Machine Learning Engineer Nanodegree Udacity

Capstone Project

Barun Mishra(9/6/2021)

Definition

a. Project Overview

This data set contains simulated data from 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.

Our 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. We 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.

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.

Someone using the app might make a purchase through the app without having received an offer or seen an offer.

Problem Statement

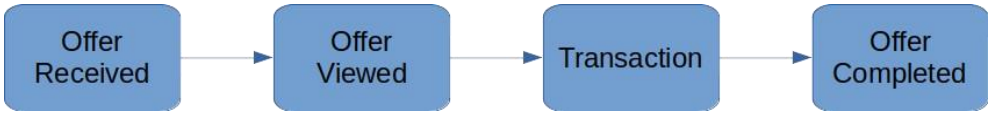
Starbucks, as well as any other company, invests money in marketing campaigns expecting to have a profit higher than the assumed before. So, identifying the most relevant offer to the correct customers is crucial for a successful campaign.

However, some targeted customers do not even see the offer sent to them, which may be a problem with the channel chosen. Other ones do not buy anything, despite seeing the offer, what might be a problem with the offer type sent, or maybe that is not a customer to be considered as a target.

The problem, which I am trying to solve, is to identify the customers to whom starbucks can provide the offer, which means finding the customers to whom Starbucks can aim and providing offers which can that is more likely to lead the customer to buy more Starbucks products

In the context of this project, an appropriate offer is that one where the customer sees the offer received and buys products under its influence, completing the offer lifecycle.

Appropriate offer



If a customer does not see an offer, it is not an appropriate one. If he or she sees the offer but does not complete it, it is not appropriate as well, since it did not lead the consumer to buy products. Similarly, if the customer buys some products, completes an offer, and receives a reward before visualizing that offer, it is not considered effective because the customer was not under the influence of that offer when decided to make a purchase.

Metrics:

I will be using F1 –score as my metrics for modelling,as we have classification problem accuracy will not suit our goal, F1score will make sure we correctly specific the customers under different bucket

Analysis

Data Sets

The data is contained in three files:

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

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

portfolio.json

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

Data loading and exploration Visualization

- Load files and present some data visualization in order to understand the distribution and characteristics of the data, and possibly identify inconsistencies.

Portfolio:

- It has 10 rows and 6 columns with no missing values
- Channels column was segregated email , mobile, social.
- Offer_type column was segregated into bogo, discount, information

Profile:

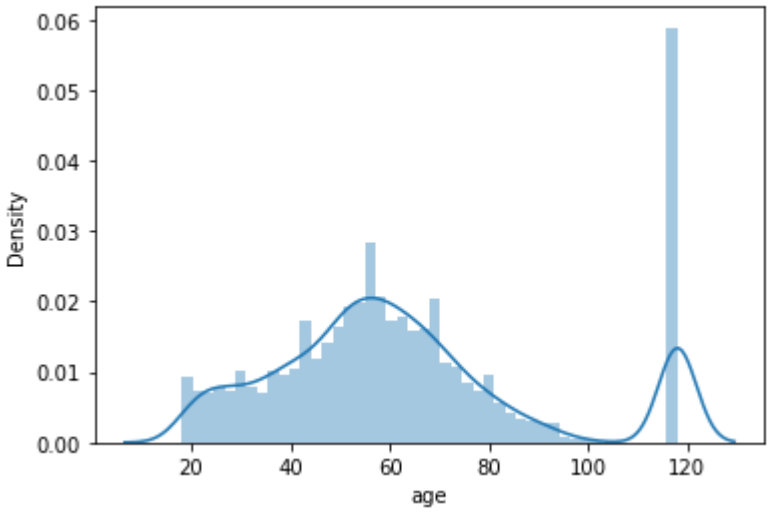
It has (17000 rows and 5 columns)

1) we have group called O apart from M and F

2) two columns data is missing it will higly vraince to impute data for missing data,gender and income missing in the same rows

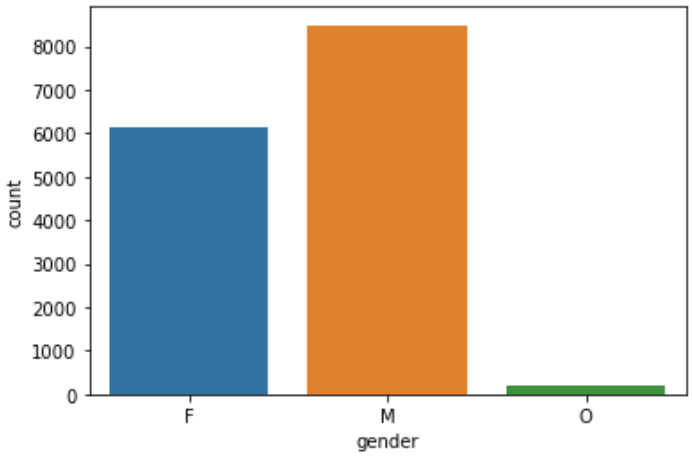
3) we will remove data for nan values.

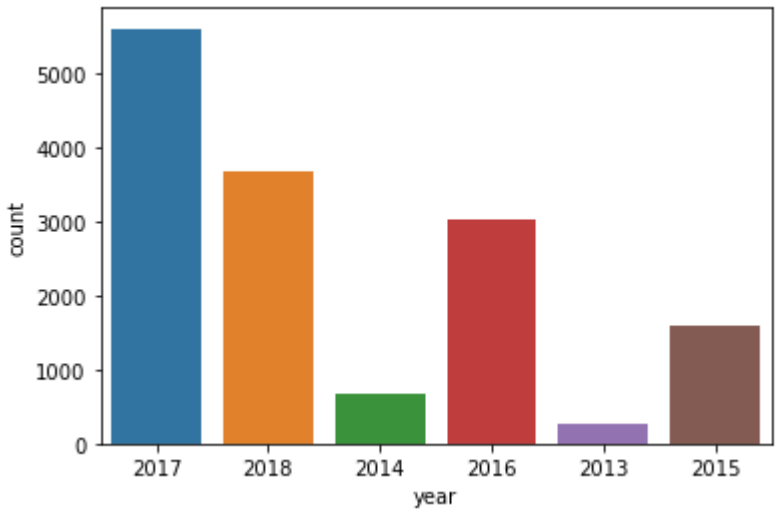
Age plot shows we have people above agae of 18 age and we have oulier around 118



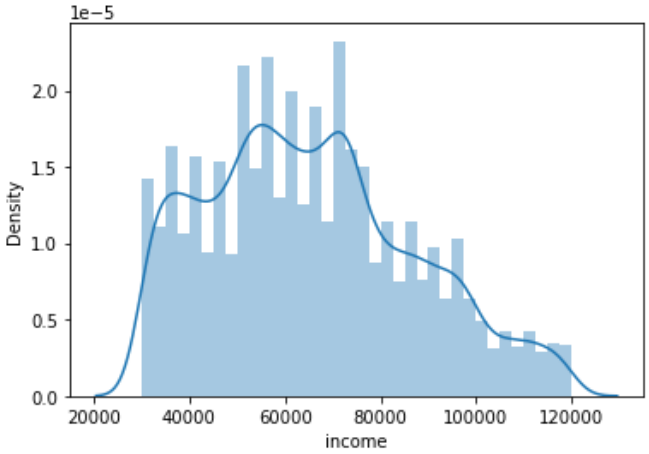
Count of Man is more than Female and ‘O’ in our dataset

Number of

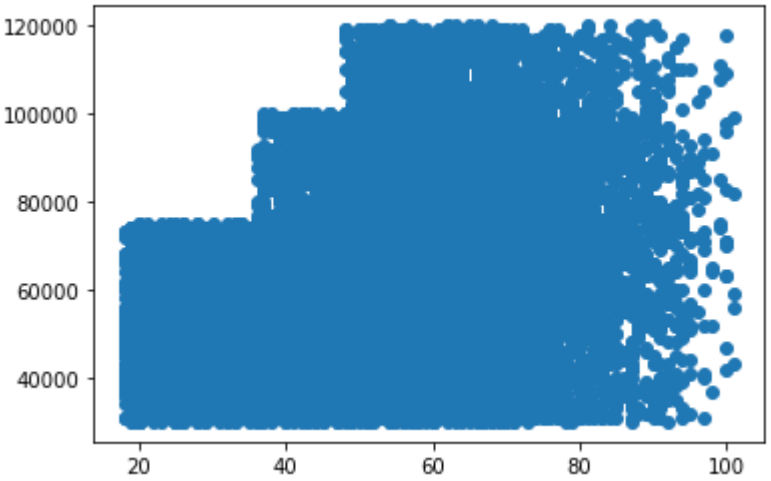




Income

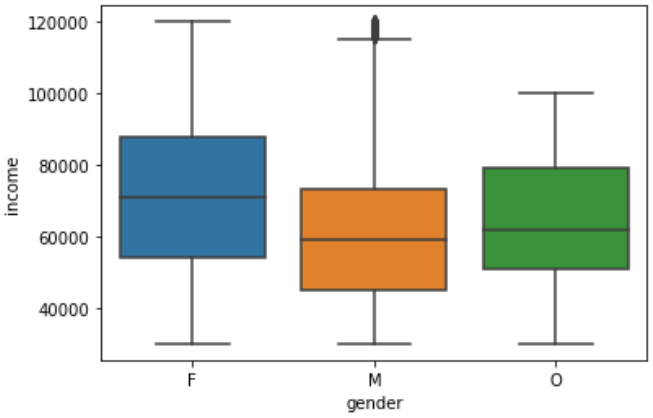


Age and income



Income and Gender

Female have higher mean income than Man and ‘O’



Transcript

1) It has 306534 rows and 4 columns

2) What we see is offer_id columns had two details one is for offer_id other is the amount spend as part of transaction

Algorithm:

I will be use Random Forest as Algorithm to model the dataset because RF does not need any normalization or feature scaling of the input data.

Benchmark:

We have used random search to get some benchmark around modeling

Methodology

Data Preprocessing:

Using below code snippet I have divided the details into two columns and ‘offer_id’ and ‘amount’.

```
transcript['offer_id'] = [[*i.values()][0]if [*i.keys()][0] in ['offer id', 'offer_id'] else None for i in transcript['value']]
transcript['amount'] = [[*i.values()][0]if [*i.keys()][0] in ['amount'] else None for i in transcript['value']]
transcript.rename(columns= {'person':'id'},inplace=True)
transcript=transcript.drop(['value'],axis=1)
transcript.head()
```

Created a clean_data func is used to derive the offer, which users received, viewed, and completed in sequential manner.
Then try to find out the records and only keep the offer_received rows so th at we can analyze the data
Creating a decision column we can track the offers which we completed by the customers. Further used that as our dependent variable for training.
Have removed the ('id','event','offer_id','email','amount') from the data
‘id’ is unique variable
‘event’ – we have divided this column into further columns
‘offer-id’- unique variable was used to join the different data sets
‘Email’- is unique value of 1
‘amount’ – it doesn’t give much meaning as per our model creation

We have merged all the three input datasets and store the output under inter mediate folder for training

Implementation:

I have used random forest as we have unbalanced datasets, random will help us to not to overfit the data.

Refinement:

1) I have tried to create a model independent of the type of offer user gets. By this I mean we can future segregate the model to tune to find the offer specific to the customers so that customer can go had buy the product. I could have trained separate model for the specific offer, which can engage specific customers.
2) We can also try to build model to capture the customer behavior over the time as we data of customer reacting to offers for specific timeframe, for that I could have used RNNs
3) For the specific to model, which I have trained could I have used other machine learning models like XG, boost with hyper parameters

Results

Model Evaluation and Validation:

Getting a F1-score of ‘82’ in test datasets.

Justification:

F1 score indicates that we have a reliable model