**Capstone Project**
**Machine Learning Engineer Nanodegree**

**Rukmini Deshpande**
**February 21st, 2022**

# Starbucks Project – Customer behaviour, demographic traits analysis and Prediction

## Content:

1. **Domain Background:**

This project is about Starbucks offers and how their customers interact with these offers. Data set given contains simulated data that mimics customer behaviour on the Starbucks rewards mobile app. Once every few days, Starbucks sends out an offer to users of the mobile app. Starbucks sometimes sends individual messages that contain offers related to its products. These offers can be an advertisement for a new product or an offer such as:

BOGO (buy one get one): User needs to reach to a certain amount to earn a reward of the same amount.
Discount: Discount is a fraction based on order price.

Not all users receive the same offers, sending offers to customers who are not likely to buy their products. On the other hand, we need to attract the new customers, so it is necessary to identify who are the people with a higher probability to respond to that specific offer we send to them. Every offer has a validity period before the offer expires. As an example, a BOGO offer might be valid for only a week. In this project, we will combine transactions data, demographic data and offers to train a model that given a customer and an offer tries to predict whether that customer will respond to that offer. Responding to an offer means viewing the offer and then making the required transactions to complete it while it is active.

2. **Project Overview:**

Promotions play an important role in driving sales. Having a discount when purchasing means saving more money for the customers. However, to those who do not make purchases based on promotions, getting excessive offers means stuffed mailbox. Different people respond to certain offers in different ways. Offers should not be sent to the population as a whole, but at an individual personalized level. In this project we will be using machine learning to make better choices about which offers to send.

3. **Problem Statement:**

Starbucks invests money in offers expecting to have higher profits in back. 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. So, selecting the most relevant offer to the correct customers is important to achieve a successful marketing campaign. However, some customers don't buy anything. Other ones don't even see the offer that was sent to them, which may be a problem with the channel chosen.
So, we have four categories:
1. Offer will not be seen nor ordered.
2. Offer will not be seen but accidently ordered.
3. Offer will be seen but not ordered.
4. Offer will be seen and ordered.

What might be a problem with the
1. offer type sent
2. The customer is not the one to be considered as a target.

4. **Datasets and Inputs:**

Our datasets here are stored in 3 ". json" files, and they are as follow:

1. portfolio.json:
    1. id (string) - offer id
    2. offer type (string) - type of offer is BOGO (4 records), discount (4 records), informational (6 records)
    3. difficulty (int) - minimum required spend to complete an offer
    4. reward (int) - reward given for completing an offer
    5. duration (int) - time for offer to be open, in days
    6. channels (list of strings)
    7. This dataset contains 10 rows and 6 columns

2. profile.json:
    1. age (int) - age of the customer
    2. became_member_on (int) - date when customer created an app account
    3. gender (str) - gender of the customer (M or F) (14825 M, 14825 F and 14825 not specified) so the data here in balanced
    4. id (str) - customer id
    5. income (float) - customer's income.
    6. This dataset contains 17000 rows and 5 columns.

3. transcript.json:
    1. event (str) - record description (ie transaction, offer received, offer viewed, etc.)
    2. person (str) - customer id
    3. time (int) - time in hours since start of test. The data begins at time t=0
    4. value - (dict of strings) - either an offer id or transaction amount depending on the record
    5. This dataset contains 306534 rows and 4 columns

We will upload these data into S3 bucket to use effectively with Sagemaker. Dataset will be feed to model for training purpose, while test dataset will be used for validation during the training.

4. **Solution Statement:**

This project will be done using XGBoost algorithm which is one of the best algorithms in Machine Learning Algorithms. The whole project will be done using the Amazon Web Services (AWS) specially Amazon Sage Maker service that I have learned to use during this nanodegree.

Our objectives for this paper are:

1. Determine what demographic traits, if any, influence purchasing behaviour.
2. Determine how well we can predict consumer behaviour after they've viewed a coupon.

**5. Evaluation Metrics:**

The metric used in this project will be accuracy. Accuracy is defined as the total of correctly predicted labels divided by the total of labels in the test dataset. The evaluation metric can be used to quantify the performance of both the benchmark model and the solution model. In this project I will use ROC-AUC. Higher the AUC value for a classifier, the better its ability to distinguish between positive and negative classes.

**6. Project Design:**

This project will be completed with the following steps:

1. Data processing and exploration:
    a. First will create a Sagemaker notebook instance with fair performance.
    b. Some Exploratory Data Analysis should be applied on the data. This helps us deleting all null values to avoid any errors that we might face after. Also, EDA helps us to find patterns in the data.
2. Data Feature Engineering: with this step, we do some statistical features for every data point.
3. Data Splitting: at this step we will split the data into three datasets:
    a. Train Data: to train the model on it.
    b. Validation Data: to validate the model and increase its performance.
    c. Test Data: this dataset is used to calculate the accuracy of the model on data that the model has not seen before.
4. Train the model: Using sagemaker tuner, we will tune model with best possible hyperparameters and when it is obtained, and when it is obtained will train model to a sagemaker estimator. Here will need to calculate the most suitable hyperparameters for tuning the model.
5. Test the model: this will be the last step in our project.

**7. Pre-processing:**

**Data exploration:**

The dataset was obtained from Udacity as part of the Machine Learning Engineer Nanodegreeand the explanations given for each feature are below. I also included a sample of the 5 top rows from each dataframe to help you visualize its structure.

The data is contained in three files:

1. portfolio.json - containing offer ids and meta data about each offer (duration, type, etc.)

2. profile.json - demographic data for each customer

3. transcript.json - records for transactions, offers received, offers viewed, and offerscompleted

***Portfolio.json (10 X 6)***

| | 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 |

***Profile.json (17000 X 5)***

| | 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 |

***Transcript.json (306534 X 4)***

| | 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'} |

First thing that about the data is, in the profile dataframe, there are some **NaN** values in the gender and income columns. With the ***df.isna.sum()*** command we check all the dataframes and we see that profile is the only dataframe with NaN values.

The age description is not accurate because people who did not input age are

considered age 118. So, we need to apply some data cleaning on all of the rows in the above picture with NaN values having unusual age. Upon checking the count of rows with that age we see that it is 2175, the same number of NaN values in the dataframe. Meaning some customers didn't to give their information or certain information when registering as members, so the income and gender got NaN values, but the age defaulted to 118.

Another problem with the current data is that there are categorical values like gender, channels and offer type. We can't feed our model text data, so we will have to convert it into numbers. Also, when there are more than two categories we will convert that to one-hot- encoding.

Lastly, the date the members became member on is an integer instead of a date.

**Visualization:**

The distribution of age before dropping the NaN rows, the outliers are obvious. Among the 17000 members, the distribution of ago peaks at 50 - 60, with a left shoulder at 20-30 years.

(Please refer figure 1, 2)

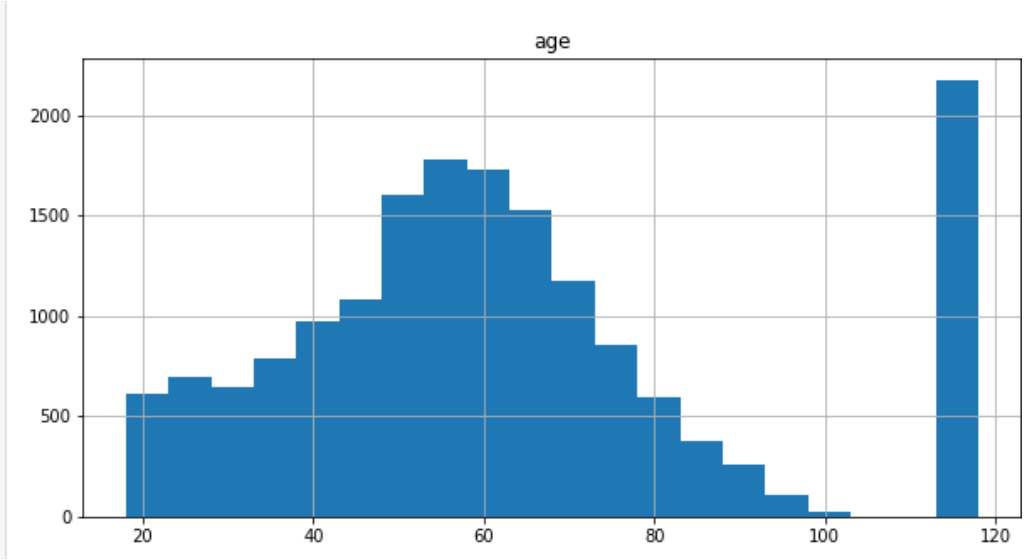



Fig. [1]

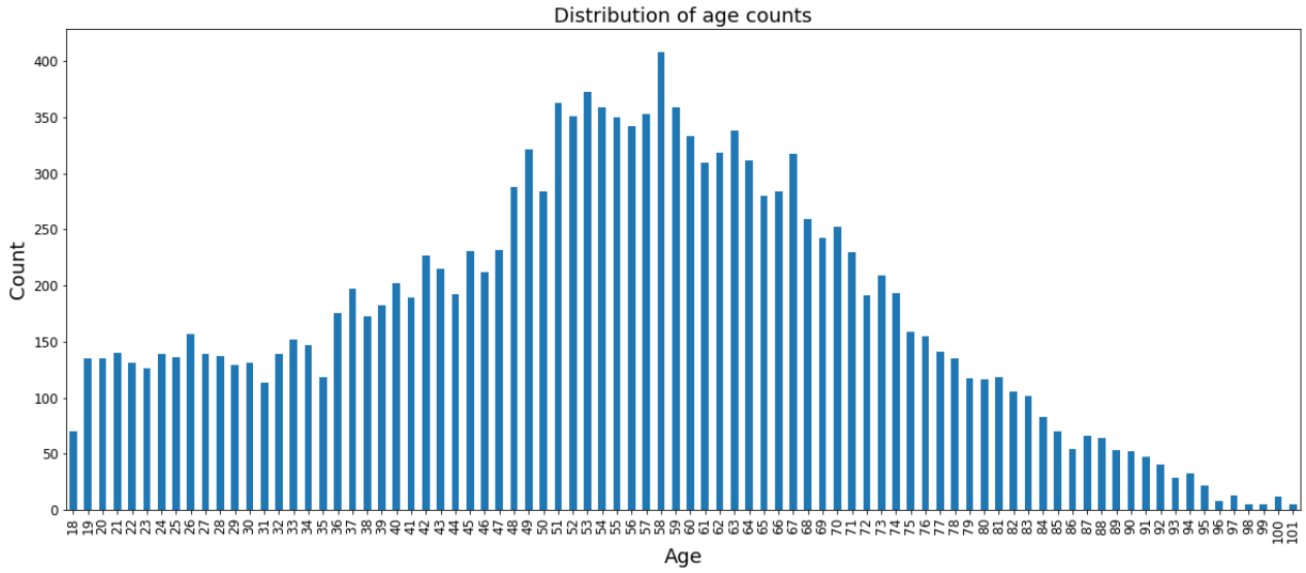



Fig. [2]

Distribution of the number of days that customers have been members. We can see that mostcustomers are relatively new. This might mean that they might not be as aware of offers as older users. (Figure 3 – member_for_days)

The salary is more evenly distributed with a range from $30, 000/year to more than $100,000/year, meaning Starbucks is loved by customers regardless of income. (Figure 3 - income)

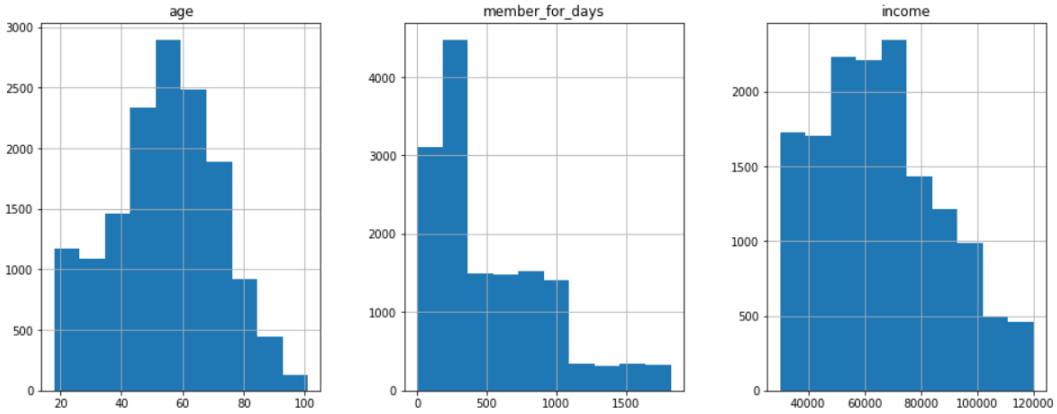



Fig. [3]

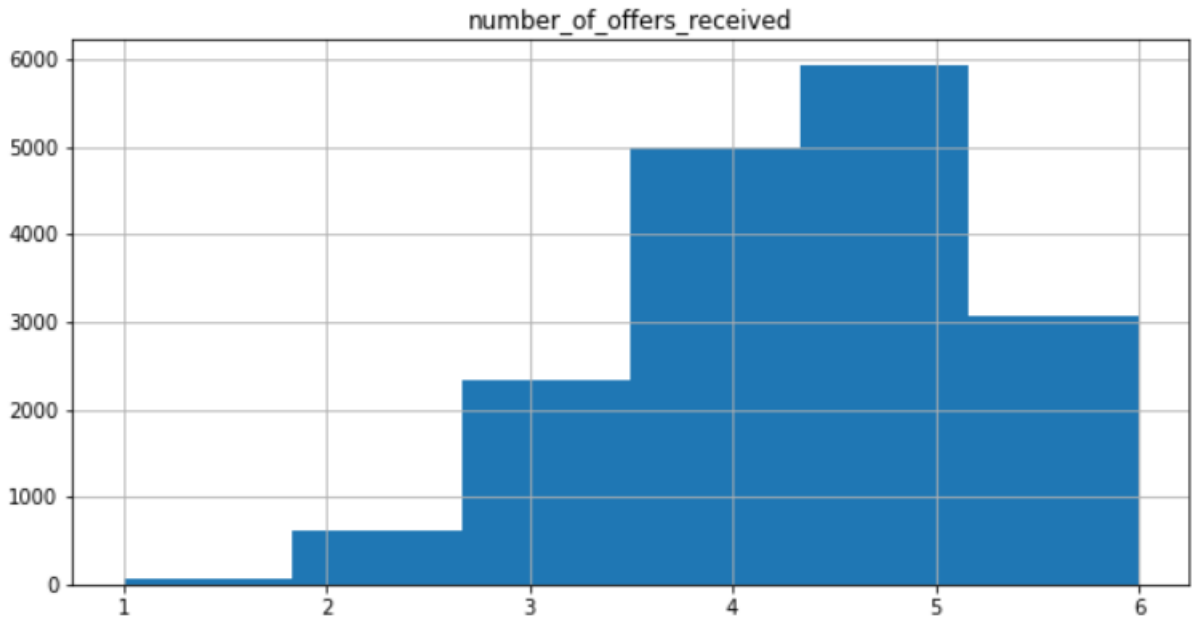


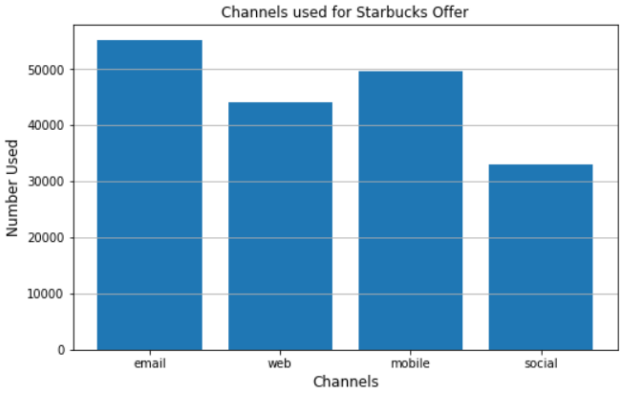

Fig. [4]

Distribution of number of offers sent to each customer. (figure 4)

Distribution of offers sent by offer_id. The axes are reversed but if you look at the x values you see that all of them have been sent a similar amount of times. (figure 5)

Looking at transaction distributions among different age groups we see older people spend more than the younger people. However, this is not the case when there is no offer.
This probably indicates the older generation is wiser with money and will not spend unless there is an offer. However, again the distribution among different offer attributes of each age group does not have high variability.

Fig. [5]



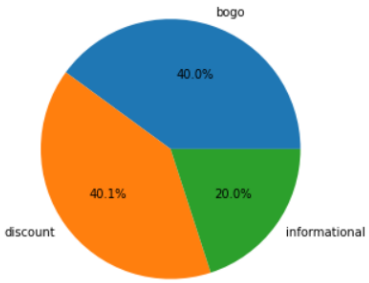




Fig. [6]

The most preferred method for sending promotion is through email, then through mobile. And the most preferred type of offer is through discount. (figure 6)
There are ~16% more male members than female members. Most of the customers joined at the year of 2017. Later, we will see if that demographic information can be used to predict offer preference.

## 8. Preparing the training code:

A logistic regression algorithm will be used in this project as a benchmark model, because it is a simple model and I expect CatBoost to be able to beat it. After applying pre-processing (discussed below) and training the model I achieved an accuracy score ofabout 0.825, which will be our target to beat with the CatBoost technique.

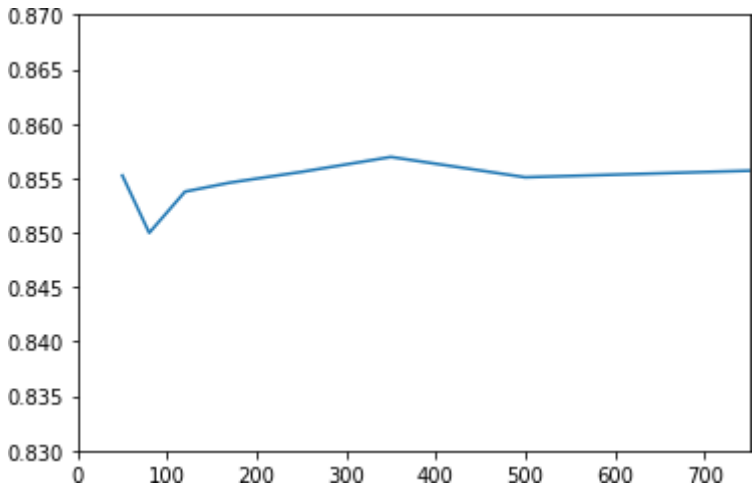
Then the transcript dataframe was looped through and processed to create a new dataframe with the following features for every offer-person combination present in the transcript which are viewed offer, completed offer, responded to offer (if someone did both), total money spent (per person) and viewing, completion and response rate (per person again). That resulting dataframe was merged with portfolio and profile, based on offer and customer id respectively.

The features that were not needed for the model (ids and labels) were dropped and the result was ready for the model. Scaling was applied to all features that needed it.

**Refinement**: I decided to test if the default value of 1000 for iterations was optimal for the problem or not. At first, I decided with bigger values to see if maybe the model was not converging yet. The accuracy did not seem to improve though, so I decided to lower the value to see maybe there was some overfitting going on. That did not seem to be the case either, in fact the accuracy stayed pretty stable with some fluctuations up and down that I assume are due randomness. I decided to cut the by about to a third to reduce the training time.

The graph below shows some samples I took. X axis is the number of iterations and y axis is the accuracy score. Mind the values in the Y axis, the fluctuations are smaller than they appear when on 0-1 axis. I decided on 350 because it peaks here. (Note: I took sample on the

following values: 50, 80, 120, 170, 250, 350, 500, 750)




### 9. Implementation and training:

The CatBoostClassifier class model is pretty straightforward and has a default values for all parameters and the project is simple and standard enough that the default values take care ofthe problem. That being said there is an important parameter, which is iterations. The defaultvalue of it is 1000, but I decided to change it and see how the accuracy would react. In the endI decided on values below (explanation in the next section, refinement):

**Model Evaluation and Justification**:

Iteration = 350 and learning rate = 0.126 (decided by the model based on iteration value)
I tested the model with 33% of data that was not in the training set and got and accuracy of 0.858.

### 10. Conclusion:

The model got an accuracy of 0.858, which means it is more accurate than the benchmark. While there is still some improvement, given a customer and an offer, the model will in most cases predict correctly whether or not the customer will respond to that offer or not, which can be useful for a company.

**Acknowledgment**

This project is fulfilled as a part of Udacity Data Science Nanodegree.

**Code**

GitHub repository with the code and data analysis.

https://github.com/MiniKara-zor-el/Starbucks-Capstone-Project.git

**11. References:**


[1] https://www.researchgate.net/publication/256048420_Machine_Learning_for_Targeted_Display_Advertising_Transfer_Learning_in_Action

[2] https://medium.com/@AmishWarlord/predicting-starbucks-customer-behavior-119fc3a43480

[3] scikit-learn library

[4] Preprocessing with sklearn: a complete and comprehensive guide

[5] matplotlib library

[6] Udacity

[7] https://auto.gluon.ai/stable/index.html

[8] https://en.wikipedia.org/wiki/Automated_machine_learning