Vienna University of Economics and Business

OeNB ILAB

OeNB ILAB: Methodology Summary

Author:
Tristan Leiter

Submission Date:
December 19, 2025

Contents

Chapter 1

Data Processing

1.1 Preprocessing and data splitting

1.1.1 Preprocessing

Based on the instructions from the client, we filter the balance sheet data based on the following constraints to remove noisy observations. To account for floating-point inconsistencies, we allow for a rounding error of 2 decimal places.

1. $f_{10} \geq 0$
2. $(f_2 + f_3) \leq f_1$
3. $(f_4 + f_5) \leq f_3$
4. $(f_6 + f_{11}) \leq f_1$

1.1.2 Data splitting

We utilize a custom function, `MVstratifiedsampling`, to perform a stratified split at the firm level rather than the observation level. This ensures that all records for a specific firm ID reside in the same partition.

1. **Aggregate to Firm Level**
   The dataset is first grouped by unique identifier (`id`). We summarize the data to create a single profile per firm, extracting the maximum default status (y) and the business sector.
2. **Create Stratification Key**
   We generate a combined key for each unique firm by interacting the stratification variables (e.g., `Sector` and `Target`).
   - This creates composite levels (e.g., `"Energy.1"`, `"Retail.0"`) used to balance the distribution of the target variable across sectors.
3. **Partition Unique IDs**
   Using the `createDataPartition` algorithm from the *caret* package, we split the unique firm IDs based on the stratification key.
   - The default split creates a training set containing 70% of the firms and a test set containing the remaining 30%.
4. **Retrieve Observation Data**
   Finally, we filter the original dataset to reconstruct the Training and Test sets based on the selected lists of IDs. This preserves the original data structure without requiring column removal.

1.2 Feature engineering

1.2.1 Standardization (Size Normalization)

Financial data often exhibits a "size effect," where absolute magnitude (e.g., total sales or debt in Euros) overshadows financial performance. To ensure comparability between large and small firms, we standardize absolute values by scaling them against a measure of firm size, typically **Total Assets**.

This step converts raw financial figures into structural ratios (e.g., EBIT $\rightarrow$ ROA), isolating efficiency from magnitude.

1.2.2 Quantile Transformation (Probability Integral Transform)

After size normalization, financial ratios typically remain highly skewed with heavy tails (non-Gaussian). We apply a **Quantile Transformation** using the Probability Integral Transform (PIT).

To preserve macro-economic signals (e.g., a global downturn increasing leverage ratios across the board), we utilize a **Frozen Reference Approach**:

1. **Training Phase:** We pool all years of the training set to establish a "Through-the-Cycle" cumulative distribution function (CDF).
2. **Testing Phase (Walking Forward):** We map future observations onto this fixed training CDF. This ensures that if the economy deteriorates (shifting the distribution right), the transformed Z-scores reflect this increased risk, rather than normalizing it away.

The transformation chain for a variable x is:

$$x_{ratio} = \frac{x_{raw}}{\text{Total Assets}} \quad \xrightarrow{ECDF_{train}} \quad u \in [0,1] \quad \xrightarrow{\Phi^{-1}} \quad z \sim N(0,1)$$

Chapter 2

Modelling

2.1 Model Selection

Selection is done by the AuC-ROC measure as per client instructions.

2.2 Hyperparameter Tuning

2.2.1 Discrete Grid Search

2.2.2 Random Grid Search

2.2.3 Bayesian Optimization

2.3 GLMs

2.3.1 Logistic Regression

2.3.2 Regularized GLMs

2.4 Decision Trees

2.4.1 Random Forest

2.4.2 AdaBoost

2.4.3 XGBoost

2.4.4 CatBoost

2.5 Neural Networks

Chapter 3

Model Assessment

3.1 Final Model

3.2 Model Evaluation

Chapter 4

Task Distribution

The following matrix outlines the distribution of project responsibilities among the four team members. Primary ownership is denoted by an **X**.

Table 4.1: Team Task Distribution Matrix

Task	**Tristan**	**Nastia**	**Leonid**	**Martin**
Data Preprocessing	**X**	**X**		
Feature Engineering (Standardization & PIT)	**X**			
Stratified Sampling within Train Set				**X**
GLMs and regularized GLMs				
Random Forest and Boosting	**X**	**X**		
Neural Networks				