# NOVO BANCO - Data Science Challenge

Vasco Fernandes

May 2020

## 1 Introduction

In online advertising, click-through rate (CTR) is a very important metric for evaluating ad performance. As a result, click prediction systems are essential and widely used for sponsored search and real-time bidding.

For this challenge, contesters were provided with data representing 11 days of interactions between the different ads and its users.

## 2 Resources

My approach to this challenge was implemented using a 13" Macbook Pro. The laptop has a dual-core CPU and 8GB of RAM memory. No GPU was available.

## 3 Structure & Approach

### 3.1 Initial Exploration

In the notebook `0.1-vfernandes-sampling.ipynb`, can be found a simple selection of a smaller sample of data, as I unfortunately have limited memory on my laptop and makes sense to have a small, representative sample of the data to further explore it and derive new features. One aspect to take into account when reducing the size of the data is to keep its representativeness. To make sure of this, I kept the class ratio as it is in the original dataset.

### 3.2 Data Exploration and Feature Engineering

In the notebook `0.2-vfernandes-ingestion.ipynb` can be found an in-depth exploration of the data. The original data has 40 million rows, for this part, I only use 500 thousand rows. This way, my life was much easier to look at the data and create new features in a timely manner. Except for the first part, I used Apache Spark in its Python flavour, PySpark. This challenge is all about the features one can extract from the data. The problem even becomes more interesting as we have a lot of features anonymized - from `C14` to `C21`. Another interesting fact is that we do not, originally, have users identifiers. In my mind,

from the beginning, the idea of modelling each customer behaviour would be a very interesting idea. I derived this from the multiple non-anonymized features related with the device from which the interaction was consumed. Another interesting part of the challenge is the temporal side of everything. From the original timestamp, `hour`, one can derive the calendar day, the day of the week and hour of the day. Another interesting idea would be to understand the temporal behaviour of each user. How much they click or did not click for each hour, calender day, and day of the week. Besides this, it would be interesting to model how the behaviour changed over time. However, given the lack of consistency over time for each user, I would incurr in data leakage, because of the fact that clicks from the same user are (as expected in some sense), very close to each other in time.

### 3.3 Numerical Model

As this is clearly a big data problem, a scalable algorithm was in order to solve this problem. This can be framed as a classification problem. The goal, at the end of the day, is to predict whether an user will click over an ad or not. Field-aware factorization machines are the state-of-the-art for for click-through-rate predictions problems, although a model like logistic regression are also used. The logistic regression model allows some degree of interpretability, as each feature as an associated weight. The "strength" of each weight can be interpreted as the importance of a specific feature. However, more modern interpretability, like SHAP values, can provide a better feature. However, a model like the logistic regression model has specific precondiitons on the data, and does not model interactions between features (although this can be overcome using combinations of features, like in generalized linear models. The sparsity of this particular data could prove to be a problem for this algorithm.

Given the very high cardinality of some of the features, as described in the `0.2-vfernandes-ingestion.ipynb` notebook, before going for a model, one has to try to reduce the sparsity of the features. For this, I used a feature hasher, as this enable me to come up with a much less sparse feature matrix, and, in my case, when memory is low, come up with a much lighter representation of the data.

Fiel-aware factorization machines are quite good for this problem for a very important reason: it is able to model interactions between features. Lets first think about a simple logistic regression:

$$\phi(w, x) = w_0 + \sum_{i=1}^{n} w_i x_i$$

As we can see, each feature has an associated weight. In the case of field-aware factorization machines:

$$\phi(w, x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_{i f_2} \cdot \mathbf{v}_{j, f_1} \rangle \, x_i x_j$$

To the function of the logistic regression, an interaction term is added, in which these interactions are explored.

This has a price in terms of interpretability, as this class of models are not particularly interpretable, but yield very good reasults. However, I was not able to reach the results of the leaderboard. My guess is more work on the features is needed.

In terms of the model chosen and tuned, I settled on a field-aware factorization machine, optimized using adagrad optimizer, as it adapts the learning rate to the parameters, performing larger updates for infrequent and smaller updates for frequent parameters. For this reason, it is well-suited for dealing with sparse data.

In terms of overfitting prevention, I chose to use $L_{2)}$ regularization with a value of 0.002.

To validate the model, cross validation was used. The performance on the test set was 0.443627 log-loss.

# 4 Final Thoughts

Defenitely, more work on the features was needed. More work was also needed to better model the evolving behaviour of each user, in a way that does not imply data leakage.

Also, another point, would be to look for a way to make these class of models more interpretable. Its not only interesting to have a moderately good result, but to know how to explain it.

I also would like to thank for the opportunity to do this challenge. It was very challenging and interesting. An awesome learning opportunity.