Starbucks Capstone Challenge

Luis Soares Evangelista Neto
August 10, 2021

1 - Capstone Proposal

Luis Soares Evangelista Neto
August 10, 2021

2 - Definition

2.1 - Project Overview

This project has the purpose of offering a solution for the Starbucks Capstone Challenge. Starbucks is an American multinational company, with the largest chain of coffee shops in the world. For this project, was given a set of customer information that simulates the customer behavior on the Starbucks rewards mobile app. The mobile app has a feature that sends promotional offers or just advertisements to the clients, but they are categorized in 3 topics:

- Informational offer;
- Discount offer;
- Buy one get one free (BOGO) offer.

2.2 - Problem Statement

Sending several offers to clients can be expensive when we don't know how our public behave. Basically we need to improve the way we communicate with our clients, by doing that, we can improve ROI (Return of Investment). Using information about demographics and transactions we can explore how the clients that respond to our campaigns usually behave and we can formulate better offers for the clients. Using a clustering model, we can identify patterns in the clients that convert to an offer, patterns that we can't identify just by observing in exploratory analysis. These patterns can be translated into personas, that is a way to describe them with a "unique" characteristics and help the marketing team to formulate better offers.

2.3 - Metrics

In this project, we are going to use the K-Prototypes model, a Mix-feature (categorical and numerical) clustering model to identify the hidden patterns in the clients that successfully convert to an offer. To evaluate the model, we are going to use the elbow plot based on the Sum of the Squared Differences (SSD), the elbow plot is a method to identify the best number of clusters (in our case, personas). To describe each persona, we make some exploratory analysis with the model output, and see which variables have “unique” characteristics.

3 - Analysis

In this section, we are going to explore the data given to us, to get a better understanding. For this project, was given 3 datasets containing information about offers, clients and transactional data from the Starbucks mobile app.

- **portfolio.json** — Contains information about Starbucks campaigns (offers);
- **profile.json** — Personal information about the clients;
- **transcript.json** — Transactional information about the behavior of the clients.

3.1 - Data exploration

We begin our analysis exploring individually all the datasets to get a better understanding of how they are structured.

3.1.1 - Portfolio dataset

As said before, this dataset contains information about the campaigns that Starbucks offers to the clients. This dataset has offers sent during a 30-day test period (10 offers x 6 fields). Below we have all the columns in the portfolio.json:

reward (*integer*)	money awarded for the amount spent
channels (*list*)	Communication channel (web, email, mobile and social)
difficulty (*numeric*)	money required to be spent to receive reward
duration (*numeric*)	time in days for an offer to be open
offer_type (*string*)	Offer type (bogo, discount and informational)
id (string)	*offer id*

3.1.2 - Profile dataset

The second dataset is the profile.json, in this dataset we have some personal information about 17000 clients. Below we show all the variables:

gender (*categorical*)	Customer gender
age (*numerical*)	customer age
id (*string*)	customer id
became_member_on (*date format*)	subscription age
income (*numeric*)	customer income

3.1.3 - Transcript database

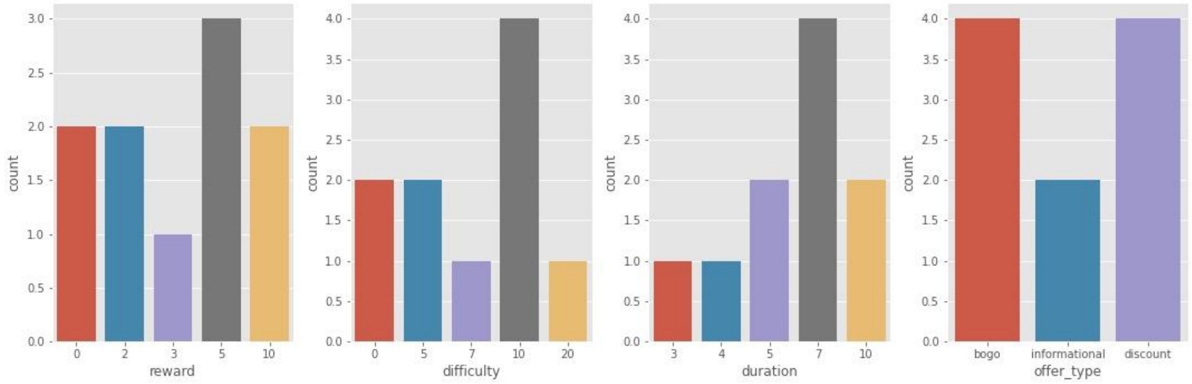
The last dataset given is the transcript.json, in this dataset we have all the transactions from our customers, the variables are:

person (*string*)	customer id
event (*string*)	state of the offer sent (offer received, offer viewed, transaction and offer completed)
value (*dictionary*)	Dictionary of event
time (*numeric*)	hours after the offer was sent

3.2 - Exploratory Visualization

3.2.1 - Portfolio dataset

Now, let’s observe how the data is distributed in the portfolio dataset in some of the variables to get some insights:

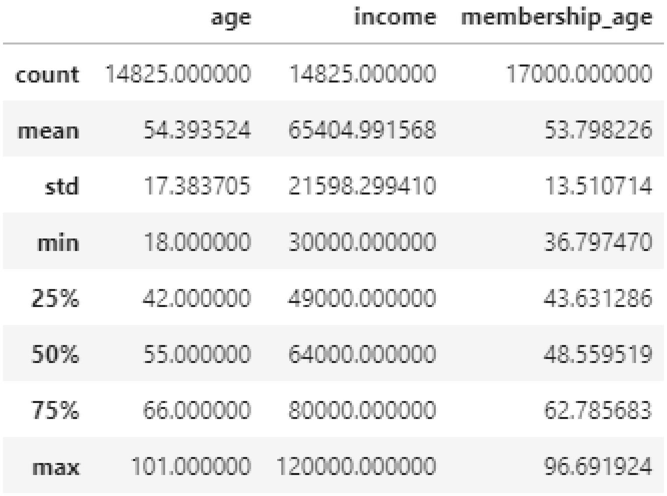


From the figure above, we can observe that we have a total of 10 campaigns, from that we observe:

- The 3 offer types mentioned before (Bogo, Informational and Discount).
- 4 offers with high difficulty, that means the minimum required to spend to complete an offer is high.
- There are 4 channels: email, mobile, social and web, and some of the offers are limited by the number of channels. .

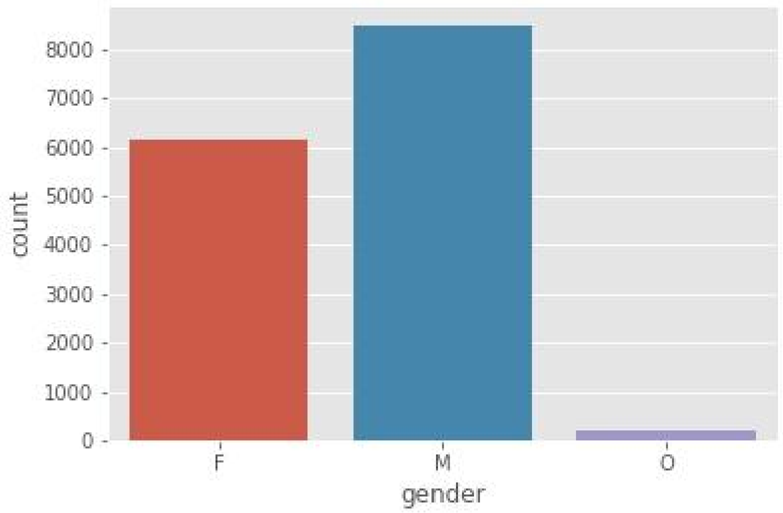
3.2.2 - Profile dataset

From the profile dataset, here are some basic statistics of the numerics features we have:

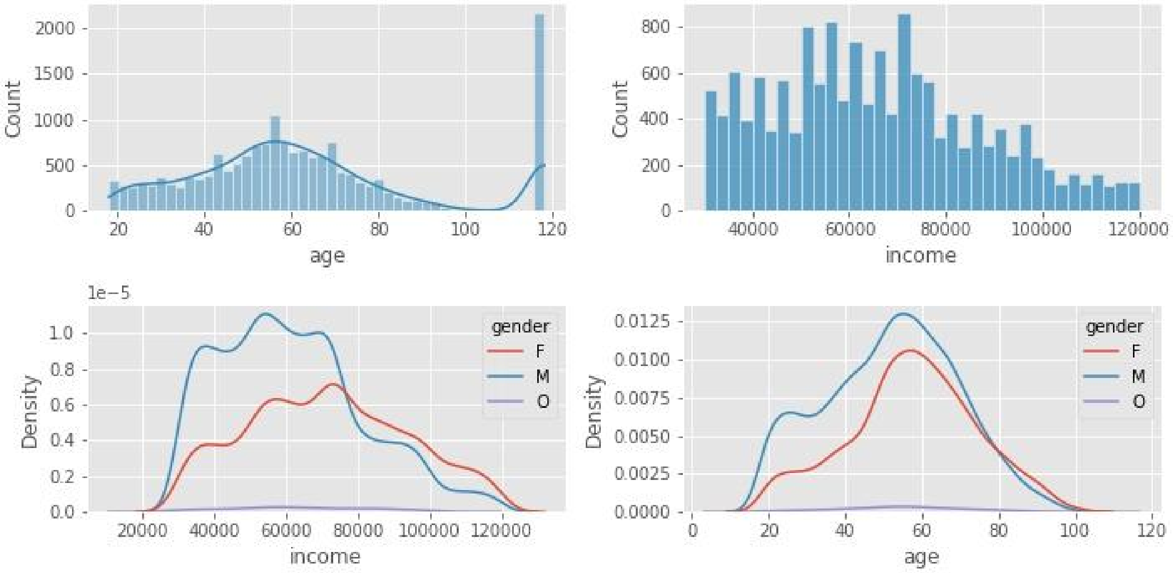


	age	income	membership_age
count	14825.000000	14825.000000	17000.000000
mean	54.393524	65404.991568	53.798226
std	17.383705	21598.299410	13.510714
min	18.000000	30000.000000	36.797470
25%	42.000000	49000.000000	43.631286
50%	55.000000	64000.000000	48.559519
75%	66.000000	80000.000000	62.785683
max	101.000000	120000.000000	96.691924

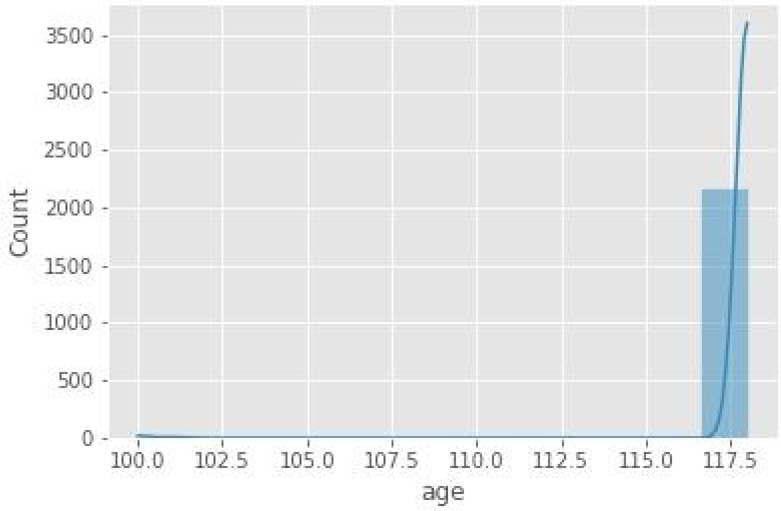
In the figure below we have the distribution from one of the variables from the profile dataset, in this case, gender.



We can see that the majority of our clients are males. When we stratify the *income* and *age* variables by the *gender* we have as shown in the figure below:



There's not much information about how these 2 variables behave together. But we observe that the variable *age* has some clients over 100 years old as shown below:



In total we have 2175 clients with the age of 118, we choose to handle these variables later in the preprocessing step. Another point to mention is that 2175 missing values exist for each variable, *gender* and *income*, so we assume that the 2175 unusual values in *age* come from the same problem of *gender* and *income*.

3.2.3 - Profile transcript

From this dataset we can observe that in the *event* variable we have 4 categories: 'offer received', 'offer viewed', 'transaction', 'offer completed'. In the *value* variable we have a dictionary that contains the related transaction based on a certain event for a specific client.

3.3 - Algorithms and Techniques

For this project, we are going to use two clustering algorithms, the K-Means and the K-prototypes. We choose these algorithms because we want to see how well they separate the personas, the K-Means uses only numerical features, and K-prototypes is a mix-feature (categorical and numerical).
To get a better understanding of the algorithms we are going to use, here's a brief explanation.

3.3.1 - K-Means

The K-Means algorithm is a classic unsupervised algorithm. K-Means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. k-means clustering minimizes within-cluster variances (squared Euclidean distances).

3.3.1 - K-Prototypes

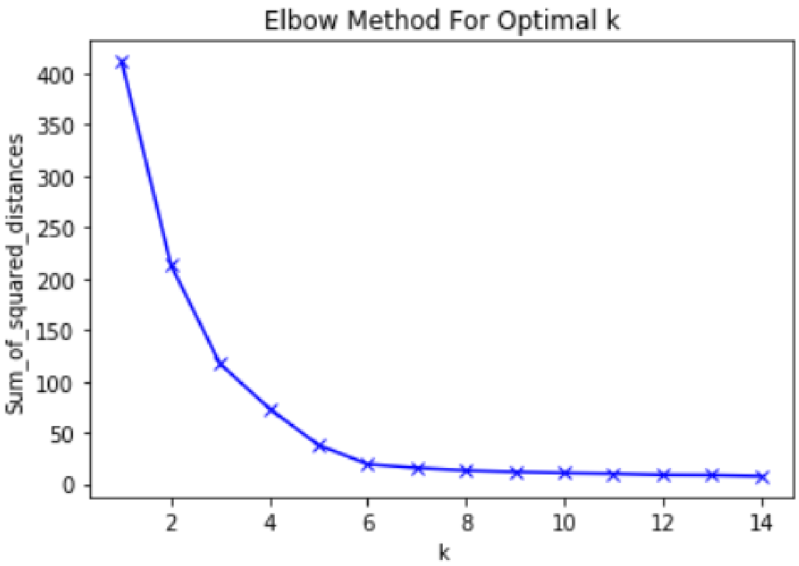
As said before, for this project we are going to use a Mix-Feature clustering model called K-Prototypes, this model has a method based on partitioning. This algorithm is an improvement of the K-Means and K-Mode clustering algorithm to handle clustering with the mixed data types. They do that with a special dissimilarity function that takes into account the cost function of K-Means and K-Modes models. Described below:

$$d(X, Y) = \sum_{j=1}^{p} (x_j - y_j)^2 + \gamma \sum_{j=p+1}^{m} \delta(x_j, y_j)$$

where

$$\delta(x, y) = \{0\ if\ (x = y)\ else\ 1\}$$

After we train the model, we need to evaluate how well the model can identify clusters. To do so, we are going to use the elbow plot technique that consists of identifying the point where we get more inflection based on the Sum of the Squared Differences (SSD), in the figure below, which is about 5 or 6, but this is not a rule.



3.4 - Benchmark

To do a benchmark, we are going to train K-Means and K-Prototypes with the objective of identifying which of them can better separate the data in personas, using Sum of the Squared Differences (SSD) at first, then, we choose the better model and explore all the clusters using the variables we used to train the model, in some cases we can use variables that are not used in the training. If we observe that some clusters have unique characteristics, we can assure that we found the personas.

4 - Methodology

4.1 - Data Preprocessing

In this section we are going to do some preprocessing in our datasets to correct some errors and create new features that may give us some insights. Below we have the steps we are going to make in each dataset:

Preprocessing Part 1

1. Portfolio
    - Creating unique columns for each *channel*.
    - Creating unique (dummies) columns for each *offer_type*.
    - Renaming the *id* to *offer_id*.
    - Renaming all the offers to a simpler label (i.e 1, 2,..)

2. Profile

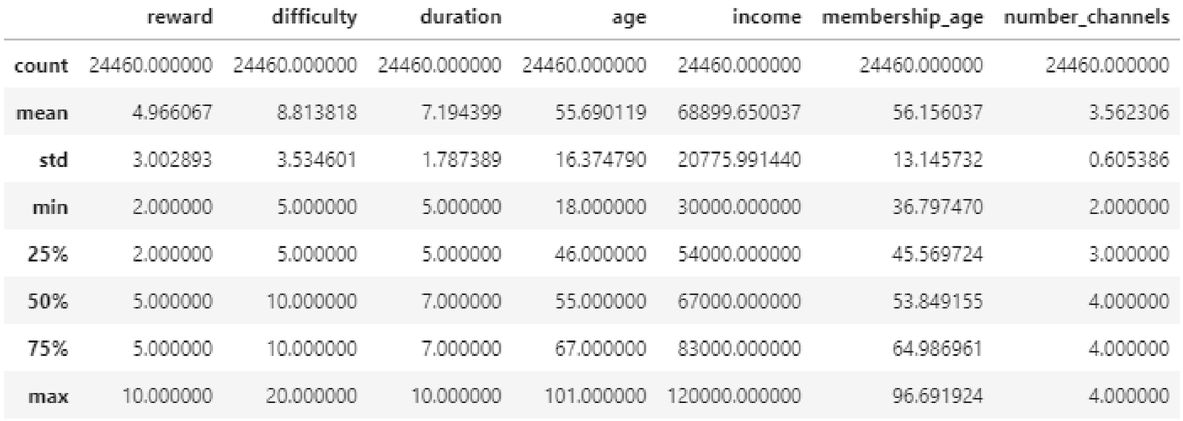
- Renaming the id to *customer_id.*
- First, let's handle the null values in the *gender* variable creating a new category “N”. Then we can create unique columns (dummies) of gender.
- Fill all the 118 in *age* with *NaN* at this moment.
- Since we have the variable became_member_on, we can create a variable that defines the membership age.
- Fill the null values from *age* and *income* with mean.
- Drop the *became_member_on* variable.

3. Transcript
    - Renaming the id to *customer_id.*
    - Extract *reward*, *offet_type* and *amount* from the dictionary variable *value.*
    - Drop the variable *value.*

Preprocessing Part 2

From the 3 variables extracted before in the transcript dataset, we can create a table with the information if the client successfully completes an offer. With that, we merge all the datasets for a more rich analysis.

With the dataset created, we need to do some exploratory analysis to investigate with the perspective of successful offers. Here are some descriptive statistics about the variables we are going to use.



	reward	difficulty	duration	age	income	membership_age	number_channels
count	24460.000000	24460.000000	24460.000000	24460.000000	24460.000000	24460.000000	24460.000000
mean	4.966067	8.813818	7.194399	55.690119	68899.650037	56.156037	3.562306
std	3.002893	3.534601	1.787389	16.374790	20775.991440	13.145732	0.605386
min	2.000000	5.000000	5.000000	18.000000	30000.000000	36.797470	2.000000
25%	2.000000	5.000000	5.000000	46.000000	54000.000000	45.569724	3.000000
50%	5.000000	10.000000	7.000000	55.000000	67000.000000	53.849155	4.000000
75%	5.000000	10.000000	7.000000	67.000000	83000.000000	64.986961	4.000000
max	10.000000	20.000000	10.000000	101.000000	120000.000000	96.691924	4.000000

Now, let's see what is the successful rate of each offer:

offer_id	**successful rate (%)**
5	70,0
7	68,18
4	58,37
9	44,63
10	44,60
8	39,40
1	38,42
6	23,09
2	0
3	0

From the table above we see that some of the offers have a high successful rate, for example, the offers 4, 5 and 7 have above of 50% successful rate, on the other hand, we see that the offers 2 and 3 have 0% impact on the public. For another point of view, let's see how the successful rate is based on some of the offers attributes.

reward	**successful rate (%)**
0	0
2	54,27
3	68,19
5	40,18
10	44,61

difficulty	**successful rate (%)**
0	0
5	48,81
7	68,18

10	49,42
20	23,09

duration	successful rate (%)
3	0
4	0
5	51,46
7	47,65
10	46,47

From the variables above, we see that some of the attributes alone don’t have an impact on the successful rate.

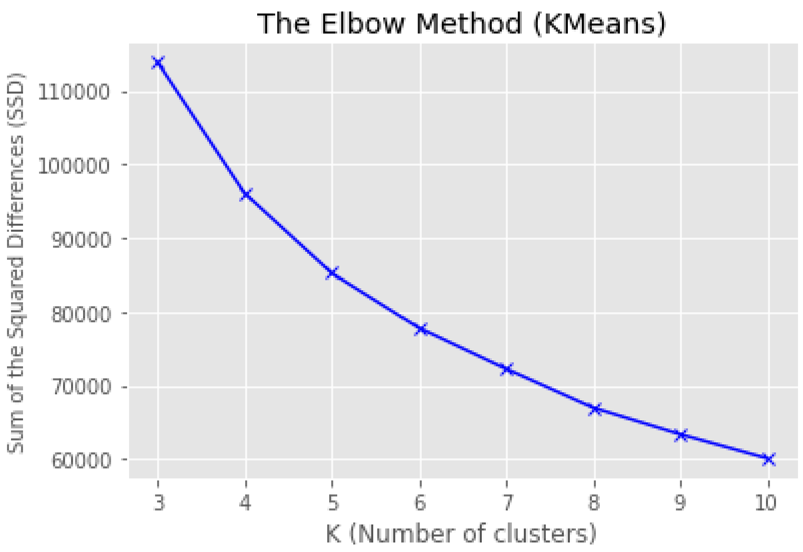
4.2 - Implementation

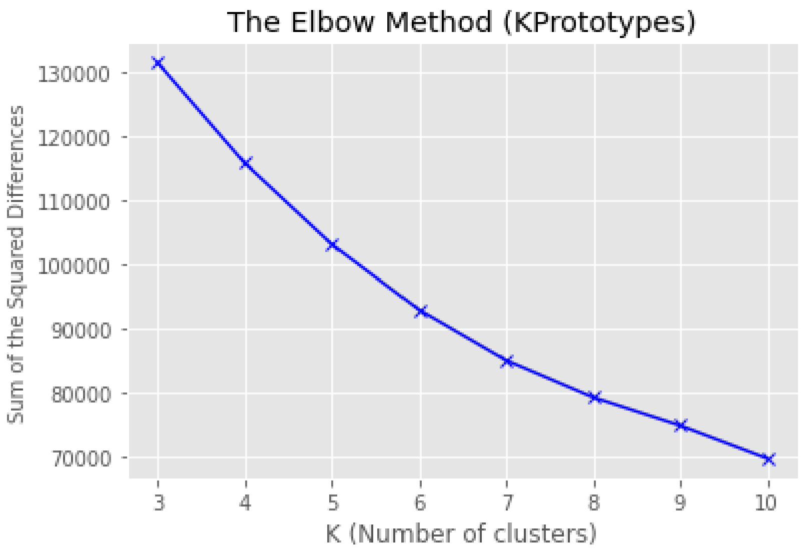
First of all, before we go to a more advanced technique, we need to understand our customers, especially the clients who successfully convert to an offer, filtering only the successful offers from the dataset obtained from the preprocessing part 2, we have 24460 observations. Now, we need to understand how this type of client behaves. In this notebook, we are going to try to cluster all the clients that convert for an offer. The variables we are going to use for a first analysis are:

numericals = 'reward', 'membership_age', 'difficulty', 'age', 'income', 'duration'

categoricals = 'channels', 'offer_type_cat', 'gender_cat', 'offer_id'

After training the K-Means, with the numerical features and K-Prototypes with both (numerical and categorical features) for 8 iterations (from 3 to 10 clusters), we have the *elbowplots* below.





From the *elbowplots*, we see that the SSD is not much different and none of the models show an explicit inflection in any of the values of K (number of clusters). So, for a better description of a first personas, we are going to choose the K-Prototypes model because we can use categorical features to describe our personas.

For the first analysis, let's choose 5 clusters just for simplicity and analyse how is the description of these clusters.

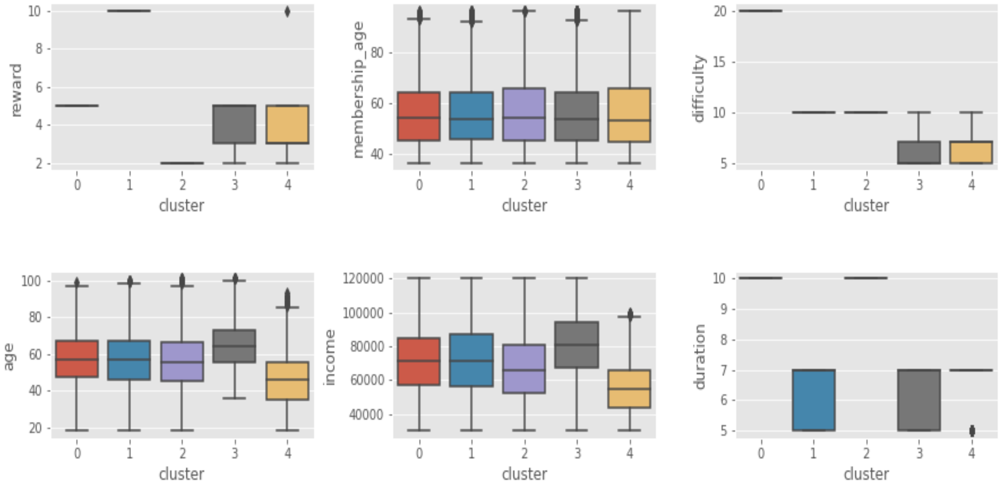
After training the K-Prototypes model for 5 clusters, we have the following results:

Cluster	Population (%)
0	1472 (6,91%)
1	5667 (23,16%)
2	4433 (18,12%)
3	6521 (26,66%)
4	6367 (26,03%)

From the table above, we see that our model distributed well our observations, let's see how the variables in each cluster are distributed.

Numericals

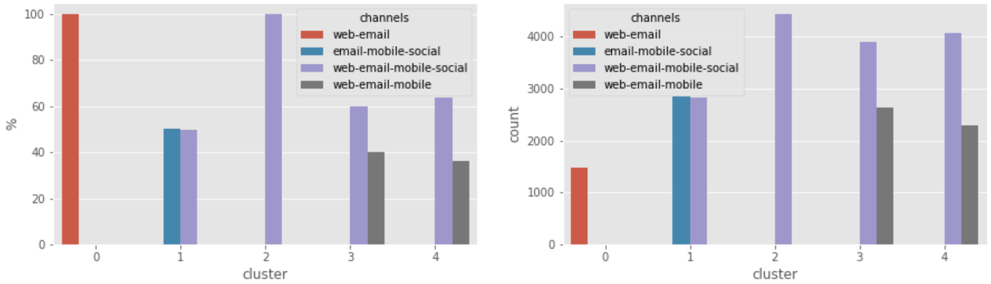
Adding the clusters into the dataset, we can explore how the variables behave when we segment by cluster, this helps us define the persona (description) of each cluster, based on the key differences in each variable. Below we have a *boxplot* showing how each variable behaves by cluster.

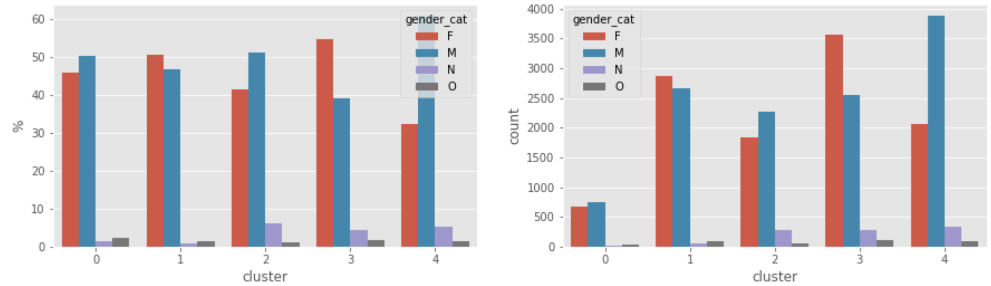


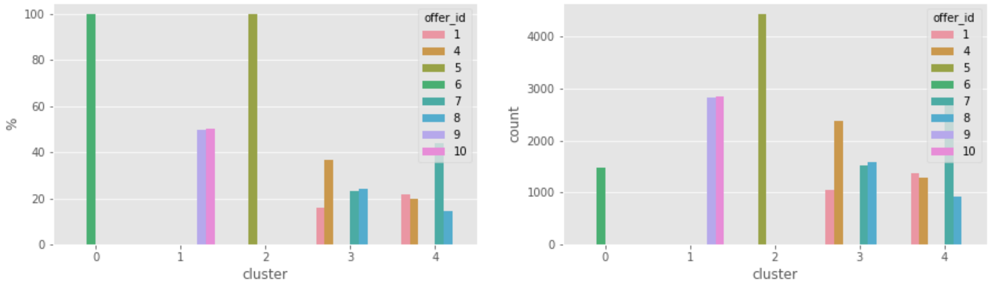
We see that some of the variables in certain clusters are basically the same, that doesn't interest us because we want a "unique" characteristic from a variable in the cluster. So for example, cluster 0 in the *reward* variable has only the value 5, and no other cluster has the same value, so this is an unique feature from this cluster.

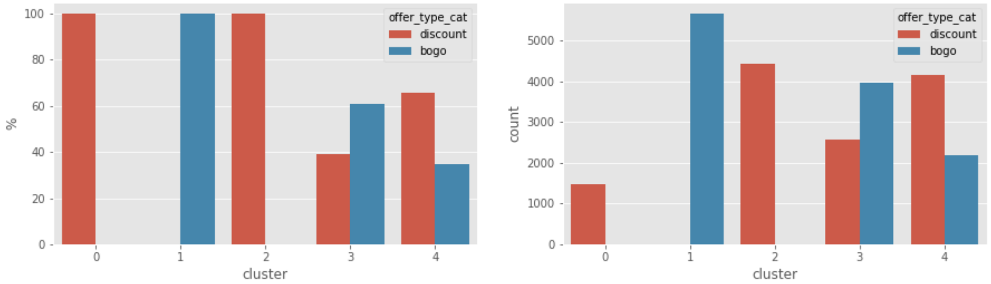
Categoricals

Exploring the categorical variables for each cluster , we have the following distributions:









From the categorical variables we have too some unique features in some of the clusters.

4.3 - Refinement

From section 4.2, we see that our *elbowplot* doesn't have an explicit inflection point among the clusters we choose. Just because we don't have an explicit inflection, doesn't mean we can't progress with our analysis, the cluster analysis with the variables that will make the verdict if the model is useful or not. But if you don't feel secure about it, basically we need to create or obtain more variables and train the model again and see the *elbowplot*. Another refinement is to use the silhouette plot.

5 - Results

5.1 - Model evaluation and Validation

Defining the personas

From the graphics above, we can aggregate all this information and define some unique characteristics of each cluster that may help us to take some action in how we communicate with our customers. The personas are:

Cluster 0 - 1472 (~6%) clients:

- Clients with **reward** equal to 5, **difficulty** equal to 20 and **duration** equal to 10;
- All these clients convert from web and email;
- The offer type they receive is **discount**;
- And all of them convert from the **offer number 6**.

Cluster 1 - 5667 (~23.2%) clients:

- Clients with **reward** equal to 10, **difficulty** equal to 10 and **duration** falls between 5 and 7;
- All the clients in this cluster convert from all channels;
- They convert from **bogo** offers;
- And all of them convert from the **offers number 9 and 10**.

Cluster 2 - 4433 (~18.1%) clients:

- Clients with **reward** equal to 2, **difficulty** equal to 10 and **duration** equals to 10;
- All the clients in this cluster convert from all channels;
- They convert from **discount** offers;

- And all of them convert from the **offer number 5**.

Cluster 3 - 6521 (~26.6%) clients:

- Clients with **reward** that falls between to 3 and 5, **difficulty** falls between to 5 and 7 and **duration** falls between to 5 and 7;
- They are on average the oldest clients and with the highest average income.
- All the clients in this cluster convert from all channels;
- They convert from **discount** (40%) and **bogo** (60%) offers;
- About 60% of the clients are males;
- They convert from multiple offers (1, 4, 7 e 8).

Cluster 4 - 6367 (~26%) clients:

- Clients with **reward** that falls between to 3 and 5, **difficulty** falls between to 5 and 7 and **duration** equals to 7;
- They are on average the youngest clients and with the lowest average income.
- All the clients in this cluster convert from all channels;
- They convert from **discount** (60%) and **bogo** (40%) offers;
- About 60% of the clients are females;
- They convert from multiple offers (1, 4, 7, 8).

5.2 - Justification

The two models we choose to explore don't have much difference when we see a metric like SSD, but, if we just used the K-Means, we would be limited in the description of the personas, in a real world problem, we would have information about the clients that can be represented in numerical and categorical way, so, the K-Prototypes model is the best option here.

From the clusters, we observe that they separate our clients into a few characteristics that we can work with the marketing team to boost sales. Working together with the marketing team or the business team, we can describe our personas even more. An important observation we can make is that not all the offers we send have a good impact on the conversion. The offer that converts very few people has to be rethought.

6 - Conclusion

6.1 - Reflection

The model developed is for sure not the best we can get from the data that was given to us. This is because our analysis only has the purpose to explore a possible solution end-to-end, a baseline model.

Our *elbowplot* don't have an explicit inflection, but anyway we obtained some unique characteristics in our personas. So, I believe this model can be used, not all clusters, but some of them in a small representative sample.

6.2 - Improvement

A way to improve the model developed is to create another model with all clients. This will give us more information about all the clients we have and investigate some bias in our first clusters.

If some of our clusters successfully change of conversion (A/B testing) in a possible test, we can go further and use some predictive machine learning model to infer what is the probability of a customer converting to a offer (y cluster). We can use a **lift chart** to verify the performance of our model and work together with the CRM teams.

Creating or getting more features is an interesting way to improve the model too. More feature engineering, more exploration is always good if we have some time to improve the model.