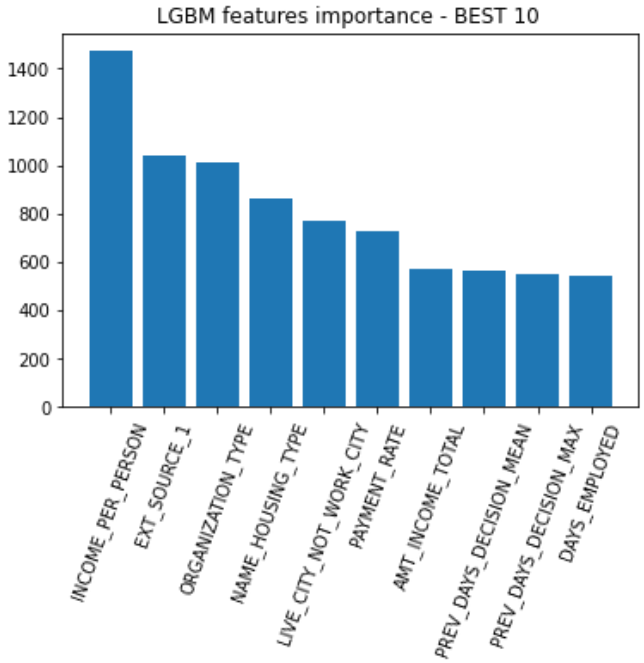
FDS – PROJECT REPORT

Riccardo Taiello 1914000

This report is a summary of the methods and approaches used to carry out the project assignment of the FDS project based on the Kaggle competition: Home Credit Default Risk.

Since many data were available on Kaggle for this project, to make my predictions I decided to concatenate all the available datasets into a single one. Before doing that though, some modifications were applied to each of the smaller dataset to clean and optimize their structure.
To do this, I took some actions like modifying the type of the elements inside the data frame, using integers instead of floats when possible (to reduce memory space), turning strings into categorical types and encoding them with numerical labels etc.
I also decided to do some missing data imputation using the MICE technique, because it is more advanced than the "average", "median" and "most frequent" approaches.
MICE is able to fill in missing data or Nan values estimating them through a linear regression on the other values of that column. As a model for the regression I chose to use the K-Neighbors one, since it allows the regression algorithm to gradually get closer to the real value without too many jumps in the ranges of values estimated.



I then moved to the feature engineering tasks. First of all, I looked for the best (most influential features) in the model, that we can see in the plot on the left.
Some of them had really plain distributions, so I focused on the ones which allowed me to make more interesting techniques.

For example, the AMT_CREDIT feature follows the power log distribution, so I decided to apply the Log-Transformation to turn it into a Gaussian and apply some outliers' deletion techniques. Doing so I was able to identify the values considered too extreme (in both directions of the distribution) and reassign to them the value corresponding to the upper and or lower threshold for detecting outliers.

Finally, I decided to drop the features with high values of correlation (higher than 98%) according to the Pearson correlation.

After all these steps I was ready to run the model. I used k-fold with LightGBM, trying out different options and obtaining the best results setting the number of folds equal to 10.
In the end I was able to obtain an **AUC score of 0.78980.**