# Capstone Project

## Data Scientist Nanodegree

Xin Meng

March 18th, 2019

# Definition

## *Project Overview*

**Consumer behaviour**[1] is the study of individuals, groups, or organizations and all the activities associated with the purchase, use and disposal of goods and services, including the consumer's emotional, mental and behavioral responses that precede or follow these activities. Characteristics of individual consumers such as demographics, personality lifestyles and behavioral variables such as usage rates, usage occasion, loyalty, brand advocacy, willingness to provide referrals, in an attempt to understand people's wants and consumption are all investigated in formal studies of consumer behavior.
This project is about a customer behavior study of Starbucks. Once every few days, Starbucks sends out an offer to users. 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. The data set contains simulated data that mimics customer behavior on the Starbucks rewards. 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.

## *Problem Statement*

The goal of the project is to combine transaction, demographic and offer data to determine which demographic groups respond best to which offer type. With the force of data science, we have develop some tools like classification prediction model and recommendation system to increase the users purchase. We approach the solutions by the following ways:

1 Found the demographic users difference in 4 groups and understood how demographic features associated with customer behaviors:

[1] Customer behaviour <https://en.wikipedia.org/wiki/Consumer_behaviour>

Group1: Users completed the offer without viewed.

Group2: Users viewed the offer and completed it.

Group3: Users viewed the offer but not completed it.

Group4: Users did not view nor complete the offers.

2 Built a classification model to predict whether a user who receive the offer would complete it.

3 Built a user-user based collaborative filtering recommendation system for new and old users.

## *Metric*

We use the accuracy, precision score and recall score to evaluate the performance of the classification model in task 2.
Accuracy is intuitively the ability of the classifier to label correct.

Accuracy = $\frac{\text{true positives + true negtives}}{\text{dataset size}}$

The precision is intuitively the ability of the classifier not to label as positive a sample that is negative. The best value is 1 and the worst value is 0.

Precision = $\frac{\text{true positives}}{\text{true posives+false positives}}$

The recall is intuitively the ability of the classifier to find all the positive samples. The best value is 1 and the worst value is 0.

Recall = $\frac{true\ \ positives}{\text{true positives+false negatives}}$

F1 score is also known as balanced F-score. The F1 score can be interpreted as a weight average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal.

F1 score = 2 * (precision * recall) / (precision + recall)

# Analysis

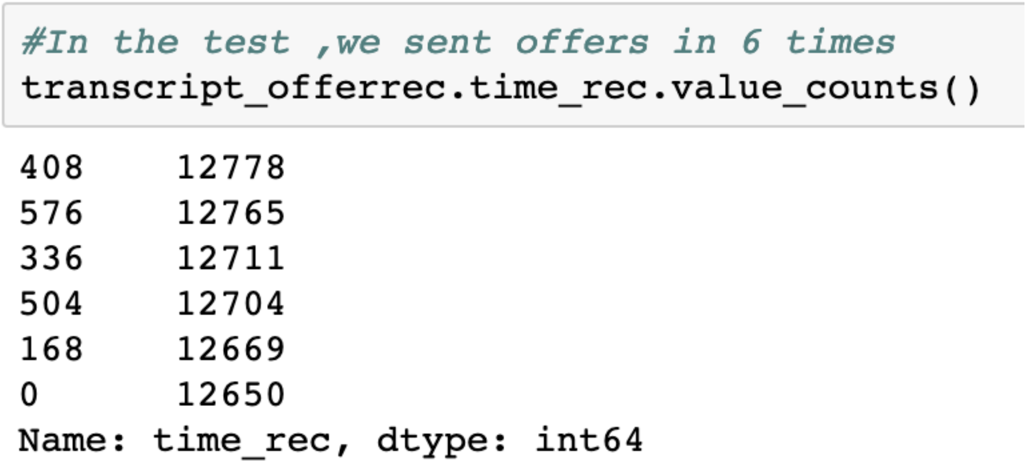
## *Data Exploration*

There are 10 offers in the portfolio dataset with the columns of channels (web/email/mobile/social), difficulty, duration, offer type, reward
There are 17000 users in the profile dataset with the demographic value of age, date of became member on, gender, income

There are 138953 rows of transaction event with userid, time, amount
There are 76277 rows of offer received event with userid, offerid, time
There are 57725 rows of offer viewed event with userid, offerid, time
There are 33579 rows of offer completed event with userid, offerid, time, reward user got

We have known that not all users received the same offer. Some users might not receive any offer during the test. User who received the offer did not view it could also complete it if he spent the amount of money greater than the offer difficulty before the offer expired. In this scenario, user was not influenced by the offer because he did not view the offer during the validity period. Users might completed the same offer and got rewards for several times. Users might receive the same offer from several different channels like email, web, social, mobile for several times.

Looked into the test we found that offers had been sent for 6 times from the first time of 0 hours to the last time of 576 hours from the test started. The max time of offer viewed event and the max time of offer completed event are both 714 hours. The max duration of offer is 10 days, which is 240 hours. So probably some offers sent in the final time did not expire when the test was end.



```
#In the test ,we sent offers in 6 times
transcript_offerrec.time_rec.value_counts()
```

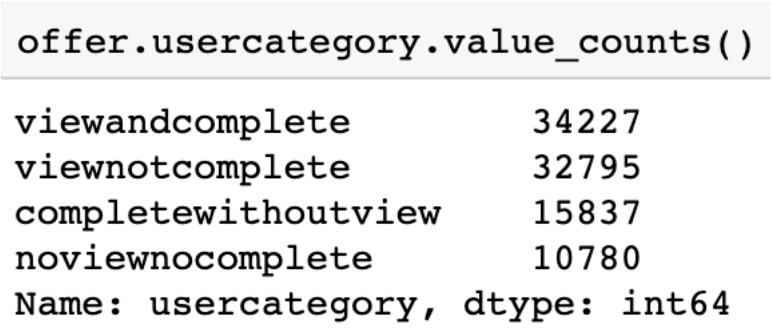
```
408    12778
576    12765
336    12711
504    12704
168    12669
0      12650
Name: time_rec, dtype: int64
```

## Exploratory Visualization

After data preprocessing which was elaborated in the Methodology-Data Preprocessing section. We split the user's behaviors into 4 groups based on their behavior of response and completion.

- Group1: Users completed the offer without view it.
- Group2: Users viewed the offer and completed it.
- Group3: Users viewed the offer but not completed it.
- Group4: Users did not view nor complete it.

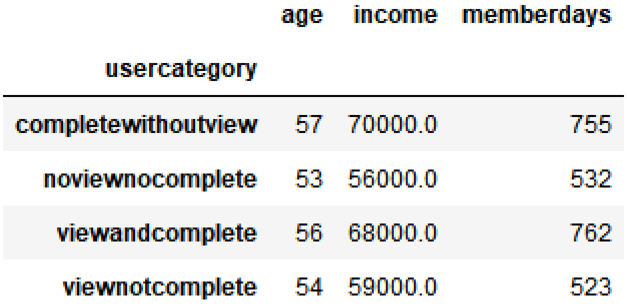
The number of user behavior in each group is:



```
offer.usercategory.value_counts()

viewandcomplete          34227
viewnotcomplete          32795
completewithoutview      15837
noviewnocomplete         10780
Name: usercategory, dtype: int64
```

We explored the demographic features of users in the 4 exclusive groups.



| usercategory | age | income | memberdays |
|---|---|---|---|
| completewithoutview | 57 | 70000.0 | 755 |
| noviewnocomplete | 53 | 56000.0 | 532 |
| viewandcomplete | 56 | 68000.0 | 762 |
| viewnotcomplete | 54 | 59000.0 | 523 |

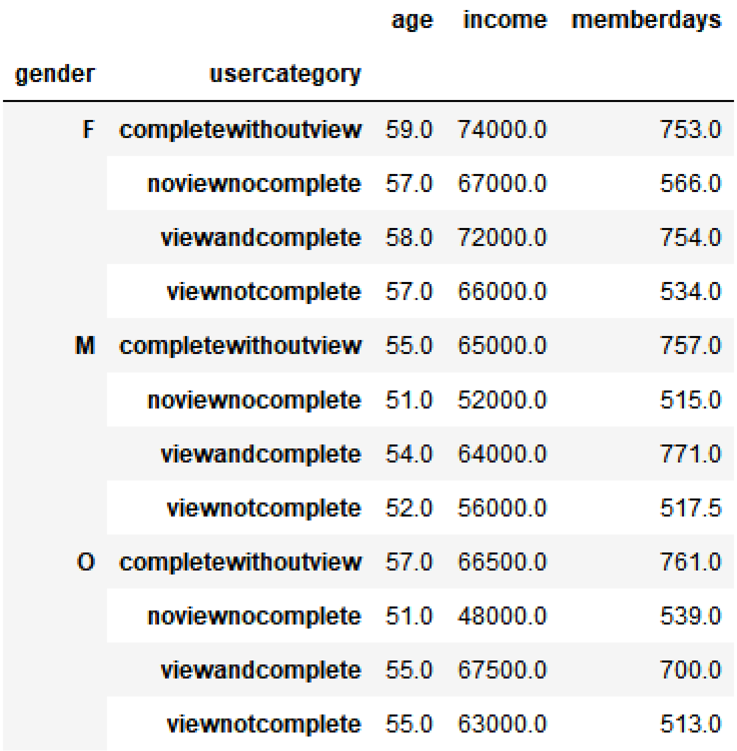
Observed the median of age,income and memberdays in the 4 groups, we saw:

The users who completed without viewed had the largest age (age median: 57), the second rank is the users who viewed and completed offers (age median: 56). Users who viewed the offers but not completed (age median: 54) and Users who did not view nor complete (median: 53) were much younger. That means elder users have more willing to buy.

The users who completed without viewed had highest income (income median: 70000).The second rank is the users who viewed and completed the offer (income median: 68000).Users who viewed the offers but not completed (income median: 59000) and users who did not view nor complete (income median:56000) had less income. That means users have higher income are less impacted by offers and also more willing to pay.

The users who completed without viewed (memberdays median: 755) and users who viewed and completed offers (membership median: 762) had the much longer period of membership than the other 2 groups. The other two groups of users who viewed but not completed and users who did not view nor complete only had membership period of about 500 days. That means we should send offers users who have long period of membership, they are more loyal and have more willing to pay.

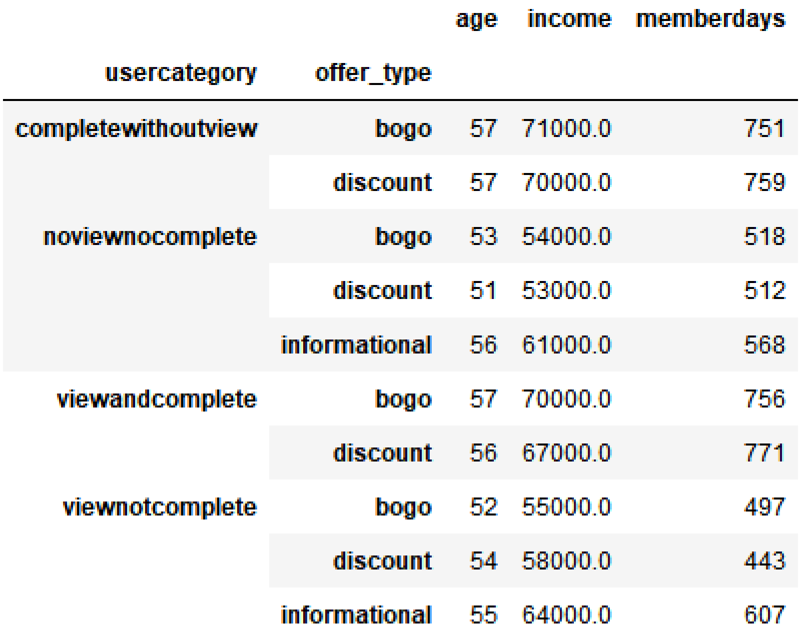
We also study the female and male users’ profile in different user groups.



| | | age | income | memberdays |
|---|---|---|---|---|
| gender | usercategory | | | |
| F | completewithoutview | 59.0 | 74000.0 | 753.0 |
| | noviewnocomplete | 57.0 | 67000.0 | 566.0 |
| | viewandcomplete | 58.0 | 72000.0 | 754.0 |
| | viewnotcomplete | 57.0 | 66000.0 | 534.0 |
| M | completewithoutview | 55.0 | 65000.0 | 757.0 |
| | noviewnocomplete | 51.0 | 52000.0 | 515.0 |
| | viewandcomplete | 54.0 | 64000.0 | 771.0 |
| | viewnotcomplete | 52.0 | 56000.0 | 517.5 |
| O | completewithoutview | 57.0 | 66500.0 | 761.0 |
| | noviewnocomplete | 51.0 | 48000.0 | 539.0 |
| | viewandcomplete | 55.0 | 67500.0 | 700.0 |
| | viewnotcomplete | 55.0 | 63000.0 | 513.0 |

We saw that female users had higher income than male users. Female users' median age was also higher than male users'.

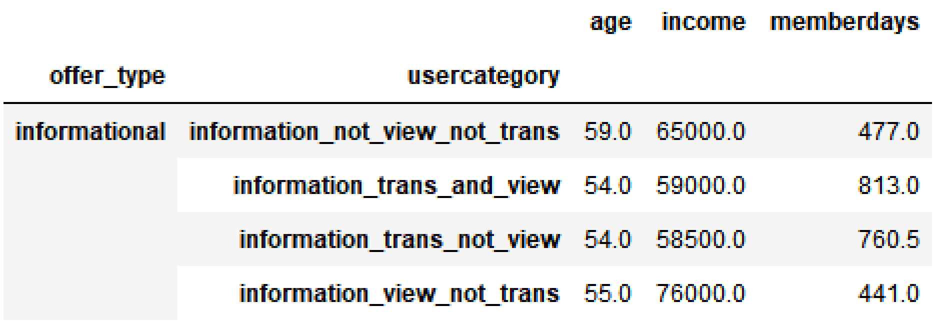
We explored the user demographic in each offer type.



| | | age | income | memberdays |
|---|---|---|---|---|
| usercategory | offer_type | | | |
| completewithoutview | bogo | 57 | 71000.0 | 751 |
| | discount | 57 | 70000.0 | 759 |
| noviewnocomplete | bogo | 53 | 54000.0 | 518 |
| | discount | 51 | 53000.0 | 512 |
| | informational | 56 | 61000.0 | 568 |
| viewandcomplete | bogo | 57 | 70000.0 | 756 |
| | discount | 56 | 67000.0 | 771 |
| viewnotcomplete | bogo | 52 | 55000.0 | 497 |
| | discount | 54 | 58000.0 | 443 |
| | informational | 55 | 64000.0 | 607 |

We saw users who completed bogo and discount offers had longer membership period (above 750 days) than users who did not completed offers (about 500days). Ages of users who completed bogo and discount offers (above 56) are higher than users did not (about

52). Users who completed bogo and discount offers have higher income (above 67000) than who did not (about 55000). We should notice that information offer do not have complete events, so next we will examine how information offers influence on transactions.



| | | age | income | memberdays |
|---|---|---|---|---|
| offer_type | usercategory | | | |
| informational | information_not_view_not_trans | 59.0 | 65000.0 | 477.0 |
| | information_trans_and_view | 54.0 | 59000.0 | 813.0 |
| | information_trans_not_view | 54.0 | 58500.0 | 760.5 |
| | information_view_not_trans | 55.0 | 76000.0 | 441.0 |

We saw users who had transactions with or without viewed had longer membership days(above 760 days) and less income (income median 59000) than users who did not had transactions with or without viewed (membership about 450 days and income above 65000)..

## *Algorithms and Techniques*

In task 2, we used DecisionTreeClassifier and RandomForestClassifier to build classification model and predict whether users who receive the offer would complete it.

In task 3, we used user-user based collaborative filtering recommendation algorithm to find similar users to the user who will be recommended offers. We used Minmaxscaler to scale the profile numerical values in the same scale (0,1) and used Euclidean distance to calculate user-user distance and found the top n most similar users to the one we will make recommendation for. Then we found out the offers which have the maximum cumulated rewards completed by the top n most similar users and recommended these offers to the user.

## *Benchmark*

In task 2, we have the performance on DecisionTreeClassifier with default parameters of:

the accuracy on testset is 0.6594181604466647
the presicion score is 0.6457960644007156,the recall score is 0.6571601941747572, the f1_score is0.6514285714285714

This could be used as benchmark and I used more advanced ensemble algorithm to beat the benchmark.

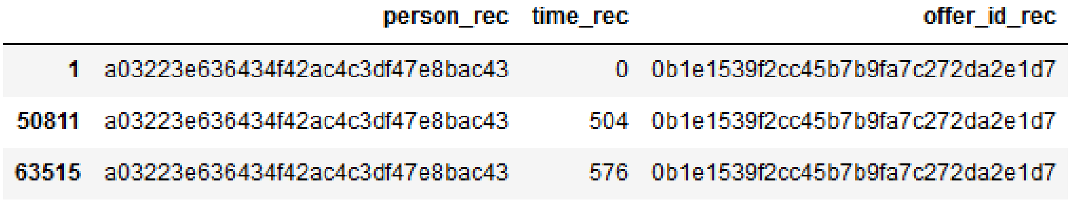
# Methodology

## *Data Preprocessing*

To study the influence of offers impact on users spending behaviors, we need to merge offer received, viewed and completed dataset. We are interested in how customer would response when they receive an offer.

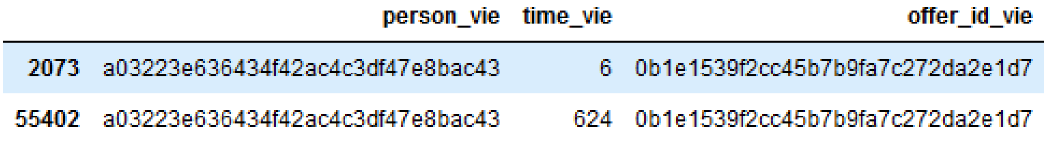
The first thing is to build a total data set from offer received, viewed and completed dataset. We took offer received dataset left joined offer viewed dataset on userid and offerid. We kept all received events rows which were sent during the test and there were some received events did not have corresponding viewed events because they were not viewed by users.

We found that offers might be sent to the same user for several times. For example: user 'a03223e636434f42ac4c3df47e8bac43' received offer '0b1e1539f2cc45b7b9fa7c272da2e1d7' for 3 times at 0 hour, 504 hours and 576 hours.



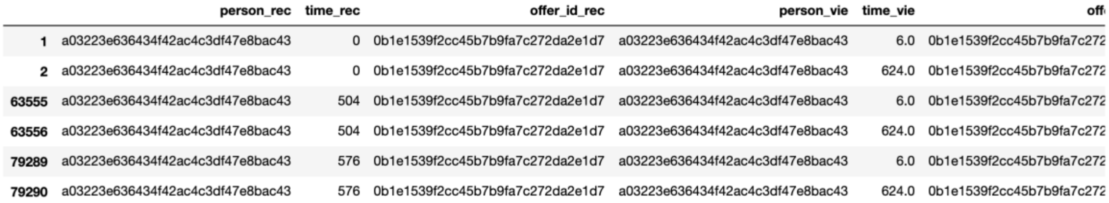
| | person_rec | time_rec | offer_id_rec |
|---|---|---|---|
| **1** | a03223e636434f42ac4c3df47e8bac43 | 0 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 |
| **50811** | a03223e636434f42ac4c3df47e8bac43 | 504 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 |
| **63515** | a03223e636434f42ac4c3df47e8bac43 | 576 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 |

The user viewed the offer for 2 times at 6hours and 624hours.



| | person_vie | time_vie | offer_id_vie |
|---|---|---|---|
| **2073** | a03223e636434f42ac4c3df47e8bac43 | 6 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 |
| **55402** | a03223e636434f42ac4c3df47e8bac43 | 624 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 |

After merged, the offer_receive_view dataset looks like：



| | person_rec | time_rec | offer_id_rec | person_vie | time_vie | off |
|---|---|---|---|---|---|---|
| **1** | a03223e636434f42ac4c3df47e8bac43 | 0 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 | a03223e636434f42ac4c3df47e8bac43 | 6.0 | 0b1e1539f2cc45b7b9fa7c272 |
| **2** | a03223e636434f42ac4c3df47e8bac43 | 0 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 | a03223e636434f42ac4c3df47e8bac43 | 624.0 | 0b1e1539f2cc45b7b9fa7c272 |
| **63555** | a03223e636434f42ac4c3df47e8bac43 | 504 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 | a03223e636434f42ac4c3df47e8bac43 | 6.0 | 0b1e1539f2cc45b7b9fa7c272 |
| **63556** | a03223e636434f42ac4c3df47e8bac43 | 504 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 | a03223e636434f42ac4c3df47e8bac43 | 624.0 | 0b1e1539f2cc45b7b9fa7c272 |
| **79289** | a03223e636434f42ac4c3df47e8bac43 | 576 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 | a03223e636434f42ac4c3df47e8bac43 | 6.0 | 0b1e1539f2cc45b7b9fa7c272 |
| **79290** | a03223e636434f42ac4c3df47e8bac43 | 576 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 | a03223e636434f42ac4c3df47e8bac43 | 624.0 | 0b1e1539f2cc45b7b9fa7c272 |

There were 6 rows in the merged dataset (for the user and the offer above). We spot some wrong rows of time_receive larger than time_view. We must drop the wrong rows after merge operation.

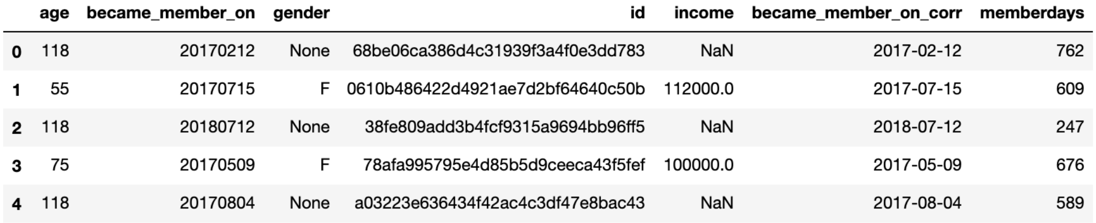
There are still be some man-made rows like the second row (time_rec=0, time_vie=624). we can't identify whether they actually happened or were only created by merge operation. We have to keep these with the risk that some rows created by merge operation which did not actually happen in the dataset we investigate.

Then we merge offer_receive_view dataset with complete event dataset. We took offer_receive_view dataset left joined offer complete dataset on person_receive and offer_id_receive, because some users completed the offer without viewed it. The same things of merge operation we need to pay attention to. We dropped the rows with time_receive larger than time_complete. We got offer_receive_view_complete dataset with wrong rows dropped.

The second step is to split user groups based on users view and complete events. We created new column of ‘usercategory’ to indicate which group user was in.

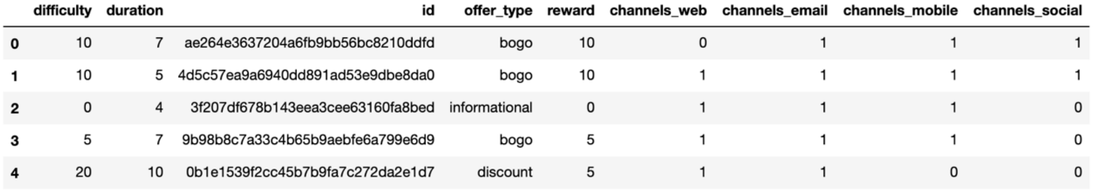
- Group1: Users completed the offer without view it.
- Group2: Users viewed the offer and completed it.
- Group3: Users viewed the offer but not completed it.
- Group4: Users did not view nor complete it.

We also preprocessing profile dataset and portfolio dataset. In the profile dataset, we created new columns of ‘memberdays’ to indicate how many days since user became a member. We dropped rows which have null value in gender and age.



| | age | became_member_on | gender | id | income | became_member_on_corr | memberdays |
|---|---|---|---|---|---|---|---|
| **0** | 118 | 20170212 | None | 68be06ca386d4c31939f3a4f0e3dd783 | NaN | 2017-02-12 | 762 |
| **1** | 55 | 20170715 | F | 0610b486422d4921ae7d2bf64640c50b | 112000.0 | 2017-07-15 | 609 |
| **2** | 118 | 20180712 | None | 38fe809add3b4fcf9315a9694bb96ff5 | NaN | 2018-07-12 | 247 |
| **3** | 75 | 20170509 | F | 78afa995795e4d85b5d9ceeca43f5fef | 100000.0 | 2017-05-09 | 676 |
| **4** | 118 | 20170804 | None | a03223e636434f42ac4c3df47e8bac43 | NaN | 2017-08-04 | 589 |

In the portfolio dataset, we created 4 columns of ‘channels_web’, ‘channels_email’, ‘channels_social’ and ‘channels_mobile’ to indicate by which channel offers were sent.



| | difficulty | duration | id | offer_type | reward | channels_web | channels_email | channels_mobile | channels_social |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 10 | 7 | ae264e3637204a6fb9bb56bc8210ddfd | bogo | 10 | 0 | 1 | 1 | 1 |
| **1** | 10 | 5 | 4d5c57ea9a6940dd891ad53e9dbe8da0 | bogo | 10 | 1 | 1 | 1 | 1 |
| **2** | 0 | 4 | 3f207df678b143eea3cee63160fa8bed | informational | 0 | 1 | 1 | 1 | 0 |
| **3** | 5 | 7 | 9b98b8c7a33c4b65b9aebfe6a799e6d9 | bogo | 5 | 1 | 1 | 1 | 0 |
| **4** | 20 | 10 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 | discount | 5 | 1 | 1 | 0 | 0 |

Finally, we got offer_p_f dataset which merged offer_receive_view_complete dataset with portfolio and profile dataset.

## *Implementation*

In task 2, we built a classifier to predict whether a user who receive the offer would view and complete it based on demographic user features and offer portfolio. For users who would complete without view are not the group we are interested in, because they are not influenced by offers and we don’t need to send offers to them.

Also, we found that information offers did not have corresponding complete events, so we removed information offers events from the dataset and create offer_p_f_2 to build the model.

We extracted X and Y columns form offer_p_f_2 dataset which merged with portfolio and profile dataset.

X columns = ['age', 'gender','income','memberdays', 'difficulty', 'duration','offer_type', 'reward','channels_web', 'channels_email', 'channels_mobile', 'channels_social']

Y columns =[‘viewandcomplete’]

We checked the statistic of the dataset and found:

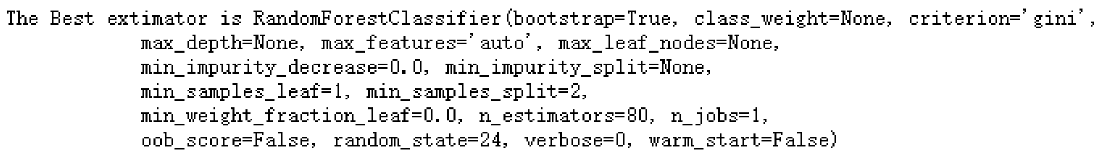
The nrows of dataset is 68057,the nrows of completed is 32772,the ratio of completed is 0.48153753471354893

The dataset is a balanced dataset, so we use accuracy to evaluate the model performance and we also use precision and recall score to explain the performance.
We used DecisionTreeClassifier with default parameters and got the performance on the test set:
The accuracy on testset is 0.6594181604466647
The presicion score is 0.6457960644007156, the recall score is 0.6571601941747572, the f1_score is0.6514285714285714

In task 3, we built a recommendation system with user-user based collaborative filtering algorithm. The idea is to find the most similar old user for the new user and recommend the old user's cumulated maximum reward offers he have completed to the new user. This recommendation system also works for old users.
We used Euclidean distance to calculate user-user distance based on profile. Since columns in profile have different scaler. For example, 'income' have values much larger than others like 'age', so column 'income' has dominate influence on the distance. That is harmful to the result. We need to rescale them to range (0,1) by Minmaxscaler. Obviously, we need to use get_dummies() to transform categorical columns 'gender' to one-hot encoder. Attention here we should keep all three possible values: F, M, O by 'drop_false=False'. However, in task 2 classification model we drop the first value. We set index to userid.
We created 'find_similar_users()' function to return the top n most similar users to the input user and we created 'recommendation()' function to find out each most similar users' largest cumulated reward amount they completed. We removed the offers which users did not influenced by from offer_p_f_2 dataset. Since similar users often have the similar offers which they got maximum cumulated rewards, sometimes we can't give us 5 recommended offers as we ask for recommendation () function. So, the 'recommendation ()' function return us probably less than the n_recom we set in parameters.

## *Refinement*

In task 2 classification model, we used advanced ensemble algorithm RandomForestClassifier and tuned parameters with GridSearchCV. We tuned the parameter of n_estimator with [20, 50, 80]…
The best estimator chose n_estimators=80. We got the best estimator is:



```
The Best extimator is RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=80, n_jobs=1,
            oob_score=False, random_state=24, verbose=0, warm_start=False)
```

# Result

## *Model Evaluation and Validation*

In task 2, classification model, we got the final model with RandomForestClassifier with the tuned parameter by GridSearchCV and the performance on testset is:

The accuracy of RandomForest clssifier on testset is 0.6898814771280243
The precision score is 0.673734610123119, the recall score is 0.6973098705501618, the f1_score is0.6853195507404831

That means we have accuracy of 68.99% of predict correctly on whether a user would complete the offer. For a predict user who would complete the offer, we have 67.37% confident he would truly complete the offer. And for users who would actually complete the offers, there are 69.73% of the users he would be identified by the model.

## *Justification*

In task 2, classification model, we got the final model and performance with RandomForestClassifier is :
The accuracy of RandomForest clssifier on testset is 0.6898814771280243
The precision score is 0.673734610123119,the recall score is 0.6973098705501618, the f1_score is0.6853195507404831

The RandomForestClassifier model beats the benchmark of DecisionTreeClassifier:
The accuracy on testset is 0.6594181604466647
The precision score is 0.6457960644007156,the recall score is 0.6571601941747572, the f1_score is0.6514285714285714

# Conclusion

We have successfully achieved the project goal by completed the 3 tasks.
In task 1, we figured out demographic difference between the 4 user groups based on user’s behavior on offers. We concluded we have found in exploratory analysis:

- Elder users (age about 56-57) have more willing to buy.
- Users have higher income (income about 70000) are less impacted by offers and also more willing to pay.
- Users who have long period of membership (about 750days), they are more loyal and have more willing to pay.

We created a typical user profile who would view and completed offers: a female, about 58, she has income of 72000 and she has been a member for 754days. Or a male, he is 54, he has income of 64000, and he has been a member for 771 days.

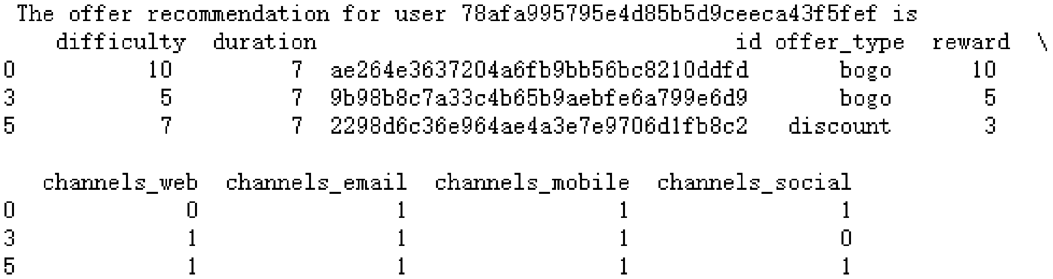
We saw users who completed bogo and discount offers had longer membership period (above 750days) than users who did not completed offers (about 500days). Ages of users who completed bogo and discount offers (above 56) are higher than users did not (about 52). Users who completed bogo and discount offers have higher income (above 67000) than who did not (about 55000). .

For users who received information offers, we saw users who had transactions with or without viewed had longer membership days(above 760 days) and less income (income median 59000) than users who did not had transactions with or without viewed (membership about 450 days and income above 65000).

In task 2, we built a classification model with RandomForestClassifier to predict users would view and complete offer he receives. For bogo and discount offers, we have accuracy of 68.99% of predict correctly on whether a user would complete the offer. For a predict user who would complete the offer, we have 67.37% confident he would truly complete the offer. And for users who would actually complete the offers, there are 69.73% of the users he would be identified by the model.

In task 3, we built a user-user based collaborative filtering recommendation system. We recommend user offers which his similar users have got cumulated maximum rewards from these offers and we remove offers the user did not influenced by according to history(from offer_p_f_2 dataset).

For example, for user’78afa995795e4d85b5d9ceeca43f5fef’ we asked for 5 offer rec ommendations, we got:



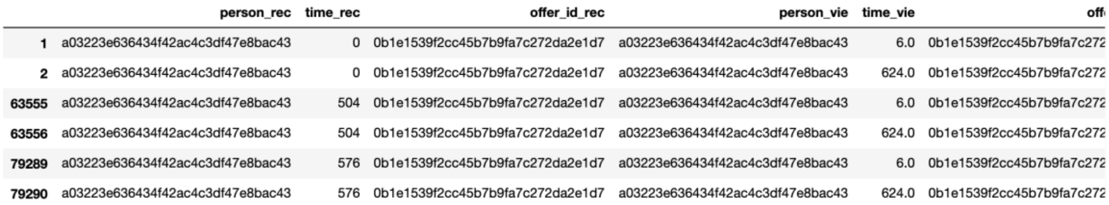
```
 The offer recommendation for user 78afa995795e4d85b5d9ceeca43f5fef is
   difficulty  duration                              id offer_type  reward  \
0          10         7  ae264e3637204a6fb9bb56bc8210ddfd       bogo      10   
3           5         7  9b98b8c7a33c4b65b9aebfe6a799e6d9       bogo       5   
5           7         7  2298d6c36e964ae4a3e7e9706d1fb8c2   discount       3   

   channels_web  channels_email  channels_mobile  channels_social  
0             0               1                1                1  
3             1               1                1                0  
5             1               1                1                1  
```

## Reflection

The most difficult part for me which consumed lots of time was finding the idea of building the offer_receive_view_complete dataset to create user groups. At the beginning, I made a mistake to consider users who viewed the offer without completed it was to subtract the unique users who completed the offers from unique users who viewed the offers. User behavior on offer must be studied on combination of user and offer. That’s why I abandoned transactions events because it only contains user information not offer. I can’t identify these transactions were with or without offers influence. I only use transactions to analyze information offers influence. For bogo and discount offers I did not use transaction to analyze. After made it clear that I needed create offer_receive_view_complete dataset with dropping wrong rows (I illustrated it in the data preprocessing section) created by merge operations, the tasks of split user groups, built classification model are much easier.

In the Data Preprocessing section, I mentioned that merge operation might introduce some rows which was not actually happened. For instance, in the following image, the reasonable receive-view rows are received at 0 hours/viewed at 6 hours, or received at 504hours/viewed at 624hours, or received at 576hours/viewed at624hours. The second row of received at 0 hour/viewed at 624 hours probably not happened and be created by merged operation. However, these man-made rows are not impact on the user group split, it might have some influence on classification model. Because some positive samples are introduced which are not really happened. So, the actual predict performance probably a little lower than the report.



| | person_rec | time_rec | offer_id_rec | person_vie | time_vie | off |
|---|---|---|---|---|---|---|
| **1** | a03223e636434f42ac4c3df47e8bac43 | 0 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 | a03223e636434f42ac4c3df47e8bac43 | 6.0 | 0b1e1539f2cc45b7b9fa7c272 |
| **2** | a03223e636434f42ac4c3df47e8bac43 | 0 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 | a03223e636434f42ac4c3df47e8bac43 | 624.0 | 0b1e1539f2cc45b7b9fa7c272 |
| **63555** | a03223e636434f42ac4c3df47e8bac43 | 504 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 | a03223e636434f42ac4c3df47e8bac43 | 6.0 | 0b1e1539f2cc45b7b9fa7c272 |
| **63556** | a03223e636434f42ac4c3df47e8bac43 | 504 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 | a03223e636434f42ac4c3df47e8bac43 | 624.0 | 0b1e1539f2cc45b7b9fa7c272 |
| **79289** | a03223e636434f42ac4c3df47e8bac43 | 576 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 | a03223e636434f42ac4c3df47e8bac43 | 6.0 | 0b1e1539f2cc45b7b9fa7c272 |
| **79290** | a03223e636434f42ac4c3df47e8bac43 | 576 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 | a03223e636434f42ac4c3df47e8bac43 | 624.0 | 0b1e1539f2cc45b7b9fa7c272 |

## *Improvement*

The project could be improved in some ways.

1. Improved the dataset. We can’t track accurately on each offer user received, viewed and completed result in man-made rows might be introduced by merge operation. One way to solve it is to create a new column of offersending_id which is unique on each sending. One offer in portfolio should have unique offersending_id on each channel at each time. We have a map of offersending_id and offerid in portfolio. This solution will give us accurate track on how user interact with the offer they receive each time form each channel. We will have a more accurate estimator on classification model performance.
2. In the user-user based collaborative filtering recommendation, I’d like to add some information offers for novelty to amuse users. Recommendation () function only return offers that have rewards. So, information offers would never recommended by recommendation () function. However, users might like to receive information offers. If

we want to recommend 5 offers to a user, while the recommendation () function only return 3 offers, we can send 2 information offers to him. The design of recommendation is decided by the intention of how to influence users and the strategy of marketing and sales.

3. We can also take A/B test study to evaluate the recommendation system. And have statistical confidence whether the recommendation increase the purchase. The study should be on bogo and discount offers.