Kaggle: Churn Prediction

Just like in the first kaggle, two problems became clear:

- What preprocessing to create a data-set exploitable from transactions and user_logs?
- Which algorithm of prediction?

Prediction algorithm

Thanks to the strong knowledge acquired during the first Kaggle, we reused the code already written and implemented algorithms Boost: **XGBoost** and **LGBoost**. As before, with a grid-seach we have optimized the parameters. As before, it is by averaging the results obtained by the two algorithms that we obtained our best score (0.12484).

Preprocessing:

If the main problem of the first kaggle was based on the missing data, here it was to build the dataset. Indeed, we had the data on the members (members.csv), we then had to use the files user_logs and transactions (daily data per user).
All scores in this part are calculated with an XGBoost algorithm.

- First idea

For the processing of user_logs and transactions we first thought to take the most frequent data. The problem with this approach is that it does not necessarily tell us about the time evolution and the recent behavior of the users (score obtained: 0.18532).

- Second idea

To overcome this problem, we have taken the most recent data for each feature. We were able to improve the score and go to 0.1484.

- Third idea

We now had to find a solution to take into account both the recent behaviour and the user's history.

About the user_logs:

- We calculate the proportion of plays at 25%, 50%... compared to the number of plays
- We added, on the history of the user, the plays for each category (num_25, num50 ...). We then calculated a score for each category with the following function:

Score_cat = sum (num _cat / diff_date) / sum(num_cat)

This allowed us to have an indicator on the recent behaviour of the user and the evolution with respect to past behaviour.

For instance: a score_25 close to 1 indicates that all plays at 25% have taken place recently. This combined information has a 25% high number of listeners compared to the number of plays at 100% can indicate a decrease of the interest of the user for the proposed music.

About the transactions:

For most of the features we chose the most recent mode (mode_de_payment, mode_auto_renew ...).

We added to this:

- a subscription cancellation rate, dividing the number of cancellations by the number of transactions.
- a total subscription time by adding the payment_plan_days.

With these modifications on the dataset, we obtained a score of 0.14570 with an XGBoost and 0.12484 by means of XGBoost and LGBoost.

Potential improvement:
- test several functions to calculate the score in user_logs
- neural network looking to minimize the log-loss function

Finally, these two projects allowed us to work on three aspects:
- create a usable data-set from unorganized data (feature-engineering)
- complete a data-set containing too much missing data
- choose and implement a prediction algorithm

On the technical side, we were able to familiarize ourselves with libraries like Pandas or XGBoost.

NB: We sent you an email 30 minutes after the deadline to tell you that we manage to deeply increase our score. Our best score yet is 0.12212, but the submission was too heavy to upload.