Starbucks Customer Behavior Analysis

Rahul Kumar

[*] Department of computer science and Engineering ‘
[**] Lovely Professional University, Phagwara, Punjab”

Abstract- For any Brand customer behavior analysis plays an important role in overall growth specially for those brands who specialized on D2C (Direct to Customer) model The case we discuss here could be a real-life marketing strategy study supported a simulated data set that mimics customer behavior on the Starbucks rewards mobile app. The goal is to mix transaction, demographic and offer data to research which demographic groups respond best to which provide type; Also, we'll build a supervised learning model(specifically, a classification problem) that predicts whether or not someone will reply to a suggestion. Similar machine learning prediction problems are Finding Donors for CharityML & Boston House Price Prediction, which both used supervised learning models. The demographic of data consists of many factors from gender to age. In D2C also this varies from what kind of D2C model or products are being used, From products like smart wearable to Restaurants chains all of them need to Know what kind of customer they have and what they Want. Brands like Starbucks also face many challenges while dealing with their customers like what kind of products they like.

I. INTRODUCTION

Machine learning plays an important role in predictive analysis which further helps brands to grow and increase their customer base and it's same for the Starbucks we are going to do the behavior analysis of Starbucks customers to find out what kind of offers they will get attracted to for this the data set we are using contain simulated data that mimics customer behavior on the Starbucks rewards mobile app. Once every few days, Starbucks sends out a proposal to users of the mobile app. a proposal are merely an advert for a drink or an actual offer sort of reduction or BOGO (buy one get one free). Some users might not receive any offer during certain weeks. Not all users receive identical offers, which is that the challenge to resolve with this data set. The task is to combine transactions, demographic, and offer data to determine which demographic groups respond best to which give type. This data set might be a simplified version of the important Starbucks app because the underlying simulator only has one product whereas Starbucks actually sells dozens of products. Every offer incorporates a validity period before the offer expires. As an example, a BOGO offer may be valid for under 5 days. within the info set that informational offer have a validity period while these ads are merely providing information a few products; as an example, if an informational offer has 7 days of validity, you will be able to assume the customer is feeling the influence of the offer for 7 days after receiving the advertisement. The Goal Use purchasing habits to arrive at discount measures to obtain and retain customers • Identify groups of individuals that are most likely to be responsive to rebates. The analysis of the given data and its findings are only observational and not the result of a formal study. This analysis comes under Starbucks Rewards Prog

II. DATA EXPLORATION

The data is contained in three files:

- portfolio.json - containing offer ids and meta data about each offer (duration, type, etc.)
- profile.json - demographic data for each customer
- transcript.json - records for transactions, offers received, offers viewed, and offers completed

Here is the schema and explanation of each variable in the files:

portfolio.json

- id (string) - offer id
- offer_type (string) - type of offer ie BOGO, discount, informational
- difficulty (int) - minimum required spend to complete an offer
- reward (int) - reward given for completing an offer
- duration (int) -
- channels (list of strings)

profile.json

- age (int) - age of the customer

- became_member_on (int) - date when customer created an app account
- gender (str) - gender of the customer (note some entries contain 'O' for other rather than M or F)
- id (str) - customer id
- income (float) - customer's income

transcript.json

- event (str) - record description (ie transaction, offer received, offer viewed, etc.)
- person (str) - customer id
- time (int) - time in hours. The data begins at time t=0
- value - (dict of strings) - either an offer id or transaction amount depending on the record

Data Preparation and Cleaning

```
#Portfolio
#1 update the column name to merge later with other dataframes
#2 Seperate values from channels columns into different columns with 0/1 value
portfolio.rename(columns={'id':'offer_id'},inplace=True)

channels = set()
for i in portfolio.channels:
    channels.update(set(i))

for i in channels:
    portfolio[i] = portfolio.channels.apply(lambda x: 1 if i in x else 0)

portfolio.drop(['channels'], axis=1, inplace=True)

portfolio.head()
```

	reward	difficulty	duration	offer_type	offer_id	email	mobile	social	web
0	10	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	1	1	1	0
1	10	10	5	bogo	4d5c57ea9a6940dd891ad53e9dbe8da0	1	1	1	1
2	0	0	4	informational	3f207df678b143eea3cee63160fa8bed	1	1	0	1
3	5	5	7	bogo	9b98b8c7a33c4b65b9aebfe6a799e6d9	1	1	0	1
4	5	20	10	discount	0b1e1539f2cc45b7b9fa7c272da2e1d7	1	0	0	1

Figure 1Portfolio Data cleaning

#Profile

1-we will replace age values with age groups to get better analysis.

2- replace become_member_on with membership year only and rename to year

3-Going to fill null values

4-rename id to the person so that we can merge later with other data frames

update age

#transcript

#1-firstly for transcript extract the value and create a new column for each value

#2-merge two resulting columns

#3-rename columns for later merge with other datasets

#4- fill numm values

extract information from the value field and create a new column for each type

The **duration** variable in the **portfolio** file would play a major role in the cleaning process where it will be compared to the **time** of each transcript provided in the **transcript** file. Therefore, it was converted to be in **hours** instead of days as the **time** variable is.
The **became_member_on** variable in the **profile** file could be used for finding trends with respect to the time most customers have joined Starbucks App. Therefore, first, the column was converted to be a **date** instead of integer and then a **membership_period** by *Month* was created where it shows the **month** and **year** when the customer become a member.

For **portfolio** data, since the values in column “channels” is in a list, and in the list are one or more values in “email”, “mobile”, “social” and “wehence we made it as dummy variables using the code below:

```
#create dummy columns
portfolio=portfolio.join(portfolio['channels'].str.join('|
```

After Doing this we Got

	difficulty	duration	id	offer_type	reward	channel_email	channel_mobile	channel_social	channel_web
0	10	7	ae264e3637204a6fb9bb56bc8210ddfd	bogo	10	1	1	1	0
1	10	5	4d5c57ea9a6940dd891ad53e9dbe8da0	bogo	10	1	1	1	1
2	0	4	3f207df678b143eea3cee63160fa8bed	informational	0	1	1	0	1
3	5	7	9b98b8c7a33c4b65b9aebfe6a799e6d9	bogo	5	1	1	0	1
4	20	10	0b1e1539f2cc45b7b9fa7c272da2e1d7	discount	5	1	0	0	1
5	7	7	2298d6c36e964ae4a3e7e9706d1fb8c2	discount	3	1	1	1	1
6	10	10	fafdcd668e3743c1bb461111dcafc2a4	discount	2	1	1	1	1
7	0	3	5a8bc65990b245e5a138643cd4eb9837	informational	0	1	1	1	0
8	5	5	f19421c1d4aa40978ebb69ca19b0e20d	bogo	5	1	1	1	1
9	10	7	2906b810c7d4411798c6938adc9daaa5	discount	2	1	1	0	1

Figure 2dummy columns

Portfolio Raw Data which clearly shows there are 10 choices of offers, and their difficulty(i.e. the threshold to get the reward), duration, offer id, offer type, reward, channel.

Next, we take a look at **profile** data:

```
profile.sample(5)
```

	age	became_member_on	gender	id	income
12202	63	20180102	M	5f29fa9bbadd4010b698f16b89b1fd80	88000.0
7434	54	20170713	M	4eab02d70ac241a8911502dc1954069d	32000.0
11481	49	20131023	M	dac5cef904764a609e3a3b3d4f0058e8	64000.0
15127	27	20170907	M	24b13db9ecca4db1b18a86c2bbc64b03	53000.0
5637	55	20160108	M	1bc9c50db8e849e78f6491bda9e9f7cb	59000.0

we can see from the data set that the age, the date when the customer became a member(in yymmdd format), gender(M, F, O or Nan), person id, and income are listed.

It is a good way to visualize the age & income distribution of our customer base. For Age, it is better to group the customers into a group for every 5 years interval using the code below:

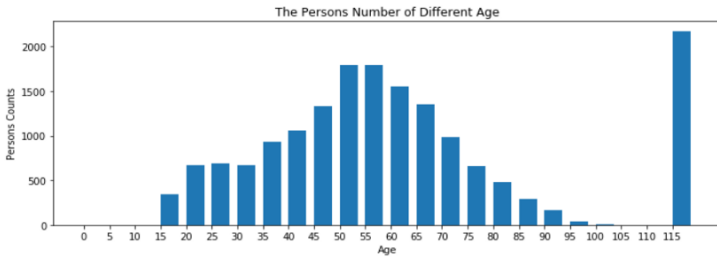

Figure 3 age vs number

Persons Number By Age groupssince those who did not enter their age are marked as 118 years old, we can see from the chart that the amount is over 2000 for this group. And apart from this, most of the customers fall in the age between 50 and 60 years old.

For income distribution we draw the pie chart below

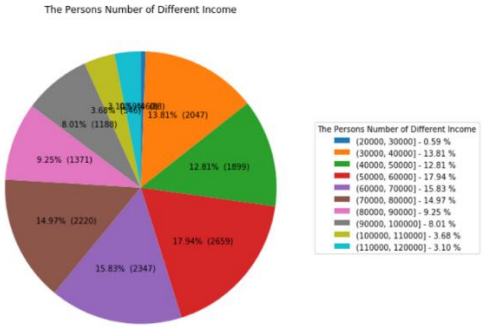

Figure 4 Income distribution pieechart

Customer income by age was also an important factor in consideration while looking at the data to ensure that what

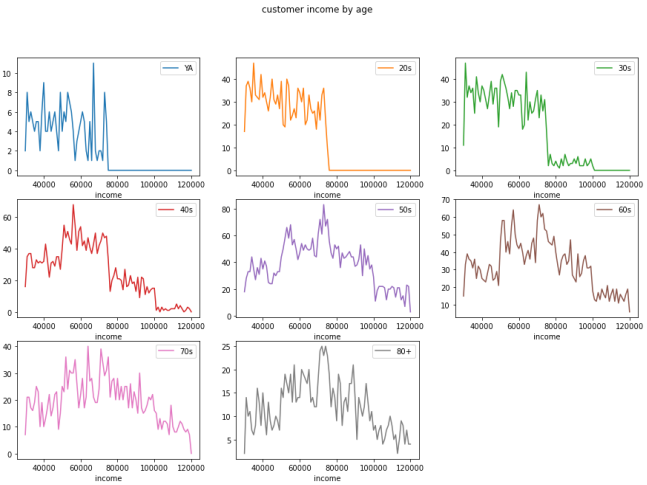

Figure 5 income by age

The Exploration and Visualization for the Profile Dataset showed the following:

- The dataset has no duplicated rows.
- The dataset has 2175 missing values on each of: ‘gender’, ’income’ variables.
- The customers’ ages range from 18 to 101. Although those 2175 customers were registered at age 118, I still considered this specific age an outlier b/c it appears clear that there is something wrong related to these 2175 rows in the dataset.

Exploring and visualizing ‘gender’, ’income’ and ’age’ variables

The missing values in ‘gender’ and ‘income’ variables which are related solely and specifically with the 2175 customers registered at age 118. In other words, customers at age 118 have no registered ‘gender’ and ‘income’. **This needs to be cleaned in the Data Pre-processing Section.**

```
profile[profile['age']==118].count()

gender              0
age              2175
id               2175
became_member_on 2175
income              0
dtype: int64
```

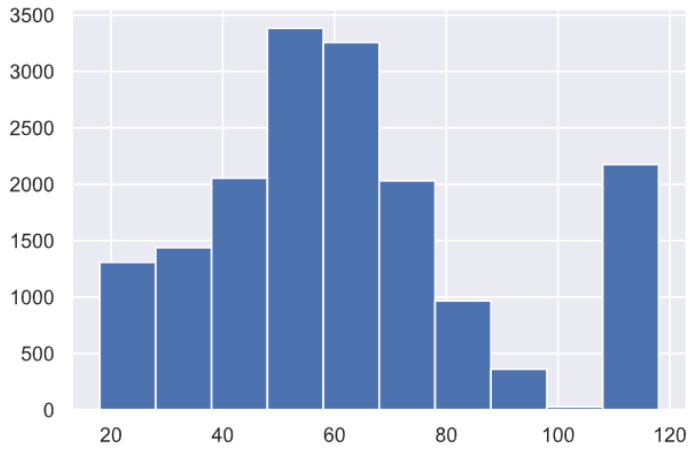

Figure 6gea

Age Distribution

According to the available data, There are three ‘gender’ categories into which the customers fall in (M, F, and O). Keeping in our mind the above observation that there are 2175 missing values, Male Customers (8484 men) are more than

Female Customers(6129 women) with 57% of customers are Males compared to 41% Females. However, there are 212 customers chose "O" as their gender.

III. Data Preprocessing

Data preprocessing was on of the major factors in consideration while analyzing datasets we used methods to make it clear of the time reward and transection for each round in transcript data set, based on the resultant data frame we want to see more clearly on the offer information as well as the related customer's information, hence we combined it with portfolio and profile.

```
person_offer_demographic.head()
```

person	age	gender	income	consolidate_offer_id	offer_type	difficulty	reward	duration	offer_time	channel_email
009655768c64bdeb2e877511632db8f	33	M	72000.0	5a8bc65990b245e5a138643cd4eb9837	informational	0	0	3	7.0	1
009655768c64bdeb2e877511632db8f	33	M	72000.0	5a8bc65990b245e5a138643cd4eb9837	informational	0	0	3	8.0	1
009655768c64bdeb2e877511632db8f	33	M	72000.0	3f207df678b143eea3cee63160fa8bed	informational	0	0	4	14.0	1
009655768c64bdeb2e877511632db8f	33	M	72000.0	3f207df678b143eea3cee63160fa8bed	informational	0	0	4	15.5	1
009655768c64bdeb2e877511632db8f	33	M	72000.0	f19421c1d4aa40978ebb69ca19b0e20d	bogo	5	5	5	17.0	1

Preprocessed Transcript with the person and other information.

Note that it is tricky to figure out *how many completed offers were from offers that the person viewed beforehand* because some of the offers were completed first and then viewed afterward. The reason to do this is that even though the customers made the transaction and completed the offer, he/she may not be aware of the offer. In other words, his/her action may not be offer-oriented, hence it should not be counted as **responded** to the offer.
Here is the pseudo-code for separating viewed completed vs. no viewed & completed offers:

Separated Viewed & completed vs. No viewed & Completed offer Records
Similar to step I attempted to separate "received and completed" vs. "completed before received" offers to see whether there are offers completed without being received. However, the result shows no such records, so I don't put the process in this article.

```
#Loop through the unique persons one by one in the transaction data. Each time:

#1.Extract all transaction records for this person(using transaction['person_id']==person_id)

#2.Sum the transaction amount of this person and mark as "total".

#3.Loop through the transaction records: For every loop, store the transaction time and amount. Find in the transcript(non-transaction
#records) satisfying "offer completed" and the same time as transaction time. Judge whether the offer(s) falls in the viewed&completed list.
#If YES, add the transaction amount to viewed&completed transaction amount. If NO, add it to noviewed&completed transaction amount.
```

Next, as discussed above, in the transcript transaction data, there are the only person, amounts, and time given, without telling us whether it is related to any offer. Hence, we need to differentiate total transaction, transaction-related to viewed&completed offers and transaction-related to noviewed&completed offers.

	index	total	view_complete_tran	noview_complete_tran
4481	4363de49f81748e8979fe15f072765b3	25.48	0	25.48
7056	6a6eace355604449a77fc84db4561f38	133.97	38.73	95.24
2327	247b5e234272439594e9b4024d7a6eea	52.09	33.98	18.11
7683	73d2f55a6c8240a2815efbe3ff95d171	134.01	49.72	84.29
4612	45546c203f724e3ba885b4e84518bca9	98.82	59.51	39.31

Tables to be Generated:
transcript_new: the transcript with offer id column name consolidated ((dup)rows are persons, columns are event, time, offer id, amount, reward)

person_and_offer: transcript_new joins portfolio((dup) adding offer info to transcript_new)

person_offer_demographic: person_and_offer joins parts of profile((dup) adding personal info to person_and_offer)

offer: (nodup)rows are persons, columns are counts of offers received, completed, viewed&completed, noviewed&completed, received bogo,received discount, received informational, v&c_bogo, v&c_discount, v&c_discount, v&c_informational, etc.

offer_record: (dup)rows are persons, columns are v&c offer id and time.

offer_norec_comp: (nodup)rows are persons, columns are counts of offers completed, completed bogo, completed discount, completed informational, received&completed, noreceived&completed, nr&c_bogo, nr&c_discount, nr&c_informational, etc.

transaction_gen: (nodup)rows are persons, columns are total amount of transactions(including those not completed), amount of transactions related to viewed&completed offers, amount of transactions related to noviewed&completed offers, etc.

IV. Customer Behavior Analysis

Behavior analysis was what we focused most on cause this was the major factor deterring our final prediction model for that I combined profile with the data set generated and got customer information offer and transaction in one table

```
person_all_information.head()
```

	person	age	gender	income	became_member_on	total	view_complete_tran	noview_complete_tran	receive	rec_bogo	...
0	0009655768c64bdeb2e877511632db8f	33	2	72000.0	736440	127.60	22.68	104.92	5	1	...
1	00116118485d4dfda04fdbaba9a87b5c	118	2	NaN	736809	4.09	0.00	4.09	2	2	...
2	0011e0d4e6b944f998e987f904e8c1e5	40	2	57000.0	736703	79.46	33.98	45.48	5	1	...
3	0020c2b971eb4e9188eac86d93036a77	59	2	90000.0	736027	196.86	34.87	161.99	5	2	...
4	0020ccbbb6d84e358d3414a3ff76cffd	24	2	60000.0	736279	154.05	36.50	117.55	4	2	...

Data Set Combing Customer Information & Offer& Transaction
For the visualization purpose, we applied the given processes to make new variable "responded" to be "T" when viewed &

completed offer amount is non-zero, otherwise “F”; And made gender variable to be 0,1,2 in order to be shown to the axis: after completing that we drew a pair plot for the variables age and gender income became_member_on view_complete_trn and fill any nan with zero

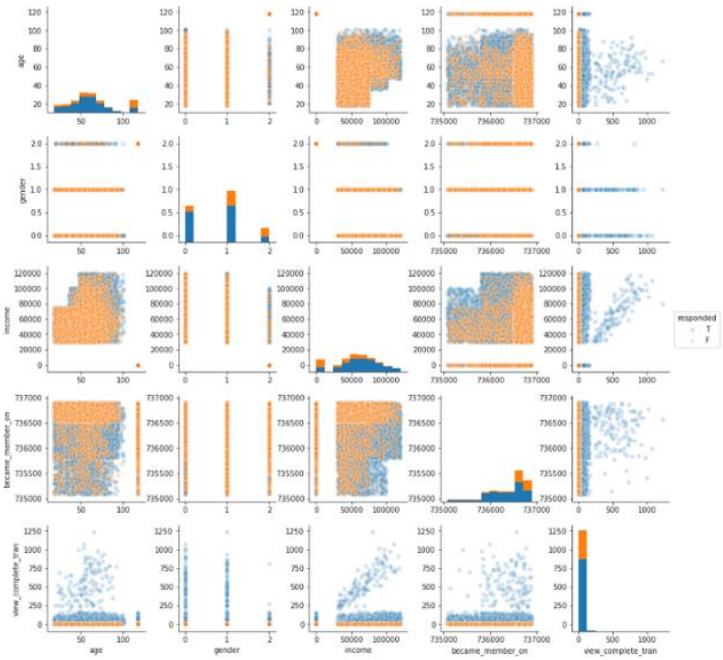

Figure 7Multiple Variable Analysis

Pair plot is one of the best graph while taking consideration the multiple variable the response of the above graph can clearly show the positive relations with age, income, and became_member_on.
Also for those customers that amight be newly registered the mobile app and have low income we can see orange line for them in the graph, from gender to gender response we can see females tends to response to more offers as compare to the male furthermore tose who did not provide any persenol information tend not to respond to the offer

V. Data Modelling

In this part, we are going to build a machine learning model that predicts whether or not someone will respond to an offer.
From the above analysis, we know that age, gender, income, membership date, offer type can affect whether a customer responds to an offer.Since the offer record generated in Part II, step 3 is customer unique and only contains the amount of each type completed/received/viewed completed offers, and each customer may receive more than one offer, we need to preprocess the data again such that we have train/test data sets without person id, only with age, income, gender, membership date, offer type, etc, as our “X”, and “responded” as our “y”.Also, we note that membership date appears in the format of yymmdd, this should be either turned into ordinal variables or extract its year, month, date to capture its real sequence. Otherwise, if you directly use this yymmdd number, we will have strange results, e.g. For the pairs 20170831 vs. 20170901, and 20170901 vs. 20170902, they are both 1-day-distant pairs, however, the number difference is not the same. Here, we chose to use extracting year&month of membership date.
Furthermore, the “gender” values are in M, F, O, and NaN, while “responded” values are in BOGO, discount, and informational, which both are categorical values. However, most machine learning algorithm requires the input to be numeric. Hence, these two variables should be turned into dummy variables.For a limited time, I just simply dropped all the NaNs and conducted the actions above(including train_test_split) and got the train/test data sets we want:

	age	income	gender_F	gender_M	gender_O	gender_nan	member_year_2013.0	member_year_2014.0	member_year_2015.0	member_year_2016.0	...	n
6050	87.0	97000.0	1	0	0	0	0	0	0	0	...	
556	59.0	83000.0	1	0	0	0	0	0	0	1	...	
10245	47.0	36000.0	0	1	0	0	0	0	0	0	...	
12685	58.0	97000.0	1	0	0	0	0	0	0	0	...	
14741	36.0	50000.0	1	0	0	0	0	1	0	0	...	

Observing the table we found that they are imbalanced so Hence, we should use an F1 score instead of accuracy, since the F1 score is a metric to balance recall&precision and to deal with imbalanced labels.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

Figure 8Precision Recall

Then we scaled the X_train & X_test to [0,1] and fit the training data into several classifiers(with its default mode) and got the F1 score(ranges from 0 to 1) for each model as shown below:

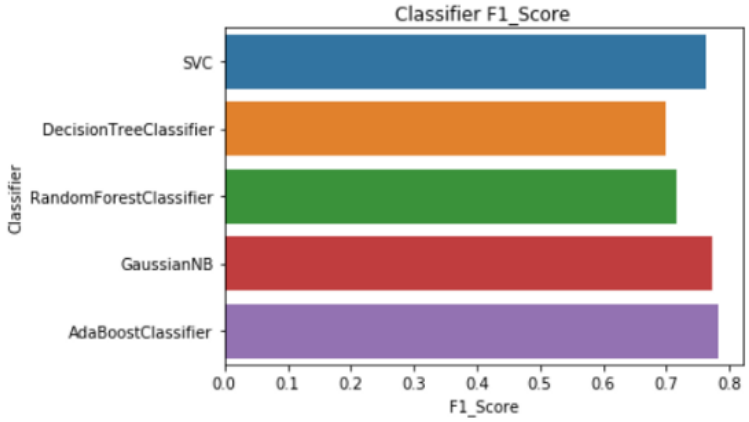

Figure 9 Model Score

Ada boost performed the best hence we can now procedd to conducting grid search on that and find best

```
param_grid = {"base_estimator__criterion" : ["gini", "entropy"],
              "base_estimator__splitter" :   ["best", "random"],
              "n_estimators": [5,  10,  20, 50],
              "learning_rate": [0.001,  0.01,  0.1,  1],
              'base_estimator__max_depth' : [1, 2, 3, 4]
             }


DTC = DecisionTreeClassifier(random_state=42)

ADA = AdaBoostClassifier(base_estimator = DTC)

grid_ada = GridSearchCV(estimator=ADA, param_grid=param_grid, scoring='f1', cv=5)
grid_ada.fit(X_train, y_train)

print('Training F1_score is:', grid_ada.score(X_train, y_train))
print('Test F1_score is:', grid_ada.score(X_test, y_test))
```

```
Training F1_score is: 0.788194444444
Test F1_score is: 0.785758459228
```

Figure 10Model improve

Best Model

2. Next, we made a prediction engine such that we take in customer info, offer type, etc, then we can transform all these information into a format which can be taken by the classifier like the test data, then predict whether the customer respond or not based on the raw profile&offer information, rather than simply seeing the test set performance without actually having any real-life usage.

Prediction Engine Testing

After building the prediction engine, we test it using a random sample of the raw profile, and chose one type of offer('BOGO',' discount' or 'informational') and found it actually works. At first glance of the data above(since the function not only replies you "respond" or not, it also returns the customer info for you to check whether this response makes sense.), the first customer has a high income, and the gender is female(in the above analysis, high income and female tends to respond to the offer), while the second person is male with lower income without responding to the offer. Also, these responses may also result from the fact that the 'discount' offer is more preferable than 'BOGO'. This result is to some extent acceptable, although more samples should be tested to validate such a conclusion.

VI. Conclusion

To conclude, during this project, we:

Cleanse the offer data such we are able to separate the finished offer data into A.viewed &completed offer and B.no viewed Compare the transaction time and therefore the offer completed time so as to work out whether the transaction is related to a completed offer(more specifically, viewed& completed offer or no viewed & completed offer or other offers), because, within the data, there are only person, amounts and time given, without telling us whether it's associated with any offer;

Next, we've got demographic, offer type and channel analysis on the entire rate.

Finally, we build a machine learning model to enable the prediction of the customer's response given a customer with age, gender, income, offer type, and other information.

Possible Enhancement within the Future:

For convenience, I drop all the NaNs; within the future, other skills will be used, like using mean, median, etc; or simply simply keep it as a worth, since within the future, there'll still be a part of customers who won't fill in their information;

This is a classification model; Alternatively, the regression model is built to predict what proportion someone will spend(i.e. transaction) supported demographics and offer type;

A web app may be built such when inputting the customer information, the prediction of response/transaction amount are often output.

References

[1] https://scikit-learn.org/stable/model_selection.html

[2] https://github.com/rahul263-stack/Starbucks-capstone-project

[3] https://www.kaggle.com/datasets/blacktile/starbucks-app-customer-reward-program-data

[4] https://pandas.pydata.org/

[5] https://numpy.org//