Social Media

Tourism Project 2

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 behaviors 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.

Objective:

To develop separate propensity models for Laptop and Mobile users to predict their likelihood of purchasing tickets based on their social and digital behavior data. Multiple models will be built, analyzed, and compared to determine the most accurate and effective approach

Data Dictionary:

User ID	Unique ID of user
Taken products	Buy ticket in next month
Yearly avg view on travel page	Average yearly views on any travel related page by user
Preferred Devices	Through which device user preferred to do login
Total likes on outstation checking given	Total number of likes given by a user on out of station checking in last year
Yearly avg outstation check in	Average number of out of station check-in done by user
Member in family	Total number of relationships 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 like on outstation checking received	Total number of likes received by a user on out of station checking in last year
Week since last outstation checking	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)
Monthly 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 min spend on traveling page	Average time spend on the company page by user on daily basis

Images Used:

Img 1	Mobile data information
Img 2	Laptop data information
Img 3	Balancing dataset
Img 4	Scaled mobile data
Img 5	Scaled laptop data
Img 6	Adding constant to Mobile dataset
Img 7	Mobile model summary
Img 8	Logistic regression mobile train model
Img 9	Logistic regression mobile test model

Img 10	Logistic regression ROC AUC curve
Img 11	Tune Logistic regression mobile train model
Img 12	Tuned Logistic regression mobile test model
Img 13	Adding constant to laptop dataset
Img 14	Laptop model summary
Img 15	Logistic regression laptop train model
Img 16	Logistic regression laptop test model
Img 17	Logistic regression ROC AUC curve
Img 18	Tuned Logistic regression train model
Img 19	Tuned Logistic regression test model
Img 20	Decision tree mobile train model
Img 21	Decision tree mobile test model
Img 22	Tuned Decision tree mobile train model
Img 23	Tuned Decision tree mobile test model
Img 24	Tuned Decision tree mobile ROC AUC curve
Img 25	Decision tree laptop train model
Img 26	Decision tree laptop test model
Img 27	Tuned Decision tree laptop train model
Img 28	Tuned Decision tree laptop test model
Img 29	Tuned decision tree laptop ROC AUC curve
Img 30	Random forest mobile train model
Img 31	Random forest mobile test model
Img 32	Random forest mobile ROC AUC curve
Img 33	Tuned Random Forest mobile train model
Img 34	Tuned Random Forest mobile test model
Img 35	Random forest laptop train model
Img 36	Random forest laptop test model
Img 37	Random forest laptop ROC AUC curve
Img 38	Tuned Random Forest laptop train model
Img 39	Tuned Random Forest laptop test model
Img 40	All model comparison for mobile data
Img 41	All model comparison for laptop user
Img 42	Important features for mobile users
Img 43	Important features for laptop users

Data Preprocessing:

Data split into mobile users and laptop users

Mobile Data Information:

```
<class 'pandas.core.frame.DataFrame'>
Index: 10652 entries, 0 to 11759
Data columns (total 15 columns):
 #   Column                                    Non-Null Count  Dtype
---  ------                                    --------------  -----
 0   Taken_product                             10652 non-null  int64
 1   Yearly_avg_view_on_travel_page            10652 non-null  float64
 2   total_likes_on_outstation_checkin_given   10652 non-null  float64
 3   yearly_avg_Outstation_checkins            10652 non-null  float64
 4   member_in_family                          10652 non-null  int64
 5   preferred_location_type                   10652 non-null  int64
 6   Yearly_avg_comment_on_travel_page         10652 non-null  float64
 7   total_likes_on_outofstation_checkin_received  10652 non-null  int64
 8   week_since_last_outstation_checkin        10652 non-null  int64
 9   following_company_page                    10652 non-null  int64
 10  montly_avg_comment_on_company_page        10652 non-null  int64
 11  working_flag                              10652 non-null  int64
 12  travelling_network_rating                 10652 non-null  int64
 13  Adult_flag                                10652 non-null  int64
 14  Daily_Avg_mins_spend_on_traveling_page    10652 non-null  int64
dtypes: float64(4), int64(11)
memory usage: 1.3 MB
```

Img 1

- There are 10652 mobile users are present in the dataset

Laptop Users Information:

```
<class 'pandas.core.frame.DataFrame'>
Index: 1108 entries, 5881 to 11758
Data columns (total 15 columns):
 #   Column                                    Non-Null Count  Dtype
---  ------                                    --------------  -----
 0   Taken_product                             1108 non-null   int64
 1   Yearly_avg_view_on_travel_page            1108 non-null   float64
 2   total_likes_on_outstation_checkin_given   1108 non-null   float64
 3   yearly_avg_Outstation_checkins            1108 non-null   float64
 4   member_in_family                          1108 non-null   int64
 5   preferred_location_type                   1108 non-null   int64
 6   Yearly_avg_comment_on_travel_page         1108 non-null   float64
 7   total_likes_on_outofstation_checkin_received  1108 non-null   int64
 8   week_since_last_outstation_checkin        1108 non-null   int64
 9   following_company_page                    1108 non-null   int64
 10  montly_avg_comment_on_company_page        1108 non-null   int64
 11  working_flag                              1108 non-null   int64
 12  travelling_network_rating                 1108 non-null   int64
 13  Adult_flag                                1108 non-null   int64
 14  Daily_Avg_mins_spend_on_traveling_page    1108 non-null   int64
dtypes: float64(4), int64(11)
memory usage: 138.5 KB
```

Img 2

- There are 1108 laptop users are present in the dataset

Balancing the dataset of target class

Mobile Users

```
Taken_product
1    9032
0    9032
Name: count, dtype: int64
Taken_product
1    0.5
0    0.5
Name: proportion, dtype: float64
```

Laptop Users

```
Taken_product
0    832
1    832
Name: count, dtype: int64
Taken_product
0    0.5
1    0.5
Name: proportion, dtype: float64
```

Img 5

- Both classes are unbalanced

- Used the SMOTE function to oversampled the data and make both class 1 and 2 balanced

Split and scaling the dataset

Split the mobile and laptop dataset into train and test set

Mobile users

```
(12644, 14)
(5420, 14)
```

Laptop users

```
(1164, 14)
(500, 14)
```

- Both data are split into a 70:30 format using the train test split

Scaled dataset for mobile users

Train data

	Yearly_avg_view_on_travel_page	total_likes_on_outstation_checkin_given	yearly_avg_Outstation_checkins	member_in_family	preferred_location_type	Yearly_avg_comment_on_travel_page
3304	0.092992	0.331764	0.142977	0.111111	0.951433	1.242493
4197	-0.562040	0.721858	0.826942	1.072705	-0.659688	0.250801
6598	-0.591152	1.675909	-0.882971	0.111111	0.629209	-0.180370
5248	0.646129	-0.758590	0.370965	0.111111	-0.337464	-0.827126
12293	-0.712736	-0.940614	1.852890	2.034298	1.918106	0.286315

Test data

week_since_last_outstation_checkin	following_company_page	montly_avg_comment_on_company_page	working_flag	travelling_network_rating	Adult_flag	Daily_Avg_mins_spend_on_traveling_page
-1.227306	1.283658	-0.135943	2.326844	-1.508427	-0.939815	-0.943089
-0.090147	-0.779024	-0.358482	-0.429767	0.314872	-0.939815	-0.394958
2.184169	-0.779024	-0.135943	-0.429767	-1.508427	1.064039	0.810931
0.667958	-0.779024	-0.358482	-0.429767	-1.508427	1.064039	1.359062
-0.469200	-0.779024	-0.318021	-0.429767	-1.508427	-0.939815	-0.285332

Img 3

Scaled dataset for Laptop users

Train data

	Yearly_avg_view_on_travel_page	total_likes_on_outstation_checkin_given	yearly_avg_Outstation_checkins	member_in_family	preferred_location_type	Yearly_avg_comment_on_travel_page
1579	-1.817537	0.271803	-0.454379	0.165109	-0.882465	-0.140236
536	-0.693083	-0.058046	1.664886	0.165109	2.094111	-1.489680
1605	-0.637183	-0.806822	0.600930	-0.820462	0.199926	-0.514652
194	-0.693083	-1.132889	0.487517	-1.806033	1.552915	0.733715
892	-0.331567	-0.561933	-0.572116	1.150679	-0.882465	0.387854

Test data

vel_page	total_likes_on_outofstation_checkin_received	week_since_last_outstation_checkin	following_company_page	montly_avg_comment_on_company_page	working_flag	travelling_network_rating	Adult_f
-0.684110	-0.003409	-0.761565	-0.750000	-1.035786	-0.393363	-1.536329	1.028
-1.753052	-0.885329	0.408873	-0.750000	-1.363152	-0.393363	0.301385	-0.972
-0.135549	-0.807483	0.018727	1.333333	0.601044	2.542181	-0.617472	-0.972
0.919304	0.812217	-0.761565	-0.750000	-0.381054	-0.393363	1.220242	1.028
0.194332	-0.808487	-1.151712	-0.750000	-0.872103	-0.393363	0.301385	-0.972

Img 4

- Scaling mobile and laptop dataset before modeling
- Using the standard scaler for scaling

Data Modeling: Logistic Regression

Logistic regression for Mobile users:

Adding constant to the data frame

	const	Yearly_avg_view_on_travel_page	total_likes_on_outstation_checkin_given	yearly_avg_Outstation_checkins	member_in_family	preferred_location_type	Yearly_avg_comment_on_travel_page
3304	1.0	0.092992	0.331764	0.142977	0.111111	0.951433	1.242493
4197	1.0	-0.562040	0.721858	0.826942	1.072705	-0.659688	0.250801
6598	1.0	-0.591152	1.675909	-0.882971	0.111111	0.629209	-0.180370
5248	1.0	0.646129	-0.758590	0.370965	0.111111	-0.337464	-0.827126
12293	1.0	-0.712736	-0.940614	1.852890	2.034298	1.918106	0.286315

	const	Yearly_avg_view_on_travel_page	total_likes_on_outstation_checkin_given	yearly_avg_Outstation_checkins	member_in_family	preferred_location_type	Yearly_avg_comment_on_travel_page
14840	1.0	-0.592641	-1.155616	-0.888626	-0.829415	0.952679	-1.364813
2484	1.0	-0.592641	-1.537364	-0.888626	2.060917	-0.330422	0.872708
3274	1.0	0.106900	-0.618103	-0.888626	1.097473	1.594230	0.349033
869	1.0	1.026173	1.626403	0.717542	-0.829415	-0.330422	-1.079172
16338	1.0	-0.771672	0.016751	1.743031	-1.792859	-0.651198	-1.308160

Img 6

- Constant is added to the train and test dataset for ensuring better capture relationship in data

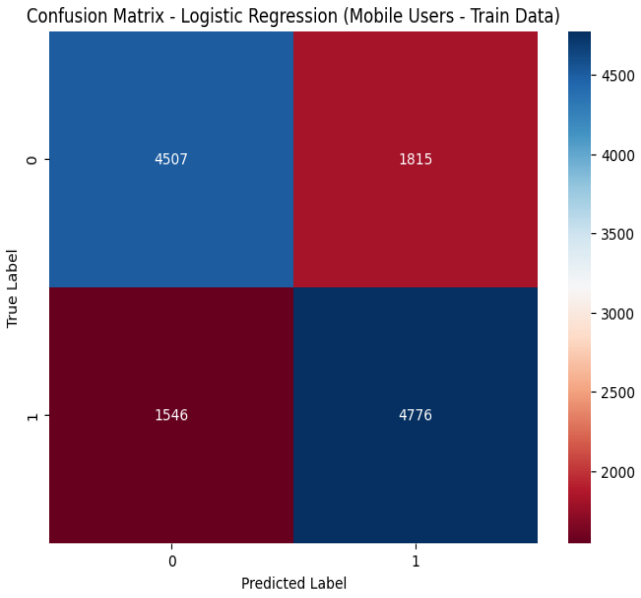
Mobile users model summary

```
Optimization terminated successfully.
         Current function value: 0.565149
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:          Taken_product   No. Observations:                12644
Model:                          Logit   Df Residuals:                    12629
Method:                           MLE   Df Model:                           14
Date:                Tue, 11 Mar 2025   Pseudo R-squ.:                 0.1847
Time:                        04:33:05   Log-Likelihood:                -7145.7
converged:                       True   LL-Null:                       -8764.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
=============================================================================================================
                                              coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------------------------
const                                      -0.0144      0.020     -0.705      0.481      -0.054       0.026
Yearly_avg_view_on_travel_page             -0.1720      0.027     -6.456      0.000      -0.224      -0.120
total_likes_on_outstation_checkin_given    -0.1393      0.021     -6.751      0.000      -0.180      -0.099
yearly_avg_Outstation_checkins              0.2483      0.021     11.981      0.000       0.208       0.289
member_in_family                           -0.0494      0.021     -2.367      0.018      -0.090      -0.008
preferred_location_type                    -0.0308      0.020     -1.505      0.132      -0.071       0.009
Yearly_avg_comment_on_travel_page          -0.0328      0.021     -1.558      0.119      -0.074       0.008
total_likes_on_outofstation_checkin_received -0.3334    0.032    -10.568      0.000      -0.395      -0.272
week_since_last_outstation_checkin          0.2965      0.022     13.530      0.000       0.254       0.339
following_company_page                      0.6591      0.021     31.327      0.000       0.618       0.700
montly_avg_comment_on_company_page         -0.0832      0.022     -3.715      0.000      -0.127      -0.039
working_flag                               -0.0442      0.022     -2.043      0.041      -0.087      -0.002
travelling_network_rating                  -0.2200      0.021    -10.701      0.000      -0.260      -0.180
Adult_flag                                 -0.6267      0.021    -30.237      0.000      -0.667      -0.586
Daily_Avg_mins_spend_on_traveling_page     -0.1550      0.037     -4.203      0.000      -0.227      -0.083
=============================================================================================================
```

Img 7

- The strongest positive predictors for buying tickets are following company page (0.659) and week since last outstation check in (0.296)
- Adult flag (-0.6267) and total likes on outstation check in received (-0.3334) has showing strong negative effect

Plotting the confusion matrix and report of training dataset



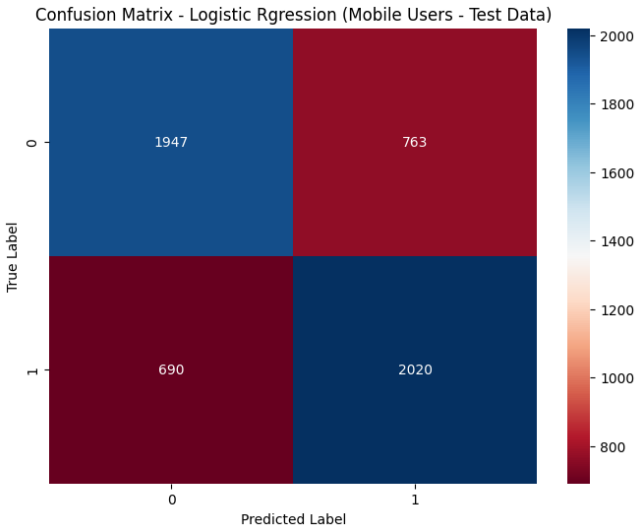
Classification Report (Train Data):

	precision	recall	f1-score	support
0	0.74	0.71	0.73	6322
1	0.72	0.76	0.74	6322
accuracy			0.73	12644
macro avg	0.73	0.73	0.73	12644
weighted avg	0.73	0.73	0.73	12644

Img 8

- This model achieves 73 % of accuracy and f1 score so, the performance is balanced in both classes
- 1815 cases are false positive and 1246 cases are false negative hence, both cases need to reduce

Plotting the confusion matrix and report of testing dataset

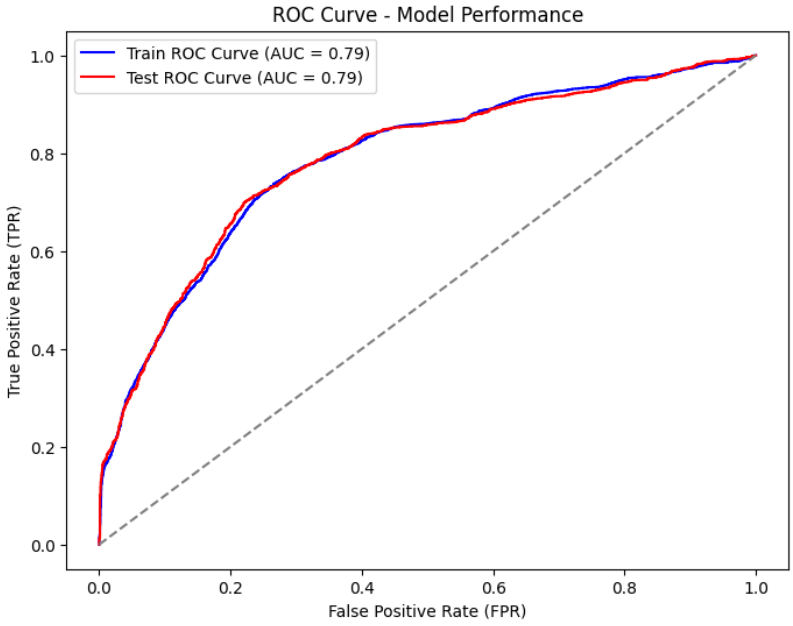


Classification Report (Test Data):				
	precision	recall	f1-score	support
0	0.74	0.72	0.73	2710
1	0.73	0.75	0.74	2710
accuracy			0.73	5420
macro avg	0.73	0.73	0.73	5420
weighted avg	0.73	0.73	0.73	5420

Img 9

- In test data, overall accuracy of 73% indicating the model perform well in both classes

ROC AUC Curve of Basic model performance



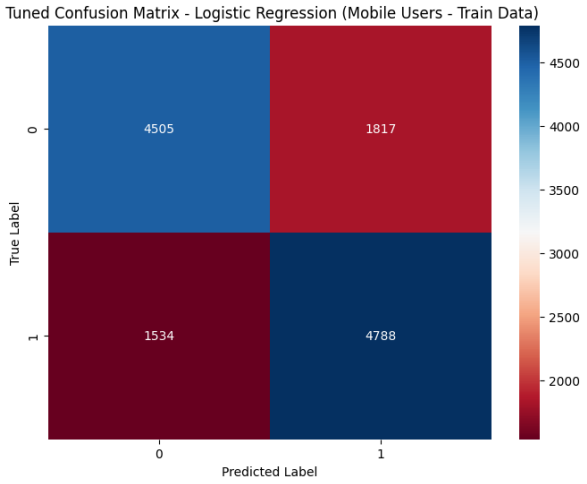
Img 10

- In both train and test dataset the AUC score is 0.79
- So, this model is good and ability to distinguish between buyers and non-buyers

Model performance improving using hyperparameters - Mobile users

- Using the GridSearchCV hyperparameter to increase model performance with the help of given algorithm

Plotting the tuned confusion matrix and report of training dataset



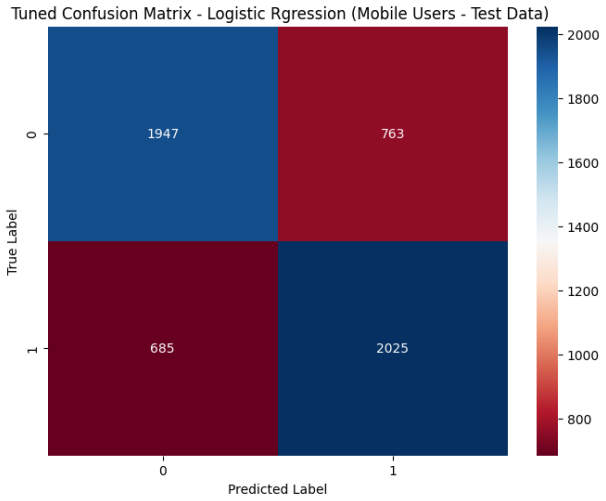
Tuned Classification Report (Train Data):

	precision	recall	f1-score	support
0	0.75	0.71	0.73	6322
1	0.72	0.76	0.74	6322
accuracy			0.73	12644
macro avg	0.74	0.73	0.73	12644
weighted avg	0.74	0.73	0.73	12644

Img 11

- Train model is still at 73% accuracy after tuning
- 76% recall for class 1 so, recall is high for positive class

Plotting the tuned confusion matrix and report of testing dataset



Tuned Classification Report (Test Data):

	precision	recall	f1-score	support
0	0.74	0.72	0.73	2710
1	0.73	0.75	0.74	2710
accuracy			0.73	5420
macro avg	0.73	0.73	0.73	5420
weighted avg	0.73	0.73	0.73	5420

Img 12

- The test model also maintains the same accuracy of 73%

Logistic regression for Laptop users:

Adding constant to the data frame

	const	Yearly_avg_view_on_travel_page	total_likes_on_outstation_checkin_given	yearly_avg_Outstation_checkins	member_in_family	preferred_location_type	Yearly_avg_comment_on_travel_page
1579	1.0	-1.817537	0.271803	-0.454379	0.165109	-0.882465	-0.140236
536	1.0	-0.693083	-0.058046	1.664886	0.165109	2.094111	-1.489680
1605	1.0	-0.637183	-0.806822	0.600930	-0.820462	0.199926	-0.514652
194	1.0	-0.693083	-1.132889	0.487517	-1.806033	1.552915	0.733715
892	1.0	-0.331567	-0.561933	-0.572116	1.150679	-0.882465	0.387854

Img 13

- Added constant to data for better capture relation between train and test dataset

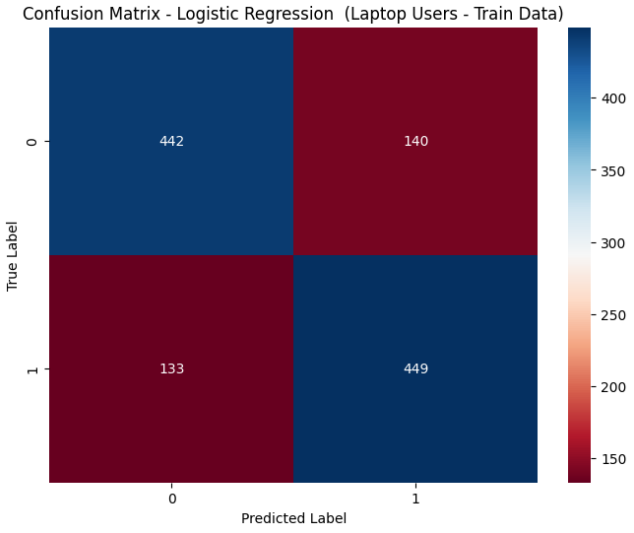
Laptop users model summary

```
Optimization terminated successfully.
         Current function value: 0.493691
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:          Taken_product   No. Observations:                 1164
Model:                          Logit   Df Residuals:                     1149
Method:                           MLE   Df Model:                           14
Date:                Tue, 11 Mar 2025   Pseudo R-squ.:                  0.2878
Time:                        04:33:06   Log-Likelihood:                -574.66
converged:                       True   LL-Null:                       -806.82
Covariance Type:            nonrobust   LLR p-value:                3.310e-90
==============================================================================================================
                                                coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------------------------
const                                         -0.0450      0.073     -0.612      0.540      -0.189       0.099
Yearly_avg_view_on_travel_page                -0.1478      0.100     -1.481      0.139      -0.343       0.048
total_likes_on_outstation_checkin_given       -0.4191      0.076     -5.519      0.000      -0.568      -0.270
yearly_avg_Outstation_checkins                 0.5112      0.079      6.479      0.000       0.357       0.666
member_in_family                              -0.0015      0.077     -0.020      0.984      -0.152       0.149
preferred_location_type                       -0.2147      0.074     -2.899      0.004      -0.360      -0.070
Yearly_avg_comment_on_travel_page              0.2595      0.073      3.531      0.000       0.115       0.404
total_likes_on_outofstation_checkin_received  -0.5230      0.115     -4.563      0.000      -0.748      -0.298
week_since_last_outstation_checkin             0.4747      0.082      5.792      0.000       0.314       0.635
following_company_page                         0.6605      0.078      8.480      0.000       0.508       0.813
montly_avg_comment_on_company_page             0.1244      0.086      1.450      0.147      -0.044       0.293
working_flag                                  -0.1641      0.086     -1.902      0.057      -0.333       0.005
travelling_network_rating                     -0.5114      0.079     -6.485      0.000      -0.666      -0.357
Adult_flag                                    -0.6910      0.076     -9.149      0.000      -0.839      -0.543
Daily_Avg_mins_spend_on_traveling_page        -0.6380      0.133     -4.797      0.000      -0.899      -0.377
==============================================================================================================
```

Img 14

- Users who follow the company page have a strong positive association with conversion
- Adult flag highly negatively impacted the conversion maybe, the younger users are high

Plotting the confusion matrix and report of training dataset

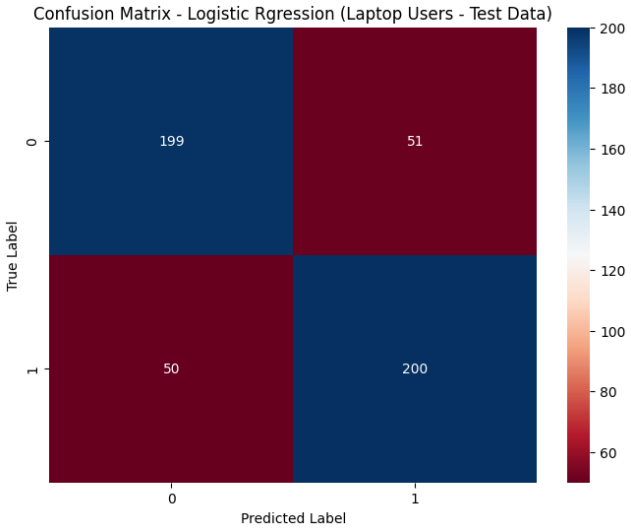



```
Classification Report (Train Data):
              precision    recall  f1-score   support

           0       0.77      0.76      0.76       582
           1       0.76      0.77      0.77       582

    accuracy                           0.77      1164
   macro avg       0.77      0.77      0.77      1164
weighted avg       0.77      0.77      0.77      1164
```

Img 15

- This model has balanced performance with equal precision, recall and f1 score of 77%
- High number of false positive (140) and false negative (133) make the model risky by targeting the wrong customers

Plotting the confusion matrix and report of testing dataset



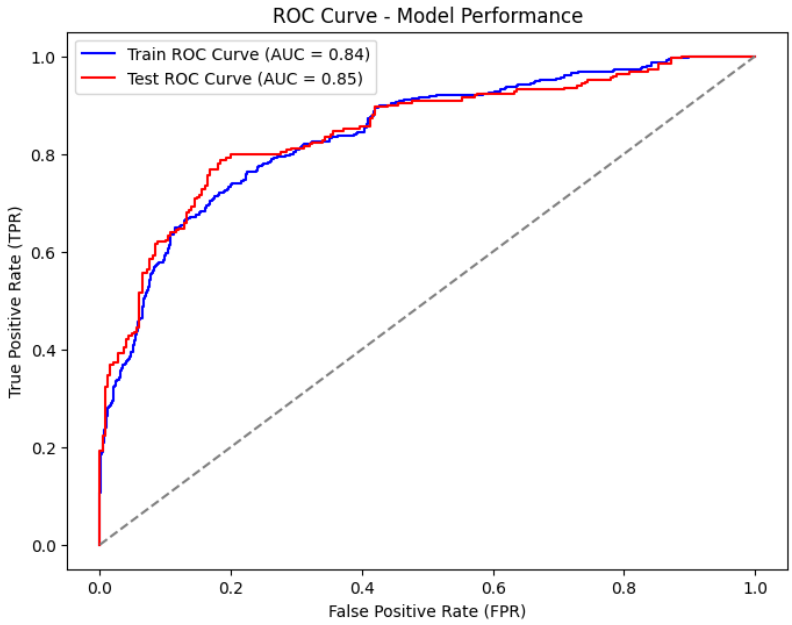
Classification Report (Test Data):	precision	recall	f1-score	support
0	0.80	0.80	0.80	250
1	0.80	0.80	0.80	250
accuracy			0.80	500
macro avg	0.80	0.80	0.80	500
weighted avg	0.80	0.80	0.80	500

Img 16

- Improved the accuracy to 80% in test performance comparing to training performance
- Balanced performance in precision, recall and f1 score
- Almost equal number of false positive (51) and false negative (50) decisions are taken and these are high with test data size

- Tuning could push the performance of the model
-

ROC AUC Curve of Basic model performance



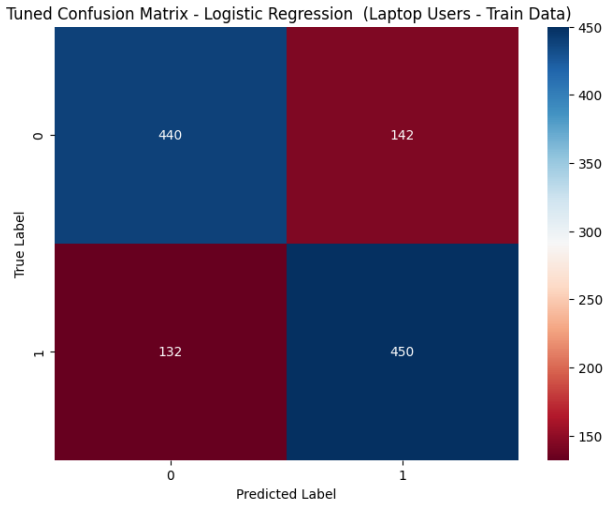
Img 17

- The AUC sore of train (0.84) and test (0.85) show that the model has strong performance
- There is a slight difference between the score due to the overfitting

Model performance improving using hyperparameters - Laptop users

- Using the GridSearchCV hyperparameter to increase model performance with the help of given algorithm

Plotting the tuned confusion matrix and report of training dataset

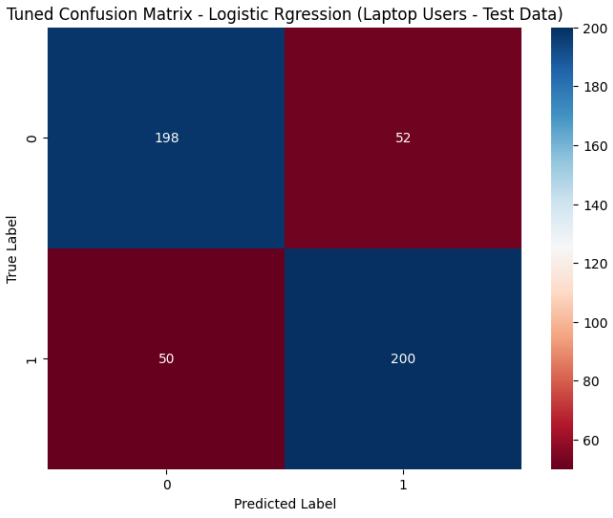



```
Tuned Classification Report (Train Data):
              precision    recall  f1-score   support

           0       0.77      0.76      0.76       582
           1       0.76      0.77      0.77       582

    accuracy                           0.76      1164
   macro avg       0.76      0.76      0.76      1164
weighted avg       0.76      0.76      0.76      1164
```

Img 18

- After doing the hyperparameter training the model performance has decreased when compare to logistic regression basic training performance

Plotting the tuned confusion matrix and report of testing dataset




```
Tuned Classification Report (Test Data):
              precision    recall  f1-score   support

           0       0.80      0.79      0.80       250
           1       0.79      0.80      0.80       250

    accuracy                           0.80       500
   macro avg       0.80      0.80      0.80       500
weighted avg       0.80      0.80      0.80       500
```

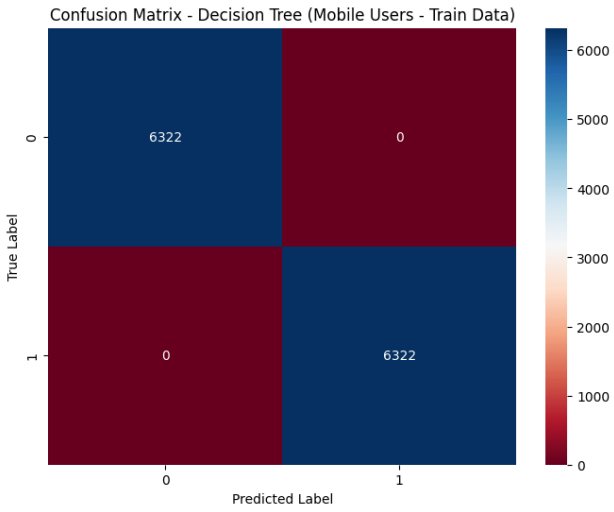
Img 19

- After tuning the model performance has reduced and still false predictions makes the model risky so, tuning doesn't increase the logistic regression model performance

Data Modeling: Decision Tree Classification

Decision tree classifier for mobile users

Plotting the confusion matrix and report of training dataset



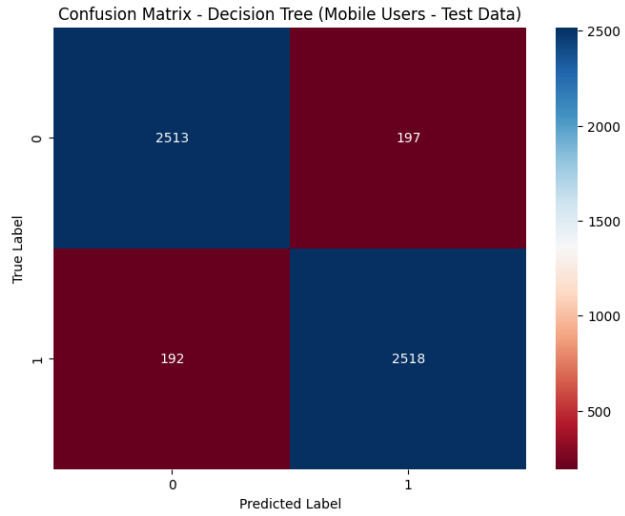
Classification Report (Train Data):

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6322
1	1.00	1.00	1.00	6322
accuracy			1.00	12644
macro avg	1.00	1.00	1.00	12644
weighted avg	1.00	1.00	1.00	12644

Img 20

- We can see that all scores are showing 100% so, the training model is overfitting

Plotting the confusion matrix and report of testing dataset



Classification Report (Test Data):

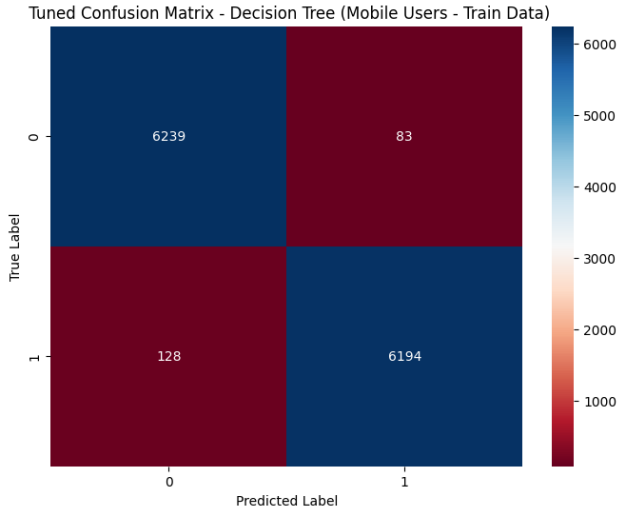
	precision	recall	f1-score	support
0	0.93	0.93	0.93	2710
1	0.93	0.93	0.93	2710
accuracy			0.93	5420
macro avg	0.93	0.93	0.93	5420
weighted avg	0.93	0.93	0.93	5420

Img 21

- In test performance 93% of accuracy for model prediction So, there is a chance of overfitting
- Need to improve generalization using hyperparameter tuning

Model performance improving using hyperparameters - Mobile users

Plotting the tuned confusion matrix and report of training dataset



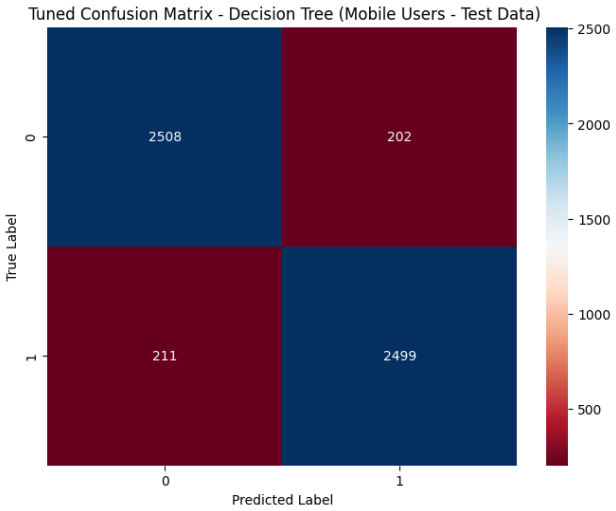
Tuned Classification Report (Train Data):

	precision	recall	f1-score	support
0	0.98	0.99	0.98	6322
1	0.99	0.98	0.98	6322
accuracy			0.98	12644
macro avg	0.98	0.98	0.98	12644
weighted avg	0.98	0.98	0.98	12644

Img 22

- Now the overfitting has reduced and the accuracy is 98% after tuning
- The average percentage of precision and recall is 98% so, the model is well balanced

Plotting the tuned confusion matrix and report of testing dataset

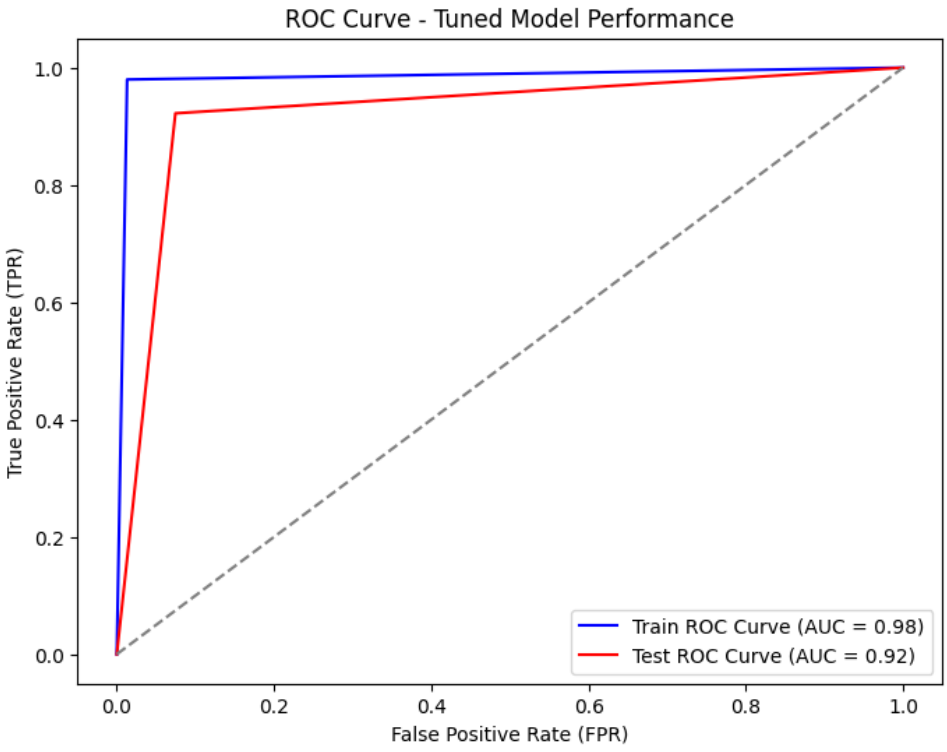



```
Tuned Classification Report (Test Data):
              precision    recall  f1-score   support

           0       0.92      0.93      0.92      2710
           1       0.93      0.92      0.92      2710

    accuracy                           0.92      5420
   macro avg       0.92      0.92      0.92      5420
weighted avg       0.92      0.92      0.92      5420
```

Img 23

- The test accuracy has remained stable
- The performance gap between train and test set has minimized so, the model has less prone to overfitting

ROC AUC Curve of tuned model performance



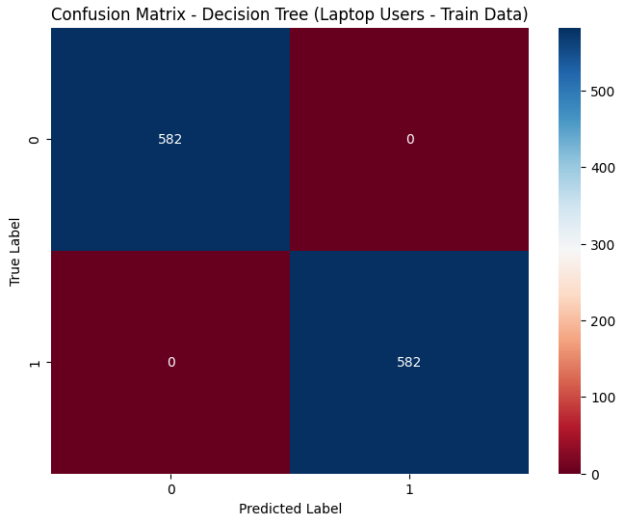
Img 24

- The AUC score of train 0.98 and test 0.92

- This is a best AUC score so; model has a strong predictive power after reducing overfitting

Decision tree classifier for Laptop users

Plotting the confusion matrix and report of training dataset



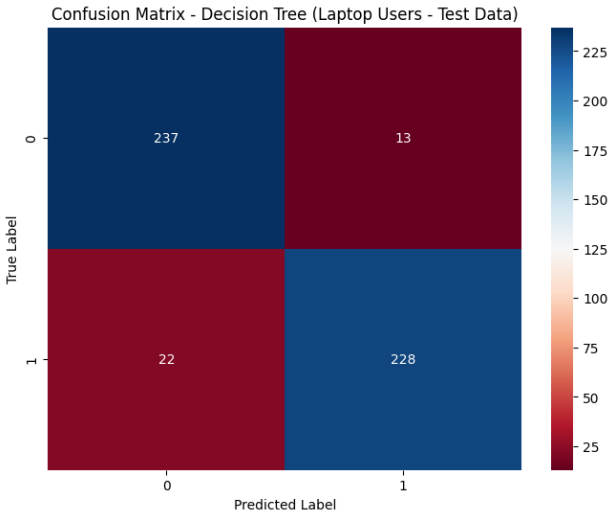
Classification Report (Train Data):

	precision	recall	f1-score	support
0	1.00	1.00	1.00	582
1	1.00	1.00	1.00	582
accuracy			1.00	1164
macro avg	1.00	1.00	1.00	1164
weighted avg	1.00	1.00	1.00	1164

Img 25

- This training model is overfitting hence, need to do hyperparameter tuning to improve generalization

Plotting the confusion matrix and report of testing dataset



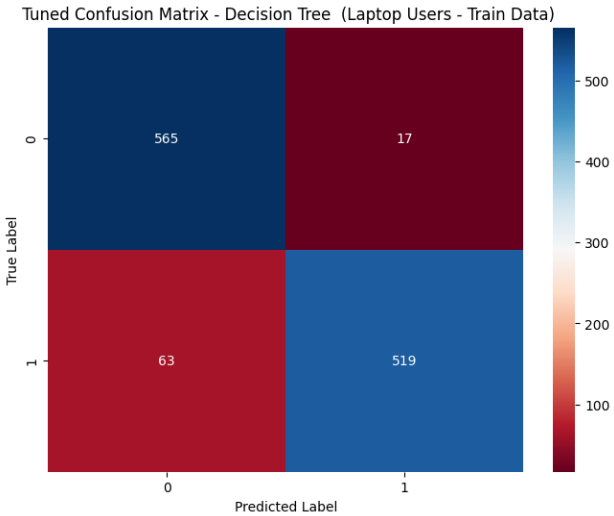
Classification Report (Test Data):	precision	recall	f1-score	support
0	0.92	0.95	0.93	250
1	0.95	0.91	0.93	250
accuracy			0.93	500
macro avg	0.93	0.93	0.93	500
weighted avg	0.93	0.93	0.93	500

Img 26

- The accuracy has dropped to 93% but, the model is still at complex
- Reduce the overfitting for better model performance

Model performance improving using hyperparameters - Laptop users

Plotting the tuned confusion matrix and report of training dataset

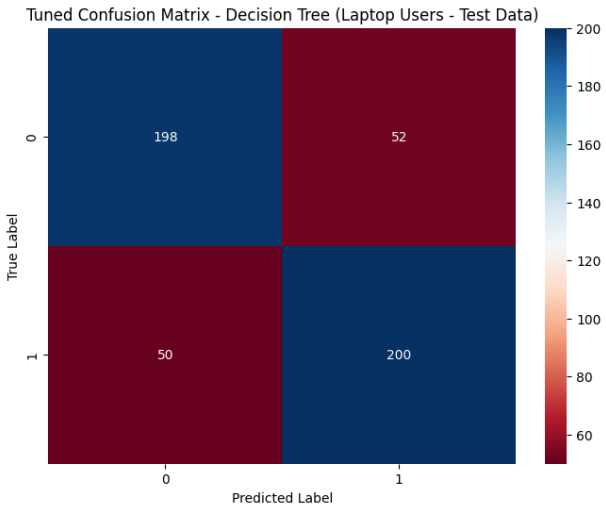


Tuned Classification Report (Train Data):	precision	recall	f1-score	support
0	0.90	0.97	0.93	582
1	0.97	0.89	0.93	582
accuracy			0.93	1164
macro avg	0.93	0.93	0.93	1164
weighted avg	0.93	0.93	0.93	1164

Img 27

- Now the overfitting has reduced and training accuracy is 93% which indicate better generalization
- The precision and recall are now balanced, with both classes having f1 score of 0.93%

Plotting the tuned confusion matrix and report of testing dataset



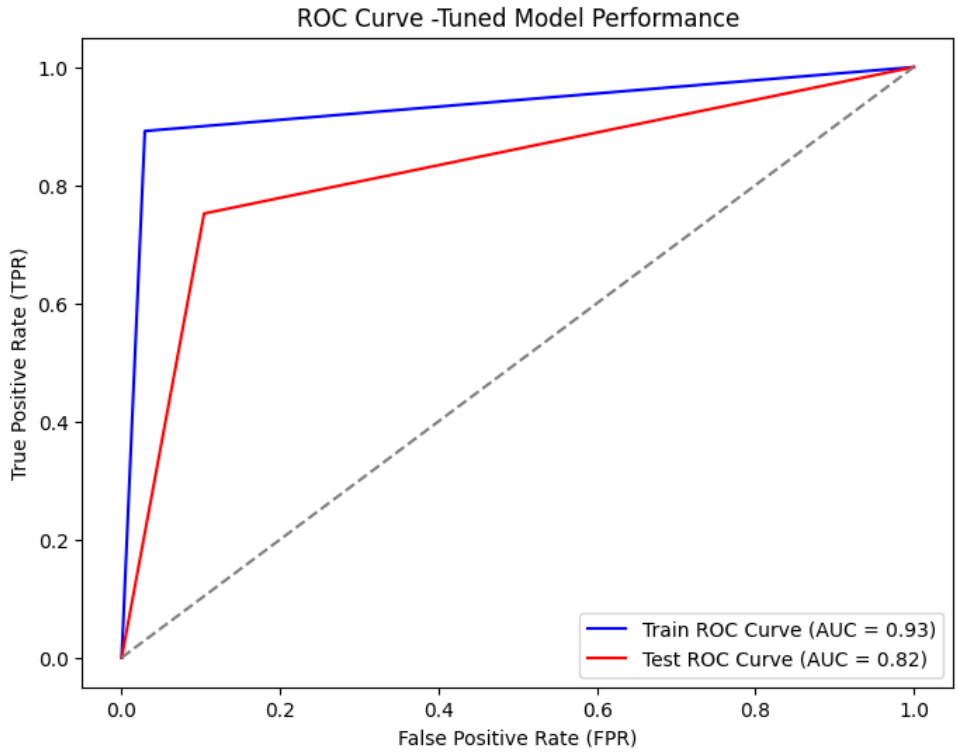
Tuned Classification Report (Test Data):

	precision	recall	f1-score	support
0	0.80	0.79	0.80	250
1	0.79	0.80	0.80	250
accuracy			0.80	500
macro avg	0.80	0.80	0.80	500
weighted avg	0.80	0.80	0.80	500

Img 28

- Tuned test accuracy has dropped to 80%, this is due to reducing overfitting
- The false positive and false negative rate also increased

ROC AUC Curve of tuned model performance



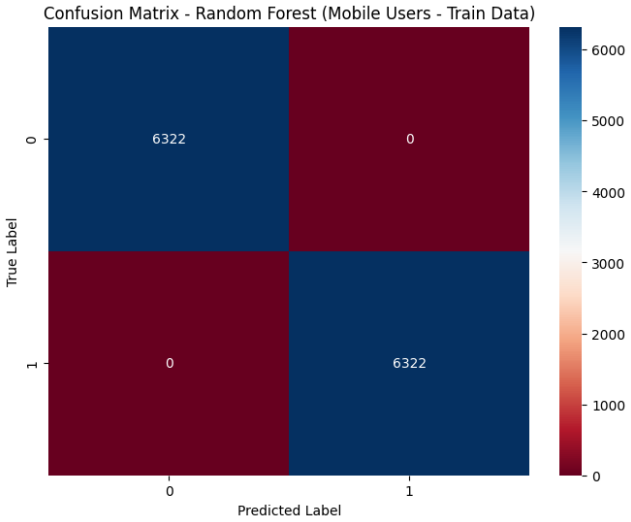
Img 29

- We can see that training performance is good at 0.93
- But test performance has reduced after tuning (0.82)
- Doing ensemble methods to handle complex pattern

Data Modeling: Random Forest Classification

Random Forest classifier for mobile users

Plotting the confusion matrix and report of training dataset

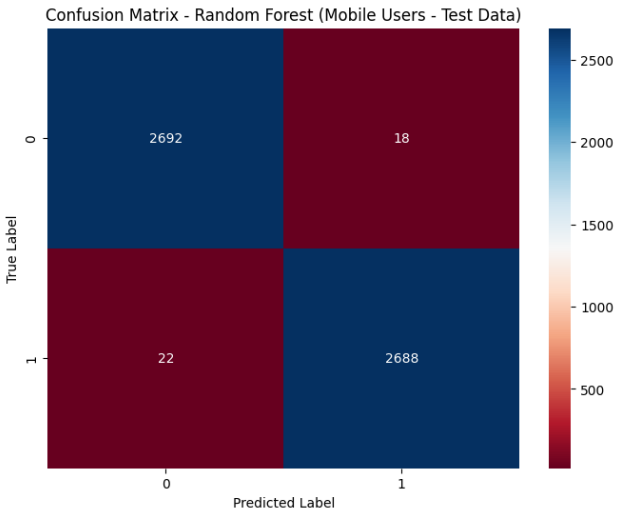


Classification Report (Train Data):				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	6322
1	1.00	1.00	1.00	6322
accuracy			1.00	12644
macro avg	1.00	1.00	1.00	12644
weighted avg	1.00	1.00	1.00	12644

Img 30

- In random forest modeling the train data is overfitting
- Set the parameters to reduce overfitting

Plotting the confusion matrix and report of testing dataset

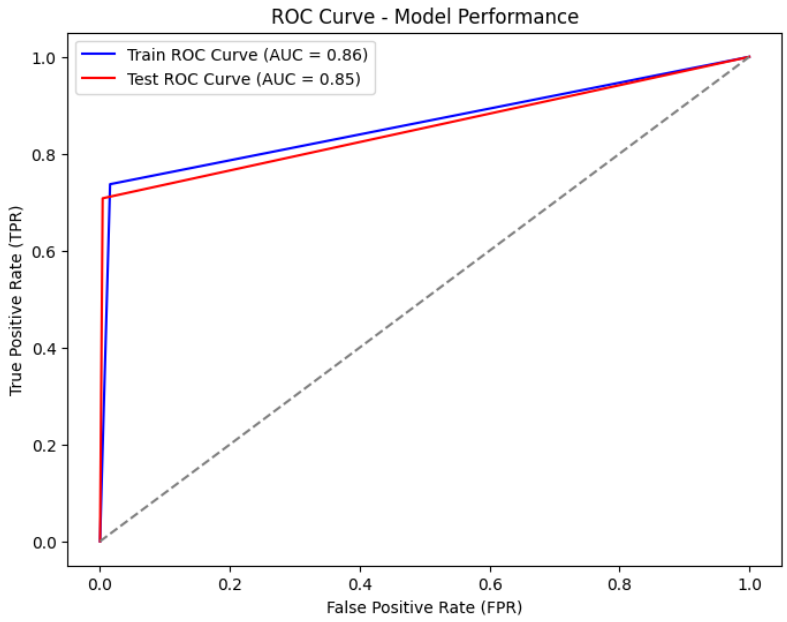


Classification Report (Test Data):				
	precision	recall	f1-score	support
0	0.99	0.99	0.99	2710
1	0.99	0.99	0.99	2710
accuracy			0.99	5420
macro avg	0.99	0.99	0.99	5420
weighted avg	0.99	0.99	0.99	5420

Img 31

- The accuracy score is still at high 0.99 so, there is a slight overfitting happened in both data
- But this overfitting is not a major issue because, the gap between train accuracy and test accuracy is 0.01%

ROC AUC Curve of tuned model performance

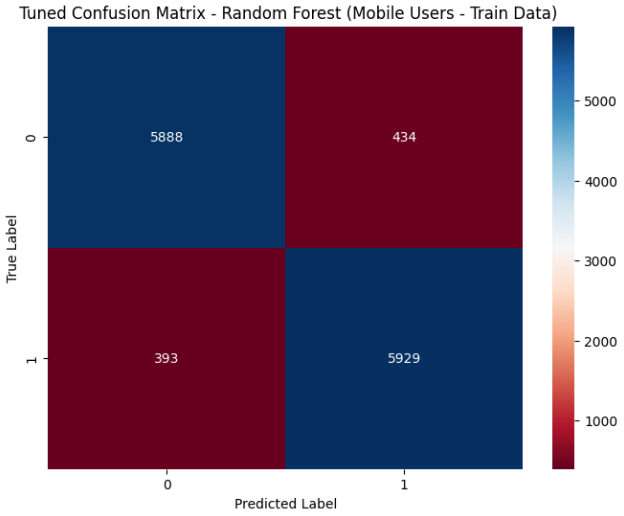


Img 32

- ROC shows model performance is good
- The AUC train score is 0.86 and test is 0.85 there is only 0.01% gap between the score so, no overfitting happened in the performance

Model performance improving using hyperparameters - Mobile users

Plotting the tuned confusion matrix and report of training dataset

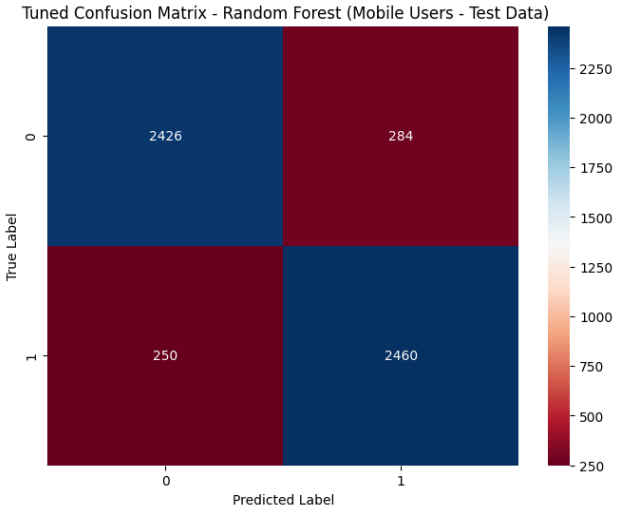


Classification Report (Train Data):	precision	recall	f1-score	support
0	0.94	0.93	0.93	6322
1	0.93	0.94	0.93	6322
accuracy			0.93	12644
macro avg	0.93	0.93	0.93	12644
weighted avg	0.93	0.93	0.93	12644

Img 33

- After tuning the overfitting has reduced to 93%
- Precision and recall are both 93% so, the model is well balanced

Plotting the tuned confusion matrix and report of testing dataset



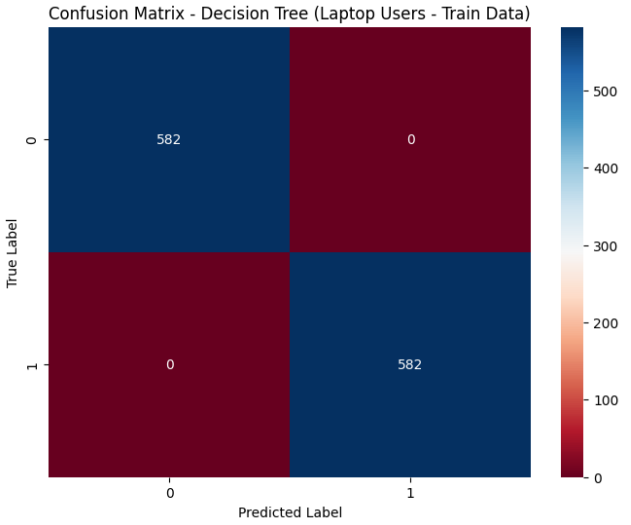

```
Classification Report (Test Data):
              precision    recall  f1-score   support

           0       0.91      0.90      0.90      2710
           1       0.90      0.91      0.90      2710

    accuracy                           0.90      5420
   macro avg       0.90      0.90      0.90      5420
weighted avg       0.90      0.90      0.90      5420
```

Img 34

- The test accuracy dropped from 99% to 90%
- false negative and false positive are minimized so, the recall and precision are balanced

Random Forest classifier for Laptop users

Plotting the confusion matrix and report of training dataset

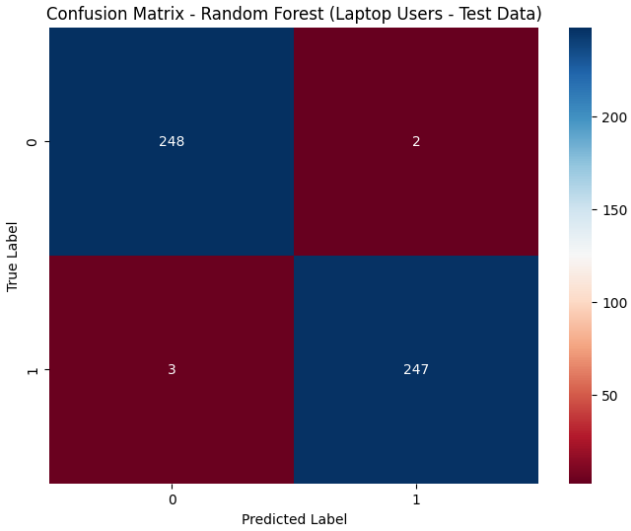



```
Classification Report (Train Data):
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       582
           1       1.00      1.00      1.00       582

    accuracy                           1.00      1164
   macro avg       1.00      1.00      1.00      1164
weighted avg       1.00      1.00      1.00      1164
```

Img 35

- Laptop train model is overfitting, just analyze the test performance also

Plotting the confusion matrix and report of testing dataset

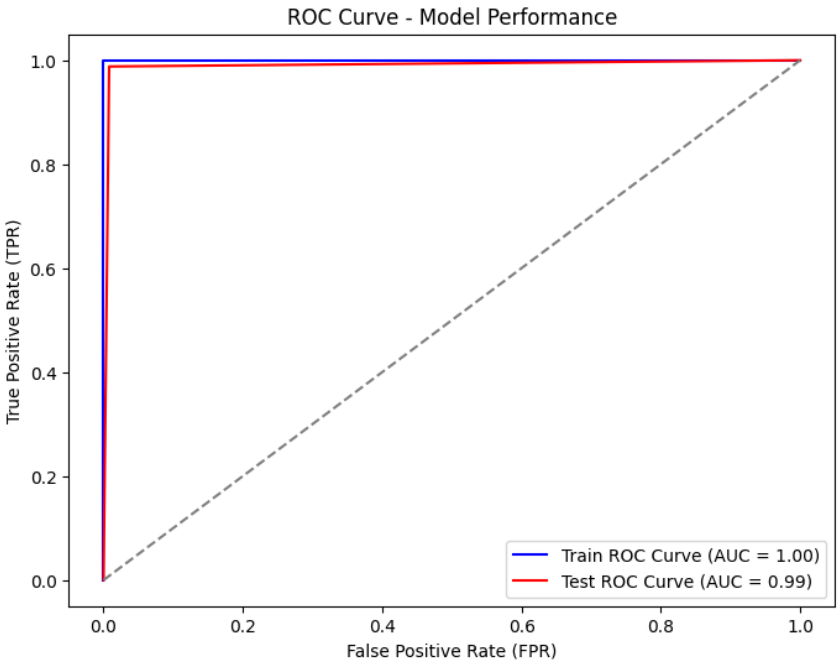



```
Classification Report (Test Data):
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       250
           1       0.99      0.99      0.99       250

    accuracy                           0.99       500
   macro avg       0.99      0.99      0.99       500
weighted avg       0.99      0.99      0.99       500
```

Img 36

- Test data is performing well, but there is a slight overfitting happened
- Recall and precision is high so the wrong prediction is less

ROC AUC Curve of tuned model performance

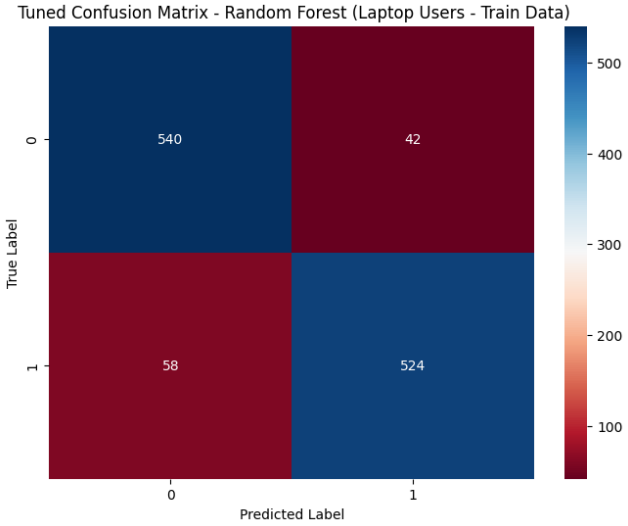


Img 37

- Train AUC score is 1.00 and test is 0.99 which means the model is performance is great
- The possibility of overfitting is high so, let's analyze the tuned performance of both dataset

Model performance improving using hyperparameters - Laptop users

Plotting the tuned confusion matrix and report of training dataset



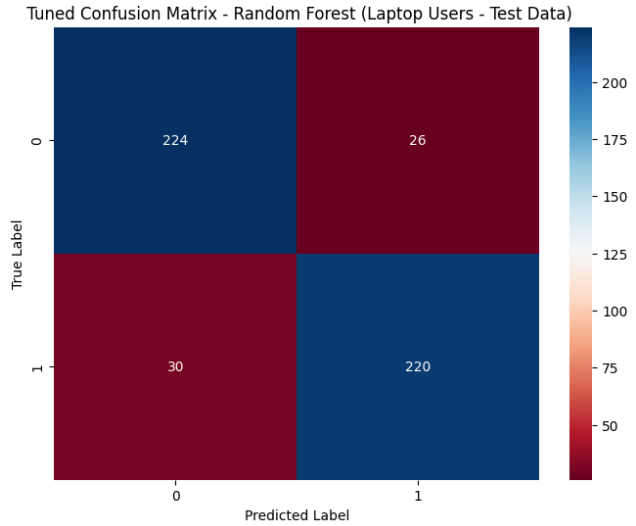
Classification Report (Train Data):

	precision	recall	f1-score	support
0	0.90	0.93	0.92	582
1	0.93	0.90	0.91	582
accuracy			0.91	1164
macro avg	0.91	0.91	0.91	1164
weighted avg	0.91	0.91	0.91	1164

Img 38

- Precision and recall are well balanced in both classes so, the tuned model performance is good
- The overfitting has reduced and now the accuracy is 91%

Plotting the tuned confusion matrix and report of test dataset



Classification Report (Test Data):

	precision	recall	f1-score	support
0	0.88	0.90	0.89	250
1	0.89	0.88	0.89	250
accuracy			0.89	500
macro avg	0.89	0.89	0.89	500
weighted avg	0.89	0.89	0.89	500

Img 39

- The test accuracy (89%) is slightly lower than train accuracy (91%)
- The performance has reduced so, the recall score is low comparing with train data

Model comparison and Final model selection

Comparing all the model build for mobile users

Training reports

Mobile training performance comparison:

	Logistic Regression	Tuned Logistic Regression	Decision Tree	Tuned Decision Tree	Random Forest	Tuned Random Forest
Accuracy	0.734182	0.690209	1.0	0.983312	1.0	0.934593
Recall	0.755457	0.786935	1.0	0.979753	1.0	0.937836
Precision	0.724624	0.659377	1.0	0.986777	1.0	0.931793
F1	0.739720	0.717531	1.0	0.983253	1.0	0.934805

Testing reports

Mobile test performance comparison:

	Logistic Regression	Tuned Logistic Regression	Decision Tree	Tuned Decision Tree	Random Forest	Tuned Random Forest
Accuracy	0.731919	0.692435	0.928229	0.923801	0.992620	0.901476
Recall	0.745387	0.787823	0.929151	0.922140	0.991882	0.907749
Precision	0.725835	0.661605	0.927440	0.925213	0.993348	0.896501
F1	0.735482	0.719218	0.928295	0.923674	0.992614	0.902090

Img 40

- When comparing with all models, Random Forest has the highest test accuracy (99.26%) for mobile users
- Logistic Regression and Tunes Logistic Regression have the least performance comparing with all models
- Decision Tree shows the sigh of overfitting due to 100% training performance and in test performance it drops to 92%
- Hence, based on overall performance Random Forest was selected as the final model prediction to understand mobile user behavior

Comparing all the model build for laptop users

Training reports

Laptop training performance comparison:

	Logistic Regression	Tuned Logistic Regression	Decision Tree	Tuned Decision Tree	Random Forest	Tuned Random Forest
Accuracy	0.765464	0.765464	1.0	0.931271	1.0	0.914089
Recall	0.771478	0.773196	1.0	0.891753	1.0	0.900344
Precision	0.762309	0.761421	1.0	0.968284	1.0	0.925795
F1	0.766866	0.767263	1.0	0.928444	1.0	0.912892

Testing reports

Laptop test performance comparison:

	Logistic Regression	Tuned Logistic Regression	Decision Tree	Tuned Decision Tree	Random Forest	Tuned Random Forest
Accuracy	0.798000	0.794000	0.930000	0.824000	0.990000	0.888000
Recall	0.800000	0.800000	0.912000	0.752000	0.988000	0.880000
Precision	0.796813	0.790514	0.946058	0.878505	0.991968	0.894309
F1	0.798403	0.795229	0.928717	0.810345	0.989980	0.887097

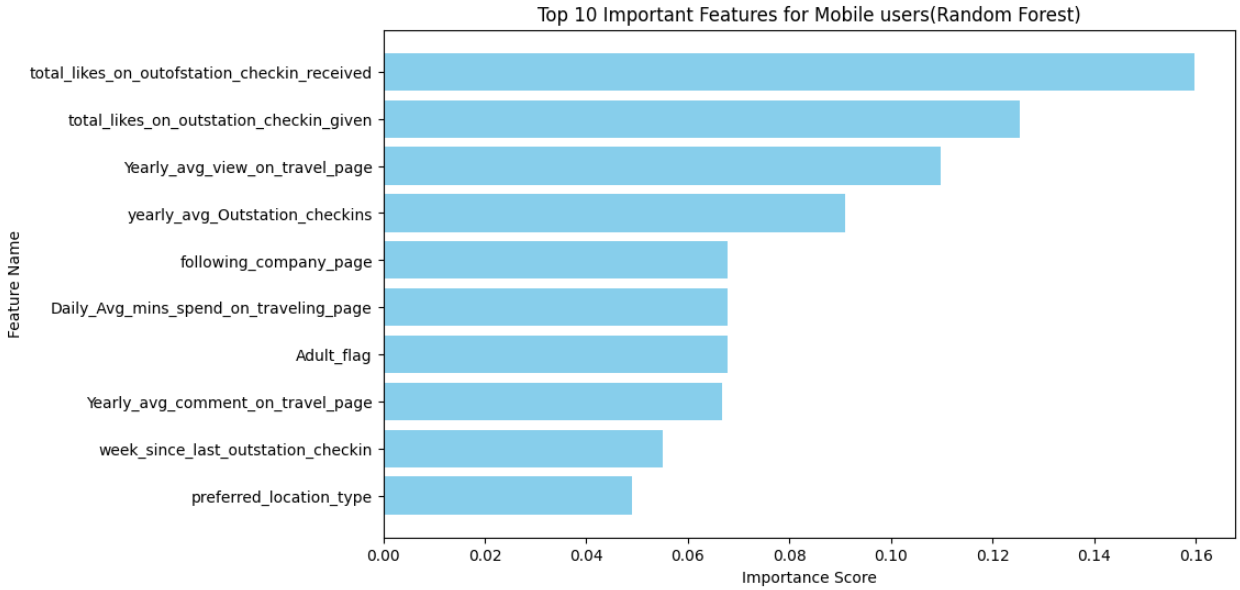
Img 41

- It can be se that the Random Forest achieved the highest test accuracy (99.0%) among all models

- Here also Decision Tree data has overfitted with train accuracy of 100% and test score dropped to 93% this reduced the stability of model performance
- The test accuracy, precision and recall score and overall stability shows that, Random Forest is the best model and final model for laptop users

Feature Importance from Random Forest

The most important 10 features for mobile users

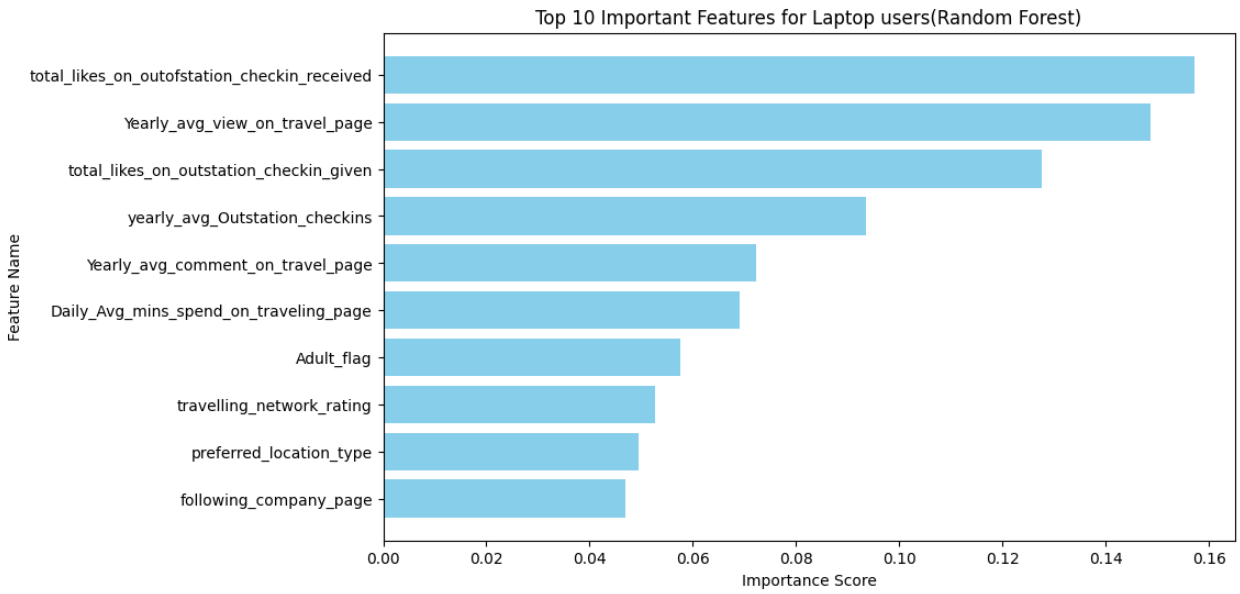


Img 42

- The most important 10 factors influencing prediction for mobile users in the Random Forest model are shown above

- Total likes on outstation checking received is that first most important feature this indicate, users mostly engagements with outstation check-in
- Total and likes on outstation chick in given and yearly avg view on travel page are the second and third most important features hence, users interact with others check in and frequently search for travel related queries

The most important 10 features for laptop users



Img 43

- The most important 10 factors influencing prediction for laptop users in the Random Forest model are shown above
- Total likes on outstation checking received is that first most important feature in both mobile and laptop users so, users always engaged with outstation check in
- Yearly avg view on travel page and total likes on outstation check in given are second and third position also play significant role in model prediction

---