Expert Systems with Applications 36 (2009) 4736–4744

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Consumer credit scoring models with limited data

Maja Šušteršič [a,*], Dušan Mramor [b], Jure Zupan [c]

[a] Petrol d.d., Ljubljana, Dunajska c. 50, 1000 Ljubljana, Slovenia
[b] Faculty of Economics, University of Ljubljana, Kardeljeva pl. 17, 1000 Ljubljana, Slovenia
[c] National Institute of Chemistry, Ljubljana, Hajdrihova ul. 19, 1000 Ljubljana, Slovenia

ARTICLE INFO

Keywords:
Consumer credit scoring
Neural networks
Genetic algorithm
Principle component analysis
Variable selection

ABSTRACT

In this paper we design the neural network consumer credit scoring models for financial institutions where data usually used in previous research are not available. We use extensive primarily accounting data set on transactions and account balances of clients available in each financial institution. As many of these numerous variables are correlated and have very questionable information content, we considered the issue of variable selection and the selection of training and testing sub-sets crucial in developing efficient scoring models. We used a genetic algorithm for variable selection. In dividing performing and nonperforming loans into training and testing sub-sets we replicated the distribution on Kohonen artificial neural network, however, when evaluating the efficiency of models, we used *k*-fold cross-validation. We developed consumer credit scoring models with error back-propagation artificial neural networks and checked their efficiency against models developed with logistic regression. Considering the dataset of questionable information content, the results were surprisingly good and one of the error back-propagation artificial neural network models has shown the best results. We showed that our variable selection method is well suited for the addressed problem.

1. Introduction

Financial institutions manage credit risks for businesses and consumers differently. In spite of the fact that procedures for granting loans to businesses are less universal, quantitative business credit scoring models were developed first (Altman, 1968; Beaver, 1966) mainly due to a wider availability of company data. There has been an impressive development from their first introduction to their later forms (Altman, 1993; Goonatilake & Treleaven, 1995; Hand, 1998; Trippi & Turban, 1996). In the past, due to the limited number of usually standardized types of consumer loans and scarce availability of data financial institutions predominantly used simple subjective qualitative methods to evaluate creditworthiness of consumer loan applicants (i.e. Sinkey, 1992).[1]. Quantitative consumer credit scoring models were developed much later than those for business credit mainly due to the problem of availability of data. In many countries legal (privacy protection) and other reasons prevented the buildup of publicly available databases. Data were limited to the own databases of financial institutions. Nowadays, some data are publicly available in several countries and financial institutions and researchers have developed many different quantitative credit scoring techniques.

Classical statistical methods that are used to develop credit scoring models are linear discriminant analysis, linear regression, logit, probit, tobit, binary tree and minimum method (Baesens, Setiono, Mues, & Vanthienen, 2003a; Baesens et al., 2003b; Thomas, 1998; West, 2000). The two most commonly used are linear discriminant method (LDA) and logistic regression (Baesens et al., 2003b; Desai, Crook, & Overstreet, 1996; Lee & Chen, 2005; Lee, Chiu, Lu, & Chen, 2002; Thomas, 2000; West, 2000). The weakness of the linear discriminant analysis is the assumption of linear relationship between variables, which is usually nonlinear and the sensitivity to the deviations from the multivariate normality assumption. The logistic regression is predicting dichotomous outcomes and linear relationship between variables in the exponent of the logistic function, but does not require the multivariate normality assumption. Because of the linear relationship between variables both LDA and logistical regression are reported to have a lack of accuracy (Thomas, 2000; West, 2000). On the other hand there are also studies showing (Baesens et al., 2003b), that most of the consumer credit scoring datasets are only weakly nonlinear and because of that LDA and logistical regression gave good performance.

There are also more sophisticated models known as artificial intelligence: expert systems, fuzzy systems, neural networks and genetic algorithms. Among these the neural networks are very

* Corresponding author. Tel.: +386 31 66 88 00/1 4714423; fax: +386 1 4366865.
E-mail address: maja.sustersic@petrol.si (M. Šušteršič).

[1] At this stage of the development the use of qualitative data was logical. It was shown that even for micro-companies their accounting data do not contain much of information, that could be used for bankruptcy prediction (see Mramor & Valentincic, 2003).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.06.016

promising (Goonatilake & Treleaven, 1995) and the alternative to the LDA and logistic regression, due to the possible complex nonlinear relationship between variables. In the literature in most cases of credit scoring problems the neural networks are more accurate than LDA and logistic regression (Desai et al., 1996; Jensen, 1996; Lee et al., 2002; Piramuthu, Shaw, & Gentry, 1994; Richeson, Zimmermann, & Barnett, 1996; West, 2000). The neural networks have their weaknesses in their long training process, and after obtaining the optimal network's architecture, the model acts as a "black box" and there is not easy to identify the relative importance of potential input variables. One can find also a few studies with genetic algorithms (Kim & Sohn, 2004; Walker, Haasdijk, & Gerrets, 1995), but in the last years the hybrid systems seem to be the most promising (Hsieh, 2005; Lee & Chen, 2005; Lee et al., 2002).

The datasets for the mentioned studies were usually collected by credit unions. They consisted of a relatively small number of variables: from 5 to 20. As these were the only available variables and as their selection was done by credit unions on the basis of past consumer loan experiences of financial institutions, researchers did not regard selection of variables as a crucial step of the model development. Because of the relative small number of variables they used all of them or their selection was based mainly on classical statistical methods like *t*-test or chi-square-test (Avery, Calem, & Canner, 2004; Kim & Sohn, 2004), multivariate adaptive regression splines (Lee & Chen, 2005) or artificial neural network (Glorfeld & Hardgrave, 1996; Hsieh, 2005; West, 2000). The weaknesses of the statistical methods usually appear when multicollinearity between a large number of variables exists and in the case of neural networks in their time consuming process especially when large number of variables exists. The highest number of variables that we found in the literature was 57 (Jacobson & Roszbach, 2003). The authors included publicly available or governmentally supplied variables, such as sex, citizenship, marital status, postal code, taxable income, taxable wealth, house ownership and variables reported by the Swedish banks like the total number of inquiries made about an individual and the number of unsecured loans and the total amount of unsecured loans. Most of the variables (41) were not used for the development of the model, because either they lacked a bivariate relation with dependent variable or displayed extremely high correlation with another variable that measured approximately the same thing but had greater explanatory power.

Contrary to the previous research, we developed consumer credit scoring models for financial institutions where data that were used in previous research are not available. We base our model selection primarily on accounting data on transactions and account balances of clients that are readily available in each financial institution. Therefore, the number of input variables is in our study larger than in other studies, many of the variables are highly correlated and for a great majority we do not know how much creditworthiness information (if any) they contain as they are currently not used in credit assessments. Hence, the issue of variable selection is a crucial and a challenging problem to solve before different credit scoring techniques are used to develop the best performing models. As it is known, different variable selection methods give different results on the same dataset. To increase the quality of variable selection we compare a statistical principal component analysis (PCA) with a nonstatistical genetic algorithm. For the genetic algorithm we divided performing and nonperforming loans into training and testing sub-sets randomly and in such a way, that both types of loans proportionally covered the whole Kohonen neural network, however, when evaluating the efficiency of models, we used *k*-fold cross-validation (Hsieh, 2005). The efficiency of models using only principle component variables was smaller.

We developed consumer credit scoring models with logistic regression and error back-propagation artificial neural networks.[2] Considering questionable information content of the dataset that we use, the results of the models are surprisingly good – prediction power of our models is approximately the same or even better than those of the latest studies. Error back-propagation neural network model using variables selected by genetic algorithm is showing the best results.

We start with short explanations of the methods, the research procedure and the data used in this study. The description of the selection of variables and the division of the master dataset into training and testing sub-sets follows. Different models and their results concerning efficiency of consumer loans classification are presented next and followed by the conclusion.

2. Principal component analysis, genetic algorithm and neural networks

Principal component analysis (PCA) is an effective transformation method for reduction of a large number of correlated variables where variable selection is hard to achieve. Namely, the result of PCA is a set of new independent variables that can be directly used by credit scoring techniques. PCA is a statistical method used frequently for reducing the dimensionality of a given dataset of correlated variables while maintaining as much of the variables' variability as possible. This efficient reduction of the number of variables is achieved by obtaining orthogonal linear combinations of the original variables – the so-called principal components (PCs). This is possible with a transformation of the co-ordinate system to a new one. The transformation is done by rotation of the old co-ordinate system into the new one in such a way that the most of the relevant information is collected around smaller number of new axes (PCs). The first principal component PC_1 preserves most of the remaining variability in the original variables, the second component PC_2 preserves the second most variability existing in the original variables, and so on. Each PC is an eigenvector of the variance–covariance matrix of the original variables. This analysis provides two important outputs: the percentage of variance explained by *i*th principal component PC_i and the correlations between each PC and the original variables. The first one is computed by dividing the eigenvalue associated to the corresponding PC by the total sum of the eigenvalues. The first output provides the importance of the component in the terms of the variability of the original variables (see Godoy & Stiglitz, 2006).

Genetic algorithm (GA) is an efficient optimization procedure. The basic principle of the genetic algorithm is inspired by the mechanisms of biological evolution. The main idea of a genetic algorithm is to start with a population of possible solutions to a given problem, and to continue by a production of series of new generations of many different solutions, assuming to find better and better ones. Genetic algorithm operates through a simple cycle consisting of the following four stages: creation of the population, evaluation, selection, and reproduction in which the last three stages are cycled until no more improvement in the evaluation stage is detected.[3]

The starting point of genetic algorithm is the creation of a population of "members" which represent candidate solutions to the problem being solved. The members (candidate solutions) are evaluated by the fitness function. This assesses the degree to which the solutions are good at solving the given problem. The value returned by this fitness function is used for the selection of members as "parents" for the production of the next generation (population)

[2] We decided to examine these two types of models, because they were most promising according to previous research (see Sustersic, 2001).

[3] For more details and applications for finance and business see Goonatilake and Treleaven (1995) and Hand (1998).

of solutions. The higher is the fitness function, the higher is the probability for a member to be selected as a parent. In the reproduction stage a completely new set of members of the new population is created from the parents through the application of genetic operators, crossover and mutation.

Artificial neural networks (ANNs) are a set of methods designed for solving many different problems from classification to modeling and optimization. Each individual ANN system is comprised of large number of highly interconnected, interacting processing units that are based on neuro-biological models. The essential features of ANN are processing units (the neurons or nodes – we will use the term neurons thereafter) and the learning algorithm used to find values of the ANNs parameters, called weights, for a particular problem. The neurons are connected to one another so that the output from one neuron can be the input to many other neurons. Each neuron transforms a multivariate input to a single value output using a predefined simple function. In most cases the form of this function is identical in all neurons, however in each set of parameters (weights) in this function are different for each neuron. The values of the weights are determined with the training sub-set consisting of data with known inputs and outputs. Network architecture is the organization of neurons and the type of connections permitted. The neurons are arranged in a series of layers with connections between neurons and other layers, but not between neurons in the same layer. The layer receiving the inputs is called the input or the first layer. The final layer providing the target output signal or answer is the output layer. Any layers between these two layers are called hidden layers.

The process of adjusting the weights to make the ANN learn the relationship between the inputs and targets is called learning or training. ANNs are divided according to the type of the learning algorithm, which can be supervised or unsupervised. At supervised learning ANN is presented a set of input and target data for all objects. The correction of the network's weights is made after each single object is sent through ANN and the produced output is compared to the actual target. In each iterative step, known as one epoch, the network's answers are compared with the targets for all objects in the training sub-set and the total error of one epoch is recorded. The training procedure is repeated till the defined acceptable mean square error (MSE) of one epoch or a prespecified number of epochs are achieved. Beside this two mentioned parameters (MSE, number of epochs) other parameters of ANN architecture should be determined. The design parameters include number of input neurons, number of output neurons, number of hidden layers,[4] number of neurons in each hidden layer,[5] and activation function selected.

On the contrast to the supervised learning in the unsupervised trained ANN the training sub-set of data does not contain the targets (answers or solutions) – only the representation of objects. Therefore such ANNs are employed for exploration of internal properties of data, such as clusters and not for modeling. The simplest unsupervised ANN is one layer Kohonen ANN. The algorithm of unsupervised learning at this network uses the principle "the winner takes all". For each input there is only one winner in the entire network. The winner is the neuron having the weights most similar to the input variables. The correction of weights during the learning process does not affect all neurons in the network, but only the winner's and those of the neighbors of the winner. Kohonen ANN is very useful for solving problems as grouping in one or two levels, classifications and transformations from multidimensional to two- or three-dimensional space etc. (Zupan & Gasteiger, 1993). If groups are well separated a logical criteria can be determined for each variable and the neural network is not a black box anymore. For this study we used 12×12 Kohonen ANN to observe the distribution of the objects from the database in the variable space projected on the 12×12 top-map.

In the study we use error back-propagation artificial neural network (EBP ANN) that according to the previous studies (Desai et al., 1996; Lee et al., 2002; Sustersic, 2001; West, 2000) gives the best results and is widely used at credit scoring models. The characteristic of EBP ANN is that weights in the learning process are changed in the opposite direction as input is traveling through the network. That means that weights in the output layer are changed first, than the weights of the hidden layers and at the end the weights of the first or input layer. The learning is supervised. At EBP ANN learning procedure the learning rate and the momentum are important. Learning rate defines ratio of the weights' change after the actual correction is evaluated. The momentum term enables that the adoption of weights in the network during the training avoids the local minimum. One of the drawbacks of EBP ANN is that it can be easily over-trained. Over-training appears if after a number of iterations, that are improving predictions on the training sub-set, the network starts yielding worse and worse predictions. For EBP ANN the architecture is crucial. Too large number of layers and/or neurons in these layers, too long training or inadequate choice of the training sub-set, can easily cause over-training of the network. Therefore, the architecture of the network should be as small as possible. The architecture of EBP ANN depends on the number of input–output variables and number of objects that are available for designing a model. Most of the authors recommend one hidden layer network to model a complex system with any desired accuracy (Hsieh, 2005), and others recommend that the number of weights do not exceed the number of objects in the training sub-set.

ANNs can be used for a number of problems in the field of accounting, finance, human resource, marketing, organization and others (Hsieh, 2005; Lee & Chen, 2005; Turban & Aronson, 1998).

3. The research procedure

The research was divided into four phases (Fig. 1). The first phase represents variable reduction from initial 84 to 67 variables. However, a relatively large number of variables still remained and that required a thorough analysis and selection before designing an optimal model. Therefore, the second phase consisted of a detailed analysis of each variable, variable selection and of the selections of various training and testing sub-sets from the entire set of 581 data objects (loan applicants). The PCA and GA were used for variables selection. The use of GA required the selection of the training and testing sub-sets. We used the distribution of objects (loan applicants) on the Kohonen 12×12 top-map for the selection of training and testing sub-sets. We compared the composition of the training and testing sub-sets with that of the equivalent sub-sets obtained with the random selection method.

The third phase was a design and optimization of the models. First ANN model (EB04) based on 21 normalized input variables was formed. This was expected to be the most efficient model on the basis of previous research (Sustersic, 2001). To test its quality, also a logit model (LOGIT01) was developed with the same training

[4] As mentioned in the literature (Baesens et al., 2003b; Hsieh, 2005; Lee et al., 2002) one-hidden layer network is sufficient to model any complex system with any desired accuracy. Desai et al. (1996) show that in such case the training time may be very high, but can be reduced by adding a second hidden layer.

[5] A number of neurons in hidden layer is also not well defined in the literature. Hsieh (2005) introduced a search method that starts with the number of hidden neurons equal to the number of inputs divided by two. Then neurons were gradually added to the hidden layer one at a time. The search process stopped when there was no further improvement in network performance. In this study the maximum number of neurons is rarely required to exceed the number of inputs by more than two times. The second rule that can be followed in determining the number of neurons is that the number of weights in the network does not exceed the number of the objects in the training sub-set.

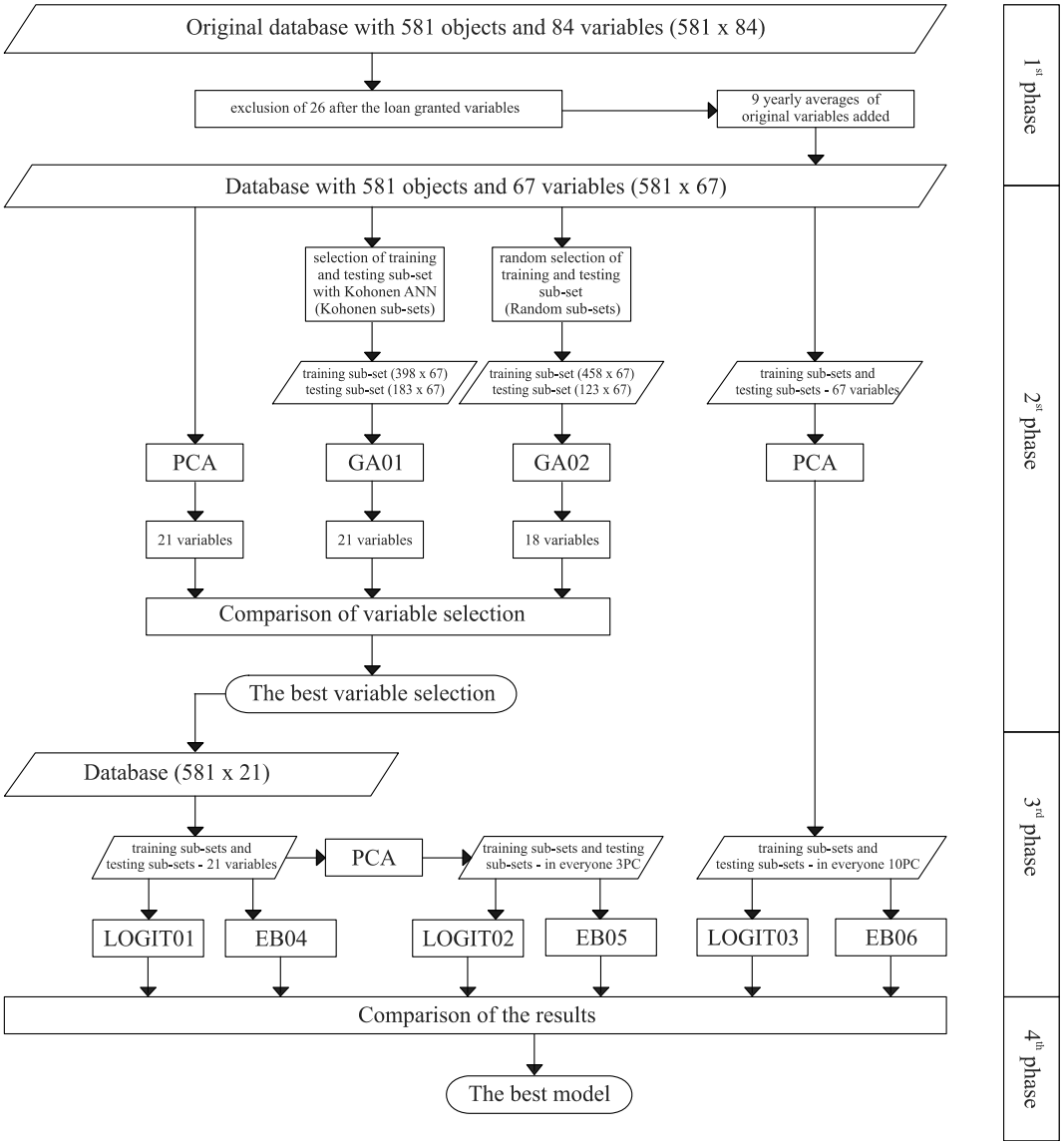

Fig. 1. Research procedure in four phases, where PCA stands for principal component analysis, GA for genetic algorithm, LOGIT for logit model and EB for error back-propagation neural network.

objects represented by the same 21 variables. The second set of models (EB05, LOGIT02) was based on training objects represented by only three variables, i.e. three PCs calculated from the 21 normalized selected variables. The third set of models (EB06, LOGIT03) was developed on objects represented by 10 PCs as input variables that were calculated from all 67 normalized selected variables obtained in the first phase. Evaluation of the results with the *k*-fold cross-validation was the fourth phase of the study. Comparing the first set of models (EB04 and LOGIT01) with the second set (EB05 and LOGIT02) enabled us to find out whether the models with PCs as input variables perform better than the models with normalized input variables. The comparison of the first and the third set (EB06 and LOGIT03) of these models showed us whether the selection of 21 variables from 67 was efficient or not. The comparison of efficiency among ANN models gave us the best ANN model, which we further compared to the statistical logit models.

4. Data

The database for this study was created by a Slovenian bank that merged all the accounting and a few other internal bank data available for 581 short term consumer loans granted to its existing and new clients in the period 1994 to 1998. The database does not include the information on the rejected applications. Developing credit scoring models only on the data of accepted customers is biased and well known in the literature (i.e. Caouette, Altman, & Narayanan, 1998; Thomas, 1998) with reject reference used to reduce the bias. In this study no such problem arises, as the rejected population consisted almost only of those loan applicants that did not fulfill the following simple legal criterion: a loan cannot be granted to the client that has or would have with a new loan the total monthly loan payments in excess of 1/3 of monthly salary.[6] In this two stage loan approval process we are developing models only for the loan applicants in the second stage.

[6] In the past in Slovenia the insurance companies secured the great majority of loans granted and consumer credit scoring was not relevant for the banks. However, in transition to the market based economy the system was changing. As a consequence to constantly decreasing security of employment and salary, the insurance companies were increasing insurance premiums. Thus, the costs of borrowing of good customers were inadequately increasing and with introduction of foreign bank competition the requirement for better consumer credit scoring of domestic banks became imminent.

The credit behavior of the client, to whom the loan was granted, was defined by dichotomous variable with value 1 if all liabilities from the loan were paid in time (performing loans) and with value 0 if this was not the case (nonperforming loans). From 581 loans 401 (69.0%) were performing and 180 (31.0%) were nonperforming.[7] Performing loans in our database were randomly selected from all performing loans the bank granted in that period and the same applies for nonperforming loans respectively. The characteristics of each client (the object of our study) were in the original database described by 84 variables. They referred to client's sex and age, characteristics of the loan, credit history with the bank before the loan was granted and detailed data on accounts balances and transactions with the bank. Twenty-six variables were describing the characteristics of clients in the period after the loan was granted. Because such variables are not available to the bank when loan decision is made, they are useless in real applications, hence we did not use them in our study. After we added nine new variables calculated as yearly averages of accounts balances from the remaining original variables (original data were only quarterly) we formed the database with 67 numerical and character variables presented in Appendix.

4.1. Selection of variables

The construction of ANN model for credit scoring with large number of variables means large number of neurons in the ANN architecture and consequently a time consuming learning and the optimization process. Besides, the variables with smaller information content and co-linear variables would create "noise", and the model would be less accurate. This is the main reason why an appropriate selection of a smaller number of variables was a crucial part of the study.

The second reason is the questionable information content of variables. We base our model selection primarily on accounting data about the transactions and account balances of clients that are readily available in each financial institution and not on some carefully selected variables in databases of credit agencies. Therefore, the number of input variables in our study is higher than in other studies, many of the variables being correlated while for a great majority of them no information on their relevance to the problem exists, as they are currently not used in credit assessments.

The selection of variables with statistical methods suggested in the literature was not used in the study, because of the known weaknesses. For example, if the co-linearity is determined with *F*-test, the variables are introduced step by step to multiregression model and in each step the *F*-test is calculated. If the added variable is significant, it is included to the model, otherwise not. The problem of this approach is that different results are achieved, when variables are added to the model in a different order (way). The next most known statistical method for the determination of the correlation between two variables is a correlation matrix. The weakness of the correlation matrix is the determination of multicollinearity, when the model has more than two specific, noncorrelated variables (Gujarati, 1995). Besides weaknesses of conventional statistical methods it is also widely known, that using different variable selection methods gain different results for the same database (Hsieh, 2005; Kim & Sohn, 2004; West, 2000), which lead us to the decision to compare a statistical PCA and a nonstatistical genetic algorithm (GA).

The original set of variables was first analyzed (e.g. minimum and maximum values, average, standard deviation, median) and the normalization of variables was performed by "Minmax" or "Auto scaling" normalization method.[8] Since ANNs, GA and statistical methods accept only numerical inputs, each character variable was transformed into a number and encoded either as 0.5 or 0, to reduce the problem of numerically imputing inappropriate weights to character values of these variables. Most of the character variables were yes/no, except sex.

After normalization the selection of variables was performed with PCA and GA.[9] In the complete set of objects (loan applicants) represented by 67 original variables PCA shows that first nine principle components (PCs) are carrying 90% of all information.[10] The second PCA output (Fig. 2) shows the distributions of original variables in the space of two or three PCs (plot of loadings), from which the most significant variables can be selected.[11] The plot of loadings was observed for the first nine PCs. The variables that were outside of the square have coefficient greater than 0.15 and were recognized as significant. The coefficient was determined so that the number of selected variables was approximately the same as with GA method. The described logic was then used to determine the significant variables (on all plots of loadings in the two dimensional space: PC1 vs. PC2, PC3, ..., PC9; PC2 vs. PC3, PC4, ..., PC9, etc.). The result was 21 selected variables with PCA method.

For GA variable selection process first objects for training and testing sub-sets were selected by using two methods: the distribution on the Kohonen ANN (Kohonen sub-sets) and the random method (random sub-sets).[12] The results of GA variable selection were the members determined with 21 variables when Kohonen sub-sets were used and the members determined with 18 variables when random sub-sets were used (see Appendix).

We than compared the quality of variable selection of the three methods. To determine the relative quality of selection we pre-tested them by designing logit models from selected variables with each method using Kohonen training sub-set. We pre-tested them with Kohonen testing sub-set and the results are presented in Table 1.

Finally, 21 variables selected with GA – Kohonen sub-sets, presented in Appendix, were used for further study as they enabled the highest accuracy in pre-testing.

4.2. Training and testing sub-sets

The models were built on the training sub-set and then tested on the testing sub-set. For the selection of the objects into these two sub-sets two approaches were used. With the first approach we have used Kohonen ANN. The objects were selected according to the distribution of performing and nonperforming loans on the top-map of the Kohonen ANN. The idea of this approach is to select the objects for the training sub-set in such a way that both types of objects (performing and nonperforming) cover the whole Kohonen space of 12×12 neurons as uniformly as possible. With this approach 398 objects were selected for the training and 183 objects for the testing sub-set. The second approach was commonly used random selection where the training sub-set consisted of 458 objects and the testing sub-set of 123 objects. With these numbers of objects we achieved approximately the same number of nonperforming objects as in the Kohonen training and testing sub-sets, although the sub-sets were very different.

[7] The ratio between performing and nonperforming (69%:31%) loans in our database is comparable with ratios in German database (70%:30%) that was used in studies by Hsieh (2005), Baesens et al. (2003b) and Kim and Sohn (2004) and Australian database (68%:32%) used in study by Hsieh (2005).

[8] With "Minmax" method we convert variable in such a way that minimum value of variable is equal zero, maximum value of variable equals 1 and the values between those two became corresponding values between zero and 1. Autoscaling normalisation method converted values in such a way that the average of the variable is equal to zero and the standard deviation is equal ±1.

[9] We use the genetic algorithm developed by Zupan and Novič, 1999.

[10] For the calculation of PCA we used MatLab.

[11] In some cases one can observe also the "plot of scores" from which the most significant objects can be determined.

[12] The sub-set selection process is explained in the next section.

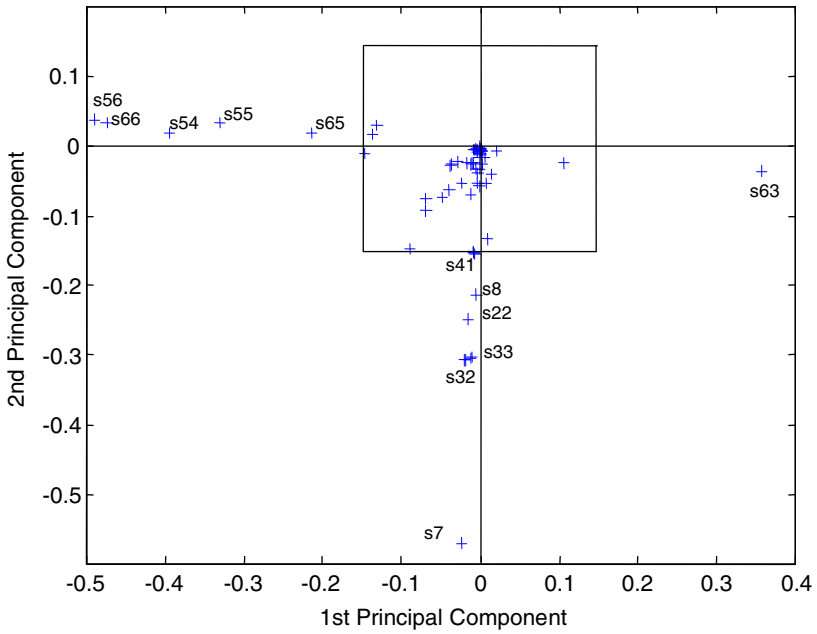

Fig. 2. Plot of loadings for 1st and 2nd principal component, where s8, s22, s32, ect. are variables (their detailed description is in Appendix). *Source*: Authors' calculations.

Table 1
Accuracy of logit models using variables selected by the three methods

Method of selection of variables	No. of selected variables	Accuracy (%)	Error Type II (%)	Error Type I (%)
Principal component analysis (PCA)	21	72.7	25.0	37.0
Genetic algorithm (GA) – Kohonen sub-sets	21	76.5	12.5	31.9
Genetic algorithm (GA)– random sub-sets	18	72.7	41.7	37.0

A proper selection of the training and testing sub-sets is important for the design of an optimal ANN architecture. The key is the optimal relation between the distributions of the performing and nonperforming loans. Due to the fact that the percentage of bad loans is usually small compared to the performing ones one can loose important information about loan applicants if they are selected randomly. We believe that much better distribution is obtained using Kohonen ANNs, especially if performing and nonperforming sub-sets are significantly different in size. It applies for the training as well as the testing sub-set. For smaller population the random method by the rule does not produce a distribution as good as for the larger group and is, therefore, inferior. For this reason we used Kohonen ANN for the selection of the sub-sets used in the determination of the optimal models – for determining the parameters of the models (number of neurons, number of hidden layers, etc.). However, when we were evaluating the efficiency of the models, we used *k*-fold cross-validation to avoid the biased selection of the sub-sets and to generate random partitions of the credit datasets.

5. Consumer credit scoring models

5.1. Neural network models

As we mentioned above, the ANN architecture is important for designing the accurate model. In ANN design it is important to optimize its parameters, which are different for different types of networks.

In the study several EBP ANNs with one or two hidden layers and a maximum of up to seven neurons in each hidden layer were investigated. The total number of neuron weights in all layers never exceeded the number of objects in the training sub-set (i.e. 398). The learning of the EBP ANN stopped when predetermined number of epochs was reached, the prespecified limit of the error obtained on the training sub-set was achieved, or the error of testing sub-set started to increase (over-training). The optimal EBP ANN was selected according to the following criteria: it should have the lowest mean square error (MSE), largest percentage of correct answers at critical value 0.5 and the lowest error type II.[13]

5.2. Logit model

The logit model is the most promising and widely used statistical credit scoring model. We designed it with the same training sub-set represented with corresponding number of variables. The forward procedure and default values of parameters were used.

We have also designed a logit model from Kohonen sub-sets described by all 67 variables, but when the *k*-fold cross-validation was performed it was obvious that the model is not stable. Test on multicollinearity showed that there were serious problems with it (condition indices were much higher than critical values). The pre-selection of variables was therefore demonstrated to be absolutely necessary.

6. Results

The relevance and the number of objects from the whole sample that are correctly classified determine the reliability of the model. It depends also on the chosen critical value. If the targets are described with two values – zero and one and the model turns the values between zero and one, then the accuracy should be highest at critical value 0.5. But at this critical value the errors of the model are not necessarily optimal. This depends on the bank's costs of granting a nonperforming loan (error type II) relative to the opportunity costs of not granting a performing loan (error type I). In this study the critical value (the threshold for decision of granting or rejecting the loan) of 0.6 turned out to be better than 0.5, as the

[13] The model is predicting a performing loan but it turns out nonperforming.

Table 2
Average accuracy and errors of testing groups with cross-validation process, where EB stands for error back-propagation neural network and LOGIT for logit model

Model	Average accuracy (%)	Standard deviation	Average error type II (%)	Average error type I (%)
EB04	79.3	0.069	17.8	29.9
EB05	73.0	0.066	11.7	39.2
EB06	70.7	0.057	15.6	42.4
LOGIT01	76.1	0.075	13.3	34.7
LOGIT02	71.3	0.068	16.1	41.6
LOGIT03	72.5	0.061	24.4	39.9

Source: Authors' calculations.

error type II is reduced considerably relative to increase of error type I. Average accuracy in the prediction of the model with the increased threshold practically did not change.

In determining the accuracy it is important to provide a reliable estimate and minimize the impact of data dependency in developing credit scoring models. To generate random partitions of the credit dataset (training and testing sub-sets) *k*-fold cross-validation was used. In this procedure, the credit dataset was divided into *k* independent groups. A model was trained using the first $k-1$ groups of samples and the trained model was tested using the *k*th group. This procedure was repeated until each of the groups has been used as a testing sub-set once. The overall scoring accuracy was reported as an average across all *k* groups. A merit of cross-validation is that the credit scorning model is developed with a large proportion of the available data and that all the data is used to test the resulting models. In this experiment the value of *k* was set to 20 and thus forms a 20-fold cross-validation. An estimate from 20-fold cross-validation is likely to be more reliable than an estimate from a common practice of using a singleton holdout set.

The results of EBP ANN models were compared with those of logit models. Table 2 summarizes the accuracy results of the models. The comparison of the results shows that EBP ANN models have the best accuracy and the lowest value for error type II.

For answering the question if our special GA (Kohonen sub-sets) selection of the 21 variables from 67 was efficient, the comparison of models EB04 with EB06 and LOGIT01 with LOGIT03 was required. In both cases the models constructed from 21 variables (EB04 and LOGIT01) perform better, having better average accuracy and significantly lower error type II than the models, which were constructed from 10 PCs selected by PCA from all 67 variables (EB06 and LOGIT03).

We have also tested, if the efficiency of ANN models increases with the reduction of input variables to less than 21. The selected variables (21) were analyzed with PCA and three PCs were selected as input to ANN models. The number of PCs as input was selected by following the criteria of variance smaller than 1%. The comparison of results between models with input of 21 selected normalized variables (EB04, LOGIT01) and with input of three PCs (EB05, LOGIT02) shows, that the overall performance of models with three PCs did not improve.[14]

7. Conclusion

The main goal of this paper is to develop comparably efficient consumer credit scoring models on a very different dataset than used in previous research. In many transition and developing countries credit agencies and credit bureaus do not exist and thus the relevant data on credit behavior of loan applicants are not available. Also financial institutions have not built relevant databases based on the past experiences with performing and nonperforming consumer loans. However, within financial institutions numerous data are available and among them consistently collected accounting data. The main research question was, how much information these data contain and how do we access this information.

Testing the models that we developed proved, that our decision to primarily search for an optimal variable selection procedure from a dataset of 67 variables yielded good results. For variable selection we used a PCA and a genetic algorithm based on the two methods of building training and testing sub-sets: on Kohonen artificial neural network and random method. We ended with 21 variables, selected with genetic algorithm using Kohonen sub-sets that were used in developing the credit scoring models.

For the development of models we used error back-propagation artificial neural networks and logistic regression. We tested the accuracy of prediction with *k*-fold cross-validation and error back-propagation neural network model showed the best average results: 79.3% accuracy, 17.8% error type II and 29.9% error type I. We have investigated also some other variable selection methods and obtained models with a lower predictive power. Considering our dataset with very questionable information content in comparison with other research, the results of the models are surprisingly good – prediction power of our best models is approximately the same as can be found in some of the latest studies.

The findings of this study are raising some very interesting questions for future research. Especially important is the question of the information content of data gathered by credit agencies and credit bureaus. Decisions on the data collected were mainly based on past consumer credit experiences of financial institutions but it seems, that either the same information is also contained in other data or the selection is not optimal. Therefore, the cluster analysis might be appropriate for the next step.

Acknowledgement

Authors wish to thanks to Anuska Ferligoj, Universitiy of Ljubljana, participants of International Conference: Applied Statistics 2006, Ribno (Bled), Slovenia, participants of the research seminar at Faculty of Economics, University of Ljubljana, and to Guofu Zhou, Washington University, St. Louis, the discussant at EFA 2007 for their helpful comments.

Appendix

Variables used in developing consumer credit scoring models and their selection, where superscript 1 represents variables selected with principal component analysis (PCA) method, 2 variables selected with genetic algorithm (GA) method, Kohonen sub-sets and 3 variables selected with genetic algorithm (GA) method, random sub-sets. Bold variables are selected variables.

Variable	Variable description
ID	Counter
Loan	Dependent variable (1 – performing, 0 – nonperforming)
s1[1]	Age
s2	Sex
s3	Number of matured and repaid loans in the year preceding loan application
s4	Sum of principle repayments in the year preceding loan application
s5[1]	Amount of loan approved
s6	Interest rate at loan approval date

[14] In the study of Sustersic (2001) conversion of selected variables into PCs slightly improved the results. However, variables were selected with combination of the three methods that caused lower average accuracy of the models. This implies that selection of variables was not optimal and the use of PCA somewhat improved the results. Even improved results were significantly inferior to those presented in Table 2.

Appendix (*continued*)

Variable	Variable description
s7[1,2,3]	Loan maturity in months
s8[1]	Payment method: client's money transfer
s9[3]	Payment method: bank automatically from transaction account
s10	Payment method: employer automatically from salary
s11	Subsidiary: 1 (=0.5, other =0)
s12[3]	Subsidiary: 2 (=0.5, other =0)
s13	Subsidiary: 3 (=0.5, other =0)
s14[2]	Subsidiary: 4 (=0.5, other =0)
s15[2]	Subsidiary: 5 (=0.5, other =0)
s16	Subsidiary: 6 (=0.5, other =0)
s17[3]	Subsidiary: 7 (=0.5, other =0)
s18[3]	Subsidiary: 8 (=0.5, other =0)
s19	Subsidiary: 9 (=0.5, other =0)
s20[3]	Subsidiary: 10 (=0.5, other =0)
s21[3]	Subsidiary: 11 (=0.5, other =0)
s22[1]	All subsidiaries in the centre (s11, s17, s18, s20, s21)
s23[2]	Subsidiaries out of the centre −1. part (s12, s13)
s24[3]	Subsidiaries out of the centre −2. part (s14, s15, s16, s19)
s25[3]	All subsidiaries out of the center (s23, s24)
s26[2]	Average foreign exchange savings account balance in the first quarter of loan approval preceding year
s27[2,3]	Average foreign exchange savings account balance in the second quarter of loan approval preceding year
s28[2]	Average foreign exchange savings account balance in the third quarter of loan approval preceding year
s29[3]	Average foreign exchange savings account balance in the fourth quarter of loan approval preceding year
s30	Average foreign exchange savings account balance in a year preceding loan approval (s26 + s27 + s28 + s29)/4
s31[1,3]	The relative difference between preceding first quarter and yearly average foreign exchange savings account balance =(s26 − s30)/s30
s32[1]	The relative difference between preceding second quarter and yearly average foreign exchange savings account balance =(s27 − s30)/s30
s33[1,2,3]	The relative difference between preceding third quarter and yearly average foreign exchange savings account balance =(s28 − s30)/s30
s34[1]	The relative difference between preceding fourth quarter and yearly average foreign exchange savings account balance =(s29 − s30)/s30
s35[2]	Average domestic currency savings account balance in the first quarter of loan approval preceding year
s36[2]	Average domestic currency savings account balance in the second quarter of loan approval preceding year
s37[2]	Average domestic currency savings account balance in the third quarter of loan approval preceding year
s38[2]	Average domestic currency savings account balance in the fourth quarter of loan approval preceding year
s39	Average domestic currency savings account balance in a year preceding loan approval (s35 + s36 + s37 + s38)/4
s40[1,3]	The relative difference between preceding first quarter and yearly average domestic currency savings account balance = (s35 − s39)/s39
s41[1]	The relative difference between preceding second quarter and yearly average domestic currency savings account balance =(s36 − s39)/s39

Appendix (*continued*)

Variable	Variable description
s42[1,2,3]	The relative difference between preceding third quarter and yearly average domestic currency savings account balance =(s37 − s39)/s39
s43[1]	The relative difference between preceding fourth quarter and yearly average domestic currency savings account balance =(s38 − s39)/s39
s44[2]	Average foreign exchange and domestic currency savings account balance in a year preceding loan approval =s30 + s39.
s45	Use of bank services over the phone: (0.5 – yes, 0 – no)
s46	Transaction account with the bank on the approval date (0.5 – yes, 0 – no)
s47[3]	Number of months of transaction account with the bank
s48[2]	Transaction account ranking: best rank (=0.5, other =0)
s49	Transaction account ranking: middle rank (=0.5, other =0)
s50[2]	Transaction account ranking: lower rank (=0.5, other =0)
s51	Transaction account ranking: lowest rank (=0.5, other =0)
s52	Transaction account reminders of insufficient funds: (0.5 – yes, 0 – no)
s53[2]	Limited number of checks approved by the bank (0.5 – yes, 0 – no)
s54[1,2,3]	Average regular monthly cash inflows in a year of loan approval
s55[1]	Average extraordinary monthly cash inflows in a year of loan approval
s56[1]	Average monthly cash outflows in a year of loan approval
s57[1]	Time deposits - balance on the loan approval date
s58[2]	Use of credit card in a year of loan approval (0.5 – yes, 0 – no)
s59[2]	Use of automatic bank transfers in a year of loan approval (0.5 – yes, 0 – no)
s60	Maximum amount of approved borrowing on credit card 1
s61[2,3]	Maximum amount of approved borrowing on credit card 2
s62	Maximum amount of approved borrowing on credit card 3
s63[1]	Average monthly free cash flow in a year of loan approval =s54 + s55 − s56
s64	Total number of credit cards
s65[1]	Total maximum amount of approved borrowing on all credit cards
s66[1]	Total cash flows: regular, extraordinary and matured time deposits
s67[1]	Loan approval date

Source: Internal bank data.

References

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance, 23*(4), 589–611.

Altman, E. I. (1993). *Corporate financial distress and bankruptcy*. NY: John Wiley & Sons.

Avery, R. B., Calem, P. S., & Canner, G. B. (2004). Consumer credit scoring: Do situational circumstances matter? *Journal of Banking and Finance, 28*, 835–856.

Baesens, B., Setiono, R., Mues, C., & Vanthienen, J. (2003a). Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science, 49*(3), 312–329.

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003b). Benchmarking state-of-art classification algorithms for credit scoring. *Journal of the Operational Research Society, 54*, 627–635.

Beaver, H. W. (1966). Financial ratios as predictors of failure. *Journal of Accounting Research, 4*, 71–127.

Caouette, J. B., Altman, E. I., & Narayanan, P. (1998). *Managing credit risk*. NY: John Wiley & Sons.

Desai, V. S., Crook, J. N., & Overstreet, G. A. Jr., (1996). A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research, 95*, 24–37.

Glorfeld, L. W., & Hardgrave, B. C. (1996). An improved method for developing neural networks: The case of evaluating commercial loan creditworthiness. *Computers and Operations Research, 23*(10), 933–944.

Godoy, S., & Stiglitz, J.E. (2006). Growth, initial conditions, law and speed of privatization in transition countries: 11 years later. National Bureau of Economic Research, Working paper 11992, pp. 1–29.

Goonatilake, S., & Treleaven, P. (Eds.). (1995). *Intelligent system for finance and business*. Chichester: John Wiley & Sons.

Gujarati, D. N. (1995). *Basic econometrics* (3rd ed.). NY: McGraw-Hill.

Hand, D. J. (Ed.). (1998). *Statistics in finance*. London: Arnold.

Hsieh, N. C. (2005). Hybrid mining approach in the design of credit scoring models. *Expert Systems with Applications, 28*, 655–665.

Jacobson, T., & Roszbach, K. (2003). Bank lending policy, credit scoring and value-at-risk. *Journal of Banking and Finance, 27*, 615–633.

Jensen, H. L. (1996). Using neural networks for credit scoring. In R. R. Trippi & E. Turban (Eds.), *Neural networks in finance and investing* (pp. 453–466). Chicago: IRWIN.

Kim, Y. S., & Sohn, S. Y. (2004). Managing loan customers using misclassification patterns of credit scoring model. *Expert Systems with Applications, 26*, 567–573.

Lee, T. S., & Chen, I. F. (2005). A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. *Expert Systems with Applications, 28*, 743–752.

Lee, T. S., Chiu, C. C., Lu, C. J., & Chen, I. F. (2002). Credit scoring using the hybrid neural discriminant technique. *Expert System with Applications, 23*, 245–254.

Mramor, D., & Valentincic, A. (2003). Forecasting the liquidity of very small private companies. *Journal of Business Venturing, 18*, 745–771.

Piramuthu, S., Shaw, M. J., & Gentry, J. A. (1994). A classification approach using multi-layer neural networks. *Decision Support Systems, 11*, 509–525.

Richeson, L., Zimmermann, R. A., & Barnett, K. G. (1996). Predicting consumer credit performance: Can neural networks outperform traditional statistical methods? In R. R. Trippi & E. Turban (Eds.), *Neural networks in finance and investing* (pp. 45–70). Chicago: IRWIN.

Sinkey, J. F. Jr, (1992). *Commercial bank financial management: In the financial-services industry* (4th ed.). NY: Macmillan Publishing Company.

Sustersic, M. (2001). Application of neural networks in consumers credit risk assessment. Master degree thesis. Ljubljana: Faculty of Economics; 2001.

Thomas, L. C. (1998). Methodologies for classifying applicants for credit. In D. J. Hand (Ed.), *Statistics in finance*. London: Arnold.

Thomas, L. C. (2000). A survey of credit and behavioral scoring: Forecasting financial risk of lending to consumers. *International Journal of Forecasting, 16*, 149–172.

Trippi, R. R., & Turban, E. (Eds.). (1996). *Neural networks in finance and investing*. Chicago: IRWIN.

Turban, E., & Aronson, Y. E. (1998). *Decision support systems and intelligent systems* (5th ed.). London: Prentice-Hall International.

Walker, R. F., Haasdijk, E. W., & Gerrets, M. C. (1995). Credit evaluation using a genetic algorithm. In S. Goonatilake & P. Treleaven (Eds.), *Intelligent system for finance and business* (pp. 39–59). Chichester: John Wiley & Sons.

West, D. (2000). Neural network credit scoring models. *Computers and Operations Research, 27*, 1131–1152.

Zupan, J., & Gasteiger, J. (1993). *Neural networks for chemists*. VCH: Weinheim.

Zupan, J., & Novič, M. (1999). Optimisation of structure representation for QSAR studies. *Analytica Chemica Acta, 388*, 243–250.