Scorecards Backtesting

Nexialog Consulting, Paris, France

September 11, 2023

Abstract

In the realm of credit risk management, financial institutions are used to assess borrowers' repayment capacity and evaluate their default probabilities through the construction of a credit scoring. Banks commonly rely on logistic regression models for building this score due to their simplicity and the clear interpretability they offer for the model's explanatory variables. However, recent researches have highlighted that logistic regression may be less performing than alternative machine learning algorithms in predicting credit default probabilities. Although these algorithms offer higher prediction performance, their outputs often lack explicit interpretability. To address this challenge, this paper proposes to use the *Catboost* model. This approach achieves a harmonious balance between prediction accuracy and the interpretability of the model's explanatory variables. However, the classic framework of backtesting used by financial institution is not fully adapted to assess this new scoring framework, particularly in terms of effectively controlling the contributions of variables. Consequently, after validating the accuracy of *Catboost* in evaluating borrowers' creditworthiness while ensuring interpretability, this research proposes and compares two methodologies to adapt the standard backtesting to accommodate this novel approach

Contents

List of Figures

1 Introduction

The banking sector plays a crucial role in providing financial support to individuals, businesses, and governments. However, the risk of borrowers defaulting on their debt obligations poses significant challenges and potential financial losses for banks. To mitigate these risks, the application of Artificial Intelligence (AI) in credit risk assessment has become increasingly important, with a focus on meeting rigorous standards and providing explicit insights.
Traditionally, banks have relied on logistic regression models to construct scorecards for credit risk assessment [3]. This approach has been favored due to its simplicity, interpretability, and ability to quickly evaluate new applications. However, recent advancements in technology and algorithms have encouraged institutions to try machine learning algorithms to build credit scores, which outperform standard logistic regression models in terms of classification performance. In particular, machine learning models excel at capturing non-linear relations between data and default probabilities.

In this paper, we propose the use of the *Catboost* model, a machine learning algorithm specifically designed to work with categorical features in machine learning tasks, to predict credit default probabilities. The primary objective is to bring to light the strengths and performance of the *Catboost* model in credit risk assessment. Additionally, we emphasize the significance of constructing a reliable score grid derived from the *Catboost* model [1]. To evaluate the model's relevance and ensure its stability over time, we introduce an adapted backtesting process. This process involves assessing the performance of the model using a large dataset of historical data. Unlike traditional backtesting methods used for logistic regression models, our approach takes into consideration the contribution of each variable in the final model. To measure variable contributions, we employ advanced techniques such as the *Shap Value* and *Prediction Value Change* methods. In this paper, we compare these methods to demonstrate their effectiveness and stability in the context of credit risk assessment.

The paper is organised as follow. Section 2 is dedicated to present the *Catboost* approach used for build the credit score. Section 3 aims to highlight the construction of the scorecard and discuss its properties. Finally Section 4 describes the process of backtesting. In particular, we showcase the results of performance and how to adapt the standard backtesting process to assess the stability.

2 CatBoost: Categorical Gradient Boosting

2.1 Description

Catboost model is a powerful gradient boosting algorithm designed by Yandex for both classification and regression tasks. In particular, it is well designed for credit scoring and other predictive modeling tasks. It belongs to the family of boosting algorithms.
An advantage of *Catboost* is its ability to handle categorical variables efficiently without the need for extensive data preprocessing. It incorporates a unique algorithm that automatically handles categorical features, eliminating the need for manual encoding or feature engineering. This capacity makes *Catboost* particularly well-suited for credit risk assessment, since categorical variables often play a significant role in evaluating creditworthiness.
Catboost model relies on gradient boosting techniques. Gradient Boosting is an ensemble learning technique used for both regression and classification tasks. It is a form of boosting, which means it combines weak learners, typically decision trees, to create a powerful predictive model. The basic idea behind Gradient Boosting is to iteratively add weak learners to the model in a sequential manner. Let's assume we have a dataset $\{(x_i, y_i)\}$ where x_i represents the feature vectors and y_i is the corresponding target label. In each iteration m, the model aims to find the optimal weak learner $h_m(x)$ that minimizes the residual errors between the true labels y_i and the current predictions $\hat{y}_i^{(m-1)}$. The prediction at iteration m is the sum of the predictions from all weak learners up to that iteration:

$$\hat{y}_i^{(m)} = \hat{y}_i^{(m-1)} + h_m(x_i).$$

The "Gradient" in Gradient Boosting comes from its optimization method, gradient descent. The model computes the negative gradient of the loss function $\mathcal{L}(y_i, \hat{y}_i^{(m-1)})$ with respect to the current predictions $\hat{y}_i^{(m-1)}$. This gradient represents the direction in which the loss function decreases most rapidly. The weak learner $h_m(x)$ is then trained to fit the negative gradient, minimizing the loss function:

$$h_m(x_i) = \arg\min_h \sum_i \left(-\frac{\partial \mathcal{L}(y_i, \hat{y}_i^{(m-1)})}{\partial \hat{y}_i^{(m-1)}} - h(x_i) \right)^2 .$$

The final prediction of the Gradient Boosting model is a weighted sum of the predictions from each individual weak learner:

$$\hat{y}_i = \sum_{m=1}^{M} \alpha_m h_m(x_i),$$

where M is the total number of weak learners, and α_m are the weights determined during the training process. Gradient Boosting models can be prone to overfitting, especially if the individual weak learners are too complex. To prevent this, regularization techniques are employed, such as limiting the depth of the decision trees or introducing learning rate parameters to control the contribution of each weak learner.

2.2 Building steps

In this section, we present the steps for building a *Catboost* model for credit risk assessment :

- **Data Preparation**: Gather a dataset that includes relevant features and the corresponding credit risk labels (default or non-default). Perform data cleaning, handle missing values, and encode categorical variables appropriately. Split the dataset into training and testing sets.

- **Feature Selection**: Analyze the dataset to identify the most informative features for credit risk assessment. Consider factors such as the relevance, correlation, and predictive power of each feature. Select a subset of features to use in the model.

- **Model Training**: Initialize a *Catboost* model and specify the desired hyperparameters such as the learning rate, depth of the trees, and number of iterations. Train the model using the training dataset and evaluate its performance using appropriate evaluation metrics such as accuracy, precision, recall, and area under the ROC curve.

- **Hyperparameter Tuning**: Optimize the model's performance by fine-tuning the hyperparameters. Utilize techniques such as grid search or random search to find the optimal combination of hyperparameters that yields the best performance on the validation set. Consider parameters such as the learning rate, regularization, and tree complexity.

- **Model Evaluation**: Evaluate the final trained model on the testing dataset to assess its generalization performance. Calculate relevant evaluation metrics to measure the model's accuracy and reliability in predicting credit default probabilities.

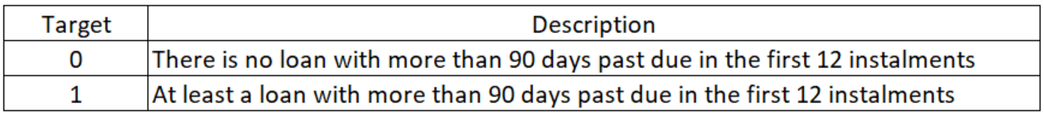
- **Interpretation of Results**: Analyze the feature importance provided by the *Catboost* model to understand which variables contribute most significantly to the credit risk assessment. This analysis can provide insights into the factors influencing default probabilities and aid in decision-making processes.

- **Model Deployment**: Once satisfied with the model's performance, deploy it for real-time credit risk assessment. Ensure that the necessary infrastructure is in place to handle the prediction requests efficiently and securely.

Note that the specific implementation details and considerations may vary depending on the characteristics of the dataset used and the requirements of credit risk assessment task. It is also important to iterate and refine the model-building process based on the specific needs and challenges of credit risk analysis.

3 Hyperparameter Tuning: Scorecard construction

As stated above, *Catboost* approach presents many advantages. Minimizing the prediction computing time and avoiding overfitting are the main ones. The target variable is the variable "Bad" described in Figure 1.



Target	Description
0	There is no loan with more than 90 days past due in the first 12 instalments
1	At least a loan with more than 90 days past due in the first 12 instalments

Figure 1 Target variable

The construction of the scorecard involves several steps, described as follows:

1. Estimation of the different default probabilities using the *Catboost* model.

2. Calibration of the default probabilities estimated in the previous step over 24 months. We present on the left of the Figure 2 (respectively on the right) the observed probability according to the probability estimated by *Catboost* before having calibrated it (respectively the probability estimated by *Catboost* after calibration). On the first graphic, the probability of default (red line) is overestimated. For example, if we look at the predicted probability on the first graphic, we expect that about 55 % of clients fall into default, whereas in reality there are only 5 % by looking the nDoD 24m Rate axis. The calibration consists in correcting the predicted probability (on the left) by applying a logistic regression model. This model inputs the predicted probabilities and outputs the calibrated predicted probabilities. After doing this calibration, we get the red line on the right graphic.

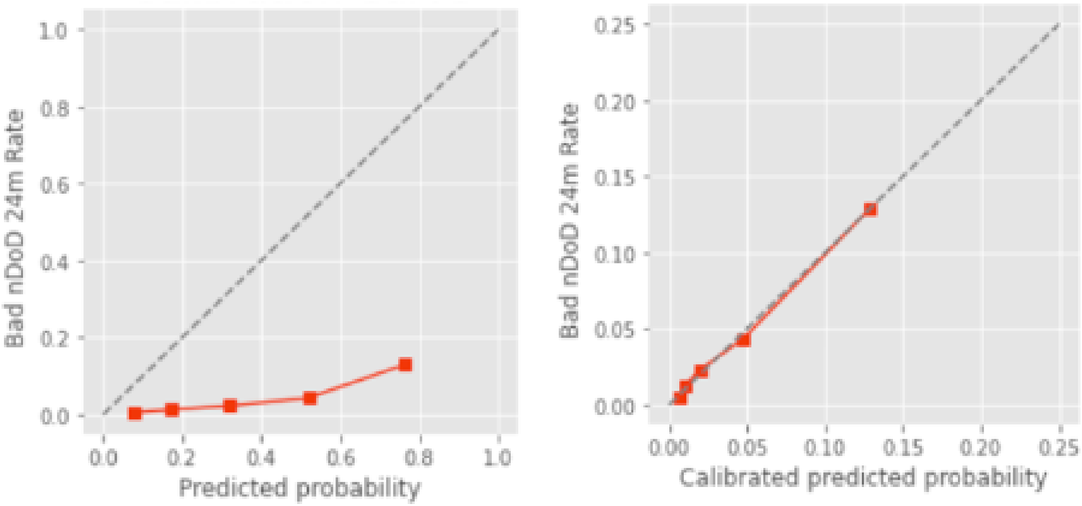


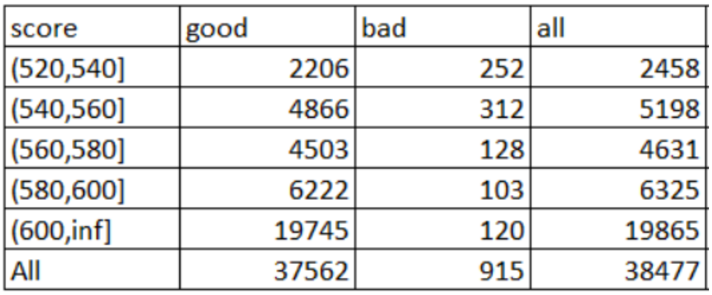
Figure 2 Probability calibration process

3. Based on internal business methodologies, we adjust all the calibrated probabilities to a standardized 12-month horizon values and transform them into scores according to the following relationship:

$$Score = Offset + Factor * log(\frac{p}{1-p}),$$

where: p is the 12-month calibrated probability, the Factor and Offset parameters are calculated using an internal business method.

4. Cut-off strategy: the values resulting from the previous step are segmented into decision risk classes and the following score grid is obtained:



score	good	bad	all
(520,540]	2206	252	2458
(540,560]	4866	312	5198
(560,580]	4503	128	4631
(580,600]	6222	103	6325
(600,inf]	19745	120	19865
All	37562	915	38477

Figure 3 Score classes with frequencies

At the end, we have a scorecard with 5 classes. The first class is the worst one as it contains the lowest number of "good" contracts ('Bad=0'). The best class is the last one, with a score higher than 600. This class contains 19745 "good" contracts and 120 "bad" ones.

4 Model evaluation : Backtesting

4.1 Backtesting framework

A backtesting methodology involves simulating the model's predictions on historical data and comparing them to the actual outcomes. Thus, it assesses the model's predictive accuracy and stability over time.

Backtesting consists on testing the relevance of a model or a strategy based on a large set of real historical data. The aim of this process is to guarantee the quality of forecasts, and if necessary, take the appropriate actions to correct any shift in terms of performance and/or stability. The performance illustrates the discriminating power of the scorecard while the stability specifies the deviation of the reference population (the development model population) compared to that used for backtesting.

Backtesting Process for *Catboost* Model in Credit Risk Assessment relies to several steps.

- Data Preparation: The historical dataset comprising borrower information, characteristics, and observed outcomes (defaults or non-defaults) is prepared. The dataset should be representative of the time period under evaluation.
- Time Split: The historical data is divided into two sets: the training set and the backtesting set. The training set covers an earlier time period and is used to train the *Catboost* model. The backtesting set covers a later time period and is used to evaluate the model's performance.
- Model Training: The *Catboost* model is trained on the training set using appropriate hyperparameters and techniques such as cross-validation for parameter tuning. The model learns to predict the probability of default based on the borrower's characteristics.
- Prediction: The trained *Catboost* model is applied to the backtesting set to generate default probability predictions for each borrower in the set. These predictions serve as the basis for evaluating the model's performance.
- Threshold Determination: A default probability threshold is defined above which a borrower is classified as a defaulter. The choice of threshold depends on the desired risk appetite and specific requirements of the credit risk assessment.
- Performance Evaluation: Various performance metrics are calculated to assess the *Catboost* model's performance on the backtesting set. These metrics include:
  - Accuracy: The proportion of correctly classified defaults and non-defaults.
  - AUC: Area Under the ROC Curve, which measures the model's ability to distinguish between defaulters and non-defaulters.
  - Gini Coefficient: A measure of the model's discriminatory power, calculated as twice the difference between the AUC and 0.5.
  - Brier Score: Measures the model's accuracy and calibration of predicted default probabilities.
  - Calibration Curve: Plots the predicted default probabilities against the observed default rates to assess the model's calibration.
  - Feature Importance: Evaluates the contribution of different borrower characteristics in predicting default probabilities.
- Monitoring and Analysis: The model's performance is monitored over time to ensure stability and consistency. Any deviations or changes in performance are analyzed to identify potential issues or areas for improvement.

Iterative Improvement: Based on the insights gained from the backtesting results, the *Catboost* model can be refined and updated. This may involve adjusting model parameters, incorporating additional data, or exploring alternative modeling techniques.

By conducting a rigorous backtesting process, we can evaluate the predictive power and reliability of the *Catboost* model in predicting default probabilities for credit risk assessment. The performance evaluation metrics provide a comprehensive assessment of the model's effectiveness and its alignment with predefined benchmarks or industry standards. The ongoing monitoring and iterative improvement ensure the model's continuous enhancement and its ability to accurately assess credit risk in practical applications.

4.2 Time split

The development period of the scorecard is between the first of January 2019 and the 31st of December 2019. Three different six-month periods are chosen to analyze the quality of the scorecard:

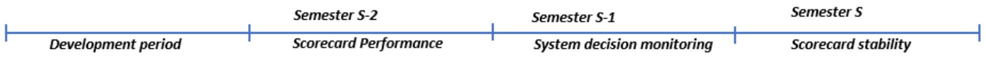


Figure 4 Stages of the backtesting process

- The Semester S-2 (from 01/01/2020 to 30/06/2020) is chosen to study the performance of the model on the backtesting database as the risk horizon is 12 months (we can then observe the default events, which are indispensable for this backtesting step). Performance indicators such as Gini index have to be calculated in this period.

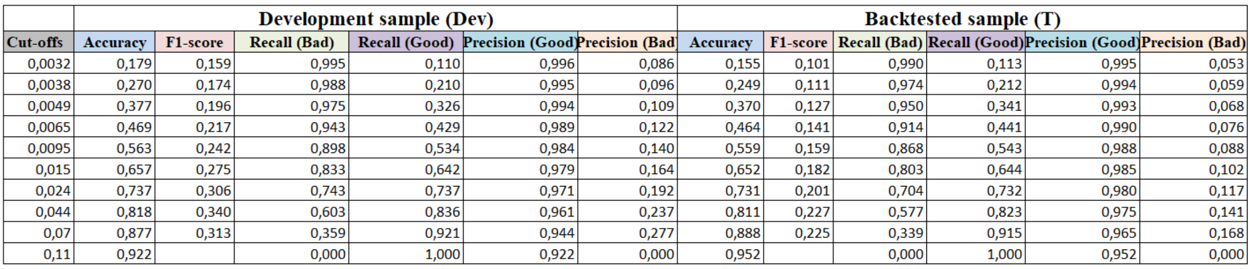
- The semester S-1 (from 01/07/2020 to 31/12/2020) is for studying overrides. It refers to manual decisions based on experts' opinions that reverse those proposed by the system (score or business rules). More precisely, the override is a measure of how well we can detect contradictions between the decisions made by the experts and those decided by the scoring system.

- The semester S (from 01/01/2021 to 30/06/2021) is for studying stability as we need the last period to have enough time to assess the evolution of the different score populations.

Note: To study the performance, we only work with the accepted credit applications (i.e. granted credits) by the system since we want to observe the risk. For stability and overrides, we work with all credit applications.

4.3 Threshold determination

In this section, we use different metrics (Accuracy, Recall, Precision and F1-score, whose formulas are presented in the annex) to compare thresholds used to decide on the attribution of a loan. For example, a threshold of 0.2 implies that if the probability of default is higher than this value, our target variable Bad is assigned the value 1. From Figure 5, we can notice that higher cut-offs implies more accuracy for both samples (i.e. with higher thresholds, the model is better to predict good and bad contracts). The results also show that the recall for good contracts is higher when higher cut-offs are considered, meaning that the model is more effective to predict good contracts when considering higher thresholds.



	Development sample (Dev)						Backtested sample (T)					
Cut-offs	Accuracy	F1-score	Recall (Bad)	Recall (Good)	Precision (Good)	Precision (Bad)	Accuracy	F1-score	Recall (Bad)	Recall (Good)	Precision (Good)	Precision (Bad)
0,0032	0,179	0,159	0,995	0,110	0,996	0,086	0,155	0,101	0,990	0,113	0,995	0,053
0,0038	0,270	0,174	0,988	0,210	0,995	0,096	0,249	0,111	0,974	0,212	0,994	0,059
0,0049	0,377	0,196	0,975	0,326	0,994	0,109	0,370	0,127	0,950	0,341	0,993	0,068
0,0065	0,469	0,217	0,943	0,429	0,989	0,122	0,464	0,141	0,914	0,441	0,990	0,076
0,0095	0,563	0,242	0,898	0,534	0,984	0,140	0,559	0,159	0,868	0,543	0,988	0,088
0,015	0,657	0,275	0,833	0,642	0,979	0,164	0,652	0,182	0,803	0,644	0,985	0,102
0,024	0,737	0,306	0,743	0,737	0,971	0,192	0,731	0,201	0,704	0,732	0,980	0,117
0,044	0,818	0,340	0,603	0,836	0,961	0,237	0,811	0,227	0,577	0,823	0,975	0,141
0,07	0,877	0,313	0,359	0,921	0,944	0,277	0,888	0,225	0,339	0,915	0,965	0,168
0,11	0,922		0,000	1,000	0,922	0,000	0,952		0,000	1,000	0,952	0,000

Figure 5 Performance metrics according to the threshold value change

From graphs 6 and 7, we can notice that the evolution of the different performance metrics as a function of the thresholds is the same for both samples (the development and backtesting samples), even though the values of these metrics are not always exactly equal in the two graphs. We can therefore conclude that we have the same impact of the threshold variation on the discriminatory power of both samples. Consequently, the performance of our model is stable over time as a function of the default threshold.

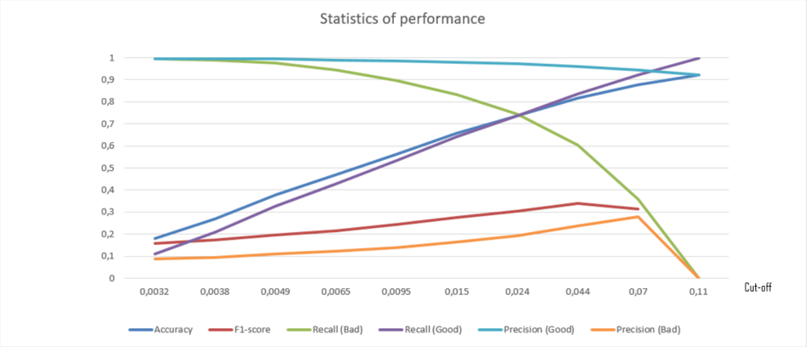


Figure 6 Evolution of performance metrics according to different thresholds (development dataframe)

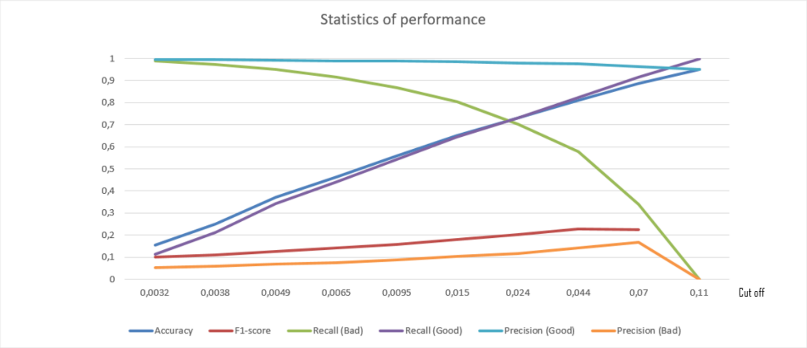


Figure 7 Evolution of performance metrics according to different thresholds (backtesting dataframe)

4.4 Performance evaluation

4.4.1 Discrimination power metric through Gini coefficient

■ Gini Index

The Gini index, also known as the Gini coefficient, is a measure of the inequality or discrimination in a predictive model's performance [4]. In the context of credit risk assessment, the Gini index is commonly used to evaluate the model's ability to distinguish between default and non-default cases.

The Gini index quantifies the degree of separation between the predicted probabilities of default for positive instances (actual defaults) and negative instances (non-defaults). It is calculated based on the Lorenz curve, which plots the cumulative proportion of default cases against the corresponding cumulative proportion of the population.

The Gini index ranges from 0 to 1, where:

- A Gini index of 0 indicates a model that performs no better than random chance. It suggests that the model cannot differentiate between default and non-default cases and has no discrimination power.
- A Gini index of 1 represents a model with perfect discrimination. It suggests that the model can perfectly separate default and non-default cases, providing maximum predictive power.

In practical terms, a higher Gini index indicates better discrimination and predictive performance of the model. A model with a higher Gini index is more effective at ranking and identifying higher-risk individuals or entities who are more likely to default on their credit obligations.

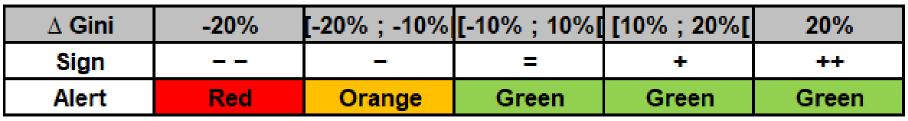
By using the Gini index, analysts and practitioners can compare the performance of different models or variations of the same model and select the one that exhibits the highest discrimination power. It serves as a valuable metric to assess the effectiveness of credit risk models and supports decision-making in risk assessment, loan approvals, and portfolio management in the banking and financial industry.

■ Δ Gini

The difference between the Gini value obtained after backtesting and the one calculated at the time of dashboard construction gives an idea of the model's performance :

$$\Delta Gini = \frac{Gini\_backtested - Gini\_reference}{Gini\_reference}$$

While a Gini higher than 30% is considered satistfactory, ΔGini, can be quantifies following this table :



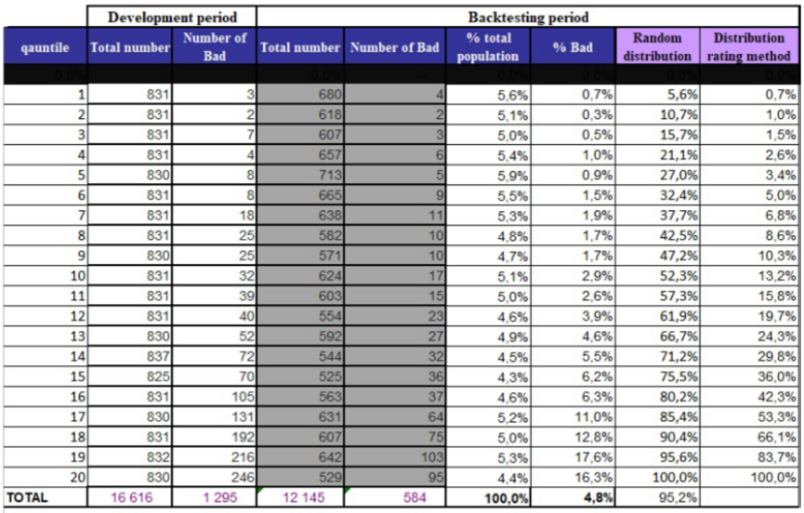
Δ Gini	-20%	[-20% ; -10%	[-10% ; 10%[	[10% ; 20%[	20%
Sign	– –	–	=	+	++
Alert	Red	Orange	Green	Green	Green

Figure 8 $\Delta Gini$ thresholds

ΔGini thresholds can be used to judge model performance:

- Green case: high level of performance.
- Orange case: Acceptable performance.
- Red case: Low level of performance.

To calculate the Gini index, scores are first divided into 20 quantiles on both the development and backtesting basis (cf. Figure 9). The first quantile represents the best scores and the last quantile is the one with the weakest contracts. "*Random distribution*" column refers to the cumulative percentage of second semester population and "*Distribution rating method*" column reveals the cumulative percentage of bad contracts.



	Development period		Backtesting period					
qauntile	Total number	Number of Bad	Total number	Number of Bad	% total population	% Bad	Random distribution	Distribution rating method
0,0%			0,0%	-	0,0%	0,0%	0,0%	0,0%
1	831	3	680	4	5,6%	0,7%	5,6%	0,7%
2	831	2	618	2	5,1%	0,3%	10,7%	1,0%
3	831	7	607	3	5,0%	0,5%	15,7%	1,5%
4	831	4	657	6	5,4%	1,0%	21,1%	2,6%
5	830	8	713	5	5,9%	0,9%	27,0%	3,4%
6	831	8	665	9	5,5%	1,5%	32,4%	5,0%
7	831	18	638	11	5,3%	1,9%	37,7%	6,8%
8	831	25	582	10	4,8%	1,7%	42,5%	8,6%
9	830	25	571	10	4,7%	1,7%	47,2%	10,3%
10	831	32	624	17	5,1%	2,9%	52,3%	13,2%
11	831	39	603	15	5,0%	2,6%	57,3%	15,8%
12	831	40	554	23	4,6%	3,9%	61,9%	19,7%
13	830	52	592	27	4,9%	4,6%	66,7%	24,3%
14	837	72	544	32	4,5%	5,5%	71,2%	29,8%
15	825	70	525	36	4,3%	6,2%	75,5%	36,0%
16	831	105	563	37	4,6%	6,3%	80,2%	42,3%
17	830	131	631	64	5,2%	11,0%	85,4%	53,3%
18	831	192	607	75	5,0%	12,8%	90,4%	66,1%
19	832	216	642	103	5,3%	17,6%	95,6%	83,7%
20	830	246	529	95	4,4%	16,3%	100,0%	100,0%
TOTAL	16 616	1 295	12 145	584	100,0%	4,8%	95,2%	

Figure 9 Table used to compute Gini indicator

Using results from the previous table, we draw the Lorentz curve (illustrated in black on Figure 10). A diagonal red line is drawn from 0% at the bottom left to 100% at the top right of the chart, representing the proportion of bad contracts in the case of a random distribution. Finally, the blue line represents the theoretical optimal case (the perfect selection distribution).

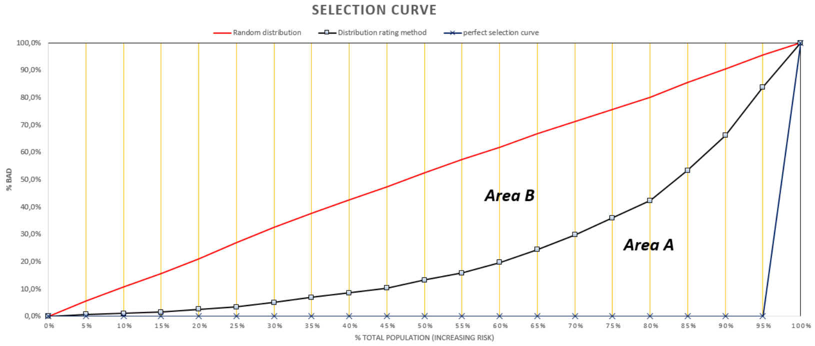
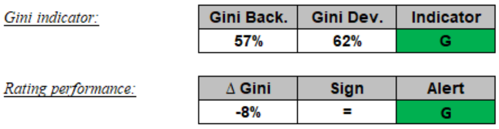


Figure 10 Lorentz curve

$$Gini = \frac{AreaB}{AreaA + AreaB}$$



Gini indicator:

Gini Back.	Gini Dev.	Indicator
57%	62%	G

Rating performance:

∆ Gini	Sign	Alert
-8%	=	G

Figure 11 The Gini and Delta Gini values obtained for both populations

We conclude from these results that our scorecard has a high level of performance as we are in the green zone of possible values for the Delta Gini indicator.

4.4.2 Kolmogorov-Smirnov (KS) Statistic

In the backtesting of the *Catboost* model for credit risk assessment, the Kolmogorov-Smirnov (KS) statistic serves as a metric for evaluating its performance. The KS statistic quantifies the model's ability to differentiate between good and bad customers based on their predicted default probabilities. The KS statistic is calculated by determining the maximum distance between the cumulative distribution functions (CDFs) of bad contracts and good contracts:

$$KS = sup|F_{good}(x) - F_{bad}(x)|$$

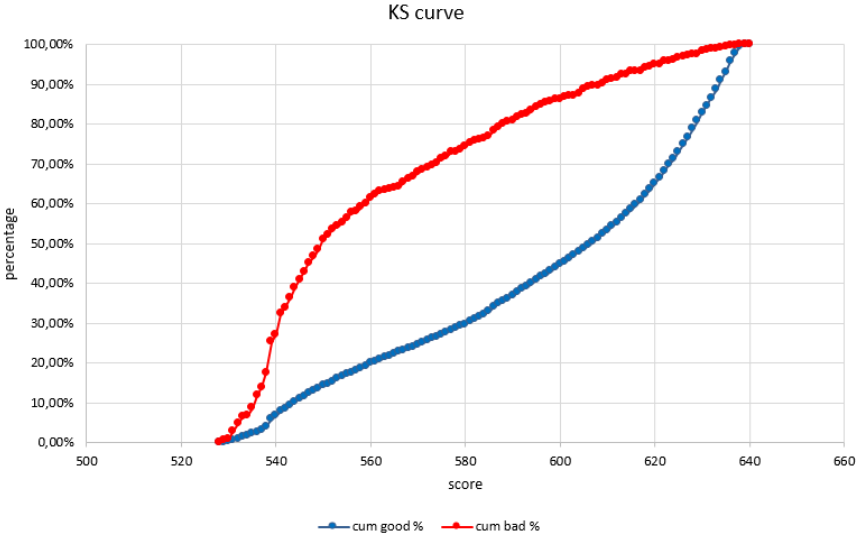
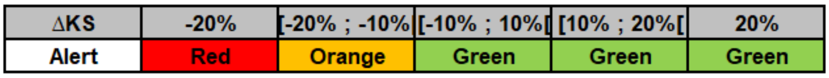


Figure 12 KS curve

Following the figure 12, it measures the maximum vertical difference between the red line (representing the cumulative bad contracts) and the blue line (representing the cumulative good contracts) on the development sample.

In our case, the KS value obtained on the development sample is 49%. This value indicates a strong discriminatory power of the scorecard in distinguishing between bad and good customers. Furthermore, when the KS is calculated on the backtesting sample, it yields a value of 45%. By comparing these two values, we can calculate the ΔKS in the same way as the ΔGini, which measures the variation of the KS indicator between the two databases. In this scenario, the ΔKS value is 8%.



ΔKS	-20%	[-20% ; -10%	[-10% ; 10%[	[10% ; 20%[	20%
Alert	Red	Orange	Green	Green	Green

Figure 13 ΔKS thresholds

Based on these results and the Figure 13, we can conclude that the *Catboost* model performs at a high level from the KS perspective. The significant KS values on both the development and backtesting samples demonstrate the model's efficacy in accurately distinguishing between defaulters and non-defaulters, thereby indicating its strong performance in credit risk prediction.

4.5 Monitoring and Analysis

4.5.1 Scorecard stability

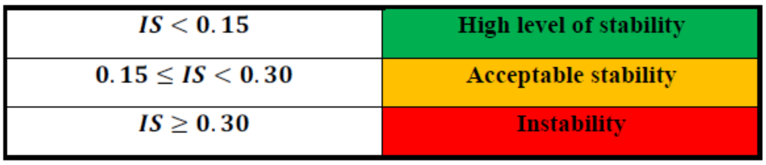
■ **Stability indicator by score buckets** IS_{score}

This indicator compares the score distribution of the backtesting population to that of the reference sample. Its purpose is to identify any shifts in the applicant profile across different score buckets. It writes

$$\text{IS}_{\text{score}} = \sum_{i=1}^{n} (p_i - b_i) \log(\frac{p_i}{b_i})$$

where:
- b_i : part of the backtesting population which belongs to score bucket i.
- p_i : part of the reference population which belongs to score bucket i.

This formula measures the entropic distance between the development population and the new backtesting one. The table below shows the alert thresholds for the IS_{score}:



$IS < 0.15$	High level of stability
$0.15 \leq IS < 0.30$	Acceptable stability
$IS \geq 0.30$	Instability

Figure 14 IS_{score} thresholds

Note: It is important to note that the IS_{score} calculation method is based on groups of deciles. Therefore, we do not use here the score grid developed in section 1.3 which contains 5 ratings. Instead, we will again classify the clients so that each score bucket contains approximately 10% of the population. This decision is motivated by the objective to improve the accuracy of the calculations and to better detect any upward or downward changes in the scorecard.

We estimate a IS_{score} equal to $1,9\%$. We have a high level of stability as our IS_{score} is smaller than 0,15.

■ **Stability Indicator by global variable** IS_{vg}

The IS_{vg} corresponds to the average of the IS by variable (IS_{v_j}), weighted by the contribution of each of these variables:

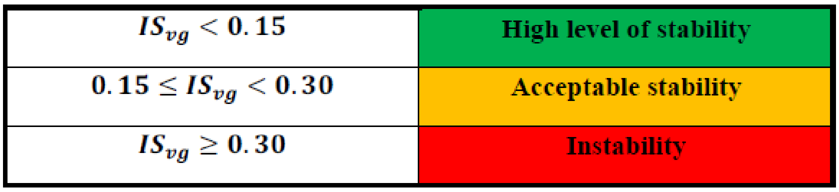
$$\text{IS}_{\text{vg}} = \sum_{j=1}^{n} (q_j * IS_{v_j})$$

where:

- q_j: contribution of the variable j (see next sub-section).

- $IS_{v_j} = \sum_{i=1}^{n} (p_{ij} - b_{ij}) \log(\frac{p_{ij}}{b_{ij}})$, where b_{ij} is the part of the backtesting population belonging to class i of variable j. If the variable is categorial, the classes are simply the modalities taken by variable j. If the variable is continuous, it is discretized into 10 equal groups in size (deciles) before calculating the IS_{v_j}. p_{ij} is the part of the development population belonging to class i of variable j.

The aim of this global indicator is to determine the average stability of the model in comparison to the reference population. The different thresholds of the IS_{vg} are described in the following table:



$IS_{vg} < 0.15$	High level of stability
$0.15 \leq IS_{vg} < 0.30$	Acceptable stability
$IS_{vg} \geq 0.30$	Instability

Figure 15 IS_{vg} thresholds

■ Contribution of variables

The estimation of the contributions of variables is the main difference between a classical logistic regression model and the *Catboost* model.
The standard method consists in using the coefficients c(i,j) of the obtained regression to find the contribution CTR(j) (i covers the modalities of the variable j).

$$CTR(j) = \frac{maxSC(j,i)}{10},$$

$$\text{with } SC(j,i) = 1000 * \frac{|c(i,j) - \alpha_j|}{\sum_j \underset{i}{max}\ c(i,j)}$$

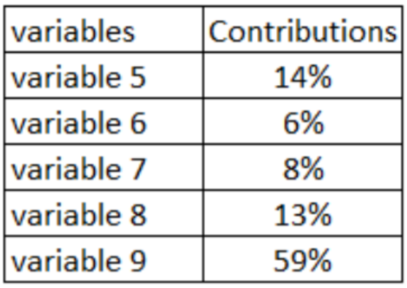
where $\alpha_j = max(c(i,j))$. The method of calculation in the case of machine learning models is less direct, meaning that one has to go through an algorithm for the computation of contributions. For the *Catboost* model, we propose to use the *Shapley Value* and *Prediction Value Change* to compute these contributions.

Prediction Value Change

The feature importance is based on the increase of the prediction error of the model after perturbation or permutation of the values of the variable in question. The more the perturbation induces a significant increase (or decrease) in the prediction error, the more important the variable is in our model. [2] summarized the feature importance calculation of the in several steps:

1. We first calculate the model error $e_{\hat{f}} = L(y, \hat{f}(X))$ where $\hat{f}$ is the model, X the matrix of predictors, y the target variable, and **L**(y, $\hat{f}$) the the loss function (MAE, RMSE, etc.).
2. For each variable, a new X_{pert} matrix is generated after perturbation of the variable in question. Then the prediction error e_{pert} is calculated.
3. We finally compute the feature importance e_{pert} - $e_{\hat{f}}$. The algorithm then uses this value to calculate the contribution of the variable.

The main advantage of this method is that it is easier to calculate and to interpret than the «Shapley Value» method. Furthermore, the contribution values are already normalized. Figure 16 presents the results issued from the *Prediction Value Change*.



variables	Contributions
variable 5	14%
variable 6	6%
variable 7	8%
variable 8	13%
variable 9	59%

Figure 16 Variable contributions using the *Prediction Value Change* approach

The most important variable is the variable 9 with a contribution equal to 59%. The less important one is the variable 6 with a contribution equal to 6%. Using these values and the different IS_{v_j} (IS of each variable), we obtain a value of IS_{vg} equal to 2.3%.

The Shapley Value

The Shapley value is a concept from cooperative game theory that provides a fair way to allocate the total contribution of a group of players to each individual player. It was introduced by Lloyd Shapley in 1953 and has since become a fundamental concept in various fields, including machine learning, economics, and political science. In the context of scoring, the following elements must be specified:

- The game: prediction related to a given observation.
- The players: set of values taken by the given observation on the different explanatory variables.
- The payoff: the predicted value of the observation minus the average prediction for all observations.

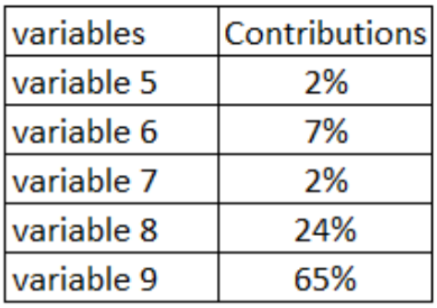
The idea of the shapley value method is to quantify the effect of each variable on the prediction of a point. For each possible subset of features excluding the one under consideration, the algorithm calculates the impact on the prediction of adding this feature. Formally

$$q_i = \sum_{S \subseteq F/\{i\}} \frac{|S|!(F-|S|-1)!}{F!}[f_{(S\cup\{i\})}(x_{(S\cup\{i\})}) - f_S(x_S)]$$

with,

i: feature
x: input
f: model

S: subset of features
F: set of all features



variables	Contributions
variable 5	2%
variable 6	7%
variable 7	2%
variable 8	24%
variable 9	65%

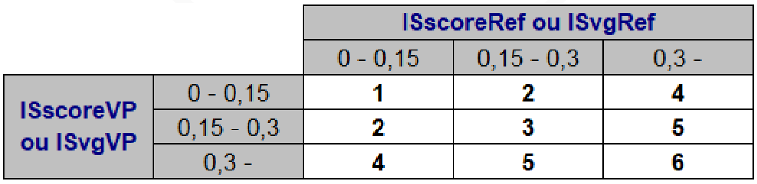
Figure 17 Variable contributions using the Shap Value approach

We obtain here a value of IS_{vg} equal to 2.1%. We therefore conclude that both approaches indicate a high level of stability for the IS_{vg} indicator.

We notive that the IS_{vg} issued from the two approaches are closed. Nevertheless, the *Shapley value* one is a little smaller than the *Prediction valu Change* one. If the bank aims to avoid redisigning the model, it would be more advantageous to employ the *Shapley value.* On the other hand, if the bank's intention is to adopt a conservative approach, it is recommended to employ the *Prediction value Change.*

4.5.2 IS status

The IS status is the intersection between IS_{score} and IS_{vg} results. It is important to know that these two indicators are not only calculated according to the reference population, but also according to the previous backtesting population. In our case, there is no previous backtesting. We therefore suppose that the IS_{score} and the IS_{vg} from the previous section belong to the "high level of stability" class, with a value of 1% for both indicators.



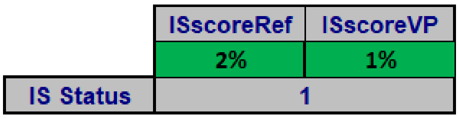
		ISscoreRef ou ISvgRef		
		0 - 0,15	0,15 - 0,3	0,3 -
ISscoreVP ou ISvgVP	0 - 0,15	1	2	4
	0,15 - 0,3	2	3	5
	0,3 -	4	5	6

Figure 18 Quantification of the IS status

-ISscoreRef (resp. ISscoreVP): IS_{score} comparing backtesting sample with the reference one (resp. the population of the previous backtesting session).

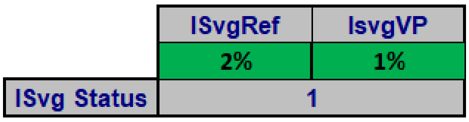
-ISvgRef (resp. ISvgVP): IS_{vg} comparing backtesting sample with the reference one (resp. the population of the previous backtesting session).

Based on Figure 18, we got:



	ISscoreRef	ISscoreVP
	2%	1%
IS Status	1	

Figure 19 IS status



	ISvgRef	IsvgVP
	2%	1%
ISvg Status	1	

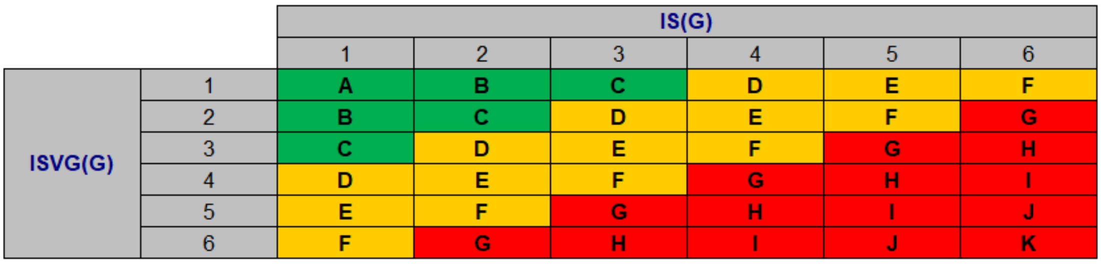
Figure 20 IS_{vg} status

Finally, since IS_{score} and IS_{vg} status are equal to 1, we conclude that our score has a high level of stability.

4.5.3 Global Indicator

The Global Indicator combines the IS and IS_{vg} status, the discrimination indicator (Gini index) and the temporal evolution of the latter ($\Delta Gini$ indicator).

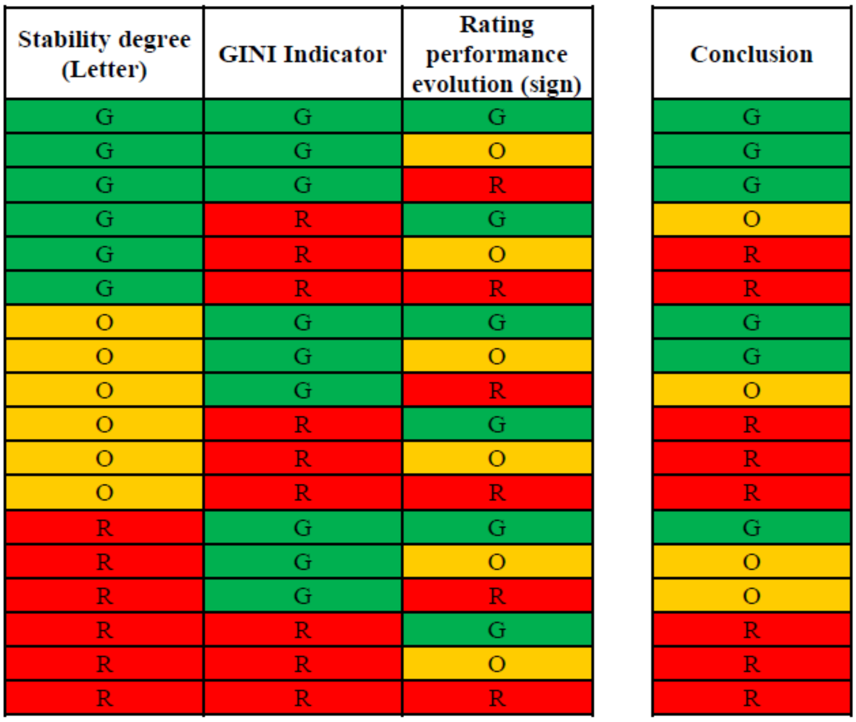
Letters (A,B,C,D,E,F,G,H,I,J or K) measure the stability degree of the rating grid by crossing the IS status with the IS_{vg} one. They are defined as follows:



		IS(G)					
		1	2	3	4	5	6
ISVG(G)	1	A	B	C	D	E	F
	2	B	C	D	E	F	G
	3	C	D	E	F	G	H
	4	D	E	F	G	H	I
	5	E	F	G	H	I	J
	6	F	G	H	I	J	K

Figure 21 Stability degrees of the rating grid

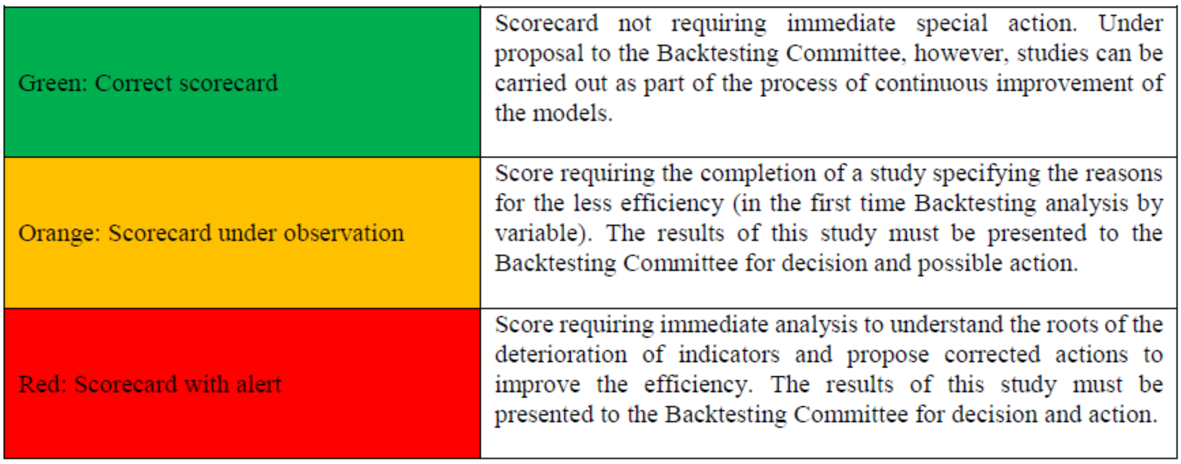
Using the stability degree obtained in the table above and the results concerning the performance levels of the Gini and *DeltaGini* indicators (see section 2.2.1), the quality of the scorecard is deduced from the following rules:



Stability degree (Letter)	GINI Indicator	Rating performance evolution (sign)	Conclusion
G	G	G	G
G	G	O	G
G	G	R	G
G	R	G	O
G	R	O	R
G	R	R	R
O	G	G	G
O	G	O	G
O	G	R	O
O	R	G	R
O	R	O	R
O	R	R	R
R	G	G	G
R	G	O	O
R	G	R	O
R	R	G	R
R	R	O	R
R	R	R	R

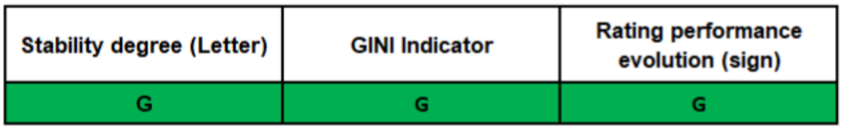
Figure 22 Scorecard quality

The following table describes the different colours defining the scorecard quality:



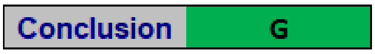
Green: Correct scorecard	Scorecard not requiring immediate special action. Under proposal to the Backtesting Committee, however, studies can be carried out as part of the process of continuous improvement of the models.
Orange: Scorecard under observation	Score requiring the completion of a study specifying the reasons for the less efficiency (in the first time Backtesting analysis by variable). The results of this study must be presented to the Backtesting Committee for decision and possible action.
Red: Scorecard with alert	Score requiring immediate analysis to understand the roots of the deterioration of indicators and propose corrected actions to improve the efficiency. The results of this study must be presented to the Backtesting Committee for decision and action.

Figure 23 Areas description



Stability degree (Letter)	GINI Indicator	Rating performance evolution (sign)
G	G	G

Figure 24 Global Indicator of our model



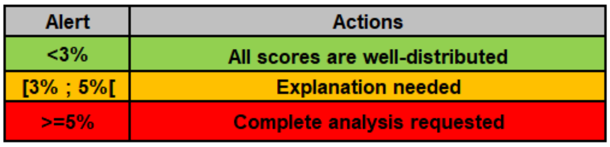
Conclusion	G

Figure 25 Backtesting conclusion

We conclude that our model is stable with a high level of performance and that our scorecard doesn't need any special action.

4.5.4 Score distribution analysis

This small analysis provides information about the distribution of scores. The goal is that there is not a big difference between the size of each rating. The following table gives the thresholds for each score size:



Alert	Actions
<3%	All scores are well-distributed
[3% ; 5%[	Explanation needed
>=5%	Complete analysis requested

Figure 26 Thresholds for each score size

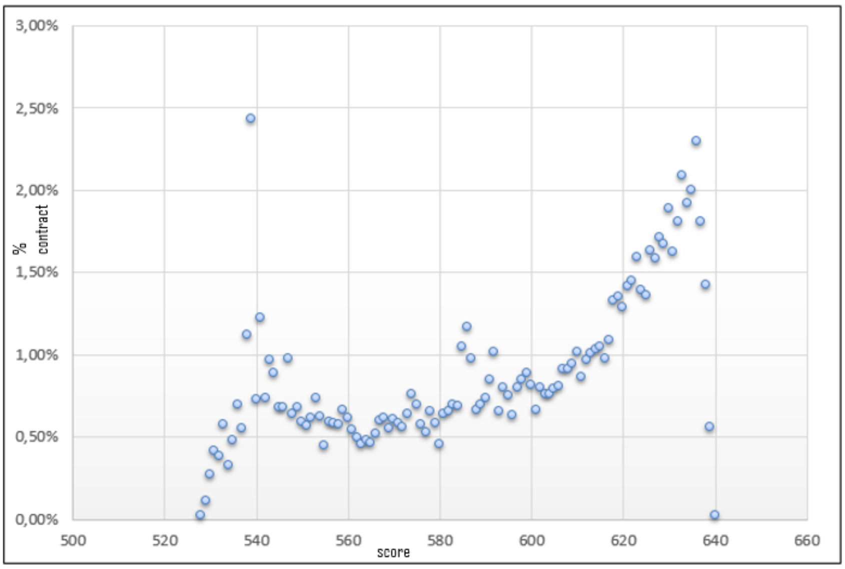


Figure 27 Our score distribution

We conclude from Figure 27 that our scores are well-distributed since all of them have less than 3% of the total population.

5 Conclusion

In this article, we used a *Catboost* that estimates the default probabilities of credit applicants. In fact, many studies have recently shown that machine learning scoring models provide now better classification performance than standard approaches as logistic regression. The choice of the *Catboost* model in particular is due to the fact that this algorithm minimizes the prediction computing time and avoids overfitting. Once the model has been implemented, we built our scorecard using the predicted values of default probabilities. We finally obtained a score grid made of 5 homogeneous risk classes.

In order to evaluate the quality of our scorecard, we performed a backtesting. The standard framework is not fully adapted to the use of *Catboost* approach. Along the study of stability, the evaluation of the contributions of variables was respectively achieved by using *Predcion Value Change* and *ShapValue* algorithms. We conducted a comparative analysis of both approaches and endeavored to draw conclusions regarding the optimal choice. Both approaches provide a high level of stability. If the bank aims to avoid redisigning the model, it would be more advantageous to employ the *Shapley value*. On the other hand, if the bank's intention is to adopt a conservative approach, it is recommended to employ the *Prediction value Change*. We also calculated the Gini and $\Delta Gini$ indexes which inferred a high level of performance, indicating that our model distinguishes well between good and bad contracts.

To deepen this study, we can apply a logistic regression model on the same data used for the *Catboost* in order to compare the results of the backtesting and make sure that our model provides better performance. We can also try another technique to compute the variables' contribution (the «LossFunctionChange» approach for example) and compare the statbily indicators with those obtained from the methods used in this study.

References

[1] Dumitrescu Elena. Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *EconomiX-CNRS, University of Paris Nanterres*, pages 1–2.

[2] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, 20(177):1–81, 2019.

[3] Naeem Siddiqi. *Credit risk scorecards: developing and implementing intelligent credit scoring*, volume 3. John Wiley & Sons, 2012.

[4] Jacques Silber. Factor components, population subgroups and the computation of the gini index of inequality. *The Review of Economics and Statistics*, pages 107–115, 1989.