112356042 資管碩一 葉兆泉

112356007 資管碩一 鄭群霓

112356019 資管碩一 黃念祺

112356045 資管碩一 謝東睿

109306030 資管四 黃則維

15 Jan. 2024

Starbucks Customer Relationship Management Data Analysis

1. Dataset Introduction

This dataset is the Starbucks customer relationship management data collected by Starbucks and Udacity, and there is abundant data about Starbucks customer information, the coupons of Starbucks, and the usage condition of the coupons. It contains three separate JSON files, which are **portfolio.json, profile.json, and transcript.json respectively.**

In the **portfolio.json** file, it describes the details that how each discount offer is sent via different channels. There are total 6 columns and 10 rows in this dataset.

- offer_id (string) - It is the primary key for the Profile.json file.
- offer_type (string) - It represents the types of different offers, ie., Buy One Get One, discount, and information.
- difficulty (int) - Each special offer will have a minimum required spend to complete it.
- reward (int) - How much a customer will save when they use the reward given for completing an offer.
- duration (int) - The time for an offer to be open (in days), while the average is 6.5 days

- channels (list of strings) - How does an offer distribute to customers? There are 10 different channel combos. and each offer type can have more than one channel.

In the **profile.json**, it contains the demographic data for each customer. There are 17,000 rows and 5 columns in the dataset

- age (int) - The age of the customer is from 18 years old to 118 years old
- became_member_on (int) - the date when the 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) - It is the customer id, and also the primary key for this JSON file.
- income (float) - It is the customer's income and the average income is $65,404

While 2,175 null values appear in the ‘gender’ and ‘income’, we find out that the information of these null values all aged 118 years old; hence we delete those rows afterward.

The **transcript.json** file describes the condition of each offer record, such as transactions, offers received, offers viewed, and offers completed. There are 306,534 rows and 4 columns in this file. Although the amount of data seems large, there are only 17,000 unique person id, which is exactly the number of customers in Profile.json file. In addition, we can see that one customer can have many consuming behaviors.

- event (str) - It represents the record description of each row (i.e. transaction, offer received, offer viewed, etc.)
- person (str) - It means ‘customer id’, and it is the same as id in **Profile.json**.
- time (int) - time in hours since the start of the test. The data begins at time t=0
- value - (dict of strings) - There are two types of data parameters in this column. One is ‘offer id’ while the other one is ‘transaction amount’. The former means the customer’s event toward a specific offer from **Portfolio.json**, for instance, once the customer received an offer, a customer viewed the offer, and used the offer, the performance will be recorded in three individual rows. The latter means the amount of money the customers had paid when they also used the special offer.

2. Objective

With the CRM data, we focus on the usage condition of the coupons they send to customers to **peak the efficiency of coupons** and **provide suggestions** toward this marketing strategy.

3. Data Preprocessing

Transcript file

'value' column

First, we deal with the transcript JSON file first. This file contains four keys (columns): 'person', 'event', 'value', 'time'. Each of them has their own {key, value} structure pair. We first deal with the 'value' column. We find there are four different keys in the 'value' column, which are 'offer id', 'amount', 'offer_id', 'reward'. Since 'offer_id' and 'offer id' are the same concept, we see them as the same. Therefore,

after joining three more columns to the original transcript file and dropping the 'value' column, we've separated 'offer_id', 'amount', and 'reward'. At the end of this section, we have six columns (features).

'event' column

Second, we deal with the 'event' column. We find there are four events: offer received, offer viewed, offer completed, and transaction.

In terms of 'offer received' and 'offer viewed' we don't have anything to do with the amount and reward column. Because I DON'T even use it, I just received an offer. So we remove the amount and reward column. And rename the 'person' column to 'customer_id'.

In terms of 'offer completed', since the customer uses the offer, they don't need to pay money, and they will get a reward. The amount should be NaN, so we should drop this column. And also rename the 'person' column to 'customer_id'.

In terms of 'transaction', the transaction means a simple purchase without using an offer. So we will be dropping the 'offer_id' and 'reward' since there will be no information about them.

One-hot encoding

We then one-hot encode 'offer_received', 'offer_viewd', and 'offer_completed' to one. We'll only show the 'offer_received' figure. While the 'transaction' data frame will be used respectively later on.

Merge the data frame

In this section, we merge these three dataframes based on 'offer_id' and 'customer_id'. And we use left-outer merge. Why specify how='left'? Because

receiving an offer doesn't mean it will be seen, and an offer being seen doesn't mean it will be used.

Drop the offers were not be seen

Since we are interested in whether customers will respond or not after viewing the offer. So we will drop the offers that have not been seen. Furthermore, We found some customers complete the offer before seeing the offer. That means he **accidentally** used the offer, and we do not want that randomness, so we dropped it too. And we also drop the incorrect time sequence (complete before view.)

Portfolio file

In this file, it represents the different combinations of offers. Each offer has six columns (features.) These are 'reward', 'channels', 'difficulty', duration', 'offer_type', and 'id'. The only operation in this file is one-hot encoding of the channels since there are four channels indicated, which are 'email', 'mobile', 'social', and 'web'.

Profile file

In this file, it represents the information about the customers. It contains five columns, which are 'gender', 'age', 'id', 'became_member_on', and 'income'.

- Age

First, We can observe that if the age is 118, which doesn't make any sense, the gender and income are both NaN. Drop these observations since they are relatively minor to the original data.

- Gender

Next, we one-hot encode the 'gender' column by using a dummy variable.

- Became_member_on

Last but not least, let's deal with the date. We want the information about how long customers remain in their membership. Therefore, we calculate the difference between the current time and the timestamp of becoming a member, resulting in a customer lifetime duration.

Furthermore, We can use the transaction dataframe extracted from the transcript dataframe to gain more information. Since one customer can purchase more than one time, we want to calculate the number of transactions. And also how much a customer spends (total_amount.)

Merge all dataframes

Eventually, we merge all the data frames above to generate a 66,367 rows x 19 columns of cleaned data frames.

4. Business Insight

First: Frequency of dispensing coupons

In this part, We are curious about 'What are the better approaches to sending customer offers?'. Therefore, the main focus of data preprocessing will be on different offers and the time when the data is released in a centralized manner.

```
df_customer.isna().sum()
```

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

```
#針對customer null值欄位細看
missing_gender = df_customer[df_customer['gender'].isna()]
missing_income = df_customer[df_customer['income'].isna()]

np.sum(missing_gender['id'] == missing_income['id'])
#比對上述的null值數量一致，表示此份資料有2157筆同時沒有性別和收入，雖然收入對我們想研究的問題很重要，但因扣除2175筆，
#我們仍有14000多筆資料可以分析，故影響因不大，下一步將會將這些空值列移除
```

```
2175
```

```
id_to_remove = missing_income['id']
df_customer_no_na = df_customer[~df_customer['id'].isin(id_to_remove)]
df_customer_no_na = df_customer_no_na.reset_index(drop = True)
df_customer_no_na.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14825 entries, 0 to 14824
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   gender            14825 non-null  object
 1   age               14825 non-null  int64
 2   id                14825 non-null  object
 3   became_member_on  14825 non-null  int64
 4   income            14825 non-null  float64
dtypes: float64(1), int64(2), object(2)
memory usage: 579.2+ KB
```

Upon examining the left chart, it was observed that there are over 2000 missing values in customer data. Upon cross-referencing with the chart on the right based on gender and income, it was noted that the numbers align. After deducting these values, we still have over 14,000 records available for analysis. As the impact is minimal, the decision has been made to remove these rows containing missing values.

Next, during the examination of data types, a particular focus was placed on determining the nature of the 'object' type entries in the dataset, specifically related to the preferred method for coupon distribution. Additionally, the 'time' column was renamed to 'hour' to better align with the predefined time units in the dataset and enhance clarity for analysis.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   reward      10 non-null     int64 
 1   channels    10 non-null     object
 2   difficulty  10 non-null     int64 
 3   duration    10 non-null     int64 
 4   offer_type  10 non-null     object
 5   id          10 non-null     object
dtypes: int64(3), object(3)
memory usage: 612.0+ bytes
```

```
#針對 channels,offer_typ,id 三個object型態的欄位細看
channels_0 = df_portfolio['channels'][0]
offer_type_0 = df_portfolio['offer_type'][0]
id_0 = df_portfolio['id'][0]

print('First value in column channels:', channels_0, '  --   Data type:', type(channels_0))
print('First value in column offer_type:', offer_type_0, '  --   Data type:', type(offer_type_0))
print('First value in column id:', id_0, '  --   Data type:', type(id_0))
#發現 portfolio 的channel 是發放管道的list組合，優惠券內容是字串，優惠券id也是字串
```

```
First value in column channels: ['email', 'mobile', 'social']   --   Data type: <class 'list'>
First value in column offer_type: bogo   --   Data type: <class 'str'>
First value in column id: ae264e3637204a6fb9bb56bc8210ddfd   --   Data type: <class 'str'>
```

```
#time欄位改名方便檢視
df_transcript.rename(columns = {'time' : 'hours_since_start'}, inplace = True)
df_transcript.head(1)
```

]:

	person	event	value	hours_since_start
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	{'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}	0

Finally, regarding the handling of offers, a two-step process was implemented. Initially, offers were sorted based on 'offer_type' and 'difficulty', and a numerical code was assigned to uniquely identify each type of coupon. Subsequently, considering that earlier validation confirmed the structure of 'value' to be in dictionary format, the 'id' and its corresponding values were separated for further analysis.

```
#依照不同種優會券類型和低消重新排序並給予offer代碼方便往後檢視
df_portfolio = df_portfolio.sort_values(['offer_type', 'difficulty']).reset_index(drop = True)

from string import ascii_uppercase
df_portfolio['offer_alias'] = [ascii_uppercase[i] for i in range(df_portfolio.shape[0])]
df_portfolio
```

9]:

	reward	channels	difficulty	duration	offer_type	id	offer_alias
0	5	[web, email, mobile]	5	7	bogo	9b98b8c7a33c4b65b9aebfe6a799e6d9	A
1	5	[web, email, mobile, social]	5	5	bogo	f19421c1d4aa40978ebb69ca19b0e20d	B
2	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	C
3	10	[web, email, mobile, social]	10	5	bogo	4d5c57ea9a6940dd891ad53e9dbe8da0	D
4	3	[web, email, mobile, social]	7	7	discount	2298d6c36e964ae4a3e7e9706d1fb8c2	E
5	2	[web, email, mobile, social]	10	10	discount	fafdcd668e3743c1bb461111dcafc2a4	F
6	2	[web, email, mobile]	10	7	discount	2906b810c7d4411798c6938adc9daaa5	G
7	5	[web, email]	20	10	discount	0b1e1539f2cc45b7b9fa7c272da2e1d7	H
8	0	[web, email, mobile]	0	4	informational	3f207df678b143eea3cee63160fa8bed	I
9	0	[email, mobile, social]	0	3	informational	5a8bc65990b245e5a138643cd4eb9837	J

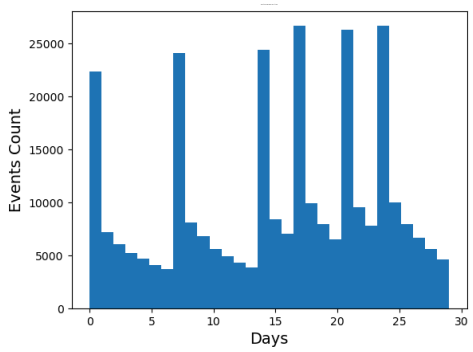
```
# 把value的key跟值拆開
value_column = df_transcript['value']
dictionary_key_column = [list(d.keys())[0] for d in value_column]
dictionary_value_column = [list(d.values())[0] for d in value_column]
value_column_split = pd.DataFrame(columns = ['dict_key', 'dict_value'])
value_column_split['dict_key'] = dictionary_key_column
value_column_split['dict_value'] = dictionary_value_column
value_column_split.head(3)
```

]:

	dict_key	dict_value
0	offer id	9b98b8c7a33c4b65b9aebfe6a799e6d9
1	offer id	0b1e1539f2cc45b7b9fa7c272da2e1d7
2	offer id	2906b810c7d4411798c6938adc9daaa5

```
# 移除原本value欄位
df_transcript_value_dic = df_transcript.drop('value', axis = 1)
df_transcript_value_dic = pd.concat([df_transcript_value_dic, value_column_split], axis = 1)
```

During the data exploration process, it was observed that the marketing campaigns in this dataset occur approximately on a monthly basis, with six prominent peak periods identified for distributing coupons to customers.

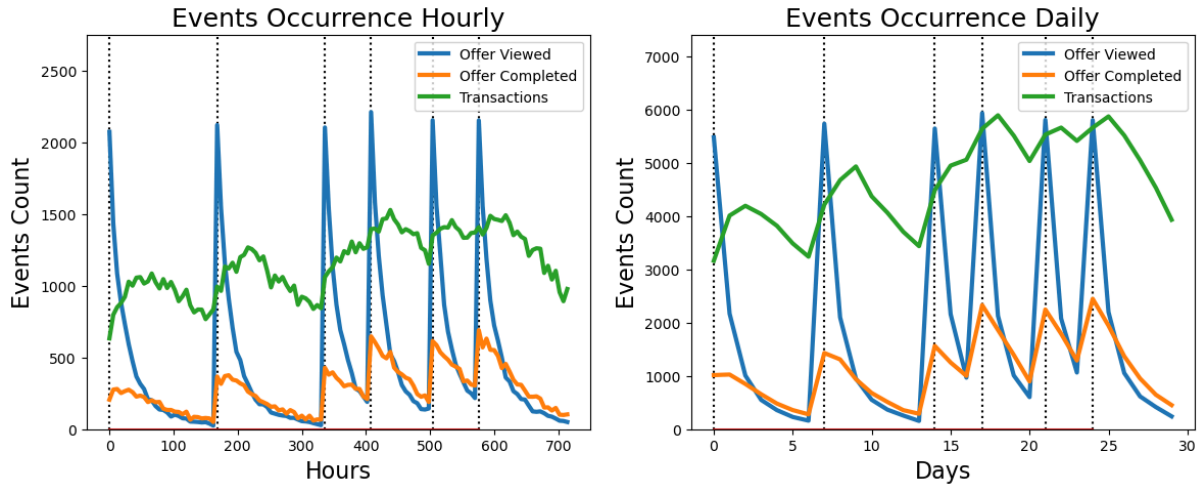



```
transcripts[transcripts['event'] == 'offer received'].groupby('hours_since_start').count()
```

32]:

hours_since_start	person	event	dict_key	dict_value	days_since_start
0	12650	12650	12650	12650	12650
168	12669	12669	12669	12669	12669
336	12711	12711	12711	12711	12711
408	12778	12778	12778	12778	12778
504	12704	12704	12704	12704	12704
576	12765	12765	12765	12765	12765

用groupby方式證實確實有六個高峰點收到優惠券，接下來繼續分析其他event的發生的時間點



Finding:

On the right-hand side, the line chart in terms of 'days' reveals a high correlation between Offer completion and offer viewing. With closer inspection using 'hours', a similar correlation is identified, albeit with a very short delay. This implies that the majority of consumers tend to **complete the offer redemption shortly after viewing the coupon, indicating that they use the coupon on the same day as viewing it.**

Transactions also exhibit a correlation with the other two events but with some dispersion. An interesting finding is that during each peak in viewed events, offer completion rapidly reaches its peak within a few hours, while transactions peak approximately 2 days later. This suggests that coupons not only directly contribute to immediate purchases with discounts or other benefits but also contribute to shaping customer purchasing habits (despite the shorter duration of individual offers).

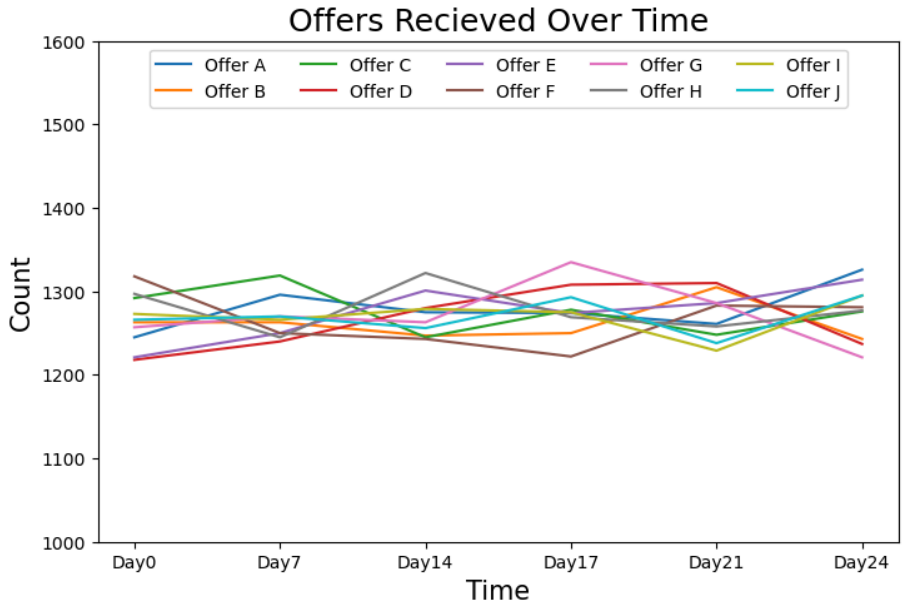
Insight:

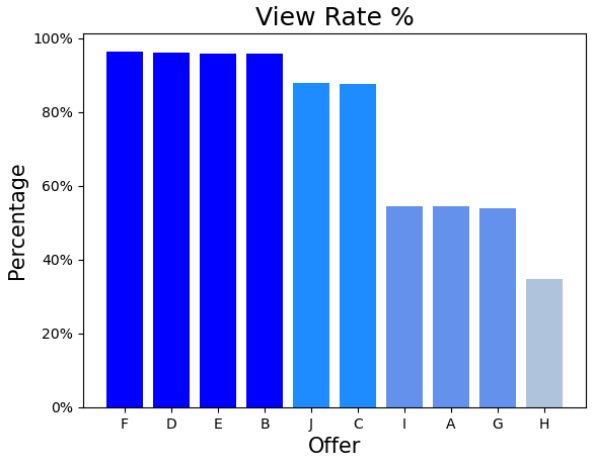
Coupons not only generate immediate revenue but also enhance customer loyalty. However, the effectiveness of coupon campaigns is transient. Therefore, Starbucks should strategically send coupons to customers at a certain frequency to help shape their buying behavior. If the frequency is too low, the cumulative cost for all customers may lead to significant marketing inefficiencies.

Second: Recommendation on coupon channel

I would like to explore the data using the view rate to assess the effectiveness of different promotional channels in terms of push notifications. Checking if all customers receive coupons in a certain order is crucial, as the sequence of receiving offers may impact the response. This can lead to variations in view rate and completion rate. From the chart, it can be observed that the quantity of each type of

coupon received at intervals 0, 7, 14, 17, 21, and 24 is roughly consistent, thereby avoiding bias.






```
offer_alias
F    0.964460
D    0.961148
E    0.959587
B    0.959451
J    0.877789
C    0.876991
I    0.544046
A    0.543311
G    0.539570
H    0.347287
Name: count, dtype: float64
```

From the chart, it can be observed that the view rates can be roughly divided into four groups. By grouping similar view rates together, it is evident that the same promotional channels are used for sending coupons within each group.

	group	reward	channels	difficulty	duration	offer_type	id	offer_alias
0	Group1	5	[web, email, mobile, social]	5	5	bogo	f19421c1d4aa40978ebb69ca19b0e20d	B
1	Group1	10	[web, email, mobile, social]	10	5	bogo	4d5c57ea9a6940dd891ad53e9dbe8da0	D
2	Group1	3	[web, email, mobile, social]	7	7	discount	2298d6c36e964ae4a3e7e9706d1fb8c2	E
3	Group1	2	[web, email, mobile, social]	10	10	discount	fafdcd668e3743c1bb461111dcafc2a4	F
4	Group2	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	C
5	Group2	0	[email, mobile, social]	0	3	informational	5a8bc65990b245e5a138643cd4eb9837	J
6	Gruop3	5	[web, email, mobile]	5	7	bogo	9b98b8c7a33c4b65b9aebfe6a799e6d9	A
7	Gruop3	2	[web, email, mobile]	10	7	discount	2906b810c7d4411798c6938adc9daaa5	G
8	Gruop3	0	[web, email, mobile]	0	4	informational	3f207df678b143eea3cee63160fa8bed	I
9	Group4	5	[web, email]	20	10	discount	0b1e1539f2cc45b7b9fa7c272da2e1d7	H

- Group 1 and Group 2 have a difference of less than 10%, indicating that web contributes minimally to the view rates.
- The view rate difference between Group 2 and Group 3 exceeds 30%, suggesting that social media significantly contributes to the view rates.
- Group 3 and Group 4 have a view rate difference close to 20%, indicating a substantial contribution from mobile devices to the view rates.

Examining Group 4 independently, it is noted that web contributes very little to the view rates. Therefore, the majority of coupons in Group 4 are viewed through email. Consequently, coupons received via email make a significant contribution to the view rates, slightly surpassing those viewed on mobile devices.

Insight :

Starbucks should consider **reducing the use of web-based** push notifications for coupons to decrease push notification costs.

Third: The factors that influence the usage of “Discount offer”

When observing the raw data, we found out that compared to the other two offer types, there are different minimum required spend for the “discount offer”. Therefore, at this stage, the primary objective we intend to observe is whether the

feature of ‘discount’ coupons will impact consumer purchasing behavior. Therefore, I will utilize two data sets: the 'transcript' and the 'portfolio'.

Since the original dataset is in JSON format, the first step involves preprocessing the data by converting it to CSV format. The 'portfolio' dataset consists of 10 types of coupons, while the 'transcript' includes all transaction records. Initially, in the preprocessing of the 'transcript', we separated 'offer_received' and 'offer_completed' to identify consumers who received and used the coupons, confirming whether the transaction was completed using the received coupon. Additionally, the coupon features are merged into the dataset.

The next step focuses on the 'discount' coupons by filtering for 'offer_type' equal to 1. This process results in a final dataset containing consumers who used 'discount' coupons, with different features and information on whether they completed transactions using the received coupons.

Using the Logits Regression in the statsmodels package, we assess whether the features of coupons, namely 'reward,' 'difficulty,' and 'duration,' affect consumer purchasing behavior. The results are as follows:

Logit Regression Results

Dep. Variable:	y	No. Observations:	36070
Model:	Logit	Df Residuals:	36066
Method:	MLE	Df Model:	3
Date:	Sat, 06 Jan 2024	Pseudo R-squ.:	0.03580
Time:	17:55:51	Log-Likelihood:	-21747.
converged:	True	LL-Null:	-22555.
Covariance Type:	non robust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	0.0960	0.076	1.270	0.204	-0.052	0.244
reward	0.1629	0.018	9.000	0.000	0.127	0.198
difficulty	-0.1598	0.005	-29.140	0.000	-0.171	-0.149
duration	0.2444	0.011	21.989	0.000	0.223	0.266

We can observe that all three features significantly influence consumer purchasing behavior. Specifically, 'reward' and 'duration' positively impact, while

'difficulty' has a negative effect. Therefore, increasing the rewards consumers receive and extending the duration of coupon usage can significantly enhance consumer willingness to purchase. Conversely, reducing the difficulty level also improves consumer purchasing behavior. It is evident that both 'reward' and 'duration' simultaneously have a positive impact on consumer purchasing behavior. Consequently, I would like to examine whether these two features exhibit interaction and whether simultaneous enhancement further strengthens consumer willingness to purchase. The results are as follows :

Logit Regression Results

Dep. Variable:	y	No. Observations:	36070
Model:	Logit	Df Residuals:	36066
Method:	MLE	Df Model:	3
Date:	Sat, 06 Jan 2024	Pseudo R-squ.:	0.03580
Time:	18:02:34	Log-Likelihood:	-21747.
converged:	True	LL-Null:	-22555.
Covariance Type:	non robust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-7.1818	0.293	-24.522	0.000	-7.756	-6.608
reward	3.0030	0.113	26.640	0.000	2.782	3.224
duration	0.9189	0.031	29.535	0.000	0.858	0.980
interaction	-0.3373	0.012	-29.140	0.000	-0.360	-0.315

Interaction refers to the interplay between 'reward' and 'duration'. It can be observed that 'reward' and 'duration' exhibit a negative interaction, and this interaction significantly affects consumer purchasing intent. This implies that when both 'reward' and 'duration' are simultaneously increased, it may lead to a decrease in users' willingness to purchase.

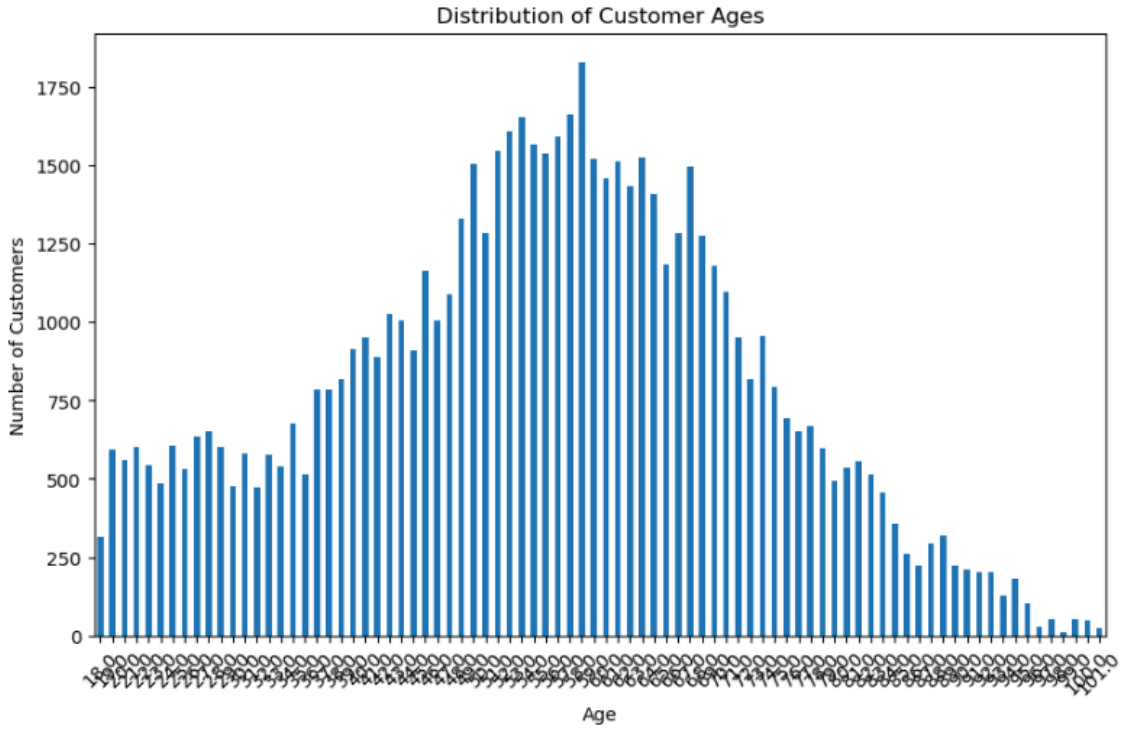
Insight :

When Starbucks is promoting 'discount' coupons, attention should be given to the coupon features. Specifically, increasing 'reward' and 'duration', and reducing difficulty can enhance consumer willingness to purchase, achieving promotional effectiveness. However, it is advisable to avoid increasing both 'reward' and 'duration'

simultaneously. Instead, selecting one of them would still significantly boost consumer purchasing behavior.

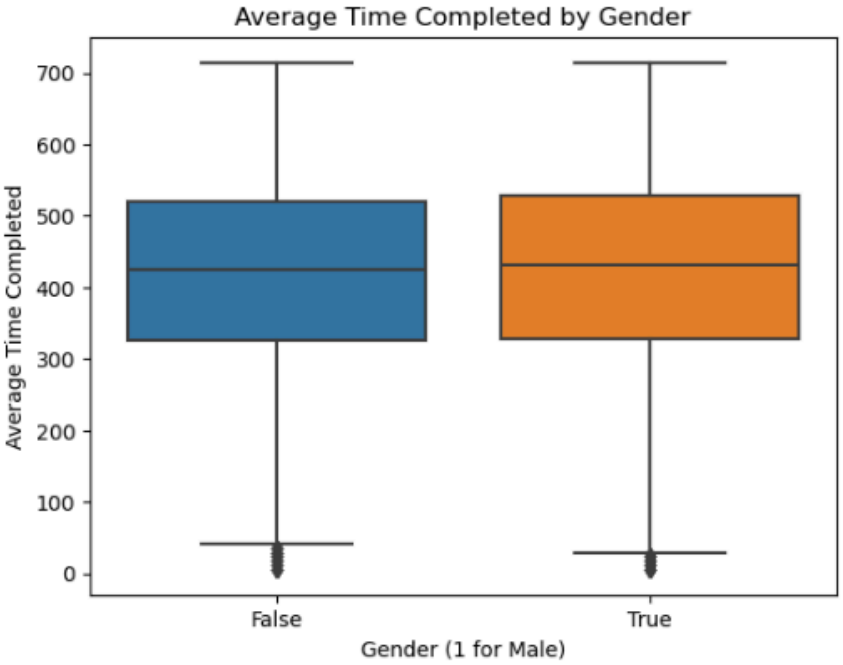
Fourth: The integrated analysis of Profile and Transcript

We would like to know the differences in the amount of time it takes for different ages (or genders) to complete a transaction after seeing the coupon. The initial hypothesis is that younger people (aged 18-40) have more time to use their mobile phones, so they check their phones and emails more frequently and are more familiar with app operations. Therefore, compared to older people, their completion time seems to be shorter.

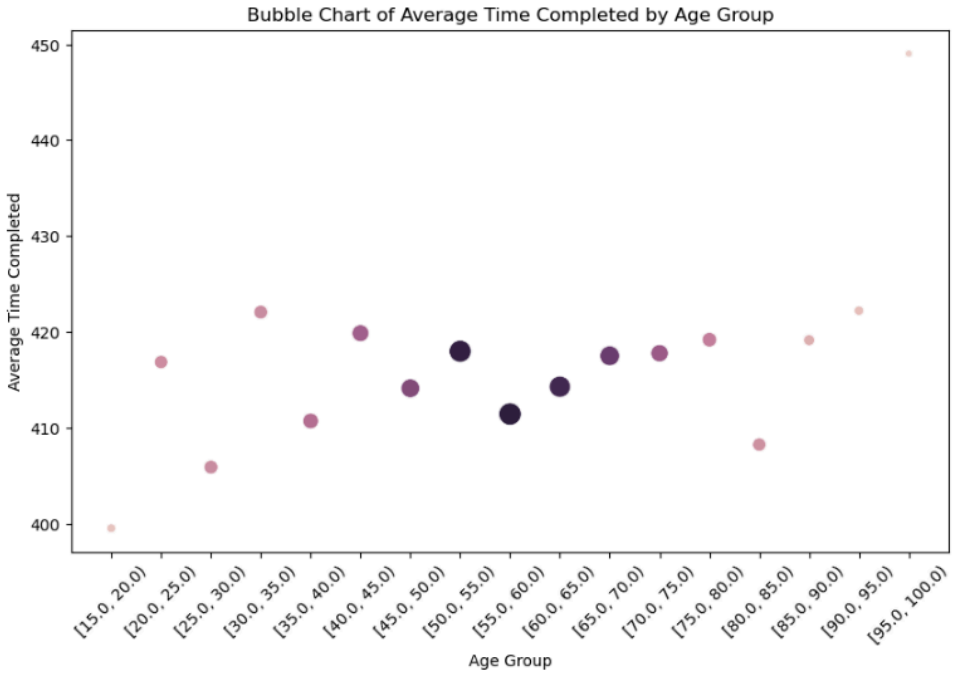


First, let's look at the age distribution of our customers above. We can see that the majority of our customers are between 35 to 70 years old, and the distribution chart shows a slight skew to the left. We reasonably speculate that middle-aged customers have more consumption power and are able to afford the

higher prices at Starbucks. Meanwhile, Starbucks is also popular among younger people.



Next, let's take a look at the boxplot above of average time completed by gender. Surprisingly, the quartiles for both genders are almost the same. We originally thought that females would have a shorter completion time during the discussion. It seems that in our dataset, the completion time for both men and women is quite evenly distributed.



Finally, we grouped all customers in 5-year age brackets to make the chart less messy and try to identify if there are any patterns in spending time. Above is the bubble chart of the average time completed by age group. We can see that as the age of the groups increases, the average time to complete a coupon becomes longer, especially in the intervals from 15-20 years to 45-50 years and from 50-55 years to 70-75 years, which matches with the initial hypothesis.

5. Machine Learning and Statistic

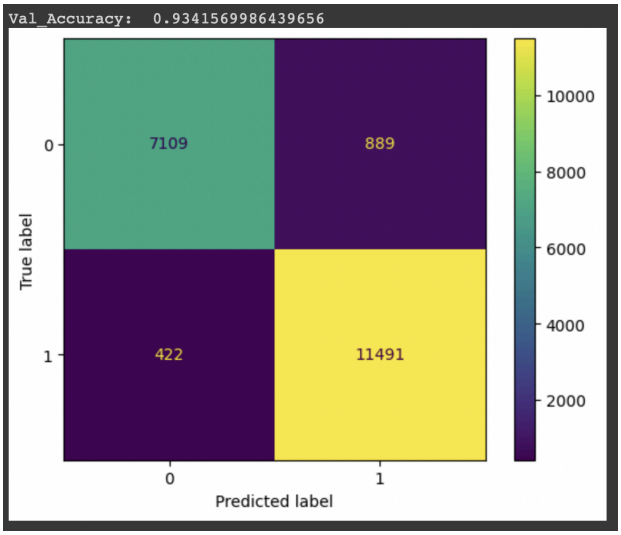
In this part, the goal is to maximize the usability of the given offer. Since not everyone will view the offer after receiving the offer, and not everyone will use the offer after viewing the offer. We want to focus on the groups who have the higher chance to view the offer and further use the offer. Therefore, we don't care much about groups that don't even bother viewing the offer, and we will not be sending them an offer. The target customers will be those who will view the offer. So the problem is formulated as follows: given the fact that each customer has already viewed the offer, whether they will respond to the offer or not. So basically, it is a binary classification problem.

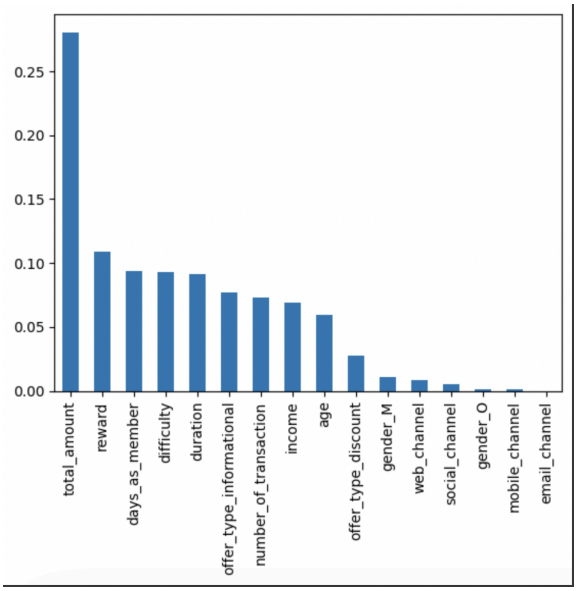
Since the first and second columns of the cleaned data frame are 'customer_id' and 'offer_id', which are not the features we are inputting into the model, we drop them. We then extract the 'customer_response', which is encoded as one and zero, as y-label, and extract the remaining data frame as X-label. The X.shape contains 66367 observations x 16 features, and the y.shape contains, certainly, 66367 corresponding y-labels. We then split the input X into 3:7, three out of ten as the validation set, and seven out of ten as the training set. And it results in

X_train.shape equals (46456, 16), X_val.shape equals (19911, 16), y_train.shape equals (46456,), and y_val.shape equals (19911,).

Random Forest

We then construct a random forest classifier to predict if a customer has viewed the offer, and whether he will respond to the offer or not. The result is quite good for the default parameter setting on random forest, which yields 93% accuracy. We are then interested in which features contribute to the result the most. It comes out 'total_amount' (The money a customer spent), 'reward' (The reward/payback a customer will receive after spending an amount of money), 'days_as_member'(how long a customer remains a member) are the top three most important features.

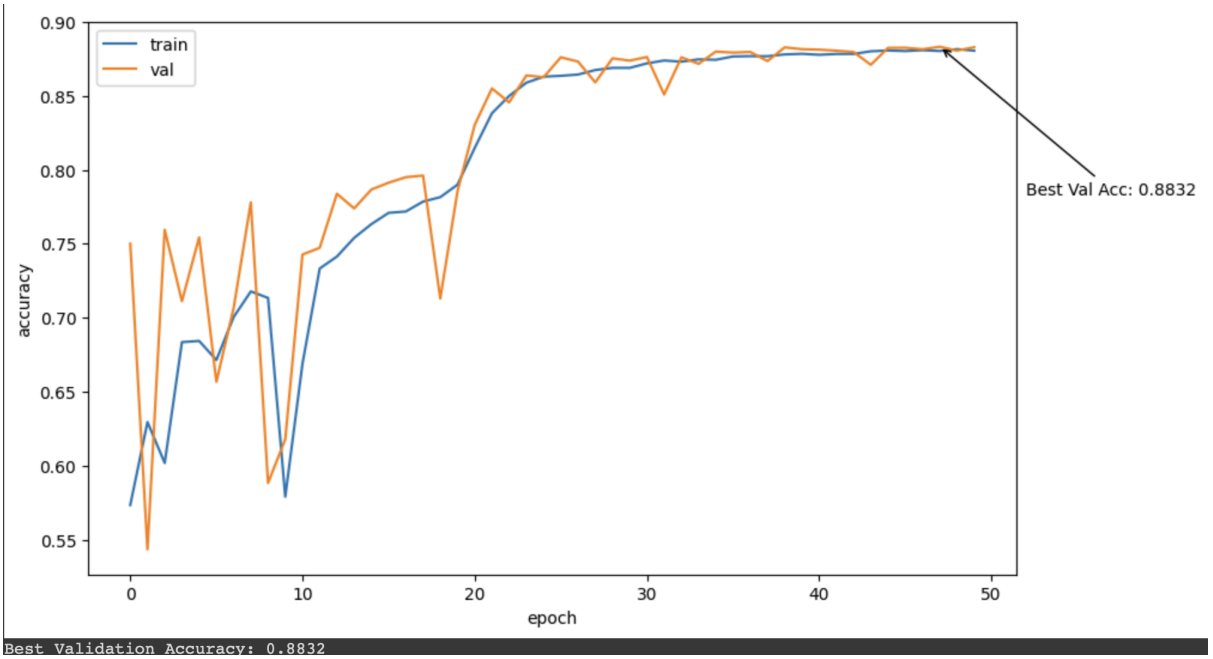




DNN(Deep Neural Network)

Finally, we are interested in neural network performance, so we simply construct an MLP with five layers, including input and output layers. The implementation details can be seen in the code provided. Last but not least, it yields

88% accuracy, which is pretty good too. However, it's still a slight step away compared to the random forest approach, which yields about 93%.



6. Summary

Based on the analysis above, we can conclude that:

In terms of coupons, the time between customers receiving coupons and viewing them is extremely swift, which means the life cycle of this marketing campaign is short. Accordingly, Starbucks will need to consider increasing the frequency of sending coupons to members. In addition, the view rate of coupons that are sent via the web is the lowest among all, thus we recommend they cut down on the amount of promotion through this channel. Once we dig deeper into the Discount offer type, we suggest that Starbucks increase either the duration or the reward of the coupons.

Moreover, the completion time for the younger age group of 20-25 years is longer than expected. Perhaps we could design interactive coupons for them, such as those including game or challenge elements, to appeal to young customers. For

middle-aged and older consumers, we could provide simple, easy-to-understand, and one-click coupons to lower the barrier to use and enhance their willingness to purchase.

As for this dataset itself, we reckon that it would be better if there were more details about the customer purchasing preference, such as the products they had bought or the price they spent on each order. With those data, we can analyze customer behavior further with customers using the coupons, and is possible to launch marketing campaigns on specific bestseller products or know customers' favor.

7. Reference

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

8. Job Distribution

鄭群霓	**insight 1, 2**
葉兆泉	**Insight 3**
黃念祺	**insight 4**
謝東睿	**data process, Machine Learning**
黃則維	**data process, Paperwork**