SOCIAL MEDIA TOURISM REPORT

ROHIT AGRAWAL

Contents

List Of Tables

List of Figures

SOCIAL MEDIA TOURISM

1.1 Introduction

a). Problem Statement

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 2models 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

b). Need of the Study

- Digital media has penetrated all aspects of tourism and have led to fundamental changes in the way tourism experiences are planned, consumed, evaluated and marketed. In the tourism industry, websites and social media provide a wealth of information with regards to experiences and review of the destination, views, likes, comments, travel check ins.
- Social media marketing generates more business exposure, increased traffic and improved search, generating leads and improved sales at lower cost. With more than 2 million reviews and being updated every minute, hospitality and tourism marketers realised its importance due to the intangibility of the goods they sell. To understand what kind of information consumers seek online and how they actually use information acquired online from other consumers to make their travel and hospitality decisions.
- The project will help the aviation company learn about the digital behaviour of the customers. It will help in identifying the group of customers who have a high propensity to take up the product. To understand what is happening and why it's like this we need to study / analyze the existing data and predict the best solutions for the future.

c). Understanding Business/Social Opportunity

- The leading trends towards the Social Networking has drawn high public attention from past ‘two’ decades. For both small businesses and large corporations, social media is playing a key role in brand building and customer communication. Apart from social networking sites like Facebook, Twitter, Instagram, Snapchat etc, other categories like news, Communication, Commenting, Marketing, Banking, Entertainment etc. are also generating huge social media content every minute.

- The understanding of the customer's behaviours on a social media platform will result in targeting advertisements according to the needs and wants of the specific set of customers that can result in high propensity to take up the product. Apart from this company can understand the problems associated with the customers that have posted bad reviews. Then, instead of calling each and every customer company can utilize its resources to improve revenue

1.2 EDA and Business Implication

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

Data is collected through social media monitoring and online marketing analytics of the company's page as well as various travelled related pages along with the monitoring of the customer's account throughout the year on daily basis.

Sample of the dataset

The dataset provided is stored as "Social+Media+Data+for+DSBA.csv". Output is displayed below for the dataset (first 5 records) after importing the file in python:

	UserID	Taken_product	Yearly_avg_view_on_travel_page	preferred_device	total_likes_on_outstation_checkin_given	yearly_avg_Outstation_checkins
0	1000001	Yes	307.0	iOS and Android	38570.0	1
1	1000002	No	367.0	iOS	9765.0	1
2	1000003	Yes	277.0	iOS and Android	48055.0	1
3	1000004	No	247.0	iOS	48720.0	1
4	1000005	No	202.0	iOS and Android	20685.0	1

Table 1. Dataset Sample First 5 Records

	UserID	Taken_product	Yearly_avg_view_on_travel_page	preferred_device	total_likes_on_outstation_checkin_given	yearly_avg_Outstation_checkins
11755	1011756	No	279.0	Laptop	30987.0	23
11756	1011757	No	305.0	Tab	21510.0	6
11757	1011758	No	214.0	Tab	5478.0	4
11758	1011759	No	382.0	Laptop	35851.0	2
11759	1011760	No	270.0	Tab	22025.0	8

Table 2. Dataset Sample Last 5 Records

Data Dictionary

The dataset consists of 17 variables. The dataset consists of information regarding Social Media. The variables are as below:

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

Table 3. Data Dictionary

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

Dimension of the dataset:

Using shape function in python it was observed that the dataset contains data of 11760 customers and 17 variables.

Summary of the Dataset

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
UserID	11760.0	NaN	NaN	NaN	1005880.5	3394.963917	1000001.0	1002940.75	1005880.5	1008820.25	1011760.0
Taken_product	11760	2	No	9864	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Yearly_avg_view_on_travel_page	11179.0	NaN	NaN	NaN	280.830844	68.182958	35.0	232.0	271.0	324.0	464.0
preferred_device	11707	10	Tab	4172	NaN	NaN	NaN	NaN	NaN	NaN	NaN
total_likes_on_outstation_checkin_given	11379.0	NaN	NaN	NaN	28170.481765	14385.032134	3570.0	16380.0	28076.0	40525.0	252430.0
yearly_avg_Outstation_checkins	11685	30	1	4543	NaN	NaN	NaN	NaN	NaN	NaN	NaN
member_in_family	11760	7	3	4561	NaN	NaN	NaN	NaN	NaN	NaN	NaN
preferred_location_type	11729	15	Beach	2424	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Yearly_avg_comment_on_travel_page	11554.0	NaN	NaN	NaN	74.790029	24.02665	3.0	57.0	75.0	92.0	815.0
total_likes_on_outofstation_checkin_received	11760.0	NaN	NaN	NaN	6531.699065	4706.613785	1009.0	2940.75	4948.0	8393.25	20065.0
week_since_last_outstation_checkin	11760.0	NaN	NaN	NaN	3.203571	2.616365	0.0	1.0	3.0	5.0	11.0
following_company_page	11657	4	No	8355	NaN	NaN	NaN	NaN	NaN	NaN	NaN
montly_avg_comment_on_company_page	11760.0	NaN	NaN	NaN	28.661565	48.660504	11.0	17.0	22.0	27.0	500.0
working_flag	11760	2	No	9952	NaN	NaN	NaN	NaN	NaN	NaN	NaN
travelling_network_rating	11760.0	NaN	NaN	NaN	2.712245	1.080887	1.0	2.0	3.0	4.0	4.0
Adult_flag	11760.0	NaN	NaN	NaN	0.793878	0.851823	0.0	0.0	1.0	1.0	3.0
Daily_Avg_mins_spend_on_traveling_page	11760.0	NaN	NaN	NaN	13.817432	9.070657	0.0	8.0	12.0	18.0	270.0

Table 4. Description of Dataset

- It clearly shows that there are high number of customers that have not purchased the product of the company.
- The average weeks since last outstation check-in is 2.62 and have below average travel rating of 2.71 meaning close friends of customers who also like travelling are very less.
- Most of the customers do not follow the company page as well and prefer “Tab” as the operating device.
- Most of the customers have a family size of 3 and travel 1 time in a year.
- The customers spend an average of 13.82 minutes on a travelling page on daily basis

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

Structure of the Dataset:

Structure of the Dataset was computed using .info () function in python. This function explains which variables are of what datatype.

```
<class 'pandas.core.frame.DataFrame'>
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)
memory usage: 1.5+ MB
```

This shows the number of columns in the data and data type of each and every column. The entire dataset consists of 3 float type variables, 7 integer type variables and 7 object or string type variables.

Checking for Missing Values

While analyzing the data. One of the key steps is that the missing values or “NA” needs to be checked and dropped from the dataset for the ease of evaluation. As null values can give errors or discrepancies in results. Missing Values was computed using .isnull().sum() function in python.

```
UserID                                          0
Taken_product                                   0
Yearly_avg_view_on_travel_page                581
preferred_device                               53
total_likes_on_outstation_checkin_given       381
yearly_avg_Outstation_checkins                 75
member_in_family                                0
preferred_location_type                        31
Yearly_avg_comment_on_travel_page             206
total_likes_on_outofstation_checkin_received    0
week_since_last_outstation_checkin              0
following_company_page                        103
```

```
montly_avg_comment_on_company_page                  0
working_flag                                        0
travelling_network_rating                           0
Adult_flag                                          0
Daily_Avg_mins_spend_on_traveling_page              0
dtype: int64
```

From the above results we can see that there is 1430 missing value present in the dataset.

Checking for Duplicates

While analyzing the data. One of the key steps is that the duplicates needs to be checked and dropped from the dataset for the ease of evaluation. Else they will affect the analysis. Duplicates was computed using .duplicated().sum() function in python. After computing from python we have found that output the dataset does not have any duplicates.

d). Univariate analysis (distribution and spread for every continuous attribute, distribution of data in categories for categorical ones)

Univariate Analysis

- To begin with Histograms and Box plot are plotted for all the numerical variables using sns.distplot and sns.boxplot function from seaborn package. Also, distribution could be viewed. Whether the data is right skewed or left skewed.

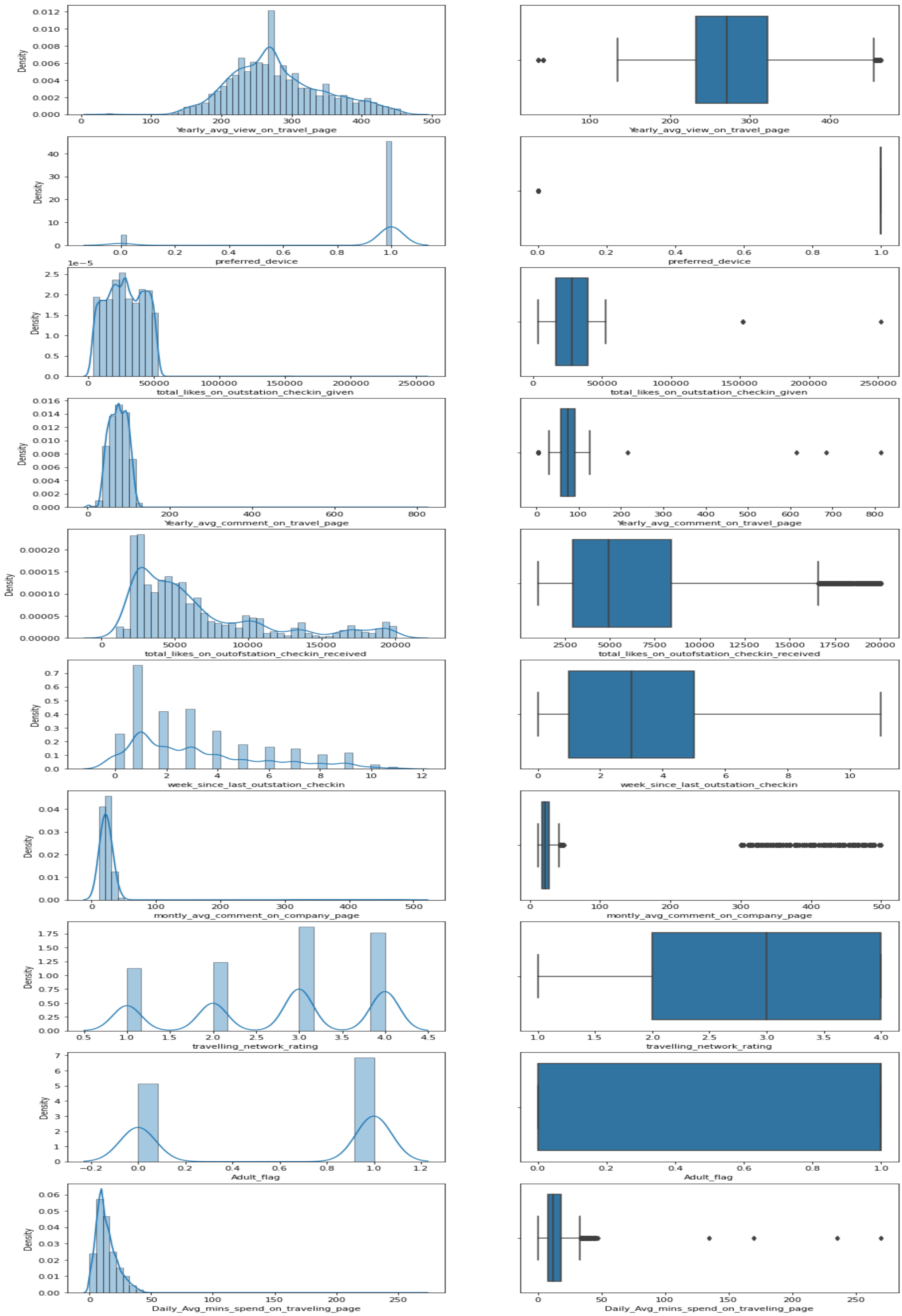
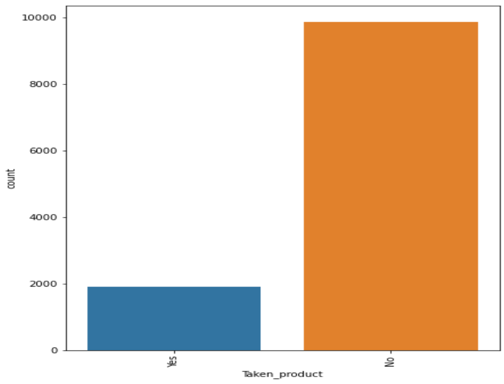


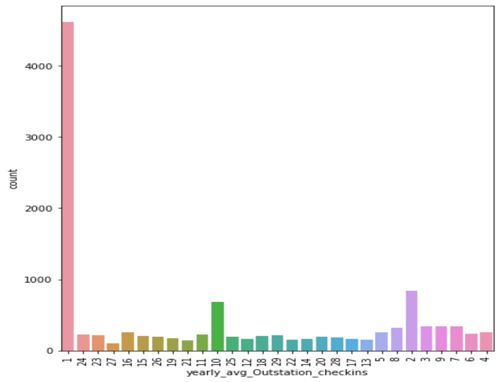
Figure 1. Histogram and Box Plot

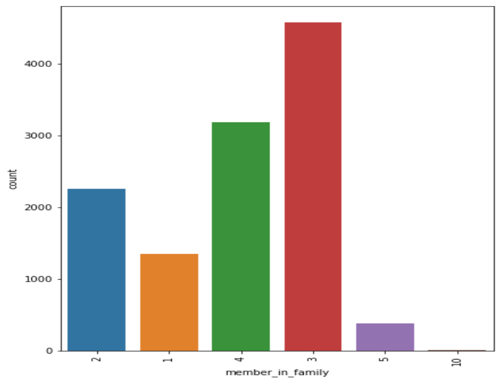
Insights

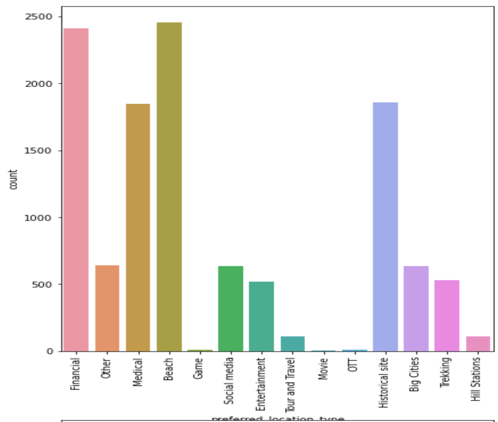
- Most of the numerical columns in the data are rightly skewed and have large number of outliers in the data set.
- The variable “Yearly average view on Travel page” is somewhat normally distributed but still have outliers on both sides of the distribution
- “week since last outstation check-in” variables has no outlier in it despite showing somewhat right skewness.

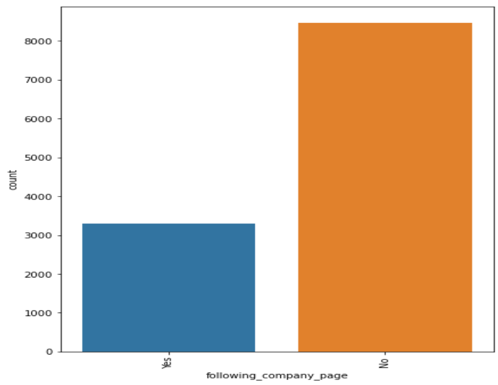
➢ In case of Categorical variable, we can observe the frequencies from count plot for Categorical variable. Using Seaborn count plot which gives the count of observations in each category.

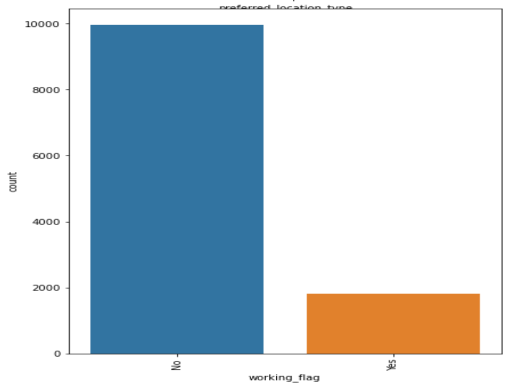












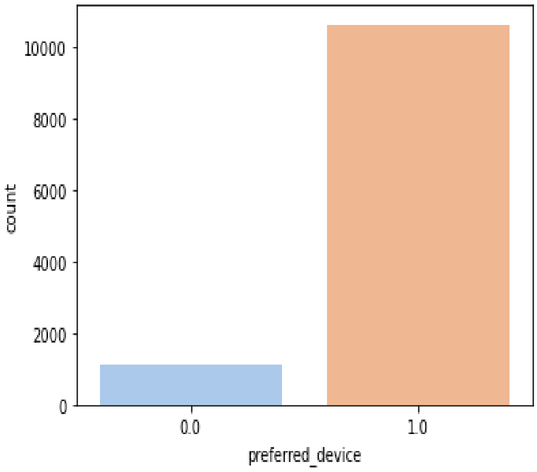


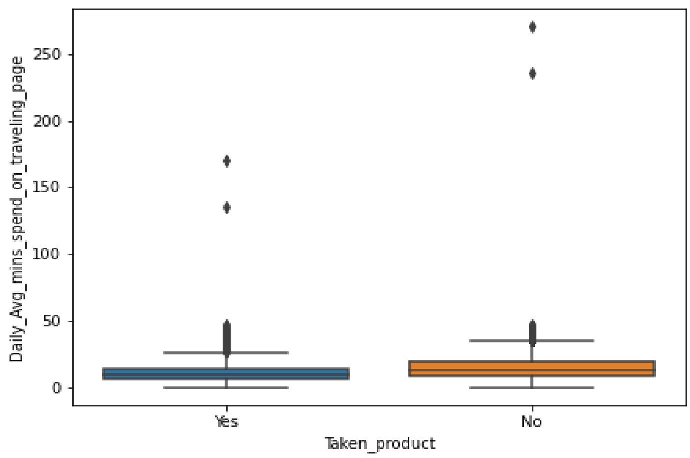
Figure 2. Count Plot for Categorical Variable

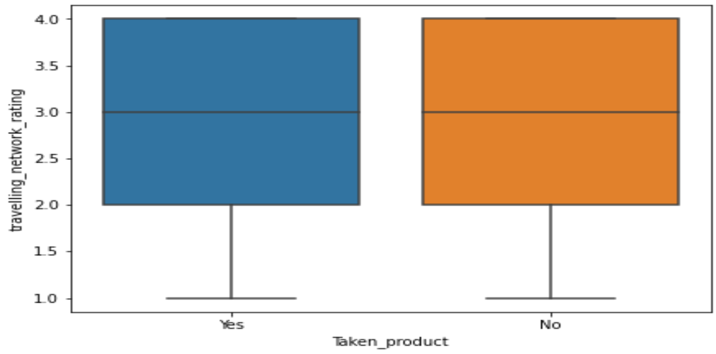
Insights

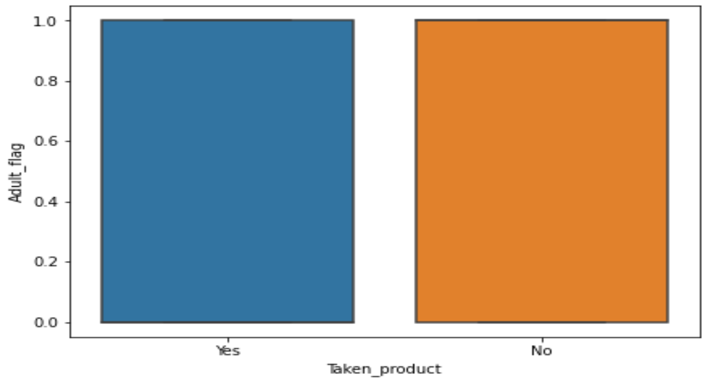
- There are 11760 customers in the data out which less customers have purchased the product whereas most of the customers did not purchase the product.
- Mobile devices are preferred as only 1108 customers prefer “Laptop” devices.
- Most of the customers travel once per year (4544 customers) followed by twice per year visits (844).
- It also shows that most customers have 3 members in the family followed by 4 members.
- Most of the customers prefer “Beach” as their location closely followed by location for “Financial” purpose.
- Most of the people are not following the company page.
- Most of the customer base is a non-working class.

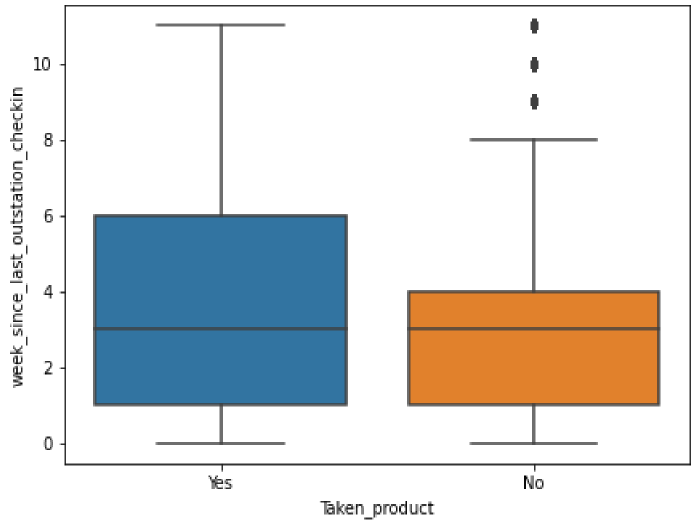
e). Bivariate analysis (relationship between different variables,correlations)

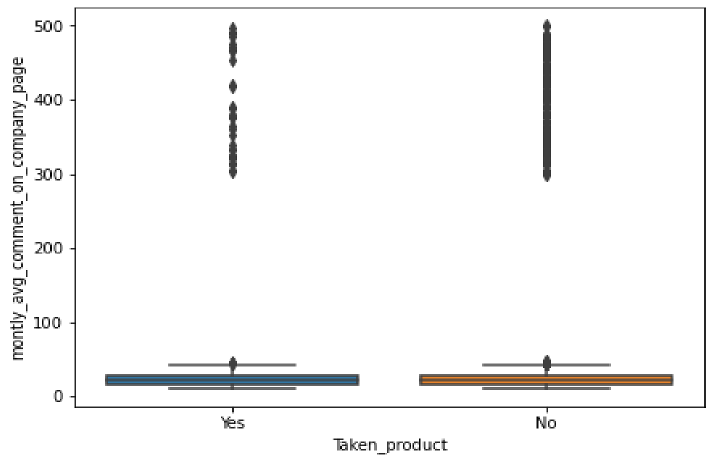
➢ We will pick one Numerical Variable and draw its relationship with variable Taken_product.




travelling_network_rating
4.0
3.5
3.0
2.5
2.0
1.5
1.0
Yes
No
Taken_product


Adult_flag
1.0
0.8
0.6
0.4
0.2
0.0
Yes
No
Taken_product


week_since_last_outstation_checkin
10
8
6
4
2
0
Yes
No
Taken_product


montly_avg_comment_on_company_page
500
400
300
200
100
0
Yes
No
Taken_product

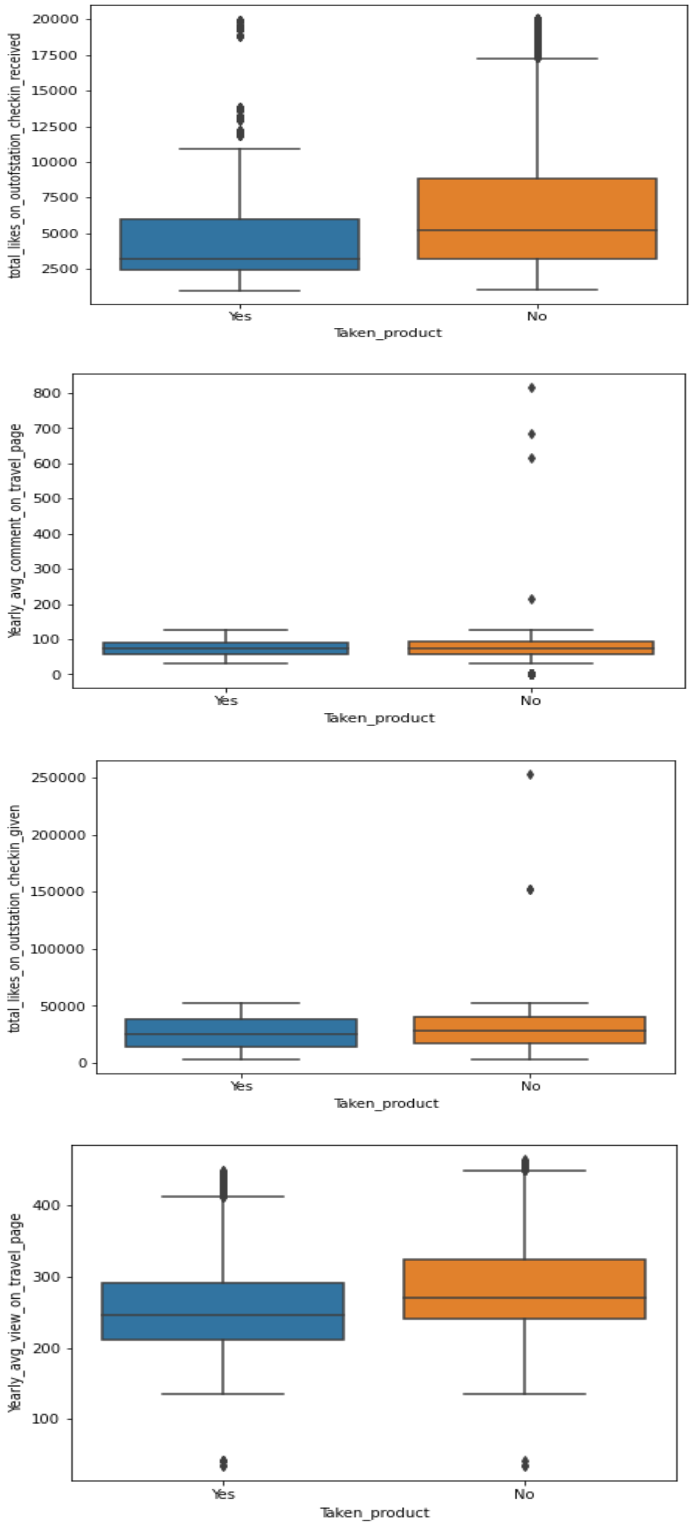


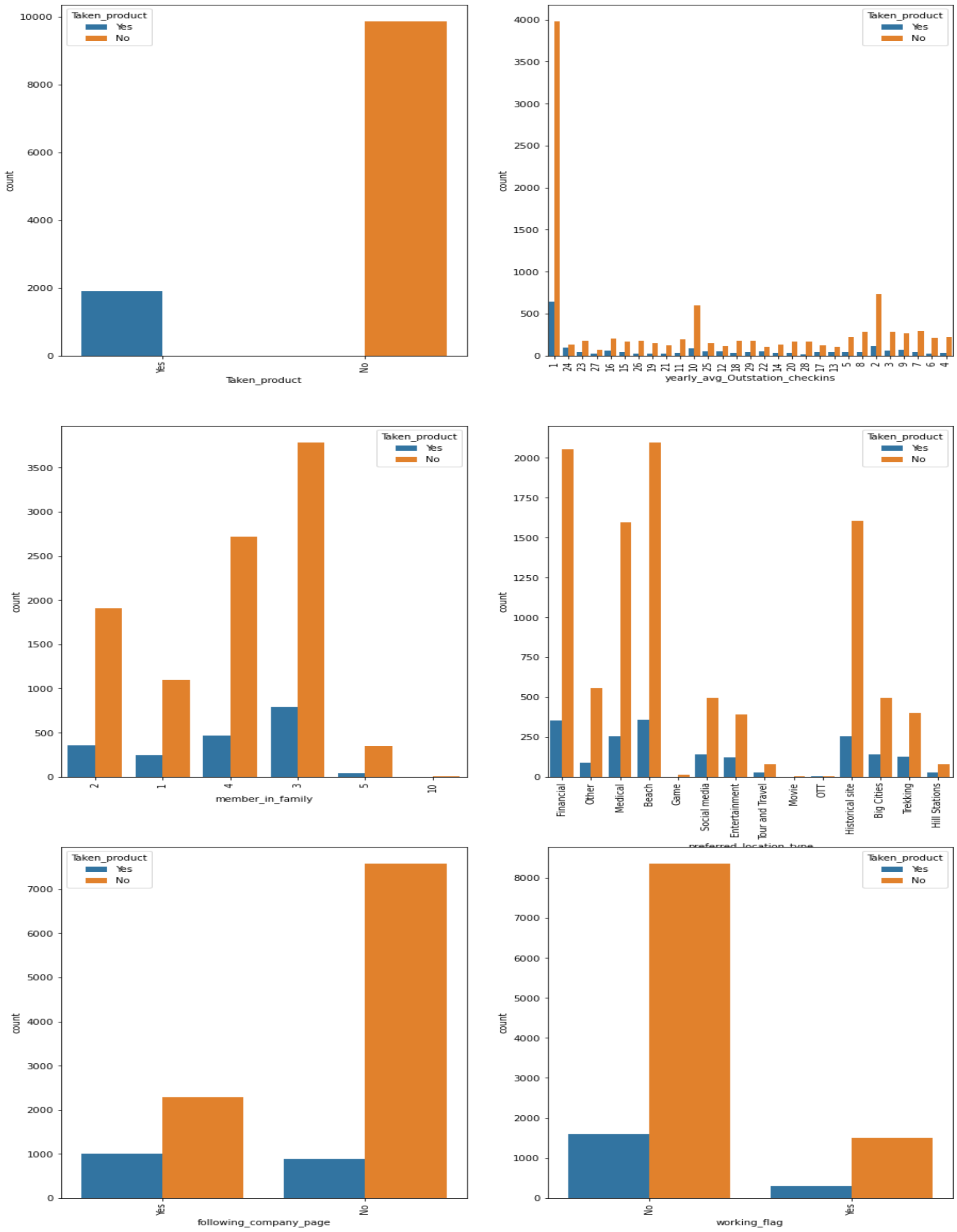
Figure 3. Numerical Variable vs Taken_product

Insights

- It can be observed that despite a greater number of “yearly average view on travel page” the customers have not purchased the product.
- Customers that have spent less on travel page viewing have high tendency to purchase the product.

- In most of the cases No cases have large number outliers.
- Fewer people have put likes in outstation check-in have purchased product whereas less people have purchased the product despite large number of likes on outstation check-in.
- Customer that have close friends which love to travel have high chances of purchasing the product but as the rating increases the likelihood of purchasing a product decline.

➢ Now we will pick one Categorical Variable and draw its relationship with variable Taken_product.



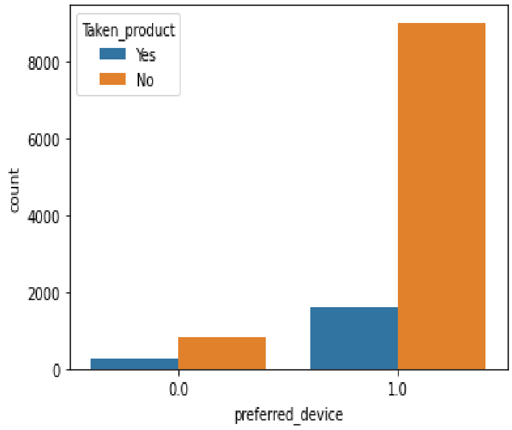


Figure 4. Categorical Variable vs Taken_product

Insights

- Customers that visits once in a year does not like to purchase the ticket from the company.
- As the family size of the customer increases the chances of purchasing the product decreases.
- The people that travel for beaches and financial purposes which are two major reasons for travel are less likely to buy the company's products.
- Customers following the company's page have high chances of purchasing the products.
- The working category have high chances of purchasing the product as compared to the people that are not working.
- It is observed that despite any preferred devices there is very high attrition rate amongst the customers as they are not interested in purchasing the product at all.

Heat Map (Relationship Analysis)

Below is Heat Map or Correlation Matrix to evaluate the relationship between different variables in our dataset. This graph can help us to check for any correlations between different variables.

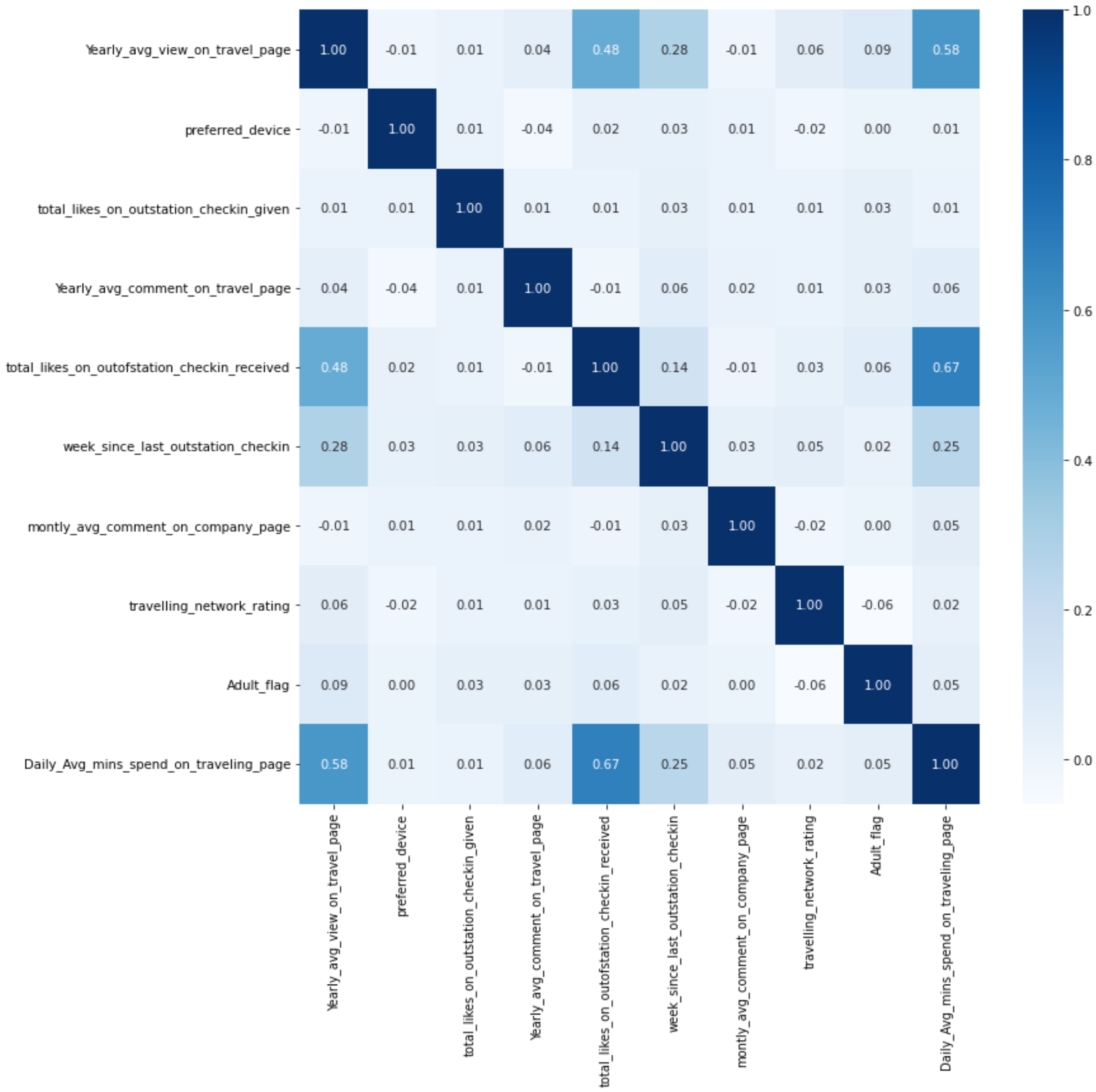


Figure 5. Heat Map

Insights

- It can be observed that there is very weak correlation amongst the variables
- There are some variables like "total likes on outstation received" and "yearly average view on travel page" that have a moderate correlation of 0.48 between them.
- Variables like "Daily average minutes spend on travelling page" and "yearly average view on travel page" also have a moderate correlation of 0.58
- "Daily average minutes spend on travelling page" and "total likes on outstation received" of moderate correlation 0f 0.67 amongst them

f). Any business insights using clustering

- Performing K-Means clustering
- Standardize the dataset using Standard Scaler function
- Identify the inertia value for multiple cluster groups and identify the cut-off
- Plot the inertia values in a line plot (elbow curve) and identify the cutoff value

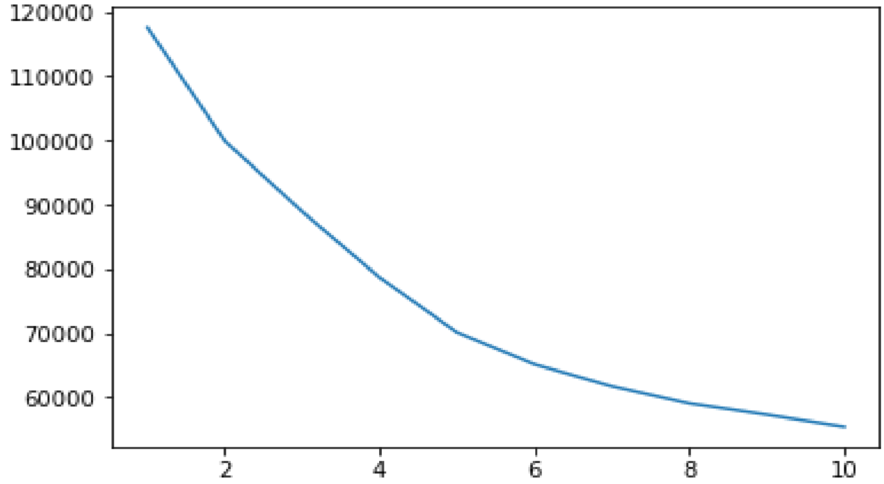


Figure 6.Elbow Curve

- As per the above plot i.e. within sum of squares (wss) method we can conclude that the optimal number of clusters is not clearly visible.

```
The Average Silhouette Score for 2 clusters is 0.20865
The Average Silhouette Score for 3 clusters is 0.2266
The Average Silhouette Score for 4 clusters is 0.17339
The Average Silhouette Score for 5 clusters is 0.16737
The Average Silhouette Score for 6 clusters is 0.17522
The Average Silhouette Score for 7 clusters is 0.1597
The Average Silhouette Score for 8 clusters is 0.16124
The Average Silhouette Score for 9 clusters is 0.15477
```

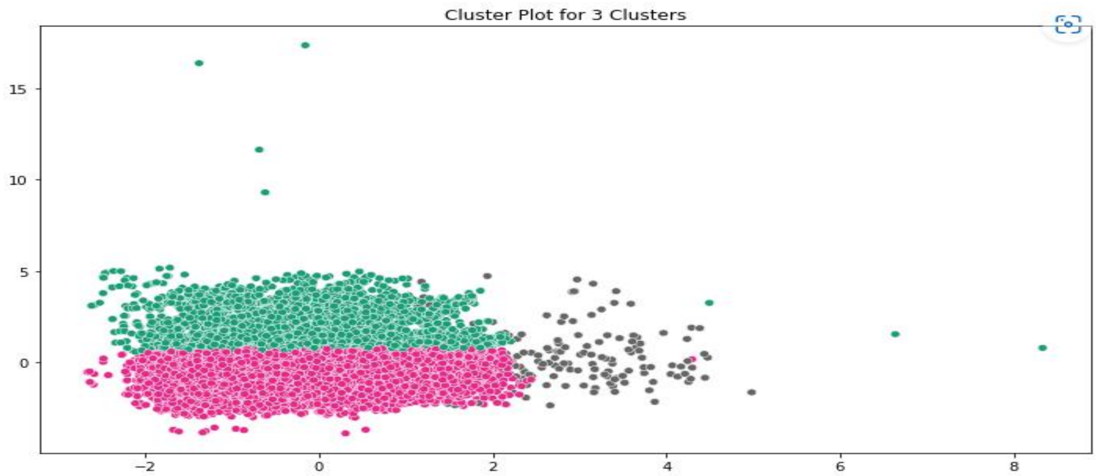


Figure 7. Cluster Plot for 3 Clusters

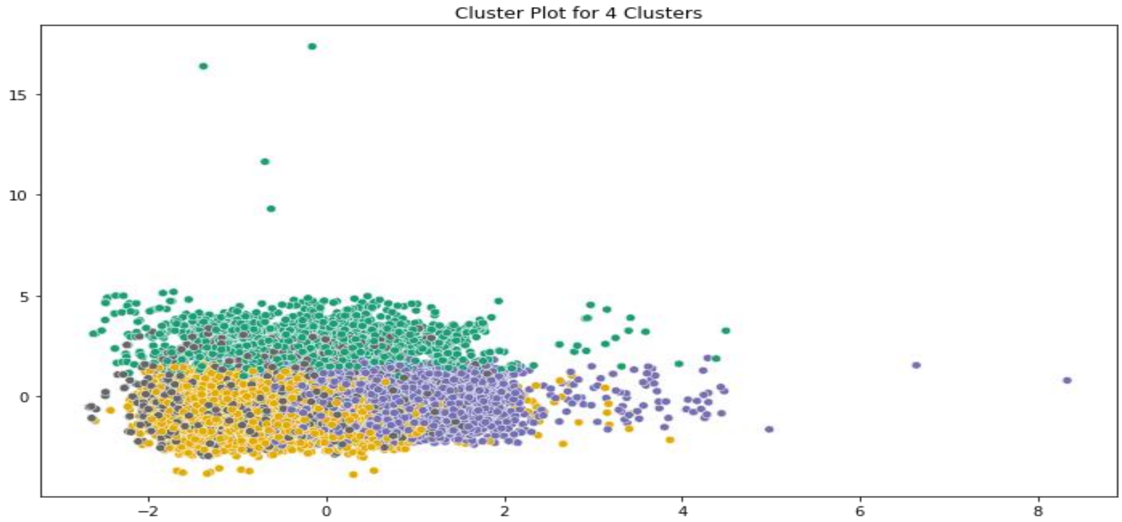


Figure 8. Cluster Plot for 4 Clusters

- It is also clear from the graph that there are overlapping of the cluster. For 3 cluster less overlapping in comparison to 4 clusters
- As there are less overlapping in cluster 3 So we will take no. of clusters is equal to 3.

Clusters are formed:

```
0    5419
1    2117
2    4224
Name: Clus_kmeans, dtype: int64
```

lling_network_rating	Adult_flag	Daily_Avg_mins_spend_on_traveling_page	working_flag_lbl	following_company_page_lbl	preferred_location_type_lbl	Clus_kmeans
1.0	0.0	8.0	0.0	1.0	13.0	2
4.0	1.0	10.0	1.0	0.0	13.0	0
2.0	0.0	7.0	0.0	1.0	10.0	2
3.0	0.0	8.0	0.0	1.0	13.0	2
4.0	1.0	6.0	0.0	0.0	11.0	0

Table 5. Sample of Dataset with 3 Clusters

Clus_kmeans	0	1	2
Taken_product	0.106108	0.092584	0.266335
Yearly_avg_view_on_travel_page	267.750784	360.917572	256.233546
preferred_device	1.000000	1.000000	1.000000
total_likes_on_outstation_checkin_given	28710.933475	28523.623288	27194.836174
yearly_avg_Outstation_checkins	8.498801	8.059991	7.822443
member_in_family	2.910685	3.059046	2.866004
Yearly_avg_comment_on_travel_page	75.201513	75.655881	73.436435
total_likes_on_outofstation_checkin_received	4896.112751	13716.650449	4628.143703
week_since_last_outstation_checkin	3.011257	4.381672	2.859848
montly_avg_comment_on_company_page	22.897767	23.299008	22.575994
Adult_flag	1.000000	0.610770	0.000000
Daily_Avg_mins_spend_on_traveling_page	11.211294	26.065187	10.510890
working_flag_lbl	0.160177	0.145961	0.149384
following_company_page_lbl	0.270530	0.292395	0.286932
preferred_location_type_lbl	11.344713	11.358999	11.307528
frequency	5419.000000	2117.000000	4224.000000

Table 6. Cluster Observation

Insights

- There are 5419 customers in Cluster 0, 2117 customers in Cluster 1 and 4224 customers in Cluster 2.
- Total likes on outstation check-ins received is the major differentiator between the clusters It shows that least likes received in Cluster 2 and Most likes received are grouped in Cluster 1.
- Yearly average view on travel page supports this grouping and shows a similar pattern. It shows that Yearly average view on travel page in cluster 0 is higher than Cluster 1 and Cluster 2
- From other variables we cannot generate more useful insights may be because clusters are not forming properly which can be attributed to the fact that data is highly imbalanced and due to which the boundaries are very less
- We cannot make much conclusions after performing clustering.

g). Other business insights

- It can be observed that most travelled location is beach and financial related travels and followed by medical related travels.
- company should come up with discount offer the user who travels for medical related travels as this will have good customer experience in these unprecedented times and it will increase brand value.
- The people who don't follow company page have high average view on company page and people who follow company page has less view.
- Social media campaigns should be there so that we can grab attention of social media mob as it clearly impact business.

1.3 Data Cleaning and Pre-processing

a). Treating Bad Data

- In Prefered_location_type column 'Tour Travel' and 'Tour and Travel' are same. We have replaced '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 and Travel      107
NaN                   31
Game                  12
OTT                    7
Movie                  5
Name: preferred_location_type, dtype: int64
```

- In yearly_avg_Outstation_checkins column '*' in data present. We are replacing with the mode. So '*' is replaced with '1'.

```
1       4544
2        844
10       682
9        340
7        336
3        336
8        320
5        261
4        256
16       255
6        236
11       229
24       223
29       215
23       215
18       208
15       206
26       199
20       199
25       198
28       180
19       176
14       167
17       160
12       159
22       152
13       150
21       143
```

```
27        96
NaN       75
Name: yearly_avg_Outstation_checkins, dtype: int64
```

- In member_in_family column 'Three' and '3' are same. We have replaced 'Three' with '3'.

```
3      4576
4      3184
2      2256
1      1349
5       384
10       11
Name: member_in_family, dtype: int64
```

- In Adult_flag column as per features there should be two factors only yes or no. However, we have 2 and 3 additional one so we will assume 2 and 3 are adult and rest are minors.

```
1    6712
0    5048
Name: Adult_flag, dtype: int64
```

- In following_company_page replacing with '1' with 'Yes' and '0' with 'No'

```
No      8360
Yes     3297
NaN      103
Name: following_company_page, dtype: int64
```

- In preferred_device column we have replaced all column with Mobile except Laptop. Replaced Mobile with 1 and Laptop with 0

```
1.0    10652
0.0     1108
Name: preferred_device, dtype: int64
```

b). Removal of unwanted variables

We have dropped User_Id column from data.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11760 entries, 0 to 11759
Data columns (total 16 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   Taken_product                           11760 non-null  object
 1   Yearly_avg_view_on_travel_page          11179 non-null  float64
 2   preferred_device                        11760 non-null  float64
 3   total_likes_on_outstation_checkin_given 11379 non-null  float64
 4   yearly_avg_Outstation_checkins          11685 non-null  object
 5   member_in_family                        11760 non-null  object
 6   preferred_location_type                 11729 non-null  object
 7   Yearly_avg_comment_on_travel_page       11554 non-null  float64
 8   total_likes_on_outofstation_checkin_received 11760 non-null  int64
 9   week_since_last_outstation_checkin      11760 non-null  int64
 10  following_company_page                  11657 non-null  object
 11  montly_avg_comment_on_company_page      11760 non-null  int64
 12  working_flag                            11760 non-null  object
 13  travelling_network_rating               11760 non-null  int64
 14  Adult_flag                              11760 non-null  int32
 15  Daily_Avg_mins_spend_on_traveling_page  11760 non-null  int64
dtypes: float64(4), int32(1), int64(5), object(6)
memory usage: 1.4+ MB
```

c). Missing Value treatment

In 1.2 C we have seen that there is 1430 missing value present in the dataset.

```
Taken_product                                   0.000000
Yearly_avg_view_on_travel_page                  4.940476
preferred_device                                0.000000
total_likes_on_outstation_checkin_given         3.239796
yearly_avg_Outstation_checkins                  0.637755
member_in_family                                0.000000
preferred_location_type                         0.263605
Yearly_avg_comment_on_travel_page               1.751701
total_likes_on_outofstation_checkin_received    0.000000
week_since_last_outstation_checkin              0.000000
following_company_page                          0.875850
montly_avg_comment_on_company_page              0.000000
working_flag                                    0.000000
travelling_network_rating                       0.000000
Adult_flag                                      0.000000
Daily_Avg_mins_spend_on_traveling_page          0.000000
dtype: float64
```

After checking above data, we have found that maximum missing values is less than 5% so we will impute those value.

We will replace missing value in numerical column using median and object column using mode.

```
Taken_product                                   0
Yearly_avg_view_on_travel_page                  0
preferred_device                                0
total_likes_on_outstation_checkin_given         0
yearly_avg_Outstation_checkins                  0
member_in_family                                0
preferred_location_type                         0
Yearly_avg_comment_on_travel_page               0
total_likes_on_outofstation_checkin_received    0
week_since_last_outstation_checkin              0
following_company_page                          0
montly_avg_comment_on_company_page              0
working_flag                                    0
travelling_network_rating                       0
Adult_flag                                      0
Daily_Avg_mins_spend_on_traveling_page          0
dtype: int64
```

d). Outlier treatment

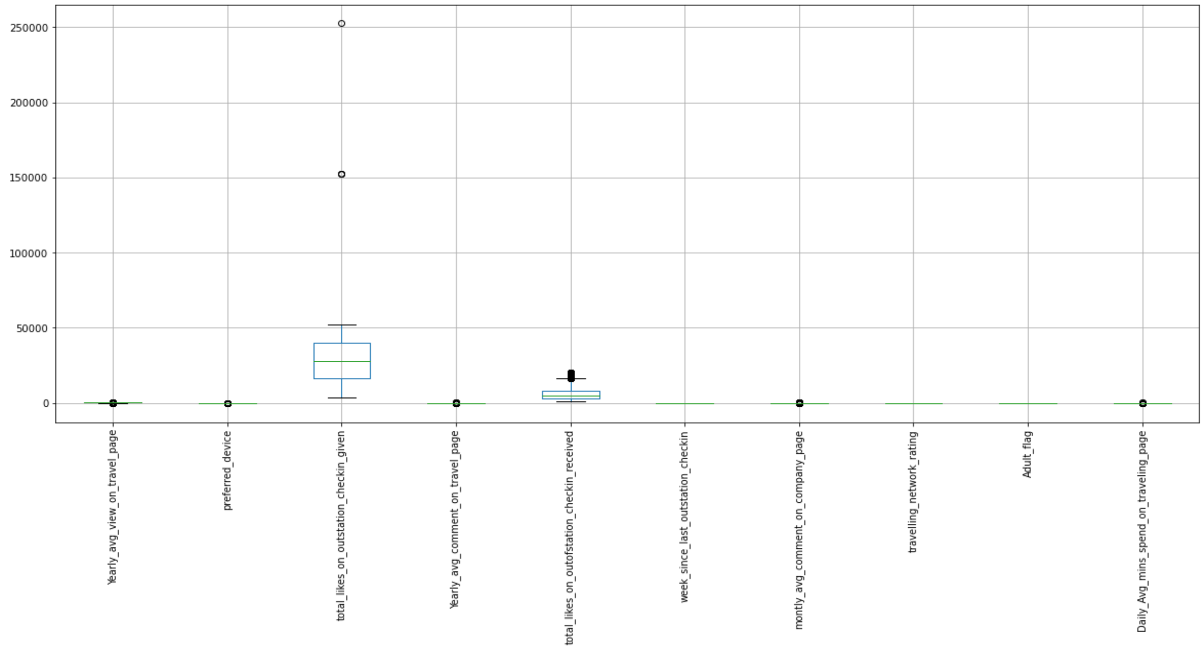


Figure 9. Box Plot to check outliers

Before, the treatment of outliers most of the variables has outliers. In order to treat those outliers in the data we replaced those outliers with the upper limit and lower limit of the particular columns. The values in the column that are greater than the upper limit are replace with its upper limit of that column and values that are lower than lower limit are replaced with the lower limit of that column.

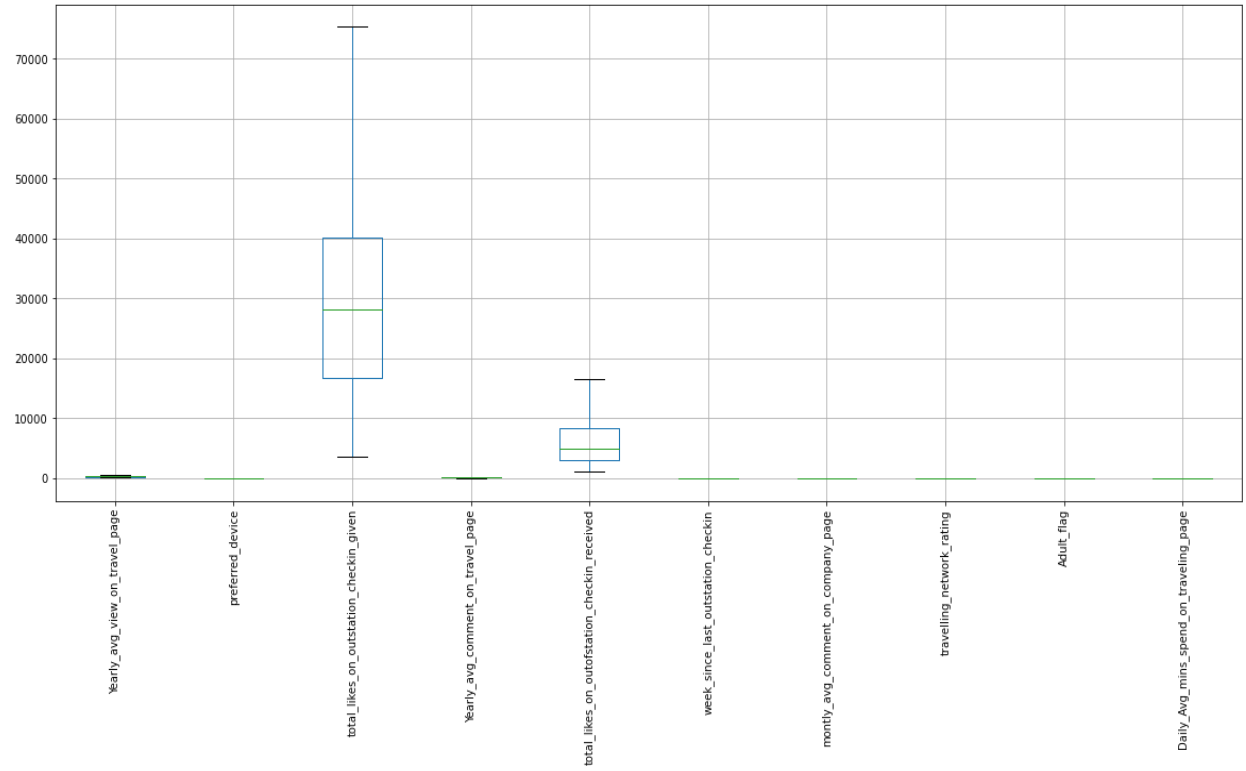


Figure 10. Box Plot after Outlier Treatment

e). Variable Transformation / Addition of New Variables

- The target variable named "Taken product" is transformed where "Yes" is turned to 1 and "No" is turned to 0 with the variable type of float.
- The new variables like "working flag label" and "following company label" are added where "Yes" is turned to 1 and "No" is turned to 0 from variables like "working flag" and "following company label" respectively with the variable type of float.
- Some variables like "member in the family", "yearly average outstation check-in" and "Adult flag" are converted to float variable type.
- "travelling network rating" is converted to category variable.
- "preferred location type label" where Location is arranged from 1-14 with 14 being marked as the most preferred location and 1 as least preferred location from the "preferred location type ".

f). Is the data unbalanced?

```
0.0    9864
1.0    1896
Name: Taken_product, dtype: int64


Normalized Score is
 0.0    0.838776
1.0    0.161224
Name: Taken_product, dtype: float64
```

- The data is highly imbalanced as out of 11760 customers there are 9864 customers that are not interested in purchasing our product which constitutes around 83.9%. This can be treated with the help of SMOTE or K-fold cross validation.
- This shows that there are customers that may purchasing the product of some other company rather than preferring the product of this company.
- It can also be observed that customers are not satisfied with company's service and ae switching to other companies.

1.4 Model Building and Model Validation

Laptop

a). Build various models (You can choose to build models for either or all of descriptive, predictive or prescriptive purposes) and test your predictive model against the test set using various appropriate performance metrics

Scaling

In regression or classification, it is often a good practice to centre the variables so that predictor have a mean of 0. This makes it easier to intercept the intercept term as the expected value of Yi when the predictor values are set to their means. Otherwise, the intercept is interpreted as the expected value of Yi when the predictors are set to 0, which may not be a realistic or interpretable situation. Another valid reason for scaling in regression is when one predictor variable has a very large scale. In that case, the regression coefficients may be on a very small order of magnitude which can be unclear to interpret. The convention that we standardize predictions primarily exists so that the units of the regression coefficients are the same. More often, the dataset contains feature highly varying in magnitudes, units and range. However, most of the machine learning algorithms use Euclidean distance between two data points in their computations, and this can be a potential problem. Also, scaling helps to standardize the independent features present in the data in a fixed range. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values. The features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes. To suppress this effect, we need to bring all features to the same level of magnitudes.

Yes, Scaling is absolutely necessary in this case as we have Variables that carry absolute numbers and we haveVariables that carry percentage. If we have data in different scales, the variables with larger scale will dominate,this is probably not what we want.After scaling there is variance look similar across all data.

Train and Test Split

Before splitting we need to determine the target variable. Hence, the target variable is "Taken_Product"

We will split the data for 70:30 ratio with a random state =1.

Train Test Data Shape

```
X_train (775, 15)
X_test (333, 15)
y_train (775,)
y_test (333,)
```

Logistic Regression Model

Logistic regression is a process of modelling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on.

We split the data into train and test using train_test_split command and fit our linear regression model into the train data and then try to predict the outcome of using the test data. Then we compare the actual against the predicted to calculate the accuracy of the model.

Performance Metrices Basic Logistic Regression Model

Model Score or Accuracy

- Accuracy for Training Data is 0.84
- Accuracy for Test Data is 0.83

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset. It will allow us to visualise the performance of the Logistic Regression Model.

- For Training Data

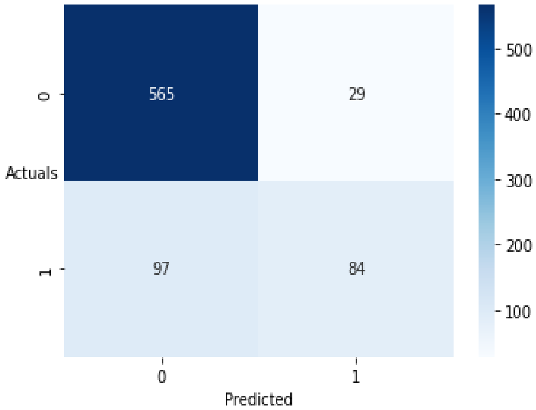


Figure 11. Confusion Matrix for Training Data in Basic Logistic Regression Model for Laptop

➤ For Test Data

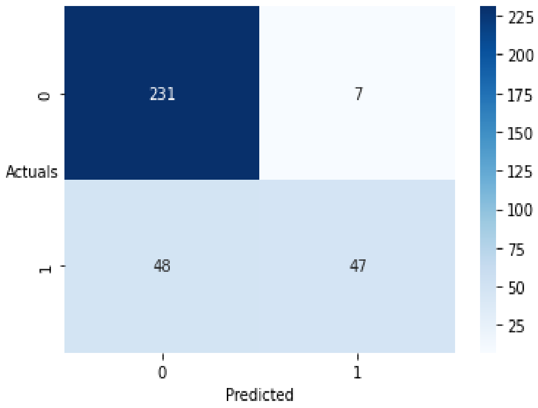


Figure 12. Confusion Matrix for Test Data in Basic Logistic Regression Model for Laptop

Classification Report

➤ For Training Data

The classification report for Logistic Regression training set is

	precision	recall	f1-score	support
0.0	0.85	0.95	0.90	594
1.0	0.74	0.46	0.57	181
accuracy			0.84	775
macro avg	0.80	0.71	0.74	775
weighted avg	0.83	0.84	0.82	775

Table 7. Classification Report for Training Data in Basic Logistic Regression Model for Laptop

➤ For Test Data

The classification report for Logistic Regression testing set is

	precision	recall	f1-score	support
0.0	0.83	0.97	0.89	238
1.0	0.87	0.49	0.63	95
accuracy			0.83	333
macro avg	0.85	0.73	0.76	333
weighted avg	0.84	0.83	0.82	333

Table 8. Classification Report for Test Data in Basic Logistic Regression Model for Laptop

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

The AUC score for Logistic Regression training set is: 0.817

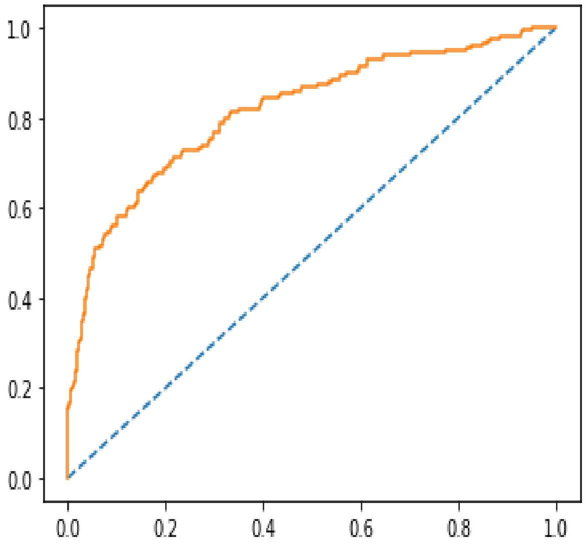


Figure 13. ROC for Training Data in Basic Logistic Regression Model for Laptop

- For Test Data

The AUC score for Logistic Regression testing set is: 0.865

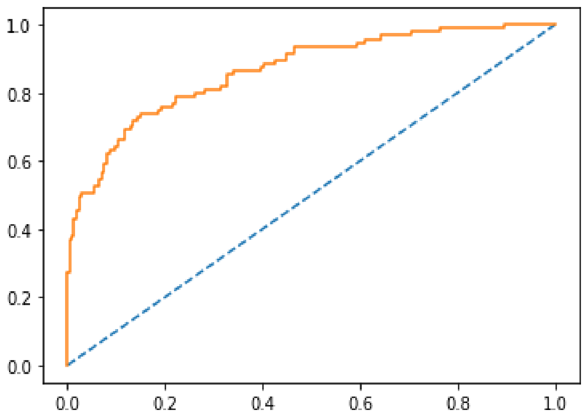


Figure 14. ROC for Test Data in Basic Logistic Regression Model for Laptop

KNN Model

KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

Performance Metrices Basic KNN Model

<u>Model Score or Accuracy</u>

- Accuracy for Training Data is 0.96
- Accuracy for Test Data is 0.88

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

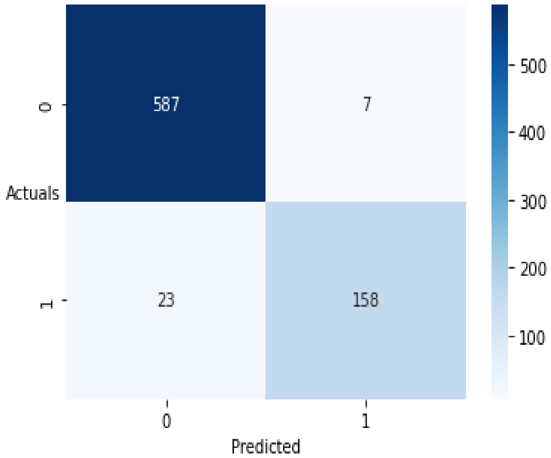


Figure 15. Confusion Matrix for Training Data in Basic KNN Model for Laptop

- For Test Data

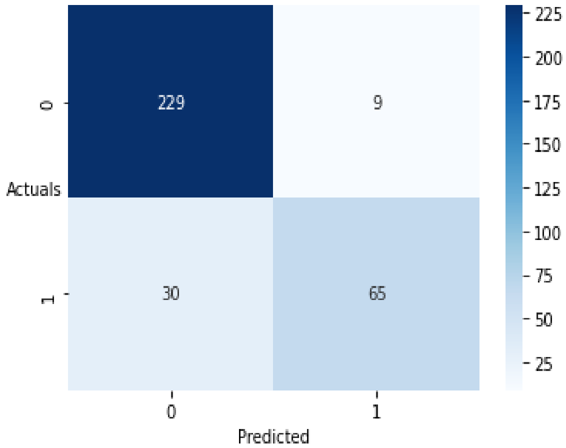


Figure 16. Confusion Matrix for Test Data in Basic KNN Model for Laptop

Classification Report

- For Training Data

The classification report for KNN set is

	precision	recall	f1-score	support
0.0	0.96	0.99	0.98	594
1.0	0.96	0.87	0.91	181
accuracy			0.96	775
macro avg	0.96	0.93	0.94	775
weighted avg	0.96	0.96	0.96	775

Table 9. Classification Report for Training Data in Basic KNN Model for Laptop

➢ For Test Data

The classification report for KNN testing set is

	precision	recall	f1-score	support
0.0	0.88	0.96	0.92	238
1.0	0.88	0.68	0.77	95
accuracy			0.88	333
macro avg	0.88	0.82	0.85	333
weighted avg	0.88	0.88	0.88	333

Table 10. Classification Report for Test Data in Basic KNN Model for Laptop

<u>ROC and AUC</u>

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

➢ For Training Data

The AUC score for KNN training set is: 0.991

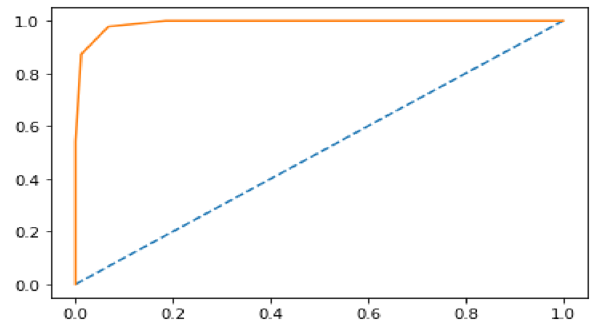


Figure 17. ROC for Training Data in Basic KNN Model for Laptop

➢ For Test Data

The AUC score for KNN testing set is: 0.937

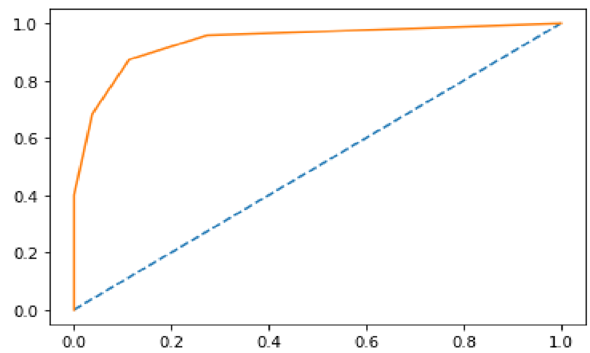


Figure 18. ROC for Test Data in Basic KNN Model for Laptop

Naïve Bayes Model

Naïve Bayes is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Performance Metrices Basic Naïve Bayes Model

Model Score or Accuracy

- Accuracy for Training Data is 0.83
- Accuracy for Test Data is 0.84

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

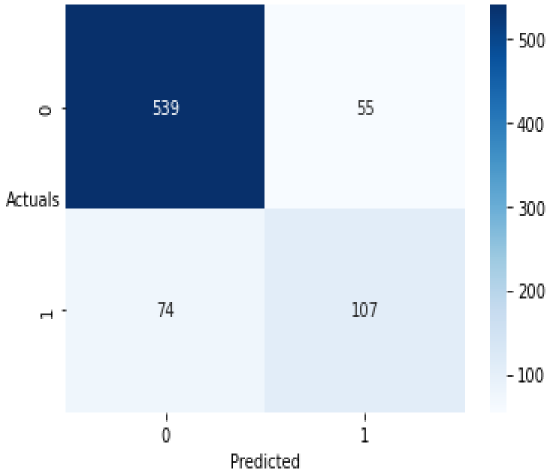


Figure 19. Confusion Matrix for Training Data in Basic Naive Bayes Model for Laptop

- For Test Data

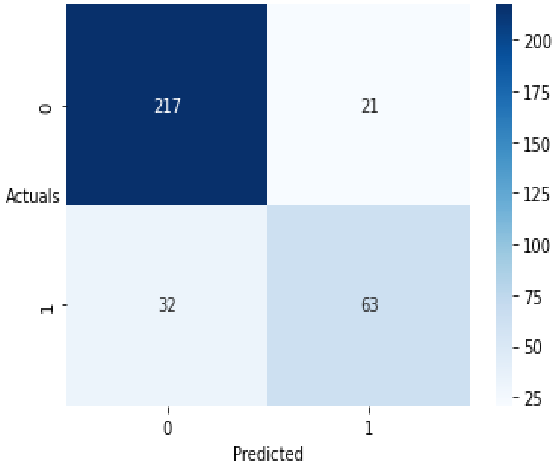


Figure 20. Confusion Matrix for Test Data in Basic Naive Bayes Model for Laptop

Classification Report

- For Training Data

he classification report for Naive Bayes Model set is

	precision	recall	f1-score	support
0.0	0.88	0.91	0.89	594
1.0	0.66	0.59	0.62	181
accuracy			0.83	775
macro avg	0.77	0.75	0.76	775
eighted avg	0.83	0.83	0.83	775

Table 11. Classification Report for Training Data in Basic Naive Bayes Model for Laptop

- For Test Data

The classification report for Naive bayes Model testing set is

	precision	recall	f1-score	support
0.0	0.87	0.91	0.89	238
1.0	0.75	0.66	0.70	95
accuracy			0.84	333
macro avg	0.81	0.79	0.80	333
weighted avg	0.84	0.84	0.84	333

Table 12. Classification Report for Test Data in Basic Naive Bayes Model for Laptop

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

The AUC score for Naive Bayes training set is: 0.809

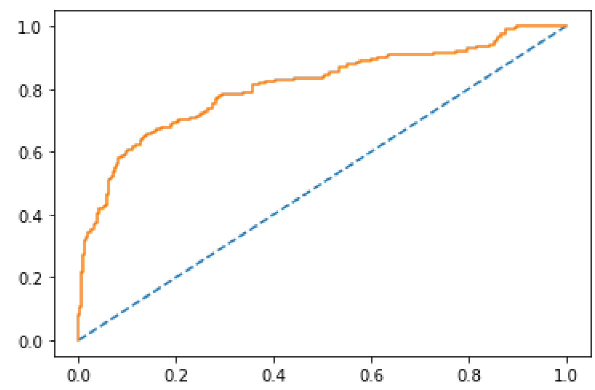


Figure 21. ROC for Training Data in Basic Naive Bayes Model for Laptop

➢ For Test Data

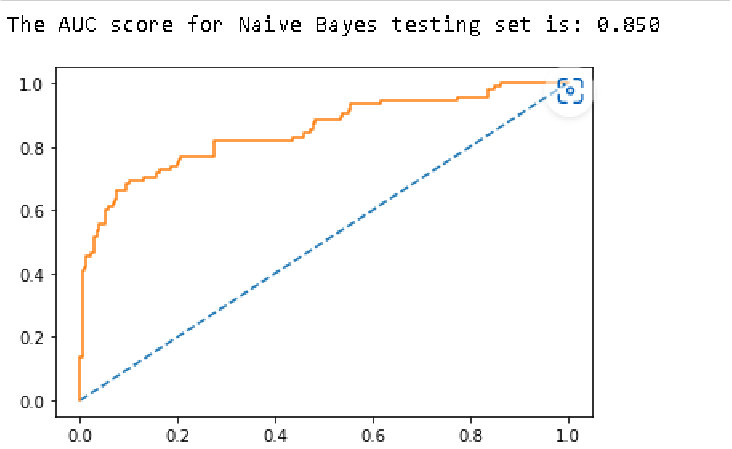


Figure 22. ROC for Test Data in Basic Naive Bayes Model for Laptop

Bagging

Bagging is designed to improve the performance of existing ML algorithms used in statistical classification or regression. It is most used with tree-based algorithms. It is a parallel method.

```
BaggingClassifier(base_estimator=RandomForestClassifier(),
n_estimators=100,random_state=1)
```

Performance Metrices Basic Bagging

Model Score or Accuracy

➢ Accuracy for Training Data is 1.0
➢ Accuracy for Test Data is 0.94

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

➢ For Training Data

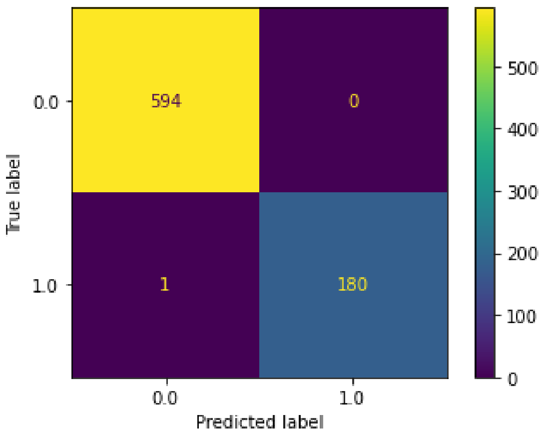


Figure 23. Confusion Matrix for Training Data in Basic Bagging for Laptop

➢ For Test Data

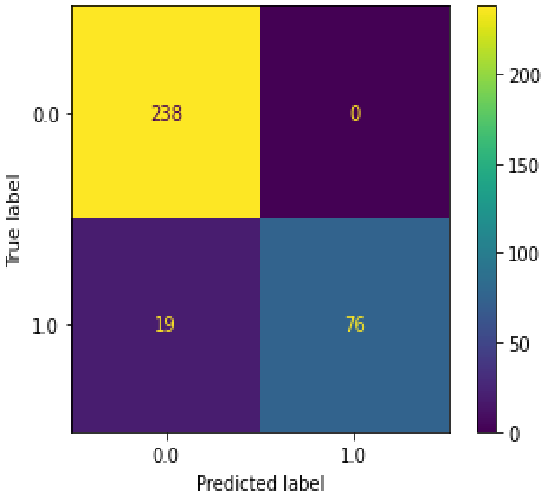


Figure 24. Confusion Matrix for Test Data in Basic Bagging for Laptop

Classification Report

➢ For Training Data

0.9987096774193548

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	594
1.0	1.00	0.99	1.00	181
accuracy			1.00	775
macro avg	1.00	1.00	1.00	775
weighted avg	1.00	1.00	1.00	775

Table 13. Classification Report for Training Data in Basic Bagging for Laptop

➢ For Test Data

0.9429429429429429

	precision	recall	f1-score	support
0.0	0.93	1.00	0.96	238
1.0	1.00	0.80	0.89	95
accuracy			0.94	333
macro avg	0.96	0.90	0.93	333
weighted avg	0.95	0.94	0.94	333

Table 14.Classification Report for Test Data in Basic Bagging for Laptop

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

➢ For Training Data

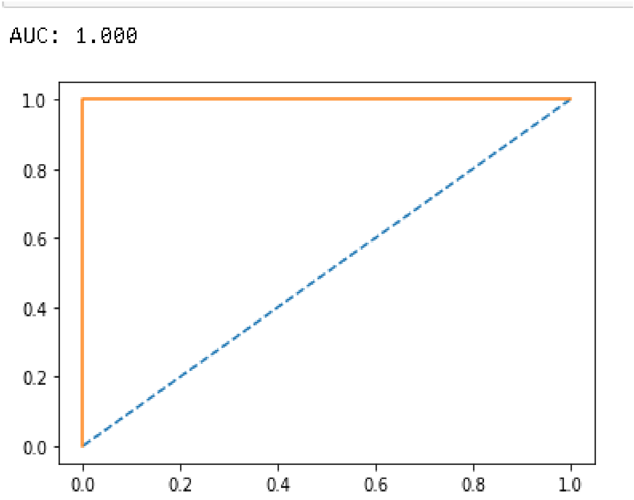


Figure 25. ROC for Training Data in Basic Bagging for Laptop

➢ For Test Data

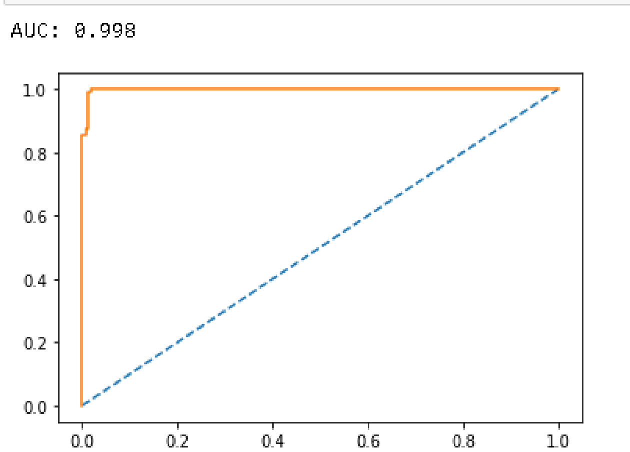


Figure 26. ROC for Test Data in Basic Bagging for Laptop

ADA Boosting

This model is used to increase the efficiency of binary classifiers, but now used to improve multiclass classifiers as well. ADA boosting can be applied on top of any classifier method to learn from its issues and bring about a more accurate model and this it is called “best out of the box classifier”

```
AdaBoostClassifier(n_estimators=100, random_state=1)
```

Performance Metrices Basic Ada Boosting

Model Score or Accuracy

- ➢ Accuracy for Training Data is 0.95
- ➢ Accuracy for Test Data is 0.87

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

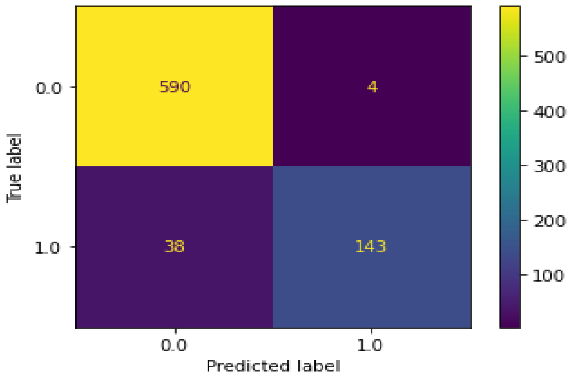


Figure 27.Confusion Matrix for Training Data in Basic Ada Boosting for Laptop

- For Test Data

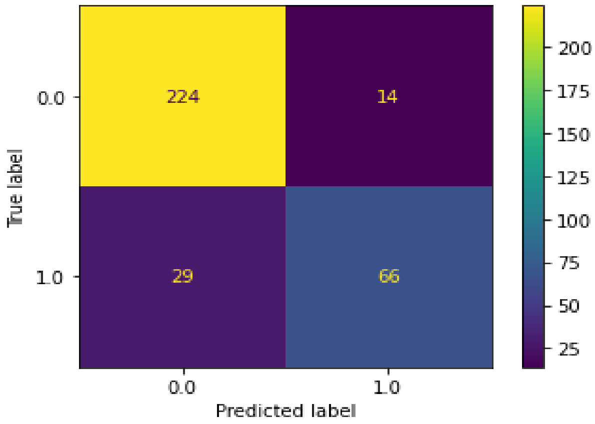


Figure 28. Confusion Matrix for Test Data in Basic Ada Boosting for Laptop

Classification Report

- For Training Data

0.9458064516129032

	precision	recall	f1-score	support
0.0	0.94	0.99	0.97	594
1.0	0.97	0.79	0.87	181
accuracy			0.95	775
macro avg	0.96	0.89	0.92	775
weighted avg	0.95	0.95	0.94	775

Table 15. Classification Report for Training Data in Basic Ada Boosting for Laptop

➢ For Test Data

0.8708708708708709

	precision	recall	f1-score	support
0.0	0.89	0.94	0.91	238
1.0	0.82	0.69	0.75	95
accuracy			0.87	333
macro avg	0.86	0.82	0.83	333
weighted avg	0.87	0.87	0.87	333

Table 16. Classification Report for Test Data in Basic Ada Boosting for Laptop

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

➢ For Training Data

AUC: 0.986

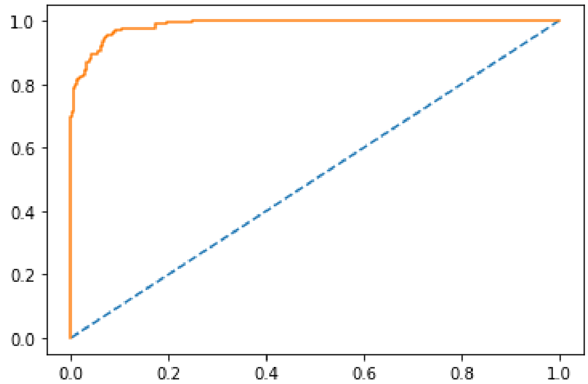


Figure 29. ROC for Training Data in Basic Ada Boosting for Laptop

➢ For Test Data

AUC: 0.926

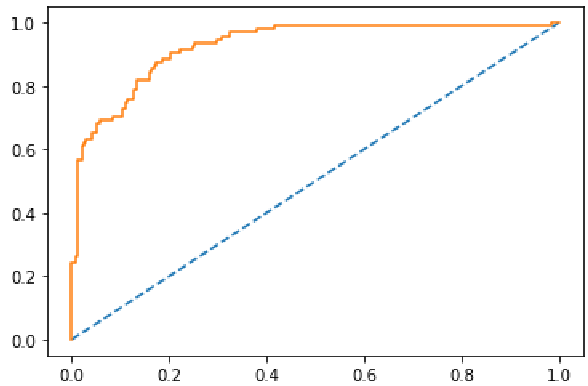


Figure 30. ROC for Test Data in Basic Ada Boosting for Laptop

Gradient Boosting

This model is just like the ADA Boosting works by sequentially adding the misidentified predictors and under-fitted predictions to the ensemble, ensuring the errors identified previously are corrected. The major difference lies in what it does with the mis-identified values of the previous weak learner.

Performance Metrices Basic Gradient Boosting

Model Score or Accuracy

- Accuracy for Training Data is 0.99
- Accuracy for Test Data is 0.96

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

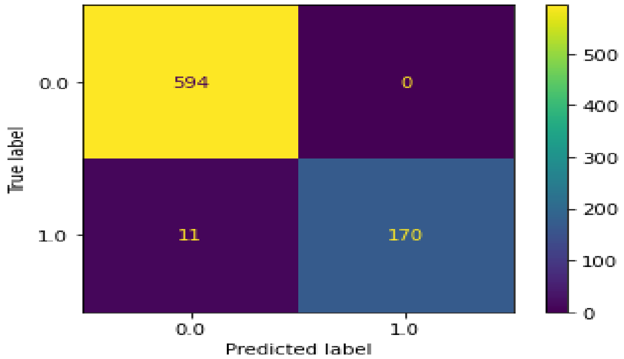


Figure 31.Confusion Matrix for Training Data in Basic Gradient Boosting for Laptop

- For Test Data

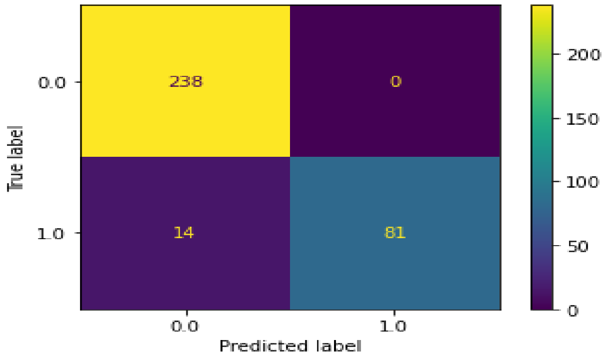


Figure 32. Confusion Matrix for Test Data in Basic Gradient Boosting for Laptop

Classification Report

- For Training Data

0.9858064516129033

	precision	recall	f1-score	support
0.0	0.98	1.00	0.99	594
1.0	1.00	0.94	0.97	181
accuracy			0.99	775
macro avg	0.99	0.97	0.98	775
weighted avg	0.99	0.99	0.99	775

Table 17. Classification Report for Training Data in Basic Gradient Boosting for Laptop

- For Test Data

0.957957957957958

	precision	recall	f1-score	support
0.0	0.94	1.00	0.97	238
1.0	1.00	0.85	0.92	95
accuracy			0.96	333
macro avg	0.97	0.93	0.95	333
weighted avg	0.96	0.96	0.96	333

Table 18. Classification Report for Test Data in Basic Gradient Boosting for Laptop

<u>ROC and AUC</u>

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

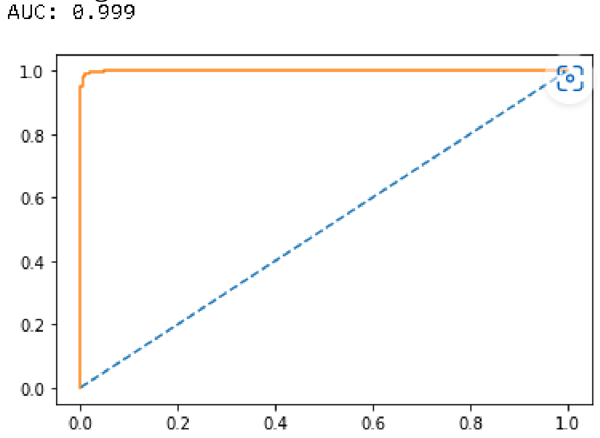


Figure 33. ROC for Training Data in Basic Gradient Boosting for Laptop

- For Test Data

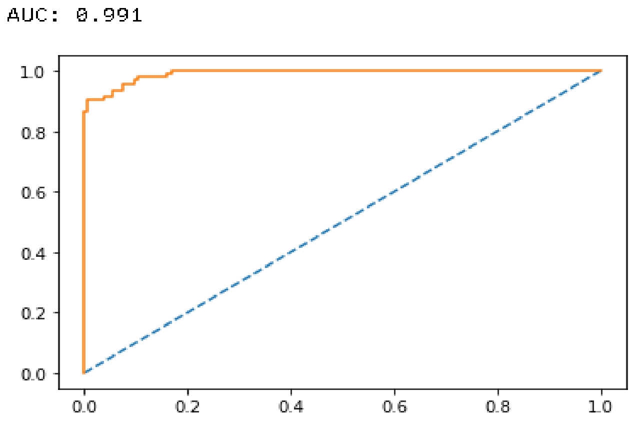


Figure 34. ROC for Test Data in Basic Gradient Boosting for Laptop

b). Interpretation of the model(s)

Basic Model		Accuracy		Precision		Recall		F1 Score		AUC	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Logistic Regression	No Taken Product	0.84	0.83	0.85	0.83	0.95	0.97	0.9	0.89	0.82	0.87
	Yes Taken Product			0.74	0.87	0.46	0.49	0.57	0.63		
KNN	No Taken Product	0.96	0.88	0.96	0.88	0.99	0.96	0.98	0.92	0.99	0.94
	Yes Taken Product			0.96	0.88	0.87	0.68	0.91	0.77		
Naïve Bayes	No Taken Product	0.83	0.84	0.88	0.87	0.91	0.91	0.89	0.89	0.81	0.85
	Yes Taken Product			0.66	0.75	0.59	0.66	0.62	0.7		
Bagging	No Taken Product	1	0.94	1	0.93	1	1	1	0.96	1	0.99
	Yes Taken Product			1	1	0.99	0.8	1	0.89		
Ada Boosting	No Taken Product	0.95	0.87	0.94	0.89	0.99	0.94	0.97	0.91	0.99	0.93
	Yes Taken Product			0.97	0.82	0.79	0.69	0.87	0.75		
Gradient Boosting	No Taken Product	0.99	0.96	0.98	0.94	1	1	0.99	0.97	0.99	0.99
	Yes Taken Product			1	1	0.94	0.85	0.97	0.92		

Table 19. Basic Models Comparisons for Laptop

- According to problem we will focus on the Customer who have taken the product.
- Logistic Regression model and KNN model provides accuracy of 84% and 83% on train set and 96% and 88% on test set respectively. In Logistic regression and KNN it can be observed that the accuracy for test set decreases.
- Naïve Bayes model have provided a decent accuracy on Training set that is 83% and applying the models to testing set, we see that the accuracy has improved a bit that is 84%
- The desired metric for the problem is Precision which is not good for the Logistic Regression and Naïve Bayes. In case of KNN for Precision is good for Train but when applied for test set it declined a bit.
- Bagging model has high score for all parameters in Training data but it has not performed well in Test data and hence it is overfitted model
- Gradient Boosting model is better than ADA model as it has high score in Accuracy, Precision, Recall, F1 score and AUC.

c) Ensemble modelling, wherever applicable and Any other model tuning measures (if applicable)

Model Tuning

Tuning is process of maximizing a model's performance without overfitting or creating too high of a variance. In ML, this is accomplished by selecting appropriate "hyper-parameters".

Logistic Regression Model – Grid Search

We split the data into train and test using train_test_split command and fit our linear regression model into the train data and then try to predict the outcome of using the test data. Then we compare the actual against the predicted to calculate the accuracy of the model. We will hyper tune the parameters that would enhance the outcome of the model.

```
GridSearchCV(cv=5, estimator=LogisticRegression(),
             param_grid={'C': [0.001, 0.009, 0.01, 0.09, 1, 5, 10, 25],
                         'penalty': ['l1', 'l2'], 'solver': ['newton-
cg']})

Best_Estimator LogisticRegression(C=1, solver='newton-cg')
```

Performance Metrices Logistic Regression Grid Search

Model Score or Accuracy

- Accuracy for Training Data is 0.84
- Accuracy for Test Data is 0.83

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset. It will allow us to visualise the performance of the Logistic Regression Model.

- For Training Data

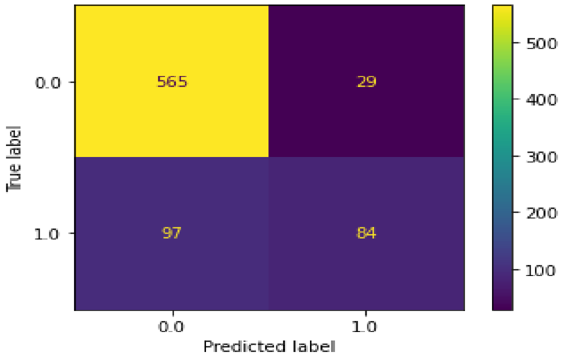


Figure 35. Confusion Matrix for Training Data in Logistic Regression Grid Search for Laptop

➢ For Test Data

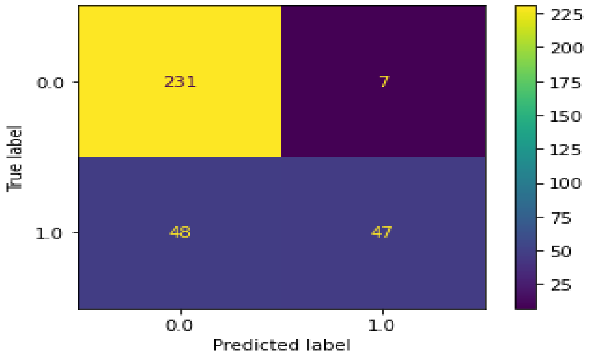


Figure 36. Confusion Matrix for Test Data in Logistic Regression Grid Search for Laptop

Classification Report

➢ For Training Data

0.8374193548387097

	precision	recall	f1-score	support
0.0	0.85	0.95	0.90	594
1.0	0.74	0.46	0.57	181
accuracy			0.84	775
macro avg	0.80	0.71	0.74	775
weighted avg	0.83	0.84	0.82	775

Table 20. Classification Report for Training Data in Logistic Regression Grid Search for Laptop

➢ For Test Data

0.8348348348348348

	precision	recall	f1-score	support
0.0	0.83	0.97	0.89	238
1.0	0.87	0.49	0.63	95
accuracy			0.83	333
macro avg	0.85	0.73	0.76	333
weighted avg	0.84	0.83	0.82	333

Table 21. Classification Report for Test Data in Logistic Regression Grid Search for Laptop

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

➢ For Training Data

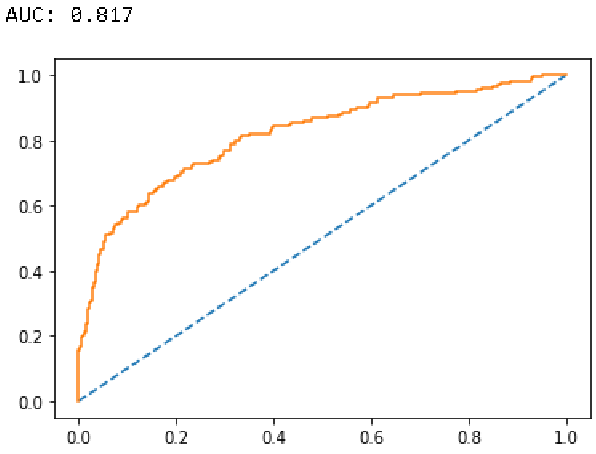


Figure 37. ROC for Training Data in Logistic Regression Grid Search for Laptop

- For Test Data

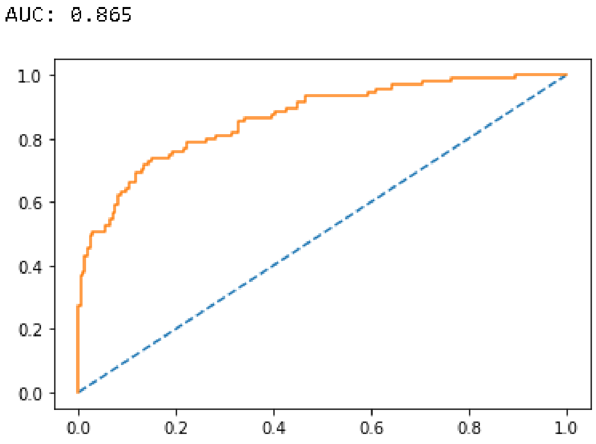


Figure 38. ROC for Test Data in Logistic Regression Grid Search for Laptop

KNN – Grid Search

We split the data into train and test using train_test_split command and fit our KNN regression model into the train data and then try to predict the outcome of using the test data. Then we compare the actual against the predicted to calculate the accuracy of the model. We will hyper tune the parameters that would enhance the outcome of the model

```
GridSearchCV(cv=5, estimator=KNeighborsClassifier(),
             param_grid={'leaf_size': [20, 30, 50], 'n_neighbors': [10, 
20, 30],'p': [1, 2]})
```

Performance Metrices Basic KNN Grid Search

Model Score or Accuracy

- Accuracy for Training Data is 0.87
- Accuracy for Test Data is 0.80

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

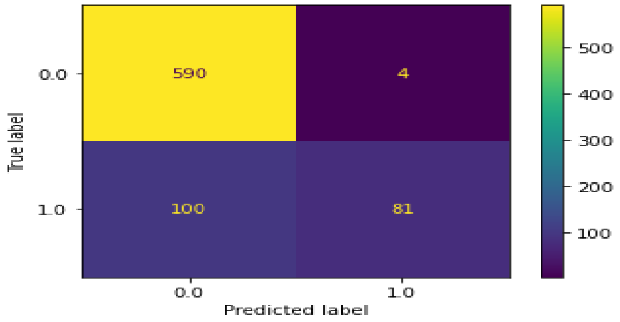


Figure 39. Confusion Matrix for Training Data in KNN Grid Search for Laptop

- For Test Data

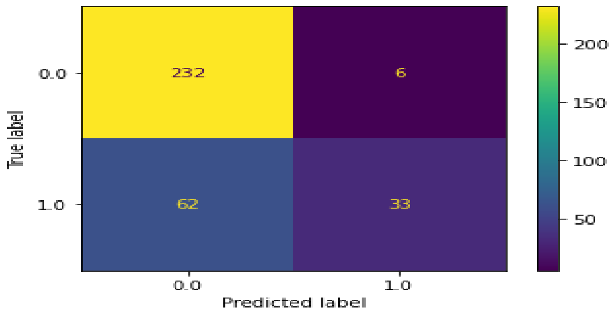


Figure 40. Confusion Matrix for Test Data in KNN Grid Search for Laptop

Classification Report

- For Training Data

0.8658064516129033

	precision	recall	f1-score	support
0.0	0.86	0.99	0.92	594
1.0	0.95	0.45	0.61	181
accuracy			0.87	775
macro avg	0.90	0.72	0.76	775
weighted avg	0.88	0.87	0.85	775

Table 22. Classification Report for Training Data in KNN Grid Search for Laptop

- F or Test Data

0.7957957957957958

	precision	recall	f1-score	support
0.0	0.79	0.97	0.87	238
1.0	0.85	0.35	0.49	95
accuracy			0.80	333
macro avg	0.82	0.66	0.68	333
weighted avg	0.81	0.80	0.76	333

Table 23. Classification Report for Test Data in KNN Grid Search for Laptop

<u>ROC and AUC</u>

➢ For Training Data

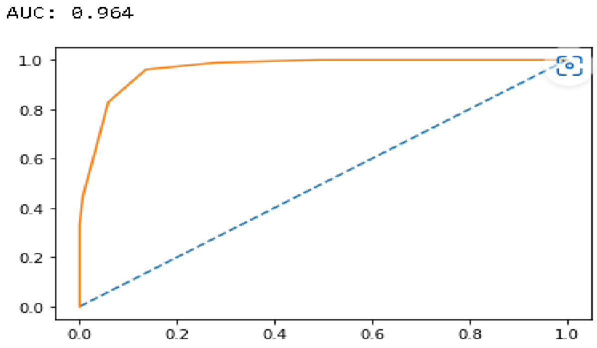


Figure 41. ROC for Training Data in KNN Grid Search for Laptop

➢ For Test Data

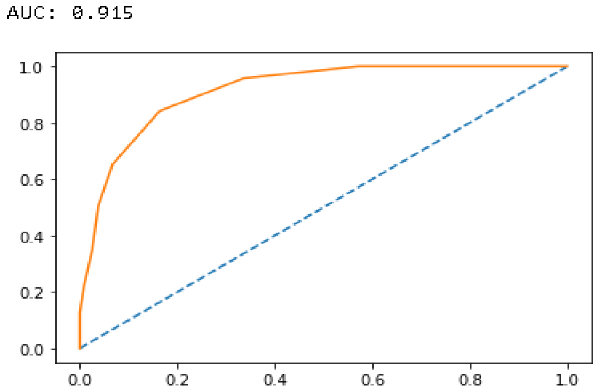


Figure 42. ROC for Test Data in KNN Grid Search for Laptop

Naïve Bayes – Grid Search

We split the data into train and test using train_test_split command and fit our Naïve Bayes model into the train data and then try to predict the outcome of using the test data. Then we compare the actual against the predicted to calculate the accuracy of the model. We will hyper tune the parameters that would enhance the outcome of the model.

```
GridSearchCV(cv=5, estimator=GaussianNB(), n_jobs=1,
                param_grid={'var_smoothing': [1e-08, 1e-07, 1e-06, 1e-05,
0.0001]},verbose=2)
```

Performance Metrices Naïve Bayes Grid Search

<u>Model Score or Accuracy</u>

➢ Accuracy for Training Data is 0.83
➢ Accuracy for Test Data is 0.84

<u>Confusion Matrix</u>

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

➢ For Training Data

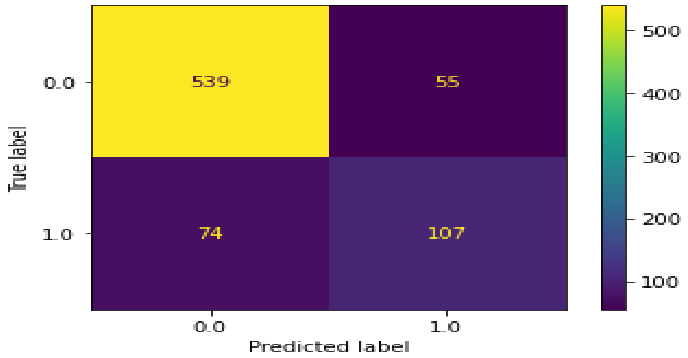


Figure 43. Confusion Matrix for Training Data in Naive Bayes Grid Search for Laptop

➢ For Test Data

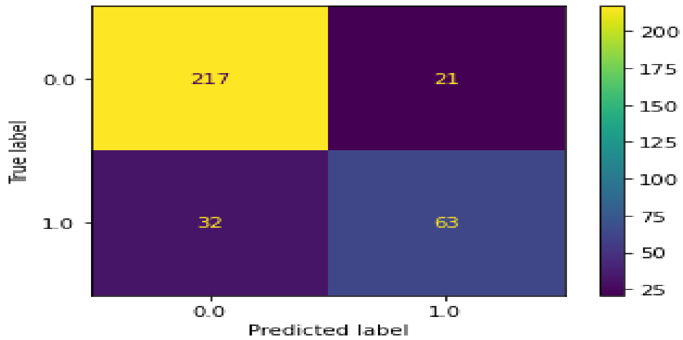


Figure 44. Confusion Matrix for Test Data in Naive Bayes Grid Search for Laptop

Classification Report

➢ For Training Data

0.8335483870967741

	precision	recall	f1-score	support
0.0	0.88	0.91	0.89	594
1.0	0.66	0.59	0.62	181
accuracy			0.83	775
macro avg	0.77	0.75	0.76	775
weighted avg	0.83	0.83	0.83	775

Table 23. Classification Report for Training Data in Naive Bayes Grid Search for Laptop

➢ For Test Data

0.8408408408408409

	precision	recall	f1-score	support
0.0	0.87	0.91	0.89	238
1.0	0.75	0.66	0.70	95
accuracy			0.84	333
macro avg	0.81	0.79	0.80	333
weighted avg	0.84	0.84	0.84	333

Table 24. Classification Report for Test Data in Naive Bayes Grid Search for Laptop

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

```
AUC: 0.809
```

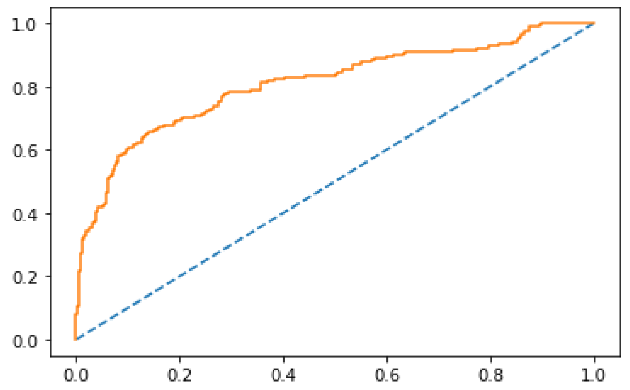


Figure 45. ROC for Training Data in Naive Bayes Grid Search for Laptop

- For Test Data

```
AUC: 0.850
```

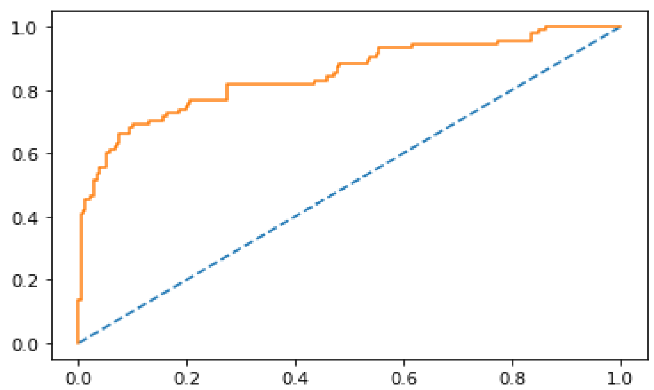


Figure 46. ROC for Test Data in Naive Bayes Grid Search for Laptop

Bagging – Grid Search

Bagging is an ensemble technique. Ensemble techniques are ML techniques that combine several base models to get an optimal model. Bagging is designed to improve the performance of existing ML algorithms used in statistical classification or regression. It is most used with tree-based algorithms. It is a parallel method.

```
GridSearchCV(cv=3,
```

```
estimator=BaggingClassifier(base_estimator=RandomForestClassifier(),n_e
stimators=100, random_state=1),
                 param_grid={'bootstrap': [True, False], 'max_features':
[1, 2, 4],'max_samples': [0.5, 1.0]})
```

Performance Metrices Bagging Grid Search

Model Score or Accuracy

- Accuracy for Training Data is 1.0
- Accuracy for Test Data is 0.89

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

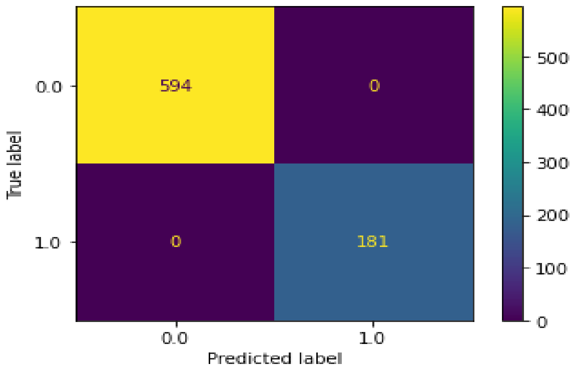


Figure 47. Confusion Matrix for Training Data in Bagging Grid Search for Laptop

- For Test Data

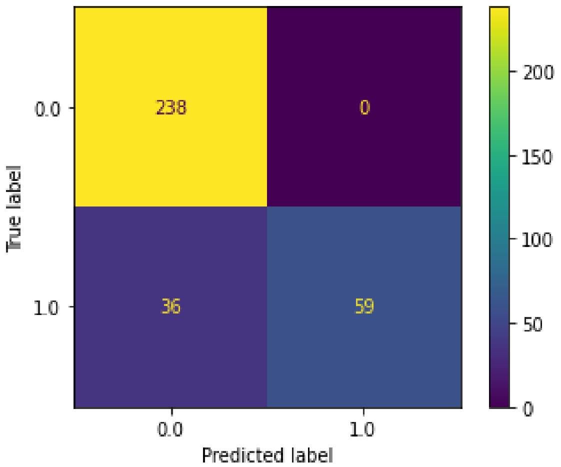


Figure 48. Confusion Matrix for Test Data in Bagging Grid Search for Laptop

Classification Report

- For Training Data

1.0

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	594
1.0	1.00	1.00	1.00	181
accuracy			1.00	775
macro avg	1.00	1.00	1.00	775
weighted avg	1.00	1.00	1.00	775

Table 25. Classification Report for Training Data in Bagging Grid Search for Laptop

- For Test Data

0.8918918918918919

	precision	recall	f1-score	support
0.0	0.87	1.00	0.93	238
1.0	1.00	0.62	0.77	95
accuracy			0.89	333
macro avg	0.93	0.81	0.85	333
weighted avg	0.91	0.89	0.88	333

Table 26. Classification Report for Test Data in Bagging Grid Search for Laptop

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

AUC: 1.000

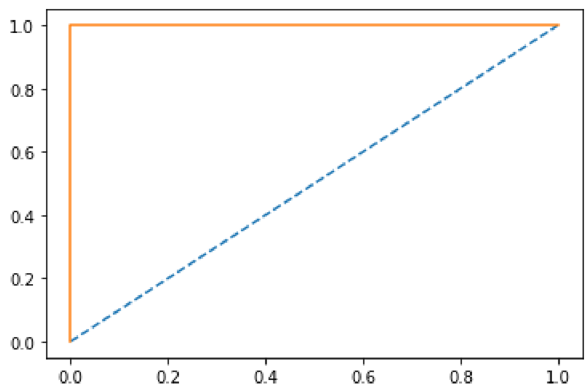


Figure 49. ROC for Training Data in Bagging Grid Search for Laptop

➢ For Test Data

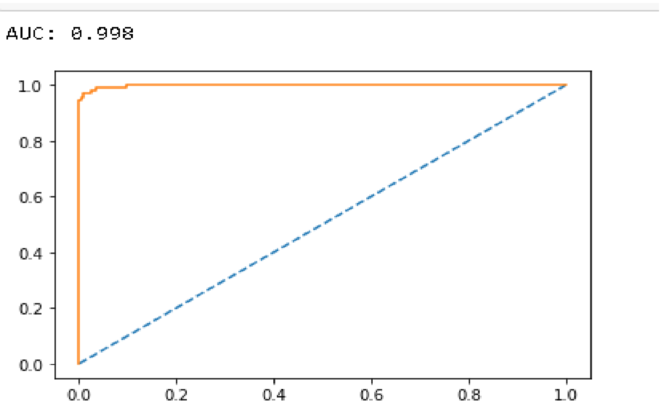


Figure 50. ROC for Test Data in Bagging Grid Search for Laptop

ADA Boosting – Grid Search

This model is used to increase the efficiency of binary classifiers, but now used to improve multiclass classifiers as well. ADA boosting can be applied on top of any classifier method to learn from its issues and bring about a more accurate model and this it is called "best out of the box classifier"

```
GridSearchCV(cv=3, estimator=AdaBoostClassifier(), n_jobs=1,
             param_grid={'learning_rate': [0.001, 0.01, 0.1],
                         'n_estimators': [500, 1000, 2000]})
```

Performance Metrices Ada Boosting Grid Search

Model Score or Accuracy

Accuracy for Training Data is 0.95
Accuracy for Test Data is 0.88

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

➢ For Training Data

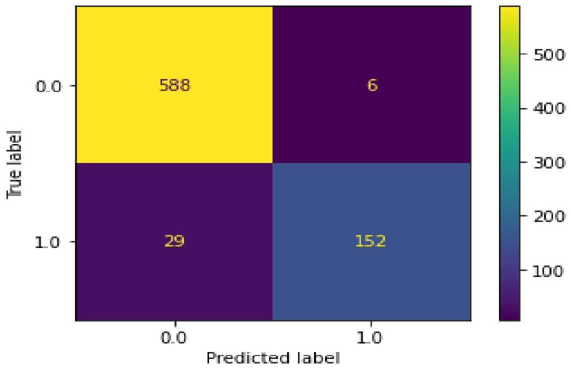


Figure 51. Confusion Matrix for Training Data in Ada Boosting Grid Search for Laptop

- For Test Data

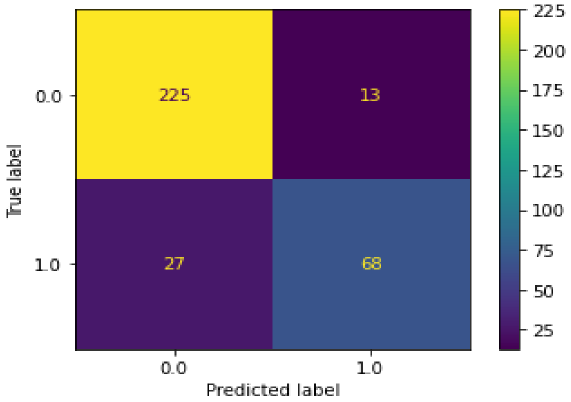


Figure 52. Confusion Matrix for Test Data in Ada Boosting Grid Search for Laptop

Classification Report

- For Training Data

0.9548387096774194

	precision	recall	f1-score	support
0.0	0.95	0.99	0.97	594
1.0	0.96	0.84	0.90	181
accuracy			0.95	775
macro avg	0.96	0.91	0.93	775
weighted avg	0.96	0.95	0.95	775

Table 27. Classification Report for Training Data in Ada Boosting Grid Search for Laptop

- For Test Data

0.8798798798798799

	precision	recall	f1-score	support
0.0	0.89	0.95	0.92	238
1.0	0.84	0.72	0.77	95
accuracy			0.88	333
macro avg	0.87	0.83	0.85	333
weighted avg	0.88	0.88	0.88	333

Table 28. Classification Report for Test Data in Ada Boosting Grid Search for Laptop

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

AUC: 0.992

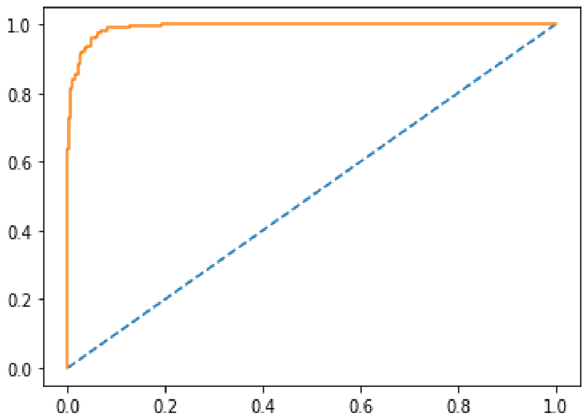


Figure 53. ROC for Training Data in Ada Boosting Grid Search for Laptop

- For Test Data

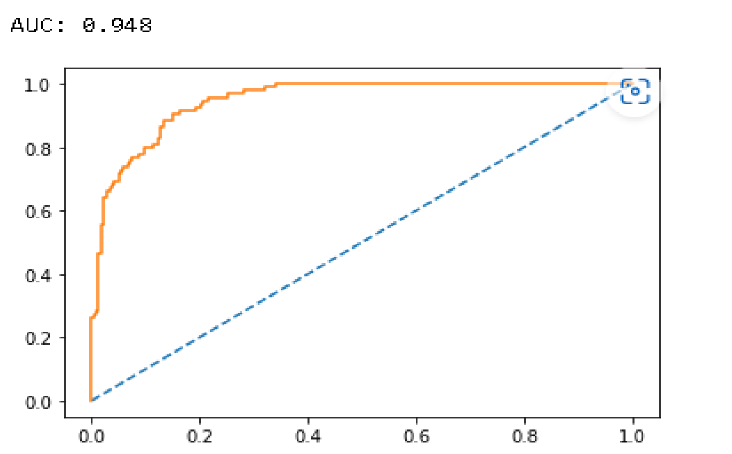


Figure 54. ROC for Test Data in Ada Boosting Grid Search for Laptop

Gradient Boosting – Grid Search

This model is just like the ADA Boosting works by sequentially adding the misidentified predictors and under-fitted predictions to the ensemble, ensuring the errors identified previously are corrected. The major difference lies in what it does with the mis-identified values of the previous weak learner.

```
GridSearchCV(cv=3, estimator=GradientBoostingClassifier(),
             param_grid={'n_estimators': range(1000, 2000, 3000)})
```

Performance Metrices Gradient Boosting Grid Search

Model Score or Accuracy

- ➢ Accuracy for Training Data is 1.0
- ➢ Accuracy for Test Data is 0.99

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- ➢ For Training Data

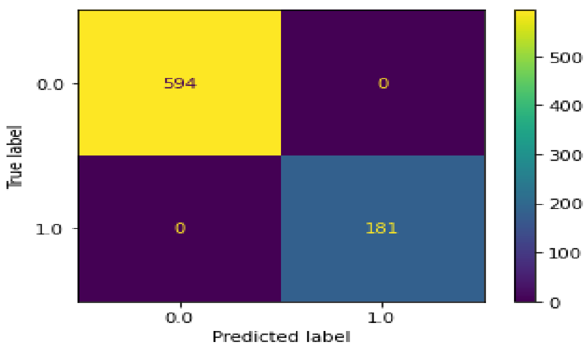


Figure 55. Confusion Matrix for Training Data in Gradient Boosting Grid Search for Laptop

- ➢ For Test Data

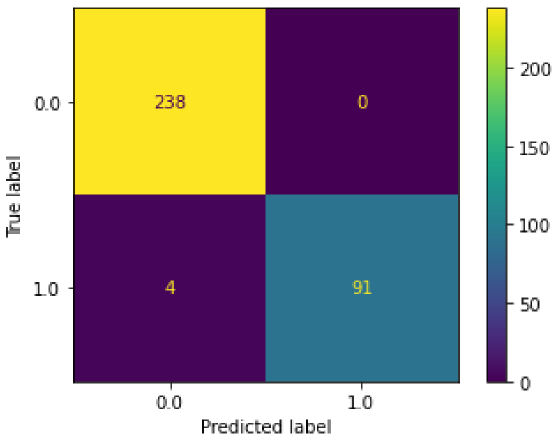


Figure 56. Confusion Matrix for Test Data in Gradient Boosting Grid Search for Laptop

Classification Report

- For Training Data

1.0

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	594
1.0	1.00	1.00	1.00	181
accuracy			1.00	775
macro avg	1.00	1.00	1.00	775
weighted avg	1.00	1.00	1.00	775

Table 29. Classification Report for Training Data in Gradient Boosting Grid Search for Laptop

- For Test Data

0.987987987987988

	precision	recall	f1-score	support
0.0	0.98	1.00	0.99	238
1.0	1.00	0.96	0.98	95
accuracy			0.99	333
macro avg	0.99	0.98	0.99	333
weighted avg	0.99	0.99	0.99	333

Table 30. Classification Report for Test Data in Gradient Boosting Grid Search for Laptop

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

➢ For Training Data

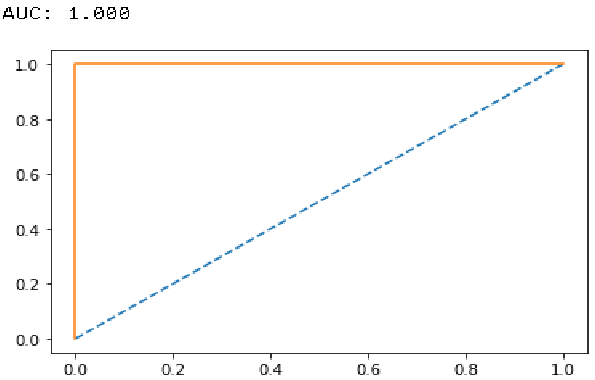


Figure 57. ROC for Training Data in Gradient Boosting Grid Search for Laptop

➢ For Test Data

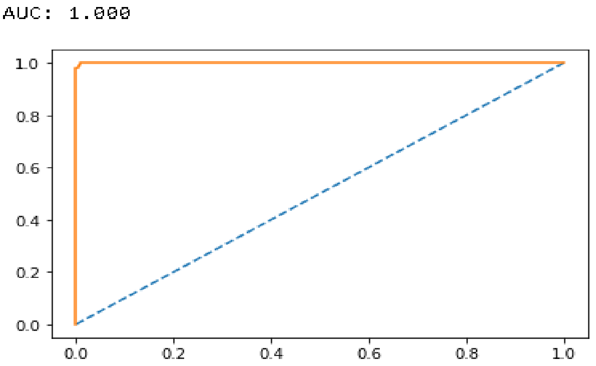


Figure 58. ROC for Test Data in Gradient Boosting Grid Search for Laptop

SMOTE

SMOTE (synthetic minority oversampling technique) is one of the most commonly used oversampling methods to solve the imbalance problem.
It aims to balance class distribution by randomly increasing minority class examples by replicating them.

SMOTE synthesises new minority instances between existing minority instances. It generates the virtual training records by linear interpolation for the minority class. These synthetic training records are generated by randomly selecting one or more of the k-nearest neighbors for each example in the minority class. After the oversampling process, the data is reconstructed and several classification models can be applied for the processed data.

```
New data shape after SMOTE (1188, 15)
```

Logistic Regression Model – SMOTE

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

Performance Metrices Logistic Regression SMOTE

Model Score or Accuracy

- ➢ Accuracy for Training Data is 0.74
- ➢ Accuracy for Test Data is 0.73

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset. It will allow us to visualise the performance of the Logistic Regression Model.

- ➢ For Training Data

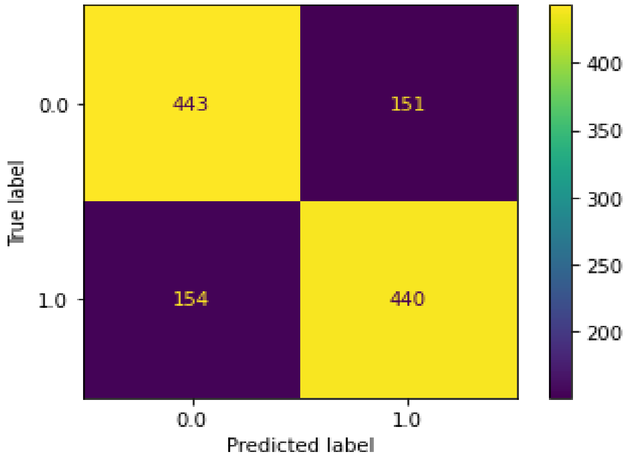


Figure 59. Confusion Matrix for Training Data in Logistic Regression SMOTE for Laptop

- ➢ For Test Data

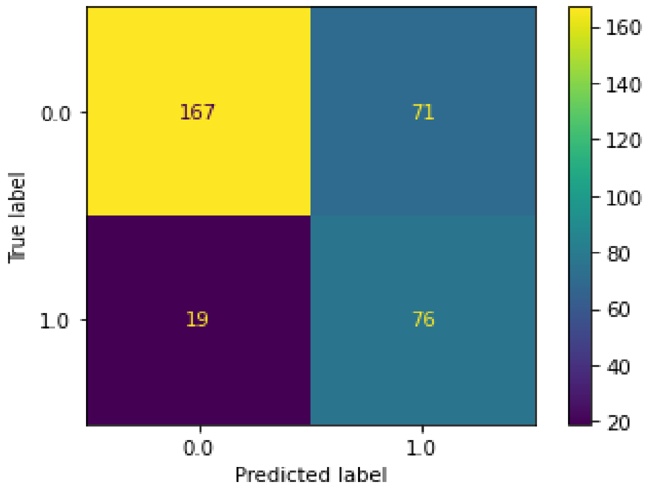


Figure 60. Confusion Matrix for Test Data in Logistic Regression SMOTE for Laptop

Classification Report

➢ For Training Data

0.7432659932659933

	precision	recall	f1-score	support
0.0	0.74	0.75	0.74	594
1.0	0.74	0.74	0.74	594
accuracy			0.74	1188
macro avg	0.74	0.74	0.74	1188
weighted avg	0.74	0.74	0.74	1188

Table 31. Classification Report for Training Data in Logistic Regression SMOTE for Laptop

➢ For Test Data

0.7297297297297297

	precision	recall	f1-score	support
0.0	0.90	0.70	0.79	238
1.0	0.52	0.80	0.63	95
accuracy			0.73	333
macro avg	0.71	0.75	0.71	333
weighted avg	0.79	0.73	0.74	333

Table 32. Classification Report for Test Data in Logistic Regression SMOTE for Laptop

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

➢ For Training Data

AUC: 0.821

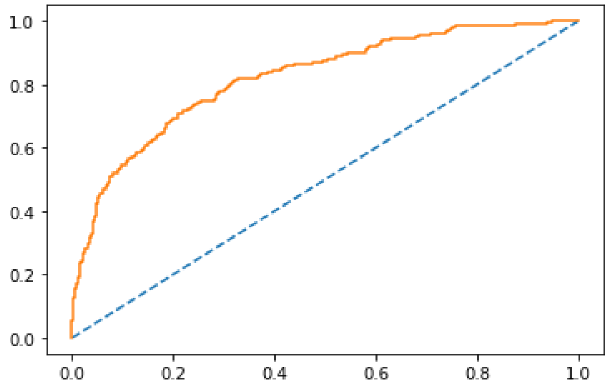


Figure 61. ROC for Training Data in Logistic Regression SMOTE for Laptop

➢ For Test Data

AUC: 0.857

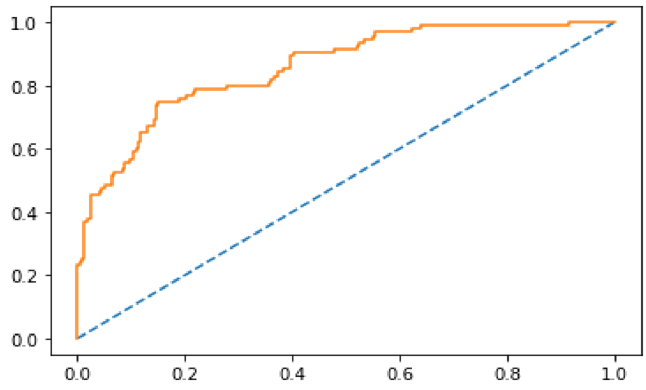


Figure 62. ROC for Test Data in Logistic Regression SMOTE for Laptop

KNN – SMOTE

Performance Metrices Basic KNN SMOTE

Model Score or Accuracy

- Accuracy for Training Data is 0.99
- Accuracy for Test Data is 0.91

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

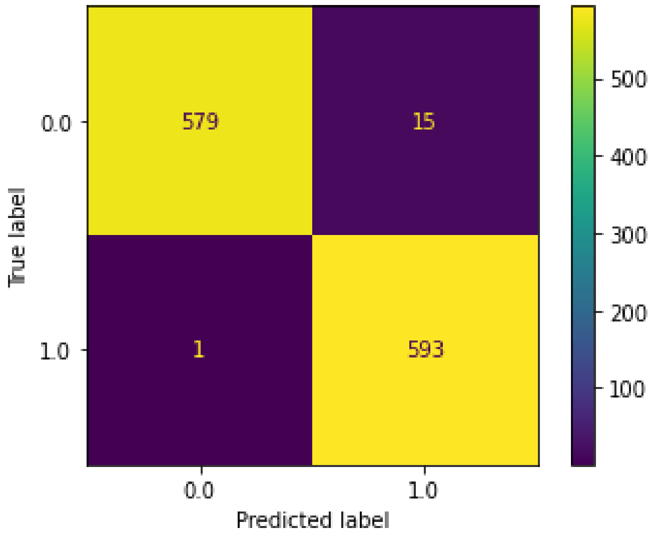


Figure 63. Confusion Matrix for Training Data in KNN SMOTE for Laptop

- ➢ For Test Data

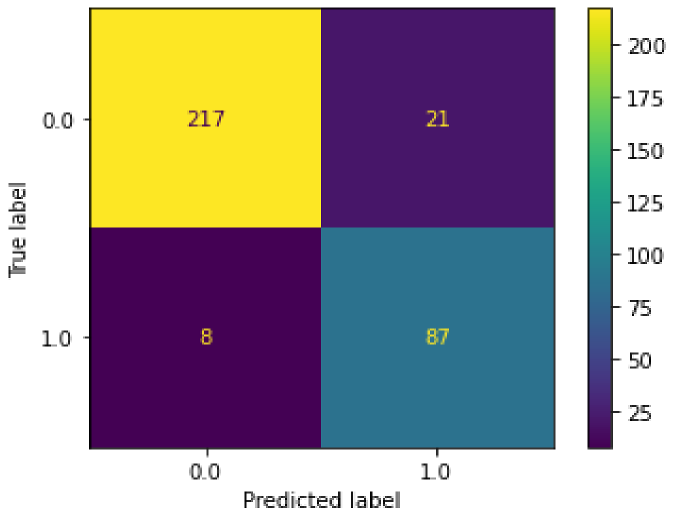


Figure 64. Confusion Matrix for Test Data in KNN SMOTE for Laptop

<u>Classification Report</u>

- ➢ For Training Data

```
0.9865319865319865
              precision    recall  f1-score   support

         0.0       1.00      0.97      0.99       594
         1.0       0.98      1.00      0.99       594

    accuracy                           0.99      1188
   macro avg       0.99      0.99      0.99      1188
weighted avg       0.99      0.99      0.99      1188
```

Table 33. Classification Report for Training Data in KNN SMOTE for Laptop

- ➢ For Test Data

0.9129129129129129

	precision	recall	f1-score	support
0.0	0.96	0.91	0.94	238
1.0	0.81	0.92	0.86	95
accuracy			0.91	333
macro avg	0.89	0.91	0.90	333
weighted avg	0.92	0.91	0.91	333

Table 34. Classification Report for Test Data in KNN SMOTE for Laptop

<u>ROC and AUC</u>

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

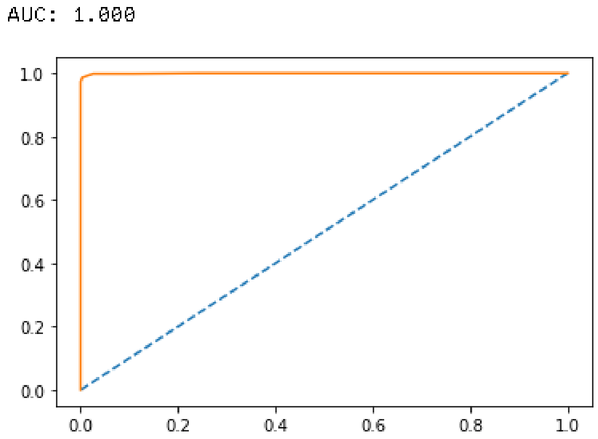


Figure 65. ROC for Training Data in KNN SMOTE for Laptop

- For Test Data

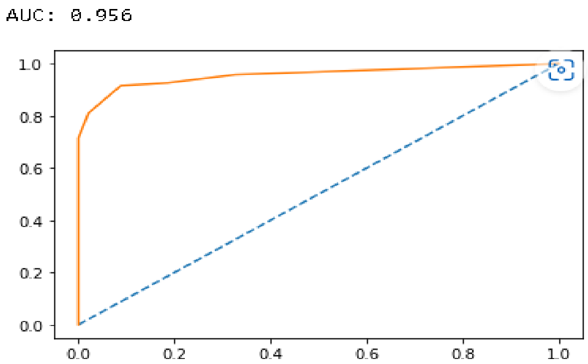


Figure 66. ROC for Test Data in KNN SMOTE for Laptop

Naïve Bayes – SMOTE

Performance Metrices Naïve Bayes SMOTE

<u>Model Score or Accuracy</u>

- Accuracy for Training Data is 0.71
- Accuracy for Test Data is 0.65

<u>Confusion Matrix</u>

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

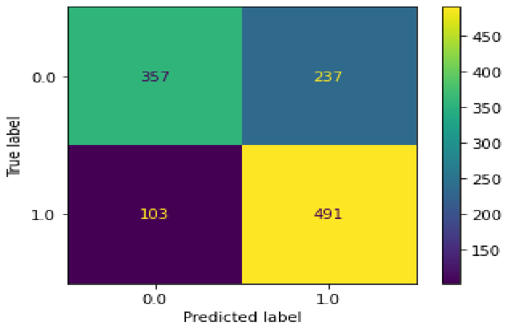


Figure 67. Confusion Matrix for Training Data in Naive Bayes SMOTE for Laptop

- For Test Data

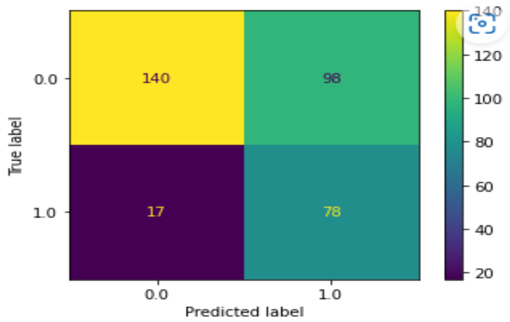


Figure 68. Confusion Matrix for Test Data in Naive Bayes SMOTE for Laptop

Classification Report

- For Training Data

0.7138047138047138

	precision	recall	f1-score	support
0.0	0.78	0.60	0.68	594
1.0	0.67	0.83	0.74	594
accuracy			0.71	1188
macro avg	0.73	0.71	0.71	1188
weighted avg	0.73	0.71	0.71	1188

Table 35. Classification Report for Training Data in Naive Bayes SMOTE for Laptop

- For Test Data

0.6546546546546547

	precision	recall	f1-score	support
0.0	0.89	0.59	0.71	238
1.0	0.44	0.82	0.58	95
accuracy			0.65	333
macro avg	0.67	0.70	0.64	333
weighted avg	0.76	0.65	0.67	333

Table 36. Classification Report for Test Data in Naive Bayes SMOTE for Laptop

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

AUC: 0.806

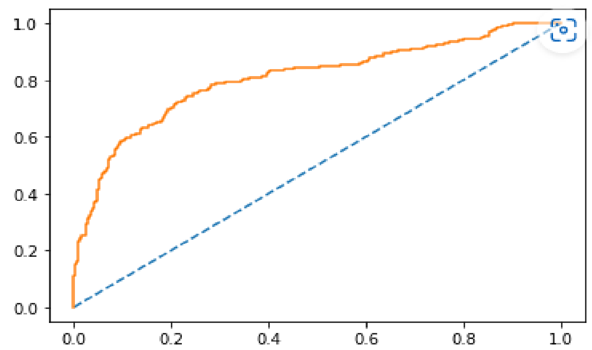


Figure 69. ROC for Training Data in Naive Bayes SMOTE for Laptop

- For Test Data

AUC: 0.840

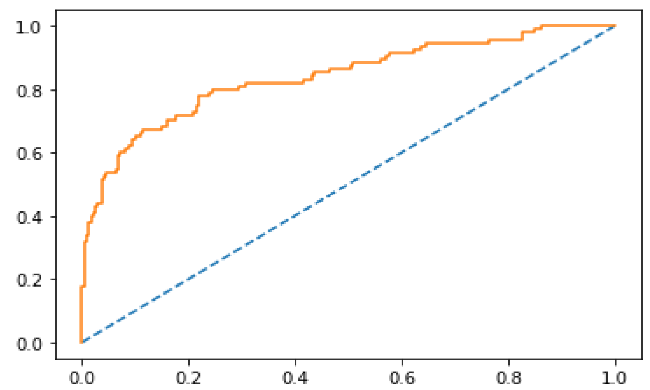


Figure 70. ROC for Test Data in Naive Bayes SMOTE for Laptop

Bagging – SMOTE

```
BaggingClassifier(base_estimator=RandomForestClassifier(),
n_estimators=100,random_state=1)
```

Performance Metrices Bagging SMOTE

Model Score or Accuracy

- Accuracy for Training Data is 1.0
- Accuracy for Test Data is 0.98

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

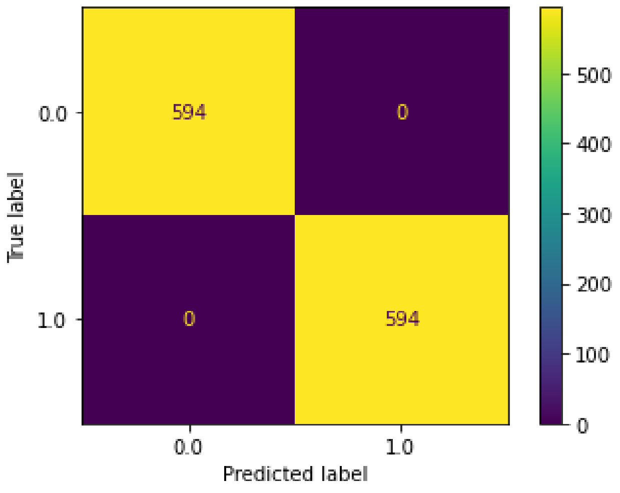


Figure 71. Confusion Matrix for Training Data in Bagging SMOTE for Laptop

- For Test Data

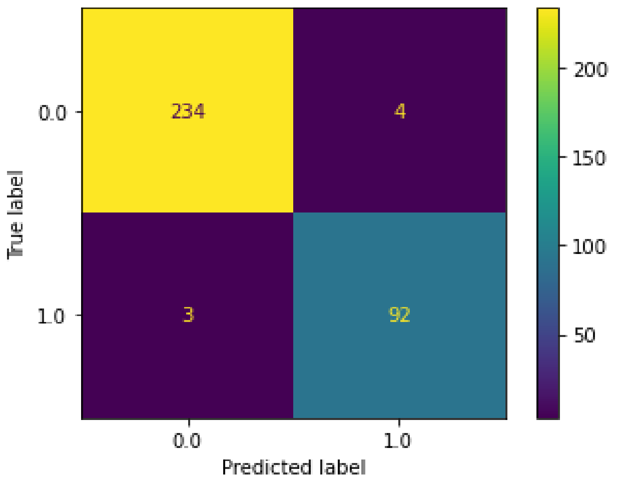


Figure 72. Confusion Matrix for Test Data in Bagging SMOTE for Laptop

Classification Report

➢ For Training Data

1.0

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	594
1.0	1.00	1.00	1.00	594
accuracy			1.00	1188
macro avg	1.00	1.00	1.00	1188
weighted avg	1.00	1.00	1.00	1188

Table 37. Classification Report for Training Data in Bagging SMOTE for Laptop

➢ For Test Data

0.978978978978979

	precision	recall	f1-score	support
0.0	0.99	0.98	0.99	238
1.0	0.96	0.97	0.96	95
accuracy			0.98	333
macro avg	0.97	0.98	0.97	333
weighted avg	0.98	0.98	0.98	333

Table 38. Classification Report for Test Data in Bagging SMOTE for Laptop

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

➢ For Training Data

AUC: 1.000

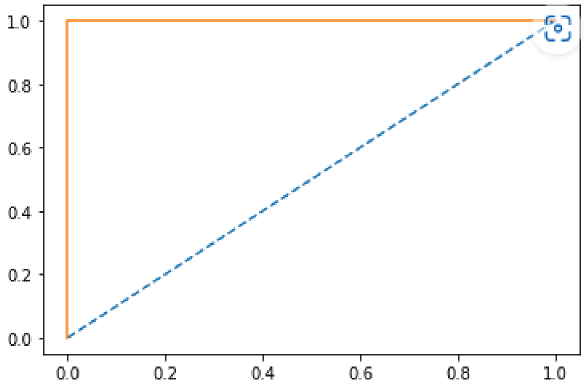


Figure 73. ROC for Training Data in Bagging SMOTE for Laptop

➢ For Test Data

AUC: 0.998

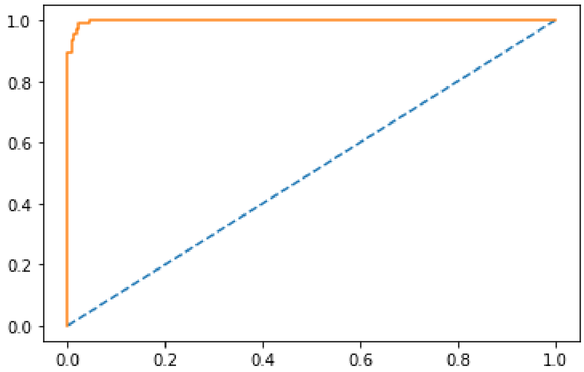


Figure 74. ROC for Test Data in Bagging SMOTE for Laptop

ADA Boosting – SMOTE

```
AdaBoostClassifier(n_estimators=100, random_state=1)
```

Performance Metrices Ada Boosting SMOTE

Model Score or Accuracy

- Accuracy for Training Data is 0.92
- Accuracy for Test Data is 0.83

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

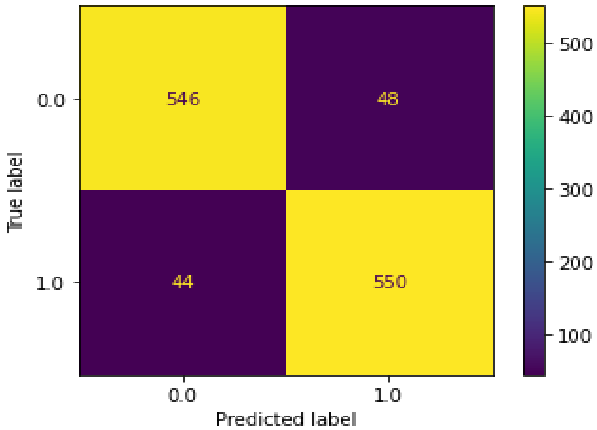


Figure 75. Confusion Matrix for Training Data in Ada Boosting SMOTE for Laptop

- For Test Data

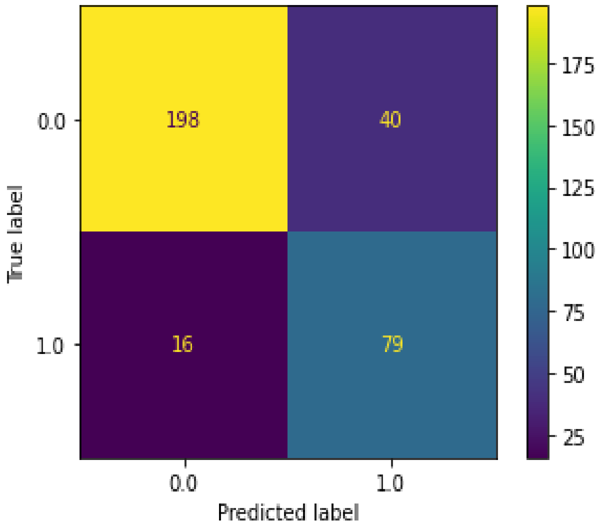


Figure 76. Confusion Matrix for Test Data in Ada Boosting SMOTE for Laptop

Classification Report

- For Training Data

0.9225589225589226

	precision	recall	f1-score	support
0.0	0.93	0.92	0.92	594
1.0	0.92	0.93	0.92	594
accuracy			0.92	1188
macro avg	0.92	0.92	0.92	1188
weighted avg	0.92	0.92	0.92	1188

Table 39. Classification Report for Training Data in Ada Boosting SMOTE for Laptop

- For Test Data

0.8318318318318318

	precision	recall	f1-score	support
0.0	0.93	0.83	0.88	238
1.0	0.66	0.83	0.74	95
accuracy			0.83	333
macro avg	0.79	0.83	0.81	333
weighted avg	0.85	0.83	0.84	333

Table 40. Classification Report for Test Data in Ada Boosting SMOTE for Laptop

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

AUC: 0.983

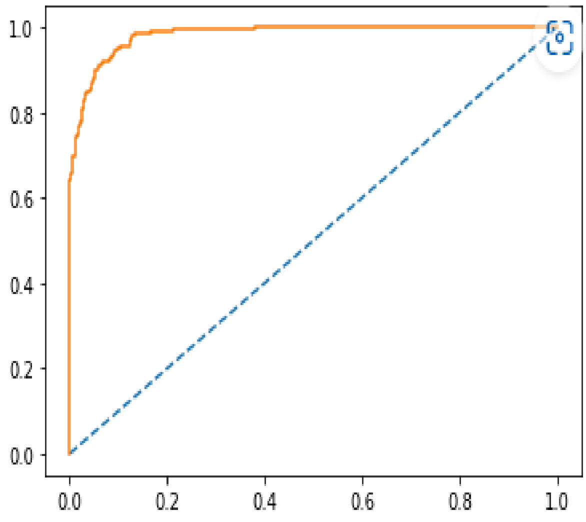


Figure 77. ROC for Training Data in Ada Boosting SMOTE for Laptop

- For Test Data

AUC: 0.932

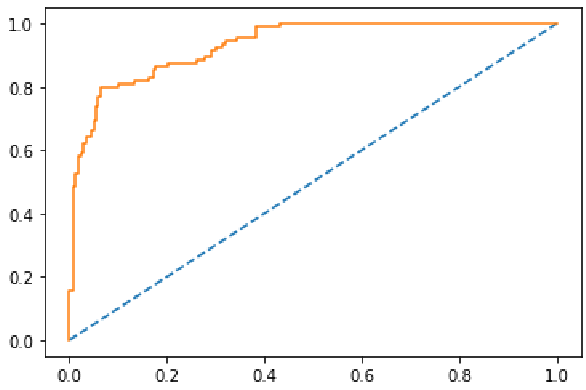


Figure 78. ROC for Test Data in Ada Boosting SMOTE for Laptop

Gradient Boosting – SMOTE

```
GradientBoostingClassifier(random_state=1)
```

Performance Metrices Gradient Boosting SMOTE

Model Score or Accuracy

- Accuracy for Training Data is 0.99
- Accuracy for Test Data is 0.0.95

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

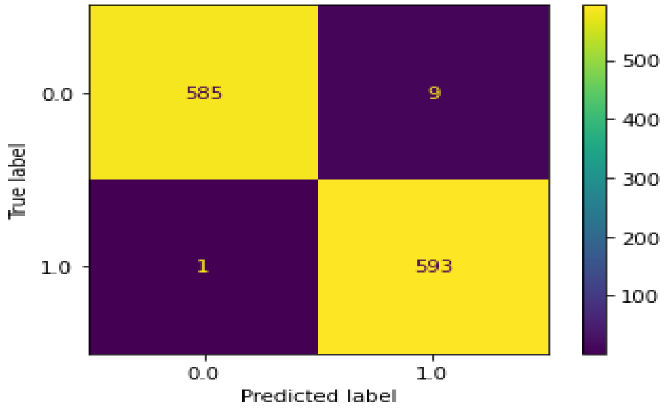


Figure 79. Confusion Matrix for Training Data in Gradient Boosting SMOTE for Laptop

- For Test Data

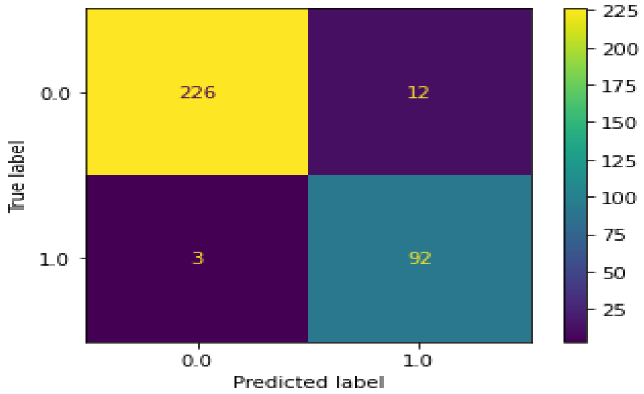


Figure 80. Confusion Matrix for Test Data in Gradient Boosting SMOTE for Laptop

Classification Report

- For Training Data

0.9915824915824916

	precision	recall	f1-score	support
0.0	1.00	0.98	0.99	594
1.0	0.99	1.00	0.99	594
accuracy			0.99	1188
macro avg	0.99	0.99	0.99	1188
weighted avg	0.99	0.99	0.99	1188

Table 41. Classification Report for Training Data in Gradient Boosting SMOTE for Laptop

- For Test Data

```
0.954954954954955
              precision    recall  f1-score   support

         0.0       0.99      0.95      0.97       238
         1.0       0.88      0.97      0.92        95

    accuracy                           0.95       333
   macro avg       0.94      0.96      0.95       333
weighted avg       0.96      0.95      0.96       333
```

Table 42. Classification Report for Test Data in Gradient Boosting SMOTE for Laptop

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

AUC: 0.999

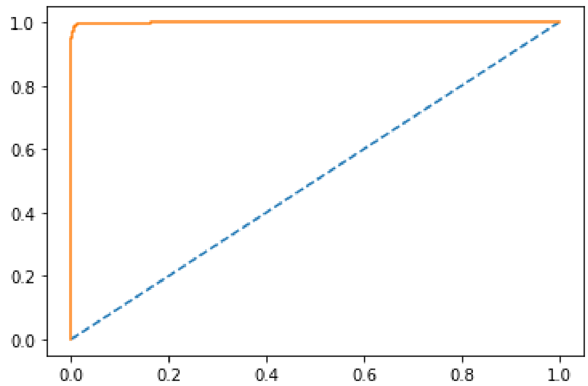


Figure 81. ROC for Training Data in Gradient Boosting SMOTE for Laptop

- For Test Data

AUC: 0.990

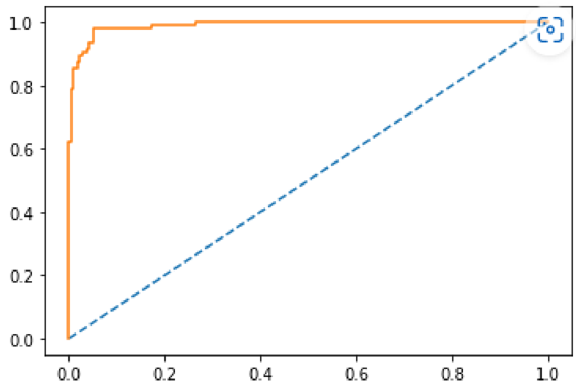


Figure 82. ROC for Test Data in Gradient Boosting Grid Search for Laptop

d). Interpretation of the hyper tuned models and Using SMOTE Techniques models.

Grid Search Model Tuning		Accuracy		Precision		Recall		F1 Score		AUC	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Logistic Regression	No Taken Product	0.84	0.83	0.85	0.83	0.95	0.97	0.9	0.89	0.82	0.87
	Yes Taken Product			0.74	0.87	0.46	0.49	0.57	0.63		
KNN	No Taken Product	0.87	0.8	0.86	0.79	0.99	0.97	0.92	0.87	0.96	0.92
	Yes Taken Product			0.95	0.85	0.45	0.35	0.61	0.49		
Naïve Bayes	No Taken Product	0.83	0.84	0.88	0.87	0.91	0.91	0.89	0.89	0.81	0.85
	Yes Taken Product			0.66	0.75	0.59	0.66	0.62	0.7		
Bagging	No Taken Product	1	0.89	1	0.87	1	1	1	0.93	1	0.99
	Yes Taken Product			1	1	1	0.62	1	0.77		
Ada Boosting	No Taken Product	0.95	0.88	0.95	0.89	0.99	0.95	0.97	0.92	0.99	0.95
	Yes Taken Product			0.96	0.84	0.84	0.72	0.9	0.77		
Gradient Boosting	No Taken Product	1	0.99	1	0.98	1	1	1	0.99	1	1
	Yes Taken Product			1	1	1	0.96	1	0.98		

Table 43. Model Tuning Comparison for Laptop

SMOTE		Accuracy		Precision		Recall		F1 Score		AUC	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Logistic Regression	No Taken Product	0.74	0.73	0.74	0.9	0.75	0.7	0.74	0.79	0.82	0.86
	Yes Taken Product			0.74	0.52	0.74	0.8	0.74	0.63		
KNN	No Taken Product	0.99	0.91	1	0.96	0.97	0.91	0.99	0.94	1	0.96
	Yes Taken Product			0.98	0.81	1	0.92	0.99	0.86		
Naïve Bayes	No Taken Product	0.71	0.65	0.78	0.89	0.6	0.59	0.68	0.71	0.81	0.84
	Yes Taken Product			0.67	0.44	0.83	0.82	0.74	0.58		
Bagging	No Taken Product	1	0.98	1	0.99	1	0.98	1	0.99	1	0.99
	Yes Taken Product			1	0.96	1	0.97	1	0.96		
Ada Boosting	No Taken Product	0.92	0.83	0.93	0.93	0.92	0.83	0.92	0.88	0.98	0.93
	Yes Taken Product			0.92	0.66	0.93	0.83	0.92	0.74		
Gradient Boosting	No Taken Product	0.99	0.95	1	0.99	0.98	0.95	0.99	0.97	0.99	0.99
	Yes Taken Product			0.99	0.88	1	0.97	0.99	0.92		

Table 44. Using SMOTE models comparison for Laptop

- According to problem we will focus on the Customer who have taken the product.
- There is not much improvement in performance for the Logistic Regression model after hyper tuning and SMOTE technique. For LR model performance declined after applying SMOTE Technique.
- For KNN after Hyper tuning model performance declined and after applying SMOTE Technique there is improvement in process but Precision is good for training set but decreases in Test Set
- There is not much improvement in performance for the Naïve Bayes model after hyper tuning and SMOTE technique. For Naïve Bayes model performance declined after applying SMOTE Technique.
- For Bagging model performance declined when hyper tuning model but in case of SMOTE technique model is performing well.
- For the ADA Boosting Model there is not much improvement in performance after hyper tuning and SMOTE technique. For ADA Boosting model performance declined after applying SMOTE Technique.

Mobile

a). Build various models (You can choose to build models for either or all of descriptive, predictive or prescriptive purposes) and test your predictive model against the test set using various appropriate performance metrics

Scaling

In regression or classification, it is often a good practice to centre the variables so that predictor have a mean of 0. This makes it easier to intercept the intercept term as the expected value of Yi when the predictor values are set to their means. Otherwise, the intercept is interpreted as the expected value of Yi when the predictors are set to 0, which may not be a realistic or interpretable situation. Another valid reason for scaling in regression is when one predictor variable has a very large scale. In that case, the regression coefficients may be on a very small order of magnitude which can be unclear to interpret. The convention that we standardize predictions primarily exists so that the units of the regression coefficients are the same. More often, the dataset contains feature highly varying in magnitudes, units and range. However, most of the machine learning algorithms use Euclidean distance between two data points in their computations, and this can be a potential problem. Also, scaling helps to standardize the independent features present in the data in a fixed range. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values. The features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes. To suppress this effect, we need to bring all features to the same level of magnitudes.

Yes, Scaling is absolutely necessary in this case as we have Variables that carry absolute numbers and we haveVariables that carry percentage. If we have data in different scales, the variables with larger scale will dominate,this is probably not what we want.After scaling there is variance look similar across all data.

Train and Test Split

Before splitting we need to determine the target variable. Hence, the target variable is “vote_Labour”

We will split the data for 70:30 ratio with a random state =1.

Train Test Data Shape

```
X_train (7456, 15)
X_test (3196, 15)
y_train (7456,)
y_test (3196,)
```

Logistic Regression Model

Logistic regression is a process of modelling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on.

We split the data into train and test using train_test_split command and fit our linear regression model into the train data and then try to predict the outcome of using the test data. Then we compare the actual against the predicted to calculate the accuracy of the model.

Performance Metrices Basic Logistic Regression Model

Model Score or Accuracy

- Accuracy for Training Data is 0.87
- Accuracy for Test Data is 0.87

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset. It will allow us to visualise the performance of the Logistic Regression Model.

- For Training Data

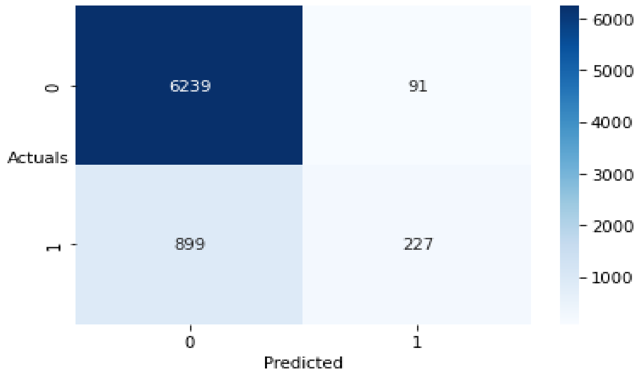


Figure 83. Confusion Matrix for Training Data in Basic Logistic Regression Model for Mobile

- For Test Data

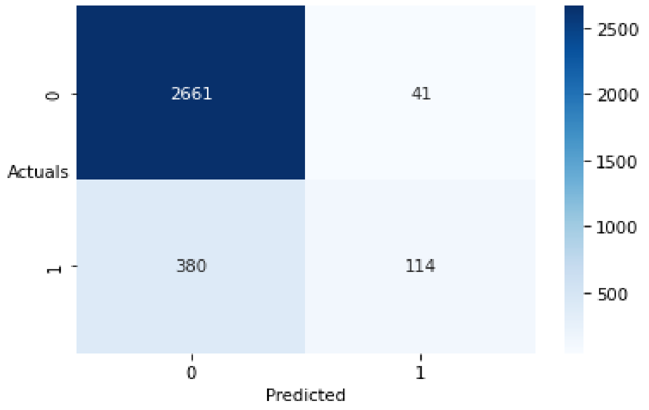


Figure 84. Confusion Matrix for Test Data in Basic Logistic Regression Model for Mobile

Classification Report

➢ For Training Data

The classification report for Logistic Regression training set is

	precision	recall	f1-score	support
0.0	0.87	0.99	0.93	6330
1.0	0.71	0.20	0.31	1126
accuracy			0.87	7456
macro avg	0.79	0.59	0.62	7456
weighted avg	0.85	0.87	0.83	7456

Table 45. Classification Report for Training Data in Basic Logistic Regression Model for Mobile

➢ For Test Data

The classification report for Logistic Regression testing set

	precision	recall	f1-score	support
0.0	0.88	0.98	0.93	2702
1.0	0.74	0.23	0.35	494
accuracy			0.87	3196
macro avg	0.81	0.61	0.64	3196
weighted avg	0.85	0.87	0.84	3196

Table 46. Classification Report for Test Data in Basic Logistic Regression Model for Mobile

<u>ROC and AUC</u>

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

➢ For Training Data

The AUC score for Logistic Regression training set is: 0.786

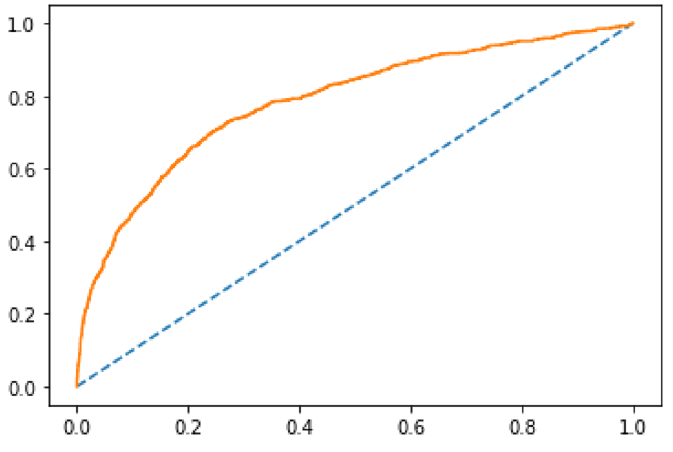


Figure 85. ROC for Training Data in Basic Logistic Regression Model for Mobile

➢ For Test Data

```
The AUC score for Logistic Regression testing set is: 0.798
```

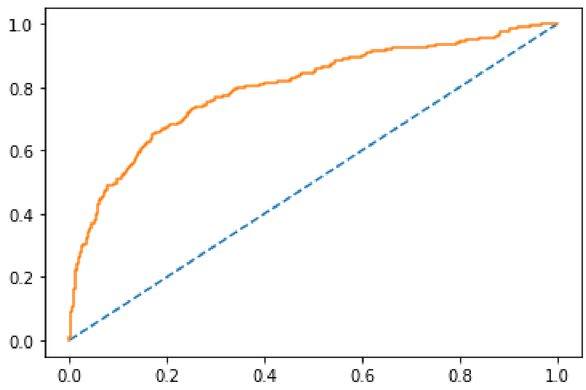


Figure 86. ROC for Test Data in Basic Logistic Regression Model for Mobile

KNN Model

KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

Performance Metrices Basic KNN Model

Model Score or Accuracy

- ➢ Accuracy for Training Data is 0.99
- ➢ Accuracy for Test Data is 0.97

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- ➢ For Training Data

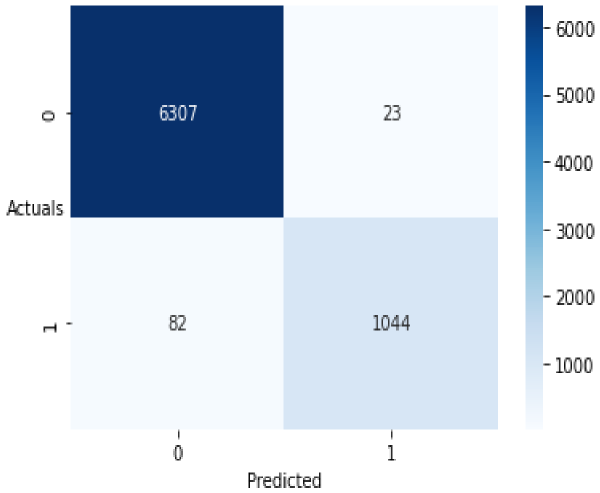


Figure 87. Confusion Matrix for Training Data in Basic KNN Model for Mobile

- ➢ For Test Data

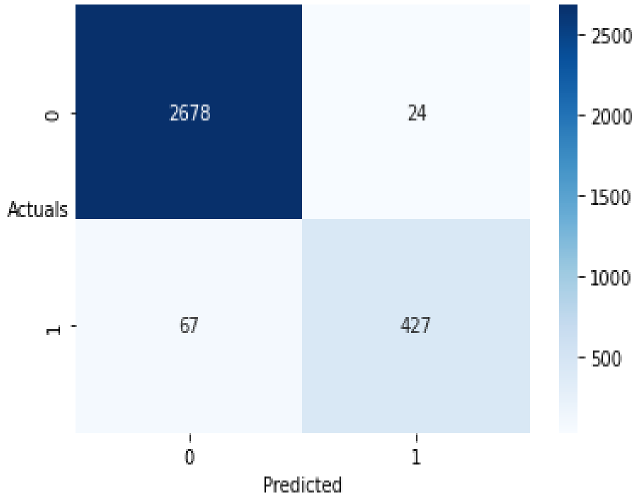


Figure 88. Confusion Matrix for Test Data in Basic KNN Model for Mobile

Classification Report

➢ For Training Data

The classification report for KNN set is

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	6330
1.0	0.98	0.93	0.95	1126
accuracy			0.99	7456
macro avg	0.98	0.96	0.97	7456
weighted avg	0.99	0.99	0.99	7456

Table 47. Classification Report for Training Data in Basic KNN Model for Mobile

➢ For Test Data

The classification report for KNN testing set is

	precision	recall	f1-score	support
0.0	0.98	0.99	0.98	2702
1.0	0.95	0.86	0.90	494
accuracy			0.97	3196
macro avg	0.96	0.93	0.94	3196
weighted avg	0.97	0.97	0.97	3196

Table 48. Classification Report for Test Data in Basic KNN Model for Mobile

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

➢ For Training Data

```
The AUC score for KNN training set is: 0.999
```

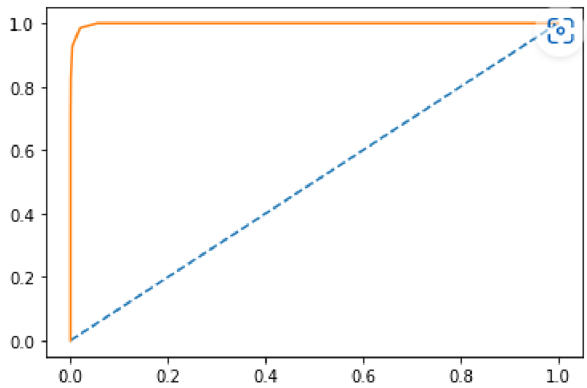


Figure 89. ROC for Training Data in Basic KNN Model for Mobile

➢ For Test Data

```
The AUC score for KNN testing set is: 0.988
```

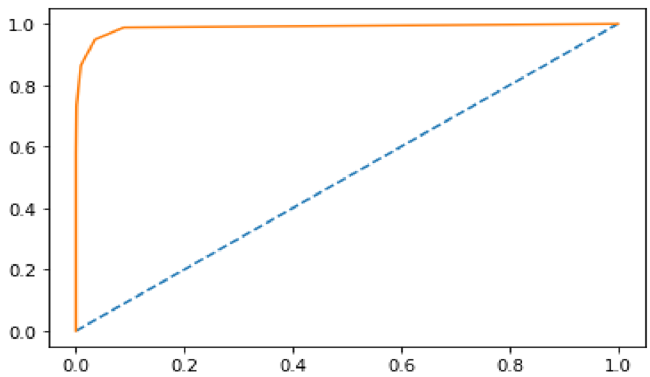


Figure 90. ROC for Test Data in Basic KNN Model for Mobile

Naïve Bayes Model

Naïve Bayes is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Performance Metrices Basic Naïve Bayes Model

Model Score or Accuracy

- ➢ Accuracy for Training Data is 0.86
- ➢ Accuracy for Test Data is 0.85

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

➢ For Training Data

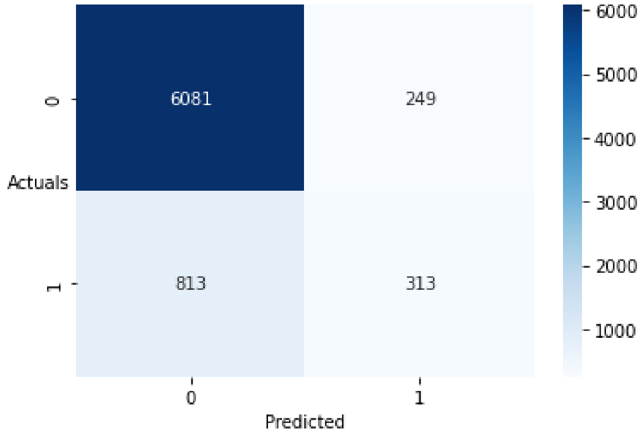


Figure 91. Confusion Matrix for Training Data in Basic Naive Bayes Model for Mobile

➢ For Test Data

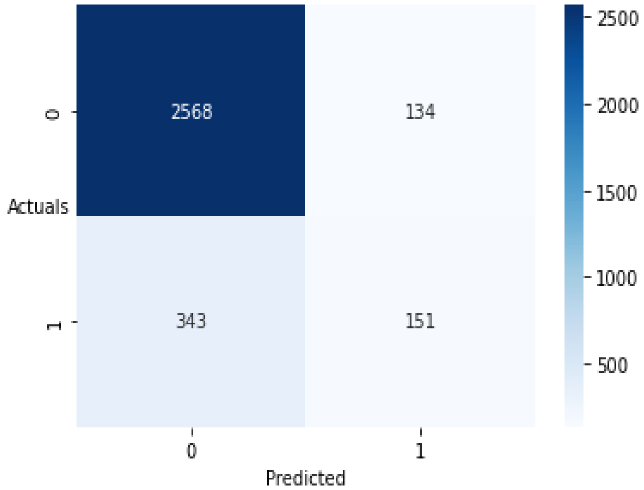


Figure 92. Confusion Matrix for Test Data in Basic Naive Bayes Model for Mobile

Classification Report

➢ For Training Data

The classification report for Naive Bayes Model set is

	precision	recall	f1-score	support
0.0	0.88	0.96	0.92	6330
1.0	0.56	0.28	0.37	1126
accuracy			0.86	7456
macro avg	0.72	0.62	0.65	7456
weighted avg	0.83	0.86	0.84	7456

Table 49. Classification Report for Training Data in Basic Naive Bayes Model for Mobile

➢ For Test Data

The classification report for Naive bayes Model testing set is

	precision	recall	f1-score	support
0.0	0.88	0.95	0.92	2702
1.0	0.53	0.31	0.39	494
accuracy			0.85	3196
macro avg	0.71	0.63	0.65	3196
weighted avg	0.83	0.85	0.83	3196

Table 50. Classification Report for Test Data in Basic Naive Bayes Model for Mobile

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

➢ For Training Data

The AUC score for Naive Bayes training set is: 0.765

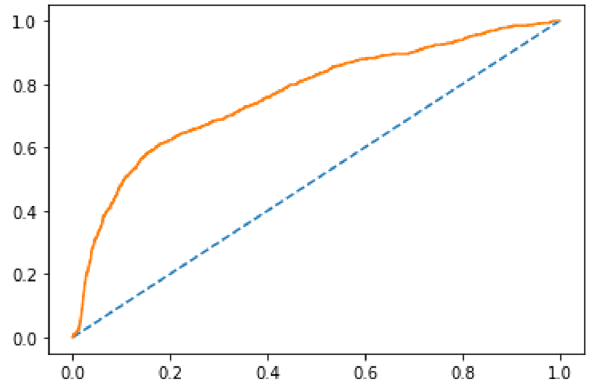


Figure 93. ROC for Training Data in Basic Naive Bayes Model for Mobile

➢ For Test Data

The AUC score for Naive Bayes testing set is: 0.774

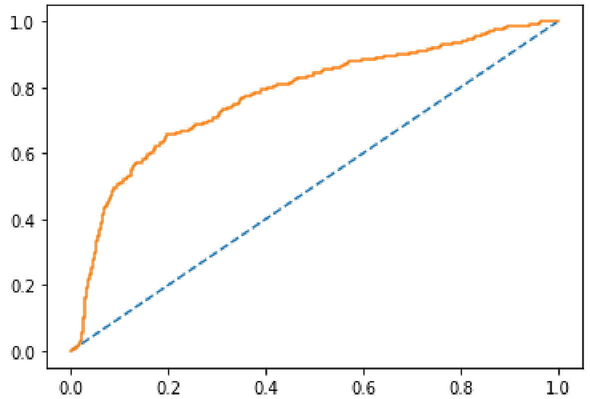


Figure 94. ROC for Test Data in Basic Naive Bayes Model for Mobile

Bagging

Bagging is designed to improve the performance of existing ML algorithms used in statistical classification or regression. It is most used with tree-based algorithms. It is a parallel method.

```
BaggingClassifier(base_estimator=RandomForestClassifier(),
n_estimators=100,random_state=1)
```

Performance Metrices Basic Bagging

Model Score or Accuracy

- Accuracy for Training Data is 1.0
- Accuracy for Test Data is 0.96

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

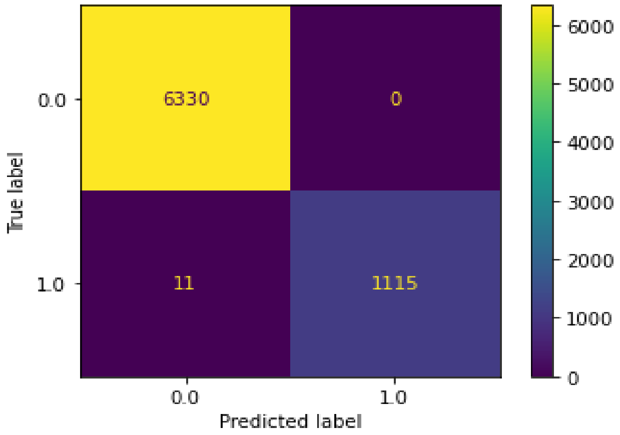


Figure 95. Confusion Matrix for Training Data in Basic Bagging for Mobile

- For Test Data

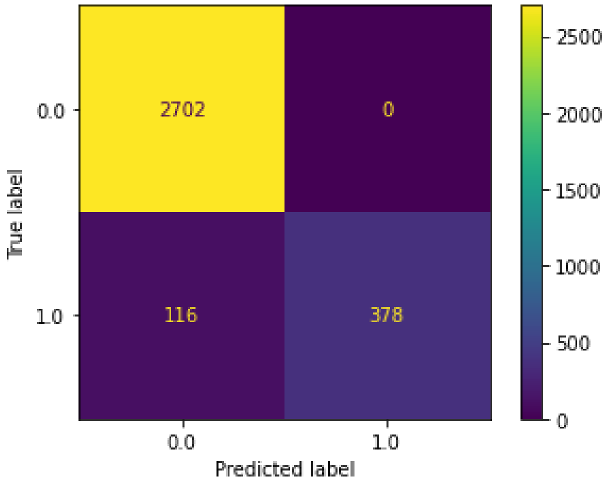


Figure 96. Confusion Matrix for Test Data in Basic Bagging for Mobile

Classification Report

- For Training Data

```
0.9985246781115880
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00      6330
         1.0       1.00      0.99      1.00      1126

    accuracy                           1.00      7456
   macro avg       1.00      1.00      1.00      7456
weighted avg       1.00      1.00      1.00      7456
```

Table 51. Classification Report for Training Data in Basic Bagging for Mobile

- For Test Data

```
0.9637046307884856
              precision    recall  f1-score   support

         0.0       0.96      1.00      0.98      2702
         1.0       1.00      0.77      0.87       494

    accuracy                           0.96      3196
   macro avg       0.98      0.88      0.92      3196
weighted avg       0.97      0.96      0.96      3196
```

Table 52.Classification Report for Test Data in Basic Bagging for Mobile

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

AUC: 1.000

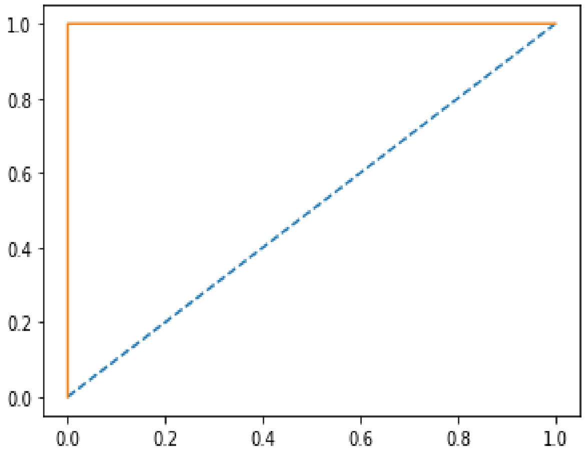


Figure 97. ROC for Training Data in Basic Bagging for Mobile

➢ For Test Data

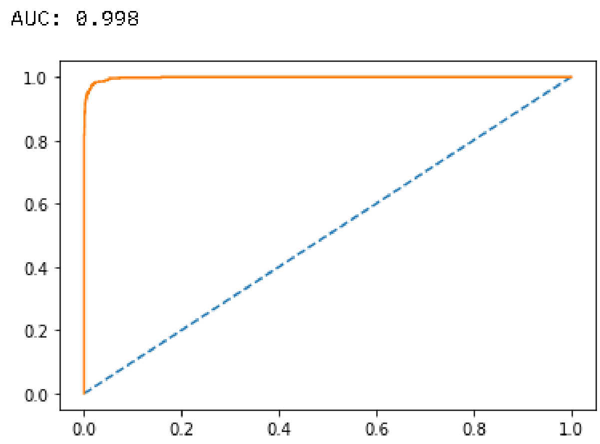


Figure 98. ROC for Test Data in Basic Bagging for Mobile

ADA Boosting

This model is used to increase the efficiency of binary classifiers, but now used to improve multiclass classifiers as well. ADA boosting can be applied on top of any classifier method to learn from its issues and bring about a more accurate model and this it is called "best out of the box classifier"

```
AdaBoostClassifier(n_estimators=100, random_state=1)
```

Performance Metrices Basic Ada Boosting

Model Score or Accuracy

- ➢ Accuracy for Training Data is 0.88
- ➢ Accuracy for Test Data is 0.88

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

➢ For Training Data

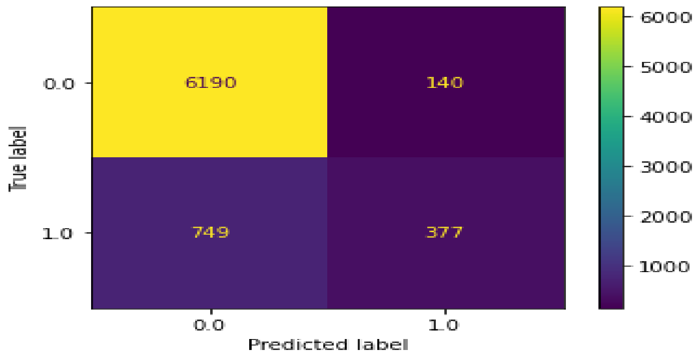


Figure 99.Confusion Matrix for Training Data in Basic Ada Boosting for Mobile

➢ For Test Data

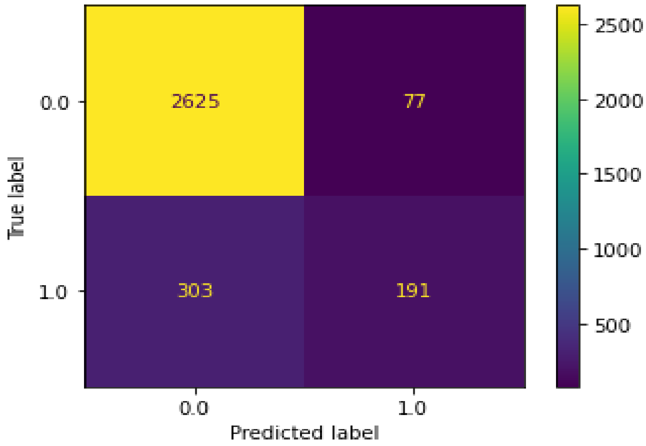


Figure 100. Confusion Matrix for Test Data in Basic Ada Boosting for Mobile

Classification Report

➢ For Training Data

0.8807671673819742

	precision	recall	f1-score	support
0.0	0.89	0.98	0.93	6330
1.0	0.73	0.33	0.46	1126
accuracy			0.88	7456
macro avg	0.81	0.66	0.70	7456
weighted avg	0.87	0.88	0.86	7456

Table 53. Classification Report for Training Data in Basic Ada Boosting for Mobile

➢ For Test Data

0.8811013767209012

	precision	recall	f1-score	support
0.0	0.90	0.97	0.93	2702
1.0	0.71	0.39	0.50	494
accuracy			0.88	3196
macro avg	0.80	0.68	0.72	3196
weighted avg	0.87	0.88	0.87	3196

Table 54. Classification Report for Test Data in Basic Ada Boosting for Mobile

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

➢ For Training Data

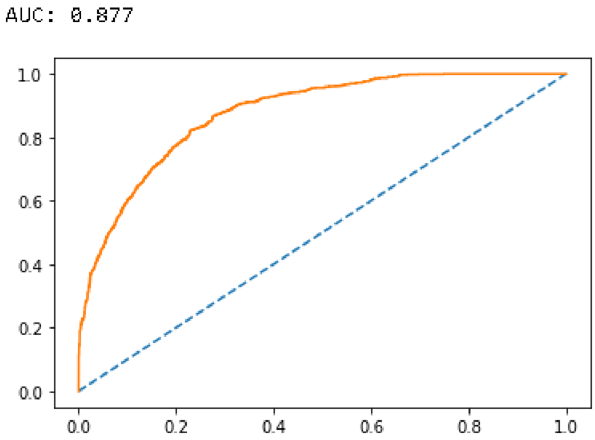


Figure 101. ROC for Training Data in Basic Ada Boosting for Mobile

➢ For Test Data

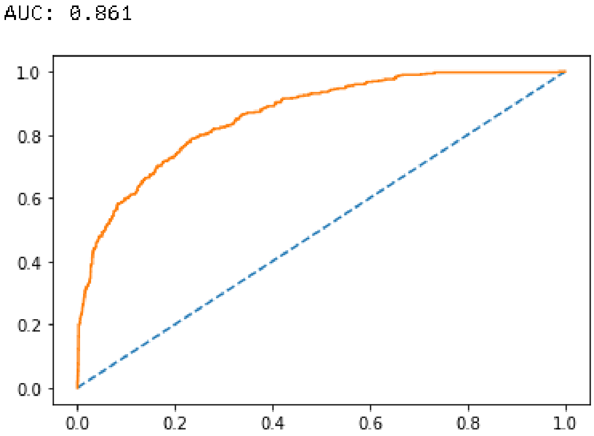


Figure 102. ROC for Test Data in Basic Ada Boosting for Mobile

Gradient Boosting

This model is just like the ADA Boosting works by sequentially adding the misidentified predictors and under-fitted predictions to the ensemble, ensuring the errors identified previously are corrected. The major difference lies in what it does with the mis-identified values of the previous weak learner.

Performance Metrices Basic Gradient Boosting

Model Score or Accuracy

- ➢ Accuracy for Training Data is 0.91
- ➢ Accuracy for Test Data is 0.90

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

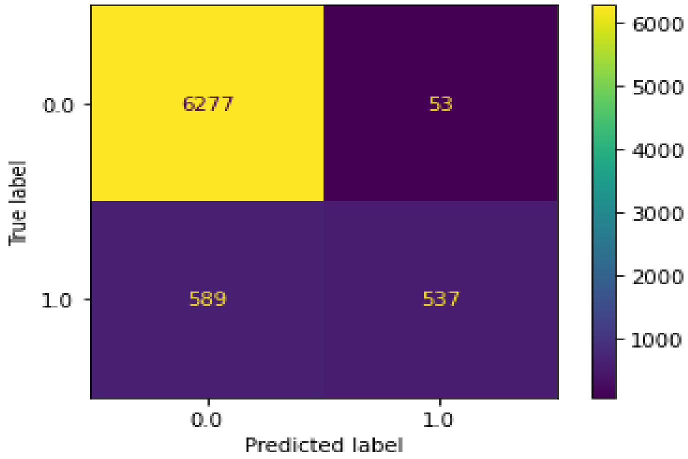


Figure 103.Confusion Matrix for Training Data in Basic Gradient Boosting for Mobile

- For Test Data

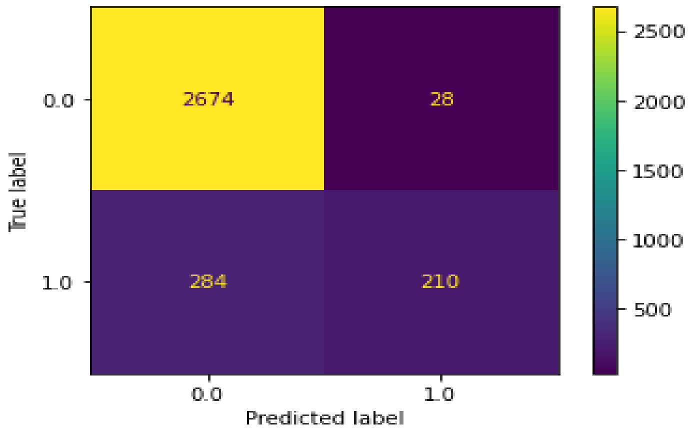


Figure 104. Confusion Matrix for Test Data in Basic Gradient Boosting for Mobile

Classification Report

- For Training Data

0.9138948497854077

	precision	recall	f1-score	support
0.0	0.91	0.99	0.95	6330
1.0	0.91	0.48	0.63	1126
accuracy			0.91	7456
macro avg	0.91	0.73	0.79	7456
weighted avg	0.91	0.91	0.90	7456

Table 55. Classification Report for Training Data in Basic Gradient Boosting for Mobile

➢ For Test Data

0.9023779724655819

	precision	recall	f1-score	support
0.0	0.90	0.99	0.94	2702
1.0	0.88	0.43	0.57	494
accuracy			0.90	3196
macro avg	0.89	0.71	0.76	3196
weighted avg	0.90	0.90	0.89	3196

Table 56. Classification Report for Test Data in Basic Gradient Boosting for Mobile

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

➢ For Training Data

AUC: 0.939

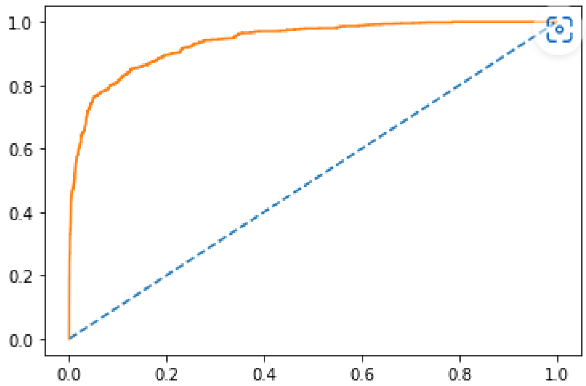


Figure 105. ROC for Training Data in Basic Gradient Boosting for Mobile

➢ For Test Data

AUC: 0.915

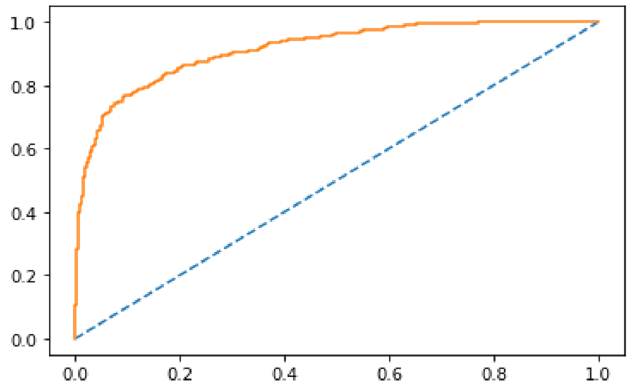


Figure 106. ROC for Test Data in Basic Gradient Boosting for Mobile

b). Interpretation of the model(s)

Basic Model		Accuracy		Precision		Recall		F1 Score		AUC	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Logistic Regression	No Taken Product	0.87	0.87	0.87	0.88	0.99	0.98	0.93	0.93	0.79	0.8
	Yes Taken Product			0.71	0.74	0.2	0.23	0.31	0.35		
KNN	No Taken Product	0.99	0.97	0.99	0.98	1	0.99	0.99	0.98	0.99	0.99
	Yes Taken Product			0.98	0.95	0.93	0.86	0.95	0.9		
Naïve Bayes	No Taken Product	0.86	0.85	0.88	0.88	0.96	0.95	0.92	0.92	0.77	0.77
	Yes Taken Product			0.56	0.53	0.28	0.31	0.37	0.39		
Bagging	No Taken Product	1	0.96	1	0.96	1	1	1	0.98	1	0.99
	Yes Taken Product			1	1	1	0.77	1	0.87		
Ada Boosting	No Taken Product	0.88	0.88	0.89	0.9	0.98	0.97	0.93	0.93	0.88	0.86
	Yes Taken Product			0.73	0.71	0.33	0.39	0.46	0.5		
Gradient Boosting	No Taken Product	0.91	0.9	0.91	0.9	0.99	0.99	0.95	0.94	0.94	0.92
	Yes Taken Product			0.91	0.88	0.48	0.43	0.63	0.57		

Table 57. Basic Models Comparisons for Mobile

- According to problem we will focus on the Customer who have taken the product.
- Logistic Regression model and KNN model provides accuracy of 87% and 87% on train set and 99% and 97% on test set respectively. In Logistic regression accuracy remain same for tr ain test and but in KNN it can be observed that the accuracy for test set decreases.
- Naïve Bayes model have provided a decent accuracy on Training set that is 86% and applying the models to testing set, we see that the accuracy has declined a bit that is 85%
- The desired metric for the problem is Precision which is not good for the Logistic Regression and Naïve Bayes. In case of KNN for Precision is good for Train but when applied for test set it declined a bit.
- Bagging model has high score for all parameters in Training data but it has not performed well in Test data and hence it is overfitted model
- Gradient Boosting model is better than ADA model as it has high score in Accuracy, Precision, Recall, F1 score and AUC.

c) Ensemble modelling, wherever applicable and Any other model tuning measures (if applicable)

Model Tuning

Tuning is process of maximizing a model's performance without overfitting or creating too high of a variance. In ML, this is accomplished by selecting appropriate "hyper-parameters".

Logistic Regression Model – Grid Search

We split the data into train and test using train_test_split command and fit our linear regression model into the train data and then try to predict the outcome of using the test data.

Then we compare the actual against the predicted to calculate the accuracy of the model. We will hyper tune the parameters that would enhance the outcome of the model.

```
GridSearchCV(cv=5, estimator=LogisticRegression(),
             param_grid={'C': [0.001, 0.009, 0.01, 0.09, 1, 5, 10, 25],
                         'penalty': ['l1', 'l2'], 'solver': ['newton-
cg']})

Best_Estimator LogisticRegression(C=1, solver='newton-cg')
```

Performance Metrices Logistic Regression Grid Search

<u>Model Score or Accuracy</u>

- ➢ Accuracy for Training Data is 0.87
- ➢ Accuracy for Test Data is 0.87

<u>Confusion Matrix</u>

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset. It will allow us to visualise the performance of the Logistic Regression Model.

- ➢ For Training Data

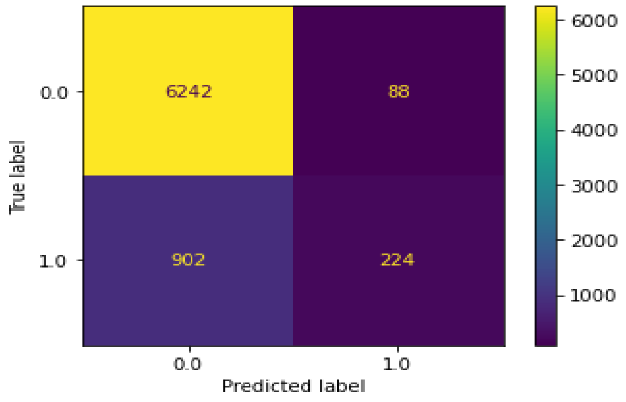


Figure 107. Confusion Matrix for Training Data in Logistic Regression Grid Search for Mobile

- ➢ For Test Data

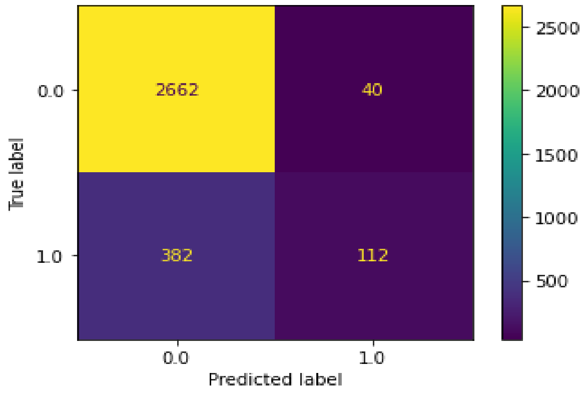


Figure 108. Confusion Matrix for Test Data in Logistic Regression Grid Search for Mobile

<u>Classification Report</u>

- For Training Data

0.8672210300429185

	precision	recall	f1-score	support
0.0	0.87	0.99	0.93	6330
1.0	0.72	0.20	0.31	1126
accuracy			0.87	7456
macro avg	0.80	0.59	0.62	7456
weighted avg	0.85	0.87	0.83	7456

Table 58. Classification Report for Training Data in Logistic Regression Grid Search for Mobile

- For Test Data

0.8679599499374218

	precision	recall	f1-score	support
0.0	0.87	0.99	0.93	2702
1.0	0.74	0.23	0.35	494
accuracy			0.87	3196
macro avg	0.81	0.61	0.64	3196
weighted avg	0.85	0.87	0.84	3196

Table 59. Classification Report for Test Data in Logistic Regression Grid Search for Mobile

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

AUC: 0.786

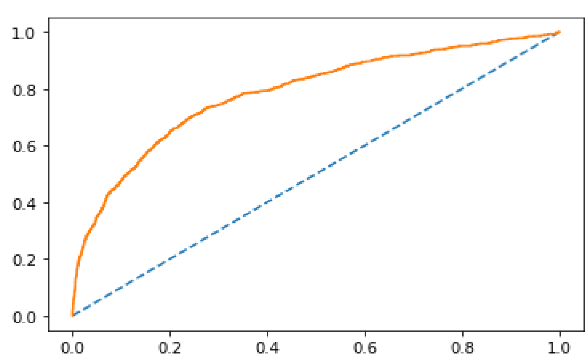


Figure 109. ROC for Training Data in Logistic Regression Grid Search for Mobile

- For Test Data

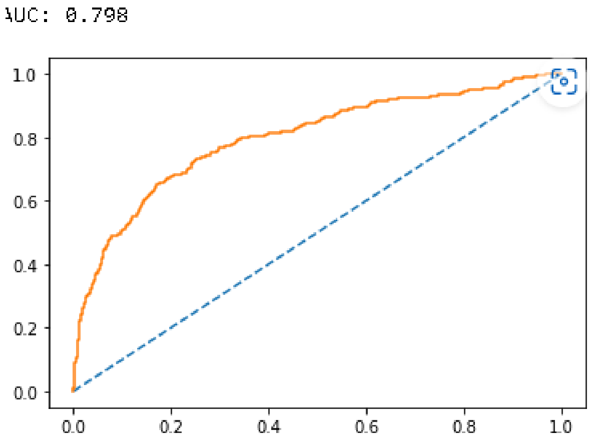


Figure 110. ROC for Test Data in Logistic Regression Grid Search for Laptop

KNN – Grid Search

We split the data into train and test using train_test_split command and fit our KNN regression model into the train data and then try to predict the outcome of using the test data. Then we compare the actual against the predicted to calculate the accuracy of the model. We will hyper tune the parameters that would enhance the outcome of the model

```
GridSearchCV(cv=5, estimator=KNeighborsClassifier(),
             param_grid={'leaf_size': [20, 30, 50], 'n_neighbors': [10, 
20, 30],'p': [1, 2]})
```

Performance Metrices Basic KNN Grid Search

Model Score or Accuracy

- Accuracy for Training Data is 0.95
- Accuracy for Test Data is 0.92

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

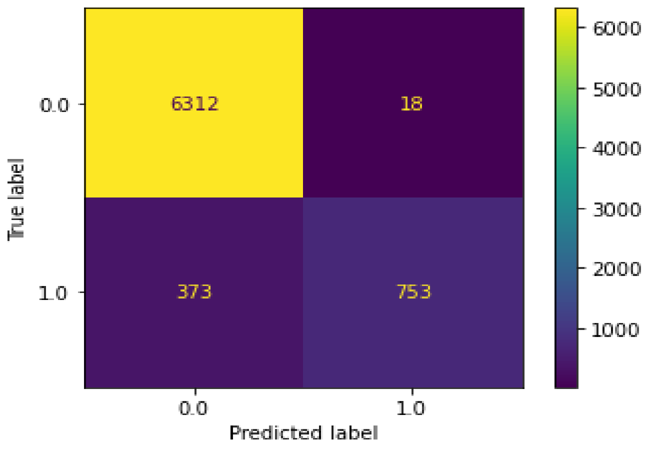


Figure 111. Confusion Matrix for Training Data in KNN Grid Search for Mobile

- For Test Data

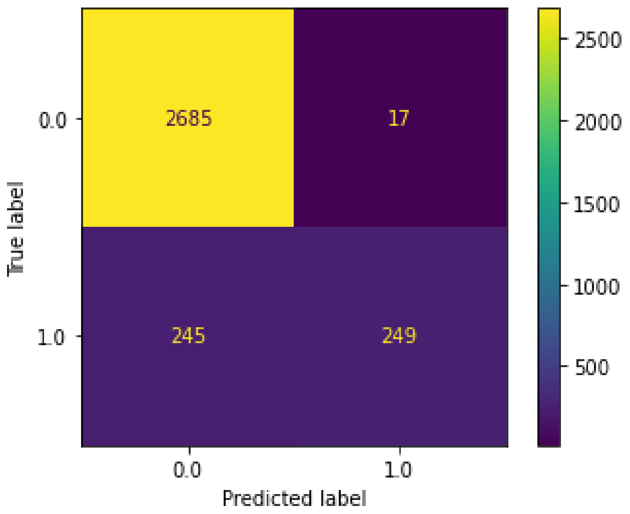


Figure 112. Confusion Matrix for Test Data in KNN Grid Search for Mobile

<u>Classification Report</u>

- For Training Data

0.9475590128755365

	precision	recall	f1-score	support
0.0	0.94	1.00	0.97	6330
1.0	0.98	0.67	0.79	1126
accuracy			0.95	7456
macro avg	0.96	0.83	0.88	7456
weighted avg	0.95	0.95	0.94	7456

Table 60. Classification Report for Training Data in KNN Grid Search for Mobile

- For Test Data

```
0.9180225281602002
              precision    recall  f1-score   support

         0.0       0.92      0.99      0.95      2702
         1.0       0.94      0.50      0.66       494

    accuracy                           0.92      3196
   macro avg       0.93      0.75      0.80      3196
weighted avg       0.92      0.92      0.91      3196
```

Table 61. Classification Report for Test Data in KNN Grid Search for Mobile

ROC and AUC

- For Training Data

AUC: 0.991

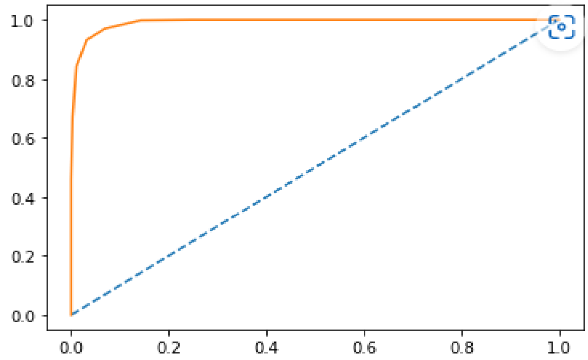


Figure 113. ROC for Training Data in KNN Grid Search for Mobile

- For Test Data

AUC: 0.973

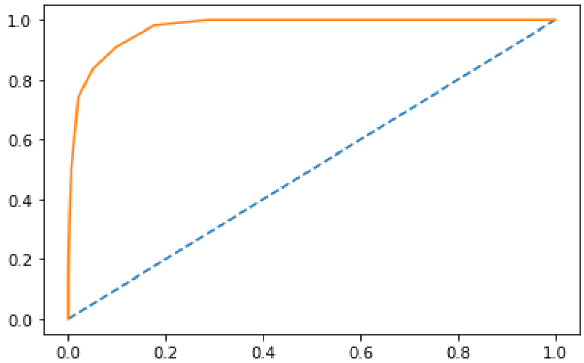


Figure 114. ROC for Test Data in KNN Grid Search for Mobile

Naïve Bayes – Grid Search

We split the data into train and test using train_test_split command and fit our Naïve Bayes model into the train data and then try to predict the outcome of using the test data. Then we compare the actual against the predicted to calculate the accuracy of the model. We will hyper tune the parameters that would enhance the outcome of the model.

```
GridSearchCV(cv=5, estimator=GaussianNB(), n_jobs=1,
```

```
                param_grid={'var_smoothing': [1e-08, 1e-07, 1e-06, 1e-05,
0.0001]},verbose=2)
```

Performance Metrices Naïve Bayes Grid Search

Model Score or Accuracy

- Accuracy for Training Data is 0.86
- Accuracy for Test Data is 0.85

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

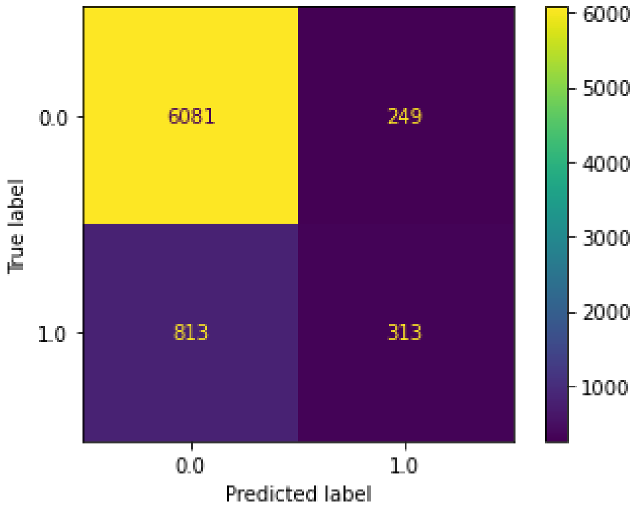


Figure 115. Confusion Matrix for Training Data in Naive Bayes Grid Search for Mobile

- For Test Data

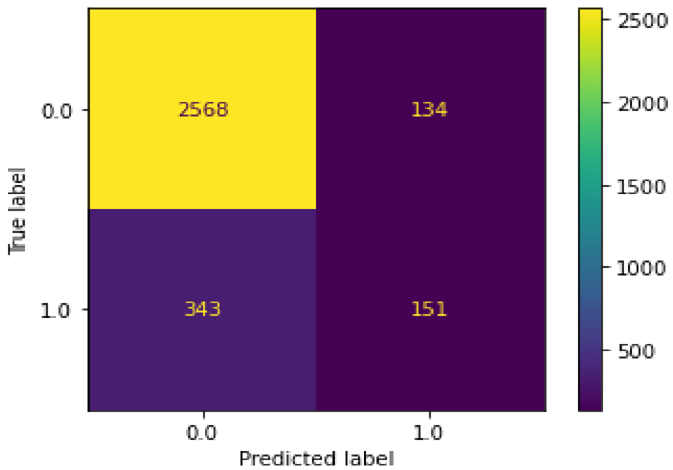


Figure 116. Confusion Matrix for Test Data in Naive Bayes Grid Search for Mobile

Classification Report

- For Training Data

0.8575643776824035

	precision	recall	f1-score	support
0.0	0.88	0.96	0.92	6330
1.0	0.56	0.28	0.37	1126
accuracy			0.86	7456
macro avg	0.72	0.62	0.65	7456
weighted avg	0.83	0.86	0.84	7456

Table 62. Classification Report for Training Data in Naive Bayes Grid Search for Mobile

- For Test Data

0.8507509386733417

	precision	recall	f1-score	support
0.0	0.88	0.95	0.92	2702
1.0	0.53	0.31	0.39	494
accuracy			0.85	3196
macro avg	0.71	0.63	0.65	3196
weighted avg	0.83	0.85	0.83	3196

Table 63. Classification Report for Test Data in Naive Bayes Grid Search for Mobile

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

AUC: 0.765

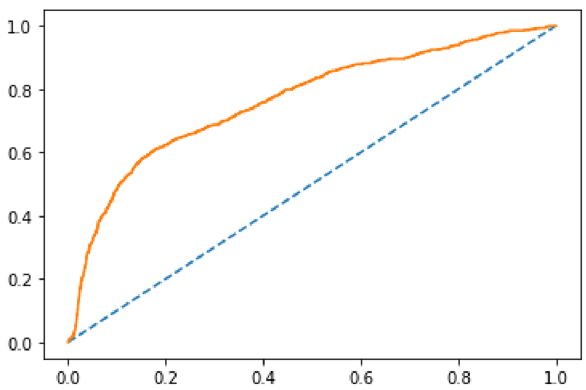


Figure 117. ROC for Training Data in Naive Bayes Grid Search for Mobile

- For Test Data

AUC: 0.774

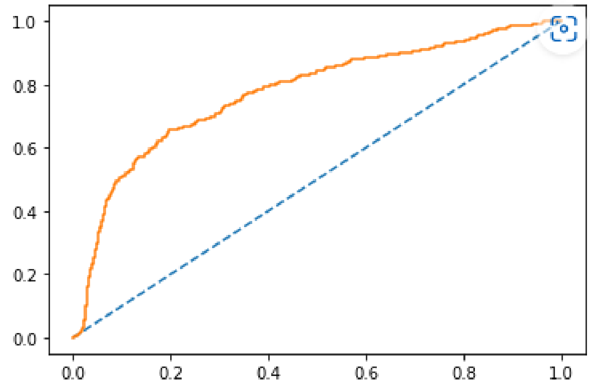


Figure 118. ROC for Test Data in Naive Bayes Grid Search for Mobile

Bagging – Grid Search

Bagging is an ensemble technique. Ensemble techniques are ML techniques that combine several base models to get an optimal model. Bagging is designed to improve the performance of existing ML algorithms used in statistical classification or regression. It is most used with tree-based algorithms. It is a parallel method.

```
GridSearchCV(cv=3,

estimator=BaggingClassifier(base_estimator=RandomForestClassifier(),n_e
stimators=100, random_state=1),
             param_grid={'bootstrap': [True, False], 'max_features':
[1, 2, 4],'max_samples': [0.5, 1.0]})
```

Performance Metrices Bagging Grid Search

Model Score or Accuracy

- Accuracy for Training Data is 1.0
- Accuracy for Test Data is 0.90

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

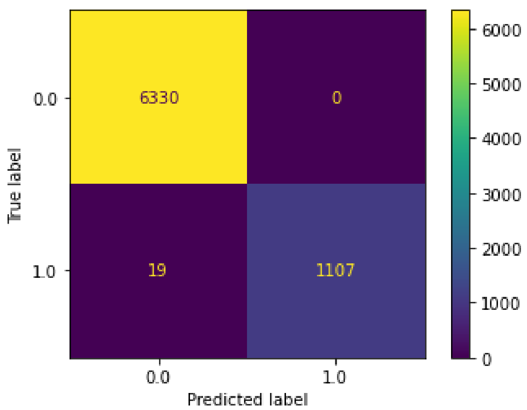


Figure 119. Confusion Matrix for Training Data in Bagging Grid Search for Mobile

➢ For Test Data

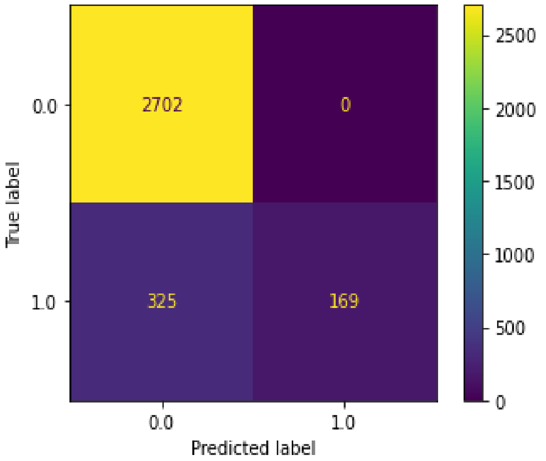


Figure 120. Confusion Matrix for Test Data in Bagging Grid Search for Mobile

Classification Report

➢ For Training Data

0.9974517167381974

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	6330
1.0	1.00	0.98	0.99	1126
accuracy			1.00	7456
macro avg	1.00	0.99	0.99	7456
weighted avg	1.00	1.00	1.00	7456

Table 64. Classification Report for Training Data in Bagging Grid Search for Mobile

➢ For Test Data

0.8983103879849812

	precision	recall	f1-score	support
0.0	0.89	1.00	0.94	2702
1.0	1.00	0.34	0.51	494
accuracy			0.90	3196
macro avg	0.95	0.67	0.73	3196
weighted avg	0.91	0.90	0.88	3196

Table 65. Classification Report for Test Data in Bagging Grid Search for Mobile

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

➢ For Training Data

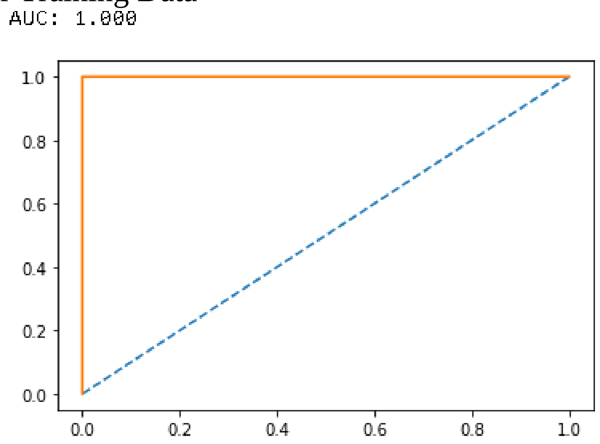


Figure 121. ROC for Training Data in Bagging Grid Search for Mobile

➢ For Test Data

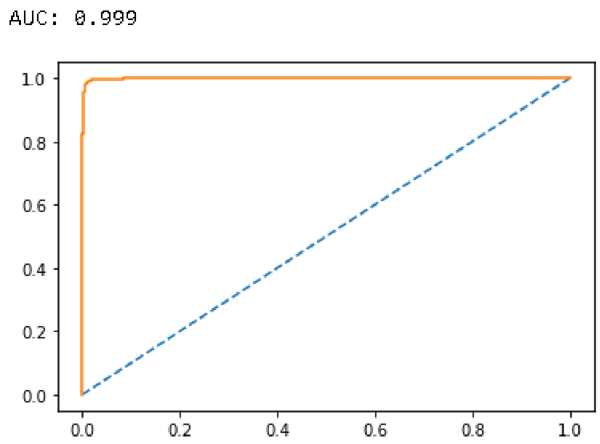


Figure 122. ROC for Test Data in Bagging Grid Search for Mobile

ADA Boosting – Grid Search

This model is used to increase the efficiency of binary classifiers, but now used to improve multiclass classifiers as well. ADA boosting can be applied on top of any classifier method to learn from its issues and bring about a more accurate model and this it is called "best out of the box classifier"

```
GridSearchCV(cv=3, estimator=AdaBoostClassifier(), n_jobs=1,
             param_grid={'learning_rate': [0.001, 0.01, 0.1],
                         'n_estimators': [500, 1000, 2000]})
```

Performance Metrices Ada Boosting Grid Search

<u>Model Score or Accuracy</u>

Accuracy for Training Data is 0.89
Accuracy for Test Data is 0.88

<u>Confusion Matrix</u>

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

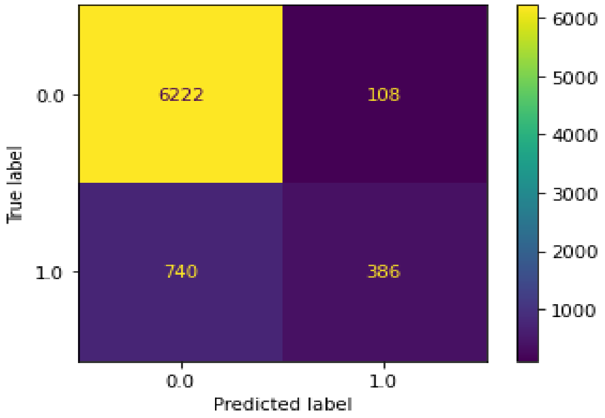


Figure 123. Confusion Matrix for Training Data in Ada Boosting Grid Search for Mobile

- For Test Data

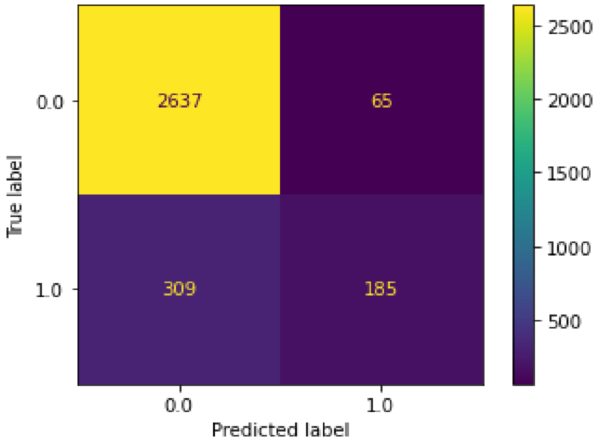


Figure 124. Confusion Matrix for Test Data in Ada Boosting Grid Search for Mobile

Classification Report

- For Training Data

0.8862660944206009

	precision	recall	f1-score	support
0.0	0.89	0.98	0.94	6330
1.0	0.78	0.34	0.48	1126
accuracy			0.89	7456
macro avg	0.84	0.66	0.71	7456
weighted avg	0.88	0.89	0.87	7456

Table 66. Classification Report for Training Data in Ada Boosting Grid Search for Mobile

- For Test Data

0.8829787234042553

	precision	recall	f1-score	support
0.0	0.90	0.98	0.93	2702
1.0	0.74	0.37	0.50	494
accuracy			0.88	3196
macro avg	0.82	0.68	0.72	3196
weighted avg	0.87	0.88	0.87	3196

Table 67. Classification Report for Test Data in Ada Boosting Grid Search for Mobile

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

AUC: 0.883

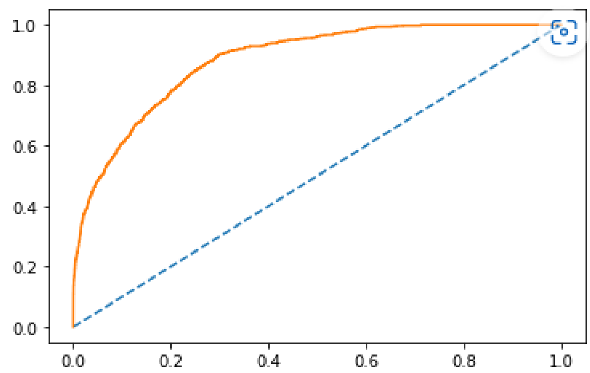


Figure 125. ROC for Training Data in Ada Boosting Grid Search for Mobile

- For Test Data

AUC: 0.872

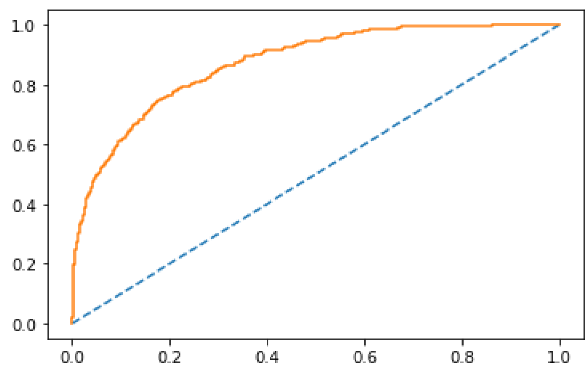


Figure 126. ROC for Test Data in Ada Boosting Grid Search for Mobile

Gradient Boosting – Grid Search

This model is just like the ADA Boosting works by sequentially adding the misidentified predictors and under-fitted predictions to the ensemble, ensuring the errors identified previously are corrected. The major difference lies in what it does with the mis-identified values of the previous weak learner.

```
GridSearchCV(cv=3, estimator=GradientBoostingClassifier(),
             param_grid={'n_estimators': range(1000, 2000, 3000)})
```

Performance Metrices Gradient Boosting Grid Search

<u>Model Score or Accuracy</u>

- Accuracy for Training Data is 1.0
- Accuracy for Test Data is 0.97

<u>Confusion Matrix</u>

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

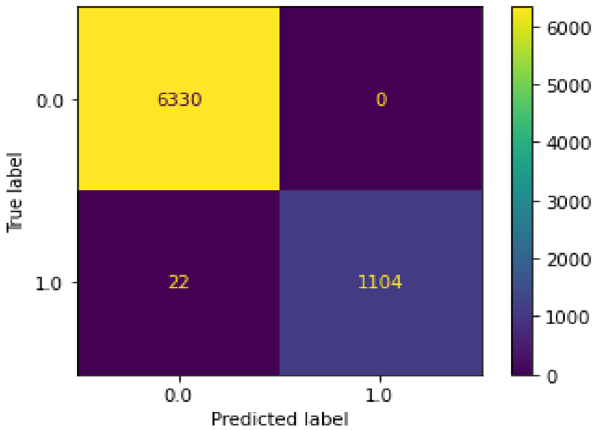


Figure 127. Confusion Matrix for Training Data in Gradient Boosting Grid Search for Mobile

- For Test Data

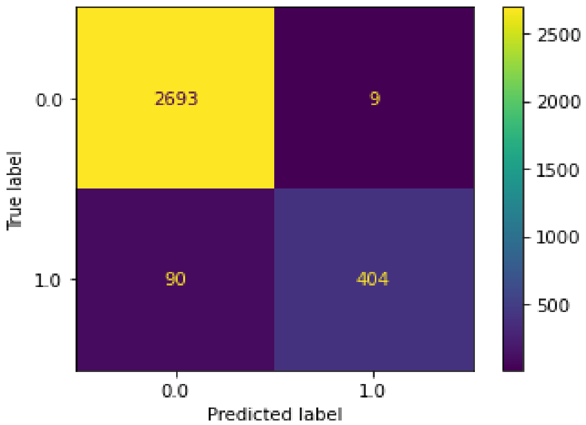


Figure 128. Confusion Matrix for Test Data in Gradient Boosting Grid Search for Mobile

Classification Report

- For Training Data

0.9970493562231759

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	6330
1.0	1.00	0.98	0.99	1126
accuracy			1.00	7456
macro avg	1.00	0.99	0.99	7456
weighted avg	1.00	1.00	1.00	7456

Table 68. Classification Report for Training Data in Gradient Boosting Grid Search for Mobile

- For Test Data

0.9690237797246558

	precision	recall	f1-score	support
0.0	0.97	1.00	0.98	2702
1.0	0.98	0.82	0.89	494
accuracy			0.97	3196
macro avg	0.97	0.91	0.94	3196
weighted avg	0.97	0.97	0.97	3196

Table 69. Classification Report for Test Data in Gradient Boosting Grid Search for Mobile

<u>ROC and AUC</u>

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

AUC: 1.000

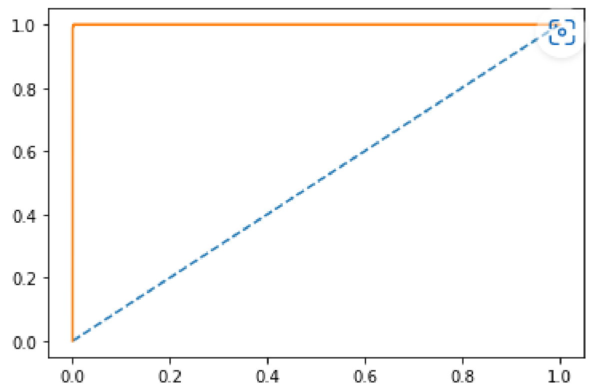


Figure 129. ROC for Training Data in Gradient Boosting Grid Search for Mobile

- For Test Data

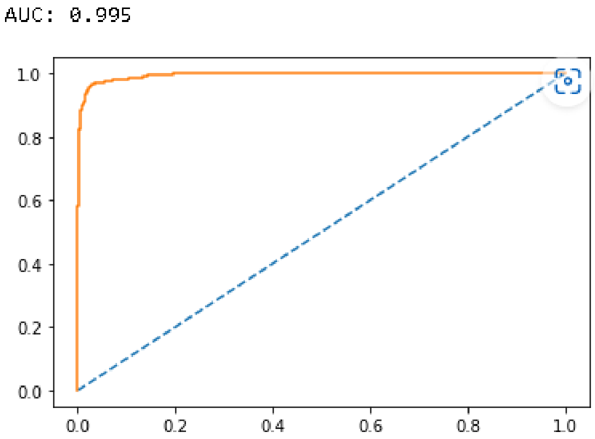


Figure 130. ROC for Test Data in Gradient Boosting Grid Search for Mobile

SMOTE

SMOTE (synthetic minority oversampling technique) is one of the most commonly used oversampling methods to solve the imbalance problem.
It aims to balance class distribution by randomly increasing minority class examples by replicating them.

SMOTE synthesises new minority instances between existing minority instances. It generates the virtual training records by linear interpolation for the minority class. These synthetic training records are generated by randomly selecting one or more of the k-nearest neighbors for each example in the minority class. After the oversampling process, the data is reconstructed and several classification models can be applied for the processed data.

```
New data shape after SMOTE (1188, 15)
```

Logistic Regression Model – SMOTE

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

Performance Metrices Logistic Regression SMOTE

Model Score or Accuracy

- Accuracy for Training Data is 0.73
- Accuracy for Test Data is 0.71

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset. It will allow us to visualise the performance of the Logistic Regression Model.

- For Training Data

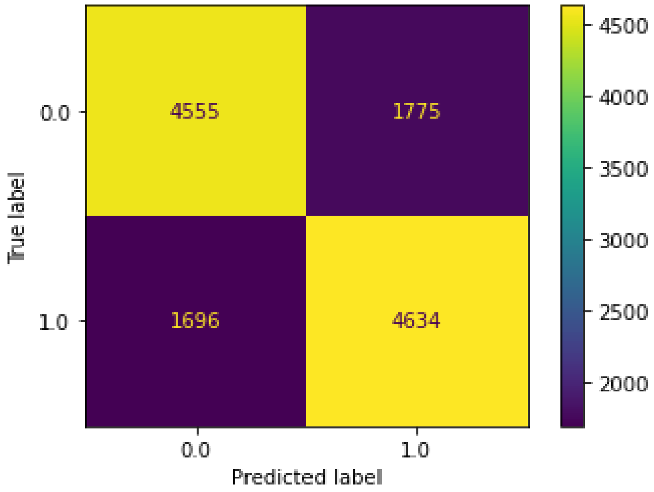


Figure 131. Confusion Matrix for Training Data in Logistic Regression SMOTE for Mobile

- For Test Data

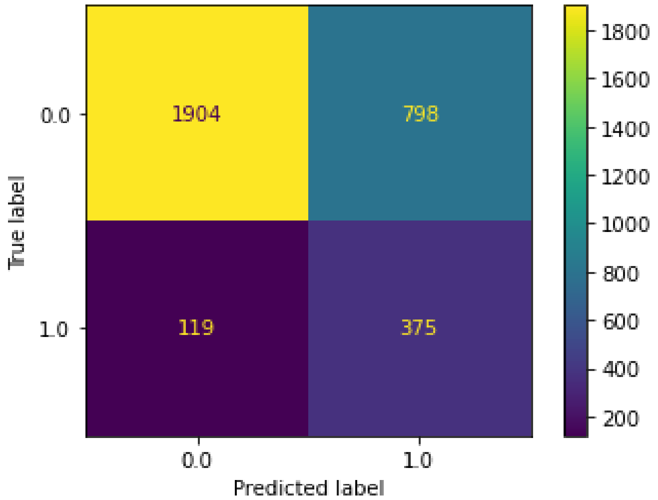


Figure 132. Confusion Matrix for Test Data in Logistic Regression SMOTE for Mobile

Classification Report

- For Training Data

0.725829383886256

	precision	recall	f1-score	support
0.0	0.73	0.72	0.72	6330
1.0	0.72	0.73	0.73	6330
accuracy			0.73	12660
macro avg	0.73	0.73	0.73	12660
weighted avg	0.73	0.73	0.73	12660

Table 70. Classification Report for Training Data in Logistic Regression SMOTE for Mobile

➢ For Test Data

0.7130788485607009

	precision	recall	f1-score	support
0.0	0.94	0.70	0.81	2702
1.0	0.32	0.76	0.45	494
accuracy			0.71	3196
macro avg	0.63	0.73	0.63	3196
weighted avg	0.85	0.71	0.75	3196

Table 71. Classification Report for Test Data in Logistic Regression SMOTE for Mobile

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

➢ For Training Data

AUC: 0.788

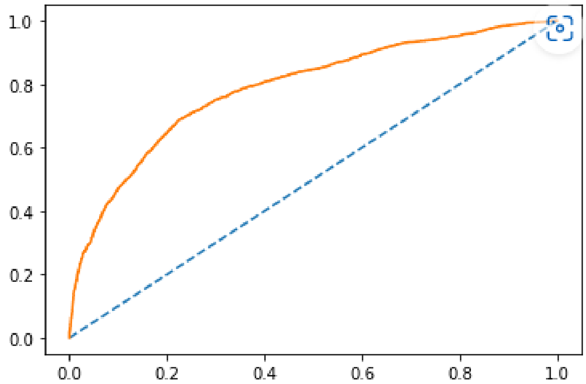


Figure 133. ROC for Training Data in Logistic Regression SMOTE for Mobile

➢ For Test Data

AUC: 0.794

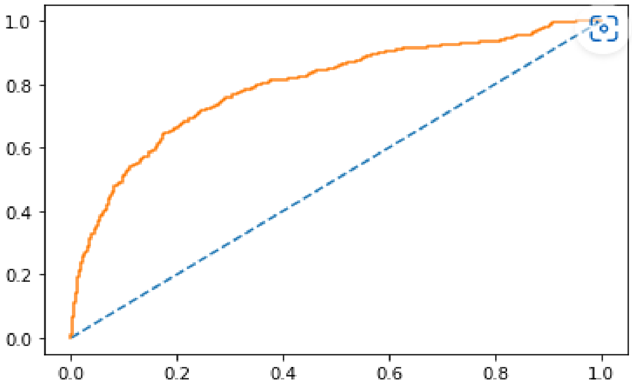


Figure 134. ROC for Test Data in Logistic Regression SMOTE for Mobile

KNN – SMOTE

Performance Metrices Basic KNN SMOTE

Model Score or Accuracy

- Accuracy for Training Data is 0.99
- Accuracy for Test Data is 0.97

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

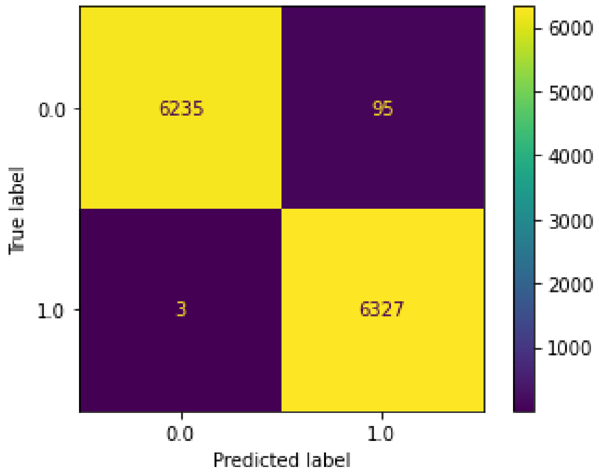


Figure 135. Confusion Matrix for Training Data in KNN SMOTE for Mobile

- For Test Data

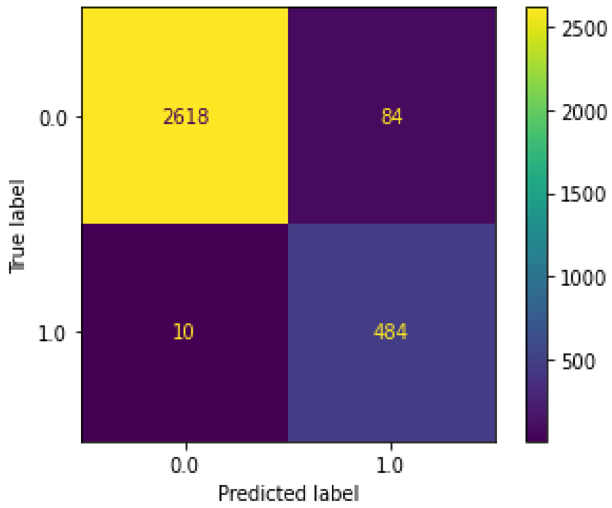


Figure 136. Confusion Matrix for Test Data in KNN SMOTE for Mobile

Classification Report

- For Training Data

0.9922590837282781

	precision	recall	f1-score	support
0.0	1.00	0.98	0.99	6330
1.0	0.99	1.00	0.99	6330
accuracy			0.99	12660
macro avg	0.99	0.99	0.99	12660
weighted avg	0.99	0.99	0.99	12660

Table 72. Classification Report for Training Data in KNN SMOTE for Mobile

- For Test Data

0.9705882352941176

	precision	recall	f1-score	support
0.0	1.00	0.97	0.98	2702
1.0	0.85	0.98	0.91	494
accuracy			0.97	3196
macro avg	0.92	0.97	0.95	3196
weighted avg	0.97	0.97	0.97	3196

Table 73. Classification Report for Test Data in KNN SMOTE for Mobile

<u>ROC and AUC</u>

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

AUC: 1.000

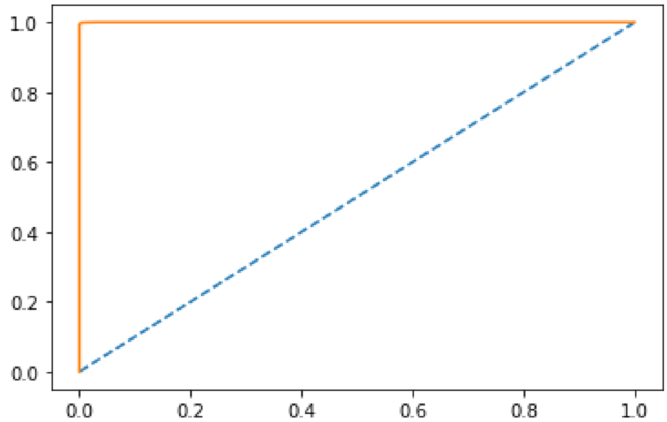


Figure 137. ROC for Training Data in KNN SMOTE for Mobile

➢ For Test Data

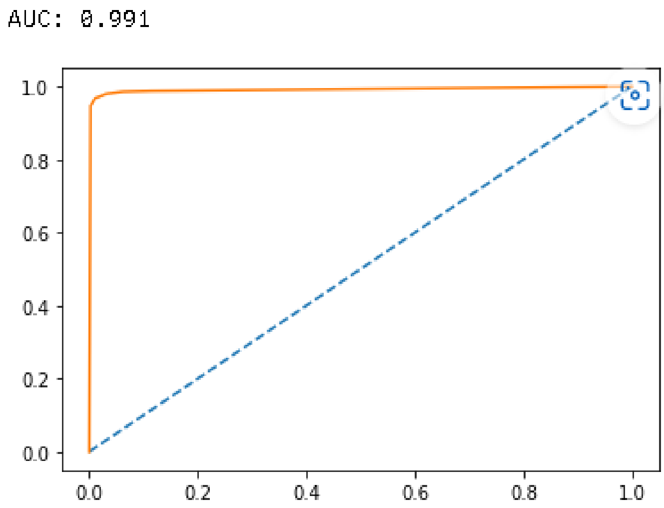


Figure 138. ROC for Test Data in KNN SMOTE for Mobile

Naïve Bayes – SMOTE

Performance Metrices Naïve Bayes SMOTE

<u>Model Score or Accuracy</u>

- ➢ Accuracy for Training Data is 0.68
- ➢ Accuracy for Test Data is 0.66

<u>Confusion Matrix</u>

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

➢ For Training Data

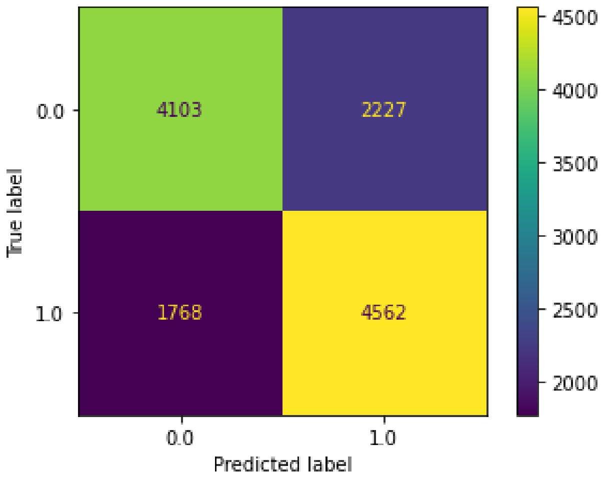


Figure 139. Confusion Matrix for Training Data in Naive Bayes SMOTE for Mobile

➢ For Test Data

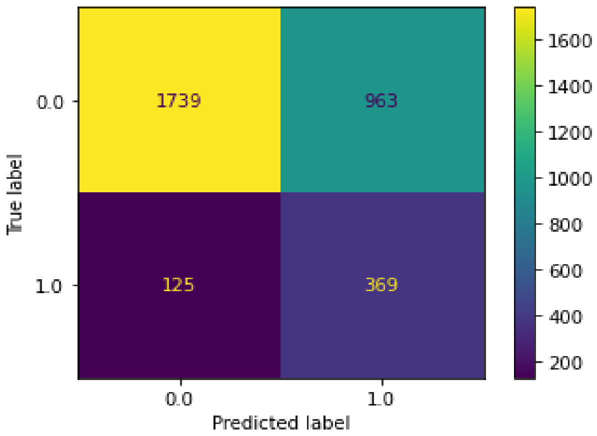


Figure 140. Confusion Matrix for Test Data in Naive Bayes SMOTE for Mobile

Classification Report

➢ For Training Data

0.6844391785150079

	precision	recall	f1-score	support
0.0	0.70	0.65	0.67	6330
1.0	0.67	0.72	0.70	6330
accuracy			0.68	12660
macro avg	0.69	0.68	0.68	12660
weighted avg	0.69	0.68	0.68	12660

Table 74. Classification Report for Training Data in Naive Bayes SMOTE for Mobile

➢ For Test Data

0.6595744680851063

	precision	recall	f1-score	support
0.0	0.93	0.64	0.76	2702
1.0	0.28	0.75	0.40	494
accuracy			0.66	3196
macro avg	0.60	0.70	0.58	3196
weighted avg	0.83	0.66	0.71	3196

Table 75. Classification Report for Test Data in Naive Bayes SMOTE for Mobile

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

AUC: 0.765

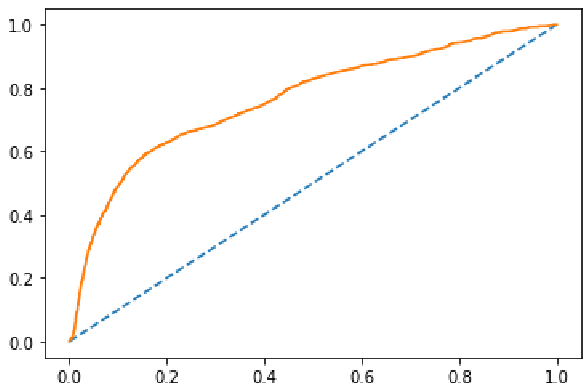


Figure 141. ROC for Training Data in Naive Bayes SMOTE for Mobile

- For Test Data

AUC: 0.770

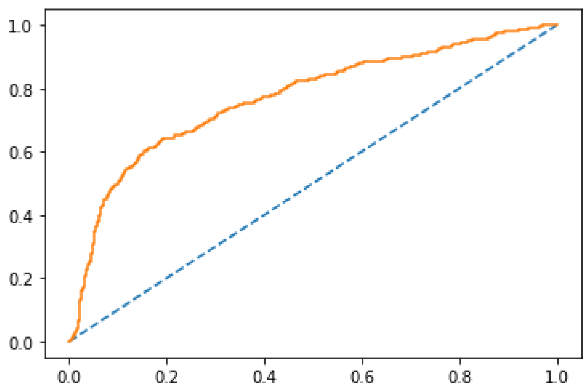


Figure 142. ROC for Test Data in Naive Bayes SMOTE for Mobile

Bagging – SMOTE

```
BaggingClassifier(base_estimator=RandomForestClassifier(),
n_estimators=100,random_state=1)
```

Performance Metrices Bagging SMOTE

Model Score or Accuracy

- Accuracy for Training Data is 1.0
- Accuracy for Test Data is 0.98

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

➢ For Training Data

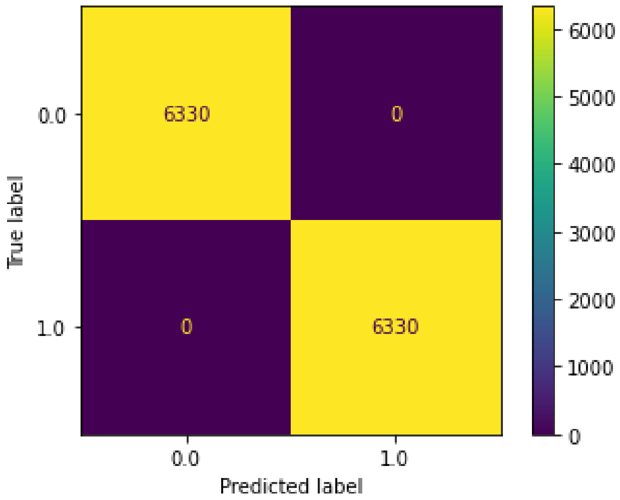


Figure 143. Confusion Matrix for Training Data in Bagging SMOTE for Mobile

➢ For Test Data

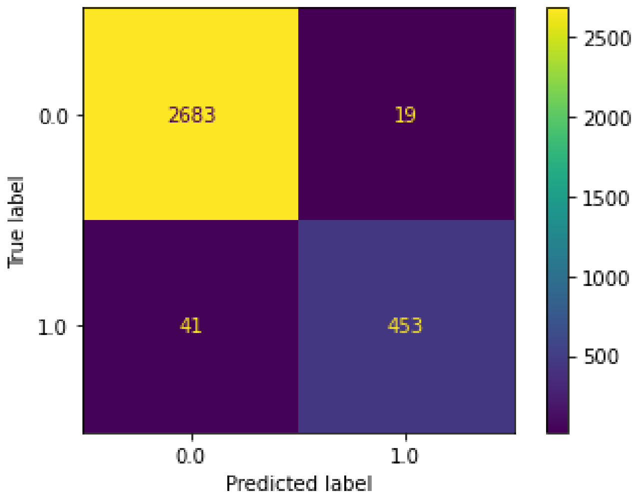


Figure 144. Confusion Matrix for Test Data in Bagging SMOTE for Mobile

Classification Report

➢ For Training Data

1.0

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	6330
1.0	1.00	1.00	1.00	6330
accuracy			1.00	12660
macro avg	1.00	1.00	1.00	12660
weighted avg	1.00	1.00	1.00	12660

Table 76. Classification Report for Training Data in Bagging SMOTE for Mobile

➢ For Test Data

0.9812265331664581

	precision	recall	f1-score	support
0.0	0.98	0.99	0.99	2702
1.0	0.96	0.92	0.94	494
accuracy			0.98	3196
macro avg	0.97	0.95	0.96	3196
weighted avg	0.98	0.98	0.98	3196

Table 77. Classification Report for Test Data in Bagging SMOTE for Mobile

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

➢ For Training Data

AUC: 1.000

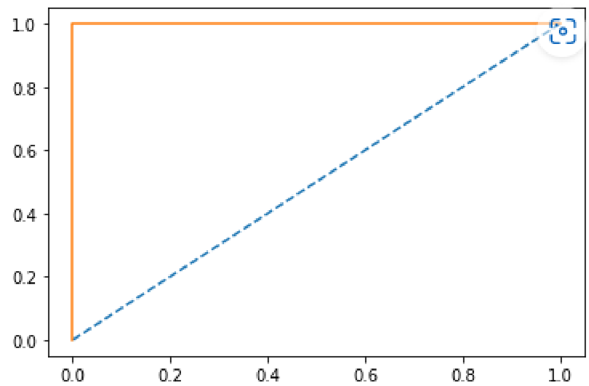


Figure 145. ROC for Training Data in Bagging SMOTE for Mobile

➢ For Test Data

AUC: 0.998

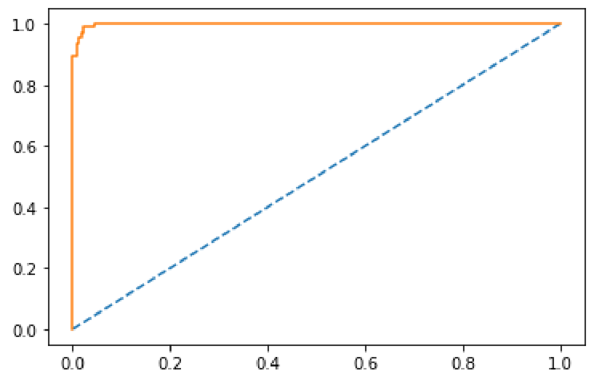


Figure 146. ROC for Test Data in Bagging SMOTE for Mobile

ADA Boosting – SMOTE

```
AdaBoostClassifier(n_estimators=100, random_state=1)
```

Performance Metrices Ada Boosting SMOTE

Model Score or Accuracy

- Accuracy for Training Data is 0.84
- Accuracy for Test Data is 0.82

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

- For Training Data

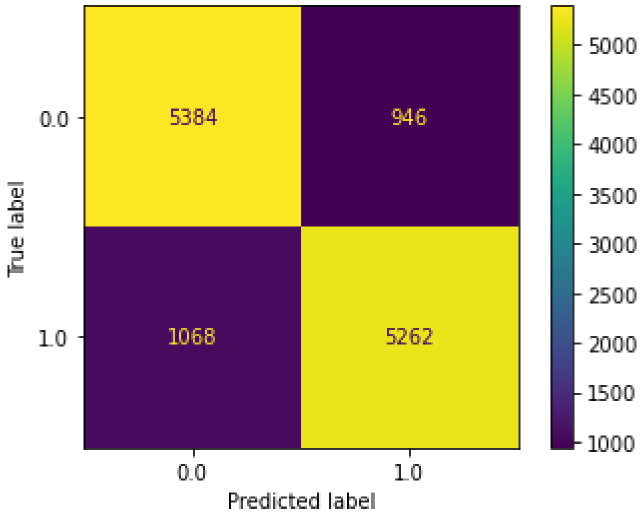


Figure 147. Confusion Matrix for Training Data in Ada Boosting SMOTE for Mobile

- For Test Data

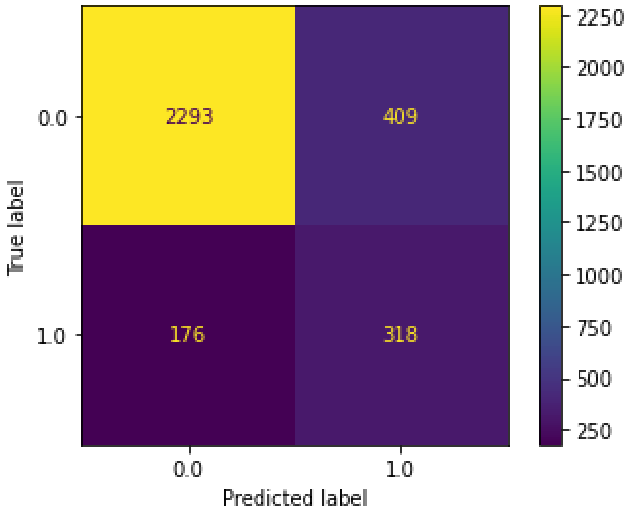


Figure 148. Confusion Matrix for Test Data in Ada Boosting SMOTE for Mobile

Classification Report

- For Training Data

0.8409162717219589

	precision	recall	f1-score	support
0.0	0.83	0.85	0.84	6330
1.0	0.85	0.83	0.84	6330
accuracy			0.84	12660
macro avg	0.84	0.84	0.84	12660
weighted avg	0.84	0.84	0.84	12660

Table 78. Classification Report for Training Data in Ada Boosting SMOTE for Mobile

- For Test Data

0.8169586983729662

	precision	recall	f1-score	support
0.0	0.93	0.85	0.89	2702
1.0	0.44	0.64	0.52	494
accuracy			0.82	3196
macro avg	0.68	0.75	0.70	3196
weighted avg	0.85	0.82	0.83	3196

Table 79. Classification Report for Test Data in Ada Boosting SMOTE for Mobile

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

AUC: 0.926

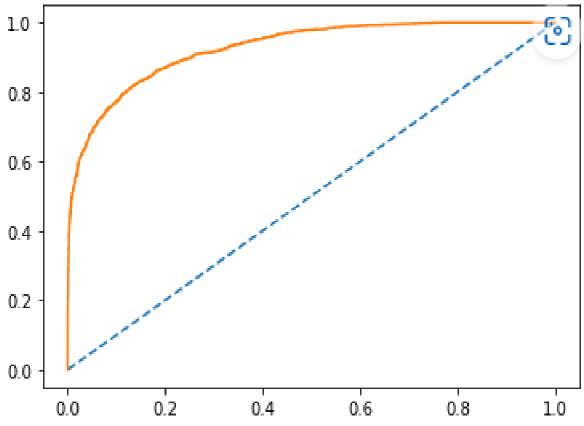


Figure 149. ROC for Training Data in Ada Boosting SMOTE for Mobile

➢ For Test Data

AUC: 0.845

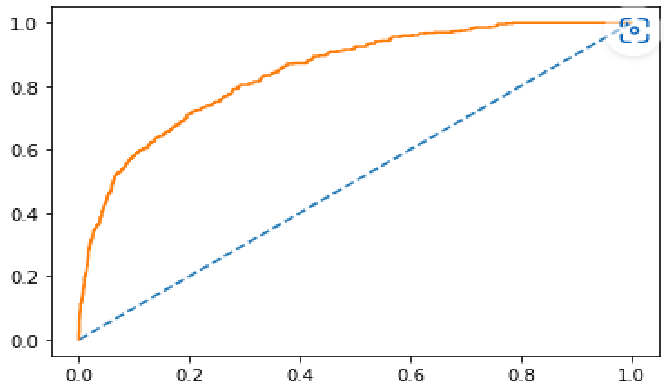


Figure 150. ROC for Test Data in Ada Boosting SMOTE for Mobile

Gradient Boosting – SMOTE

```
GradientBoostingClassifier(random_state=1)
```

Performance Metrices Gradient Boosting SMOTE

Model Score or Accuracy

- ➢ Accuracy for Training Data is 0.90
- ➢ Accuracy for Test Data is 0.0.87

Confusion Matrix

We will now create a Confusion Matrix which will be used to describe the performance of a classifier on our Test and Train Dataset.

➢ For Training Data

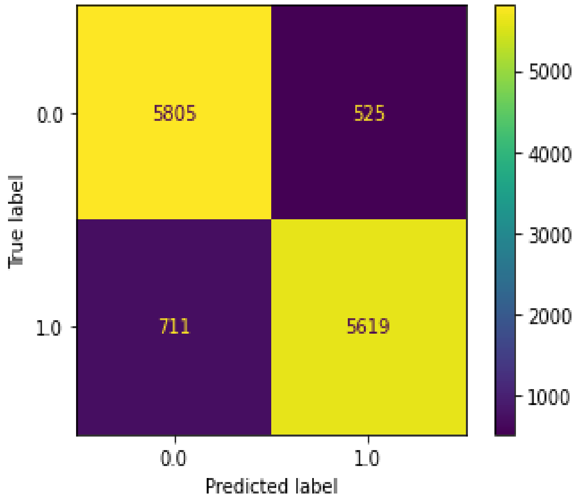


Figure 151. Confusion Matrix for Training Data in Gradient Boosting SMOTE for Mobile

➢ For Test Data

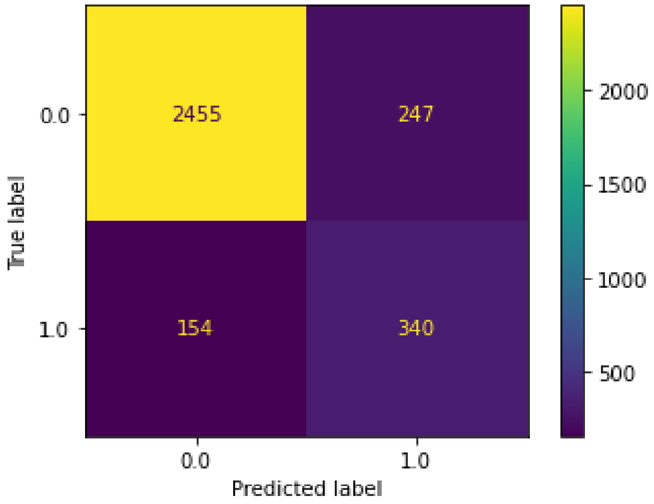


Figure 152. Confusion Matrix for Test Data in Gradient Boosting SMOTE for Mobile

Classification Report

➢ For Training Data

0.9023696682464455

	precision	recall	f1-score	support
0.0	0.89	0.92	0.90	6330
1.0	0.91	0.89	0.90	6330
accuracy			0.90	12660
macro avg	0.90	0.90	0.90	12660
weighted avg	0.90	0.90	0.90	12660

Table 80. Classification Report for Training Data in Gradient Boosting SMOTE for Mobile

➢ For Test Data

0.8745306633291614

	precision	recall	f1-score	support
0.0	0.94	0.91	0.92	2702
1.0	0.58	0.69	0.63	494
accuracy			0.87	3196
macro avg	0.76	0.80	0.78	3196
weighted avg	0.89	0.87	0.88	3196

Table 81. Classification Report for Test Data in Gradient Boosting SMOTE for Mobile

ROC and AUC

An ROC Curve or Receiver Operating Characteristic curve is a graph showing the performance of a classification model. The curve is plotted between sensitivity and (1-specificity). (1-specificity) is also known as false positive rate and sensitivity is also known as True Positive Rate.

- For Training Data

AUC: 0.966

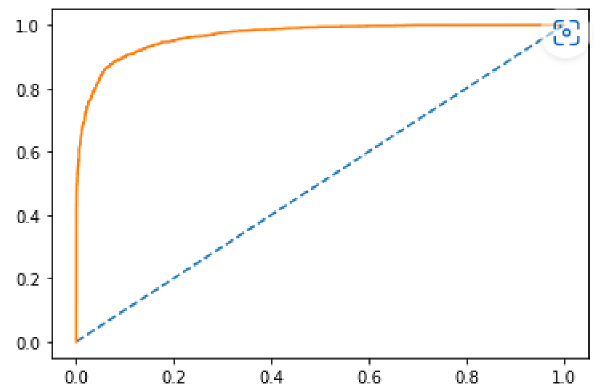


Figure 153. ROC for Training Data in Gradient Boosting SMOTE for Mobile

- For Test Data

AUC: 0.897

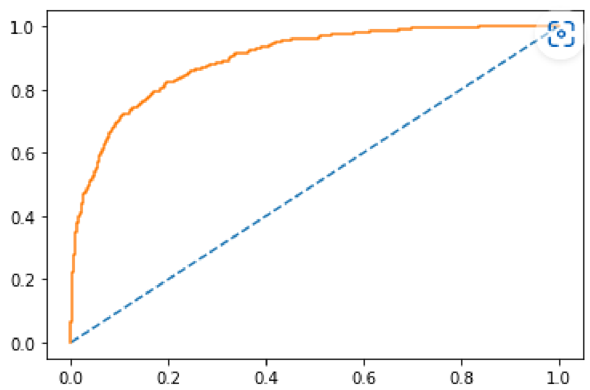


Figure 154. ROC for Test Data in Gradient Boosting Grid Search for Mobile

d). Interpretation of the hyper tuned models and Using SMOTE Techniques models.

Grid Search Model Tuning		Accuracy		Precision		Recall		F1 Score		AUC	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Logistic Regression	No Taken Product	0.87	0.87	0.87	0.87	0.99	0.99	0.93	0.93	0.79	0.8
	Yes Taken Product			0.72	0.74	0.2	0.23	0.31	0.35		
KNN	No Taken Product	0.95	0.92	0.94	0.92	1	0.99	0.97	0.95	0.99	0.97
	Yes Taken Product			0.98	0.94	0.67	0.5	0.79	0.66		
Naïve Bayes	No Taken Product	0.86	0.85	0.88	0.88	0.96	0.95	0.92	0.92	0.77	0.77
	Yes Taken Product			0.56	0.53	0.28	0.31	0.37	0.39		
Bagging	No Taken Product	1	0.9	1	0.89	1	1	1	0.94	1	0.99
	Yes Taken Product			1	1	0.98	0.34	0.99	0.51		
Ada Boosting	No Taken Product	0.89	0.87	0.89	0.9	0.98	0.98	0.94	0.93	0.88	0.87
	Yes Taken Product			0.78	0.74	0.34	0.37	0.48	0.5		
Gradient Boosting	No Taken Product	1	0.97	1	0.97	1	1	1	0.98	1	0.99
	Yes Taken Product			1	0.98	0.88	0.82	0.99	0.89		

Table 82. Model Tuning Comparison for Mobile

SMOTE		Accuracy		Precision		Recall		F1 Score		AUC	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Logistic Regression	No Taken Product	0.73	0.71	0.73	0.94	0.72	0.7	0.72	0.81	0.79	0.79
	Yes Taken Product			0.72	0.32	0.73	0.76	0.73	0.45		
KNN	No Taken Product	0.99	0.97	1	1	0.98	0.97	0.99	0.98	1	0.99
	Yes Taken Product			0.99	0.85	1	0.98	0.99	0.91		
Naïve Bayes	No Taken Product	0.68	0.66	0.7	0.93	0.65	0.64	0.67	0.76	0.77	0.77
	Yes Taken Product			0.67	0.28	0.72	0.75	0.7	0.4		
Bagging	No Taken Product	1	0.98	1	0.98	1	0.99	1	0.99	1	0.99
	Yes Taken Product			1	0.96	1	0.92	1	0.94		
Ada Boosting	No Taken Product	0.84	0.93	0.83	0.93	0.85	0.85	0.84	0.89	0.93	0.85
	Yes Taken Product			0.85	0.44	0.83	0.64	0.84	0.52		
Gradient Boosting	No Taken Product	0.9	0.87	0.89	0.94	0.92	0.91	0.9	0.92	0.97	0.9
	Yes Taken Product			0.91	0.58	0.89	0.69	0.9	0.63		

Table 83. Using SMOTE models comparison for Mobile

- According to problem we will focus on the Customer who have taken the product.
- There is not much improvement in performance for the Logistic Regression model after hyper tuning and SMOTE technique. For LR model performance declined after applying SMOTE Technique.
- For KNN after Hyper tuning model performance declined and after applying SMOTE Technique there is improvement in process but Precision is good for training set but decreases in Test Set
- There is not much improvement in performance for the Naïve Bayes model after hyper tuning and SMOTE technique. For Naïve Bayes model performance declined after applying SMOTE Technique.
- For Bagging model performance declined when hyper tuning model but in case of SMOTE technique model is performing well.
- For the ADA Boosting Model there is not much improvement in performance after hyper tuning and SMOTE technique. For ADA Boosting model performance declined after applying SMOTE Technique.
- For the Gradient Boosting model improvement in performance after hyper tuning but model performance declined after applying SMOTE Technique.

1.5 Final interpretation / recommendation

Interpretation of the most optimum model

- Based on our model evaluation, performing visual inspection, stacking and bagging models. We finally are able to combine all the results and can clearly infer that after Using SMOTE Technique Bagging using base estimator as Random Forest is the best performing model for both Mobile phone users and Laptop users, with the highest accuracy of 98%.
- Bagging is performing well in terms of Recall, Precision and F1-Score for both Laptop and Mobile users.

- The desired metric for this problem which is Precision, is also observed to be significantly the highest for Random Forest models with 96% for Laptop users and 96% for Mobile phone users.
- From this observation Precision quantifies the number of positive class predictions that actually belong to the positive class. Hence this should be interpreted as 96% of total customers who use Laptops who were predicted to purchase the product actually purchases the product.
- Similarly, among the total customers who use Mobile phones 96% of all the customers predicted to purchase the product actually buys the product.
- Hence, on building two different models based on preferred device of our customers, both the model provided highly satisfactory results using SMOTE Technique Bagging using base estimator as Random Forest whose results are statistically significant and are safe to be deployed for further evaluation of test cases.

Business Implications

- By selecting the right model, the prediction capabilities of that model greatly increase. This makes the model more reliable for decision making. While using Bagging base estimator as Random Forest while applying SMOTE Technique, we then train the model considering the whole dataset as train set, and the resulting model will be ready to make predictions provided with independent variables.
- In our case, provided with the social media components of personnel which includes the time Customers spends in travel websites, number of likes received and given, etc. we will now be able to predict the likeliness of that customer to purchase the travel packages offered by the aviation company with an accuracy of 98%.
- We the help of this model company can identify which customers can purchase the product in the new future. Also, this helps in better reach and target the audience accordingly.
- This can also increase the traffic on the company’s site resulting in better minimizing the click per cost expense for the company.

Recommendations

- Target the customers who have not checked in in the last few weeks as outstation checkin is the most important feature.
- Plan a campaign for people who spend a lot of time on the page and target the customers more on the mobile device.
- Adults play a critical role in the buying decision, thus they should be targeted more wisely.
- The aviation company should invest their budget in acquiring the dataset from the networking platform to learn about their behavior and target these customers.
- Using optimum model i.e. SMOTE Technique Bagging using base estimator as Random Forest can also increase the traffic on the company’s site resulting in better minimizing the click per cost expense for the company.
- As, the higher the number of hits on website increases, more chances of purchasing the product also increases bringing in the surge in revenues for the company.

- This in turn provided a targeted approach for the aviation company to approach their customer base, thereby reducing cost and making the most returns out of the expenditure they put in digital marketing campaigns.
- Company should come up with discount offer the user who travels for medical related travels as this will have good customer experience in these unprecedented times and it will increase brand value.

THE END!!!