Comparative Analysis of Probability of Default Models: errata DRAFT

Date of preparation: 17. 06. 2024

Author: Stevan Vujcic

After submitting the Comparative Analysis of Probability of Default Models Master's thesis on 26th May, 2024, the thesis solver identified an unintended usage of a variable in Chapter 3. The variable "behavioral_score" was used. Initially, it was assumed that this variable reflected the behavioral score of clients related to their non-home loan exposure lines. However, upon opponent's review the thesis solver studied the construction of this variable in internal modeling documentations of ČSOB. It was found the variable represents the fitted scores for the home loan portfolio itself. Although it can be argued that the variable can still be used for modeling, it was not the thesis solvers' intention to do so.

The list of corrections documents all parts of the submitted thesis which are replaced with an adjusted calculation in which the "behavioral_score" is excluded. Note that the thesis solver still allows the usage of another variable – "retail_behavioral_score". This is due to the fact that the variable reflects the initial intention to use the behavioral score of clients on their non-home loan exposure lines.

In the remainder of this document, the descriptions of the corrections are formatted *italic* whilst the actual corrections are formatted **bold**. The unchanged parts of the submitted thesis that are retained for context and completeness have standard, unhighlighted text formatting. Note that even when performance metrics are the same as in the submitted thesis text using 2 decimal points rounding, the new figures are formatted **bold** due to discrepancies that emerge as a result of rounding and just the fact that the models are not the same. The same applies for the hyperparameters and any other model-specific figures.

1 List of Corrections

The corrections are listed as follows.

1. *The binning of the variable "behavioral_score" is removed from Section 3.3.3.1 on page 47. Figure 6 is replaced with the binning of the variable "retail_behavioral_score". The text describing Figure 6 on the same page remains unchanged due to the fact that it still corresponds to the updated figure. The corrected Figure 6 is shown below.*

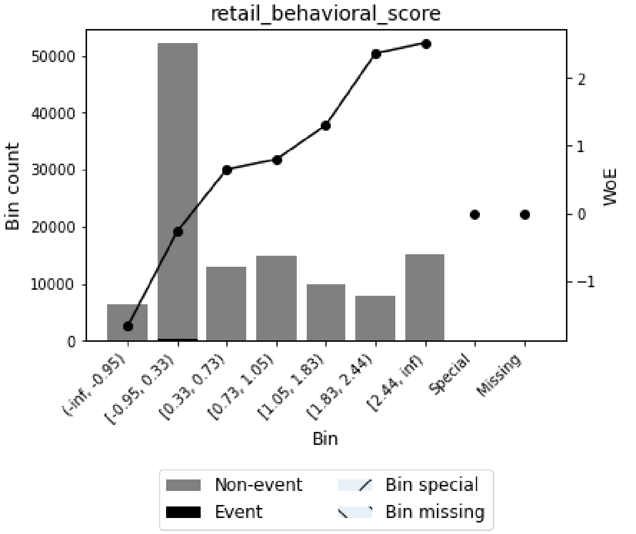

Figure 6: binning and WoE values of behavioral score, source: author

2. *In Section 3.3.3.1, the list of features upon shortlisting shown in Table 12 on pages 49 and 50 is replaced with a new list of features for that same table. The new list of features does not contain the "behavioral_score" variable and it newly contains the "retail_behavioral_score" variable. The corrected Table 12 is summarized below.*

Table 12: feature shortlist obtained after data preprocessing and feature selection

Feature	# of bins	IV	Gini
retail_behavioral_score	**7**	**1.022**	**0.481**
debt_summary_2qs_max_amt	**2**	**0.779**	**0.432**
days_in_deliquency_6m_avg_count	**2**	**1.035**	**0.384**
interest_paid_to_next_installment_6m_min_ratio	**9**	**0.504**	**0.377**
fee_paid_mtd_amt	**6**	**0.416**	**0.325**
collateral_required_amt	**8**	**0.359**	**0.315**

Feature	# of bins	IV	Gini
education_categorical	**3**	**0.420**	**0.315**
penalty_interest_paid_mtd_amt	**2**	**0.694**	**0.304**
interest_paid_6m_max_to_next_payment_cat	**3**	**0.487**	**0.288**
product_type_cd	**4**	**0.346**	**0.288**
fee_mtd_to_installment_ratio	**6**	**0.297**	**0.285**
brki_installment_amt	**3**	**0.318**	**0.265**
credit_turnover_2qs_avg_amt	**7**	**0.240**	**0.252**
income_to_expense_all_applicants_ratio	**10**	**0.214**	**0.249**
principal_paid_6m_avg_amt	**9**	**0.185**	**0.229**
paid_to_limit_ratio	**7**	**0.223**	**0.214**
interest_paid_3m_min_amt	**7**	**0.217**	**0.205**
principal_paid_to_outstanding_6m_max_ratio	**7**	**0.129**	**0.202**
no_fee_flg	**2**	**0.214**	**0.193**
expense_all_aplicants_to_next_installment_ratio	**8**	**0.124**	**0.192**
age	**10**	**0.114**	**0.184**
main_obj_value_amt	**5**	**0.151**	**0.183**
od_limit_utilization_amt	**2**	**0.234**	**0.181**
interest_paid_6m_max_amt	**5**	**0.182**	**0.179**
client_capital_to_paid_mtd_ratio	**5**	**0.166**	**0.172**
since_live_acc_opening_mths_count	**5**	**0.120**	**0.172**
client_income_amt	**8**	**0.104**	**0.161**
principal_paid_6m_max_amt	**8**	**0.084**	**0.159**
ltv_at_loan_origination_ratio	**6**	**0.139**	**0.156**
dsti_ratio	**7**	**0.080**	**0.154**
main_applicant_expense_amt	**6**	**0.080**	**0.150**
installments_count	**6**	**0.086**	**0.148**
collateral_value_to_outstanding_ratio	**6**	**0.067**	**0.134**
fixation_to_installments_ratio	**6**	**0.068**	**0.132**
all_applicants_expense_amt	**7**	**0.082**	**0.128**
principal_paid_to_outstanding_3m_avg_ratio	**8**	**0.050**	**0.123**
debt_summary_mtd_amt	**1**	**0.053**	**0.111**
fixation_period_mths_count	**3**	**0.064**	**0.109**
limit_pct	**3**	**0.056**	**0.105**

Source: author

3. *In Section 3.3.3.1, the output of the batch (1) logistic regression model as shown in Table 13 on page 50 is replaced in accordance with the introduced changes about the "behavioral_score" variable. The new Table 13 is summarized below.*

Table 13: logistic regression summary

Variable	Coefficient	Std. error	[0.025]	[0.975]
penalty_interest_paid_mtd_amt	**-0.295**	**(0.018)*****	**-0.330**	**-0.260**
fee_paid_mtd_amt	**-0.289**	**(0.017)*****	**-0.323**	**-0.256**
interest_paid_6m_max_amt	**-0.226**	**(0.029)*****	**-0.283**	**-0.169**
days_in_deliquency_6m_avg_count	**-0.578**	**(0.016)*****	**-0.610**	**-0.547**
installments_count	**-0.616**	**(0.038)*****	**-0.690**	**-0.543**
age	**-0.750**	**(0.031)*****	**-0.812**	**-0.689**
main_applicant_expense_amt	**-0.803**	**(0.042)*****	**-0.885**	**-0.720**
limit_pct	**1.313**	**(0.054)*****	**1.208**	**1.418**
main_obj_value_amt	**-2.595**	**(0.049)*****	**-2.691**	**-2.500**
brki_installment_amt	**-10.104**	**(0.152)*****	**-10.401**	**-9.807**
debt_summary_2qs_max_amt	**-0.598**	**(0.017)*****	**-0.632**	**-0.563**
retail_behavioral_score	**-0.625**	**(0.014)*****	**-0.652**	**-0.598**
product_type_cd	**-0.542**	**(0.022)*****	**-0.584**	**-0.499**
paid_to_limit_ratio	**1.343**	**(0.026)*****	**1.292**	**1.395**
collateral_value_to_outstanding_ratio	**-0.482**	**(0.05)*****	**-0.580**	**-0.384**
income_to_expense_all_applicants_ratio	**0.010**	**(-0.024)**	**-0.037**	**0.057**
expense_all_aplicants_to_next_installment_ratio	**-0.060**	**(0.029)***	**-0.117**	**-0.002**
interest_paid_to_next_installment_6m_min_ratio	**-0.270**	**(0.018)*****	**-0.305**	**-0.235**
education_categorical	**-0.446**	**(0.016)*****	**-0.477**	**-0.414**

*** represents significance at 0.01, ** at 0.01 and * at 0.05, source: author

4. *In Section 3.3.3.1, the ROC curves plot and AUCs reporting in Figure 9 on page 51 is replaced to reflect the removal of the "behavioral_score" variable from the modeling procedure. The corrected Figure 9 is shown below.*

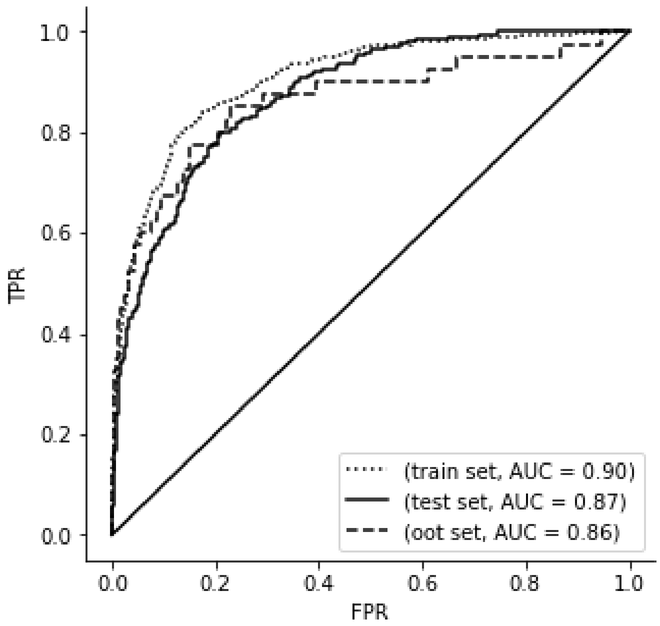

Figure 9: ROC curves of logistic regression model from batch (1), source: author

5. *In Section 3.3.3.2 on pages 51 and 52, the reported AUCs and hyperparameter specifications of the batch (1) models are updated to reflect the removal of the "behavioral_score" variable from the modeling procedure. This is performed by updating the entries of Table 14. Furthermore, the description corresponding to that table is updated in parts that are formatted bold.*

The batch (1) models are estimated using the full sample of train data (no undersampling), WoE-transformed features as well as a multicollinearity removal have the specifications summarized in the following table. The LR, ANN and boosting models exhibit solid performance with a gradual decrease in AUC across different samples. KNN and bagging overfit the train data. **[removed sentence]**. **[removed_sentence**]. **The** test and OOT performance **of the SVM model** is **[removed part of sentence]** alike the one of KNN. Overall, it can be concluded that **most** models estimated under these settings have a solid performance.

Table 14: optimal hyperparameters of batch (1) models

Model type	Hyperparameter specification	AUC train	AUC test	AUC OOT
LR	n.a.	**0.90**	**0.89**	**0.83**
ANN	Activation: **tahn** Hidden layers: **[7, 7]**	**0.88**	**0.87**	**0.81**

Model type	Hyperparameter specification	AUC train	AUC test	AUC OOT
	Learning rate: **constant** Solver: **adam**			
KNN	# of neighbors: **10** Weights: **uniform** Distance: **Euclidean**	**0.99**	**0.62**	**0.55**
SVM	C: **0.25** Polynomial degree: **1** Kernel: **linear**	**0.52**	**0.50**	**0.52**
Bagging	# of estimators: **100**	**1.00**	**0.73**	**0.75**
RF	Maximum tree depth: **1** Minimum samples per leaf: **10** # of estimators: **10**	**0.85**	**0.84**	**0.80**
Boosting	# of estimators: **10**	**0.87**	**0.85**	**0.81**

Source: author

6. *In Section 3.3.3.2 on pages 52 and 53, the reported AUCs and hyperparameter specifications of the batch (2) models are updated to reflect the removal of the "behavioral_score" variable from the modeling procedure. This is performed by updating the entries of Table 15. Furthermore, the description corresponding to that table is updated in parts that are formatted bold.*

The batch (2) models are estimated using the full sample of train data (no undersampling), WoE-transformed features whilst multicollinearity is not removed prior to model estimation. The LR model is estimated using the elastic net approach, where the L1 ratio of **0.9** points that both Ridge and Lasso are used. The result of that estimation is similar to the one above. The ANN has a slightly **better** performance than the previously estimated one. In addition, the algorithm retained the **similar** simple structure of the neural network [**removed part of sentence**]. It appears that the dataset with correlated features further complicates the performance of **[removed part of sentence]** SVM. A milder drop is noticed in the case of RF and bagging as well. Boosting is largely consistent with the previous estimate.

Table 15: optimal hyperparameters of batch (2) models

Model type	Hyperparameter specification	AUC train	AUC test	AUC OOT
LR	L1 ratio: **0.9**	**0.91**	**0.88**	**0.83**

Model type	Hyperparameter specification	AUC train	AUC test	AUC OOT
ANN	Activation: **sigmoid** Hidden layers: **[7]** Learning rate: **constant** Solver: **adam**	**0.88**	**0.87**	**0.83**
KNN	# of neighbors: **10** Weights: **uniform** Distance: **Manhattan**	**0.99**	**0.62**	**0.58**
SVM	C: **0.25** Polynomial degree: **1** Kernel: **Polynomial**	**0.42**	**0.44**	**0.33**
Bagging	# of estimators: **100**	**1.00**	**0.80**	**0.72**
RF	Maximum tree depth: **1** Minimum samples per leaf: **10** # of estimators: **10**	**0.79**	**0.76**	**0.78**
Boosting	# of estimators: **10**	**0.87**	**0.85**	**0.82**

Source: author

7. *In Section 3.3.3.2 on page 53, the reported AUCs and hyperparameter specifications of the batch (3) models are updated to reflect the removal of the "behavioral_score" variable from the modeling procedure. This is performed by updating the entries of Table 16. Furthermore, the description corresponding to that table is updated in parts that are formatted bold.*

The batch (3) models are estimated using the full sample of train data (no undersampling), dummy-transformed features and a multicollinearity removal. No feature selection is performed. Again, the LR estimation's performance is overlapping with the previous two models. Along with boosting **and RF**, LR is the only model that maintains decent performance as measure

d by AUC. The remaining estimates – ANN, KNN, SVM **and** bagging **[removed part of sentence]** do not perform well.

Table 16: optimal hyperparameters of batch (3) models

Model type	Hyperparameter specification	AUC train	AUC test	AUC OOT
LR	**n.a.**	**0.90**	**0.89**	**0.81**

Model type	Hyperparameter specification	AUC train	AUC test	AUC OOT
ANN	Activation: **sigmoid** Hidden layers: **[7]** Learning rate: **constant** Solver: **adam**	**0.59**	**0.61**	**0.69**
KNN	# of neighbors: **10** Weights: **uniform** Distance: **Manhattan**	**0.99**	**0.58**	**0.58**
SVM	C: **0.25** Polynomial degree: **1** Kernel: **linear**	**0.71**	**0.65**	**0.62**
Bagging	# of estimators: **30**	**1.00**	**0.66**	**0.64**
RF	Maximum tree depth: **1** Minimum samples per leaf: **10** # of estimators: **10**	**0.80**	**0.80**	**0.79**
Boosting	# of estimators: **30**	**0.89**	**0.88**	**0.83**

Source: author

8. *In Section 3.3.3.2 on page 54, the reported AUCs and hyperparameter specifications of batch (4) models are updated to reflect the removal of the "behavioral_score" variable from the modeling procedure. This is performed by updating the entries of Table 17. Furthermore, the description corresponding to that table is updated in parts that are formatted bold.*

The batch (4) models have the same specifications as batch (1) with the only difference being that undersampling is used in the case of batch (4). All models have test performances between **0.85 and 0.88**. OOT is slightly lower and in the range of **0.82 to 0.86**. Overall, it is concluded that no single model stands out from the rest. What makes batch (4) different from batch (1) is that KNN, SVM **and bagging** perform substantially better. Hence, it can be concluded that class balance is a necessary prerequisite to achieve good performance of these models.

Table 17: optimal hyperparameters of batch (4) models

Model type	Hyperparameter specification	AUC train	AUC test	AUC OOT
LR	**n.a.**	**0.90**	**0.87**	**0.86**
ANN	Activation: **tahn** Hidden layers: **[40, 40, 20]**	**0.90**	**0.88**	**0.86**

Model type	Hyperparameter specification	AUC train	AUC test	AUC OOT
	Learning rate: **constant** Solver: **adam**			
KNN	# of neighbors: **30** Weights: **distance-based** Distance: **Manhattan**	**1.00**	**0.87**	**0.84**
SVM	C: **0.5** Polynomial degree: **1** Kernel: **linear**	**0.90**	**0.87**	**0.85**
Bagging	# of estimators: **150**	**1.00**	**0.86**	**0.85**
RF	Maximum tree depth: **10** Minimum samples per leaf: **10** # of estimators: **150**	**0.94**	**0.88**	**0.86**
Boosting	# of estimators: **30**	**0.90**	**0.85**	**0.82**

Source: author

9. *In Section 3.3.3.2 on pages 54 and 55, the reported AUCs and hyperparameter specifications of the batch (5) models are updated to reflect the removal of the "behavioral_score" variable from the modeling procedure. This is performed by updating the entries of Table 18. The text corresponding to Table 18 is not changed as it still holds.*

Table 18: optimal hyperparameters of batch (5) models

Model type	Hyperparameter specification	AUC train	AUC test	AUC OOT
LR	L1 ratio: **0.1**	**0.93**	**0.87**	**0.84**
ANN	Activation: **logistic** Hidden layers: **[40, 40, 20]** Learning rate: **constant** Solver: **adam**	**0.92**	**0.88**	**0.86**
KNN	# of neighbors: **10** Weights: **uniform** Distance: **Euclidean**	**0.92**	**0.84**	**0.76**
SVM	C: **1** Polynomial degree: **1** Kernel: **rbf**	**0.95**	**0.86**	**0.80**
Bagging	# of estimators: **150**	**1.00**	**0.87**	**0.86**

Model type	Hyperparameter specification	AUC train	AUC test	AUC OOT
RF	Maximum tree depth: **15** Minimum samples per leaf: **10** # of estimators: **50**	**0.95**	**0.87**	**0.87**
Boosting	# of estimators: **150**	**0.95**	**0.87**	**0.82**

Source: author

10. *In Section 3.3.3.2 on pages 55 and 56, the reported AUCs and hyperparameter specifications of the batch (6) models are updated to reflect the removal of the "behavioral_score" variable from the modeling procedure. This is performed by updating the entries of Table 19. Furthermore, the description corresponding to that table is updated in parts that are formatted bold.*

The batch (6) models have the same specifications as the batch (3) with the only difference being that undersampling is used in the case of batch (6). At first, an attempt was made to remove multicollinearity at a 0.5 threshold. However, at 0.5 there still are perfect correlations between various dummy variables. This is driven by the fact that the undersampled dataset is relatively small. In order to remedy this problem, the optimal multicollinearity tolerance level is found to be at 0.2. The LR **[removed part of sentence] and boosting models maintain good performance**. **[removed sentence].** When compared to batch (3), it can be seen that the performance of all the remaining models improved significantly.

Table 19: optimal hyperparameters of batch (6) models

Model type	Hyperparameter specification	AUC train	AUC test	AUC OOT
LR	n.a.	**0.92**	**0.88**	**0.83**
ANN	Activation: **tahn** Hidden layers: **[40, 40, 20]** Learning rate: **constant** Solver: **adam**	**0.92**	**0.88**	**0.84**
KNN	# of neighbors: **10** Weights: **distance-based** Distance: **Euclidean**	**1.00**	**0.82**	**0.78**

Model type	Hyperparameter specification	AUC train	AUC test	AUC OOT
SVM	C: **0.75** Polynomial degree: **2** Kernel: **polynomial**	**0.97**	**0.86**	**0.85**
Bagging	# of estimators: **100**	**1.00**	**0.87**	**0.87**
RF	Maximum tree depth: **15** Minimum samples per leaf: **10** # of estimators: **50**	**0.92**	**0.88**	**0.83**
Boosting	# of estimators: **150**	**0.92**	**0.87**	**0.83**

Source: author

11. *In Section 3.3.4 on page 56, the reported AUCs are updated to reflect the removal of the "behavioral_score" variable from the modeling procedure. This is performed by updating the entries of Table 20.*

Table 20: average AUCs on train, test and OOT samples

Model type	Average AUC train	Average AUC test	Average AUC OOT
LR	**0.91**	**0.88**	**0.83**
ANN	**0.85**	**0.83**	**0.82**
KNN	**0.98**	**0.73**	**0.68**
SVM	**0.75**	**0.70**	**0.66**
Bagging	**1.00**	**0.80**	**0.78**
RF	**0.88**	**0.84**	**0.82**
Boosting	**0.90**	**0.86**	**0.82**

Source: author

12. *In Section 3.3.4 on page 57, the reported AUCs are updated to reflect the removal of the "behavioral_score" variable from the modeling procedure. This is performed by updating the entries of Table 20.*

Table 21: average AUCs on train, test and OOT samples without batch 3 estimates

Model type	Average AUC train	Average AUC test	Average AUC OOT
LR	**0.91**	**0.88**	**0.84**
ANN	**0.90**	**0.88**	**0.84**
KNN	**0.98**	**0.75**	**0.70**
SVM	**0.75**	**0.71**	**0.67**
Bagging	**1.00**	**0.83**	**0.81**
RF	**0.89**	**0.85**	**0.83**
Boosting	**0.90**	**0.86**	**0.82**

Source: author

13. *In Section 3.3.4 on page 56, the reported AUCs are updated to reflect the removal of the "behavioral_score" variable from the modeling procedure. This is performed by updating the entries of Table 20.*

Table 22: best AUCs on train, test and OOT samples

Model type	Batch number	Best* AUC train	Best* AUC test	Best AUC OOT
LR	**5**	**0.93**	**0.87**	**0.84**
ANN	**5**	**0.92**	**0.88**	**0.86**
KNN	**4**	**1.00**	**0.87**	**0.84**
SVM	**4**	**0.90**	**0.87**	**0.85**
Bagging	**6**	**1.00**	**0.87**	**0.87**
RF	**5**	**0.95**	**0.87**	**0.87**
Boosting	**3**	**0.89**	**0.88**	**0.83**

* means that the reported AUC is not necessarily the best, it is the AUC that corresponds to the OOT AUC which is the key to determination of the best model by this metric, source: author