Predicting Probability of Credit Default

Labinot Polisi

I. Exploratory Data Analysis

The first step in solving this problem was getting familiar with the dataset via *exploratory data analysis* (EDA). The distributions of different features was investigated and if the training set was representative of the test set.

After getting accounted with the different columns of the dataset, the percentage of missing values was investigated. Some features had a significant high percentage of missing values. Closer inspection showed that a great share of these were due to conditional reasons, e.g. the column *account_status* had missing values when the columns *account_worst_status_x_ym* were 0.

The distribution of the target variable was checked. The ratio between the two classes was around 68.9, indicating an *imbalanced class problem*.

Lastly both correlations to the target and collinearity between the features was investigated. Different binning strategies was involved for plotting numerical features with the number of default occurrences, both in relative and in absolute measures.

II. Model

The model building was done in an iterative manner, i.e. starting with a base model and then performing small changes while keeping track if the change resulted in an improvement or not. This part of the case study also involved going back and forth between the EDA and evaluating the model.

The base model was constructed by using an Gradient Boosting Classifier from *LightGBM* and feeding the model all features with minimal processing. The minimal processing required was to convert non-numerical values to numerical since the model does not have support for non-numerical values.

Each evaluation step involved training the model with a *stratified 5-fold cross-validation*, using *binary log-loss* as the loss function and *AUC-score* for validation. *F1-scores* was also calculated but not used in the model selection.

From the EDA it was shown that some columns had a high percentage of missing values. Different strategies was involved into imputing these values, e.g. iterative imputation, statistical imputation and manually setting values. I was found that manual imputation slightly increased the performance.

Using information about the correlations, dropping columns with low predictive power or high collinearity was investigated. In the end it was found that dropping columns had a very slight negative impact on the model performance. The columns were kept, even though one could argue that they could be dropped given the low impact and the cost of overhead in maintaining many features in an production environment.

The categorical features with string values were *one-hot encoded* (OHE). The valuation showed that this increased model performance.

The different binning strategies from the EDA also showed an improvement to the model performance, and was thus included into the model as new features.

Lastly, hyper-parameter tuning was performed using the python package *hyperopt*.

III. Analysis

The model was analyzed by plotting the *ROC-curve* and *normalized confusion matrix*. Depending on the business value, one can decide which part of the confusion matrix to optimize for.

Feature importance was analyzed using *SHAP*, cf. Fig. 1. SHAP gives the opportunity to also analyze how different features interact with each-other.

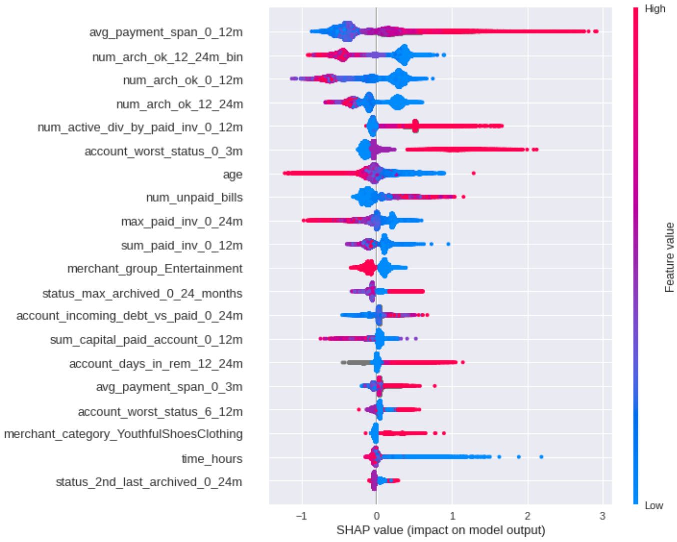

Fig. 1. SHAP feature importance