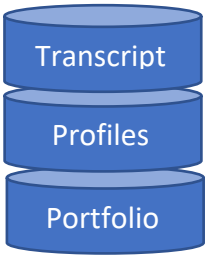
Udacity – Starbucks Project

Evaluation of different Offer types

Data:

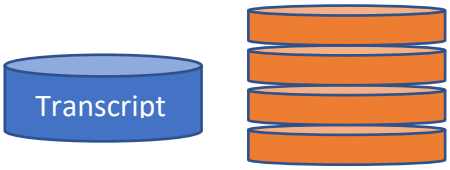
- Transcript (Loging Data of the app)
- Profiles (Info about the customer)
- Portfolio (Info about the Offers)



Split **Transcript** into several Parts

- Transactions
- Offer received
- Offer viewed
- Offer completed

There is an explicit connection between a transaction that completes an offer. The transactions index is one lower than the completion index. (bear that in mind.)



Transactions can describe the user's behavior. **Python's describe tool** is one good way to get the average (explicit the median) spent money that can be **compared with offer-influenced visits**, and to get the counts of visits (=count transactions), as well as the standard deviation.
Transactions are connected with a date, so we can get a count (*'dummie_count'*) on which day of the week the user is more likely to visit a store.

Those columns

- total money spent,
- visit count,
- average money spent,
- standard deviation,
- doW-0, doW-1, doW-2, doW-3, doW-4, doW-5, doW-6

… are merged to the Profiles DF.

CRISP DM (Cross Industrial Process for Data Mining)

1. Business Understanding

The provided Data includes besides the demographic data on each customer also a transcript over roughly one month. In this period offers of different types were sent out to the customers, aiming to get more 'traffic' and therefor more revenue.

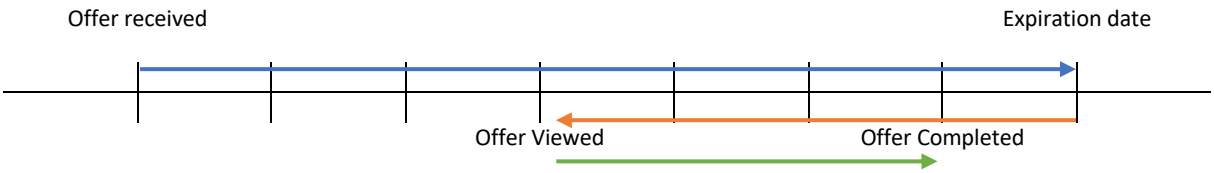
Transcript includes beside the timestamp and the customer-id, an event col, that signalizes whether the offer was sent, viewed or completed, or if it was a 'normal' transaction

2. Data Understanding

Explaining the Transcript, in order to get an influence score on the offer complete transaction:

For the score I am using 5 terms and apply them on each offer sent out.

$$\frac{\text{Bogo + discount Offers viewed}}{\text{Num visits}} * \frac{\text{Offer amount spent}}{\text{Median amount spent}} * \frac{\text{Expiration Time (-2Days)}}{\text{View to expire time}} * \frac{\text{Offer reaction time}}{\text{View to expire time}} * \frac{1}{1 + \text{visits\_count}}$$



In my eyes it is key to get a controlled basis. Whereas the offer received/sent out time is just half controlled, the other half will be fulfilled by the user. I use the Offer viewed as a startingpoint for the analyzes. At this very moment the offer is in kind of a consideration set of the customer (actually since there is no need, I would probably call it different…) and can have influence on the decision, to complete or not to complete the offer. So the difficulty can be measured by keeping an eye on the rest time till the offer expires and the attractivity can be somehow be descripted on a number called reaction time (between the time of view and completion). Those two findings were written in 2 Terms, whereas one of them is linked to the actual completion and will make the whole indication zero if the offer is not completed.

The intention of an offer is to make the customer first of all visiting the store and then make him, her or it spent more money, that he usually would. Talking in statistic terms, I am gathering the transaction data over the period and compare the median amount spent vs. the amount spent on a specific offer. (By the way, the transaction that leads into a completed offer is one record ahead of the event, so it can be linked by ease).

Some Customers do see all the offers, but never or rarely show up at the store.
So why not bring the visits into account on the calculation. By setting the offers viewed count (discount and Bogo only – informational cannot be completed) relative to the count of visits I'll get on the one end a very small number – where the visits are greater than the offers viewed, that indicates a small influence of the offer on the visit. On the other end, this number can be great (all offers viewed, but only one visit) and should be limited to something like max 4 (take a look in the code, for the actual limit).

Between the recognition of an offer and the completion (maybe by incidence) can be other visits. I do have written a completion by coincidence because, if the count of visits gets greater and greater, the actual influence of the offer is very small. That's the last term, that is in max (without one visit between seeing and acting) getting the number of one.

3. Prepare Data

To get those vectors I am extracting the event and dissolve the value column, that is a dictionary.

Calculating the influence

4. Data Modelling

Dividing the Profile DataFrame into some Clusters, based on a Principal Components Analyses.
For each Cluster I am calculating the total amount spent without regarding the influence score and as a second time, just the influenced transactions.

Those influenced transactions can be regarded as extra sales.
I am not evaluating the extra profit – maybe there is none, because the costs are on the same level as the income.

I will show the offer-types and the extra sales

Another diagram I am showing regards the avg amount spent in each cluster, for non-influenced transcations vs influenced transactions

5. Evaluate Data

In A/B Testing the clustering is done on training data and the evaluation is between split sets (test and control group) in each cluster. In this case I am using an influenced set to train the clustering model. Evaluation can therefor just be done by looking on the influence scores and if there is a correlation. (Maybe using the r-squared value).

Story:

Setting Terms to calculate and evaluate Offer influenced Transactions:

The QUICK and DIRTY way to evaluate Offers

Normally test design requires a structural proceed step by step, that excludes the dependence of observed factors. Those can be found in a long-term regard, with a seasonal ETS-decomposition plot (error trend seasonality). Based on findings, a Test design can be found and set up for example by using control and treatment groups.

Facing the given Data set those steps cannot be done. The data shows transaction data and provides information about the offers that were sent out. Since the offers were sent out by day 1 of the record, I cannot follow the usual procedure.
Therefor I will try a quick and dirty way, that gives me clusters and the chance on calculating an influence score. And with those two rough and clearly non-independent numbers, I will evaluate the offer-strategy.

First of all, let's get a common understanding when an offer is successful and when it is not. I would call it successful, if the customer is attracted and completes the offer on an extra visit. This extra visit is really important, because if the offer is used in a regular visit, the store reduces its profit.
This definition leads into the negative side - the profit reducing side of the offer. Unseen offers can be completed, but they do not help the store to increase the profit. Trying to avoid those transactions can be a strategy to follow up.
Therefor I will in the end have unsuccessful offers, non-influenced completion and influenced completion.

So let's take a look at the steps I take along the way.

Step 1: Clustering.
Clustering on customer profiles (including the transaction data) is in this case dependent on the offers itself. But by using the median I will give it a try. (One term of calculating the offer-influence needs the median amount spent – that would lead into at least a second iteration, to recalculate the profiles.)
I do value the advantage in using the transaction data, greater than the disadvantages in being dependent. Values on frequent visits and average consumption will help me setting the clusters.

Step 2: Calculating the influence score. (Feature Design)
The so-called Influence score is a scalar based on five features, that fit in the AIDA-model. AIDA describes the steps in a sales process. First by getting attention, the offer gets into a consideration set, second there might be interest in the offer, that leads into a Desire. Completing the offer is regarded as the final step - the Action.

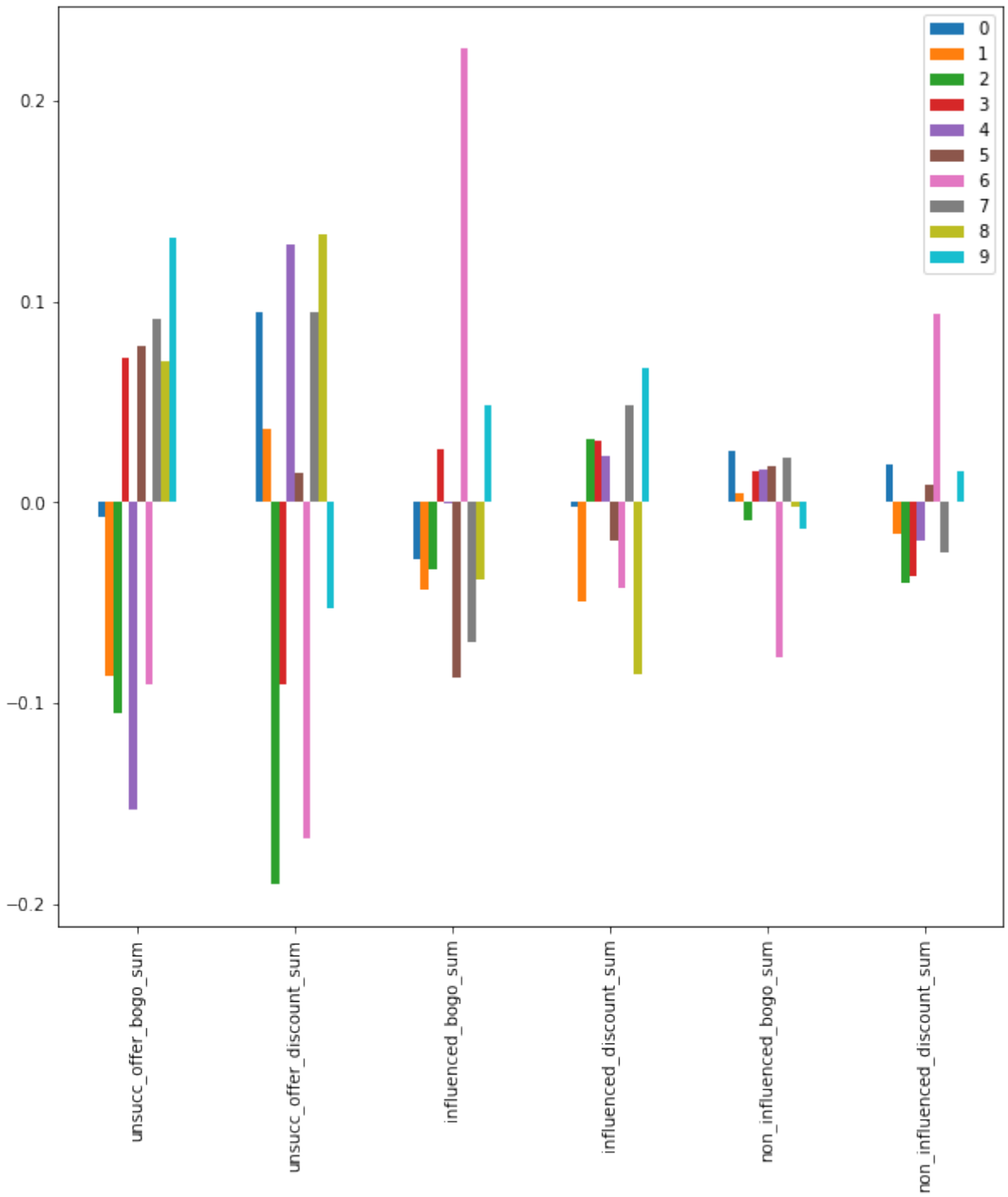
I do hope, that the Term-names and Mathematical definitions are telling my thoughts. Otherwise take a look in the READ-me.

1.)	Interested in Offers	(view / received *0.75)
2.)	Willingness to Overpaying	(Amount spent on Offer/ median amount spent)
3.)	own Difficulty	(Expiration Time – 48h / view to expire time)
4.)	Reactiveness	(Offer reaction time / view to expire time)
5.)	Fulfillment by coincidence	(1 / (1+ visits))

Step 3: Evaluating
If the scalar number exceeds ‘0.2’ as a critical swell, I call the transaction influenced. This number seams verry low, but if you score 5 terms more or less between 0 and 1, the number gets very small.

Step 4:
I have the categories and I have got the clusters – so now it is time to combine those two. Because the clusters are different in size, I have to look at the average amount spent, that I get by the count and the sum of each category.
Further I look at the performance as one number better or worse than the average score in this category. So a positive number indicates that customers of this cluster are x-times more likely to complete one offer of this type (for example ‘bogo’).

As told, it is better to send out offers, that will be seen and completed, than offers that are completed by accident.
So unsuccessful offers and non-influenced offers are indicating the other way round (negative cluster is performing better).



Further investigations can be:

- The number of clusters
- Error indication bars in the plot

By taking a look on the error indication, the whole estimation can be measured performance wise.

$$Influence\ Score =$$

1.) Interested in Offers

$$\frac{Offers\ viewed}{0.75 * Offers\ received}$$

2.) Willingness to Overpaying

$$\frac{Money\ spent}{median\ amount\ Spent}$$

3.) own Difficulty

$$\frac{Expiration\ Time}{rest\ Time\ till\ Expiration}$$

4.) Reactiveness

$$\frac{Offer\ reaction\ time}{0.5 *\ rest\ Time\ till\ Expiration}$$

5.) Fulfillment by coincidence

$$\frac{1}{(1 + visits\ till\ completion)}$$

Offers, Clusters and Profit

You save, we earn