Machine Learning Engineer Nanodegree

Capstone Project - Starbucks Offers

Alejandro Llanos

May 4th, 2021

Definition

Project Overview

Starbucks sends to their users offers through the Starbucks reward app. This offer could be a discount or a BOGO (buy one get one free). These promotions aren't displayed to each user at the same time so the goal is to determine which group responds best to which offer type.

Some rules about the offers:

- An offer has a validity period before it expires.
- A user can complete an offer but never actually received or seen the offer.

The data is displayed by a unique user ID and time (t) in hours, starting from t=0. So the problem has a temporal issue to work on it, i.e. if an offer is completed it's necessary check this criterions in the past:

- The offer completed is valid only if it has been seen by the customer.
- The offer was successfully completed by the minimum sum of transactions.
- The offer is in a valid period when it is completed.

In the data some customer profiles are given like: age, when the customer created an app account, gender and customer's income. Also some characteristics of the offer are provided: offer type, minimum required spend, reward given, duration and channels.

With this information it is possible to analyze at least the success of an offer by groups, mixing customer's profile information and offer information. If this analysis reveals some correlations or tendencies, it is possible to build a binary classification model (ex: logistic regression, decision tree, etc.)

Problem Statement

There are two proposals of problems to solve in this challenge. One of the problems to be solved is to predict if a customer will respond to an offer. Since each offer given to the client means a cost to the company it's important to be efficient sending these offers.

Another problem to solve, for those customers who have completed an offer, is to determine which variables are correlated with increasing the amount of transaction, given an offer difficulty, ie, since a customer spent 40 and the difficulty was 10, determine if there is some variable correlated with these extra spending.

The first challenge is to work the data properly. Match the three tables and construct a rule to determine if an offer was properly completed. Also we can create several kinds of variables according to customer behaviors. In addition, it's important to understand the data, making some descriptive analysis.

For predicting if a customer will respond to an offer, logistic regression is a good approach to solve this problem. With these we can identify which variables are influenced in this decision and make it interpretable. On the other hand, if a customer doesn't respond to an offer, according to the model, it is also useful information for the company to try new ways or types of offers.

In the case of estimating the correlation from the sum amount of transaction (of completed offers) and some other variables, we can use first some BoxPlot to have a visual interpretation.

Finally, the following steps are necessary to achieve the solution.

Data exploration: Understand the datasets, and make some descriptive analysis, like: minimum, average, maximum, distribution, among others.

Data processing: Join the datasets and create a mark for completed offers according to the business rules and data analysis.

Data transformation: Prepare data for the model, adapting columns to model requirements and with a sense of business.

Modelling: Split the data in training and test and run the models with different parameters.

Evaluation: Compare the metrics proposed for the models.

Conclusions: Discussion of findings from the data analysis and the model results.

Metrics

From a business point of view we need to consider if the user has the Starbucks rewards app, so they want to be part of promotions or fidelity programs. So to solve these problems we need to avoid false negatives rather than false positives, because sending an offer to a client that isn't completed is not a problem and even could have a positive future effect in a marketing context.

The way to measure these models is comparing the following metrics:

1. Precision: Positive predictive value.
2. Recall: True positive rate.
3. ROC curve: False positive rates for each threshold.
4. F1 Score: weighted average of the precision and recall.

Analysis

Datasets and Inputs

The dataset for the problem consists of three files: 1) Portfolio, 2) Profile and 3) Transcription. The first one, contains the metadata of offers ids, for example duration, reward, etc. The profile file contains demographic information for each customer and the Transcript file contains the information of all the processes, when a customer buys something, receives an offer, when they see the offer and when this is completed.

Also we can relate all the dataset as follow:

- The transcript dataset can be joined with the profile dataset through the customer id.
- For those records in the transcript dataset that contains the offer id we can join them with the information of the portfolio dataset.

The dictionary and descriptive statistics of these three datasets is presented in the next tables:

Portfolio

Name	Data Type	Description
id	string	Offer id
offer_type	string	Type of offer, ie, BOGO, discount or informational
difficulty	int	Minimum required spend to complete an offer
reward	int	Reward given for completing an offer
duration	int	Time for offer to be open (in days)
channels	list	Channels

Some descriptive statistics of the portfolio dataset:

	reward	channels	difficulty	duration	offer_type	id
count	10.0	10	10.000000	10.000000	10	10
unique	NaN	4	NaN	NaN	3	10
top	NaN	[web, email, mobile, social]	NaN	NaN	bogo	0b1e1539f2cc45b7b9fa7c272da2e1d7
freq	NaN	4	NaN	NaN	4	1
mean	4.2	NaN	7.700000	6.500000	NaN	NaN
std	3.58	NaN	5.831905	2.321398	NaN	NaN
min	0.0	NaN	0.000000	3.000000	NaN	NaN
25%	2.0	NaN	5.000000	5.000000	NaN	NaN
50%	4.0	NaN	8.500000	7.000000	NaN	NaN
75%	5.0	NaN	10.000000	7.000000	NaN	NaN
max	10.0	NaN	20.000000	10.000000	NaN	NaN

Profile

Name	Data Type	Description
age	int	Age of the customer
became_member_on	int	Date when customer created an app account
gender	string	Gender of the customer (some entries contain ‘O’ for other rather than F or M)
id	string	Customer id

income	float	Time for offer to be open (in days)

Some descriptive statistics of the profile dataset:

	gender	age	id	became_member_on	income
count	14825	17000	17000	1.700000e+04	14825.0
unique	3	NaN	17000	NaN	NaN
top	M	NaN	033d0a511a5c452ea2be37a23f8c6dcb	NaN	NaN
freq	8484	NaN	1	NaN	NaN
mean	NaN	62.5	NaN	2.016703e+07	65404.9
std	NaN	26.7	NaN	1.167750e+04	21598.2
min	NaN	18	NaN	2.013073e+07	30000.0
25%	NaN	45	NaN	2.016053e+07	49000.0
50%	NaN	58	NaN	2.017080e+07	64000.0
75%	NaN	73	NaN	2.017123e+07	80000.0
max	NaN	118	NaN	2.018073e+07	120000.0

Transcript

Name	Data Type	Description
event	string	Record description: transaction, offer received, offer viewed, etc.
person	string	Customer id
time	int	Time in hours since the start of the test. The data beings at time t=0
value	Dict of strings	Either an offer id or transaction amount depending on the record

Some descriptive statistics of the transcript dataset:

	person	event	value	time
count	306534	306534	306534	306534.0
unique	17000	4	5121	NaN
top	94de646f7b6041228ca7dec82adb97d2	transaction	{'offer id': '2298d6c36e964ae4a3e7e9706d1fb8c2'}	NaN
freq	51	138953	14983	NaN
mean	NaN	NaN	NaN	366.3
std	NaN	NaN	NaN	200.3
min	NaN	NaN	NaN	0.0
25%	NaN	NaN	NaN	186.0
50%	NaN	NaN	NaN	408.0
75%	NaN	NaN	NaN	528.0
max	NaN	NaN	NaN	714.0

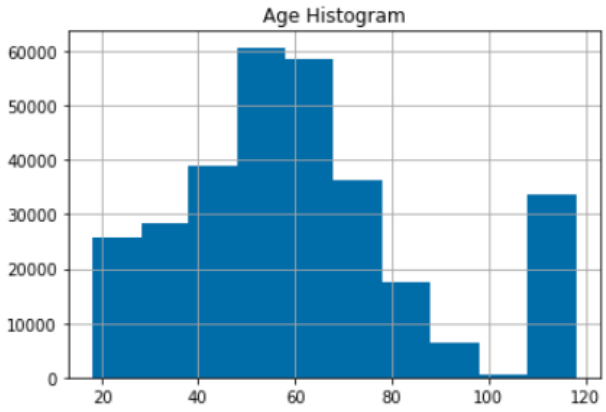
Data Exploration and Exploratory Visualization

This section shows some relevant findings about the datasets. One of the key variables is "event" from the Transcript dataset. This is distribution of the variable:

Value	Count	Percentage
offer completed	33.579	11%
offer received	76.277	25%
offer viewed	57.725	19%
transaction	138.953	45%

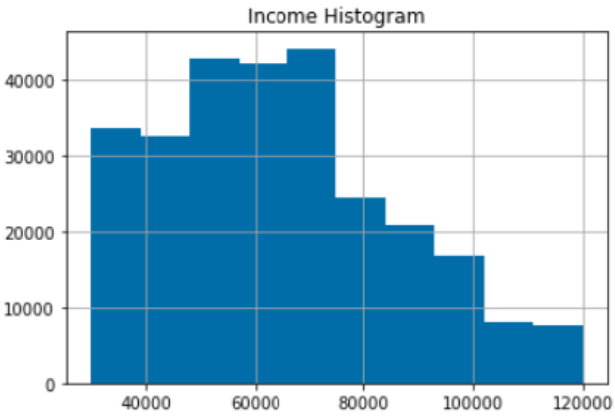
So if we consider all the offers received, we get the 25% of the information available. Then the customers view just the 75% of the and the less of the half complete an offer (44%). This information also considered the "informational offer".

About the age of the customers we can see the follow distribution:



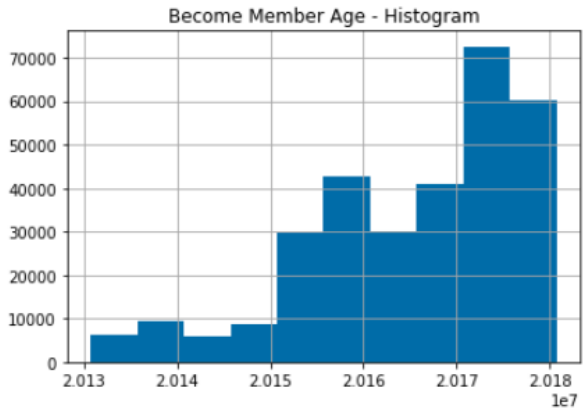
In the age column there is some value above 100, these values can be treated like an anomaly. Later we could give a special value, something appropriate to the model.

Distribution of the income's customer:



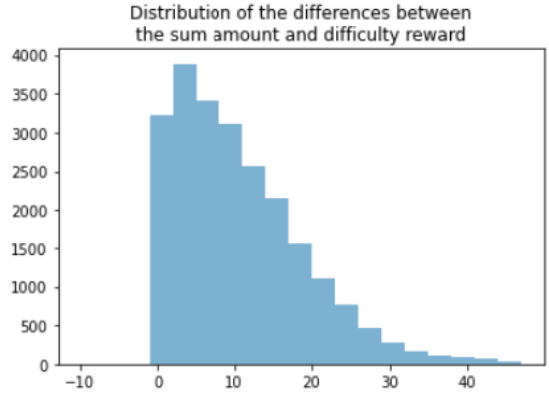
There isn't any anomaly detected on this variable.

Distribution of the become a member variable:



We can see that most of the customers have started joining the app since 2015 and there isn't any anomaly detected on this variable.

Also we can see that when a completed offer occurs (BOGO or Discount), customers spend more than the offer difficulty in most of the cases.



The bar chart of these extra spending shows there is linear decreasing behavior and most of the customers spend between 0 and 10 additional dollars. There is also a significant amount of customers with the range of spending: 10-20 and above of 20. This extra spending is one of the problems to understand in the project.

Offer completed

In order to know if an offer will be completed, we developed an algorithm to get this status according to the following rules:

- Consider only BOGO and Discount offers.
- Achieve the minimum difficulty in the offer completed status.

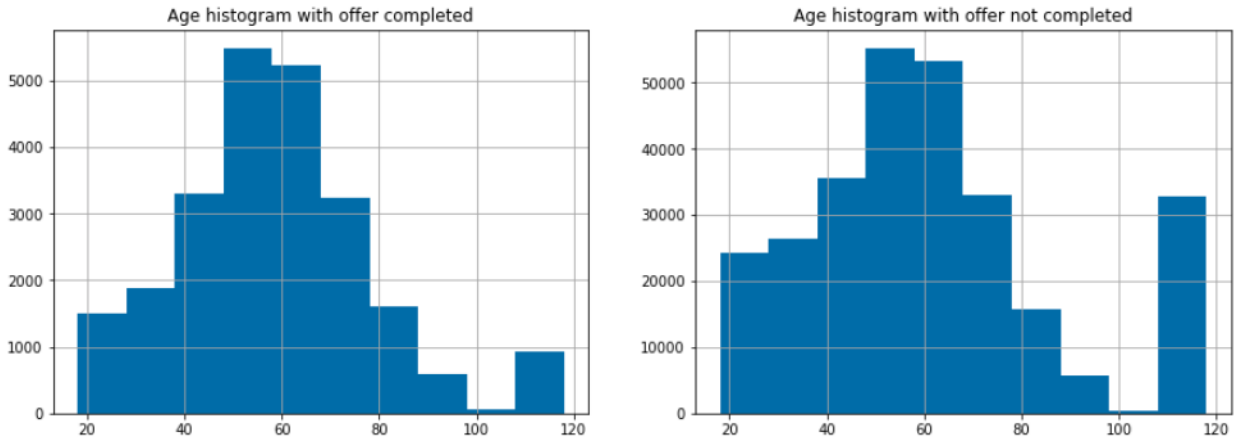
- The offer is still available at the moment when it's completed.
- The offer was viewed.

Basically if an offer were completed, we would bring this status with the sum amount in the **offer received** time. The following image shows an example of this process:

	days	event	channels	amount	reward	duration	difficulty	offer_type	sum_amount	offer_is_completed	amount_when_completed
280	0.00	offer received	[web, email, mobile]	NaN	2.0	7.0	10.0	discount	0.00	0	0.00
281	5.50	transaction	NaN	2.32	NaN	NaN	NaN	NaN	0.00	0	0.00
282	7.00	offer received	[web, email, mobile, social]	NaN	2.0	10.0	10.0	discount	0.00	1	15.04
283	7.00	transaction	NaN	6.86	NaN	NaN	NaN	NaN	0.00	0	0.00
284	8.50	offer viewed	[web, email, mobile, social]	NaN	2.0	10.0	10.0	discount	0.00	0	0.00
285	9.75	transaction	NaN	3.07	NaN	NaN	NaN	NaN	0.00	0	0.00
286	10.25	transaction	NaN	5.11	NaN	NaN	NaN	NaN	0.00	0	0.00
287	10.25	offer completed	NaN	NaN	NaN	NaN	NaN	NaN	15.04	0	0.00
288	11.00	transaction	NaN	2.20	NaN	NaN	NaN	NaN	0.00	0	0.00

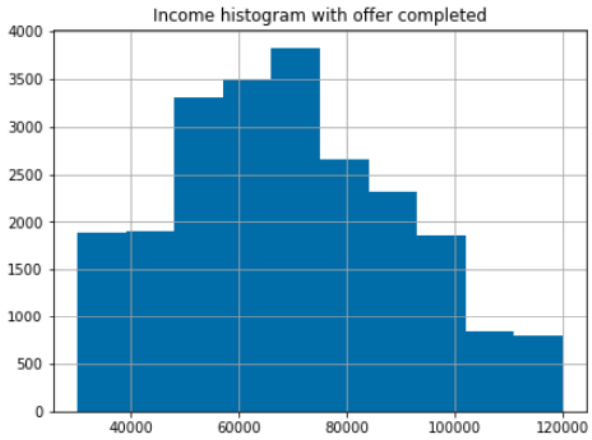
Bringing the information at this level helps us to more easily work distinct operations with the datasets. For example the following analysis can be done.

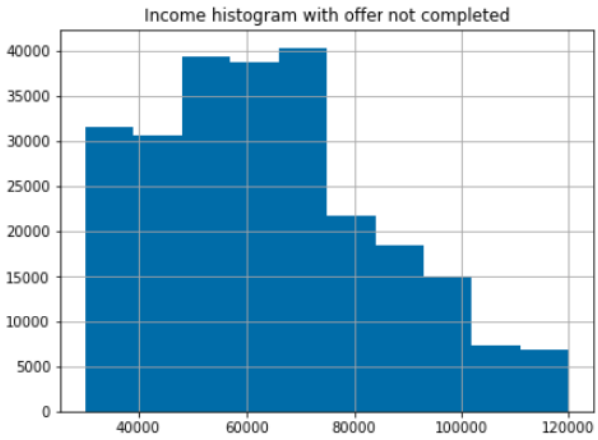
Age's histogram comparing if an offer were completed or not:



Clearly the ages over 100 are an anomaly. Also that group with those ages below 40 have a lower completed offer rate in contrast with the other ranges.

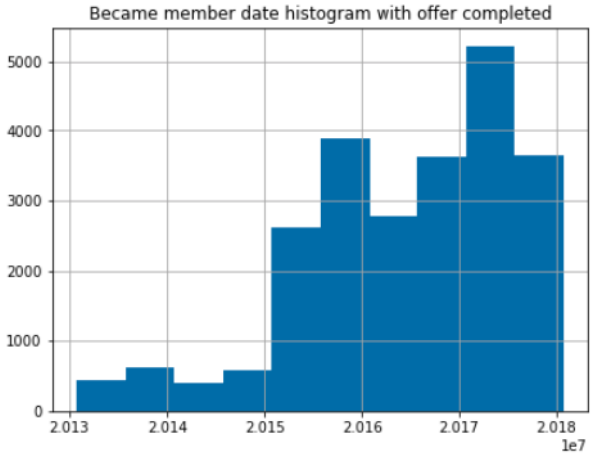
Income's histogram comparing if an offer were completed or not:

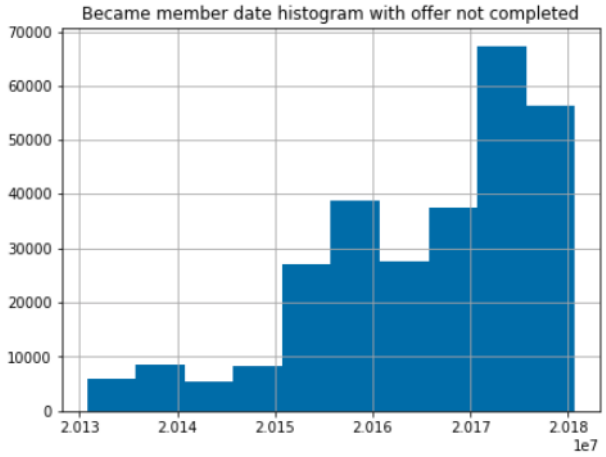




For incomes below the 50.000 the rate is lower.

Became a member's histogram comparing if an offer were completed or not:





Algorithms and Techniques

In this section we are going to talk first about the binning process in each variable and then some theoretical facts about the three models selected for the machine learning technique selected.

Weight of Evidence

The approach that we are going to use to solve this problem is finding an optimal binning for each continuous and categorical variable given our binary target (offer completed). We are going to use the **optbinning** library in Python. In this process each binning is associated with a numerical value called Weight of Evidence (**WoE),** that represents the predictive power of an independent variable in relation to the dependent variable. WoE is defined for each bin as follows:

$$WoE = ln \frac{\left(\frac{Number\ of\ not\ completed\ offers\ in\ the\ bin}{Number\ of\ total\ not\ completed\ offers}\right)}{\left(\frac{Number\ of\ completed\ offers\ in\ the\ bin}{Number\ of\ total\ completed\ offers}\right)}$$

WoE is inversely related to the offer completed event. So the output is a numerical value and the goal is to give this value to the logistic regression model. This step increases the performance of the model because we remove this complexity.

For example this is the result when we make this process for the variable ages:

Final Group for variable: age

	Bin	Count	Count (%)	Event rate	WoE	IV
0	[-inf, 8.50)	9799	0.128466	0.095418	1.45907	0.189198
1	[8.50, 28.50)	6552	0.085897	0.255037	0.281788	0.006437
2	[28.50, 35.50)	4184	0.054853	0.280354	0.152568	0.001239
3	[35.50, 40.50)	4154	0.054459	0.351709	-0.178605	0.001793
4	[40.50, 52.50)	13841	0.181457	0.355682	-0.195986	0.007212
5	[52.50, 72.50)	27895	0.365707	0.359742	-0.213655	0.017323
6	[72.50, 79.50)	4916	0.064449	0.362083	-0.223805	0.003355
7	[79.50, inf)	4936	0.064712	0.370948	-0.261988	0.004644
8	Special	0	0.000000	0.000000	0	0.000000
9	Missing	0	0.000000	0.000000	0	0.000000
Totals		76277	1.000000	0.312139		0.231201

Also an important metric to analyse is Information value (IV) or Jeffrey's divergence, this metric helps us to rank variables on the basis of their importance. It's calculated by the following expression:

$$IV = \sum_{i\ in\ Bins} (\%\ of\ not\ completed_i - \%\ of\ completed_i) * WoE_i$$

Working with this technique, it is also important to consider that:

- Variable with IV less than 0.02 is not a good predictor.

- In the case of numerical variables WoE's needs to be monotonic.

Machine Learning Techniques

In this section we are going to give a small review of some theoretical facts and properties of the models selected for the benchmark.

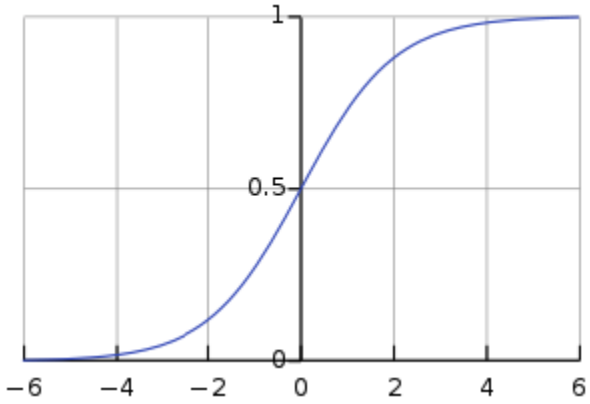
Logistic Regression

The logistic regression is a good model to solve problems when the response variable is qualitative with two outcomes for example: good or bad, some financial status like if a customer is going to have a default status or in health like high blood pressure or not, among others. Usually the outcome is represented by a binary variable with the values 0 or 1.

In the logistic regression the response function has a sigmoidal shape, with asymptotes at 0 and 1. In the case of one variable the function has the following expression:

$$E[Y] = \frac{exp(\beta_0 + \beta_1 X)}{1 + exp(\beta_0 + \beta_1 X)}$$

Where Y is a Bernoulli random variable with the state 0 or 1 and X represent observations (the predictor variable). The following figure give us an example of the response variable:



For higher dimensions the expression can be easily extended, replacing $\beta_0 + \beta_1 X$ by $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$.

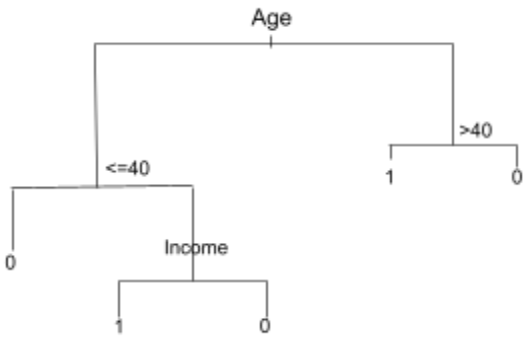
More detail of how to find the maximum likelihood estimator can be reviewed in [3].

Some properties or observations of this methods:

- Convergence can present difficulties when the predictor variables are highly correlated and also when we have a large number of predictors.
- The difficulty in this technique for making predictions is in determining a cutoff point (ex: 0 if Y<=0.4, 1 otherwise).
- When the error rate in the test dataset is similar to the train dataset we have a reliable model.

Decision Tree

Decision tree is a method commonly used in operations research and can be applied for regression and classification problems. In the case of classification problems, the method uses binary splitting to grow the tree and the error is the fraction of the training observation that don't belong to their class. The following diagram give us a notion how this technique works:



In this method we always start with a root node (in the case of the previous figure the variable is Age). Then the method continues splitting the dataset in sub-nodes. When we cannot find any other splitting we finish in that node and we call it a leaf or terminal node. We can find in the list of variables another way to split the dataset in the sub-nodes phase and continue repeating the process, but also there is not always a constructive branch and we can also have a pruning phase.

To know if a variable will be a node the method can use a selection measurement like: entropy, Gini index, among others.

In the case of Gini Index, the method tried to minimize the error for each split:

$$Gini = 1 - \sum_{i=1}^{N} (p_i)^2$$

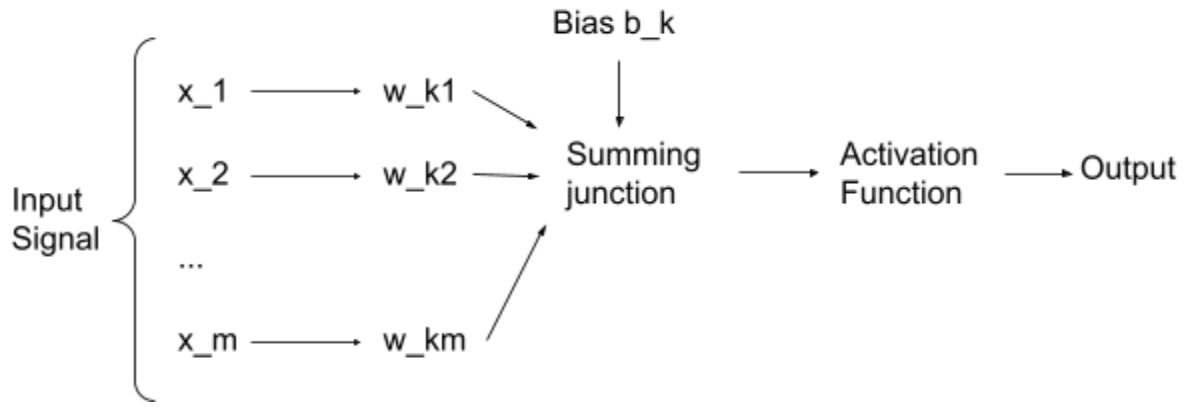
Where N is the amount of subnodes and p_i is the probability of an element being classified for a distinct class.

Neural Networks

This technique was inspired by the human brain and the interactions between neurons. Three elements are important in this kind of models:

1. Synapsis: Characterized by a weight w_{kj}.
2. Adder: Summing the input signals.
3. Activation function: Limiting the amplitude of the output's neuron.

The follow diagram show the is the process for a set of data $x = (x_1,..., x_m)$:



The bias b_k has the effect of increasing or lowering the input. The activation function can be: threshold function, piecewise linear function or sigmoid function. A generalized neural network is when the input can go to several activation functions in a layer format, this is called hidden layer and the output of this layer can go to a new layer making the neural network more complex.

Working with this technique we need to try several quantities of hidden layers with distinct activation functions in there. So the hyperparameter that we need to consider, at least, are the followings:

- Number of hidden layers.
- Number of activation functions (neuron) in each layer.
- Type of activation function.

This model can easily have overfitting issues, so it is important to have a validation dataset to avoid this. In our case we used the library Sklearn and the function MLPClassifier that already split a fraction of the data for validation, so we need just for now set the previous hyperparameter.

Benchmark

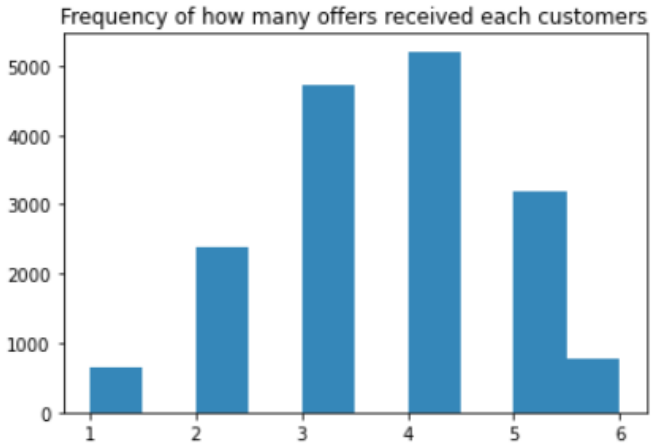
A good way to make a benchmark with the logistic regression is comparing it with other binary classifications techniques like: random forest or neural network, since these two models are more complex and can achieve a good precision but with a lack of interpretability. In the result section we compare those models.

Methodology

Data Preprocessing, Implementation and Refinement

Training and testing data

Since we need to transform our variables in WoE, we need to split the data in training and testing subsets previous to this process. This is because the WoE transformation depends on the total of events selected to train the model.
Another decision that we are going to take is selecting just one offer received for each customer, this method avoids any bias for those customers with more offers in front of those with 1 or 2 offers.



Selecting a unique offer for each customer we get a total number of 16,928 offers. The final result of the splitting process is:

	Total Offers	Completed Offers	% of completed
Total Offer Selected	16,928	6,259	37%
Training set	13,542	4,998	37%
Testing set	3,386	1,261	37%

Transformation

For each variable we are going to apply an Optimal Binning through *optbinning* library. This method consists in getting an optimal discretization for continuous and categorical variables given our binary target. The optimal binning ends up with a discretization represented by the Weight of Evidence (WoE), that shows how the category of the bin is related to the completed offer rate.

The method consist in the following steps for both, continuous and categorical variables:

1. Analyze the proposed discretization given by the optbinning library.
2. Replace some values if it's necessary to keep the monotonic tendency of completed offer rate. For example: In case of variable age, some values are above 100 (probably an anomaly of the data), so these values could be replaced for example by -1, to keep the tendency of completed offer rate in an ascending order. But this process can be done

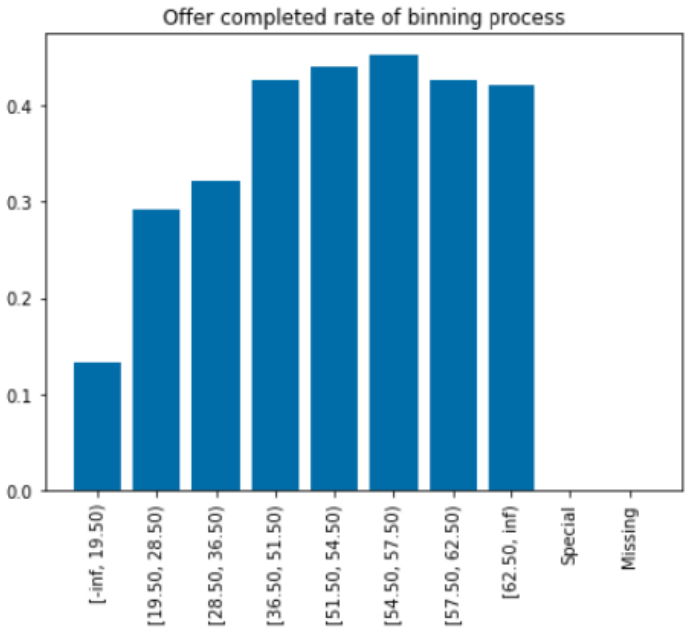
only for missing or default values in the case of continuous variables. In the case of categorical variables any kind of grouping can be done.

3. Get again the WoE variable through the optbinning library.

So with these steps we can have both, WoE data and also maintain the raw data for our logistic regression model. The raw data can be used for other kinds of models like Neural Network.

We are going to describe the previous steps in the case of the age variable, but the same process can be applied to the other variables.

Starting with the **age** variable, we can see that some values are above 100, that's clearly an anomaly of the data. So we need to transform this value to one that makes sense for the WoE transformation. Because the data over 100 has a low rate of offer completed, they need to be located with younger ages to keep monotonic behavior.



So ages over 100 we are going to assign value -1, with this change the binning process is shown in the table:

Final Group for variable: age

	Bin	Count	Count (%)	Event rate	WoE	IV
0	[-inf, 8.50)	9799	0.128466	0.095418	1.45907	0.189198
1	[8.50, 28.50)	6552	0.085897	0.255037	0.281788	0.006437
2	[28.50, 35.50)	4184	0.054853	0.280354	0.152568	0.001239
3	[35.50, 40.50)	4154	0.054459	0.351709	-0.178605	0.001793
4	[40.50, 52.50)	13841	0.181457	0.355682	-0.195986	0.007212
5	[52.50, 72.50)	27895	0.365707	0.359742	-0.213655	0.017323
6	[72.50, 79.50)	4916	0.064449	0.362083	-0.223805	0.003355
7	[79.50, inf)	4936	0.064712	0.370948	-0.261988	0.004644
8	Special	0	0.000000	0.000000	0	0.000000
9	Missing	0	0.000000	0.000000	0	0.000000
Totals		76277	1.000000	0.312139		0.231201

We also have a good IV for the variable and a monotonic decreasing WoE.

Model Implementation

For the selected models we use SKlearn library in python. Some of the advantages of using this library is because it is easy to use, has a numerous authors and contributors and it's well documented.

We need first to save the training and testing dataset in arrays, called: X_train, Y_train, X_test anad Y_test. This process is explained in the notebook.

The following table describe the function that we use and some hyperparameters used:

Functions	Parameters
LogisticRegression	random_state=0, solver='lbfgs', multi_class='ovr', class_weight = 'balanced'
RandomForestClassifier	n_estimators=100, max_depth=4, random_state=0, class_weight = 'balanced'
MLPClassifier	solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(7, 2), random_state=1

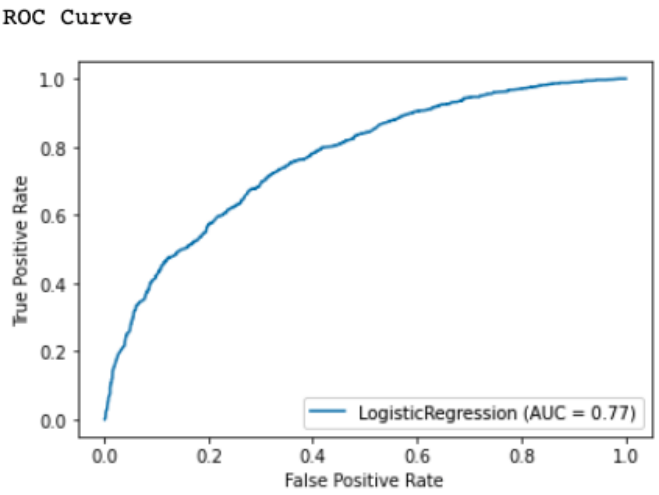
In the case of the LogisticRegression and RandomForestClassifier the balanced parameter play a key role, since if we omit it, then the recall metrics lower their performance. For the Neural Network method, the hidden layer is something that we need to play a little bit, we found after some iterations that increasing the hidden layer the model doesn't perform better and if we reduce it we lose some performance.

Results

Model Evaluation and Validation

For our model we use Logistic Regression with WoE variables and for the benchmark we are using Random Forest and Neural Network. Here are the results on the test dataset, with the four metrics that we explain in the section “Metrics”. Since our data is **imbalance**, we need to set this parameter to our models and also we need care more about **recall** metrics than precision or accuracy, nevertheless we are going to show all typical metrics for this kind of problem:

Model	Recall	Precision	Accuracy	F1 Score
Both Offers - Logistic	0.74	0.56	0.69	0.64
Both Offers - R. Forest	0.73	0.56	0.69	0.63
Both Offers - Neural Network	0.52	0.67	0.73	0.58



As we can see, The logistic model on the test dataset gives us a good performance with recall metric. That means that we can have a better prediction in a successful offer. The precision doesn’t have a good performance since we prefer to optimize the successful offers.

Notice that if we were selected an unbalanced setting this would be the result of the logistic regression:

```
Metrics for Logistic Regression.

prediction (col)     0    1
actual (row)
0                 1783  342
1                  616  645

Recall:     0.511
Precision:  0.653
Accuracy:   0.717
F1 Score:   0.574
```

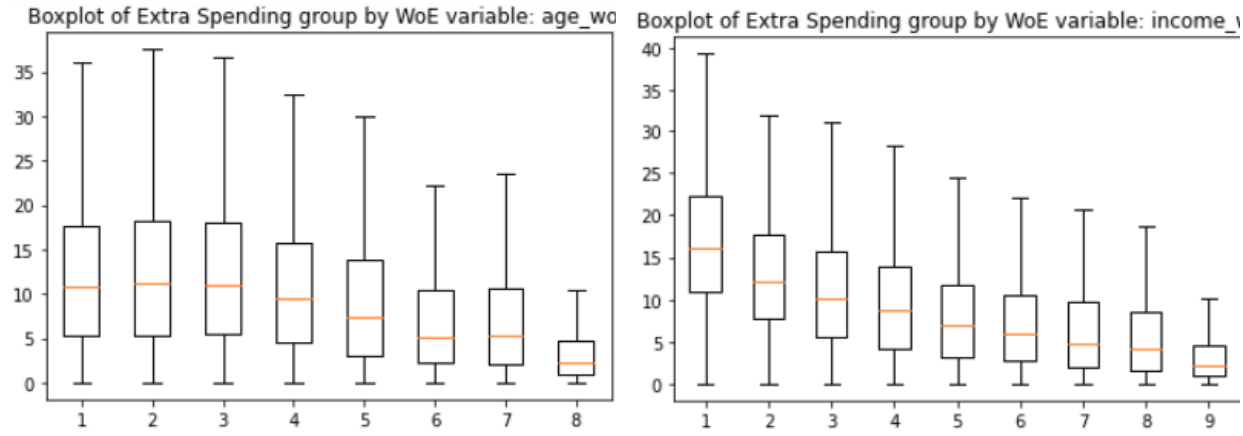
We lost performance in the recall metric (-15%) but we increased our precision by 10%.

Also we tried modeling each offer separately by BOGO and Discount, but the performance doesn't improve significantly. Here are the performance for each model:

Model	Recall	Precision	Accuracy	F1 Score
BOGO Offer - Logistic	0.73	0.51	0.65	0.60
BOGO Offer - R. Forest	0.68	0.51	0.66	0.59
BOGO Offer - Neural Network	0.49	0.60	0.70	0.54
Discount Offer - Logistic	0.74	0.62	0.72	0.67
Discount Offer - R. Forest	0.77	0.60	0.71	0.66
Discount Offer - Neural Network	0.56	0.74	0.75	0.64

After analyzing this result we prefer a logistic regression model like our final model, because we can get a clear interpretation of the input dataset and also this information can be useful from a marketing point of view. Also this binning method helps with interpretability.

Finally, about the extra spending, two variables are clearly related with this behavior, age and income:



Ages under 53 and those with anomalies (over 100) are more likely to have an extra spending. Same result occurs for those with incomes below 63,500.

Justification

We could achieve a model that can identify if a customer would respond to an offer. Also the benchmark with other models like neural network and random forest tell us that the result of the logistic regression has a good prediction and not too far from those models, even if we separate the problem by offers (BOGO and Discount). Also if we are more interested in precision the model can be easily changed in the training process removing the balanced parameter.

Conclusion

Our proposed model and analysis shows us that we can have a robust marketing strategy, having the information of the customers at the moment of sending an offer. We can predict with good metrics those offers that will be completed. Also our data processing strategy avoid having any bias for those customers with more offer completed, for example an users achieving a completed offers status several times means that user weigh more than others, this could end up avoiding important user's inherent information.

At this point we can have a good model with the logistic regression technique and also with a strong interpretability. The use of the binning process before the modeling helps us to reduce the complexity of the model.

About the results of the benchmark model aren't far from the logistic regression. Choosing this kind of model gives us a good interpretability, making it easy for other departments to understand the model behavior.

Finally we can say that those customers with ages below 53 and incomes below 63,500 are more likely to make an extra spending from the offer difficulty. So this is a clear advantage to marketing strategy.

References

1. G. James, D. Witten, T. Hastie and R. Tibshirani. "*An Introduction to Statistical Learning with Applications in R"*. Springer, 2013.
2. J. Foreman. "*Data Smart: Using Data Science to Transform Information into Insight*". Wiley, 2013.
3. J. Neter, M. Kutner, C. Nachtsheim, W. Wasserman. "*Applied Linear Statistical Models*". Fourth Edition, McGraw-Hill, 1996
4. F. Harrel. "*Regression modeling strategies: with applications to linear models, logistic regression, and survival analysis*". Springer, 2001.
5. Guillermo Navas-Palencia. "Optimal binning: mathematical programming formulation"- https://arxiv.org/pdf/2001.08025.pdf
6. R. Anderson. "The Credit Scoring Toolkit: Theory and Practice for Retail Credit Risk Management and Decision Automation". Oxford University Press 2007.
7. S. Haykin. "Neural Network: A comprehensive foundation". Second Edition, Prentice Hall International 1999.