UDACITY

Capstone project

An application of Random Forest Classifier to Starbucks' direct marketing system

Saverio Pertosa

October 2020

1 Description of the task (from Udacity)

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.

2 Proposal

My proposal consists of the application of a Machine Learning technique called "Random Forest Classifier" to predict the likelihood that an offer, characterized by a given set of features and customer's data, will be completed.

Being able to correctly predict users' behavior is crucial for any business, since knowing who to send an offer to would:
- increase the revenue of the campaign by maximizing the number of completed offers,
- decrease the chance of sending offers to people who do not appreciate them, and thus might get frustrated of the company,
- optimize marketing costs.

Thus, a marketing campaign performance is deeply linked to the ability to spot correctly customers who are more prone to accept an offer.

The project will aim at classifying offers sent based on two sets of categories.

- The first (later called **Y1**) consists in **4 labels**:
  - 0: Viewed the offer + Completed the offer
  - 1: Viewed the offer + Did not complete the offer
  - 2: Did not view the offer + Completed the offer (this includes the cases in which a customer first completes the offer, and then views it)
  - 3: Did not view the offer + Did not complete the offer.
- The second (later called **Y2**) consists in **2 labels**:
  - 0: Customer completed a specific offer being influenced by it (equal to the previous label 0)
  - 1: All other cases (equal to the previous labels 1-2-3)

Contents

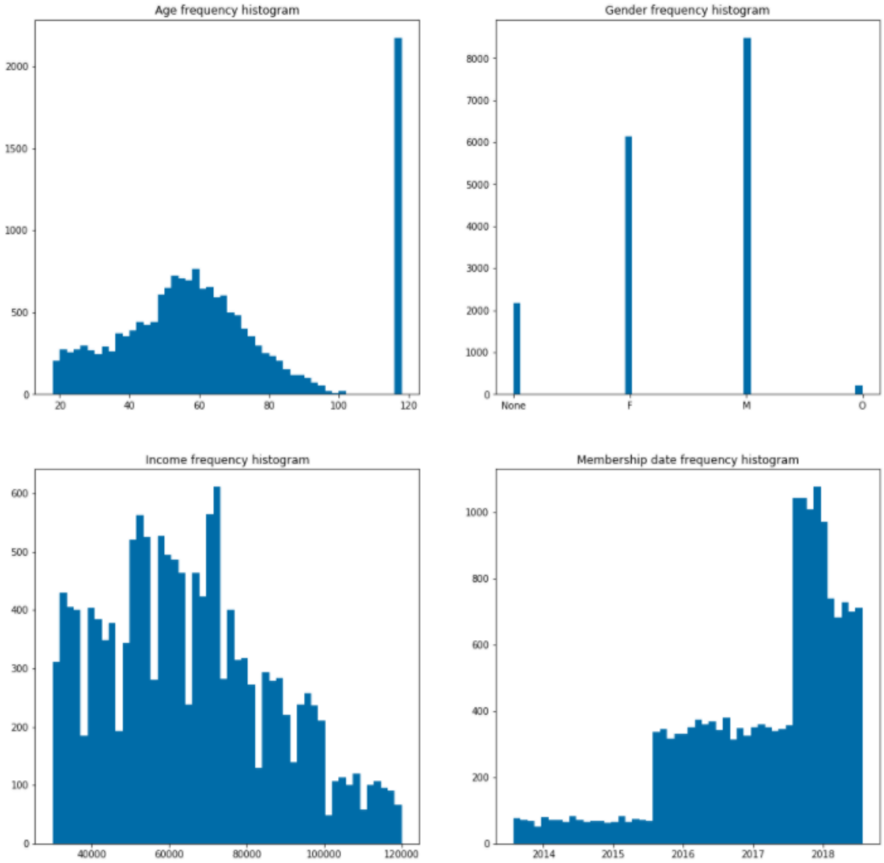
Part I

Data and inputs

Starbucks provided 3 .json files, containing simulated data mimicking customer behaviour.

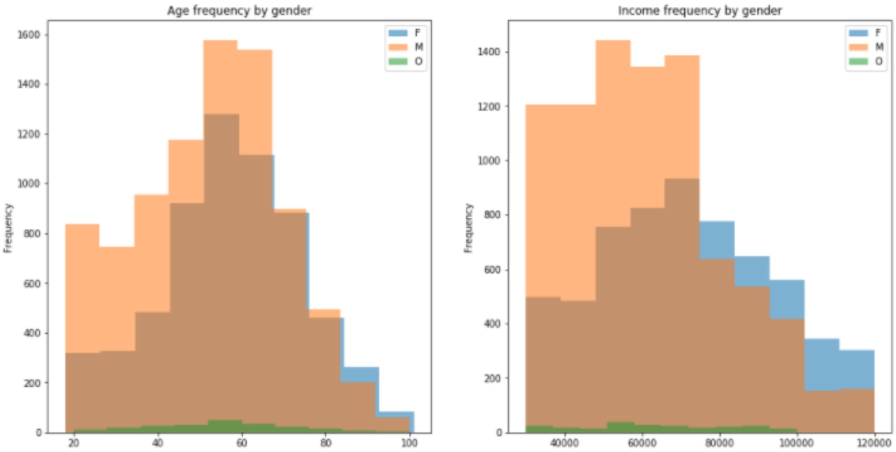
3 Datasets' description

- **profile.json** - Customers' data *(17000 users x 5 fields)*
  - gender: (categorical) M, F, O, or null
  - age: (numeric) missing value encoded as 118
  - id: (string/hash)
  - became_member_on: (date) format YYYYMMDD
  - income: (numeric)



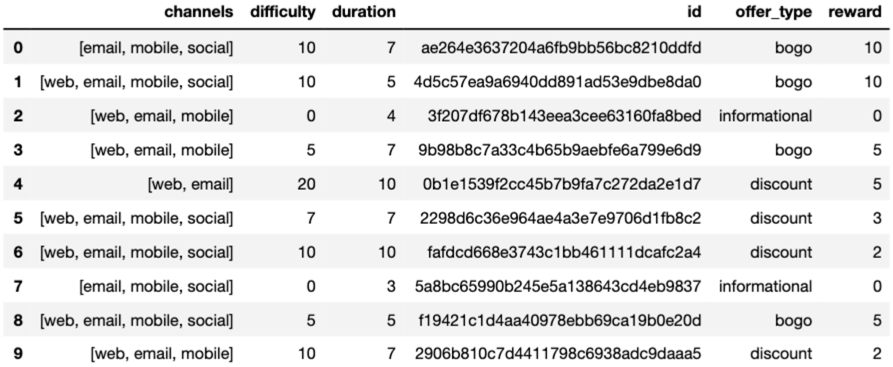
Beyond some missing values (missing age is encoded as *118*), the distributions do not show outliers or noticeable issues.

The offers were sent mostly to customers who recently subscribed membership to Starbucks, who are on average 54 years old and have a salary of about 65k $. Both distributions are slightly skewed, but still average and median values are quite similar.



Also, the gender-specific distributions of age and income have still similar average and median values. On average female customers in the dataset both are older and have a higher income.

- **portfolio.json** - Offers sent during 30-day test period *(10 offers x 6 fields)*
  - reward: (numeric) money awarded for the amount spent
  - channels: (list) web, email, mobile, social
  - difficulty: (numeric) money required to be spent to receive reward
  - duration: (numeric) time for offer to be open, in days
  - offer_type: (string) bogo, discount, informational
  - id: (string/hash)



	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

All offers were sent by e-mail and at least another channel.

- **transcript.json** - Event log *(306648 events x 4 fields)*
  - person: (string/hash)

- event: (string) offer received, offer viewed, transaction, offer completed
- value: (dictionary) different values depending on event type
- offer id: (string/hash) not associated with any "transaction"
- amount: (numeric) money spent in "transaction"
- reward: (numeric) money gained from "offer completed"
- time: (numeric) hours after start of test

4 Some initial statistics on completion

Based on the *Transcript* data, we can do some first initial assessments on the performance of the campaign done

- Total offers sent: 76277
- Total offers viewed: 57725
- Total offers completed: 33579
- Total transactions: 138953
- High-level % of opened offers: 75.68%
- High-level % of completed offers: 44.02%
- High-level % of completed offers, if viewed: 58.17%

Part II

Data cleaning and feature engineering

5 Data cleaning

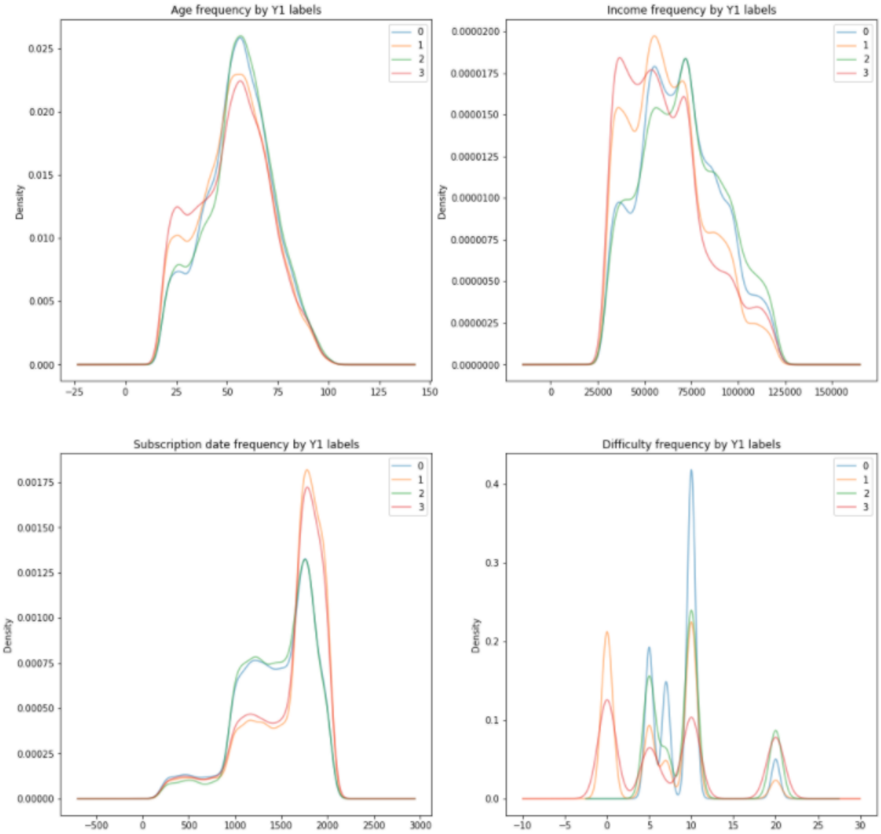
First of all, I joined the three databases, and filled in missing values.

To create an analytic database ready to be passed to a Random Forest algorithm I then made every row represent an offer sent to a specific customer. After that, I added two columns with the two events following the offer received.

Based on that, we can tell which offer was opened and/or completed.

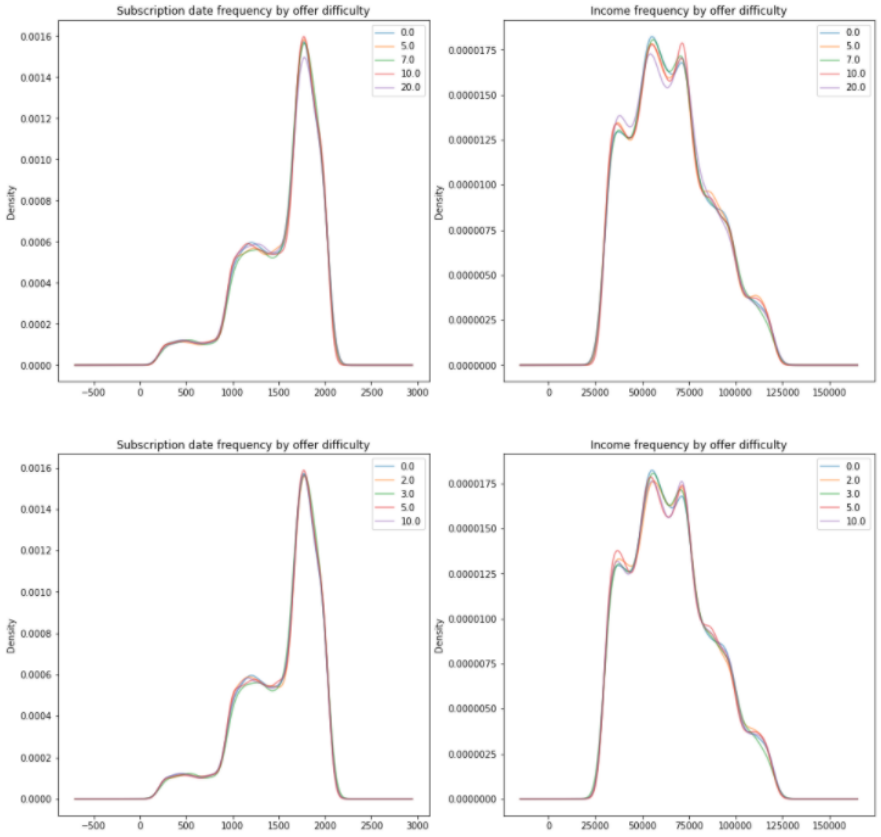
As explained in the **Proposal** section, the project will use two sets of categories:

- The first (later called **Y1**) consists in **4 labels**:
  - 0: Viewed the offer + Completed the offer
  - 1: Viewed the offer + Did not complete the offer
  - 2: Did not view the offer + Completed the offer (this includes the cases in which a customer first completes the offer, and then views it)
  - 3: Did not view the offer + Did not complete the offer.



Some first interesting insights are:

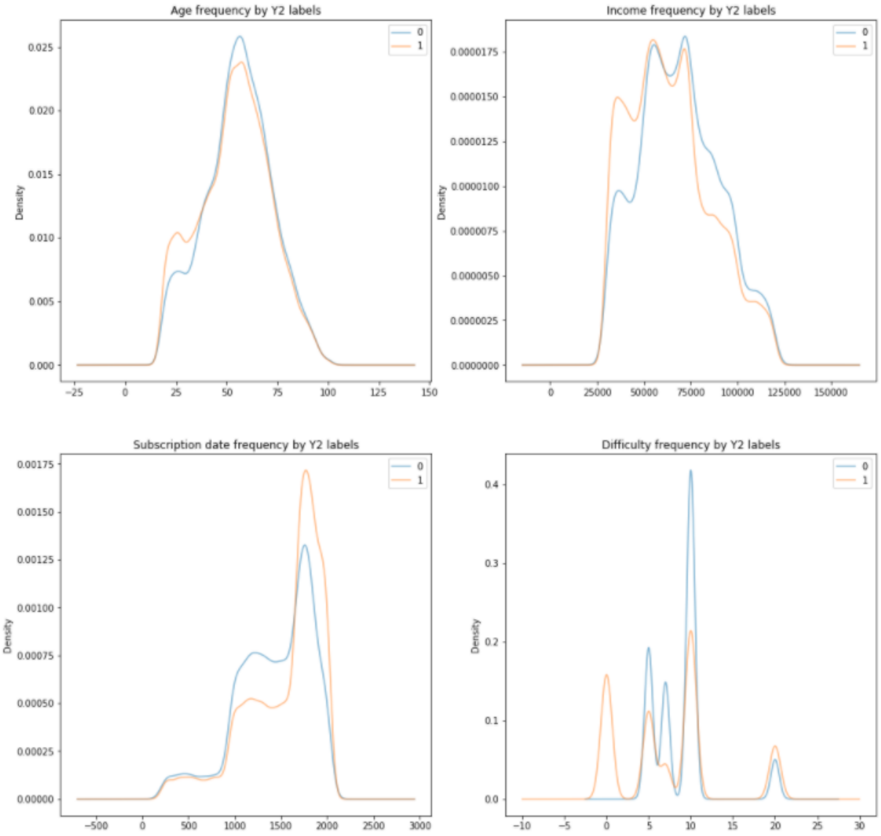
  - Younger people tend to neither view nor complete the offer.
  - Income is positively correlated with the likelihood of completing the offer (regardless of actually seeing the offer). This is of course a consequence of the higher purchasing power/lower price sensitivity.
  - Interestingly enough, the longer the customer has been a member, the lower the chance that they will complete an offer.
  - Most of the viewed-completed offers had an average (10) difficulty.



It's curious to see that the difficulty and the reward of offers don't depend on those variables that most correlated with the probability of completing an offer: income and membership duration.

I think it would be interesting to experiment with offers having a level of difficulty customized on users' features.

- The second (later called **Y2**) consists in **2 labels**:
  - 0: Customer completed a specific offer being influenced by it (equal to the previous label 0)
  - 1: All other cases (equal to the previous labels 1-2-3)

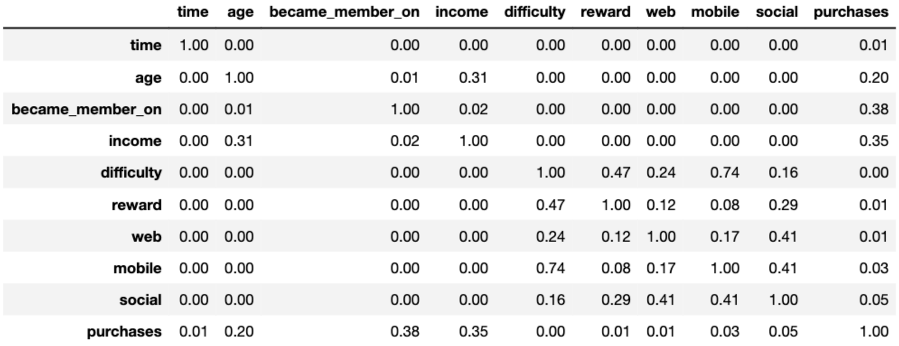


The density distributions with Y1 labeling provide similar information to what was highlighted before.

6 Feature engineering

I converted the *became_ member_ on* variable into an integer (by calculating the difference in days from 2013-01-01, a date preceding the initial membership of every customer in the dataset) and computed the number of transactions occurred *outside* the context of an offer as a proxy of the natural proclivity of a user to buy Starbucks' products (name: ‘purchases‘).

The correlation matrix shows that the features have overall low correlation values to one another. Thus, none will be excluded from the analysis.



	time	age	became_member_on	income	difficulty	reward	web	mobile	social	purchases
time	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
age	0.00	1.00	0.01	0.31	0.00	0.00	0.00	0.00	0.00	0.20
became_member_on	0.00	0.01	1.00	0.02	0.00	0.00	0.00	0.00	0.00	0.38
income	0.00	0.31	0.02	1.00	0.00	0.00	0.00	0.00	0.00	0.35
difficulty	0.00	0.00	0.00	0.00	1.00	0.47	0.24	0.74	0.16	0.00
reward	0.00	0.00	0.00	0.00	0.47	1.00	0.12	0.08	0.29	0.01
web	0.00	0.00	0.00	0.00	0.24	0.12	1.00	0.17	0.41	0.01
mobile	0.00	0.00	0.00	0.00	0.74	0.08	0.17	1.00	0.41	0.03
social	0.00	0.00	0.00	0.00	0.16	0.29	0.41	0.41	1.00	0.05
purchases	0.01	0.20	0.38	0.35	0.00	0.01	0.01	0.03	0.05	1.00

In the end, since Decision Trees are both robust to outliers and not impacted by feature scales, it is not necessary to modify or cut out values

Part III

Model

Since the problem is a classification one, in a supervised setting, it can be tackled with a Random Forest Classifier, one of the most known learning ensemble methods.

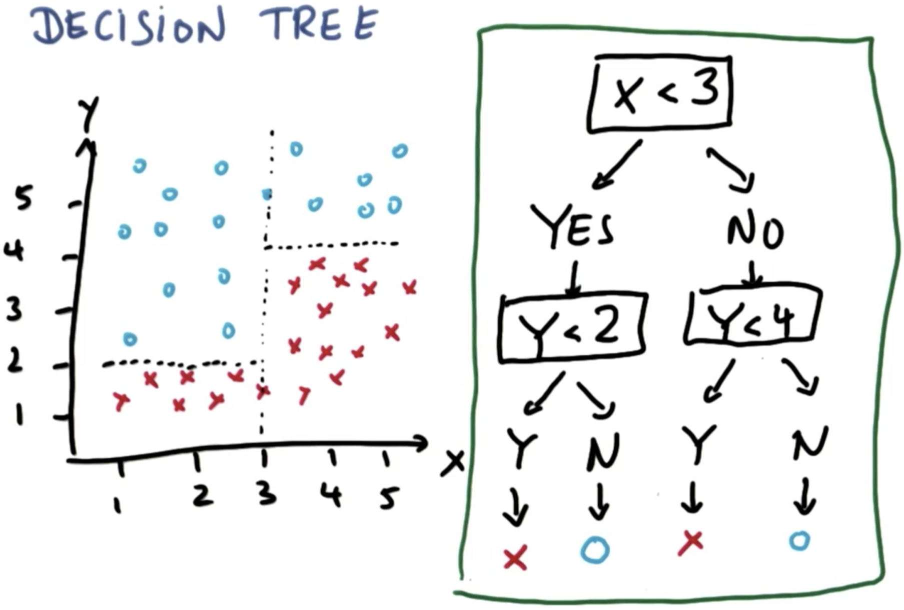
7 Random forest classifier

Decision tree

A *Decision tree* is a learning method whose goal is to create a model predicting the value of a target variable by learning simple decision rules inferred from the data features.

They are used to classify non-linearly separable data, and each node defines a partition and a local linear rule on top of it.

The following image, taken from a lesson in Udacity's *Intro to Machine learning*, explains how this works, showing a 2-level-deep decision tree with two *internal* nodes, each one with two *leaf* nodes.



Random forest classifier

A *Random forest* consists of an ensemble of multiple decision trees, that are merged in the end to get a more accurate and stable prediction. When used in classification settings, the final assigned label will be determined by a "majority" vote among all trees.

Random forests models build "forests" such that each one employs only a subset of the available features and uses bootstrap replicas, meaning that they subsample the input data with replacement.

8 Hyperparameters

The models' hyperparameters have been chosen with a *GridSearch* on some of the parameters offered by scikit-learn's *RandomForestClassifier*.

- *n_ estimators*: The number of trees in the forest;
- *max_ depth*: The maximum depth of the tree;
- *min_ samples_ split*: The minimum number of samples required to split an internal node;
- *min_ samples_ leaf*: The minimum number of samples required to be at a leaf node.

9 Performance metrics

For each model, a few performance metrics have been computed on the test set. In this section they will be described analytically

- **Accuracy** Accuracy is simply the ratio between the number of correct predictions and the sample size. It is computed as

  $$Accuracy(\hat{y}, y) = \frac{1}{n}\sum_{i} \mathbf{1}(\hat{y} = y)$$

  where n is the sample size and $\mathbf{1}(.)$ is a function taking 1 as value if the condition is satisfied, 0 otherwise.

- **Balanced accuracy** Balanced accuracy avoids potential biases implied in having a sample skewed towards one class. It is computed as the average of recall obtained on each class.

- **Recall** Recall is the ratio between the number of correctly identified positive results and the number of all samples that should have been identified as positive.

  $$Recall = \frac{TP}{TP + FN}$$

- **Precision** Precision is the ratio between the number of correctly identified positive results and the number of positively labeled predictions.

  $$Precision = \frac{TP}{TP + FP}$$

- **Confusion matrix** A table-like way to see the distribution of FP, FN, TP and TN.

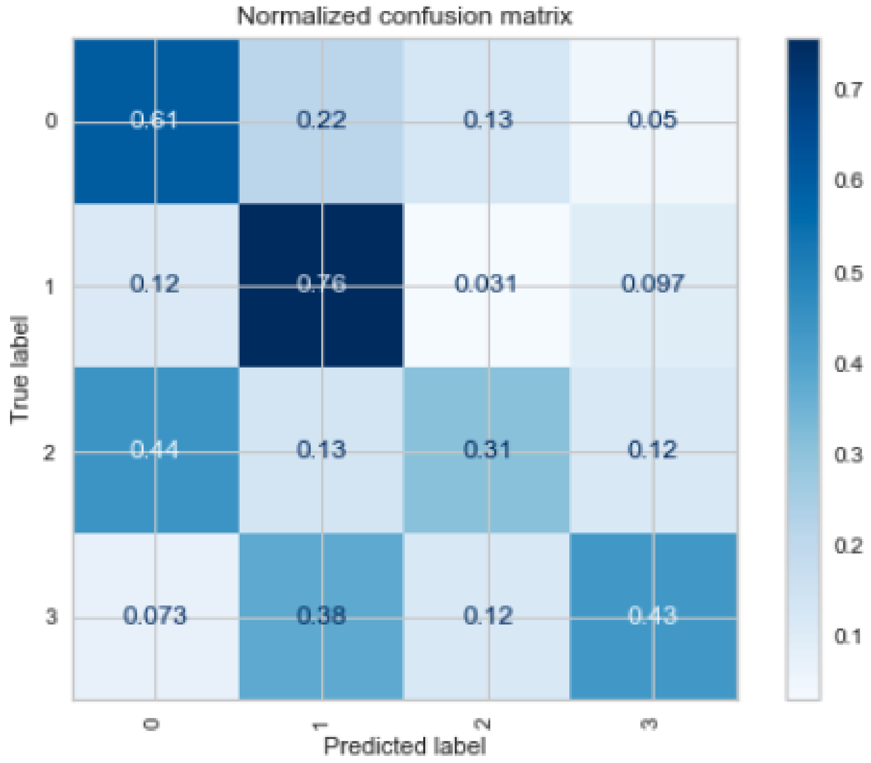
- **ROC-AUC** The AUC (*Area under curve*) shows the strength of the relationship between false positives and true positives. It's computed as the area under the ROC (*Receiver Operating Characteristic*) curve, which describes the locus of the trade-off between **Sensitivity** (computed as $\frac{TP}{TP+FN}$) and **Specificity** (computed as $\frac{TN}{TN+FP}$) of a binary classifier.
In the **Y1** model, 4 ROC curves will be computed, one per label.

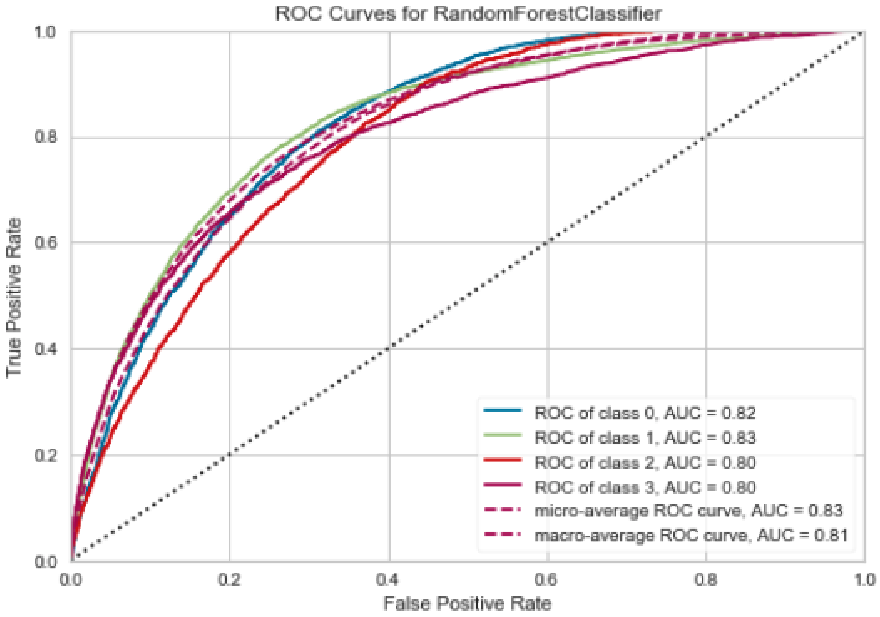
10 Implementation

After training on 80% of the data points, leaving the remaining 20% for testing, each model shows quite good performance levels.

Y1 model

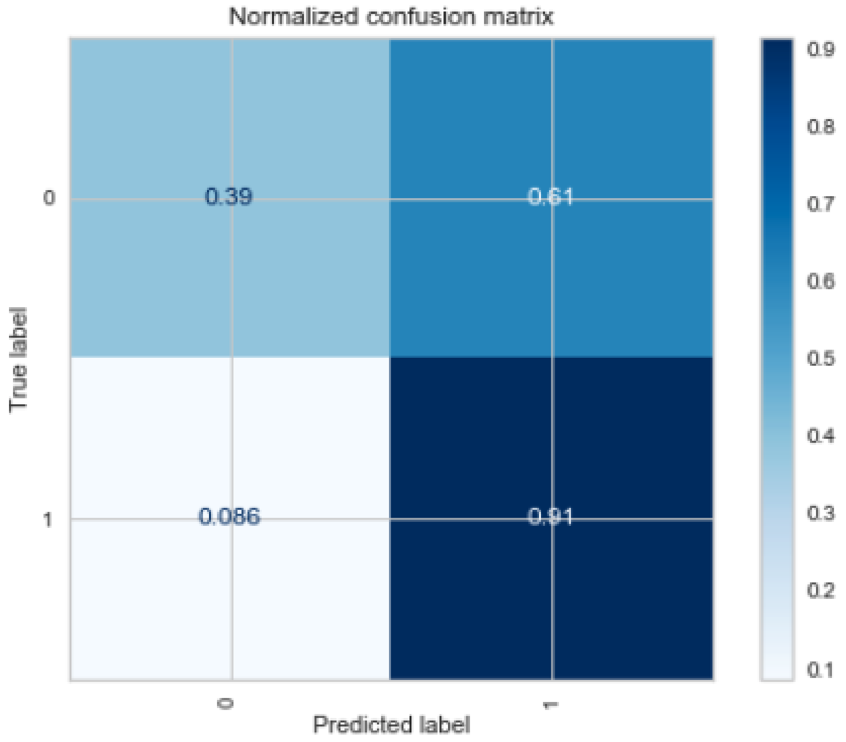
The first model uses a 4-category labeling system, reaching an Accuracy of **57%** (balanced accuracy is **53%**) and similar levels of Recall and Precision, respectively **57%** and **56%**.
The accuracy level of the benchmark model defined in the Proposal document is **44%**.

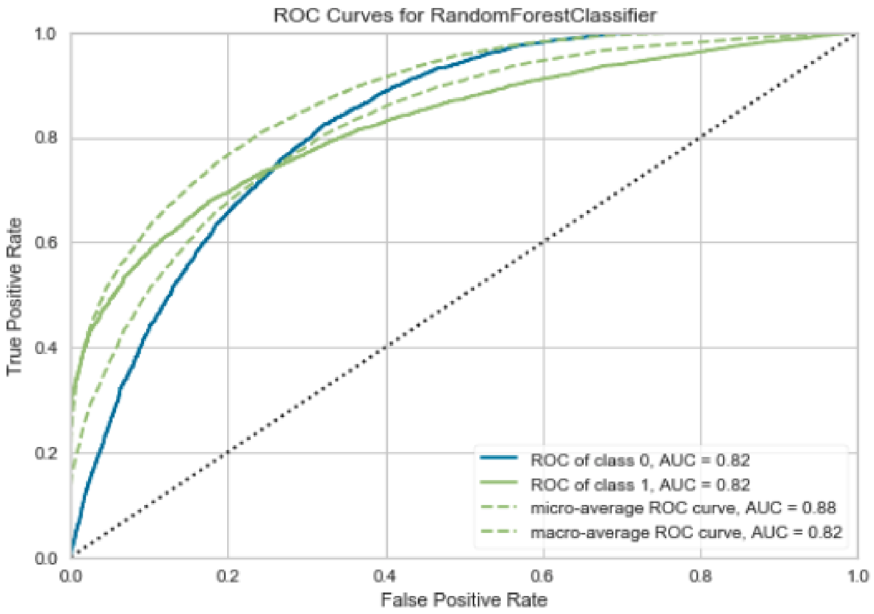




Y2 model

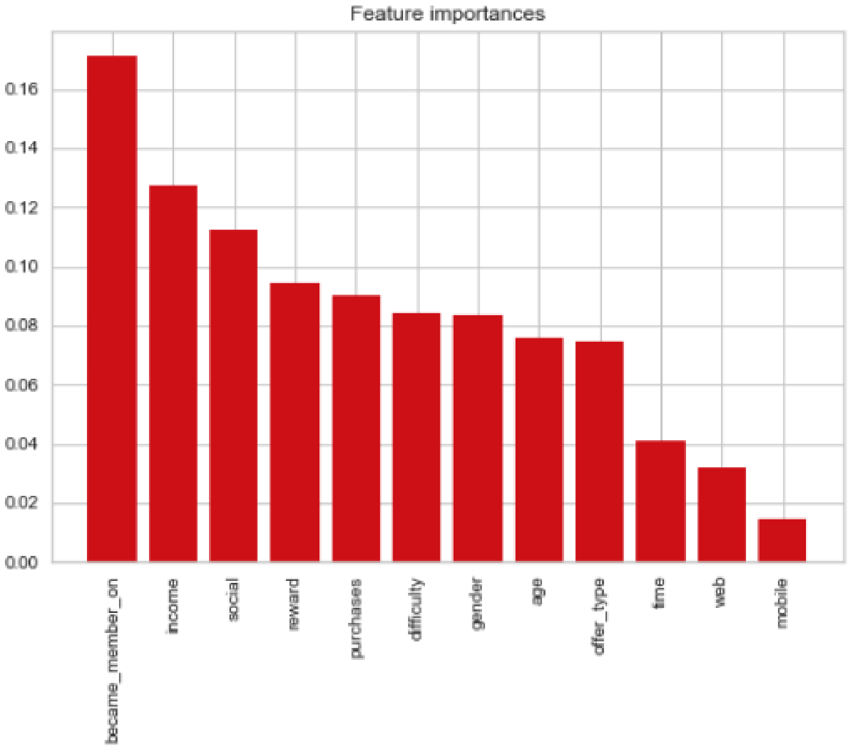
The second model is intrinsically less "complex", having only 2 labels, and is able to reach an Accuracy of **78%** (balanced accuracy is **65%**) and similar levels of Recall and Precision, respectively **78%** and **76%**.
The accuracy level of the benchmark model is **0.71%**.

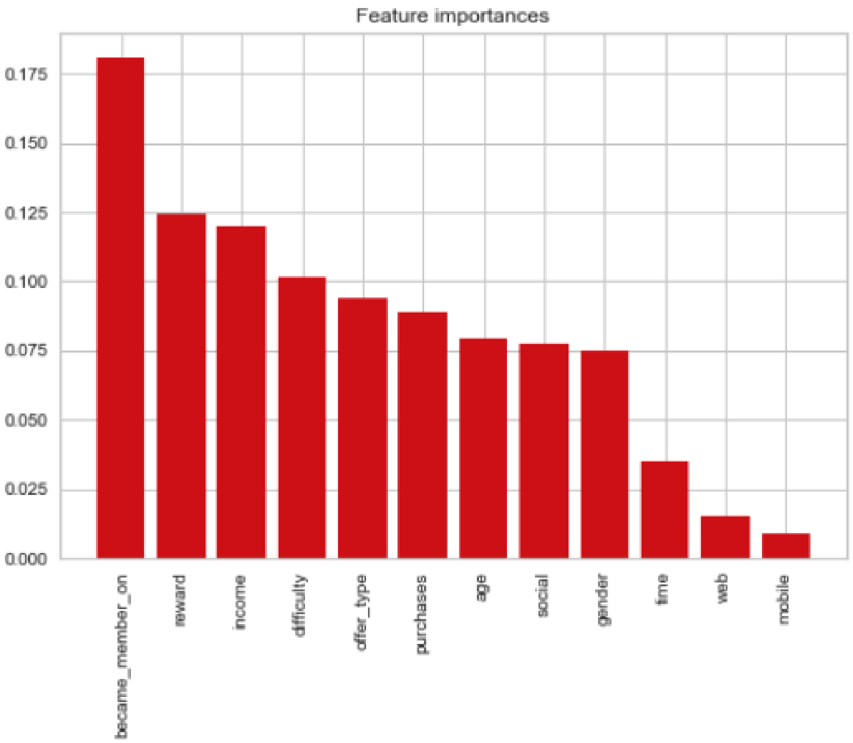




Feature importance

The following plots show the histograms of feature importance, respectively of the first and the second model. The rankings are quite similar.





These rankings show some first important insights:

- *Mobile* and *web* channels have low impact in offer completion rates, while *Social* networks seem to be the best one.
- The *membership* duration is important, but the direction of the impact is non obvious, since older members are less prone to complete offers than newer. This might be a consequence of older members having experimented with more of Starbuck's catalog and thus being less responsive to offers.
- Monetary variables, i.e. *reward*, *income* and *frequency*, are strongly correlated with the probability of completing the offer.

11 Final results

The two models reach quite good performance levels, and can inform the next direct marketing campaign with relatively few predictive features from Starbucks' customers.
The models, even in their relative simplicity, show also some second-level effects that were not so obvious before, such that old users likely saturated Starbucks' catalog and thus showing less proclivity to complete offers.
Based on what has been observed up to now, I can suggest a few recommendations to Starbucks.

- Right now there is no target-customization of neither *reward* nor *difficulty*. I think that differentiating these variables according to some other predictors, such as *age* or *income*, might be an interesting marketing strategy to experiment with.

- Long-time members are less inclined to react and complete offers. Starbucks could test some other approaches, such as trying different ratios *difficulty-reward*, offering them discounts on their favorite products (if the information is available), or even inviting them in stores for a sneak preview of new products.

- Investing more on *Social* network campaigns since the channel has a higher level of engagements (e.g. with more influencer marketing campaigns).