UDACITY

DATA SCIENTIST NANODEGREE
FINAL PROJECT
DATA SCIENCE CAPSTONE
STARBUCKS CASE

DECEMBER 31, 2023

Tran Huu Nhat Huy

Japan

Contents

I. Introduction

The Starbucks Capstone Project is a key component of the Udacity Data Science Nanodegree program, providing students with a hands-on opportunity to apply their skills and knowledge gained throughout the program to a real-world business challenge. This project is developed in collaboration with Starbucks, a global coffeehouse chain where the author enjoys sipping their matcha latte every day, making it an engaging and relevant experience for aspiring data scientists such as the author.

Starbucks, a renowned brand in the food and beverage industry, has provided simulated data that mimics customer behavior on their rewards mobile app. The dataset spans diverse aspects of customer interactions, including offers received, viewed, and completed, as well as transaction details. The primary objective of the capstone project is to analyze this dataset and develop a machine learning model that predicts whether a customer will respond to a particular offer.

II. Methodology

1. Datasets

This data set contains simulated data that mimics customer behavior on the Starbucks rewards mobile app. Once every few days, Starbucks sends out an offer to users of the mobile app. An offer can be merely an advertisement for a drink or an actual offer such as a discount or BOGO (buy one get one free). Some users might not receive any offer during certain weeks. Not all users receive the same offer, and that is the challenge to solve with this data set.

Your task is to combine transaction, demographic and offer data to determine which demographic groups respond best to which offer type. This data set is a simplified version of the real Starbucks app because the underlying simulator only has one product whereas Starbucks actually sells dozens of products. Every offer has a validity period before the offer expires.

As an example, a BOGO offer might be valid for only 5 days. You will see in the data set that informational offers have a validity period even though these ads are merely providing information about a product; for example, if an informational offer has 7 days of validity, you can assume the customer is feeling the influence of the offer for 7 days after receiving the advertisement. You will be given transactional data showing user purchases made on the app including the timestamp of purchase and the amount of money spent on a purchase. This transactional data also has a record for each offer that a user receives as well as a record for when a user actually views the offer. There are also records for when a user completes an offer.

Regarding metadata, the data consists of 3 datasets in 3 JSON 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 is BOGO, discount, informational.
    - difficulty (int) - minimum required spend to complete an offer.
    - reward (int) - reward given for completing an offer.
    - duration (int) - time for offer to be open, in days.
    - 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 (transaction, offer received/viewed, etc.)
    - person (str) - customer id
    - time (int) - time in hours since start of test. The data begins at time t=0
    - value - (dict of strings) - either an offer id or transaction amount depending on the record

2. Strategies & targets

1. Data exploratory & cleaning to further understand the nature of datasets & patterns.
2. Try to find the data distribution among customers (Profiles), the offers (Portfolio), and the events (Transcripts).
3. Try to find correlation between customer's behaviors and offers, based on gender & age.
4. Try to find the most successful offers, what are they and how they do.
5. Try to predict the successful offers.

3. Data acquisition and initial exploratory

In this session, we read the 3 JSON files. For each file, extract DataFrame, and then have a brief look at data distribution. Then, we simply do ordinary data cleanings like NaN analysis and process, data distribution assessment, dropping duplicates, etc.

4. Data wrangling

In this session, we try to further manipulate the data by converting data types, transforming data formats/structures, and see if we can possibly merge the 3 datasets together.

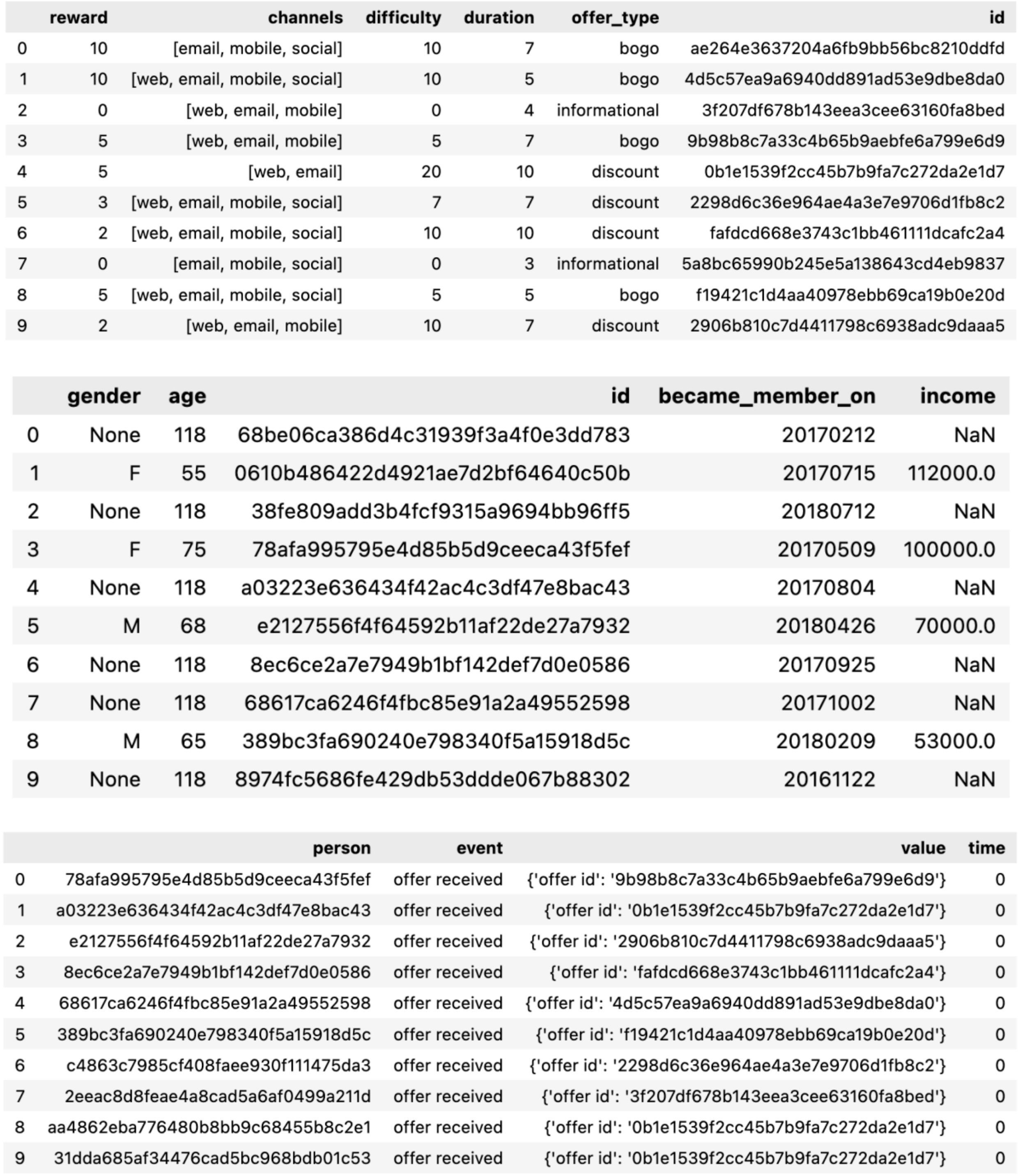
5. Model training with evaluation and improvement

Finally, we try to predict the successful offer cases, based on other features, all of these are one-hot encoded. Next, we try to further improve the model's performance.

III. Results and Discussions

1. Data acquisition and initial exploratory

In this session, we read the 3 JSON files. For each file, extract DataFrame, and then have a brief look at data distribution.



	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

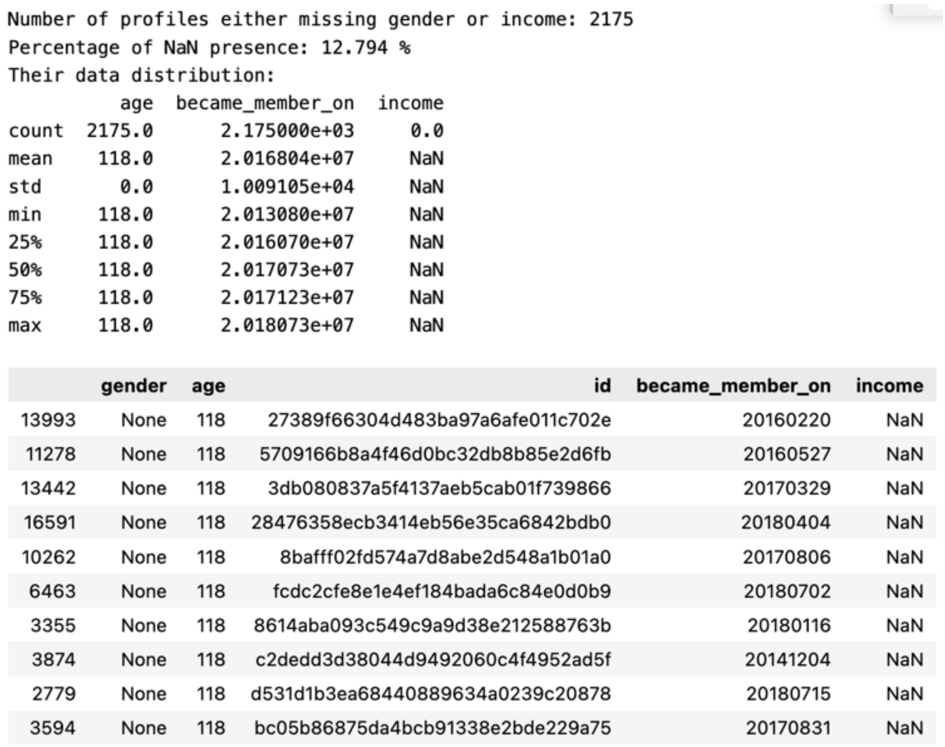
	gender	age	id	became_member_on	income
0	None	118	68be06ca386d4c31939f3a4f0e3dd783	20170212	NaN
1	F	55	0610b486422d4921ae7d2bf64640c50b	20170715	112000.0
2	None	118	38fe809add3b4fcf9315a9694bb96ff5	20180712	NaN
3	F	75	78afa995795e4d85b5d9ceeca43f5fef	20170509	100000.0
4	None	118	a03223e636434f42ac4c3df47e8bac43	20170804	NaN
5	M	68	e2127556f4f64592b11af22de27a7932	20180426	70000.0
6	None	118	8ec6ce2a7e7949b1bf142def7d0e0586	20170925	NaN
7	None	118	68617ca6246f4fbc85e91a2a49552598	20171002	NaN
8	M	65	389bc3fa690240e798340f5a15918d5c	20180209	53000.0
9	None	118	8974fc5686fe429db53ddde067b88302	20161122	NaN

	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
5	389bc3fa690240e798340f5a15918d5c	offer received	{'offer id': 'f19421c1d4aa40978ebb69ca19b0e20d'}	0
6	c4863c7985cf408faee930f111475da3	offer received	{'offer id': '2298d6c36e964ae4a3e7e9706d1fb8c2'}	0
7	2eeac8d8feae4a8cad5a6af0499a211d	offer received	{'offer id': '3f207df678b143eea3cee63160fa8bed'}	0
8	aa4862eba776480b8bb9c68455b8c2e1	offer received	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}	0
9	31dda685af34476cad5bc968bdb01c53	offer received	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}	0

Fig. 1. Head samples of Portfolio, Profile, Transcript respectively.

From the above session, we can see that among 3 data sources, only Profile has null data in 2 out of 5 columns, which are:

- gender
- income



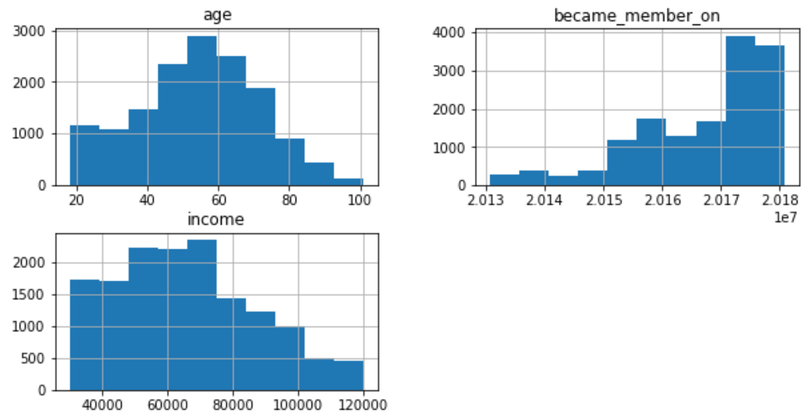
```
Number of profiles either missing gender or income: 2175
Percentage of NaN presence: 12.794 %
Their data distribution:
          age  became_member_on  income
count  2175.0      2.175000e+03     0.0
mean    118.0      2.016804e+07     NaN
std       0.0      1.009105e+04     NaN
min     118.0      2.013080e+07     NaN
25%     118.0      2.016070e+07     NaN
50%     118.0      2.017073e+07     NaN
75%     118.0      2.017123e+07     NaN
max     118.0      2.018073e+07     NaN
```

	gender	age	id	became_member_on	income
13993	None	118	27389f66304d483ba97a6afe011c702e	20160220	NaN
11278	None	118	5709166b8a4f46d0bc32db8b85e2d6fb	20160527	NaN
13442	None	118	3db080837a5f4137aeb5cab01f739866	20170329	NaN
16591	None	118	28476358ecb3414eb56e35ca6842bdb0	20180404	NaN
10262	None	118	8bafff02fd574a7d8abe2d548a1b01a0	20170806	NaN
6463	None	118	fcdc2cfe8e1e4ef184bada6c84e0d0b9	20180702	NaN
3355	None	118	8614aba093c549c9a9d38e212588763b	20180116	NaN
3874	None	118	c2dedd3d38044d9492060c4f4952ad5f	20141204	NaN
2779	None	118	d531d1b3ea68440889634a0239c20878	20180715	NaN
3594	None	118	bc05b86875da4bcb91338e2bde229a75	20170831	NaN

Fig. 2. Data distribution and 10 samples of missing values in Profile.

Upon taking a deeper look, from the data distribution above, we can see these records are totally nonsense, with the same very high age of 118, and different membership starting dates, thus we can conclude all of them are indeed data errors. They take 12.8% of the data, although not very small, but still an acceptable amount to drop. Thus, we can drop them, and so did we.

After NaNs are dealt with, we examine the data distribution and composition of these 3 tables.



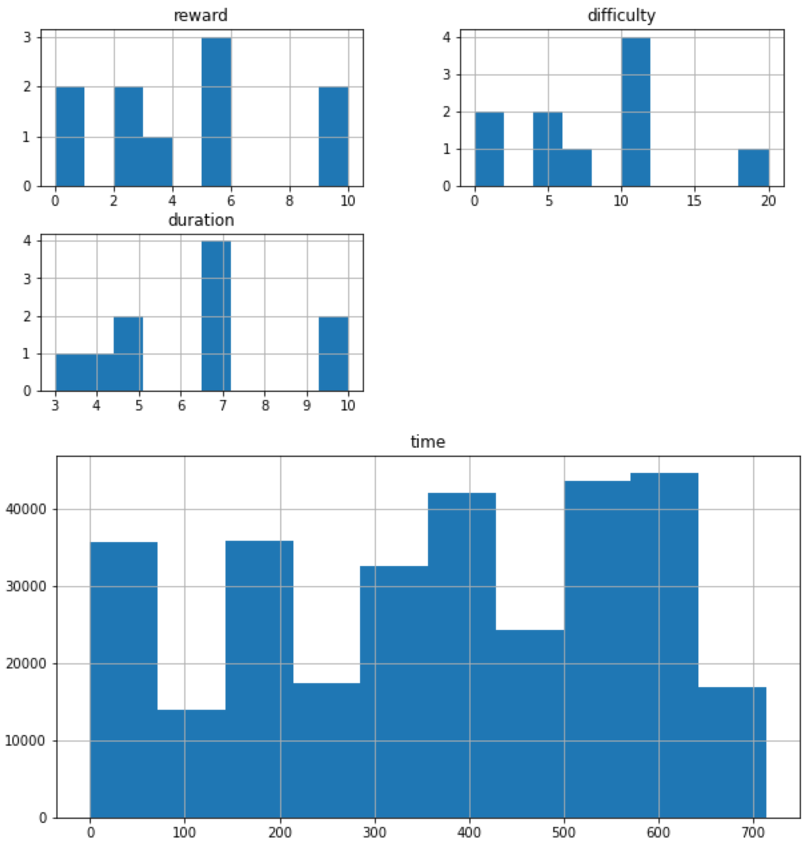


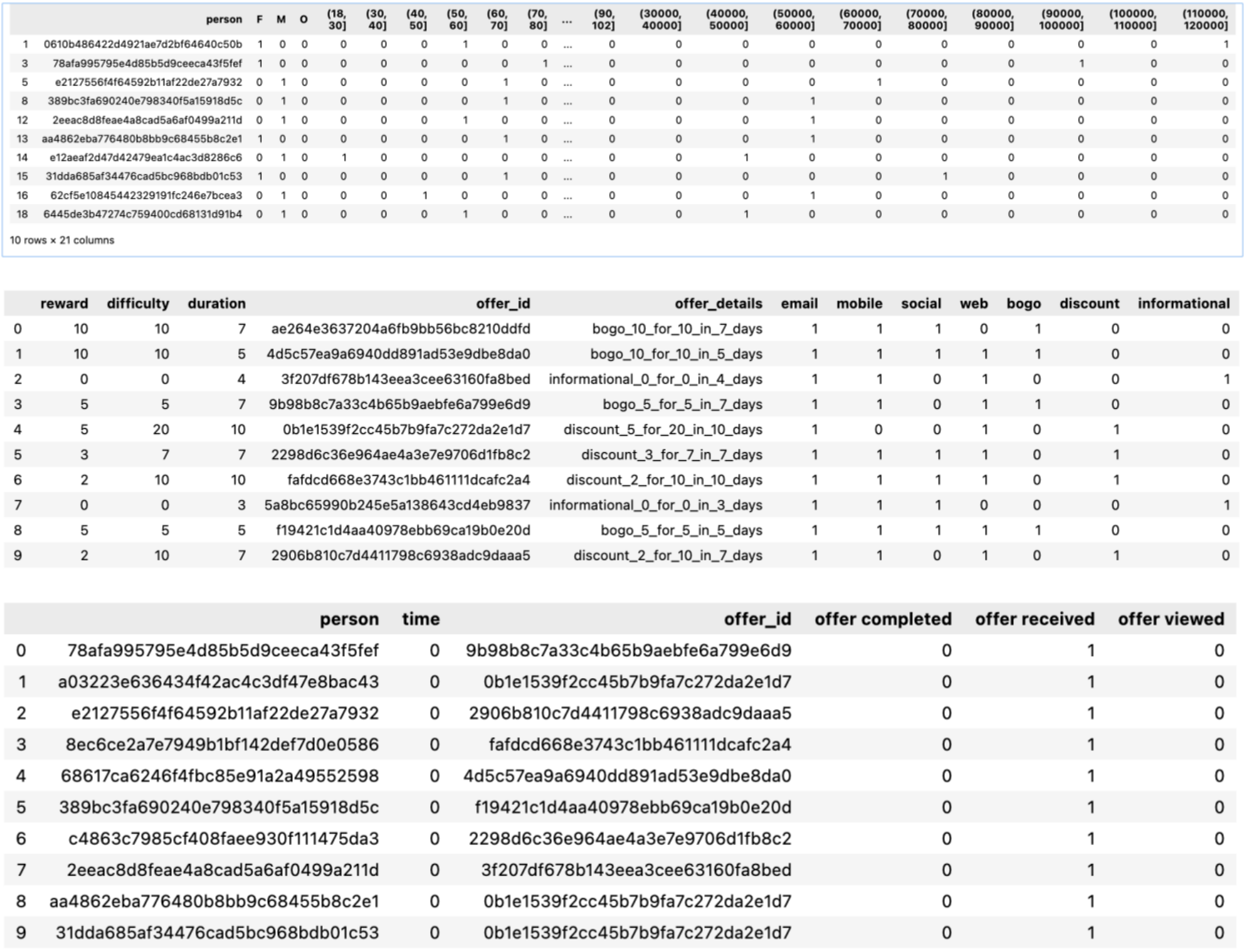
Fig. 3. Data distribution of numerical columns in Profile, Portfolio, and Transcript respectively.

From this, we can acquire several initial understandings:

- Profile
  - became_member_on should be DateTime. But on second thought, since this is the ONLY DateTime column in our whole datasets, and we do not have any other reference timestamp, so this column is practically useless and can be dropped.
  - gender should be somewhat categorical one-hot encoded.
- Portfolio
  - channels should be categorical one-hot encoded.
  - offer_type should be one-hot.
- Transcript
  - event should be one-hot.
  - value : extract the offer_id inside and turn that column into offer_id.

These can be address in data cleaning session.

2. Data wrangling



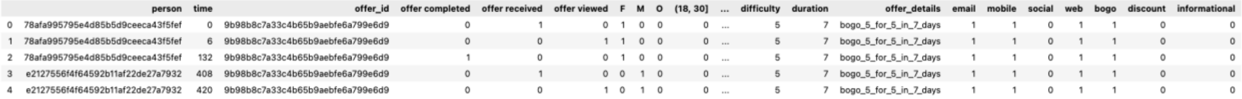
	person	F	M	O	(18, 30]	(30, 40]	(40, 50]	(50, 60]	(60, 70]	(70, 80]	...	(90, 102]	(30000, 40000]	(40000, 50000]	(50000, 60000]	(60000, 70000]	(70000, 80000]	(80000, 90000]	(90000, 100000]	(100000, 110000]	(110000, 120000]
1	0610b486422d4921ae7d2bf64640c50b	1	0	0	0	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0	1
3	78afa995795e4d85b5d9ceeca43f5fef	1	0	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0	1	0	0
5	e2127556f4f64592b11af22de27a7932	0	1	0	0	0	0	0	1	0	...	0	0	0	0	1	0	0	0	0	0
8	389bc3fa690240e798340f5a15918d5c	0	1	0	0	0	0	0	1	0	...	0	0	0	1	0	0	0	0	0	0
12	2eeac8d8feae4a8cad5a6af0499a211d	0	1	0	0	0	0	1	0	0	...	0	0	0	1	0	0	0	0	0	0
13	aa4862eba776480b8bb9c68455b8c2e1	1	0	0	0	0	0	0	1	0	...	0	0	0	1	0	0	0	0	0	0
14	e12aeaf2d47d42479ea1c4ac3d8286c6	0	1	0	1	0	0	0	0	0	...	0	0	1	0	0	0	0	0	0	0
15	31dda685af34476cad5bc968bdb01c53	1	0	0	0	0	0	0	1	0	...	0	0	0	0	0	1	0	0	0	0
16	62cf5e10845442329191fc246e7bcea3	0	1	0	0	0	1	0	0	0	...	0	0	0	1	0	0	0	0	0	0
18	6445de3b47274c759400cd68131d91b4	0	1	0	0	0	0	1	0	0	...	0	0	1	0	0	0	0	0	0	0

10 rows × 21 columns

	reward	difficulty	duration	offer_id	offer_details	email	mobile	social	web	bogo	discount	informational
0	10	10	7	ae264e3637204a6fb9bb56bc8210ddfd	bogo_10_for_10_in_7_days	1	1	1	0	1	0	0
1	10	10	5	4d5c57ea9a6940dd891ad53e9dbe8da0	bogo_10_for_10_in_5_days	1	1	1	1	1	0	0
2	0	0	4	3f207df678b143eea3cee63160fa8bed	informational_0_for_0_in_4_days	1	1	0	1	0	0	1
3	5	5	7	9b98b8c7a33c4b65b9aebfe6a799e6d9	bogo_5_for_5_in_7_days	1	1	0	1	1	0	0
4	5	20	10	0b1e1539f2cc45b7b9fa7c272da2e1d7	discount_5_for_20_in_10_days	1	0	0	1	0	1	0
5	3	7	7	2298d6c36e964ae4a3e7e9706d1fb8c2	discount_3_for_7_in_7_days	1	1	1	1	0	1	0
6	2	10	10	fafdcd668e3743c1bb461111dcafc2a4	discount_2_for_10_in_10_days	1	1	1	1	0	1	0
7	0	0	3	5a8bc65990b245e5a138643cd4eb9837	informational_0_for_0_in_3_days	1	1	1	0	0	0	1
8	5	5	5	f19421c1d4aa40978ebb69ca19b0e20d	bogo_5_for_5_in_5_days	1	1	1	1	1	0	0
9	2	10	7	2906b810c7d4411798c6938adc9daaa5	discount_2_for_10_in_7_days	1	1	0	1	0	1	0

	person	time	offer_id	offer completed	offer received	offer viewed
0	78afa995795e4d85b5d9ceeca43f5fef	0	9b98b8c7a33c4b65b9aebfe6a799e6d9	0	1	0
1	a03223e636434f42ac4c3df47e8bac43	0	0b1e1539f2cc45b7b9fa7c272da2e1d7	0	1	0
2	e2127556f4f64592b11af22de27a7932	0	2906b810c7d4411798c6938adc9daaa5	0	1	0
3	8ec6ce2a7e7949b1bf142def7d0e0586	0	fafdcd668e3743c1bb461111dcafc2a4	0	1	0
4	68617ca6246f4fbc85e91a2a49552598	0	4d5c57ea9a6940dd891ad53e9dbe8da0	0	1	0
5	389bc3fa690240e798340f5a15918d5c	0	f19421c1d4aa40978ebb69ca19b0e20d	0	1	0
6	c4863c7985cf408faee930f111475da3	0	2298d6c36e964ae4a3e7e9706d1fb8c2	0	1	0
7	2eeac8d8feae4a8cad5a6af0499a211d	0	3f207df678b143eea3cee63160fa8bed	0	1	0
8	aa4862eba776480b8bb9c68455b8c2e1	0	0b1e1539f2cc45b7b9fa7c272da2e1d7	0	1	0
9	31dda685af34476cad5bc968bdb01c53	0	0b1e1539f2cc45b7b9fa7c272da2e1d7	0	1	0

Fig. 4. Profile, Portfolio, Transcript after data wrangling.



	person	time	offer_id	offer completed	offer received	offer viewed	F	M	O	(18, 30]	...	difficulty	duration	offer_details	email	mobile	social	web	bogo	discount	informational
0	78afa995795e4d85b5d9ceeca43f5fef	0	9b98b8c7a33c4b65b9aebfe6a799e6d9	0	1	0	1	0	0	0	...	5	7	bogo_5_for_5_in_7_days	1	1	0	1	1	0	0
1	78afa995795e4d85b5d9ceeca43f5fef	6	9b98b8c7a33c4b65b9aebfe6a799e6d9	0	0	1	1	0	0	0	...	5	7	bogo_5_for_5_in_7_days	1	1	0	1	1	0	0
2	78afa995795e4d85b5d9ceeca43f5fef	132	9b98b8c7a33c4b65b9aebfe6a799e6d9	1	0	0	1	0	0	0	...	5	7	bogo_5_for_5_in_7_days	1	1	0	1	1	0	0
3	e2127556f4f64592b11af22de27a7932	408	9b98b8c7a33c4b65b9aebfe6a799e6d9	0	1	0	0	1	0	0	...	5	7	bogo_5_for_5_in_7_days	1	1	0	1	1	0	0
4	e2127556f4f64592b11af22de27a7932	420	9b98b8c7a33c4b65b9aebfe6a799e6d9	0	0	1	0	1	0	0	...	5	7	bogo_5_for_5_in_7_days	1	1	0	1	1	0	0

Fig. 5. 5 head samples of merged table from 3 original tables.

3. Further exploratory data analysis

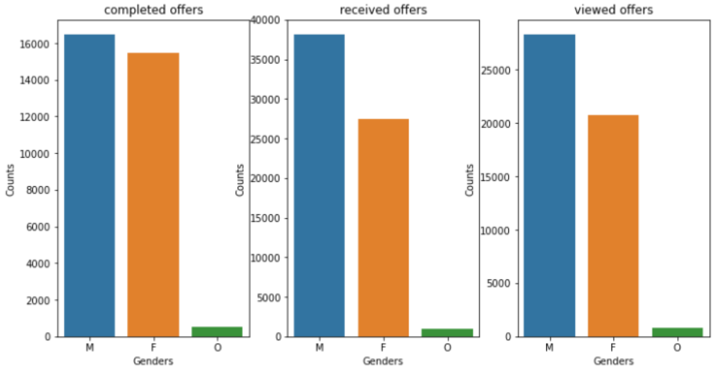


Fig. 6. Offer status, by genders.

We can see a general trend where males view, receive and complete more offers than females (we don't really care about Other since they are close to non-existent in the data). One interesting insight is females have higher offer completion over received & view than those in males.

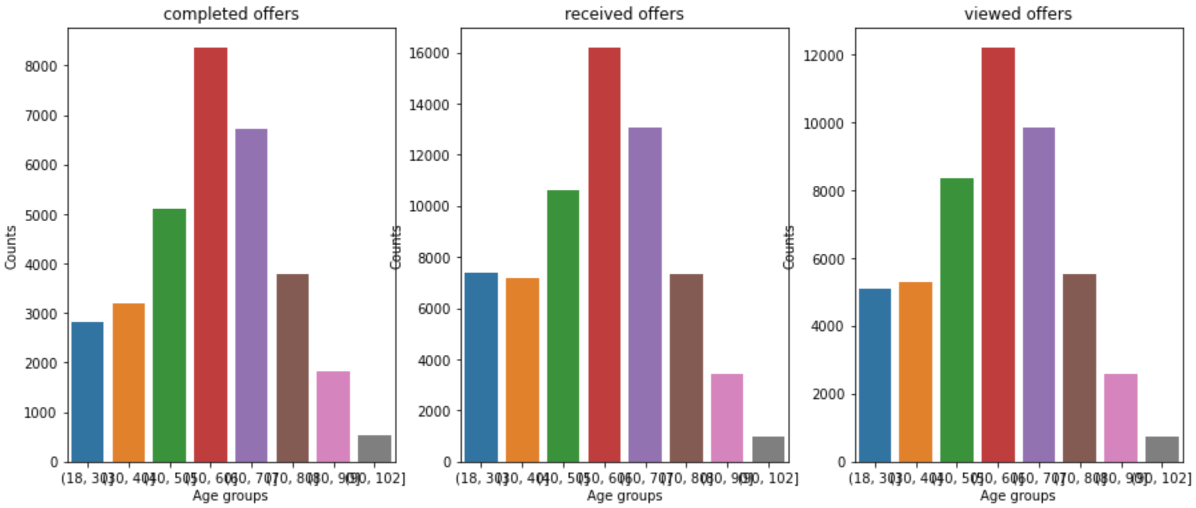


Fig. 7. Offer status, by age groups.

Although different scales, we can see a very similar distribution patterns between age groups in a Gaussian-like distribution with 50-60 age group taking dominance (like honestly, this data was taken in Japan or something? I'm sitting at Starbucks right now and a lot of elder people around me, just a normal day in Japan amidst the aging population).

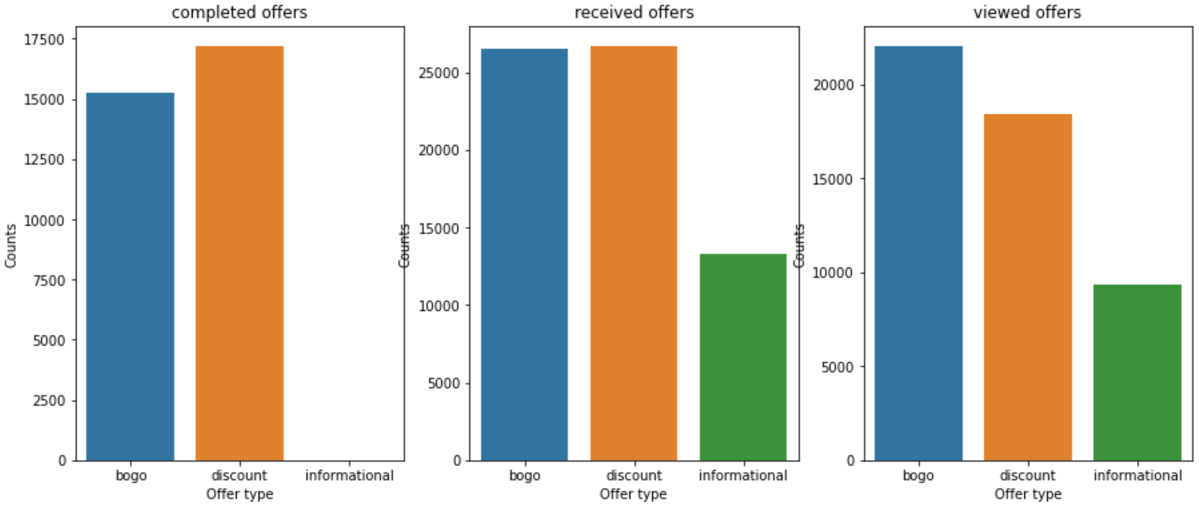


Fig. 8. Offer status, by age groups.

Looks like although BOGO and Discount share the same received counts, Discount offer type is more successful than BOGO, kinda understandable since people tend to buy a cup of coffee at a discounted price rather than 2 cups of coffee at the price of one, since most of the time we don't have such company to drink along.

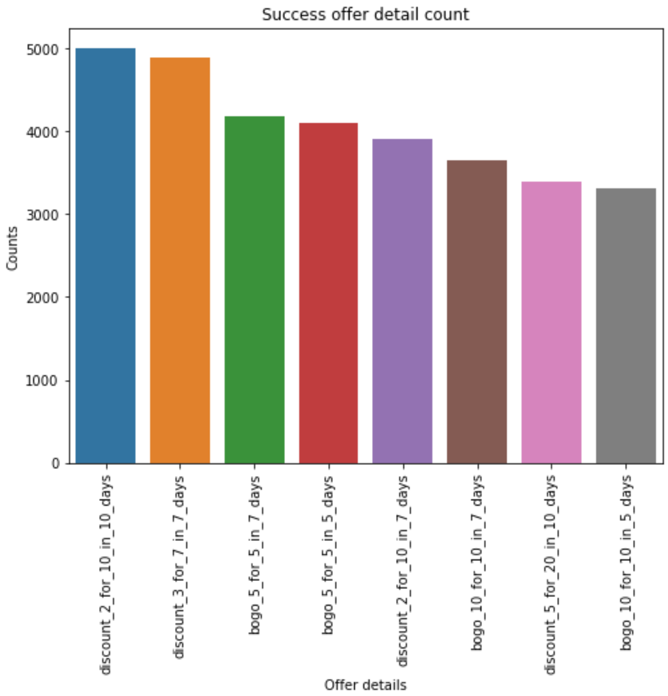


Fig. 9. Success offers by details, with the most 2 prominent ones being Discount.

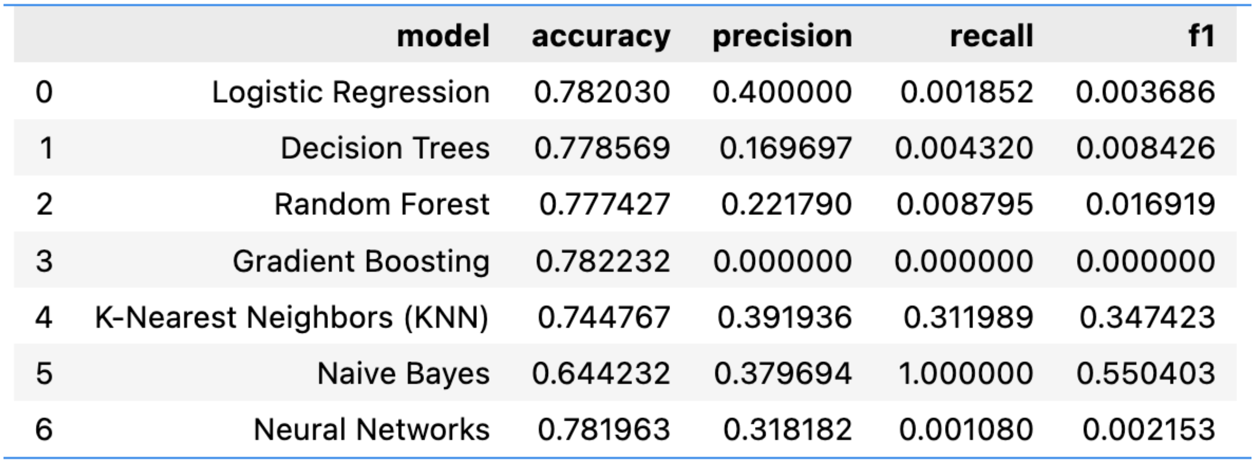
4. Model building

We have a bunch of one-hot encoded features, all X and Y are either 1 and 0. Y is Offer Accepted, a scalar, and X comprises of over 30 features remaining. After splitting into X/Y train/test sets with test ratio 0.2, I will use a bunch of models, each has default simple, rudimentary hyperparameter settings, to test which model performs to which extent, which we can choose one and bring it to the hyperparameter tuning round.

List of models:

- Logistic Regression
- Decision Trees
- Random Forest
- Gradient Boosting
- K-Nearest Neighbors (KNN)
- Naive Bayes
- Neural Networks

After training of all models, this is the result:



	model	accuracy	precision	recall	f1
0	Logistic Regression	0.782030	0.400000	0.001852	0.003686
1	Decision Trees	0.778569	0.169697	0.004320	0.008426
2	Random Forest	0.777427	0.221790	0.008795	0.016919
3	Gradient Boosting	0.782232	0.000000	0.000000	0.000000
4	K-Nearest Neighbors (KNN)	0.744767	0.391936	0.311989	0.347423
5	Naive Bayes	0.644232	0.379694	1.000000	0.550403
6	Neural Networks	0.781963	0.318182	0.001080	0.002153

Fig. 10. Results of 7 models.

The models seem to have challenges in effectively identifying positive instances (low recall), which may be influenced by the significant class imbalance. Naive Bayes shows high recall but sacrifices precision, possibly due to the imbalance in class distribution. The F1 Score provides a balanced measure, highlighting the trade-off between precision and recall.

In much greater details for each model:

- Logistic Regression: the model has relatively high accuracy but low precision and recall. The precision indicates that when the model predicts the positive class, it is correct 40% of the time. However, the recall is very low, suggesting that the model misses a significant number of positive instances.
- Decision Trees: similar to Logistic Regression, the Decision Trees model has low precision and recall. It seems to struggle in identifying positive instances, as reflected by the low recall.
- Random Forest: the Random Forest model shows improvement over Decision Trees in terms of precision and recall. However, the recall is still quite low, indicating that there's room for improvement in capturing positive instances.
- Gradient Boosting: the model appears to have issues with both precision and recall, resulting in an F1 Score of 0. This suggests that the model struggles to correctly classify positive instances.
- K-Nearest Neighbors (KNN): KNN shows a decent balance between precision and recall. The F1 Score suggests a more balanced performance compared to the earlier models, though there is still room for improvement.
- Naive Bayes: the model has high recall, but this comes at the cost of precision, as reflected in the F1 Score. The model seems to be biased toward identifying positive instances, possibly due to the class imbalance.
- Neural Networks: the model has high accuracy but, similar to Logistic Regression and Decision Trees, struggles with both precision and recall.

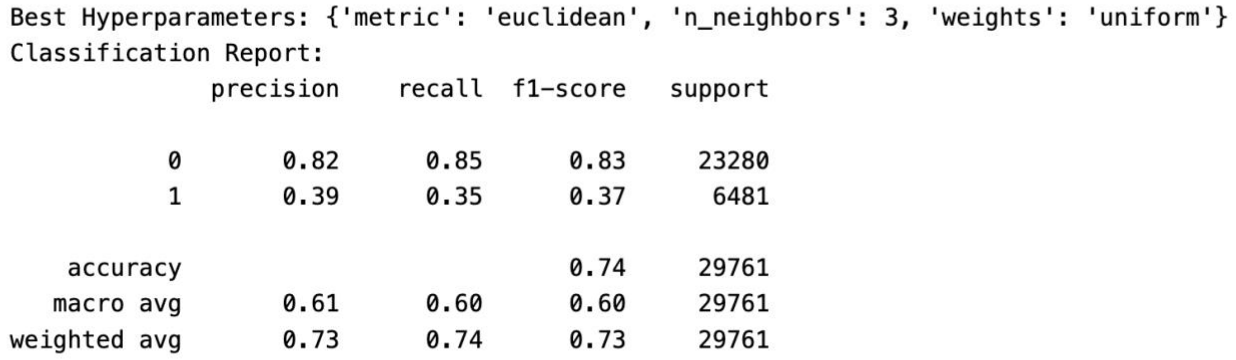
As I found out above that negative records significantly outnumber positive records with a 4:1 ratio which is extremely imbalance, it is crucial to consider the balance between precision and recall. Precision and recall are often in tension with each other - increasing one typically leads to a decrease in the other (precision-recall trade-off). Achieving a balance between precision and recall is important to ensure that the model is making predictions that are both accurate (high precision) and comprehensive (high recall).

Thus, out of 7 models with very rudimentary hyperparameter settings, I decided to choose KNN, with most balanced precision-recall, high accuracy, and an acceptable F1 score.

5. Further improve KNN

In this session, I aim to further improve the KNN model with better hyperparameters, using GridSearchCV for hyperparameter tuning.

```
param_grid = {
  "n_neighbors": [3, 5, 7],
  "weights": ["uniform"],
  "metric": ["euclidean"]
}
```



```
Best Hyperparameters: {'metric': 'euclidean', 'n_neighbors': 3, 'weights': 'uniform'}
Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.85      0.83     23280
           1       0.39      0.35      0.37      6481

    accuracy                           0.74     29761
   macro avg       0.61      0.60      0.60     29761
weighted avg       0.73      0.74      0.73     29761
```

Fig. 11. Result of the tuning and evaluation.

We can see that after tuning, our KNN model with the specified hyperparameters shows promise, notably at increased precision-recall balance and F1-score for class 0. However, there are also areas for improvement, particularly in capturing positive instances of class 1. Fine-tuning the model, adjusting the decision threshold, and considering other modeling approaches can potentially enhance its effectiveness.

IV. Conclusions

In summary, the analysis suggests that while some models have high accuracy, there is room for improvement in identifying positive instances, especially in the context of imbalanced datasets. Consideration of precision, recall, and F1 Score provides insights into the trade-offs made by each model. Of course, there are still rooms for improvements:

- Addressing the class imbalance and potentially adjusting the decision threshold may improve the models' performance.
- Further hyperparameter tuning and feature engineering could be explored to enhance model effectiveness.
- Evaluation using precision-recall curves and AUC-PR can provide a more detailed understanding of model performance across different decision thresholds.

Also, during data cleaning and exploring for the model, I have found several key insights:

- Regarding coffee consumption, people in middle ages like 50 to 60 drink most coffee compared to other age groups, indicating high level of coffee consumption of this age.
- Income shows a lot of people with low or average income going to Starbucks. People with higher incomes, especially those six-figure guys seem to not really enjoying Starbucks coffee compared to the lower income groups.
- Regarding offer reception, we can see a general trend where males view, receive and complete more offers than females and others. On the other hand, females have higher ratio of offer completion over received & view than those in males.
- Although BOGO and Discount share the same received counts, Discount offer type is more successful than BOGO, with the most prominent two being Discount type.

V. Acknowledgements

- Data Source: Starbucks.
- This wonderful course we I can learn, showcase and sharpen my Data skills: Udacity.