Machine Learning Engineer Nanodegree

Capstone Project

Yusuke Kawanabe December 31st, 2050

I. Definition

Project Overview

Nowadays Understanding user behavior and taking actions based on data is a key for customer success and profitability. Starbuck coffee, a coffee chain in America has successfully developed a mobile application platform to achieve this. Once every few days, Starbuck sends out an offer to users of the mobile app. An offer can be merely an advertisement fo a drink or an actual offer such as a discount or BOGO (buy one get one free). Some users might not receive any offers during certain weeks.

Problem Statement

It is important for marketers to be able tell how well an offer will perform in order to run effective campaigns. This problem can break down to multiple sections, such as choosing the right parameter for a target audience, finding the audience that can give the most outcome, analyzing user behavior after they used an offer, predicting how many user will use a certain offer. In this project, I will build a model which can give prediction on wether or not a user will complete the offer or not by using machine learning predictors. The model is going to be binary classifier because the outcome we want to expect is binary, if a user will complete the offer or not. The model will take an offer as input and gives how much percent of users will complete the offer. In order to achieve this, I will first explore the dataset to have better understanding. Secondly, I will clean up the dataset so that machine learning models can utilize the data, which includes normalization. Once the dataset is ready, I will use XGBoost as a benchmark model. XGBoost is an optimized distributed gradient boosting library, which AWS Sagemaker estimator provides easy to use support. Finally, I will use other models, such as the support vector machine, the logistic regression, the k-nearest neighbors vote, to compare the performance suited for our purpose to predict offer completion.

Metrics

I will use ROC-AUC as a main metric. This is a graphical plot using the true positive rate (TPR) and the false positive rate (FPR) at various threshold to illustrate optimal model. ROC-AUC is suited for balanced data because TPR and FPR only depends on positives and if the dataset is imbalanced ROC-AUC won't capture precision. In this case, dataset is balanced as shown in data exploration section and we can use ROC-AUC as the main metric. Additionally, accuracy with validation data will be accounted as main metrics.

II. Analysis

Data Exploration

The dataset is provided by Udacity and downloaded to my project. There is 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)

There is three kinds of offer type:

- bogo
- informational
- discount

For channels, we have below as value:

- web
- email
- mobile
- social

Portofolio data contains 10 rows and the data looks like below:

	channels	difficulty	duration	id	offer_type	reward
0	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	bogo	10
1	[web, email, mobile, social]	10	5	4d5c57ea9a6940dd891ad53e9dbe8da0	bogo	10
2	[web, email, mobile]	0	4	3f207df678b143eea3cee63160fa8bed	informational	0
3	[web, email, mobile]	5	7	9b98b8c7a33c4b65b9aebfe6a799e6d9	bogo	5
4	[web, email]	20	10	0b1e1539f2cc45b7b9fa7c272da2e1d7	discount	5
5	[web, email, mobile, social]	7	7	2298d6c36e964ae4a3e7e9706d1fb8c2	discount	3
6	[web, email, mobile, social]	10	10	fafdcd668e3743c1bb461111dcafc2a4	discount	2
7	[email, mobile, social]	0	3	5a8bc65990b245e5a138643cd4eb9837	informational	0
8	[web, email, mobile, social]	5	5	f19421c1d4aa40978ebb69ca19b0e20d	bogo	5
9	[web, email, mobile]	10	7	2906b810c7d4411798c6938adc9daaa5	discount	2

Sample of protfolio.json

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

The data sample looks like below:

	age	became_member_on	gender	id	income
0	118	20170212	None	68be06ca386d4c31939f3a4f0e3dd783	NaN
1	55	20170715	F	0610b486422d4921ae7d2bf64640c50b	112000.0
2	118	20180712	None	38fe809add3b4fcf9315a9694bb96ff5	NaN
3	75	20170509	F	78afa995795e4d85b5d9ceeca43f5fef	100000.0
4	118	20170804	None	a03223e636434f42ac4c3df47e8bac43	NaN

Sample of profile.json

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

This json file contains 4 types of events. Those events represents key events for offer completion.

- transaction
- offer received
- offer viewed
- offer completed

Each of these has different dictionary value set in the value column. For example, for transaction event, the dictionary contains amount of money transacted, and for offer completed, it's offer id.

The data has 306534 rows and the data sample looks like this.

]:

	event	person	time	value
0	offer received	78afa995795e4d85b5d9ceeca43f5fef	0	{'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}
1	offer received	a03223e636434f42ac4c3df47e8bac43	0	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}
2	offer received	e2127556f4f64592b11af22de27a7932	0	{'offer id': '2906b810c7d4411798c6938adc9daaa5'}
3	offer received	8ec6ce2a7e7949b1bf142def7d0e0586	0	{'offer id': 'fafdcd668e3743c1bb461111dcafc2a4'}
4	offer received	68617ca6246f4fbc85e91a2a49552598	0	{'offer id': '4d5c57ea9a6940dd891ad53e9dbe8da0'}

Sample of transcript.json

Exploratory Data Analysis

portfolio.json

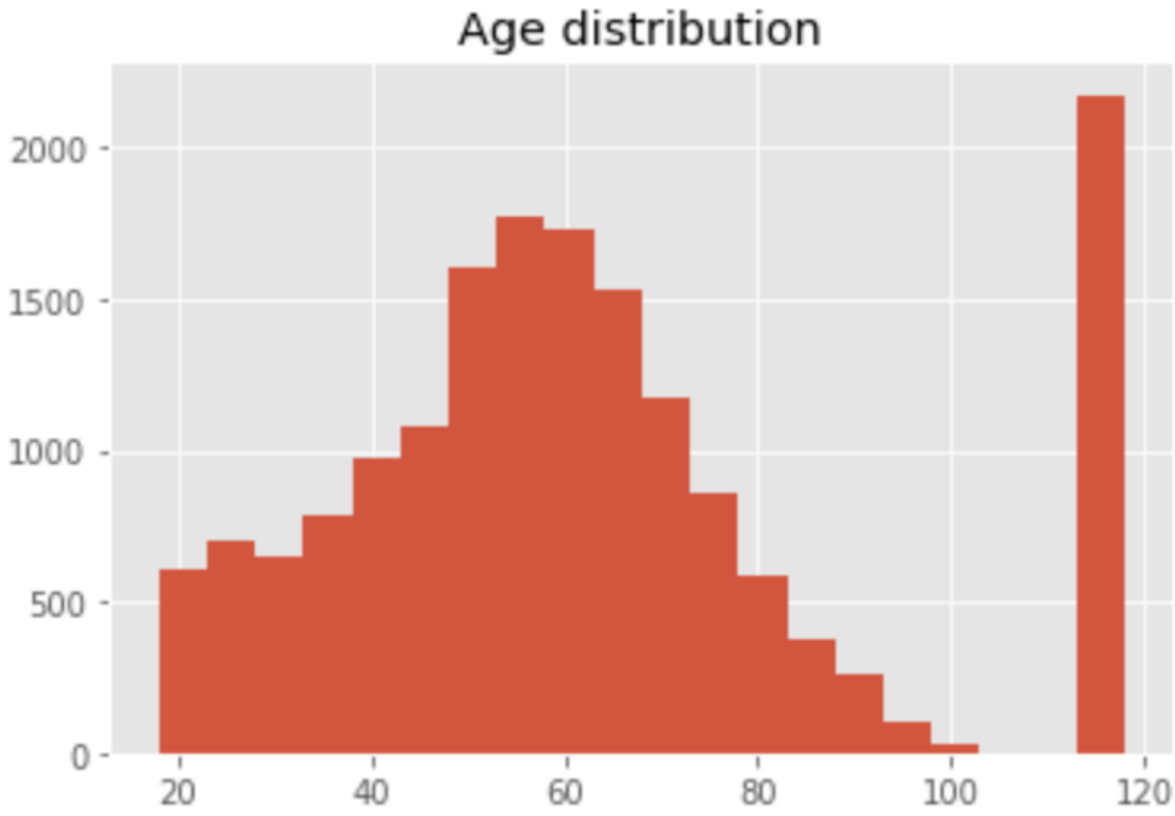
This data only contains 10 rows.

Channel row contains arrays of strings, and this is not machine learning model friendly, although it is expected that different channel have different influence on users. I will have to separate them to separate columns.

profile.json

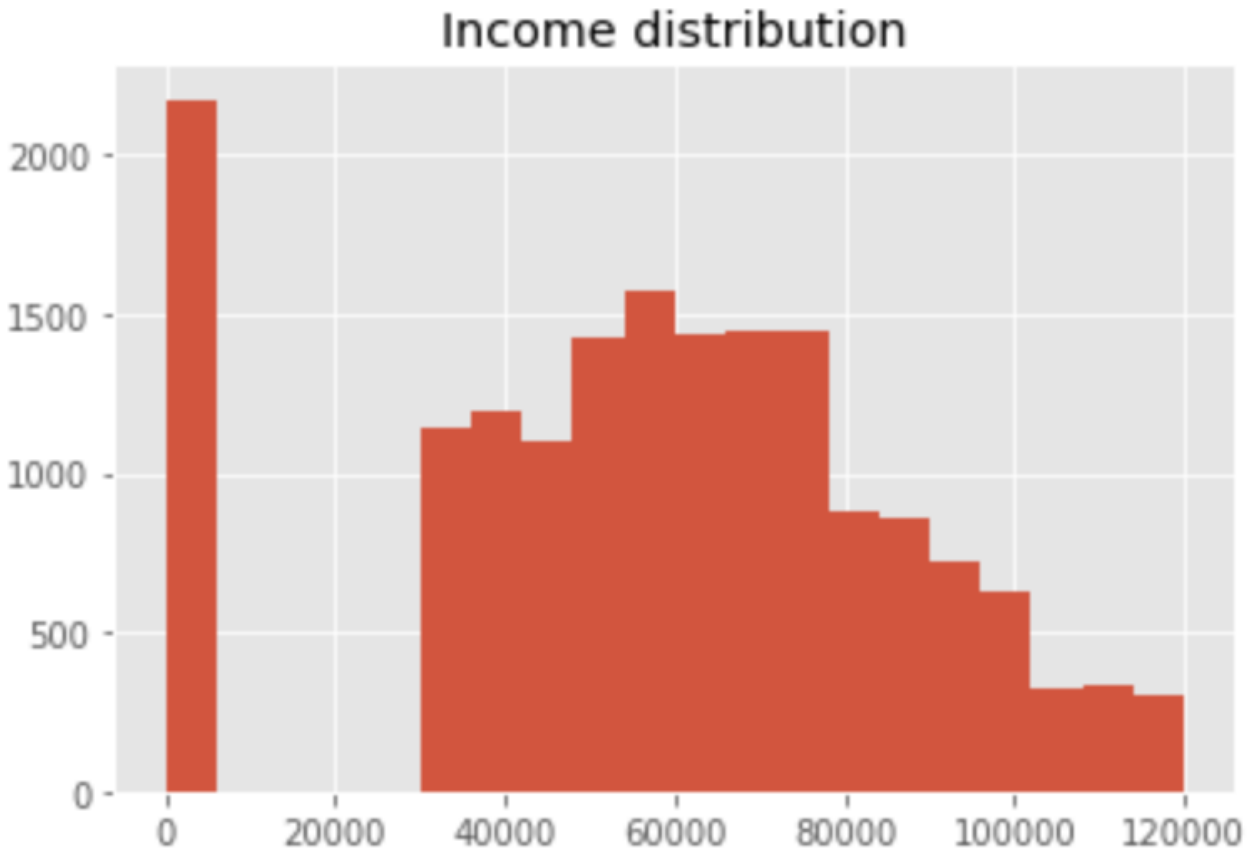
We have 17000 customer data set. The data seems incomplete because some data contains 118 for age, None for gender and NaN for income. Let's take a look at distribution of each features.

Age distribution is shown below. This shows people who is over 118 years old is the biggest population, which seems not realistic given average life expectancy in the U.S. is 72 years old. It is most likely the default value set by the system. I will replace this value with the average in the data preprocessing phase.



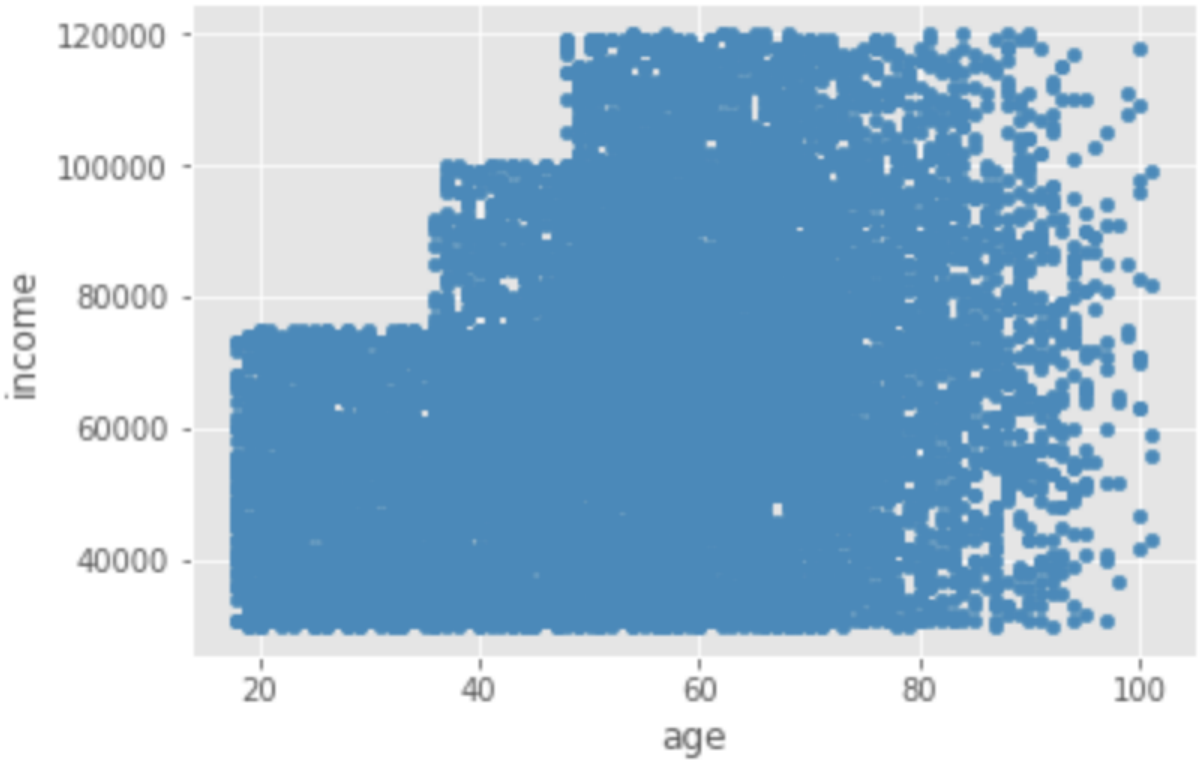
Age distribution in profile.json

The income distribution is shown below, and it seems reasonably distributed given it looks like normal distribution except for NaN rows. I will replace this NaN values with the average.



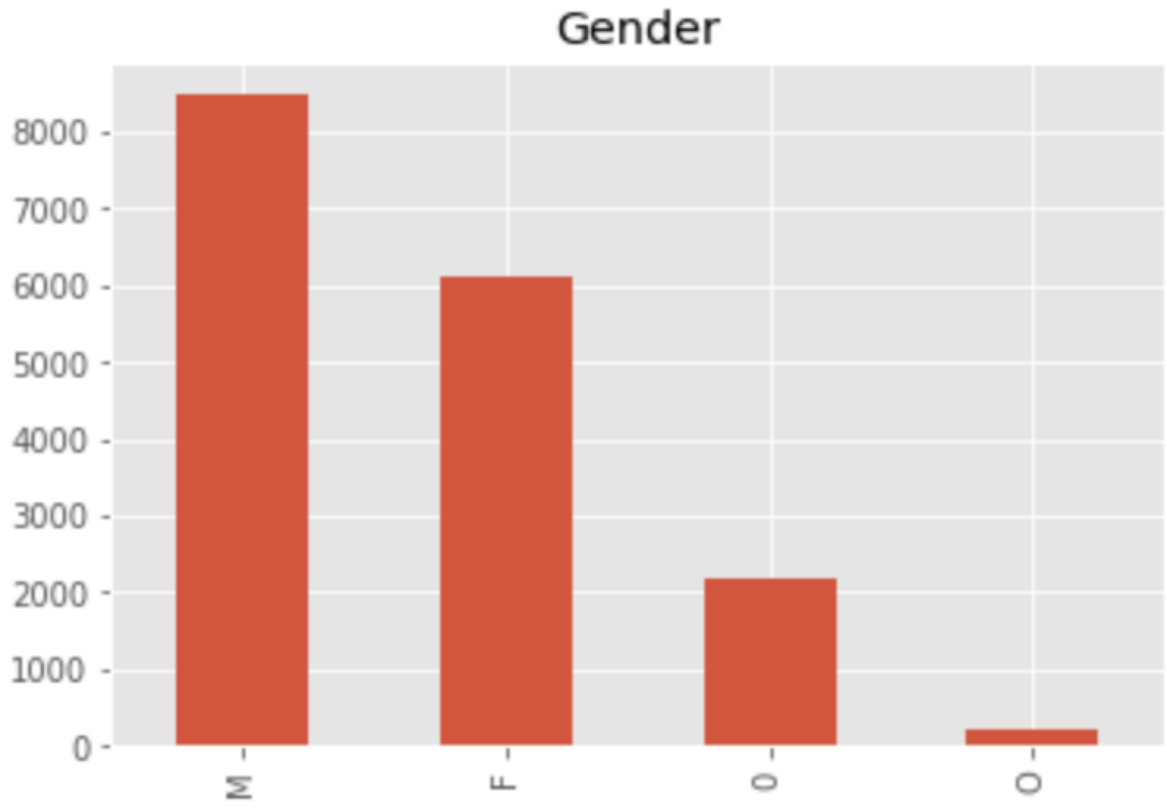
Income distribution in profile.json

The image above is a scatter plot with income and age. This highlights that there is income cut off for each age groups. For example, people in 20s and 30s have income cutoff somewhere around 750K. This indicates that the profile data is arbitrarily created. This may make prediciton simpler because user segmentation is clearler.



Income and age scatter plot from profile.json

The chart below is gender distribution from profile.json. This shows gender column has some anomally values, 0 and O. I made an assumption they are default vaue or indicates user has not selected gender.

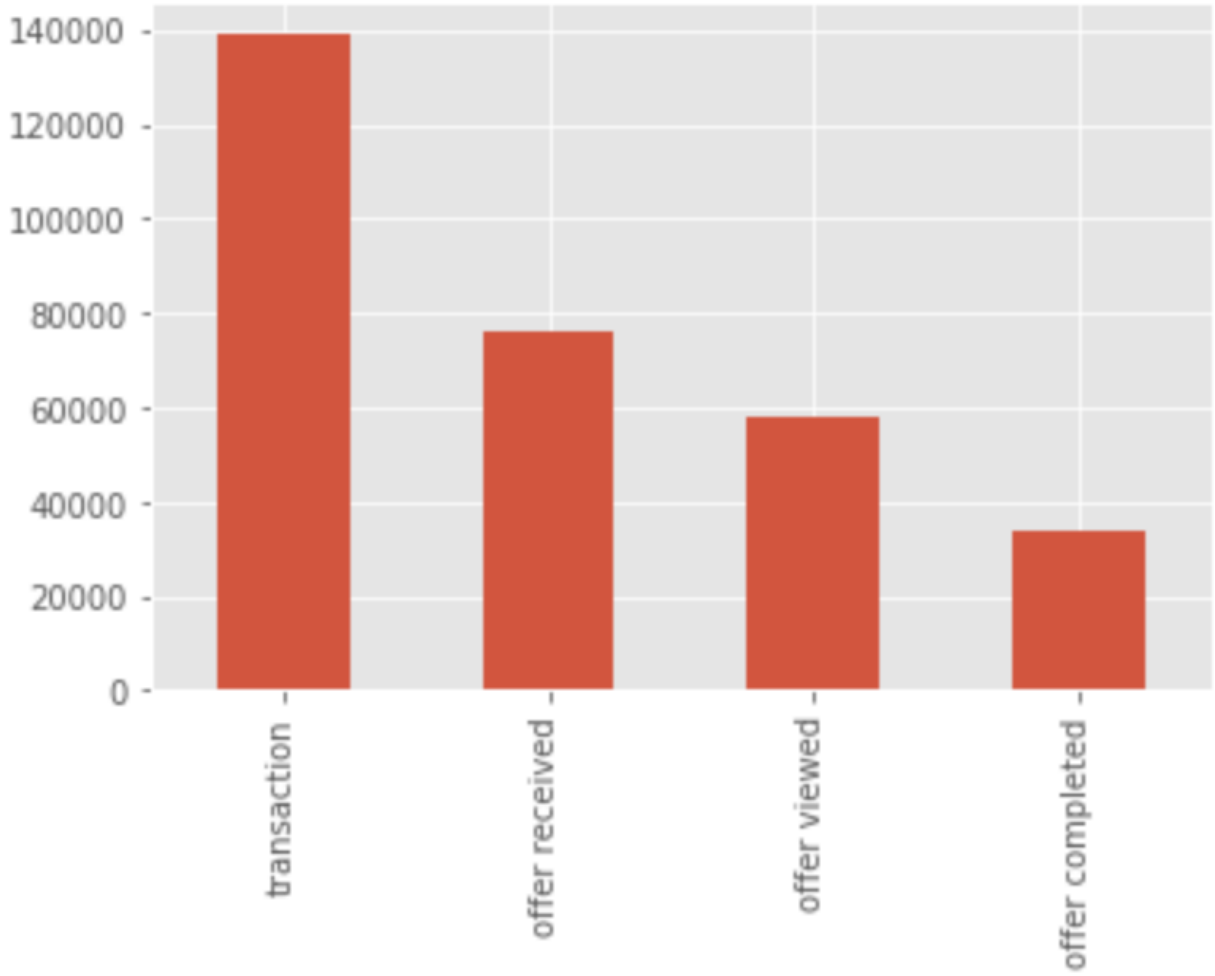


Gender distribution in profile.json

transcript.json

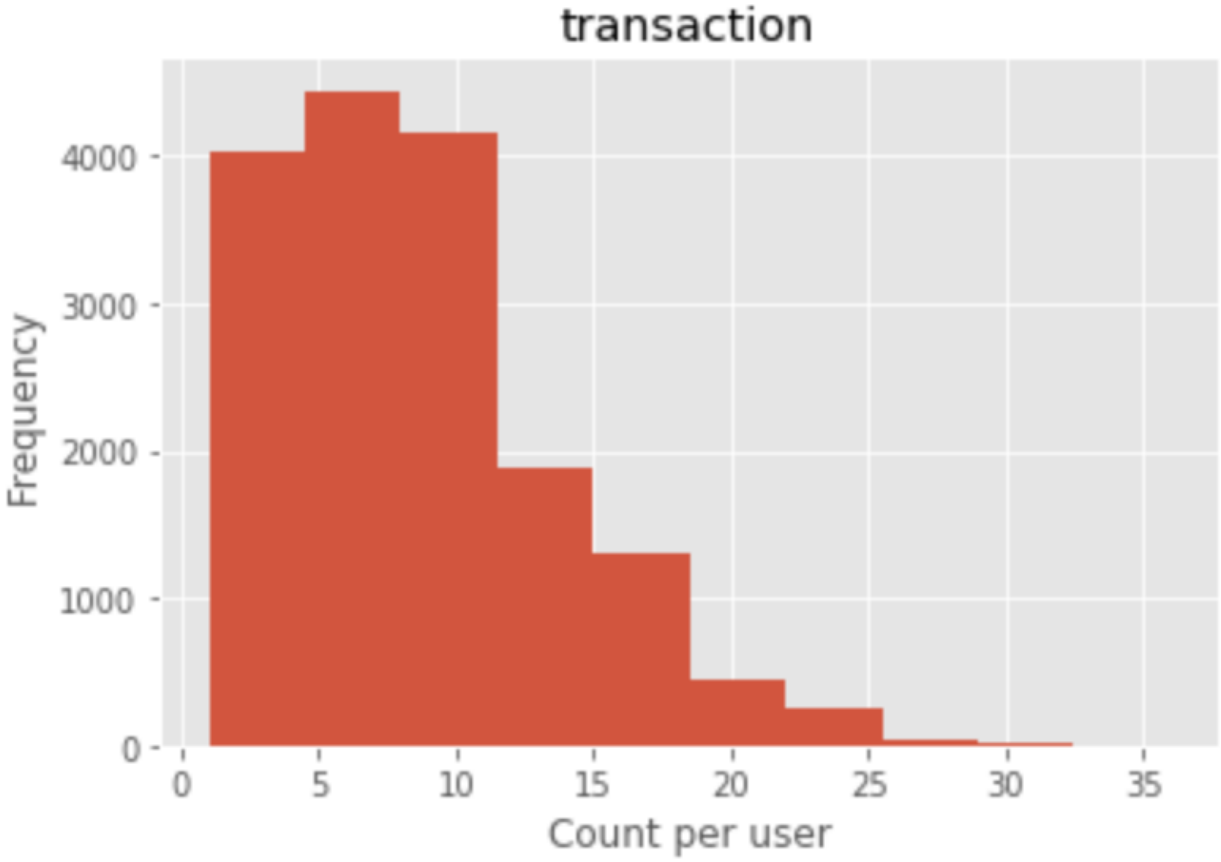
The data has 306534 rows. This transcript data contains for types of events and each event has different values. Therefore, we will need to separate them during data preprocessing.

Let's first take a look at number of each event as shown below. As expected, the number of events decrease as offer funnel proceeds from offer received to offer completed. Additionally, although transaction has the biggest data, this event should be treated differently because the nature of the event is different from other events.



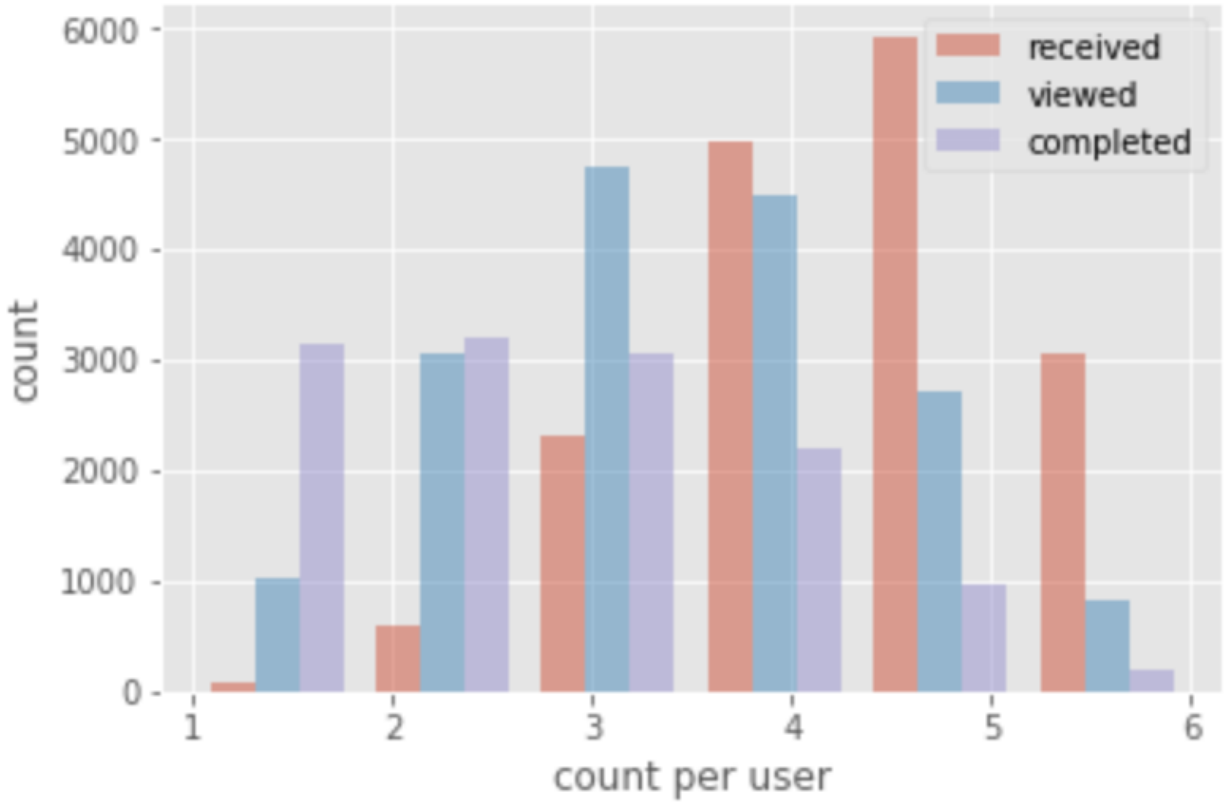
Event distribution in transcript.json

Here is distribution of transaction count per user. More users had a few transaction and less users had many transactions as expected.



Distribution of transaction count in transcript.json

The box graph below shows distribution of offer event including received, viewed and completed per user. As expected number is skewed high for the received, middle for the viewed and low for completed. This indicates that offer events has very realistic distribution. Additionally, the number of offer received data is 76277 and the number offer completed is 33579. This indicates that data is balanced.



Distribution of offer event count in transcript.json

Algorithms and Techniques

The goal for this project is to come up with a model that can accurately predict wether user will complete the offer or not. This is supervised binary classification problem.

I will use the logistic regression model, multi layer perceptron, random forrest classifier, the k-nearest neighbors vote, support vector machine and AdaBoost classifier.

Logistic regression is a probability model which is used for 0/1 problems. This model is used in variety of field such as medical and engineering.In this model, sigmoid function is used. It takes takes in user profile and offer data as input and gives an steep curve at which y value changes. an arbitral threshold is selected and depends on the threshold you can change sensitiveness of the model. I will use `LogisticRegression()` in sklearn for this problem.

Multi layer perceptron is consists of multiple layer of perceptorons as the name suggests. Perceptron is mathematic model represented by an activation function such as a step function or a sigmoid function, which sends output to the next layer if it hits certain threshold. The benefit of the model is it can learn non-linear models, which allows us to create complex models. However, this is sensitive to to feature scaling. Hence, we will normalize the data in preprocessing. `MLPClassifier()` in sklearn is used in this paper.

Random forrest classifier is a classification method uses multiple decision trees from randomly selected subset of training data. Sometimes simple random forest classifier performs better than other complicated models. `RandomForestClassifier()` from sklearn is used in the implementation.

The k-nearest neighbors vote classifies using majority vote of it's neighbors, with the object being assigned to the class most common to its k-nearest neighbor. The main feature of this model is being non-parametric method.

This makes the model suitable for data with a lot of unknowns and outliers. `KNeighborsClassifier()` from sklearn will be used in the implementation.

Additionally, support vector machine is a classifier model which finds a hyper plane which separates one or the other of two categories. It is represented as `SVN()` in sklearn.

And finally AdaBoost classifier is a classifier used in conjunction of bunch of weak learners. The benefit of the class is this model provides relatively accurate predictions without much of parameter tuning. However, AdaBoost is sensitive to outlier and noisy data. `AdaBoostClassifier()` class from sklearn is used in this project.

Benchmark

As benchmark, I used XGBoost binary classifier provided on Sagemaker. I ended up using XGBoost because the model can be easily accessed on AWS. Additionally, it is known for providing good predictions and won some Kaggle competitions.

The parameter of predictor is defined as below:

```
xgb.set_hyperparameters(max_depth=5,
                        eta=0.2,
                        gamma=4,
                        min_child_weight=6,
                        subsample=0.8,
                        silent=0,
                        objective='binary:logistic',
                        early_stopping_rounds=10,
                        num_round=500)
```

20 % of data is used for test data to calculate accuracy score of the model, which turned out 0.795.

III. Methodology

(approx. 3-5 pages)

Data Preprocessing

In this section, all of your preprocessing steps will need to be clearly documented, if any were necessary. From the previous section, any of the abnormalities or characteristics that you identified about the dataset will be addressed and corrected here. Questions to ask yourself when writing this section:

- *If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented?*
- *Based on the **Data Exploration** section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected?*
- *If no preprocessing is needed, has it been made clear why?*

Implementation

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

- *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?*
- *Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?*
- *Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

Refinement

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

- *Has an initial solution been found and clearly reported?*
- *Is the process of improvement clearly documented, such as what techniques were used?*
- *Are intermediate and final solutions clearly reported as the process is improved?*

IV. Results

(approx. 2-3 pages)

Model Evaluation and Validation

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model's solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

- *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?*
- *Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?*
- *Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?*
- *Can results found from the model be trusted?*

Justification

In this section, your model's final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

- *Are the final results found stronger than the benchmark result reported earlier?*
- *Have you thoroughly analyzed and discussed the final solution?*
- *Is the final solution significant enough to have solved the problem?*

V. Conclusion

(approx. 1-2 pages)

Free-Form Visualization

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

- *Have you visualized a relevant or important quality about the problem, dataset, input data, or results?*
- *Is the visualization thoroughly analyzed and discussed?*
- *If a plot is provided, are the axes, title, and datum clearly defined?*

Reflection

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

- *Have you thoroughly summarized the entire process you used for this project?*
- *Were there any interesting aspects of the project?*
- *Were there any difficult aspects of the project?*
- *Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?*

Improvement

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section:

- *Are there further improvements that could be made on the algorithms or techniques you used in this project?*
- *Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?*
- *If you used your final solution as the new benchmark, do you think an even better solution exists?*

Before submitting, ask yourself. . .

- Does the project report you've written follow a well-organized structure similar to that of the project template?

- Is each section (particularly **Analysis** and **Methodology**) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
- Would the intended audience of your project be able to understand your analysis, methods, and results?
- Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?
- Are all the resources used for this project correctly cited and referenced?
- Is the code that implements your solution easily readable and properly commented?
- Does the code execute without error and produce results similar to those reported?
-

Reference

- XGBoost Documentation
- ROC and AUC, Clearly Explained!
- Receiver operating characteristic
- Logistic regression
- scikit-learn
- Machine Learning101
- AdaBoost