Home Credit Default Risk

Alexandru Melnic 1692625 & Dario Russo 1714011
Data Science Class: 29942

The aim of this homework is to be able to find among the unbanked population who can repay a loan and who not. The final output must be a file that contains a header and two columns one with the **SK_ID_CURR** (the ID of each person) and **TARGET** (the probability that that person will repay the loan). Our predictions are evaluated by the **ROC curve**.
Before running the whole code ***REMEMBER*** to go into the header library and set the paths to your file.

The first part of the main concerns the ***EDA***, with the purpose to produce as outcome a cleaned *application_train* set and some information that allows to reproduce the same transformations on the *application_test* set. All the tools needed for the exploration data are present in the classes *train_analysis** and *utils** in the library *df_cleaner**. The functions contained in the class *train_analysis** has been thought to plot some useful graph or to calculate features importance. While the ones in *utils** are useful both for the training and the test set to ***pre-process*** the data. During the analysis, variables have been grouped per "affinity", to study correlation, importance and generate new variables in a clearer environment. All the transformations needed ***to obtain the final sets*** are in the class *generate_test* in* the library *df_cleaner*.* Despite the name, this class can generate both train (not needed) and test. The most important variables which have been created are: *DURATION,* which is the term of the loan and AMT_ANNUITY_INCOMES_RATIO, which is the proportion of the incomes that is yearly used to repay the loan. The other ones are useful to make a summary of many highly correlated or almost empty of meaning columns.

When the train and test are ready, the focus switches on the ***other databases***, which have as foreign key to the train and test set *SK_ID_CURR.* For *Bureau* and *Previous Application,* the adopted strategy consists in grouping the IDs and taking the max, min, mean/median and standard deviation for ***numerical variables*** and the mode and frequency for ***categorical*** ones.
In the *Instalments payments* data frame are stored the clients' loans, and for each loan there is the complete monthly history of the payments. Two columns have been created: *DIFF_INST_DAY* = *DAYS_INSTALMENT -DAYS_ENTRY_PAYMENT*, which represents late payments, and *DIFF_INST_CASH* = AMT_INSTALMENT - AMT_PAYMENT, which represents the difference in cash between the expected payment for that instalment and the actual one. Since for each SK_ID_CURR corresponds many rows, the min, max, median and standard deviation has been saved to be used in train and test. Finally, the *CNT_PREV_LOANS_HC* has been created to represent the amount of previous loans calculated on the *Home Credit* data frame. After this, all new attributes can be added respectively to train and test.

At this point, our data are ready to be processed, the chosen model to train and predict data is the ***lightGBM*** one. The distinctively trait of this model is that data can be fit without being imputed before. Anyway, encoding is still necessary, a double strategy is used: ***label encoding*** for categorical variables with 2 values, and ***one hot encoding*** for all the other categorical variables. Before running the model, the ***multicollinearity*** has been reduced dropping one of the columns of each pair which has a correlation equal or higher than 0.90. Also, features with more than 75 percent of null has been dropped. The last step is scaling all numerical variables through the *MinMaxScaler*. After running the first lightGBM model, all features with zero importance have been detected and deleted. The result is evaluated thanks to the ***private AUC score***, which returns a value of ~0.76.

At the end of our work, the result didn't satisfy our goals, therefore we decided to build a ***new model***, which contains in the train and test most of the original features, plus the one we had created during the EDA and the ones from the other data frames. Also the categorical features haven't been grouped in smaller categories as opposed to the previous model. All the functions needed to generate the new train and test sets are in in the class *No_Out** and *New_Attributes** in the library *Fix_Transform*.* This time the ***private AUC score*** is 0.78173 and this score can be considered a good one.

*Further Information about the functions used in the project are in the libraries.