CAPSTONE PUBLIC PROJECT SUMMARY

CHALLANGER CORPORATE EARLY WARNING SIGNAL MODEL

By
Greta Zsikla

Submitted to Central European University - Private University
Department of Economics/ Central European University /Business Analytics MSc

In partial fulfilment of the requirements for the degree of Master of Business Analytics

Supervisor: Eduardo Arino de la Rubia

Vienna, Austria
2025

Copyright Notice

[1]

Author's declaration

I, the undersigned, **Greta Zsikla**, candidate for the MSc degree in Business Analytics declare herewith that the present thesis titled "Challenger corporate early warning signal model" is exclusively my own work, based on my research and only such external information as properly credited in notes and bibliography.

I declare that no unidentified and illegitimate use was made of the work of others, and no part of the thesis infringes on any person's or institution's copyright.

I also declare that no part of the thesis has been submitted in this form to any other institution of higher education for an academic degree.

I also declare that no part of the thesis has been submitted in this form to any other institution of higher education for an academic degree.

Vienna, 02 June 2025

Greta Zsikla

Table of contents

Introduction

This capstone project was completed in partnership with a major commercial bank, with the goal of improving the institution's ability to detect early signs of financial distress among its corporate clients. The bank already had an internal Early Warning Signal (EWS) model in place, designed to flag companies at risk of defaulting on their loans. However, there were several areas where the existing model could be improved, particularly in terms of the data it used, the way it handled missing information, and the techniques used for selecting features.

In simple terms, the objective was to build a new version of the model, a so-called "challenger model" that could be compared against the bank's current system. This challenger model aimed to provide more accurate alerts when a company might be heading toward financial trouble, allowing the bank to take preventive action.

Approach and Methodology

To build this model, I worked with real transactional data covering the financial behavior of thousands of companies from 2015 to 2023. One of the key elements of the dataset was the inclusion of a pre-defined reference point for each company, referred to as the "event date." This date was either the first known occurrence of a default or a randomly assigned point in time for companies without any default history. Using this event date, I was able to consistently collect and analyze one year of historical data preceding the event across all companies.

Next, I carried out feature engineering, which involves transforming raw data into meaningful indicators that a machine learning model can understand. This step was particularly important because I was working with over 3,000 different variables, and many of these had missing values or showed limited variation. I introduced new types of aggregations, such as calendar-based timeframes and percentile calculations, to better capture trends in the financial behavior of each company.

To handle missing data, I applied a unique strategy that categorized different reasons for missingness and assigned distinct placeholder values accordingly. This approach was meant to preserve the informative value of missing data, which can sometimes signal underlying issues in a company's operations.

For modeling, I used two machine learning algorithms: LightGBM and Random Forest. These models work by combining many decision trees to predict whether a company is likely to default. I experimented with four different modeling approaches, each combining different techniques such as LASSO (a form of regularization used to select important features) and sign-based dummy variables that help the model understand the direction of change in financial patterns.

Finally, I evaluated each model using standard performance metrics such as recall (how well the model identifies actual defaults), precision (how often flagged companies actually default), and AUC (a measure of how well the model distinguishes between defaults and non-defaults). Special attention was given to recall, as early identification of at-risk companies is the most valuable outcome for the bank.

Outcomes and Value to the Client

The project produced several valuable insights and demonstrated the potential of alternative modeling strategies. While the new feature engineering techniques did not consistently improve the model's performance across all metrics, certain approaches, especially those incorporating sign dummies, led to improvements in recall and AUC, two of the most important indicators for early warning systems.

One of the most promising models was the Random Forest model built using the second approach, which balanced high recall with relatively stable performance across training and test datasets. This indicates that the model is likely to generalize well to unseen data and could be a strong candidate for future implementation.

More broadly, the bank now has a structured comparison of modeling approaches that can be used to guide future improvements to its EWS system. This includes a clearer understanding of how feature engineering, missing value treatment, and model tuning interact to affect performance.

Personal Learning Experience

This project was a rich and challenging learning experience. One of the key lessons I took away is that real-world data science is far more complex than textbook examples. In textbooks, datasets are often clean and balanced, and algorithms perform as expected. In this project, I had to work with massive datasets, where even basic operations like aggregations could strain computational resources. This forced me to carefully consider the computational cost of each step and think strategically about performance.

It was also my first time using PySpark, a distributed computing tool used for processing large datasets. At the beginning, writing efficient code in PySpark was unfamiliar and challenging. But as I worked through the project, I gained confidence and learned to write optimized code that could scale to large volumes of data.

Another major learning point was working with high-dimensional data, thousands of features. Managing and selecting relevant variables from such a large set required creativity and a structured approach. I realized how crucial good feature engineering is.

Ultimately, this capstone gave me the chance to apply my technical skills to a real-world financial problem, while also sharpening my project management, critical thinking, and communication abilities. It also deepened my appreciation for the challenges and trade-offs involved in deploying predictive models in a business setting.