Starbucks capstone project

Thomas Heinemann

December 30, 2023

1 Project overview

This is a recommendation engine project proposed by Starbucks to find out which demographic groups of customers respond best to certain types of offers. The underlying data sets include transactional and customer data, as well as information about the types of offers sent. The full set of files, including the main Jupyter file with the detailed task, can be found at [1]. In addition, an approach for a simpler version of such a problem can be found here [2].

References

[1] All project files: `https://github.com/thomasheinemann/starbucks_promotion_type_exercise/`

[2] Simpler version of this project: `https://github.com/thomasheinemann/starbucks_promotion_exercise/`

2 Introduction

To begin with we first present the original introduction and task (the full description is here [1]):

> *This data set contains simulated data that mimics customer behavior on the Starbucks rewards mobile app. Once every few days, Starbucks sends out an offer to users of the mobile app. An offer can be merely an advertisement for a drink or an actual offer such as a discount or BOGO (buy one get one free). Some users might not receive any offer during certain weeks.*
>
> *Not all users receive the same offer, and that is the challenge to solve with this data set.*
>
> *Your task is 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.*
>
> *Every offer has a validity period before the offer expires. As an example, a BOGO offer might be valid for only 5 days. You'll see in the data set that informational offers have a validity period even though these ads are merely providing information about a product; for example, if an informational offer has 7 days of validity, you can assume the customer is feeling the influence of the offer for 7 days after receiving the advertisement.*
>
> *You'll be given transactional data showing user purchases made on the app including the timestamp of purchase and the amount of money spent on a purchase. This transactional data also has a record for each offer that a user receives as well as a record for when a user actually views the offer. There are also records for when a user completes an offer.*
>
> *Keep in mind as well that someone using the app might make a purchase through the app without having received an offer or seen an offer.*

So how do we proceed? A simple statistical evaluation based on the frequency of offer completions or views per customer may not work as we do not have huge data sets for each person with and without receiving, viewing and completing a considered offer type. Also, a customer being researched may not have exactly the same profile as those being provided. We therefore improve the statistics by reducing the value set of the customer profile data. However, this is not enough, so we train models to predict as accurately as possible whether the customer is worth promoting or not. The advantage of this is that these models also make good use of information from other more or less similar customer profiles.

In the following sections, we provide definitions, present the modeling approach and its implementation, summarize the results for each type of offer, conclude with our main findings in the conclusion section and finally provide an outlook with a discussion on how to improve the current survey.

3 Definitions

Main offer types to be considered:

- BOGO (buy one get one free) offer: If the customer buys the praised product of this offer then this person gets the same product on top for free.
- discount offer: If the customer spends a certain amount then this person receives a discount.
- informational offer: These offers just show the customer an advertisement for a product, e.g., a drink. In the given data, however, there is no offer completion record. For this purpose, we define an offer instance as "completed" if the hourly spend exceeds that of an no-offer instance, and vice versa.

Offer instance: An offer sent for one specific customer, at one specific time (offer_received_time). So each offer instance is addressed with the key **[offer_id, person, offer_received_time]** which is found being unique. No-offer times of a customer are treated as "no-offer" offer instances and addressed with [offer_id="no offer", person, offer_received_time=0].

Offer events:

- offer received
- offer viewed (can also happen after offer is completed or no longer valid)
- transaction = the customer makes a transaction of a certain amount (can happen during an offer or a "no-offer" time)
- offer completed = the difficulty challenged by the offer has been overcome with the current transaction

Offer event times: Times of the events above are further denoted as "offer_received_time", "offer_viewed_time", "transaction_time", and "offer_completed_time".

Offer event record (OER): An OER consists of an offer received event, a possible offer viewed or completed event, and all transaction events associated with a specific offer instance. Similar to each offer instance, each OER is addressed using the unique key [offer_id, person, offer_received_time]. For each person we also define one OER associated with no offer times which is addressed with [offer_id="no offer", person, offer_received_time=0]. Each OER has an offer validation interval starting at its offer_received_time and ending at

$$\text{offer\_time\_out} = \text{offer\_received\_time} + 24 \times \text{duration}.$$

The duration is the maximum offer-specific time interval in days during which the offer can be valid if not completed earlier.

States of an offer instance:
Each offer instance is characterized by the binary state variables "promoted", "viewed" and "completed". Their calculation requires the corresponding OER and is shown in the following table.

state variables of an offer instance	**calculation in terms of logical expressions**
promoted	offer_id $\neq$ "no offer"
viewed	offer_viewed_time $\leq$ offer_end_time with: offer_end_time = Min(offer_time_out, offer_completed_time)
completed	offer_completed_time $\leq$ offer_time_out

Consumption during offer: It is a quantity related to an OER. It is defined as the sum of all transactions made in an OER where each transaction lies in the time interval [offer_received_time, offer_time_out].

4 Modeling approach

4.1 Data flow for new customers

The data flow along our prediction pipeline starts with the original customer profile attributes and ends with a decision on whether or not to promote that person. Additional attributes deduced from the persons' OERs are not shown here as they are needed for fitting and scoring but theoretically not for pure prediction. The first step is to get rid of NaN values and introduce categorical variables for the gender attribute. In the second step, model 1 predicts an early viewing probability, which is used to sort out "lazy viewers". To predict the final decision to send the offer, in step 3 we used model 2, which is fitted separately for each type of offer.

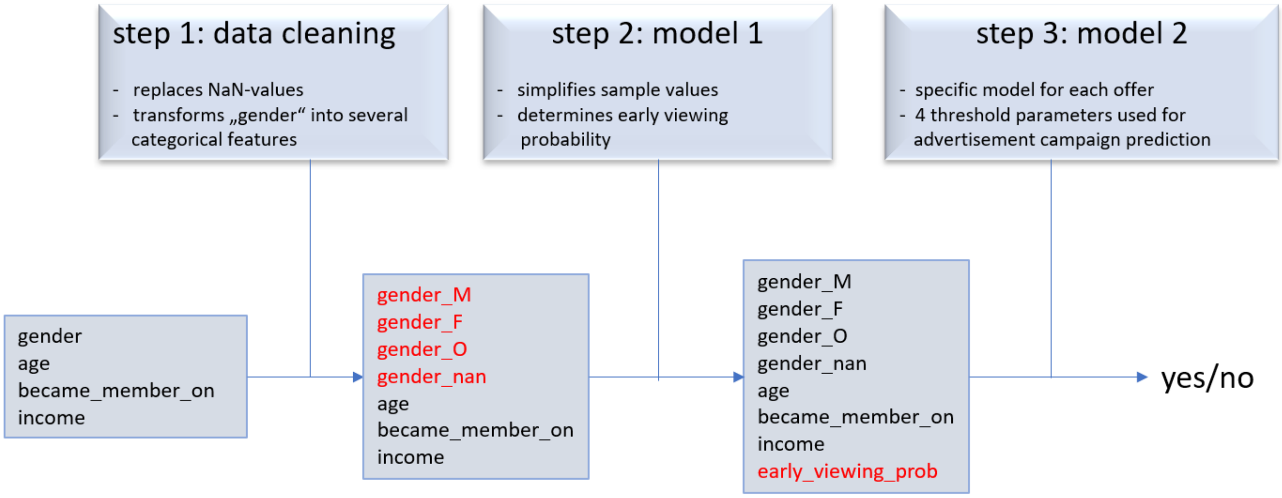

Figure 1: Data flow showing a prediction pipeline for the customer profile attributes "gender", "age", "became_member_on", "income" to a "Yes/No" prediction for an advertisement campaign. For each offer type a differently fitted version of model 2 is used. Models 1 and 2 were fitted using customer profile attributes and customers' offer event records (OERs).

4.2 Step 1: data preparation

The customer profile data is first subjected to a NaN value treatment for the income. In particular, customers who do not report an income, are marked as having an income of -10,000 dollars. Another issue treated in this step is the gender attribute which is categorical without order and therefore split into one-hot encoded attributes for each class label, resulting in "gender_M", "gender_F", "gender_O", and "gender_nan" (indicating if no gender is provided).

4.3 Step 2: model 1

Model 1 has the task of preventing overfitting and improving statistics by simplifying the value sets of the customers' integer features "age", "become_member_on" and "income" using a transformer to achieve 10 year steps for the "age", 1 year steps for "become_member_on" and 10,000 dollar steps for "income". Then, in a second transformation, a test-set customer receives, within the framework of model 1, an early viewing probability encoded in the attribute, "early_viewing_prob" (probability that the offer will be viewed within 48 h). This transformer also produces viewing preference information with the self-explanatory attributes "early_viewing_pref" and "late_viewing_pref" to predict whether the customer prefers to view within 48 hours, prefers to view after 48 hours or does not tend to view at all. These two attributes are not actively used as feature variables in subsequent steps of the pipeline, as they are only predicted from the demographic attributes, but are useful for supporting the visualization of a customer's demographics in the results section. The underlying model of the second transformer is fitted with all the cleaned customer profile data from the profile training set, together with information about their OERs. The offer instances to be considered include only the customers' non-overlapping offer instances. This restriction was chosen because multi-offer effects can be quite complex and would be a further step of investigation that would likely require more data.

4.4 Step 3: model 2 (offer specific model)

In this step, an offer-specific model comes into play. Accordingly, a separate instance of this model is used for each offer type. The model itself uses cleaned customer profile data and all the customers' OERs of non-overlapping offer instances of the considered offer type and their corresponding no-offer OERs. The "offer completion" states for the no-offer instances are calculated in this model as if the customer had been promoted with the offer type considered in this model. To be precise, a difficulty had to be overcome that scales with the ratio between the no-offer and the offer duration.

Regarding the states of the offer instances, we generally distinguish six since an offer can be sent or not, can be viewed (if a promotion exists) or not, and completed or not. A grouping of these six classifications into four different labels forms the basis of the current model and is shown in the table below.

label	states of offer instances		
	promoted	viewed	completed
I	yes	yes	yes
II	no	no	yes
	yes	no	yes
III	yes	yes	no
IV	no	no	no
	yes	no	no

The aim of Model 2 is to make use of the upper 4-label classification of offer instance states to predict customers who only complete the offer when viewed. Ideally, these customers will only have offers with states labelled I and IV, respectively. Fig. 2 shows a Venn diagram depicting ideal customer sets along with the one we are interested in. We hereby define an ideal customer as one for whom the offer completion status is solely dependent on the viewing status, or is even independent of the latter. This definition leads to the four different types of ideal customers shown in the figure below.

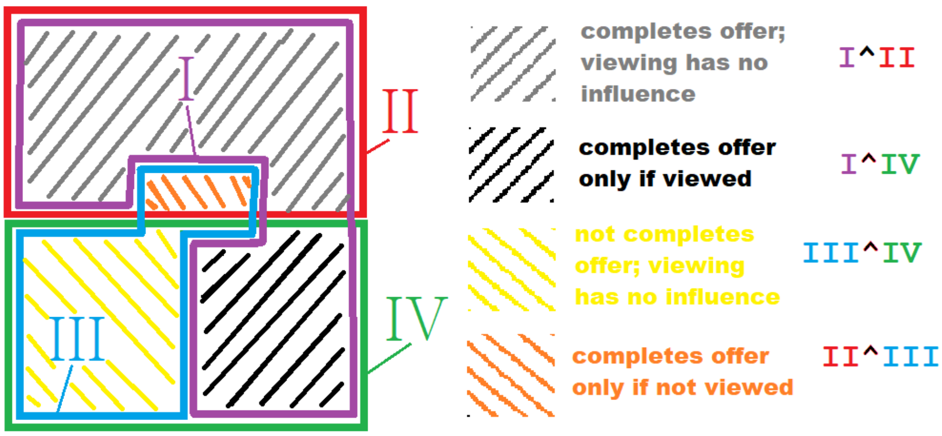

Figure 2: A Venn diagram is shown, representing a grouping of ideal customers into four disjunct sets filled with differently colored diagonal stripes. In each set there are only two different offer instance state labels (represented by Roman numbers).

Unfortunately, our customers are not ideal, so each customer will behave differently in relation to each individual offer instance. Therefore, the only way we can predict what behavior the customer is most likely to reveal is by assuming a correlation between the customer's profile and the states of their offer instances.

In order to determine whether a customer will "complete the offer only if viewed", we create one predictor to distinguish between customer profiles with states labelled II vs. those with states labelled IV, and analogously a second to distinguish between I vs. III. Using their prediction probabilities for both label I and label IV customers, we can get a good estimate of the black-striped area. To do this, the threshold parameters p_1 and p_2 must be exceeded by both prediction probabilities.

However, some customers are actually discouraged from buying when they see an ad (see orange-striped area). These customers are not in the black-striped area, but since the boundaries are fuzzy, we also included a third predictor (with threshold parameter p_3) that distinguishes between customer sets with labels I $\cup$ IV vs. II $\cup$ III. This allows separate tuning of the boundary between the orange and black striped area and the boundary between the orange and grey striped area. In addition, we have implemented a fourth threshold parameter p_4 for the early viewing probability determined in model 1 to weed out lazy viewers in the result set. This can prevent too many customers from being unknowingly rewarded.

The threshold parameters are optimized on a hyperparameter grid for each model 2 instance using a 3-fold cross-validation procedure with 1250 grid points. The scoring metrics for the validation sets and the test set used for every model 2 instance are defined below.

4.5 Scoring metric

The scoring metric was chosen to take into account two important aspects. Firstly, we want to maximize the number of customers who complete an offer when they receive it. The underlying metric to be maximized is the "incremental response rate" ("irr"). On the other hand, we aim to gain consumption by maximising a shifted version of "net incremental revenue" ("nir"). This prevents our recommendation algorithm from promoting too many customers whose rewards make them spend less because they are too satisfied. As a scoring metric, we therefore used a compromise that is a product of the two previous metrics, i.e.

$$\text{scoring metric} = irr \times nir \tag{1}$$

Both metrics are defined in the following.

Incremental response rate "irr" (for offer type X):

$$irr = \frac{n_{prom,compl}}{n_{prom}} - \frac{n_{no\ prom,compl}}{n_{no\ prom}} \tag{2}$$

$n_{prom,compl}$: number of instances of offer type X which are completed

n_{prom} : number of instances of offer type X

$n_{no\ prom,compl}$: number of no-offer instances whose consumption rate would have completed an offer of offer type X

$n_{no\ prom}$: number of no-offer instances

Shifted net incremental revenue "nir" (for offer type X): We take the sum of the consumption of completed offers (of offer type X) minus the sum of the consumption of no-offer instances whose consumption rate would satisfy an offer completion event of offer type X. Each consumption is normalized by its corresponding offer/no-offer duration. The function of the shift is to take care of the imbalanced data sets of completed offer instances vs. completed no-offer ones.

$$nir = \sum_{\substack{i \in \text{"completed" offer} \\ \text{instances in} \\ \text{test set}}} \frac{\text{consumption(i)}}{\text{duration(i)}} - \sum_{\substack{k \in \text{"completed"} \\ \text{no-offer instances} \\ \text{in test set}}} \frac{\text{consumption(k)}}{\text{duration(k)}} - \text{shift} \cdot n_{\text{test}} \tag{3}$$

consumption (x) : consumption during offer/no-offer instance x

duration (x) : duration of offer/no-offer instance x

n_{test} : number of all offer/no-offer instances (in the test set)

n_{train} : number of all offer/no-offer instances (in the training set)

$$\text{shift} = \frac{1}{n_{\text{training}}} \left(\sum_{\substack{i \in \text{"completed" offer} \\ \text{instances in} \\ \text{training set}}} \frac{\text{consumption(i)}}{\text{duration(i)}} - \sum_{\substack{k \in \text{"completed"} \\ \text{no-offer instances} \\ \text{in training set}}} \frac{\text{consumption(k)}}{\text{duration(k)}} \right)$$

5 Implementation of the modeling approach

In this section we present the practical implementation of our modeling approach using examples of the data provided. Every important step in the data flow is described separately below.

5.1 Step 1: data preparation

First, the customer profile data is prepared by converting NaN values in the "income" field to -10,000 dollars and the "gender" attribute to one-hot encoded attributes, as described in 4.2 and shown in Fig. 3.

	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
5	M	68	e2127556f4f64592b11af22de27a7932	20180426	70000.0
6	None	118	8ec6ce2a7e7949b1bf142def7d0e0586	20170925	NaN
7	None	118	68617ca6246f4fbc85e91a2a49552598	20171002	NaN

⇩

	age	id	became_member_on	income	gender_F	gender_M	gender_O	gender_nan
0	118	68be06ca386d4c31939f3a4f0e3dd783	20170212	-10000.0	0	0	0	1
1	55	0610b486422d4921ae7d2bf64640c50b	20170715	112000.0	1	0	0	0
2	118	38fe809add3b4fcf9315a9694bb96ff5	20180712	-10000.0	0	0	0	1
3	75	78afa995795e4d85b5d9ceeca43f5fef	20170509	100000.0	1	0	0	0
4	118	a03223e636434f42ac4c3df47e8bac43	20170804	-10000.0	0	0	0	1
5	68	e2127556f4f64592b11af22de27a7932	20180426	70000.0	0	1	0	0
6	118	8ec6ce2a7e7949b1bf142def7d0e0586	20170925	-10000.0	0	0	0	1
7	118	68617ca6246f4fbc85e91a2a49552598	20171002	-10000.0	0	0	0	1

Figure 3: Data cleaning of customer profile data is shown.

The provided transcript and portfolio data sets are then joined into the transcript unfolded portfolio dataframe ("df_transcript_unfolded_portfolio"; see Fig. 4). More specifically, the attribute "value" of the transcript dataframe, which is provided in terms of dictionary objects, is unfolded first and then augmented with information from the portfolio dataframe.

person	time	event	value
0009655768c64bdeb2e877511632db8f	168	offer received	{'offer_id': '5a8bc65990b245e5a138643cd4eb9837'}
	192	offer viewed	{'offer_id': '5a8bc65990b245e5a138643cd4eb9837'}
	228	transaction	{'amount': 22.16}
	336	offer received	{'offer_id': '3f207df678b143eea3cee63160fa8bed'}
	372	offer viewed	{'offer_id': '3f207df678b143eea3cee63160fa8bed'}
	408	offer received	{'offer_id': 'f19421c1d4aa40978ebb69ca19b0e20d'}
	414	offer completed	{'offer_id': 'f19421c1d4aa40978ebb69ca19b0e20d'}
		transaction	{'amount': 8.57}
	456	offer viewed	{'offer_id': 'f19421c1d4aa40978ebb69ca19b0e20d'}

	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
5	3	[web, email, mobile, social]	7	7	discount	2298d6c36e964ae4a3e7e9706d1fb8c2
6	2	[web, email, mobile, social]	10	10	discount	fafdcd668e3743c1bb461111dcafc2a4
7	0	[email, mobile, social]	0	3	informational	5a8bc65990b245e5a138643cd4eb9837
8	5	[web, email, mobile, social]	5	5	bogo	f19421c1d4aa40978ebb69ca19b0e20d
9	2	[web, email, mobile]	10	7	discount	2906b810c7d4411798c6938adc9daaa5

				amount	reward	channels	difficulty	duration	offer_type
person	time	event	offer_id						
0009655768c64bdeb2e877511632db8f	168	offer received	5a8bc65990b245e5a138643cd4eb9837	NaN	0.0	[email, mobile, social]	0.0	3.0	informational
	192	offer viewed	5a8bc65990b245e5a138643cd4eb9837	NaN	0.0	[email, mobile, social]	0.0	3.0	informational
	228	transaction	NaN	22.16	NaN	None	NaN	NaN	None
	336	offer received	3f207df678b143eea3cee63160fa8bed	NaN	0.0	[web, email, mobile]	0.0	4.0	informational
	372	offer viewed	3f207df678b143eea3cee63160fa8bed	NaN	0.0	[web, email, mobile]	0.0	4.0	informational
	408	offer received	f19421c1d4aa40978ebb69ca19b0e20d	NaN	5.0	[web, email, mobile, social]	5.0	5.0	bogo
	414	offer completed	f19421c1d4aa40978ebb69ca19b0e20d	NaN	5.0	[web, email, mobile, social]	5.0	5.0	bogo
		transaction	NaN	8.57	NaN	None	NaN	NaN	None
	456	offer viewed	f19421c1d4aa40978ebb69ca19b0e20d	NaN	5.0	[web, email, mobile, social]	5.0	5.0	bogo

Figure 4: The transformation of the original transcript (top left) and portfolio (center right) records into the "df_transcript_unfolded_portfolio" dataframe is shown.

In the next step, the latter dataframe is transformed into a so-called main dataframe "df_main" (see Fig. 5) by several intermediate steps. All records with the key [person, offer_id, offer_received_time] are now combined into one record. There is also always one entry per person containing aggregated information about the no-offer times. For records related to an offer type, the "promoted" variable is set to 1, otherwise 0 for no-offer instances. If the offer instance is viewed before it expires, it is marked with "viewed=1", otherwise with 0. Similarly, the new variable "completed" is set. The remaining new attributes "duration" and "consumption" represent the duration of the offer instance and the sum of all transactions made within its valid time interval. In addition to these data aggregations, there are also offer instances that have not been considered in the fitting and prediction. In particular, offers that overlap by more than one hour are removed, as we do not consider the influence of multiple offers on a customer.

	person	offer_id	offer_received_time	offer_viewed_time	promoted	viewed	completed	duration	consumption
0	0009655768c64bdeb2e877511632db8f	5a8bc65990b245e5a138643cd4eb9837	168	192.0	1	1	0	72.0	22.16
1	0009655768c64bdeb2e877511632db8f	no offer	0	NaN	0	0	0	263.0	0.00
2	00116118485d4dfda04fdbaba9a87b5c	f19421c1d4aa40978ebb69ca19b0e20d	168	216.0	1	1	0	120.0	0.00
3	00116118485d4dfda04fdbaba9a87b5c	f19421c1d4aa40978ebb69ca19b0e20d	576	630.0	1	1	0	120.0	0.00
4	00116118485d4dfda04fdbaba9a87b5c	no offer	0	NaN	0	0	0	473.0	4.09
5	0011e0d4e6b944f998e987f904e8c1e5	2298d6c36e964ae4a3e7e9706d1fb8c2	168	186.0	1	1	1	168.0	11.93
6	0011e0d4e6b944f998e987f904e8c1e5	3f207df678b143eea3cee63160fa8bed	0	6.0	1	1	0	96.0	0.00
7	0011e0d4e6b944f998e987f904e8c1e5	5a8bc65990b245e5a138643cd4eb9837	336	354.0	1	1	0	72.0	0.00

Figure 5: The main dataframe ("df_main") is displayed, which is an aggregated version of the "df_transcript_unfolded_portfolio" dataframe.

In a final step, the main dataframe is merged with the cleaned profile dataframe on the "person" attribute separately for train and test set customers (see Fig. 6).

	person	offer_id
0	0009655768c64bdeb2e877511632db8f	5a8bc65990b245e5a138643cd4eb9837
1	0009655768c64bdeb2e877511632db8f	no offer
2	0011e0d4e6b944f998e987f904e8c1e5	2298d6c36e964ae4a3e7e9706d1fb8c2
3	0011e0d4e6b944f998e987f904e8c1e5	3f207df678b143eea3cee63160fa8bed
4	0011e0d4e6b944f998e987f904e8c1e5	5a8bc65990b245e5a138643cd4eb9837
...	...	...
36850	fffad4f4828548d1b5583907f2e9906b	9b98b8c7a33c4b65b9aebfe6a799e6d9
36851	fffad4f4828548d1b5583907f2e9906b	f19421c1d4aa40978ebb69ca19b0e20d
36852	fffad4f4828548d1b5583907f2e9906b	f19421c1d4aa40978ebb69ca19b0e20d
36853	fffad4f4828548d1b5583907f2e9906b	no offer
36854	ffff82501cea40309d5fdd7edcca4a07	no offer

•••

consumption	age	became_member_on	income	gender_F	gender_M	gender_O	gender_nan
22.16	33	20170421	72000.0	0	1	0	0
0.00	33	20170421	72000.0	0	1	0	0
11.93	40	20180109	57000.0	0	0	1	0
0.00	40	20180109	57000.0	0	0	1	0
0.00	40	20180109	57000.0	0	0	1	0
...	...	...	...	...	...	...	...
18.04	34	20170123	34000.0	0	1	0	0
26.62	34	20170123	34000.0	0	1	0	0
12.18	34	20170123	34000.0	0	1	0	0
26.36	34	20170123	34000.0	0	1	0	0
0.00	45	20161125	62000.0	1	0	0	0

Figure 6: The main dataframe with demographic attributes is shown for train set customers (name of dataframe is "df_train"). For simplicity, not all columns are shown. The whole set of columns are: person, offer_id, offer_received_time, offer_viewed_time, promoted, viewed, completed, duration, consumption, age, became_member_on, income, gender_F, gender_M, gender_O, gender_nan.

5.2 Step 2: model 1

In model 1, the value sets of the demographic variables of "df_train" and "df_test" are simplified ($10,000 steps for "income", 10 year steps for "age", annual steps for "become_member_on") using the "simplify_feature_values" transformer. The second transformer "add_viewing_pref_features" adds viewing preference features. The whole model is a transformer pipeline and is shown in Fig. 7. After fitting with "df_train", the transformer pipeline produces the following new attributes as described in section 4.3: early_viewing_pref, late_viewing_pref, early_viewing_prob.

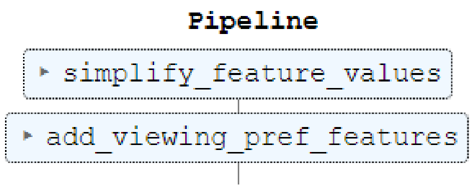

Figure 7: The composition of model 1 is shown, which consists of a transformer pipeline consisting of the transformers "simplify_feature_values" for reducing the value set of demographic attributes and "add_viewing_pref_features" for predicting viewing preferences and an early viewing probability.

5.3 Step 3: model 2

This offer-specific model is instantiated separately for each offer type. The entire model 2 is an estimator pipeline and optimized with a 3-fold cross-validation hyperparameter grid search (see Fig. 8). In the first step, carried out using the "select_specific_offer_and_no_offer" transformer, the outgoing dataframe of model 1 is taken and only the records related to the considered offer and all the no-offer records are filtered. Then the attribute "completion" is adjusted within the framework of the "adjust_completion_feature" transformer for the no-offer records and for all information offer instances as these have no completion records. Specifically, if a person has a higher hourly spend during an information offer instance than during a no-offer instance, then the information offer instance is marked as completed and vice versa. The last step consists in using the classifier "promotion_classifier" to decide whether or not it is fruitful for the person to receive a promotion of this considered offer, by providing a vector of "Yes" or "No" entries. We would like to point out that the scoring on the evaluation sets within the framework of the cross-validation uses the same metric as the scoring in the test set (for details on the metric, see Sec. 4.5). With respect to the hyperparameter grid search for optimizing the model, we used for the threshold parameters $p_1, \ldots, p_4$ (explained in Sec. 4.4) the following values: $p_1, \ldots, p_3 \in (0, 0.25, 0.5, 0.75, 1)$ and for $p_4 \in (0, 0.1, \ldots, 0.9)$.

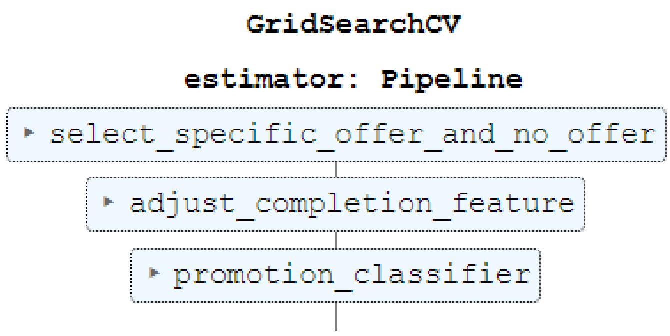

Figure 8: The composition of model 2 is shown, which consists of a cross-validation hyperparameter search on an estimator pipeline. The latter consists of the transformers "select_specific_offer_and_no_offer" and "adjust_completion_feature" and the classifier "promotion_classifier".

6 Results

6.1 Cluster analysis for all customers in the test set

In this section we present a cluster analysis of all customers using five histograms covering gender, income, age, membership begin year and viewing preference, with the bin counts for each cluster stacked on top of each other using different colours. Clustering was performed in a five-dimensional space spanned by all of the above demographic attributes, each represented by a histogram. The optimal number of clusters was determined using the kmeans++ algorithm with the help of the silhoutte score and is limited here to five for simplicity.

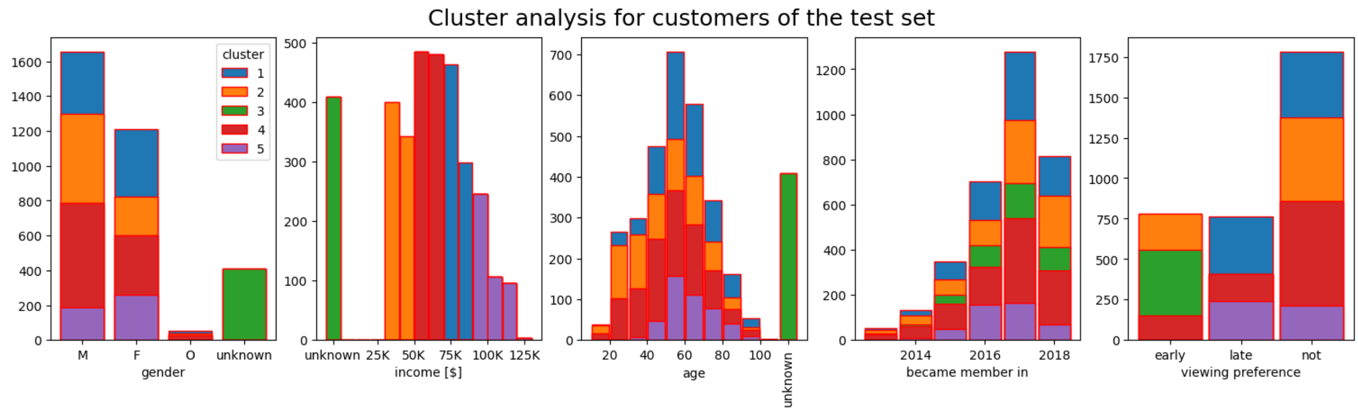

Figure 9: Cluster analysis for all customers in the test set using 5 clusters. In terms of clustering, we can see that people who do not provide information on gender, income and age are all early viewers. Customers earning more than $70,000 are slightly dominated by women. Those earning more than $90,000 are not under 40 years old and do not tend to view offers within two days.

6.2 Analyzing the promotion strategy for each offer type

In this subsection we briefly analyze, for each offer type (grouped by main offer types), which customers we would and would not recommend for a promotion using the strategy presented here. Each figure represents the results of one offer type analysis and consists of three rows of graphs covering the demographic information of gender, income, age, membership begin year and viewing preference. The top row of graphs shows data for all customers and highlights the customers who should be promoted according to the strategy found. The middle row contains data only for customers to be promoted, while the bottom row contains data only for customers not to be promoted. The customer demographics in the corresponding histograms have been clustered in the five-dimensional space spanned by all the aforementioned demographic attributes. Clustering was performed separately for target and non-target customers. In each analysis, the values for the "irr" and "nir" metrics, the threshold parameters of model 2 (p_1, p_2, p_3, p_4), as well as the metric score and the target (non-target) to test set ratio or "coverage" of all test set customers are shown. The values for the threshold parameters shown in this investigation, each multiplied by 100, refer to the percentile of the corresponding prediction probability values determined in the training set.

6.2.1 Diagrams for BOGO offers

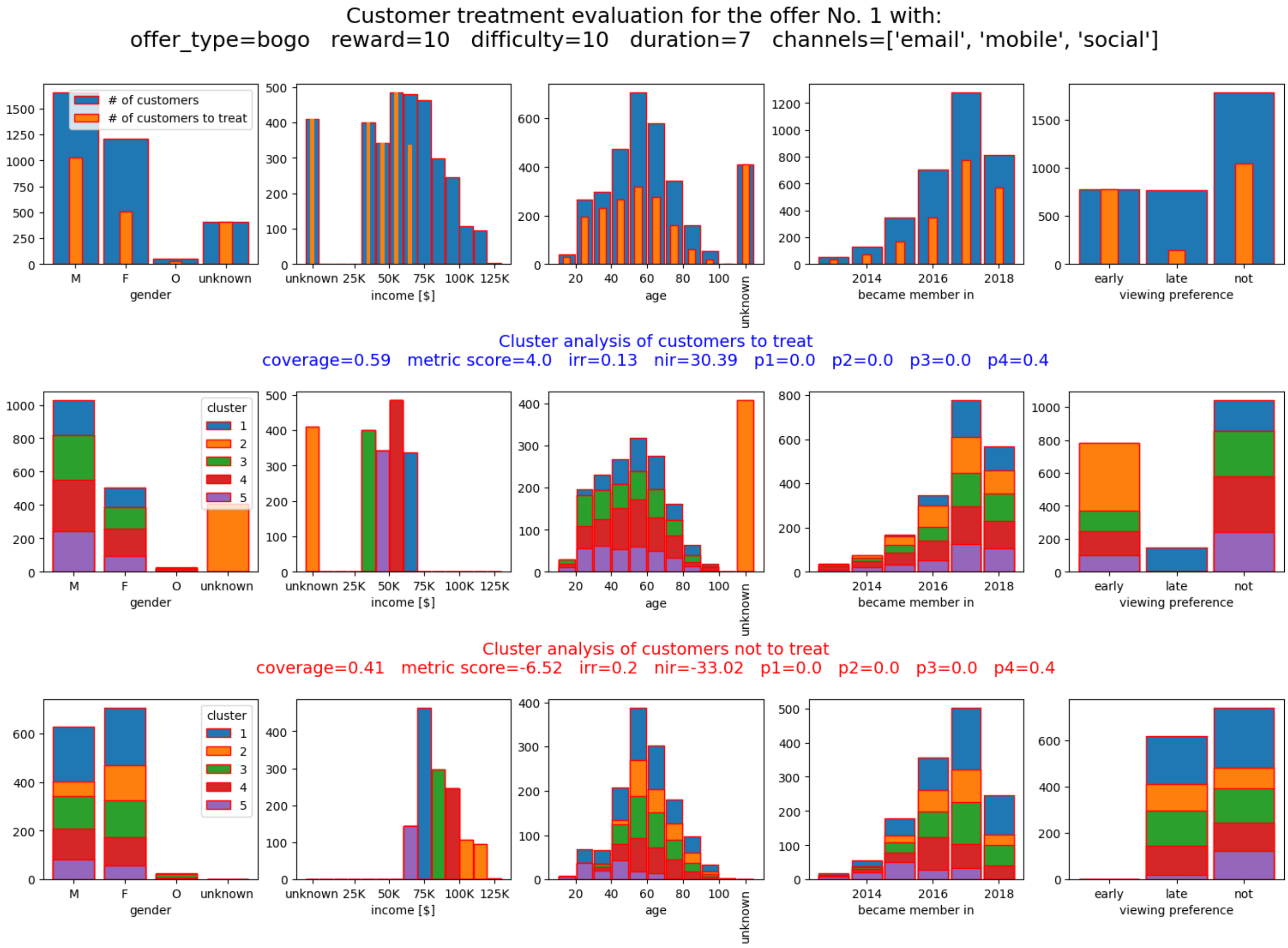

Figure 10: Customers with incomes up to $70,000 are worth targeting with this offer. Those who do not provide demographic information should be included as well. The considered customers tend to view early or rather not.

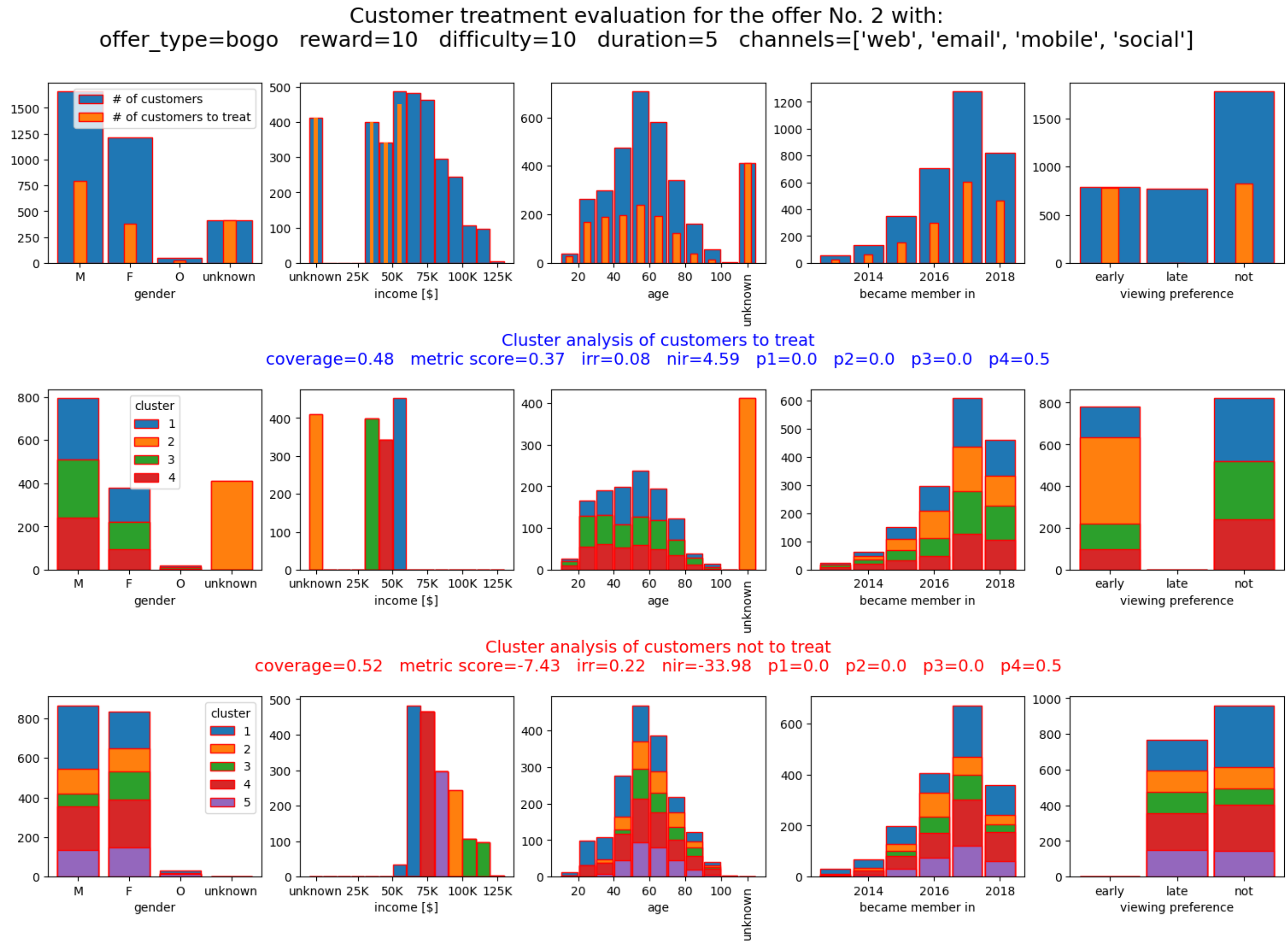

Figure 11: Customers with incomes up to \$60,000 are worth targeting with this offer. Those who do not provide demographic information should be included as well. The considered customers tend to view early or rather not.

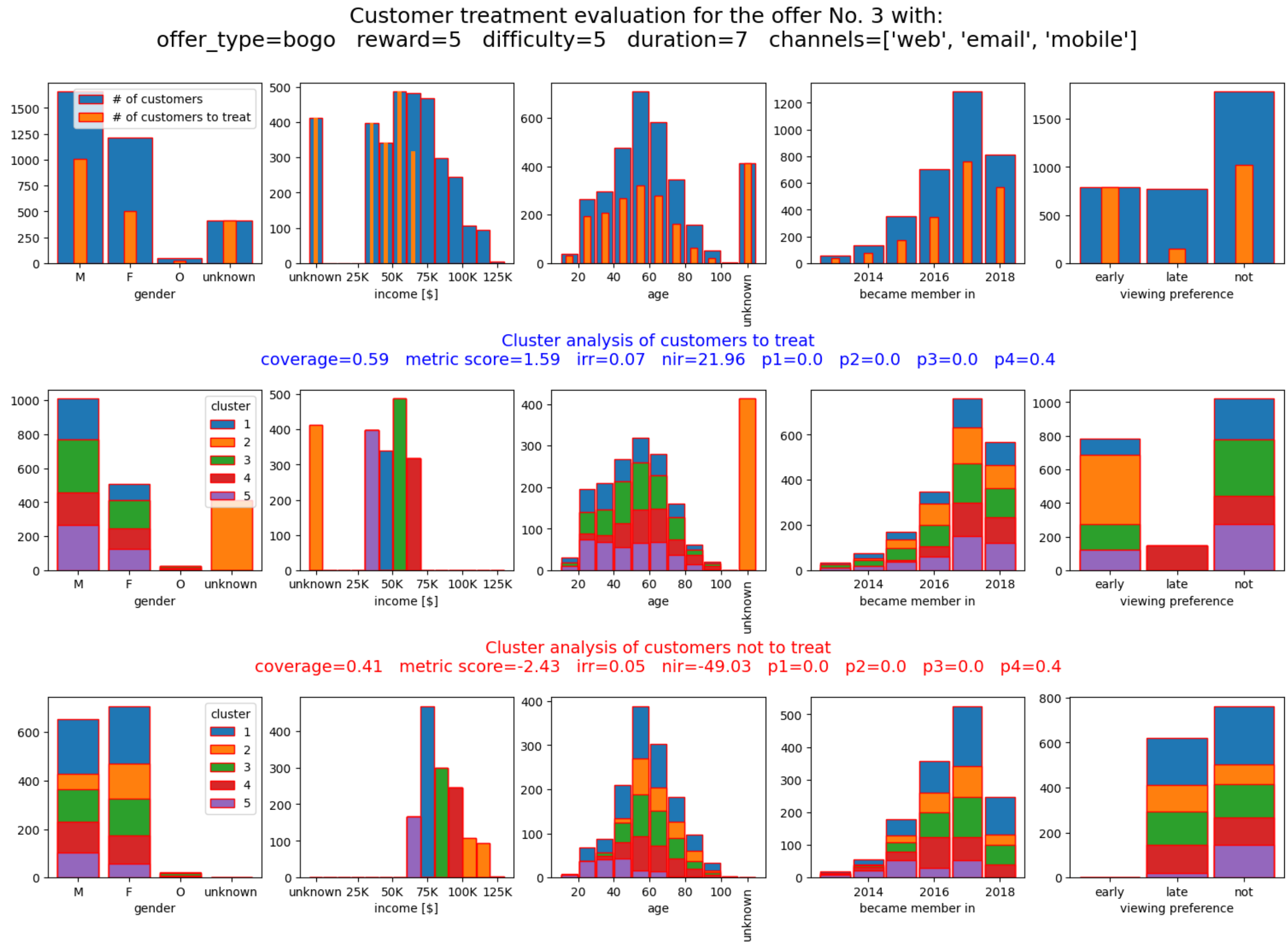

Figure 12: Customers worth targeting with this offer are basically the same as in offer type 1. However, this offer type has a weaker value in the metric score.

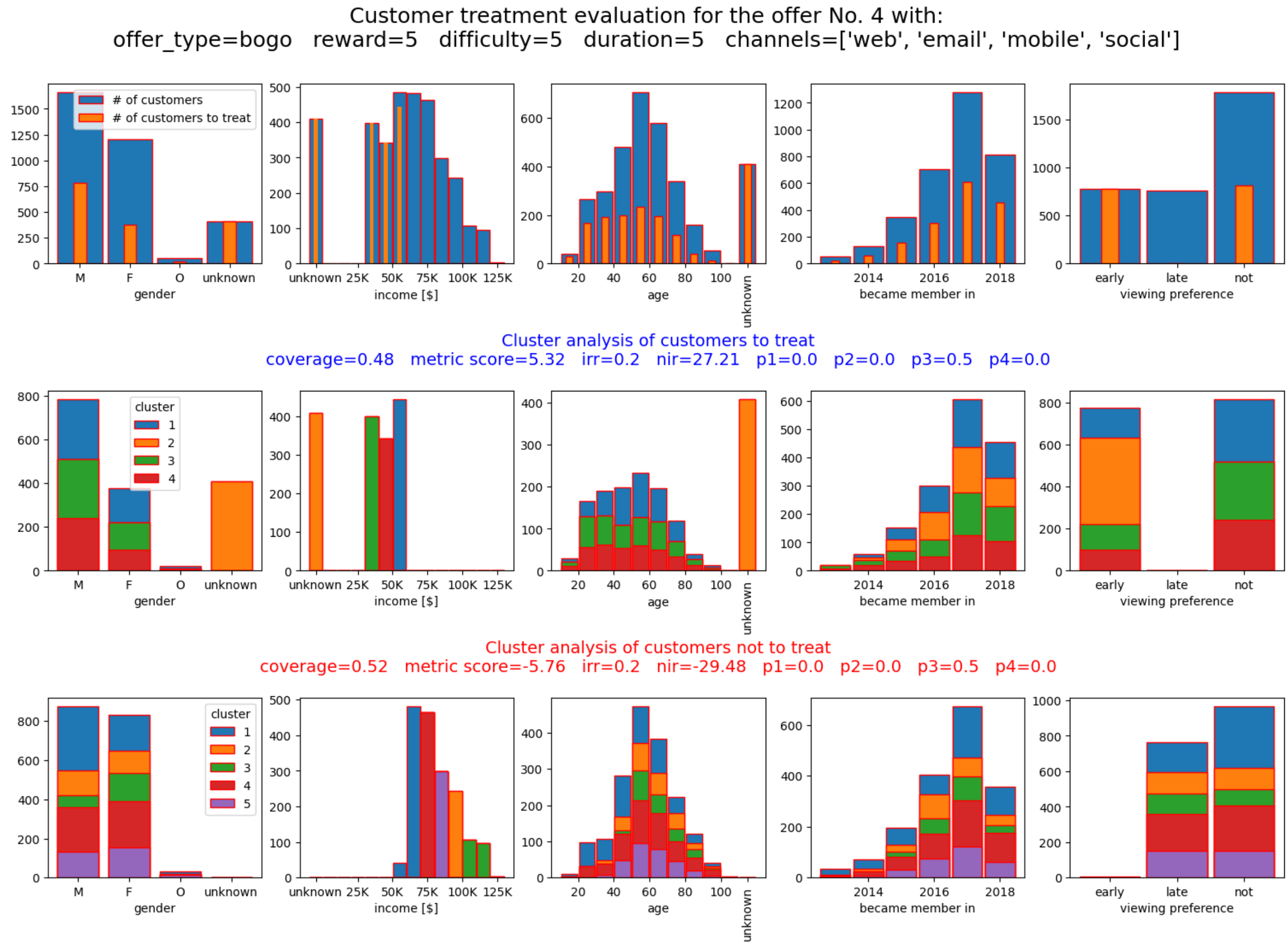

Figure 13: Customers worth targeting with this offer are basically the same as in offer type 2. However, this offer type has a stronger value in the metric score.

6.2.2 Diagrams for discount offers

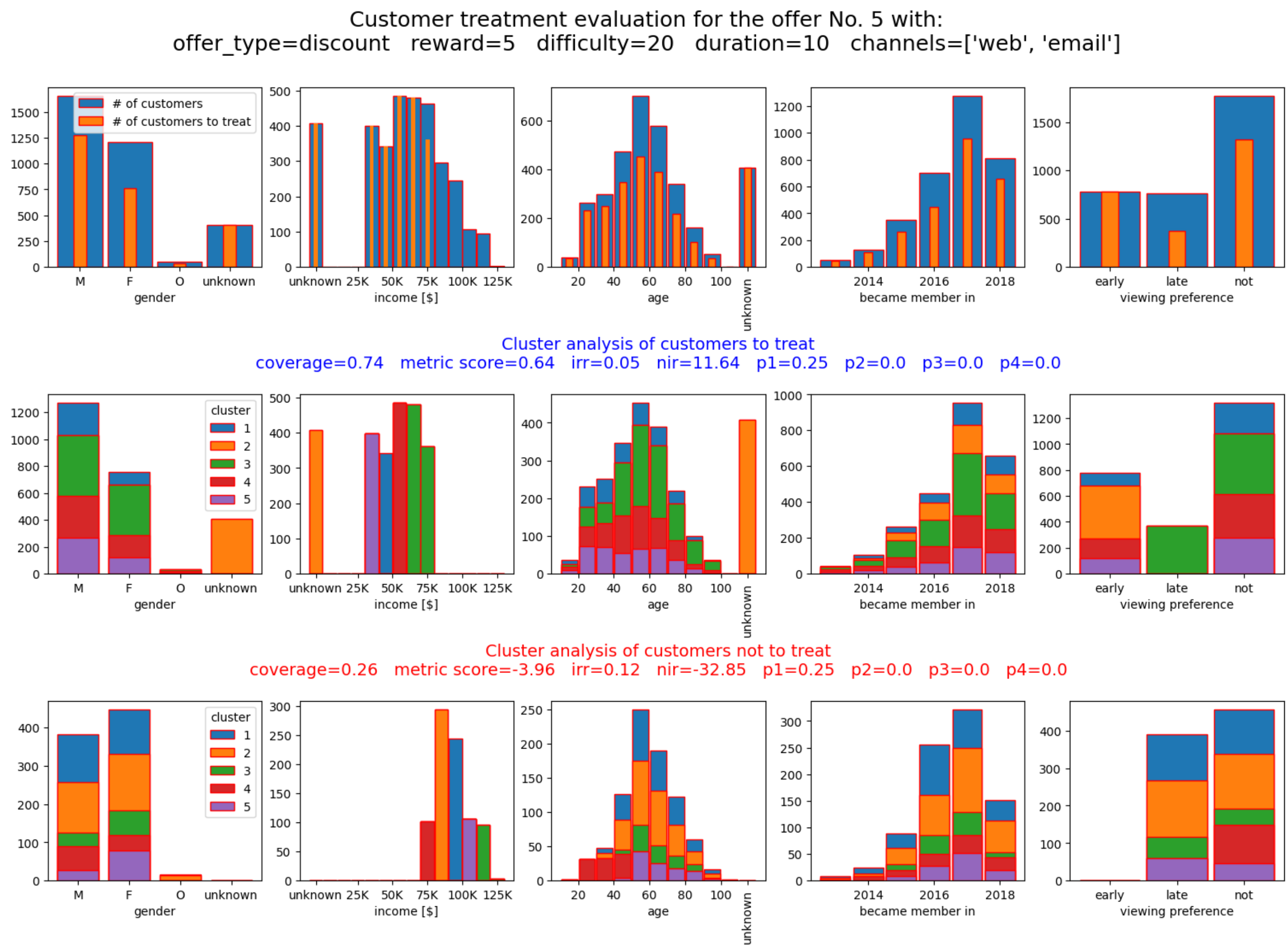

Figure 14: Customers with incomes up to $80,000 are worth targeting with this offer. Those who do not provide demographic information should be included as well.

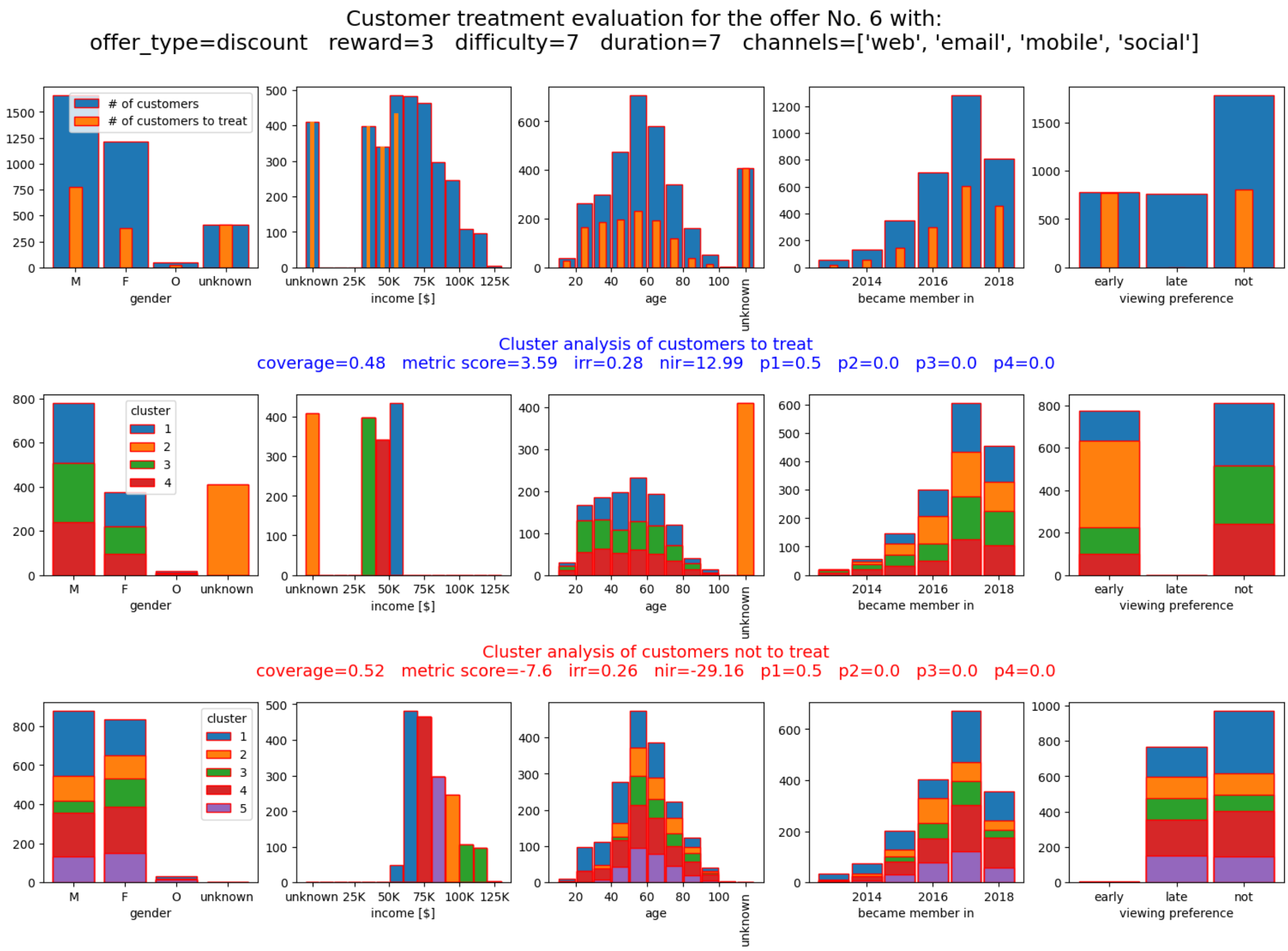

Figure 15: Customers worth targeting with this offer are basically the same as in offer type 2 or 4.

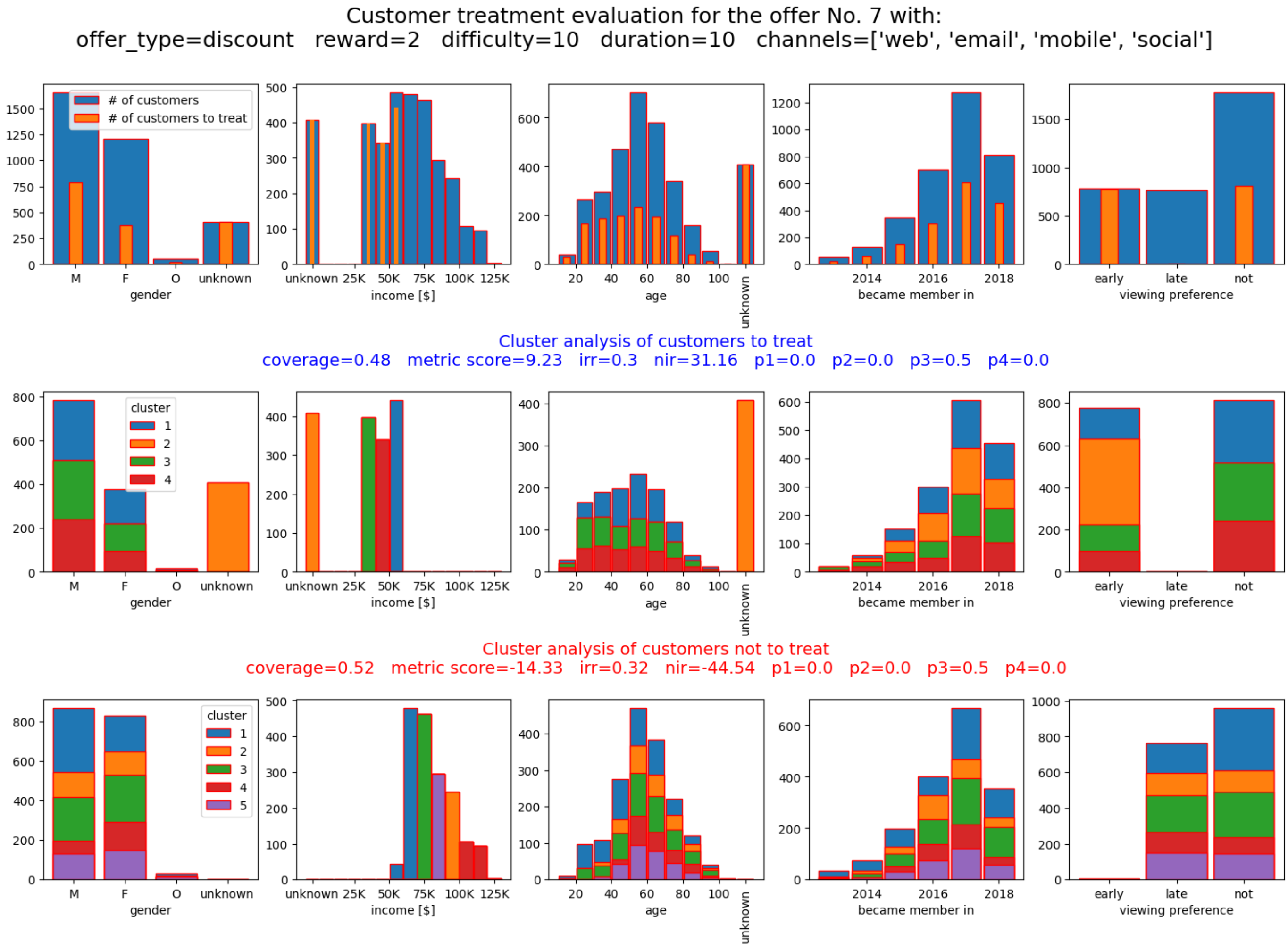

Figure 16: Customers worth targeting with this offer are basically the same as in offer type 2,4 or 6. However, this offer type has a stronger value in the metric score.

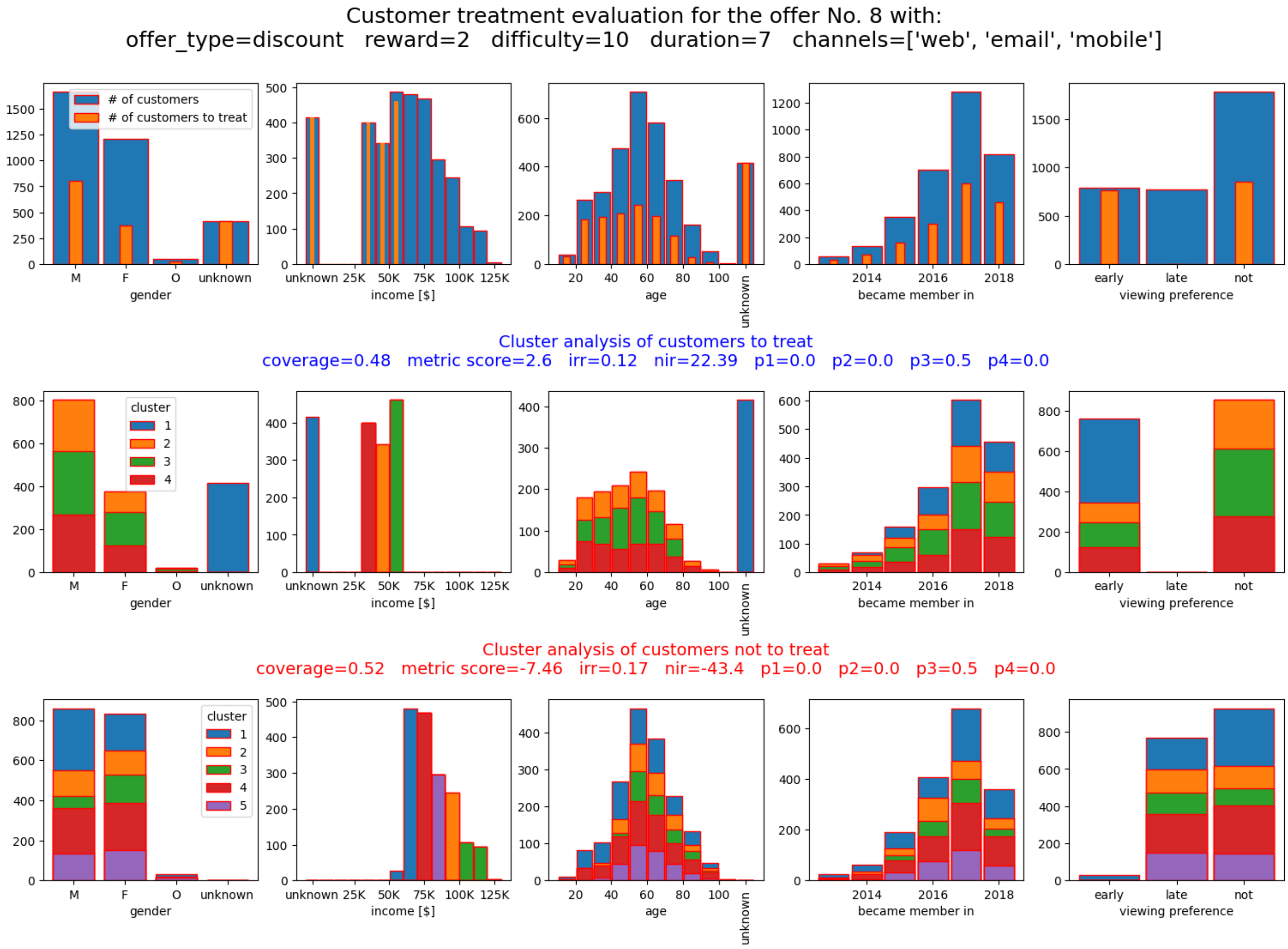

Figure 17: Customers worth targeting with this offer are basically the same as in offer type 2,4, 6 or 7.

6.2.3 Diagrams for informational offers

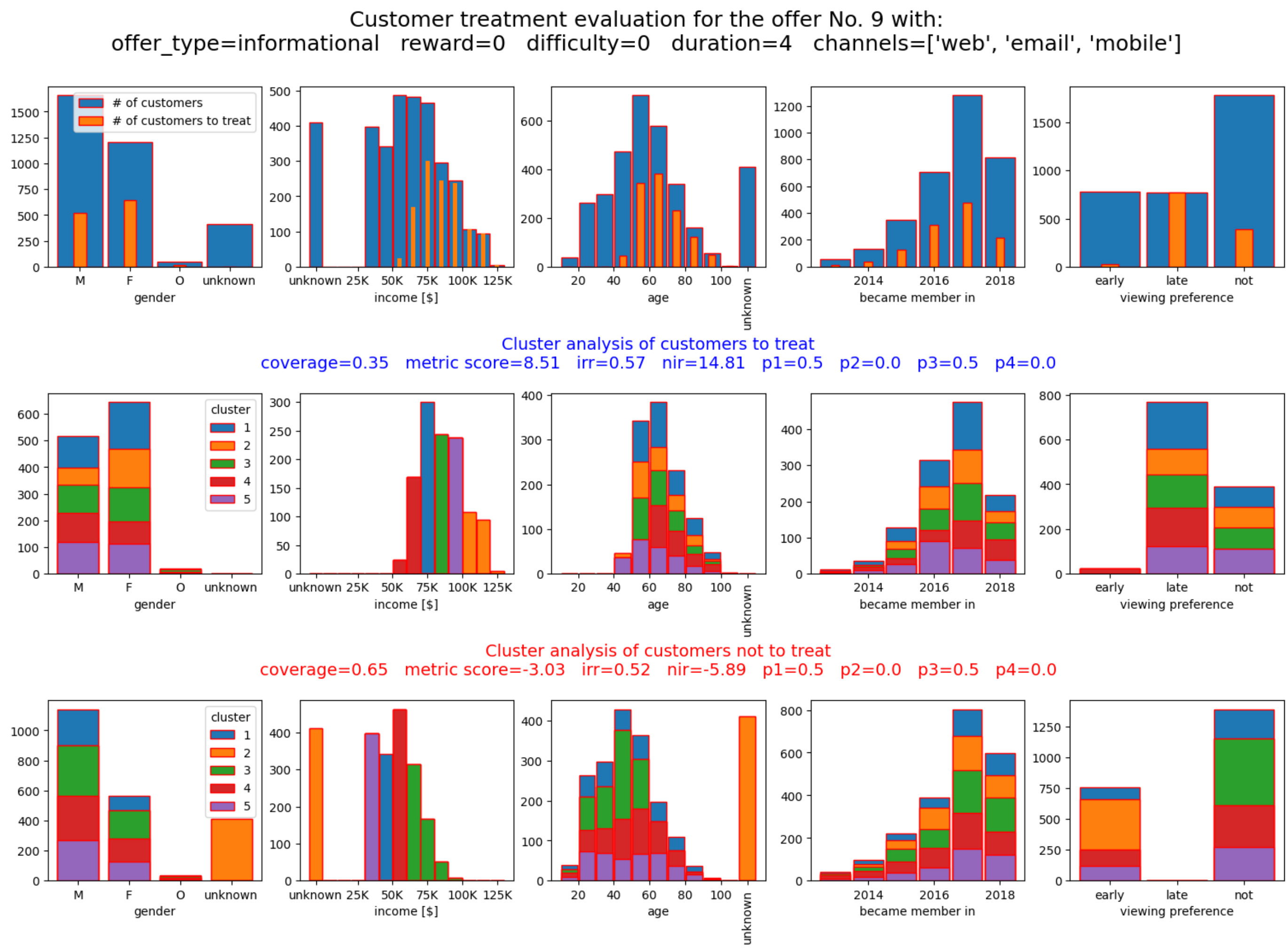

Figure 18: A majority of customers earning at least \$70,000 should be targeted by this offer. These customers are more likely female and tend to view lately or rather not. Customers not providing demographic information should be excluded.

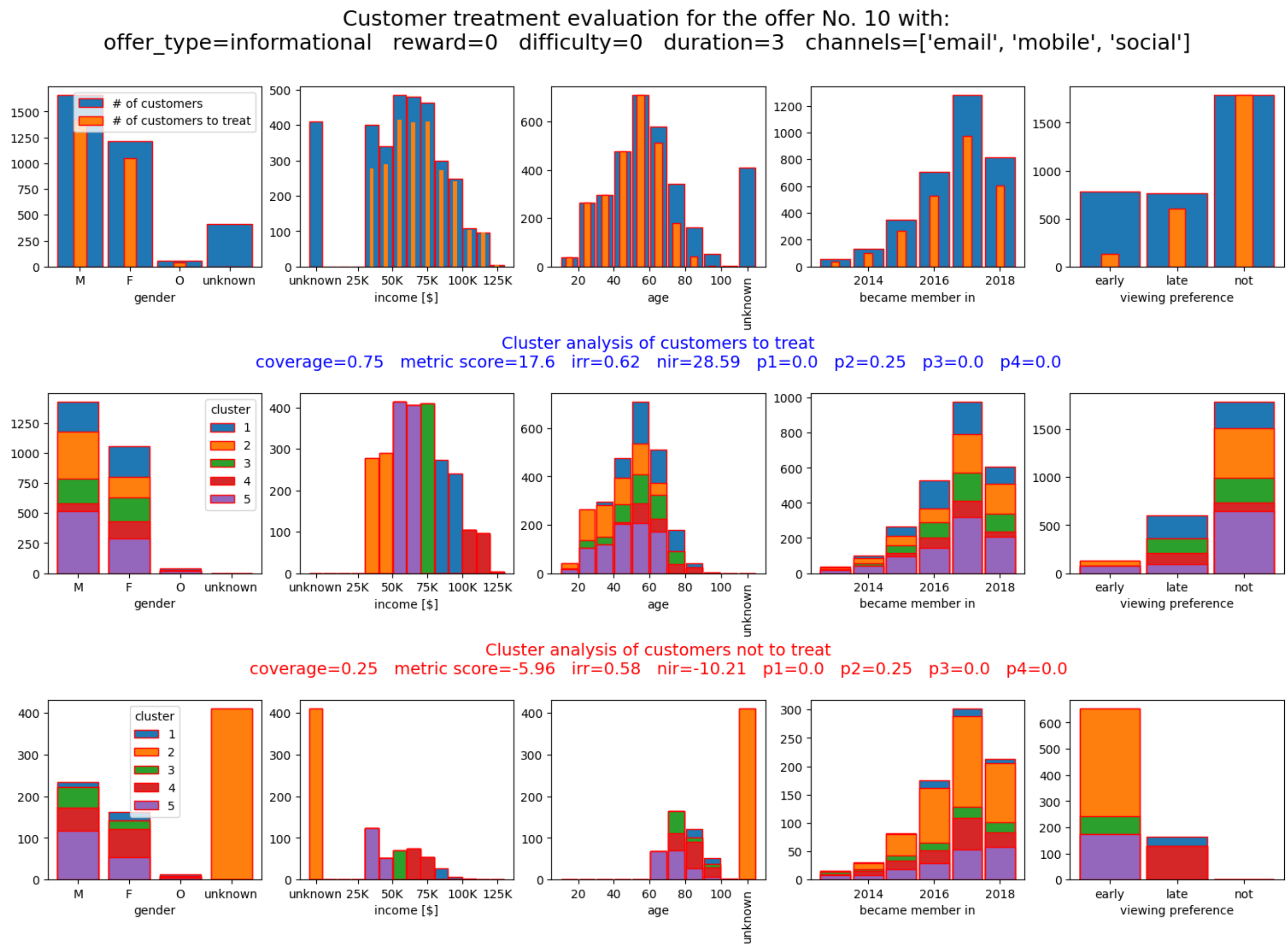

Figure 19: A staggering 75% of customers are worth targeting with this offer, with the highest metric score found. Young and middle-aged customers are particularly worth targets. These customers are more likely to view lately or rather not. Customers who do not provide demographic information should be excluded. Of those with an income of $60,000 or less, only those who tend to view lately are in the target group.

7 Conclusion

In this study, we can conclude that BOGO and discount offers are primarily targeted at low- and middle-income customers. The best BOGO offer is offer type No. 4 and is designed for customers earning up to $60,000. It has a 5-day validity period and a $5 threshold. The best discount offer is offer type No. 7, which has a 7-day validity period, a $10 difficulty level and a $2 reward. Offer type No. 4 is also designed for customers earning up to $60,000, but has a metric score that is twice as high as the best BOGO offer. Nonetheless, both the BOGO and discount offers do not target higher income customers. On the other hand, we can see that the information offers are intended for high income customers who do not tend to look at offers immediately. In contrast to the other types of offers, customers who do not provide demographic information are not included in the target group. It may be that people who do not provide any information about themselves do not want to be informed by the app without any direct benefit. In fact, the second information offer (offer type No. 10) with the 3 day duration has even the highest metric score of all offers and a coverage of 75%. This seems to be the best choice of offer to send for most customers who provide demographic information. The remaining customers should receive offer No. 5 or 7 depending on their demographics. However, it is difficult to make an exact decision within the scope of this research, as a finer screening of the parameters and more data would probably provide more clarity. With regard to the choice of demographic attributes for a possible promotion decision, we can say that the starting date of membership does not seem to be very characteristic, at least when looking at the year. However, the impact of the "income" attribute, as well as the decision to provide demographic data at all, seems to be very profound in this context. There may also be a tendency for targeting via social networks to increase the metric score.

8 Discussion and outlook

There are several ways to improve this survey. One simple way is to increase the sampling rate in the four-dimensional hyperparameter grid. This is particularly useful when hardware is available with a large number of computing cores, as each parameter setting can be computed in parallel. So far, a hyperparameter grid of 1250 constellations has been used for the threshold parameters $p_1, \ldots, p_4$.

We would also like to point out that a lot of data representing people under the influence of multiple offers have been removed. More data with single offer influence or a modeling approach to deal with multiple offer influence could be very helpful. In this respect, from a scientific point of view, it is useful to carry out a large scale or finite size analysis to scrutinize the stability of the target group when the amount of data increases.

In this research it is also not clear how customers behave depending on the order of the offers or the time between them. Even the frequency of spending may be characteristic but is not investigated here. The choice for the used metric is also an area for improvement. For example, the consumption of promoted customers who do not complete the offer, as well as non-promoted customers who would not have completed the offer, is not taken into account in terms of their consumption, only the number of these customers appears in the irr metric.

Concerning our predictor model (model 2) there is also a possibility for improvement. In model 2, we have treated promoted and non-promoted customers together in the offer instances with state labels II and IV respectively (see table in Sec. 4.4). In this sense, we group people who do not view the offer together with people who might view it. We can justify this approach due to its simplicity and the lack of data, as well as the fact that our ideal customer approach so far examines separately the event of viewing on consumption in model 2 and the viewing behavior in the framework of model 1. Accordingly, one could tackle this issue and build a classifier for model 2 that includes all six states of the offer instances, but this would increase the hyperparameter space and the need for data and powerful hardware.

With respect to our results, we noticed that the membership start date in terms of years does not seem to be a characteristic demographic attribute. This annual date mapping was done for simplicity and to avoid overfitting. Perhaps a more fine-grained version that takes into account seasonal influences, for example, might lead to a more distinctive demographic characteristic. The time between signing up in the app and receiving the offer may also have an effect. For example, someone who has just signed up for the app might be very receptive to advertising. And of course, more demographic data, even if it is hard to get, seems to be helpful. But also other categories of data like employment or geographic data may improve the modeling approach. So if people live or work near a Starbucks café or store, they might go there more often and perhaps buy more if they receive ads, even if they are not belonging to the best target group within the current investigation.

A final point of further investigation we would like to mention concerns the estimators used. In particular, we used throughout the "LogisticRegression"-estimator as regressors with a fixed setting for the regularization, number of iterations and so on. This is because it was used successfully on a simpler promotion problem formulated by Starbucks (see Ref. [2]).