Relax Challenge

# Data Wrangling

In this section I did some exploration of the features available. We have 12000 different users, but in the **user\_engagement** data there are only 8823 unique users, therefore more than 3000 of registeres users never visited the website.

Graphical user interface, text, application

Description automatically generatedThe next step was to create the adopted\_users feature using the users’ engagement.

We have 1445 users that had 3 ore more logins within a 7-day time span.

We used this to create a Target Feature, **adopted,** in our **users’** database.

# Feature Engineering

Graphical user interface, table

Description automatically generated1. I used **last\_session\_creation\_time**  to create **time\_since\_last\_sesion** and **at\_least\_1\_visit** features:

Time since last visit is simply the difference between the amx timestamp and the date of last sesion

At leat 1 visit is a binomial feature, 0 for those who had no visit records, and 1 for the rest.

2. Using c**reation\_time** to create **creation\_year, creation\_month** and **creation\_dayofweek**

3. Lastly, I used **invited\_by\_user\_id** to create **number\_of\_members\_invited** feature with a simple groupby and left merge on users DataFrame.

After the removal of unnecesary feature, this is the data before modeling.

# Chart Description automatically generatedModeling

I used 3 classifiers: XGBoost, RFC and KNN. XGBoost had the highest score with 0.97 accuracy and 0.99 ROC, folowed closely by RFC.

Graphical user interface, text, application

Description automatically generated

Chart

Description automatically generatedThe very high scores are ussualy explained by the fact that there might be some data leackeage in our features.

Using Shap library we can identify the most important features for our XGBoost model. It seems that time\_since\_last\_sesion, creatin\_year and creation\_month are the main features in our prediction. So all we need to make a somewhat accurate predcition are the creation date and sime since last session to predict **adoption.**