CAPSTONE PROJECT

PROJECT REPORT

By: PRADEEP PAL

PROJECT NOTES - 1

TABLE OF CONTENTS

LIST OF TABLES

LIST OF TABLES

LIST OF FIGURES

BUSINESS OBJECTIVE

An aviation company that provides domestic as well as international trips to the customers now wants to apply a targeted approach instead of reaching out to each of the customers. This time they want to do it digitally instead of tele calling. Hence they have collaborated with a social networking platform, so they can learn the digital and social behaviour of the customers and provide the digital advertisement on the user page of the targeted customers who have a high propensity to take up the product. Propensity of buying tickets is different for different login devices. Hence, you have to create 2 models separately for Laptop and Mobile. [Anything which is not a laptop can be considered as mobile phone usage.] The advertisements on the digital platform are a bit expensive; hence, you need to be very accurate while creating the models.

Data Dictionary

Variable	Description
UserID	Unique ID of user
Buy_ticket	Buy ticket in next month
Yearly_avg_view_on_travel_page	Average yearly views on any travel related page by user
preferred_device	Through which device user preferred to do login
total_likes_on_outstation_checkin_given	Total number of likes given by a user on out of station checkings in last year
yearly_avg_Outstation_checkins	Average number of out of station check-in done by user
member_in_family	Total number of relationship mentioned by user in the account
preferred_location_type	Preferred type of the location for travelling of user
Yearly_avg_comment_on_travel_page	Average yearly comments on any travel related page by user
total_likes_on_outofstation_checkin_received	Total number of likes received by a user on out of station checkings in last year
week_since_last_outstation_checkin	Number of weeks since last out of station check-in update by user
following_company_page	Weather the customer is following company page (Yes or No)
montly_avg_comment_on_company_page	Average monthly comments on company page by user
working_flag	Weather the customer is working or not
travelling_network_rating	Does user have close friends who also like travelling. 1 is highs and 4 is lowest
Adult_flag	Weather the customer is adult or not
Daily_Avg_mins_spend_on_traveling_page	Average time spend on the company page by user on daily basis

1. Introduction of the business problem

a. Defining problem statement

Defining Problem Statement: The problem at hand involves an aviation company that wishes to transition from a mass marketing approach to a more targeted digital advertising strategy. Specifically, the company wants to utilize the data from a social networking platform to create two separate predictive models, one for users accessing the platform via laptops and another for those using mobile devices. The primary goal is to identify and target potential customers who are highly likely to purchase domestic and international flight tickets.

Problem Statement: To develop accurate predictive models for two distinct user groups (Laptop and Mobile) based on their digital and social behavior, allowing the aviation company to deliver cost-effective, precisely targeted digital advertisements to users with a high propensity to purchase their services.

b. Need of the study/project

The shift towards digital advertising through a social networking platform is driven by several key factors:

- Cost Efficiency: Traditional telecalling methods can be expensive and time-consuming. Targeted digital advertising is more cost-effective as it allows for precise audience selection.
- Improved Customer Engagement: Digital advertisements can be tailored to match users' preferences and behaviors, resulting in higher engagement and conversion rates.
- Data-Driven Marketing: Leveraging the data available on the social networking platform, the company can gain insights into user behaviors and preferences, enhancing the effectiveness of their marketing efforts.
- Competitive Advantage: In the highly competitive aviation industry, the ability to reach potential customers more accurately and efficiently can give the company a competitive edge.

c. Understanding business/social opportunity

The business opportunity here lies in harnessing the power of data and digital advertising to make marketing efforts more effective and efficient.

By identifying high-propensity customers based on their device usage, the aviation company can increase its ROI in marketing. This approach not only optimizes advertising spending but also enhances the customer experience by offering relevant and engaging content.

In doing so, the company can potentially increase ticket sales, improve customer loyalty, and solidify its presence in the highly competitive aviation market. Furthermore, this project demonstrates the growing importance of data-driven decision-making and the integration of digital platforms to achieve business goals.

2. Data Report

a. Understanding how data was collected in terms of time, frequency and methodology

The data set is consist of 11760 entries around 17 columns. This dataset contain information related to users and their behaviors on a social networking platform, which could be linked to a travel and product purchase scenario. The data includes various types of variables, such as binary, categorical, and numerical.

While specific details regarding the data collection timeframe, frequency and methodology are unavailable, an initial observation of the dataset suggests that the data likely spans a period exceeding one year and may have been obtained through the analysis of a social media application.

b. Visual inspection of data (rows, columns, descriptive details)

Let's start with looking at top few rows and columns.

Taken_product	Yes	No	Yes	No	No
Yearly_avg_view_on_travel_page	307.0	367.0	277.0	247.0	202.0
preferred_device	iOS and Android	iOS	iOS and Android	iOS	iOS and Android
total_likes_on_outstation_checkin_given	38570.0	9765.0	48055.0	48720.0	20685.0
yearly_avg_Outstation_checkins	1	1	1	1	1
member_in_family	2	1	2	4	1
preferred_location_type	Financial	Financial	Other	Financial	Medical
Yearly_avg_comment_on_travel_page	94.0	61.0	92.0	56.0	40.0
total_likes_on_outofstation_checkin_received	5993	5130	2090	2909	3468
week_since_last_outstation_checkin	8	1	6	1	9
following_company_page	Yes	No	Yes	Yes	No
montly_avg_comment_on_company_page	11	23	15	11	12
working_flag	No	Yes	No	No	No
travelling_network_rating	1	4	2	3	4
Adult_flag	0	1	0	0	1
Daily_Avg_mins_spend_on_traveling_page	8	10	7	8	6

Table 1.1: Top rows and columns of Dataset

Shape of data set is given below

```
The number of rows (observations) is 11760
The number of columns (variables) is 17
```

Table 1.2: Shape of dataset

Let's now examine the datatype information of the variables.

```
RangeIndex: 11760 entries, 0 to 11759
Data columns (total 17 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   UserID                                  11760 non-null  int64
 1   Taken_product                           11760 non-null  object
 2   Yearly_avg_view_on_travel_page          11179 non-null  float64
 3   preferred_device                        11707 non-null  object
 4   total_likes_on_outstation_checkin_given 11379 non-null  float64
 5   yearly_avg_Outstation_checkins          11685 non-null  object
 6   member_in_family                        11760 non-null  object
 7   preferred_location_type                 11729 non-null  object
 8   Yearly_avg_comment_on_travel_page       11554 non-null  float64
 9   total_likes_on_outofstation_checkin_received 11760 non-null  int64
 10  week_since_last_outstation_checkin      11760 non-null  int64
 11  following_company_page                  11657 non-null  object
 12  montly_avg_comment_on_company_page      11760 non-null  int64
 13  working_flag                            11760 non-null  object
 14  travelling_network_rating               11760 non-null  int64
 15  Adult_flag                              11760 non-null  int64
 16  Daily_Avg_mins_spend_on_traveling_page  11760 non-null  int64
dtypes: float64(3), int64(7), object(7)
```

Table 1.3: Information about Data types of dataset

The dataset comprises of 10 variables with numeric datatype and 7 variables with object datatype.

Now, let's examine the statistical summary of the dataset.

	count	mean	std	min	25%	50%	75%	max
UserID	11760.00	1005880.50	3394.96	1000001.00	1002940.75	1005880.50	1008820.25	1011760.00
Yearly_avg_view_on_travel_page	11179.00	280.83	68.18	35.00	232.00	271.00	324.00	464.00
total_likes_on_outstation_checkin_given	11379.00	28170.48	14385.03	3570.00	16380.00	28076.00	40525.00	252430.00
Yearly_avg_comment_on_travel_page	11554.00	74.79	24.03	3.00	57.00	75.00	92.00	815.00
total_likes_on_outofstation_checkin_received	11760.00	6531.70	4706.61	1009.00	2940.75	4948.00	8393.25	20065.00
week_since_last_outstation_checkin	11760.00	3.20	2.62	0.00	1.00	3.00	5.00	11.00
montly_avg_comment_on_company_page	11760.00	28.66	48.66	11.00	17.00	22.00	27.00	500.00
travelling_network_rating	11760.00	2.71	1.08	1.00	2.00	3.00	4.00	4.00
Adult_flag	11760.00	0.79	0.85	0.00	0.00	1.00	1.00	3.00
Daily_Avg_mins_spend_on_traveling_page	11760.00	13.82	9.07	0.00	8.00	12.00	18.00	270.00

Table 1.4: Statistical summary of dataset

The table above illustrates considerable disparity in the values of various columns.

Missing values in data set

```
Yearly_avg_view_on_travel_page              581
preferred_device                             53
total_likes_on_outstation_checkin_given     381
yearly_avg_Outstation_checkins               75
preferred_location_type                      31
Yearly_avg_comment_on_travel_page           206
following_company_page                      103
```

Table 1.5: Columns having missing values present

Seven variables within the dataset contain missing values, accounting for 12.16% of the overall data.

There are no duplicated rows.

```
The sum of duplicate rows :0
```

Table 1.6: Duplicated row's sum

c. Understanding of attributes (variable info, renaming if required)

Now checking the categorical columns and modifying the attributes in case of discrepencies.

- Taken_Product: This is our Target column indicating whether the user has purchased a product or not.

No	9864
Yes	1896

Table 1.7: Count of whether a person taken product

- preferred_device: A variable indicating whether the user prefers a laptop or mobile device. We have combined all the devices other than laptop as mobile

Tab	4172
iOS and Android	4134
Laptop	1108
iOS	1095
Mobile	600
Android	594
Other	4

Mobile	10599
Laptop	1108

Table 1.8: Preferred device before and after renaming

- member_in_family: A variable indicating the number of family members. We have combined 'Three' with '3'.

3	4561
4	3184
2	2256
1	1349
5	384
Three	15
10	11

3	4576
4	3184
2	2256
1	1349
5	384
10	11

Table 1.9: Family members before and after renaming

- preferred_location_type: A variable representing the user's preferred location type. We have combined 'Tour Travel' with 'Tour and Travel'.

Beach	2424
Financial	2409
Historical site	1856
Medical	1845
Other	643
Big Cities	636
Social media	633
Trekking	528
Entertainment	516
Hill Stations	108
Tour Travel	60
Tour and Travel	47
Game	12
OTT	7
Movie	5

Beach	2424
Financial	2409
Historical site	1856
Medical	1845
Other	643
Big Cities	636
Social media	633
Trekking	528
Entertainment	516
Hill Stations	108
Tour and Travel	107
Game	12
OTT	7
Movie	5

Table 1.10: Preferred location before and after renaming

- following_company_page: A variable indicating whether the user follows the company's page or not. We have combine ‘0’ with ‘No’ and ‘1’ with ‘Yes’

No	8355
Yes	3285
1	12
0	5

No	8360
Yes	3297

Table 1.11: Following status before and after renaming

- working_flag: A variable indicating whether the user is currently employed.

No	9952
Yes	1808

Table 1.12: User’s working status

- travelling_network_rating: A variable representing the user's rating within a travel network.

3	3672
4	3456
2	2424
1	2208

Table 1.13: User’s travel Rating

- Adult_flag: A variable indicating whether the user is an adult or not. We have combined all values other than '0' to '1'.

0	5048
1	4768
2	1264
3	680

1	6712
0	5048

Table 1.14: Adult status before and after renaming

We have also changed the data type of few columns given below:

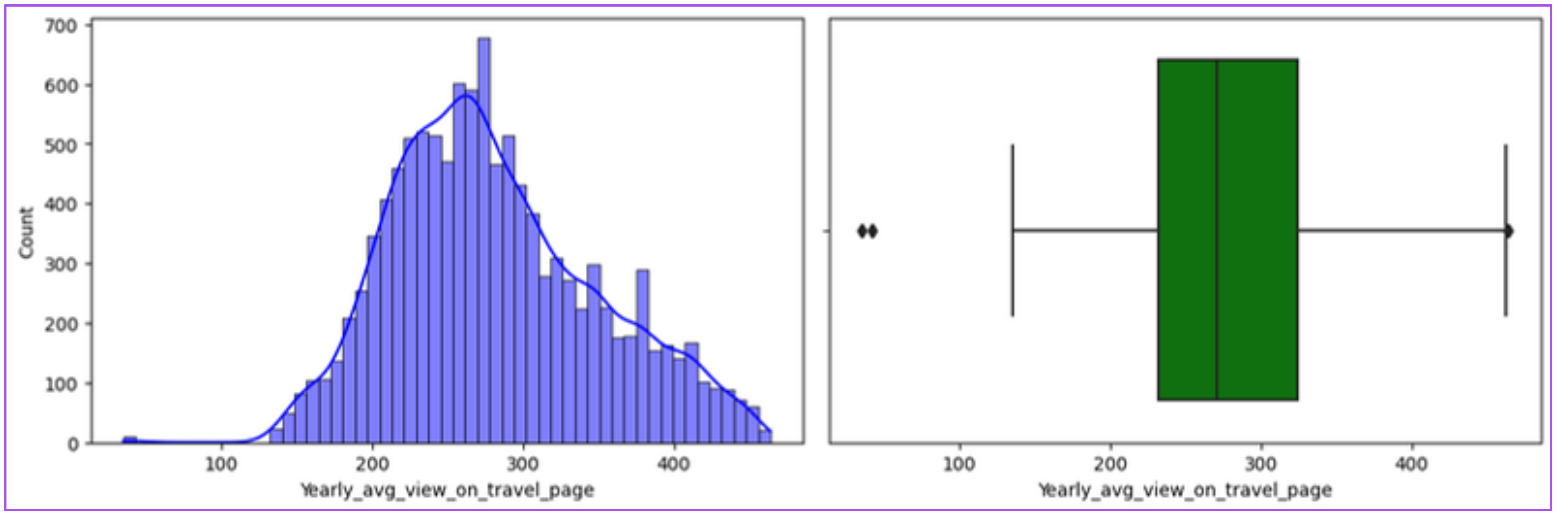
'yearly_avg_Outstation_checkins' : 'object' to 'float64'
'travelling_network_rating' : 'int64' to 'object'
'Adult_flag' : 'int64' to 'object'
'member_in_family' : 'object' to 'int64'

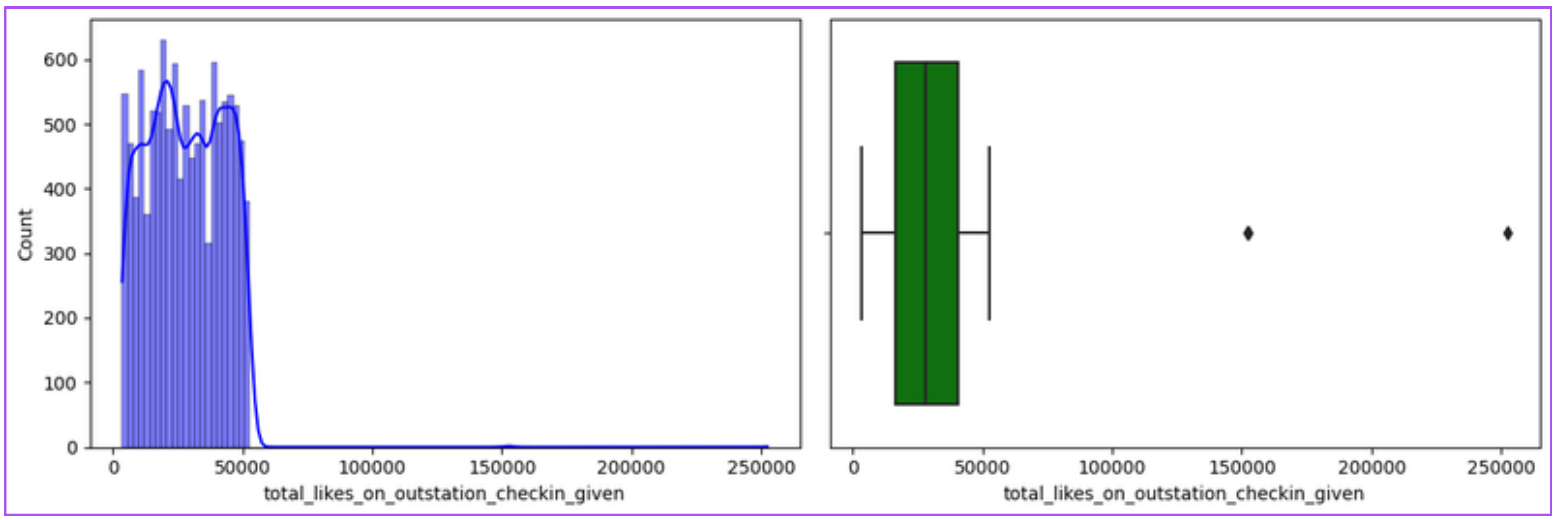
3. Exploratory data analysis

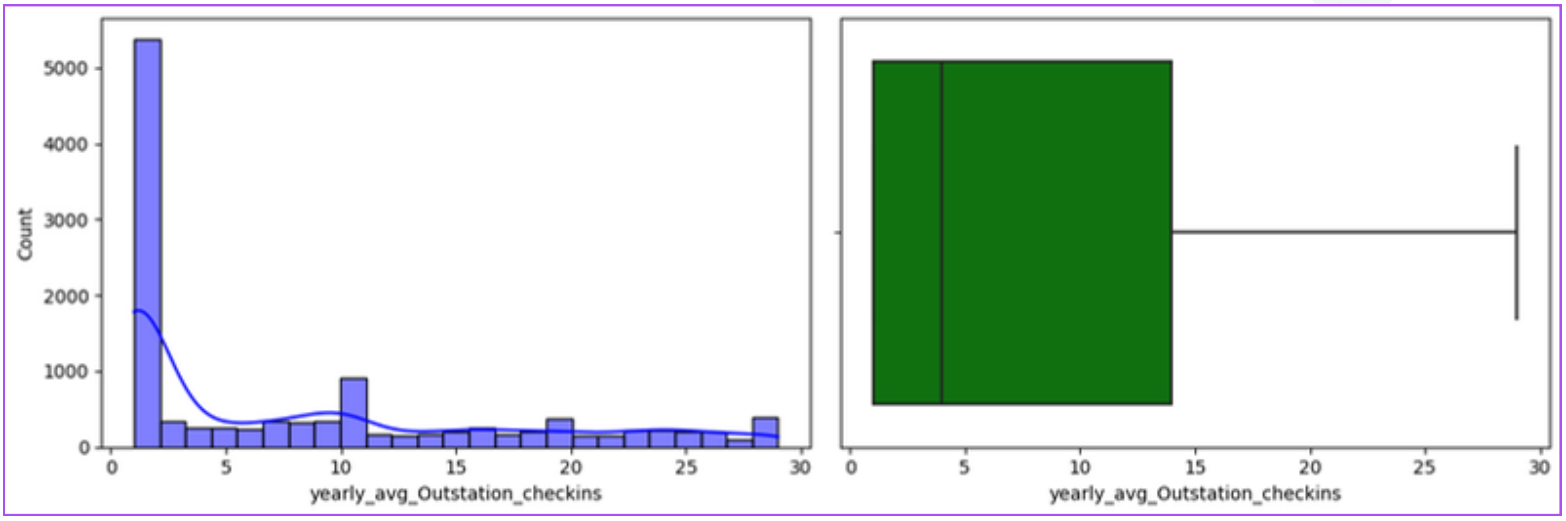
a) Univariate analysis

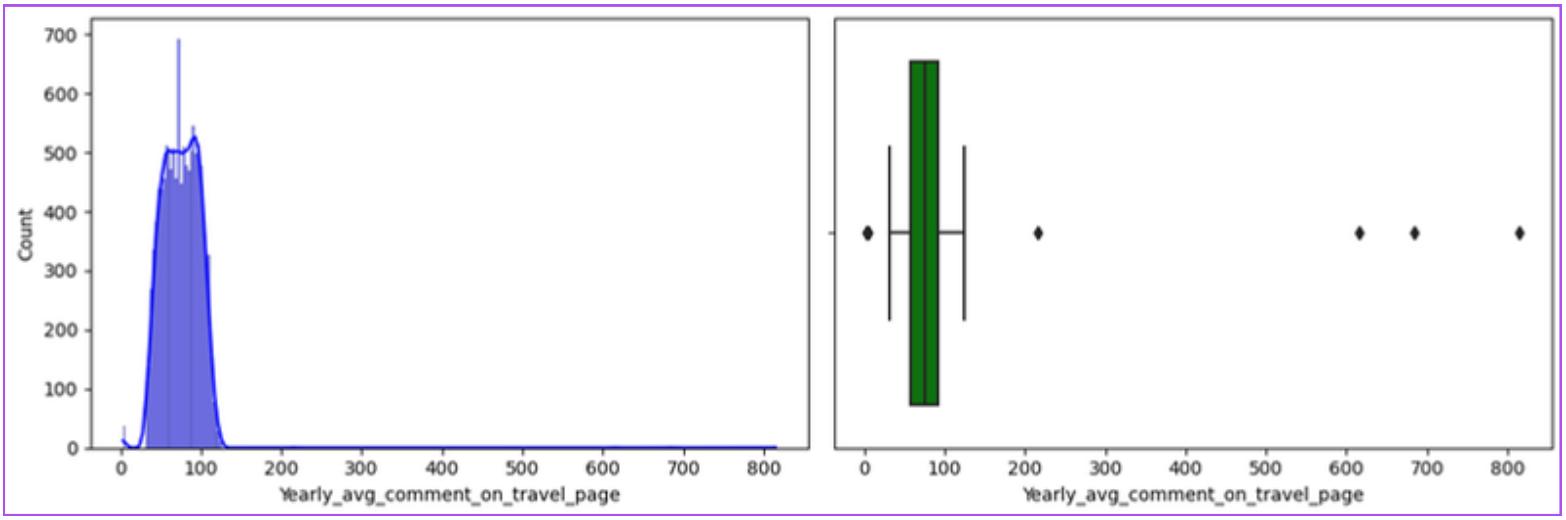
Univariate analysis is a statistical method used to analyze and describe the distribution, central tendency, and variability of a single variable or a single feature in a dataset.

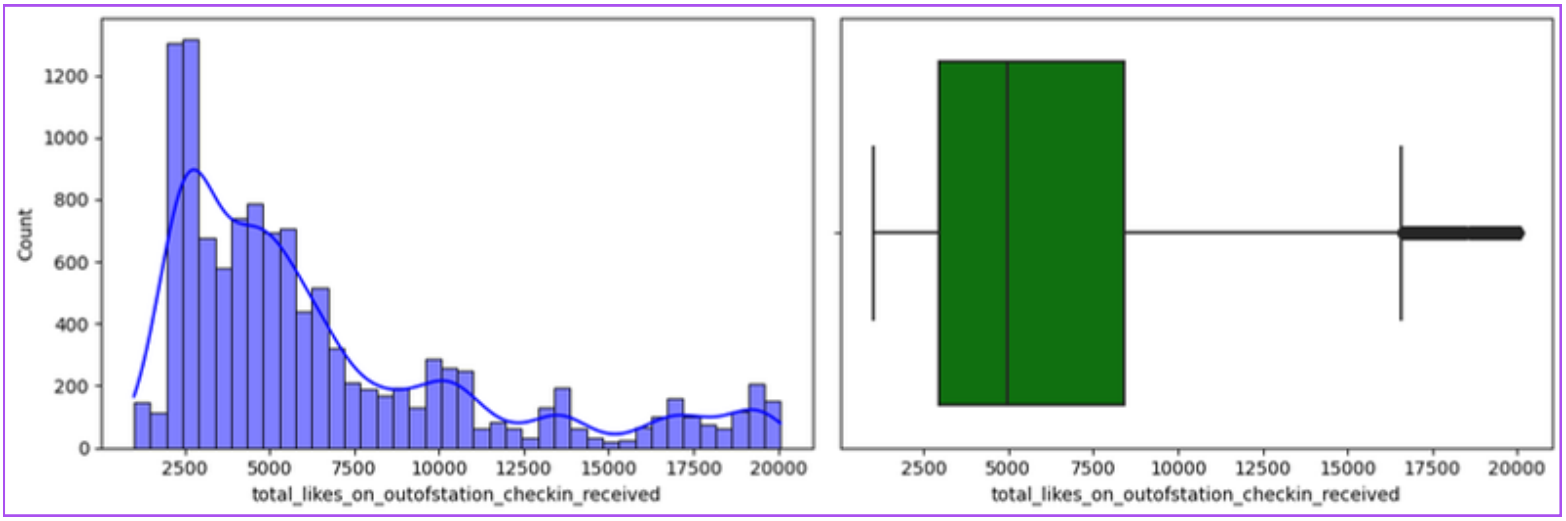
Below is univariate analysis for Numerical variables






Count
yearly_avg_Outstation_checkins
yearly_avg_Outstation_checkins


Count
Yearly_avg_comment_on_travel_page
Yearly_avg_comment_on_travel_page


Count
total_likes_on_outofstation_checkin_received
total_likes_on_outofstation_checkin_received

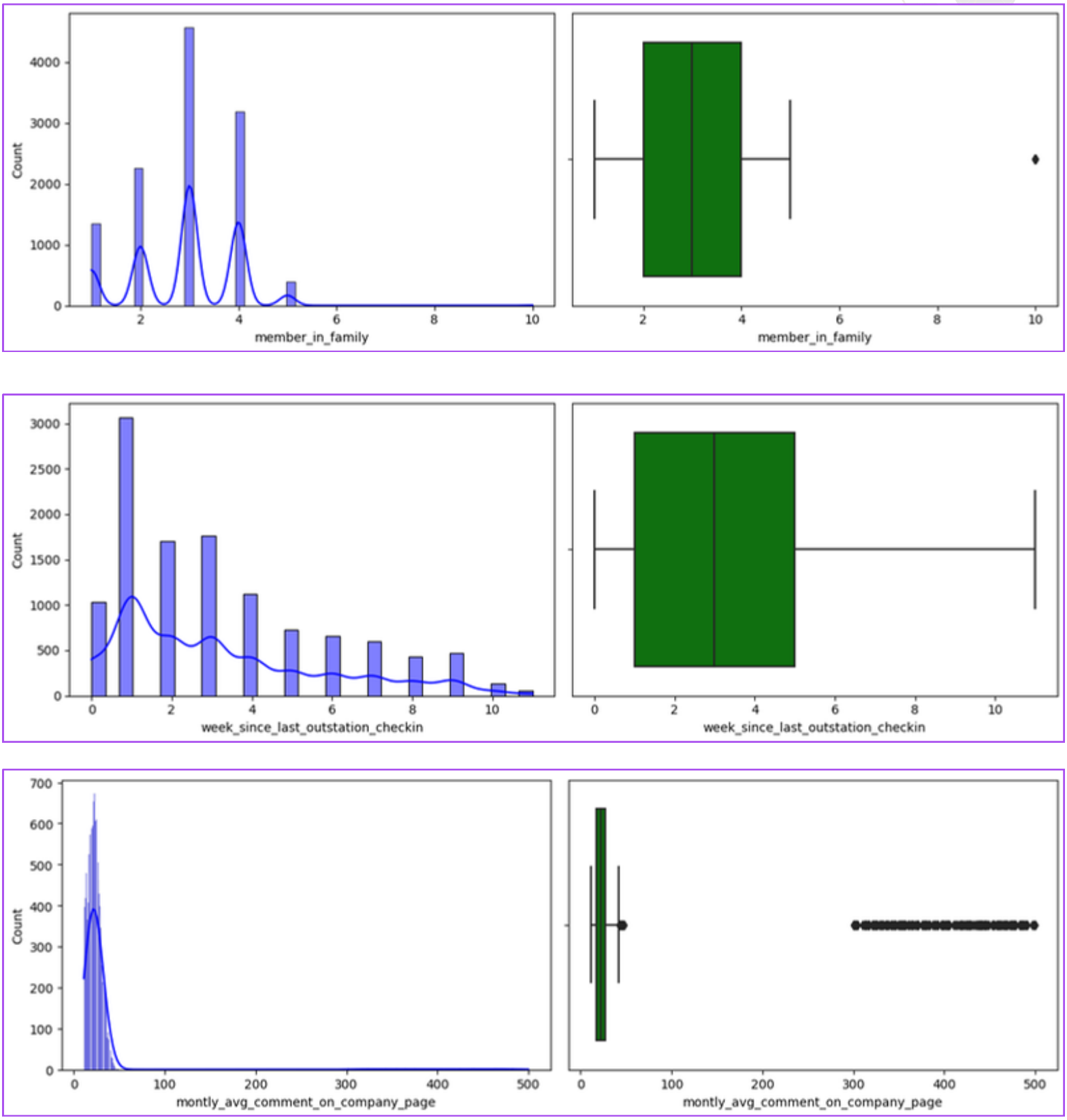


Figure 1.1: Univariate analysis of Numerical columns

Insights:

Yearly Average Views on Travel Page:

- On average, users view travel-related pages approximately 280.83 times per year.
- The data exhibits a standard deviation of 68.18, suggesting some variation in viewing frequencies.
- Views range from a minimum of 35 to a maximum of 464 times per year.

Total Likes on Outstation Check-ins Given:

- Users provide a mean of 28,170.48 likes on outstation check-ins, with a standard deviation of 14,385.03.
- The distribution indicates variability in the extent of likes given.
- The minimum number of likes given is 3, while the maximum reaches 252,430.

Yearly Average Comments on Travel Page:

- Users tend to comment on travel-related pages with an average of 74.79 comments per year.
- The standard deviation is 24.03, reflecting varying engagement levels.
- The range of comments varies from a minimum of 3 to a maximum of 815.

Total Likes on Out-of-Station Check-ins Received:

- Users receive an average of 6,531.70 likes on their out-of-station check-ins.
- The standard deviation is 4,706.61, indicating differing levels of user popularity.
- The number of likes received ranges from a minimum of 1,009 to a maximum of 20,065.

Weeks Since Last Outstation Check-in:

- On average, users have checked in for out-of-station activities approximately 3.20 weeks ago.
- The standard deviation is 2.62, signifying variation in the recency of check-ins.
- The data ranges from 0 weeks (very recent) to 11 weeks (less recent).

Monthly Average Comments on Company Page:

- Users contribute an average of 28.66 comments on the company's page each month.
- The standard deviation is relatively high at 48.66, indicating considerable variability in engagement.
- Comment frequencies range from a minimum of 11 to a maximum of 500 comments per month.

Daily Average Minutes Spent on Traveling Page:

- Users spend an average of 13.82 minutes daily on traveling-related pages.
- The standard deviation is 9.07, suggesting variation in user engagement.
- Daily time spent ranges from a minimum of 0 minutes to a maximum of 270 minutes, indicating diverse user behaviors.

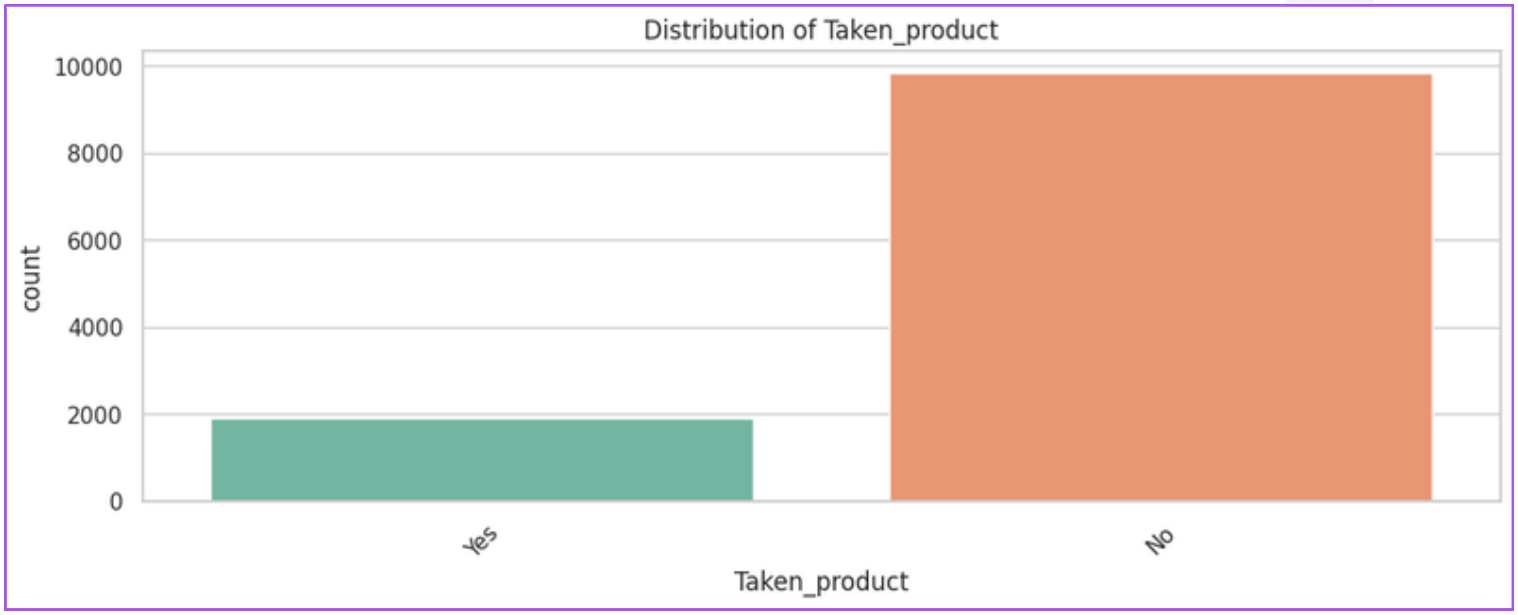
Member in Family:

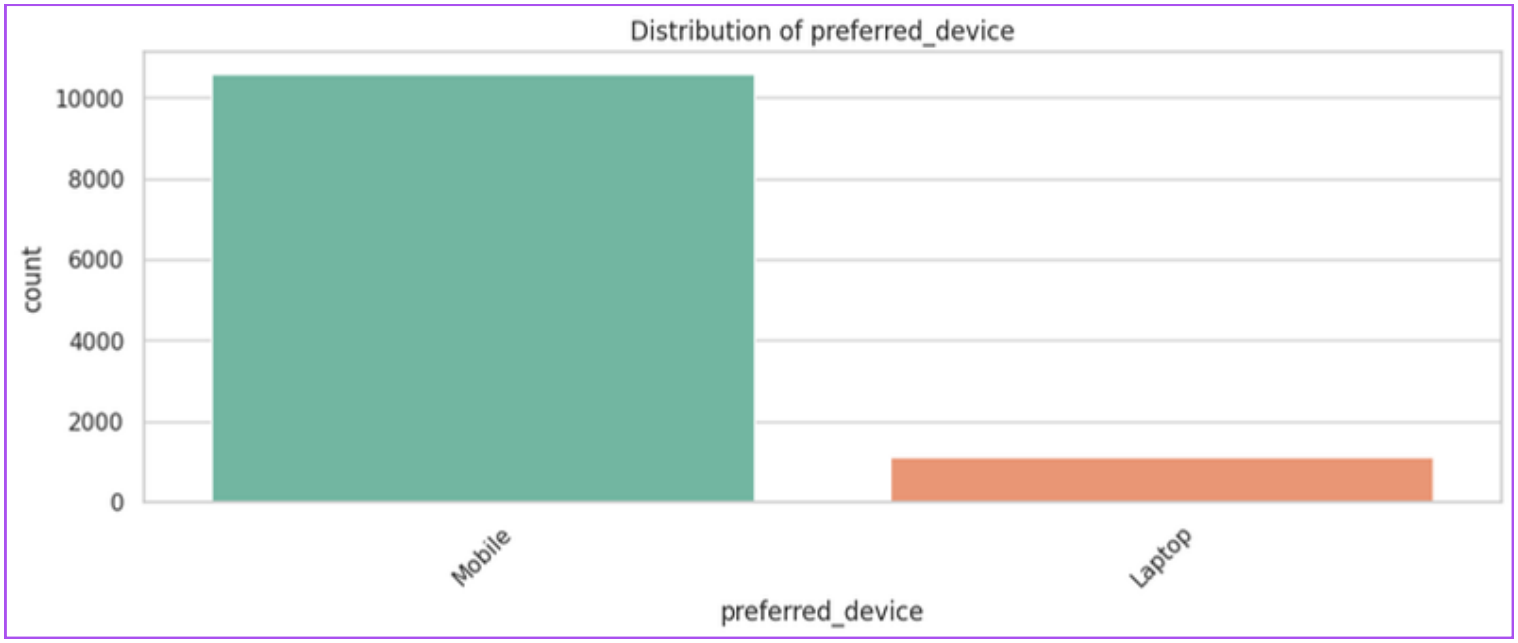
- On average, users report being part of a family with approximately 2.92 members.
- The standard deviation is relatively low, indicating consistent reporting.
- The majority of users have 2 to 4 family members.

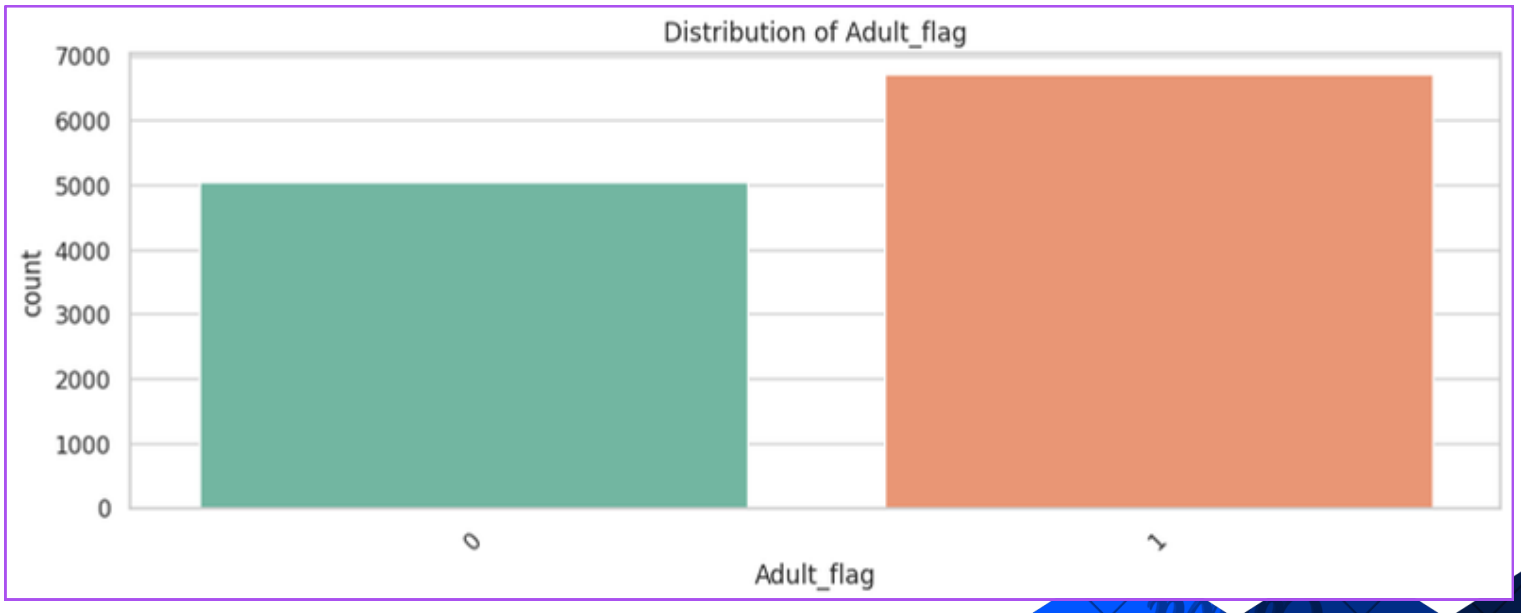
Yearly Avg Outstation Checkins:

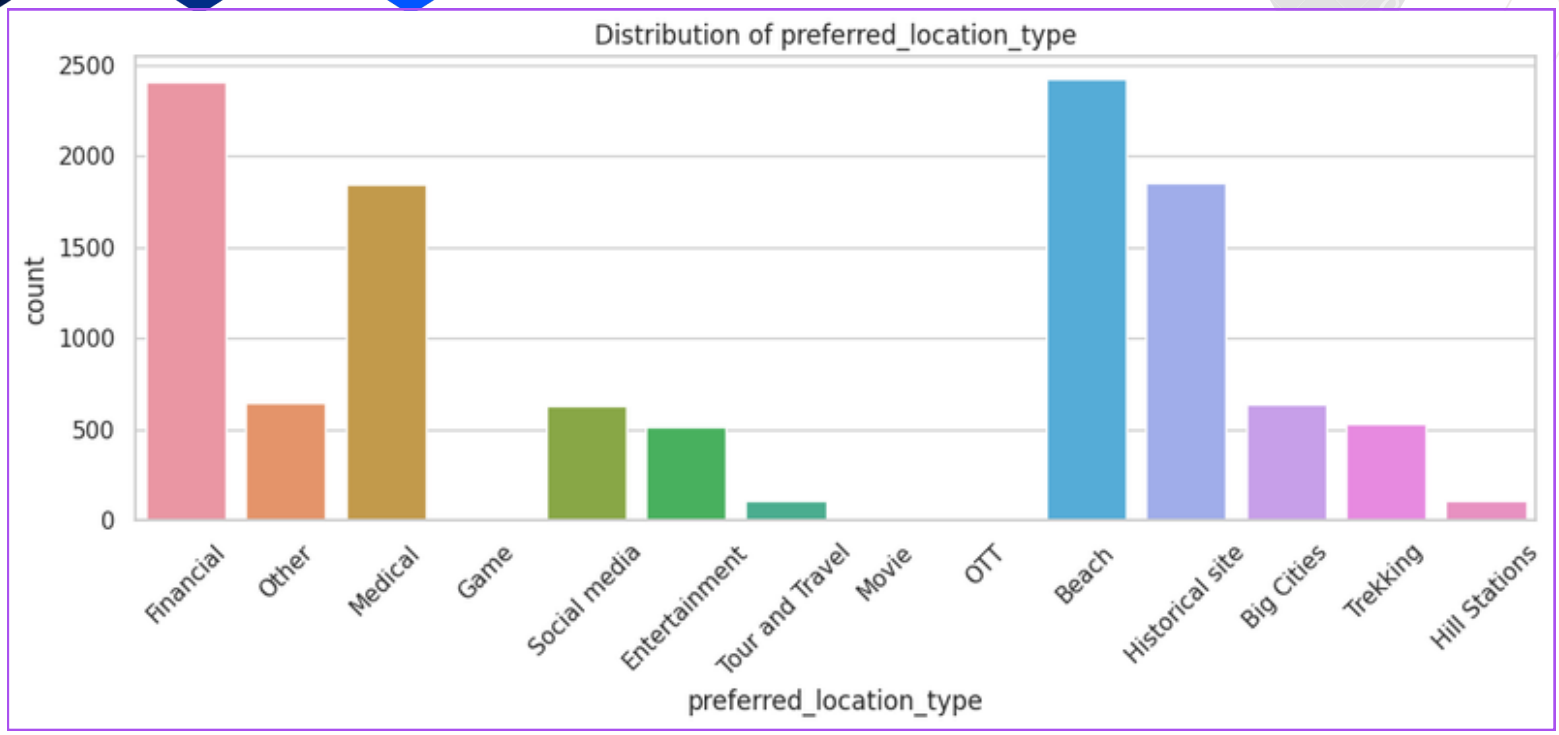
- The average yearly count of outstation check-ins is about 8.22.
- Users have a wide distribution of outstation check-ins, with the minimum at 1 and the maximum at 29.
- The median (50th percentile) is 4, indicating that half of the users have four or fewer outstation check-ins.

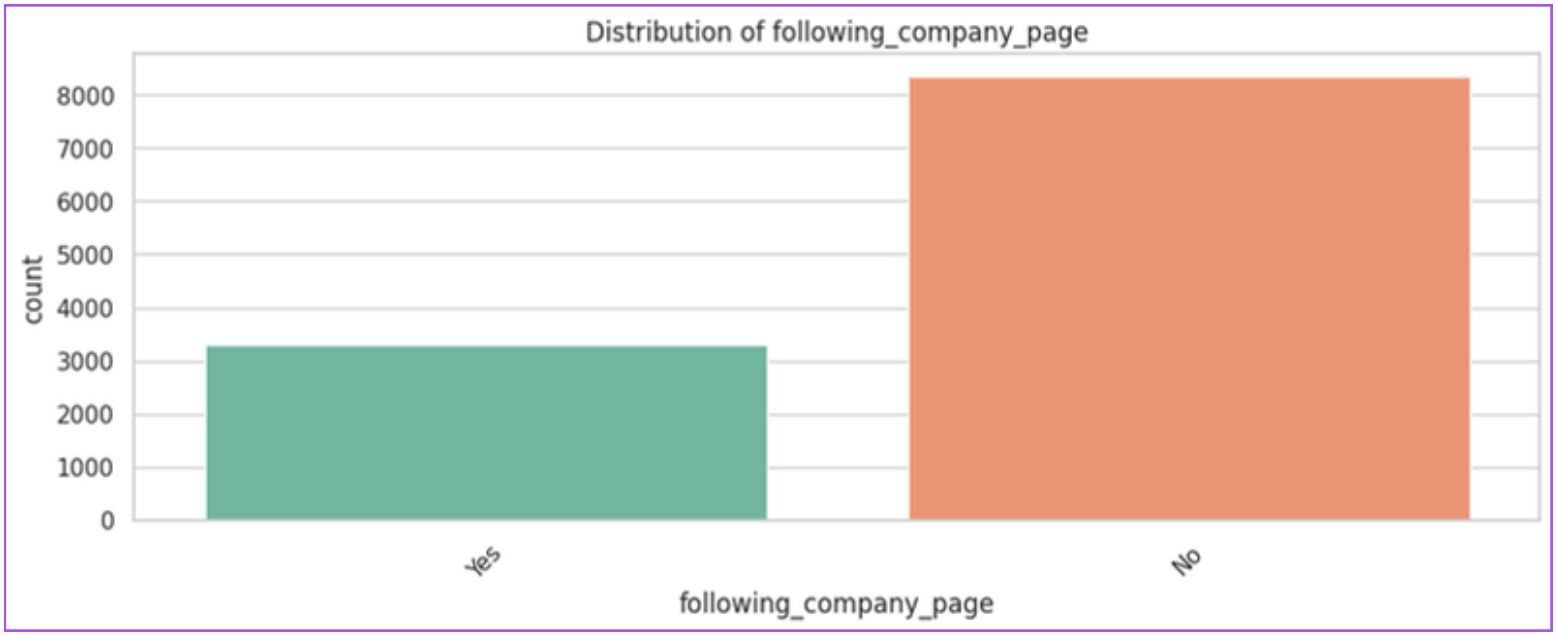
Below is univariate analysis for Categorical variables

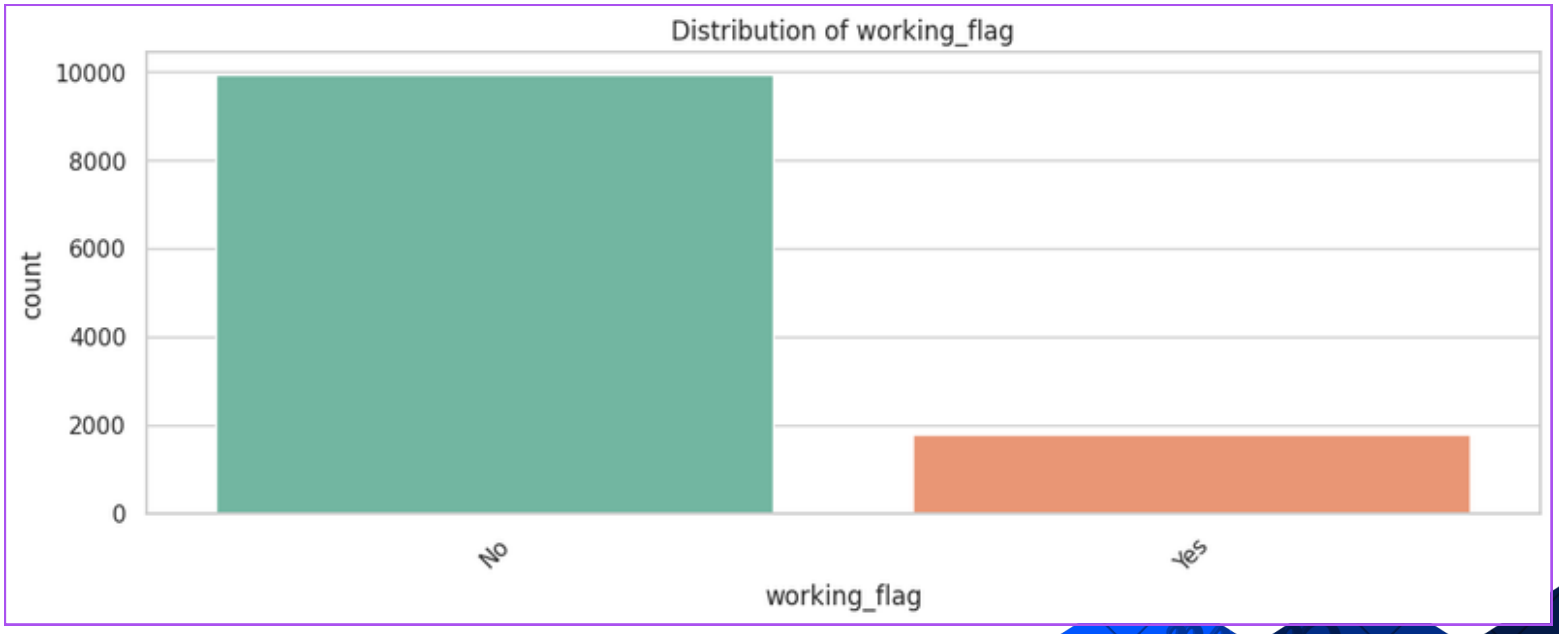







Distribution of preferred_location_type
count
2500
2000
1500
1000
500
0
Financial
Other
Medical
Game
Social media
Entertainment
Tour and Travel
Movie
OTT
Beach
Historical site
Big Cities
Trekking
Hill Stations
preferred_location_type


Distribution of following_company_page
count
8000
7000
6000
5000
4000
3000
2000
1000
0
Yes
No
following_company_page


Distribution of working_flag
count
10000
8000
6000
4000
2000
0
No
Yes
working_flag

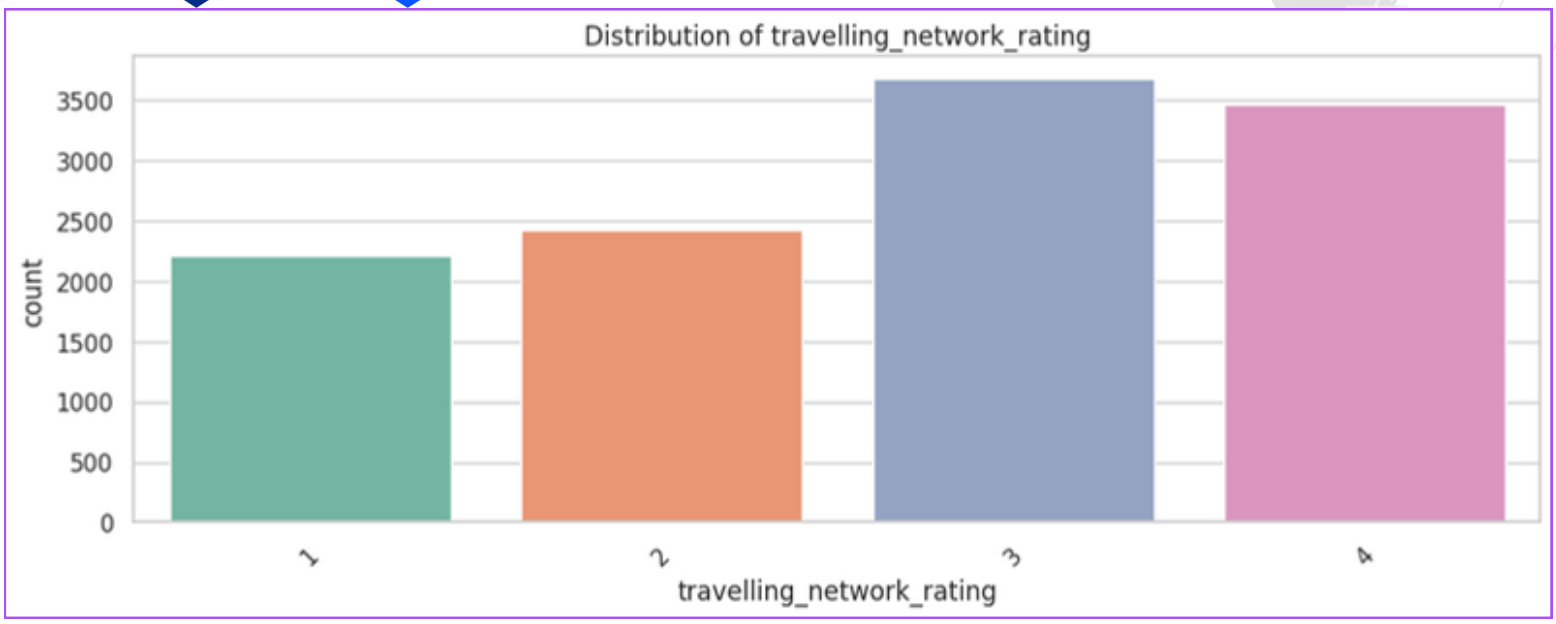


Figure 1.2: Univariate analysis of Categorical columns

Insights:

- A significant majority of individuals within the dataset have not made a purchase of the product.
- The primary choice for accessing the platform is through mobile devices.
- The most favored destination types among users are "Financial" and "Beach."
- A substantial proportion of users do not follow the company's page, and a similar majority falls within the non-working category.
- The majority of users possess a traveling network rating of 3.
- The dataset contains a larger representation of adult users in comparison to non-adults.

b) Bivariate analysis

Bivariate analysis is a statistical analysis technique that focuses on examining the relationship or association between two variables in a dataset.

Below is Correlation heat map for numeric features

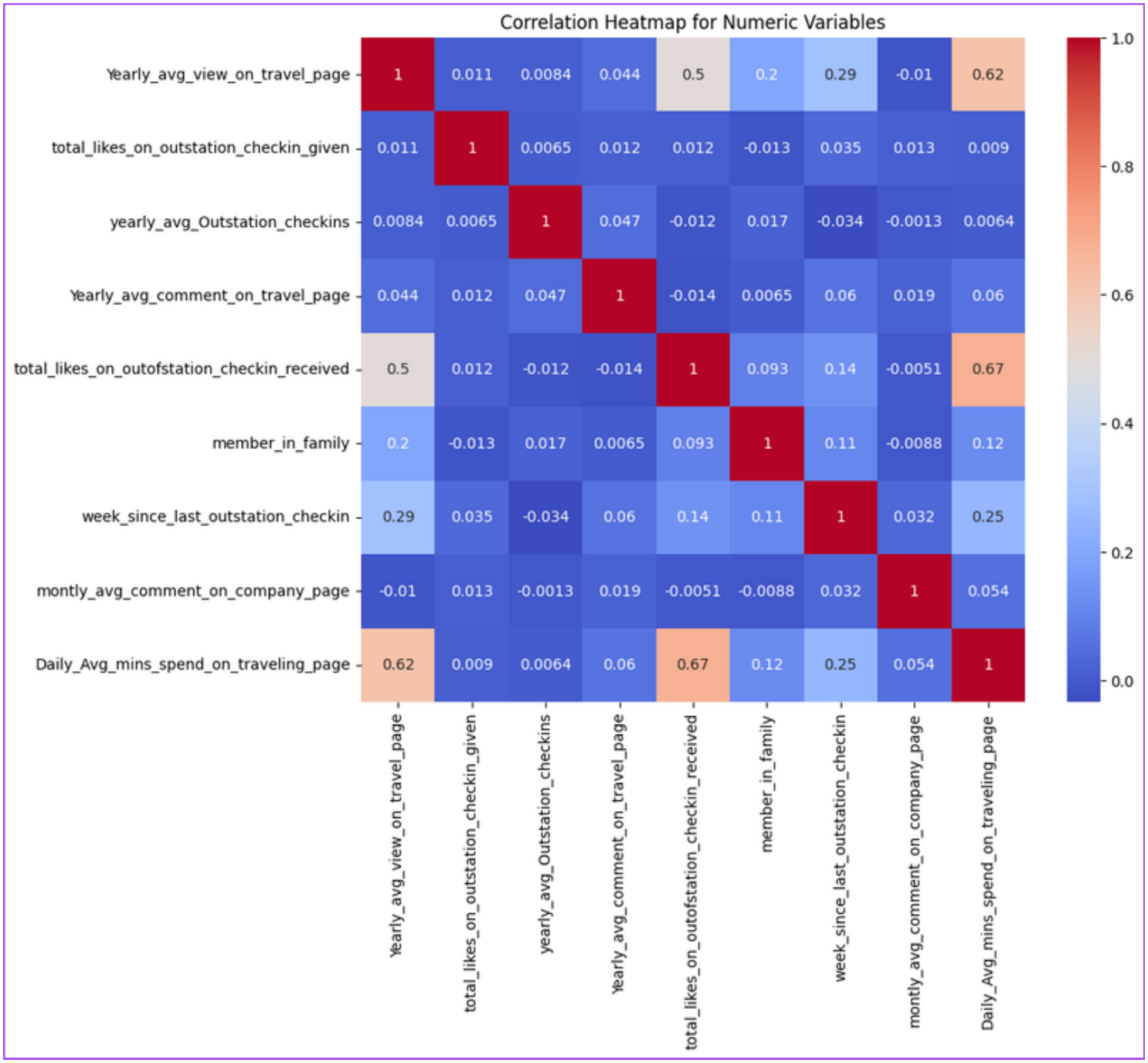


Figure 1.3: Heat map of Numerical columns

From the above graph here are few observations:

Strong positive correlations:

- Daily_Avg_mins_spend_on_traveling_page vs. total_likes_on_outofstation_checkin_received (0.67)
- Yearly_avg_view_on_travel_page vs. Daily_Avg_mins_spend_on_traveling_page (0.62)

Weak positive correlations:

- week_since_last_outstation_checkin vs. montly_avg_comment_on_company_page (0.03)
- Yearly_avg_view_on_travel_page vs. Yearly_avg_comment_on_travel_page (0.04)
- total_likes_on_outstation_checkin_given vs. total_likes_on_outofstation_checkin_received (0.01)
- member_in_family vs. week_since_last_outstation_checkin (0.11)

Negative correlations:

- total_likes_on_outstation_checkin_given vs. yearly_avg_Outstation_checkins (-0.01)
- total_likes_on_outofstation_checkin_received vs. Yearly_avg_comment_on_travel_page (-0.01)

Now lets look at Pairplot for significant features

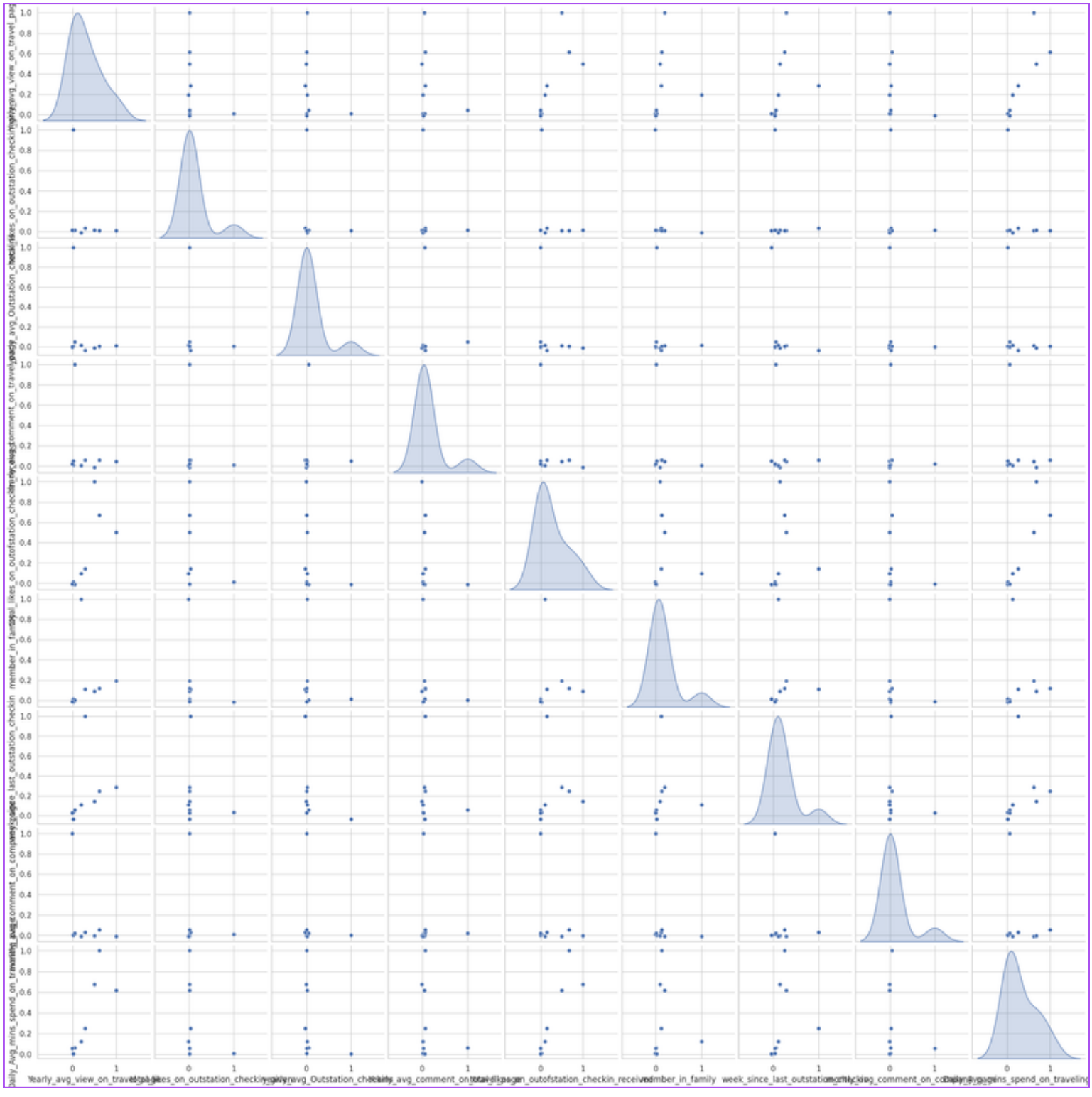
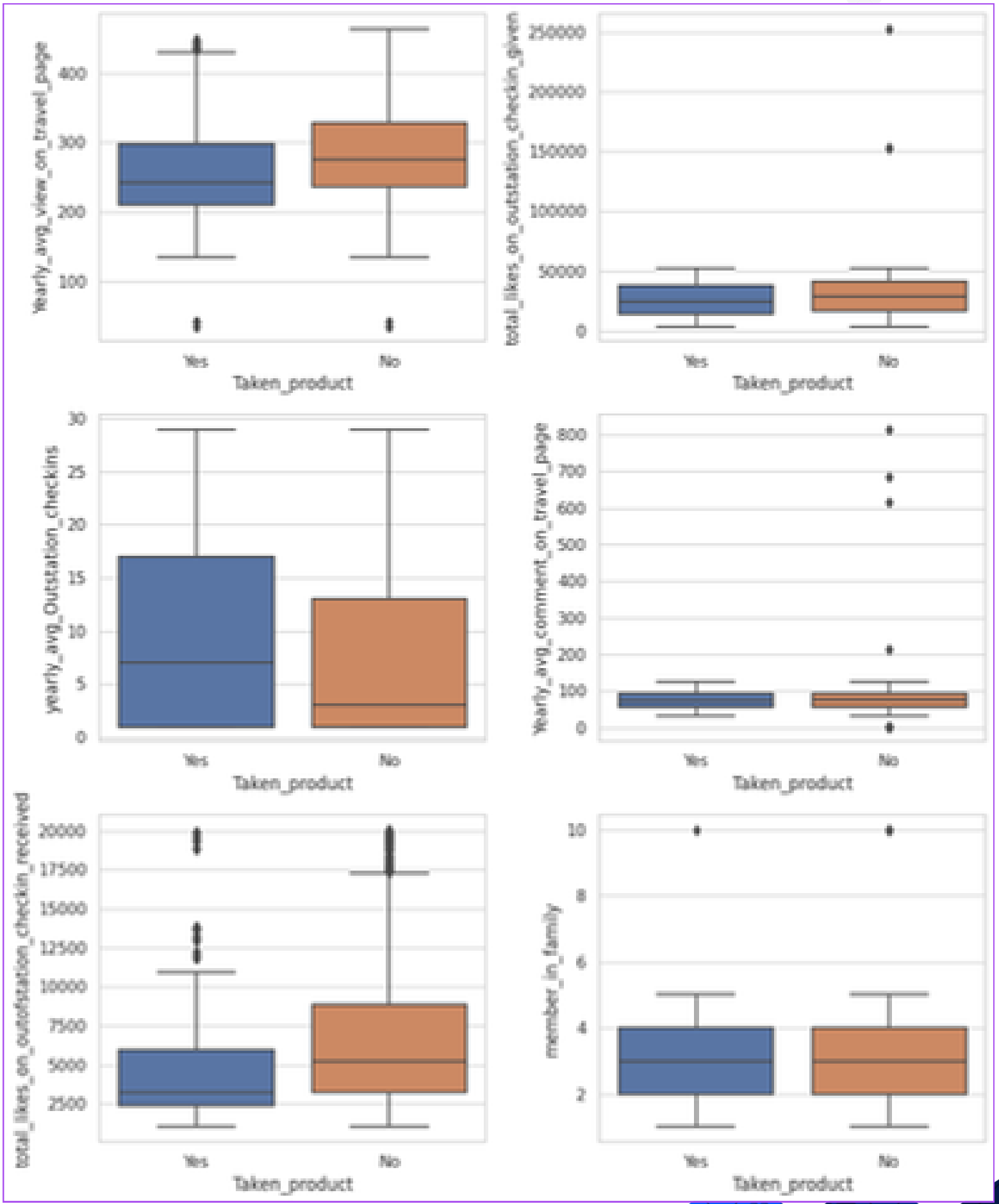
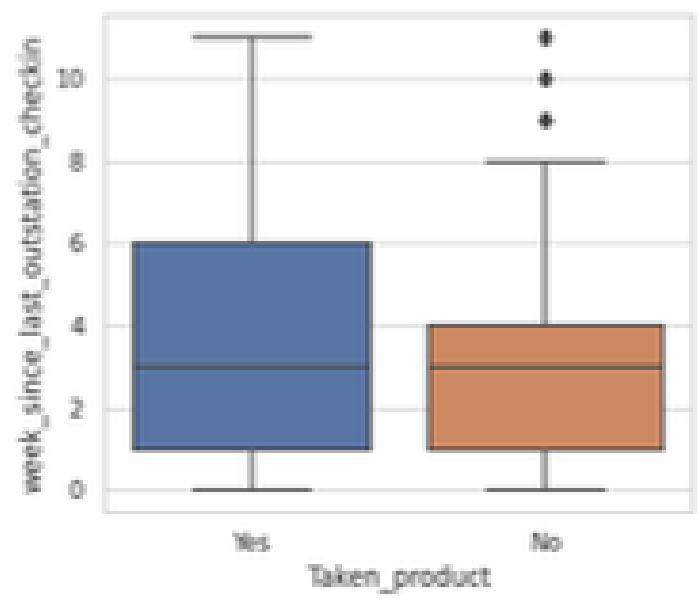
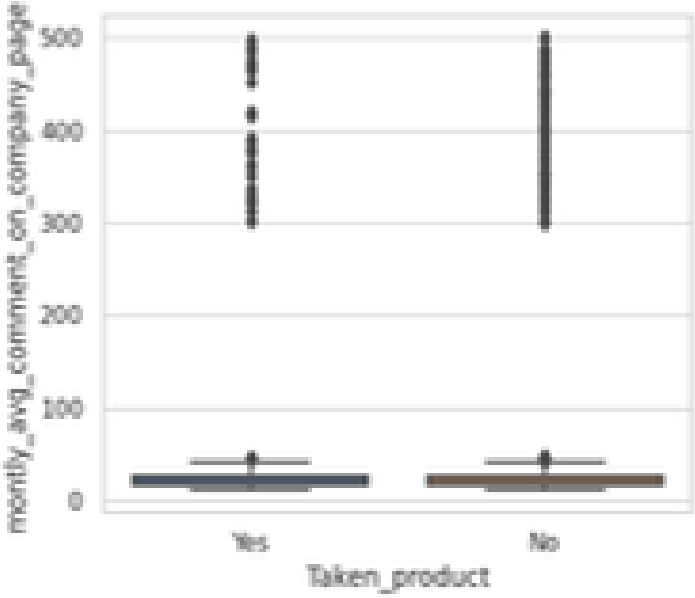


Figure 1.4: Pair Plot

Boxplot for continuous variables vs 'Product Taken'.







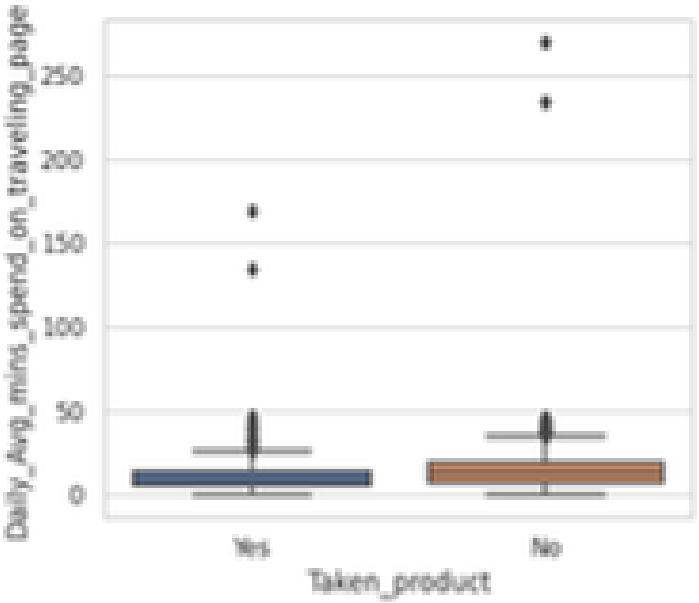


Figure 1.5: Continuous variables vs Product_taken Boxplot

Based on the graphs above, the probability of taking product remains relatively constant across different variables, with the exception of ‘total_likes_on_outstation_checkin_given ‘ and 'week_since_last_outstation_checkin'.

c) Removal of unwanted variables

Within the present dataset, it is discernible that all variables, with the exception of the 'UserID' attribute, contribute unique and distinct information that holds value for constructing an optimally predictive model. Consequently, we have excluded the 'UserID' column from our analysis, utilizing the remaining dataset in its entirety for modeling purposes.

```
RangeIndex: 11760 entries, 0 to 11759
Data columns (total 16 columns):
 #   Column
---  ------
 0   Taken_product
 1   Yearly_avg_view_on_travel_page
 2   preferred_device
 3   total_likes_on_outstation_checkin_given
 4   yearly_avg_Outstation_checkins
 5   member_in_family
 6   preferred_location_type
 7   Yearly_avg_comment_on_travel_page
 8   total_likes_on_outofstation_checkin_received
 9   week_since_last_outstation_checkin
 10  following_company_page
 11  montly_avg_comment_on_company_page
 12  working_flag
 13  travelling_network_rating
 14  Adult_flag
 15  Daily_Avg_mins_spend_on_traveling_page
```

Figure 1.6: Columns after removal of 'UserID'

d) Missing Value Treatment

Lets find out how many missing/null values present in dataset

```
Total number of Null values present: 1430
```

```
Percentage of Null values present: 0.76%
```

Table 1.15: Missing values in dataset

To address missing or null values, we will utilize imputation techniques.

Some of the popular imputation techniques are below:

- Mean, Median, or Mode Imputation
- Linear Interpolation
- Polynomial Interpolation
- K-Nearest Neighbors (KNN) Imputation

Out of the techniques mentioned above, we opted to use the mode imputation for treating missing or null values in our model. This is due to presence of non numeric columns and small quantity of missing values across columns.

```
Total number of Null values after imputation: 0
```

Table 1.16: Missing values in dataset after imputation

Here are some rows after imputation of data.

Taken_product	Yearly_avg_view_on_travel_page	preferred_device	total_likes_on_outstation_checkin_given	yearly_avg_Outstation_checkins
Yes	307.00	Mobile	38570.00	1.00
No	367.00	Mobile	9765.00	1.00
Yes	277.00	Mobile	48055.00	1.00
No	247.00	Mobile	48720.00	1.00
No	202.00	Mobile	20685.00	1.00

Table 1.17: Data after imputation

e) Outlier treatment

Let's examine the outliers that exist in the various columns.

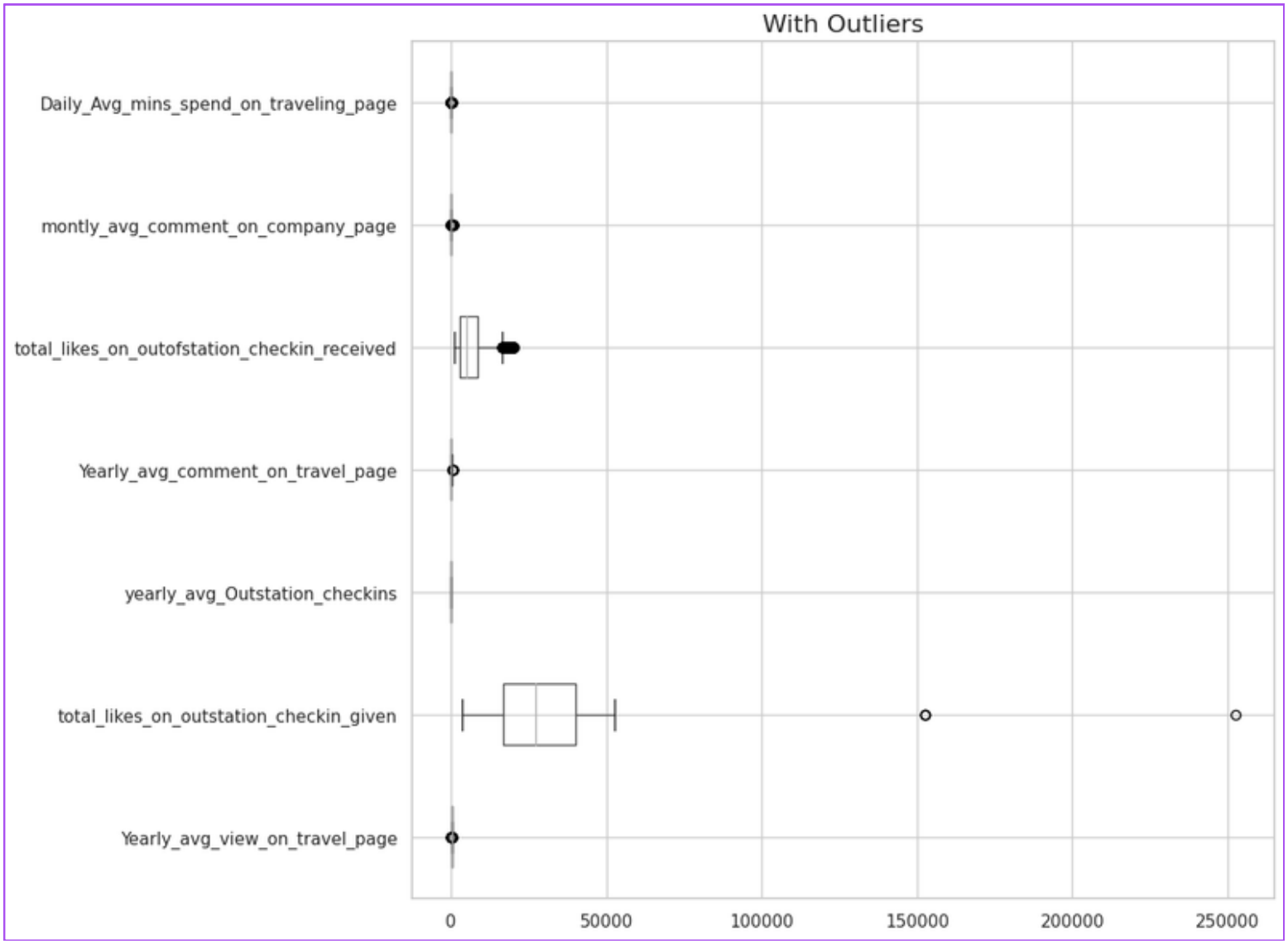


Figure 1.7: Outliers present in columns

Outlier above are identified using Inter-Quartile Rule (IQR). The IQR method defines outliers as data points that fall below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR, where Q1 is the first quartile (25 percentile) and Q3 is the third quartile (75 percentile).

In order to handle outliers, we are performing winsorizing. This approach offers multiple benefits, as opposed to merely dropping the outliers.

- By replacing the extreme values with quartile values (e.g., the upper quartile, Q3), we can mitigate the impact of outliers on your analysis.
- Winsorizing retains the overall shape of the data distribution.
- Winsorizing enhances the robustness of statistical tests and models.
- In some cases, winsorizing can lead to better model performance.

After outliers treatment.

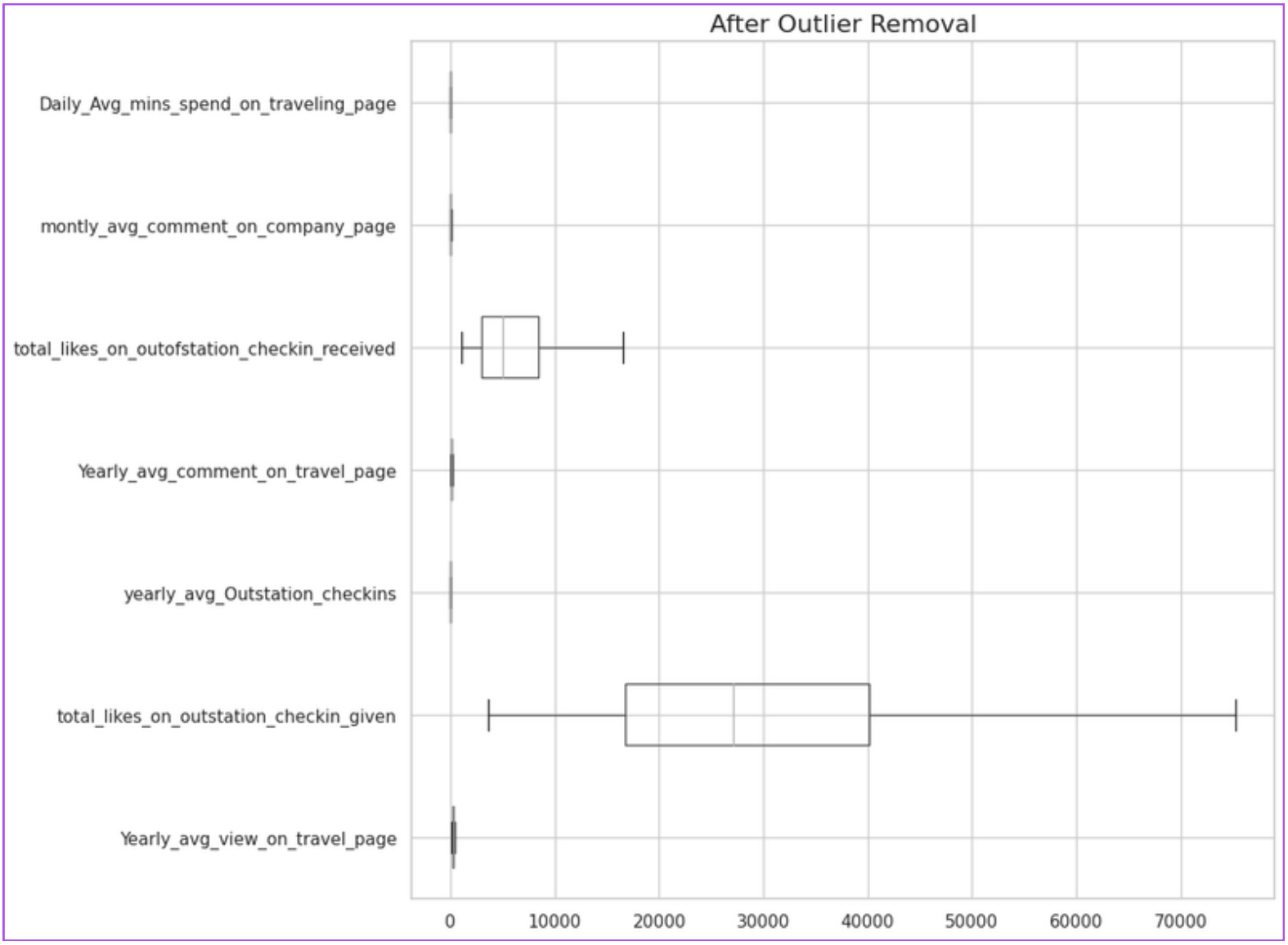


Figure 1.8: Box plot after outlier treatment

f) Variable transformation

We have performed label encoder to convert categorical columns to numerical values

Here is top few rows of data after encoding.

Taken_product	preferred_device	member_in_family	preferred_location_type	following_company_page	working_flag	travelling_network_rating	Adult_flag
1	1	1	3	1	0	0	0
0	1	0	3	0	1	3	1
1	1	1	10	1	0	1	0
0	1	3	3	1	0	2	0
0	1	0	7	0	0	3	1

Table 1.18: Top rows after encoding

Later we have scaled the data using StandardScaler to standardize the features of dataset. It scales your data so that it has a mean of 0 and a standard deviation of 1.

Here is top few rows of data after scaling.

	0	1	2	3	4	5	6	7
0	2.28	0.32	-0.89	-0.44	1.60	-0.43	-1.58	-1.15
1	-0.44	0.32	-1.87	-0.44	-0.62	2.35	1.19	0.87
2	2.28	0.32	-0.89	1.39	1.60	-0.43	-0.66	-1.15
3	-0.44	0.32	1.05	-0.44	1.60	-0.43	0.27	-1.15
4	-0.44	0.32	-1.87	0.60	-0.62	-0.43	1.19	0.87

Table 1.19: Top rows after scaling

g) Addition of new variables

The incorporation of additional variables is not a prerequisite; however, post-clustering, it may become necessary to introduce a cluster column to the dataset.

4. Business insights from EDA

a) Is the data unbalanced? If so, what can be done? Please explain in the context of the business

The data is indeed unbalanced, as evidenced by the class distributions for the "Taken_product" feature, where "No" has 9864 instances, and "Yes" has 1896 instances. This class imbalance can have implications for the analysis and model performance.

In the context of the business, addressing this class imbalance is important for several reasons:

- Biased Model: Class imbalance can lead to a biased machine learning model. In this case, the model may perform better on the majority class (e.g., "No" for "Taken_product") but poorly on the minority class ("Yes"). This is problematic if the business wants to make predictions for both classes with equal accuracy.

- Decision-Making: If the business's decisions are based on the model's predictions, an imbalanced dataset can lead to suboptimal decisions. For example, in this scenario, not identifying customers who might be interested in a product ("Yes" for "Taken_product") could result in missed revenue opportunities.
- Loss of Information: Class imbalance may result in the model having limited information to learn from the minority class, potentially leading to poor generalization for those instances.

Here are some strategies that can be considered to address the class imbalance:

- Resampling: One common approach is to either oversample the minority class (create more instances of "Yes") or undersample the majority class (remove some instances of "No"). This helps balance the class distribution. Techniques like SMOTE (Synthetic Minority Over-sampling Technique) can be used to create synthetic samples for the minority class.

- Weighted Loss: In many machine learning algorithms, you can assign different weights to classes. You can assign a higher weight to the minority class to penalize misclassifications more.
- Collect More Data: If possible, collect more data for the minority class to balance the distribution.
- Evaluate Metrics Carefully: When assessing model performance, use metrics beyond accuracy, such as precision, recall, F1-score, or area under the ROC curve (AUC-ROC). These metrics provide a more comprehensive view of how the model performs for both classes.
- Business Process Adjustment: Consider modifying business processes to address the class imbalance. For instance, marketing strategies could be tailored to reach out to customers in the minority class ("Yes" for "Taken_product").
- Ensemble Methods: Ensemble methods like Random Forest and Gradient Boosting can handle class imbalance more effectively than some other algorithms. These models can be fine-tuned to give better performance on the minority class.

b) Any business insights using clustering.

We have employed the K-means clustering technique for the given dataset. In our analysis, we conducted several assessments, including the generation of an elbow plot, the calculation of the Within Sum of Squares (WSS), and the determination of the silhouette score. These measures were instrumental in elucidating the optimal number of clusters for the dataset.

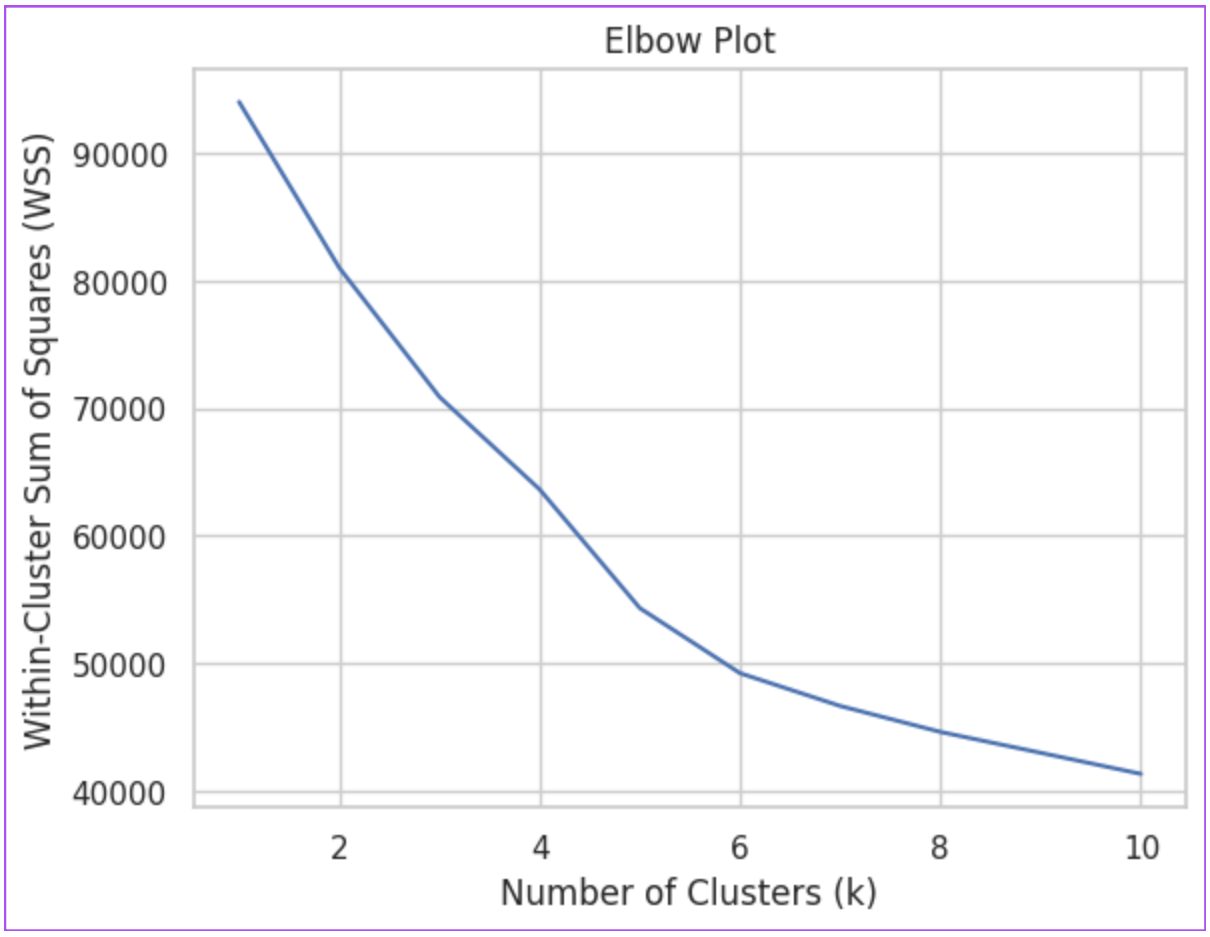


Figure 1.9: Elbow plot

```
The WSS value for Cluster 0 is 94080.00000000012
The WSS value for Cluster 1 is 81055.45162091937
The WSS value for Cluster 2 is 70914.23006426313
The WSS value for Cluster 3 is 63649.49794403698
The WSS value for Cluster 4 is 54328.31675713566
The WSS value for Cluster 5 is 49218.962904236876
The WSS value for Cluster 6 is 46646.15261366091
The WSS value for Cluster 7 is 44609.30044556396
The WSS value for Cluster 8 is 42971.29089175603
The WSS value for Cluster 9 is 41314.652689581075
```

Table 1.20: WSS score

From the elbow plot and WSS score it looks like elbow point is around cluster 5.

```
for k = 2, silhouette score is = 0.1839515024857806 8
for k = 3, silhouette score is = 0.2606712183431087
for k = 4, silhouette score is = 0.1944616913340177 8
for k = 5, silhouette score is = 0.2154436929326154
for k = 6, silhouette score is = 0.2229492703908036 7
for k = 7, silhouette score is = 0.1935783915241741 4
for k = 8, silhouette score is = 0.1883179562810776
for k = 9, silhouette score is = 0.2096023314233905 5
for k = 10, silhouette score is = 0.216879497870042
```

Table 1.21: Silhouette score

From silhouette score we can see the optimum number of cluster is 3. We will proceed with 3 clusters only.

Now we have divided data into clusters and same can be seen in below table

working_flag	travelling_network_rating	Adult_flag	Daily_Avg_mins_spend_on_traveling_page	Clus_kmeans3
No	1	0	8.00	2
Yes	4	1	10.00	1
No	2	0	7.00	2
No	3	0	8.00	1
No	4	1	6.00	1

Table 1.22: Top few rows of dataset after clustering

Insights from clustering:

Cluster 0 :

- Cluster Size**:** Cluster 0 contains 1108 data points.
- Taken Product (No vs. Yes): Within this cluster, 832 users have not taken the product ("No"), while 276 users have taken the product ("Yes").

Business Insights:

- Cluster 0 represents a group of customers with mixed responses to the product. A significant portion (about 25%) of users within this cluster have shown interest in the product ("Yes"), while the majority have not taken the product.

- This cluster presents an opportunity for targeted marketing or engagement strategies to encourage more users to take the product. Understanding the common characteristics or behaviors of the users who have taken the product can help tailor marketing efforts.

Cluster 1 :

- Cluster Size: Cluster 1 is the largest cluster and contains 9032 data points.
- Taken Product (No vs. Yes)**:** In this cluster, all users fall into the "No" category

Business Insights:

- Cluster 1 appears to represent a group of users who have not shown interest in taking the product. This cluster's size indicates that it might be challenging to convert users within this group into product adopters.
- The business may want to further analyze the characteristics and behaviors of users in this cluster to understand why they are not taking the product. This information can help refine marketing or product improvement strategies.

Cluster 2 :

- Cluster Size: Cluster 2 contains 1620 data points.
- Taken Product (No vs. Yes): Within this cluster, all 1620 users have taken the product ("Yes").

Business Insights:

- Cluster 2 represents a group of users who have shown a strong interest in and have taken the product. These users are already customers, which is a positive sign.
- For this cluster, the business focus could be on customer retention, cross-selling, or upselling additional products or services to this engaged customer group.

- This cluster presents an opportunity for targeted marketing or engagement strategies to encourage more users to take the product. Understanding the common characteristics or behaviors of the users who have taken the product can help tailor marketing efforts.

c) Any other business insights

Overall, the clustering results provide valuable insights into different segments of your customer base. By understanding the characteristics and behaviors of users within each cluster, the business can tailor its marketing, product development, and customer engagement strategies to optimize customer acquisition, retention, and revenue generation. The specific actions to take would depend on the business objectives and the context of the data.

PROJECT NOTES - 2

1). Model building and interpretation

In order to conduct our analysis, we have classified the data based on the user's preferred device into two categories: mobile and laptop.

We have tried variety of classification algorithms to see which one performs best for this problem. The algorithms included are:

- Logistic Regression: Logistic Regression is a statistical and machine learning model used for binary classification. The logistic regression model is expressed as follows:

$$P(Y = 1|X) = 1 / (1 + e^{-(\beta 0 + \beta 1X1 + \beta 2X2 + ... + \beta nXn)})$$

Where:

- P(Y = 1|X) is the probability of the target variable Y being 1 given the input features X.
- e is the base of the natural logarithm.
- β0, β1, β2, ..., βn are the coefficients or weights associated with each feature X0, X1, X2, ..., Xn.

- LDA: LDA stands for Linear Discriminant Analysis, which is a dimensionality reduction and classification technique in machine learning and statistics. LDA is often used in the context of supervised learning for classification and feature extraction
- Decision Trees: A Decision Tree is a tree-like structure, where each internal node represents a feature or attribute, each branch represents a decision or rule, and each leaf node represents an outcome or class label. The path from the root node to a leaf node represents a sequence of decisions based on the feature values.
- k-Nearest Neighbors (KNN): -Nearest Neighbors (KNN) is a simple yet effective machine learning algorithm used for both classification and regression tasks. When a prediction is required, KNN searches the dataset for the k-nearest data points (or neighbors) to the input data point and makes predictions based on their properties.

- Naive Bayes: These models are based on Bayes' theorem, which is a fundamental concept in probability theory. The "naive" part of the name comes from the simplifying assumption that features are conditionally independent, given the class label. This assumption makes the model computationally efficient and easy to implement, but it may not always hold true in real-world data.

We have encoded the categorical columns to numeric values using LabelEncoder. Top few rows are below.

Taken_product	Yearly_avg_view_on_travel_page	preferred_device	total_likes_on_outstation_checkin_given	yearly_avg_Outstation_checkins
1	172	1	5545	0
0	232	1	952	0
1	142	1	7183	0
0	112	1	7300	0
0	67	1	2755	0

Table 1.23: Top few rows of dataset after Imputation

We have labelled ‘mobile’ devices as 1 and ‘laptop’ as 0. First we will make models for ‘Mobile’ followed by ‘Laptop’.

We have divided our data between mobile and laptop users and then partitioned data in 70:30, where test data is 30%.

```
For training data of Mobile:
The number of rows (observations) is 745
The number of columns (variables) is 14

For test data of Mobile:
The number of rows (observations) is 319
The number of columns (variables) is 14

For training data of Laptop:
The number of rows (observations) is 775
The number of columns (variables) is 14

For test data of Laptop:
The number of rows (observations) is 333
The number of columns (variables) is 14
```

Now we will check this data against various models and check model metrics for test data

Models for Mobile Data

- Logistic Regression

Classification Report:	precision	recall	f1-score	support
0	0.86	1.00	0.92	2743
1	0.61	0.03	0.06	453
accuracy			0.86	3196
macro avg	0.74	0.51	0.49	3196
weighted avg	0.83	0.86	0.80	3196

Table 1.24: Classification report for logistic regression

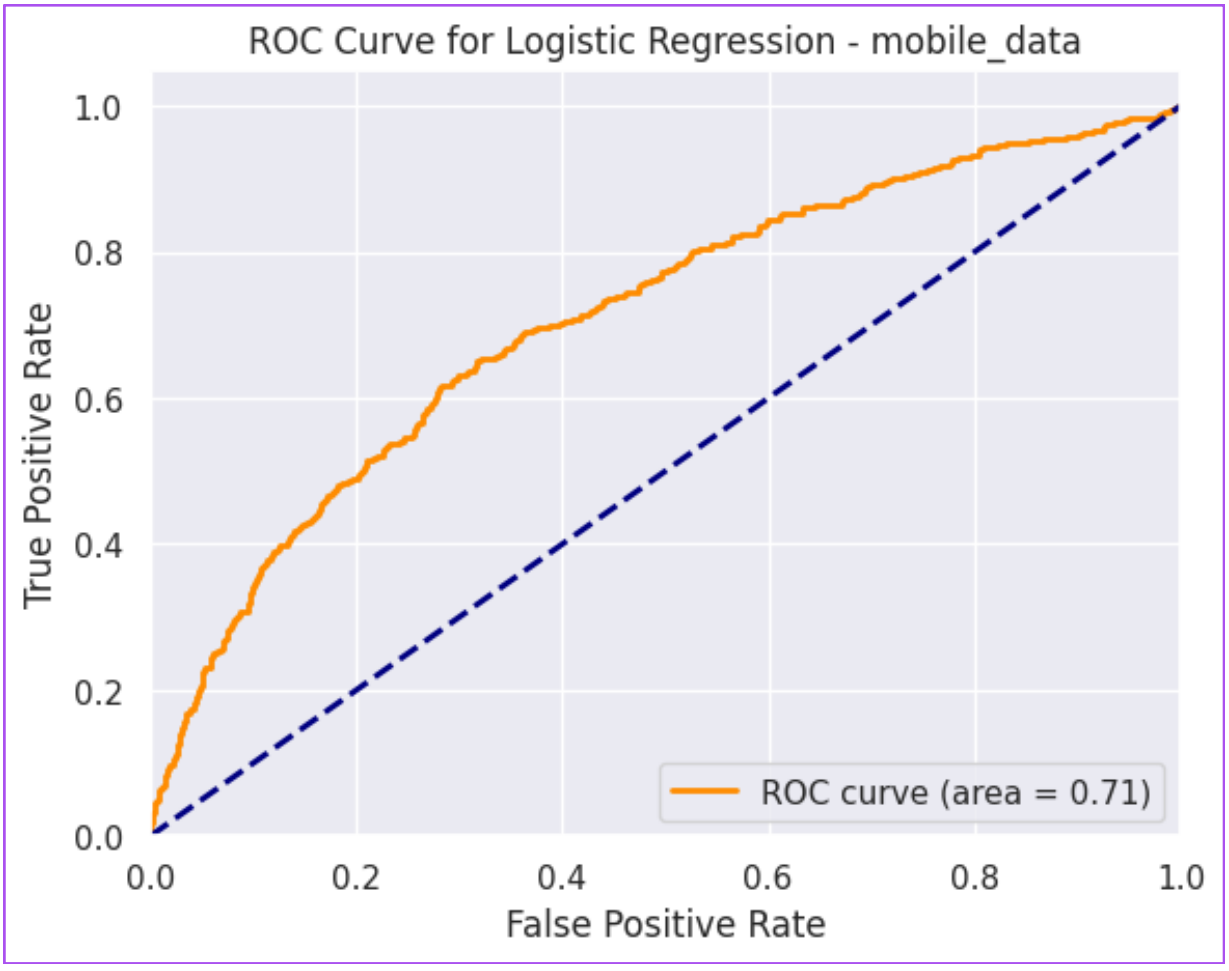


Figure 1.10: ROC curve for logistic regression

Insights:

- Confusion Matrix:
  - True Positives (TP): 14
  - True Negatives (TN): 2734
  - False Positives (FP): 9
  - False Negatives (FN): 439
- RMSE (Root Mean Squared Error): The RMSE value is approximately 0.3744, which indicates the average error in the predicted values.
- AUC (Area Under the Curve): The AUC value is approximately 0.7102. AUC is a measure of the model's ability to distinguish between positive and negative classes. A value of 0.5 represents random guessing, so an AUC of 0.7102 suggests a moderate level of discrimination.
- Accuracy: The accuracy for the test set is approximately 0.8598, which means that the model correctly classifies about 85.98% of the test instances.

- Classification Report:
  - Precision for class 0 (No product taken) is 0.86, indicating that 86% of the predicted instances for class 0 were correct.
  - Recall for class 0 is 1.00, indicating that 100% of the actual instances for class 0 were correctly predicted.
  - F1-score for class 0 is 0.92, which is the harmonic mean of precision and recall.
  - Precision for class 1 (Product taken) is 0.61, indicating that 61% of the predicted instances for class 1 were correct.
  - Recall for class 1 is 0.03, indicating that only 3% of the actual instances for class 1 were correctly predicted.
  - F1-score for class 1 is 0.06, which is low due to the low recall.

These metrics show that the model performs well in predicting class 0 (No product taken) but poorly in predicting class 1 (Product taken).

- LDA

Classification Report:

	precision	recall	f1-score	support
0	0.89	0.98	0.93	2743
1	0.64	0.25	0.36	453
accuracy			0.87	3196
macro avg	0.76	0.62	0.65	3196
weighted avg	0.85	0.87	0.85	3196

Table 1.25: Classification report for LDA

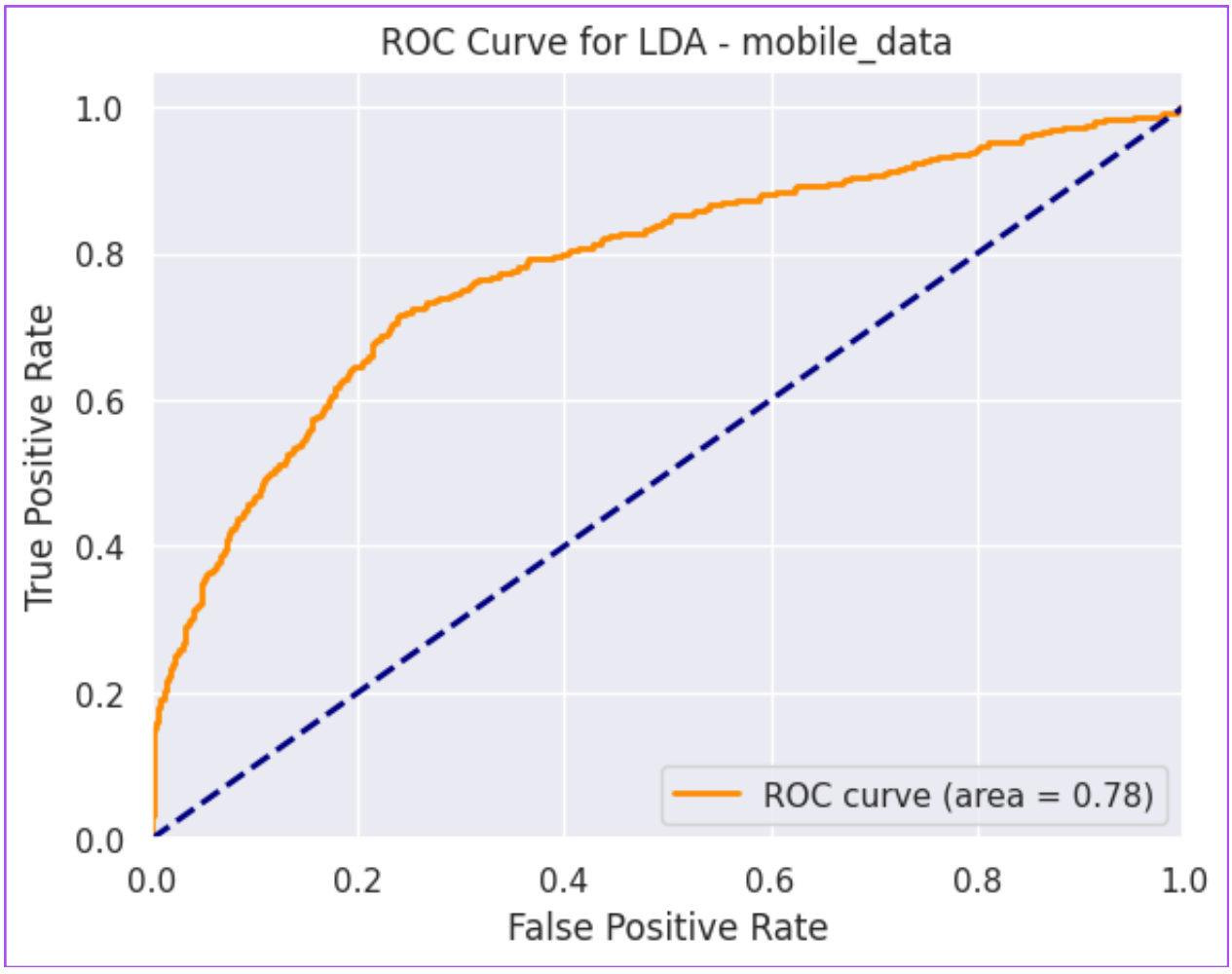


Figure 1.11: ROC curve for LDA

Insights:

- Confusion Matrix:
  - True Positives (TP): 115
  - True Negatives (TN): 2678
  - False Positives (FP): 65
  - False Negatives (FN): 338
- RMSE (Root Mean Squared Error): The RMSE value is approximately 0.3551, indicating the average error in the predicted values.
- AUC (Area Under the Curve): The AUC value is approximately 0.7825. AUC of 0.7825 suggests a good level of discrimination.
- Accuracy: The accuracy for the test set is approximately 0.8739, which means that the model correctly classifies about 87.39% of the test instances. The accuracy is relatively high.

- Classification Report:
  - Precision for class 0 (No product taken) is 0.89, indicating that 89% of the predicted instances for class 0 were correct.
  - Recall for class 0 is 0.98, indicating that 98% of the actual instances for class 0 were correctly predicted.
  - F1-score for class 0 is 0.93, which is high and indicates a good balance between precision and recall.
  - Precision for class 1 (Product taken) is 0.64, indicating that 64% of the predicted instances for class 1 were correct.
  - Recall for class 1 is 0.25, indicating that 25% of the actual instances for class 1 were correctly predicted.
  - F1-score for class 1 is 0.36, which is moderate.

These metrics show that the LDA model performs well in predicting class 0 (No product taken). However, it struggles to predict instances of class 1 (Product taken), with lower precision, recall, and F1-score.

- Decision Tree

Classification Report:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	2743
1	0.88	0.91	0.89	453
accuracy			0.97	3196
macro avg	0.93	0.94	0.94	3196
weighted avg	0.97	0.97	0.97	3196

Table 1.26: Classification report for Decision Tree

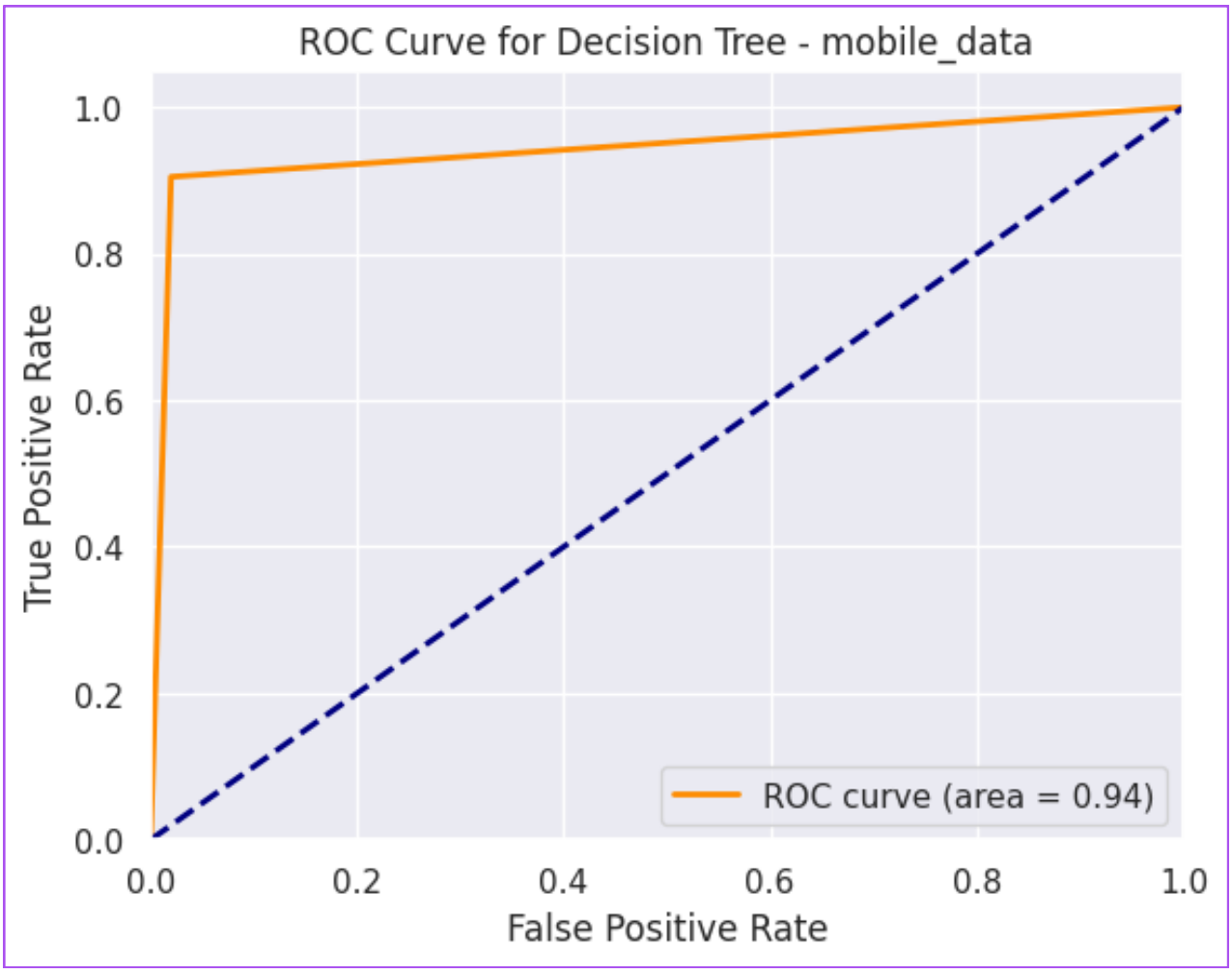


Figure 1.12: ROC curve for Decision tree

Insights:

- Confusion Matrix:
  - True Positives (TP): 410
  - True Negatives (TN): 2689
  - False Positives (FP): 54
  - False Negatives (FN): 43
- RMSE (Root Mean Squared Error): The RMSE value is approximately 0.1742, indicating a low average error in the predicted values.
- R-squared (R^2): The R-squared value is 0.7505. In this context, a high positive R-squared indicates that the model explains a significant portion of the variance in the data, which is generally good.
- AUC (Area Under the Curve): The AUC value is approximately 0.9427. AUC of 0.9427 suggests a high level of discrimination.
- Accuracy: The accuracy for the test set is approximately 0.9696, which means that the model correctly classifies about 96.96% of the test instances.

- Classification Report:
  - Precision for class 0 (No product taken) is 0.98, indicating that 98% of the predicted instances for class 0 were correct.
  - Recall for class 0 is 0.98, indicating that 98% of the actual instances for class 0 were correctly predicted.
  - F1-score for class 0 is 0.98, which is high and indicates an excellent balance between precision and recall for this class.
  - Precision for class 1 (Product taken) is 0.88, indicating that 88% of the predicted instances for class 1 were correct.
  - Recall for class 1 is 0.91, indicating that 91% of the actual instances for class 1 were correctly predicted.
  - F1-score for class 1 is 0.89, which is also high and indicates a good balance between precision and recall for this class.

These metrics show that the Decision Tree model performs exceptionally well, with high precision, recall, and F1-scores for both classes.

- KNN

Classification Report:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	2743
1	0.91	0.83	0.87	453
accuracy			0.96	3196
macro avg	0.94	0.91	0.92	3196
weighted avg	0.96	0.96	0.96	3196

Table 1.27: Classification report for KNN

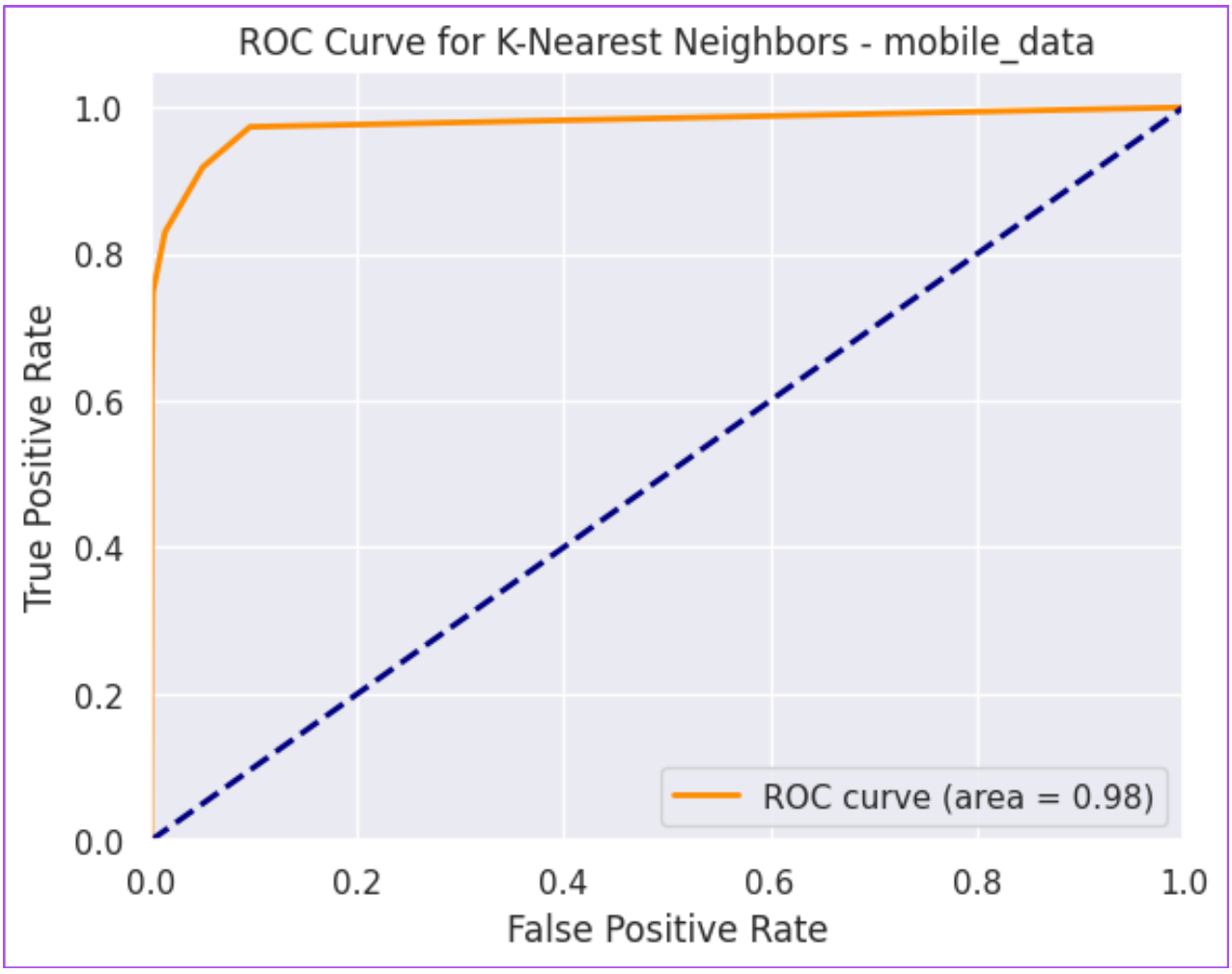


Figure 1.13: ROC curve for KNN

Insights:

- Confusion Matrix:
  - True Positives (TP): 376
  - True Negatives (TN): 2705
  - False Positives (FP): 38
  - False Negatives (FN): 77
- RMSE (Root Mean Squared Error): The RMSE value is approximately 0.1897, indicating a low average error in the predicted values.
- R-squared (R^2): The R-squared value is approximately 0.7042. In this context, a high positive R-squared indicates that the model explains a significant portion of the variance in the data, which is generally good.
- AUC (Area Under the Curve): The AUC value is approximately 0.9775. AUC of 0.9775 suggests a very high level of discrimination.
- Accuracy: The accuracy for the test set is approximately 0.9640, which means that the model correctly classifies about 96.40% of the test instances.

- Classification Report:
  - Precision for class 0 (No product taken) is 0.97, indicating that 97% of the predicted instances for class 0 were correct.
  - Recall for class 0 is 0.99, indicating that 99% of the actual instances for class 0 were correctly predicted.
  - F1-score for class 0 is 0.98, which is high and indicates an excellent balance between precision and recall for this class.
  - Precision for class 1 (Product taken) is 0.91, indicating that 91% of the predicted instances for class 1 were correct.
  - Recall for class 1 is 0.83, indicating that 83% of the actual instances for class 1 were correctly predicted.
  - F1-score for class 1 is 0.87, which is also high and indicates a good balance between precision and recall for this class.

These metrics show that the K-Nearest Neighbors (KNN) model performs exceptionally well, with high precision, recall, and F1-scores for both classes.

- Naive Bayes

Classification Report:

	precision	recall	f1-score	support
0	0.89	0.97	0.93	2743
1	0.62	0.30	0.41	453
accuracy			0.87	3196
macro avg	0.76	0.64	0.67	3196
weighted avg	0.85	0.87	0.86	3196

Table 1.28: Classification report for Naive Bayes

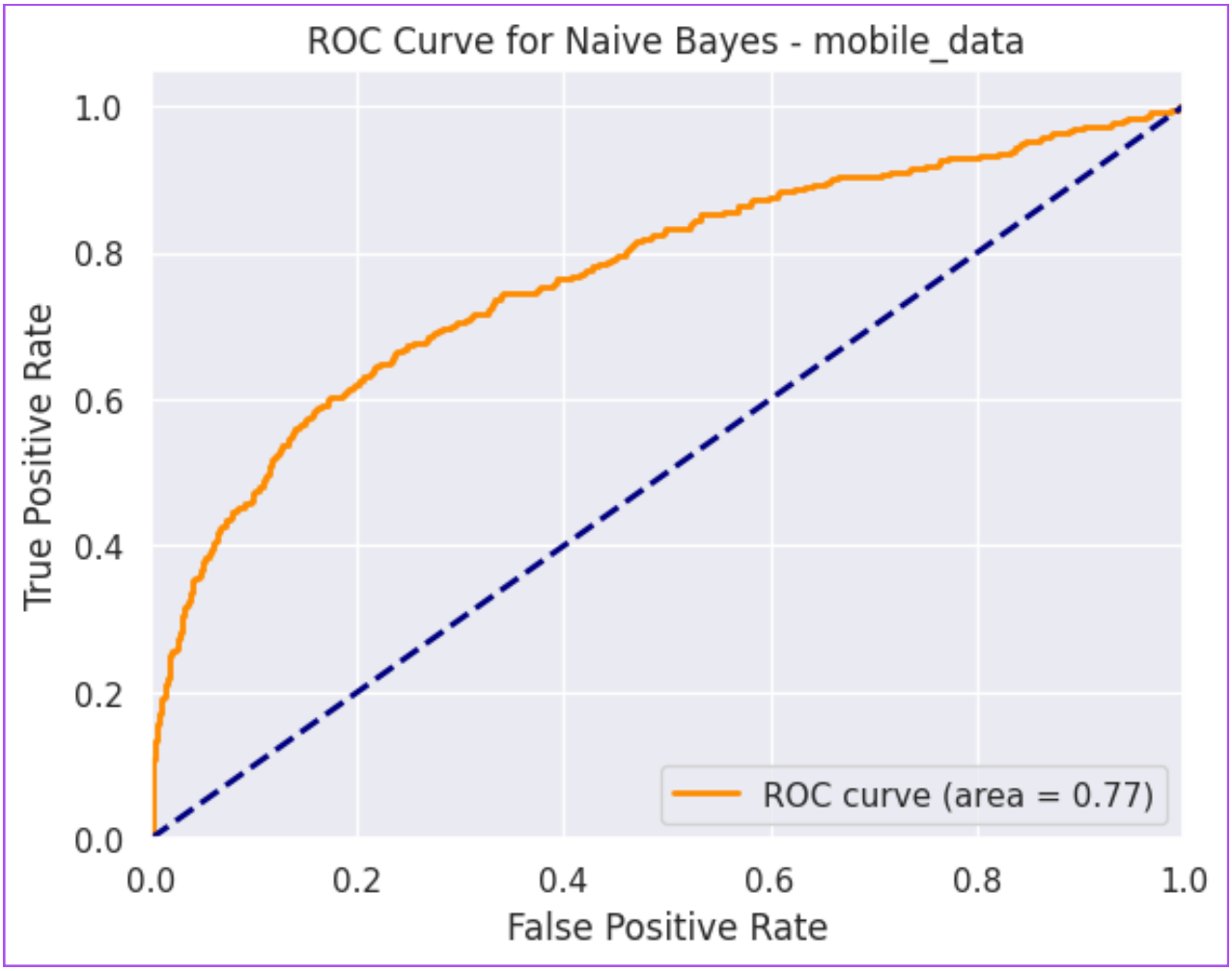


Figure 1.14: ROC curve for Naive Bayes

Insights:

- Confusion Matrix:
  - True Positives (TP): 137
  - True Negatives (TN): 2658
  - False Positives (FP): 85
  - False Negatives (FN): 316
- RMSE (Root Mean Squared Error): The RMSE value is approximately 0.3542, indicating the average error in the predicted values.
- AUC (Area Under the Curve): The AUC value is approximately 0.7702. AUC of 0.7702 suggests a moderate level of discrimination.
- Accuracy: The accuracy for the test set is approximately 0.8745, which means that the model correctly classifies about 87.45% of the test instances.

- Classification Report:
  - Precision for class 0 (No product taken) is 0.89, indicating that 89% of the predicted instances for class 0 were correct.
  - Recall for class 0 is 0.97, indicating that 97% of the actual instances for class 0 were correctly predicted.
  - F1-score for class 0 is 0.93, which is high and indicates a good balance between precision and recall for this class.
  - Precision for class 1 (Product taken) is 0.62, indicating that 62% of the predicted instances for class 1 were correct.
  - Recall for class 1 is 0.30, indicating that only 30% of the actual instances for class 1 were correctly predicted.
  - F1-score for class 1 is 0.41, which is moderate.

These metrics show that the Naive Bayes model performs well in predicting class 0 (No product taken) but poorly in predicting class 1 (Product taken).

Models for Laptop Data

- Logistic Regression

Classification Report:

	precision	recall	f1-score	support
0	0.83	0.90	0.86	258
1	0.52	0.37	0.43	75
accuracy			0.78	333
macro avg	0.68	0.64	0.65	333
weighted avg	0.76	0.78	0.77	333

Table 1.29: Classification report for Logistic Regression

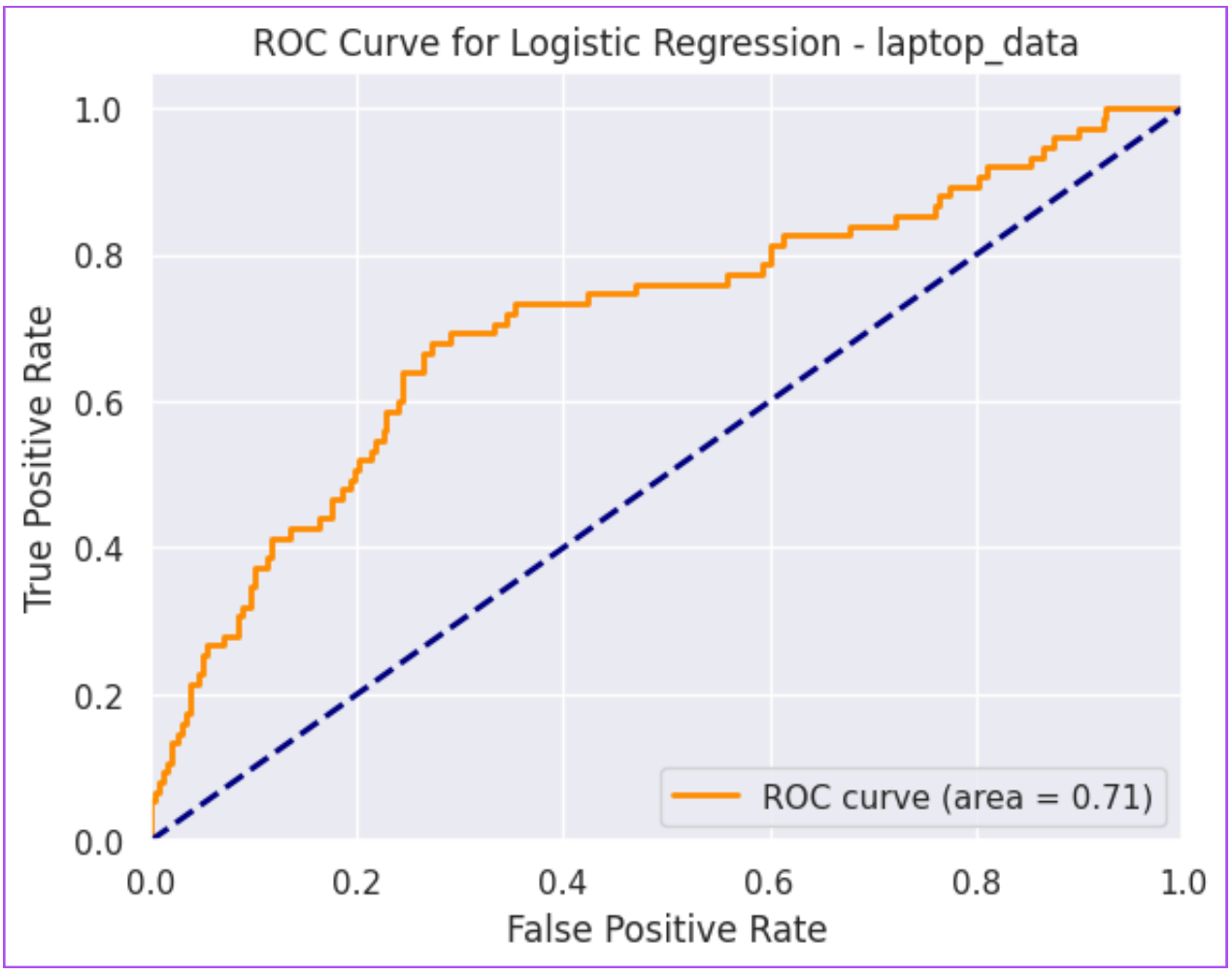


Figure 1.15: ROC curve for Logistic Regression

Insights:

- Confusion Matrix:
  - True Positives (TP): 28
  - True Negatives (TN): 232
  - False Positives (FP): 26
  - False Negatives (FN): 47
- RMSE (Root Mean Squared Error): The RMSE value is approximately 0.4682, indicating the average error in the predicted values.
- AUC (Area Under the Curve): The AUC value is approximately 0.7107.. AUC of 0.7107 suggests a moderate level of discrimination.
- Accuracy: The accuracy for the test set is approximately 0.7808, which means that the model correctly classifies about 78.08% of the test instances.

- Classification Report:
  - Precision for class 0 (No product taken) is 0.83, indicating that 83% of the predicted instances for class 0 were correct.
  - Recall for class 0 is 0.90, indicating that 90% of the actual instances for class 0 were correctly predicted.
  - F1-score for class 0 is 0.86, which is relatively high and indicates a good balance between precision and recall for this class.
  - Precision for class 1 (Product taken) is 0.52, indicating that 52% of the predicted instances for class 1 were correct.
  - Recall for class 1 is 0.37, indicating that only 37% of the actual instances for class 1 were correctly predicted.
  - F1-score for class 1 is 0.43, which is moderate.

These metrics show that the Logistic Regression model performs relatively well in predicting class 0 (No product taken) but less effectively in predicting class 1 (Product taken).

- LDA

Classification Report:

	precision	recall	f1-score	support
0	0.86	0.96	0.90	258
1	0.76	0.45	0.57	75
accuracy			0.84	333
macro avg	0.81	0.71	0.74	333
weighted avg	0.83	0.84	0.83	333

Table 1.30: Classification report for LDA

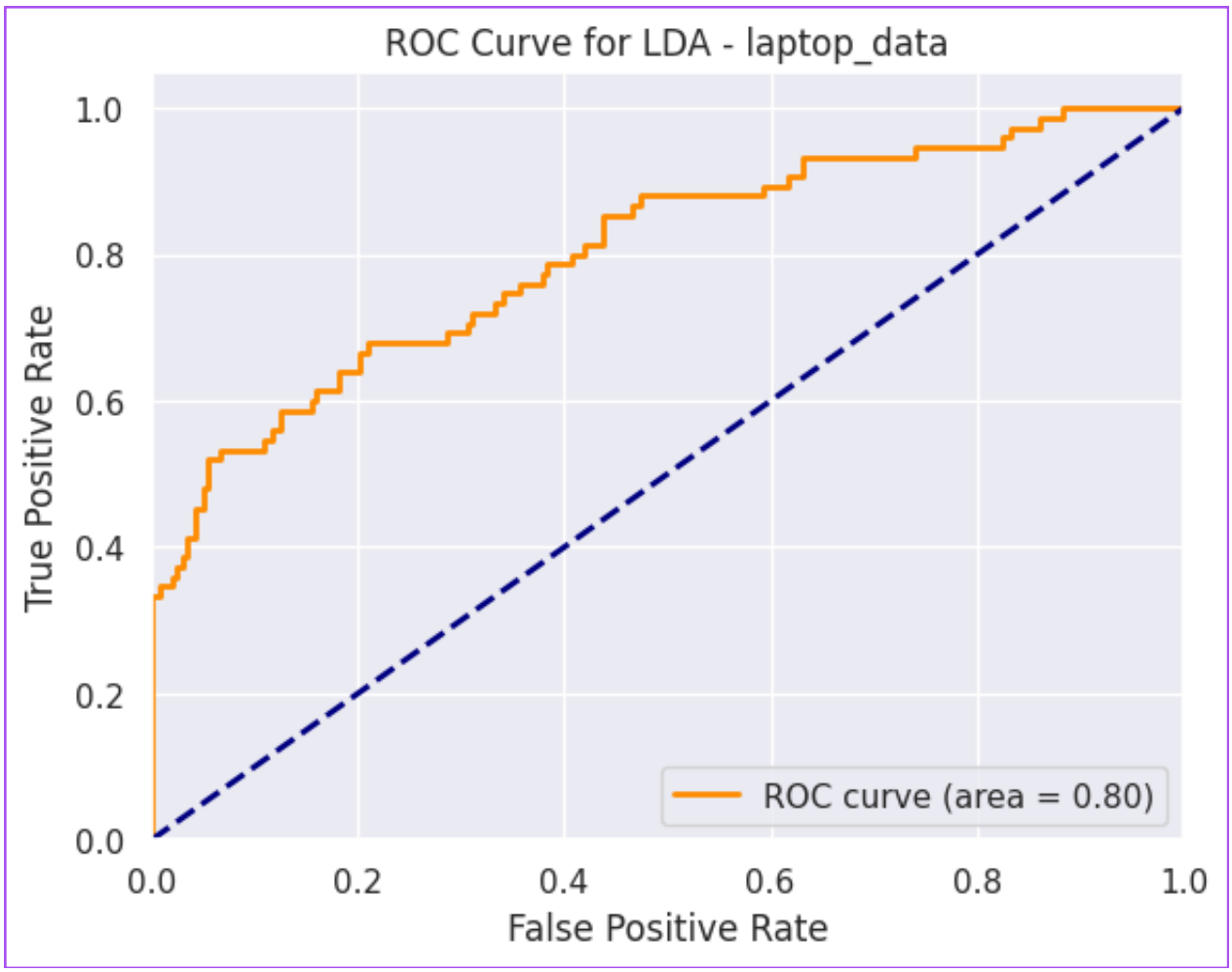


Figure 1.16: ROC curve for LDA

Insights:

- Confusion Matrix:
  - True Positives (TP): 34
  - True Negatives (TN): 247
  - False Positives (FP): 11
  - False Negatives (FN): 41
- RMSE (Root Mean Squared Error): The RMSE value is approximately 0.3952, indicating the average error in the predicted values.
- AUC (Area Under the Curve): The AUC value is approximately 0.8032. An AUC of 0.8032 suggests a good level of discrimination.
- Accuracy: The accuracy for the test set is approximately 0.8438, which means that the model correctly classifies about 84.38% of the test instances.

- Classification Report:
  - Precision for class 0 (No product taken) is 0.86, indicating that 86% of the predicted instances for class 0 were correct.
  - Recall for class 0 is 0.96, indicating that 96% of the actual instances for class 0 were correctly predicted.
  - F1-score for class 0 is 0.90, which is relatively high and indicates a good balance between precision and recall for this class.
  - Precision for class 1 (Product taken) is 0.76, indicating that 76% of the predicted instances for class 1 were correct.
  - Recall for class 1 is 0.45, indicating that 45% of the actual instances for class 1 were correctly predicted.
  - F1-score for class 1 is 0.57, which is moderate.

These metrics show that the LDA model performs well in predicting class 0 (No product taken) but less effectively in predicting class 1 (Product taken).

- Decision Tree

Classification Report:

	precision	recall	f1-score	support
0	0.98	0.97	0.97	258
1	0.91	0.92	0.91	75
accuracy			0.96	333
macro avg	0.94	0.95	0.94	333
weighted avg	0.96	0.96	0.96	333

Table 1.31: Classification report for Decision Tree

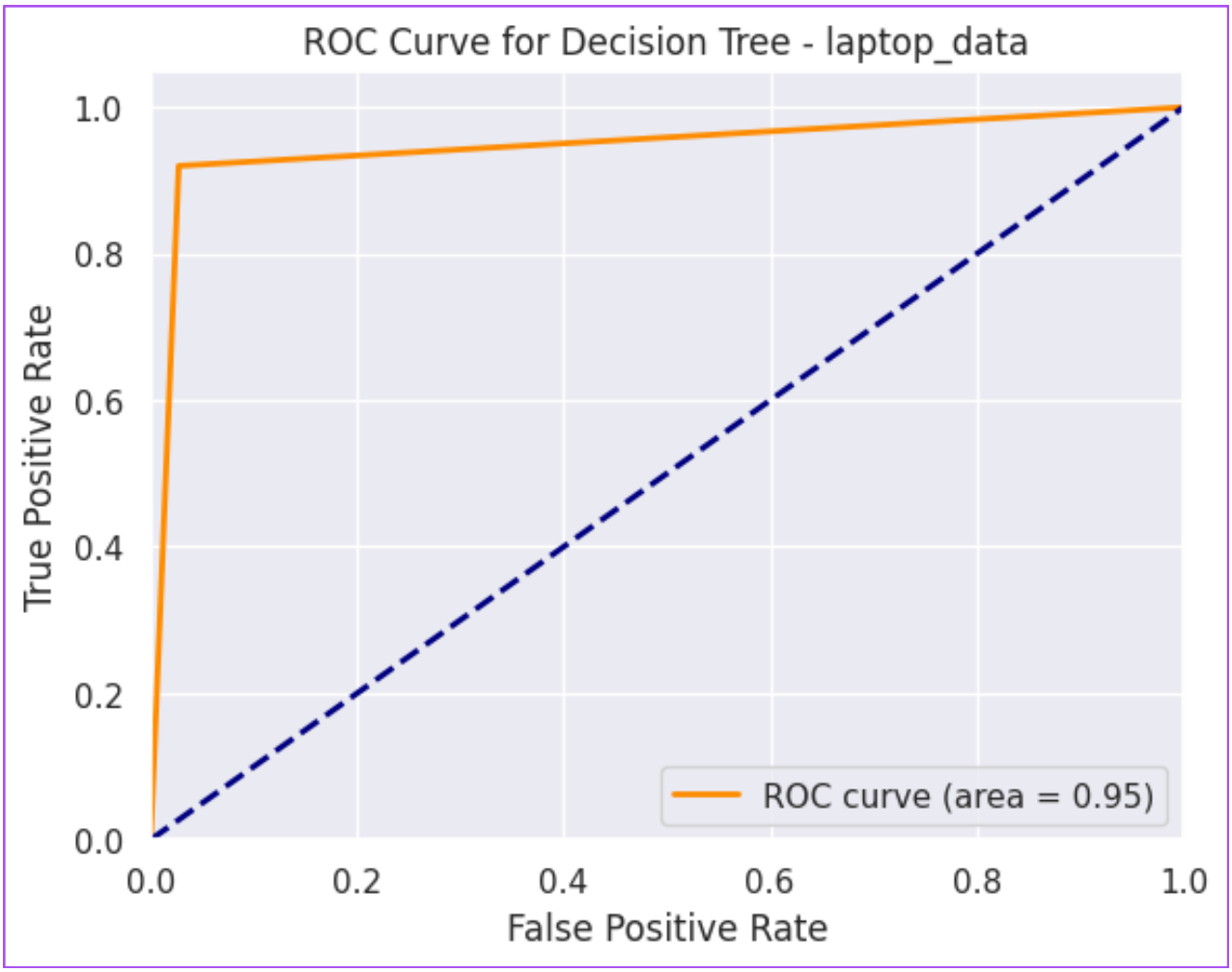


Figure 1.17: ROC curve for Decision tree

Insights:

- Confusion Matrix:
  - True Positives (TP): 69
  - True Negatives (TN): 251
  - False Positives (FP): 7
  - False Negatives (FN): 6
- RMSE (Root Mean Squared Error): The RMSE value is approximately 0.1976, indicating a low average error in the predicted values.
- R-squared (R^2): The R-squared value is approximately 0.7763. A high positive R-squared indicates that the model explains a significant portion of the variance in the data, which is generally good.
- AUC (Area Under the Curve): The AUC value is approximately 0.9464. An AUC of 0.9464 suggests excellent ability to discriminate between classes.
- Accuracy: The accuracy for the test set is approximately 0.9610, which means that the model correctly classifies about 96.10% of the test instances.

- Classification Report:
  - Precision for class 0 (No product taken) is 0.98, indicating that 98% of the predicted instances for class 0 were correct.
  - Recall for class 0 is 0.97, indicating that 97% of the actual instances for class 0 were correctly predicted.
  - F1-score for class 0 is 0.97, which is high and indicates an excellent balance between precision and recall for this class.
  - Precision for class 1 (Product taken) is 0.91, indicating that 91% of the predicted instances for class 1 were correct.
  - Recall for class 1 is 0.92, indicating that 92% of the actual instances for class 1 were correctly predicted.
  - F1-score for class 1 is 0.91, which is high and indicates a good balance between precision and recall for this class.

These metrics show that the Decision Tree model performs exceptionally well, with high precision, recall, and F1-scores for both classes.

- KNN

Classification Report:

	precision	recall	f1-score	support
0	0.88	0.93	0.91	258
1	0.72	0.57	0.64	75
accuracy			0.85	333
macro avg	0.80	0.75	0.77	333
weighted avg	0.85	0.85	0.85	333

Table 1.32: Classification report for KNN

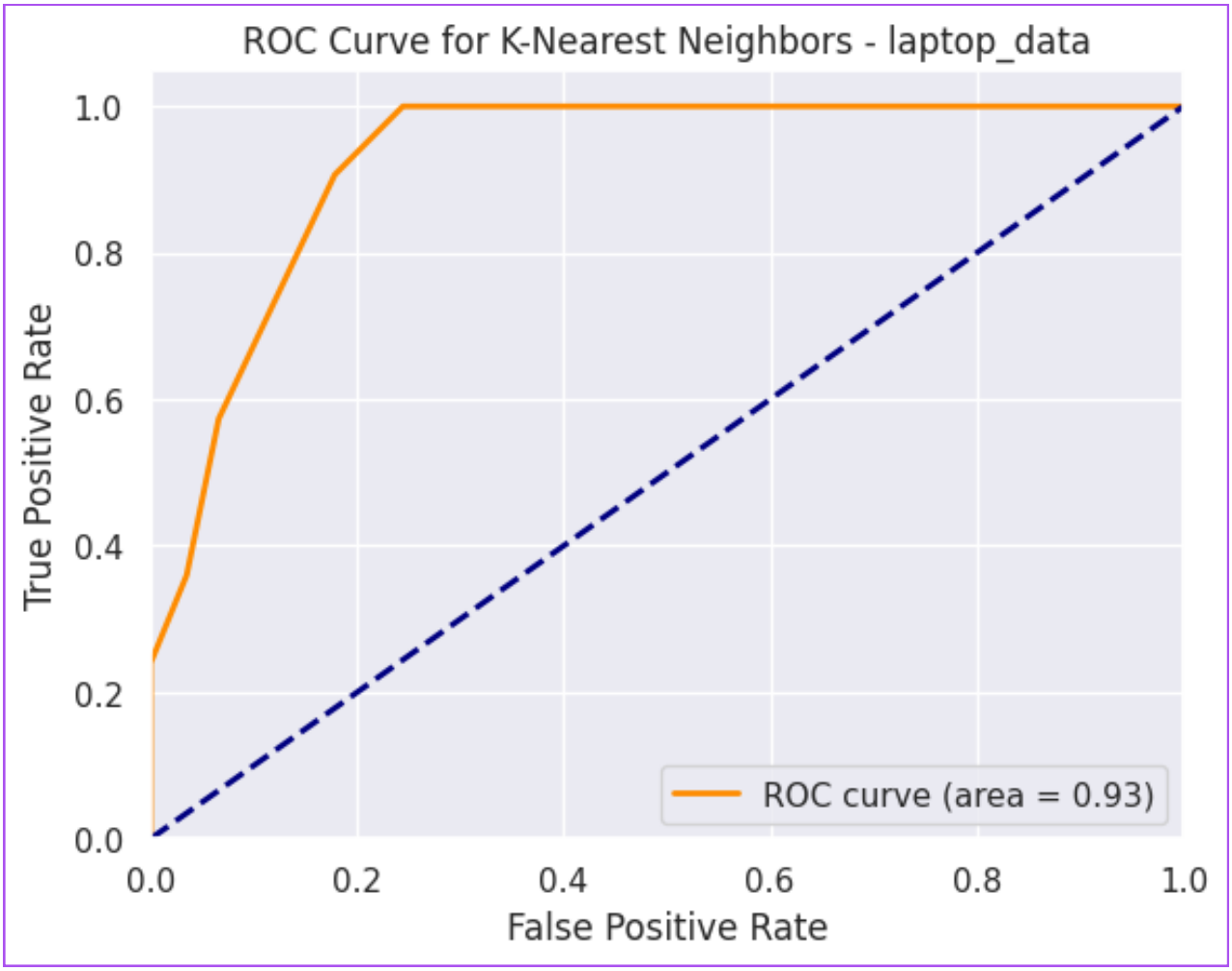


Figure 1.18: ROC curve for KNN

Insights:

- Confusion Matrix:
  - True Positives (TP): 43
  - True Negatives (TN): 241
  - False Positives (FP): 17
  - False Negatives (FN): 32
- RMSE (Root Mean Squared Error): The RMSE value is approximately 0.3836, indicating a low average error in the predicted values.
- R-squared (R^2): The R-squared value is approximately 0.1567. Relatively low value suggests that the model may not capture a significant portion of the variance.
- AUC (Area Under the Curve): The AUC value is approximately 0.9267. An AUC of 0.9267 suggests a high ability to discriminate between classes.
- Accuracy: The accuracy for the test set is approximately 0.8529, which means that the model correctly classifies about 85.29% of the test instances.

- Classification Report:
  - Precision for class 0 (No product taken) is 0.88, indicating that 88% of the predicted instances for class 0 were correct.
  - Recall for class 0 is 0.93, indicating that 93% of the actual instances for class 0 were correctly predicted.
  - F1-score for class 0 is 0.91, which is high and indicates a good balance between precision and recall for this class.
  - Precision for class 1 (Product taken) is 0.72, indicating that 72% of the predicted instances for class 1 were correct.
  - Recall for class 1 is 0.57, indicating that 57% of the actual instances for class 1 were correctly predicted.
  - F1-score for class 1 is 0.64, which is moderate.

These metrics show that the K-Nearest Neighbors (KNN) model performs relatively well in predicting class 0 (No product taken) but is less effective in predicting class 1 (Product taken).

- Naive Bayes

Classification Report:

	precision	recall	f1-score	support
0	0.87	0.88	0.87	258
1	0.56	0.53	0.54	75
accuracy			0.80	333
macro avg	0.71	0.70	0.71	333
weighted avg	0.80	0.80	0.80	333

Table 1.33: Classification report for Naive Bayes

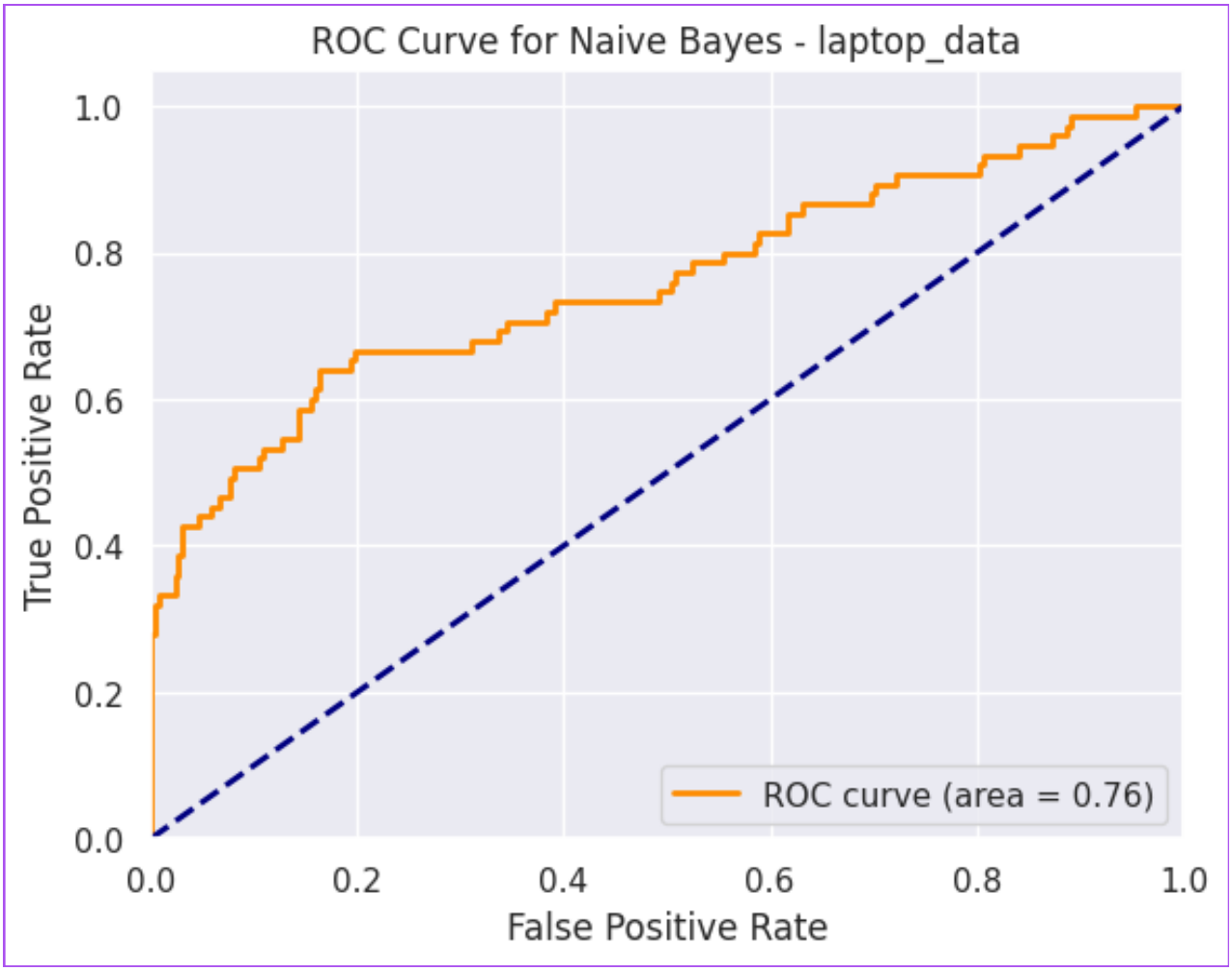


Figure 1.19: ROC curve for Naive Bayes

Insights:

- Confusion Matrix:
  - True Positives (TP): 40
  - True Negatives (TN): 226
  - False Positives (FP): 32
  - False Negatives (FN): 35
- RMSE (Root Mean Squared Error): The RMSE value is approximately 0.4486, indicating the average error in the predicted values.
- AUC (Area Under the Curve): The AUC value is approximately 0.7602. An AUC of 0.7602 suggests a moderate ability to discriminate between classes.
- Accuracy: he accuracy for the test set is approximately 0.7988, which means that the model correctly classifies about 79.88% of the test instances.

- Classification Report:
  - Precision for class 0 (No product taken) is 0.87, indicating that 87% of the predicted instances for class 0 were correct.
  - Recall for class 0 is 0.88, indicating that 88% of the actual instances for class 0 were correctly predicted.
  - F1-score for class 0 is 0.87, which is relatively high and indicates a good balance between precision and recall for this class.
  - Precision for class 1 (Product taken) is 0.56, indicating that 56% of the predicted instances for class 1 were correct.
  - Recall for class 1 is 0.53, indicating that 53% of the actual instances for class 1 were correctly predicted.
  - F1-score for class 1 is 0.54, which is moderate.

These metrics show that the Naive Bayes model performs reasonably well in predicting class 0 (No product taken) but less effectively in predicting class 1 (Product taken).

Aggregated result table for various models

Model	Data	Confusion Matrix	RMSE	R-squared	AUC	Train Accuracy	Test Accuracy
Logistic Regression	mobile_data	[[2734, 9], [439, 14]]	0.37	-0.15	0.71	0.84	0.86
LDA	mobile_data	[[2678, 65], [338, 115]]	0.36	-0.04	0.78	0.86	0.87
Decision Tree	mobile_data	[[2689, 54], [43, 410]]	0.17	0.75	0.94	1.00	0.97
K-Nearest Neighbors	mobile_data	[[2705, 38], [77, 376]]	0.19	0.70	0.98	0.98	0.96
Naive Bayes	mobile_data	[[2658, 85], [316, 137]]	0.35	-0.03	0.77	0.86	0.87
Logistic Regression	laptop_data	[[232, 26], [47, 28]]	0.47	-0.26	0.71	0.80	0.78
LDA	laptop_data	[[247, 11], [41, 34]]	0.40	0.11	0.80	0.83	0.84
Decision Tree	laptop_data	[[251, 7], [6, 69]]	0.20	0.78	0.95	1.00	0.96
K-Nearest Neighbors	laptop_data	[[241, 17], [32, 43]]	0.38	0.16	0.93	0.95	0.85
Naive Bayes	laptop_data	[[226, 32], [35, 40]]	0.45	-0.15	0.76	0.80	0.80

Table 1.34: Aggregate table of results for various models

2). Model Tuning and business implication

We will now perform cross validation for all the models to find the best base model and then perform model tuning on that model.

Cross Validation: Cross-validation is a technique used in machine learning to assess the performance and generalization ability of a predictive model. It helps in estimating how well a model will perform on unseen data, which is essential for evaluating and selecting the best model and avoiding overfitting.

The basic idea behind cross-validation is to split the available dataset into multiple subsets, train and test the model on different combinations of these subsets, and then aggregate the results to get a more accurate estimate of the model's performance.

The most common form of cross-validation is k-fold cross-validation, where the data is divided into k subsets (or folds). We have used this technique to compare models.

Result of cross-validation:

Model	Data	Accuracy Range
Logistic Regression	mobile_data	(0.84, 0.85)
LDA	mobile_data	(0.85, 0.86)
Decision Tree	mobile_data	(0.95, 0.97)
K-Nearest Neighbors	mobile_data	(0.93, 0.96)
Naive Bayes	mobile_data	(0.86, 0.86)
Logistic Regression	laptop_data	(0.78, 0.82)
LDA	laptop_data	(0.79, 0.86)
Decision Tree	laptop_data	(0.83, 0.92)
K-Nearest Neighbors	laptop_data	(0.79, 0.85)
Naive Bayes	laptop_data	(0.77, 0.83)

Table 1.35: Cross validation results for various models

From the above model we can see that best model for both 'mobile_data' and 'laptop_data' is Decision Tree. Now we will proceed with model tuning for Decision tree.

Ensemble modeling: Ensemble modeling techniques combine the predictions of multiple machine learning models to improve overall predictive performance. They are a powerful approach for increasing the accuracy and robustness of predictions. Here are some common ensemble modeling techniques

Bagging (Bootstrap Aggregating):

- Bagging combines multiple instances of a base model, typically decision trees, to reduce variance and improve model stability.
- Each model is trained on a random subset of the data with replacement (bootstrapping).
- The final prediction is often an average (for regression) or a majority vote (for classification) of the individual model predictions.
- Random Forest is a well-known ensemble method that uses bagging with decision trees as the base model.

Boosting:

- Boosting iteratively builds multiple models, with each model giving more weight to the misclassified instances from the previous model.
- The final prediction is a weighted combination of the individual model predictions.

For Bagging we have used Random forest as base estimator

Parameters: n_estimators=100

n_estimators=100: This parameter specifies the number of base estimators to create and train.

For boosting we have used grid search on decision tree to find best model and then used it as base estimator.

Parameters: cv=5, scoring='roc_auc'

cv=5: This parameter specifies the number of cross-validation folds.

scoring='roc_auc': This parameter specifies the scoring metric to be used to evaluate the performance of different hyperparameter combinations

Mobile_Data

- Bagging using Random Forest

Classification Report (mobile_data):

	precision	recall	f1-score	support
0	0.99	1.00	1.00	2747
1	1.00	0.95	0.97	449
accuracy			0.99	3196
macro avg	0.99	0.98	0.98	3196
weighted avg	0.99	0.99	0.99	3196

Table 1.36: Classification report for Bagging

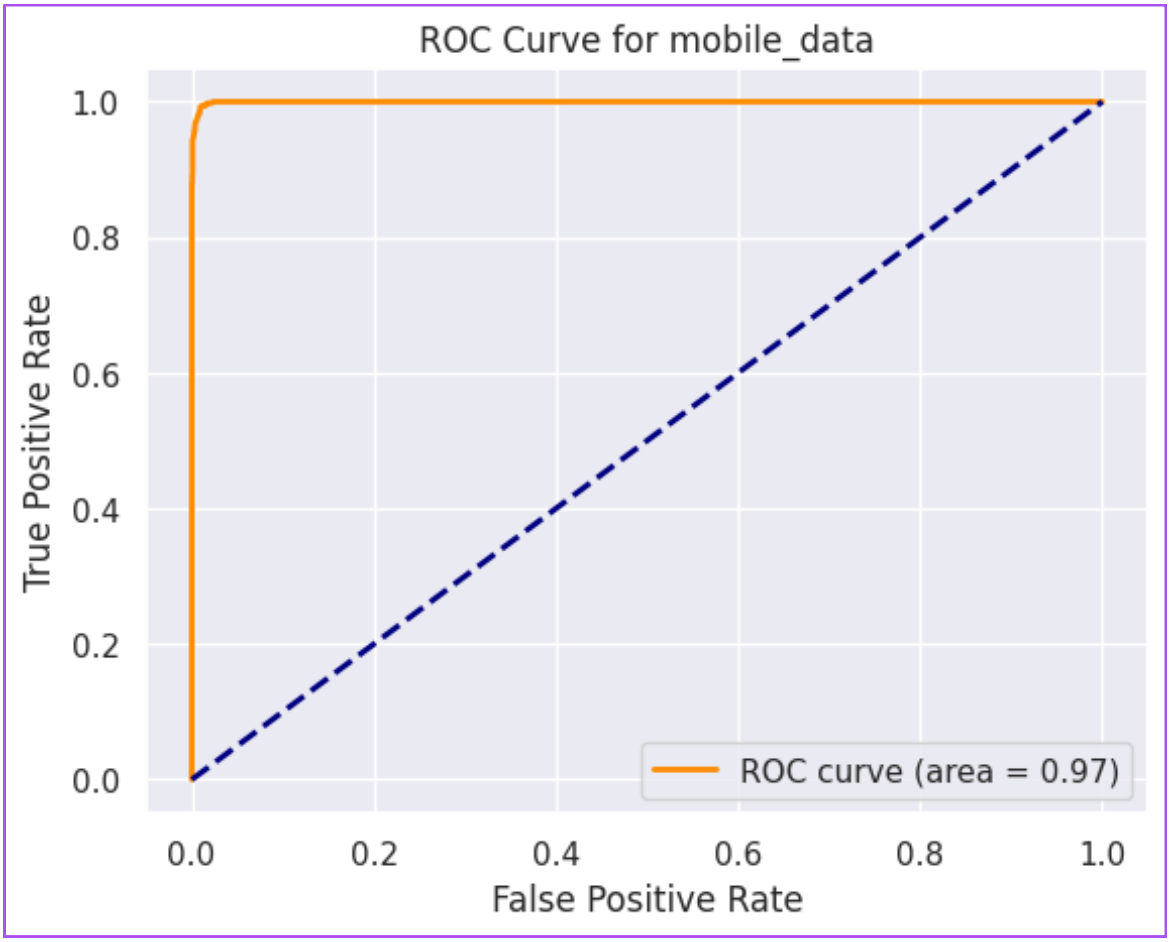


Figure 1.20: ROC curve for bagging

Insights:

- Confusion Matrix:
  - True Positives (TP): 428
  - True Negatives (TN): 2745
  - False Positives (FP): 2
  - False Negatives (FN): 21
- RMSE (Root Mean Squared Error): The RMSE value is 0.0848, which is quite low, indicating that the model's predictions are close to the actual values.
- AUC (Area Under the Curve): The AUC (Area Under the ROC Curve) score is very high at 0.9996, close to 1, indicating that the model's ability to distinguish between the two classes is excellent.
- Accuracy: The accuracy is approximately 0.9928, which means that the model correctly classifies about 99.28% of the test instances.

- Classification Report:
  - Precision for class 0 (No product taken) is 0.99, indicating that 99% correct prediction.
  - Recall for class 0 is 1.00, indicating that 100% of correct prediction.
  - F1-score for class 0 is 1.00, which is high and indicates an excellent balance between precision and recall for this class.
  - Precision for class 1 (Product taken) is 1.00, indicating that 100% of correct prediction.
  - Recall for class 1 is 0.95, indicating that 95% correct prediction.
  - F1-score for class 1 is 0.97, which is also high and indicates a good balance between precision and recall for this class.

These metrics show that the model performs exceptionally well, with high precision, recall, and F1-scores for both classes. The model excels in correctly classifying instances for both class 0 and class 1.

- Boosting using Grid Search

Classification Report (mobile_data):

	precision	recall	f1-score	support
0	0.98	0.98	0.98	2747
1	0.86	0.89	0.87	449
accuracy			0.96	3196
macro avg	0.92	0.93	0.93	3196
weighted avg	0.96	0.96	0.96	3196

Table 1.37: Classification report for Boosting

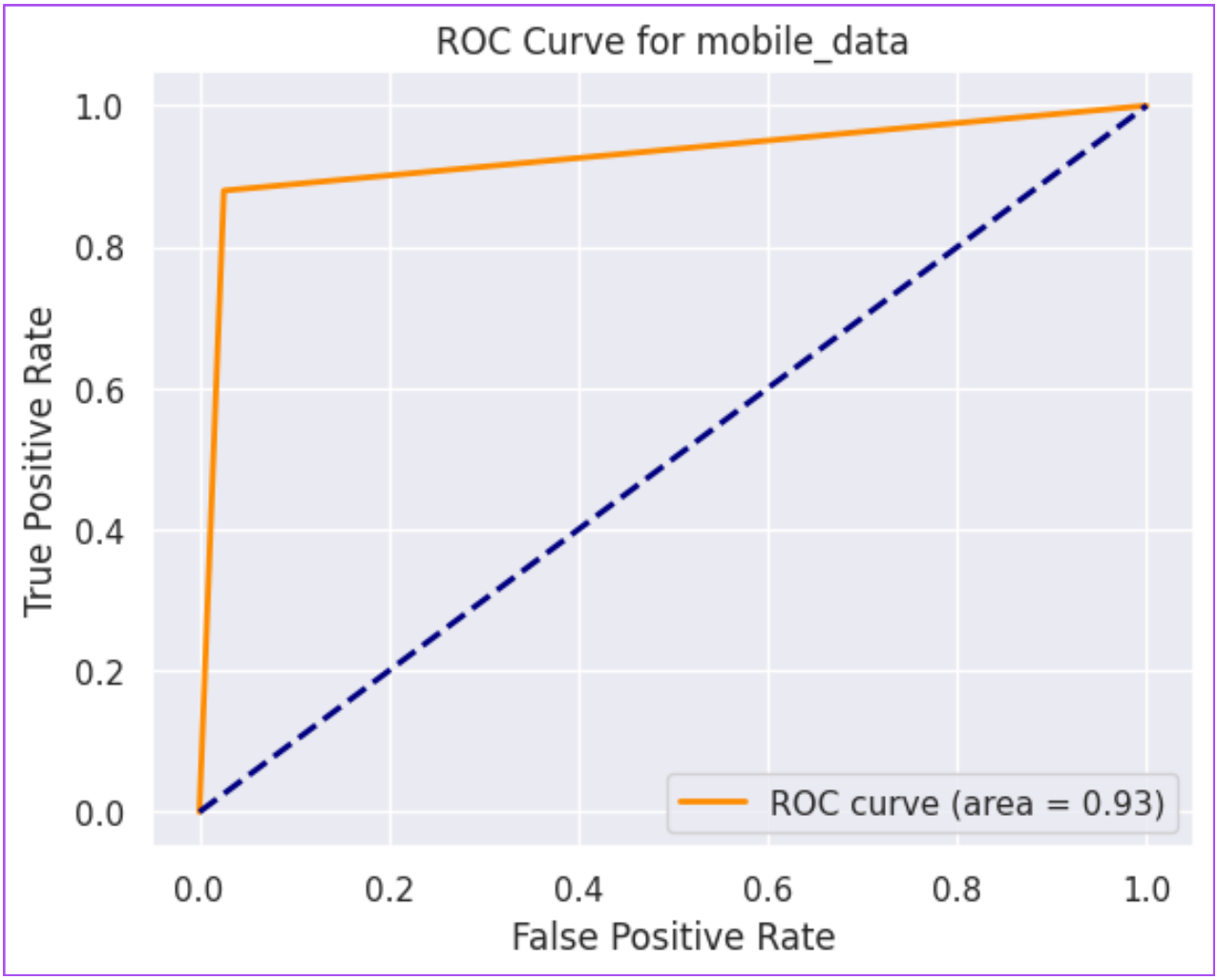


Figure 1.21: ROC curve for boosting

Insights:

- Confusion Matrix:
  - True Positives (TP): 395
  - True Negatives (TN): 2676
  - False Positives (FP): 71
  - False Negatives (FN): 54
- RMSE (Root Mean Squared Error): The RMSE is 0.1978, a low RMSE value suggests that the model's predicted probabilities are well-calibrated and close to the true class labels.
- AUC (Area Under the Curve): The AUC (Area Under the ROC Curve) score is very high at 0.9269, close to 1, indicating that the model's ability to distinguish between the two classes is excellent.
- Accuracy: The overall accuracy of the model on the mobile_data is 96.09%. This indicates that the model correctly predicts the class label for approximately 96.09% of the instances in the dataset.

- Classification Report:
  - Precision and Recall: The classification report shows that the model's precision for class 0 is high at 98%, indicating that when the model predicts class 0, it is correct 98% of the time. The precision for class 1 is 85%, meaning that when the model predicts class 1, it is correct 85% of the time. The recall for class 0 is 97%, indicating that the model correctly identifies 97% of the instances of class 0. The recall for class 1 is 88%, indicating that the model correctly identifies 88% of the instances of class 1.
  - F1-Score: The F1-score is a harmonic mean of precision and recall and provides a balance between the two. For class 0, the F1-score is high at 98%, indicating a good balance between precision and recall. For class 1, the F1-score is 86%, which is also a reasonably balanced value.

Laptop_Data

- Bagging using Random Forest

Classification Report (laptop_data):

	precision	recall	f1-score	support
0	0.97	1.00	0.99	249
1	1.00	0.92	0.96	84
accuracy			0.98	333
macro avg	0.99	0.96	0.97	333
weighted avg	0.98	0.98	0.98	333

Table 1.38: Classification report for Bagging

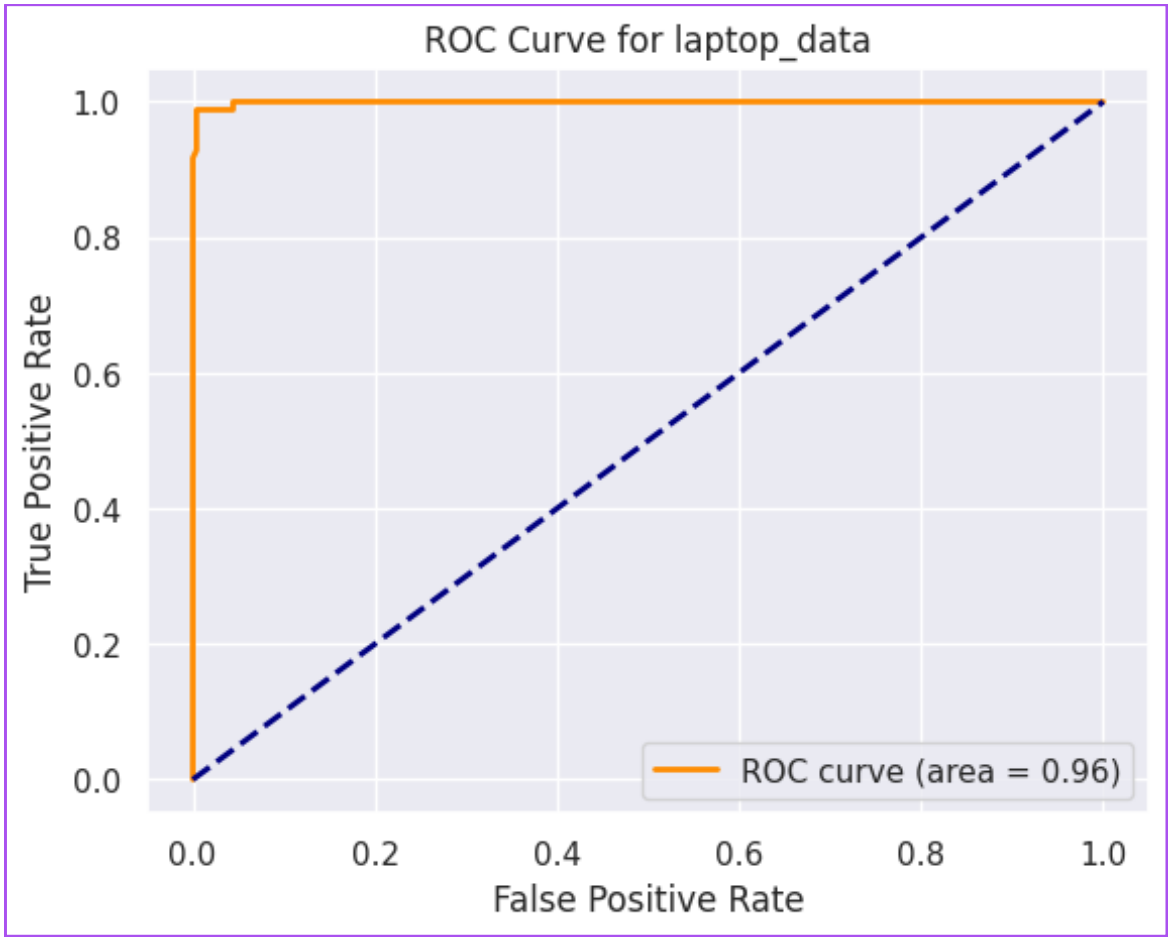


Figure 1.22: ROC curve for bagging

Insights:

- Confusion Matrix:
  - True Positives (TP): 77
  - True Negatives (TN): 249
  - False Positives (FP): 0
  - False Negatives (FN): 7
- RMSE (Root Mean Squared Error): The RMSE is 0.1450, a low RMSE value suggests that the model's predicted probabilities are well-calibrated and close to the true class labels.
- AUC (Area Under the Curve): The AUC (Area Under the ROC Curve) score is very high at 0.9992, close to 1, indicating that the model's ability to distinguish between the two classes is excellent.
- Accuracy: The overall accuracy of the model on the laptop_data is 97.90%. This indicates that the model correctly predicts the class label for approximately 97.90% of the instances in the dataset. It's a high accuracy score, reflecting the model's strong performance.

Classification Report:

- Precision and Recall: The classification report shows that the model's precision for class 0 is very high at 97%, indicating that when the model predicts class 0, it is correct 97% of the time. The precision for class 1 is even higher at 100%, meaning that when the model predicts class 1, it is correct 100% of the time. The recall for class 0 is perfect at 100%, indicating that the model correctly identifies all instances of class 0. The recall for class 1 is 92%, indicating that the model correctly identifies 92% of the instances of class 1.
- F1-Score: The F1-score is a harmonic mean of precision and recall and provides a balance between the two. For both classes 0 and 1, the F1-scores are very high at 99% and 96%, respectively, indicating an excellent balance between precision and recall.

- Boosting using Grid Search

Classification Report (laptop_data):

	precision	recall	f1-score	support
0	0.97	0.93	0.95	249
1	0.82	0.92	0.87	84
accuracy			0.93	333
macro avg	0.89	0.92	0.91	333
weighted avg	0.93	0.93	0.93	333

Table 1.39: Classification report for Boosting

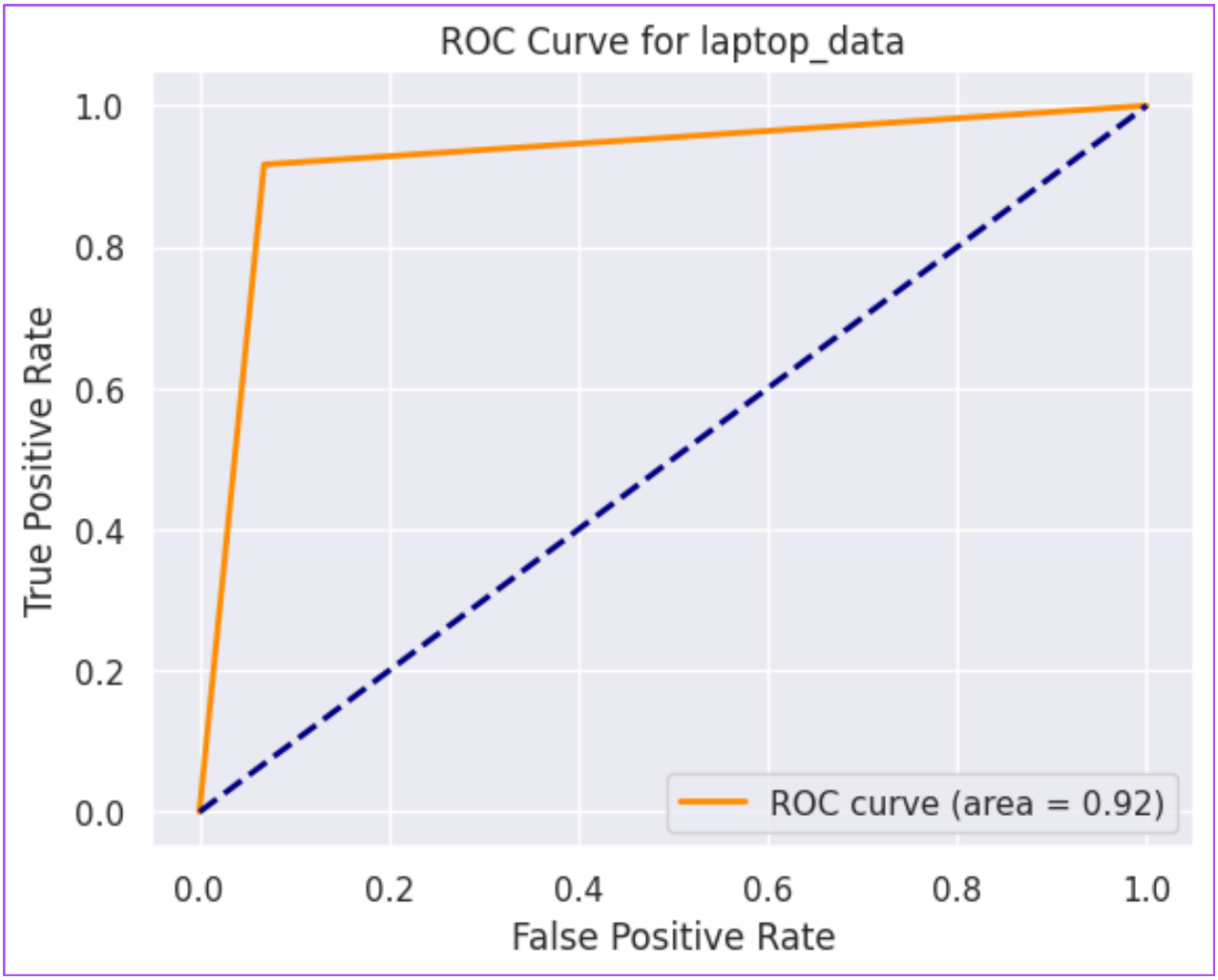


Figure 1.23: ROC curve for boosting

Insights:

- Confusion Matrix:
  - True Positives (TP): 77
  - True Negatives (TN): 232
  - False Positives (FP): 17
  - False Negatives (FN): 7
- RMSE (Root Mean Squared Error): he RMSE is 0.2685, a low RMSE value suggests that the model's predicted probabilities are well-calibrated and close to the true class labels.
- AUC (Area Under the Curve): The AUC (Area Under the ROC Curve) score is very high at 0.9242, close to 1, indicating that the model's ability to distinguish between the two classes is excellent.
- Accuracy: The overall accuracy of the model on the laptop_data is 0.9279. This indicates that the model correctly predicts the class label for approximately 97.90% of the instances in the dataset. It's a high accuracy score, reflecting the model's strong performance.

Classification Report:

- Precision and Recall: The classification report shows that the model's precision for class 0 is 97%, indicating that when the model predicts class 0, it is correct 97% of the time. The precision for class 1 is 82%, meaning that when the model predicts class 1, it is correct 82% of the time. The recall for class 0 is 93%, indicating that the model correctly identifies 93% of the instances of class 0. The recall for class 1 is 92%, indicating that the model correctly identifies 92% of the instances of class 1.
- F1-Score: The F1-score is a harmonic mean of precision and recall and provides a balance between the two. For class 0, the F1-score is 95%, indicating a good balance between precision and recall. For class 1, the F1-score is 87%, which also represents a reasonably balanced value.

Selecting Best Model for Mobile Users:

Let's compare the evaluation metrics of the Decision Tree, Bagging, and Boosting models.

1. Decision Tree - mobile_data:
    - Accuracy: 0.9675
    - AUC: 0.9387
    - Precision (class 1): 0.88
    - Recall (class 1): 0.90
    - F1-Score (class 1): 0.89
2. Bagging (Mobile_data_bagging):
    - Accuracy: 0.9912
    - AUC: 0.9996
    - Precision (class 1): 1.00
    - Recall (class 1): 0.94
    - F1-Score (class 1): 0.97
3. Boosting (Mobile_data_boosting):
    - Accuracy: 0.9609
    - AUC: 0.9269
    - Precision (class 1): 0.85
    - Recall (class 1): 0.88
    - F1-Score (class 1): 0.86

The **Bagging model** stands out as the best-performing model based on multiple metrics. It has the highest accuracy, AUC, precision, and F1-Score for class 1, while maintaining a high recall. This suggests that the Bagging model has a good balance between precision and recall and is effective at correctly classifying positive cases (class 1).

Interpretation of the Optimum Model (Bagging):

- High Accuracy: The Bagging model has an accuracy of 99.12%, indicating that it correctly predicts the class label for the majority of instances in the mobile_data.
- High Precision (class 1): The precision for class 1 is 100%, which means that when the model predicts class 1, it is correct 100% of the time. This is important because it minimizes false positives, which can be costly in certain business contexts.
- High AUC Score: The AUC score of 99.96% is exceptionally high. It indicates that the model's ability to distinguish between the two classes is almost perfect.

- High F1-Score (class 1): The F1-Score for class 1 is 0.97, reflecting a strong balance between precision and recall. This is essential in situations where both false positives and false negatives need to be minimized.
- Good Recall (class 1): The recall for class 1 is 94%, indicating that the model correctly identifies 94% of the instances of class 1. This is crucial for capturing as many positive cases as possible.

Implications on Business:

- The Bagging model's superior performance means that the business can confidently use this model to target customers with a high propensity to take up the product.
- Cost-Efficiency: By accurately identifying potential customers, the business can avoid wasting resources on those less likely to convert, leading to cost-efficiency in advertising campaigns.
- Improved Conversion Rates: The high precision and recall of the Bagging model ensure that the advertisements reach the right audience, increasing the likelihood of higher conversion rates.
- Competitive Advantage: The model's ability to identify potential customers with precision gives the business a competitive advantage by optimizing its advertising efforts.

Selecting Best Model for Laptop Users:

Let's compare the evaluation metrics of the Decision Tree, Bagging, and Boosting models.

1. Decision Tree (laptop_data):
    - Accuracy: 0.9550
    - AUC: 0.9426
    - Precision (Class 1): 0.88
    - Recall (Class 1): 0.92
    - F1-Score (Class 1): 0.90
2. Bagging (laptop_data_bagging):
    - Accuracy: 0.9790
    - AUC: 0.9992
    - Precision (Class 1): 1.00
    - Recall (Class 1): 0.92
    - F1-Score (Class 1): 0.96
3. Boosting (laptop_data_boosting):
    - Accuracy: 0.9279
    - AUC: 0.9242
    - Precision (Class 1): 0.82
    - Recall (Class 1): 0.92
    - F1-Score (Class 1): 0.87

The **Bagging (laptop_data_bagging)** model outperforms the other two models.Here are some key implications of selecting the Bagging model for the business

Interpretation of best model:

1. High Accuracy: The Bagging model achieves a high accuracy of 97.90%. This indicates that the model is effective in correctly predicting customer behavior, which is crucial for targeted advertising.
2. High AUC Score: The AUC score of 0.9992 for the Bagging model suggests that it excels in distinguishing between potential customers who are likely to take up the product and those who are not. This is essential for identifying the right audience for digital advertisements.
3. High Precision and Recall: The Bagging model achieves both high precision and recall for class 1 (potential customers). This means that the model is not only good at identifying potential customers but is also precise in doing so. It minimizes the risk of advertising to customers who are unlikely to convert.

Implications on Business:

- The Bagging model's superior performance means that the business can confidently use this model to target customers with a high propensity to take up the product.
- Cost-Efficiency: By accurately identifying potential customers, the business can avoid wasting resources on those less likely to convert, leading to cost-efficiency in advertising campaigns.
- Improved Conversion Rates: The high precision and recall of the Bagging model ensure that the advertisements reach the right audience, increasing the likelihood of higher conversion rates.
- Competitive Advantage: The model's ability to identify potential customers with precision gives the business a competitive advantage by optimizing its advertising efforts.