Estimation of Bankruptcy Risk

1. Introduction

This document contains information about the estimation of bankruptcy risk of companies in Valuatum Platform. The paper starts by discussing the bankruptcy risk estimation in general and by presenting some of the more common prediction models, including a more detailed description of gradient boosting model, which is the main tool in predicting the bankruptcy risk in Valuatum system. After that, an overview of model training and development is presented, followed by a more detailed explanation of model validation.

1.1. What is bankruptcy risk estimation?

The goal of the bankruptcy risk functionality in the Valuatum Platform is to provide an accurate estimation of how likely a company is to go insolvent in the near future (e.g. in the next year or two). The predictions are made based on historical financial statement data for companies, and several different indicators are used (for instance profitability, debt ratio and current ratio). The models use raw figures and ratios calculated from the historical financial statements as inputs and provide a probability of how likely the company is to go bankrupt in the following *n* years (where *n* is usually 2 or 3).

It is worth noting that the predicted probability is just a statistical estimate. Therefore, a company with a very low estimated bankruptcy risk may go bankrupt due to internal or external shocks. The rate of bankruptcy in general may also increase or decrease significantly due to changes in the economic climate. The estimated probability should therefore be viewed in relation to other companies (i.e. is the probability of bankruptcy high or low when compared to other similar companies) instead of taking the figure as an absolute truth.

2. Models

Predicting the bankruptcy risk of a company is called a *classification problem*, since the outputs are binary indicators of whether the company will go bankrupt or not. Bankruptcy estimation models are therefore called *binary classifiers*. Historically, logistic regression has been the most common method for estimating the bankruptcy risk, but recently more powerful and complex machine learning algorithms have emerged. This chapter provides and overview of some of the most common bankruptcy estimation models.

Even though the models can differ from each other significantly in terms of complexity and accuracy, the general principle is always the same. The model takes financial data as inputs and gives the bankruptcy risk as an output. Figure 1 shows a schematic view of a bankruptcy risk model.

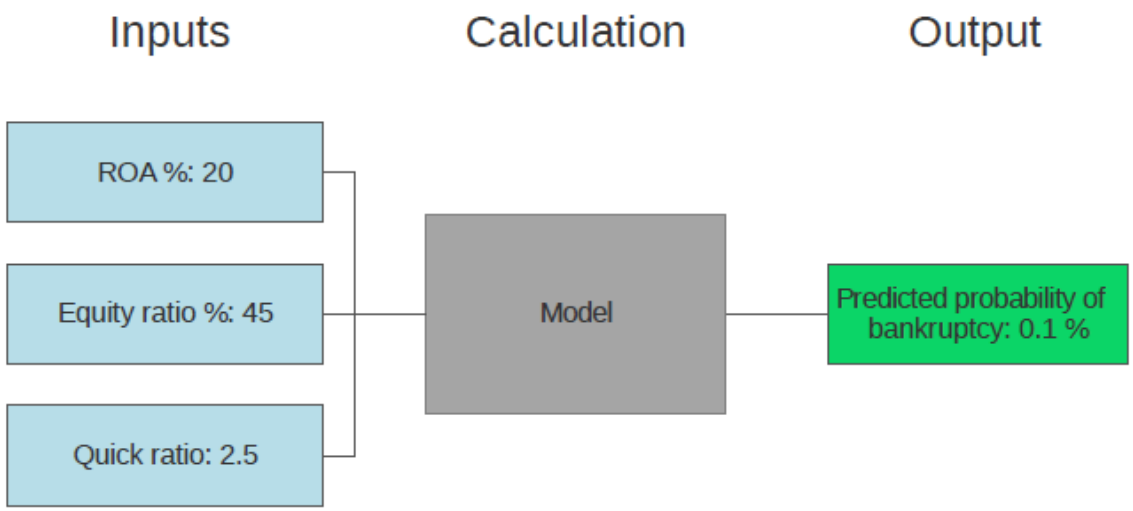

Figure 1: A schematic view of a bankruptcy risk model.

Similarly, the development of the model always follows the same general guidelines. The model is first trained with a large sample of historical data. This includes financial data as well as the information whether the company went bankrupt or not. This sample of data is called the training set, and it comprises of tens or hundreds of thousand of companies and includes millions of data points. The model performance is then evaluated using a similar data set comprised of different companies and data points. The details of evaluating a model depend largely on the type of the model. Training and validation will be discussed in more detail later in the paper.

2.1. Logistic regression

One of the most common methods in estimation of bankruptcy risk is *logistic regression*. It is a widely used statistical method that can provide probabilities for an event given numerical inputs. It is thus (in theory) well suited to perform bankruptcy risk estimation. In logistic regression (as in linear regression) each of the inputs a_i are multiplied with a weight coefficient w_i, and the results are summed. To be able to interpret the result as a probability the result is then passed through the *logistic function* that always outputs a number between 0 and 1 (giving a risk of bankruptcy between 0% and 100%). Logistic regression is widely used in estimation

of bankruptcy risk due to its simplicity, but the downside is that the weights assigned to different variables are always constant.

2.2. Machine learning algorithms

We have studied several different machine learning algorithms to find the one that is best suited for estimation of bankruptcy risk. This chapter starts by briefly covering two of the most prominent models, random forest model and artificial neural network, after which a more detailed explanation is provided on gradient boosting model which is the cornerstone of bankruptcy estimation in the Valuatum Platform. Machine learning models are generally much more accurate than logistic regression and can observe much more complex relationships between different variables. However, the increased complexity means that the models are so-called black box models. This means that it is often hard or even impossible to accurately know which parameters affect the final prediction and by how much.

Random forest model is built using decision trees. A decision tree is a flowchart-like structure where each node consists of a test on one of the input attributes, e.g., whether ROA-% is positive or negative. Each branch of the tree structure represents an outcome and at the end of each branch is a decision, in this case the bankruptcy risk. When training a random forest model, each node is set in a way that the estimated bankruptcies match the actual bankruptcies as well as possible. The final model consists of hundreds or even thousands of decision trees generated with different subsets of the training data. These decisions are then merged to form the final estimation for bankruptcy risk.

Artificial neural network is a model inspired by the biological neural networks found in the human brain. Neural network consists of nodes called neurons that are connected to each other. Each neuron receives as input one or more data points, the neuron weighs these inputs, sums them, and passes the outcome through a non-linear function called the *activation function*. This allows the neural network to model non-linear dependencies between the inputs and the output. The output of the activation function is then passed on to other neurons in the network. The neural network is trained by setting the weights in each neuron in a way that the predicted probability of bankruptcy is as close as possible to the actual status of the company. To achieve this, a so-called backpropagation algorithm is used.

While we were able to obtain promising results with both random forest model and artificial neural network, and by combining the two, we were not quite satisfied with them and continued development, which led us to study the gradient boosting model.

Gradient boosting model

Gradient boosting is a machine learning technique used for regression and classification problems that has recently been dominating applied machine learning competitions. Gradient boosting is an ensemble model, meaning that we build several models and combine them at the end to form the final prediction, just like with the random forest model. However, the key difference to the random forest model is that instead of building several *independent* models, with gradient boosting several *sequential* models are built. This means that each model that is being built learns from the mistakes of previous models. This feature is called *boosting*. Each model in a boosting process is called a *weak model*. While each individual model may indeed be weak, the idea is that together they will form a strong and accurate model. Boosting can be illustrated with a simple analogy of a golfer trying to hit a hole-in-one[1].

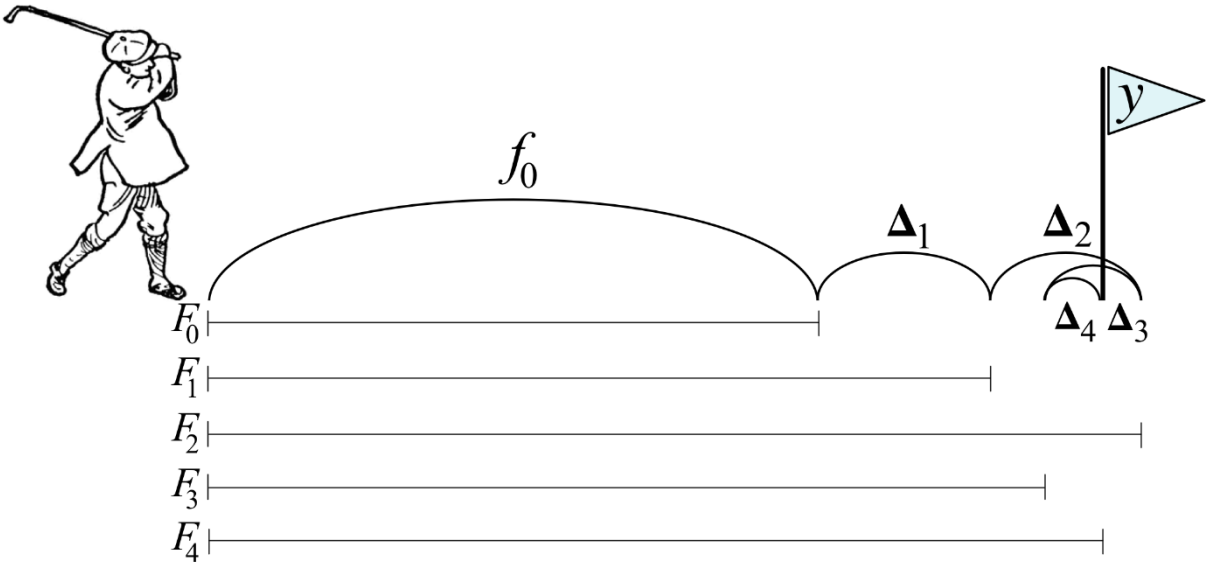

Figure 2: Simple illustration of model boosting.

Let us assume that the hole is 100 meters long and denoted by y. After the first stroke F_0 the golfer only gets to 70 meters, denoted by f_0, leaving him $y - f_0$ from the hole. F_0 is called the first approximation, or in this case the first weak model, and the difference to the hole is called the *residual* Δ_m, i.e., the mistake made by the first model.

The golfer's objective is now to train the model so that Δ_m approaches $y - F_{m-1}$. With perfect recall and precision (more about this in Chapter 5.2), he would be able to predict Δ_m

[1] The example from: https://explained.ai/gradient-boosting/. See website for more information about gradient boosting

perfectly, but as with any machine learning models the prediction is noisy. This results in second approximation, or weak model, being $F_1 = F_0 + \Delta_1$. The next time around he repeats the process using the previous model as a base model, and this time the approximation is $F_2 = F_1 + \Delta_2$. After the fifth approximation he has been able to appropriately train Δ_m. Table 1 illustrates the boosting process.

Table 1: Illustration of boosting where each model learns from the mistakes of previous models.

Stage m	Boosted Model	Model Output $\hat{y}$	Train Δ_m on $y - F_{m-1}$	Noisy Prediction Δ_m
0	F_0	70		
1	$F_1 = F_0 + \Delta_1$	70+15=85	100-70=30	$\Delta_1 = 15$
2	$F_2 = F_1 + \Delta_2$	85+20=105	100-85=15	$\Delta_2 = 20$
3	$F_3 = F_2 + \Delta_3$	105-10=95	100-105=**-5**	$\Delta_3 =$ **-10**
4	$F_4 = F_3 + \Delta_4$	95+5=100	100-95=5	$\Delta_4 = 5$

Boosting a model takes advantage of the mistakes made by the previous models F_{m-1}. In this content a model that does not utilize boosting could be described with a situation where the golfer tries to hit a hole-in-one five times but does not know where each shot lands. The final model would then be the average of these shots.

Estimating the bankruptcy risk of a company using dozens of variables is of course much more complicated than estimating a straight golf shot, but the principle is the same. We start with a weak base model whose residual (i.e., the error) is then passed on to the following model. Unlike in our golfer example, with bankruptcy risk estimation the noise in the prediction can never be eliminated completely but by running thousands of iterations the results can become very good.

Valuatum uses a gradient boosting algorithm called *XGBoost* (e**X**treme **G**radient **B**oosting)[2], which uses decision trees as weak models to develop our bankruptcy prediction model. As a result, we get a forest of decision trees whose results are added together. The biggest advantages of XGBoost over other gradient boosting methods are execution speed and model performance. XGBoost is also well suited for classification problems such as bankruptcy risk estimation.

[2] For more information about XGBoost, see, e.g., https://en.wikipedia.org/wiki/XGBoost or https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

3. Choosing the input variables

The bankruptcy risk model analyses a company's bankruptcy by looking at these four areas of the company: profitability, liquidity, indebtedness and size. We use 30 different variables to train our bankruptcy risk model because they give a thorough look at all categories mentioned above, giving an accurate overview of the company's situation which is used to estimate its bankruptcy risk as accurately as possible. These variables can be either raw data like net sales or ratios like equity ratio.

3.1. Liquidity

Liquidity is the most obvious aspect in determining a company's bankruptcy risk, since if a company does not have enough liquid assets to pay its current debtors it will go bankrupt. This means that if a company intentionally keeps little liquid assets on hand, it is more susceptible to unexpected shocks, such as an increase in interest rate or a temporary decrease in sales. Key variables that our model uses to determine a company's liquidity include quick ratio and current ratio.

3.2. Profitability

If a company has made profit in the past, it shows that the company has a working business model and it is likely to continue to work in the future. If a company has been profitable in the past, it at least has the potential to be profitable in the future and if a company will be profitable in the future, its bankruptcy risk will decrease significantly. On the flipside, if a company has not been profitable in the past, it could mean that the company has systematic issues that will lead to bankruptcy in the future. Key variables that our model uses to determine a company's profitability include return on assets, return on investment, return on equity and net earnings.

3.3. Indebtedness

In order to operate, companies require capital, and to get capital companies (almost always) need to acquire debt. This makes the ability to take loans essential for companies, because often it can be the only way that a company can afford to invest money into things like research and development or new equipment, which can be crucial for a company's growth and future profitability. But despite debt being essential for a company's success, it can also be a substantial contributor in a company's failure. For example, if a company takes a loan to start a new project and it plans on financing the loan with future profits from the project, but the

project fails and is not as profitable as the company hoped, the company is left to pay back the loan with money it might not have. This can lead to liquidity problems, and it is easy to see why indebtedness is an important factor in determining a company's bankruptcy risk. Key variables that our model uses to determine a company's profitability include equity ratio and gearing.

3.4. Company size

If a firm has grown to a large size, it is again an indicator of previous success and a working business plan. Additionally, larger companies generally have more resources that can be used to get the company back on its feet if it gets into financial trouble. These are some reasons why a company's size matters when considering its bankruptcy risk and why we have included these variables in our bankruptcy risk model. Key variables that our model uses to measure a company's size include net sales and balance sheet total.

These variables (and many more) are used in out bankruptcy risk model since it is necessary to look at a company from as many perspectives as possible when performing analysis such as credit risk analysis. For example, consider two types of companies; young companies that generally are not profitable since they are still developing their product which requires R&D, and banks that have very low liquidity due to their business models. Estimating either company's bankruptcy risk without thinking about all the aspects listed above can lead to very misleading results.

4. Model training

Once the variables used in the model training are decided, data with hundreds of thousands of data points from different companies is provided to the machine learning algorithm. This data contains both companies that have gone bankrupt and companies that have not gone bankrupt to ensure that the model can estimate bankruptcy risks that are statistically significant. The model then assigns weights to each of these variables, so that when it is given any company it can estimate its bankruptcy risk based on the values of the company's variables. It is important to note, however, that the machine learning algorithm does not produce a simple polynomial equation like, e.g., logistic regression. Instead, the machine learning algorithm is able to adjust the weights of individual parameters on a case-by-case basis based on other parameters. For example, while company's solvency plays an important role in bankruptcy risk estimation in

general, poor solvency might not increase the bankruptcy risk if the company is highly profitable and has low indebtedness.

We only use data from one year to train the model as this reduces the effect of macroeconomic factors that affect the bankruptcy risk but that companies have no control over. That said, the variables used in the model training include some variables that take data from many years. For example, one such variable is a variable that counts how many years the net sales of the company in question has been growing (or reducing). This gives the model some knowledge of the company's situation before the year that is used for training, and we have found that including variables like this improves the model's performance.

5. Model performance

The bankruptcy rate of companies in general is usually very low. Usually only about 1% of companies go bankrupt in the given year. This means that evaluating bankruptcy prediction models based on accuracy alone is not very informative, since a model that predicts that no companies go bankrupt will have an accuracy of about 99%. It is therefore necessary to use other methods to evaluate the performance of the models.

5.1. Receiver operating characteristic curve

A commonly used way to evaluate the performance of binary classifiers is to plot the Receiver operating characteristic curve, i.e. the *ROC curve*. Each point on the ROC curve is created by first selecting a *threshold* for the model. Selecting a threshold means that we consider that the model predicts that a company will go bankrupt if the estimated probability is above the threshold. If the probability is below the threshold then the prediction is that the company will not go bankrupt. For each threshold we then calculate the *true positive rate* (the proportion of bankruptcies that are correctly identified as such) and the *false positive rate* (the proportion of companies that the model predicted to go bankrupt but did not) of the model.

Table 2:Definition of true positive rate and false positive rate

	Did not go bankrupt	Went bankrupt
Predict not bankrupt	True negative (TN)	False negative (FN)
Predict bankrupt	False positive (FP)	True positive (TP)

$$\text{True positive rate} = TP / (TP + FN)$$

$$\text{False positive rate} = FP / (FP + TN)$$

By repeating the process for several different threshold values and plotting the points (false positive rate to x-axis and true positive rate to y-axis) we get a complete ROC curve.

To quantify the graphical illustration, area under curve (AUC) is calculated. This measures the percentage of area under the ROC curve relative to the total area. Figure 3 presents a ROC curve with an AUC value of 0.902. A random model would result in an AUC of 0.50 (represented by the red dashed line) while a perfect model would have an AUC of 1.0.

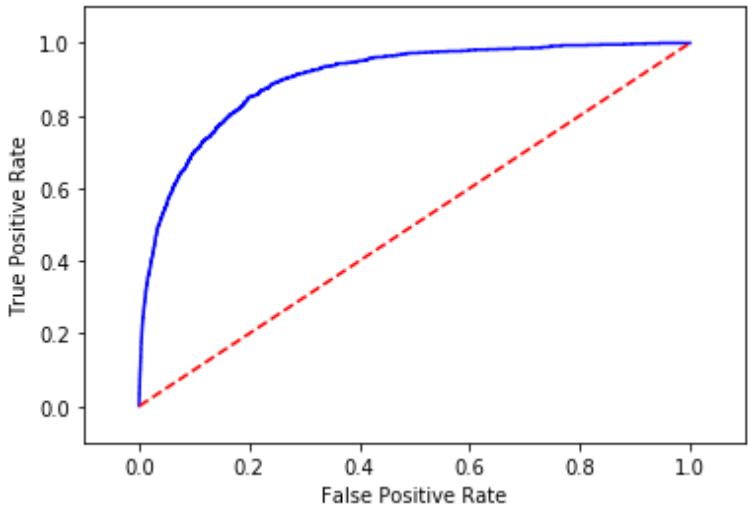

Figure 3: ROC curve with AUC value of 0.902.

When the threshold is decreased, more companies are predicted to go bankrupt increasing both the true positive rate and the false positive rate.

Naturally, we want the model to have a high true positive rate and a low false positive rate. Therefore, the closer the ROC curve gets to the upper left corner the higher the AUC and the better the model performance is. What can be considered a good ROC-AUC value is very dependent on the problem, but a rough guide is that an AUC of over 0.8 is good and a value of 0.9 or more can be considered excellent.

5.2. Precision-recall curve

Another less common way to visualize the performance of binary classifiers is to plot the precision-recall curve, i.e. the *PR curve* presented in Figure 4. The points on the curve are created by varying the threshold as in the case of the ROC curve, but instead of plotting the true positive rate and the false positive rate we plot the precision (how many of the companies predicted to go bankrupt actually went bankrupt) and recall (how many of the companies that went bankrupt are predicted to go bankrupt). They are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

where *TP*, *FP*, and *FN* stand for true positive, false positive, and false negative, respectively.

Just like with the ROC curve, we can calculate the area under the PR curve. The resulting value is called the PR-AUC number. A higher PR-AUC number is better. A random model will get an AUC number that is equal to the rate of bankruptcies in the data set (about 0.01). A perfect model will have an PR-AUC number 1. There is no general rule for what can be considered a good PR-AUC value, since this is very dependent on the problem. The PR-AUC values are not widely used in bankruptcy risk estimation literature, so no indication of what a good value for PR-AUC could be is available.

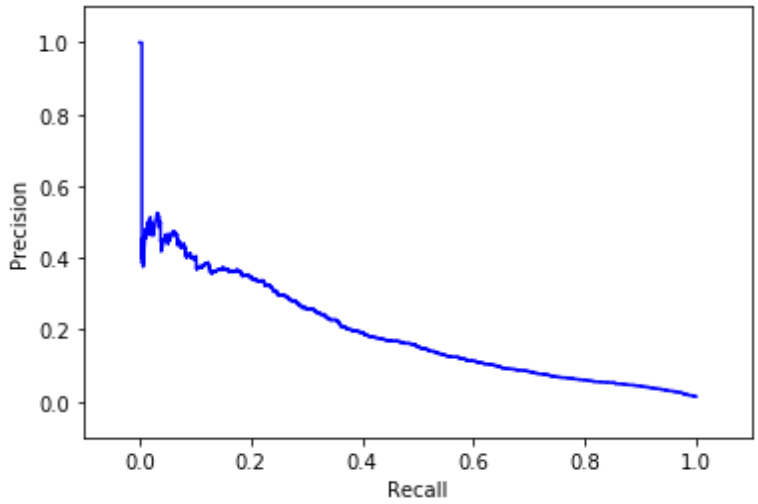

Figure 4: PR curve with AUC value of 0.192.

5.3. Moving average graph

Another way to visualize the performance of the model is to draw a graph that plots the moving average of predicted bankruptcy risk and moving average of the actual bankruptcy rate in the same graph. The graph is made by ordering all the companies in the test set in the order of their predicted bankruptcy, and then calculating the moving average for a certain group size. The graph gives a visual way to confirm that the average probabilities predicted at different levels of risk corresponds to the actual average risk.

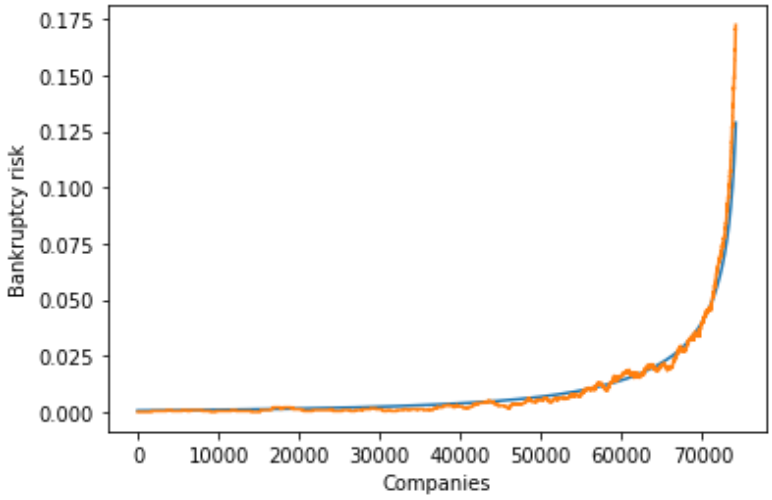

Figure 5: Moving average graph with a group size of 2500.

Figure 5 shows the moving average graph with a group size of 2500 companies. Orange curve represents the actual bankruptcies and the blue curve is the moving average of predicted bankruptcies.

6. Determining the interest rates

Once we have estimated the bankruptcy risk for a company, it can be used to determine the optimal interest rate that should be charged for a loan given to the company. However, determining the interest rate is not our expertise, and hence our model does not determine what interest rates should be given to companies.

Instead, we leave this to our customers. Since our customers are investment banks and credit institutions, they are much better equipped at setting interest rates as they have years of experience setting interest rates. Our role to provide our customers with information that is as accurate as possible, providing support to our customers so that they can make the correct decisions regarding lending and interest rates.