Link to the Github repo.

Introduction

In this project we are going to predict corporate exit. We are going to predict the probability of firm exit and classify firms into prospective exiting firms and prospective staying in business firms. We are going to use `bisnode-firms` dataset. The data was collected and cleaned by Bisnode, a major European business information company.

Our study will focus on companies in the electronic product manufacturing industry. We will utilize data from before 2015 to train and test a model to predict whether companies remained in business in 2015.

We do not have an explicit y variable to indicate if a company exited the market. Instead, we will define an indicator variable where a value of 1 denotes companies with sales greater than 0 in 2014 but did not exist in 2015 or had zero or missing sales.

Data Cleaning and Feature Engineering

We have started with dropping the columns with more than 70% missing values, because it wouldn't be very informative and decisive to keep them. Subsequently, we created all possible combinations of firms and years, intending to create a new column named "status_alive." This column would indicate whether a company is still in operation based on non-missing and positive sales values.

To assess the financial indicators, we created flag variables to indicate potential issues. These flags identify negative values, missing data, and values that are too high or too low. In cases like sales, where the distribution has a long right tail, we also calculated logarithmic measures. To address the issue of negative values in the sales data, we replaced them with a value of 1 and created a corresponding flag variable. We generated new variables as sales ratios for profit and loss, and balance sheet indicators.

For excessively high and low values, we employed the winsorization technique. For excessively low values, we established a threshold and substituted the same values for all excessively low values. We used the same technique in reverse for excessively high values. Additionally, we imputed missing values in numerical data with means and categorical data with mode. Flagged data that had been imputed.

Define Holdout and Work sets

To study firm default, we created a holdout dataset using specific filtering criteria. The dataset includes firms in the "Manufacture of computer, electronic, and optical products" industry (ind2 == 26) with 2014 sales figures between 1,000 EUR and 10 million EUR. This selection ensures that we focus on small or medium enterprises (SMEs) that existed in 2014, had positive sales, and did not exist in 2015 (either due to zero sales or missing data), meeting the definition of default. Our successful sample design resulted in a dataset of 1037 firms, of which 56 defaulted, and 981 remained active. The dataset shows an average sales figure of 0.4902 million EUR, with minimum and maximum sales recorded at 0.00107 million EUR and 9.57648 million EUR, respectively, adhering to the assignment's specified criteria.

For our work dataset, we selected all small and medium-sized enterprises (SMEs) from 2014 with sales ranging between 1000 EUR and 10 million EUR, excluding companies in the manufacturing electronic products field. Additionally, we constructed categorical variable matrices for various categories in the dataset. To establish reference categories for comparisons, we removed the first level of each category.

Modeling

To facilitate analysis, we organized all variables into several groups. `rawvars` primarily contains financial raw indicators. `qualityvars` include balance sheet indicators like "balsheet_length" and "balsheet_notfullyear." `engvar1` comprises engineered variables, including calculated ratios. `engvar2` contains more intricate indicators, such as quadratic forms. `d1` contains differences and logarithms. `hr` includes variables related to human resources, such as labor, CEO age, gender, and the number of directors. Lastly, the remaining flags are grouped in the last variable group.

Using these groups of variables we built 4 logistic regression models.

- Model 1: only using rawvars
- Model 2: adding qualityvars and categorical variables
- Model 3: adding engvar1 and engvar2
- Model 4: adding d1, hr and flags

We applied 5 fold cross validation.

	CV RMSE	CV AUC
X1	0.299777	0.731431
X2	0.298861	0.709731
X3	0.294360	0.741380
X4	0.289897	0.767637

The most sophisticated model, **X4**, exhibited the best performance in terms of RMSE and AUC, with lower RMSE and higher AUC being more favorable. This model performed exceptionally well on the holdout set, achieving an RMSE of 0.214 and an AUC of 0.813 (see Appendix 1).

Next, we started to determine the optimal threshold (0.148) for classification purposes, which resulted in the generation of a confusion matrix for our logistic model.

	Predicted no default	Predicted default
Actul no defaul	909	72
Actual default	28	28

The results were quite interesting, with a false positive rate of only 7%, but a relatively high false negative rate of approximately 50%.

We continued our experiment using the Random Forest model.
When comparing the confusion matrix, we obtained similar results. However, other indicators yielded marginally more promising outcomes. It was observed that Random Forest outperformed Logistic Model 4, as evidenced by a lower RMSE (0.209 vs. 0.214). The AUC value was also higher (0.86 vs. 0.813), as shown in Appendix 2. Despite Random Forest's superiority in other measures, Logistic Regression exhibited a slightly lower expected loss (0.613 vs. 0.616).

	Predicted no default	Predicted default
Actul no defaul	898	83
Actual default	26	30

Since the Random Forest model is a black box, we included a variable importance plot (Appendix 3) to illustrate the variables with the most significant contributions. Additionally, we provide a partial dependence plot (Appendix 4) to demonstrate the relationship between the probability of default and one of the most important variables identified in the previous plot.

Conclusion

Comparing the performance metrics of Logistic Regression and Random Forest models:

- Accuracy: Both models show high accuracy, with logistic regression slightly outperforming the random forest.
- Sensitivity (Recall): The random forest has a higher sensitivity, meaning it is better at correctly identifying positive cases.
- Specificity: Logistic regression exhibits higher specificity, indicating it is more adept at correctly identifying negative cases.
- Precision: Logistic regression has a marginally higher precision than the random forest, suggesting it has fewer false positives relative to true positives.
- AUC: The Area Under the Curve is higher for the random forest, suggesting it is better at distinguishing between the classes across all thresholds.
- Expected Loss: The expected loss, which factors in the cost of false negatives and positives, is very similar between the two models, despite the higher cost associated with false negatives.
- RMSE: The Root Mean Square Error is slightly lower for the random forest, indicating better performance in terms of predicted probability accuracy.
- Optimal Threshold: The optimal threshold for classification is higher for the random forest, which may be contributing to its lower specificity.
- Brier Score: The Brier score, which measures the accuracy of probabilistic predictions, is lower for the random forest, indicating better performance in this aspect.

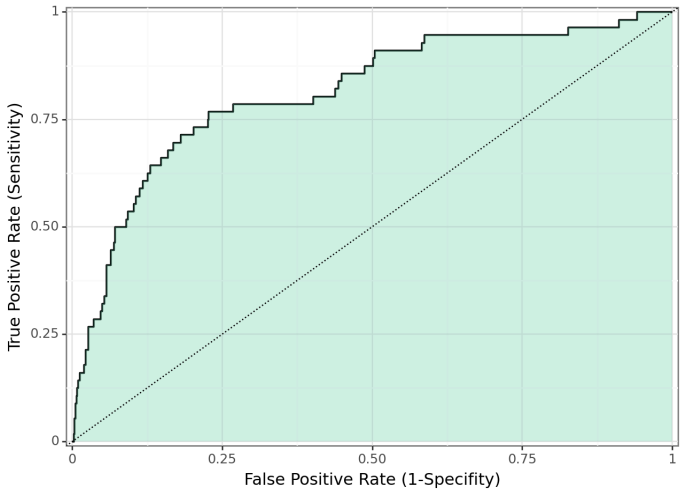
	Logistic regression	Random forest
Accuracy	0.904	0.895
Sensitivity	0.500	0.536
Specificity	0.927	0.915
Precision	0.280	0.265
AUC	0.813	0.860
Expected loss	0.613	0.616
RMSE	0.214	0.209
Optimal threshold	0.148	0.192
Brier score	0.176	0.165

Between Logistic Regression and Random Forest, the latter emerges as the better model when considering the balance between all the metrics provided, particularly in the context where false negatives carry a higher cost than false positives. The Random Forest model exhibits a higher Sensitivity, which is crucial in minimizing false negatives—this means it is more effective at identifying cases that are truly defaulting, a vital characteristic in financial risk assessment where the consequences of missing a default can be very costly. Additionally, the Random Forest has a higher AUC, indicating a stronger ability to differentiate between defaulting and non-defaulting cases over various threshold settings, a key advantage in creating a more robust and discriminative model.

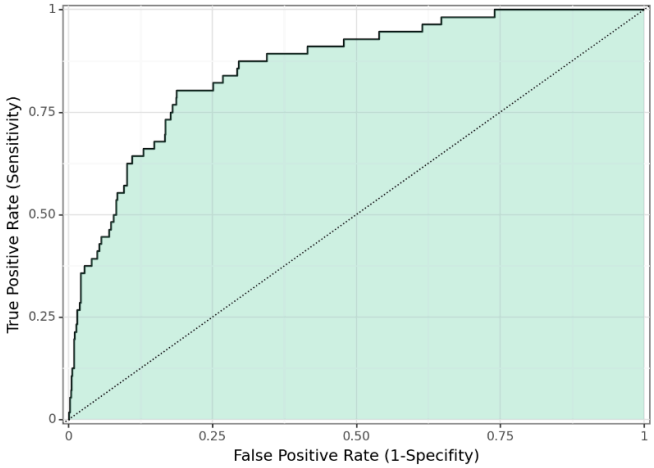
Furthermore, despite a slightly higher Expected Loss and lower Precision, the Random Forest model has a better Brier score, which shows it provides more accurate probability estimates, and a lower RMSE, suggesting its predictions are closer to the actual outcomes. In practice, the Random Forest model's strengths in Sensitivity, AUC, Brier score, and RMSE would likely translate to better performance in real-world default prediction scenarios, where accurately identifying default risk and estimating the probabilities of default are of great importance. Hence, the Random Forest model is preferred for its comprehensive performance and suitability for complex risk prediction tasks.

However in this particular case, when the goal is to get the model with the smallest expected loss the logistic regression could be a better choice!

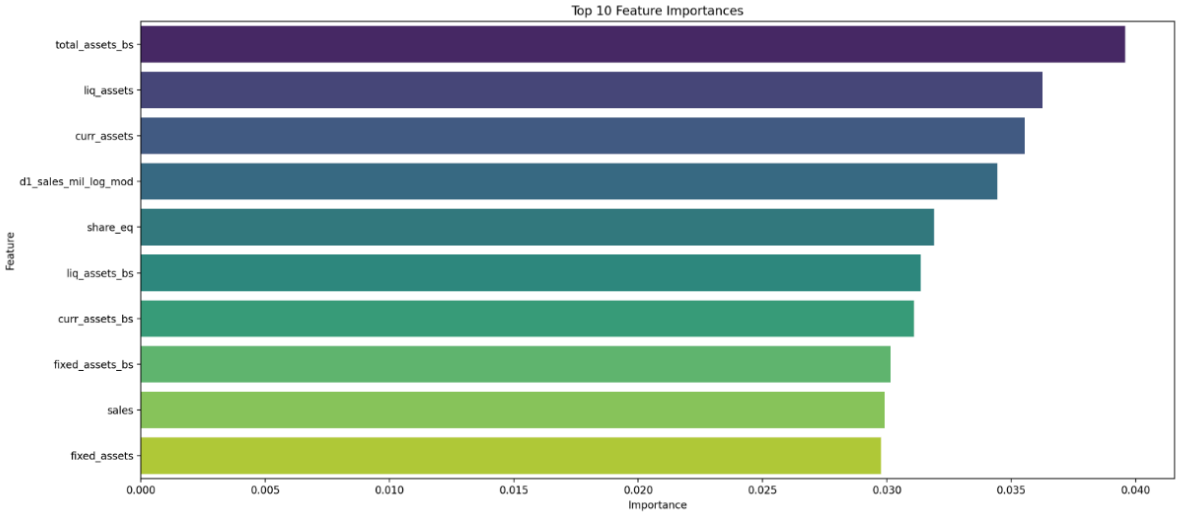
Appendix 1



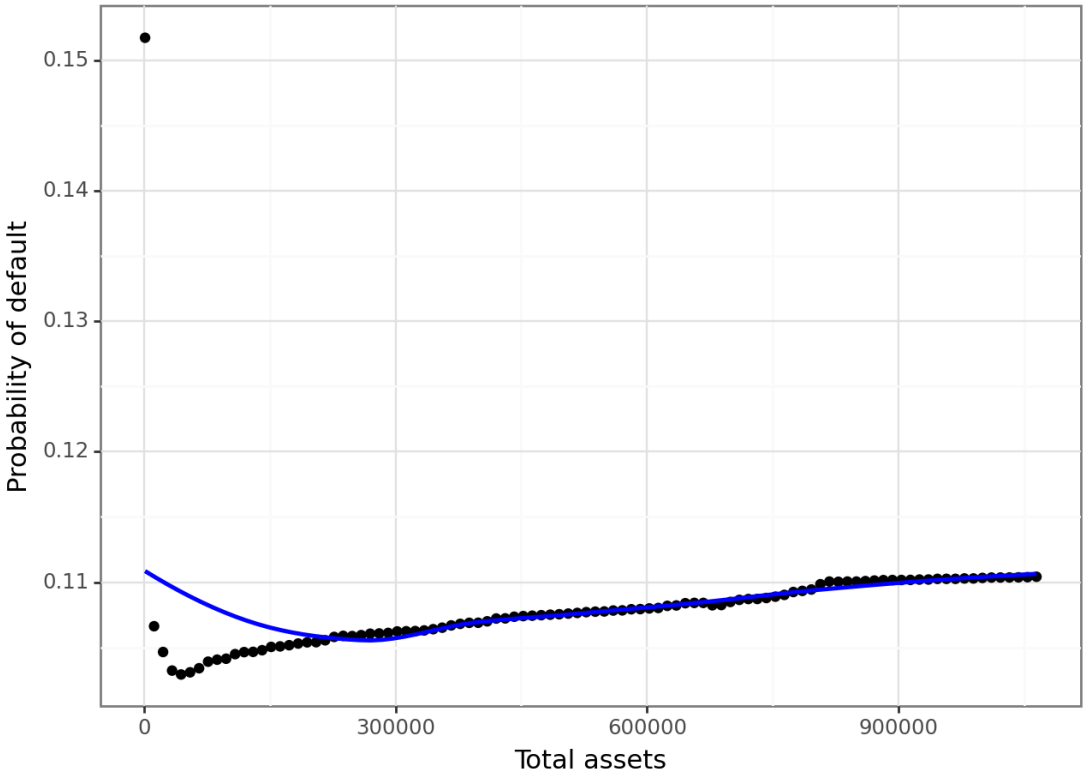
Appendix 2

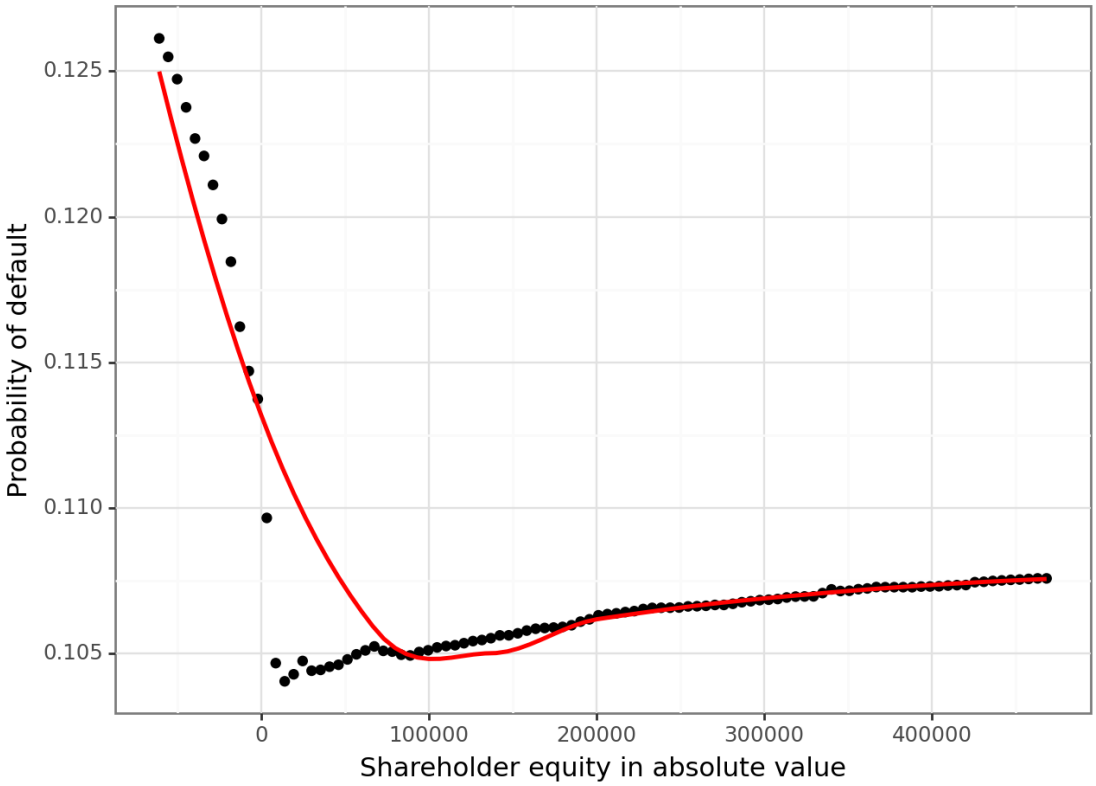


Appendix 3



Appendix 4


0.15
0.14
0.13
0.12
0.11
Probability of default
0
300000
600000
900000
Total assets


0.125
0.120
0.115
0.110
0.105
Probability of default
0
100000
200000
300000
400000
Shareholder equity in absolute value