Capstone Report

Project Overview

This project is the final project of Udacity Machine Learning Engineer Nanodegree Program. The 3 datasets provided by Udacity are simulated data that mimics behaviors of customers of Starbucks rewards app. Once a customer registered this app, he/she will receive offers such as discounts, buy-one-get-one-free, or advertisements.

Customers response differently to offers. Some complete after viewing the offers, and some doesn't; some complete without viewing the offers, and some view after completing the offers; some complete offer A but not offer B, and some does the opposite; some complete soon after receiving the offers, and some wait till the last second to complete the offers.

The goal is to build a model that finds the best offer to each customer.

Data Cleaning

1. portfolio.json – information about offers
    a) reward – reward a customer gets once complete the offer (0-10)
    b) channels – channel the offer is distributed (web, email, mobile, social)
        i. I split this column into 4 columns using pandas.explode(), and pandas.get_dummies(), so that if an offer is distributed by a specific channel (eg. web), then the value of that column is 1, otherwise 0.
    c) difficulty – minimum dollars a customer has to spend to complete the offer (0-10)
    d) duration – the validity period of the offer (3-10)
        i. the duration was in days, and I change it to hours, because the time in transcript.json is in hours and name the column as duration_h
    e) offer_type – bogo, discount, informational
    f) id – unique id of each offer
        i. I simplify the id as first character of offer_type plus index (eg. b0, b1, d5, i9) and name the column as offer_id
2. profile.json – information about customers
    a) gender – F for female, M for male, O for other
        i. I replace genders with numbers because most algorithms don't work with string (M->1, Female->2, O->3, unknown->4)
    b) age
        i. there are 2175 customers with age 118, and unknown gender and income, so I treat age 118 as unknown
        ii. I use pd.cut() to separate age into 4 bins: (18,30], (30,50], (50,70], (70,101] and name the column as age_bin

c) id – unique id of each customer
    i. I rename the column as member_id
d) became_member_on – date that the customer registered the app
    i. I change the data type to datetime, and then extract year, month, and calculate membership days and name the columns as year, month, and membership_days respectively
e) income
    i. I use pd.qcut() to separate income into 4 bins: (29999.99,49000], (49000,64000], (64000,80000], (80000,120000] and name the column as income_bin

3. transcript.json – record of events
    a) person – unique id of each customer
        i. I rename the column as member_id
    b) event – offer_received, offer_viewed, offer_completed, transaction
    c) value – details of the event, including offer_id, amount of money, or number of rewards
        i. I separate the column into offer_id, amount, rewards columns
    d) time – hours after the first event

Feature Engineering

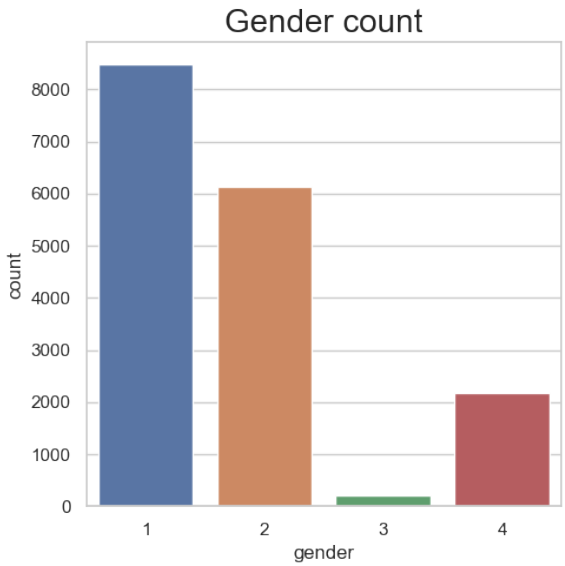
I create the following features:

1. number of offers received
2. number of offers viewed
3. number of offers completed
4. number of offers completed without viewing
5. number of offers viewed after completing
6. percentage of offers viewed (view/receive)
7. percentage of offers completed (complete/receive)
8. percentage of offers completed without viewing (complete without view/complete)
9. percentage of offers viewed after completing (view after complete/view)
10. hours between receive and view offers
11. hours between receive and complete offers
12. number of transactions made
13. number of transactions made after viewing offers
14. amount of money spent
15. amount of money spent after viewing offers

Exploratory Analysis

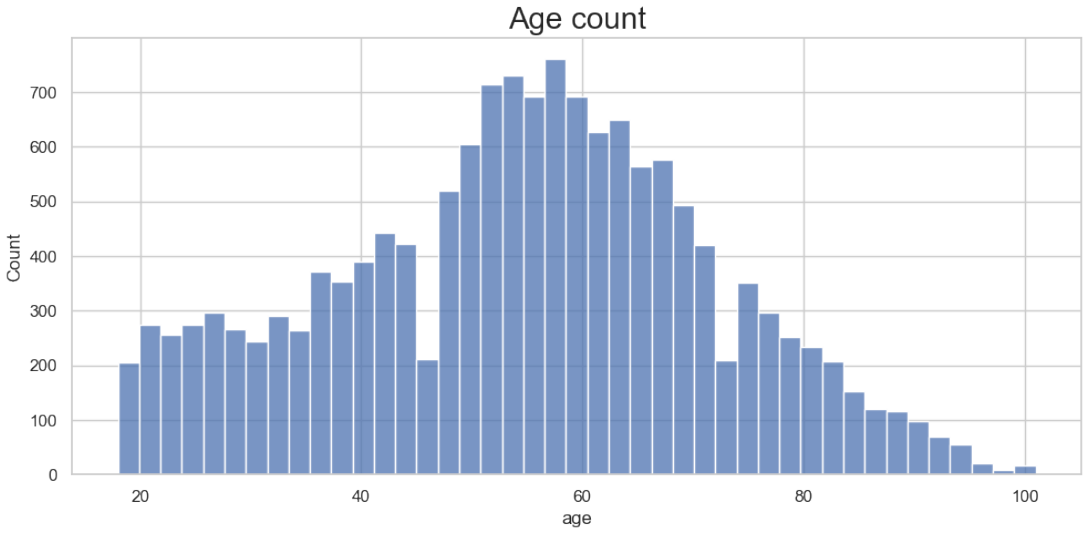
In order to analyze this data, I asked the following questions:

1. **What are the gender distribution?**

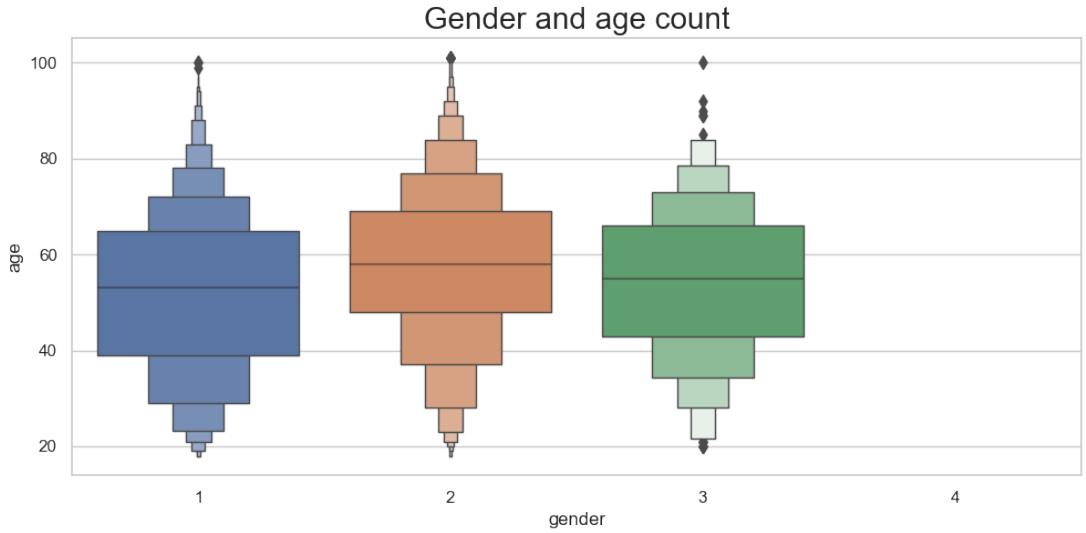


The nearly half of population is male, 36% is female, 1% is other gender, and 13% is unknown gender.

2. How old are the customers?

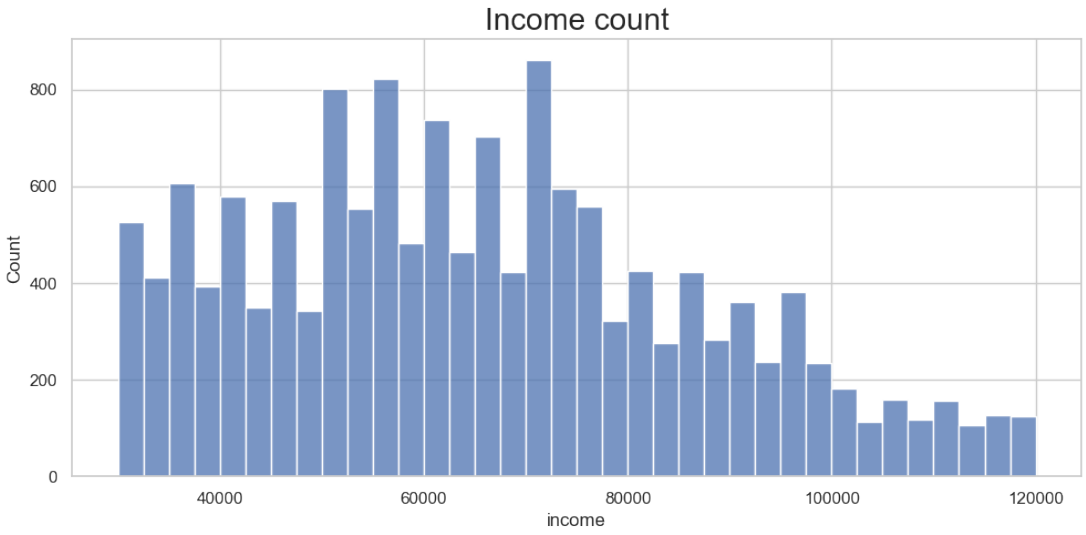


The majority of customers are between age 50 and 70, so this dataset consists mostly of old people.



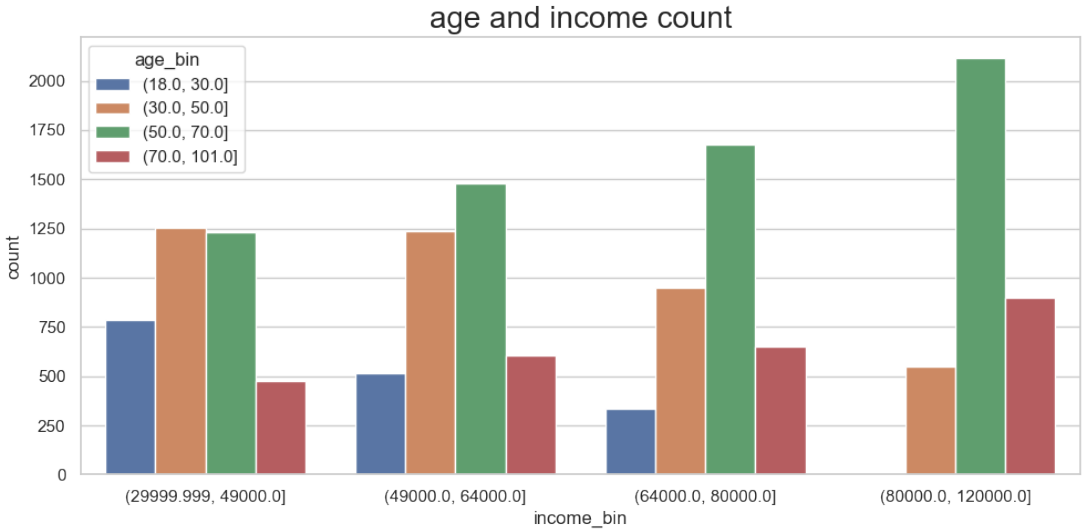
The percentage of old people in female group is higher than others.

3. How much do they earn each year?



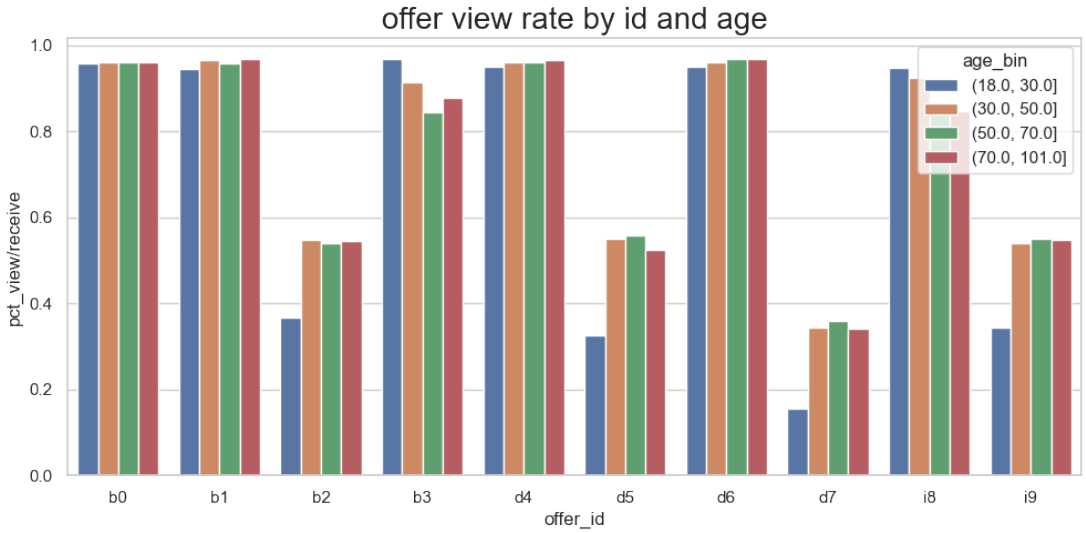
Majority of customers have annual income of $49000 to $80000.

Female customers earn more than other genders.



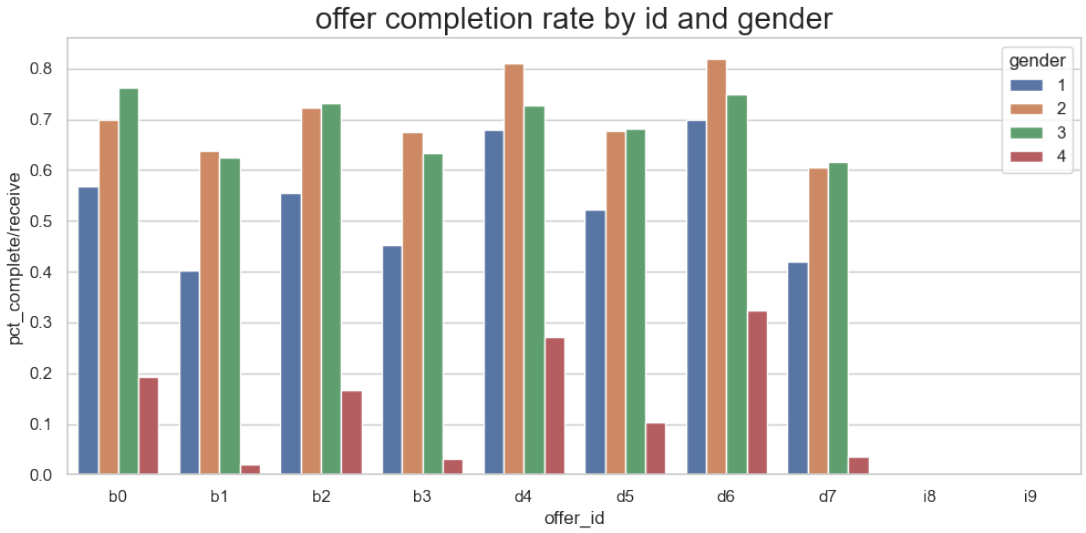
Age and income are positively correlated, and the poorest group is below age 30, the richest group is between age 50 and 70.

4. which type of offers are most attractive?

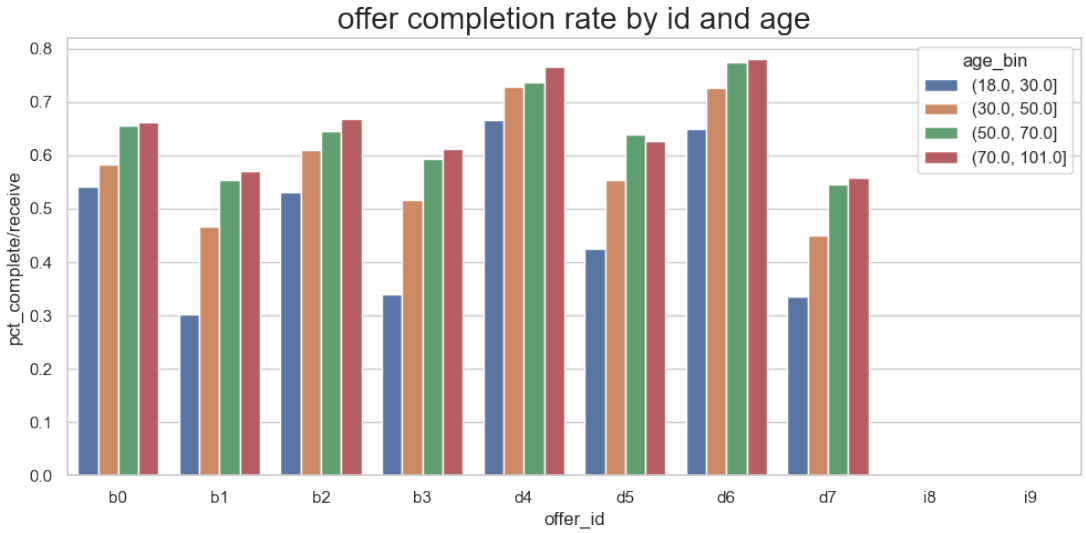


The offers with highest view rate are b0, b1, d4, and d6, because these 4 offers are distributed via all 4 channels (email, mobile, social, and web). And those offers with low view rate are not distributed via social channel.

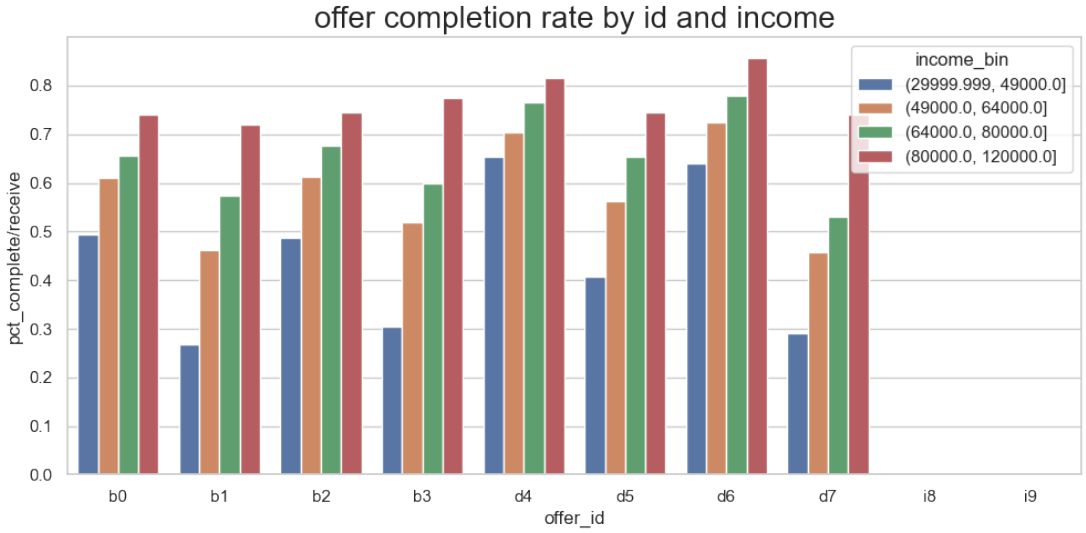
5. Which offer has highest completion rate?



Completion rate of d4 and d6 are slightly higher than others, and female customers have higher completion rate.

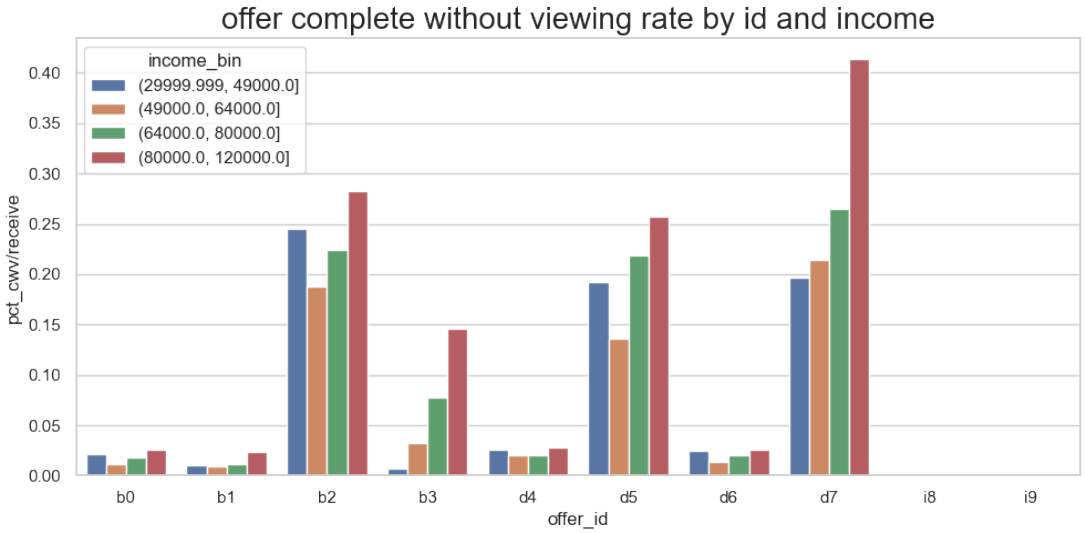


No matter which offer, age is positively correlated with completion rate



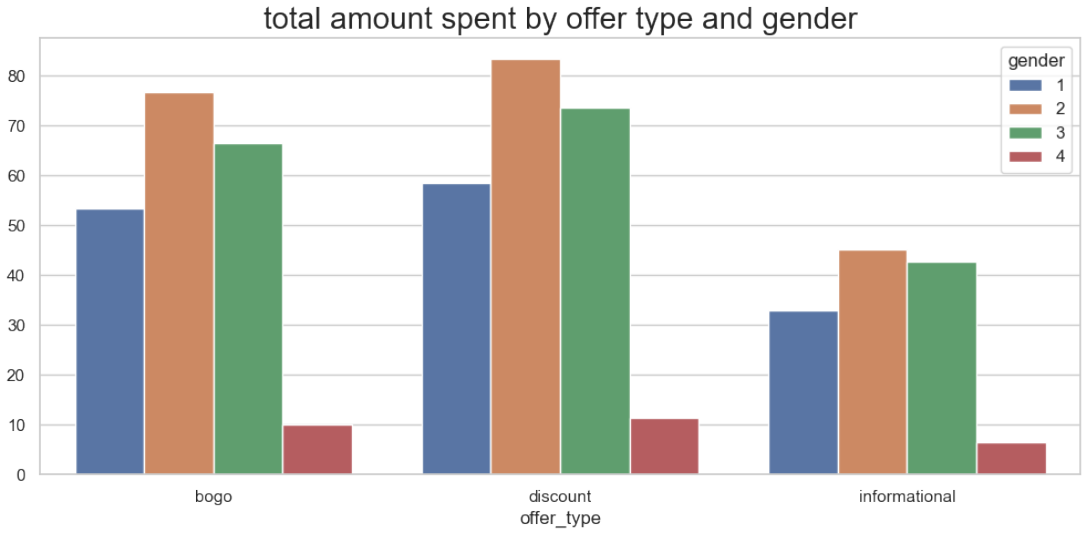
Income is also positively correlated with completion rate, because age is positively correlated with income as well.

6. Who is most likely to complete an offer without viewing it?

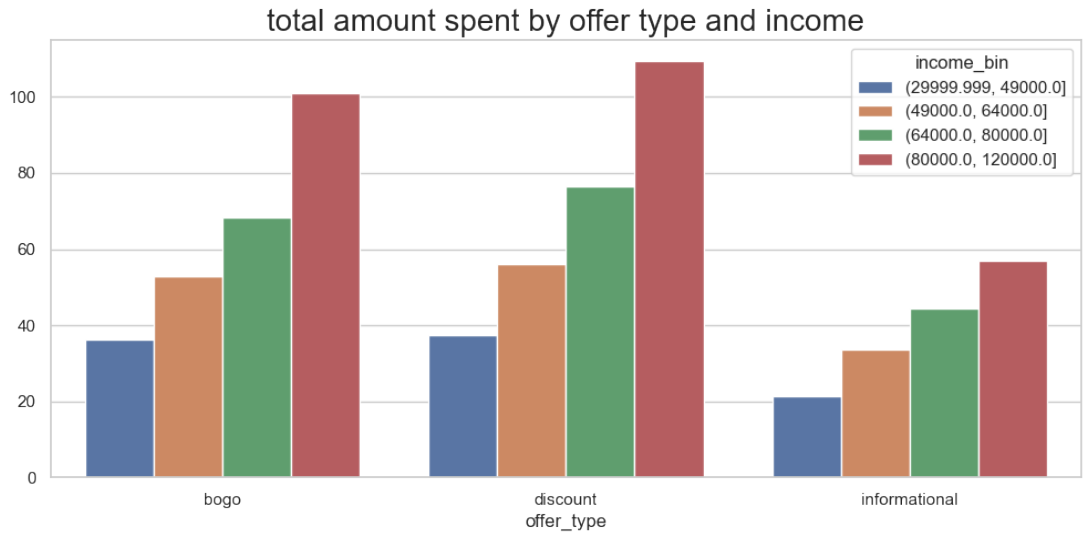


Offer b2, d5, and d7 have the highest complete_without_view_rate, because they are not distributed via social channel, and customers in highest income group have the highest compelte_without_view_rate, I guess it's because they are not sensitive to these offers.

7. How much did they spend after receiving each offer?

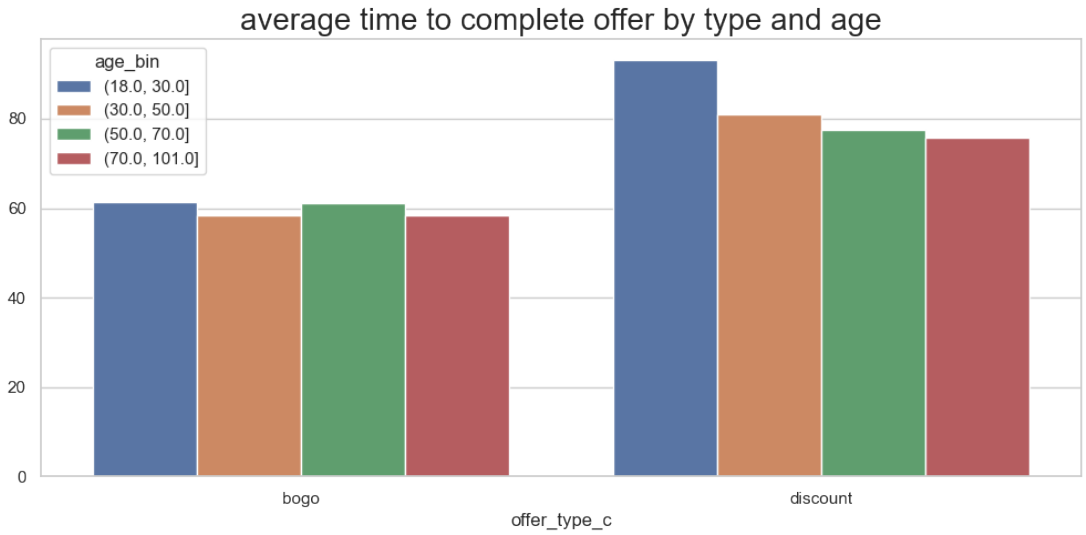


No matter which type of offers, females spent the most, and the unknown gender customers spent the least. And no matter which gender, customers spent most after viewing discount offers.



The amount of money spent is also positively correlated with income.

8. How long did it take a customer to complete an offer after receiving it?



It generally takes longer time for customers to complete discount offers than bogo, and younger people especially acted faster to complete the discount offers.

Predictive Modeling

I calculate the completion rate of each offer for each customer, and the one with highest rate is the final pick for that customer. If the customer hasn't completed any offer, then I will randomly choose one that hasn't been sent to him/her as the final pick.

This is a multiclassification issue, so I choose accuracy score and confusion matrix as metrics. I first use naïve bayes algorithm (sklearn.naive_bayes.BeroulliNB) and get accuracy score of 0.49, and then use decision tree algorithm (sklearn.tree.DecisionTreeClassifier) and get accuracy score of 0.74, and then use

gradient boosting algorithm (sklearn.ensemble.GradientBoostingClassifier) and get accuracy score of 0.83.

Improvements

Firstly, since this data is simplified with limited time duration and number of transactions, my analysis and the model I built may not be useful in the real world. Secondly, I set the rule by merely ranking the completion rate and pick the one with highest rate for each customer, which may not be most appropriate. Finally, the accuracy score of 0.83 is acceptable, so I didn't tune the hyperparameters or try more algorithms. Without the issues mentioned above, the model may perform much better.