Improving Starbuck's Offers' Success Rate: A Study to Analyze Purchasing Decisions

A Abhyuday Singh 22 hours ago · 21 min read

> *An analysis of data by Starbucks to deduce how users ' hidden traits influence the success rate of different offers.*

Image Source: Cloudinary

INTRODUCTION

The users are provided with offers every day- especially from brands they use or those whose membership they signed up for. These offers are of different types- a regular discount offer, buy X get Y, referral offers, etc. Additionally, their mode of delivery varies- it could be via emails, social media, TV advertisements, OTT ads, pop-up ads on some websites, etc. Some users love getting offers and actually fulfill those to get the rewards. Some are only irritated by these and never completely participate in these offers. A business needs to analyze these trends for better customer acquisition and retention.

In collaboration with the Udacity Data Science Nanodegree program, Starbucks has provided data for the various offers rolled out to the users of the Starbucks rewards mobile app over time through various channels. The data contains information about offers, users, and the events such as offers received, viewed, or completed.

Our job is to analyze these three data sets and figure out the patterns that would improve the offer success rates, i.e., figuring out which type of users would be better for which types of offers. In other words, customer segmentation- here, this means grouping customers according to the type of offers.

I decided to take it a step further and tried to fit the data into a classification model that would predict if a particular type of offer with its given attributes would be successful for a given user.

PROBLEM STATEMENT

In this project, we try to look at the data sets and answer the following questions:

1. What is the distribution of the offers that were rolled out? In this, we look at the data and try to figure out how the offers were distributed in terms of different offers and types of offers.

2. How many offers were viewed and completed? In this, we try to figure out how many offers were successful, i.e., were both viewed and completed.

3. What is the completion rate of each offer and each type of offer? Here, we would explore the completion rate of different types of offers to analyze which offers are more likely to be successful.

4. What is the success rate of offers? The completion rate and success rate differ in that an offer is only considered to be successful if it was viewed and completed. The offers which were completed without being viewed were wasted since the customer anyways spent the money on the product.

5. What does the demographic of the users look like? Here we will explore what type of users are present.

6. What is the correlation between attributes of users and offers and offer success rate? Finally, we will look at how does user demographics influence the offer success rate along with the offer attributes that might affect the offer success rate.

MODEL EVALUATION:

Our end goal here is to fit the data into a classification model so as to predict whether an offer would be suitable for a user.

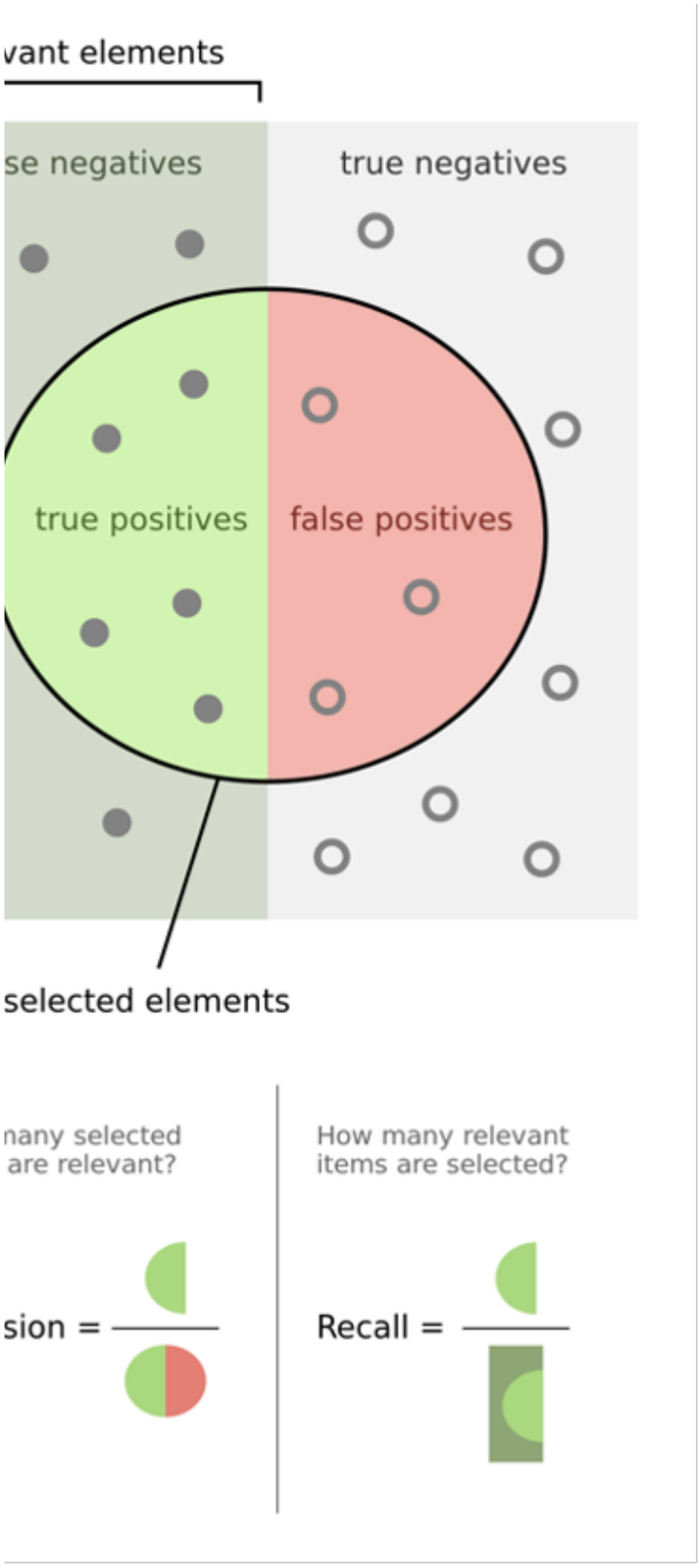


Image Source: Wikipedia

In other words, we want to predict if a user would complete the presented offer. This is a problem for binary classification- 0 if the offer would not be successful, 1 if it will be. In this, we try to fit a Logistic Regressor and a Random Forest Classifier and evaluate both models.

To evaluate classifiers, we use the metrics of precision, recall, F-1 score, and accuracy.

Precision: Precision is the ratio of the values that were correctly predicted as positive to the total number of values that were predicted as positive. That is, the ratio of true positives to the total predicted positives. This metric describes how precise the model is. In our model, to avoid wasting offers, we want it to have high precision because we want to maximize the success rate of the offer and minimize the failure rate as this helps in increasing revenues.

Recall: Recall is the ratio of the true positives out of all the positives, i.e., how many positive values were recalled. The recall is more important when there is a high cost associated with false

negatives. In our case, a false negative would mean a customer that would have completed the offer but was not presented with the offer because our model predicted they would not complete it.

F-1 Score: F1 Score is a function of precision and recall. It is another measure of the accuracy of the model. It is the harmonic mean of precision and recall. It applies additional weights to one of the two.

Accuracy: Percentage of the total items that were classified correctly.

AUC — ROC Curve: This measures the separability. AUC is the area under the curve and ROC is the probability curve. Higher the AUC, better the model as better the separability. In other words, the model will be able to separate 0s and 1s with more accuracy. This is plotted with True Positive Rate (TPR) against False Positive Rate (FPR).

These are the metrics that would be used to measure the predictive performance of our models.

DATA DESCRIPTION AND PREPROCESSING:

The data simulates customer behavior. There are three data sets: portfolio, profile, and transcript. Here we would explore each of them and clean them for further analysis and data modeling.

1. profile.json:

Rewards program users (17000 users x 5 fields)

- gender: (categorical) M, F, O, or null
- age: (numeric) missing value encoded as 118
- id: (string/hash)
- became_member_on: (date) format YYYYMMDD
- income: (numeric)

Starting with the profile dataset, our objectives for this dataset are:

1. Check for null values and drop, if any. Maintain the dropped user ids, as we would be dropping the corresponding records in the transcript dataset.

2. Map the hashed value of user id to a sequence value starting from 1. Maintain this map using a dictionary, as this will be a universal mapping of user id for all data sets.

3. Format the member_since attribute using pd.datetime.

4. Separate the day, month, and year from the above-formatted attribute.

5. Create the dummies of the gender attribute.

The head of the original dataset:

	gender	age	id	became_member_on	income
0	None	118	68be06ca386d4c31939f3a4f0e3dd783	20170212	NaN
1	F	55	0610b486422d4921ae7d2bf64640c50b	20170715	112000.0
2	None	118	38fe809add3b4fcf9315a9694bb96ff5	20180712	NaN
3	F	75	78afa995795e4d85b5d9ceeca43f5fef	20170509	100000.0
4	None	118	a03223e636434f42ac4c3df47e8bac43	20170804	NaN

head of the original dataset

```
profile.isna().sum()
```

```
gender              2175
age                    0
id                     0
became_member_on       0
income              2175
```

null values in the dataset

After dropping null values, the shape of the dataset becomes: **(14825, 5)**

After performing all the steps, the final cleaned dataset looks like this:

	age	income	user_id	gender_F	gender_M	gender_O	signup_day	signup_month	signup_year
1	55	112000.0	1	1	0	0	15	7	2017
3	75	100000.0	2	1	0	0	9	5	2017
5	68	70000.0	3	0	1	0	26	4	2018
8	65	53000.0	4	0	1	0	9	2	2018
12	58	51000.0	5	0	1	0	11	11	2017

final cleaned profile dataset

2. **portfolio.json**

Offers sent during the 30-day test period (10 offers x 6 fields)

- reward: (numeric) money awarded for the amount spent
- channels: (list) web, email, mobile, social
- difficulty: (numeric) money required to be spent to receive

reward

- duration: (numeric) time for offer to be open, in days
- offer_type: (string) bogo, discount, informational
- id: (string/hash)

There are three types of offers:

(i) Informational offer, which is merely an advertisement for a drink

(ii) Discount offer, where the user has to spend some amount to get a smaller amount back as rewards

(iii) BOGO, or buy one get one offer, where the user buys one drink and gets another one for free.

The original dataset looks like this:

	reward	channels	difficulty	duration	offer_type	id
0	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd
1	10	[web, email, mobile, social]	10	5	bogo	4d5c57ea9a6940dd891ad53e9dbe8da0
2	0	[web, email, mobile]	0	4	informational	3f207df678b143eea3cee63160fa8bed
3	5	[web, email, mobile]	5	7	bogo	9b98b8c7a33c4b65b9aebfe6a799e6d9
4	5	[web, email]	20	10	discount	0b1e1539f2cc45b7b9fa7c272da2e1d7
5	3	[web, email, mobile, social]	7	7	discount	2298d6c36e964ae4a3e7e9706d1fb8c2
6	2	[web, email, mobile, social]	10	10	discount	fafdcd668e3743c1bb461111dcafc2a4
7	0	[email, mobile, social]	0	3	informational	5a8bc65990b245e5a138643cd4eb9837
8	5	[web, email, mobile, social]	5	5	bogo	f19421c1d4aa40978ebb69ca19b0e20d
9	2	[web, email, mobile]	10	7	discount	2906b810c7d4411798c6938adc9daaa5

original portfolio dataset

Now, with the portfolio dataset, our objectives are:

1. Check for null values in the dataset.

There are no null values present.

2. Divide the channels list into individual columns of 1 or 0. For this, we need to first get the unique channels.

There are 4 channels: {‘mobile’, ‘email’, ‘web’, ‘social’}

3. Map the offer id (the id field) from hashed value to a sequence value starting from 1. Maintain this map using a dictionary, as this will be a universal mapping of offer id for all data sets.

4. Get the dummy variables of offer_type. There are 3 types of offers as described above.

5. Convert the duration from days to hours, as the timeline of offers in the transcript dataset is given in hours.

After cleaning, the dataset looks like this:

	reward	difficulty	duration	mobile	email	web	social	offer_id	offer_type_bogo	offer_type_discount	offer_type_informational
0	10	10	168	1	1	0	1	1	1	0	0
1	10	10	120	1	1	1	1	2	1	0	0
2	0	0	96	1	1	1	0	3	0	0	1
3	5	5	168	1	1	1	0	4	1	0	0
4	5	20	240	0	1	1	0	5	0	1	0
5	3	7	168	1	1	1	1	6	0	1	0
6	2	10	240	1	1	1	1	7	0	1	0
7	0	0	72	1	1	0	1	8	0	0	1
8	5	5	120	1	1	1	1	9	1	0	0
9	2	10	168	1	1	1	0	10	0	1	0

cleaned portfolio dataset

3. transcript.json

Event log (306648 events x 4 fields)

- person: (string/hash)

- event: (string) offer received, offer viewed, transaction, offer completed
- value: (dictionary) different values depending on event type
- offer id: (string/hash) not associated with any “transaction”
- amount: (numeric) money spent in “transaction”
- reward: (numeric) money gained from “offer completed”
- time: (numeric) hours after start of test

Before processing, the dataset looks like this:

	person	event	value	time
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	{'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}	0
1	a03223e636434f42ac4c3df47e8bac43	offer received	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}	0
2	e2127556f4f64592b11af22de27a7932	offer received	{'offer id': '2906b810c7d4411798c6938adc9daaa5'}	0
3	8ec6ce2a7e7949b1bf142def7d0e0586	offer received	{'offer id': 'fafdcd668e3743c1bb461111dcafc2a4'}	0
4	68617ca6246f4fbc85e91a2a49552598	offer received	{'offer id': '4d5c57ea9a6940dd891ad53e9dbe8da0'}	0

original transcript dataset

The transcript dataset would be cleaned in the following ways:

1. Drop the records for the dropped user ids.

This reduces the number of rows from 306534 to 272762.

2. Using the map for offer id, put the corresponding value of offer id into the offer id field of the value attribute.

3. Using the map for user-id, put the corresponding user if into the person attribute.

4. Extract offer id from value attribute and put it into a separate column.

5. Drop the duplicate records.

There are 374 duplicate records present.

Post cleaning, the dataset looks like this:

	event	time	user_id	transaction	offer_id
0	offer received	0	2	{'offer id': 4}	4
2	offer received	0	3	{'offer id': 10}	10
5	offer received	0	4	{'offer id': 9}	9
7	offer received	0	5	{'offer id': 3}	3
8	offer received	0	6	{'offer id': 5}	5

cleaned transcript dataset

EXPLORATORY DATA ANALYSIS

After cleaning the data, now we will explore the data using visualizations, and try to answer some of the questions using the data. Here, we will further shape the data in a way that it can be used for classification.

There are 3 types of offers in the portfolio dataset: BOGO, Discount, and Informational. These 3 can be categorized into 2 types: Reward offers (BOGO and Discount) and Informational Offers.

For informational offers, there is not ‘offer complete’ event. So, we would need to see if there was any transaction in the period the

offer was supposed to be influential.

In addition to that, we would explore the data to answer the following questions:

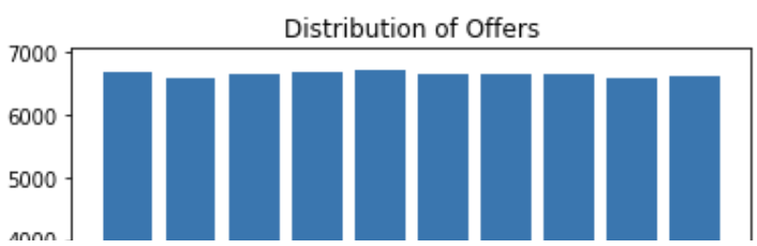
> 1. What is the distribution of the offers that were rolled out?
>
> 2. How many offers were viewed and completed?
>
> 3. What is the completion rate of each offer and each type of offer?
>
> 4. What is the success rate of offers?
>
> 5. What does the demographic of the users look like?
>
> 6. What is the correlation between attributes of users and offers and offer success rate?

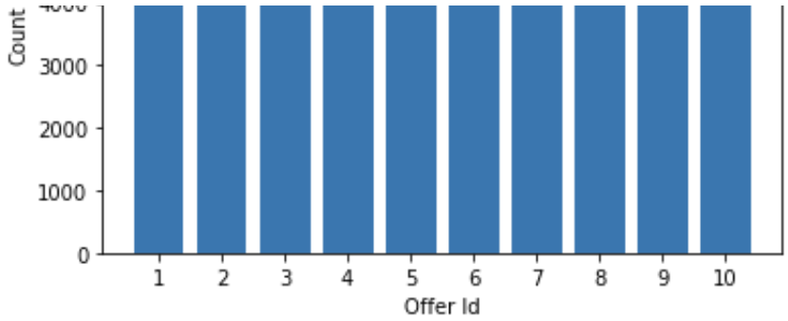
1. What is the distribution of the offers that were rolled out?

There are 4 types of events: offer received, offer viewed, offer completed, and transaction event.

In this, we look at the data and try to figure out how the offers were distributed in terms of different offers and types of offers.

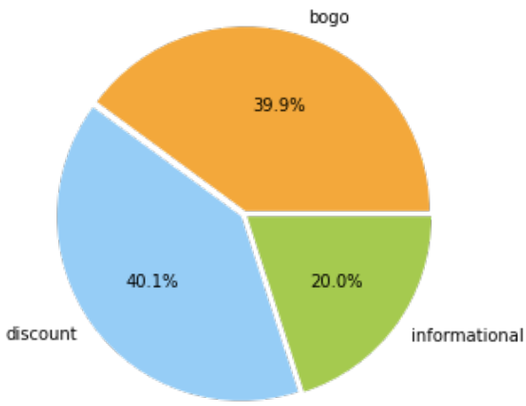
We group the offers according to the offer ids. The distribution is as follows:





distribution of offers

As we can see, almost the same number of offers were rolled out for every offer in the portfolio.



	offer_type	count
0	bogo	26537
1	discount	26664
2	informational	13300

(L) Distribution according to offer type. (R) Offer type counts

The number of BOGO and discount offers were similar — 39.9% and 40.1% respectively. The least number of offers were rolled out for informational offers. Therefore, 80% of the offers that were sent out were rewards offers and only 20% of the offers were informational offers.

2. How many offers were viewed and completed?

Apart from the ‘offer received’ event, there is events ‘offer viewed’ and ‘offer completed’ in the transcripts dataset for rewards offers. For informational offers, only an‘ offer viewed’ event is present. So, the success of the offers will be deduced using different techniques.

For reward offers, we would look into the sorted dataset for an offer received, viewed, and completed datasets and mark them as viewed and completed accordingly.

The reward offers statistics are:

Total Reward Offers: 53201

Viewed: 40500

Completed: 32070

Separating the Informational Offers. For these types of offers, we would look at 'offer received', 'offer viewed', and 'transaction' records. The informational offers have offer id of 3 or 8 and transactional events have offer id of -1.

Since there is no 'offer complete' event for these offers, their completion would be judged based on the transactions done during the period the offer had an influence over the customer. We assume that if there is any transaction present in this influential period, then the offer is completed.

We match the offers 'offer received' with 'offer viewed' and transaction to fill in the 'viewed' and 'completed' values. For 'completed' values, we will check whether the transaction event is before the expiry of the offer.

Informational Offers Statistics:

Total Information Offers: 13300

Viewed: 9360

Completed: 0

Final statistics:

Viewed and Completed: 4932

Viewed but not completed: 2591

Not viewed but completed: 3847

Not viewed and not completed: 1930

Now that we have completed and viewing values for both types of offers, we will concatenate them in one data frame in order to perform statistics on them.

The concatenated dataset looks like this:

	user_id	offer_id	viewed	completed
0	1	4	0	1
2	2	1	1	1
5	2	4	1	1
8	2	9	0	1
11	3	4	1	1
14	3	7	1	1
17	3	10	1	0
19	4	4	1	0
21	4	4	1	1
24	4	7	1	1

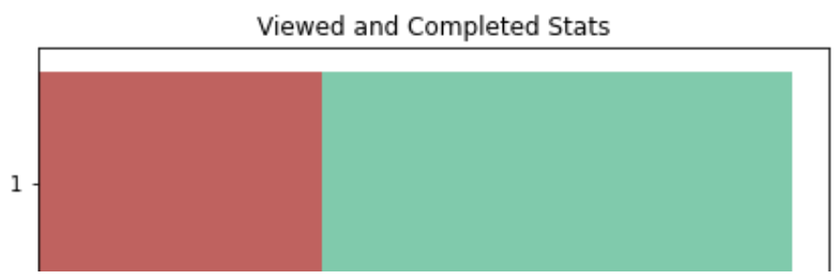
concatenated dataset

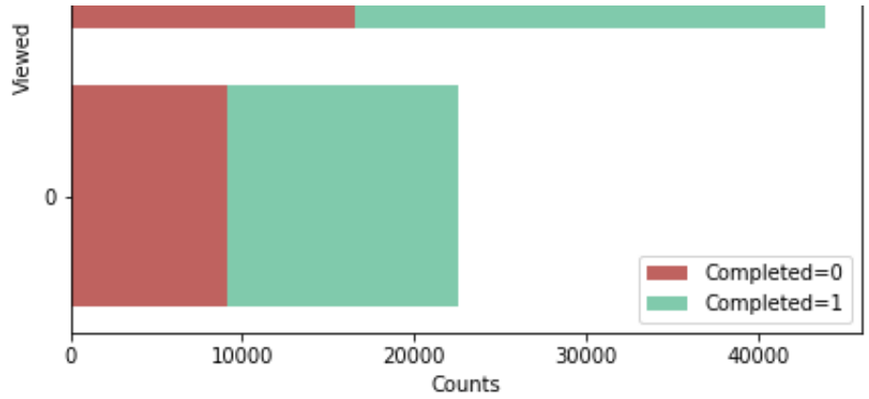
We now merge this with portfolio and profile datasets:

	user_id	offer_id	viewed	completed	reward	difficulty	duration	mobile	email	web	...	offer_type_discount	offer_type_informational	age	income	gender_F	gender_M	gender_O	signup_day	signup_month	signup_year
0	1	4	0	1	5	5	168	1	1	1	...	0	0	55	112000.0	1	0	0	15	7	2017
1	1	3	0	1	0	0	96	1	1	1	...	0	1	55	112000.0	1	0	0	15	7	2017
2	2	4	1	1	5	5	168	1	1	1	...	0	0	75	100000.0	1	0	0	9	5	2017
3	2	1	1	1	10	10	168	1	1	0	...	0	0	75	100000.0	1	0	0	9	5	2017
4	2	9	0	1	5	5	120	1	1	1	...	0	0	75	100000.0	1	0	0	9	5	2017

merged dataset

Finally, we get the completion and view data:





completion and viewing stats

As seen from the graph, the majority of the offers that were viewed were completed. A lot of offers were not viewed and yet, were completed. These are what we call wasted offers.

3. What is the completion rate of each offer and each type of offer?

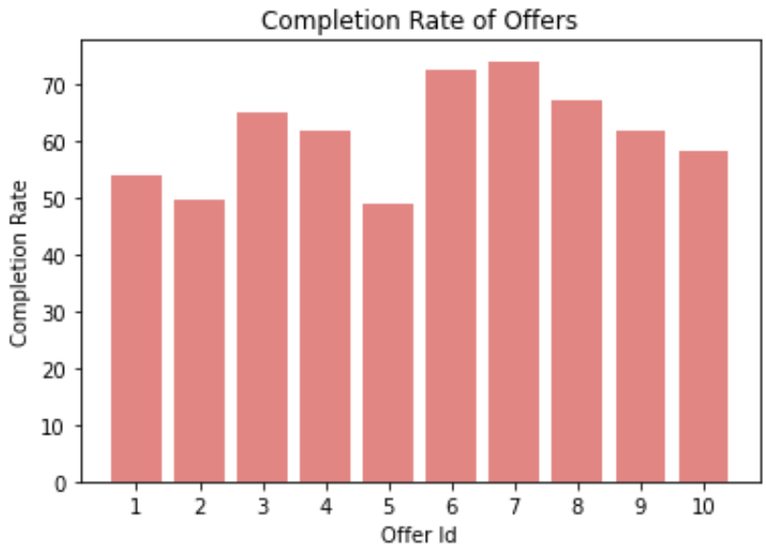
Here, the type of offer means- BOGO, Discount, and Informational. The completion of the offer is stored in the 'completed' field of the merged data frame.

We create an offer descriptor for each offer by combining its attributes:

```
0              bogo/Spend:10/Reward:10/Days:7
1              bogo/Spend:10/Reward:10/Days:5
2     informational/Spend:0/Reward:0/Days:4
3               bogo/Spend:5/Reward:5/Days:7
4        discount/Spend:20/Reward:5/Days:10
5         discount/Spend:7/Reward:3/Days:7
6        discount/Spend:10/Reward:2/Days:10
7     informational/Spend:0/Reward:0/Days:3
8               bogo/Spend:5/Reward:5/Days:5
9         discount/Spend:10/Reward:2/Days:7
```

offer descriptors

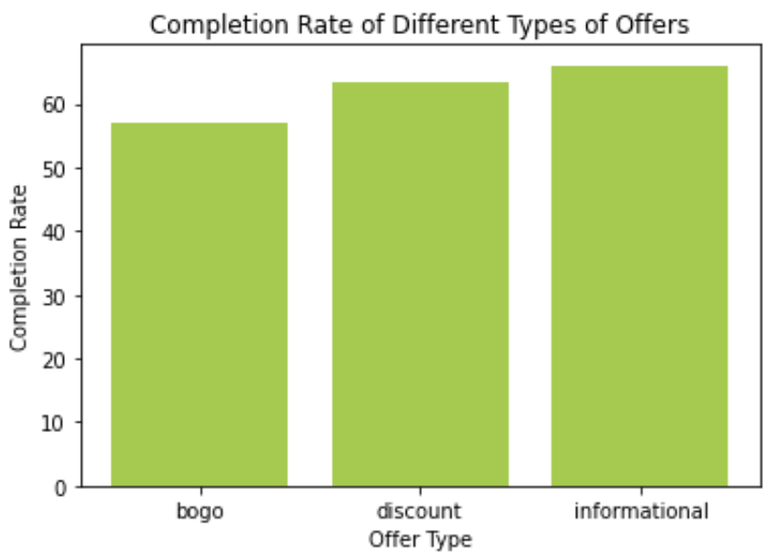
We obtain the following graph for the completion rates of the offers:



completion rate of offers

As we can see from the graph, the offer with id=7, i.e. (discount/ Spend:10/ Reward:2/ Days:10) had the highest completion rate among all. The offer with id=5 (discount/Spend:20/Reward:5/Days:10) had the lowest completion rate. The offer with id 5 required higher spendings and was more difficult to achieve.

For different types of offers:



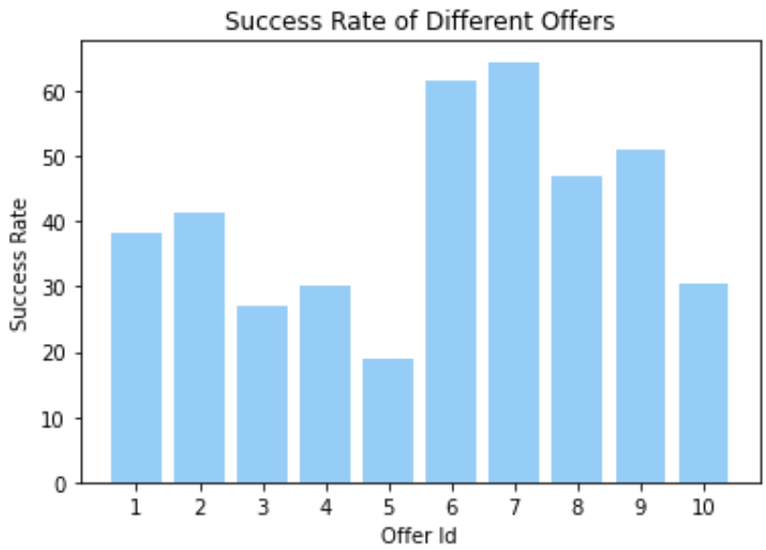
completion rate of different types of offers

As we can see, the most offers that were completed were informational offers. The BOGO offers were the least completed.

4. What is the success rate of offers?

The success rate differs from the completion rate in that an offer is only considered to be successful if it was viewed and completed. The offers which were completed without being viewed were wasted since the customer anyways spent the money on the product.

We obtain the success rates as:



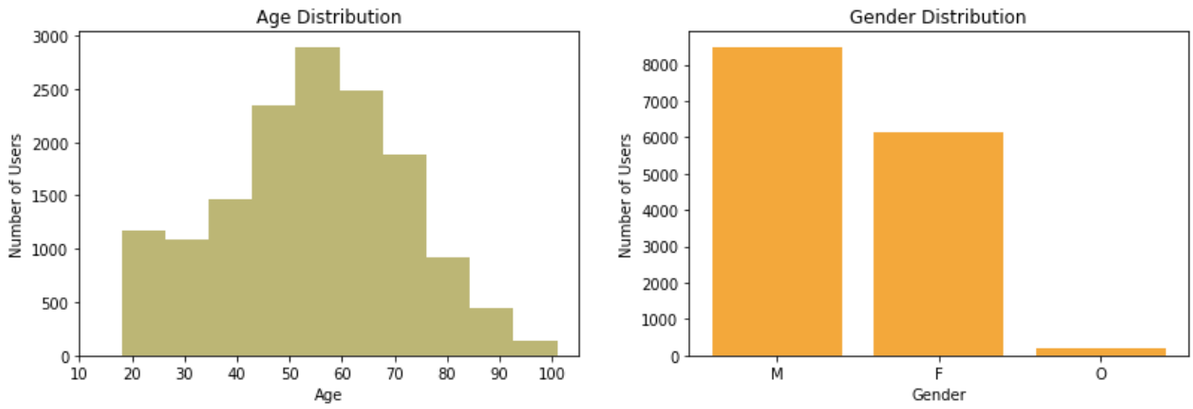
the success rate of the offers

As with the completion rate, the offer with id=7, i.e. discount/ Spend:10/ Reward:2/ Days:10 had the highest success rate amongst all 10. The offer with id=5 (discount/Spend:20/Reward:5/Days:10) had the lowest success rate.

5. What does the demographic of the users look like?

Here we will explore what type of users are present. Later we will

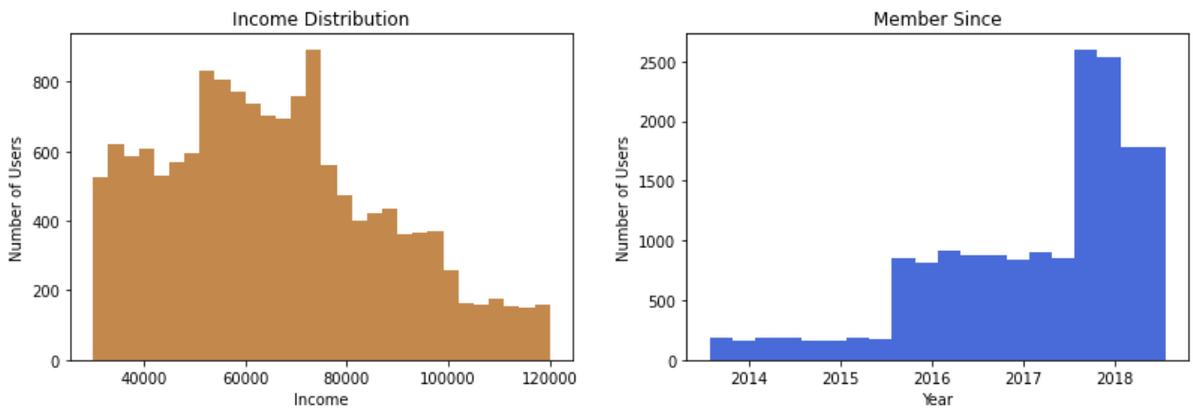
explore the relationship between the user and the offer completion rate.



(L) Age distribution. (R) Gender distribution.

The users are distributed normally, with most users present in the ages of 40–70.

There is the maximum number of males present, followed by females, and then those who do not identify as either male or female.



(L) Income distribution. (R) Membership age distribution

Most of the users earn less than 80,000. There are some users with salaries reaching up to 120,000.

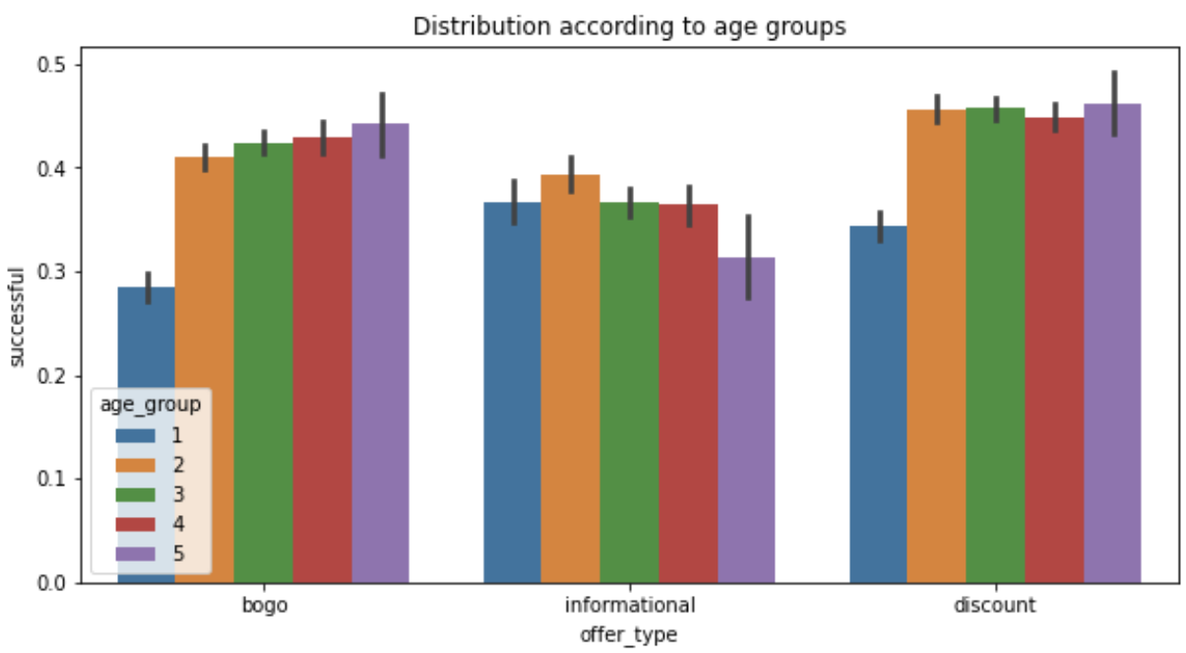
Most of the users are members since mid-2017. Very few users have been members since 2014.

6. What is the correlation between attributes of users and offers and offer success rate?

Finally, we will look at how does user demographics influence the offer success rate along with the offer attributes that might affect the offer success rate. Particularly, we will look at different types of users based on their ages, incomes, genders, and signup dates, and different types of channels that are used to advertise the offers. To reiterate, the successful offers are those which are viewed and completed.

What is the distribution of successful offers among the user age?

Since age is a continuous variable, we divide the users into 5 age groups. To do this, we use np.histogram with bins=5 to get 5 bins for the distribution of ages. The decrement of lower value of first bin is done so people with the lowest age will fall into the 1st group (instead of 0th group). Then, we will plot the graph of what was the success rate of each offer type (BOGO, Discount, and Informational offers) — in terms of how many offers of that type were successful out of those that were rolled out.



Conclusions:

1. The minimum age of the users is 18 and the oldest user is 101 years old.

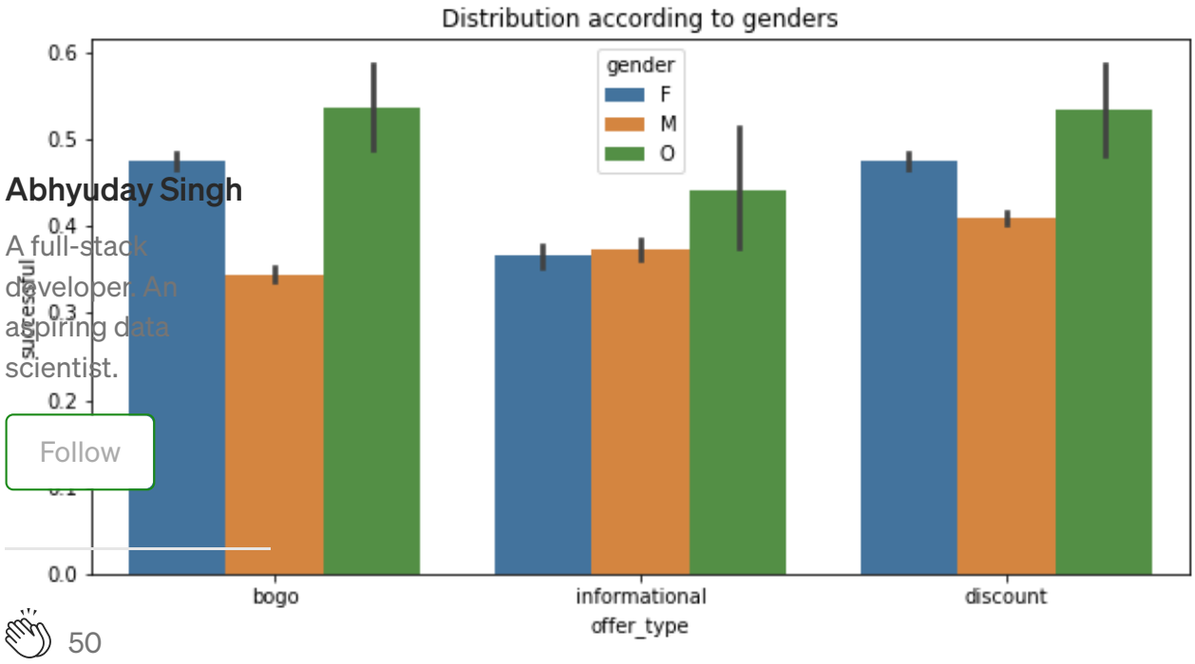
2. BOGO was less popular among the youngest age group of 17–34.6. This could be because the BOGO offers are the hardest to complete since to claim the rewards of the BOGO offer, the user has to make one big transaction, whereas benefits of discount offer can be collected with multiple smaller transactions.

3. The oldest users (84.4–101 years) preferred discount offers more than BOGO and BOGO more than informational offers.

4. Overall, people of all ages (except the youngest group) preferred discount offers. People in the age groups 2–4 (34.6–84.4) preferred BOGO over informational, and among the youngest people, informational offers had the highest success rates since those are easiest to complete.

What is the distribution of successful offers according to the users' gender?

Here, we will look at if people of any gender have a preference for one type of offer over the other. To arrange the data, we would need to first put mapped user ids in the *profile* table and merge it with the *merged* table. There are 3 categories for gender- Male, Female, and all those who do not identify as either male or female, Others. As seen from previous analysis, the highest number of users identify as males, and the least number of users identify as others.



distribution according to gender

Conclusions:

1. The success rate of BOGO and Discount offers appears to be the same for females and people of other genders.

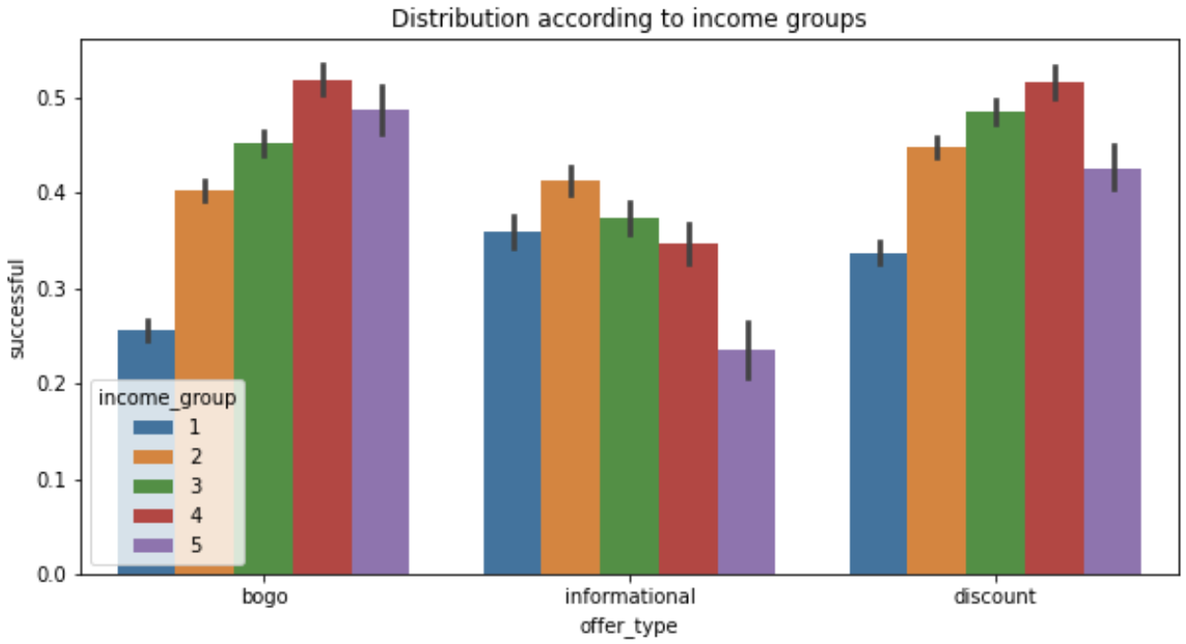
2. Males preferred discount offers more than informational offers and informational offers more than BOGOs.

3. Females and non-binary users preferred BOGO and Discount offers more than informational offers.

4. For all offer types combined, females had higher offer success rates than men.

What is the distribution of successful offers according to the users' income?

Like ages, the income of users is also continuous. Therefore, we divide the income into 5 groups using np.histogram and bins=5. This gives us income groups from 1–5, the lowest income being that of 30,000 and the highest being 120,000.



distribution according to income

Conclusions:

1. As might have been evident from the problem statement, the BOGO offer, being the most difficult, is least popular among the people in the lowest income group. This group prefers informational and discount offers over BOGOs.

2. People in the highest income group prefer BOGOs more than any other type of offers.

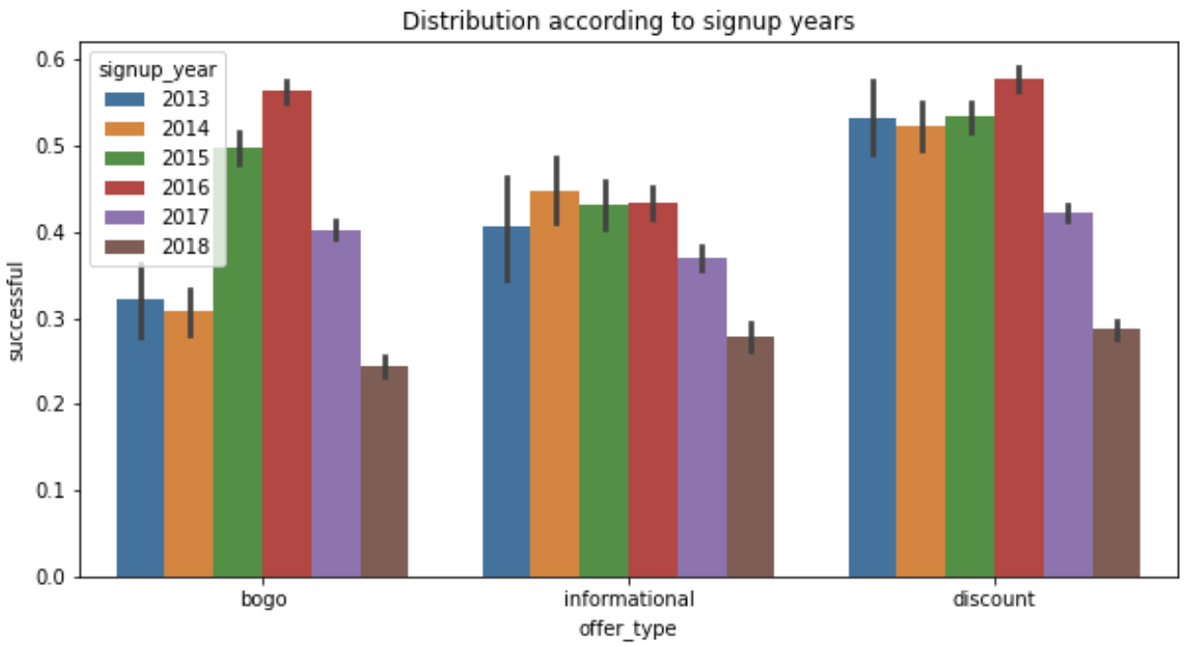
3. Generally, as the income increases, success rates for BOGOs and Discount offers increases, and that for informational offer decreases.

4. People in the second income group have similar distribution for success rates of the three types of offers.

What is the distribution of successful offers according to the users' signup date?

Here, we will look at the success rates of the different types of

offers based on the year the user signed up in.



distribution according to membership age

Conclusions:

1. For users who signed up in and after 2015, there is a drop in the success rates of the offers for every offer type.

2. Users since 2016 have the highest success rate.

3. Generally, users from all the years complete discount offers more than the other 2 types of offers.

DATA MODELLING

Now we will model the data to predict the success of offers using various classifiers, and try to improve the model. In particular, we would be looking at Logistic Regression and Random Forest Tree classifiers.

The relevant columns to train a classifier on the data are

> *age, income, gender_F, gender_M, gender_O, signup_day, signup_month, signup_year, difficulty, duration, reward, social, mobile, email, web, offer_type_bogo, offer_type_discount, offer_type_informational, and successful.*

Therefore, we would combine all-offer data set with profile and portfolio. Finally, our dataset looks like this:

	successful	age	income	gender_F	gender_M	gender_O	signup_day	signup_month	signup_year	reward	difficulty	duration	mobile	email	web	social	offer_type_bogo	offer_type_discount	offer_type_informational
0	0	55	112000.0	1	0	0	15	7	2017	5	5	168	1	1	1	0	1	0	0
1	1	75	100000.0	1	0	0	9	5	2017	5	5	168	1	1	1	0	1	0	0
2	1	68	70000.0	0	1	0	26	4	2018	5	5	168	1	1	1	0	1	0	0
3	0	65	53000.0	0	1	0	9	2	2018	5	5	168	1	1	1	0	1	0	0
4	1	65	53000.0	0	1	0	9	2	2018	5	5	168	1	1	1	0	1	0	0

final merged dataset

· There are **66501** records in our final dataset.

· We **split** the dataset into test and training sets, with a test size of 30%.

· **Scaling the dataset**: first, we will fit and transform the X_train, and then transform the X_test using that. Doing this ensures that our model does not see any part of our test set and only trains on the information from the training set.

CLASSIFIERS:

Here, we use the following two classifiers to model our data- Logistic Regressor and Random Forest Classifier. The LR is a simple linear model, and we would use the results of this as a baseline to assess further models. RFC on the other hand uses multiple decision trees to give the final prediction. The output of RFC is generally better than LR (unless overfitting occurs) because, unlike LR, RFC is not a linear model and is an ensemble

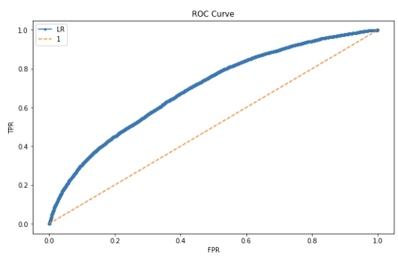
model in that it combines the output of several decision trees.

LOGISTIC REGRESSION

Firstly, we will use logistic regression as a binary classifier to predict whether a particular offer for a particular user will be successful or not. Also, this model gives the baseline model for prediction performance. The default parameters for this model are:

{ “C”:1.0, “class_weight”:null, “dual”:false, “fit_intercept”:true, “intercept_scaling”:1, “l1_ratio”:null, “max_iter”:100, “multi_class”:”auto”, “n_jobs”:null, “penalty”:”l2", “random_state”:491, “solver”:”lbfgs”, “tol”:0.0001, “verbose”:0, “warm_start”:false }

	precision	recall	f1-score	support
0	0.67	0.79	0.73	11655
1	0.61	0.46	0.52	8296
accuracy			0.65	19951
macro avg	0.64	0.63	0.63	19951
weighted avg	0.65	0.65	0.64	19951



(L) classification report of logistic regression. (R) roc-curve

The precision for positives is 0.61, i.e., out of all the predicted positives, 61% were correctly predicted. In terms of our data, this means that of all the offers that would be rolled out to users, it is predicted that 61% of them would be successful. The recall of this model is low- 0.46. This means that out of all the offers that would have been successful, only 46% were actually rolled out. The accuracy also improved by 2%, i.e., the percentage of correct predictions was improved by 2%.

Random Forest Classifier

Now, we would try to improve our results by training a random forest classifier. As mentioned above, RFC is an ensemble classifier and merges the results of various decision trees. Hence, it is likely to give better performance, unless it is subjected to overfitting of data.

The default parameters for this model are:

{ “bootstrap”:true, “ccp_alpha”:0.0, “class_weight”:null, “criterion”:”gini”, “max_depth”:null, “max_features”:”auto”, “max_leaf_nodes”:null, “max_samples”:null, “min_impurity_decrease”:0.0, “min_impurity_split”:null, “min_samples_leaf”:1, “min_samples_split”:2, “min_weight_fraction_leaf”:0.0, “n_estimators”:100, “n_jobs”:null, “oob_score”:false, “random_state”:null, “verbose”:0, “warm_start”:false }

	precision	recall	f1-score	support
0	0.70	0.76	0.73	11655
1	0.62	0.54	0.58	8296
accuracy			0.67	19951
macro avg	0.66	0.65	0.65	19951
weighted avg	0.66	0.67	0.67	19951

classification report of random forest regressor

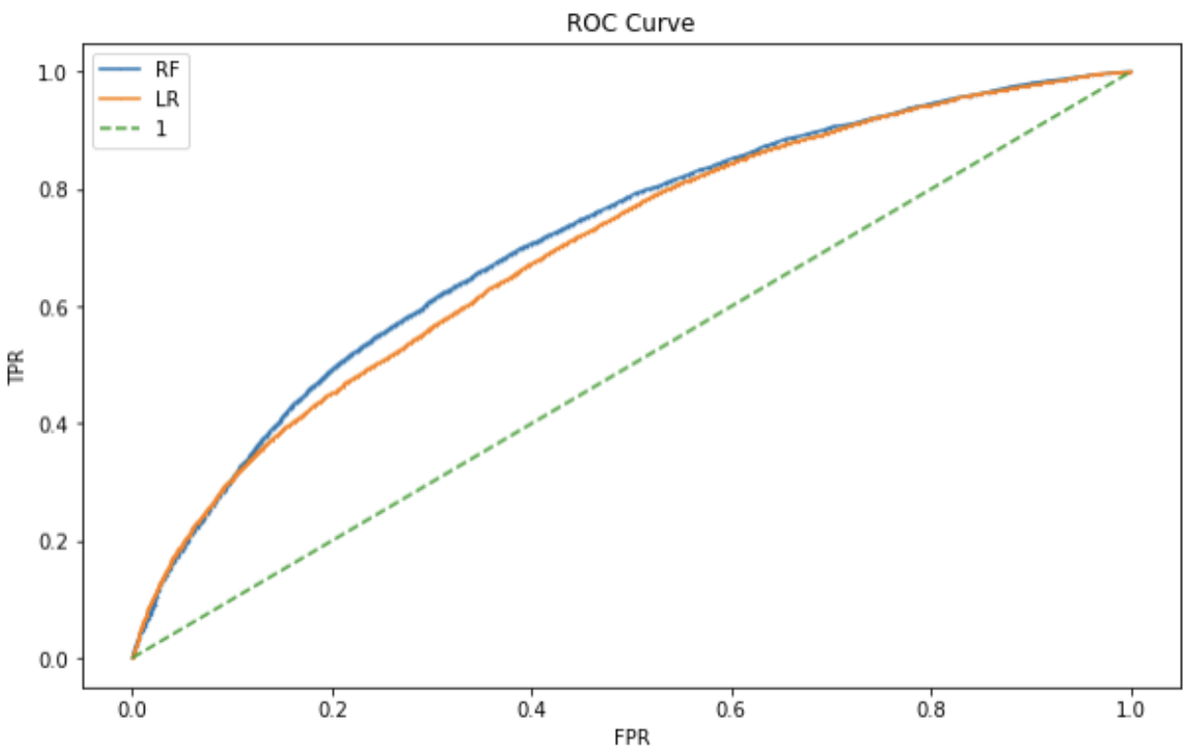
As we can see, the random forest classifier gives better accuracy for the classification than the logistic regression.

As we can see, the random forest classifier gives better accuracy for the classification than the logistic regressor.

In RFC, the precision for positives saw a slight increase to 0.62, i.e., out of all the predicted positives, 62% were correctly

predicted. In terms of our data, this means that of all the offers that would be rolled out to users, it is predicted that 62% of them would be successful. This is a 1% increase from the LR model. The recall of this model saw a big jump of 0.8. This means that out of all the offers that would have been successful, RFC predicted that only 54% of them would be successful. There is a 2% increase in accuracy as well.

> In particular, the random forest classifier has better precision and recall for those for whom the offer will be successful.



roc-curve for RFC

The ROC Curve for RFC and LR shows that the area under the curve for RFC is more than that of LR, meaning there is better separability of successful and unsuccessful offers in the RFC model.

GridSearchCV

Lastly, we try to improve the accuracy of the RFC model by performing a grid search over a variety of parameters. This is known as **hyperparameter tuning**. Hyperparameters are parameters of an algorithm that can be adjusted to tune the performance of the model. With the RFC model, these hyperparameters can be the number of decision trees, the depth of the tree, etc.

We perform Grid Search on the data with cross-validation of 4, scoring of 'roc-auc', number of jobs 8, and following hyperparameters:

{

'max_features': ['auto', None],

'min_samples_split': [2, 5],

'n_estimators': [100, 200]

}

· The cross-validation would validate the model on a separate dataset- the validation set taken out of the training set on each fold.

· The max_features is the number of features on which the model would be trained. The default is 'auto', which takes the sqrt(n_features) and None means all features.

· The min_samples_split is the number of minimum samples a node should contain before further splitting. The default is 2.

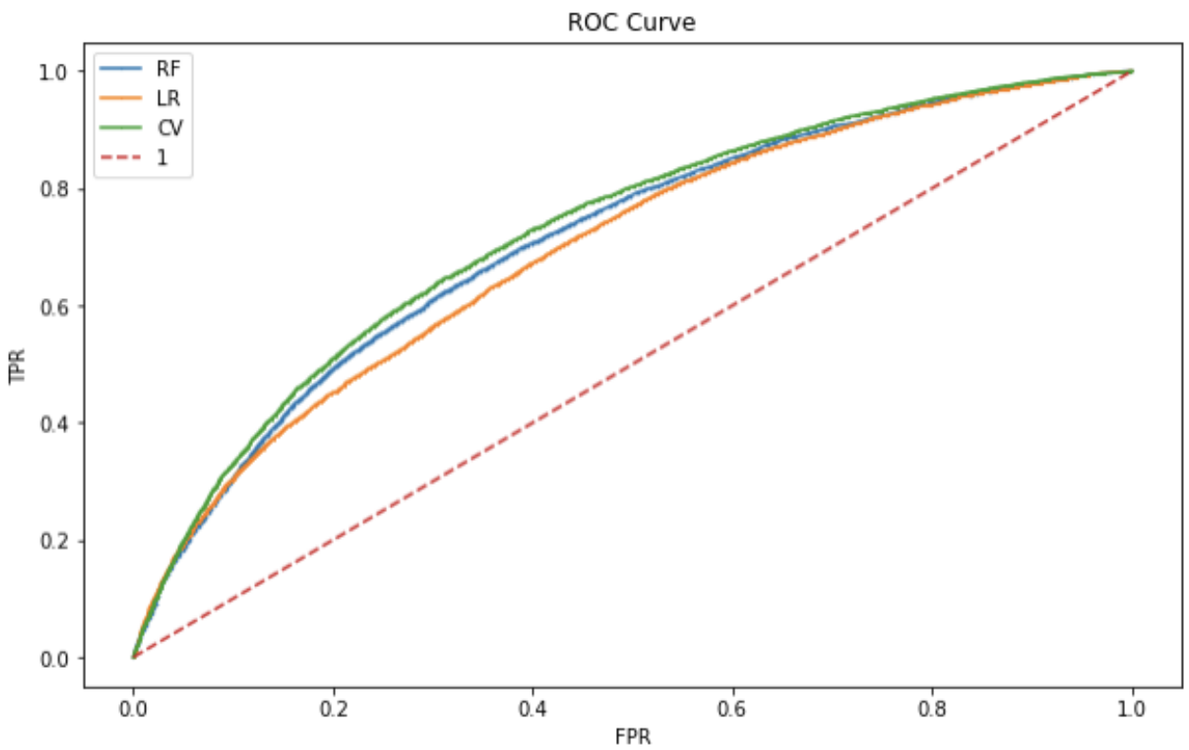
· The n_estimators is the number of decision trees. A higher

number of decision trees means better performance on the training set. However, this could lead to overfitting of data, dropping the performance on the test set and real-world data. The default is 100 and here we are also evaluating for 200 trees as well.

The model after training returned the best score of 0.725202919830348 and the following features:

{‘max_features’: None, ‘min_samples_split’: 5, ‘n_estimators’: 200}

This implies that the model performs better with 200 estimators, minimum of 5 samples at a node before splitting, and all the features.



roc-curve for all three models

The ROC Curve shows that through hyperparameter tuning, we achieved better separability than the RFC model and the LR model.

CONCLUSION:

This project aims at analyzing the customers' reception to the offers rolled out by Starbucks through their mobile app. We probed through the provided data sets and deduced the patterns that would help Starbucks in improving its ad-targeting.

We were able to unearth the following insights from the data:

1. More number of BOGO and discount offers were rolled out than informational offers.

2. Majority of the offers that were viewed were completed. There were around 1400 offers that were wasted, i.e., users completed them through their regular spending without being aware of them.

3. The offer with id=7, i.e. discount/Spend:10/Reward:2/Days:10 had the highest completion rate among all. The offer with id=5 (discount/Spend:20/Reward:5/Days:10) had the lowest completion rate. The offer with id 5 required higher spending and was more difficult to achieve.

4. Also, the most offers that were completed were informational offers. The BOGO offers were the least completed.

5. The offer with id=7, i.e., discount/Spend:10/Reward:2/Days:10 had the highest success rate amongst all 10. The offer with id=5 (discount/Spend:20/Reward:5/Days:10) had the lowest success

rate.

6. Regarding users, most were males. The age group of 40–70 years had the highest users. Most of the users earn less than 80,000. There are some users with salaries reaching up to 120,000. Most of the users are members since mid-2017. Very few users have been members since 2014.

7. Correlation between offers' success and user factors:

(i) Age-

· BOGO was less popular among the youngest age group of 17–34.6.

· The oldest users (84.4–101 years) preferred discount offers more than BOGO and BOGO more than informational offers.

· Overall, people of all ages (except the youngest group) preferred discount offers.

(ii) Gender-

· Males preferred discount offers more than informational offers and informational offers more than BOGOs.

· Females and non-binary users preferred BOGO and Discount offers more than informational offers.

· For all offer types combined, females had higher offer success rates than men.

(iii) Income-

· People in the highest income group prefer BOGOs more than any other type of offers.

· The BOGO offer, being the most difficult, is least popular among the people in the lowest income group.

· Generally, as the income increases, success rates for BOGOs and Discount offers increases, and that for informational offer decreases. Therefore, for people with higher incomes, reward offers would be better.

(iv) Signup Date-

· Users since 2016 has the highest success rate.

· Generally, users from all the years complete discount offers more than the other 2 types of offers.

· For users who signed up in and after 2015, there is a drop in the success rates of the offers for every offer type. This means users from 2015 are more likely to complete the presented offers.

FUTURE SCOPE:

1. Grid Search- the model can be put through GridSearchCV for various classifiers to improve the performance through validation. In the end, we could get hyperparameters that would give the best performance out of all the hyperparameters.

2. Neural Networks- training the data on neural networks might give better prediction performance.

3. Feature Engineering- more data can be generated from the given data. For example, PCA can be used to extract features, or

features can be combined to generate polynomial features.

4. Channels- more channels of advertising the offers can be explored and analyzed.

> *You can find the full code on my GitHub repository: https://github.com/singhabhyuday01/starbucks-project*