 'PreferredLoginDevice', 'Occupation', 'Gender', 'ProductPitched', 'MaritalStatus', 'Designation' data encoding needs to be done before building models by using pd.get_dummies() and dropping the first column to make sure we don’t have a pair of collinear variables in our dataset.

4. Model building:

- Clear on why was a particular model(s) chosen.
- Effort to improve model performance.

Split the Dataset:

- Since we have less than 10,000 observations we will choose to split the data into train and test sets in 70:30 ratio.
- We will use train_test_split function with arguments test_size = 0.30 to get 70:30 split, random state = 1 for reproduce able results and stratify = y (target variable), so that target variable has equal proportion of classes in train and test set.
- We will drop 'NumberOfPersonVisited' and 'NumberofChildrenVisited', since we are not building model based on the number of person or children visited to purchase a tourism package.
- The train set now has 3421 rows, 43 features whereas test has 1467 rows, 43 features.

```
Number of rows and columns of the training set for the independent variables: (3421, 43)
Number of rows and columns of the training set for the dependent variable: (3421,)
Number of rows and columns of the test set for the independent variables: (1467, 43)
Number of rows and columns of the test set for the dependent variable: (1467,)
```

Model Building:

Model building is an iterative process. Various models needs to be built in order to meet the business requirement to get the best optimum model out of all the models. Hence the model performance on both train and test dataset can be improved by using various permutations and combinaton to refine the model. Feature engineering, feature extraction, hyper-parameter tuning (by changing the combinations of various parameters) methods can be used to get better output.

Following classification models were built at initial stage without any tuning parameters:

1. Logistic Regression
2. Linear Discriminant Analysis
3. Decision Tree
4. Random Forest

Later Model tuning was performed for the above models using gridsearch parameter and with scoring parameter has better recall score and cross validation has 10.

Finally following Ensemble models were built to check model performance:

1. Adaptive Boost [AdaBoost]
2. Gradient Boost [GBBoost]
3. Extreme Gradient Boosting [XGBoost]

5. Model validation:

- How was the model validated? Just accuracy, or anything else too?

Model Evaluation:

- Customer purchased package and model predicted customer will purchase travel package: True Positive (observed=1,predicted=1).
- Customer did not purchase package and model predicted customer will purchase travel package: False Postive (observed=0,predicted=1).
- Customer didnot purchase package and model predicted customer will not purchase travel package: True Negative (observed=0,predicted=0).
- Customer purchased package and model predicted customer will not purchase travel package: False Negative (observed=1,predicted=0).

Model can make wrong predictions as:

- Predicting a customer will purchase travel package but the customer ultimately does not purchase the package. Type-1 error or false positive. (Actual=0, predicted=1).

- Predicting a customer will not purchase travel package but the customer purchases the travel package. Type-2 error or false negative. (Actual=1, predicted=0).

Which case is more important?

- Type-2 error needs to be reduced. In order to do so Recall needs to be increased. If Recall increases, Sales will increase and it will bring good revenue to the company.

- Type 1 error anyhow we know those customer will not buy the plan but if due to our model miss if we target those customer it is good if there's even 10-20% conversion rate.(so not a issue).

Model Outputs

1. Logistic Regression [Without Hyper-parameter Tuning]:

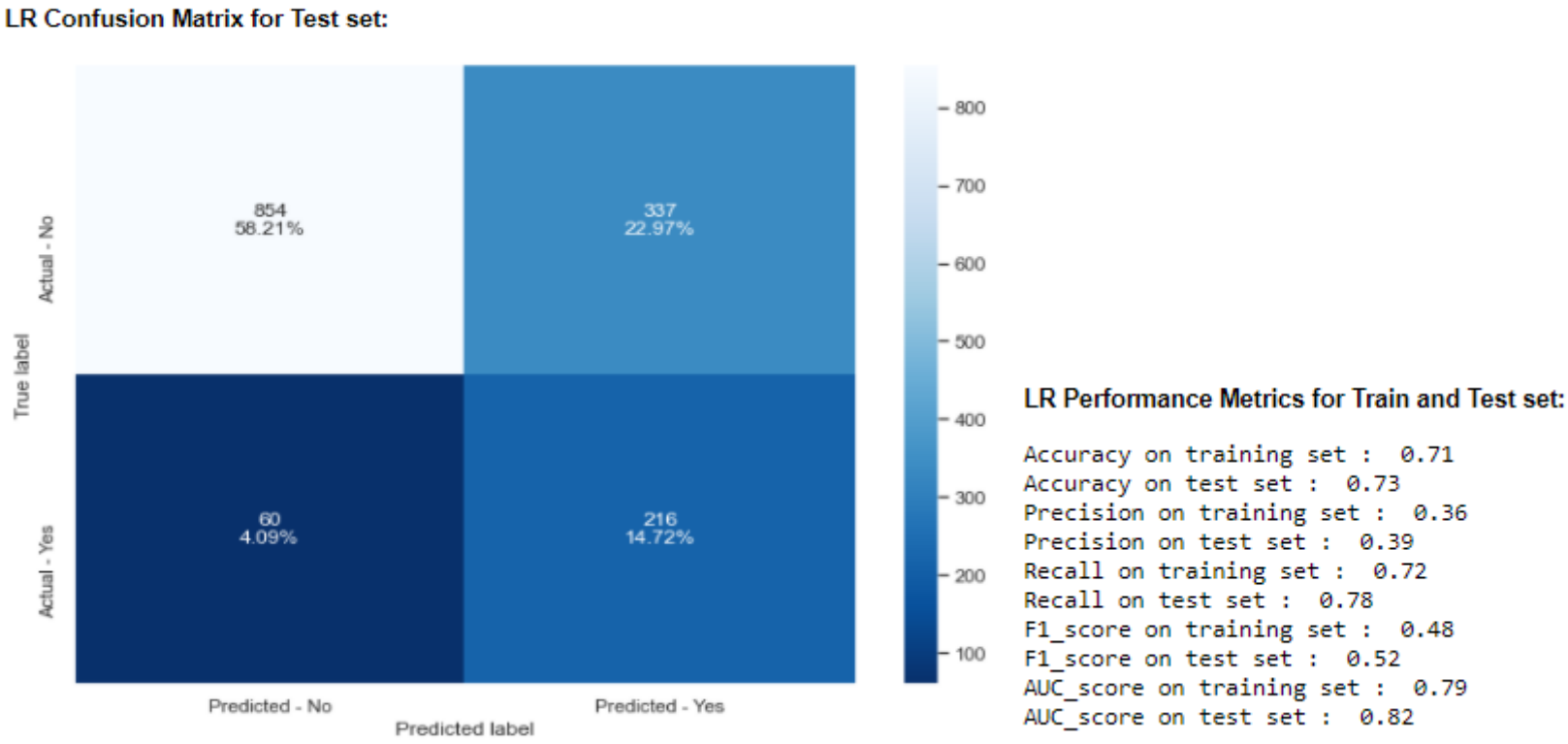


Figure-6a: Model Output of Logistic Regression with class weights [Before Hyper-parameter tuning]

2. Linear Discriminant Analysis [Without Hyper-parameter Tuning]:

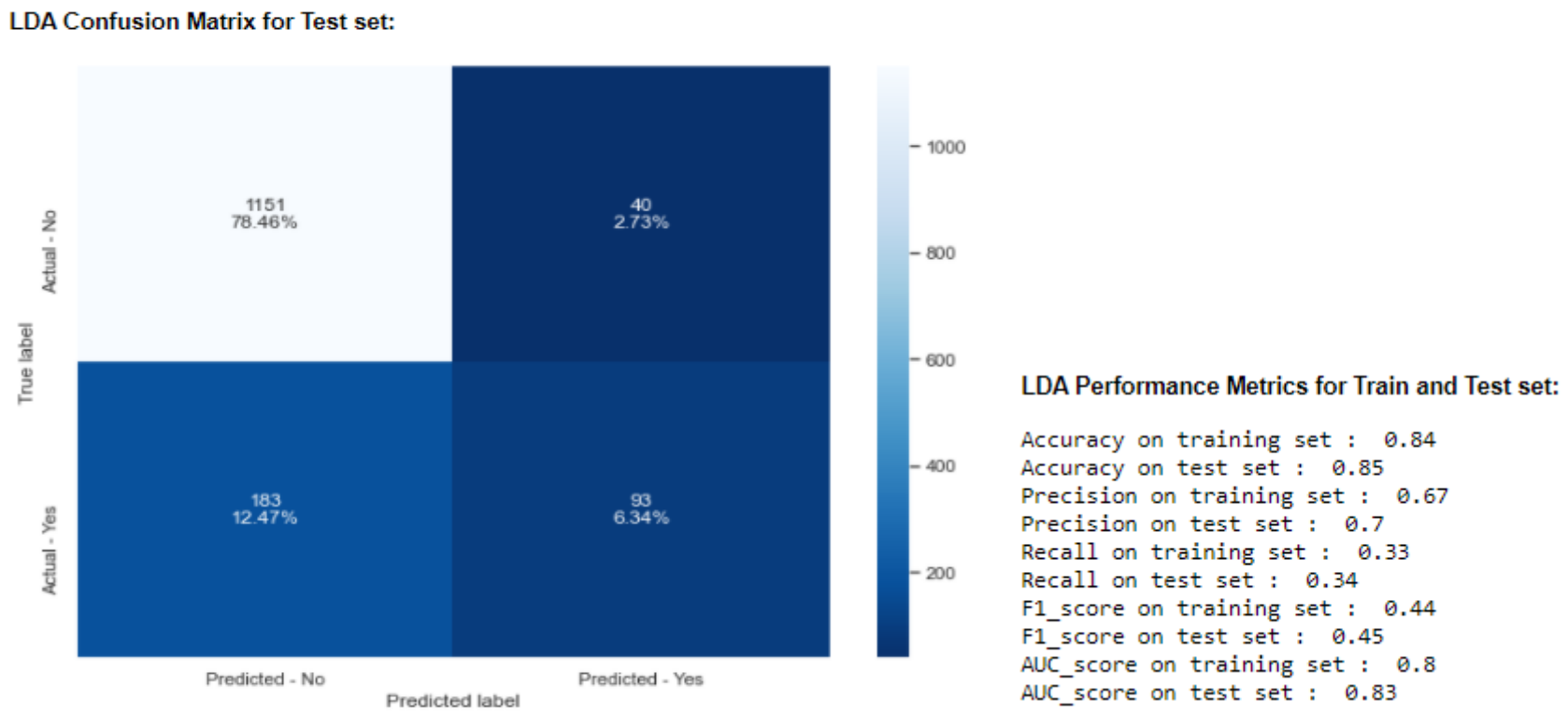


Figure-6b: Model Output of LDA [Before Hyper-parameter tuning]

3. Decision Tree [Without Hyper-parameter Tuning]:

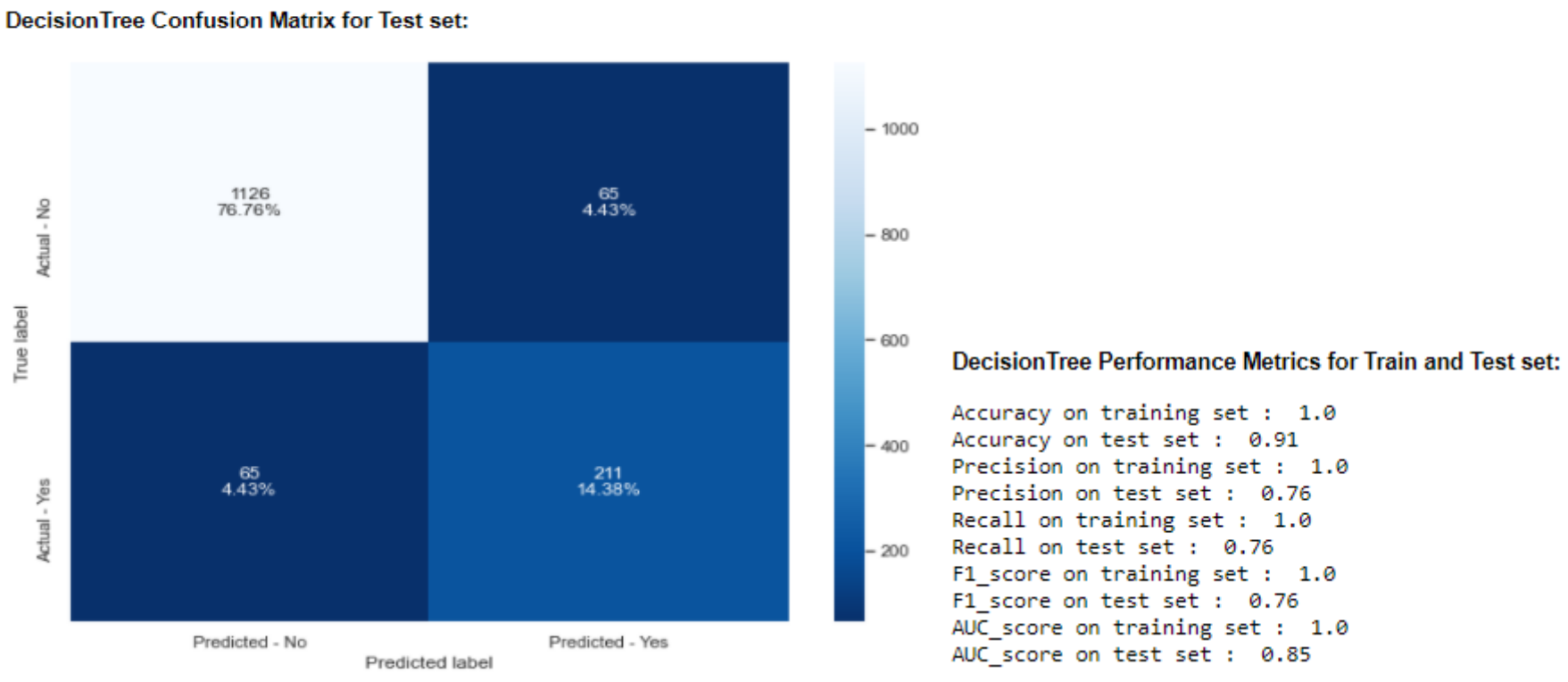


Figure-6c: Model Output of Decision Tree [Before Hyper-parameter tuning]

4. Random Forest [Without Hyper-parameter Tuning]:

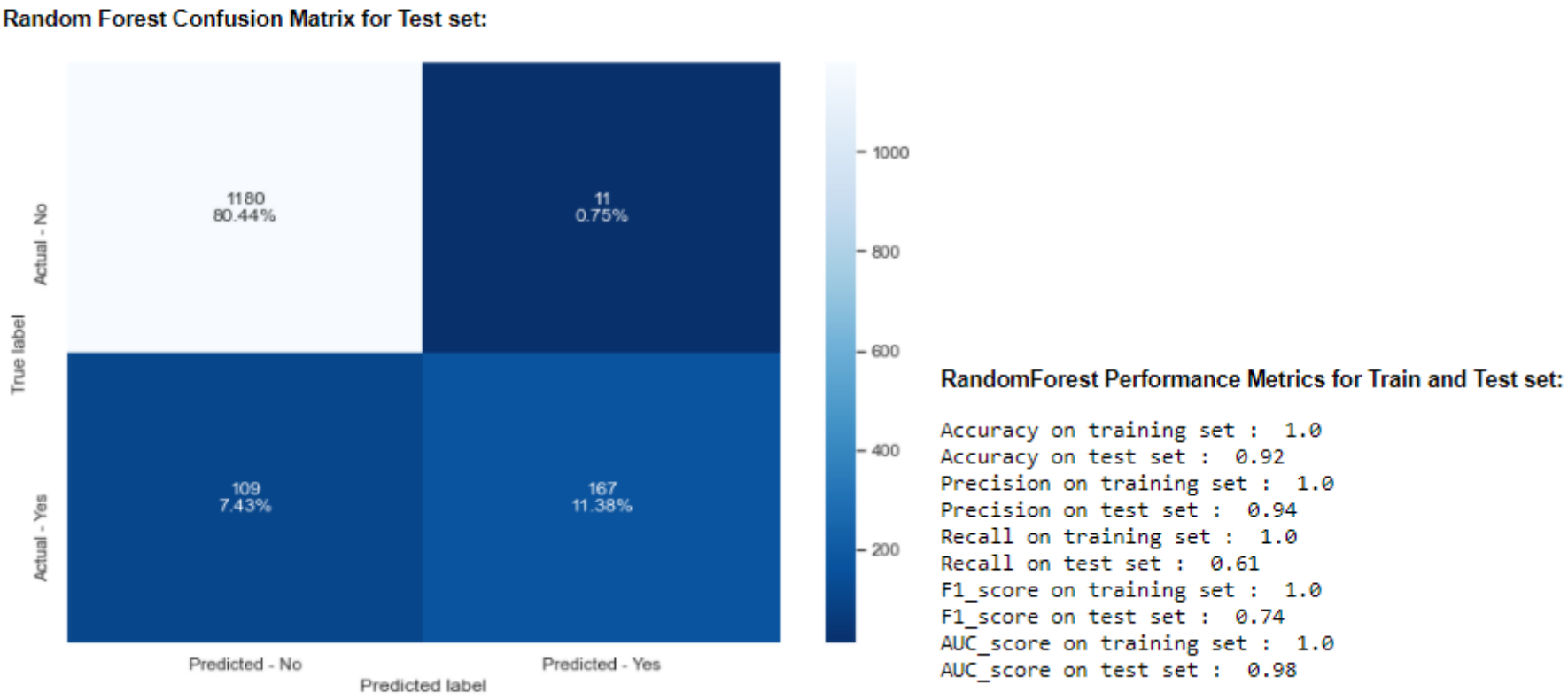


Figure-6d: Model Output of Random Forest [Before Hyper-parameter tuning]

5. Logistic Regression [With Hyper-parameter Tuning]:

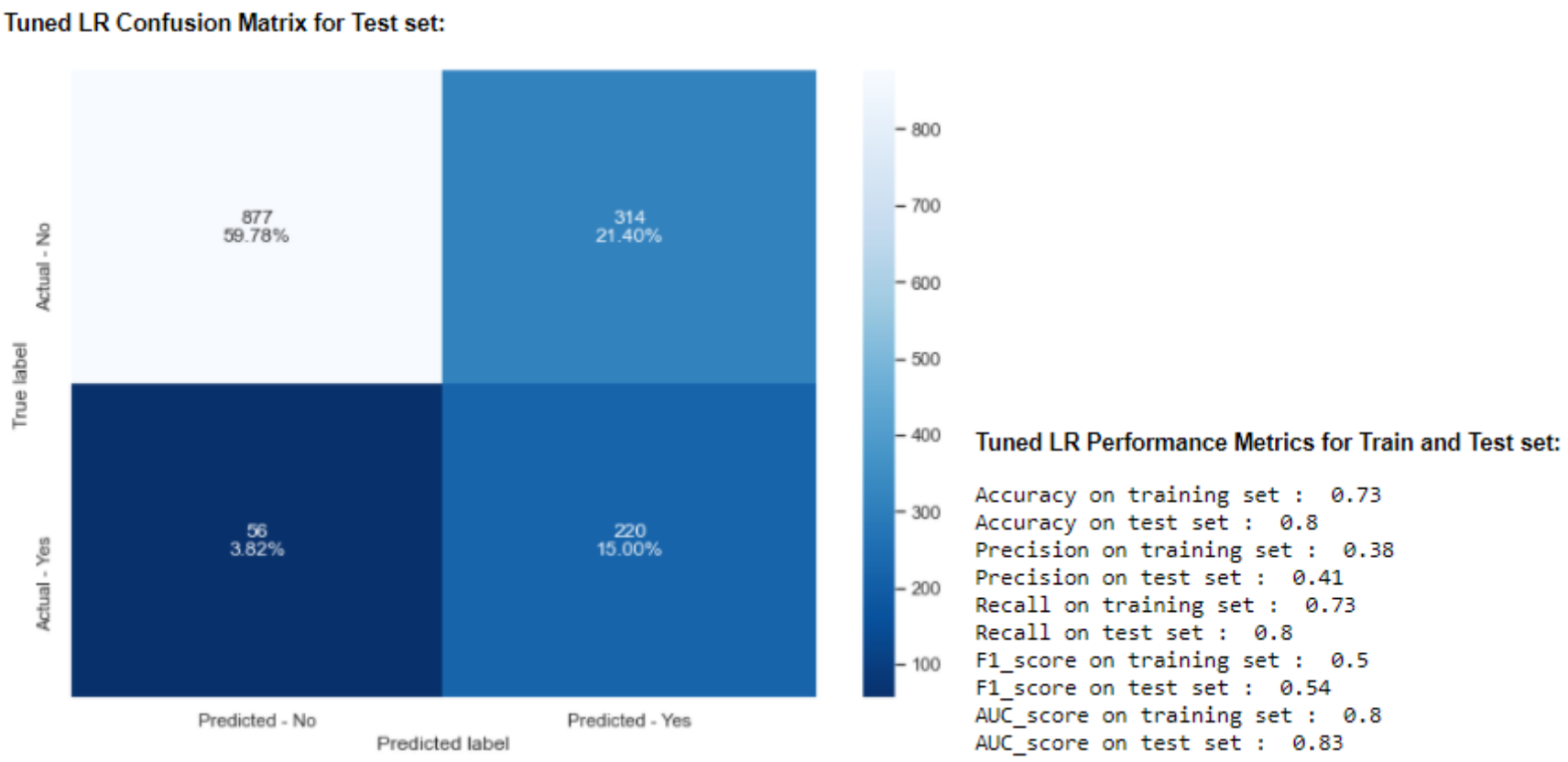


Figure-6e: Model Output of LR [After Hyper-parameter tuning]

6. Decision Tree [With Hyper-parameter Tuning]:

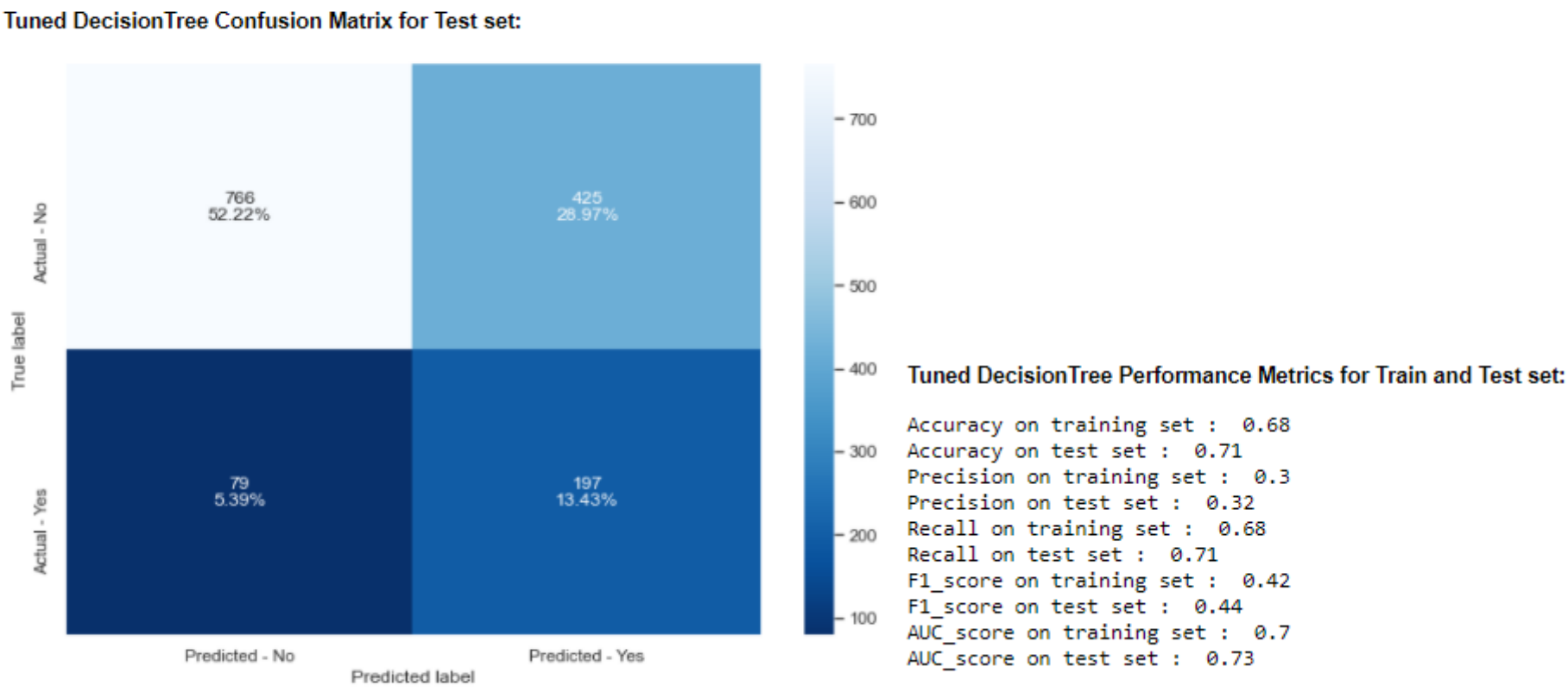
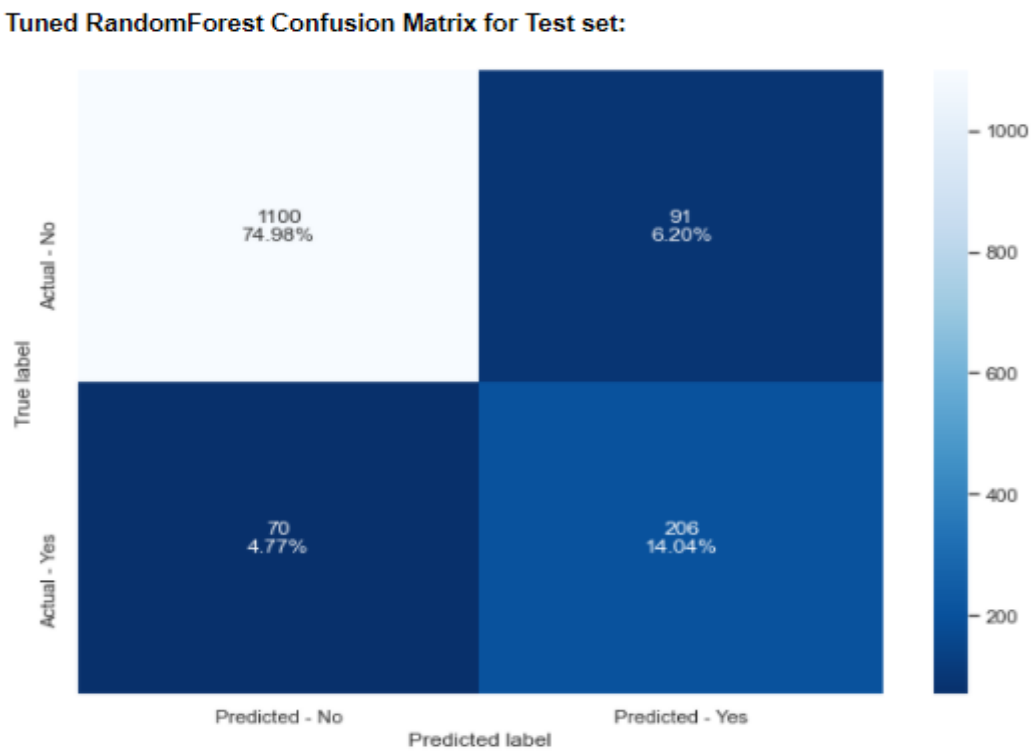


Figure-6f: Model Output of Decision Tree [After Hyper-parameter tuning]

7. Random Forest [With Hyper-parameter Tuning]:

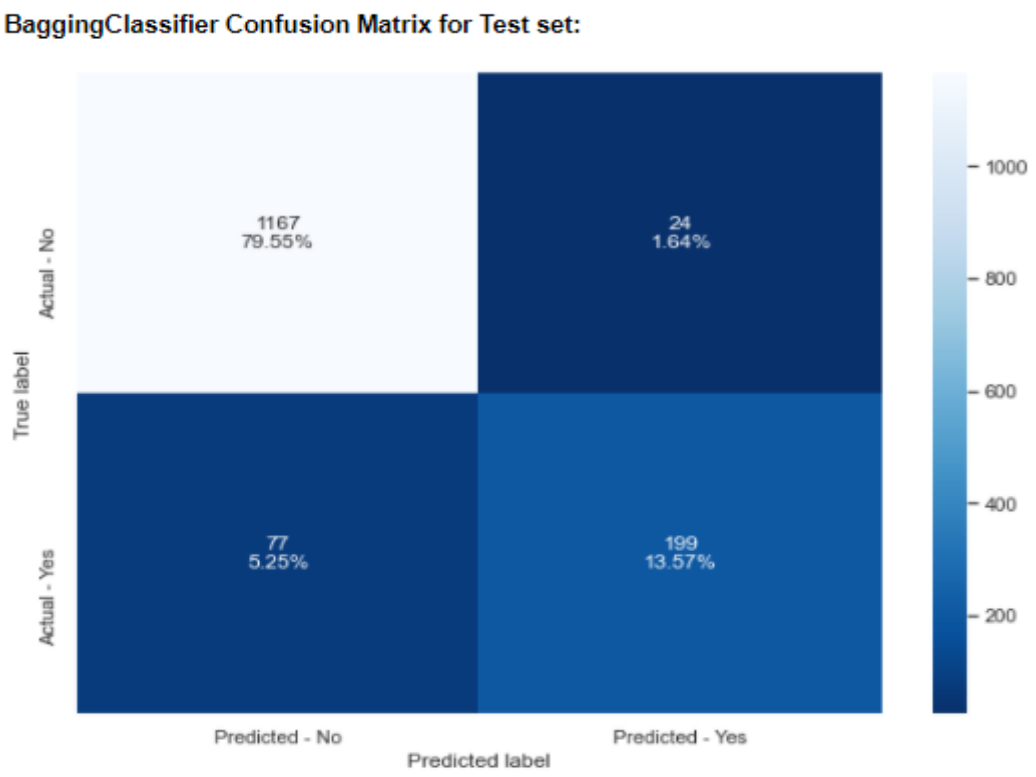


Tuned Random Forest Performance Metrics for Train and Test set:

```
Accuracy on training set :  0.96
Accuracy on test set :  0.75
Precision on training set :  0.79
Precision on test set :  0.69
Recall on training set :  0.96
Recall on test set :  0.75
F1_score on training set :  0.87
F1_score on test set :  0.72
AUC_score on training set :  0.99
AUC_score on test set :  0.93
```

Figure-6g: Model Output of Random Forest [After Hyper-parameter tuning]

8. Ensemble Model –Bagging classifier [Without Hyper-parameter Tuning]:

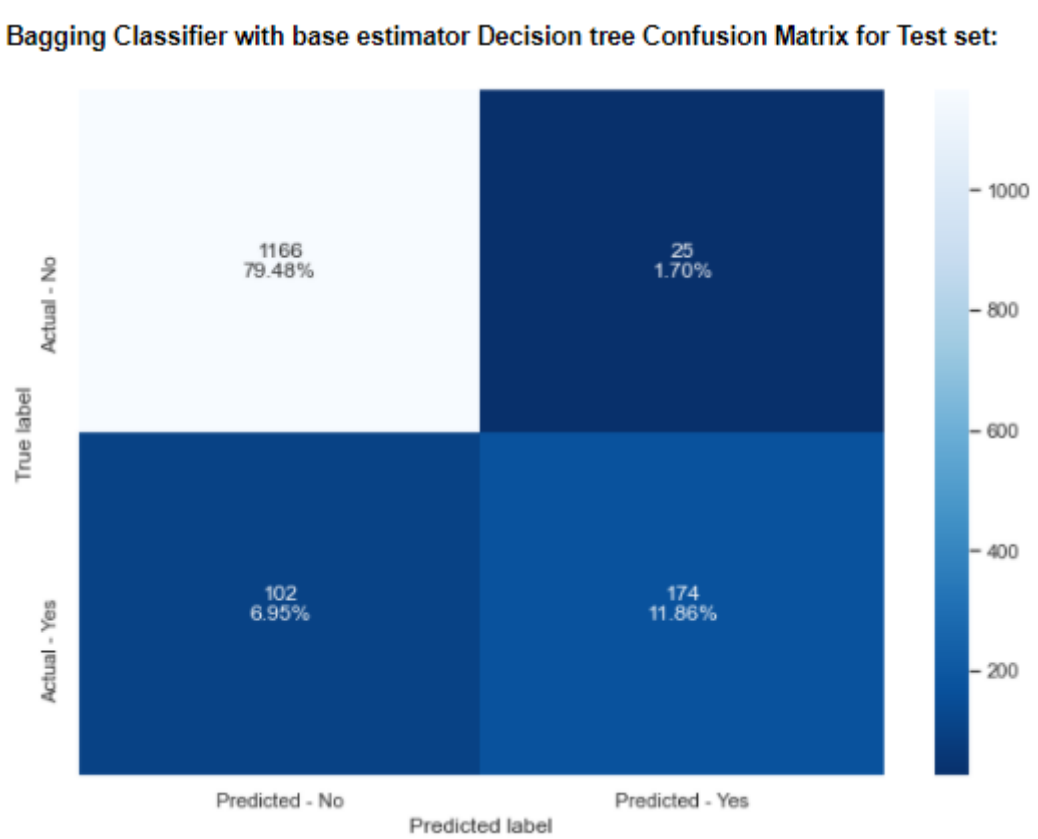


Bagging Classifier Performance Metrics for Train and Test set:

```
Accuracy on training set :  1.0
Accuracy on test set :  0.93
Precision on training set :  1.0
Precision on test set :  0.89
Recall on training set :  0.98
Recall on test set :  0.72
F1_score on training set :  0.99
F1_score on test set :  0.8
AUC_score on training set :  1.0
AUC_score on test set :  0.95
```

Figure-6h: Model Output of Bagging Classifier [Before Hyper-parameter tuning]

9. Bagging classifier with base estimator: Weighted Decision tree



Bagging Classifier with base estimator Decision tree Performance Metrics for Train and Test set:

```
Accuracy on training set :  0.99
Accuracy on test set :  0.91
Precision on training set :  1.0
Precision on test set :  0.87
Recall on training set :  0.98
Recall on test set :  0.63
F1_score on training set :  0.99
F1_score on test set :  0.73
AUC_score on training set :  1.0
AUC_score on test set :  0.94
```

Figure-6i: Model Output of Bagging with base estimator: weighted Decision tree [Before Hyper-parameter tuning]

10. Bagging classifier with base estimator: Weighted Random Forest

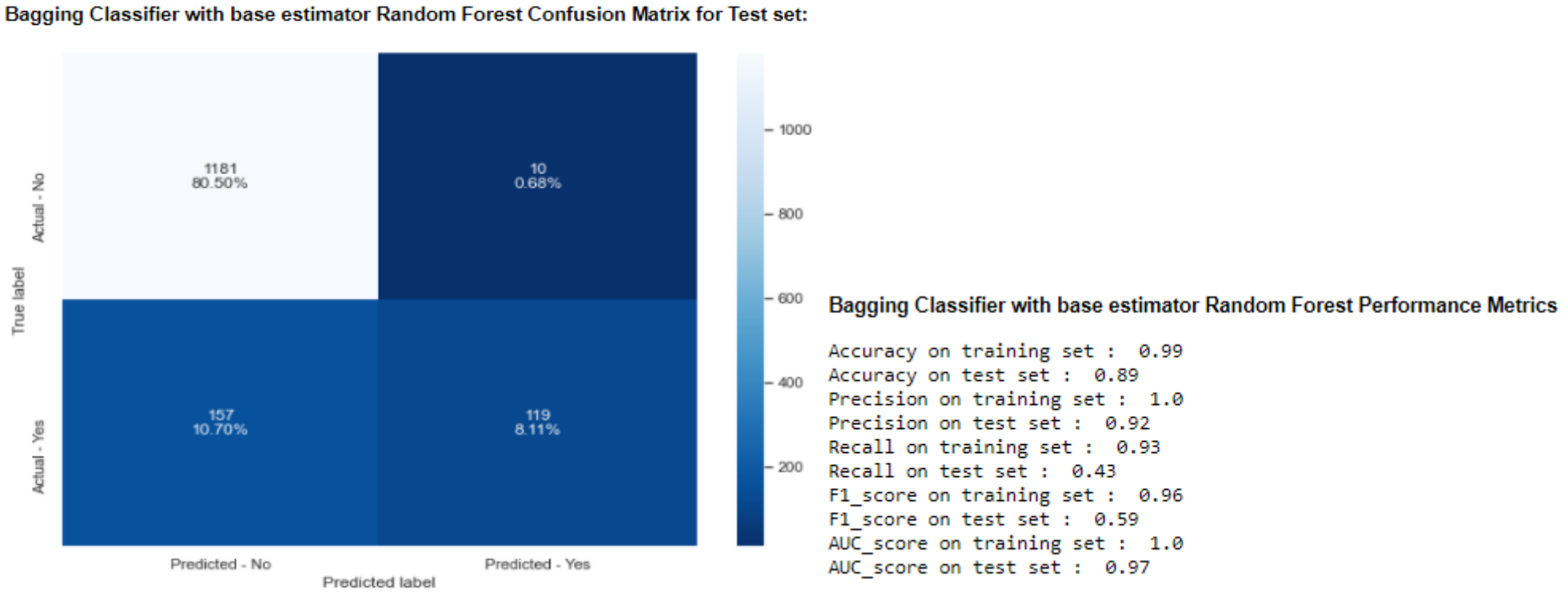


Figure-6j: Model Output of Bagging with base estimator: weighted Random Forest [Before Hyper-parameter tuning]

11. Bagging classifier with base estimator: Weighted Decision Tree [After tuning]

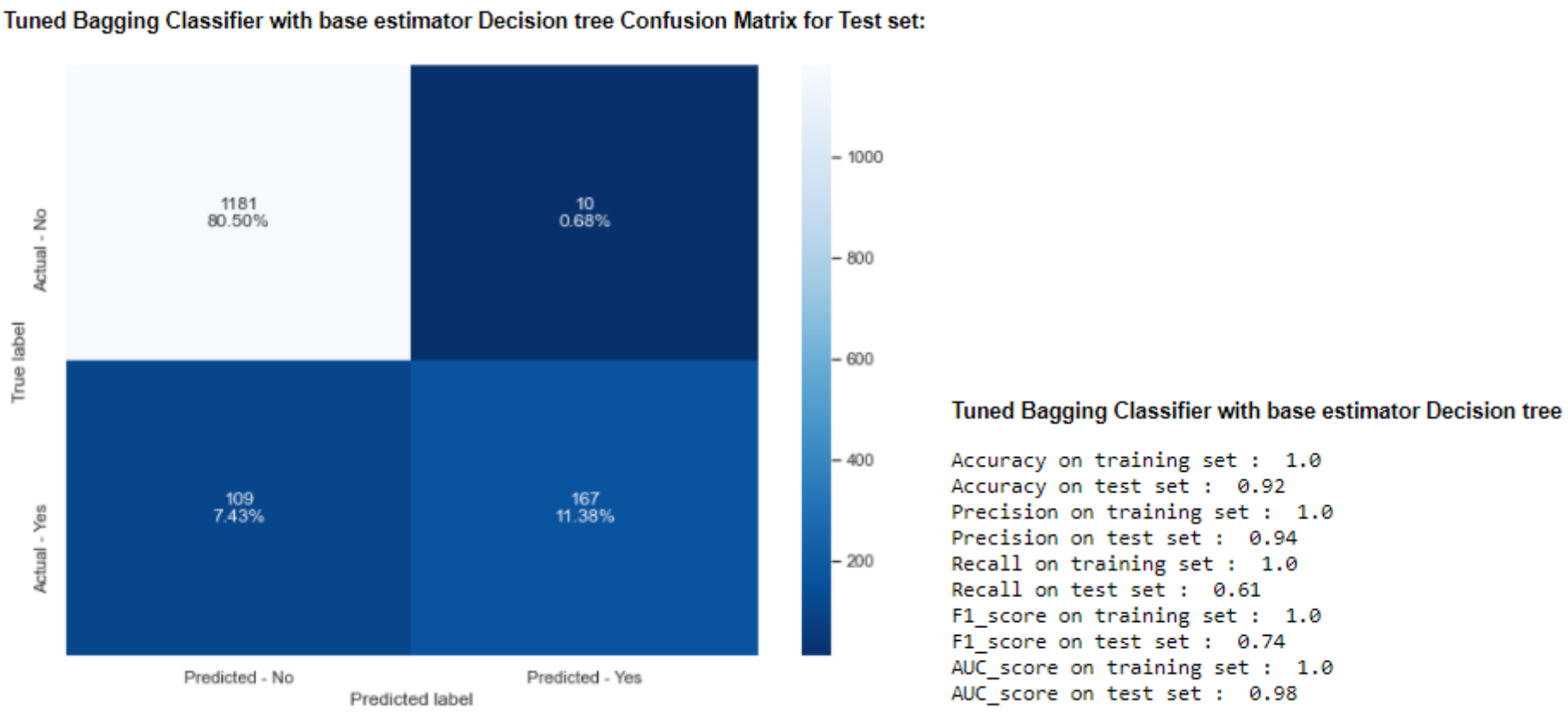


Figure-6k: Model Output of Bagging with base estimator: weighted Decision tree [After Hyper-parameter tuning]

12. Ensemble Modeling-AdaBoost [Before tuning]

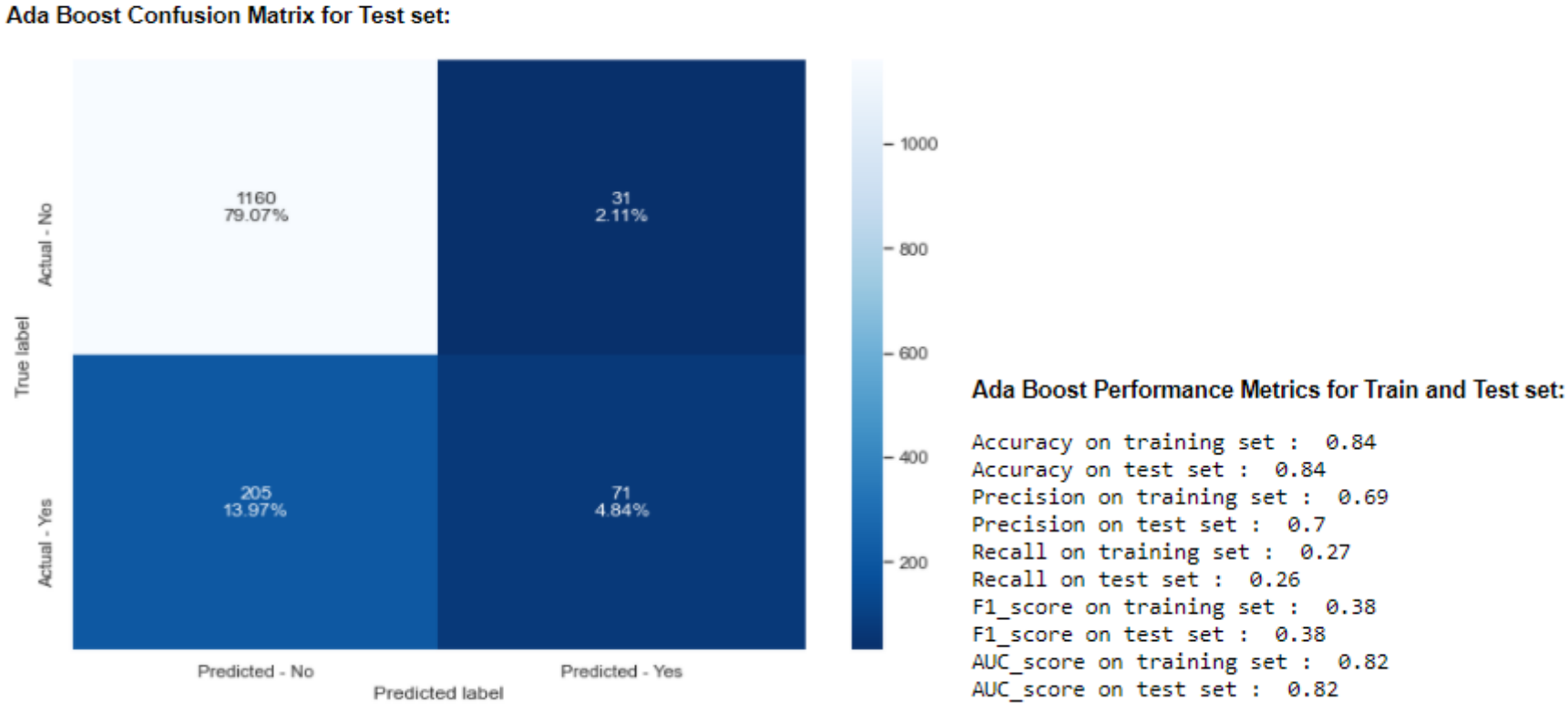
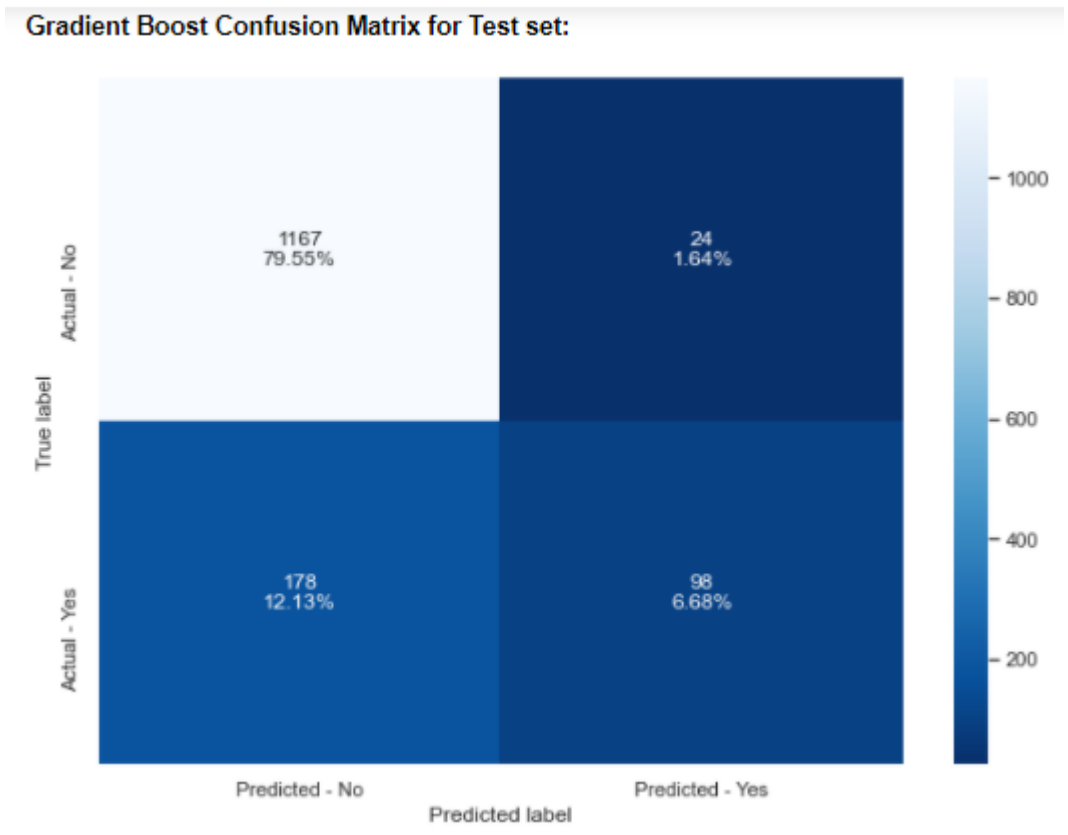


Figure-6L: Model Output of Ada Boost [Before Hyper-parameter tuning]

13. Ensemble Modeling-Gradient Boost [Before tuning]

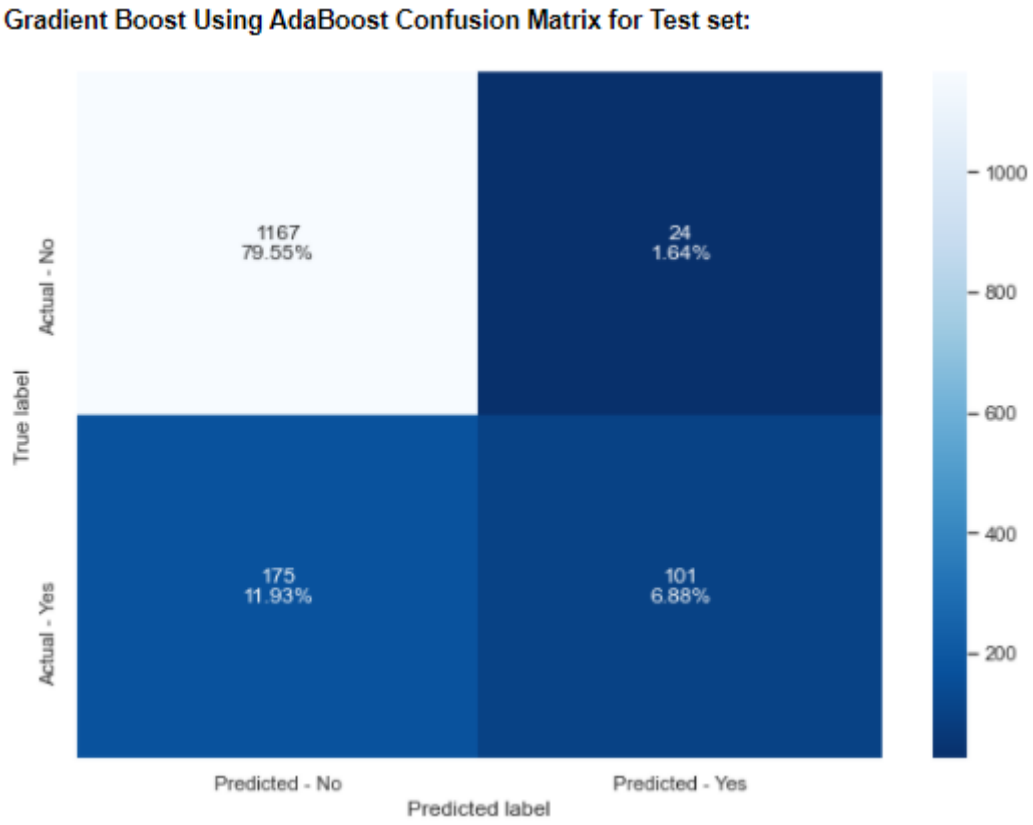


Gradient Boost Performance Metrics for Train and Test set:

```
Accuracy on training set :  0.88
Accuracy on test set :  0.86
Precision on training set :  0.88
Precision on test set :  0.8
Recall on training set :  0.44
Recall on test set :  0.36
F1_score on training set :  0.58
F1_score on test set :  0.49
AUC_score on training set :  0.9
AUC_score on test set :  0.87
```

Figure-6M: Model Output of Gradient Boost [Before Hyper-parameter tuning]

14. Ensemble Modeling- Gradient Boost using Ada boost initial predictions [Before tuning]

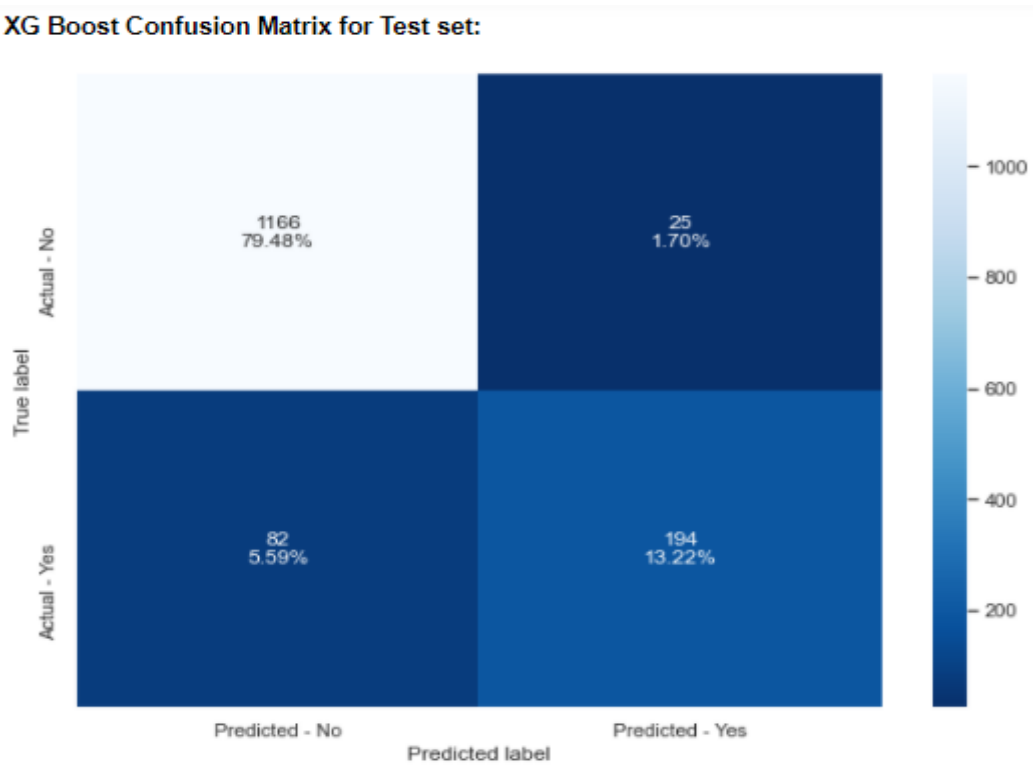


Gradient Boost Using AdaBoost Performance Metrics for Train and Test set:

```
Accuracy on training set :  0.88
Accuracy on test set :  0.86
Precision on training set :  0.86
Precision on test set :  0.81
Recall on training set :  0.44
Recall on test set :  0.37
F1_score on training set :  0.58
F1_score on test set :  0.5
AUC_score on training set :  0.9
AUC_score on test set :  0.87
```

Figure-6N: Model Output of Gradient Boost using Ada boost initial predictions [Before Hyper-parameter tuning]

15. Ensemble Modeling- XGBoost [Before tuning]



XG Boost Performance Metrics for Train and Test set:

```
Accuracy on training set :  1.0
Accuracy on test set :  0.93
Precision on training set :  1.0
Precision on test set :  0.89
Recall on training set :  1.0
Recall on test set :  0.7
F1_score on training set :  1.0
F1_score on test set :  0.78
AUC_score on training set :  1.0
AUC_score on test set :  0.95
```

Figure-6o: Model Output of XG Boost [Before Hyper-parameter tuning]

16. Ensemble Modeling-AdaBoost [After tuning]

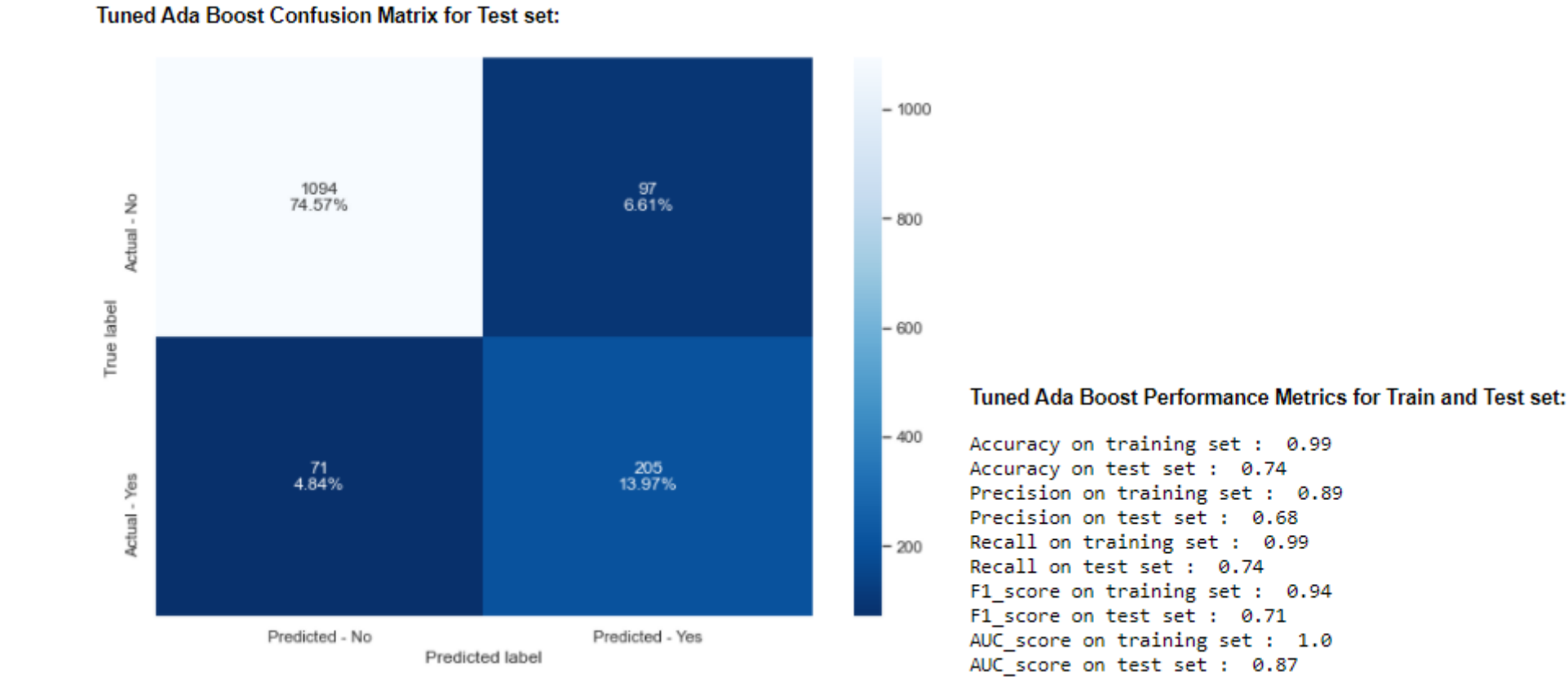


Figure-6p: Model Output of Ada Boost [After Hyper-parameter tuning]

17. Ensemble Modeling- Gradient Boost using Ada boost initial predictions [After tuning]

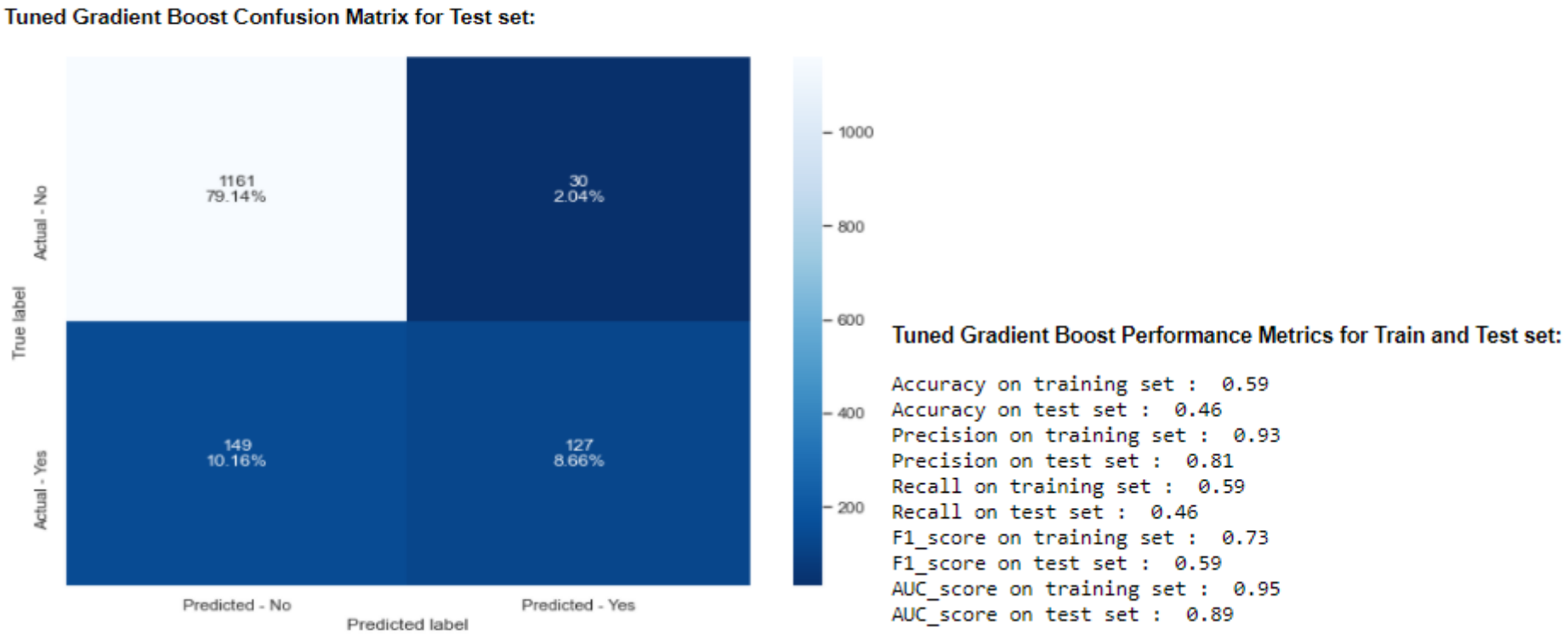


Figure-6q: Model Output of Gradient Boost using Ada boost initial predictions [After Hyper-parameter tuning]

18. Ensemble Modeling- XGBoost [After tuning]

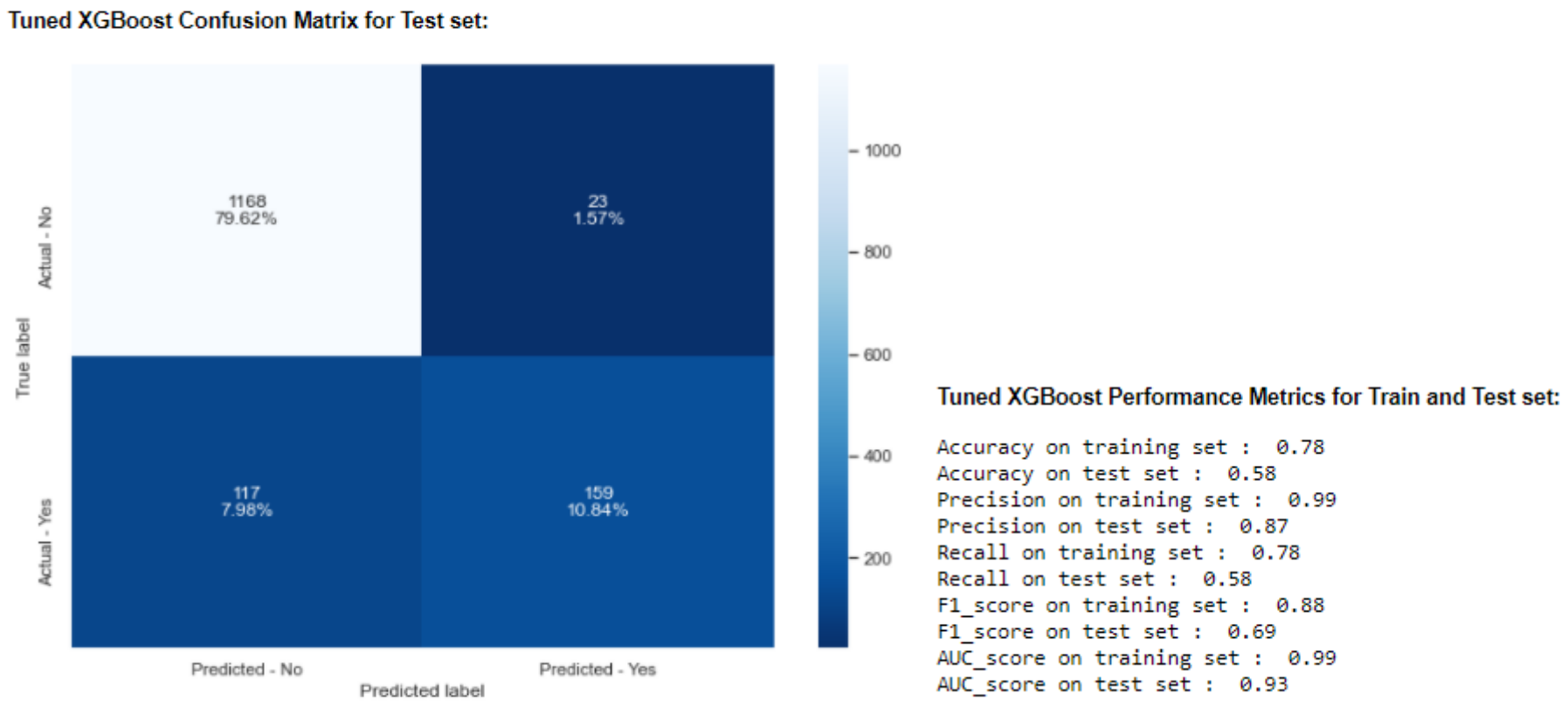


Figure-6r: Model Output of XG Boost [Before Hyper-parameter tuning]

6. Final interpretation / recommendation:

- Detailed recommendations for the management/client based on the analysis done.

Final Interpretation - Model Comparison:

	Model	Train_Accuracy	Test_Accuracy	Train_Precision	Test_Precision	Train_Recall	Test_Recall	Train_F1	Test_F1
0	LR	0.71	0.73	0.36	0.39	0.72	0.78	0.72	0.78
1	LDA	0.84	0.85	0.67	0.70	0.33	0.34	0.33	0.34
2	Decision Tree	1.00	0.91	1.00	0.76	1.00	0.76	1.00	0.76
3	Random Forest	1.00	0.92	1.00	0.94	1.00	0.61	1.00	0.61
4	LR Tuned	0.73	0.80	0.38	0.41	0.73	0.80	0.73	0.80
5	Decision Tree Tuned	0.68	0.71	0.30	0.32	0.68	0.71	0.68	0.71
6	Random Forest_tuned	0.96	0.75	0.79	0.69	0.96	0.75	0.96	0.75
7	Bagging	1.00	0.93	1.00	0.89	0.98	0.72	0.98	0.72
8	Bagging_DTCL	0.99	0.91	1.00	0.87	0.98	0.63	0.98	0.63
9	Bagging_RF	0.99	0.89	1.00	0.92	0.93	0.43	0.93	0.43
10	AdaBoost	0.84	0.84	0.69	0.70	0.27	0.26	0.27	0.26
11	GradientBoost	0.88	0.86	0.88	0.80	0.44	0.36	0.44	0.36
12	GBC_init	0.88	0.86	0.86	0.81	0.44	0.37	0.44	0.37
13	XGBOOST	1.00	0.93	1.00	0.89	1.00	0.70	1.00	0.70
14	Bagging_DTCL_tuned	1.00	0.92	1.00	0.94	1.00	0.61	1.00	0.61
15	AdaBoost_tuned	0.99	0.74	0.89	0.68	0.99	0.74	0.99	0.74
16	GradientBoost_tuned	0.59	0.46	0.93	0.81	0.59	0.46	0.59	0.46
17	XGBoost_tuned	0.78	0.58	0.99	0.87	0.78	0.58	0.78	0.58

Table-8: Model Comparison

Below are the models that performed best on accuracy and recall score:

- Weighted Logistic Regression
- Tuned Weighted Decision Tree

The final best optimum model selected is simple weighted Logistic Regression with a better accuracy score of 71% on train and 73% on test set and followed by recall score of 72% on train and 78% on test set.

Features Importance of Weighted Logistic Regression:

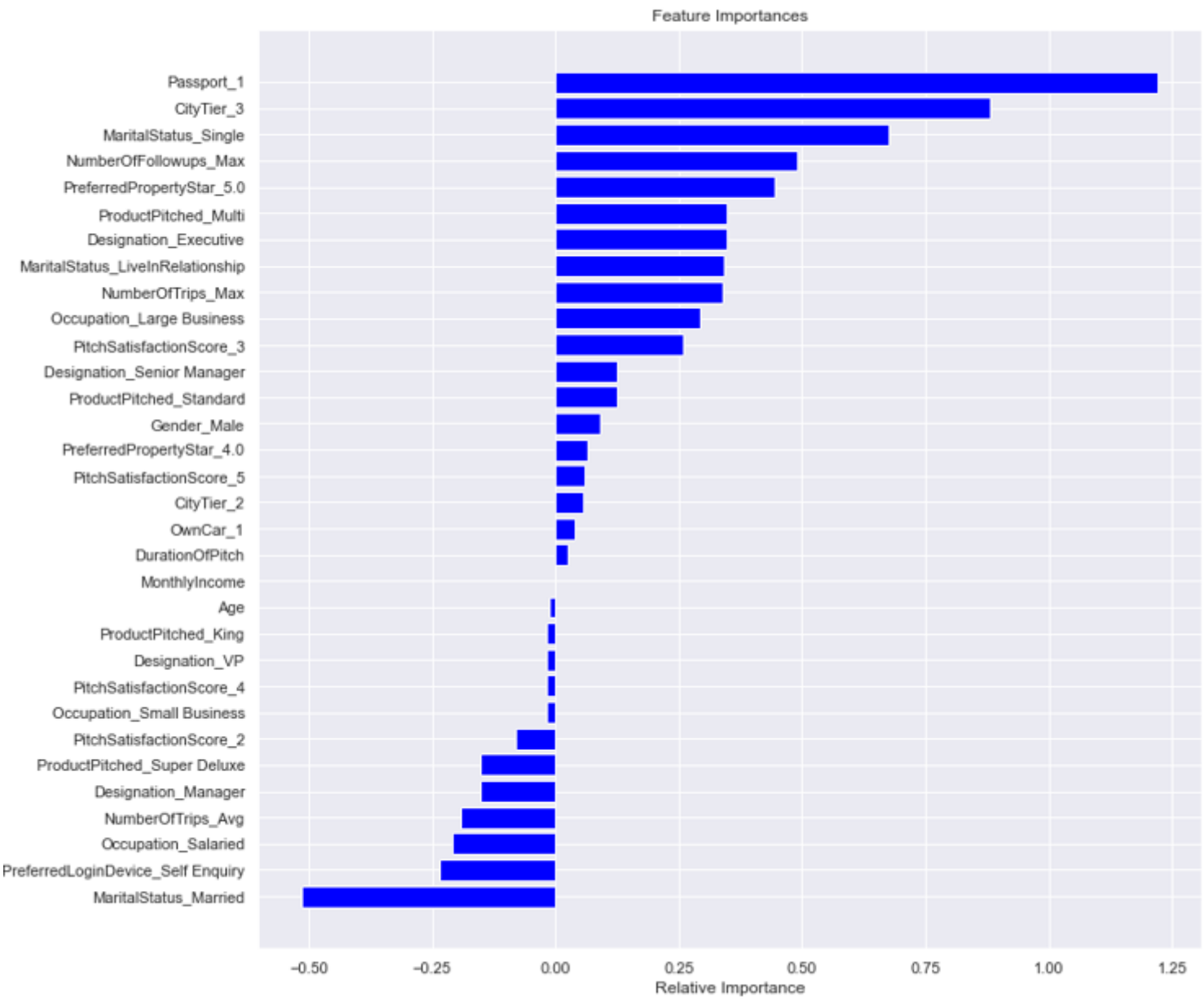


Figure-7: Feature Importance of Weighted Logistic Regression

- Passport_1, City Tier_3, MaritalStatus_Single, NumberOfFollowups_Max, PreferredPropertyStar_5.0 are the top 5 important significant features to increase sales.
- MaritalStatus_Married, PreferredLoginDevice_SelfEnquiry, Occupation_Salaried, NumberOfTrips_Avg, Designation_Manager are 5 significant features which will reduce the sales.

Business Insights:

- To increase sales revenue Target customers whose:
  - Age is less than 40 years.
  - Gender is Male.
  - Marital Status is Single and LiveInRelationship.
  - Occupation belongs to Large Business.
  - Designation belongs to Executive and Senior Manager.

- Target Customer who have passport has higher chances of buying tourism package.
- Target Customer who belongs to city tier 3 has higher chances of buying the tourism package.
- Target Customer who prefers to stay at 5 star rating property has higher chances of buying.

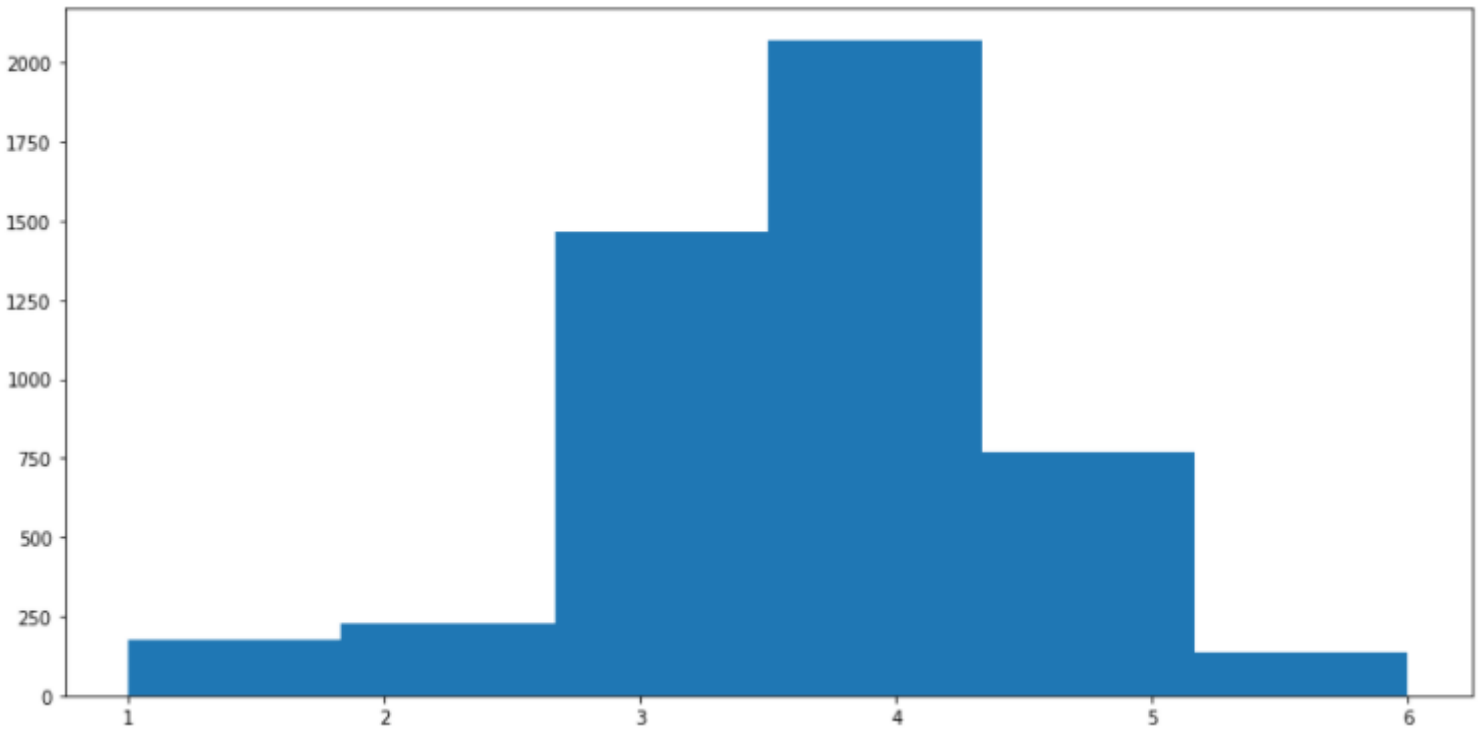
Business Recommendations:

- Suggest Business to expand their business at city tier 3 in order to increase the revenue.
- Suggest Business to increase the duration of sales pitch and also to increase the number of follow-ups made by sales team which will definitely generate a good amount revenue to the company.
- Product Super Deluxe is having no impact to business, recommend to change product policy or offers, so that in future could help to increase sales.
- Suggest business to focus on collecting more data whose occupation is free Lancers has we had only 2 records and both the customers bought the tourism package.
- Suggest business to improve or attract the customers who comes in contact with company through self-enquiry. At least with 10-20% conversion rate would increase company.

THANK YOU

Appendix:

1. NumberofFollowups: Creating Bins



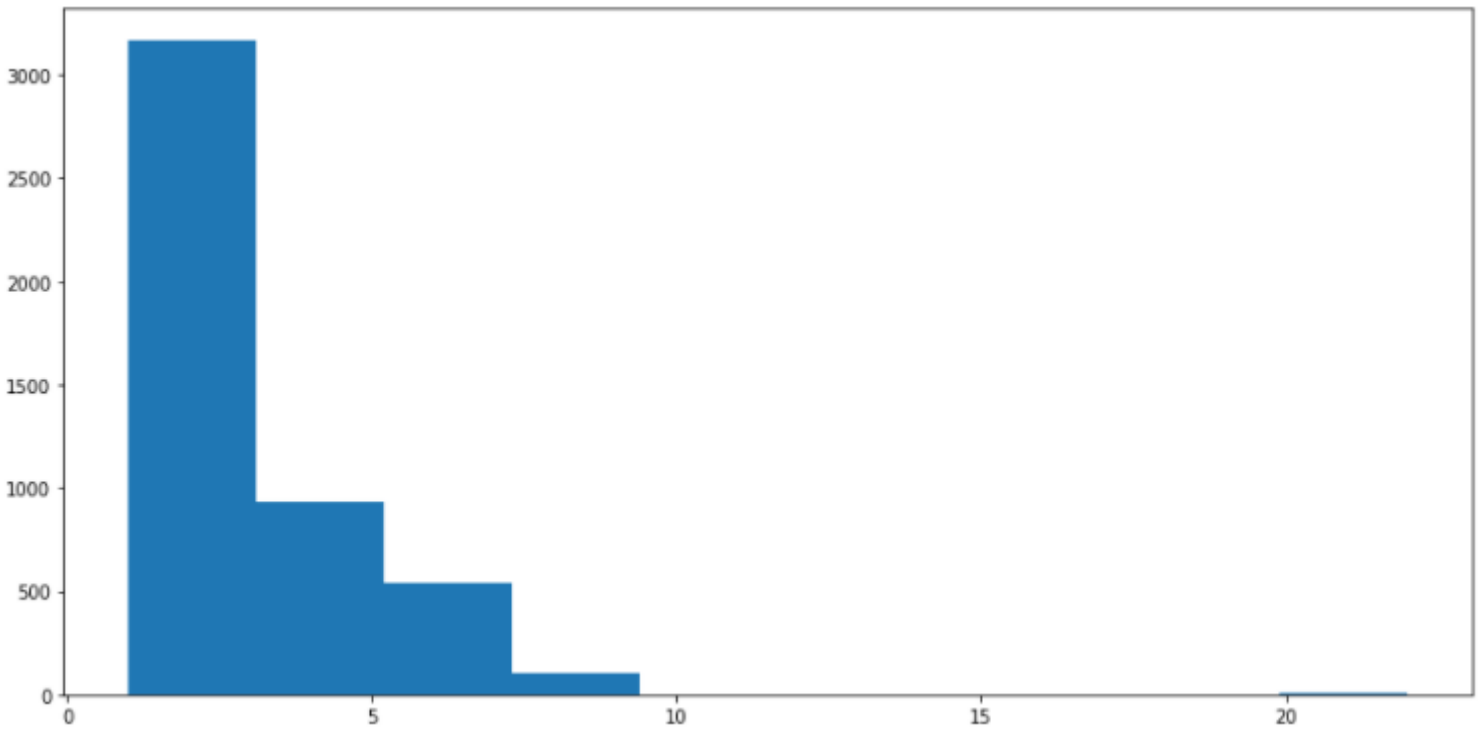
Raw code:

```
# For Age Variable: (Since Age variable is normally distributed use qcut)
bin_labels = ['Min','Max']
df['NumberOfFollowups'] = pd.cut(df['NumberOfFollowups'], bins=[0, 3, 6], labels=bin_labels)
```

Output:

```
Min    1871
Max    2972
Name: NumberOfFollowups, dtype: int64
```

2. NumberofTrips: Creating Bins



Raw Code:

```
# For Age Variable: (Since Age variable is normally distributed use qcut)
bin_labels = ['Min','Avg','Max']
df['NumberOfTrips'] = pd.cut(df['NumberOfTrips'], bins=[0, 2, 5, 25], labels=bin_labels)
```

Output:

```
Min    2084
Avg    2015
Max     649
Name: NumberOfTrips, dtype: int64
```

Model Building:

1. Weighted Logistic Regression:

```
LogisticRegression(class_weight='balanced', random_state=1)
```

2. Weighted Decision Tree:

```
DecisionTreeClassifier(class_weight='balanced', random_state=1)
```

3. Weighted Random Forest:

```
RandomForestClassifier(class_weight='balanced', random_state=1)
```

4. Weighted Logistic Regression: [Hyper-parameters]

```
GridSearchCV(cv=StratifiedKFold(n_splits=10, random_state=None, shuffle=False),
             estimator=LogisticRegression(class_weight='balanced',
                                          random_state=1),
             n_jobs=-1,
             param_grid={'max_iter': [100], 'penalty': ['l2'],
                         'solver': ['newton-cg'], 'tol': [0.01]},
             scoring=make_scorer(recall_score))
```

5. Weighted Decision Tree: [Hyper-parameters]

```
GridSearchCV(cv=StratifiedKFold(n_splits=10, random_state=None, shuffle=False),
             estimator=DecisionTreeClassifier(class_weight='balanced',
                                              random_state=1),
             param_grid={'max_depth': array([ 2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
       19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]),
                         'max_leaf_nodes': [10],
                         'min_impurity_decrease': [0.0001],
                         'min_samples_leaf': [1, 2, 5, 10, 15]},
             scoring=make_scorer(recall_score))
```

6. Weighted Random Forest: [Hyper-parameters]

```
GridSearchCV(cv=StratifiedKFold(n_splits=10, random_state=None, shuffle=False),
             estimator=RandomForestClassifier(class_weight='balanced',
                                              random_state=1),
             n_jobs=-1,
             param_grid={'max_features': ['log2'],
                         'max_samples': [0.7, 0.9, None],
                         'min_samples_leaf': array([1, 2, 3, 4, 5]),
                         'n_estimators': [110]},
             scoring=make_scorer(recall_score))
```

7. Bagging Classifier with base estimator Weighted Decision Tree:

```
BaggingClassifier(base_estimator=DecisionTreeClassifier(class_weight='balanced',
                                                        random_state=1),
                  random_state=1)
```

8. Bagging Classifier with base estimator Weighted Random Forest:

```
BaggingClassifier(base_estimator=RandomForestClassifier(class_weight='balanced',
                                                        random_state=1),
                  random_state=1)
```

9. Bagging Classifier with base estimator Weighted Decision Tree: [Hyper-parameters]

```
GridSearchCV(cv=StratifiedKFold(n_splits=10, random_state=None, shuffle=False),
             estimator=BaggingClassifier(random_state=1),
             param_grid={'base_estimator': [DecisionTreeClassifier(class_weight='balanced',
                                                                   random_state=1)],
                         'max_features': [0.7], 'n_estimators': [101]},
             scoring=make_scorer(recall_score))
```

10. Ada Boost Classifier: [Hyper-parameters]

```
GridSearchCV(cv=StratifiedKFold(n_splits=10, random_state=None, shuffle=False),
             estimator=AdaBoostClassifier(random_state=1),
             param_grid={'base_estimator': [DecisionTreeClassifier(class_weight='balanced',
                                                                   max_depth=3)],
                         'learning_rate': [0.8], 'n_estimators': [100]},
             scoring=make_scorer(recall_score))
```

11. Gradient Boost Classifier: [Hyper-parameters]

```
GridSearchCV(cv=StratifiedKFold(n_splits=10, random_state=None, shuffle=False),
             estimator=GradientBoostingClassifier(init=AdaBoostClassifier(random_state=1),
                                                  random_state=1),
             param_grid={'max_features': [0.9], 'n_estimators': [250],
                         'subsample': [1]},
             scoring=make_scorer(recall_score))
```

12. XG Boost Classifier: [Hyper-parameters]

```
GridSearchCV(cv=StratifiedKFold(n_splits=10, random_state=None, shuffle=False),
             estimator=XGBClassifier(base_score=None, booster=None,
                                     colsample_bylevel=None,
                                     colsample_bynode=None,
                                     colsample_bytree=None,
                                     eval_metric='logloss', gamma=None,
                                     gpu_id=None, importance_type='gain',
                                     interaction_constraints=None,
                                     learning_rate=None, max_delta_step=None,
                                     max_depth=None, mi...
                                     missing=nan, monotone_constraints=None,
                                     n_estimators=100, n_jobs=None,
                                     num_parallel_tree=None, random_state=1,
                                     reg_alpha=None, reg_lambda=None,
                                     scale_pos_weight=None, subsample=None,
                                     tree_method=None, validate_parameters=None,
                                     verbosity=None),
             param_grid={'colsample_bylevel': [1], 'colsample_bytree': [1],
                         'learning_rate': [0.1], 'n_estimators': [90],
                         'subsample': [1]},
             scoring=make_scorer(recall_score))
```

Feature Importance of Decision Tree: