Credit Risk Modeling - ILAB OeNB

By Anastasiia Pryshchepa, Leonid Kuzmishin,
Tristan Leiter, Martin Yago Tortajada

WU WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

ONB OESTERREICHISCHE NATIONALBANK EUROSYSTEM

Motivation and Research Tasks

1. Develop and evaluate supervised credit scoring models for Austrian SMEs using raw balance-sheet data.

2. Does the use of financial ratios improve model performance and generalization compared to the raw-data approach?

3. How do time-series dynamics and lagged variables influence the model's predictive power and feature importance over time?

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

OESTERREICHISCHE NATIONALBANK EUROSYSTEM

Data Preparation

Summary Statistics

Size	Number of observations	Share of observations	Default rate
Small	191 693	78.54%	0.86%
Tiny	52 375	21.46%	1.22%

Table 1: Size Characteristics.

Group Membership	Number of observations	Share of observations	Default rate
0	249 912	99.12%	0.86%
1	2 156	0.88%	0.51%

Table 2: Group Membership Characteristics.

Sector	Number of observations	Share of observations	Default rate
Service	77 764	31.86%	1.04%
Real estate	65 545	26.86%	0.60%
Construction	28 260	11.58%	0.84%
Retail	25 542	10.47%	1.02%
Manufacture	22 838	9.36%	1.01%
Wholesale	19 492	7.99%	0.91%
Energy	4 627	1.90%	0.24%

Table 3: Sector Characteristics

Filtering conditions:

- $f10 \geq 0$
- $(f2 + f3) \leq f1$
- $(f4 + f5) \leq f3$
- $(f6 + f11) \leq f1$

Allowing for rounding error of 2, we eliminated 333 observations, which account for 0.14% of original dataset.

Univariate Analysis Results

Comment .. AUC with the features doesn't make sense?

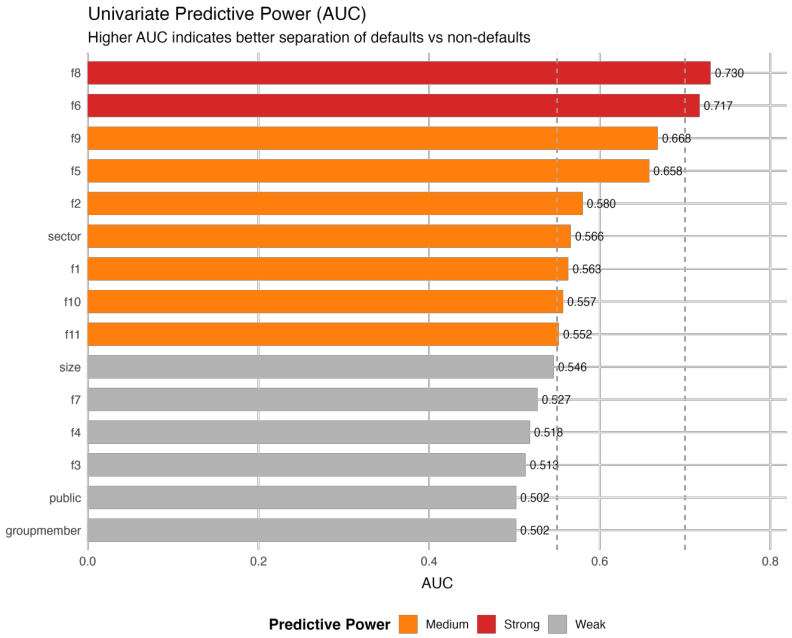


Chart 1: Comparison of univariate predictive power per feature.

- Equity and net profit stand out as the most informative predictors, with AUC score above 0.70, indicating strong stand-alone discrimination power.
- Retained earnings (f9), cash, invested capital, liabilities, total assets, provisions and sector exhibit moderate predictive ability (AUC between 0.55 and 0.70), suggesting useful signal when combined with other variables.
- Firm characteristics such as size, sector, group membership, as well as Retained Earnings (f7), inventories, current assets and public status show AUCs close to 0.50, indicating limited individual explanatory power.
- Overall, the financial positions dominate univariate performance, while structural firm attributes contribute mainly in a multivariate setting.

Data splitting

We enforce group integrity and stratification to eliminate data leakage and preserve information in the dataset.

Validation strategy:
We implemented Stratified Group K-Fold Cross-Validation (k = 5) for hyperparameter tuning.

Preventing Data Leakage:
We enforce a strict separation of entities ($I_{train} \cap I_{test}$ $\emptyset$; $CV_{fold1} \cap CV_{fold2} = \emptyset$)
All firms are assigned exclusively to a single fold to prevent data leakage.

Stratification:
The default-rate (overall and for each sector) is maintained across folds despite the grouping constraints.
Thus, we mitigate variance in the CV-estimates caused by class imbalance.

Data splitting

Analysis: Train set (70%)

- Number of firms: 53 888
- Number of observations: 170 654
- Overall default rate: 0.86%

Sector	Firms	Defaulted firms	Default rate
Service	17 285	556	3.22%
Real estate	15 268	255	1.67%
Construction	5 978	162	2.71%
Retail	5 475	179	3.27%
Manufacture	4 798	159	3.31%
Wholesale	4 177	121	2.90%
Energy	907	8	0.88%

2018	2019	2020	2021	2022
14.25%	19.85%	20.87%	22.16%	22.87%

Table 4: Summary of the training set.

Analysis: Test set (30%)

- Number of firms: 23 084
- Number of observations: 73 414
- Overall default rate: 0.85%

Sector	Firms	Defaulted firms	Default rate
Service	7 406	237	3.20%
Real estate	6 541	108	1.65%
Construction	2 561	69	2.69%
Retail	2 345	76	3.27%
Manufacture	4 798	159	3.24%
Wholesale	1 789	51	2.85%
Energy	388	3	0.77%

2018	2019	2020	2021	2022
14.26%	19.8%	20.9%	22.22%	22.83%

Table 5: Summary of the test set.

Quantile Transformation

Rank-Gauss Transformation via the Probability Integral Transform (PIT).

Motivation:

The balance sheet items and size variables exhibit right skewness and heavy tails, violating the linearity assumption.

Methodology:

We apply the PIT to map features to a standard normal distribution.

Transformation: $x_{final} = \Phi^{-1}(ECDF_{train}(x_{input}))$

Frozen Reference Approach:

The empirical CDF is fitted exclusively on the training set.

The test set observations are mapped onto this fixed training ECDF („walking forward approach“).

Hyperparameter Tuning

Superior Convergence via Bayesian Optimization with Gaussian Processes.

Baseline Strategies – Discrete Grid

Deterministic, but computationally expensive for high-dimension fine-tuning of the hyperparameters.

Baseline Strategies – Random search

By using stochastic uniform sampling the algorithm is more efficient, but it lacks memory of the past evaluations.

Bayesian Optimization:

Prior: Gaussian Process with a Matern 5/2 kernel to model the non-smooth objective landscape of tree-based models

We relied on the Upper Confidence Bound (UCB) with a high confidence parameter ($\kappa = 2.576$) to guide the search. Expected Improvement (EI) can theoretically result in a higher performance (it targets the max. marginal gain), but we observed that it induces instability.

Bayesian optimization results in the highest CV-AUC with the lowest no. of iterations.

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

OESTERREICHISCHE NATIONALBANK EUROSYSTEM

Model Development

Regularized GLMs (Elastic Net)

All hyperparameter tuning methods result in a very similar AUC-Score (CV).

Hyperparameter Tuning:

- All three hyperparameter tuning methods yield very similar results.
- We select Bayesian Optimization as the preferred model.

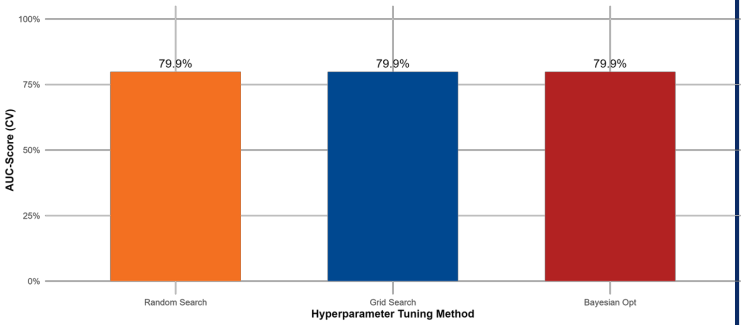


Chart 2: Comparison of hyperparameter tuning methods for GLM by CV-AUC.

Learning Curve:

- First, a rapid increase in the AUC-Score can be observed as the penalty strength gets relaxed.
- The absence of a performance drop-off on the RHS indicates that the model is robust and likely not overfitted.

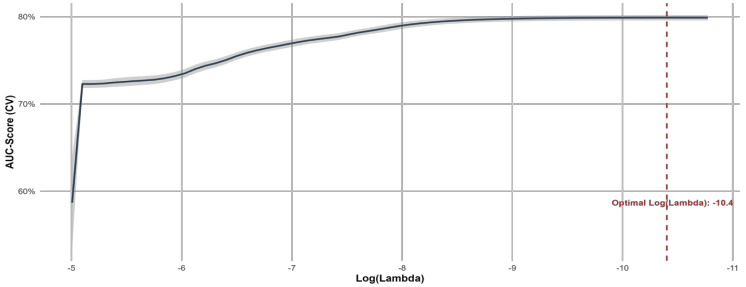


Chart 3: Learning curve of the Bayesian optimization hyperparameter tuning model (CV-AUC).

Regularized GLMs (Elastic Net)

All hyperparameter tuning methods result in a very similar AUC-Score (CV).

Hyperparameter Tuning:

- All three hyperparameter tuning methods yield very similar results.
- We select Bayesian Optimization as the preferred model.

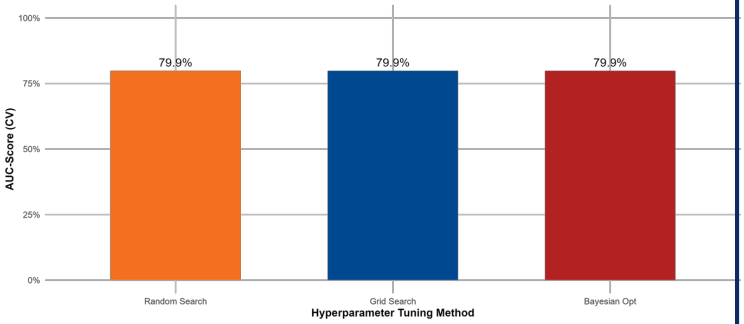


Chart 2: Comparison of hyperparameter tuning methods for GLM by CV-AUC.

Optimal model:

- All three methods imply a (very) low penalty strength, which results in a model close to a full Ridge regularization implementation.

Method	Mixing Param (α)	Penalty Strength (λ)	CV AUC (Validation)
Grid Search	1.00	3.03×10^{-5}	79.9%
Random Search	1.00	3.03×10^{-5}	79.9%
Bayesian Opt	**1.00**	$\mathbf{3.04 \times 10^{-5}}$	**79.9%**

α: Elastic Net mixing parameter (0 = Ridge, 1 = Lasso). Controls variable selection;
λ: Regularization penalty strength. Controls the magnitude of coefficient shrinkage.

Table 6: Hyperparameters and specifications of the different methods for the GLM.

Regularized GLMs (Elastic Net)

The elastic net implementation of the baseline GLM model results in a strong performance in the test set.

Generalization Performance:

- Achieved a final Test AUC of 81.8%, confirming strong predictive power on unseen entities.
- The generalization gap (Train 79.9% vs. Test 81.8%) implies that the model generalizes well to “new” datapoints, implying a robust model.

Model Selection Strategy:

- We compare the "Best Fit" model against the one following the 1-Standard Error (1-SE) rule.
- Decision: We go for the sparser 1-SE model as the Test-AUC is very similar. In this case, the weaker penalty parameter leads to the variable “f10” being dropped.

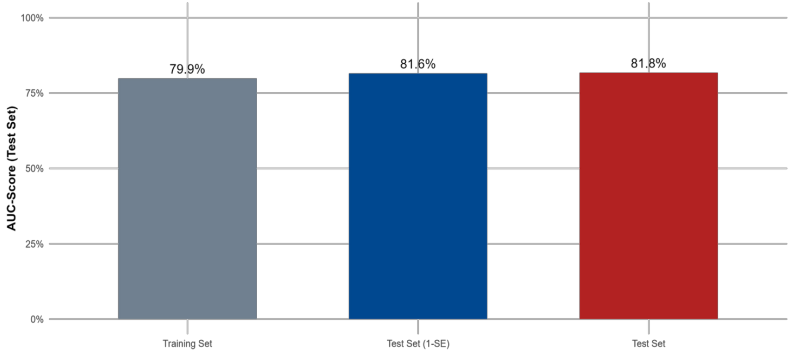


Chart 5: Comparison of Test-AUC with the Test-AUC using the 1-SE rule and the CV-AUC.

Metric	Best-fit (Lambda Min)	1-SE (1-SE Rule)
Test AUC	**81.8%**	81.6%
CV AUC	79.9%	–
N Features	20	19
Optimal Hyperparameters		
Mixing Param (α)		1.00 (Lasso)
Penalty (λ)	3.04×10^{-5}	8.45×10^{-4}

Note: The 1-SE Rule removed one additional variable to improve parsimony with negligible performance loss.

Table 7: Model outputs of the GLM model trained via Bayesian Optimization compared to 1-SE.

Regularized GLMs (Elastic Net)

Even though the Test-AUC is high (81.6%), the model severely underestimates the highest risk deciles.

show for different factors and categories

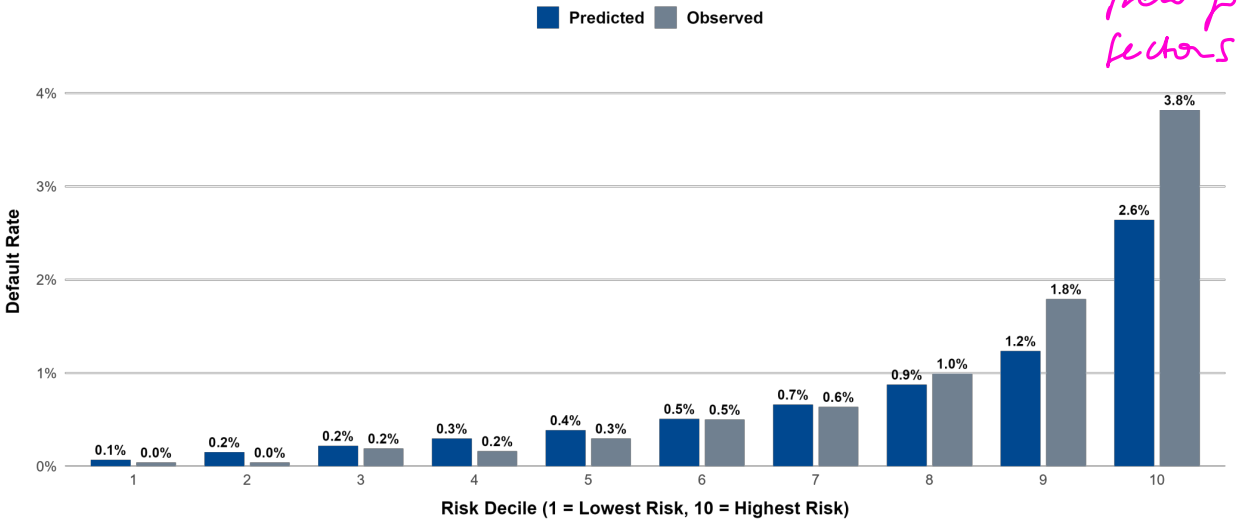


Chart 6: Calibration chart of the optimal GLM-model (1-SE).

The linear GLM-model is too rigid for the highest-risk deciles without interactions (tail compression).

Random Forest

Bayesian optimization leads to the highest CV-AUC. The optimal hyperparameters are used to train the Random Forest model.

Hyperparameter Tuning:

- Bayesian Optimization resulted in an AUC-Score (CV) of 80.8%
- Similarly, random search achieved an AUC-Score of 80.6%.

Thus, both methods yield a similar score.

➢ We select the optimal model found via Bayesian Optimization as the preferred model.

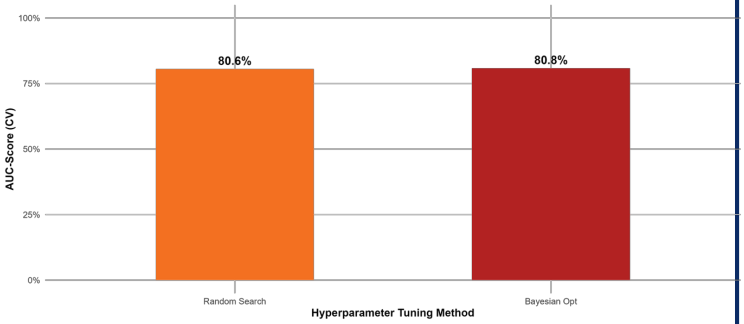


Chart 7: Comparison of hyperparameter tuning methods for Random Forest by CV-AUC.

Learning Curve:

- First, a rapid increase in the AUC-Score can be observed as the penalty strength gets relaxed.
- The absence of a performance drop-off on the RHS indicates that the model is robust and likely not overfitted.

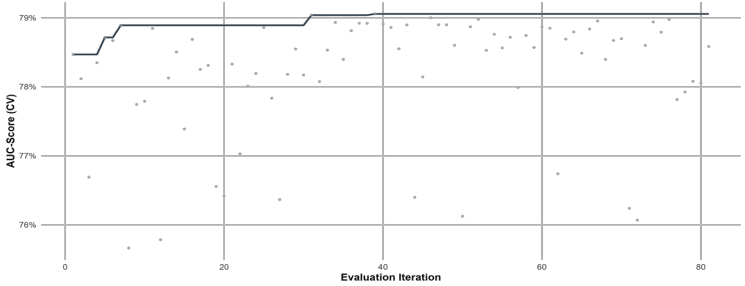


Chart 8: Learning curve of the Bayesian optimization hyperparameter tuning model (CV-AUC).

Random Forest

Bayesian optimization leads to the highest CV-AUC. The optimal hyperparameters are used to train the Random Forest model.

Hyperparameter Tuning:

- Bayesian Optimization resulted in an AUC-Score (CV) of 80.8%
- Similarly, random search achieved an AUC-Score of 80.6%.

Thus, both methods imply a similar Score.

➢ We select the optimal model found via Bayesian Optimization as the preferred model.

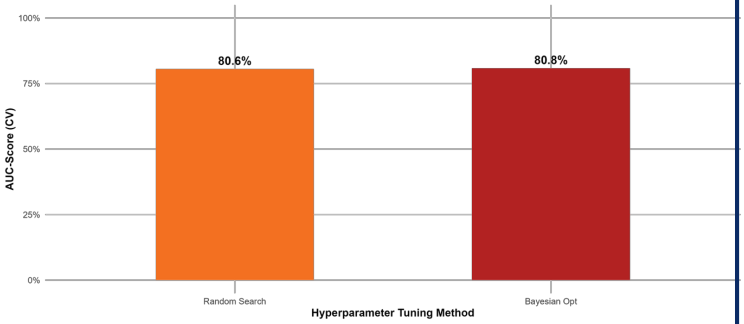


Chart 9: Comparison of hyperparameter tuning methods for Random Forest by CV-AUC.

Optimal model:

- The Optimizer selected ExtraTrees (Randomized Thresholds) over Gini splits, suggesting a need to decorrelate trees to handle feature noise.
- Sampling without replacement (Fraction: 0.87) was preferred, reducing the bias typically introduced by bootstrapping in imbalanced datasets.

Method	Vars/Split (`mtry`)	Min Node (Size)	Sample Frac (Fraction)	Split Rule (Type)	CV AUC (Validation)
Random Search	4	37	0.39	ExtraTrees	78.9%
Bayesian Opt	**3**	**48**	**0.87**	**ExtraTrees**	**79.1%**

`mtry`: Number of variables candidates at each split;
Min Node: Minimum size of terminal nodes (controls depth/complexity);
ExtraTrees: Extremely Randomized Trees (randomly selected thresholds).

Table 8: Hyperparameters and specifications of the two tuning methods.

Random Forest

Robust Generalization and Monotonic Risk Ranking on Out-of-Sample Data.

Generalization Performance:

- Achieved a final Test AUC of 81%, confirming strong predictive power on unseen entities.
- The generalization gap (80.8% vs. Test 81%) is also (very) low. Similar to the regularized GLM, the Random Forest model generalizes rather well to "new" datapoints.

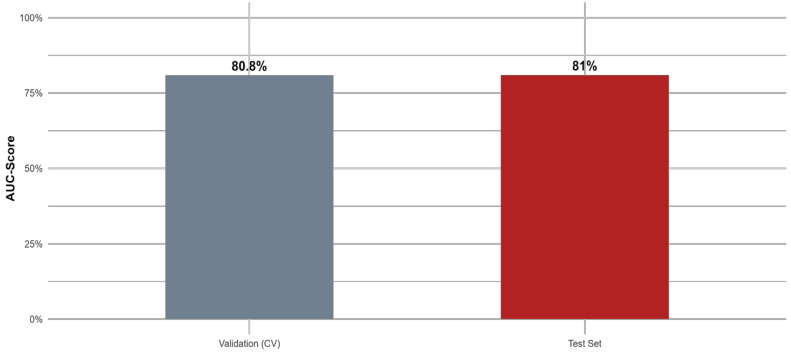


Chart 10: Comparison of Test-AUC and the CV-AUC (Bayesian Optimization).

Metric	Best-fit (Bayesian Opt)
Test AUC	**81.0%**
CV AUC	79.1%
N Features	21
Optimal Hyperparameters	
Split Rule	ExtraTrees
Vars per Split (`mtry`)	3
Min Node Size	48
Sample Fraction	0.87 (No Replace)

Table 9: Hyperparameters and specifications of the final Random Forest model.

Random Forest

The RF-model underestimates the average default-risk in the two highest risk-deciles.

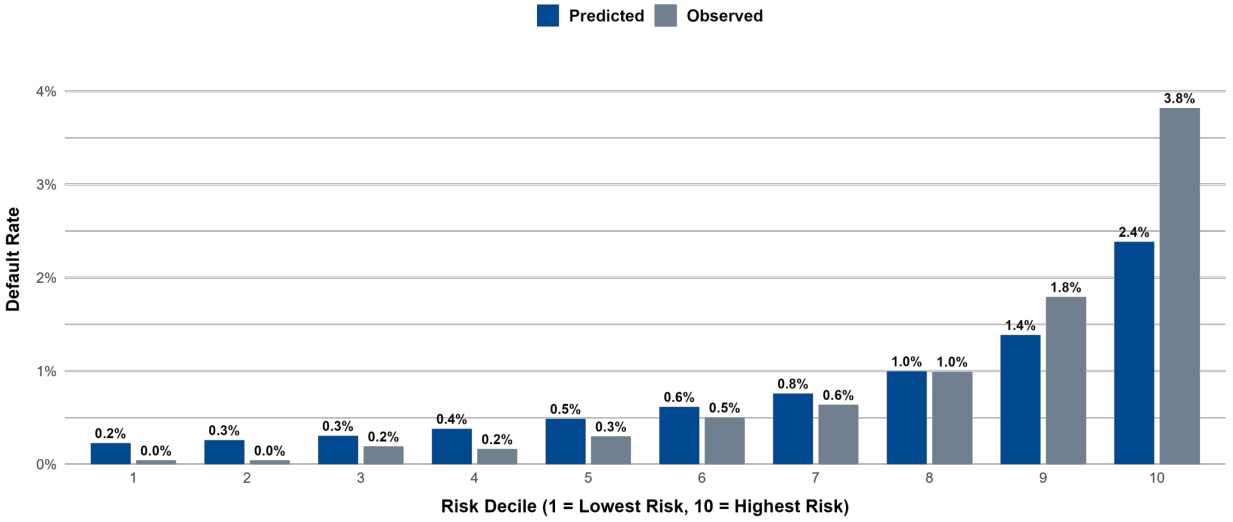


Chart 11: Calibration chart of the optimal RF-model.

The RF-model is conservative in the lower risk deciles (1-8),
but underestimates the average default-risk in the two highest deciles (9-10).

XGBoost

Bayesian optimization leads to the highest CV-AUC. The optimal hyperparameters are used to train the XGBoost-model.

Hyperparameter Tuning:

- Bayesian Optimization: Achieved the highest CV-AUC by utilizing a Gaussian Process prior to model the objective function.
- Random Search: Strong baseline (80.7%) but less efficient at fine-tuning specific interactions.
- Grid Search: Lowest performance (80.6%).

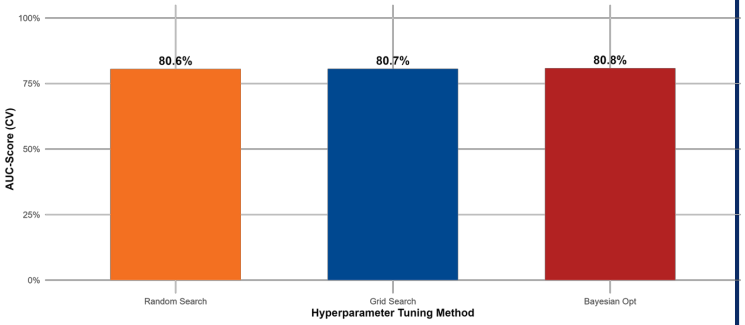


Chart 12: Comparison of hyperparameter tuning methods for XGBoost by CV-AUC.

Learning Curve:

- Convergence: The learning curve plateaus optimally, indicating the model converges without overfitting.
- Robustness: The narrow standard deviation ribbon (grey area) confirms consistent performance across the stratified firm-level folds.

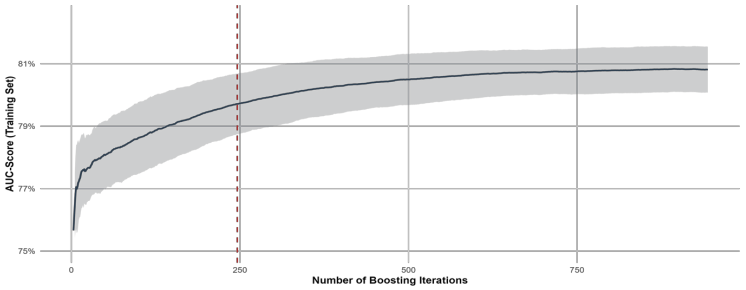


Chart 13: Learning curve of the Bayesian optimization hyperparameter tuning model (CV-AUC).

XGBoost

Bayesian optimization leads to the highest CV-AUC. The optimal hyperparameters are used to train the XGBoost-model.

Hyperparameter Tuning:

- Bayesian Optimization: Achieved the highest CV-AUC by utilizing a Gaussian Process prior to model the objective function.
- Random Search: Strong baseline (80.7%) but less efficient at fine-tuning specific interactions.
- Grid Search: Lowest performance (80.6%).

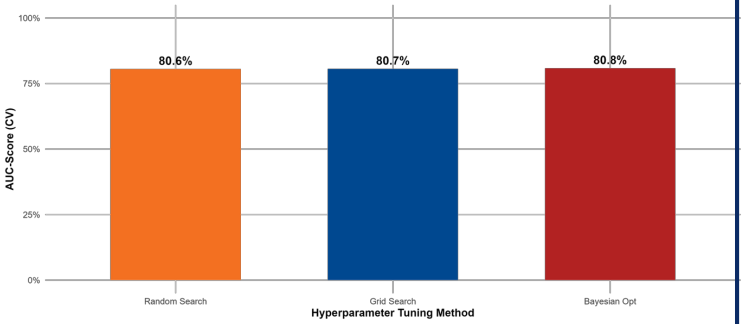


Chart 14: Comparison of hyperparameter tuning methods for XGBoost by CV-AUC.

Optimal model:

- Bayesian Optimization leads to a slightly lower tree depth, resulting in less complex interactions.
- Additionally, the fraction of datapoints used per tree (Row Sample) is much lower.
- The bayesian optimizer discovered that aggressive bagging (50% of the data per tree) is superior to complex feature interactions.

Method	Learn Rate (η)	Tree Depth (J)	Row Sample (Subsample)	Feat Sample (Colsample)	CV AUC (Validation)
Grid Search	0.010	5	0.70	0.90	80.7%
Random Search	0.030	5	0.92	0.78	80.6%
Bayesian Opt	**0.010**	**4**	**0.50**	**0.76**	**80.8%**

Note: CV AUC represents the mean performance across 5 stratified validation folds.
η: Learning Rate; J: Interaction Depth;
Subsample: Fraction of training instances (rows) used per tree;
Colsample: Fraction of features (columns) used per tree.

Table 10: Hyperparameters and specifications of the different methods.

XGBoost

Robust Generalization and Monotonic Risk Ranking on Out-of-Sample Data.

Generalization Performance:

- Achieved a final Test AUC of 83.6%, confirming strong predictive power on unseen entities.
- The generalization gap (Train 80.8% vs. Test 83.6%) implies that the model generalizes well to “new” datapoints, implying a robust model.

Model Selection Strategy:

- We compared the "Best Fit" model against the 1-Standard Error (1-SE) rule.
- Decision: Retain the Standard Bayesian Model (83.6%) as it outperformed the sparser 1-SE alternative (82.8%).

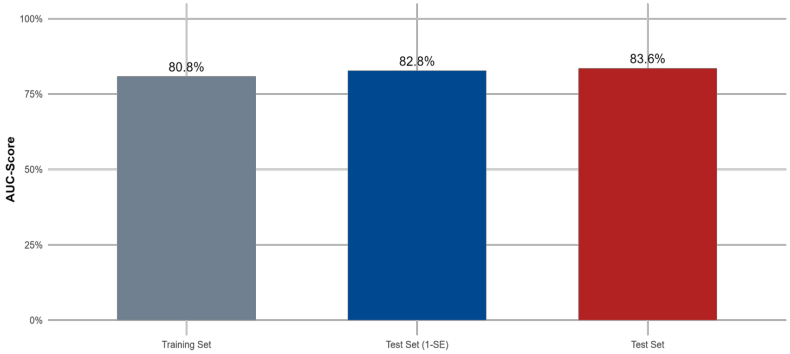


Chart 15: Comparison of Test-AUC with the Test-AUC using the 1-SE rule and the CV-AUC.

Metric	Best-fit (Max AUC)	1-SE (1-SE Rule)
Test AUC	**83.6%**	82.8%
CV AUC	80.8%	–
N Features	21	21
Optimal Hyperparameters		
Learning Rate (η)	0.010	
Tree Depth (J)	4	
Row Sampling	0.50	
Feat. Sampling	0.76	

Note: The 1-SE model uses fewer boosting iterations to reduce overfitting variance.

Table 11: Hyperparameters and specifications of the different methods.

XGBoost

The calibration chart shows that the model is a good fit for tail calibration and strong rank ordering.

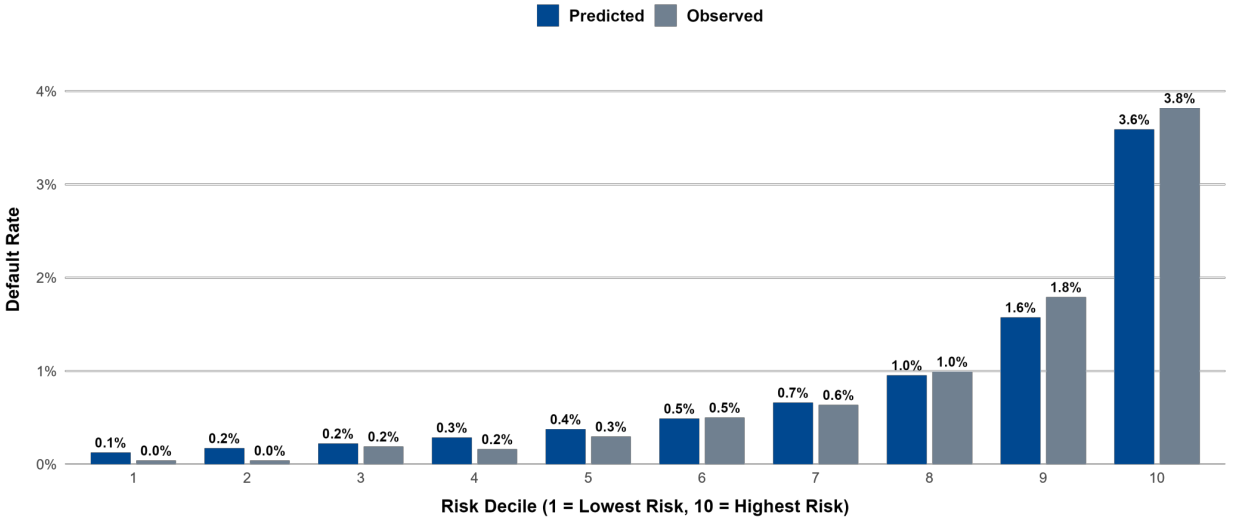


Chart 16: Calibration chart of the optimal XGBoost-model.

The predicted probabilities closely track the observed default-rates.

Neural Networks

Severe class imbalance constrains model complexity.

Set up:

- Only 11 balance sheet items are used to maintain a high signal-to-noise ratio
- Rare default events (~1%) limit feasible network depth
- Neural network ensemble (PyTorch + FastAI) improves robustness under imbalance

Hyperparameter choices :

- Learning rate: 0.001.
- Batch size: 512.
- Dropout rate: 0.1.
- Model complexity is controlled via early stopping,

Performance metrics:

- AUC - 0.802
- Recall – 77.11%
- Precision – 2.16%

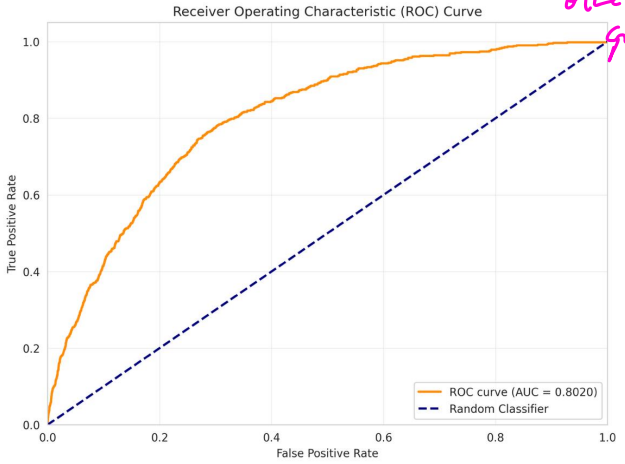


Chart 17: ROC curve out-of-sample neural network.

NN hyperparameters were fixed to stable default values and model complexity was controlled via early stopping, as full hyperparameter tuning would be unreliable under extreme class imbalance.

Neural Networks

Threshold Analysis

- The standard 0.5 classification threshold is inappropriate under strong class imbalance.
- For the out-of-sample (stratified) test set, the optimal threshold is 0.0086, selected to maximize the Matthews Correlation Coefficient (MCC) subject to a regulatory constraint.
- We impose an upper bound on the False Negative Rate as an illustrative and conservative risk tolerance.
- This threshold satisfies the ≤ 28% False Negative Rate (FNR) requirement, ensuring that at least 72% of defaults are correctly identified.
- Imposing the FNR constraint does not materially reduce model performance, indicating that the OOS results are stable and robust.

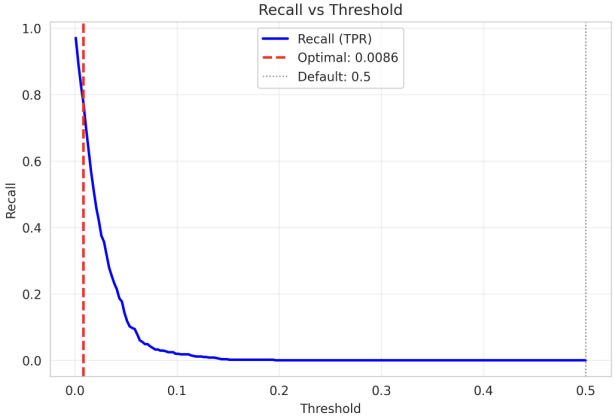


Chart 18: Comparison of Recall and Threshold.

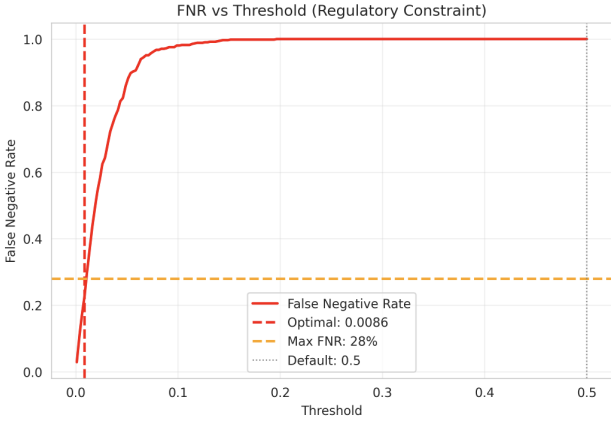


Chart 19: Comparison of False Negative Rate and Regulatory Constraint.

Neural Networks

Threshold Analysis

- Lower thresholds increase recall but reduce precision, leading to more false alarms.
- The selected threshold represents a balanced operating point, capturing most defaults without overly aggressive classification*.
- Very low thresholds generate a sharp increase in false positives, while higher thresholds suppress default detection.
- The commonly used 0.5 threshold is inappropriate in this setting, as it would classify almost all observations as non-default.

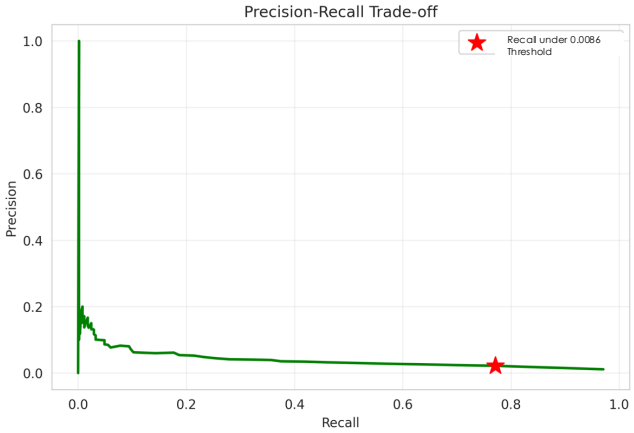


Chart 20: Precision-Recall Trade-off.

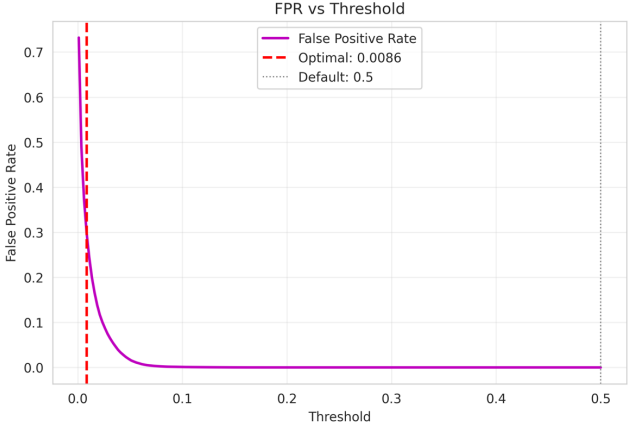


Chart 21: Comparison of False Negative Rate and Threshold.

*Aggressive classification refers to using very low decision thresholds that maximize default detection but generate many false alarms.

Neural Networks

Non-default firms:
- 51,171 correctly classified (true negatives 70%)
- 21,516 incorrectly classified as default (false positives 29%)

Default firms:
- 475 correctly identified defaults (true positives 0.6%)
- 141 defaults missed (false negatives 0.2%)

- False positive rate – 29.6%
- False negative rate – 22.9%
- Recall – 77.1%
- Precision – 2.16%

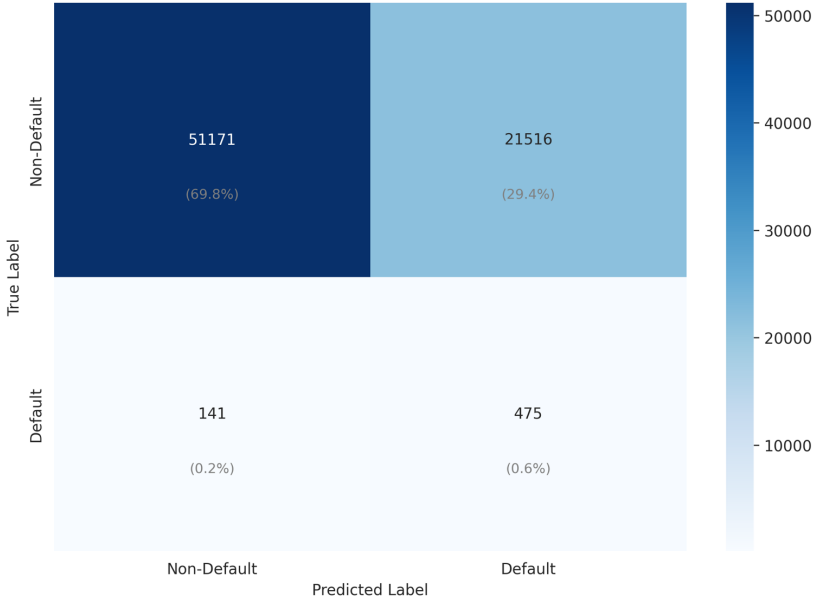


Chart 22: Confusion matrix.

Model selection

Sacrifice AUC to get the compression in the tail, we love XGBOOST.

Model selection

Performance in the test set:

- XGBoost achieved the highest AUC-Score in the test set (83.58%), followed by GLM (1-SE) and Random Forest.
- The hyperparameter tuning for XGBoost prevented the model from overfitting whilst extracting a strong signal from the information.

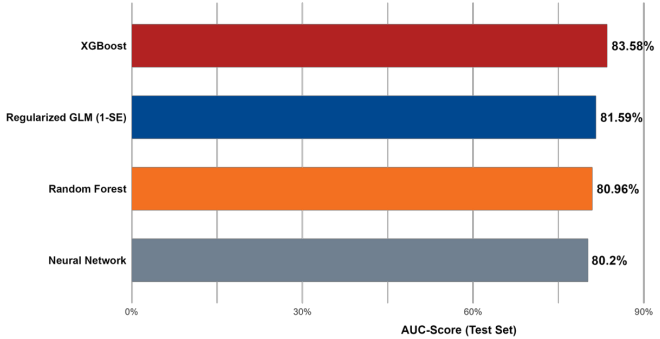


Chart 23: Comparison of Test-AUC for the different models.

XGBoost results in the highest AUC-Score (test set) and the lowest Brier-Score, implying that the model is reliable and separates defaulters and non-defaulters well.

Model	Test AUC	Brier Score	Rank
XGBoost	**83.6%**	**0.00823**	**1**
Regularized GLM (1-SE)	81.6%	0.00828	2
Random Forest	81.0%	0.00829	3
Neural Network	80.2%	0.00834	4

Table 13: Test-AUC and Brier-Score of the different models.

Metric	Best-fit (Max AUC)	1-SE (1-SE Rule)
Test AUC	**83.6%**	82.8%
Brier Score	**0.0082**	–
CV AUC	80.8%	–
N Features	21	21
Optimal Hyperparameters		
Learning Rate (η)	0.010	
Tree Depth (J)	4	
Row Sampling	0.50	
Feat. Sampling	0.76	

Table 14: Hyperparameters of the preferred XGBoost-model.

We select XGBoost as the preferred model.

Outlook

+ Add visual representation ...
Pdep path of the BASE LEARNERS,
to understand how the shrinkage
actually looks like

Other models

AdaBoost

- Estimated only on 10 000 observations using random sampling.
- Due to high computational costs, we were unable to estimate the model on the full sample.
- Hyperparameter tuning applied: grid search, random search, Bayesian optimization.

Key results	
Best method	Grid search
Test AUC	0.74
Train AUC	0.81

Table 15: Summary of ADABoost results.

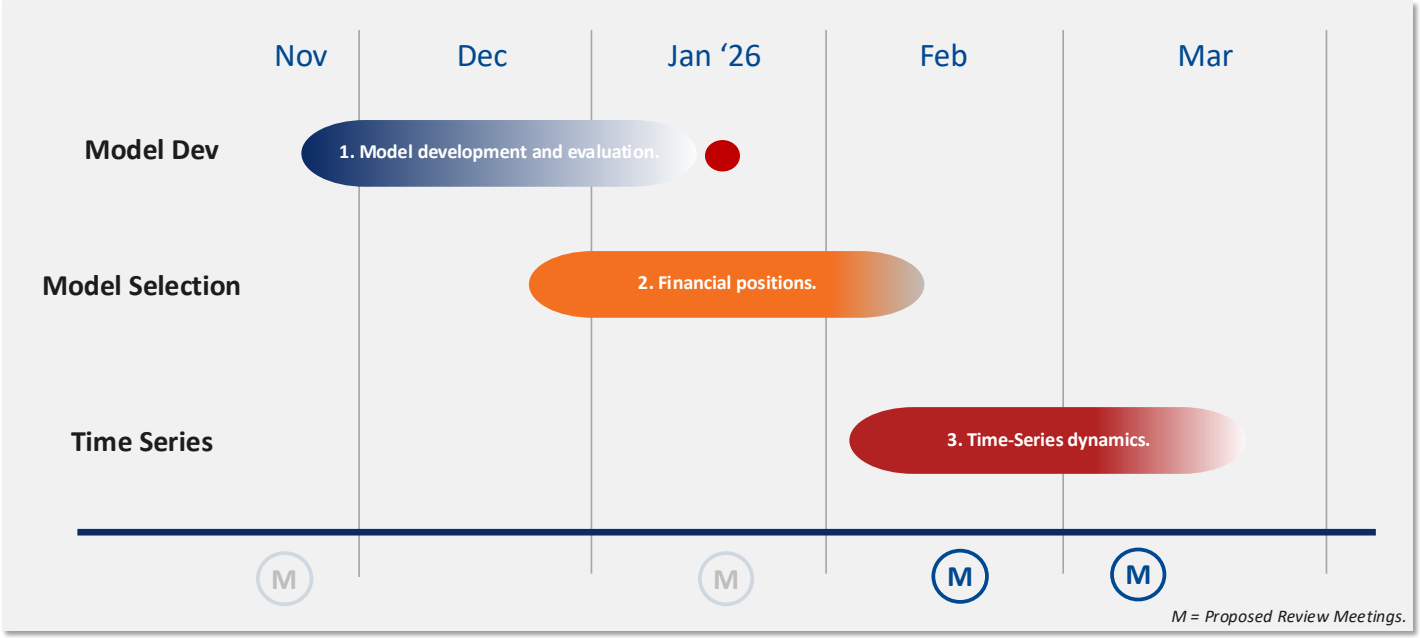
Neural Network out-of-time sampling

- Train years: 2018 -2021.
- Test years: 2022.
- 5 CV folds.
- The optimal threshold is 0.0088, selected to maximize the Matthews Correlation Coefficient (MCC) subject to a regulatory constraint.

Key results	
AUC	0.8142
Recall	77.44%
Precision	2.75%

Table 16: Summary of Neural Network out-of-time results.

Model selection



Richter: wants more info about the models, which features for which models..
Kornik: show partial dependence plots ..
show bivariate interaction.
MARGINAL RESPONSES

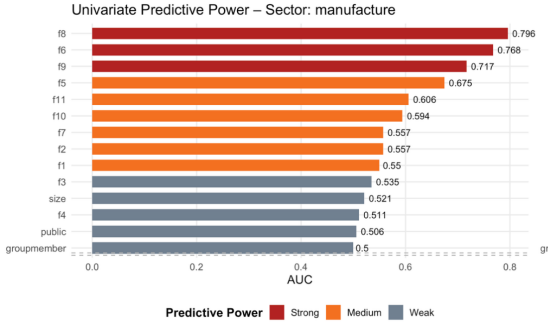
THANK YOU FOR YOUR ATTENTION!

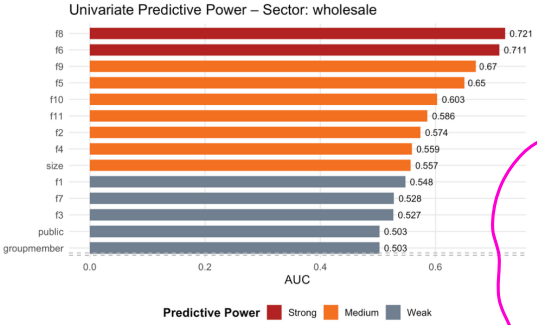
Kornik: RANK transformation ..

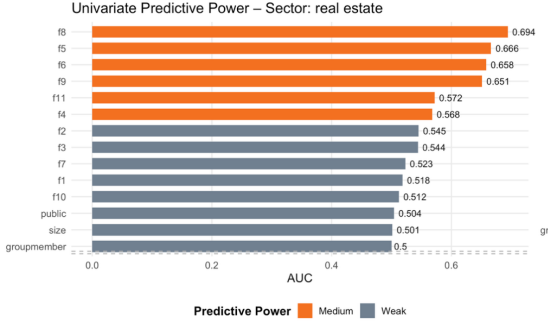
One feature at a time for the PART. DEP. PLOTS

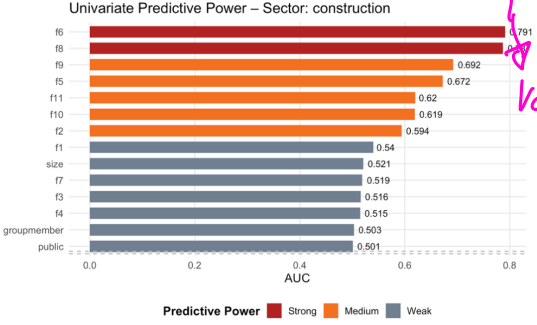
What does XGBoost see? Find specific exemplary balance sheets to see clearly what the XGBoost catches ..

Appendix 1 – Univariate Analysis Results by sector





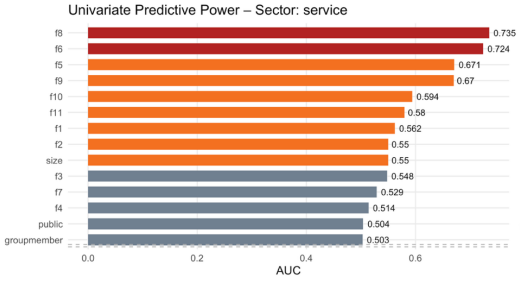


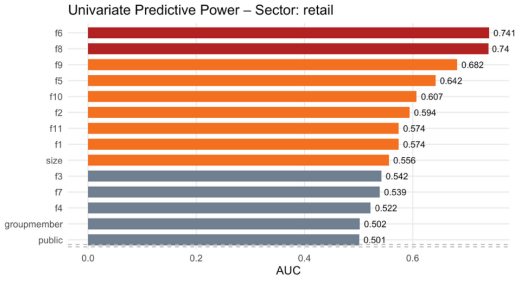


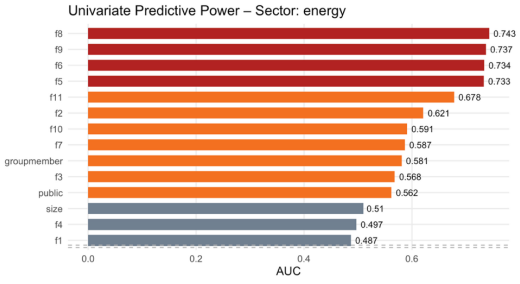
H2O package for variational autoencoders ...

THERE WAS ONE PAST GOOD result: Var. Autoencoders + GLM.

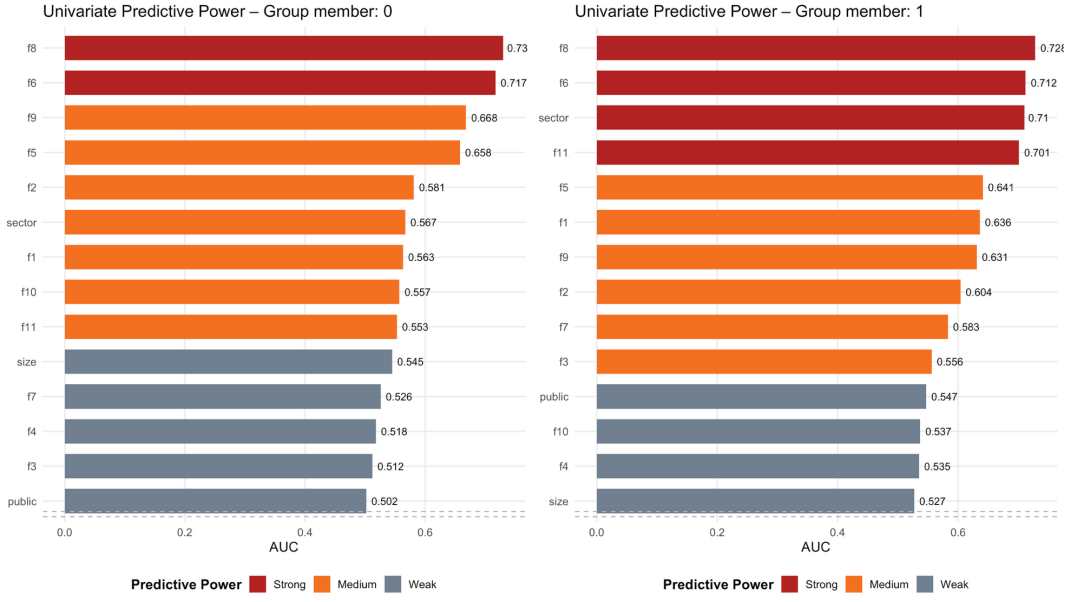
Appendix 1 – Univariate Analysis Results by sector



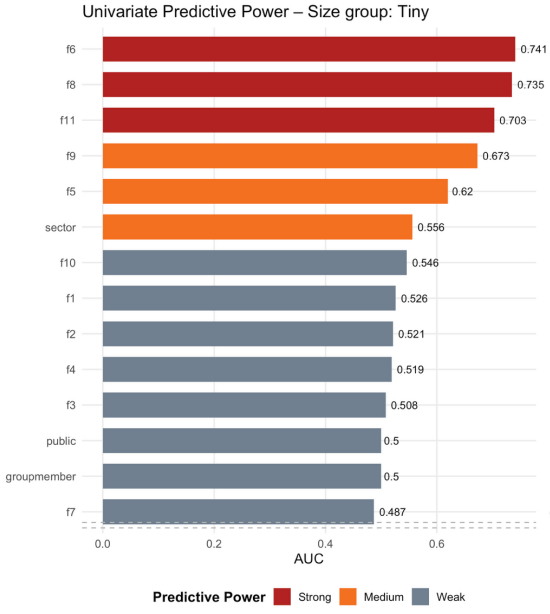


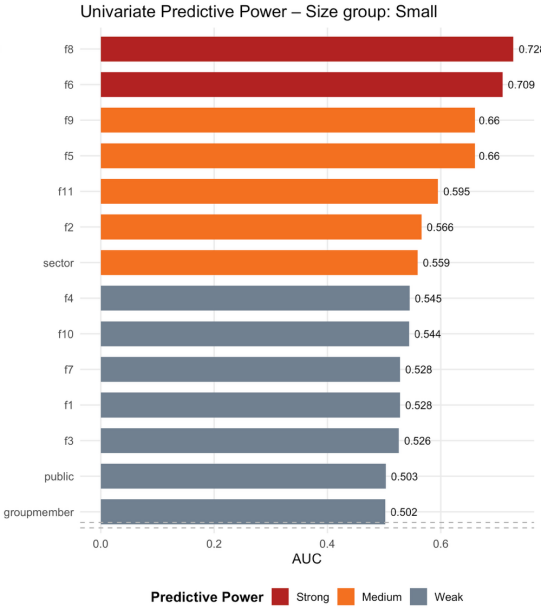


Appendix 2 - Univariate Analysis Results by group member



Appendix 3 - Univariate Analysis Results by size





Appendix 3 - Matthews Correlation Coefficient (MCC)

- MCC is a balanced measure of classification performance based on all outcomes in the confusion matrix (true positives, true negatives, false positives, false negatives).
- Matthews, B. W. (1975) introduced the Matthews Correlation Coefficient (MCC) as a correlation-based performance measure for binary classification.
- Chicco, D., & Jurman, G. (2020). demonstrate that MCC is superior to accuracy and F1 in imbalanced classification problems and recommend MCC for such settings.
- MCC evaluates how well predicted and actual class labels agree overall, rather than favoring the majority class.
- Values range from −1 (complete misclassification) to +1 (perfect classification), with 0 indicating random performance.
- Maximizing MCC yields a well-balanced decision threshold that effectively captures defaults while controlling false alarms.

- $$\mathrm{MCC} = \frac{\mathrm{TN}\times\mathrm{TP} - \mathrm{FN}\times\mathrm{FP}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}$$

References:
1. Matthews, B. W. (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta (BBA) – Protein Structure, 405(2), 442–451.
2. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics, 21(1), Article 6.

Appendix 4 – Decile-based Calibration Plot

- Start with the GLM-model (1-SE) predictions for the probability of default for every observation.

1. Sorting: The algorithm sorts all observations in the dataset based on their predicted probability (lowest to highest).
2. Partitioning: The sorted data is split into 10 equal-sized deciles.
3. For each decile, the algorithm computes the average predicted probability in that bin and the average of the actual observed default-rate (= default rate of this bin).
4. Visualize the bins.

Rank (Risk Order)	Actual (Default)	GLM (P_{def})	Random Forest (P_{def})	XGBoost (P_{def})
1	0	0.323	0.055	0.243
2	0	0.267	0.037	0.374
3	0	0.217	0.039	0.088
4	0	0.198	0.059	0.165
5	0	0.193	0.051	0.125
6	0	0.193	0.040	0.187
7	0	0.192	0.057	0.302
8	**1**	**0.163**	**0.053**	**0.182**
9	0	0.159	0.056	0.319
10	0	0.157	0.056	0.155
11	0	0.154	0.062	0.187
12	0	0.145	0.047	0.277
13	0	0.144	0.093	0.228
14	**1**	**0.142**	**0.038**	**0.070**
15	0	0.140	0.027	0.156

Note: Defaults (Rows 8, 14) are captured despite low absolute probabilities.
Observation: RF probabilities are consistently lower than GLM/XGB.

Table 17: Summary of observed versus predicted probabilities.

Appendix 5 – Data Splitting

1. Aggregate to firm level: Group firms by id.
2. Stratification Key: Create one key for each unique firm by interaction the strat. Variables (e.g., Sector and Target).
3. Creates composite levels (Energy.1, Retail.0) to balance the distribution of the default-rate across sectors.
4. Partition the unique ids: using “createDataPartition” from the “caret” package, we split the unique firm ids based on the prior key.

Appendix 6 – Feature description

Variable	Description (DE)	Description (EN)
f1	Aktiva	Total assets
f2	Anlagevermögen	Invested capital
f3	Umlaufvermögen	Current assets
f4	Vorräte	Inventories
f5	Kassenbestand	Cash
f6	Eigenkapital	Equity
f7	Gewinnrücklage	Retained earnings
f8	Bilanzgewinn	Net profit
f9	Gewinnvortrag	Retained earnings
f10	Rückstellungen	Provisions
f11	Verbindlichkeiten	Liabilities

Table 18: Feature descriptions.