ING Lion's Den 2024

Team: RiskBusters

Authors: Michał Bryzik, Michał Niegierewicz, Kacper Gruca, Jan Ślusarek

Below our Team presents in-depth description of our proposition for default predicting model as the answer for ING Lion's Den preliminary–task.

Dataset description

As a first step, it was decided to remove rejected applications. Since they do not have a default flag and have not been approved by the credit policy for launch, they do not add significant value for us. The number of accepted and rejected applications is shown below. Removing the rejected variables solved the problem of missing data for the target variable.

Accepted	: **36718**
Rejected	: **13282**

After removing the rejected applications, the missing values for the variables are as follows. The next section of this document will detail the process of cleaning the collection of problematic values. At this point, it is worth noting that the described missing data for most variables have their economic interpretation. Only for the variable Var17, i.e. cost estimation, it is difficult to find a justification for the occurrence of missing values. Missing data can be explained by the absence of a second borrower (Var10,Var12), a different purpose of the loan (Var18,Var19, Var8) and the absence of a current and savings account (Var18,Var19).

VARIABLE	NA
customer_id	0
application_date	0
target	0
Application_status	0
Var1	0
Var2	1018

Var3	1018
Var4	0
Var5	0
Var6	0
Var7	0
Var8	20538
Var9	0
Var10	28043
Var11	0
Var12	28043
Var13	0
Var14	0
Var15	0
Var16	0
Var17	32
Var18	27125
Var19	20538
Var20	0
Var21	0
Var22	0
Var23	0
Var24	0
Var25	7401
Var26	14649
Var27	0
Var28	0
Var29	0
Var30	0
_r_	0

Table 1 Missing values

The chart below shows the default rate calculated for each quarter. The values are compared on the chart to the default rate on the full sample, and the default rate through the cycle on the 7-year window. From the graph, we can see that the default rate has the characteristic of white noise, with no longer periods with a lower or higher default rate visible, possibly representing conjunctural cycles. Moreover, the DR appears consistent over time which allows us to use the full time window in our modeling. DR on the full sample is 3.07%. DRttc as an average of annual DR for the most recent 7 years is 3.04%.

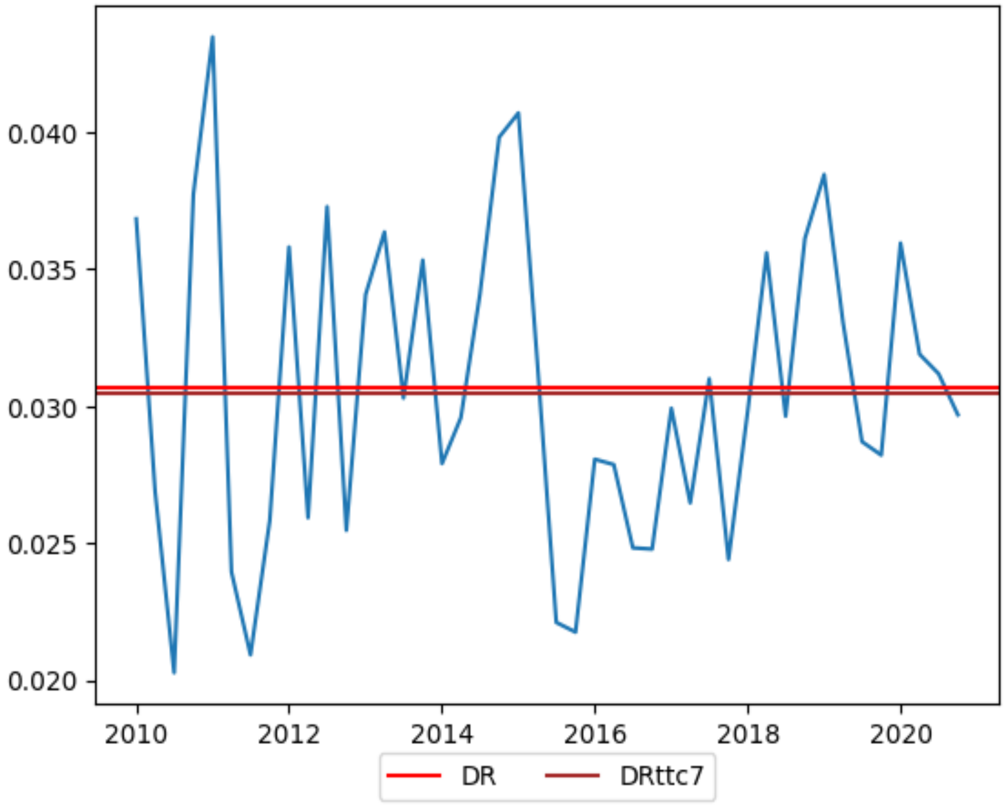


Chart 1 DR in time

Descriptive statistics of the variables were also analyzed as part of the modeling. Detailed results due to their size were not included in the final report. They can be found in the file 2_data_analysis.ipynb. The dataset after removing non-processed applications contains 36 columns, including identifiers like ID, Client_ID. It is worth mentioning here that at the client level there are cases with more than one exposure in the sample. The total number of observations is 36 718.

Feature engineering and the data treatments

As mentioned earlier, the data used had many minor problems. In order to use them in the modeling process, it was necessary to clean the data. Several additional variables were also created.

The data cleaning process began by setting a common date format for the application_date and Var13 variables. For the Var13 variable, more than 700 exposures had an NA value.

Based on these two variables, a new column was created to represent the working time in months from employment to the application date. This variable was named working_moths. A MOB variable was also created to represent the time in months from the date of application to today.

For categorical and continuous variables, the following steps were applied:

- Coded according to the dictionary of Direct and Online values for variable Var3.
- Completed the gaps in variable Var2 with information from variables Var8, Var18 and Var19, which clearly indicate for what purpose the loan was taken.
- An income variable was created - which is the sum of the income of the main borrower with the second borrower. The main idea behind this variable is that the loan installment can be spread over more than one salary, which should theoretically increase the quality of the loan application.
- The variable ii_ratio was created - it represents the ratio of installment to the sum of the income of the main borrower and the second borrower. As expected, as the ratio increases, credit risk increases, due to the greater financial burden on the borrower
- The variable idi_ratio was created - the variable is an improvement of the coefficient ii_ratio. It represents the ratio of installment to disposable income, calculated as the difference between the borrower's main income and estimated expenses. The economic interpretation is identical to the ii_ratio variable.
- Created variable loan_desc - based on variables Var2, Var18 and Var19, a new variable was created with more precise categories describing the purpose of the loan.
- Removed exposures with missing data in variable Var3 - this variable describes how the loan is sold, it was considered that observations without a preset sales channel are erroneous and not representative of the modeled phenomenon.

- Variable _r_ was removed - the variable was removed by its lack of description. The use of such a variable misses the business purpose. There was a suspicion of output from the previous model, which in addition may have imported into the model the risks associated with the poor quality of the previous model.

In the next step, due to the different needs of the algorithms used in modeling, it was decided to create two parallel samples. One was prepared for the fine and coarse classing process, in which data gaps can be managed by adding a new category. The sample containing continuous data plus one-hot encoding of categorical data could not contain missing data. To manage missing data for both samples, gaps in the Var17 variable were filled by the KNN algorithm.

For the sample on which fine categorization is applied, the following operations were carried out:

- for the variables loan_desc and Var12 data gaps were managed by adding a new category.
- variables Var2, Var18 and Var19 were removed - information from these variables is carried by a new variable labeled loan_desc. Moreover, the values of vehicles for which credit was taken were considered irrelevant.

As mentioned, a second sample, not adapted to the fine-classing approach, was created in parallel. On this sample, the management of missing data was carried out as follows:

- Var8 was removed,
- Deficiencies in variable var25 and var26 were filled with zeros.

The data were then subjected to fine classing and coarse classing. First, the values of the variables were grouped into appropriate categories (e.g., for continuous variables by deciles) and their value replaced by the Weight of Evidence statistic. In the fine classing process, the values of the categorical variables were expertly concatenated. New categories were created for missing data.

On such prepared data, we tested the discriminatory power using the Information Value statistic. In the first step, we removed variables that had a value of this statistic below 0.02 - this level can be interpreted as unusable variables. The variables are listed in the table below:

	VARIABLE	IV
2	Var11	0.237090

16	Var28	0.159687
25	idi_ratio	0.151682
14	Var26_q	0.122043
28	loan_desc	0.111645
27	income	0.091549
29	working_months_1	0.072218
17	Var29	0.067985
4	Var14	0.062022
13	Var25_q	0.060927
26	ii_ratio	0.056983
15	Var27	0.056081
20	Var4	0.046328
7	Var17_1	0.044461
24	Var8_q	0.037895
21	Var5	0.030886
3	Var12	0.028132
18	Var3	0.020321
10	Var22_1	0.016821
0	MOB	0.013903
11	Var23_1	0.012936
9	Var21_1	0.012588
6	Var16	0.009623
19	Var30	0.006522
5	Var15	0.006377
8	Var20_1	0.005253
22	Var6	0.003846
1	Var1	0.003693
23	Var7	0.003219
12	Var24	0.002038

Table 2 IV statistic for variables

The next step in the selection of variables was to calculate correlations between variables and remove pairs for which Kendall's Tau coefficient was greater than 0.6. Due to size, the correlation matrix was not included in the report. It can be found in the codes, in the file 1_5_1_WOE_analysis.ipynb.

Such a high correlation applied only to the variables:

- Var12_woe, Var1_woe
- Var15_woe, Var16_woe
- Var20_1_woe, Var21_1_woe
- Var21_1_woe, Var22_1_woe
- Var21_1_woe, Var23_1_woe
- Var22_1_woe, Var23_1_woe

The occurrence of such pairs was considered logical, Var20-Var23 variables represent the number of inquiries in the bio, Var15 and Var16 are the number of children and the number of dependents, respectively.

Variables in the pairs were ranked by IV value and variables that had lower discriminatory power were removed. The following variables were finally removed: Var1_woe, Var15_woe, Var20_1_woe, Var21_1_woe, Var21_1_woe, Var23_1_woe.

For the test data, the cleaning process was repeated analogously to the training data. For fine classing, deciles and WoE values calculated on the training sample were used. Finally, due to time constraints, the coarse grading process was not carried out. This carries the risk of overtraining the model.

LOGIT

In our analysis, the dependent variable within the logistic regression model is designated as 'target,' which assumes binary values. The independent variables incorporated into the model include: 'const', 'Var3_woe', 'Var4_woe', 'working_months_1_woe', 'Var27_woe', 'ii_ratio_woe', 'Var8_q_woe', 'loan_desc_woe', 'Var26_q_woe', 'idi_ratio_woe', 'Var14_woe', 'Var5_woe', 'Var25_q_woe', 'Var12_woe', 'Var29_woe', 'income_woe', 'Var17_1_woe', 'Var28_woe', and 'Var11_woe'. We initiated our analysis by constructing the simplest possible model, that is, by estimating parameters on the training set and subsequently evaluating its performance on the test set. It's important to highlight that these datasets (both the training and the test sets) have undergone a comprehensive and intricate Weight of Evidence (WOE) transformation. This transformation is a hallmark technique for addressing weak entropy in this type of data. For the sake of ease in interpretive and descriptive analysis, the estimation of the logit model was primarily conducted using the Statsmodels package in Python.

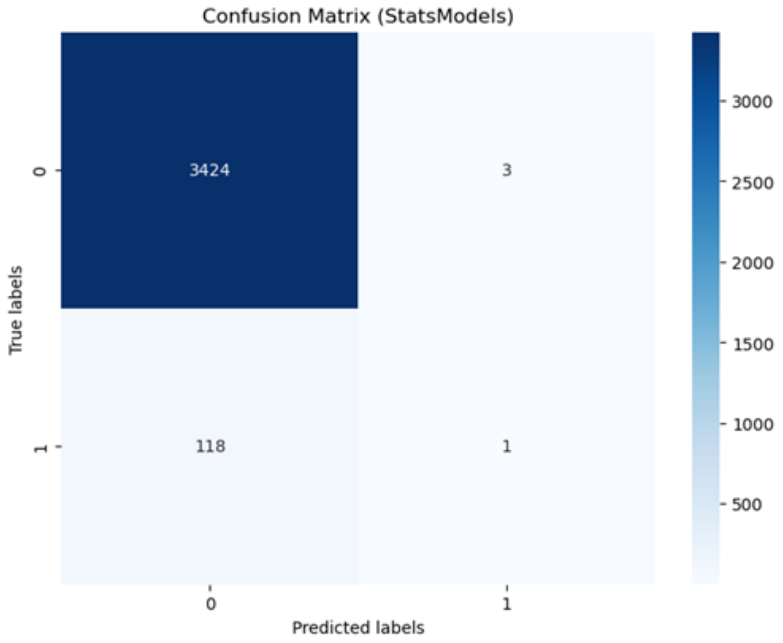


Chart 2 Confusion matrix

Regrettably, the predictive capabilities of the model, as demonstrated by the confusion matrix, proved to be quite deficient. Consequently, our subsequent step involved lowering the threshold, a common practice in handling more intricate classifications.

Following the analysis of the numerical iterations of the threshold, we determined that the optimal threshold would be 0.026. Below is a table featuring the confusion matrix for the test set, utilizing the same model but with the threshold set at precisely 0.026.

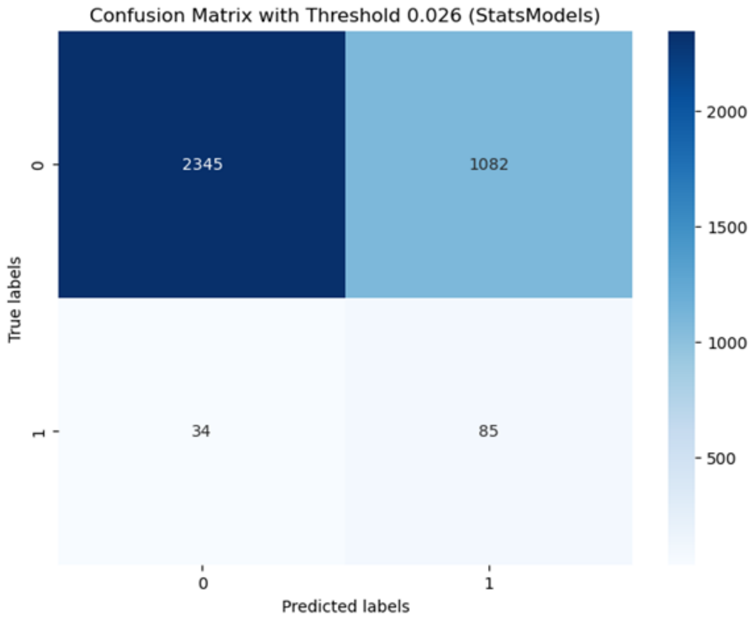


Chart 3 Confusion matrix

Regrettably, the predictive capabilities of the model, as demonstrated by the confusion matrix, proved to be quite deficient. Consequently, our subsequent step involved lowering the threshold, a common practice in handling more intricate classifications.

Following the analysis of the numerical iterations of the threshold, we determined that the optimal threshold would be 0.026. Below is a table featuring the confusion matrix for the test set, utilizing the same model but with the threshold set at precisely 0.026.

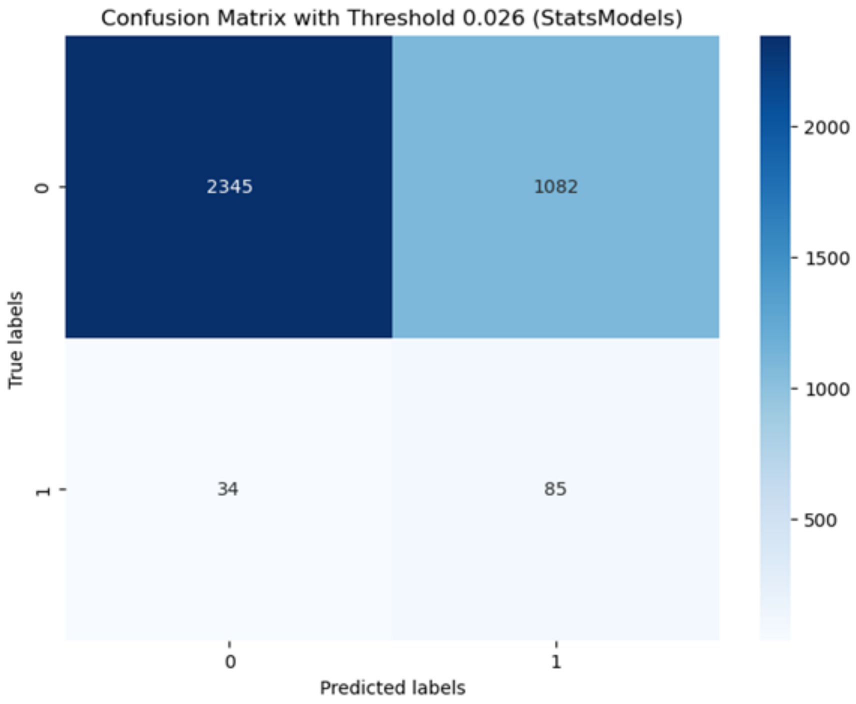
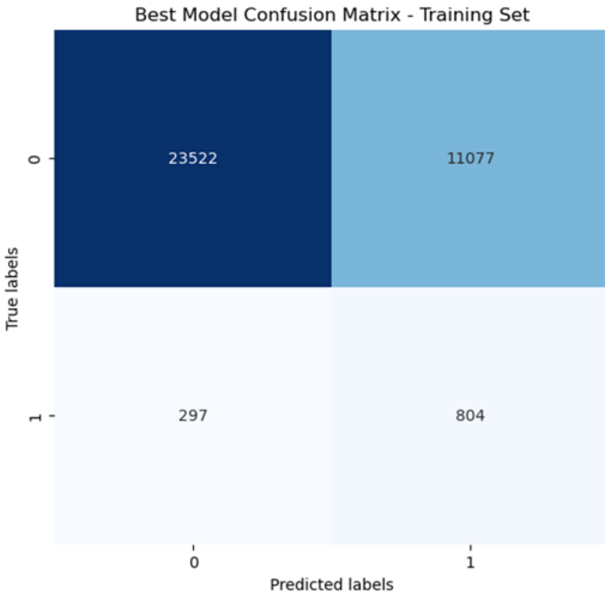
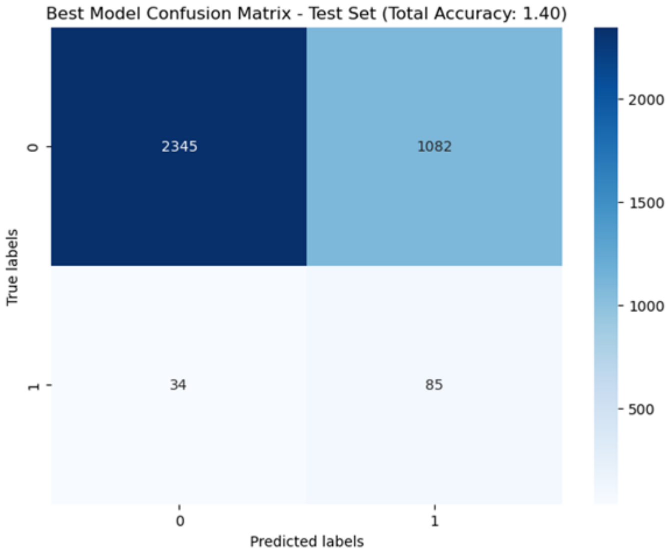


Chart 4 Confusion matrix

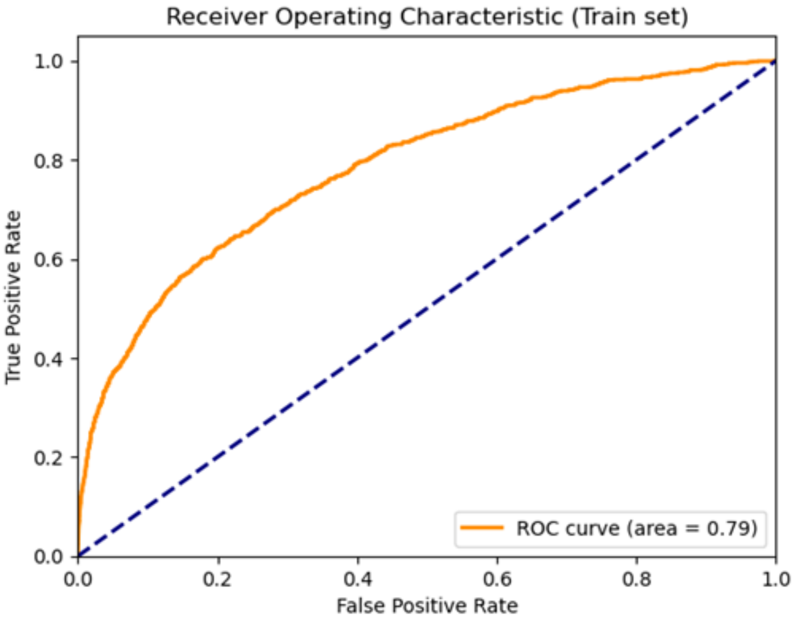
The results outlined above demonstrate a nearly astronomical improvement in the model's performance, which is highly encouraging. Moving forward, we will apply the Recursive Feature Elimination (RFE) method to ascertain whether a rank-based sequential elimination of variables from the model will further enhance its effectiveness. To achieve this, we developed an intricate algorithm in Python that optimized the model selection based on maximizing the sum of the percentage of TRUE POSITIVES and TRUE NEGATIVES. It later became evident that eliminating any variable from the model (including the constant) significantly diminished its predictive capabilities. Below is a visualization of the model's key characteristics.

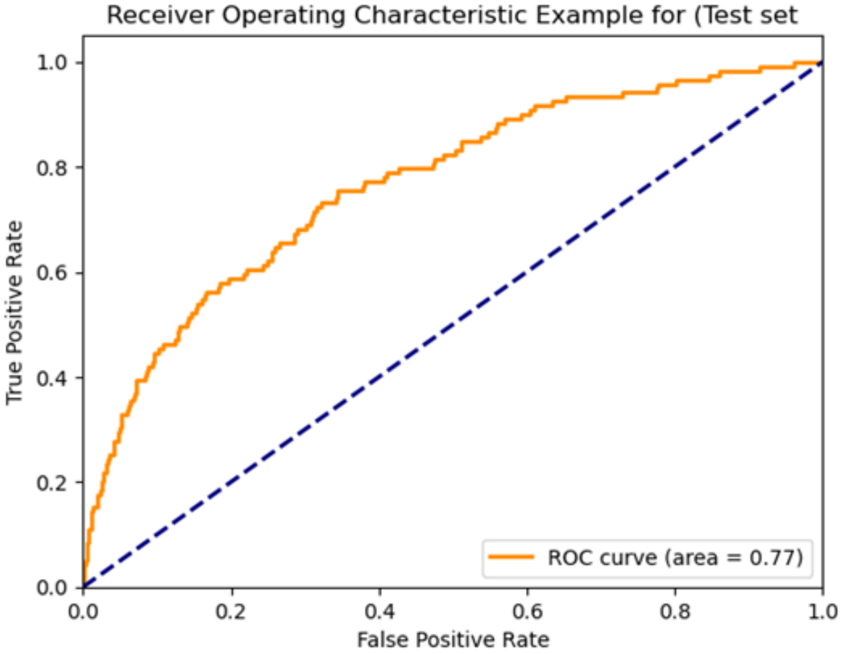


Best Model Confusion Matrix - Training Set



Best Model Confusion Matrix - Test Set (Total Accuracy: 1.40)


Receiver Operating Characteristic (Train set)
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
ROC curve (area = 0.79)


Receiver Operating Characteristic Example for (Test set
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
ROC curve (area = 0.77)

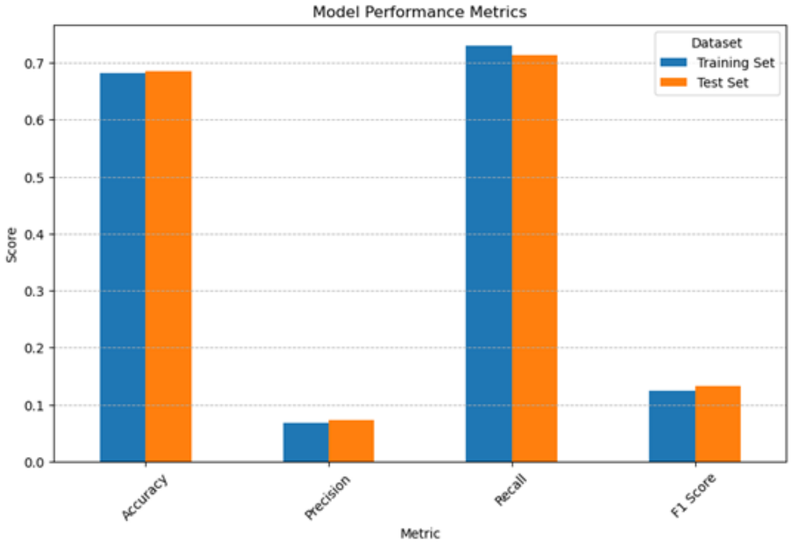


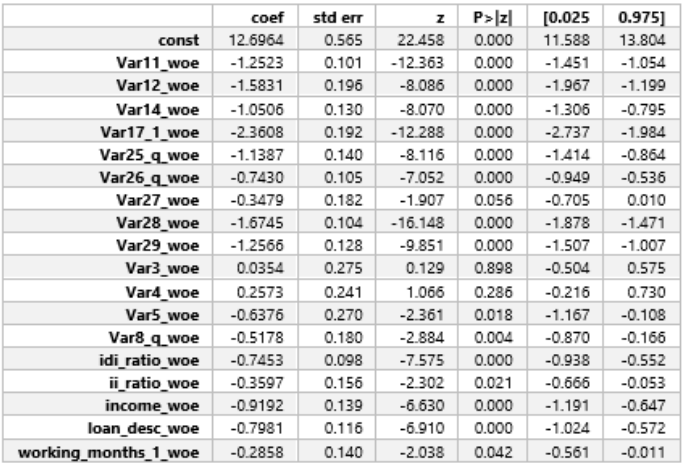
Chart 5 Classification metrices

For both the training and testing sets, the Area Under the ROC Curve (AUC) ranges between 0.7 and 0.8, indicating a fair classification capability. Moreover, the confusion matrix for the training set does not significantly differ from that of the testing set in terms of the percentage of correct classifications.

We further performed parameter tuning by dividing the training set into a smaller training subset and an internal validation subset, after which the model was evaluated again on the actual test set. However, this technique did not yield any statistically significant improvements in this case.

In summary, the model we have developed is relatively well-suited for bankruptcy risk classification. This is evidenced by the reasonably good true classification rates for both 0s and 1s on both the training and test sets.

Below is a comprehensive description of the variables in the optimal model.



	coef	std err	z	P>|z|	[0.025	0.975]
const	12.6964	0.565	22.458	0.000	11.588	13.804
Var11_woe	-1.2523	0.101	-12.363	0.000	-1.451	-1.054
Var12_woe	-1.5831	0.196	-8.086	0.000	-1.967	-1.199
Var14_woe	-1.0506	0.130	-8.070	0.000	-1.306	-0.795
Var17_1_woe	-2.3608	0.192	-12.288	0.000	-2.737	-1.984
Var25_q_woe	-1.1387	0.140	-8.116	0.000	-1.414	-0.864
Var26_q_woe	-0.7430	0.105	-7.052	0.000	-0.949	-0.536
Var27_woe	-0.3479	0.182	-1.907	0.056	-0.705	0.010
Var28_woe	-1.6745	0.104	-16.148	0.000	-1.878	-1.471
Var29_woe	-1.2566	0.128	-9.851	0.000	-1.507	-1.007
Var3_woe	0.0354	0.275	0.129	0.898	-0.504	0.575
Var4_woe	0.2573	0.241	1.066	0.286	-0.216	0.730
Var5_woe	-0.6376	0.270	-2.361	0.018	-1.167	-0.108
Var8_q_woe	-0.5178	0.180	-2.884	0.004	-0.870	-0.166
idi_ratio_woe	-0.7453	0.098	-7.575	0.000	-0.938	-0.552
ii_ratio_woe	-0.3597	0.156	-2.302	0.021	-0.666	-0.053
income_woe	-0.9192	0.139	-6.630	0.000	-1.191	-0.647
loan_desc_woe	-0.7981	0.116	-6.910	0.000	-1.024	-0.572
working_months_1_woe	-0.2858	0.140	-2.038	0.042	-0.561	-0.011

· const (12.6964): The constant term is quite high, which suggests that when all other variables are at zero, the log-odds of the default is significantly positive. However, since all other variables will not be zero in practice, this number mainly serves as an anchor point for the model.

· Var11_woe (Profession of main applicant, -1.2523): The profession of the main applicant has a negative association with the default probability. This suggests that certain professions are less likely to default on their loans.

· Var12_woe (Profession of second applicant, -1.5831): Similar to Var11, the profession of the second applicant also shows a negative relationship with defaulting, indicating that the profession of both applicants is crucial in predicting loan performance.

· Var14_woe (Marital status of main applicant, -1.0506): The marital status of the main applicant has a significant negative coefficient, suggesting that certain marital statuses are associated with a lower risk of default.

- Var17_1_woe (Spendings estimation, -2.3608): The estimated spendings have a strong negative effect on the default probability. This might mean that applicants with higher estimated spending are less likely to default, perhaps due to better financial management or higher disposable income.

- Var25_q_woe (Amount on current account, -1.1387): A negative coefficient indicates that higher amounts in the current account are associated with a lower probability of default.

- Var26_q_woe (Amount on savings account, -0.7430): This variable also shows a negative relationship with the default risk, meaning that more savings correlate with a reduced risk of default.

- Var28_woe (Arrear in last 12 months, -1.6745): Past arrears are strongly negatively related to defaulting, suggesting that applicants without past dues are less likely to default.

- Var29_woe (Credit bureau score, -1.2566): A higher credit score is associated with a lower default risk, which aligns with the expected behavior that a good credit score reflects better creditworthiness.

- Var5_woe (Credit duration, -0.6376): Longer credit duration is associated with a lower probability of default, which might indicate that loans with longer terms have more manageable repayment schedules.

- Var8_q_woe (Value of the goods, car, -0.5178): The negative coefficient implies that higher car values are associated with a lower risk of default. This may reflect applicants with more expensive cars having better financial stability.

- idi_ratio_woe (-0.7453) and ii_ratio_woe (-0.3597): These variables, presumably related to income or indebtedness ratios (as the name suggests), both have negative coefficients, indicating that better ratios (e.g., higher income to debt or installment to income ratios) are protective against default.

- income_woe (-0.9192): A negative coefficient for income suggests that higher incomes are associated with lower default risks.

- loan_desc_woe (-0.7981): This could be related to the description or quality of the loan, with a negative coefficient indicating that certain characteristics of the loan are linked with reduced default risk.

- working_months_1_woe (-0.2858): The negative coefficient suggests that applicants with a longer duration of current employment are less likely to default, likely reflecting job stability as a factor in creditworthiness.

Despite the p-values indicating that the variables: Var3_woe (p-value: 0.898), Var4_woe (p-value: 0.286), and Var27_woe (p-value: 0.056) are statistically insignificant, it was found that they hold predictive significance. The exclusion of these variables did not lead to a positive effect on the model's performance.

Task. 1.2.

Methodology Overview

Our task was to develop a robust credit risk assessment model to estimate the probability of default for customers applying for an installment loan product. To meet the challenging task, we adopted a systemic approach, focusing on developing the highest possible predictive power, even at the cost of some model parsimony.

To achieve this, we decided to start with 6 different classifiers:

- Random Forest
- XGBoost
- LightGBM
- Support Vector Machine (SVC)
- CatBoost
- K-Nearest Neighbors (KNN)

Then, their parameters are hypertuned using grid search cross-validation and the prediction is performed on a test dataset.

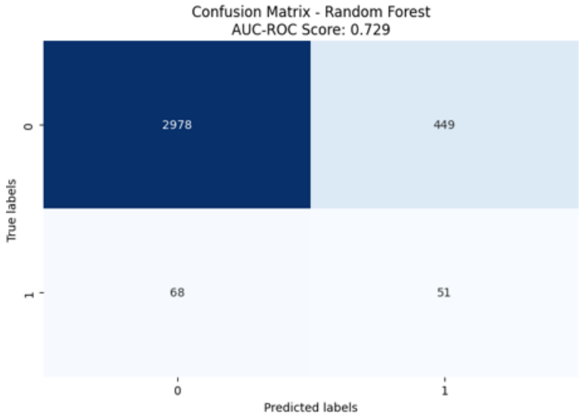
As the given dataset is characterized by a significant imbalance of the dependent variable, we have tried to work with upsampling [SMOTE, random oversampling] and downsampling [random downsampling, NearMiss]. However, due to the higher computational complexity [using upsampling results in a larger training dataset] and the lack of statistical difference between the upsampled and downsampled approaches, we have decided to continue with the downsampling method as our way of dealing with unbalanced data.

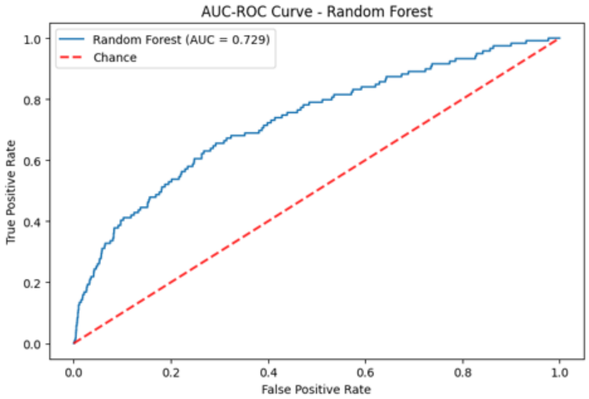
Evaluation Metrics

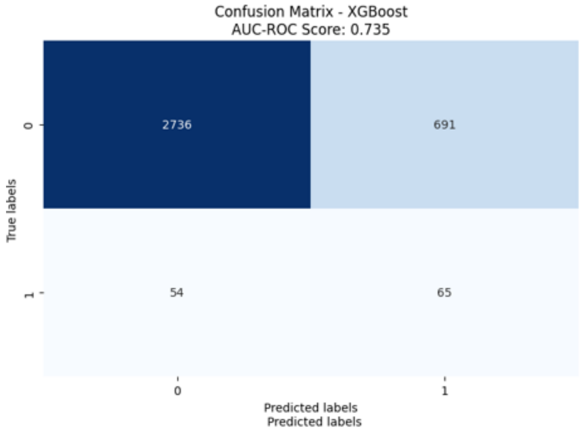
In order to assess the predictive power of a particular configuration of parameters for each of the models, we need to use specific metrics. Since we are dealing with unbalanced data, we have decided to compare AUC ROC, F1 score [also taking into account F0.5 score and F2 score], PR AUC and the balanced accuracy. However, in the end, because of the use of downsampling, we can also use the accuracy metric without fear of misleading performance.

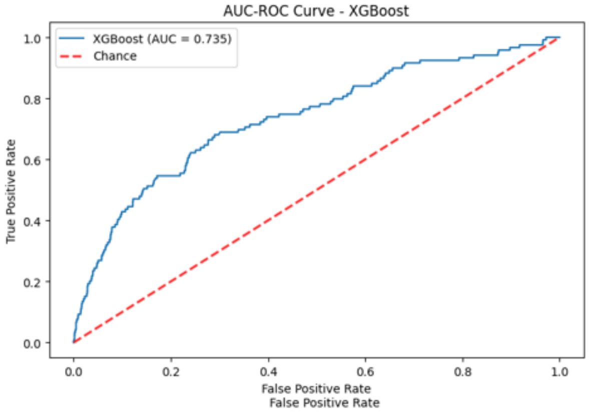
Predictive Results

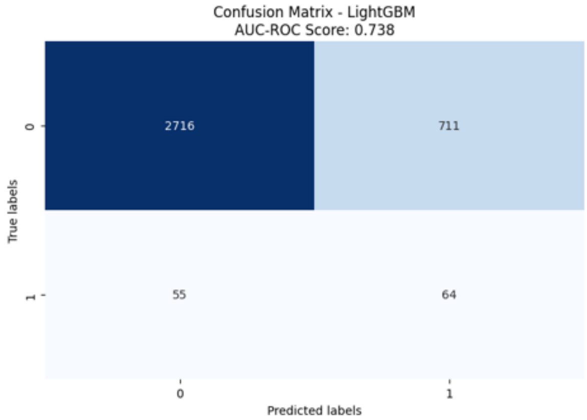
In order to asses the specific models performance let’s look at the charts below:

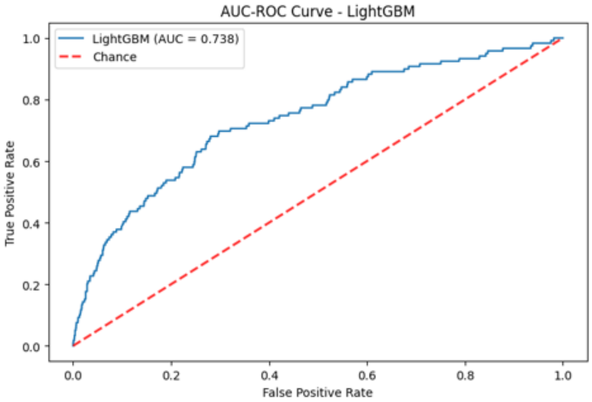


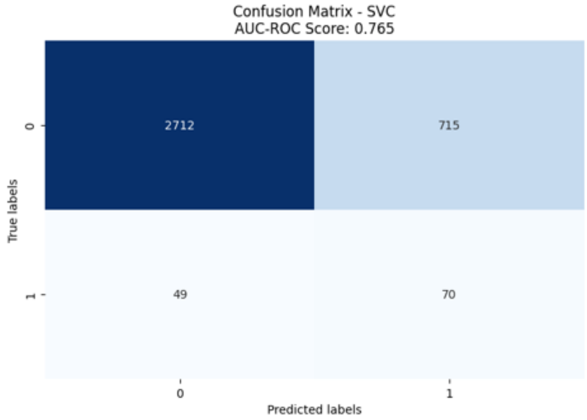


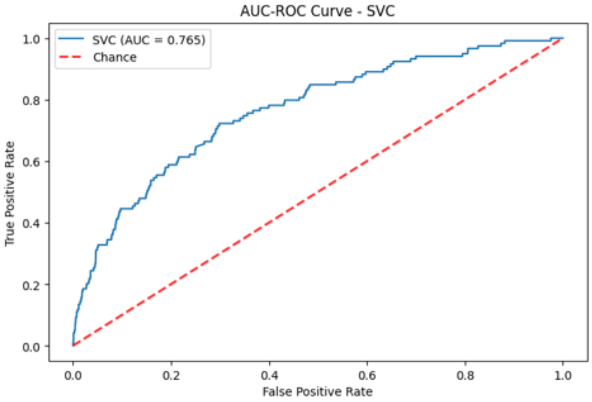


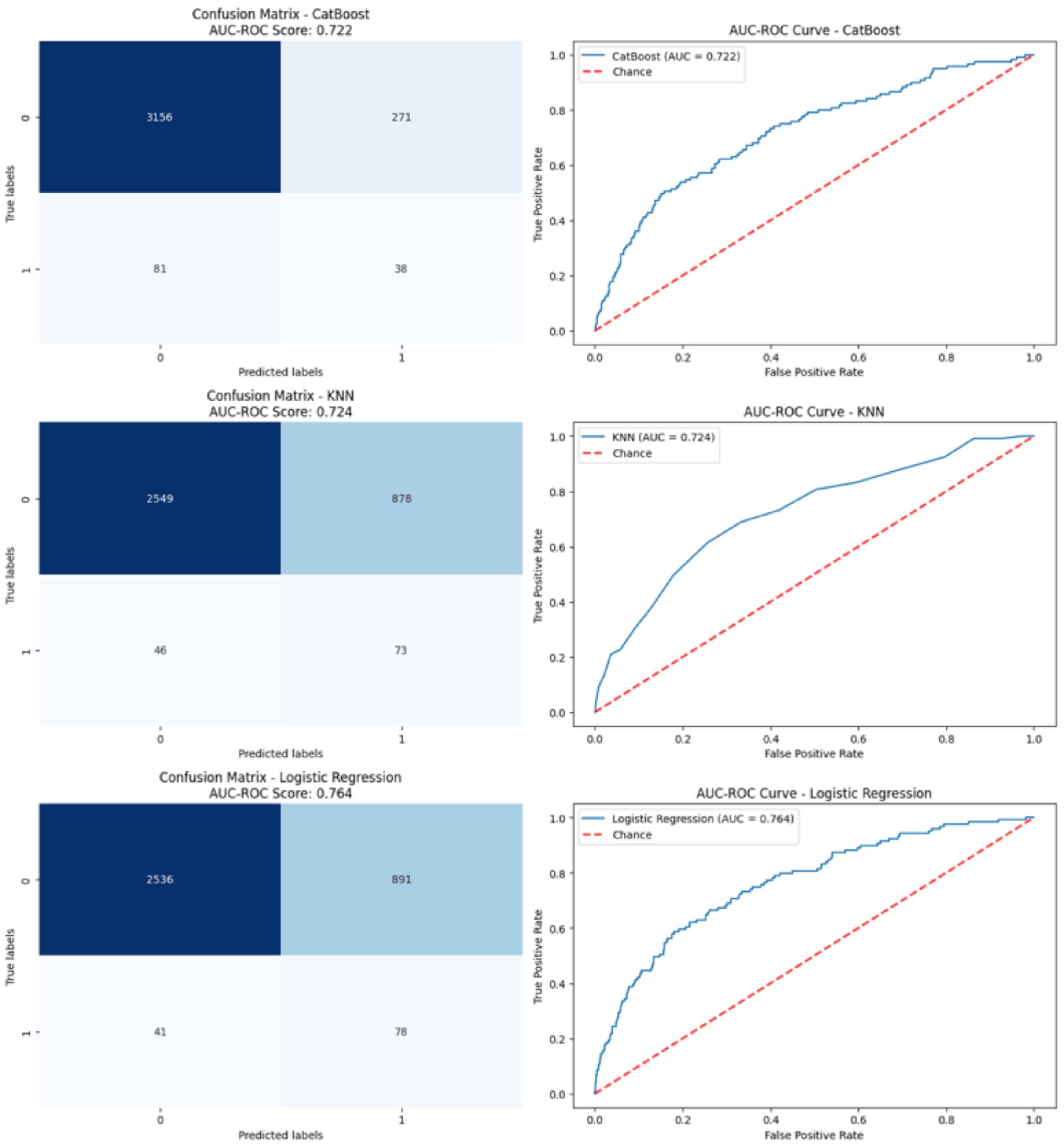




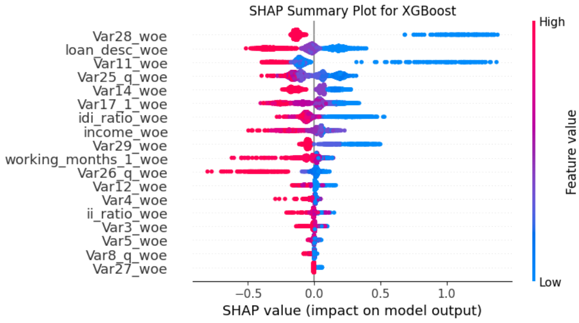


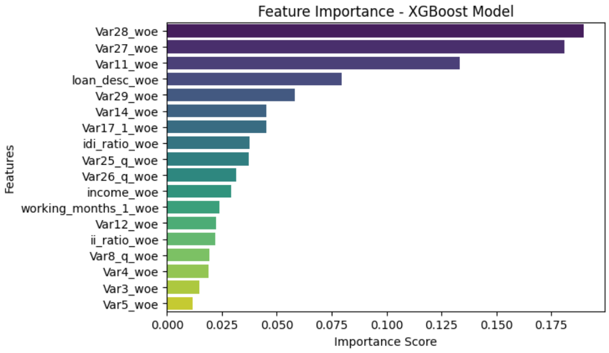






As we can see from the graphs above, the machine learning models didn't really bring much improvement in predictive capabilities, in most cases they even had a worse predictive performance than the proposed logit model in the first step. As our final decisive metric was the area under the curve of the ROC curve, we should also consider the SVC [Support Vector Classifier] model in addition to the logistic regression model, as both of them have a Gini index just above 50%, which could classify them as acceptable, obviously depending on different business needs. For the needs of our analysis, we used WoE transformed variables and decided to use all variables for prediction [as the use of e.g. Recursive Feature Elimination didn't really have an impact and the computational complexity wasn't an issue at this point], although some variables were consistently more impactful than others, as seen in the graphs below.





Support Vector Classifier

The main purpose and reason for using Support Vector Machines to predict the standard classification was their ability to reflect non-linear relationships, their effectiveness with small/medium datasets [as we used downsampling], and their high tunability [as we observed high score improvement with further iterations of hyperparameter tuning].

However, the approach itself has its own flaws, which have been reflected throughout the whole process. The relatively high computational dependency has made upsampling extremely cost-ineffective, and we were forced to try downsampling instead. The high tuning dependency has resulted in a significant amount of time spent on tuning the model, which could have been used to gain a broader understanding of the economic background for the topic and to perform deeper feature engineering. As a result, we observed that logistic regression, tuned with L1 regularization and calculated with the Saga solver, performed similarly or even better than the proposed model. Since the dataset was average, there was not much room for gradient boosting models to perform better than simple logistic regression.

Conclusion

Although the models may produce accurate results, machine learning models are often referred to as 'black boxes' and can be difficult to interpret without specialized knowledge. This can require banks to have higher computational capabilities and specific workplaces to test, store, and compare numerous models and methods for interpreting the results. However, as XAI (Explainable Artificial Intelligence) continues to develop and become more widespread, we may see an increase in the implementation of machine learning models.