

Università degli Studi di Torino



School of Management and Economics

Master's degree in Economics

Use of Genetic Algorithms in Credit Scoring. An Experiment on Italian Firms.

Candidate:

Iuliia Tolkacheva

Supervisor:

Prof. Pietro Terna

Opponent:

Prof. Sergio Margarita

Academic Year 2015 - 2016

ACKNOWLEDGMENTS

First of all I would like to express my deepest gratitude to Professor Pietro Terna, not only for being an excellent supervisor, but also for the continuous support, motivation and insightful advises and comments that greatly improved my thesis.

I would also like to thank Professor Sergio Margarita for accepting to be the opponent for my thesis and for the interest shown to my work.

My sincere thanks also go to Cerved S.p.A and, in particular, to Dr.ssa Stefania Mazzone, Dr.ssa Rossella Momo and Dr.ssa Natalia Leonardi who greatly assisted my research and provided me with the data for creating the credit scoring models. They gave me a unique opportunity to perform a study based on real data.

Finally I would like to thank my family and Guido for the continuous support and encouragement throughout my master degree.

CONTENTS

1	INTRODUCTION.....	4
2	CREDIT SCORING	6
2.1	What is credit scoring	6
2.2	Main credit scoring models	7
2.2.1	Discriminant Analysis	8
2.2.2	Regression Analysis	10
2.2.3	Neural Networks	11
2.2.4	Other non-parametric techniques	12
2.3	Benefits and limitations of credit scoring models	14
2.3.1	Basel II and internal ratings-based models (IRB)	15
3	GENETIC ALGORITHMS.....	18
3.1	Main characteristics of genetic algorithms	18
3.2	Use of genetic algorithms in credit scoring – Literature review	20
4	AGENT-BASED MODELLING.....	22
4.1	What is agent-based modelling.....	23
4.2	Motivations for agent-based modelling	24
4.3	Weaknesses of agent-based modelling.....	24
5	THE SOFTWARE PACKAGES USED.....	26
5.1	NetLogo	26
5.2	BehaviorSearch	28
6	RESEARCH DESIGN	32
6.1	Data	32
6.2	Data preparation	34
6.3	Choice of financial indicators.....	34
6.4	The research method	36
7	CREDIT SCORING MODEL №1	37
7.1	How it works	37
7.2	Experiments in BehaviorSearch	41
7.3	Comparison of the results obtained.....	48
8	CREDIT SCORING MODELS №2.1 AND №2.2.....	49
8.1	Model №2.1	50
8.2	Model №2.2	55
8.3	Comparison of models №2.1 and №2.2	59
9	CREDIT SCORING MODELS №3.1 AND №3.2.....	61
9.1	Model №3.1 with non-normalized financial indicators	63
9.2	Model №3.1 with normalized financial indicators.....	71
9.3	The results of credit scoring model №3.1.....	78
9.4	Model №3.2 with non-normalized financial indicators	80
9.5	Model №3.2 with normalized financial indicators.....	86
9.6	The results of credit scoring model №3.2.....	93
9.7	Comparison of models №3.1 and №3.2	95
10	TESTING THE MODELS.....	97
10.1	Experiment №1	97
10.2	Experiment №2	99
10.3	Experiment №3	101
10.4	Experiment №4	102
10.5	The results	104
11	REPLICATION OF VARETTO'S MODEL (1998).....	106

12	FINAL RESULTS AND DISCUSSION.....	115
12.1	Further research	122
13	CONCLUSIONS	123
14	REFERENCES.....	126
A	APPENDIX.....	129
A1	The Python code.....	129
A2	The financial ratios used in the 14 credit scoring studies analyzed.....	129
A3	The NetLogo code of models №3	131

1 INTRODUCTION

In the 1990s, the Italian banking system underwent great changes: the liberalization of financial markets and the developments in information and communication technologies. In particular, the expansion of banks in new geographical and product areas and the profound reorganization of banks' loan-granting decision resulted in easier access to credit for households and companies. With the banking system growing at a rapid pace, it became crucial to develop and adopt valid and effective credit scoring techniques.

Credit scoring models gained even more importance after Basel II entered into force. In the late 2000s, credit scoring systems were widespread among medium-large banks and smaller players belonging to large groups. They were used mostly to evaluate small and medium firms, and to a much smaller extent to evaluate large companies. According to data from the Bank of Italy, in 2009, 68.8% of banks were using credit scoring models in order to assess corporate loan opportunities.

For the banking industry the role of credit scoring is essential, as effective and reliable credit scoring models are needed to distinguish healthy firms from unhealthy ones. As the models used in practice by banks are often proprietary, it is very hard to study these models and to understand the fundamental principles that drive these systems, to assess their performance and to measure their error rates.

Nevertheless, the objective of this thesis is to develop an agent-based credit scoring model using the sample of 1013 Italian industrial firms, to apply genetic algorithms to this model and to evaluate the performance of this sample model.

The thesis consists of two parts: a theoretical part (Sections 2 to 5) and a modelling part (Sections 6 to 13).

Section 2 presents a literature review of credit scoring models, focusing on their uses, on their benefits and on their limitations.

Section 3 narrows the focus to a specific type of credit scoring model – genetic algorithms – discussing its main characteristics and presenting examples.

Section 4, then, explains the methodology around agent-based modelling. In particular, the main features of this methodology and the advantages of this analytical approach are presented.

Next, a short operating guide for the two software applications used in the modeling section – NetLogo and BehaviorSearch – are presented in Section 5.

The initial model design is explained in Section 6. This section includes a description of

the dataset used for the analysis and the selection process used to identify the 17 financial indicators applied in the credit scoring models.

Several credit scoring agent-based models are developed in the next three sections (7, 8 and 9). The first model, discussed in Section 7, is a very simplistic one, used to familiarize the reader with the modeling techniques used throughout the paper. Section 8 covers two more sophisticated models, which are estimated using two sets of 17 indicators. Consequently, the second credit scoring model consists of two versions: model №2.1 and model №2.2. Finally, a third model is developed in Section 9. Once again, this third model is estimated on two sets of indicators, used both in their normalized and non-normalized versions. In addition, the synergies between two financial ratios will be added to our model to relax the strong assumption of linearity. The four best configurations of this third model are selected and used for subsequent tests, in Section 10

Section 11, instead, provides a glimpse into more sophisticated modeling techniques, presenting a replication of a credit scoring model developed by Varetto (1998).

Section 12 then evaluates and compares the models developed in this thesis, discussed the final results and provided some suggestions for further research.

Section 13 concludes the thesis with several final observations.

2 CREDIT SCORING

It is hard to imagine our modern world without credit, a key driver of economic growth. For a long time, borrowing money has been almost a privilege and credit granting decisions have been based mostly on personal relationships between banks and borrowers.¹ In contrast, credit is now accessible to a much larger audience.

With the growth of the credit industry seen over the last few decades, it has become crucial to develop accurate tools for decision-making and risk assessment that can assist banks in their provision of credit to the economy. Credit scoring models have been one of the earliest developments in financial risk management addressing this need.

2.1 What is credit scoring

Following Anderson (2007), in order to define credit scoring, the term should be broken it into two components, credit and scoring:

- *Credit* simply means “buy now, pay later”. It is derived from the Latin word “credo”, which means, “I trust in”. The term credit refers indeed to personal obligations, whereby lenders grant credit under appropriate compensation.
- *Scoring* refers to “the use of a numerical tool to rank order cases according to some real or perceived quality in order to discriminate between them, and ensure objective and consistent decisions”. To put it simply, a score is a number that summarizes some quality over a reference population.

Accordingly, credit scoring is a set of techniques that help transform data into numerical measures that can provide assistance in credit-granting decisions.

Although the basics of modern credit-scoring techniques were first described in the 1930s by Fisher (1936) and Durand (1941), the real boost to the development and spread of these models came in the 1960s, with the inception of credit cards. On the one hand, a rapidly increasing number of households applied for credit card loans. On the other hand, the advent of computers experienced in those decades made it easier to manipulate and exploit financial information. In particular, FICO was the first company to devise a credit scoring system for banks and other financial institutions.

These credit scoring models proved very successful in the credit card industry² and in the

¹ See Anderson (2007)

² See Anderson (2007)

1980s these models spread to other areas, such as insurance, collections and others. For instance, in 1995 credit scoring was used by Freddie Mac and Fannie Mae who started assigning credit scores to mortgage-backed securities.

The 1990s saw further spreading of credit scoring models, which were mainly used in a dichotomic manner, assigning credit applicants to one of two groups: a “good credit” group – likely to meet their financial obligations – or a “bad credit” group – with a high probability of defaulting on their financial obligations.

Given the importance of these models to the finance industry, research on more accurate credit scoring models is still ongoing. A credit scoring model that helps a bank price risk in a correct manner represents an important source of comparative advantage in modern financial markets. Even a fraction of a percent increase in credit scoring accuracy can result in a significant increase in profits for a bank.

2.2 Main credit scoring models

Credit scoring models have evolved over the years in many different ways and they can be classified into two broad categories³: parametric techniques, which make assumptions about the data and non-parametric techniques, which do not require any (strong) assumptions on the data. *Parametric techniques* include:

- Discriminant analysis (DA);
- Regression analysis (linear probability modeling, logit and probit);

These methods are based on a deductive approach and demand some a priori assumptions that do not necessary hold. Because of these assumptions, *non-parametric techniques*, originally developed in the machine-learning field, were applied to financial analysis. These techniques include:

- Neural networks (NNs);
- Genetic algorithms (GAs);
- Classification trees;
- Trait recognition;
- K-nearest neighbors.

³ This categorization is based on Resti and Sironi (2007) and Anderson (2007), focusing only on the main credit scoring models.

These models follow a purely empirical inductive approach. The main critiques raised over them are the lack of transparency and the potential for over-fitting.⁴

Other techniques – such as linear programming – exist too, but their effectiveness and practical usefulness is still to be ascertained.⁵

Overall, no silver bullet exists when it comes to choosing one of these techniques and the choice seems to be very context-dependent. For instance, the logistic regression is one of the most widely accepted techniques, especially due to the simplicity of converting its scores into probability estimates. NNs, instead, are a good solution for fast-changing environments, like the ones that characterize fraud detection problems.

2.2.1 Discriminant Analysis

Discriminant analysis (DA) is a parametric statistical technique that is applied to a group of variables associated to the credit applicants and that helps identify which of these variables are most helpful to separate firms into a “healthy” group and an “unhealthy” one.

Usually, this process is performed in a stepwise manner: initially, all variables are analyzed and evaluated in order to determine which ones are most helpful in separating firms into two groups. The assessment is then repeated on this subset of variables and the process is repeated until all remaining variables are relevant to the discrimination of firms.

Alternatively, one could also perform the assessment in the opposite manner, choosing an initial set of variables known to have high explanatory power and selecting one by one variables that can be individually added to the model to improve the scoring system.

Fisher (1936) was the first to describe discriminant analysis, which he initially used to classify flowers. In 1941, David Durand started developing the first financial applications of DA techniques.

The most popular application of DA techniques is the model introduced by Altman in 1968. Altman developed the first operational scoring model for predicting future corporate bankruptcy and applied it to listed US companies. His linear DA model, commonly referred to as the Z-Score model, includes only five financial indicators:

⁴ Generally speaking, *over-fitting* is a situation when a model relies too much on a particular training dataset and as a result performs worse on the new test dataset. The key problem is that instead of learning to generalize the dataset, a model memorizes it. Over-fitting usually happens with models that are too complex for the dataset they are based on.

⁵ See Anderson (2007)

$$Z_i = 1.2 * x_1 + 1.4 * x_2 + 3.3 * x_3 + 0.6 * x_4 + 1.0 * x_5$$

Where:

x_1 – Working capital / Total assets;

x_2 – Retained earnings / Total assets;

x_3 – Earnings before interest and tax (EBIT) / Total assets;

x_4 – Market value of equity / Book value of total liabilities;

x_5 – Sales / Total assets.

A higher Z-score indicates higher creditworthiness of the firm (low probability of default). After testing the model on samples including firms known to have defaulted, Altman identified the value of 1.81 as an appropriate cut-off between “healthy” and “unhealthy” companies.

The original Z-score model has been revised and modified many times to increase its accuracy for particular classes of firms. For example, Altman developed alternative versions of its model that could help score private firms, non-manufacturing firms and firms in emerging markets.

Like every other credit scoring model, DA models predict bankruptcy only up to some degree of probability. With DA models, the probability of making prediction errors is represented by 2 types of mistakes:

- Classifying an insolvent company as healthy (type I error);
- Classifying a healthy company as insolvent (type II error).

The costs associated to *type I errors* are the costs incurred as a result of the default of a company that was wrongly classified as “healthy” and was granted a loan. The costs associated to *type II errors* are costs corresponding to the foregone profit which would have been earned if the “healthy” applicant had not been classified as distressed and turned down.

The cut-off point for the score can be chosen to balance these costs. For example, a bank could decide not to grant a loan to a firm if the expected cost of a type I error exceeds the expected cost of a type II error.

DA models rely on an assumption of homoscedasticity of the data (specifically, equal variance and covariance matrices of the independent variables for the two groups of companies). In reality, as evidenced by empirical data, this assumption is often violated and adjusted versions of DA models (such as heteroscedastic or quadratic discriminant analysis) involve the estimation

of many additional parameters. In the absence of large datasets, this can reduce the statistical accuracy of the scores.

In addition, DA techniques assume a multivariate normal distribution of the data. - an assumption which is often empirically contradicted. The usage of a distribution with an unbounded domain is particularly inappropriate when the model includes financial ratios – which by definition lie in the $[0; 1]$ interval.

All these problems related to DA models represent significant challenges for practitioners. Regardless of these issues, though, discriminant analysis is still one the most commonly used techniques used in credit scoring applications.

2.2.2 Regression Analysis

(i) Linear probability modeling

Like discriminant analysis, regression analysis characterized early credit scoring models. In the most basic version of these models, probabilities of default were regressed on a number of appropriately selected independent variables. Early models assumed a linear relationship between the dependent and independent variables. These models are commonly referred to as Linear Probability Models (LPM) and they are usually estimated using ordinary least squares. The resulting estimates are used to predict the probability of default of companies applying for loans.

Just like any linear regression used to predict binary outcomes, LPM exhibit heteroscedasticity which reduces the robustness of coefficient estimates. Also, predicted probabilities of default may not lie in the $[0; 1]$ range. For these reasons, linear regressions are hardly used in practice, and non-linear specifications such as probit and logit models are preferred.

(ii) Logit and probit regressions

Logit and probit regressions are used to estimate the probability of an event (such as a corporate default) by regressing a binary dependent variable on a set of independent variables.

Logit and probit regressions relax the normality assumption required by DA models and, instead, they allow for different parametric distributions.

The logit model uses the following exponential transformation:

$$y_i = f(w_i) = \frac{1}{1 + e^{-w_i}}$$

Where the independent variable w_i is given by the function of the financial indicators x_{ij} :

$$w_i = \alpha + \sum_{j=1}^m \beta_j x_{ij}$$

That gives the logit model:

$$y_i = \frac{1}{1 + e^{-\alpha - \sum_j \beta_j x_{ij}}} + \varepsilon_i$$

The logistic function generates predictions for y that always lie in the $[0; 1]$ interval and which can be correctly interpreted as probabilities of default.

In probit models, instead, a normal cumulative density function is used.

The main difference between logit and probit models is that the logistic function has “fatter” tails. However, the predictions yielded by logit and probit models are often very similar.

The main shortcoming of these regressions is their high sensitivity to missing values. On the other hand, logit and probit regressions yield results that are easy to interpret and they arguably offer accurate default predictions. West (2000), for instance, used German and Australian data to compare neural network models with more traditional methods of credit scoring and found that the logistic regression minimized credit scoring error among all the methods used.

2.2.3 Neural Networks

The first two types of models described (DA and regression analysis) follow the so-called structural approach. They start with some assumptions made a priori (e.g. if a company has high D/E ratio, it will probably fail) and try to confirm them using the information contained in the dataset. Usage of this deductive approach may result in models that can be understood and gamed by the companies requesting loans. For instance, firms may succeed in identifying the variables that matter the most in banks’ credit-granting decisions and they may decide to adjust their accounting policies to obtain a loan. As a result, these models risk losing their effectiveness over time.

Neural networks (NNWs), instead, are learning systems that use an inductive approach. Starting from a given dataset, some empirical regularities are identified (e.g. unhealthy companies exhibit values of variable x_{ij} that happen to be higher than their counterparts on healthy firms) and they are used then to predict default probabilities. The empirical regularities detected in the data are not always easy to make sense of but they can still yield accurate predictions. The “obscurity” of these “black-box” models may also represent a strength of NNW models as they prevent credit applicants from understanding and gaming the credit-granting

assessment.⁶

NNWs try to mimic the learning mechanism characterizing human knowledge and memory. They are inspired by the human brain and its ability to adapt to new information. The black-box nature of NNWs and the assumption about non-linear relation among variables make them very different from classical credit scoring system such as Altman's Z-score.

In general, a neural network consists of a large number of elements – “neurons” – that are connected to each other. The links between the neurons are given a weight which determines the type and intensity of the information exchanged. The weights define the network behavior.

Neurons are arranged in “layers”; a neuron in a layer receives an input, from neurons in other layers and yields an output, which is transmitted to the neurons in the next layer. This process is repeated through all layers of neurons, after which the network generates a final result, such as a numerical score for the companies. This is the most frequently used neural network architecture for credit scoring – it is called multilayer perceptron (MLP). There exist many other types of NNW, such as mixture of experts (MOE), radial basis function (RBF), learning vector quantization (LVQ), fuzzy adaptive resonance (FAR) etc.

NNWs were first described in the 1960s but the first applications to credit scoring date back to the beginning of the 1990s. For instance, Altman et al. (1994) estimated NNW and DA credit scoring models on a dataset of 1,000 industrial Italian firms and found that – despite the accuracy of NNW models – DA models outperformed NNW ones. Tam and Kiang (1992), on the other hand, came to opposite conclusions in their study of Texas bank failures, finding that NNW models outperformed DA models, which in turn outperformed logistic regressions.

2.2.4 Other non-parametric techniques

(i) Classification Trees

Classification tree is a method that uses a tree-like structure to describe the decisions and its potential consequences. The idea of classification trees was developed in statistics independently by Breiman and Friedman in 1973. The concept of trees was soon applied to credit scoring by Makowski (1985) and Coffman (1986).⁷

The top of the classification tree is the *root node* which represents the sample of all applicants (good and bad firms). In order to construct a tree the initial set of all firms should be split into two subgroups using a particular discriminating variable. These subgroups, called *child*

⁶ See Resti, Sironi (2007)

⁷ See Thomas et al (2002)

nodes, are then split again, using another distinguishing feature. When no more discriminant variables can be used to divide the observations, the tree reaches a *leaf node*. The splitting process is repeated until all the observations are in leaf nodes.

Several rules define the classification trees:

- The splitting rule – a rule according to which the population should be split;
- The stopping rule – a rule defining when to stop creating new node;
- The assignment rule – a rule classifying each node as good or bad.

The assignment rule is the easiest to make, whereas the splitting rule is the most complex one. There are various measures defining whether a split is the best for a particular characteristic, such as Kolmogorov–Smirnov statistic, Gini index or entropy index.

The classification tree approach provides results that are transparent and are easy to use. Moreover, the technique is computationally simple and is able to identify patterns. On the other hand, the classification trees require a lot of data, but as they become larger, the number of cases in the nodes decreases that is leading to potential over-fitting and confusing results.⁸

(ii) Trait Recognition

Following Kolari et al. (1996), trait recognition (TR) is a nonparametric technique that does not require any underlying distributional hypothesis on the variables. In contrast to classification trees, TR is not guided by decision rules. Instead, its algorithm is based on the interactions between the variables of interest. The existence of these interrelationships between the variables is the main advantage of TR technique.

The TR algorithm starts with deriving binary string code for each observation that contains the information about financial ratios. Then all the string codes are used to construct a trait matrix where all possible combinations of the variables are included. The next step is to identify “good” and “bad” traits – distinctive features of failed and non-failed companies. In particular, “good” features are traits found in the majority of healthy companies and in a small number of failed companies. The opposite holds for “bad” features. “Good” and “bad” traits vote on each company in the sample and, finally, the voting matrix is derived. The matrix consists of safe and unsafe cells that assist modelers in determining whether a firm lies in the safe or unsafe traits space. Performing the algorithm for several iterations could guarantee the best result.

Several studies examined the performance of TR technique. Kolari et al. (2002) developed

⁸ See Anderson (2007)

an early warning system of US bank failures using logit and TR method. It was found that TR model generally outperforms logit model and specifically in overall accuracy and large bank failure accuracy. Lanine et al. (2006) also applied Logit and TR approaches to predict failures of Russian banks. Better results were obtained using TR technique.

(iii) K-nearest Neighbor

The K-nearest neighbor (KNN) method is a seeking patterns technique that is searching for similarities between a new input element and the elements in an already existing set. In the context of credit scoring, this new input element is an applicant and it is classified as good or bad based on the votes given by its K closest neighbors. The KNN approach was introduced by Fix and Hodges in 1952 and was employed in credit scoring purposes for the first time by Chatterjee and Barcun in 1970.

In order to apply the method, one needs to define the metric. The metric serves for measuring the distance between the new and the already existing applicants. Though several forms of the metrics exist, one of the most common is the Euclidian metric. As soon as the metric is defined, the estimation of distance between the applicants and the corresponding number of neighbors K is required. Henley and Hand (1996) applied KNN to credit scoring and used an adjusted version of Euclidian metric. After a large number of experiments, the results obtained with KNN technique were compared with other discriminant techniques revealing that the method proved to be effective.

The KNN technique is considered to be a simple technique⁹ and it allows automatic updating of the model, but it is not widely used in credit scoring because no actual final scoring is provided, which in fact, complicates the interpretation of results.

2.3 Benefits and limitations of credit scoring models

In the banking sector, credit scoring models provide an objective analysis of a customer's creditworthiness. They focus on relevant information only, ruling out discriminatory practices and personal subjectivity from banks' decision-making processes. Moreover, the use of credit scoring models reduces the costs of processing loan applications and allows automation of the credit-granting process.

Despite these benefits, a number of shortcomings associated to credit scoring models have been highlighted in the literature:

- The “true” credit scores change over time. For example the characteristics of a healthy

⁹ See Anderson (2007) and Henley et al (1996)

applicant could change over 5 years meaning that some applicants may require re-classification. The credit scoring models should be designed so as to remain relevant over time.

- Credit-scoring models ignore qualitative factors. Traits such as a company's reputation, the quality of its management and the outlook for the industry can be important factors in determining the insolvency of a company.
- There is no standard credit scoring model and different solutions may be required for different markets. Financial indicators differ very much from one industry to another, both in the values of certain variables and in their relative importance.

Even with the limitations highlighted above, the models described are still popular tools for predicting credit risk. In addition, following the introduction of the Internal Ratings-Based models under the Basel II regulations, the benefits of developing accurate credit scoring models have increased, given that these modeling techniques can now be used by banks to influence their capital requirements. This change is briefly discussed in the next subsection.

2.3.1 Basel II and internal ratings-based models (IRB)

Basel II is an international accord presenting a series of recommendations on banking laws and regulations introduced by the Basel Committee on Banking Supervision (BSBS). The New Accord is a successor of Basel I, which was introduced in 1988 in order to increase the safety of the international banking system. While Basel I focused on credit risk only, Basel II, instead, expanded its own scope to credit, operational, and market risks.

The new Basel II rules are centered around 3 pillars:

- 1) Minimum capital requirements;
- 2) Supervisory review;
- 3) Market discipline.

The first pillar deals with a bank's core capital requirement, calculated for the three main types of risks faced by a bank: credit risk, operational risk and market risk. The second pillar refers to a set of tools that national regulatory authorities can use to deal with risks and circumstances not captured by the pillar 1. The third pillar requires enhancing disclosure requirements in order to stimulate oversight by market participants.

Within its first pillar, Basel II provided banks, for the first time, with a choice between two approaches for calculating the capital requirement for credit risk. Under both approaches, capital

requirements are calculated as a percentage of risk-weighted assets.

Under the *standardized approach*, in order to determine the amount of required capital for a loan, banks are obliged to use ratings from external credit assessment institutions, such as rating agencies and export credit agencies. The higher the rating of the counterpart, the lower the risk weight. Compared to the Basel I framework, the standardized approach under Basel II was supposed to increase the sensitivity of risk weights to actual credit risk. For instance, the relationships between ratings and risk weights were differentiated based on the type of creditor (sovereigns, banks, private companies). The standardized approach is the basic, default approach and it does not require much modeling from the bank. It is relatively simple, though it is considered to be less flexible.

Under the *internal ratings-based approach (IRB)*, banks are allowed to use their own risk measures to calculate their regulatory capital, provided their risk measurement systems are approved by the national supervisory authority. In particular, this approach involves the use of a credit risk assessment model specified by the Basel Committee, which has 6 different inputs, corresponding to major drivers of risk¹⁰:

- Risk of default measured by the probability of loan default (PD);
- Recovery risk measured by the loss given default (LGD);
- Exposure risk measured by the exposure at default (EAD) ;
- Maturity of the loan;
- Granularity of the exposures (few large loans vs many small loans);
- Correlation among borrowers (diversification of the bank's portfolio).

Whilst the last two parameters are mandated by Regulatory Supervisors, banks can use their own models to estimate the remaining four parameters. There are two possibilities:

- *Foundation IRB approach* – whereby banks estimate their clients' PDs, while the values of the other parameters (LGD, RWA) are based on Supervisors' recommendations;
- *Advanced IRB approach* – whereby PD, LGD and EAD are estimated using internally-developed models.

Many banks and institutions have chosen to adopt the IRB, possibly in an attempt to optimize their capital requirements and cost of capital. For example, according to the European

¹⁰ See Anderson (2007)

Central Bank, over 7000 internal models have been developed by banks under its supervision. In Italy, 5 banks have adopted models approved by the Bank of Italy – Unicredit, Intesa San Paolo, MPS, UBI and Banco Popolare, while Banca Popolare di Milano and Banca Popolare dell'Emilia are waiting for official approval.¹¹ Even though the adoption of the IRB approach involves high costs of implementation and maintenance of the system, the internal rating system can be tailored to the bank's needs, portfolio allocation and risk management practices and may allow better credit risk quantification and higher transparency over risk measures used by banks.

¹¹ See Milano Finanza N° 161 of 18.08.2015, p. 9

3 GENETIC ALGORITHMS

The conventional statistical methods have several restrictive assumptions such as linearity, normality and independence among variables that limit the effectiveness and validity of the methods. These shortcomings have led to a rise in research of more accurate and sophisticated methods. Machine learning techniques are not affected by the restrictive hypothesis on which traditional approaches are founded. Moreover, the methods of automatic learning extract the rules from the data and search for regularities that could be too complicated to be explored by engineers.

Genetic algorithms are one of these techniques. They were invented and developed by John Holland in the 1960s at the University of Michigan. Holland's original goal was to apply natural adaptation mechanisms to computer systems. However, the method proposed was quickly adopted as a heuristic approach suitable for multi-parameter optimization and demonstrated to be effective in searching very large spaces.

3.1 Main characteristics of genetic algorithms

Genetic algorithms (GAs) are optimization algorithms that replicate the principle of survival of the fittest (Darwin's principles of natural selection) to identify the optimum of a generic problem. Just like natural selection leads to a continuous improvement of a species, genetic algorithms can be used to improve some measure of fitness of candidate solutions to a problem. Specifically, genetic algorithms replicate three fundamental processes involved in the evolution and adaptation of the species:

- Selection;
- Reproduction;
- Genetic mutation.

Selection refers to the identification of those individuals that survive to the next generation and that will reproduce. The survived elements are not the individuals with the best value of a one or another parameter, but the fittest ones overall that represent the best solution found to date.

Reproduction is the process whereby two individuals mate and genetic recombination (crossover) takes place, leading to offspring with improved characteristics (superfit). The underlying principle is that the populations of individuals evolve constantly, meaning that the species from each following generation are better suited to the environment than the individuals that they were created from.

Genetic mutation refers to the casual and sudden introduction of genetic changes in the population. These modifications occur very rare so as not to destroy the genetic heritage accumulated through previous selections.

The process continues through many generations and it terminates either after a certain number of iterations or if the improvements from one generation to the next are below a predetermined threshold values.

All these principles are transferred to credit scoring models. The generic scoring formula has the following form¹²:

$$z = \alpha_0 + \alpha_1 * x_1 + \alpha_2 * x_2 + \alpha_m * x_m$$

The vector $\alpha_o = [\alpha_0 \alpha_m]'$ contains the algebraic sign and weight with which the financial indicators enter in the construction of z and it represents a potential (candidate) solution.

Optimization through genetic algorithms occurs according to the following steps:

- 1) The initial population of N candidate solutions (chromosomes), which are usually encoded as binary strings, is generated randomly. The choice of population size is an important issue – if it is too small, the GA will likely underperform, whereas too large population can increase the processing time;
- 2) For every candidate, the fitness (the accuracy of the scores that are associated to that vector) is calculated. It is connected to the fitness function that measures the performance of each member of the current population. Whether a candidate will be included in the next generation depends on its fitness value;
- 3) All the candidates are ordered on the basis of their fitness. The selection algorithm is applied, identifying the individuals that will “survive” and those who will “die”;
- 4) The surviving members undergo a crossover or recombination procedure, which results in a second generation of solutions with properties (genes) inherited from their parents;
- 5) The mutation algorithm is applied to the new generation. Random genetic modifications occur with a small probability within the population, otherwise a high probability could lead to the loss of good quality solutions;

The sequence is repeated from bullet point (2) onwards. The cycle continues until a predetermined level of fitness is reached or until a stable level of fitness is observed through subsequent iterations.

¹² The formula is taken out from Resti, Sironi (2007)

The advantage of genetic algorithms is that they do not explore the solution space in a random manner. Using the fitness function and the genetic operators of selection, crossover and mutations, GA explore the space of solutions in a more effective way, using the information obtained in past searches and performing the so-called “adaptive search”. All the solutions compete with each other and that, in turn, helps identify the optimal solution. Though there is no guarantee that GAs will find the absolute optimum, GA represent an efficient way to identify optima more rapidly than with other optimization techniques.

The simplicity and effectiveness of GAs has made them a popular tool used in many different fields, such as criminalistics. Even when GAs perform worse compared to another procedure, they can still be used to improve on the solutions obtained with the other method.

3.2 Use of genetic algorithms in credit scoring – Literature review

GAs have been adapted to a very wide range of financial applications. For example, Colin (1994) and Deboeck (1994) used GAs to construct trading rules, Mahfoud et al. (1995) used them for stock selection, Kim et al (2012) – used them to predict the stock price index and Packard (1990) – used GAs for budget allocation exercises.

As for credit scoring applications, one of the first attempts to use GAs for bankruptcy prediction was that of Back et al. (1996). Using a sample of Finnish companies and then comparing the performances of GA with DA and logit regressions, they found that the best prediction result was associated to GA.

Varetto (1998) compared the LDA methodology with GAs, using data from a sample of Italian companies. He used these models for one-year-ahead and three-year-ahead bankruptcy predictions and concluded that LDA models yielded slightly better results than GA models.. As he points out, however, GAs involve shorter computation times and are much less dependent on financial analysts’ contributions.

Galvao et al (2002) developed corporate distress classification models using GAs and DA. Estimating his models on a dataset of 60 British corporations, the authors found that GA models had better cross-validation performance than standard Z-score methodologies. They also commented that GA models were more parsimonious than Z-score models, as the selection of financial indicators in GA models is completely data-driven., with the role of financial analysts being limited to a pre-selection of the variables to include in the model.

Shin and Lee (2002) generated rules for bankruptcy prediction using GAs and commented that GA-based predictions of corporate failures appeared to be promising, with an average classification accuracy of ca. 80%. Likewise, Davalos et al (2007) applied GA-based rule

extraction to the bankruptcy classification of US airlines. The rules developed through GAs forecasted insolvencies with an accuracy ratio of 91%.

Sexton et al (2003) used GAs to identify input variables for 30 NN models and found that GA-based pre-selection of variables improved the cross-validation performance of NN models and improved the parsimony of the models.

Finlay (2006) applied GA models on a dataset of UK credit applications of known performance and compared them to methods as NN, linear regression and logistic regression. Despite not being able to find a methodology that outperformed all other in all situations, Finlay noted the lower computational requirements of GA models.

4 AGENT-BASED MODELLING

Most economic models assume that rules are fixed, that the state of the world is static and that agents are rational, know their preferences and have all the information they need to make decisions. A rational agent maximizes its own utility under given constraints and manages to identify the optimal solution. Though this approach is widely used, it has important limitations in complex environments where uncertainty is high and agents exhibit bounded rationality, unpredictable behavior and a complex set of social interactions. In an increasingly globalized world, complexity can no longer be reduced through standard economic assumptions and conventional modelling tools are losing their power.

The need to incorporate complexity in modern economic models spurred research efforts in the field of Agent-based modeling (ABM). Agent-based modeling (ABM) consists in modeling economic agents (decision-makers) that interact with each other through a set of predefined rules. Agents included in the models may differ from one another and, through their interaction, new patterns and behaviors may arise that had not been explicitly programmed into the model. These features make the ABM technique innovative in its attempt at capturing complexity. A more detailed discussion of the modeling approach is given in section 4.1.

The origins of ABM can be traced back to John von Neumann and Stanislaw Ulam who in 1940s discovered cellular automata – a system capable of reproducing and regulating itself. Later in the 1970s, John Conway tried to build upon the original concept of cellular automata and developed the Game of Life, a two-dimensional cellular automata. The applications of these systems were very diverse. For instance, Thomas Schelling applied cellular automata to social science and explained the phenomenon of segregation – the separation of individuals into distinct racial groups.

With the appearance of specialized software for computer simulations, agent-based models (ABMs) experienced a surge in popularity in the early 1990s and found applications in diverse fields, such as social sciences or natural sciences. For instance, Griffin et al (2007) modeled the rise and fall of the polities that inhabited the Lake Titicaca Basin between AD 550-1000. Mast et al. (2007) used ABM to develop scenarios of offshore wind energy implementation in the Netherlands. With their ‘Santa Fe’ artificial stock market, Arthur et al. (1997) developed an Agent-Based asset pricing model. Janssen et al (1999) used ABM to investigate consumer behavior that can lead to “lock-in” effects – situations whereby one product is preferred over all others despite the absence of superior characteristics of the good or technology. Deffuant et al (2002) modeled the evolution of political opinions and the spreading of extremist ideas within a population, using ABM. Charania et al (2006) adopted ABM to simulate future market scenarios

for sub-orbital space tourism. Ehlert (2001), instead, analysed traffic flows with an Agent-Based model with agents with different driving styles interacted in a simulated urban environment.

ABMs have also had natural science applications. For example, Kreft et al (1998) undertook an ABM simulation of the growth and behavior of bacteria, Pogson et al (2006) used ABM for the study of intracellular chemical interactions and Folcik et al (2007) introduced an Agent-Based model to analyze the interactions among cells of the innate and adaptive immune system.

4.1 What is agent-based modelling

Following Gilbert (2008), *agent-based modeling* is a computational method for simulating the actions and interactions of individual agents within an environment. ABM is typically a computer simulation aiming to demonstrate how the diversity between agents' characteristics and behavioral patterns affects the evolution of the entire system.

Usually ABMs have three elements:

- *A set of agents.* Interestingly, there is no universal definition of an agent. Jennings (2000) uses a computer science description of agents, arguing that it is autonomous behavior that essentially characterizes them. According to Bonabeau (2001), instead, any independent element of a system can be considered to be an agent. Other authors, instead, focus on adaptive behavior as the defining trait of an agent. Macal and North (2009) provide a definition of agents based on three properties. First, agents operate autonomously on the basis of their perceptions of the outside environment and of their decisions. Second, agents have boundaries, meaning that an observer can tell whether something is or is not a part of an agent. Finally, agents interact with each other, according to specific protocols. Agents could also have additional non-defining properties, such as, for example, goal orientation or possession of resources. These additional properties however are context-dependent and vary depending on the modeling objectives.
- *A set of agents' relationships.* The essential part of ABM is the ability of agents to interact with each other so as to exchange information and to use that information to choose their behavior.
- *An environment where agents interact.* The environment is a virtual world typically representing a geographical area in which agents act and interact. Depending on the model, it may be designed to influence agents or not. It also worth mentioning that interaction among agents does not occur in an "all-to-all" fashion. Instead, agents usually

interact within a restricted group of neighboring agents, making the model more realistic.

4.2 Motivations for agent-based modelling¹³

The ABM approach has proved to be a useful and powerful tool in many different domains. Indeed, Agent-Based models help overcome the limitations of traditional theoretical modelling tools, for example by relaxing unwarranted assumptions.

In particular, one of the main critiques to economic research relates to the strict separation between theoretical and empirical models, with empirical findings often failing to reconcile with theoretical models and with theorists often favoring logical consistency over empirical relevance and empirical studies often. The excessive formality of some theoretical models is arguably a contributing factor to this separation. Thanks to their less structured framework, ABMs may allow a higher degree of integration between theoretical models and empirical data.

The lower degree of formality of ABMs also allows a more realistic representation of the phenomenon under investigation. As an example, it is easier to design an agent-based model that describes costumers' movements in a store, than to derive an equation explaining the changes in the density of costumers. The more realistic modeling allowed by ABMs facilitates the identification of causal connections within the system. In contrast, in equation-based models it is sometimes complicated to determine causality.

An additional advantage of ABMs is the presence of agents' heterogeneity, which represents a break from the standard representative agent approach of traditional economic models. ABMs generate a great number of unique interacting agents. Additionally, the agents are endowed with complex behavior: they can learn from and adapt to changes in the environment, with further differentiation being introduced in the model.

To sum up, agent-based modelling is an effective analytical technique that addresses some of the most common critiques to the standard economic approach to modeling phenomena.

4.3 Weaknesses of agent-based modelling

A number of shortcomings characterize ABMs. First of all, the result of single runs of Agent-based models cannot be interpreted as a general theory. Replication of these results over multiple model runs is required to develop more general theories. Also, testing the model over a broad range of parameter combinations is required to test the robustness of one's findings. This is actually the second problematic aspect of ABMs: the huge dimension of the parameter space, which makes it extremely difficult to define the whole set of potential model settings. In order to

¹³ Sections 4.2 and 4.3 draw upon Boero (2015)

solve the problem, the application of neural networks and genetic algorithms (see Section 2.2.3 and Section 3) is often recommended. These algorithms use learning and evolving techniques to explore the parameter space in a more efficient manner.

The great number of parameters that usually characterizes ABMs also makes it hard to understand which of these parameters is responsible for a particular result. Building the model in a step-by-step manner can partially solve this issue, as model features are added gradually, keeping complexity under control.¹⁴

Finally, modelers should be very careful with the results of a simulation: though unexpected findings may sometimes be caused by interesting model dynamics, more often they are the result of code errors and bugs.

¹⁴ See Cederman (1997)

5 THE SOFTWARE PACKAGES USED

5.1 NetLogo

NetLogo is an agent-based programming language used for modelling environment and simulating diverse phenomena. It is especially very useful for modelling complex systems.

The world of NetLogo consists of agents – “*turtles*” that can be of any size, shape, color or location and that follow the commands. The turtles move around the world, interact with each other and fulfill their tasks. “*Links*” connect the turtles one with another. Different types of turtles – “*breeds*” – can be defined. Turtles appertaining to different breeds can differ dramatically one from another. Studying the interactions between the turtles is extremely important for understanding their behavior and emerging patterns.

The world of turtles represents a grid of “*patches*”, as well belonging to the class of agents. Patches are square cells with coordinates. As other agents, they are settable by a modeler. The last agent in NetLogo is “*observer*”. In a model there could be only one observer. It could be an author or a user of the model. It does not have any location and its aim is to control the model and to give instructions to other agents.

The interface of NetLogo, shown in Figure 5.1, is represented by three tabs:

- 1) Interface;
- 2) Information;
- 3) Code.

The *interface tab* is where the model with its graphical elements is created. From here the user runs the model and can see how the agents behave in the world.

Nine visual interface elements can be created here:

- *Button* that launches the procedures either once or forever;
- *Slider* that sets the value of a global variable within a specified range;
- *Switch* that sets the value of a Boolean variable to either on or off;
- *Chooser* that selects the value of a global variable from a list of alternatives defined by a modeler;
- *Input* that allows a user to insert the value for a global variable;
- *Monitor* that shows the value of a variable, updated as a model runs;

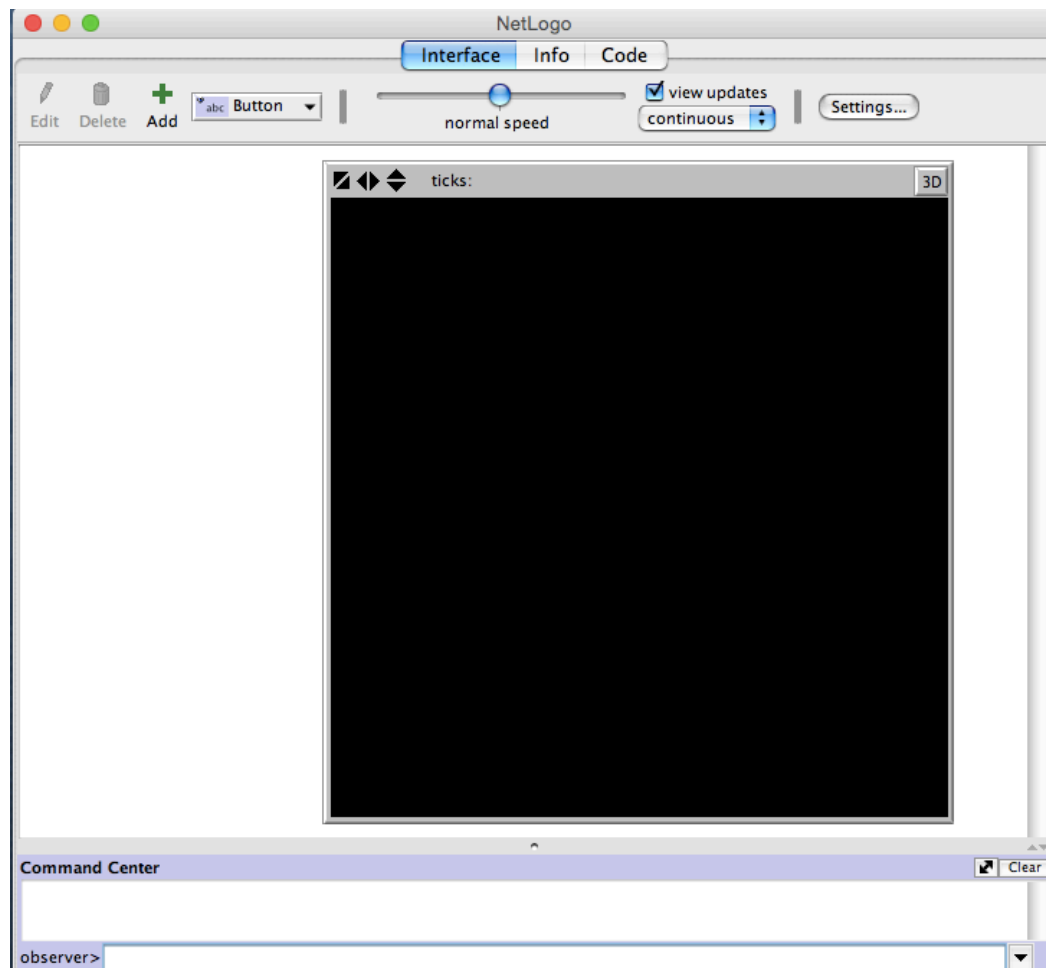


Figure 5.1 NetLogo interface

- *Plot* that makes a graph, updated as a model runs;
- *Output* that displays text, updated as a model runs;
- *Note* that displays a text label for user's convenience.

The *speed slider* allows the user to modify the speed with which the model runs. Some features of the world can be changed with a *setting* button. Such modifications are the location of origin, size of patch, font and frame and the display of tick counter. “*Tick*” is a kind of time unit in NetLogo. At each tick agents do what they were instructed to. Tick counter calculates the total number of ticks passed. Using *command center*, a user can send the instructions to observer, patches or turtles and receive the results right away.

The *information tab* contains information on the NetLogo model. In particular, a user can learn how to run, manipulate and understand the model. The information tab consists of 9 sections, among which there are “How it works”, “Things to notice” and “Extending the model”.

The *procedures tab* contains the rules defining features, behavior and interactions between the agents. Basically, it is the programming code of *primitives*. Primitives are built-in NetLogo

procedures determining the actions of agents. Each type of agent has its own primitives. What can be used for patches is not suitable for turtles. There are two categories of primitives: *commands* and *reporters*. Commands tell agents what to do, whereas reporters calculate a value and then report it back as an output. All of the primitives are described in NetLogo dictionary.

5.2 BehaviorSearch

A NetLogo model that has a lot of parameters is getting very complex and can lead to various behaviors and results. BehaviorSearch is a tool that is able to find the parameter settings meeting the objectives of a modeler. It uses special techniques, such as genetic algorithms, to search the parameter space in an efficient way.

Basically the procedure is the following: the user specifies the parameters of interest and its ranges and the objective function, then selects a heuristic search algorithm and decides whether the objective function should be minimized or maximized. BehaviorSearch runs the search, finds the best result and, in addition, provides with a list of all the simulations executed.

The interface of BehaviorSearch, shown on Figure 5.2, is graphically divided on 3 sections.

In the first section the NetLogo model that will be optimized is chosen.

The screenshot shows the BehaviorSearch interface with the following components:

- Model Selection:** A text field containing "Some Model.nlogo" and a "Browse for model..." button.
- Parameter Specification:**
 - A text area containing: `["integerParameter" [0 1 10]]`, `["continuousParameter" [0.0 "C" 1.0]]`, and `["choiceParameter" "near" "far"]`.
 - A "Load param ranges from model interface" button.
- Search Method Configuration:**
 - A dropdown menu set to "StandardGA".
 - A table with parameters and values:

Parameter	Value
mutation-rate	0.03
population-size	50
crossover-rate	0.7
population-model	generational
tournament-size	3
 - A checkbox for "Use fitness caching" which is checked.
 - A dropdown for "Search Encoding Representation" set to "GrayBinaryChromosome".
- Objective / Fitness Function:**
 - A "Goal" dropdown set to "Maximize Fit...".
 - A "Collected measure" dropdown set to "AT_FINAL_STEP".
 - A "Fixed Sampling" dropdown set to "10".
 - A "Combine replicates" dropdown set to "MEAN".
 - Two checkboxes: "Take derivative?" (unchecked) and "Use ABS value?" (unchecked).
 - A "w.r.t." dropdown set to "----" and a "Δ=" field set to "0.0".
 - An "Evaluation limit" field set to "300" model runs.
 - A "BestChecking replicates" field set to "0".
 - A "Run BehaviorSearch" button.
- Setup and Measure:**
 - "Setup:" field set to "setup".
 - "Step:" field set to "go".
 - "Measure:" field set to "mean [energy] of turtles".
 - "Measure If:" field set to "true".
 - "Stop If:" field set to "count turtles > 1000".
 - "Step Limit:" field set to "100" model steps.

Figure 5.2 BehaviorSearch interface

The second section is where the parameter space and the ranges are specified and behavioral measures are designed. The parameters of interest can be load directly from the model selected above by clicking "*Load param ranges from model interface*". They can either be set to constant or have a range of allowable values with an increment step. The type of chosen parameters can be numerical or Boolean. The more parameters were chosen and the larger were their ranges, the greater will be the search space.

The objective function is a quantative measure of behavior the modeler is interested in exploring. Unfortunately, for some models of qualitative nature this task becomes very difficult to accomplish. However, as soon as the objective function is defined, some BehaviorSearch parameters should be specified:

- *Setup* – a NetLogo command that setup a model (to setup);
- *Step* – a NetLogo command that is executed at each tick (to go);
- *Measure* – the objective function;
- *Measure if* – a condition defining when the measure is collected;
- *Stop if* – an optional condition obliging the model stop;
- *Step limit* – how many times the step command should be run.

The third section is the design of the objective function and the following specification of the search algorithm. First of all, the additional information on the qualitative measure of interest is required:

- *Goal* – whether the objective (fitness) function should be maximized or minimized;
- *Collected measure* – during one model run a measure can be picked several times:
 - *At final step* – the last recorded measure;
 - *Mean across steps* – the average of measures taken during the ticks defined in the second section;
 - *Median across steps* – the median of measures taken during the ticks defined in the second section;
 - *Min across steps* – the least value of measure taken during the ticks defined in the second section;
 - *Max across steps* – the highest value of measure taken during the ticks defined in the second section;
 - *Variance across steps* – the variance of all objective function values taken during the

ticks defined in the second section;

- *Sum across steps* – sum of all objective function values taken during the ticks defined in the second section.
- *Sampling* – it is possible that one model run cannot give good results. For this not to happen, the number of model runs should be specified;
- *Combine replicates* – running the model several times (see Sampling) leads to multiple values of objective function. These results can be combined in 6 different ways: the average (*Mean*), the median result (*Median*), the lowest result (*Min*), the highest result (*Max*), the variance of the result (*Variance*) or the standard deviation of the results (*StDev*).
- *Take derivative* – it is an optional setting allowing to maximize or minimize the approximate derivative of the objective function with respect to a one of four parameters and define the amount of changes (delta). A modeler can decide to consider only positive values of reported difference by choosing “*Use ABS value*”;
- *Evaluation limit* – it is the total number of times the model should be run before stopping;
- *BestChecking replicate* – since ABMs are usually random, some noises can occur. As a result, the best results obtained could be due this random noise. In order to be sure in the unbiasedness of results, this optional setting can be used.

As soon as the objective function is completely designed, the search algorithm and search space should be specified. The scope of search algorithm is to obtain the quantitative measure of behavior of interest. In particular, it fixes the order of different parameter settings trials. BehaviorSearch offers four different search algorithms are available:

- *RandomSearch* is a basic search algorithm that randomly tries various sets of parameters and then chooses the best explored. This algorithm is very simple and unbiased, though the search space it provides is not most effective;
- *MutationHillClimber* is a search algorithm that starts with a random set of parameters (location), mutates it and moves to a new set only if it performs better than the previous one. The modeler must specify the probability of mutation (*mutation-rate*) and the number of attempts after which the algorithm must select a new location if it was trapped (*restart-after-stall-count*);
- *SimulatedAnnealing* is a search algorithm that is similar to MutationHillClimber, but unlike it, SimulatedAnnealing can also move to inferior location with a certain

probability. This probability is derived from the temperature of the system that decreases as the system evolves. As in *MutationHillClimber*, the modeler must specify *mutation-rate* and *restart-after-stall-count*. Additionally, there are 2 more settings: original system's temperature (*initial-temperature*) and the number reflecting temperature changes after each move, that must be less than 1 (*temperature-change-factor*);

- *StandardGA* is a classic genetic algorithm. The model must specify the probability of mutation (*mutation-rate*), number of individuals in each generation (*population-size*), the probability of using two parental individuals to create an offspring (*crossover-rate*), one of three available *population-model* (*generational* – if the whole population is replaced at once, *steady-state-replace-random* – if a single individual is replaced or *steady-state-replace-worst* – if a current worst individual is replaced) and the number of individuals competing to be selected, which is usually 2 or 3 (*tournament-size*).

The search space is a collection of all possible combinations of model parameters' settings. There are four search space encodings:

- *StandardBinaryChromosome* – is an encoding where each parameter is represented as a string of binary digits. All the strings are combined together into one large bit array;
- *GrayBinaryChromosome* – is an encoding where each parameter is converted into a string of binary digits using a Gray code, which is considered to give better results;
- *MixedTypeChromosome* – is an encoding where each parameter keeps its own data type and the mutation occurs to each parameter separately;
- *RealHypercubeChromosome* – is an encoding where each parameter is described by a real-valued continuous variable.

The last thing to do before running the simulation is to decide whether to *use fitness catching*. This feature is mostly used while running a simulation of a complex model. In particular, fitness catching allows to the search algorithm to memorize all the evaluation results of the objective function attained during the simulation. Thus, if in the course of simulation the search algorithm returns to the parameter settings it has already evaluated, it does not recalculate it.

After all simulation settings are selected, but before executing the search, the modeler has to decide how many times the search process should be repeated – more searches can improve the results. When the search is running, *BehaviorSearch* provides a graph with real-time search progress. After the search has finished, an .csv file with the model run history is created.

6 RESEARCH DESIGN

6.1 Data

The data used in this research was kindly provided by Cerved S.p.A. The dataset contains the 2010 financial statement of randomly selected Italian firms. Overall there are 1013 corporations (limited liability companies) operating in the industrial sector, which are located in eight regions of North-East and North-West Italy, in particular: Emilia-Romagna, Friuli-Venezia Giulia, Liguria, Lombardia, Piemonte, Trentino-Alto Adige, Valle d'Aosta and Veneto. The distribution of the companies in the regions can be seen on the Figure 6.1.

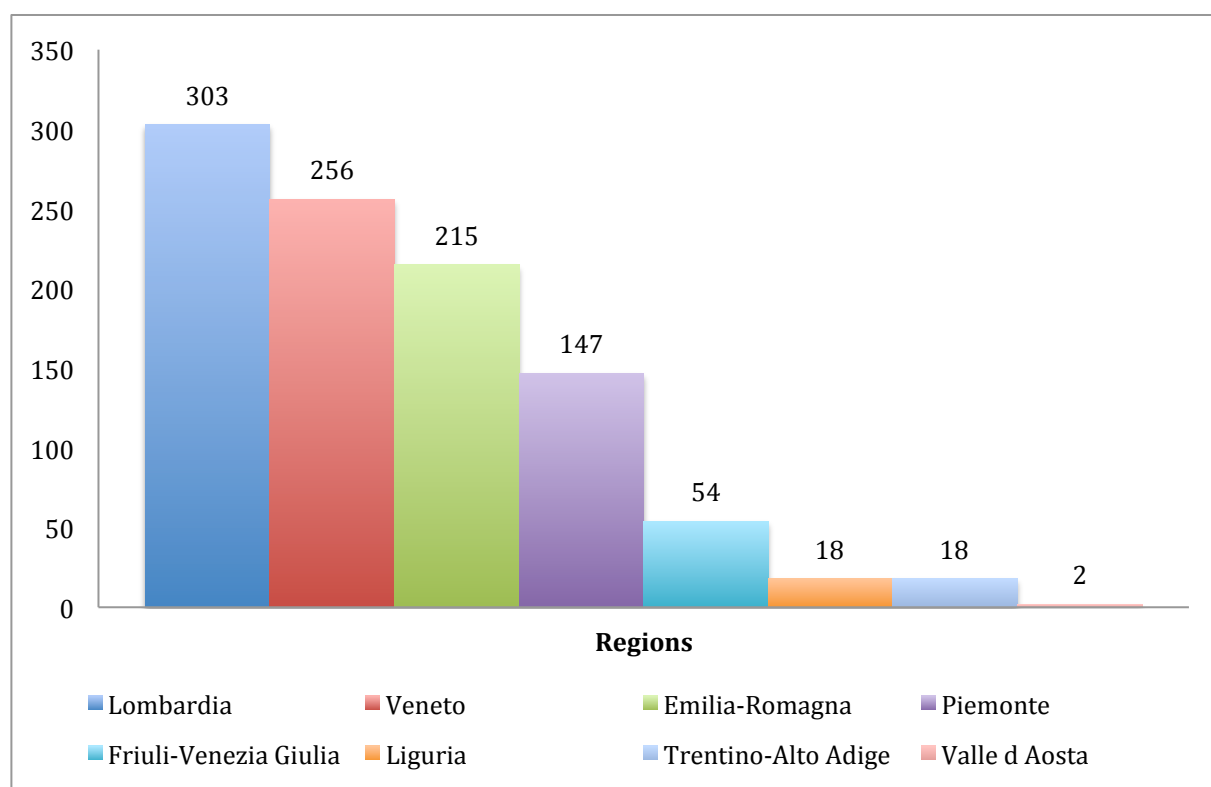


Figure 6.1 The distribution of firms by region

The breakdown of the firms by size in the data sample is reported in Table 6.1. It is based on the European Commission categorization, according to which the firm's size definition could be determined by its annual revenues. Enterprises with annual revenues ≤ 2 million € are considered micro, with ≤ 10 million € – small, with ≤ 50 million € – medium and with > 50 million € – large. Micro firms are excluded from the data sample. Furthermore, small companies account for more than 67 % of the firms represented in the sample.

All the 1013 firms in the dataset are considered trusted by Cerved. That is the companies with zero financial borrowings, zero bank financing and zero financial expenses are excluded from the data.

Type	Annual revenues (in million euros)	Number of firms
Small firms	≤ 10	687
Medium firms	≤ 50	251
Large firms	> 50	75

Table 6.1 Distribution of firms by size

Besides the financial statements, the information about companies' future defaults is available. All the defaults have occurred in 2012, 2013 and 2014. Moreover, two types of default are recorded: legal default and broad default.

A company is considered to be in *legal default* when it is undergoing bankruptcy proceedings. These judicial procedures are significant events in a firm's life, since they can result in a firm ceasing its activities. Because of the difficulty of surviving these proceedings, companies usually experience legal defaults only once throughout their lifetimes.

Broad defaults, instead, refer not only to legal defaults but also to situations where a firm's financial difficulties have resulted in the issuance of court orders to pay and judicial bonds – outside of formal bankruptcy proceedings, though. This latter type of situations may go on for multiple years and they do not prevent a company from continuing its operations. If a firm is considered to be in legal default, it is also in broad default, but not vice versa – broad default does not lead to legal default.

For the forthcoming simulations a firm considered to be insolvent if it files for legal or broad default. Of the total 1013 companies in the dataset, 109 of them experienced one of the two types of default in 2012, 2013 or 2014. In particular, 91 firms were at least once in legal default and 109 firms – in broad default. Table 6.2 reports how the number of insolvent firms was changing in each accounting period.

Year	Broad defaults initiating	Legal defaults initiating
2012	35	20
2013	49	40
2014	25	31
Total:	109	91

Table 6.2 The number of insolvent firms

In addition, according to Italian law, firms could submit either *balance sheet in normal form* ("Il bilancio ordinario") or *in abbreviated form* ("Il bilancio abbreviato"). Drawing up the normal balance sheet could be costly for small firms. Moreover for such companies it may even

be unnecessary to provide the same information that big corporations are required to compile. For these reasons, a small firm has the possibility to draw up the *balance sheet in abbreviated form* if during the first year of business or for the two subsequent years it has not passed two of three thresholds: a) 4 400 000 € of total current assets; b) 8 800 000 € of proceeds of sales or loans; c) 50 employees. In the dataset there are 220 firms with abbreviated balance sheets. In these cases, the supplementary note is not sufficiently detailed to allow separation between financial and nonfinancial debt. Therefore, in this study an estimate of financial debt based on interest paid is used.

6.2 Data preparation

The dataset provided by Cerved is in .csv format (comma separated values) which is usually used for import and export of spreadsheets and databases. The problem is that NetLogo cannot read an .csv file. For this reason, it is crucial to convert it into .txt format using Python. The following changes were done in Python in order to avoid any problems with reading of the dataset with NetLogo:

- Removal of headers;
- Replacement of semicolon by space;
- Addition of indexes for every observation;
- Putting strings in inverted commas.

The Python code is provided in Appendix A1.

6.3 Choice of financial indicators

The choice of the financial variables for credit scoring models is a very important part in the model development since they determine the value of credit score. Numerous studies have been done in order to determine what financial indicators can help predict firms' insolvency. For example Beaver in 1966 suggested that the most useful variable is cash flow to total debt ratio, whereas liquid asset ratios do not exhibit high predictive power.

Unfortunately, there is still no unique guideline for determining the “right” variables and the motivation for choosing them. For these reasons the selection of the financial indicators for this research will be mostly based on literature analysis. 14 papers regarding credit scoring models were analyzed. The financial ratios used in these studies are presented in Appendix A2.

After that, the financial indicators that will be used for discriminate the healthy firms from the insolvent ones were selected on the basis of theoretical reasoning. A lot of variables used in

these studies are similar to each other – they capture the same economic meaning. Moreover, it is computationally impossible to try all the ratios – it cannot guarantee an efficient and accurate credit scoring model and it is very time-consuming for the softwares used.

In order to choose the most suitable financial variables, meaning of each indicator, its repetition in the previous studies, theoretical similarity to other indicators and the availability of variables for its calculation in dataset were taken into account. The final 17 financial ratios are provided in Table 6.3. They all have one property in common – a higher value of any indicator has a positive impact on the firm's economic health.

Name of the ratio	Nº of repetitions in the studies	Definition of the financial ratio
Cash / Current liabilities (Cash ratio)	1	It measures a firm's ability to pay off its current liabilities with only cash and cash equivalents.
Cash / Sales	1	It indicates the effectiveness of the firm's credit and collection policies, and the amount of cash required as buffer for unexpected delays in cash collection.
Cash / Total assets	1	It measures the portion of a company's assets held in cash or marketable securities.
Current assets / Current liabilities (Current ratio)	3	It measures the ability of a firm to pay its near-term obligations.
EBIT / Sales	2	It indicates how much profit a firm makes after paying for costs of production such as wages, raw materials, equipment etc. (but before interest and tax).
EBITDA / Current liabilities	1	It indicates the resources available each period to the firm to pay off its short-term debt, ignoring the factors of interest, taxes, depreciation and amortization.
Equity / Current liabilities	1	It indicates how much short-term debt a company is using to finance its assets relative to the amount of value represented in shareholders' equity.
Equity / Total assets (Equity ratio)	1	It determines financial health and long-term profitability of a company and it measures the proportion of the total assets that are financed by stockholders, and not debt.
Long term funding / Fixed assets	1	It shows the proportion of fixed assets that are being funded by long term funding, comprising long term debt and shareholders' equity.
Return on assets (ROA)	4	It measures the amount of profit earned relative to the firm's level of investment in total assets.
Return on equity (ROE)	4	It measures how much profit is generated with the money shareholders have invested.
Sales / Current assets	1	It measures how well a company is making use of its assets in generating sales.

Sales / Total assets (Asset turnover)	2	It measures the value of a company's revenues generated relative to the value of its assets.
Total assets / Current liabilities	1	It measures a company's financial leverage and liquidity – it indicates how much short term debt a company is using to finance its assets.
Value added / Total assets	2	It gives an idea as to how efficient management is at using its assets to generate earnings (value added).
Working capital / Equity	2	It indicates the financial strength of a company.
Working capital / Total assets	5	Measures the company's ability to pay its short term financial obligations.

Table 6.3 Selected financial variables

6.4 The research method

In the research an agent-based credit scoring model will be created and genetic algorithms will be applied to it in order to search for optimal parameters.

The idea to create agent-based model is motivated by the facility of GAs application to it. When the model is created in NetLogo, a software BehaviorSearch may be use for the search of parameter-space and the model exploration that result in a required target behavior.

In the following sections several different credit scoring models will be created. GAs will be applied to each model with a scope to find the specific values of parameters of interest in order to maximize/minimize one of three objective functions. The fitness functions are the following:

- 1) $\alpha \cdot \text{Firms that received a loan and defaulted subsequently} + \beta \cdot \text{Firms that did not receive a loan and did not default}$ ($\alpha + \beta = 1$);
- 2) Profit;
- 3) Profit – lost business opportunities.

As soon as all the models will be developed, their performance in terms of earned profit, error rates and accuracy will be compared.

7 CREDIT SCORING MODEL №1

This simulation is the first attempt to create a credit scoring model. It is a preliminary model, where some of the concepts used are simplified. Overall, the model serves as a foundation for more sophisticated and advanced versions.

7.1 How it works

The interface of the NetLogo model is depicted in Figure 7.1.

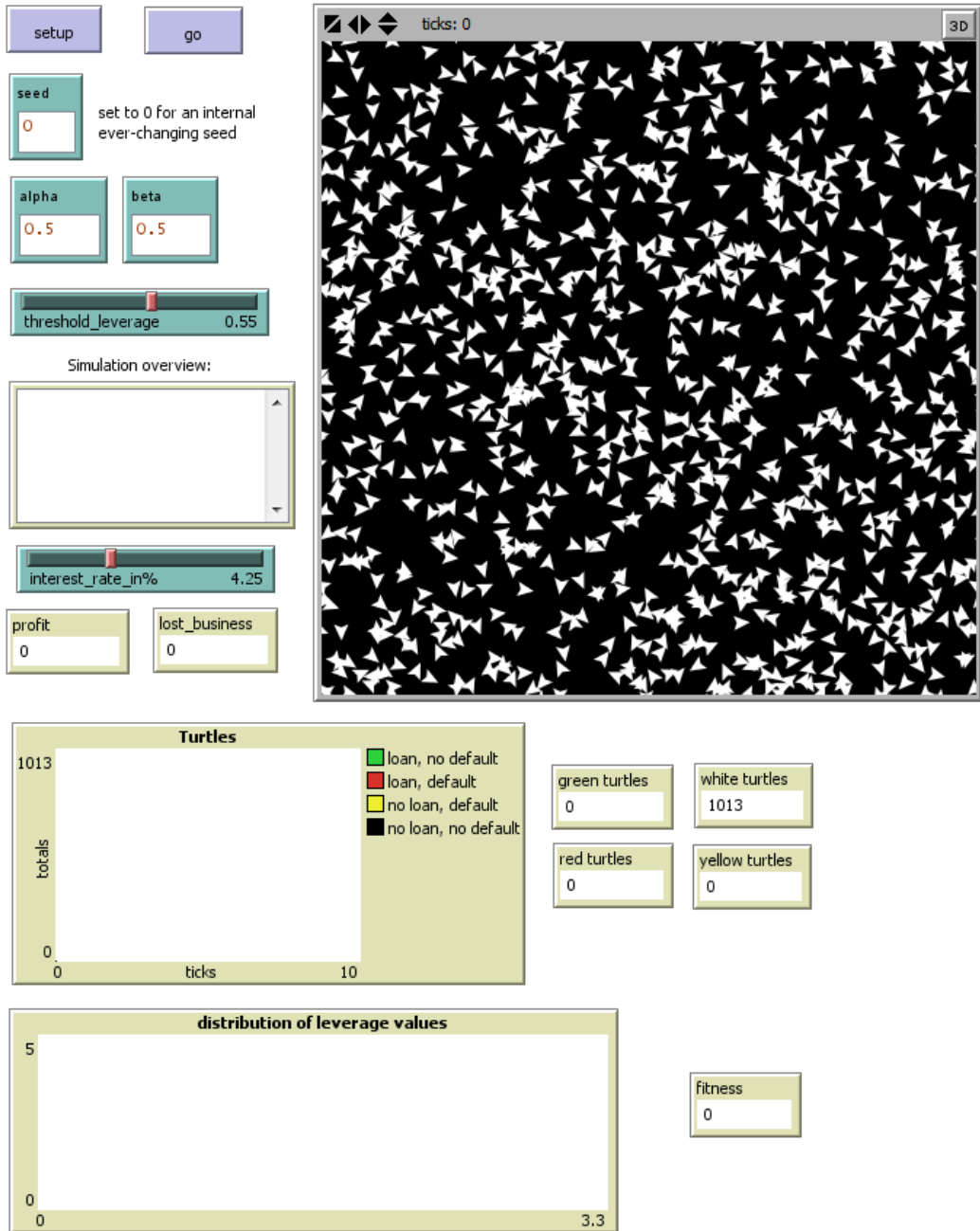


Figure 7.1 Setup of the credit scoring model №1

By clicking “setup”, 1013 firms, represented by randomly-located white turtles, are created. Each of them is endowed with parameters – information about each firm and its

financial situation obtained from balance sheet:

```
create-turtles 1013 [setxy random-xcor random-ycor set color 9.9]
```

The information about the firms is downloaded from the dataset in .txt format to NetLogo. Overall there are 135 parameters, including location of the firm, type of the balance sheet, information about future default and 122 financial variables:

```
file-open "datasetfornetlogo.txt"
while [not file-at-end?]
[ask turtle file-read [
  set codimp file-read
  set annorif file-read
  set datacost file-read
  set area_geografica file-read
  set regione file-read
  ...
  set V03001 file-read
  ...
  set V06044 file-read]
]
```

There is another parameter, which is created during the setup procedure – *leverage*. Since this is the very first version of the model, for initial simplicity leverage is considered to be the ratio of estimator of financial debts to total assets:

```
set leverage (debiti_finanziari_stimati / V03023)
```

The variable takes on a value from the range [0; 3.29]. The histogram of the leverage values is showed in Figure 7.2.

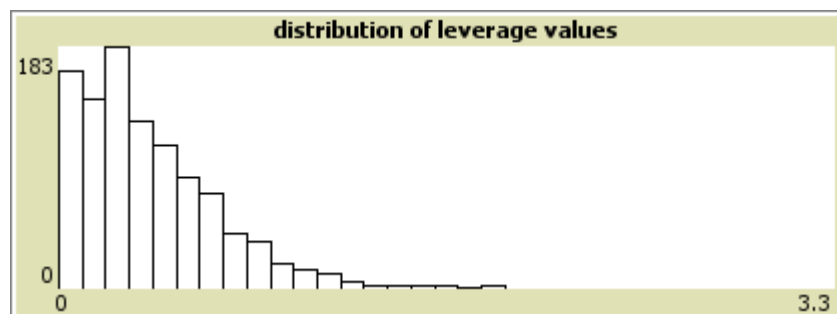


Figure 7.2 The distribution of the leverage values

Though the range is supposed to be [0; 1], there are 36 firms having the leverage higher than 1¹⁵. The value of this variable is particularly important – whether a firm receives a loan depends on its leverage.

¹⁵ Values greater than 1 are characteristic of firms with negative equity, i.e. situations where the accounting value of total assets has dropped below the book value of debt (e.g. due to fluctuations in the market value of the assets which have to be reflected in book values according to National Accounting Standards)

Parameters alpha and beta do not influence the run of the model, though they will be used for the optimization of the objective functions in the following sections.

The model proceeds for 4 ticks. Tick 0 represents 2010, the year when all the firms presented their balance sheets. Hence tick 1, 2 and 3 are 2012, 2013 and 2014. In these years, there is information about each firm's legal status. In particular, it can be seen whether a firm defaulted or not in.

```
to go
  if ticks <= 4 [
    if ticks = 0 [ setturtles ]
    if ticks = 1 [ default2012 ]
    if ticks = 2 [ default2013 ]
    if ticks = 3 [ default2014 ]
  ]
```

Threshold_leverage is a value of leverage above which a firm will not receive a loan. This value can be chosen with a slider. The values of threshold can vary from 0 to 1 with an increment of 0.05. At tick 0, turtles with leverage value lower than the threshold change the color from white to green. Green turtles are firms that were granted a credit, whereas white turtles are firms that have not been approved.

```
;Colors used: white=9.9; red=15; yellow=45; green=65
to setturtles
  ask turtles [
    set lev (debiti_finanziari_stimati / V03023)
    if lev <= threshold_leverage [set color 65]
  ]
end
```

At ticks 1, 2 and 3 turtles continue to change colors. If a firm that was granted a credit does not fail, it remains green, but if it fails, it changes its color from green to red. If a firm that was not granted a credit does not fail, it remains white, whereas if it fails, the color becomes yellow. After tick 4 nothing else changes in the model.

```
;Colors used: white=9.9; red=15; yellow=45; green=65
to default2012
  ask turtles [
    let index max (list default_esteso_2012 default_legale_2012)
    if index = 1 [
      if color = 9.9 [set color 45]
      if color = 65 [set color 15]
    ]
  ]
end

to default2013
  ask turtles [
    let index max (list default_esteso_2013 default_legale_2013)
    if index = 1 [
```



```

    if color = 9.9 [set color 45]
    if color = 65 [set color 15]
  ]]
end

to default2014
  ask turtles [
    let index max (list default_esteso_2014 default_legale_2014)
    if index = 1 [
      if color = 9.9 [set color 45]
      if color = 65 [set color 15]
    ]
  ]
end

```

During all the ticks, the number of turtles of each of the four colors is calculated. An example of turtle's evolution is shown in Figure 7.3. For these run threshold leverage was equal to 0.4. At first tick more than a half of the firms received a loan (619 green turtles). At the fourth tick only a small part of them failed (16 red turtles).

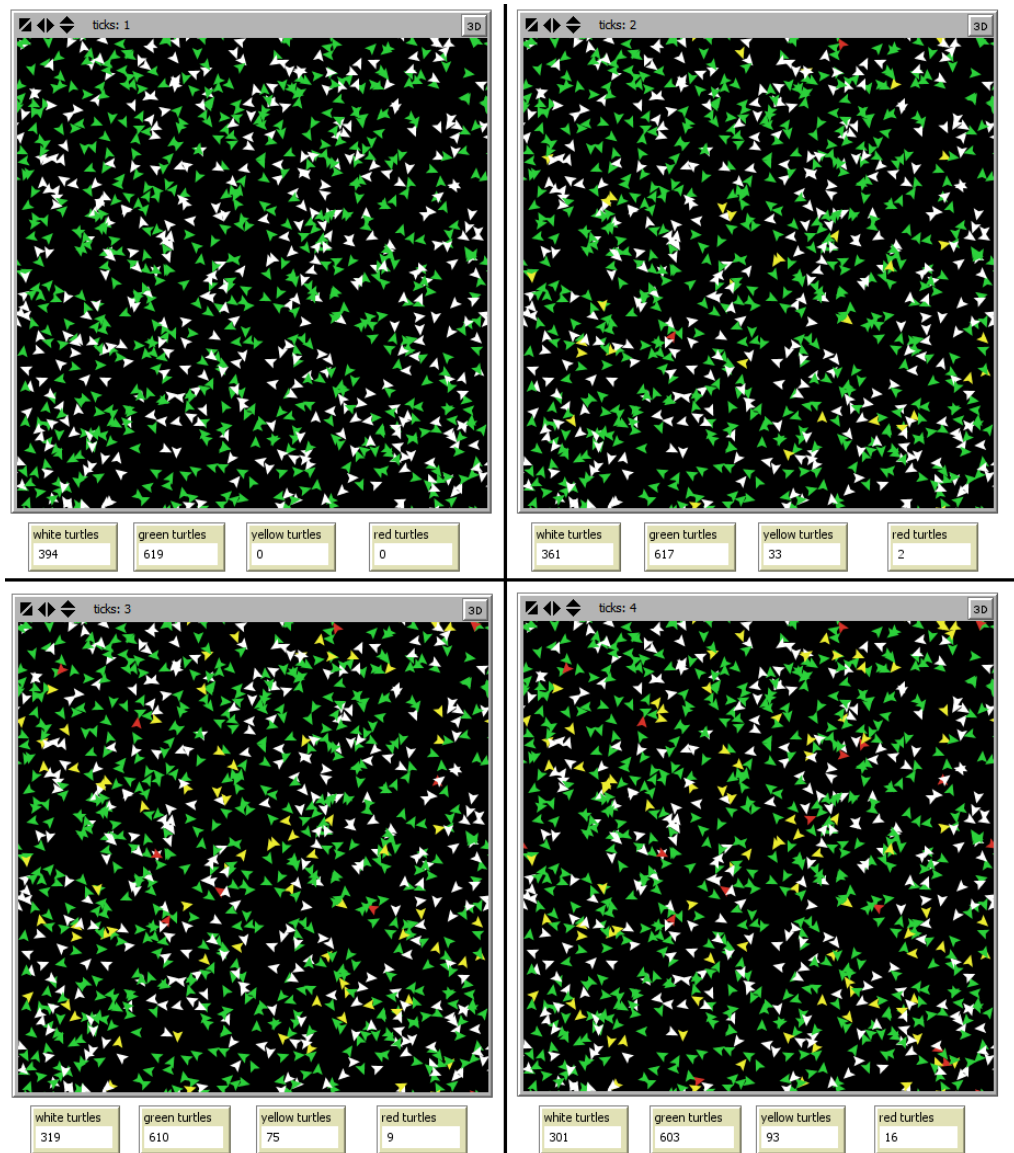


Figure 7.3 Evolution of the model as ticks pass

Based on these numbers, NetLogo plots the graph named “Turtles” showing how the quantity of turtles of each color changes during the ticks. In addition, following the credit scoring theory, NetLogo calculates the type I and type II errors¹⁶ as well as percentage of successful applicants and the overall accuracy. These four values are showed in the monitor after each tick. The output provided by NetLogo is showed on Figure 7.4.

```

;Calculating type I error in % - number of firms that were granted a loan but then went bankrupt
  output-type "typeIerror " output-print precision typeIerror 2
;Calculating type II error in % - number of firms that were not granted a loan but then did not go bankrupt
  output-type "typeIIerror " output-print precision typeIIerror 2
;Calculating % of successful applicants - number of firms received a loan to the total number of firms
  output-type "% of succesful applicants " output-print precision (100 * ((count turtles with [color = 65] + count turtles with [color = 15]) / count turtles )) 2
;Calculating the accuracy - number of firms correctly classified as solid as solid and unsolid
  output-type "% of correct decision " output-print precision (100 * ((count turtles with [color = 15] + count turtles with [color = 45]) / count turtles )) 2

```

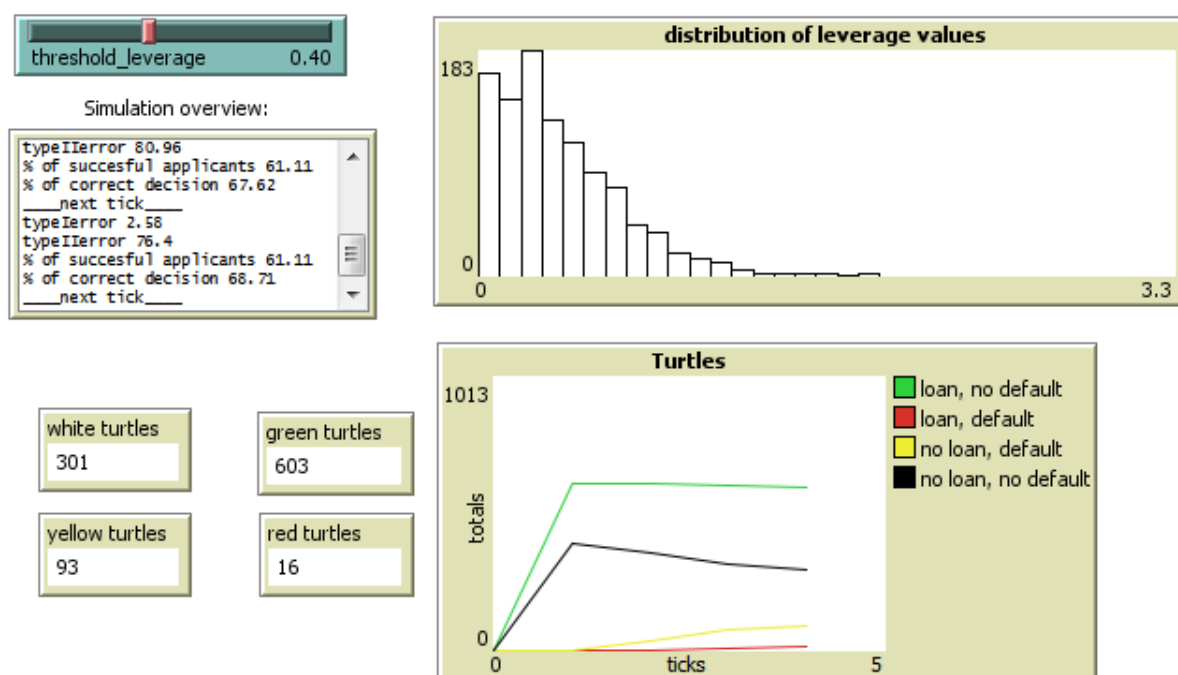


Figure 7.4 Output of the NetLogo simulation after the 4th tick

7.2 Experiments in BehaviorSearch

After the NetLogo model is fully designed, the genetic algorithm can be applied using BehaviorSearch¹⁷ software. The goal of GA is to find the specific values of parameters of interest that maximize or minimize the fitness function. The three objective functions are used:

¹⁶ See section 2.2.1

¹⁷ See section 5.2 for BehaviorSearch manual

- 1) $\alpha \cdot \text{Firms that received a loan and defaulted subsequently} + \beta \cdot \text{Firms that did not receive a loan and did not default (or } \alpha \cdot \text{Red turtles} + \beta \cdot \text{White turtles)}$;
- 2) Profit = gains – losses;
- 3) Profit – lost business opportunities.

After the simulations with different objective functions are executed, their outcomes will be compared.

(i) Fitness function №1

The first objective function is of the following form: $\alpha \cdot \text{count turtles with } [color = 15] + \beta \cdot \text{count turtles with } [color = 9.9]$.

In words, the fitness function is the sum of the firms that that were considered as solid but then failed (red turtles) and that were considered as unsolid but have not failed (white turtles). There are two coefficients α and β that should be specified by the user and that allow to give different weights to the amount of turtles of each color. It is important that the coefficients sum to 1: $\alpha + \beta = 1$.

The goal of BehaviorSearch is to choose the value of threshold_leverage in order to minimize the fitness function – that is to minimize the number of incorrectly classified healthy and unhealthy firms.

Since the NetLogo model runs only for four ticks, the 4th tick is crucial for BehaviorSearch analysis. All the measurement must be collected at the last tick, otherwise the results might be incorrect.

Before running BehaviorSearch, it is important to specify some search parameters (see Figure 7.5):

- *Measure*: $\alpha * \text{count turtles with } [color = 15] + \beta * \text{count turtles with } [color = 9.9]$;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: minimize function at final step.

Parameter Specification

```
[{"threshold_leverage" [0 0.05 1]}
{"alpha" 0.5}
{"beta" 0.5}
{"interest_rate_in%" 4.25}]
```

Load param ranges from model interface

Setup: setup

Step: go

Measure: fitness

Measure If: true

Stop If:

Step Limit: 4 model steps

Search Method Configuration

StandardGA

Parameter	Value
mutation-rate	0.03
population-size	50
crossover-rate	0.7
population-model	generational
tournament-size	3

☒ Use fitness caching

Search Encoding Representation

StandardBinaryChromos...

Objective / Fitness Function

Goal: Minimize Fitn...

Collected measure: AT_FINAL_STEP

Fixed Sampling: 10

Combine replicates: MEAN

☐ Take derivative? ☐ Use ABS value?

w.r.t. ---- $\Delta = 0.0$

Evaluation limit: 300 model runs

BestChecking replicates: 0

Run BehaviorSearch

Figure 7.5 The specifications of BehaviorSearch run

For the first experiment, alpha and beta were both set to 0.5. In the subsequent experiments the values of coefficients were not equal anymore. Setting the different values of allows assigning more importance to minimizing the amount of firms that were incorrectly classified as solid (red turtles), than to those that were incorrectly classified as unhealthy (white turtles) if $\alpha > \beta$ or vice versa if $\beta > \alpha$. Overall 12 such experiment were performed.

The 13th simulation differs from the first twelve – in additional to threshold, the GA has also to choose the values of alpha and beta that minimize the objective function. Since $\alpha + \beta = 1$ it can be assumed that $\beta = 1 - \alpha$. Then the fitness function №1 can be written as: $\alpha \cdot \text{count turtles with } [color = 15] + (1 - \alpha) \cdot \text{count turtles with } [color = 9.9]$.

The results of all 13 experiments with different values of coefficients alpha and beta are presented in Table 7.1.

It worth mentioning that not all of the results obtained were expected. The coefficients in trial №1 were set equal and the result could not be predicted. The obtained threshold_leverage = 1 means that loans were granted to almost all firms, in particular to more than 86% of applicants. Interestingly, in spite of the excessive type I error, the highest accuracy was achieved exactly in this trial.

№ of trial	Values of α and β	Threshold	Fitness	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
1	$\alpha=0.5; \beta=0.5$	1	60.5	880	97	24	12	9.93	88.06
2	$\alpha=1; \beta=0$	0	0	6	0	898	109	0	11.35
3	$\alpha=0.95; \beta=0.05$	0.35	29.7	538	12	366	97	2.18	62.69
4	$\alpha=0.9; \beta=0.1$	0.4	44.5	603	16	301	93	2.58	68.71
5	$\alpha=0.85; \beta=0.15$	0.45	57.35	652	23	252	86	3.41	72.85
6	$\alpha=0.8; \beta=0.2$	0.55	64.6	733	38	171	71	4.93	79.37
7	$\alpha=0.75; \beta=0.25$	0.55	71.25	733	38	171	71	4.93	79.37
8	$\alpha=0.7; \beta=0.3$	1	75.1	880	97	24	12	9.93	66.67
9	$\alpha=0.6; \beta=0.4$	1	67.8	880	97	24	12	9.93	66.67
10	$\alpha=0.3; \beta=0.7$	1	45.9	880	97	24	12	9.93	66.67
11	$\alpha=0.2; \beta=0.8$	1	36.8	880	97	24	12	9.93	66.67
12	$\alpha=0; \beta=1$	0.95	33	871	96	33	13	9.93	87.27
13*	$\alpha=0.96; \beta=0.04$	0.3	28.48	479	12	424	97	2.44	56.86

Table 7.1 The results of the NetLogo simulations – fitness function №1

In trial №2, setting $\beta=0$ means minimizing only the amount of red turtles, which in turn implies very low threshold – exactly the same that BehaviorSearch suggested. In trials №3, №4, №5, №6 and №7 the value of alpha was gradually changing from 0.95 to 0.75. As the coefficient was decreasing, the threshold was rising, which is what was expected. However, it was noticed that while moving from $\alpha=1$ to $\alpha=0.95$, the threshold raised sharply from 0.05 to 0.35, though after it was moving more gradually.

Starting from trial №8 threshold was not changing and remained constantly equal to 1 for experiments №8, №9, №10 and №11. Moving from trial №7 to №8 the value of alpha has been changed only by 0.05, but the resulted threshold almost doubled. This is probably due to the fact that values of coefficients are moving towards equal values ($\alpha=0.5$ and $\beta=0.5$), though such an abrupt increase is surprising. Trail №12 assumes $\alpha=0$ which means minimizing only the number of firms that were not granted a credit and have not failed (white turtles). One could expect a high threshold and a great number of firms that receive a loan, which, in fact, coincides with the result explored by BehaviorSearch.

The outcome of the trial №13 is very similar to the one of the trial №3. It should be noticed that as the choice of coefficient alpha was delegated to GA, the algorithm did find the least fitness value, though the type I error is not the best one. In fact, the lowest type I error (trial №3)

was achieved by randomly choosing the values of coefficients alpha and beta.

Overall, in spite of the latter experiment, the applied GA is resulted to be sufficiently sensitive. Still, further experiments with other objective functions should be done.

(ii) Fitness function №2

Since the primary goal of a credit organization is maximizing its profit, the fitness function could be changed to the calculation of profit. The new fitness function is: $profit = gains - losses$.

To define the profit function, one, first of all, needs to determine gains and losses. Gains are calculated as total value of all credits granted to firms that have not failed multiply an interest rate. Since gains are calculated as cumulative sum of gains each year, it is very important to define *gains* as a global variable.

In order to decide upon the value of interest rate, the statistical database of Banca d'Italia¹⁸ was analyzed. Table 7.2 presents the 2011 bank interest rates on euro loans to non-financial corporations (new business) up to 250,000 €.

Observation date	Interest rate - loans to non-financial corporations - up to 250,000 euro - new business (in %)
Jan 2011	3,5732
Feb 2011	3,6524
Mar 2011	3,6945
Apr 2011	3,8604
May 2011	3,9646
Jun 2011	4,052
Jul 2011	4,3062
Aug 2011	4,473
Sep 2011	4,5061
Oct 2011	4,7495
Nov 2011	4,9187
Dec 2011	5,2757
Average:	4.25

Table 7.2 Interest rates on loans up to 250 000 euro in 2011

Thus, the average of 12 interest rates – 4.25% – will be chosen for NetLogo calculation of

¹⁸ Statistical database of Banca d'Italia, cube – [TTI30100] Bank interest rates on euro loans to non-financial corporations: new business
(link: <https://infostat.bancaditalia.it/inquiry/#eNory0wt3IOdXL1dAm2DQnxNDYwNDDQBwBRewZG>)

profit. It can easily be modified by the user with a slider “*interest_rate_in%*”. Actually, the interest rate can be seen as a weight coefficient with which a particular level of importance is given to gains. Losses are total value of all credits granted to firm that have failed. The profit is the difference between gains and losses and it is cumulative of three years (2012, 2013 and 2014):

```
if ticks >= 1 [
  let index sum [loan_amount] of turtles with [color = 65]
  set gains gains + index * (interest_rate_in% / 100)
  let losses sum [loan_amount] of turtles with [color = 15]
  set profit gains - losses ]
```

All the firms are endowed with a new parameter – *loan_amount*, which is for simplicity considered to be $\frac{1}{4}$ of total assets of a firm:

```
set loan_amount (V03023 / 4)
```

The parameters specified in BehaviorSearch are similar to what was used before. The coefficients alpha and beta are not included into the fitness function and so can be disregarded. The goal of GA is to choose the value of threshold that maximizes the profit (see Figure 7.6):

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step.

The result of the experiment in BehaviorSearch is presented in Table 7.3.

Thre shold	Profit	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
0.35	366 142.5	538	12	366	97	2.18	62.69

Table 7.3 The results of the NetLogo simulation – fitness function №2

Interestingly, the outcome of this simulation coincides with the results obtained in trial №3 using the first fitness function, which was the best result in terms of fitness and type I error. Though for now it is hard to say whether the result in terms of profit is bad or good. It can be done after trying the last objective function.

Parameter Specification

```
["threshold_leverage" [0 0.05 1]]
["interest_rate_in%" 4.25]
```

Load param ranges from model interface

Setup: setup
Step: go
Measure: profit
Measure If: true
Stop If:
Step Limit: 4 model steps

Search Method Configuration

StandardGA

Parameter	Value
mutation-rate	0.03
population-size	50
crossover-rate	0.7
population-model	generational
tournament-size	3

☒ Use fitness caching

Search Encoding Representation

StandardBinaryChromos...

Objective / Fitness Function

Goal: Maximize Fit...

Collected measure: AT_FINAL_STEP

Fixed Sampling: 10

Combine replicates: MEAN

☐ Take derivative? ☐ Use ABS value?

w.r.t.: ----- **Δ=** 0.0

Evaluation limit: 300 model runs

BestChecking replicates: 0

Run BehaviorSearch

Figure 7.6 The specifications of BehaviorSearch run – fitness function №2

(iii) Fitness function №3

Considering the importance of profit for credit organizations, it makes sense to try to calculate the lost business opportunities. In particular, one needs to consider the firms that were not assigned with a loan, but that did not go bankrupt. These firms are represented in the model by white turtles. The sum of the loans that could have been granted to these clients multiplied an interest rate (4.25%) gives the costs of lost opportunities. Like the variables *gains* and *losses*, *lost_business* is calculated as cumulative sum of lost opportunities in each year:

```
let ind sum [loan_amount] of turtles with [color = 9.9]
set lost_business lost_business + ind * (interest_rate_in% / 100)
```

The third fitness function is: profit – lost_business.

The goal of GA is to choose a threshold value in order to maximize the fitness function. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit – lost_business;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step.

The result of the experiment is presented in Table 7.4.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
0.55	295 050.3	163 782.1	733	38	171	71	4.93	79.37

Table 7.4 The results of the NetLogo simulation – fitness function №3

7.3 Comparison of the results obtained

The outcomes of the four simulations are compared and presented in Table 7.5.

Fitness function	Profit	Lost business	Type I error	Classification accuracy (%)
FF№1: $0.96 \cdot \text{Red turtles} + 0.04 \cdot \text{White turtles}$	-	-	2.44	56.86
FF№2: Profit	366 142.5	-	2.18	62.69
FF№3: Profit – lost business	295 050.3	163 782.1	4.93	79.37

Table 7.5 Comparison of the results of all configurations of model №1

As it can be seen, the best configuration in terms of profit is the one that uses *profit* as fitness function. This result was expected since GA was maximizing profit only in the second simulation. Besides, while using this objective function, the type I error is also very low (2.18).

Despite the 13 experiments were performed with the fitness function №1, only two configurations are included in the table (trials №3 and №13). The one with $\alpha=0.95$ and $\beta=0.05$ guarantees a lower type I error and a higher classification accuracy with respect to another trial.

Even though there is no unique best simulation, it is clear that the third fitness function performed worse than others, since it has the highest type I error. An explanation could be the fact that the costs of lost business opportunities are not real costs – they do not exist in reality since they are connected to potential gains. In spite of that, the classification accuracy is the highest one.

To sum up, the algorithm developed so far could seem too simple and it has to be expanded.

8 CREDIT SCORING MODELS №2.1 AND №2.2

In the first credit scoring model the score function was too simple and consisted of only one financial indicator (Estimated financial debt / Total assets). This was a very naïve assumption that should be changed in order to make the model more realistic. Indeed, there is probably no credit scoring model that uses only one financial indicator for decision making.

Taking into account the literature about credit scoring techniques, it is getting clear that there is no unique variable that can distinguish between sound and unsound firms. For these reason, 17 variables were selected in Section 6.3 (see Table 6.3).

The next step in exploring and expanding the model is to try to use these variables instead of the score used before. The new score function is of the following form:

$$Score = \frac{w_1}{w_1 + w_2 + \dots + w_n} i_1 + \frac{w_2}{w_1 + w_2 + \dots + w_n} i_2 + \dots + \frac{w_n}{w_1 + w_2 + \dots + w_n} i_n$$

Where:

i_i – financial indicator i ;

w_i – weight assigned to the financial indicator i .

It would be too complicated to choose all the financial indicators for the score calculation. Therefore, the two versions of credit scoring model will be created – each of them using a combination of 10 indicators at a time.

The interface of the NetLogo model is showed in Figure 8.1. As it can be seen, it has changes a little bit. There are 10 new inputs for coefficients w_i that must be inserted by the user from the range $[0; 10]$. The distribution of the score function will be different for two models, since different collections of indicators are used.

For both versions of the model three simulations (with three different fitness functions) will be run in BehaviorSearch.

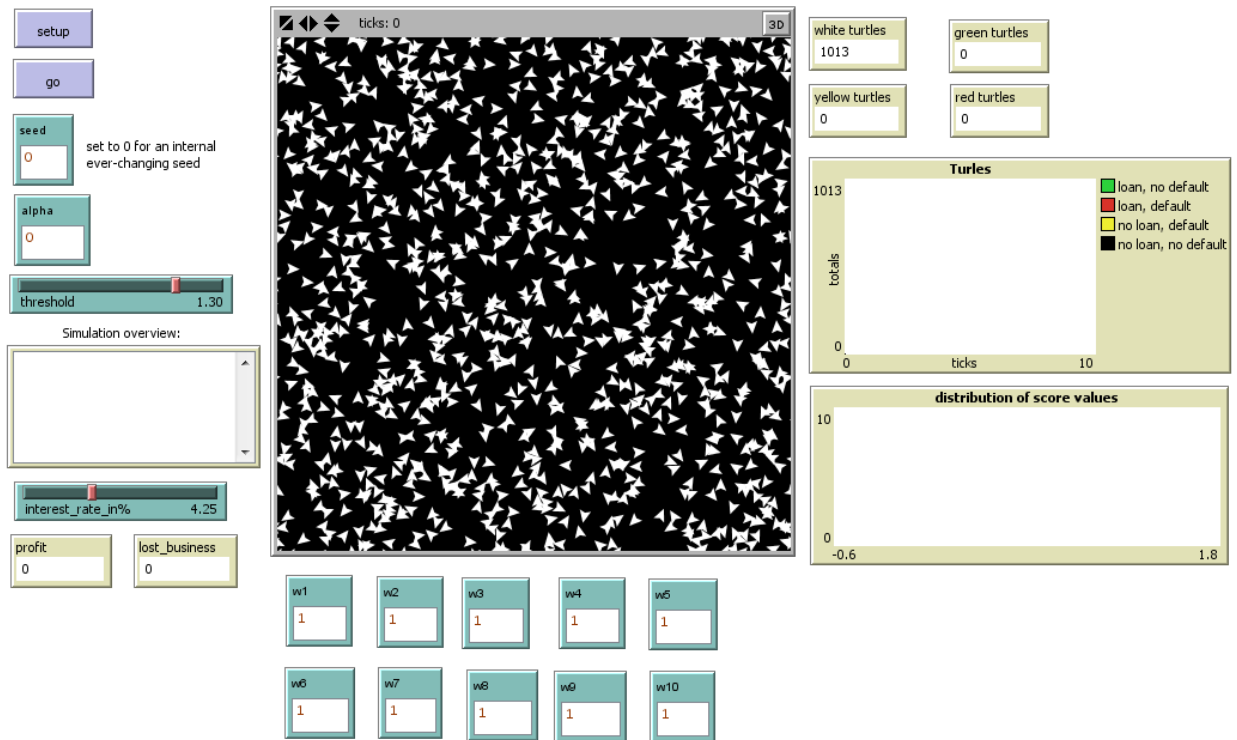


Figure 8.1 Setup of the credit scoring model №2

8.1 Model №2.1

The financial indicators used in the first experiment are listed in Table 8.1. The ratios were chosen in such a way as to have similar ranges of values.

i_i	Financial indicator	Minimum value	Maximum value	Mean
1	Cash / Total Assets	0,0000119402	0,7079295154	0,055165
2	Working capital / Total assets	-0,8575216972	0,8471866043	0,1208353
3	Cash / Current liabilities	0,0000234808	2,5453824812	0,143331
4	Cash / Sales	0,0000126699	1,2980769231	0,05624
5	EBIT / Sales	-1,1457100592	0,7742162733	0,031734
6	ROA	-1,4023625844	0,596875	0,035231
7	Equity / Total assets	-1,1457100592	0,8848114169	0,235671
8	Equity / Current Liabilities	-0,7292954265	8,5940594059	0,66363
9	Sales / Total assets	0,0769614425	4,8356512682	1,114965
10	EBITDA / Current liabilities	-1,4305684995	2,4384133612	0,162667

Table 8.1 Financial indicators used in model №2.1

Ten new variables representing the chosen financial indicators and the score are created in NetLogo. The granting decision rule differs from the previous model – a firm with the score higher than a threshold will receive a loan (it means that turtles become green):

```

to setturtles
ask turtles [
set loan_amount V03023 / 4

```

```

set cash/ta V03020 / V03023
set wc/ta ((V03022 - V04035) / V03023)
set cash/carliab V03020 / V04035
set cash/sales V03020 / V06001
set ebit/sales V06019 / V06001
set roa V06019 / V03023
set e/ta V04015 / V03023
set e/curliab V04015 / V04035
set sales/ta V06001 / V03023
set ebitda/curliab V06044 / V04035
set score (w1 * cash/ta + w2 * wc/ta + w3 * cash/curliab + w4 * cash/sales
+ w5 * ebit/sales + w6 * roa + w7 * e/ta + w8 * e/curliab + w9 * sales/ta
+ w10 * ebitda/curliab) / (w1 + w2 + w3 + w4 + w5 + w6 + w7 + w8 + w9 + w10)
if score >= threshold [set color 65]
]
end

```

Because of the particular form of the score function, the distribution of score values does not depend on the coefficients w_i . The histogram of score values is presented in Figure 8.2. The threshold has the range of values similar to the score function: $[-1; 2]$.

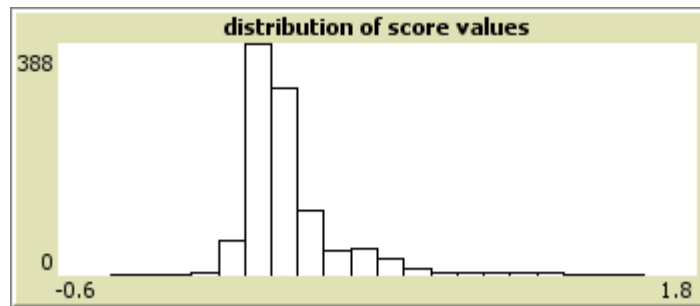


Figure 8.2 The distribution of the score values in model №2.1

The scope of GA for BehaviorSearch experiments is to find the values of the coefficients and of the threshold in order to achieve the best fitness value. As before, GA will use three different fitness functions. However, since the score function is more complex, it is necessary to perform two searches for each BehaviorSearch simulation. In fact, this strategy is advised by the application creators – repeating the search process two times can improve confidence, though it is very time-consuming.

(i) Fitness function №1

The first objective function is: $\alpha \cdot \text{count turtles with } [color = 15] + (1-\alpha) \cdot \text{count turtles with } [color = 9.9]$.

This time there is only one coefficient α and that should be specified by the user and that allows to give different weights to the amount of turtles of each color.

The goal of BehaviorSearch is to choose the values of threshold, of coefficients w_i and of α

in order to minimize the fitness function. The parameters specified in BehaviorSearch are the following (see Figure 8.3):

- *Measure*: $\alpha \cdot \text{count turtles with [color = 15]} + (1-\alpha) \cdot \text{count turtles with [color = 9.9]}$;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: minimize function at final step;
- *Number of searches*: 2.

The screenshot shows the BehaviorSearch configuration window with the following sections:

- Parameter Specification:** A text area containing a list of parameters and their ranges:


```
["threshold" [-1 0.01 2]]
["interest_rate_in%" 4.25]
["alpha" [0 0.01 1]]
["w1" [0 0.01 10]]
["w2" [0 0.01 10]]
["w3" [0 0.01 10]]
["w4" [0 0.01 10]]
["w5" [0 0.01 10]]
["w6" [0 0.01 10]]
["w7" [0 0.01 10]]
["w8" [0 0.01 10]]
["w9" [0 0.01 10]]
["w10" [0 0.01 10]]
```

 Below the text area is a button: "Load param ranges from model interface".
- Search Method Configuration:**
 - Algorithm: StandardGA (dropdown)
 - Table of parameters and values:

Parameter	Value
mutation-rate	0.03
population-size	50
crossover-rate	0.7
population-model	generational
tournament-size	3
 - Use fitness caching: ☒
 - Search Encoding Representation: StandardBinaryChromosome (dropdown)
- Objective / Fitness Function:**
 - Goal: Minimize Fitness (dropdown)
 - Collected measure: AT_FINAL_STEP (dropdown)
 - Fixed Sampling: 10 (input), replicates
 - Combine replicates: MEAN (dropdown)
 - Take derivative?: ☐ Use ABS value?: ☐
 - W.r.t.: ---- (dropdown) $\Delta =$ 0.0 (input)
 - Evaluation limit: 300 (input) model runs
 - BestChecking replicates: 0 (input)
 - Run BehaviorSearch button
- Setup:** setup (input)
- Step:** go (input)
- Measure:** fitness (input)
- Measure If:** true (input)
- Stop If:** (input)
- Step Limit:** 4 (input) model steps

Figure 8.3 The specifications of BehaviorSearch run – fitness function №1

Since all the financial indicators have the same sign in terms of economic-financial significance, the values of coefficients w_i can vary from 0 to 10 with an increment of 0.01. The negative values of coefficients are not allowed. The value of α is assumed to be in the range [0; 1] with an increment of 0.01.

The resulted values of coefficients w_i , α and threshold obtained with BehaviorSearch, are showed in Table 8.2.

Threshold	α	$1-\alpha$	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10
-0.27	0.02	0.98	2.76	7.03	3.31	3.67	8.52	6.98	3.34	9.53	6.25	0.73

Table 8.2 The results of the experiment in BehaviorSearch – fitness function №1

The results of the NetLogo simulation with the coefficients found by BehaviorSearch are presented in Table 8.3.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
-0.27	-125 747	0	904	108	0	1	10.67	89.34

Table 8.3 The results of the NetLogo simulation – fitness function №1

It turns out that the results of NetLogo simulation are not satisfying – the algorithm has assigned loan to almost all firms. In spite of that, the profit is negative, which is due to the fact that the failed firms received huge loans, whereas the gains from healthy clients were not high enough. The type I error in this configuration is very high.

(ii) Fitness function №2

The second objective function is: *profit = gains – losses*.

The parameters specified in BehaviorSearch are similar to what was used before. The coefficient alpha and is not included into the fitness function and so it can be disregarded. The goal of GA is to choose the value of threshold that maximizes the profit:

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The values of the threshold and the coefficients w_i found with BehaviorSearch are presented in Table 8.4.

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀
0.12	9.7	2.61	3.82	5.4	3.96	6.04	4.88	2.94	2.15	6.33

Table 8.4 The results of the experiment in BehaviorSearch – fitness function №2

The results of the NetLogo simulation with the values found by BehaviorSearch are presented in Table 8.5.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classificatio n accuracy (%)
0.12	381 818.1	175 027.7	565	27	339	82	4.56	63.87

Table 8.5 The results of the NetLogo simulation – fitness function №2

Looking at the results of the previous simulation, it is getting clear that in this experiment the profit is not negative anymore and it seems very significant and the value of type I error is low.

(iii) Fitness function №3

The last objective function is: profit – lost_business.

The goal of BehaviorSearch is to choose values of the coefficients w_i and of the threshold in order to maximize the difference between profit and lost business. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit – lost_business;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The values of the threshold and the coefficients w_i found with BehaviorSearch are presented in Table 8.6.

Threshold	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
0.18	7.33	7.37	1.98	9.4	1	5.76	6.44	0.24	8.11	2.24

Table 8.6 The results of the experiment in BehaviorSearch – model fitness function №3

The results of the NetLogo simulation with the coefficients found by BehaviorSearch are presented in Table 8.7.

Thre shold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classificatio n accuracy (%)
0.18	338 203.5	157 347.9	726	54	178	55	6.92	77.1

Table 8.7 The results of the NetLogo simulation – fitness function №3

The comparison of the results obtained after three experiments with different objective functions is presented in Table 8.8.

Fitness function	Thresh old	Profit	Lost business	Type I error	Classification accuracy (%)
FF№1: 0.02 * Red turtles + 0.98 * White turtles	-0.27	-125 747	0	10.67	89.34
FF№2: Profit	0.12	381 818.1	175 027.7	4.56	63.87
FF№3: Profit – lost business	0.18	338 203.5	157 347.9	6.92	77.1

Table 8.8 Comparison of the results of all configurations of model №2.1

The first objective function exhibited the worst performance – the loans were granted to all but one client. Type I error in this experiment achieved its highest value, not to mention negative profit. The classification accuracy is very high because almost all firms received a loan.

Using the second fitness function allows to achieve the highest value of profit and the lowest type I error. The last experiment neither provided better results nor had a bad performance as the first simulation.

8.2 Model №2.2

The indicators chosen for the second trial are listed in Table 8.9. Even though the ranges of values of financial ratios are quite similar, there are some indicators that can adopt values higher than before (for example “Working capital/Equity” and “Long term funding/Fixed assets”).

i _i	Financial indicator	Minimum value	Maximum value	Mean value
1	ROA	-1,4023625844	0,596875	0,035231
2	ROE	-8,75	34,5	0,052441
3	EBIT / Sales	-1,1457100592	0,7742162733	0,031734
4	Sales / Current assets	0,1263557006	8,5815789474	1,633208
5	Value added / Total assets	-0,531816566	1,7311111111	0,268243
6	Total assets / Current liabilities	0,64091471	19,98785837	2,104989
7	Working Capital / Equity	-56,1667	35,7727	0,3737125
8	Current assets / Current liabilities	0,1709662921	7,9603960396	1,380198
9	Equity / Current liabilities	-0,7292954265	8,5940594059	0,66363
10	Long term funding / Fixed assets	-7,0789473684	67,5	2,653785

Table 8.9 Financial indicators used in model №2.2

The indicators are defined in NetLogo. In the dataset there is a firm with zero equity ($V04015 = 0$). Since dividing on 0 is impossible, this observation should not be considered and the corresponding turtle should be eliminated:

```
to setturtles
  ask turtles[
    if V04015 = 0 [die]
    set loan_amount V03023 / 4
    set roa V06019 / V03023
    set roe V06037 / V04015
    set ebit/sales V06019 / V06001
    set sales/curass V06001 / V03022
    set va/ta V06009 / V03023
    set ta/curliab V03023 / V04035
    set wc/e (V03022 - V04035) / V04015
    set curass/curliab V03022 / V04035
    set e/curliab V04015 / V04035
    set ltfund/fixedass V04025 / V03010
    set score (w1 * roa + w2 * roe + w3 * ebit/sales + w4 * sales/curass + w5 * va/ta +
w6 * ta/curliab + w7 * wc/e + w8 * curass/curliab + w9 * e/curliab
+ w10 * ltfund/fixedass) / (w1 + w2 + w3 + w4 + w5 + w6 + w7 + w8 + w9 + w10)
    if score >= threshold [set color 65]
  ]
end
```

The NetLogo interface is the same as in model №2.1 – the only difference is in the values of the score function and the threshold. Since the ranges of values of four financial indicators described above are high, the distribution of the score values is larger than before. The histogram is presented in Figure 8.5. The threshold has the range of values similar to the score function: [-5; 8].

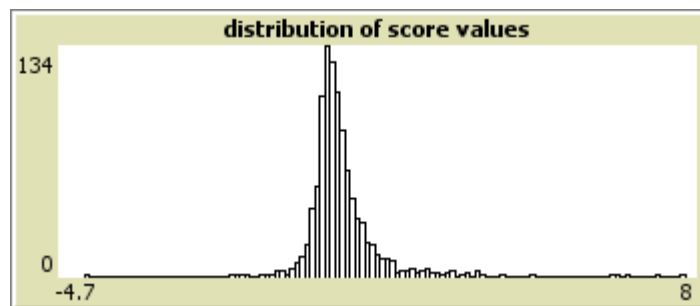


Figure 8.4 The distribution of the score values in model №2.2

In order to find the best fitness value, as before, GA will use three different fitness functions. For each BehaviorSearch simulation two searches will be performed.

(i) Fitness function №1

The first objective function is: $fitness = \alpha \cdot count\ turtles\ with\ [color = 15] + (1 - \alpha) \cdot count\ turtles\ with\ [color = 9.9]$.

The goal of BehaviorSearch is to choose values of coefficients w_i , α and threshold in order to minimize the fitness function. The parameters specified in BehaviorSearch are the following:

- *Measure*: fitness;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: minimize function at final step;
- *Number of searches*: 2.

Since all the financial indicators have the same sign in terms of economic-financial significance, the values of coefficients w_i can vary from 0 to 10 with an increment of 0.01. The negative values of coefficients are not allowed.

Two different BehaviorSearch simulations were run. In the first one α was in the range [0; 1] with an increment of 0.01 and in the second one – in the range [0.5; 1] with an increment of 0.01. The latter simulation was run in order to assign more importance to the minimization of red turtles and in case of obtaining $\alpha=0$ in the first experiment.

The values of coefficients w_i , α and threshold found by BehaviorSearch are presented in Table 8.10.

Threshold	Fitness	α	$1-\alpha$	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
-2.46	1	0	1	7.83	8.46	7.5	4.67	3.58	0.42	8.59	5.94	9.4	0.54
-4.58	108	1	0	3.02	3.91	9.5	0.89	0.24	1.44	6.49	5.49	4.42	3.65

Table 8.10 The results of the experiments in BehaviorSearch – fitness function №1

The results of two NetLogo simulations with values of coefficients, of α and of the threshold found by BehaviorSearch are presented in Table 8.11.

Threshold	α	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
-2.46	0	-126 599	1.3	902	108	1	1	10.69	89.23
-4.58	1	-126 599	1.3	902	108	1	1	10.69	89.23

Table 8.11 The results of the NetLogo simulations – fitness function №1

Two NetLogo simulations are the same – almost all firms receive a loan. These results are

very similar to what was found in model №2.1.

(ii) Fitness function №2

The second fitness function is: $profit = gains - losses$.

The goal of GA is to find values of coefficients w_i and threshold in order to maximize the profit. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The results of BehaviorSearch simulation are presented in Table 8.12

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀
0.64	6.39	2.69	3.11	5.04	2.57	0.68	9.77	5.62	9.79	5.07

Table 8.12 The results of the experiment in BehaviorSearch – fitness function №2

The results of a NetLogo simulation with the values of coefficients and threshold found by BehaviorSearch are presented in Table 8.13.

Thre shold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
0.64	308 380.5	246 986	558	38	345	71	6.38	62.15

Table 8.13 The results of the NetLogo simulation – fitness function №2

The results seem normal – profit is high and type I error is relatively low.

(iii) Fitness function №3

The last fitness function used is: $fitness = profit - lost_business$.

The goal of BehaviorSearch is to choose values of coefficients w_i and threshold in order to maximize the difference between profit and lost business. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit – lost_business;

- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The threshold and the values of coefficients w_i found with BehaviorSearch are presented in Table 8.14.

Threshold	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
-0.07	4.74	9.62	1.23	7.65	1.74	3.06	4.95	1.36	2.19	6.69

Table 8.14 The results of the experiment in BehaviorSearch – fitness function №3

The results of NetLogo simulation after applying these coefficients and the threshold are presented in Table 8.15.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
-0.07	201 143.2	3 271	890	100	13	9	10.1	88.83

Table 8.15 The results of the NetLogo simulation – fitness function №3

After all the simulations are run, their results should be analyzed. The comparison of the model's behavior with three different objective functions is presented in Table 8.16

Fitness function	Thres hold	Profit	Lost business	Type I error	Classification accuracy (%)
FF№1: White turtles	-2.46	-126 599	1.3	10.69	89.23
FF№2: Profit	0.64	308 380.5	246 986	6.38	62.15
FF№3: Profit – lost business	-0.07	201 143.2	3 271	10.1	88.83

Table 8.16 Comparison of the results of all configurations of model №2.2

The highest profit and the lowest type I error are achieved while fitness function №2 was used. The highest percentage of correct decision was achieved with the first objective function, even though this simulation results in negative profit and high type I error.

8.3 Comparison of models №2.1 and №2.2

To understand which of the two versions of the model performed better, it is crucial to

compare them. This task could also be important for determining the best collections of financial indicators. The comparison of the models is presented in Table 8.17.

Model №	Fitness function	Profit	Lost business	Type I error	Classification accuracy (%)
2.1	FF№1: $0.02 \cdot \text{Red turtles} + 0.98 \cdot \text{White turtles}$	-125 747	0	10.67	89.34
	FF№2: Profit	381 818.1	175 027.7	4.56	63.87
	FF№3: Profit – lost business	338 203.5	157 347.9	6.92	77.1
2.2	FF№1: White turtles	-126 599	1.3	10.69	89.23
	FF№2: Profit	308 380.5	246 986	6.38	62.15
	FF№3: Profit – lost business	201 143.2	3 271	10.1	88.83

Table 8.17 Comparison of the results of models №2.1 and №2.2

Analyzing the results more closely, several issues arise.

Firstly, the use of the first fitness function did not lead to valid results – in both models loans were granted to more than 99 % of firms. Using this objective function, the model №2.1 achieved the highest percentage of correct decision, but the model cannot be considered successful and applied simply because a credit granting organization probably will not find it profitable to approve all applicants. In fact, this choice leads to a negative profit. Thus, it is highly dubious if this configuration can actually be applied in reality. The same reasoning is applicable to the results obtained with the first fitness function in model №2.2,

Secondly, the best model in terms of profit and type I error is the model №2.1 model that uses profit as the objective function. Indeed, this simulation seems more promising – loans are granted to more than 50 % of companies, not to every firm, as happened with fitness function №1. The unique drawback is low percentage of correct decision, though there is another simulation with even lower one.

Nextly, it is evident that the model with the first collection of coefficients overall performed better.

Finally, the minimum type I error achieved in the models №2.1 and №2.2 (4.56) is higher than the one of the first credit scoring model (developed in section 7) that achieved the level of 2.18, which is twice less.

The credit scoring models developed in this section require more improvements and adjustments, such as changing the score function.

9 CREDIT SCORING MODELS №3.1 AND №3.2

Analyzing the results of the credit scoring models №2.1 and №2.2, it was found that the applied score function was not totally correct. In particular, since most of the financial indicators used in the previous sections may adopt negative values, the weighted average coefficients in the score function cannot be used. Besides, the function was too complicated to analyze by BehaviorSearch and this process was very time-consuming.

The new score function is:

$$Score = w_1 i_1 + w_2 i_2 + \dots + w_n i_n$$

Another drawback of the models was setting the variable `loan_amount` equal to $\frac{1}{4}$ of total assets. This was done because of the lack of information about loan amount granted to each firm. However, the latest Mediobanca report on cumulative data of 2055 Italian firms suggests other values. Since the firms in the dataset provided their balance sheets in 2010, loans were assigned to them only in 2011. The enterprises' Debt/Equity ratios in 2011 grounded on the report are showed in Table 9.1.

Type of the firm	Financial debt towards banking system in 2011 (as a % of equity)
All 2055 firms	33.6 %
Manufactures	38.3 %
Industries in the narrow sense	35.1 %
Industries	35.7 %
Average:	35.7%

Table 9.1 Amount of financial debt towards banks as a ratio of equity

It worth mentioning that firms in the dataset belong to the industrial sector and that is why it is reasonable to consider loan amount as equity multiply an average of four values from the table: 35.7%. In NetLogo it could be written as:

```
set loan (V04015 * 0.357)
```

The range of values of this variable is: [-1 658.04; 220 216.9]. It turned out that in the dataset there are 12 companies with negative equity and for such firms the loan amount would be negative, which does not make much sense. Thus, for the calculation of loan amount in NetLogo a more complex formula should be applied:

```
if V04015 = 0 [die]
set loan_amount min ( list (max list 10 (V04015 * 0.357)) 220216.9)
```

In words, it means that the variable `loan_amount` adopts the least of the two numbers: *max*

list 10 loan) and 220216.9. The first number is either equal to 10, which is the minimum amount of loan a firm can ask for, or to a value of loan variable of a firm. The second number is simply the maximum value of loan variable. This calculation guarantees a positive value of loan for firms with negative equity. Besides, since all the 12 companies with negative equity are small firms (revenues ≤ 10 million €), it makes sense for them to ask for a small amount of money, for example 10 000 €.

All the numbers are in thousands of euros. Thus, the minimum loan is 10 000 € and the maximum loan is 220 216 000.9 €. More than 80% of firms received a loan below 2000 (i.e. 2 000 000 €). The histogram of *loan_amount* values above 2000 is showed in Figure 9.1. the mean of the variable is 2629.2.

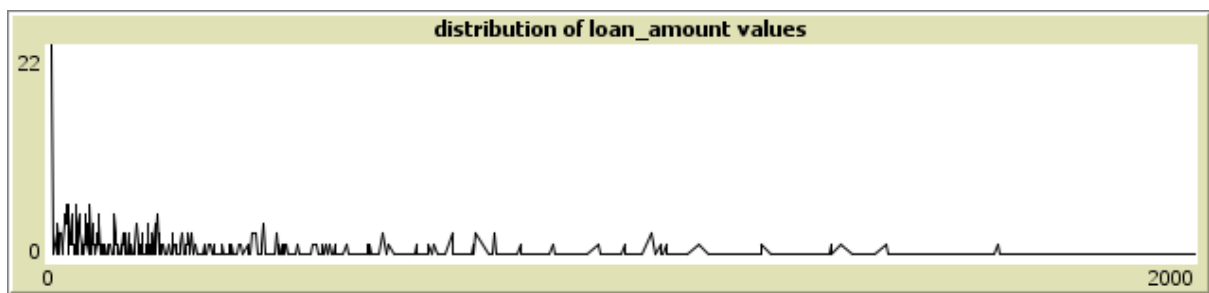


Figure 9.1 The distribution of *loan_amount* values above 2000

The full code is provided in Appendix A3.

Another important adjustment of the model is the normalization of the values of financial indexes. All 17 financial ratios have different ranges of values and scaling them may improve the performance of the model.

The last new feature of the third credit scoring model is synergies between several financial indicators. Using the synergies, could allow the model could to go outside the linearity. The score function will become:

$$Score = w_1 * i_1 + w_2 * i_2 + \dots + w_{10} * i_{10} + w_{11} * i_x * i_y$$

Overall the novelties of the scoring model №3 are the following:

- New score function;
- New method of loan amount calculation;
- Normalization of financial indicators;
- Synergies of the financial indicators.

As the previous section, two models will be developed: model №3.1 and model №3.2.

Each of them will be done with non-scaled and scaled financial variables. Hence, the experiments in BehaviorSearch will be performed for four directions:

- Model №3.1 with non-normalized financial indicators;
- Model №3.1 with normalized financial indicators;
- Model №3.2 with non-normalized financial indicators;
- Model №3.2 with normalized financial indicators.

Two synergies will be added to every best-performed fitness function in each of the four configurations presented above. Each synergy will be done for a pair of indicators that will be chosen on the basis of their weight coefficient (w_i) obtained in previous BehaviorSearch simulation. In such a way the ratios in the first pair are the indicators with the two highest values of w_i , whereas the ratios in the second pair have the 3rd and 4th highest values of w_i .

In the end, the results of different configurations will be analyzed and compared, the final discussion will be provided.

9.1 Model №3.1 with non-normalized financial indicators

The ratios used in model № 3.1 are the same from model № 2.1 (see Table 8.1). The new score function is determined in NetLogo:

```
set score (w1 * cash/ta + w2 * wc/ta + w3 * cash/curliab + w4 * cash/sales
+ w5 * ebit/sales + w6 * roa + w7 * e/ta + w8 * e/curliab + w9 * sales/ta
+ w10 * ebitda/curliab)
```

This time the range of values of score function depends on coefficients w_i . The maximum range of score values is reached when all coefficients w_i are equal to 10. The histogram of the values is presented in Figure 9.2. Consequently, the threshold range of values is set to [-5; 130].

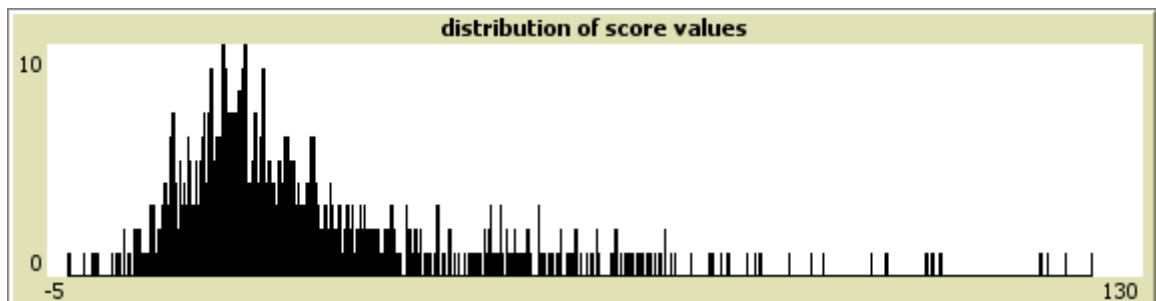


Figure 9.2 The distribution of the score values as $w_i=10$

(i) Fitness function №1

The first objective function is: $\alpha \cdot \text{count turtles with } [color = 15] + (1-\alpha) \cdot \text{count turtles with } [color = 9.9]$

The goal of GA is to find the values of alpha, of coefficients w_i and of the threshold in order to minimize the fitness function (see Figure 9.3). The parameters specified in the BehaviorSearch are the following:

- *Measure*: fitness;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: minimize function at final step;
- *Number of searches*: 2.

The screenshot displays the BehaviorSearch configuration window, divided into several sections:

- Parameter Specification:** A text area containing a list of parameters and their ranges:


```
[ "threshold" [-5 0.01 130]]
[ "interest_rate_in%" 4.25]
[ "alpha" [0 0.01 1]]
[ "w1" [0 0.01 10]]
[ "w2" [0 0.01 10]]
[ "w3" [0 0.01 10]]
[ "w4" [0 0.01 10]]
[ "w5" [0 0.01 10]]
[ "w6" [0 0.01 10]]
[ "w7" [0 0.01 10]]
[ "w8" [0 0.01 10]]
[ "w9" [0 0.01 10]]
[ "w10" [0 0.01 10]]
```

 Below this is a button "Load param ranges from model interface".
- Setup:** A section with input fields for "Setup:" (value: setup), "Step:" (value: go), "Measure:" (value: fitness), "Measure If:" (value: true), and "Stop If:" (empty).
- Step Limit:** A field showing "4" model steps.
- Search Method Configuration:**
 - Algorithm: StandardGA (dropdown).
 - Table of parameters and values:

Parameter	Value
mutation-rate	0.03
population-size	50
crossover-rate	0.7
population-model	generational
tournament-size	3
 - Checkboxes: "Use fitness caching" (checked).
 - Search Encoding Representation: StandardBinaryChromosome (dropdown).
- Objective / Fitness Function:**
 - Goal: Minimize Fitness (dropdown).
 - Collected measure: AT_FINAL_STEP (dropdown).
 - Fixed Sampling: 10 replicates (dropdown and input).
 - Combine replicates: MEAN (dropdown).
 - Checkboxes: "Take derivative?" (checked), "Use ABS value?" (unchecked).
 - w.r.t.: ---- (dropdown), $\Delta =$ 0.0 (input).
 - Evaluation limit: 300 model runs (input).
 - BestChecking replicates: 0 (input).
 - Run BehaviorSearch button.

Figure 9.3 The specifications of BehaviorSearch run – fitness function №1

The results of the BehaviorSearch simulations are presented in Table 9.2. As in the credit scoring model №1, two simulations were performed. In the first search the ranges of values of alpha was [0; 1] with an increment 0.01, whereas in the second search it was [0.5; 1] with the same increment.

Thres hold	Fitne ss	α	$1-\alpha$	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
0.76	8.2	0.05	0.95	6.05	2.8	0.32	8.37	4.15	6.73	1.52	0.24	9.42	8.83
56.99	9.01	0.99	0.01	6.56	0.1	6.75	5.55	2.3	3.94	6.08	3.48	4.73	6.16

Table 9.2 The results of the experiments in BehaviorSearch – fitness function №1

The results of two NetLogo simulations with the substituted values of coefficients w_i and α and of threshold, found by BehaviorSearch, are presented in Table 9.3.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classificati on accuracy (%)
0.76	151 158.4	809.4	900	107	3	2	10.63	89.13
56.99	534.2	321 160.7	2	0	901	109	0	10.97

Table 9.3 The results of the NetLogo simulations – fitness function №1

The results obtained after two simulations are exactly opposite. In the first experiment, only 5 of all applicants were rejected, whereas in the second one – only 2 of them were granted a loan. This is probably due to the big difference between values of alpha (0.05 and 0.99). However, since the goal was to minimize the fitness, the first simulation with $\alpha=0.05$ is the best one.

(ii) *Fitness function №2*

The second objective function is: $profit = gains - losses$.

The goal of GA is to find the values of coefficients w_i and threshold in order to maximize the profit. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The results of BehaviorSearch simulation are presented in Table 9.4.

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀
17.14	2.34	1.06	7.95	4.91	9.99	7.27	8.84	10	8.22	4.53

Table 9.4 The results of the experiment in BehaviorSearch – fitness function №2

The results of a NetLogo simulation with the values of coefficients and threshold found by BehaviorSearch are presented in Table 9.5.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classificatio n accuracy (%)
17.14	227 935.1	80 440.9	476	14	427	95	2.86	56.42

Table 9.5 The results of the NetLogo simulation – fitness function №2

(iii) Fitness function №3

The fitness function №3: *profit – lost_business*.

The goal of BehaviorSearch is to choose values of coefficients w_i and threshold in order to maximize the difference between profit and lost business. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit – lost_business;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The threshold and the values of coefficients w_i found with BehaviorSearch are presented in Table 9.6.

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀
10.68	0.76	3.4	7.13	6.56	3.73	0.16	8.14	9.72	5.45	6.73

Table 9.6 The results of the experiment in BehaviorSearch – fitness function №3

The results of NetLogo simulation after applying these coefficients and the threshold are presented in Table 9.7

Thre shold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classificatio n accuracy (%)
10.68	223 509.2	19 345.1	649	44	254	65	6.35	70.55

Table 9.7 The results of the NetLogo simulation – fitness function №3

The comparison of the simulations with three different objective functions is presented in Table 9.8.

Fitness function	Thresh old	Profit	Lost business	Type I error	Classification accuracy (%)
FF№1: 0.05 * Red turtles + 0.95 * White turtles	0.76	151 158.4	809.4	10.63	89.13
FF№2; Profit	17.14	227 935.1	80 440.9	2.86	56.42
FF№3: Profit – lost business	10.68	223 509.2	19 345.1	6.35	70.55

Table 9.8 Comparison of the results of model №3.1 with non-normalized indicators

According to the results, the highest profit and lowest type I error are achieved with the use of fitness function №2. The highest percentage of correct decision is obtained with the first objective function and $\alpha=0.05$. This simulation results in assigning loan to almost all clients, which is not a strategy the credit organizations could use.

Since the second fitness function seems to be the best for this configuration, two synergies will be applied to it. The indicators and weights resulted from the simulation in BehaviorSearch with fitness function №2 are listed in Table 9.9.

i_i	Financial indicator	w_i
1	Cash / Total Assets	2.34
2	Working capital / Total assets;	1.06
3	<i>Cash / Current liabilities</i>	7.95
4	Cash / Sales	4.91
5	<i>EBIT / Sales</i>	9.99
6	ROA	7.27
7	<i>Equity / Total assets</i>	8.84
8	Equity / Current Liabilities	10
9	<i>Sales /Total assets.</i>	8.22
10	EBITDA / Current liabilities	4.53

Table 9.9 Coefficients w_i found by BehaviorSearch using fitness function №2 – model №3.1 with non-normalized indicators

(iv) Synergy *EBIT/Sales* · *Equity/Total assets*

The first synergy score function is:

$$Score = w_1 * i_1 + w_2 * i_2 + \dots + w_{10} * i_{10} + w_{11} * i_5 * i_7$$

The score function is described in NetLogo in the following way:

```
set score (w1 * cash/ta + w2 * wc/ta + w3 * cash/curliab + w4 * cash/sales
+ w5 * ebit/sales + w6 * roa + w7 * e/ta + w8 * e/curliab + w9 * sales/ta
+ w10 * ebitda/curliab + w11 * ebit/sales * e/ta)
```

With the 11th coefficient the range of score values changes (before it was [-5; 130]). The histogram of the new values is showed in Figure 9.4. The threshold leverage now has the range [-50; 140].

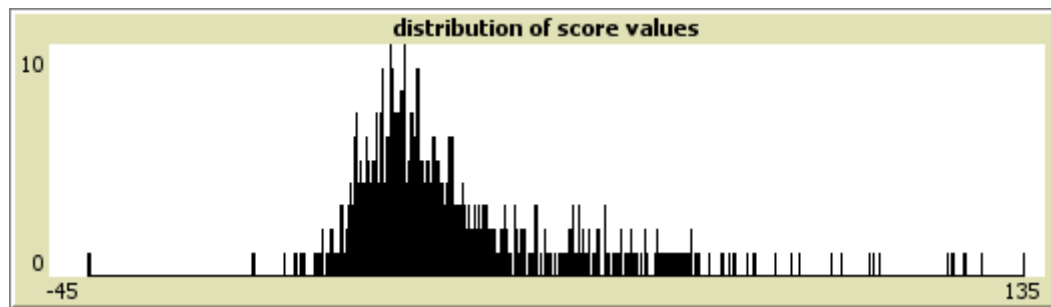


Figure 9.4 The distribution of the score values as $w_{11}=10$

The goal of GA has not change – BehaviorSearch has to find the values of coefficients w_i and threshold in order to maximize the profit. The parameters specified in BehaviorSearch are the following (see Figure 9.5):

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The results of BehaviorSearch simulation are presented in Table 9.10.

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀	w ₁₁
15.72	3.28	7.35	6.09	0.91	8.91	7.84	6.13	9.1	7.9	6.88	8.98

Table 9.10 The results of the experiment in BehaviorSearch

The results of a NetLogo simulation with the values of coefficients and threshold found by BehaviorSearch are presented in Table 9.11.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
15.72	239 452.1	57 509.2	519	21	384	88	3.89	59.98

Table 9.11 The results of the NetLogo simulation

The screenshot displays the BehaviorSearch interface with the following settings:

- Parameter Specification:** A list of parameters including "threshold" [-50 0.01 140], "interest_rate_in%" 4.25, and weights w1 through w11, each with a range [0 0.01 10].
- Search Method Configuration:** Set to "StandardGA" with parameters: mutation-rate (0.03), population-size (50), crossover-rate (0.7), population-model (generational), and tournament-size (3).
- Objective / Fitness Function:** Goal is "Maximize Fitness", Collected measure is "AT_FINAL_STEP", Fixed Sampling is 10 replicates, and Combine replicates is "MEAN".
- Other Settings:** "Use fitness caching" is checked, "Search Encoding Representation" is "StandardBinaryChromosome", and "Evaluation limit" is 300 model runs.

Figure 9.5 The specifications of BehaviorSearch run with the 11th coefficient

(v) Synergy Cash/Current liabilities · Sales/Total assets

The second synergy score function is:

$$Score = w_1 * i_1 + w_2 * i_2 + \dots + w_{10} * i_{10} + w_{11} * i_3 * i_9$$

The score function is described in NetLogo in the following way:

```
set score (w1 * cash/ta + w2 * wc/ta + w3 * cash/curliab + w4 * cash/sales
+ w5 * ebit/sales + w6 * roa + w7 * e/ta + w8 * e/curliab + w9 * sales/ta
+ w10 * ebitda/curliab + w11 * cash/curliab * sales/ta)
```

With the 11th coefficient the range of score values does not change much with respect to synergy A and the threshold leverage has the same range [-40; 150].

The goal of GA has not change – BehaviorSearch has to find values of coefficients w_i and threshold in order to maximize the profit. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The results of BehaviorSearch simulation are presented in Table 9.12.

Threshold	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}
3.87	6.6	1.48	1.52	7.48	0.71	0.49	5.17	0.55	2	0.6	2.8

Table 9.12 The results of the experiment in BehaviorSearch

The results of a NetLogo simulation with the values of coefficients and threshold found by BehaviorSearch are presented in Table 9.13.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classificatio n accuracy (%)
3.87	230 680.6	72 957.6	585	27	318	82	4.41	65.91

Table 9.13 The results of the NetLogo simulation

Table 9.14 provides the comparison of the results obtained in model №3.1 with non-normalized indicators, using the second fitness function, and of the results after two synergies.

Fitness function	Profit	Lost business	Type I error	Classification accuracy (%)
Model №3.1; non-normalized indicators; FF №2: Profit; No synergies	227 935.1	80 440.9	2.86	56.42
...+ Synergy $EBIT/Sales \cdot Equity/Total\ assets$	239 452.1	57 509.2	3.89	59.98
...+ Synergy of $Cash/Current\ liabilities \cdot Sales/Total\ assets$	230 680.6	72 957.6	4.41	65.91

Table 9.14 Comparison of the of the outcomes of synergies

As it results, there is no unique best model. The lowest type I error was achieved when GA was maximizing profit in model №3.1 and the inclusion of synergies did not improve it. Though, synergy of *EBIT/Sales* and *Equity/Total assets* performed well in terms of profit, whereas the synergy of *Cash/Current liabilities* and *Sales/Total assets* obtained the highest % of correct decisions. The unique conclusion so far is that in that synergies are able to improve the results of the main configuration.

9.2 Model №3.1 with normalized financial indicators

In section 9.1 the financial ratios were non-normalized. In Table 9.15 scaled financial ratios from the first collection of indicators are presented. After the normalization the ranges are getting more similar to each other – all the values of 10 ratios are in the range [0; 1].

i_i	Financial indicator	Range of values	Scale	Normalized range of values
1	Cash / Total Assets	[0,0000119402; 0,7079295154]	0,7079414556	[0,000016866; 0,999983134]
2	Working capital / Total assets	[-0,8575216972; 0,8471866043]	1,7047083015	[-0,5030313376; 0,4969686624]
3	Cash / Current liabilities	[0,0000234808; 2,5453824812]	2,545405962	[0,0000092248; 0,9999907752]
4	Cash / Sales	[0,0000126699; 1,2980769231]	1,2980895929	[0,0000097604; 0,9999902396]
5	EBIT / Sales	[-1,1457100592; 0,7742162733]	1,9199263325	[-0,5967468854; 0,4032531146]
6	ROA	[-1,4023625844; 0,596875]	1,9992375844	[-0,7014486899; 0,2985513101]
7	Equity / Total assets	[-1,1378977821; 0,8848114169]	2,022709199	[-0,5625612335; 0,4374387665]
8	Equity / Current Liabilities	[-0,7292954265; 8,5940594059]	9,3233548324	[-0,0782224252; 0,9217775748]
9	Sales / Total assets	[0,0769614425; 4,8356512682]	4,9126127108	[-0,3697532196; 0,6302467804]
10	EBITDA / Current liabilities	[-1,4305684995; 2,4384133612]	3,8689818607	[-0,3697532196; 0,6302467804]

Table 9.15 Normalized financial ratios – model №3.1

Calculation of financial indicators in NetLogo has changed – every ratio should be divided on its scale:

```

set cash/ta ((V03020 / V03023) / 0,7079414556)
set wc/ta (((V03022 - V04035) / V03023)) / 1.7047083015 )
set cash/curliab ((V03020 / V04035) / 2.5454059620)
set cash/sales ((V03020 / V06001) / 1.2980895929)

```



```

set ebit/sales ((V06019 / V06001) / 1.9199263325)
set roa ((V06019 / V03023) / 1.9992375844)
set e/ta ((V04015 / V03023) / 2.0227091990)
set e/curliab ((V04015 / V04035) / 9.3233548324)
set sales/ta ((V06001 / V03023) / 4.9126127108)
set ebitda/curliab ((V06044 / V04035) / 3.8689818607)

```

As before, the range of values of score function depends on the values of w_i . The maximum range of score values is reached when all coefficients w_i are equal to 10. The histogram of the values is presented in Figure 10.3. Thus, the threshold range of values is set to $[-8; 45]$.

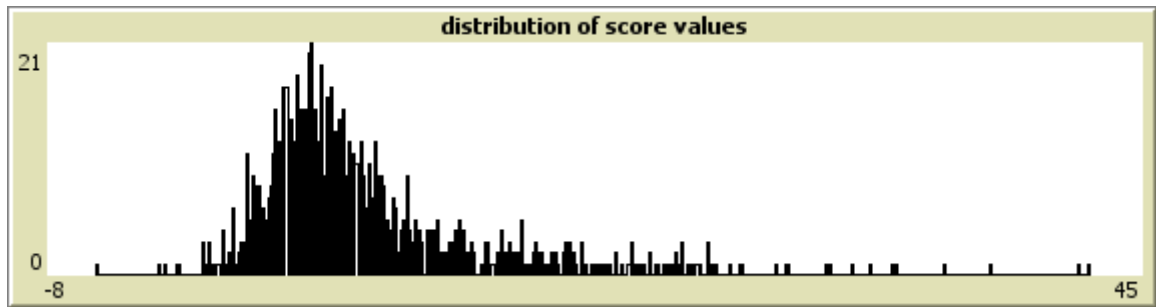


Figure 9.6 The distribution of the score values as $w_i=10$

(i) *Fitness function №1*

The first objective function is: $\alpha \cdot \text{count turtles with } [color = 15] + (1-\alpha) \cdot \text{count turtles with } [color = 9.9]$

The goal of GA is to find the values of α , of coefficients w_i , of α and of the threshold in order to minimize the fitness function. The parameters specified in the BehaviorSearch are the following:

- *Measure*: fitness;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: minimize function at final step;
- *Number of searches*: 2

The result of the experiment in BehaviorSearch is presented in Table 9.16. Two simulations were performed with the different ranges of values of α . In the first one α was in the range $[0; 1]$ and in the second one – in the range $[0; 0.99]$.

Thres hold	Fitne ss	α	$1-\alpha$	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
18.97	0	1	0	9.01	6.3	8.52	7.01	3.5	9.18	2.08	4.62	6.2	5.44
-4.38	4.32	0.04	0.96	2.3	2.85	7.17	4.97	8.83	4.75	4.96	7.87	7.59	0.9

Table 9.16 The results of the experiments in BehaviorSearch – fitness function №1

The results of the NetLogo simulation with values of coefficients w_i , α and threshold found by BehaviorSearch are presented in Table 9.17.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classificatio n accuracy (%)
18.97	2 915.2	318 779.8	10	0	893	109	0	11.76
-4.38	143 047.8	0	903	108	0	1	10.68	89.33

Table 9.17 The results of the NetLogo simulations – fitness function №1

As before, different values of alpha lead to absolutely different results. In the first simulation only 10 firms received a loan, but neither of them went bankrupt, whereas in the second experiment all but one companied received a loan. The simulation with $\alpha=1$ has the lowest fitness value. However, both simulations cannot be seen as valid since they did not provide good results.

(ii) Fitness function №2

The second objective function is: $profit = gains - losses$.

The goal of GA is to find values of coefficients w_i and threshold in order to maximize the profit. The parameter α is not used anymore. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The results of BehaviorSearch simulation are presented in Table 9.18

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀
3.11	1.91	6.05	2.51	2.57	0.82	6.87	5.58	7.27	8.19	5.44

Table 9.18 The results of the experiment in BehaviorSearch – fitness function №2

The results of a NetLogo simulation with the values of coefficients and threshold found by BehaviorSearch are presented in Table 9.19.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classificatio n accuracy (%)
3.11	219 525.3	72 982.8	594	34	309	75	5.41	66.11

Table 9.19 The results of the NetLogo simulation – fitness function №2

(iii) Fitness function №3

The last objective function is: profit – lost_business.

The goal of BehaviorSearch is to choose values of coefficients w_i and threshold in order to maximize the difference between profit and lost business. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit – lost_business;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The threshold and the values of coefficients w_i found with BehaviorSearch are presented in Table 9.20.

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀
2.48	4.54	6.64	0.79	4.13	1.61	4.59	9.91	0.51	4.77	4.96

Table 9.20 The results of the experiment in BehaviorSearch – fitness function №3

The results of NetLogo simulation after applying these coefficients and the threshold are presented in Table 9.21.

Thre shold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
2.48	215 678.4	33 022.6	664	45	239	64	6.35	71.94

Table 9.21 The results of the NetLogo simulation – fitness function №3

The comparison of the three simulations with three different objective functions is presented in Table 9.22.

Fitness function	Threshold	Profit	Lost business	Type I error	Classification accuracy (%)
FF№1: Red turtles	18.97	2 915.2	318 779.8	0	11.76
FF№2: Profit	3.11	219 525.3	72 982.8	5.41	66.11
FF№3: Profit – lost business	2.48	215 678.4	33 022.6	6.35	71.94

Table 9.22 Comparison of the results of model №3.1 with normalized indicators

For the first time the zero type I error is achieved while using the first fitness function. The absence of the error means that the algorithm assigns loan only to solvent clients. Unfortunately, in this case, only 10 of 1012 firms are approved – that is why the classification is very low.

As it can be seen, the model using fitness function №2 guarantees the highest profit and the best (not zero) type I error. The highest % of correct decision is achieved with the third fitness function, though other parameters are lower.

Since the second fitness function seems to be the best for this configuration, two synergies will be applied to it. The indicators and weights resulted from the simulation in BehaviorSearch with fitness function №2 are listed in Table 9.23

i_i	Financial indicator	w_i
1	Cash / Total Assets	1.91
2	<i>Working capital / Total assets;</i>	6.05
3	Cash / Current liabilities	2.51
4	Cash / Sales	2.57
5	EBIT / Sales	0.82
6	<i>ROA</i>	6.87
7	Equity / Total assets	5.58
8	<i>Equity / Current Liabilities</i>	7.27
9	<i>Sales / Total assets.</i>	8.19
10	EBITDA / Current liabilities	5.44

Table 9.23 Coefficients w_i found by BehaviorSearch using fitness function №2 – model № 3.1 with normalized indicators

(iv) Synergy Equity/Current liabilities x Sales/Total assets

The first synergy score is:

$$Score = w_1 * i_1 + w_2 * i_2 + \dots + w_{10} * i_{10} + w_{11} * i_8 * i_9$$

The first score function is described in NetLogo in that way:

```
set score (w1 * cash/ta + w2 * wc/ta + w3 * cash/curliab + w4 * cash/sales
+ w5 * ebit/sales + w6 * roa + w7 * e/ta + w8 * e/curliab + w9 * sales/ta
+ w10 * ebitda/curliab + w11 * e/curliab * sales/ta)
```

With the 11th coefficient the range of score values does not changes. The threshold still has the range [-8; 45].

The goal of GA is to find the values of coefficients w_i and threshold in order to maximize the profit. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The results of BehaviorSearch simulation are presented in Table 9.24

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀	w ₁₁
1.1	4.73	0.12	5.13	2.42	0.83	2.89	0.25	5.03	1.13	2.46	8.93

Table 9.24 The results of the experiment in BehaviorSearch

The results of a NetLogo simulation with the values of coefficients and threshold found by BehaviorSearch are presented in Table 9.25.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
1.1	216 504.9	97 864.7	439	13	464	96	2.88	52.87

Table 9.25 The results of the NetLogo simulation

(v) Synergy Working capital/Total assets x ROA

The second synergy score is:

$$Score = w_1 * i_1 + w_2 * i_2 + \dots + w_{10} * i_{10} + w_{11} * i_2 * i_6$$

The second score function is described in NetLogo in the following way:

```
set score (w1 * cash/ta + w2 * wc/ta + w3 * cash/curliab + w4 * cash/sales
+ w5 * ebit/sales + w6 * roa + w7 * e/ta + w8 * e/curliab + w9 * sales/ta
+ w10 * ebitda/curliab + w11 * wc/ta * roa)
```

With the 11th coefficient the range of score values does not change and the threshold still has the range [-8; 45].

The goal of GA has not change – BehaviorSearch has to find values of coefficients w_i and threshold in order to maximize the profit. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The results of BehaviorSearch simulation are presented in Table 9.26.

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀	w ₁₁
3.78	5.48	3.66	0.64	5.98	6.09	7.04	7.48	4.85	9.12	5.47	2.54

Table 9.26 The results of the experiment in BehaviorSearch

The results of a NetLogo simulation with the values of coefficients and threshold found by BehaviorSearch are presented in Table 9.27.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classificatio n accuracy (%)
3.78	219 700.1	85 656	531	22	372	87	3.98	61.07

Table 9.27 The results of the NetLogo simulation

(vi) *Comparison of the results achieved with synergies*

Table 9.28 provides the comparison of the results obtained in model №3.1 with normalized indicators, using the second fitness function, and of the results after two synergies.

Fitness function	Profit	Lost business	Type I error	Classification accuracy (%)
Model №3.1; normalized indicators; FF №2: Profit; No synergies	219 525.3	72 982.8	5.41	66.11
...+ Synergy <i>Equity/Current liabilities</i> · <i>Sales/Total assets</i>	216 504.9	97 864.7	2.88	52.87
...+ Synergy <i>Working capital/Total assets</i> · <i>ROA</i>	219 700.1	85 656	3.98	61.07

Table 9.28 Comparison of the of the outcomes of synergies

Unfortunately, still no unique best model is obtained. However, the use of synergies allowed to decrease the type I error (synergy *Equity/Current liabilities* · *Sales/Total assets*) or to increase profit (synergy *Working capital/Total assets* · *ROA*). The highest accuracy was achieved without synergies.

9.3 The results of credit scoring model №3.1

The comparison of profit, error rates and accuracy of the configurations of the credit scoring model №3.1 is presented in Table 9.29.

Model configuration	Profit	Type I error (%)	Type II error (%)	Classification accuracy (%)
Model №3.1; non-normalized indicators; FF №1: 0.05 * Red turtles + 0.95 * White turtles	151 158.4	10.63	60	89.13
Model №3.1; non-normalized indicators; FF №2: Profit	227 935.1	2.86	81.8	56.42
... + synergy <i>EBIT/Sales</i> · <i>Equity/Total assets</i>	239 452.1	3.89	81.35	59.98
... + synergy <i>Cash/Current liabilities</i> · <i>Sales/Total assets</i>	230 680.6	4.41	79.5	65.91
Model №3.1; non-normalized indicators; FF№3: Profit – lost business	223 509.2	6.35	79.62	70.55
Model №3.1; normalized indicators; FF №1: Red turtles	2 915.2	0	89.12	11.76
Model №3.1; normalized indicators; FF №2: Profit	219 525.3	5.41	80.47	66.11
... + synergy <i>Equity/Current liabilities</i> · <i>Sales/Total assets</i>	216 504.9	2.88	82.9	52.87
... + synergy <i>Working capital/Total assets</i> · <i>ROA</i>	219 700.1	3.98	81.05	61.07
Model №3.1; normalized indicators; FF№3: Profit – lost business	215 678.4	6.35	78.88	71.94

Table 9.29 Comparison of the results of all configurations of model №3.1

The average classification accuracy of the credit scoring model №3.1 is 60.57%. The highest accuracy was achieved in with normalized indicators and the first fitness function ($0.05 \cdot \text{Red turtles} + 0.95 \cdot \text{White turtles}$). A clear pattern emerges – a configuration with a high accuracy rate is the one with the highest type I error because loans were assigned to almost all clients. Thus, the experiment with the highest accuracy (89.13%) has poor discrimination abilities and cannot be considered desirable bank classification models.

It should be noticed that in both configurations (with normalized and non-normalized indicators) *profit-lost business* objective function had worse performance in terms of profit and type I error than *profit* fitness function.

The configuration with zero type I error performs very badly – it has the lowest profit and the lowest accuracy. Even though loans were assigned only to healthy firms that did not go bankrupt, these clients represent a small portion of applicants (less than 1%), which means that many potentially good firms were rejected. From a point of view of credit-granting organization, these configurations could not be used since most of the possibilities to gain are lost.

Another thing to notice is that both the highest profit and the lowest type I error are achieved in configurations with non-normalized indicators. Particularly, the lowest type I error (2.86 %) was obtained even without additional synergies. This configuration seems a good one since, besides the type I error, the amount of profit is relatively close to the highest one. The same is valid for a configuration with the highest profit (239 452.1) – its type I error is the third best among 10 experiments presented in the table (3.89%).

Thus, the final observations of the credit scoring model 3.1 are the following:

- 1) Average accuracy – 60.57%;
- 2) The configurations with the highest accuracy rate are those granting loan to almost every applicant;
- 3) The use of non-normalized financial indicators seems to give better results;
- 4) The use of synergies did not permit to find a perfect configuration, but helped to improve some parameters;
- 5) The results obtained with the *profit-lost business* fitness function are worse than the one achieved with *profit*.
- 6) No unique best model has been found so far.

The two best configurations of the credit scoring model №3.1 are the one with the lowest

values of type I error:

- Model №3.1; non-normalized indicators; FF №2;
- Model №3.1; normalized indicators; synergy $Equity/Current\ liabilities \cdot Sales/Total\ assets$.

9.4 Model №3.2 with non-normalized financial indicators

The ratios that will be used in model №3.2 are the same as in model №2.2 (see Table 8.9).

The new score function is determined in NetLogo in the following way:

```
set score (w1 * roa + w2 * roe + w3 * ebit/sales + w4 * sales/curass + w5 * va/ta
+ w6 * ta/curliab + w7 * wc/e + w8 * curass/curliab + w9 * e/curliab
+ w10 * ltfund/fixedass)
```

The range of values of score function depends on the value of coefficients w_i . The maximum range of score values is reached when all the coefficients are equal to 10. The histogram of the values is presented in Figure 9.7. Thus, the threshold range of values is set to [-70; 480].

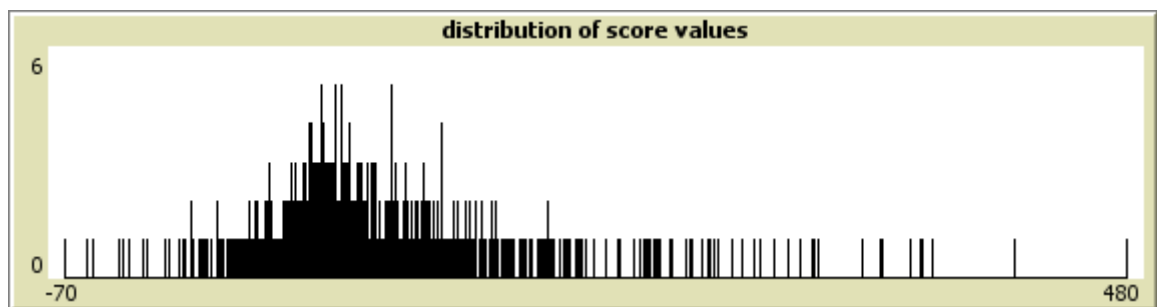


Figure 9.7 The distribution of the score values as $w_i=10$

(i) Fitness function №1

The first objective function is: $\alpha \cdot \text{count turtles with } [color = 15] + (1-\alpha) \cdot \text{count turtles with } [color = 9.9]$

The goal of GA is to find the values of coefficients w_i , of α and of the threshold in order to minimize the fitness function. The parameters specified in the BehaviorSearch are the following:

- *Measure*: fitness;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: minimize function at final step;

- *Number of searches: 2*

The results of the BehaviorSearch simulations are presented in Table 9.30. As in previous credit configurations, two simulations were performed. In the first search the range of values of alpha was [0; 1] with an increment 0.01, whereas in the second search it was [0.5; 1].

Thresh old	Fitne ss	α	$1-\alpha$	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
20.29	2.02	0.01	0.99	6.1	2.87	9.1	9.91	2.96	5.15	0.65	8.39	4.59	8.46
119.17	8.81	0.99	0.01	5.66	4.75	3.41	1.8	9.05	1.62	3.98	0.35	3.98	5.68

Table 9.30 The results of the experiments in BehaviorSearch – fitness function №1

The results of s NetLogo simulation with values of coefficients w_i , α and threshold found by BehaviorSearch are presented in Table 9.31.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
20.29	190 888.6	2.5	902	103	1	6	10.25	89.72
119.17	7 999.2	313 695.7	22	0	881	109	0	12.94

Table 9.31 The results of the NetLogo simulations – fitness function №1

The results of the first simulation with $\alpha=0.01$ are not satisfying – loans are assigned to more than 99% of firms. The second experiment with $\alpha=0.99$ performed differently – 22 companies were granted a loan and neither of them went bankrupt. The outcome is similar to the results of minimization of the first fitness function in model №3.1

(ii) Fitness function №2

The second objective function is: *profit = gains – losses.*

The goal of GA is to find values of coefficients w_i and threshold in order to maximize the profit. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The results of BehaviorSearch simulation are presented in Table 9.32.

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀
47.42	8.58	3.79	9.37	7.75	2.88	8.3	0.75	6.21	5.51	6.25

Table 9.32 The results of the experiment in BehaviorSearch – fitness function №2

The results of a NetLogo simulation with the values of coefficients and threshold found by BehaviorSearch are presented in Table 9.33.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classificatio n accuracy (%)
47.42	225 329.7	57 431.8	539	33	364	76	5.77	60.77

Table 9.33 The results of the NetLogo simulation – fitness function №2

(iii) Fitness function №3

The last objective function is: profit – lost business.

The goal of BehaviorSearch is to choose values of coefficients w_i and threshold in order to maximize the difference between profit and lost business. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit – lost_business;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The threshold and the values of coefficients w_i found with BehaviorSearch are presented in Table 9.34

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀
10.48	6.92	3.43	6.65	4.1	4.09	0.78	9.35	1.48	0.56	6.02

Table 9.34 The results of the experiment in BehaviorSearch – fitness function №3

The results of NetLogo simulation after applying these coefficients and the threshold are presented in Table 9.35.

Thre shold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classificatio n accuracy (%)
10.48	210 855.4	14 196.7	787	82	116	27	9.44	80.43

Table 9.35 The results of the NetLogo simulation – fitness function №3

The comparison of the three simulations with three different objective functions is presented in Table 9.36.

Fitness function	Threshold	Profit	Lost business	Type I error	Classification accuracy (%)
FF№1: 0.01* Red turtles + 0.99 * White turtles	20.29	190 888.6	2.5	10.25	89.72
FF№2: Profit	47.42	225 329.7	57 431.8	5.77	60.77
FF№3: Profit – lost business	10.48	210 855.4	14 196.7	9.44	80.43

Table 9.36 Comparison of the results of model №3.2 with non-normalized indicators

The best result in terms of profit and type I error belongs to the configuration with the second fitness function (225 329.7 and 5.77% accordingly). The highest accuracy is achieved while using the first fitness function and $\alpha=0.01$, though in this simulation almost all applicants were approved and this cannot be considered a valid result. The third fitness function did not produce better results.

Since the second fitness function seems to be the best for this configuration, two synergies will be applied to it. The indicators and weights resulted from the simulation in BehaviorSearch with fitness function №2 are listed in Table 9.37.

i_i	Financial indicator	w_i
1	ROA	8.58
2	ROE	3.79
3	EBIT / Sales	9.37
4	Sales / current assets	7.75
5	Value added / Total assets	2.88
6	Total assets / Current liabilities	8.3
7	Working Capital / Equity	0.75
8	Current assets/current liabilities	6.21
9	Equity / Current liabilities	5.51
10	LT funding / Fixed assets	6.25

Table 9.37 Coefficients w_i found by BehaviorSearch using fitness function №2 – model №3.2 with non-normalized coefficients

(iv) Synergy ROA · EBIT/Sales

The first synergy score function is:

$$Score = w_1 * i_1 + w_2 * i_2 + \dots + w_{10} * i_{10} + w_{11} * i_1 * i_3$$

The score function is described in NetLogo in the following way:

```
set score (w1 * roa + w2 * roe + w3 * ebit/sales + w4 * sales/curass
+ w5 * va/ta + w6 * ta/curliab + w7 * wc/e + w8 * curass/curliab
+ w9 * e/curliab + w10 * Itfund/fixedass + w11 * roa * ebit/sales)
```

With 11th coefficient the range of score values does not change. The threshold leverage and the score have the range [-70; 480].

The goal of GA has not change – BehaviorSearch has to find values of coefficients w_i and threshold in order to maximize the profit. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The results of BehaviorSearch simulation are presented in Table 9.38.

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀	w ₁₁
24.33	1.71	0.63	0.55	8.38	8.09	1.38	2.28	2.37	7.72	0.11	1.17

Table 9.38 The results of the experiment in BehaviorSearch

The results of a NetLogo simulation with the values of coefficients and threshold found by BehaviorSearch are presented in Table 9.39.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
24.33	244 689.4	51 904.5	577	28	326	81	4.63	65.02

Table 9.39 The results of the NetLogo simulation

(v) Synergy Sales/Current assets · Total assets/Current liabilities

The second synergy score function is:

$$Score = w_1 * i_1 + w_2 * i_2 + \dots + w_{10} * i_{10} + w_{11} * i_4 * i_6$$

The second score function is described in NetLogo in that way:

```
set score (w1 * roa + w2 * roe + w3 * ebit/sales + w4 * sales/curass
+ w5 * va/ta + w6 * ta/curliab + w7 * wc/e + w8 * curass/curliab
+ w9 * e/curliab + w10 * ltfund/fixedass + w11 * sales/curass * ta/curliab)
```

With 11th coefficient the range of score values does not change. The threshold and the score have the range [-70; 480].

The goal of GA has not change – BehaviorSearch has to find values of coefficients w_i and threshold in order to maximize the profit. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The results of BehaviorSearch simulation are presented in Table 9.40

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀	w ₁₁
47.15	6.67	3.33	3.03	7.65	1.83	0.83	4.41	9.28	4.7	3.91	4.2

Table 9.40 The results of the experiment in BehaviorSearch

The results of a NetLogo simulation with the values of coefficients and threshold found by BehaviorSearch are presented in Table 9.41.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
47.15	231 517.1	64 688.1	551	30	352	79	5.16	62.65

Table 9.41 The results of the NetLogo simulation

Table 9.42 provides the comparison of the results obtained in model №3.2 with non-normalized indicators, using the second fitness function, and of the results after two synergies.

Fitness function	Profit	Lost business	Type I error	Classification accuracy (%)
Model №3.2; non-normalized indicators; FF №2: Profit; No synergies	225 329.7	57 431.8	5.77	60.77
...+ Synergy $ROA \cdot EBIT/Sales$	244 689.4	51 904.5	4.63	65.02
...+ Synergy $Sales/Current\ assets \cdot Total\ assets/Current\ liabilities$	231 517.1	64 688.1	5.16	62.65

Table 9.42 Comparison of the outcomes of synergies

As it can be seen in the table, for the first time the best configuration in terms of all three parameters of interest (high profit, low type I error and high accuracy) is found. Moreover, this configuration was achieved with the use of a synergy $ROA \cdot EBIT/Sales$.

9.5 Model №3.2 with normalized financial indicators

In section 9.4 the financial ratios were not normalized. In Table 9.43 scaled financial ratios from section 9.4 are presented. After normalization the ranges are getting more similar to each other – all the values of 10 ratios are in the range [-0.7; 1.44].

i_i	Financial indicator	Range of values	Scale	Normalized range of values
1	ROA	[-1,4023625844; 0,596875]	1,9992375844	[-0,7014486899; 0,2985513101]
2	ROE	[-8,75; 34,5]	43,25	[-0,2023121387; 0,7976878613]
3	EBIT / Sales	[-1,1457100592; 0,7742162733]	1,9199263325	[-0,5967468854; 0,4032531146]
4	Sales / current assets	[0,1263557006; 8,5815789474]	8,7079346479	[0,0145104098; 0,9854895902]
5	Value added / Total assets	[-0,531816566; 1,7311111111]	1,1992945452	[-0,4434411614; 1,4434411614]
6	Total assets / Current liabilities	[0,64091471; 19,98785837]	20,6287730833	[0,0310689689; 0,9689310311]
7	Working Capital / Equity	[-56,1667; 35,7727]	91,9394	[-0,6109096902; 0,3890903098]
8	Current assets/current liabilities	[0,1709662921; 7,9603960396]	8,1313623317	[0,021025541; 0,9789744590]
9	Equity / Current liabilities	[-0,7292954265; 8,5940594059]	9,3233548324	[-0,0782224252; 0,9217775748]
10	LT funding /Fixed assets	[-7,0789473684; 67,5]	74,5789473684	[-0,0949188426; 0,9050811574]

Table 9.43 Normalized financial ratios – model №3.2

The indicators have been changed in NetLogo as follows:

```

set roa ((V06019 / V03023) / 1.9992375844)
set roe ((V06037 / V04015) / 43.25)
set ebit/sales ((V06019 / V06001) / 1.9199263325)
set sales/curass ((V06001 / V03022) / 8.7079346479)
set va/ta ((V06009 / V03023) / 1,1992945452)
set ta/curliab ((V03023 / V04035) / 20,6287730833)
set wc/e (((V03022 - V04035) / V04015) / 91.9394)
set curass/curliab ((V03022 / V04035) / 8.1313623317)
set e/curliab ((V04015 / V04035) / 9.3233548324)
set ltfund/fixedass ((V04025 / V03010) / 74.5789473684)

```

The range of values of score function depends on coefficients w_i . The maximum range of score values is reached when all coefficients are equal to 10. The histogram of the values is presented in Figure 9.8. The threshold range of values is set to $[-11; 35]$.

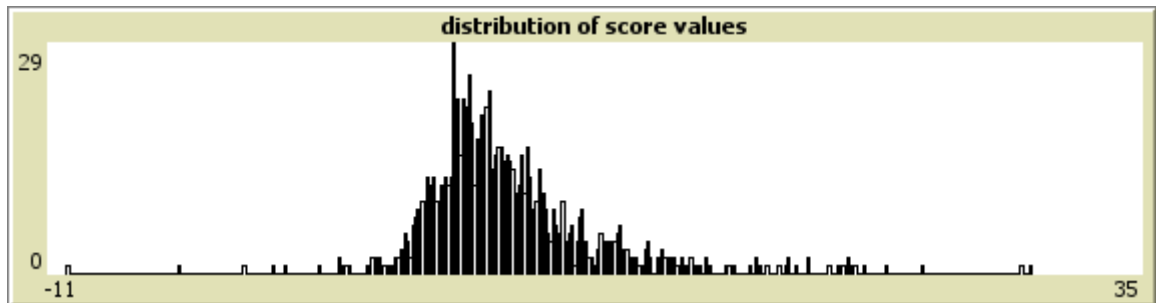


Figure 9.8 The distribution of the score values as $w_i=10$

(i) *Fitness function №1*

The first objective function is: $\alpha \cdot \text{count turtles with } [color = 15] + (1-\alpha) \cdot \text{count turtles with } [color = 9.9]$

The goal of GA is to find the values of α , of coefficients w_i and of the threshold in order to minimize the fitness function (see Figure). The parameters specified in the BehaviorSearch are the following:

- *Measure*: fitness;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: minimize function at final step;
- *Number of searches*: 2

The result of the experiment in BehaviorSearch is presented in Table 9.44. Only one

simulation was performed – the values of α could be in the range [0; 1].

Thres hold	Fitne ss	α	$1-\alpha$	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
-0.85	0	0.03	0.97	0.09	8.25	6.25	8.18	0.31	4.16	7.23	6.91	5.09	0.61

Table 9.44 The results of the experiment in BehaviorSearch – fitness function №1

The results of s NetLogo simulation with values of coefficients w_i , α and threshold found by BehaviorSearch are presented in Table 9.45.

Thre shold	Profit	Lost busin ess	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classificatio n accuracy (%)
-0.85	143 036.5	1.3	902	109	1	0	10.78	89.13

Table 9.45 The results of the NetLogo simulation – fitness function №1

The screenshot displays the BehaviorSearch configuration window, divided into several sections:

- Parameter Specification:** A text area containing a list of parameters and their ranges, such as ["threshold" [-11 0.01 35]], ["alpha" [0 0.01 1]], and various ["w" parameters [0 0.01 10]].
- Search Method Configuration:** A dropdown menu set to "StandardGA" and a table of parameters (mutation-rate, population-size, crossover-rate, population-model, tournament-size) with their respective values.
- Objective / Fitness Function:** A section for defining the goal (Minimize Fitness), collected measure (AT_FINAL_STEP), and other optimization settings like sampling and replicates.
- Setup and Step Configuration:** Fields for Setup (setup), Step (go), Measure (fitness), Measure If (true), and Stop If.
- Step Limit:** A field set to 4 model steps.
- Run BehaviorSearch:** A button to execute the search process.

Figure 9.9 The specifications of BehaviorSearch run – fitness function №1

As it can be seen, only one firm was rejected a loan. Moreover, the type I error is very high. Hence, the model does not provide good results.

(ii) Fitness function №2

The second objective function is: profit = gains – losses.

The goal of GA is to find values of coefficients w_i and threshold in order to maximize the profit. The parameter α is not used anymore. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The results of BehaviorSearch simulation are presented in Table 9.46.

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀
5.14	7.42	4.79	1.9	8.87	6.82	9.83	3.91	3.92	7.31	9.53

Table 9.46 The results of the experiment in BehaviorSearch – fitness function №2

The results of a NetLogo simulation with the values of coefficients and threshold found by BehaviorSearch are presented in Table 9.47.

Thre shold	Profit	Lost busines s	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classificatio n accuracy (%)
5.14	218 928.5	83 467	557	24	346	85	4.13	63.44

Table 9.47 The results of the NetLogo simulation – fitness function №2

(iii) *Fitness function №3*

The last objective function is: profit – lost business.

The goal of BehaviorSearch is to choose values of coefficients w_i and threshold in order to maximize the difference between profit and lost business. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit – lost_business;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;

- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The threshold and the values of coefficients w_i found with BehaviorSearch are presented in Table 9.48.

Threshold	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
2.39	1.78	7.01	5.43	9.39	0.69	9.98	9.84	4.17	3.99	5.39

Table 9.48 The results of the experiment in BehaviorSearch – fitness function №3

The results of NetLogo simulation after applying these coefficients and the threshold are presented in Table 9.49.

Threshold	Profit	Lost businesses	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
2.39	206 397.6	3 193.6	872	86	31	23	8.98	88.44

Table 9.49 The results of the NetLogo simulation – fitness function №3

The comparison of the three simulations with three different objective functions is presented in Table 9.50.

Fitness function	Threshold	Profit	Lost business	Type I error	Classification accuracy (%)
№1. $0.03 * \text{Red turtles} + 0.97 * \text{White turtles}$	-0.85	143 036.5	1.3	10.78	89.13
№2. Profit	5.14	218 928.5	83 467	4.13	63.44
№3. Profit – lost business	2.39	206 397.6	3 193.6	8.98	88.44

Table 9.50 Comparison of the results of model №3.2 with normalized indicators

As it can be seen, the model using fitness function №2 guarantees the highest profit and the lowest type I error. The third fitness function did not achieve any good result, but its outcome is very similar to the one of the second objective function. The highest accuracy is achieved with the first fitness function. Looking at the amount of approved firms in that experiment – more than 99% of all applicants – it can hardly be considered successful.

Since the second fitness function seems to be the best for this configuration, two synergies will be applied to it. The indicators and weights resulted from the simulation in BehaviorSearch with fitness function №2 are listed in Table 9.51.

i_i	Financial indicator	w_i
1	ROA	7.42
2	ROE	4.79
3	EBIT / Sales	1.9
4	Sales / current assets	8.87
5	Value added / Total assets	6.82
6	Total assets / Current liabilities	9.83
7	Working Capital / Equity	3.91
8	Current assets/current liabilities	3.92
9	Equity / Current liabilities	7.31
10	LT funding /Fixed assets	9.53

Table 9.51 Coefficients w_i found by BehaviorSearch using profit as an objective function in model №3.2 with normalized coefficients

(iv) Synergy *Total assets/Current liabilities* · *Long-term funding/Fixed assets*

The first synergy score is:

$$Score = w_1 * i_1 + w_2 * i_2 + \dots + w_{10} * i_{10} + w_{11} * i_8 * i_9$$

The first score function is described in NetLogo in that way:

```
set score (w1 * roa + w2 * roe + w3 * ebit/sales + w4 * sales/curass
+ w5 * va/ta + w6 * ta/curliab + w7 * wc/e + w8 * curass/curliab
+ w9 * e/curliab + w10 * ltfund/fixedass + w11 * ta/curliab * ltfund/fixedass)
```

With the 11th coefficient the range of score values does not changes. The threshold still has the range [-11; 35].

The goal of GA has not change – BehaviorSearch has to find values of coefficients w_i and threshold in order to maximize the profit. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The results of BehaviorSearch simulation are presented in Table 9.52.

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀	w ₁₁
2.31	3.85	4.91	8.31	4	0.29	1.66	7.87	5.81	9.94	2.3	1.69

Table 9.52 The results of the experiment in BehaviorSearch

The results of a NetLogo simulation with the values of coefficients and threshold found by BehaviorSearch are presented in Table 9.53.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classificatio n accuracy (%)
2.31	215 594.7	38 757.2	590	36	313	73	5.75	65.51

Table 9.53 The results of the NetLogo simulation

(iv) Synergy ROA · Sales/Current assets

The second synergy score is:

$$Score = w_1 * i_1 + w_2 * i_2 + \dots + w_{10} * i_{10} + w_{11} * i_1 * i_4$$

The second score function is described in NetLogo in that way:

```
set score (w1 * roa + w2 * roe + w3 * ebit/sales + w4 * sales/curass
+ w5 * va/ta + w6 * ta/curliab + w7 * wc/e + w8 * curass/curliab
+ w9 * e/curliab + w10 * ltfund/fixedass + w11 * roa * sales/curass)
```

The score and the threshold leverage have the same range as before: [-15; 35].

The goal of GA has not change – BehaviorSearch has to find values of coefficients w_i and threshold in order to maximize the profit. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The results of BehaviorSearch simulation are presented in Table 9.54.

Threshold	w ₁	w ₂	w ₃	w ₄	w ₅	w ₆	w ₇	w ₈	w ₉	w ₁₀	w ₁₁
3.6	8	1.67	8.02	9.37	1.14	2.08	9.8	1.84	9.31	2.45	7.85

Table 9.54 The results of the experiment in BehaviorSearch

The results of a NetLogo simulation with the values of coefficients and threshold found by BehaviorSearch are presented in Table 9.55.

Thres hold	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
3.6	220 239	87 701.5	394	16	509	93	3.9	48.12

Table 9.55 The results of the NetLogo simulation

It is necessary to compare the results obtained after two experiments with the best outcome of model №3.2 with normalized indicators. The results are presented in Table 9.56.

Fitness function	Profit	Lost business	Type I error	Classification accuracy (%)
Model №3.2; normalized indicators; FF №2: Profit; No synergies	218 928.5	83 467	4.13	63.44
...+ Synergy <i>Total assets/Current liabilities · Long-term funding/Fixed assets</i>	215 594.7	38 757.2	5.75	65.51
...+ Synergy <i>ROA · Sales/Current assets</i>	220 239	87 701.5	3.9	48.12

Table 9.56 Comparison of the of the outcomes of synergies

It turned out, that the synergy *ROA · Sales/Current assets* achieved a higher profit and a lower type I error than the original configuration. The synergy *Total assets/Current liabilities · Long-term funding/Fixed assets* obtained the highest accuracy. In that way, both synergies outperformed the original configuration.

9.6 The results of credit scoring model №3.2

The comparison of profit, error rates and accuracy of most of the configurations of the credit scoring model №3.2 is presented in Table 9.57.

Model configuration	Profit	Type I error (%)	Type II error (%)	Classification accuracy (%)
Model №3.2; non-normalized indicators; FF №1: 0.01 * Red turtles + 0.99 * White turtles	190 888.6	10.25	88.98	89.72
Model №3.2; non-normalized indicators; FF №2: Profit	225 329.7	5.77	82.7	60.77
... + synergy <i>ROA · EBIT/Sales</i>	244 689.4	4.63	80.1	65.02
... + synergy <i>Sales/Current assets · Total assets/Current liabilities</i>	231 517.1	5.16	81.67	62.65
Model №3.2; non-normalized indicators; FF №3: Profit – lost business	210 855.4	9.44	81.12	80.43
Model №3.2; normalized indicators; FF №1: 0.03 * Red turtles + 0.97 * White turtles	143 036.5	10.78	100	89.13
Model №3.2; normalized indicators; FF №2: Profit	218 928.5	4.13	80.3	63.44
...+ synergy <i>Total assets/Current liabilities · Long-term funding/Fixed assets</i>	215 594.7	5.75	81.1	65.51
...+ synergy <i>ROA · Sales/Current assets</i>	220 239	3.9	84.55	48.12
Model №3.2; normalized indicators; FF №3: Profit – lost business	206 397.6	8.98	57.4	88.44

Table 9.57 Comparison of the results of all configurations of model №3.2

The average classification accuracy of the credit scoring model 3.2 is 71.32 %. The highest accuracy was achieved in model №3.2 with non-normalized indicators and the first fitness function (0.01 · Red turtles + 0.99 · White turtles). The pattern explored in the model 3.1 is also consistent for model 3.2 – a configuration with a high accuracy rate is the one with the highest type I error because loans were assigned to almost all clients. Thus, the experiments with the two highest accuracies (89.72% and 89.13%) have poor discrimination abilities and cannot be considered desirable bank classification models

In comparison with the model №3.1, the highest profit and the lowest type I error are achieved in different configurations, but both with the use of synergies. The highest profit (244 689.4) was obtained when indicators were non-normalized and the synergy *ROA · EBIT/Sales* was applied. The lowest type I error (3.9 %), instead, was achieved with normalized financial ratios and the synergy *ROA · Sales/Current assets*. One can notice that ROA was applied in both synergies and for some reason it had improved the results.

As in the credit scoring model №3.1, the experiments that used the fitness function №3 and

did not perform better than those with the second objective function.

The final observations of the credit scoring model №3.2 are the following:

- 1) Average accuracy – 71.32%:
- 2) The configurations with the highest accuracy rate are those assigning loan to almost everyone;
- 3) The use of synergies of financial indicators seems to give better results;
- 4) Including ROA in the synergies allowed to improve the values of parameters of interest;
- 5) No unique best model has been found so far.

The two best configurations of the credit scoring model №3.2 are those with the lowest values of type I error:

- Model №3.2; normalized indicators; FF №2: Profit;
- Model №3.2; normalized indicators; synergy $ROA \cdot Sales/Current\ assets$.

9.7 Comparison of models №3.1 and №3.2

Four best configurations of credit scoring models №3.1 and №3.2 are presented in Table 9.58.

Model configuration	Profit	Type I error (%)	Type II error (%)	Classification accuracy (%)	Approved clients (%)
Model №3.1; non-normalized indicators; FF №2: Profit	227 935.1	2.86	81.8	56.42	48.42
Model №3.1; normalized indicators; synergy $Equity/Current\ liabilities \cdot Sales/Total\ assets$.	216 504.9	2.88	82.9	52.87	44.66
Model №3.2; normalized indicators; FF №2: Profit	218 928.5	4.13	80.3	63.44	57.41
Model №3.2; normalized indicators; synergy $ROA \cdot Sales/Current\ assets$	220 239	3.9	84.55	48.12	40.51

Table 9.58 Comparison of the models №3.1 and №3.2

First of all, the average classification accuracy of model №3.1 is lower than of the model №3.2 (60.57% in the model №3.1 and 71.32% in the model №3.2). The average classification accuracy of the best four configurations of model №3 is 55.21%.

The lowest type I error (2.86%) is achieved in model №3.1 with non-normalized financial

ratios. The model also accounts for the highest profit (227 935.1).

Two findings result from the table above:

- The models using the first set of coefficients seem to have better performances;
- The lowest type I error is achieved when non-scaled financial ratios were applied.

The next step is to test these 4 models using different sub-datasets.

10 TESTING THE MODELS

After the four best configurations of credit scoring models №3.1 and №3.2 are chosen, it is important to test them on different datasets. It will be done in three different ways:

- 1) The entire dataset will be divided into three subsets, each containing information about a particular dimension-type of the companies – *small*, *medium* and *large firms* datasets.
- 2) The entire dataset will be divided into four subsets, each containing information about firms from a particular Italian region. The regions that will be considered are the one with the highest amount of firms in dataset. They are *Lombardy*, *Emili-Romagna*, *Veneto* and *Piedmont*;
- 3) The entire dataset will be randomly divided into two subsets – *training set*, accounting for 80% of the companies in the dataset (809 firms) and *validation set* that contains remaining 20% of companies (204 firms).

So far, four configurations of credit scoring model №3 exhibited good results. Thus, four experiments will be executed. In each experiment one of the four configurations will be tested on seven sub datasets and on training and validation sets. After the tests, the results in terms of profit, classification accuracy and errors will be compared with the original 1013 firms dataset result.

10.1 Experiment №1

(i) Seven sub-datasets

The first configuration to be tested is the model №3.1, containing 10 non-normalized financial indicators from the first collection. The non-normalized financial indicators and coefficients w_i in front of them are presented in Table 10.1.

i_i	Financial indicator	w_i
1	Cash / Total Assets	2.34
2	Working capital / Total assets;	1.06
3	Cash / Current liabilities	7.95
4	Cash / Sales	4.91
5	EBIT / Sales	9.99
6	ROA	7.27
7	Equity / Total assets	8.84
8	Equity / Current Liabilities	10
9	Sales /Total assets.	8.22
10	EBITDA / Current liabilities	4.53

Table 10.1 Coefficients w_i of model 3.1 with non-normalized financial indicators

The parameters of the configuration are the following:

- Threshold – 17.14;
- The resulted type I error (%) – 2.86;
- The resulted type II error (%) – 81.8;
- Classification accuracy (%) – 56.42;
- Approved clients (%) – 48.42.

The results after testing the model on seven sub datasets are presented in Table 10.2.

Trial №	Dataset	Number of observations in dataset	Type I error (%)	Type II error (%)	Classification accuracy (%)	Approved clients (%)
1	Small size firms	687	4.03	84.02	50.73	43.44
2	Medium size firms	251	0.71	75.45	66.53	56.18
3	Large size firms	75	1.96	75	74.67	68
4	Firms from Lombardy	303	3.21	82.31	58.42	51.49
5	Firms from Veneto	256	2.14	81.9	61.72	54.69
6	Firms from Emilia-Romagna	215	3.23	82.64	51.87	43.46
7	Firms from Piedmont	147	3.03	81.48	53.74	44.9

Table 10.2 The results of the experiment №1 on 7 sub-datasets

One can notice that 3 of 7 sub datasets achieved a lower type I error with respect to the basic configuration – *medium-size firms*, *large-size firms* and *firms from Veneto*. In particular, the lowest type I error was obtained using the subset of *medium-size firms*.

The highest accuracy, the largest number of approved clients and the smallest type II error are achieved with *large-size firms* subset. It could be due to the dimension of the subset – it is the smallest one.

4 of 7 sub datasets exhibits a higher classification accuracy – *medium-size firms*, *large-size firms*, *firms from Lombardy* and *firms from Veneto*.

(ii) Training and validation sets

The model №3.1, containing 10 non-normalized financial indicators from the first collection was run again on training dataset. BehaviorSearch was used for searching for a values of coefficients w_i and of the threshold that allow to maximize the profit.

The values of coefficients and of the threshold were substituted into NetLogo, first for training set and then for validation set. The results of the simulations are presented in Table 10.3.

Dataset	Type I error (%)	Type II error (%)	Classification accuracy (%)	Approved clients (%)
Training set	5.92	75.3	79.83	79.46
Validation set	9.7	64.1	79.9	80.88

Table 10.3 The results of the experiment №1 on training and validation sets

With respect to the original configuration, the value of type I error, using training set, has almost doubled (from 2.86 to 5.92). The simulation performed on validation set has showed an even higher type I error (9.7). However, the test on control set has a higher accuracy and a higher amount of approved clients.

10.2 Experiment №2

(i) Seven sub-datasets

The second configuration to be tested is the model №3.1, containing 10 normalized financial indicators from the first collection and a synergy of two ratios (*Equity/Current liabilities* · *Sales/Total assets*). The normalized financial indicators and coefficients w_i in front of them are presented in Table 10.4.

i_i	Financial indicator	w_i
1	Cash / Total Assets	4.73
2	Working capital / Total assets;	0.12
3	Cash / Current liabilities	5.13
4	Cash / Sales	2.42
5	EBIT / Sales	0.83
6	ROA	2.89
7	Equity / Total assets	0.25
8	Equity / Current Liabilities	5.03
9	Sales /Total assets.	1.13
10	EBITDA / Current liabilities	2.46
11	Equity/Current liabilities · Sales/Total assets	8.93

Table 10.4 Coefficients w_i of model №3.1 with normalized financial indicators and a synergy *Equity/Current liabilities* · *Sales/Total assets*

The parameters of the configuration are the following:

- Threshold – 1.1;
- The resulted type I error (%) – 2.88;
- The resulted type II error (%) – 82.86;
- Classification accuracy (%) – 52.87;

- Approved clients (%) – 44.66.

The results after testing the model on seven sub datasets are presented in Table 10.5.

Trial №	Dataset	Number of observations in dataset	Type I error (%)	Type II error (%)	Classification accuracy (%)	Approved clients (%)
1	Small size firms	687	3.91	84.44	48.54	40.96
2	Medium size firms	251	1.54	78.51	61.35	51.79
3	Large size firms	75	0	79.41	64	54.67
4	Firms from Lombardy	303	2.67	82.35	57.1	49.5
5	Firms from Veneto	256	3.6	86.21	49.61	43.36
6	Firms from Emilia-Romagna	215	4.3	83.47	50.93	43.46
7	Firms from Piedmont	147	0	80	53.74	42.18

Table 10.5 The results of the experiment №2 on 7 sub-datasets

From the experiment it results that four sub-datasets achieved a lower type I error with respect to the basic configuration – *medium-size firms*, *large-size firms*, *firms from Lombardy* and *firms from Piedmont*. Importantly, *firms from Lombardy* and *firms from Piedmont* obtained zero type I error, meaning that no loans were granted to unhealthy clients. This is a good result, even though these two sub-datasets are relatively small and do not include many observations.

The highest classification accuracy and the highest percentage of approved loans are obtained while using *large-size firms* subset. The same result was achieved in the previous experiment.

(ii) Training and validation sets

The model №3.1, containing 10 normalized financial indicators from the first collection and synergy *Equity/Current liabilities* · *Sales/Total assets* was run again on training dataset. BehaviorSearch was used for searching for a values of coefficients w_i and of the threshold that allow to maximize the profit.

The values of coefficients and of the threshold were substituted into NetLogo, first for training set and then for validation set. The performance of the simulations is presented in Table 10.6.

Dataset	Type I error (%)	Type II error (%)	Classification accuracy (%)	Approved clients (%)
Training set	6.52	73.68	82.43	83.54
Validation set	10.24	65.79	79.41	81.37

Table 10.6 The results the experiment №2 on training and validation sets

With respect to the original configuration, the value of type I error, using training set, has sharply increased (from 2.88 to 6.52). The simulation performed on validation set has generated an even higher type I error=10.24.

10.3 Experiment №3

(i) Seven sub-datasets

The next configuration to be tested is the model №3.2, containing 10 normalized financial indicators from the second collection. The normalized financial indicators and coefficients w_i in front of them are presented in Table 10.7.

i_i	Financial indicator	w_i
1	ROA	7.42
2	ROE	4.79
3	EBIT / Sales	1.9
4	Sales / current assets	8.87
5	Value added / Total assets	6.82
6	Total assets / Current liabilities	9.83
7	Working Capital / Equity	3.91
8	Current assets/current liabilities	3.92
9	Equity / Current liabilities	7.31
10	LT funding /Fixed assets	9.53

Table 10.7 Coefficients w_i of model №3.2 with normalized financial indicators and the

The parameters of the configuration are the following:

- Threshold – 5.14;
- The resulted type I error (%) – 4.13;
- The resulted type II error (%) – 80.3;
- Classification accuracy (%) – 63.44;
- Approved clients (%) – 57.41.

The results after testing the model on seven sub datasets are presented in Table 10.8.

According to the test, three sub datasets achieved a lower type I error with respect to the basic configuration – *medium-size firms*, *firms from Veneto* and *firms from Piedmont*. As in the previous experiment, *firms from Piedmont* subset achieved zero type I error. Moreover, this sub-dataset obtained the lowest type I error and the highest classification accuracy

Trial №	Dataset	Number of observations in dataset	Type I error (%)	Type II error (%)	Classification accuracy (%)	Approved clients (%)
1	Small size firms	687	4.37	80.81	62.54	56.71
2	Medium size firms	251	2.78	77.57	65.34	57.37
3	Large size firms	75	6.25	85.19	65.33	64
4	Firms from Lombardy	303	4.71	82.71	61.06	56.11
5	Firms from Veneto	256	3.14	80.41	67.58	62.11
6	Firms from Emilia-Romagna	215	6.61	82.8	60.28	56.54
7	Firms from Piedmont	147	0	72.58	69.39	59.82

Table 10.8 The results of the experiment №3 on 7 sub-datasets

The highest percentage of approved loans are obtained with *large-size firms* subset. The same result was achieved in the previous experiment – the *large-size firms* dataset is the one with the lowest number of observations in it.

(ii) Training and validation sets

The model №3.2, containing 10 normalized financial indicators from the second collection was run again on training dataset. BehaviorSearch was used for searching for a values of coefficients w_i and of the threshold that allow to maximize the profit.

The values of coefficients and of the threshold were substituted into NetLogo, first for training set and then for validation set. The performance of the simulations is presented in Table 10.9.

Dataset	Type I error (%)	Type II error (%)	Classification accuracy (%)	Approved clients (%)
Training set	4.36	81.19	66.83	62.5
Validation set	5.88	67.65	73.53	66.67

Table 10.9 The results of the experiment №3 on training and validation sets

With respect to the original configuration, the value of type I error, using training set, has increased a little bit (from 4.13 to 4.36). The simulation performed on validation set has generated an higher type I error=5.88. The classification accuracy and the amount of approved clients are higher when the training set was applied.

10.4 Experiment №4

(i) Seven sub-datasets

The last configuration to be tested is the model №3.2, containing 10 normalized financial indicators from the second collection and a synergy $ROA \cdot Sales/Current\ assets$. The normalized

financial indicators and coefficients w_i in front of them are presented in Table 10.10.

i_i	Financial indicator	w_i
1	ROA	8
2	ROE	1.67
3	EBIT / Sales	8.02
4	Sales / current assets	9.37
5	Value added / Total assets	1.14
6	Total assets / Current liabilities	2.08
7	Working Capital / Equity	9.8
8	Current assets/current liabilities	1.84
9	Equity / Current liabilities	9.31
10	LT funding /Fixed assets	2.45
11	ROA · Sales/Current assets	7.85

Table 10.10 Coefficients w_i of model №3.2 with normalized financial indicators and the synergy
ROA · Sales/Current assets

The parameters of the configuration are the following:

- Threshold – 3.6
- The resulted type I error (%) – 3.9;
- The resulted type II error (%) – 84.55;
- Classification accuracy (%) – 48.12
- Approved clients (%) – 40.51

The results after testing the model on seven sub datasets are presented in Table 10.11.

Trial №	Dataset	Number of observations in dataset	Type I error (%)	Type II error (%)	Classification accuracy (%)	Approved clients (%)
1	Small size firms	687	4.76	85.71	44.02	36.73
2	Medium size firms	251	1.72	80.74	55.78	46.22
3	Large size firms	75	4.76	84.85	60	56
4	Firms from Lombardy	303	5.22	85.8	49.83	44.22
5	Firms from Veneto	256	0.89	84.03	52.34	43.75
6	Firms from Emilia-Romagna	215	6.49	86.13	42.52	35.98
7	Firms from Piedmont	147	1.82	82.61	47.62	37.41

Table 10.11 The results of the experiment №4 on 7 sub-datasets

The results suggest that three sub datasets achieved a lower type I error with respect to the basic configuration – *medium-size firms*, *firms from Veneto* and *firms from Piedmont*. The least

type I error and the highest accuracy and number of approved clients are obtained while using *firms from Veneto* subset.

The highest classification accuracy and the highest percentage of approved loans are obtained with *large-size firms* subset. The same result was achieved in the previous experiments – the *large-size firms* dataset is the one with the lowest number of observations in it.

(ii) Training and validation sets

The model №3.2, containing 10 normalized financial indicators from the second collection and synergy of two ratios ($ROA \cdot Sales/Current\ assets$) was run again on training dataset. BehaviorSearch was used for searching for a values of coefficients w_i and of the threshold that allow to maximize the profit.

The values of coefficients and of the threshold were substituted into NetLogo, first for training set and then for validation set. The results of the simulations are presented in Table 10.12.

Dataset	Type I error (%)	Type II error (%)	Classification accuracy (%)	Approved clients (%)
Training set	5.78	86.34	53.34	49.26
Validation set	7.77	78.22	57.35	50.49

Table 10.12 The results of the experiment №4 on training and validation sets

With respect to the original configuration, the value of type I error, using training set, has increased (from 3.9 to 5.78). The simulation performed on validation set has generated an even higher type I error=7.77. The classification accuracy and the amount of approved clients are higher when the training set was applied.

10.5 The results

The four best configurations of models №3.1 and №3.2 were estimated on 7 different partitions of the dataset. Restricting the dataset to *medium-sized firms* appears to reduce type I error, for all four estimated models. Even though the subset does not contain many observations (only 251), these results are very promising.

Also, restricting the dataset to *firms from Veneto* or to *firms from Piedmont* reduces type I error in three out of the four model configurations. For instance, estimating the 2nd and 3rd model configurations on the sub-dataset *firms from Piedmont* brings type I error all the way down to zero, meaning that no loans were assigned to unsound firms. This indicates a very powerful model.

The sub-datasets *firms from Lombardy* and *large-sized firms* achieved lower error rates in model configurations 2 and 1 times respectively.

Hence, it appears that the models which displayed the best performance on the entire dataset (see Section 9) do not work as well on the subsets *small-sized firms* and *firms from Emilia-Romagna*.

11 REPLICATION OF VARETTO'S MODEL (1998)

Since the massive spread of credit scoring models developed in 1960s, many studies and researches were performed. One of the most significant contributions to the bankruptcy prediction models in Italy was done by Franco Varetto in 1998. In particular, he applied GA to bankruptcy prediction and carried out the experiments using two different approaches: genetic linear score and genetic score by rule. Both of them were found to be powerful instruments for corporate distress diagnosis

In this section the genetic linear score experiment will be replicated.

The genetic linear score (GLS) function assumes the following form:

$$GLS = \alpha_0 + \alpha_1 R_{k1} + \alpha_2 R_{k2} + \dots + \alpha_i R_{ki}$$

Where:

α_0 – constant;

α_i – i th coefficient;

R_{ki} – k th financial ratio of the i th family of variables.

17 financial indicators, chosen in Section 6.3 (see Table 6.3), should be split into several families. Four financial ratio families emerge: liquidity, profitability, solvency in the long term, and efficiency. The families and indicators belonging to them are presented in Table 11.1.

It is important that all ratios in a family have the same sign. Indeed, all 17 financial variables have positive sign, which means that an increase in one of the indicators has a positive influence on the firm's economic health.

Nº of family	Family name	Variables in the family
1	Liquidity	1. Cash / Current liabilities; 2. Cash / Sales; 3. Cash / Total assets; 4. Current assets / Current liabilities; 5. EBITDA / Current liabilities; 6. Equity / Current liabilities; 7. Total assets / Current liabilities; 8. Working capital / Equity; 9. Working capital / Total assets.
2	Profitability	1. EBIT / Sales. 2. ROA; 3. ROE.
3	Solvency in the long term	1. Equity / Total assets; 2. LT funding / Fixed assets.
4	Efficiency	1. Sales / Current assets; 2. Sales / Total assets; 3. Value added / Total assets.

Table 11.1 Four families of financial ratios

The NetLogo model interface has change a little bit with respect to the models developed in the previous sections. The interface is depicted in Figure 11.1.

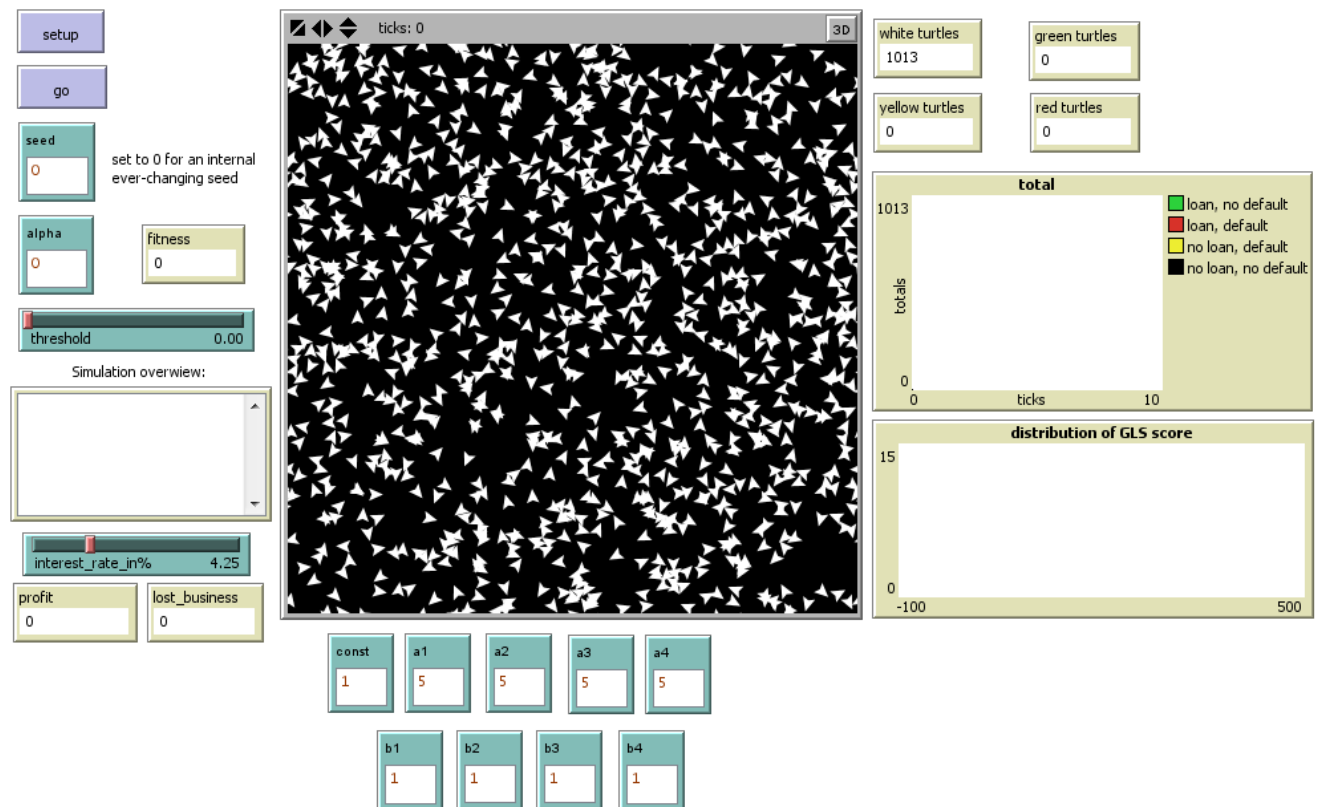


Figure 11.1 Setup of GLS model

There are 9 inputs: a *constant* with the range of values [-5; 5], 4 coefficients a_n in front of the financial indicators and 4 financial variables b_n that will be chosen from the families. All the coefficients a_n can take a value from the range [0; 5] with an increment of 0.01. The coefficients b_n have different ranges of values – it depends on the quantity of ratios in each family. For example b_1 can adopt a value from the range [1; 9] with an increment of 1, whereas b_4 is in the range [1; 3] with the same increment.

The value of GLS score for each firm depends on the coefficients a_n and a particular financial ratio b_n . Thus, the distribution of GLS values is unknown and in NetLogo it is set to the range [-100; 500]

The structure of “*to go*” procedure is similar to the models created before:

```
to go
  if ticks <= 4 [
    if ticks = 0 [ lending_decision ]
    if ticks = 1 [ default2012 ]
    if ticks = 2 [ default2013 ]
    if ticks = 3 [ default2014 ]
  ]
```

However, there is a new “*lending decision*” procedure at tick 0, in which 17 variables, each corresponding to a financial ratio, are defined. The loan_amount variable is similar to the one used in section 9:

```
to lending_decision
  ask turtles [
    if V04015 = 0 [die]
    set loan_amount min ( list (max list 10 (V04015 * 0.357)) 220216.9)
    set cash/curliab V03020 / V04035
    set cash/sales V03020 / V06001
    set cash/ta V03020 / V03023
    set curas/curliab V03022 / V04035
    set e/curliab V04015 / V04035
    set wc/e ((V03022 - V04035) / V04015)
    set wc/ta ((V03022 - V04035) / V03023)
    set ebitda/curliab V06044 / V04035
    set ta/curliab V03023 / V04035
    set roa V06019 / V03023
    set roe V06037 / V04015
    set ebit/sales V06019 / V06001
    set ltfund/fixedass V04025 / V03010
    set e/ta V04015 / V03023
    set sales/curas V06001 / V03022
    set va/ta V06009 / V03023
    set sales/ta V06001 / V03023
    calc_gls
```

The GLS function itself is defined by the procedure “*to calc_gls*”. Each local $pick_n$

variable takes the value b_n that corresponds to one of financial indicators from its family. Using these coefficients the global variable gls is calculated.

```

to calc_gls
;set pick1 to the chosen liquidity ratio
  let pick1 0
  if b1 = 1 [ set pick1 cash/curliab]
  if b1 = 2 [ set pick1 cash/sales]
  if b1 = 3 [ set pick1 cash/ta]
  if b1 = 4 [ set pick1 curas/curliab]
  if b1 = 5 [ set pick1 ebitda/curliab]
  if b1 = 6 [ set pick1 e/curliab]
  if b1 = 7 [ set pick1 ta/curliab]
  if b1 = 8 [ set pick1 wc/e]
  if b1 = 9 [ set pick1 wc/ta]
;set pick2 to the chosen profitability ratio
  let pick2 0
  if b2 = 1 [ set pick2 ebit/sales]
  if b2 = 2 [ set pick2 roa]
  if b2 = 3 [ set pick2 roe]
;set pick3 to the chosen solvency ratio
  let pick3 0
  if b3 = 1 [ set pick3 e/ta]
  if b3 = 2 [ set pick3 ltfund/fixedass]
;set pick4 to the chosen efficiency ratio
  let pick4 0
  if b4 = 1 [ set pick4 sales/curas]
  if b4 = 2 [ set pick4 sales/ta]
  if b4 = 3 [ set pick4 va/ta]
;calculate genetic linear score (GLS)
  set gls const + a1 * pick1 + a2 * pick2 + a3 * pick3 + a4 * pick4
end

```

According to Varetto's algorithm a firm is considered healthy if its value of GLS is positive, while if the value is negative, the firm is unsound:

```

if gls >= threshold [set color 65]

```

The GA has to choose the constant $const$, the coefficients a_1 , a_2 , a_3 , a_4 and one variable from each of the financial ratio family (b_1 , b_2 , b_3 , b_4).

Even though Varetto used only one fitness function, in the thesis 2 additional objective functions will be used: *profit* and *profit – lost_business*.

(i) *Fitness function №1*

The first objective function is: $count\ turtles\ with\ [color = 65] + \alpha \cdot count\ turtles\ with\ [color = 45]$

In words, it represents the number of correctly classified healthy and unhealthy firms. Varetto suggested $\alpha=1.2$, in that way, assigning more importance to the companies that were

correctly classified as unhealthy.

The goal of GA is to find the values of constant and of coefficients a_n and b_n in order to maximize the fitness function. The parameters specified in the BehaviorSearch are the following (see Figure 10.2):

- *Measure*: fitness
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The screenshot displays the BehaviorSearch configuration window, divided into several sections:

- Parameter Specification:** A text area containing a list of parameters and their ranges, such as `["threshold" 0]`, `["interest_rate_int" 4.25]`, `["const" [-5 0.01 5]]`, `["alpha" 1.2]`, and various `["a1" ...]`, `["b1" ...]` parameters. A button "Load param ranges from model interface" is located below.
- Setup:** A dropdown menu set to "setup".
- Step:** A dropdown menu set to "go".
- Measure:** A dropdown menu set to "fitness".
- Measure If:** A dropdown menu set to "true".
- Stop If:** An empty dropdown menu.
- Step Limit:** A numeric input field set to "4" with the unit "model steps".
- Search Method Configuration:** A dropdown menu set to "StandardGA". Below it is a table:

Parameter	Value
mutation-rate	0.03
population-size	50
crossover-rate	0.7
population-model	generational
tournament-size	3

 Below the table is a checkbox "Use fitness caching" which is checked.
- Search Encoding Representation:** A dropdown menu set to "StandardBinaryChromosome".
- Objective / Fitness Function:** A dropdown menu set to "Maximize Fitness". Below it is a dropdown menu set to "AT_FINAL_STEP".
 - Collected measure:** A dropdown menu set to "AT_FINAL_STEP".
 - Fixed Sampling:** A dropdown menu set to "10" with the unit "replicates".
 - Combine replicates:** A dropdown menu set to "MEAN".
 - Take derivative?** and **Use ABS value?** are unchecked checkboxes.
 - W.r.t.:** A dropdown menu set to "----" with a $\Delta = 0.0$ input field.
 - Evaluation limit:** A numeric input field set to "300" with the unit "model runs".
 - BestChecking replicates:** A numeric input field set to "0".
- A "Run BehaviorSearch" button is located at the bottom right.

Figure 11.2 The specifications of BehaviorSearch run – GLS model, fitness function №1

The model was run several times. In the first run $\alpha=1.2$, as Varetto suggested. Overall 7 different values of alpha were tried. In the 8th experiment GA had also to choose the value of coefficient alpha from the range [0; 15].

The results of different BehaviorSearch trials are presented in Table 11.2.

№ of trial	α	Fitness	const	a1	a2	a3	a4	b1	b2	b3	b4
1	1.2	906.6	-1.08	2.1	3.58	3.01	1.39	6	1	2	1
2	3	955	-2.63	2.25	3.4	0.84	1.97	3	3	2	1
3	5	1033	-4.66	2.87	2.63	4.55	3.1	9	3	1	1
4	7	1058	-2.01	2.29	5	0.41	1.56	9	1	2	2
5	9	1196	-1.91	4.62	4.84	0.96	1.68	9	2	1	2
6	11	1390	-0.58	3.06	0.71	0.03	0.51	2	3	1	2
7	13	1555	-3.57	1.49	3.52	0.8	2.35	6	1	2	3
8	14.79*	1745.7	-3.69	4.24	1.22	1.18	0.83	6	1	1	1

Table 11.2 The results of the experiments in BehaviorSearch – fitness function №1

The values of coefficients found by BehaviorSearch are substituted into the NetLogo model. The results are presented in Table 11.3.

№ of trial	α	Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
1	1.2	143 067.4	0.4	903	106	0	3	10.51	89.63
2	3	216 434.2	21 999	850	74	53	35	8.01	87.45
3	5	221 315.2	25 636	763	55	140	54	6.72	80.73
4	7	161 780.2	121 881.7	694	57	209	52	7.59	80.08
5	9	190 944.7	89 182.5	683	52	220	57	7.07	73.12
6	11	160 225	142 268.7	598	37	305	72	5.83	66.21
7	13	167 636.8	149 353.5	307	13	59	96	4.06	39.82
8	14.79*	228 665.2	55 609	385	17	518	92	4.23	47.13

Table 11.3 The results of the simulations in NetLogo – fitness function №1

In the trial №1, using $\alpha=1.2$ and coefficients generated by BehaviorSearch, more than 99% of firms received loan, among which more than 10% failed. Too much companies were approved and that is why the type I error is too high. The classification accuracy is the highest one among all trials – it is not because of the high predicting power of the configuration, but because the loans were assigned to almost all applicants.

In trial №2 and trial №3 the results were improving – type I error was decreasing, whereas the profit was rising. In the trials №4, №5 and №6 type I error was decreasing gradually, even though the experiments did not provide the best results.

The best result in terms of type I error was achieved in trial №7 (4.06). Additionally, in this configuration only 31% of applications were approved and because of that the classification accuracy rate is the lowest one (39.82%).

In the trial №8 GA found the value of alpha to be equal to 14.79. Surprisingly, the best result in terms of profit was achieved in this experiment, even though only 39% of firms were granted a loan. This configuration has also achieved one of the lowest type I errors (4.23).

Based on the results of eight experiments, a pattern emerges – as α rises, the number of approved companies decreases (green + red turtles) together with the number of failed among them (type I error).

Regardless the high value of the classification accuracy, the first configuration cannot be seen as a good one. Consequently, the best configuration is the one with the highest fitness (i.e, the highest α) – it is the trial №8.

(ii) *Fitness function №2*

The second objective function is: *profit = gains – losses*.

The goal of GA is to find the values of constant and of the coefficients a_n and b_n that maximize the profit. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The results of the BehaviorSearch simulation are presented in Table 11.4.

Constant	a1	a2	a3	a4	b1	b2	b3	b4
-2.8	1.52	4.59	3.68	1.32	3	2	1	1

Table 11.4 The results of the experiment in BehaviorSearch – fitness function №2

The results of a NetLogo simulation with the values of coefficients and threshold found by BehaviorSearch are presented in Table 11.5.

Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
240 698.3	44 855.8	546	23	357	86	4.04	62.45

Table 11.5 The results of the simulation in NetLogo – fitness function №2

The outcome seems good – 56% of applicants were granted a loan and the type I error is relatively low (4.04).

(iii) Fitness function №3

The last objective function is: profit – lost_business.

The goal of BehaviorSearch is to choose values of coefficients a_n and b_n that maximize the difference between profit and lost business. The parameters specified in BehaviorSearch are the following:

- *Measure*: profit – lost_business;
- *Step limit*: 4 model steps;
- *Search algorithm*: Standard GA;
- *Search encoding representation*: Standard Binary Chromosomes;
- *Goal*: maximize function at final step;
- *Number of searches*: 2.

The results of the BehaviorSearch simulation are presented in Table 11.6.

Constant	a1	a2	a3	a4	b1	b2	b3	b4
-1.19	0.41	1.58	0.73	0.3	7	2	1	1

Table 11.6 The results of the experiment in BehaviorSearch – fitness function №3

The results of a NetLogo simulation with the values of coefficients and threshold found by BehaviorSearch are presented in Table 11.7.

Profit	Lost business	Loan; No default	Loan; Default	No loan; No default	No loan; Default	Type I error	Classification accuracy (%)
195 271.7	17 182.7	674	54	229	55	7.42	72.04

Table 11.7 The results of the simulation in NetLogo – fitness function №3

The comparison of the three simulations with three different objective functions is presented in Table 11.8

The highest profit and the lowest type I error were obtained when the goal of GA was to maximize the profit. Moreover, this configuration has achieved the second best classification accuracy (62.45).

The configuration that was using the first fitness function obtained type I errors similar to

the lowest one, but the quantity of approved clients was relatively low there.

Fitness function	Profit	Lost business	Type I error (%)	Type II error (%)	Classification accuracy (%)	Approved clients (%)
FF№1: Green turtles + 14.79 · Yellow turtles	228 665.2	55 609	4.23	84.92	47.13	39.72
FF№2: Profit	240 698.3	44 855.8	4.04	80.59	62.45	56.23
FF№3: Profit – lost business	195 271.7	17 182.7	7.42	80.63	72.04	71.94

Table 11.8 Comparison of the results of GLS model

The highest accuracy is achieved with *profit – lost business* as the objective function. Besides, with this configuration more than 70% of companies were granted a loan, but the value of type I error is too high.

Hence, the best configuration of this model is the one with *profit* as fitness function.

12 FINAL RESULTS AND DISCUSSION

The main goal of this work was to create a credit scoring model for corporate distress analysis. The overall idea was to build an agent-based model using a sample of 1013 Italian firms and then to apply genetic algorithms to the search for the optimal parameters of the model.

In the thesis six different credit scoring models were developed. They were created following a “*learning by doing*” principle – each new version of the model was meant to improve on the previous one. For every model, genetic algorithms were used to find the values of the model parameters that would maximize/minimize one of three fitness functions (i.e. objective functions). These functions are the following:

- 1) $\alpha \cdot \text{Firms that received a loan and subsequently defaulted} + \beta \cdot \text{Firms that did not receive a loan and did not default} (\alpha + \beta = 1)$;
- 2) Profit;
- 3) Profit – lost business opportunities.

Hence, every model consists of at least three configurations – each configuration has parameter values found by maximizing/minimizing a different objective function. Once the results of all model configurations were computed, modelling outcomes were compared in terms of profit, error rates and accuracy rates.

Credit scoring model №1

This model is a preliminary experiment, which is meant to give the reader a simplistic overview of the main concepts and calculations used throughout this paper. In particular, the score function used for this initial version of the model was simply the *ratio of estimated financial debt to total assets*. It is clear, that an effective credit scoring model cannot be based just on one financial indicator, but this assumption was made for the sake of initial simplicity.

As no information was available in the dataset regarding the size of the loans granted to the companies, a proxy variable *loan_amount* was created, estimating the size of the loan requested by each company as $\frac{1}{4}$ of its total assets.

Applying GA to this model suggests that the best result in terms of profit was obtained using the second fitness function, whereas the lowest type I error was achieved when using the first objective function.

Credit scoring models №2.1 and №2.2

This models adopt a more sophisticated calculation of the credit score assigned to a firm:

$$Score = \frac{w_1}{w_1 + w_2 + \dots + w_n} i_1 + \frac{w_2}{w_1 + w_2 + \dots + w_n} i_2 + \dots + \frac{w_n}{w_1 + w_2 + \dots + w_n} i_n$$

The respective score functions of the two models are based on two different sets of 10 financial indicators, listed in Table 6.3 in Section 6.3. GA-based parameterization is undertaken three times for each sub-model (using three different fitness functions). In both versions the best results in terms of profit and type I error were obtained when the second objective function was maximized. However, the sub-model with the first collection of coefficients (model №2.1) displayed a better overall performance than the other one.

Credit scoring models №3.1 and №3.2

This models use a different, more intuitive, formula for calculating firms' credit scores, as the complexity of the formula used in models №2 did not seem to warrant superior performance of the model:

$$Score = w_1 i_1 + w_2 i_2 + \dots + w_n i_n$$

The calculation of *loan_amount* was also altered – for each firm, the size of the requested loan was estimated as the lowest between $A = \max [10; (V04015 * 0.357)]$ and $B = 220216.9$ ¹⁹. This formula was used to ensure that requested loans would have a cap and a floor, at 220 millions and 10 thousand euros respectively. Within this range, the size of the requested loan was assumed to be 35.7% of the firm's equity.

Each of the two models is based on a different set of 10 financial indicators. As the financial indicators varied within different ranges, it seemed appropriate to normalize them. Consequently, each sub-model is estimated twice – with non-scaled and scaled financial ratios.

Hence, the experiments in BehaviorSearch were performed for four configurations:

- Model №3.1 with non-normalized financial indicators;
- Model №3.1 with normalized financial indicators;
- Model №3.2 with non-normalized financial indicators;
- Model №3.2 with normalized financial indicators.

Another novelty of the third credit scoring model was the addition of synergies between financial indicators to the credit scoring function. For each of the four configurations above, the four indicators with the highest coefficients²⁰ were multiplied by each other to create two

¹⁹ All numbers are in thousands of euros

²⁰ Using the fitness function yielding the best performance

“synergy-variables” that entered the score function.

In model № 3.1 with non-normalized indicators, the configuration with the highest profit and lowest type I error was obtained using the second fitness function. Adding synergies to this configuration improves the results in terms of profit and in terms of type I error, but not in terms of both simultaneously. When the normalized indicators were used, the situation remained essentially the same – the best configuration was attained using the second fitness function and synergies improved profit or type I error, but only separately. Comparing the results obtained with scaled vs non-scaled financial ratios, it appeared that a better performance was achieved when non-normalized indicators were used.

The two best configurations of credit scoring model №3.1 are:

- Model №3.1; non-normalized indicators; FF №2;
- Model №3.1; normalized indicators; synergy $Equity/Current\ liabilities \cdot Sales/Total\ assets$.

The same procedure was applied to model № 3.2. Using non-normalized financial ratios, the best results in terms of profit and type I error were achieved with fitness function №2. The addition of the synergy $ROA \cdot EBIT/Sales$ has improved the results in terms of both profit and type I error. Using normalized financial indicators the best results were achieved, again, with fitness function №2 and the synergy $ROA \cdot Sales/Current\ assets$ has improved the results in terms of both performance measures.

In comparing the results with scaled and non-scaled financial ratios, it was observed that the configuration with the highest profit was the one with non-normalized indicators, whereas the configuration with the lowest type I error was obtained with normalized ratios. As with model №3.1, no unique best configuration was found.

The two best configurations of credit scoring model №3.2 are:

- Model №3.2; normalized indicators; FF №2;
- Model №3.2; normalized indicators; synergy $ROA \cdot Sales/Current\ assets$.

The scheme of the three credit scoring models is presented in Figure 12.1.

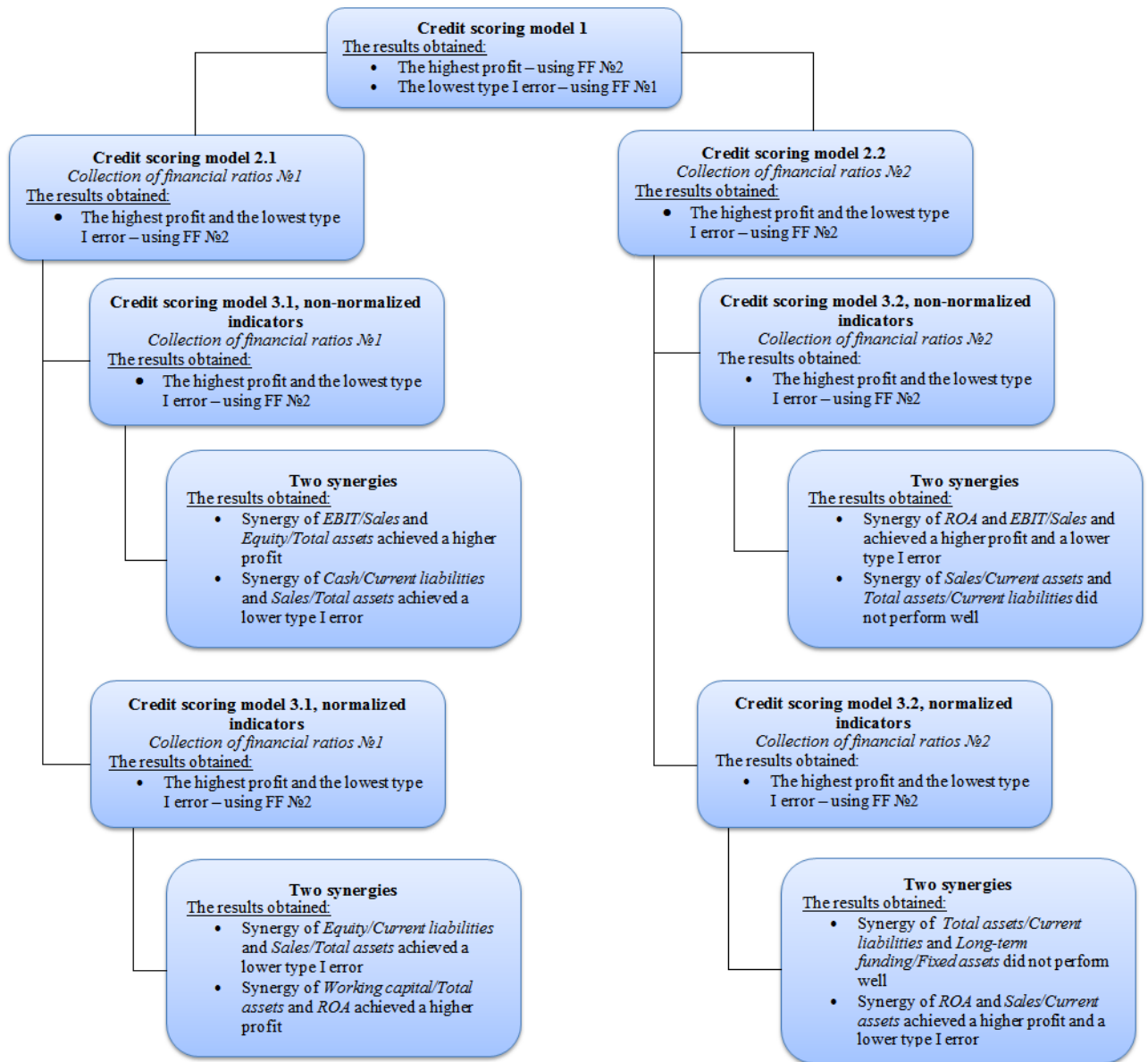


Figure 12.1 The scheme of three credit scoring models developed in the thesis

Genetic linear score model, Varetto (1998)

In addition to the three models developed above, the genetic linear score model of Varetto (1998) was replicated, due to the significance of this contribution in the empirical literature on Italian firms. The same 17 financial indicators of the previous models were used, but the score function was taken from Varetto's model. The genetic linear score (GLS) is:

$$GLS = \alpha_0 + \alpha_1 R_{a1} + \alpha_2 R_{b2} + \dots + \alpha_i R_{ki}$$

Where:

α_0 – constant;

α_i – i th coefficient;

R_{ki} – k th financial ratio of the i th family of variables.

The ratios were divided into four families of financial ratios: liquidity, profitability, long-term solvency and efficiency. Genetic algorithms not only chose the four coefficients in front of each ratio, but they also selected one indicator within each family. Three different objectives functions were used:

- 1) Firms that received a loan and did not default + α · Firms that did not receive a loan and defaulted subsequently;
- 2) Profit;
- 3) Profit – lost business.

The best configuration of this GLS model, in terms of profit and type I error, was obtained by maximizing the second fitness function.

Comparison of the results of the models

Overall, eight models were developed: model №1, model №2.1, model №2.2, model №3.1 with non-scaled ratios, model №3.1 with scaled ratios, model №3.2 with non-scaled ratios, model №3.2 with scaled ratios and the GLS model. The best performing configuration of each model²¹ was identified and assessed against the other models on the basis of their type I and type II errors, classification accuracy, the percentage of approved clients and the profit earned.

Type I error refers to situations whereby a bad costumer is classified as a good one, resulting in losses to the banking institute. *Type II error*, instead, refers to situations whereby a good costumer is classified as a bad one and its loan application is rejected. Type I error is a more serious mistake as it brings actual losses to a bank. However, Type II error is also important, as reducing it will increase the revenues of a credit institution.

Total error is the proportion of all applicants that were incorrectly classified as either sound or unsound (firms that did not receive a loan and that did not default and firms that received a loan and defaulted subsequently). *Classification accuracy* is the proportion of clients that were correctly classified as either sound or unsound (firms that received a loan and did not default and firms that did not receive a loan and defaulted subsequently). *The percentage of approved clients* is the ratio of clients that received a loan to the total number of applicants.

In addition to comparing the error and accuracy rates, the *profit* earned with each model specification was measured and compared across models. As two different formulas were used

²¹ For each model, the best configuration was assumed to be the one with the lowest type I error.

across the thesis to estimate the size of the requested loans, comparisons in terms of profit can only be performed among models №1, №2.1 and №2.2, with a separate comparison required among models №3.1, №3.2 and the GLS model.

As no model outperformed all other models in terms of all 6 performance measures, the main parameter used for the comparison was type I error.

The descriptive results associated to eight best configurations are presented in Table 12.1.

	Type I error (%)	Type II error (%)	Classification accuracy (%)	% of clients approved	Profit
Model №1; FF№2	2.18	79.05	62.69	54.29	366 142.5
Model №2.1; FF№2	4.56	80.52	63.87	58.4	381 818.1
Model №2.2; FF№2	6.38	82.9	62.15	58.9	308 380.5
Model №3.1 with non-scaled ratios; FF№2	2.86	81.8	56.42	48.4	227 935.1
Model №3.1 with scaled ratios + synergy <i>Equity/Current liabilities · Sales/Total assets</i>	2.88	82.9	52.87	44.66	216 504.9
Model №3.2 with non-scaled ratios; FF№2	4.13	80.3	63.44	57.41	218 928.5
Model №3.2 with scaled ratios + synergy <i>ROA · Sales/Current assets</i>	3.9	84.5	48.12	40.51	220 239
GLS model; FF №2	4.04	80.59	62.45	56.23	240 698.3

Table 12.1 Comparison of the models' performances

The lowest type I and type II errors were achieved with model №1 (2.18 and 79.05). Overall the first model had a very good performance, despite the simplified assumptions introduced for building this model and the use of a single financial indicator in the score function. This setup resulted in a limited parameter space being searched by BehaviorSearch.

Model №2.1 achieved the highest accuracy rate among all other models (with values of 63.87). This model also obtained the highest profit among the models that used the simplified estimate of the loan size – models №1, №2.1 and №2.2.

Model №2.2, displayed a worse performance than model №2.1 in terms of all measures. In fact, the highest type I error was obtained with model 2.2, indicating that many bad clients were included in the sound group (i.e. weak classification performance of the model). Hence, the second set of coefficients did not seem to provide satisfactory results.

Model №3.1, in its versions with non-normalized and normalized indicators, exhibited the 2nd- and the 3rd- lowest values of type I error. In particular, using normalized indicators and adding the synergy *Equity/Current liabilities · Sales/Total assets* achieved a very low type I error (2.88), signaling a reasonably good performance of this model. Whilst the addition of an 11th coefficient to the score function may sometimes worsen the results very much, this did not seem to be the case here. Model №3.1 appeared to be a valid model, even though its average accuracy was not the highest. Regarding the use of non-scaled vs scaled indicators, the first choice seemed to lead to better performance.

Model №3.2, in its versions with non-normalized and normalized indicators, displayed fairly high values of type I error. However, the scaled version of the model included a synergy of financial indicators and appeared to improve on the general model. Model №3.2 with scaled ratios and the synergy *ROA · Sales/Current assets* displayed the lowest percentage of clients approved among all models, even though the achieved profit is not the smallest one. Overall, it appeared that – as with models №2.1 and №2.2 – the second set of financial indicators performed worse than the first one. Unlike model №3.1, the use of scaled ratios resulted in a lower value of type I error.

The GLS model did not seem to perform well among the models estimated. Its type I error is higher than that of models №3.1 and model №1, but lower than that of models №2.1, №2.2 and №3.2. This poor performance may due to the fact that the financial indicators included in the GLS model could be different from the ones used in Varetto's original paper (which does not list them explicitly). However, among all configurations of model №3 and GLS model, the latter achieved the highest profit (240 698.3). The GLS model developed here included the same ratios as models №2 and №3, although the score function and the algorithm differed.

In light of the results above, it is possible to draw the following conclusions on these credit scoring models:

- 1) The models displayed a reasonably good performance, reaching accuracy rates well above 50%;
- 2) The use of the first set of indicators seemed to give better results in models №2 and №3;
- 3) The normalization of financial indicators did not seem to alter the performance of model №3 significantly;
- 4) Model №3.1 outperforms all other models except for model №1;
- 5) The inclusion of synergies among financial indicators arguably improved the model performance;

- 6) There is no “unique best model” that outperformed all other on all five performance measures.

12.1 Further research

The results of this thesis provide interesting insights into the application of ABMs and GAs to credit scoring models.

Importantly, throughout this thesis it appears that the predictive power of the models is crucially linked to the specific variables included in the scoring function. That is why two sets of indicators were compared in this work. Even though one of them yields undeniably better results than the other, a different set of indicators yielding even better performances could be identified.

In general, despite the good performance of the models developed in this thesis, further extensions and improvements are possible.

First of all, the availability of information on real loan sizes – as opposed to estimated sizes – may increase the accuracy and the performance of the models. Moreover, the interest rate applied for calculating potential gains from loans was assumed to be 4.25%. Differentiating the interest rates by applicant would surely increase the predictive power of the scoring model. Indeed, in reality, interest rates charged on corporate loans are not the same and vary according to the size and financial situation of a company.

In addition, different credit scoring models can be developed for different types of companies. In Section 10, the best four versions of models №3.1 and №3.2 were tested on different portions of the datasets, highlighting the differential appropriateness of these scoring models to different types of firms. Hence, developing different scoring models for SMEs vs large companies could be an interesting research attempt.

Another feature that could enhance the model performance would be the modeling of networks among firms. With a sufficiently large dataset, it is very likely that the companies in it are linked to each other in one way or the other – they could be business partners or contractors. Thus, the financial problems of one player can lead to difficulties for another player. Such network relations could, in fact, greatly benefit the accuracy of credit scoring models.

13 CONCLUSIONS

The main topic of this thesis – credit scoring – is of high importance to financial institutions. Accurately predicting whether a loan applicant will meet its financial obligations or not is the essence of the banking business and is central to a bank's competitive advantage.

The goal of the thesis was to construct an agent-based credit scoring model using a dataset of 1013 Italian firms provided by Cerved S.p.A. Genetic algorithms were used to develop and estimate a credit scoring model on the dataset.

The theory underpinning the main credit scoring techniques was outlined in Section 2, together with the strengths and weaknesses of such techniques. The main focus of the section was on those methods that are most widely used in practice – parametric techniques (e.g. discriminant and regression analysis) and non-parametric techniques (e.g. neural networks, classification trees, trait recognition, k-nearest neighbor method). Also, the internal ratings-based approach of Basel II was discussed in the Section.

Section 3 provided the main notions of another non-parametric credit scoring technique – genetic algorithms. Genetic algorithms (GAs) identify the optimum coefficients of a model using the Darwin's principles of natural selection. GAs belong to machine learning techniques that are superior to the conventional credit scoring approaches and seem to be very popular among practitioners and scholars alike. The main characteristics and applications of GAs are discussed in the section.

Section 4 presented the main characteristics of agent-based modelling (ABM), a technique used in the thesis to simulate the functioning of the credit scoring models developed in the thesis. The main applications, advantages and shortcomings of ABM are discussed in the section.

The main features of the two software applications used in this thesis – NetLogo and BehaviorSearch – were then reviewed in Section 5. NetLogo is an agent-based programming language used for modelling complex systems. With this software package, the loan application and scoring process were simulated for the 1013 Italian firms in the sample. BehaviorSearch is a tool that uses genetic algorithms to search the parameter space for values that can maximize some objective function.

The modeling part of the thesis starts with Section 6, where the dataset is described. The dataset contains 1013 industrial Italian firms, among which 67.8% are small companies and 10.76% are firms that are known to have defaulted. Within this dataset, 17 financial ratios are selected for inclusion in the scoring formula.

5 versions of the credit scoring agent-based model are developed in Sections 7, 8 and 9.

The first model, described in Section 7, is an elementary version of a credit scoring model that uses simplistic assumptions for the estimated size of requested loans, for the fitness function and for the scoring formula. Surprisingly, this apparently naïve model achieves a low error rate.

Instead, the two more advanced models, covered in Section 8, use the 17 indicators chosen in Section 6. The difference between the models is the use of two different sets of 10 indicators. The first collection of indicators appears to yield better performances compared to the second one.

Section 9 presents the other models, which use, again, the same 17 indicators, in their non-scaled and scaled versions. Thus, four models emerge: i) Model №3.1 with non-normalized financial indicators; ii) Model №3.1 with normalized financial indicators; iii) Model №3.2 with non-normalized financial indicators; iv) Model №3.2 with normalized financial indicators. One important novelty of this section was the introduction of synergies between financial indicators that allow to get rid of the linearity assumption in the credit scoring models. Model №3.1 displays better performance than model №3.2.

The four versions of model 3 yielding the best performances are identified in Section 10 and re-estimated over partitions of the original dataset (e.g. firms of different dimensions, firms from different regions and training and validation sets). According to these experiments, the models have the highest explanatory power for medium firms, firms from Veneto and firms from Piedmont. Also, the experiments on training and validation revealed poor out-of-sample performance of GA-based credit scoring models.

In addition to the previous models, a genetic linear score (GLS) model – first developed by Varetto in 1998 – was replicated in Section 11. The methodology behind this model differed greatly from that of the other models, although the same 17 indicators were included in the model (after being divided into four families). Whilst the specific financial ratios used in the original model of Varetto were not known, the results of the replicated model seemed very promising,

Finally, the results of the four models were compared, evaluated and discussed in Section 12. The main observation that emerged from this research is that very low error rates (approximately 3%) can be achieved by some of the best-performing credit scoring models. Also, the first set of indicators seemed to be more relevant than the second set at predicting corporate distress. Even though model №1 had the lowest error rate, model №3.1 also provided satisfactory results and outperformed all other versions of model 3. It is important to note that,

overall, no model can be regarded as superior to all other based on an individual measure. For instance, low error rates may be accompanied by low profits or few accepted loan applications. The section finishes off with several suggestions for further research.

Section 13 concludes the thesis with some final remarks.

14 REFERENCES

- Altman, E. (1968). *Financial ratios, discriminant analysis and the prediction of corporate bankruptcy*. The Journal of Finance, September 1968, pp.589-609.
- Altman, E. (2002). *Revisiting credit scoring models in a Basel 2 environment*.
- Altman, E., Marco, G., Varetto, F. (1994). *Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the Italian experience)*. Journal of Banking and Finance 18, pp. 505–529.
- Anderson, R. (2007). *The Credit Scoring Toolkit: theory and practice for retail credit risk management and decision automation*. New York: Oxford University Press. Chapters 1, 2, 7, 36.
- Angelini, E., Tollo, G., and Roli, A. (2007). *A Neural Network Approach for Credit Risk Evaluation*. The Quarterly Re-view of Economics and Finance, Vol. 48, No. 4, 2008, pp. 733-755.
- Back, B., Laitinen, T., and Sere, K. (1996). *Neural networks and genetic algorithms for bankruptcy predictions*. Expert System with Applications, 11(4), 407–413.
- Basel Committee on Banking Supervision. (2003). Overview of the New Basel Capital Accord. BIS consultative document.
- Beaver, W. (1966). *Financial ratios as predictors of failure*. *Empirical Research in Accounting: Selected Studies*, 1966, supplement to vol.5, Journal of Accounting Research, pp. 71-111.
- Boero, R., Morini, M., Sonnessa, M., Terna, P. (2015). *Agent-based models of the economy: from theories to applications*. Palgrave Macmillan.
- Cannata, F. (2001). *Rating esterni e dati di bilancio: un'analisi statistica*. Studi e note di economia n. 3.
- Davalos, S., Gritta, R., Goodfriend, J., Adrangi, B. (2005). *The Use of a Genetic Algorithm in Forecasting Air Carrier Financial Stress and Insolvency*. Credit and Financial Review, 11(2), June 2005.
- Davalos, S., Gritta, R., & Adrangi, B. (2010). *Deriving rules for forecasting air carrier financial stress and insolvency: A genetic algorithm approach*. Journal of the Transportation Research Forum, 46(2).

- du Jardin, P. (2010). *Predicting bankruptcy using neural networks and other classification methods: The influence of variable selection techniques on model accuracy*. Neurocomputing, 73(10-12), 2047- 2060.
- D'Annunzio, N., Falavigna, G. (2004). *Modelli di analisi e previsione del rischio di insolvenza. Una prospettiva delle metodologie applicate*.
- Etemadi, H., Rostamy, A., Dehkordi, H. (2009). *A genetic programming model for bankruptcy prediction: Empirical evidence from Iran*. Expert Systems with Applications 36 (2/2): 3199-3207.
- Finlay, S.M. (2006). *Using genetic algorithms to develop scoring models for alternative measures of performance*. Technical Report 2006/022, Department of Management Science, Lancaster University, UK.
- Fletcher, D., Goss, E. (1993). *Forecasting with neural networks: An application using bankruptcy data*. Information and Management 24 (3): 159–167.
- Galvao, H.K.H., V.M. Becerra, and M. Abou-Seada. (2002). *Variable Selection For Financial Distress Classification Using a Genetic Algorithm*. Congress on Evolutionary Computation, May 12-17, Honolulu, Hawaii, (2002).
- Gilbert, N. (2008). Agent-based models. Sage Publications.
- Gordini N. (2014) *A genetic algorithm approach for SMEs bankruptcy prediction: Empirical evidence from Italy*. Expert Systems with Applications, Volume 41, Issue 14, 15 October 2014, pp. 6433-6445.
- Gurný, P., Gurný, M. (2013). *Comparison Of Credit Scoring Models On Probability Of Default Estimation For US Banks*. Prague Economic Papers, 2, pp. 163-181.
- Jakubik, P., Teply, P. (2011). *The JT index as an indicator of financial stability of corporate sector*. Prague Economic Papers, vol. 2, June 2011, pp. 155-175.
- Henley, W. E. (1995). *Statistical aspects of credit scoring*. Ph.D. Thesis, The Open University, Milton Keynes.
- Lazzari, V., Laruccia, E. (2000). *La mizurazione del rischio di credito per un portafoglio di finanziamenti bancari*.
- Lee, K., Booth, D., Alam, P. (2005) *A comparison of supervised and unsupervised neural networks in predicting bankruptcy of Korean firms*. Expert Systems with Applications 29, (1), pp. 1–16.

- Kolari J., Caputo M., Wagner D. (1996) *Trait recognition: an alternative approach to early warning systems in commercial banking*. Journal of Business Finance and Accounting, 23(9/10).
- Kolari J., Glennon D., Shin H., Caputo M., (2002), *Predicting large US commercial bank failures*, Journal of Economics and Business, 54, pp. 361-387.
- Macal, C.M., and North, M.J. (2009). *Agent-based modeling and simulation*. Proceedings of the 2009 Winter Simulation Conference (WSC), 86–98.
- Pierri, F., Burchi, A., Stanghellini, A. (2011) *La Capacita' predittiva degli indicatori di bilancio: Un metodo per le PMI*.
- Resti, A., Sironi A. (2007). *Risk management and shareholders' value in banking. From Risk Measurement Models to Capital Allocation Policies*. Wiley Finance, Chichester. Chapters 10, 20.
- Sexton, R.S., Sriram, R.S., Etheridge, H. (2003). *Improving Decision Effectiveness of Artificial Neural Networks: A Modified Genetic Algorithm Approach*. Decision Sciences 34 (3), (2003): 421-442.
- Shin, K. S., Lee, Y. J. (2002). *Genetic algorithm application in bankruptcy prediction modeling. Expert System with Applications*, 23, pp. 321–328.
- Stanghellini, E. (2009), *Introduzione ai metodi statistici per il credit scoring*, Springer, Milano, Italia.
- Thomas, L. C., Edelman, D. B., Crook, J. N. (2002). *Credit scoring and its applications*. Philadelphia: Society for Industrial and Applied Mathematics. Philadelphia, USA.
- Varetto, F. (1998). *Genetic algorithms applications in the analysis of insolvency risk*. Journal of Banking and Finance, 22, pp. 1421–1439.
- West, D. (2000). *Neural network credit scoring model*. Computers and Operations Research, 27 (11/12), pp.1131-1152.

A APPENDIX

A1 The Python code

```
#check current directory
import os
print "Old directory: " + os.getcwd()
#change directory to where the code and the datasets are saved on your PC
os.chdir('C:/Users/Yuliya/Desktop/UnItO/Tesi/code/')
print "New directory: " + os.getcwd()
#open the dataset of interest
f=open("Tolkacheva_20150804_campione_imprese.csv","r")
#create new file (or rewrite over pre-existing file)
#This second file will be imported into netlogo
g=open("datasetfornetlogo.txt","w")

i = -1
for line in f:
    if i != -1: #skip headers
        #find position of strings which need to be enclosed in quotes
        #(only three columns contain strings in the dataset at hand)
        one = line.find(";",line.find(";",line.find(";",0)+1)+1)
        two = line.find(";",line.find(";",line.find(";",line.find(";",0)+1)+1)+1)
        three = line.find(";",line.find(";",line.find(";",line.find(";",line.find(";",0)+1)+1)+1)+1)
        four =
line.find(";",line.find(";",line.find(";",line.find(";",line.find(";",line.find(";",0)+1)+1)+1)+1)+1)

        #edit lines by :
        #(1) adding quotes;
        line=line[0:one + 1] + "\"" + line[one + 1:two] + "\";\\"" + line[two + 1:three]
        + "\";\\"" + line[three + 1:four] + "\";\" + line[four + 1:-1]
        #(2) substituting semicolons with spaces;
        line=line.replace(";", " ")
        #(3) adding an index at the start of each line (and writing on the new file)
        g.write(str(i) + " " + line + "\n")

    i = i + 1

#print i

f.close()
g.close()
```

A2 The financial ratios used in the 14 credit scoring studies analyzed

Study cited	Suggested financial variables
Altman (1968)	<u>5 variables</u> : Working capital / Total assets; Retained earnings / Total assets; EBIT / Total assets; Market value of Equity / Book value of total liabilities; Sales / Total assets.
Fletcher et al. (1993)	<u>3 variables</u> : Current assets / Current liabilities; Quick ratio [(Cash + other near cash assets) / Current liabilities]; Net income / Working capital.
Back et al. (1996)	<u>31 variables</u> : Cash / Current liabilities; Cash flow / Current liabilities; Cash flow / Total assets; Cash flow / Total debt; Cash / Net sales; Cash / Total assets;

	Current assets / Current liabilities; Current assets / Net sales; Current assets / Total assets; Current liabilities / Equity; Equity / Fixed assets; Equity / Net sales; Inventory / Net sales; Long term debt / Equity; MV of equity / Book value of debt; Total debt / Equity; Net income / Total assets; Net quick assets / Inventory; Net sales / Total assets; Operating income / Total assets; EBIT / Total interest payments; Quick assets / Current liabilities; Quick assets / Net sales; Quick assets / Total assets; Rate of return to common stock; Retained earnings / Total assets; Return on stock; Total debt / Total assets; Working capital / Net sales; Working capital / Equity; Working capital / Total assets.
Lazzari et al. (2000)	<u>7 variables:</u> Capitale netto / (Capitale netto + debiti finanziari); Autofinanziamento / Debiti finanziari; Immobilizzi / Capitali permanenti; Debiti a breve termine / Liquidità; Oneri finanziari / Fatturato; ROE; (Attività correnti – magazzino) / Passività correnti.
Cannata (2001)	<u>15 variables:</u> Totale indebitamento / (Capitale + riserve – att. immateriali); EBIT / Interessi passive; Interessi Passivi / (Reddito operativo + non operativo); Attivo corrente / Passivo corrente; (Attivo corrente – magazzino) / Passivo corrente; EBITDA / Passivo corrente; Cash flow / Passività totali; Margine operative; Margine operativo ante imposte = Utile ante imposte / Fatturato; EBIT / Fatturato; EBITDA / Fatturato; ROA; ROE; ROE2 =Utile Netto / (capitale + riserve); Attivo totale investito = Attivo Totale – Passivo Corrente.
Shin et al. (2002)	<u>9 variables:</u> Value added to total asset; Net income to stockholder's equity; Quick ratio; Liquidity ratio; Current liability / Total assets; Retained earnings / Total assets; Stockholders' equity / Total assets; Financial expenses / Sales; Operating income / Operating expenses.
D'Annunzio (2004)	<u>10 variables:</u> Cash flow / Attivo totale; Leverage; Cash flow / Debiti a breve; Cash flow / Oneri finanziari; Attività correnti / Passività correnti; Fondo di Capitale disponibile netto / Attivo totale; Free cash flow / Attivo totale; Debiti a breve / Attivo totale; Oneri finanziari / Valore aggiunto; Variazione % Valore aggiunto d'impresa / Variazione % Valore aggiunto di settore (o stesso indice su fatturato).
Lee et al. (2005)	<u>5 variables:</u> EBIT / Total assets; Market capitalization / Total debt; Retained earnings / Total assets; Sales / Total assets; Working capital / Total assets.
Angelini et al. (2007)	<u>16 variables:</u> Cash flow / Total debt; Turnover / Inventory; Current liability / Turnover; Equity / Total assets; Financial costs / Total debts; Net working capital / Total assets; Trade accounts receivables / Turnover; Value added / Total assets; Utilized credit line / Accorded credit line; Unsolved effects (quantity) / Under usual reserve effects (quantity); Unsolved effects (value) / Under usual reserve effects (value) and with the overall Italian Banking System; Transpassing short-term / Accorded credit line short-term; Transpassing medium-long term / Accorded credit line medium-long term; Utilized credit line short-term / Accorded credit line short-term; Utilized credit line medium-long term / Accorded credit line medium-long term.
Davlos et al. (2007)	<u>9 variables:</u> Cash flow / Total assets; Current liabilities / Total assets; Current assets / Operating revenue; Current assets / Total assets; Net income / Equity;

	Retained earnings / Total assets; Profit / Operating expenses; Operating expenses / Revenue; Total liabilities / Total assets.
Etemadi et al. (2009)	<i>5 variables:</i> Operational income / Sales; Total liability / Total assets; Sales / Current assets; Interest expense / Gross profit; Quick assets / total assets.
Pierri et al. (2011)	<i>3 variables:</i> EBITDA / Vendite (%); Rotazione del capitale circolante lordo (volte); Indice di disponibilità.
Jakubik et al. (2011)	<i>7 variables:</i> Debt / Equity; (Long-term debt + long-term bonds) / Equity; Interest coverage; Return on equity; Gross profit margin; Cash ratio; Inventory ratio.
Giordini (2013)	<i>8 variables:</i> ROE; ROI; EBITDA / Turnover; Interest charges / EBITDA; Cash flow / Total debts; Financial debts / Equity; Total debts / EBITDA; Current ratio.

A3 The NetLogo code of models №3

```

turtles-own [codimp annorif datacost area_geografica regione tipologia_bil
debiti_finanziari_stimati default_esteso_2012 default_esteso_2013 default_esteso_2014
default_legale_2012 default_legale_2013 default_legale_2014 V03001 V03002
V03003 V03004 V03005 V03006 V03007 V03008 V03009 V03010 V03011 V03012
V03013 V03014 V03015 V03016 V03017 V03018 V03019 V03020 V03021 V03022
V03023 V03024 V03025 V03026 V03027 V03028 V03029 V03030 V03031 V03032
V03033 V03034 V03035 V03036 V03037 V03038 V04001 V04002 V04003 V04004
V04005 V04006 V04007 V04008 V04009 V04010 V04011 V04012 V04013 V04014
V04015 V04016 V04017 V04018 V04019 V04020 V04021 V04022 V04023 V04024
V04025 V04026 V04027 V04028 V04029 V04030 V04031 V04032 V04033 V04034
V04035 V04036 V04037 V04038 V04039 V04040 V06001 V06002 V06003 V06004
V06005 V06006 V06007 V06008 V06009 V06010 V06011 V06012 V06013 V06014
V06015 V06016 V06017 V06018 V06019 V06020 V06021 V06022 V06023 V06024
V06025 V06026 V06027 V06028 V06029 V06030 V06031 V06032 V06033 V06034
V06035 V06036 V06037 V06038 V06039 V06040 V06041 V06042 V06043 V06044
loan_amount
;Choose one of the two sets of indicators
;First collection of indicators used in models n3.1
cash/ta wc/ta cash/curliab cash/sales ebit/sales roa e/ta e/curliab sales/ta
ebitda/curliab score
;Second collection of indicators used in models n3.2
roa roe ebit/sales sales/curass va/ta ta/curliab wc/e curass/curliab e/curliab
ltfund/fixedass score
]

globals [fitness profit lost_business gains]

;In setup procedure 1013 turtles are created; each of them is endowed with 135 parameters
to setup
clear-all
clear-output
reset-ticks
set gains 0
set lost_business 0
if seed != 0 [random-seed seed]
create-turtles 1013 [setxy random-xcor random-ycor set color 9.9]

```

```

file-open "datasetfornetlogo.txt"
while [not file-at-end?]
    [ask turtle file-read [set codimp file-read
        set annorif file-read
        set datacost file-read
        set area_geografica file-read
        set regione file-read
        set tipologia_bil file-read
        set debiti_finanziari_stimati file-read
        set default_esteso_2012 file-read
        set default_esteso_2013 file-read
        set default_esteso_2014 file-read
        set default_legale_2012 file-read
        set default_legale_2013 file-read
        set default_legale_2014 file-read
        set V03001 file-read
        set V03002 file-read
        set V03003 file-read
        set V03004 file-read
        set V03005 file-read
        set V03006 file-read
        set V03007 file-read
        set V03008 file-read
        set V03009 file-read
        set V03010 file-read
        set V03011 file-read
        set V03012 file-read
        set V03013 file-read
        set V03014 file-read
        set V03015 file-read
        set V03016 file-read
        set V03017 file-read
        set V03018 file-read
        set V03019 file-read
        set V03020 file-read
        set V03021 file-read
        set V03022 file-read
        set V03023 file-read
        set V03024 file-read
        set V03025 file-read
        set V03026 file-read
        set V03027 file-read
        set V03028 file-read
        set V03029 file-read
        set V03030 file-read
        set V03031 file-read
        set V03032 file-read
        set V03033 file-read
        set V03034 file-read
        set V03035 file-read
        set V03036 file-read
        set V03037 file-read
        set V03038 file-read
        set V04001 file-read

```

set V04002 file-read
set V04003 file-read
set V04004 file-read
set V04005 file-read
set V04006 file-read
set V04007 file-read
set V04008 file-read
set V04009 file-read
set V04010 file-read
set V04011 file-read
set V04012 file-read
set V04013 file-read
set V04014 file-read
set V04015 file-read
set V04016 file-read
set V04017 file-read
set V04018 file-read
set V04019 file-read
set V04020 file-read
set V04021 file-read
set V04022 file-read
set V04023 file-read
set V04024 file-read
set V04025 file-read
set V04026 file-read
set V04027 file-read
set V04028 file-read
set V04029 file-read
set V04030 file-read
set V04031 file-read
set V04032 file-read
set V04033 file-read
set V04034 file-read
set V04035 file-read
set V04036 file-read
set V04037 file-read
set V04038 file-read
set V04039 file-read
set V04040 file-read
set V06001 file-read
set V06002 file-read
set V06003 file-read
set V06004 file-read
set V06005 file-read
set V06006 file-read
set V06007 file-read
set V06008 file-read
set V06009 file-read
set V06010 file-read
set V06011 file-read
set V06012 file-read
set V06013 file-read
set V06014 file-read
set V06015 file-read
set V06016 file-read

```

set V06017 file-read
set V06018 file-read
set V06019 file-read
set V06020 file-read
set V06021 file-read
set V06022 file-read
set V06023 file-read
set V06024 file-read
set V06025 file-read
set V06026 file-read
set V06027 file-read
set V06028 file-read
set V06029 file-read
set V06030 file-read
set V06031 file-read
set V06032 file-read
set V06033 file-read
set V06034 file-read
set V06035 file-read
set V06036 file-read
set V06037 file-read
set V06038 file-read
set V06039 file-read
set V06040 file-read
set V06041 file-read
set V06042 file-read
set V06043 file-read
set V06044 file-read]]

file-close
end

;Colors used: white=9.9; red=15; yellow=45; green=65
to go
  if ticks <= 4 [
    if ticks = 0 [ setturtles ]
    if ticks = 1 [ default2012 ]
    if ticks = 2 [ default2013 ]
    if ticks = 3 [ default2014 ]
  ]
  if ticks >= 1 [
;Fitness function n1
    set fitness (alpha * count turtles with [color = 15] + (1 - alpha) * count turtles with [color
= 9.9])
;Fitness function n2
    let index sum [loan_amount] of turtles with [color = 65]
    set gains gains + index * (interest_rate_in% / 100)
    let losses sum [loan_amount] of turtles with [color = 15]
    set profit gains - losses
;Fitness function n3
    let ind sum [loan_amount] of turtles with [color = 9.9]
    set lost_business lost_business + ind * (interest_rate_in% / 100)
  ]
;Calculating type I error in % - number of firms that were granted a loan but then went bankrupt
  output-type "typeIerror " output-print precision (100 * (count turtles with [color = 15] /
(count turtles with [color = 65] + count turtles with [color = 15]))) 2

```

;Calculating type II error in % - number of firms that were not granted a loan but then did not go bankrupt

```
output-type "typellerror " output-print precision (100 * (count turtles with [color = 9.9] /
(count turtles with [color = 9.9] + count turtles with [color = 45]))) 2
```

;Calculating % of succesful applicants - number of firms received a loan to the total number of firms

```
output-type "% of succesful applicants " output-print precision (100 * ((count turtles with
[color = 65] + count turtles with [color = 15]) / count turtles )) 2
```

;Calculating classification accuracy - number of firms correctly classified as solid and unsolid

```
output-type "% of correct decision " output-print precision (100 * ((count turtles with [color
= 65] + count turtles with [color = 45]) / count turtles )) 2
```

```
output-type "____next tick" output-print ("____")
```

```
tick
```

```
end
```

;Turtles with the score values higher than the threshold are granted loans and change the color from white to green

to setturtles

```
ask turtles [
```

```
if V04015 = 0 [die]
```

```
set loan_amount min ( list (max list 10 (V04015 * 0.357) ) 220216.9)
```

;Defining indicators from the first set – for model n3.1

```
set cash/ta (V03020 / V03023)
```

```
set wc/ta (((V03022 - V04035) / V03023))
```

```
set cash/curliab (V03020 / V04035)
```

```
set cash/sales (V03020 / V06001)
```

```
set ebit/sales (V06019 / V06001)
```

```
set roa (V06019 / V03023)
```

```
set e/ta (V04015 / V03023)
```

```
set e/curliab (V04015 / V04035)
```

```
set sales/ta (V06001 / V03023)
```

```
set ebitda/curliab (V06044 / V04035)
```

;Normalized values of indicators – for model n3.1

```
set cash/ta ((V03020 / V03023) / 0.7079414556)
```

```
set wc/ta (((V03022 - V04035) / V03023)) / 1.7047083015)
```

```
set cash/curliab ((V03020 / V04035) / 2.5454059620)
```

```
set cash/sales ((V03020 / V06001) / 1.2980895929)
```

```
set ebit/sales ((V06019 / V06001) / 1.9199263325)
```

```
set roa ((V06019 / V03023) / 1.9992375844)
```

```
set e/ta ((V04015 / V03023) / 2.0227091990)
```

```
set e/curliab ((V04015 / V04035) / 9.3233548324)
```

```
set sales/ta ((V06001 / V03023) / 4.9126127108)
```

```
set ebitda/curliab ((V06044 / V04035) / 3.8689818607)
```

```
set score (w1 * cash/ta + w2 * wc/ta + w3 * cash/curliab + w4 * cash/sales
+ w5 * ebit/sales + w6 * roa + w7 * e/ta + w8 * e/curliab + w9 * sales/ta
+ w10 * ebitda/curliab)
```

;Defining indicators from the second set – for model n3.2

```
set roa V06019 / V03023
```

```
set roe V06037 / V04015
```

```
set ebit/sales V06019 / V06001
```

```
set sales/curass V06001 / V03022
```

```
set va/ta V06009 / V03023
```

```
set ta/curliab V03023 / V04035
```



```

set wc/e (V03022 - V04035) / V04015
set curass/curliab V03022 / V04035
set e/curliab V04015 / V04035
set Itfund/fixedass V04025 / V03010
;Normalized values of indicators – for model n3.2
set roa ((V06019 / V03023) / 1.9992375844)
set roe ((V06037 / V04015) / 43.25)
set ebit/sales ((V06019 / V06001) / 1.9199263325)
set sales/curass ((V06001 / V03022) / 8.7079346479)
set va/ta ((V06009 / V03023) / 1.1992945452)
set ta/curliab ((V03023 / V04035) / 20.6287730833)
set wc/e (((V03022 - V04035) / V04015) / 91.9394)
set curass/curliab ((V03022 / V04035) / 8.1313623317)
set e/curliab ((V04015 / V04035) / 9.3233548324)
set Itfund/fixedass ((V04025 / V03010) / 74.5789473684)

set score (w1 * roa + w2 * roe + w3 * ebit/sales + w4 * sales/curass + w5 * va/ta
+ w6 * ta/curliab + w7 * wc/e + w8 * curass/curliab + w9 * e/curliab
+ w10 * Itfund/fixedass)

if score >= threshold [set color 65]
]
end

;Check for default in 2012, 2013 and 2014.
;If a firm that was granted a loan defaults - the color changes from green to red.
;If a firm that was not granted a loan defaults - the color changes from white to yellow.
to default2012
ask turtles [
let index max ( list default_esteso_2012 default_legale_2012)
if index = 1 [
if color = 9.9 [set color 45]
if color = 65 [set color 15]
]]
end

to default2013
ask turtles [
let index max ( list default_esteso_2013 default_legale_2013)
if index = 1 [
if color = 9.9 [set color 45]
if color = 65 [set color 15]
]]
end

to default2014
ask turtles [
let index max ( list default_esteso_2014 default_legale_2014)
if index = 1 [
if color = 9.9 [set color 45]
if color = 65 [set color 15]
]]
end

```