



APP SUBSCRIPTION ANALYSIS

TEAM MEMBERS:

A. ROHAN GANESH
SREECHARAN KOMMULA
M. SAI PAVAN ADITYA



PROBLEM STATEMENT

Build a classifier using Machine Learning models which predicts whether the user of an app of a fintech company enrolls for it's subscription or not based on the data collected the by the company.

The details of data collection by the company and its motives are shown next.



Details of the Company and application

The application is from a Fintech company used for financial purposes like bank loans, savings, etc. in one place. The firm want to grow their business. But the problem arises while deciding who to recommend this app and offer who really wants it.

So the company gives a free trial of their premium version to customers for 24 hours and collects data from them. The goal of the firm is to build a Machine learning model which predicts whether the newly enrolled customer is interested to buy their product or not. In this scenario, some customers buy the app and some do not. According to the data available, They want to give special offer to the customer who are not interested to buy without the offer and grow their business.



MOTIVATION AND BENEFITS


This problem of classification presents a unique challenge of representing text data consisting of a list of screens visited as meaningful numerical data required for feeding into a machine learning model. This was our motivation for choosing this project.

This project is beneficial for companies as it helps them gain an insight into what are the features in their application which makes users enroll for a premium subscription thus allowing them to focus more on providing more such features.

DATASET

We have used a dataset consisting of 50,000 data entries where each entry consists of 11 features and 1 label. The following is a part of the dataset containing 20 data points.

user	first_open	dayofweek	hour	age	screen_list	numscreens	minigame	used_premium_f	enrolled	enrolled_date	liked
235136	14:51.3	3	2:00:00	23	idscreen,joinscreen,Cycle,product_review	15	0	0	0		0
333588	16:00.9	6	1:00:00	24	joinscreen,product_review,product_review	13	0	0	0		0
254414	19:09.2	1	19:00:00	23	Splash,Cycle,Loan	3	0	1	0		1
234192	08:46.4	4	16:00:00	28	product_review,Home,product_review,Lo	40	0	0	1	11:49.5	0
51549	50:48.7	1	18:00:00	31	idscreen,joinscreen,Cycle,Credit3Contair	32	0	0	1	56:37.8	1
56480	58:15.8	2	9:00:00	20	idscreen,Cycle,Home,ScanPreview,Verifi	14	0	0	1	59:03.3	0
144649	33:18.5	1	2:00:00	35	product_review,product_review2,ScanPr	3	0	0	0		0
249366	07:49.9	1	3:00:00	26	Splash,Cycle,Home,Credit3Container,Cn	41	0	1	0		0
372004	22:01.6	2	14:00:00	29	product_review,product_review2,ScanPr	33	1	1	1	24:54.5	0
338013	22:16.0	4	18:00:00	26	Home,Loan2,product_review,product_rev	19	0	0	1	31:58.9	0
43555	48:27.6	1	4:00:00	39	Splash,idscreen,Home,RewardsContain	14	0	0	1	02:17.2	0
317454	07:07.4	1	11:00:00	32	product_review,Home,Loan2,Credit3Con	25	1	1	0		0
205375	28:45.9	0	6:00:00	25	idscreen,joinscreen,Cycle,product_review	11	0	0	0		0
307608	52:31.8	5	19:00:00	23	Alerts,ProfilePage,Home,Credit3Contain	4	0	0	1	27:42.8	0
359855	48:48.9	0	4:00:00	17	joinscreen,product_review,product_review	9	0	0	0		0
284938	41:35.7	5	18:00:00	25	idscreen,joinscreen,Cycle,Loan2,product	26	1	0	1	10:04.5	0
235143	07:35.1	6	16:00:00	21	product_review,product_review,product_j	6	0	0	1	24:09.1	0
141402	12:46.9	5	21:00:00	55	joinscreen,Cycle,product_review,Loan2,p	20	0	0	1	35:03.1	0
257945	59:43.4	4	5:00:00	32	Splash,product_review,Home,Loan2,pro	15	0	0	1	29:36.9	1

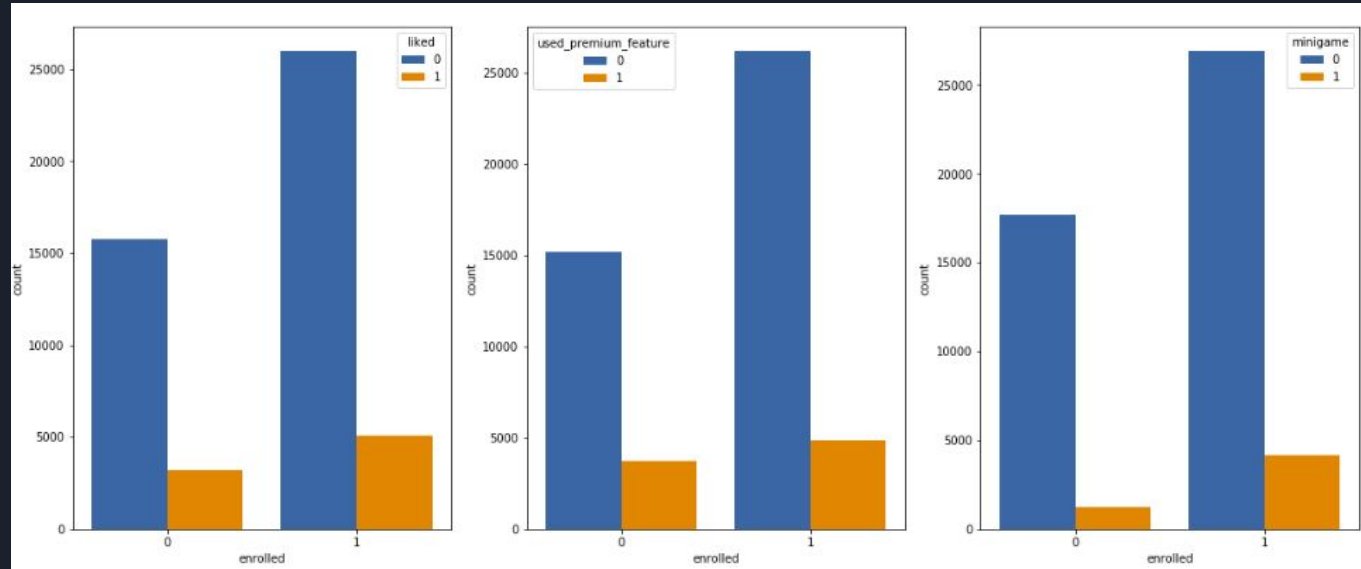
- 
- The Dataset contains categorical data, continuous data as well as sequence data which are processed accordingly. The following is the dataset description:

- | # | Column | Non-Null | Count | Dtype |
|----|----------------------|----------|----------|--------|
| 0 | user | 50000 | non-null | int64 |
| 1 | first_open | 50000 | non-null | object |
| 2 | dayofweek | 50000 | non-null | int64 |
| 3 | hour | 50000 | non-null | object |
| 4 | age | 50000 | non-null | int64 |
| 5 | screen_list | 50000 | non-null | object |
| 6 | numscreens | 50000 | non-null | int64 |
| 7 | minigame | 50000 | non-null | int64 |
| 8 | used_premium_feature | 50000 | non-null | int64 |
| 9 | enrolled | 50000 | non-null | int64 |
| 10 | enrolled_date | 31074 | non-null | object |
| 11 | liked | 50000 | non-null | int64 |

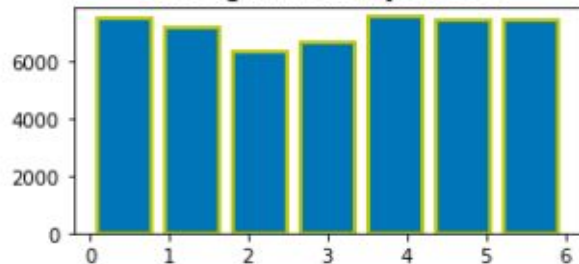
- The data is split into Train, Validation and Test Datasets with their sizes as follows:
 - Train Dataset : 40000
 - Test Dataset : 5000
 - Validation Dataset : 5000
- The dependent variables are enrolled (and enrolled date but it is ignored for the current task) and the rest are independent variables.

RESULTS OF EDA

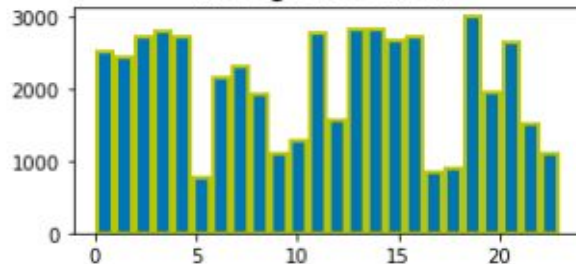
The following are the distributions of various features of the data with respect to whether the user has enrolled or not



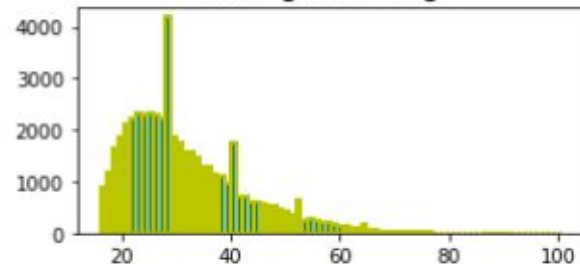
Histogram of dayofweek



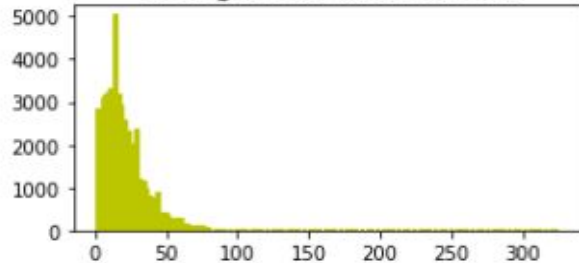
Histogram of hour



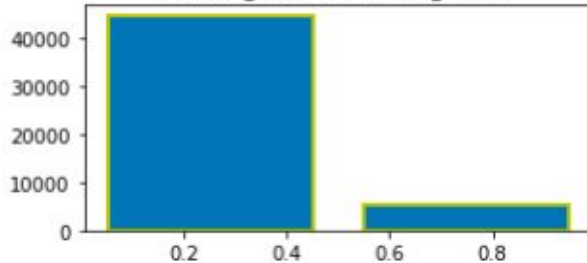
Histogram of age



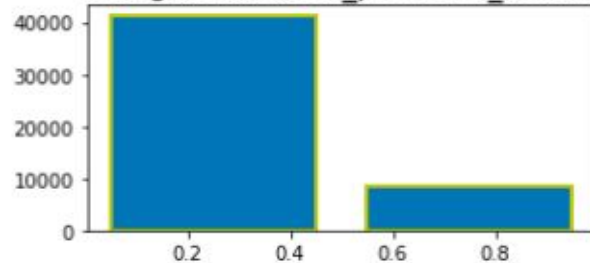
Histogram of numscreens



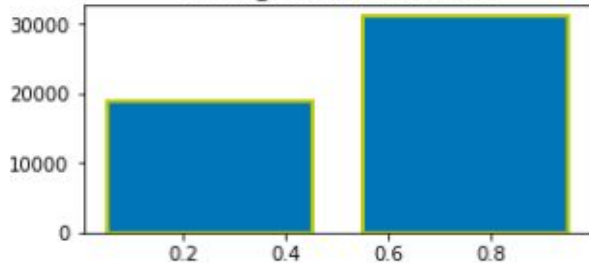
Histogram of minigame



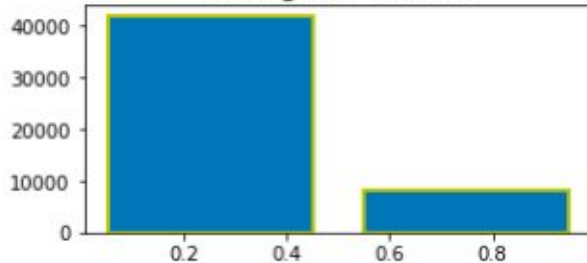
Histogram of used_premium_feature

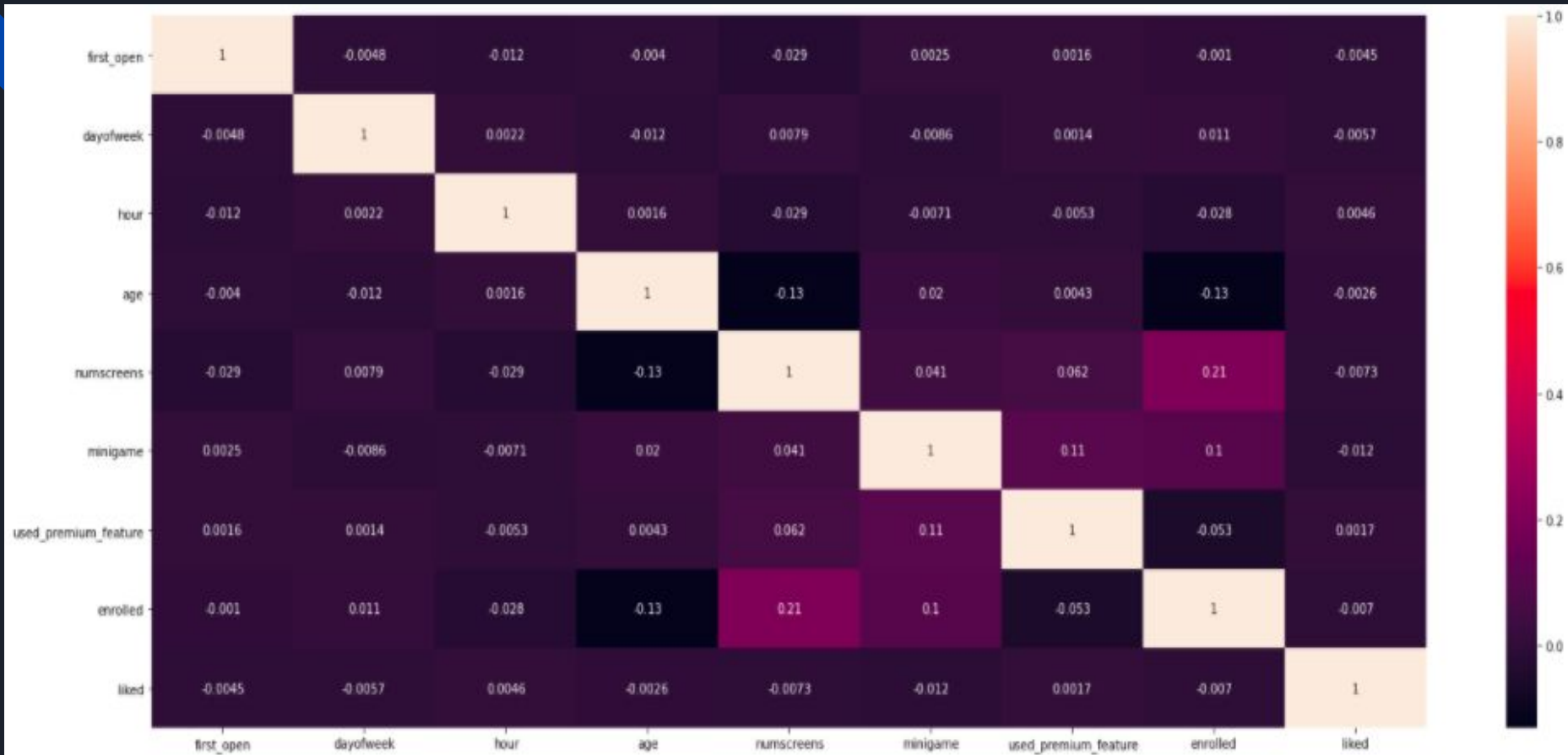


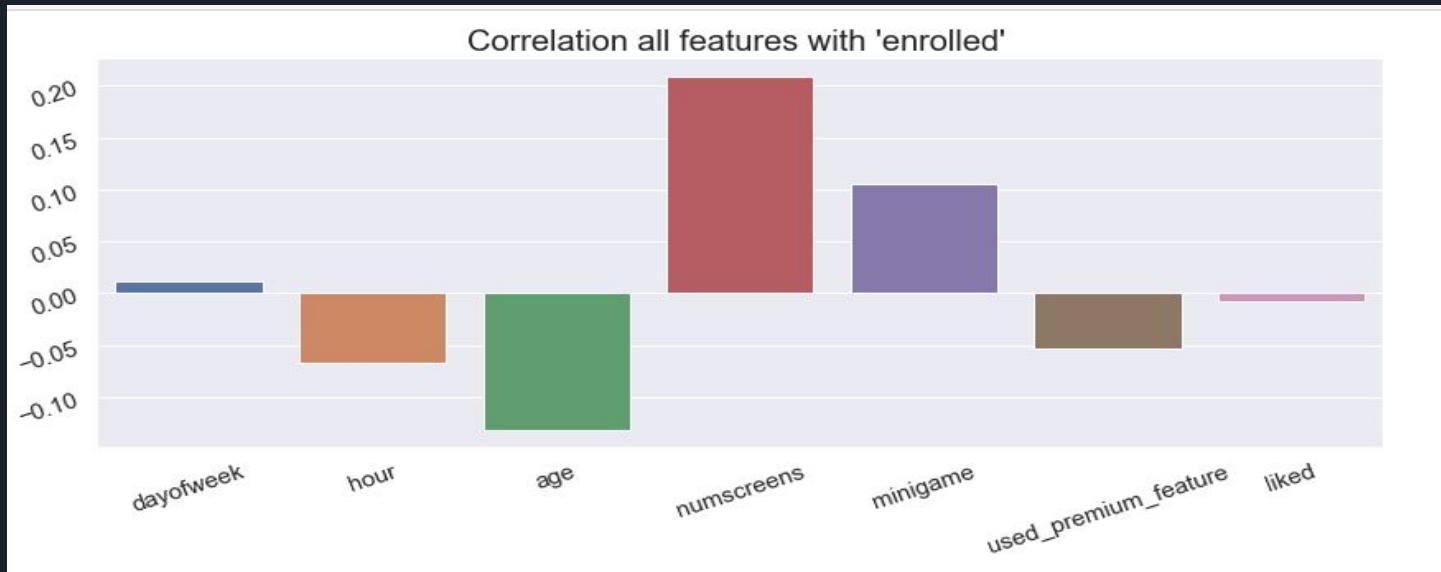
Histogram of enrolled



Histogram of liked









DATA PRE-PROCESSING

- The enrolled data column is dropped as our task is to predict whether a user enrolls or not and not predicting the enrolled date.
- The first_open column is converted from date time format to a timestamp which is a floating point with a precision of 3 decimals
- The hour column is dropped as it is part of the first_open column.



PROCESSING SCREEN LIST

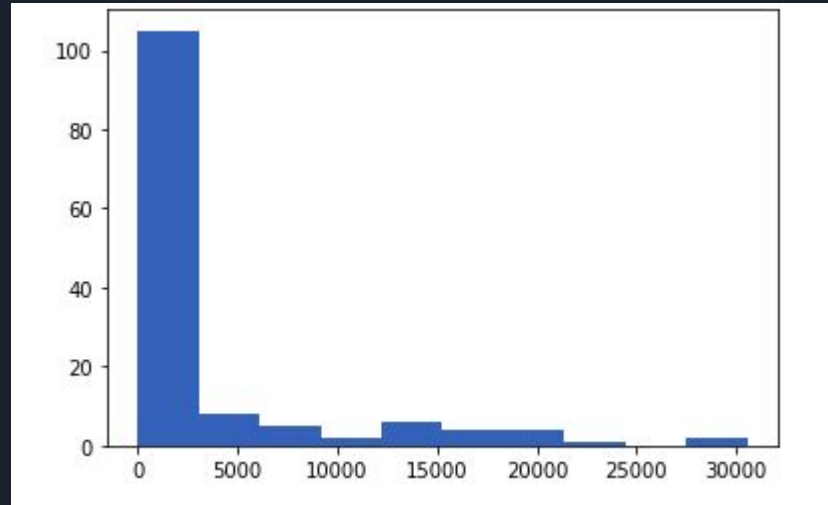
- Firstly, we create a vocabulary of all the pages present in the app (This is done on the total dataset). The size of this vocabulary comes out to be 137.
- Then, for each word in each data point, we find the tf-idf weight which is calculated as follows:

$$w_{i,j} = tf_{i,j} \times \log \left(\frac{N}{df_i} \right)$$

Here w denotes the weight, tf denotes the frequency of that word in the particular sentence and the log term (called idf) ensures that pages visited very frequently by everyone (eg. Home) are not given importance. (df is the number of sentences in which the word has occurred).

- The calculation of the df for words is done only on the Training Data.

DOCUMENT FREQUENCY OF WORDS



This histogram shows the document frequency of words (frequency across sentences and not counting repetitions within the sentence)

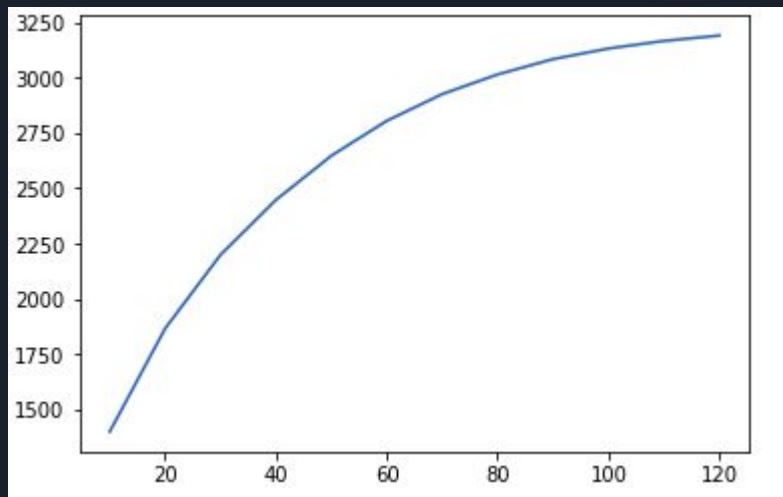


CONTINUED...

- Then, we assign each word in the vocabulary a one hot vector whose dimension is the size of the vocabulary.
- Using these vectors and the calculated weights, we compute a vector for each screen list as the weighted sum of these vectors with the weights being the tf-idf.
- We optionally perform PCA (described in the next slide) on these vectors to reduce the dimensionality.
- The obtained vectors are concatenated with the dataset and remove the screen list column and then the features are scaled using a standard scaler. The resultant dataset contains 48 dimensional vectors (if PCA is applied) and 145 dimensional vectors if not applied.

VIABILITY OF PCA

Following is the sum of eigenvalues(variances) if PCA is applied for various target dimensions.



Looking at the curve, we see sufficient saturation is not achieved at a lower dimension making PCA not viable.

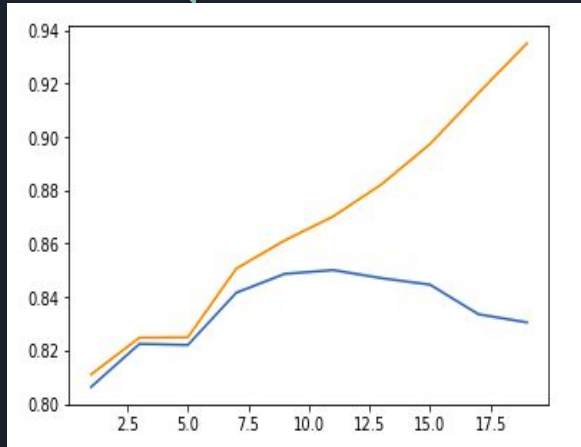


TRAINING, VALIDATION AND TESTING

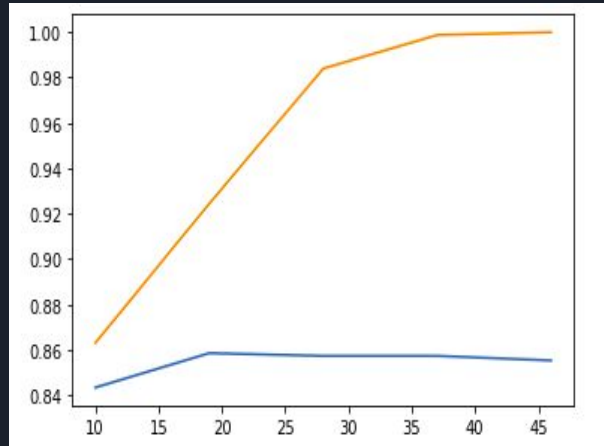
We train four Machine Learning models on the training dataset . The parameters of the models and train and validation accuracies are as follows:

Model Name	Train Accuracy	Validation Accuracy	Parameters
Random Forest	0.998	0.849	Depth : 10
Decision Tree	0.942	0.862	N_estimators : 100, Depth : 19
Logistic Regression	0.852	0.842	C : 0.5
XgBoost	0.99	0.856	Eta : 0.3 Max_Depth : 8 Steps : 15

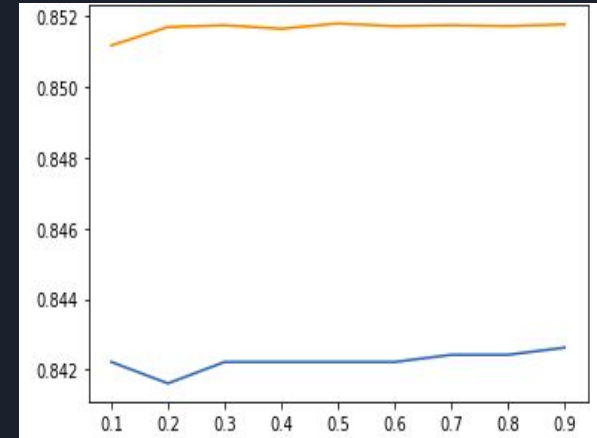
PLOTS OF PARAMETER TUNING ON VALIDATION DATA



Decision Tree (acc vs depth)



Random Forest (acc vs depth)



Logistic Regression (acc vs c
(regularisation parameter))



RESULTS OF GOODNESS OF FIT MEASURES ON TEST DATA

	model	accuracy	precision	recall	f1score
0	DecisionTree	0.8536	0.919609	0.839770	0.877878
1	RandomForest	0.8640	0.942620	0.833706	0.884824
2	Logistic_Regression	0.8468	0.920725	0.826684	0.871174
3	Xgboost	0.8676	0.943308	0.839132	0.888176

The following are the results of goodness of fit measures



INFERENCES

- It is seen that among all models, Random Forest and Xgboost perform the best for all evaluation metrics.
- Also, for our current task as the company focuses on business expansion and on the non-enrolled users, the recall would be the metric to test our models on as a higher recall would mean lesser false negatives which would result in lesser special offers being given by the company.



THANK YOU