Fundamentals of Data Science - Project Report

- Peconi F. (1823570), Ribaldi F. (1763737), Trombetta V. (1455172)

- Details: Language: Python3, Private Score: 0.79125

1 - Preprocessing: homdefault_traintest

The following techniques are equally applied to *application_train/test.csv*: EDA, handling of null/nan values, one-hot categorical features, handling of noisy features (featexp package), handling of anomalies in features, simple features engineering, align and store resulting *train.csv,test.csv*

2 - Merging with the remaining Datasets: homedefault_merge

Handling of anomalies and manual features engineering is carried on every dataset left. After that, all datasets are joined together under the current client's id key. At each merge, statistics of grouped features are computed with special care for categorical ones. Sparse features are removed. The same logic is applied for both train and test. Align and store resulting *train.csv,test.csv*

3 - Feature Selection: homedefault_featen{_pimp}

In *_featen*: remove multicollinearity and newly created noisy features. In *_pimp* we compute feature importance using the research-state method of permutation importance available in the last version of Sklearn, using light Gbm as classifier. Note, the two notebooks need to strictly follow this order when run.

4 - Train and Predict: _lgb, _catb, _xgb

Cross validate LightGbm, CatBoost and XGBoost models over important features as returned from the previous phase and store they're predictions.

5 - Blendings : homedefault_ensemble

A weighted-voting phase is performed between different predictions given by different classifiers to improve the overall score.

6 - Possible improvements

We realized not to have enough computing power and time to try the following, but surely meaningful approaches:

- Create a meta-classifier which learns upon classifier's predictions (stacking) or perform a Bayesian optimization to find the optimal coefficients for the voting phase (we found them manually)
- Try different thresholds when selecting features e.g less eager

Trivia

The *utility.py* script contains procedures used throughout the project. In order to flawlessly run the code, the required packages are listed inside the *Readme*. Additional details and information concerning all the aforementioned sections are present inside notebooks as comments.