1. Introduction:

This report briefly summarizes the modeling procedure and results obtained for the probability of default project, which aims at creating a predictive model to estimate the probability of default for customers in each financial portfolio. The goal is to aid decision-making on offering financial products while balancing the identification of defaults and minimizing false positives.

2. Data Study, Preparation and Exploratory Data Analysis (EDA):

This project includes a training and a test dataset, containing 120,000 and 30,000 records, respectively. Both datasets contain the target variable *PD_2years* (probability of default 2 years), and the following ten independent features:

personal_credit_%	Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits, percentage.
age	Age of borrower in years, integer.
A30-60_Counter	Number of times borrower has been 30-59 days past due but no worse in the last 2 years, integer.
A60-90_Counter	Number of times borrower has been 60-89 days past due but no worse in the last 2 years, integer.
A90+_Counter	Number of times borrower has been 90 days or more past due, integer.
expense_ratio	Monthly debt payments, alimony, living costs divided by monthly gross income, percentage.
income	Monthly income, real.
num_loans	Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards), integer.
num_home_loans	Number of mortgage and real estate loans including home equity lines of credit, integer.
num_dependents	Number of dependents in family excluding themselves (spouse, children etc.), integer.

Initial exploratory data analysis revealed the presence of missing values in '*income*' and '*num_dependents*' features, which was addressed by imputing missing '*income*' values with the median and '*num_dependents*' with the mode, respecting their distributions. Additionally, a significant class imbalance in the target variable *PD_2years* was discovered (~93% to 7%). Since this can eventuate in having bias towards the majority negative class, and decreasing the model ability to detect defaults accurately, the Synthetic Minority Over-sampling Technique (SMOTE) is explored in modeling process. This method artificially generates samples for the minority class to provide a more balanced training dataset. Moreover, this step also observed the correlations between features and the statistical properties of data.

3. Model Development and Evaluation

The model development strategy pursued in this project includes the following steps: 1) model-type selection, 2) feature-efficiency analysis for potential feature engineering, 3) model hyperparameters finetuning, 4) optimal classification-threshold calculation, 5) final model training/testing on entire data, with a focus on potential overfitting detection, 6) reliability analysis using the calibration-curve, and 7) robustness tests like sensitivity analysis and noise addition. The remainder of this section explains every step in more detail.

3.1. This step aimed at rapid prototyping of different models on a smaller chunk of data (30% of entire training data), using a multi-model Grid-Search equipped with k(5)-fold cross validation. Initial trials encompassed a variety of models, including *Logistic Regression*, *Random Forest*, *XGB*, *LGBM*, *Gradient Boosting*, and *Cat Boost* classifiers, each with a vast set of hyperparameters, and 6 different SMOTE ratios that simulate a wide range of data imbalance, from the highly-imbalanced structure of original data (no-oversampling) to full-equality of classes.

Since this step mostly targeted to analyze the overall potential of different models in addressing the problem in hand, rather than developing the final model, and considering the existence of data imbalance, the best set of hyperparameters and SMOTE ratio for all models were selected according to the Precision-Recall Area Under Curve (PR AUC) values. Then, all these best models were trained and tested on the entire training set to check how these best models perform on the entire data. Finally, the Cat Boost classifier with a specific hyperparameter set and zero SMOTE was selected as the main model, considering its superior PR AUC on both the smaller and the entire training sets.

3.2. In this step, a deliberate feature-engineering was performed based on the general knowledge of the field, in which a) three highly correlated *A30-60_Counter*, *A60-90_Counter*, and *A90+_Counter* features were summed and replaced by a new *Total_Late_Payments* feature, and b) the *Age_NumLoans_Interaction* feature was generated by multiplying the *age* and *num_loans* features. Additionally, further investigations on defining interaction terms, measuring the feature importance, and model performance after iterative removal of features, were also conducted. However, after training all the best models from the previous step on both the original and the engineered datasets, and carefully investigating the results and calibration curves for all models, the original feature set was selected. In a real-world project, further feature engineering steps based on domain knowledge can be implemented.

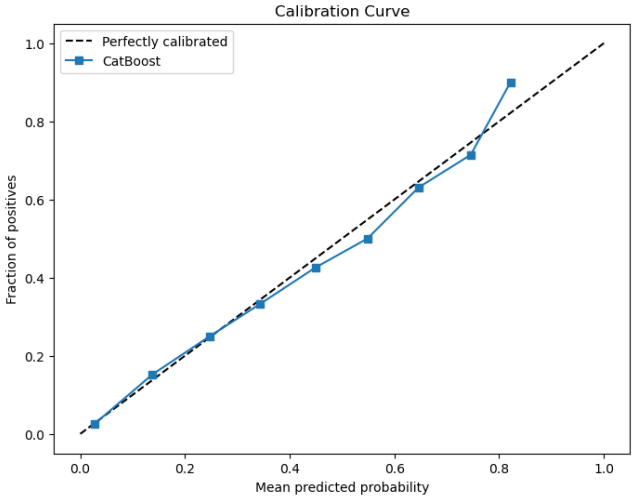
3.3. This step of development further finetuned the *Cat Boost* classifier on slightly bigger chunk of training dataset (~50% of the entire dataset). This step also included the k(5)-fold cross validation implemented using Grid Search tool

on a more detailed set of hyperparameters around the previously selected best hyperparameters, and finally, the model generating the highest PR AUC was selected as the best model.

3.4. Then, the chosen classifier was further examined to discover its optimal threshold value on the entire dataset. Rather than adhering to a conventional 0.5 cutoff for classification, a data-driven approach was employed to identify the threshold (0.21) that generates the highest F1 score (0.44). Additionally, the F1-Threshold curve was plotted, based on which the highest threshold yielding a minimum F1 of 0.4 (assumed to be the lowest acceptable F1) was discovered (0.38). Finally, after examining both thresholds based on Precision, Recall, F1, and the calibration curve, the threshold of 0.21 was selected, considering the higher importance of Recall rate for this problem. However, in a real-world scenario, several factors like the resources to handle the predictions of the model need to be considered.

3.5. The obtained model with its chosen hyperparameters was trained and tested on entire training and test sets. With this, the lack of overfitting was also confirmed, since the model performance did not drop compared to the last step. The Precision, Recall, F1 for the positive default class was 0.4, 0.5, and 0.44, respectively, on the test set. The ROC AUC and the PR AUC scores were also calculated as 0.86 and 0.40, respectively.

3.6. The model reliability was confirmed by plotting the following calibration curve, in which the curve highly correlates with the diagonal line, verifying the model is well-calibrated.



3.7. Finally, the robustness of the model was examined by the sensitivity analysis and noise addition to dataset. The sensitivity analysis which was implemented by adding a fixed amount between +/- 3 to integer features and multiplying the continuous features by a percentage between +/- 20%, demonstrated that the most impactful feature was *personal_credit_%*, that decreased the F1 score from 0.44 to 0.38 during the test. Other features did not provide significant changes. Therefore, one approach of improvement can be decreasing the reliance of the model on the *personal_credit_%* feature by proposing new features or considering this feature as a risk factor.

Furthermore, the Gaussian noise was added to the test set, where the noise level was set as a percentage (1% in this case) of the standard deviation of each feature. This robustness test simulates real-world scenarios where data might not be perfectly clean or might suffer from small inaccuracies. The addition of noise led to a decrease in all performance metrics. Specifically, the F1 score dropped from 0.4436 to 0.3446, ROC AUC from 0.8646 to 0.7732, and PR AUC from 0.4020 to 0.3032. This degradation indicates the model's sensitivity to variations in the input data, which could be addressed through various means, such as incorporating noise resistance during the training phase, exploring more robust modeling techniques, or applying data augmentation methods that include noise as part of the training process.

4. Conclusion

In this project, a predictive model was developed for customer default probability, tackling initial data challenges like missing values and class imbalance with techniques such as SMOTE and feature engineering. The Cat Boost Classifier, chosen for its superior performance metrics, underwent rigorous hyperparameter tuning and threshold optimization, ensuring a balanced precision-recall trade-off. Despite achieving good performance, sensitivity analysis and noise addition tests highlighted the model's vulnerability to input data variations, suggesting a need for enhanced robustness. These findings underscore the project's success in leveraging machine learning for financial risk prediction while also marking a path for future enhancements, particularly in model resilience and data quality assurance.