Starbucks customer offer success predictor
Nikhil Tank | University of Florida
Udacity Nanodegree – Machine learning Engineer

Domain background

Starbucks started as a single store in Seattle's Pike Place Market in 1971. From the starting the company only celebrated coffee and the rich tradition, and brought the feeling of connection. Starbucks is ranked 227 among the Fortune 500 companies in 2019. The company has more than 31,000 stores across the world and this figure has almost got doubled in the last decade. They have an app as a convenient way to pay in store or skip the line and order ahead. Offering rewards and stars earning free drinks and food with every purchases. The promotion offer can be an advertisement for a drink or an actual offer such as a discount or BOGO (buy one get one free).

Problem statement

The goal of the project is to predicting the influence of rewards and offer on customers based on their response to the previously sent offers, built a machine learning model that determines which demographic groups will respond best to which offer type and duration; resulting better marketing strategies and improving customer reach and satisfaction. Training XGBoost model as a baseline and training PyTorch model and improve the results.

Datasets and inputs

The project uses the data which simulates how people make purchasing decisions and how those decisions are influenced by promotional offers.

Each person in the simulation has some hidden traits that influence their purchasing patterns and are associated with their observable traits. People produce various events, including receiving offers, opening offers, and making purchases. In the dataset three types of files are included:

- profile.json - demographic data for each customer like age, gender and income.
- portfolio.json - containing offer ids and meta data about each offer, such as the duration and the amount a customer' spending required to qualify for an offer. These offers can be delivered via multiple channels and specific time validity of an offer.
- transcript.json - records for transactions, offers received, offers viewed, and offers completed.

```
In [23]: profile.head()
```

Out[23]:

	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

```
In [24]: print("profile: Rows = {0}, Columns = {1}".format(str(profile.shape[0]), str(profile.shape[1])))
profile: Rows = 17000, Columns = 5
```

```
In [8]: portfolio.head(10)
```

Out[8]:

	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

```
In [9]: print("portfolio: Rows = {0}, Columns = {1}".format(str(portfolio.shape[0]), str(portfolio.shape[1])))
portfolio: Rows = 10, Columns = 6
```

```
In [25]: transcript.head()
```

Out[25]:

	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'}

```
In [26]: print("transcript: Rows = {0}, Columns = {1}".format(str(transcript.shape[0]), str(transcript.shape[1])))
transcript: Rows = 306534, Columns = 4
```

As a simplification, there are no explicit products to track. Only the amounts of each transaction or offer are recorded.

There are three types of offers that can be sent:

- Buy-one-get-one (BOGO): a user needs to spend a certain amount to get a reward equal to that threshold amount.
- Discount: a user gains a reward equal to a fraction of the amount spent.
- Informational: there is no reward, but neither is there a requisite amount that the user is expected to spend.

Solution statement

This is a binary classifier problem because the label is in form of 1 and 0. We have data about profile of customers and features like offers and their description but no information about the store transactions.

For binary classification we have options like Random Forest Classifier, Decision Tree Classifier, neural networks, support vector machines. In this project I will be using XGBoost model and NN using PyTorch as solution model.

Benchmark model

I will be using XGBoost model as me benchmark. I have calculated the RO_AUC score and it is 0.7159 for the following Hyperparameters:

- max-depth =5
- eta =0.2

- gamma = 4
- min_child_weight=6
- objective='binary:logitics'
- subsample=0.8

Evaluation metrics

Receiver operation characteristic ROC-AUC will be used for model evaluation. The probabilistic interpretation of ROC-AUC score is that if we randomly choose a positive case and a negative case, the probability that the positive case outranks the negative case according to the classifier is given by the AUC. ROC-AUC score is independent of the threshold set for classification because it only considers the rank of each prediction and not its absolute value. The same is not true for F1 score which needs a threshold value in case of probabilities output. ROC-AUC is used because we don't want to tune the threshold in our model.

Outline of the project design

The step for the execution of the project is as follows:

1) Loading the data into the Jupyter Notebook
2) Cleaning the data as needed
3) Feature Engineering
    a. Determine correlation between the features and reducing them if necessary, using the PCA.
    b. Preparing final data set by reducing the feature and keeping the valuable data
4) Train the XGBoost model in SageMaker
5) Train Deep Learning Model on SageMaker
6) Hyperparamter tuning for the PyTorch model based on the comparison from the benchmark
7) Conclusion