Nubank data challenge - Credit Risk Analysis

by Adriano Freitas

Approval flow

We decided to use the approval flow described below for the credit analysis.

1. **Fraud prediction:** to determine the probability of an application being a fraud
2. **Default prediction:** to determine the probability of an applicant becoming defaulted
3. **Spending prediction:** to determine how much an applicant should spend

This approval process will decide:

1. If the application will be **accepted or not**
2. The credit **line**

Fraud prediction

In fraud prediction we are concerned on predicting the probability of an application being a fraud.

It is very common for fraud to be committed by family members or friends of a person, for this we have even a special type of fraud, but we have decided to only predict whether the application will be fraud or not, regardless of its type.

According to this site (https://www.ecommercebrasil.com.br/noticias/quase-2-das-compras-na-black-friday-foram-feitas-com-cartoes-clonados/), almost 2% of attempts to buy on Black Friday 2017 were fraud.

We decide to mark as fraud if our model predict a probability grater than 1,5%.

Default prediction

In default prediction we are concerned to predict the probability of an applicant becoming defaulted.

Accordind to this site (https://www.valor.com.br/financas/5623039/juro-do-cartao-de-credito-cai-243-em-maio-para-cliente-regular), the default rate for credit cards turns around 33% and 34%.

We decide that 30% is an acceptable rate.

Spending prediction

Based on applicant characteristics and on our database, we predict a probable amount of expenses for this applicant.

Final rule

All Applicants that passes in the first 2 filters (fraud and default) are pre-approved.

So we need to calculate a credit line (limit) for them, and we used a simple rule to determine a value:

$$credit\ line = expenses\ predicted \times (1 - probability\ of\ default)$$

It would be great if we could accept every request that comes to us, but we need to limit it. We can issue up to 1500 cards per month, so only the 1500 best orders are accepted. But how we will decide the best applicants? Well, we just will sortby credit line descending.

Mybe this is not a fair way to decide, but we can inprove this in future.

Considerations

Although a very good model is very important, due to time constraints, I decided to sacrifice a bit of accuracy metrics in order to develop a more robust and complete solution.

The current solution, as it is structured, can be easily adapted to put into production, incorporating into a micro service for example. Your code is clean, simple and easy to maintain.

All the rules cited in this document are flexible and can be changed once the scenario changes, or even after we measure performance in the real world, which will be done constantly.

Next steps

Publish, measure, improve, repeat.

Acknowledgment

I would like to thank the opportunity to participate in this challenge, it was a period of great dedication to achieve these results and I gave my best. I hope you enjoy the results as much as I enjoyed doing this work.