pISSN 2073-8005
eISSN 2311-9438

Banking

Translated Article[†]

DATA MINING TECHNIQUES: MODERN APPROACHES TO APPLICATION IN CREDIT SCORING

Elena S. VOLKOVA
Financial University under Government of Russian Federation, Moscow, Russian Federation
EVolkova@fa.ru
Corresponding author

Vladimir B. GISIN
Financial University under Government of Russian Federation, Moscow, Russian Federation
VGisin@fa.ru

Vladimir I. SOLOV'EV
Financial University under Government of Russian Federation, Moscow, Russian Federation
VSoloviev@fa.ru

Article history:
Received 4 July 2017
Received in revised form
9 August 2017
Accepted 24 August 2017
Translated 4 December 2017
Available online 14 December 2017

JEL classification: C38, C55, D81

Keywords: loan scoring, credit score, machine learning, data mining

Abstract

Importance This article examines the current state of research in machine learning and data mining, which computational methods get combined with conventional lending models such as scoring, for instance.

Objectives The article aims to classify the modern methods of credit scoring and describe models for comparing the effectiveness of the various methods of credit scoring.

Methods To perform the tasks, we have studied relevant scientific publications on the article subject presented in Google Scholar.

Results The article presents a classification of modern data mining techniques used in credit scoring.

Conclusions and Relevance Credit scoring models using machine learning procedures and hybrid models using combined methods can provide the required level of efficiency in the modern environment.

The editor-in-charge of this article was Irina M. Vechkanova
Authorized translation by Andrey V. Bazhanov

Introduction

Credit scoring can be defined as a technology that helps a credit organization decide on the granting of credit to the applicant in the light of its characteristics, such as age, income, marital status, etc. Such technologies have emerged and been developed along with the emergence of trade and the need for credit. The ideas and methods of scoring, consistent with their modern understanding, were first formulated in the work by D. Durand [1].

[†]For the source article, please refer to: Волкова Е.С., Гисин В.Б., Соловьев В.И. Современные подходы к применению методов интеллектуального анализа данных в задаче кредитного скоринга. Финансы и кредит. 2017. Т. 23. Вып. 34. С. 2044–2060. URL: https://doi.org/10.24891/fc.23.34.2044

Please cite this article as: Volkova E.S., Gisin V.B., Solov'ev V.I. Data Mining Techniques: Modern Approaches to Application in Credit Scoring. *Digest Finance*, 2017, vol. 22, iss. 4, pp. 400–412. https://doi.org/10.24891/df.22.4.400

Following the adoption of the Basel II and especially, Basel III Accords, it has become possible and necessary to apply internal ranking procedures to assess the basic risk parameters. This makes the role of credit scoring more significant and urges financial institutions to continually improve the quantitative models they use.

The article by D.J. Hand and W.E. Henley gives a fairly complete idea of the works on the classical methods of credit scoring [2]. The articles by V. García, A.I. Marqués, J.S. Sánchez [3] and S. Lessmann, H.-V. Seow, B. Baesens, and L.C. Thomas [4] provide reviews of later publications on the subject. Numerous reviews are devoted to selected technologies of credit scoring and comparative analysis of methods used.

The present review article focuses on the works which apply and describe the methods of credit scoring based on the data mining methodology.

In recent years, there has been a significant increase in the number of publications describing so-called hybrid methods.

Section 1 of the article provides a brief overview of the basic techniques of credit scoring.

Section 2 provides a brief description of common test data sets to compare the effectiveness of credit scoring techniques.

Sections 3 and 4 describe how different models and techniques of credit scoring can be compared.

Section 5 provides an analysis of the software implementation of machine learning algorithms.

1. Basic Machine-Learning Techniques in Credit Scoring

1.1. Linear Regression

Linear regression links the borrower's characteristics represented by the $x \in R^n$ vector to the target variable $y \in \{-1; 1\}$:

$$y = \beta_0 + \langle \beta, x \rangle + \varepsilon,$$

where ε is a random error with zero mean. When deciding whether to assign y to a class, the $\beta_0 + \langle \beta, x \rangle$ value is interpreted as a conditional mathematical expectation $E(y|x)$. The work by D.J. Hand and M.G. Kelly [5] presents scorecards built by means of linear regression. Note that in recent years, linear regression has not been used alone, although it is still used as an important tool in mixed models.

1.2. Logistic Regression

Logistic regression is one of the main tools of credit scoring. In publications, logistic regression is typically used to compare with other techniques, for instance, the works by B.W. Yap (et alias) [6], N.G. Pavlidis (et alias) [7], Z. Khemais (et alias) [8], or in combination with other techniques, for instance, the works by F. Louzada (et alias) [9], and Z. Li (et alias) [10].

The logistic regression technique is used in credit scoring to calculate the $P(y=1|x)$ probability of rejection of loan issue to a borrower with x characteristics. This probability can be presented as

$$P(y=1|x) = \frac{1}{1+e^{-(\alpha+\beta^T x)}}.$$

The maximum-likelihood technique is used to estimate the α and β_i coefficients (the coordinates of β vector). For estimation, a learning set is used.

1.3. Discriminant Analysis

Discriminant analysis is one of the most frequently used score techniques, in credit scoring in particular. Discriminant analysis goes back to the work by R.A. Fisher [11]. This was one of the first techniques used to build credit scoring systems. R.A. Eisenbeis' article [12] analyzes problems related to the application of discriminant analysis in credit scoring. Currently, the discriminant analysis continues to be used in credit scoring directly [13]. Discriminant analysis often serves as a benchmark against which other techniques are compared, as is done, for instance, in the article by S. Akkoç [14]. A number of studies are related to improving the accuracy of the discriminant analysis through applying the new procedures [15].

1.4. Decision Trees

This technique originates from the work by L. Breiman (et alias) [16]. W.-Y. Loh's work [17] gives an insight into the state of the art. In the case of credit scoring, decision trees are mainly used for classification [4].

We next briefly describe the techniques involved in building decision trees. Variable *X* is stated to be an order one if the numerical values it adopts are ordered as significant for the classification. Otherwise, the variable is called categorical.

The algorithm for *Automatic Interaction Detector Analysis* (AID), one of the first algorithms for building classification trees, sequentially breaks data in each node. In the case of an order variable, branching occurs according to the $X \le c$, type conditions, in the case of a categorical variable, it occurs according to the $X \in A$. Assume that $S(t)$ is a set of data numbers in the learning sample that are related to *t* node. Let us denote the average (for $S(t)$) value of the interpretable variable *Y* by $\overline{y_t}$. The $imp(t) = \sum_{i \in S(t)} (y_i - y_t)^2$ value is a measure of the contamination of *t* node. The AID algorithm chooses such a splitting, which minimizes the amount of contamination indices by direct successor node. The split process ends when the contamination level becomes less than the predetermined threshold.

The *THeta Automatic Interaction Detection* (THAID) algorithms extend the described technique to categorical variables. Here, entropy or the Gini coefficient are used as a contamination measure.

The newer algorithms, like *Classification and Regression Trees* (CART) algorithms replace the stopping rules used in the AID and THAID algorithms by the rules for creating and pruning new branches. *Chi-square Automatic Interaction Detection* (CHAID) and *C4.5* algorithms are used as well.

The article by S.M. Finlay [18] shows the comparative characteristics of the various algorithms of credit scoring, including the CART algorithms. It notes that the CART algorithms are less efficient than the other ones. However, some new ideas and improvements to the modeling of trees make it possible to significantly increase the algorithms' efficiency (See D. Zhang (et alias) [19] and Q. Hu (et alias) [20]).

Knowledge extraction algorithms, e.g. *Rule*, *Extraction*, and *RX* big data-centric algorithms can be attributed to algorithms associated with decision trees (See Y. Hayashi (et alias) [21]).

1.5. Support Vector Machine

Support Vector Machine (SVM) as a model for statistical classification was proposed by Vladimir N. Vapnik [22]. The principle of the technique is as follows.

Assume a given learning set $\{(x^{(j)}, y^{(j)})\}_{j=1,2,\ldots,l}$, where $x^{(j)} \in X \subset R^n$ is the attribute description of the object, and $y^{(j)} \in \{-1; 1\}$ is the binary classifier. The equation of $\langle w, x \rangle - w_0 = 0, w \in R^n$ type specifies a hyperplane with *w* normal vector that separates the classes of "good" $y^{(j)} = 1$ and "bad" $y^{(j)} = -1$ objects in R^n space.

The best separating hyperplane is defined as an optimization solution:

$$\|w\| \to \min;$$
$$y^{(j)}(\langle w, x^{(j)} \rangle - w_0) \ge 1, j = 1,2,\ldots,l.$$

If there is a separating hyperplane, the $\frac{2}{\|w\|}$ value is the width of the band between points of different classes. The problem of finding the best separating hyperplane can be solved by using the Karush–Kuhn–Tucker (KKT) Conditions. Assume that

$$L(w, w_0, \lambda) = \frac{1}{2}\langle w, w \rangle -$$

$$-\sum_{j=1}^{l} \lambda_j (y^{(j)}(\langle w, x^{(j)} \rangle - w_0) - 1)$$

is the corresponding Lagrange function.

Learning sample object $x^{(j)}$ is called a support vector if $\lambda_j > 0$ and $\langle w, x^{(j)} \rangle - w_0 = y^{(j)}$. Vector *w* is a linear combination of support vectors:

$$w = \sum_j \lambda_j y^{(j)} x^{(j)}.$$

Therefore, a relatively small number of learning sample objects are used to actually build the *w* vector. This sparseness property distinguishes the SVM technique from the classical linear separators of the Fischer's linear discriminant type.

If there is no separating hyperplane (the learning sample cannot be linearly separated), the optimization problem gets adjusted: the amount

Please cite this article as: Volkova E.S., Gisin V.B., Solov'ev V.I. Data Mining Techniques: Modern Approaches to Application in Credit Scoring. *Digest Finance*, 2017, vol. 22, iss. 4, pp. 400–412. https://doi.org/10.24891/df.22.4.400

of error penalties gets added to the objective function.

Switching to a non-linear separator using a kernel is possible as well. The kernel is the $K(x,x')$, $x,x' \in X$ function such that $K(x,x')=\langle \varphi(x), \varphi(x') \rangle$ is for some $\varphi: X \to R^m$ mapping. If the φ mapping is used, linear separator can be built in the R^m space [23].

The quadratic optimization problem through the SVM technique can be formulated in a dual form: we find

$$\max_{\lambda} \left(\sum_j \lambda_j + \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y^{(i)} y^{(j)} K(x^{(i)}, x^{(j)}) \right)$$

under the preceding conditions $0 \le \lambda_j \le C_j$ for all *j* and $\sum_j \lambda_j y_j = 0$.

The C_j parameters control the relative value of the indicators. The most common kernel functions are as follows:

$K(x^{(i)}, x^{(j)}) = \langle x^{(i)}, x^{(j)} \rangle$ – a linear model;

$K(x^{(i)}, x^{(j)}) = (\langle x^{(i)}, x^{(j)} \rangle + 1)^d$ – a polynomial degree d model;

$K(x^{(i)}, x^{(j)}) = \exp\left(-\frac{\|x^{(i)} - x^{(j)}\|}{2\sigma^2} \right)$ – the Gaussian radial base function (RBF) with the σ parameter.

For a new object, the prediction is based on a formula $y = sgn\left(\sum_j \lambda_j y^{(j)} K(x^{(i)}, x) + b_j \right)$, where $b_j = \sum_j \lambda_j y^{(j)} K(x^{(i)}, x^{(j)})$.

The work by W. Chen (et alias) [24] is one of the first to use the SVM technique to solve the problem of credit scoring. For credit scoring, Y. Ling (et alias) [25] used the SVM system in respect to the class of kernels.

1.6. Bayesian Network

The work by N. Friedman (et alias) [26] was probably the starting point for the use of *Bayesian networks* (BN) in credit scoring. This work extends the so-called simple (naïve) Bayesian technique, according to which the decision with the highest *a posteriori* information is chosen. The naïve Bayesian technique is used when a particular feature is independent.

As the authors point out, in credit scoring, this assumption is unrealistic: for instance, the correlation of parameters such as age, education, and income cannot be ignored. The authors have developed the ideas of J. Pearl [27].

In general terms, the Bayesian network is a *directed acyclic graph* (DAG). The training generates a conditional distribution of the probability of $P(Y | X_1, \ldots, X_k)$, where *Y* is the vertex, and $X_1, \ldots, X_k$ is the *parents* on graph.

The Bayesian network defines the joint vertex distribution. For instance, the naïve Bayesian technique is obtained by taking a categorical variable as the root vertex and all attributes as its *children*.

Informal learning of the Bayesian network consists of maximizing its adaptation to the learning set. Optimization is performed relative to scoring function. Bayesian scoring function and the function based on the *minimum description length principle* (MDL) are the most used ones. These functions asymptotically lead to the same learning outcome, but the MDL function has proved to be better in the finite sets.

Assume that $B = (G, \Theta)$ is a Bayesian network (*G* is a graph, Θ is the appropriate probability distribution) and $D = \{u_1, \ldots, u_n\}$ is a learning set (each u_i assigns values to all the vertices of the graph). In this case $MDL(B|D) = \frac{\log N}{2} |B| - LL(B|D)$, where $|B|$ is the number of network settings and $LL(B|D) = \sum_{i=1}^{N} \log(P_B(u_i))$ measures the amount of information required to describe *D* based on the P_B probability distribution.

The MDL scoring function is asymptotically correct.

We shall indicate several works in which the Bayesian networks were used for credit scoring: P. Giudici [28], J. Gemela [29], A.C. Antonakis (et alias) [30, 31], W.-W. Wu [32], and H. Zhu (et alias) [33].

1.7. (Artificial) Neural Network

(Artificial) neural networks (ANNs) convert a set of input variables into a set of output variables and model both linear and non-linear transformations. Transformations are carried out using neurons, which are a simplified model of animal/human brain neurons. Neurons are connected to the network by one-way channels of communication. Each neuron can be activated by incoming input signals and output signals will be issued in the active state. The neural network has a layer of input neurons, which are the neurons that receive input variables, and a layer of output neurons, the output signals of which form output variables and the hidden layers. Neural networks differ in structure, number of hidden layers, and activation function.

The work by D. West [34] analyzes the five models of neural network used in the credit scoring:

- Multilayer Perceptron (MLP);
- Mixture of Experts (MOE);
- Network of Radial Basis Function (RBF);
- Learning Vector Quantization (LVQ);
- Fuzzy Adaptive Resonance Theory (Fuzzy ART).

The efficiency of the neural networks of the listed types in credit scoring has been compared with the efficient use of classical parametric techniques (linear discriminant analysis and logistic regression), non-parametric methods (*k*-nearest neighbors algorithm (*k*-NN) and kernel density estimation (KDE), and the classification tree method (CTM).

The results obtained confirm that multilayer perceptrons show far less than the highest accuracy, the networks of mixture of experts and the network of radial basis function show satisfactory results in credit scoring. Logistic regression is the most accurate technique of the classical ones. Networks based on the fuzzy adaptive resonance theory fall within the least accurate. Being as efficient as the other nets to identify a bad borrower, the Fuzzy ART-based networks are essentially less efficient at recognizing a good borrower.

1.8. Genetic Algorithm

The specific application of *genetic algorithms* (GA) in credit scoring is that the population is formed by classification trees. Mutation and crossover algorithms are applied to trees. In other respects, the algorithm structure is standard. Once the initial population is created, the mutation and crossover processes get iterated and then evaluated. The relative number of classification errors is taken as an estimate. The work by C.-S. Ong (et alias) [35] shows that the results of genetic algorithms (with 1,000 generations) are among the best in the test suites.

1.9. Ensemble Methods

Hybrid and ensemble methods are those which use different techniques of credit scoring to improve efficiency. *Bootstrap Aggregating*, also called *Bagging*, *Boosting*, and *Stacking* are the most common three machine learning ensemble meta-algorithms.

Bagging (stands for **B**ootstrap **Agg**regating) was proposed and introduced by Leo Breiman [36]. The basic idea of this technique is to build a series of predictors, which, in aggregate (after a certain aggregation), produce a better predictor with improved predictive force.

Schematically, in the case of credit scoring, the bagging approach can be outlined like the following. Assume that there is a training algorithm that, by the learning set *L*, builds the predictor φ(*x*, *L*), which gives *y* upon given *x*. Based on the learning set *L*, one can build a set of learning sets $\{L_k\}_{k=1,\dots,K}$ (usually, the same amount as *L*). These sets consist of the same objects selected randomly from *L* (possibly, with repetitions). Let K_+ is equal to the number of those *k* for which φ (x,L_k) gives an affirmative answer. The aggregated predictor produces an affirmative answer if $K_+ > \frac{1}{2}$.

Bagging is particularly effective where the basic training algorithm is instable, viz, strongly dependent on small changes in the learning set.

The basic idea of **Boosting** is to build a strong classification algorithm based on the weak (in terms of accuracy) algorithm. In the process of forming

Please cite this article as: Volkova E.S., Gisin V.B., Solov'ev V.I. Data Mining Techniques: Modern Approaches to Application in Credit Scoring. *Digest Finance*, 2017, vol. 22, iss. 4, pp. 400–412. https://doi.org/10.24891/df.22.4.400

a strong algorithm, the weak algorithm "improves itself" through a redistribution of sample weights from the training sample: in the case of correct recognition, the weight decreases. If recognition is wrong, the weight increases. The boosting approach can be illustrated with the following example.

Assume that X, the space $\{(x^{(j)}, y^{(j)})\}_{j=1,2,\ldots,l}$ – is a training sample. The basic algorithm runs in a series of rounds $t = 1,\ldots,T$. Let us denote the weight assigned to the object in round t by $D_t(j)$ (initial distribution of the weights $D_0(j)$ can be uniform). The learning task is to find in round t such a $h_t(x)$ mapping with values in {–1;1} that minimizes the error probability $\varepsilon_t = \sum_{h_t(x^{(j)}) \neq y^{(j)}} D_t(j)$.

The weights are updated as follows: assume that $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$.

Then $D_{t+1}(j) = \frac{D_t(j)\exp\left(-\alpha_t y^j h_t(x^{(j)})\right)}{Z_t}$, where Z_t is the normalizing factor (so that $D_{t+1}(j)$ is a distribution). The final recognition algorithm is

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

This technique of boosting, based on the exponential loss function, is called *AdaBoost*. It provides an advantage for the algorithm to get improved when the noise use cases are abundant. To minimize this effect, the logistic loss function can be used (this algorithm is called *LogitBoost*).

Stacking (sometimes called *Stacked Generalization*) is a technique which combines several learning algorithms by means of a combiner. A single-layer logistic regression model is often used as the combiner. The theoretical principles of stacking were provided in the work by D.H. Wolpert [37].

The ensemble methods are extensively applied in the credit scoring. The works by S. Akkoç [14], S. Vukovic (et alias) [38], A.I. Marqués (et alias) [39] are illustrative of that.

1.10. Fuzzy Logic Techniques

There are quite a few publications on the application of fuzzy logic techniques in credit scoring. We shall mention some of them: F. Hoffmann (et alias) [40], J. Ignatius (et alias) [41], A. Lahsasna (et alias) [42], A. Kaur (et alias) [43], R. Malhotra, D. Malhotra [44].

However, given the enormous number of publications on credit scoring, the quantity of the above-mentioned publications is relatively small. The works using fuzzy logic for credit scoring can be roughly divided into two groups.

The first group includes the studies applying fuzzy logic within traditional techniques. Typically, these works are related to neural networks and SVM.

The second group includes the studies applying the technique derived from fuzzy set theory. Primarily, these are the works based on fuzzy logic systems, in particular the Mamdani and Takagi-Sugeno-Kang (TSK) type models.

The second part of the review deals with a detailed analysis of the fuzzy logic application in credit scoring.

2. Test Data Sets

The *German Credit* and *Australian Credit Approval* Data Sets are commonly used by developers of credit scoring algorithms.

The Australian Credit Approval Data Set contains a combined total of 690 borrowers (instances), 307 of which are solvent (paying towards a loan) and 383 are insolvent. The description of each particular borrower includes 14 attributes: six continuous and eight categorical ones.

The German Credit Data Set contains 1,000 records of borrowers (instances), 700 of which are solvent and 300 are insolvent. The description of each particular borrower contains 20 attributes.

Both the data sets are publicly available at UCI Repository of Machine Learning[1].

3. General Credit Scoring Model Concept and the Comparison of Credit Scoring Models

Let us agree to call the outcome *good* if $y = 0$ and *bad*, if $y = 1$. In the classical setting, the prediction

[1] Statlog (Australian Credit Approval) Data Set. URL: http://archive.ics.uci.edu/ml/datasets/Statlog+(Australian+Credit+Approval); Statlog (German Credit Data) Data Set. URL: http://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

task is to define $E[Y|x]=E[y=1|x]$. by the set of objects under observation (*x*, *y*). If the $P(Bad|x)$ conditional probabilities were known, it would not be difficult to make optimal decisions on credit granting. Attribute space is usually too large to allow for empirical evaluation of the probabilities $P(Bad|x)$. The standard approach is to build a scoring function *s*(*x*). The $P(Bad|s)=P(Bad|s(x)=s)$ posterior probability is used to build predictions, replacing $P(Bad|x)$.

Assume that *A* is a credit scoring model. The scoring function $s_A(x)$ value can be considered as realization of a random value s_A. Let us denote the probability density of the s_A conditional prediction upon given *y* by $f(s_A|y)$, and denote the probability that the scoring function value will be equal to s_A by $v(s_A)$.

The basic principles of comparing credit scoring models are found in the works by R.T. Clemen (et alias) [45], M. DeGroot (et alias) [46, 47], and H. Zhu (et alias) [33].

Assume that *A* and *B* are scoring models. It is stated that the *A* model is sufficient for the *B* model if there is an *h* function with the following properties:

1) $h(s_B|s_A)\geq 0$ for any s_A, s_B;

2) $\sum_{S_B} h(s_B, s_A)=1$ at any s_A;

3) $\sum_{s_A} h(s_B|s_A) f_A(s_A|y)=f_B(s_B|y)$ for any s_B and *y*.

If *A* is sufficient for *B*, then *B* can be considered more undefined, since the *h* function gives additional randomness to the s_B values.

It is stated that the *B* model is not related to the *A* model, if *y* is independent of the s_B upon given s_A, that is $P(y|s_A, s_B)=P(y|s_A)$.

For the assumed *A* and *B* scoring models, we define a combined scoring model *C*, assuming that $s_C=P(Good|s_A, s_B)$. The combined model is sufficient for the *A* and *B* models. The model is sufficient for the *C* model when and only when the *B* model and *A* model are unrelated.

Now, let us briefly run through the scoring model in terms of utility value.

Suppose that credit granting to a good borrower yields an income of 1, and in the case of a bad borrower, it amounts to $-\alpha\leq 0$ (loss). Assume that $\pi(s)=P(Good|s)$. When a loan is granted to the borrower with *s* scoring, the expected income *R* is:

$$E[R|s]=\pi(s)-\alpha(1-\pi(s)).$$

We also assume that the credit denial yields an income of 0 regardless of the type of borrower. The decision to grant a loan is made if $E[R|s]\geq 0$. Since the $\pi(s)$ function increases monotonically, there is a value of s^*, that $E[R|s^*]=0$. If the scoring function value is greater than s^*, the loan granting is approved, if not, the borrower's credit application is rejected. Thus, $E[R]=\sum_{s\geq s^*} E[R|s]v(s)$.

Since s^* depends on α, the mathematical expectation value of *E*[*R*] income also depends on α. This value can be used to compare credit scoring models: scoring model *A* is sufficient for scoring model *B* when and only when $E_A[R]\geq E_B[R]$ is for all α.

4. Evaluating the Quality of Credit Scoring Algorithms

One of the ways to determine the quality of a machine learning model is to split the sample into a *training dataset*, which is used to identify algorithm parameters, and a *validation dataset* for each object of which the algorithm-predicted and true classes are compared.

The most common techniques of credit scoring algorithm evaluation are based on the confusion matrix: all sample objects are divided into four categories, depending on the combination of the true *y* response and the α(*x*) response supplied by the algorithm:

	$\alpha(x)=1$	$\alpha(x)=0$
$y=1$	*TP*	*FN*
$y=0$	*FP*	*TN*

(*TP* is the abbreviation for *True Positive*, *FN* is the abbreviation for *False Negative*, and similarly in the two remaining cases).

Please cite this article as: Volkova E.S., Gisin V.B., Solov'ev V.I. Data Mining Techniques: Modern Approaches to Application in Credit Scoring. *Digest Finance*, 2017, vol. 22, iss. 4, pp. 400–412. https://doi.org/10.24891/df.22.4.400

Since the purpose of applying classification algorithms in credit scoring is to sort the scoring objects into *good* and *bad*, the algorithms efficiency is evaluated through matching of the algorithm-predicted class and true class of the object for each particular one from the validation dataset.

The credit scoring task has two features.

First, the classification of bad credit as a good one is more costly than the classification of good credit as a bad one.

Second, there are always more good customers than bad ones in a learning sample.

Due to the first feature, the following algorithm quality measures are used in the credit scoring task:

$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$ is the proportion of loans classified correctly;

$Precision = \frac{TP}{TP + FP}$ is the proportion of bad loans classified correctly among all observations and classified by the algorithm as bad loan;

$Recall = \frac{TP}{TP + FN}$ is the completeness, i.e. assessment of the ability of the algorithm to recognize bad loans;

$Negative\ Predictive\ Value = \frac{TN}{TN + FN}$ is the proportion of good loans classified correctly among all observations and classified by the algorithm as good loan;

$Specificity = \frac{TN}{TN + FP}$ is the assessment of the ability of the algorithm to recognize good loans;

$F1\,Score = \frac{2(Precision \cdot Recall)}{Precision + Recall}$ is the harmonic mean of precision and recall;

$False\ Negative\ Rate = \frac{FN}{TP + FN}$ is the proportion of bad loans incorrectly classified as good loan;

$False\ Positive\ Rate = \frac{FP}{TN + FP}$ is the proportion of good loans incorrectly classified as bad loan.

A *Receiver Operating Characteristic Curve* (ROC curve) is a graphical plot that illustrates the change in the ratio of *Recall* correctly classified bad loans in their total number to the *False Positive Rate* of good loans incorrectly classified as bad loan, as the decision rule threshold is varied.

The ROC curve is obtained as follows:

Assume that the result of the α(*x*) algorithm depends on some parameter, for example, the threshold value, and the algorithm is as follows $\alpha(x) = Entier[s(x) > s^*]$.

If $s^* = \infty$ we get *SEN* = 0 and *FPR* = 0, if $s^* = -\infty$ – *SEN* = 1 and *FPR* = 1. When s^* changes from $-\infty$ to ∞, the point (*FPR*, *SEN*) describes a curve called the ROC curve. The *Area Under the ROC Curve* (AUC) serves as a quality characteristic of the algorithm.

The equality of *AUC* = 0.5 means that the algorithm categorizes objects at random. The more *AUC* value, the better the algorithm. Another most commonly used measure is called the *Gini coefficient* (sometimes expressed as a *Gini ratio* or a normalized *Gini index*) that can be thought of as the ratio of the area that lies between the diagonal line and the curve: $Gini = 2\,AUC - 1$.

An important problem in building scoring models is the fact that the proportion of bad loans in the sample is significantly lower than the proportion of good loans (typically, between 2 and 30 percent). In this situation, a model that offers to recognize all customers as the good ones can provide a minor error in the training and test samples.

Possible solutions to this problem are the introduction of different cost of Type I and Type II errors or the modification of the training sample to change the representativeness of the sample.

There are two main techniques used to change the representativeness of sample: *Oversampling* and *Undersampling*.

The limitation of the first technique is that a simple use case duplication may not affect some training methods in any way but may lead to the overfitting of others. If removing the instances that belong to a majority class, some information important for

the classification may be lost, which is also not desirable.

Synthetic Minority Oversampling Technique (SMOTE) is the most frequently used statistical technique for increasing the number of cases in the dataset in a balanced way:

- The difference $d = x_b - x_a$ between the x_a, x_b vectors of features of nearest neighbors a, b of minority class is calculated;
- A vector of features for the new instance is generated $x_{\tilde{d}} = x_a + cd$, where $c \sim N(0,1)$.

There are different variations of the SMOTE technique, where the nearest neighbors from both the minority and majority classes are used to generate minority class cases, and the generated instances are closer to or away from the margin of class separation.

In practice, however, the SMOTE technique very often results in the overfitting of models. It is also a computationally expensive and time-consuming approach. Moreover, the problem of unbalanced classes in scoring samples is usually a separate challenge.

5. Software Implementation of Machine Learning Algorithms in Credit Scoring

The software that is used to automate the data mining and machine learning task handling can be divided into three classes:

– Commercial statistical package;

– Open-source framework;

– Cloud solution.

Generally, commercial software has been used in banks to address data analysis, particularly related to scoring. SAS software suite has been most frequently used, IBM SPSS Statistics and Statistica software packages have been used less frequently.

These three software product lines provide similar features. These capabilities include analytical data preparation tools, ready and custom machine learning algorithm templates, including linear and logistic regression models, decision trees and forests, gradient boosting, support vectors, neural networks, etc. In addition, these packages can set up model parameters and use interactive quality assessment techniques.

In recent years, the banks, while not completely refusing to use the commercial SAS-type packages, have also used the free and open-source Python/R/Spark software framework.

The advantage of these software frameworks is that one can use many more algorithms than commercial software packages offer.

But if, for example public health and industry institutions and organizations have largely stopped using commercial software packages in favor of open-source libraries, the banks still use SAS-type packages more frequently than Python and R software.

The *open-source R programming language* was created as a special tool for statistical computing. It became the first open-source software environment to be used extensively for data analysis.

The most commonly used libraries for machine learning in R are as follows:

– **rpart** and **CARET** (classification and regression algorithms);

– **randomForest** (random forest algorithm);

– **nnet** (neural networks);

– **e1071** (one of the first machine learning libraries in R that contains the implementation of the support vector machine, naïve Bayes classifier, and a number of other techniques);

– **kernlab** (support vector machine);

– **gbm** (gradient boosting);

– **ROCR** (visualization of the performance of scoring classifiers).

The *Python programming language* has become the most popular tool for analyzing data after a perfectly documented **scikit-learn** library was released that implements a large number of machine-based learning algorithms. In addition to the scikit-learn library, the **TensorFlow** and **Theano** open-source software libraries are also popular (these libraries also implement different data analysis techniques but outnumber scikit-learn only in techniques implemented with neural networks).

Please cite this article as: Volkova E.S., Gisin V.B., Solov'ev V.I. Data Mining Techniques: Modern Approaches to Application in Credit Scoring. *Digest Finance*, 2017, vol. 22, iss. 4, pp. 400–412. https://doi.org/10.24891/df.22.4.400

The **Pyinference** library is used for Bayesian and fuzzy reasoning in Python. The main advantage of Python over R is the faster script execution speed.

Apache Spark, an open-source cluster-computing scalable framework oriented to compute in RAM is an alternative solution for analyzing data when Python performance is not enough. The **MLib** Apache Spark's scalable machine learning library is still substantially inferior to the scikit-learn library by the number of algorithms, but is actively developing.

Cloud-based machine learning platforms have emerged in recent years.

The main advantage of these systems is flexible scalability, viz the allocation and release of computing resources occurs instantaneously according to the tasks to be performed.

The *Amazon Machine Learning* service implements only the basic algorithms for binary and multi-class classifications, as well as regression.

The *Google Cloud Machine Learning Engine* service provides the ability to run TensorFlow models in a cloud environment.

The *Microsoft Azure Machine Learning Studio* environment provides a powerful tool to build machine learning models through a simple graphical interface using a variety of standard classification, regression, cluster analysis, and anomaly detection algorithms, and embed native code in these models in SQL, Python, and R.

A similar solution, the *Watson Machine Learning* service is expected from IBM in the near future.

However, despite the benefits of the cloud-based data mining tools, they are not virtually used in banks because of security concerns about the transfer of confidential customer data to cloud storage.

References

1. Durand D. Risk Elements in Consumer Installment Financing. New York, National Bureau of Economic Research Books, 1941, 163 p.
2. Hand D.J., Henley W.E. Statistical Classification Methods in Consumer Credit Scoring: A Review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 1997, vol. 160, iss. 3, pp. 523–541. URL: https://doi.org/10.1111/j.1467-985X.1997.00078.x
3. García V., Marqués A.I., Sánchez J.S. An Insight into the Experimental Design for Credit Risk and Corporate Bankruptcy Prediction Systems. *Journal of Intelligent Information Systems*, 2015, vol. 44, iss. 1, pp. 159–189. URL: https://doi.org/10.1007/s10844-014-0333-4
4. Lessmann S., Seow H.-V., Baesens B., Thomas L.C. Benchmarking State-of-the-Art Classification Algorithms for Credit Scoring: An Update of Research. *European Journal of Operational Research*, 2015, vol. 247, iss. 1, pp. 124–136. URL: https://doi.org/10.1016/j.ejor.2015.05.030
5. Hand D.J., Kelly M.G. Superscorecards. *IMA Journal of Management Mathematics*, 2002, vol. 13, iss. 4, pp. 273–281.
6. Yap B.W., Ong S.H., Husain N.H.M. Using Data Mining to Improve Assessment of Credit Worthiness via Credit Scoring Models. *Expert Systems with Applications*, 2011, vol. 38, iss. 10, pp. 13274–13283. URL: https://doi.org/10.1016/j.eswa.2011.04.147
7. Pavlidis N.G., Tasoulis D.K., Adams N.M., Hand D.J. Adaptive Consumer Credit Classification. *Journal of the Operational Research Society*, 2012, vol. 63, iss. 12, pp. 1645–1654. URL: https://doi.org/10.1057/jors.2012.15
8. Khemais Z., Nesrine D., Mohamed M. Credit Scoring and Default Risk Prediction: A Comparative Study between Discriminant Analysis & Logistic Regression. *International Journal of Economics and Finance*, 2016, vol. 8, iss. 4, pp. 39–53. URL: http://dx.doi.org/10.5539/ijef.v8n4p39

9. Louzada F., Anacleto-Junior O., Candolo C., Mazucheli J. Poly-bagging Predictors for Classification Modelling for Credit Scoring. *Expert Systems with Applications*, 2011, vol. 38, iss. 10, pp. 12717–12720. URL: https://doi.org/10.1016/j.eswa.2011.04.059

10. Li Z., Tianb Y., Li K. et al. Reject Inference in Credit Scoring Using Semi-supervised Support Vector Machines. *Expert Systems with Applications*, 2017, vol. 74, pp. 105–114. URL: https://doi.org/10.1016/j.eswa.2017.01.011

11. Fisher R.A. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 1936, vol. 7, iss. 2, pp. 179–188. URL: https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

12. Eisenbeis R.A. Problems in Applying Discriminant Analysis in Credit Scoring Models. *Journal of Banking & Finance*, 1978, vol. 2, iss. 3, pp. 205–219. URL: https://doi.org/10.1016/0378-4266(78)90012-2

13. Mylonakis J., Diacogiannis G. Evaluating the Likelihood of Using Linear Discriminant Analysis as a Commercial Bank Card Owners Credit Scoring Model. *International Business Research*, 2010, vol. 3, no. 2, pp. 9–20. URL: https://doi.org/10.5539/ibr.v3n2p9

14. Akkoç S. An Empirical Comparison of Conventional Techniques, Neural Networks and the Three Stage Hybrid Adaptive Neuro Fuzzy Inference System (ANFIS) Model for Credit Scoring Analysis: The Case of Turkish Credit Card Data. *European Journal of Operational Research*, 2012, vol. 222, iss. 1, pp. 168–178. URL: https://doi.org/10.1016/j.ejor.2012.04.009

15. Falangis K., Glen J.J. Heuristics for Feature Selection in Mathematical Programming Discriminant Analysis Models. *Journal of the Operational Research Society*, 2010, vol. 61, no. 5, pp. 804–812. URL: https://doi.org/10.1057/jors.2009.24

16. Breiman L., Friedman J., Stone C.J., Olshen R.A. Classification and Regression Trees. Monterey, CA, Wadsworth & Brooks/Cole Advanced Books & Software, 1984, 368 p.

17. Loh W.-Y. Fifty Years of Classification and Regression Trees. *International Statistical Review*, 2014, vol. 82, iss. 3, pp. 329–348. URL: https://doi.org/10.1111/insr.12016

18. Finlay S. Multiple Classifier Architectures and Their Application to Credit Risk Assessment. *European Journal of Operational Research*, 2011, vol. 210, iss. 2, pp. 368–378. URL: http://dx.doi.org/10.1016/j.ejor.2010.09.029

19. Zhang D., Zhou X., Leung S.C.H., Zheng J. Vertical Bagging Decision Trees Model for Credit Scoring. *Expert Systems with Applications*, 2010, vol. 37, iss. 12, pp. 7838–7843. URL: https://doi.org/10.1016/j.eswa.2010.04.054

20. Hu Q., Che X., Zhang L. et al. Rank Entropy-Based Decision Trees for Monotonic Classification. *IEEE Transactions on Knowledge and Data Engineering*, 2012, vol. 24, iss. 11, pp. 2052–2064. URL: https://doi.org/10.1109/TKDE.2011.149

21. Hayashi Y., Tanaka Y., Takagi T. et al. Recursive-Rule Extraction Algorithm with J48graft and Applications to Generating Credit Scores. *Journal of Artificial Intelligence and Soft Computing Research*, 2016, vol. 6, iss. 1, pp. 35–44. URL: https://doi.org/10.1515/jaiscr-2016-0004

22. Vapnik V.N. Statistical Learning Theory. New York, John Wiley, 1998, 768 p.

23. Bellotti T., Crook J. Support Vector Machines for Credit Scoring and Discovery of Significant Features. *Expert Systems with Applications*, 2009, vol. 36, iss. 2-2, pp. 3302–3308. URL: https://doi.org/10.1016/j.eswa.2008.01.005

Please cite this article as: Volkova E.S., Gisin V.B., Solov'ev V.I. Data Mining Techniques: Modern Approaches to Application in Credit Scoring. *Digest Finance*, 2017, vol. 22, iss. 4, pp. 400–412. https://doi.org/10.24891/df.22.4.400

24. Chen W., Ma C., Ma L. Mining the Customer Credit Using Hybrid Support Vector Machine Technique. *Expert Systems with Applications*, 2009, vol. 36, iss. 4, pp. 7611–7616. URL: https://doi.org/10.1016/j.eswa.2008.09.054

25. Ling Y., Cao Q., Zhang H. Credit Scoring Using Multi-Kernel Support Vector Machine and Chaos Particle Swarm Optimization. *International Journal of Computational Intelligence and Applications*, 2012, vol. 11, iss. 3, pp. 12500198:1–12500198:13.

26. Friedman N., Geiger D., Goldszmidt M. Bayesian Network Classifiers. *Machine Learning*, 1997, vol. 29, iss. 2-3, pp. 131–163. URL: https://doi.org/10.1023/A:1007465528199

27. Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, 1988, 552 p.

28. Giudici P. Bayesian Data Mining, with Application to Benchmarking and Credit Scoring. *Applied Stochastic Models in Business and Industry*, 2001, vol. 17, iss. 1, pp. 69–81. URL: https://doi.org/10.1002/asmb.425

29. Gemela J. Financial Analysis Using Bayesian Networks. *Applied Stochastic Models in Business and Industry*, 2001, vol. 17, iss. 1, pp. 57–67. URL: https://doi.org/10.1002/asmb.422

30. Antonakis A.C., Sfakianakis M.E. Naïve Bayes as a Means of Constructing Application Scorecards. In: L. Moutinho and K.-H. Huarng (eds), Advances in Doctoral Research in Management. Singapore, World Scientific Publishing Co. Pte. Ltd, 2008, vol. 2, pp. 47–62.

31. Antonakis A.C., Sfakianakis M.E. Assessing Naïve Bayes as a Method for Screening Credit Applicants. *Journal of Applied Statistics*, 2009, vol. 36, iss. 5-6, pp. 537–545. URL: https://doi.org/10.1080/02664760802554263

32. Wu W.-W. Improving Classification Accuracy and Causal Knowledge for Better Credit Decisions. *International Journal of Neural Systems*, 2011, vol. 21, iss. 4, pp. 297–309. URL: https://doi.org/10.1142/S0129065711002845

33. Zhu H., Beling P.A., Overstreet G.A. A Bayesian Framework for the Combination of Classifier Outputs. *Journal of the Operational Research Society*, 2002, vol. 53, iss. 7, pp. 719–727. URL: https://doi.org/10.1057/palgrave.jors.2601262

34. West D. Neural Network Credit Scoring Models. *Computers & Operations Research*, 2000, vol. 27, iss. 11-12, pp. 1131–1152. URL: https://doi.org/10.1016/S0305-0548(99)00149-5

35. Ong C.-S., Huang J.-J., Tzeng G.-H. Building Credit Scoring Models Using Genetic Programming. *Expert Systems with Applications*, 2005, vol. 29, iss. 1, pp. 41–47. URL: https://doi.org/10.1016/j.eswa.2005.01.003

36. Breiman L. Bagging Predictors. *Machine Learning*, 1996, vol. 24, iss. 2, pp. 123–140. URL: https://doi.org/10.1007/BF00058655

37. Wolpert D.H. Stacked Generalization. *Neural Networks*, 1992, vol. 5, no. 2, pp. 241–259.

38. Vukovic S., Delibašić B., Uzelac A., Suknovic M. A Case-Based Reasoning Model That Uses Preference Theory Functions for Credit Scoring. *Expert Systems with Applications*, 2012, vol. 39, iss. 9, pp. 8389–8395. URL: https://doi.org/10.1016/j.eswa.2012.01.181

39. Marqués A.I., García V., Sánchez J.S. Two-Level Classifier Ensembles for Credit Risk Assessment. *Expert Systems with Applications*, 2012, vol. 39, iss. 12, pp. 10916–10922. URL: https://doi.org/10.1016/j.eswa.2012.03.033

40. Hoffmann F., Baesens B., Mues C. et al. Inferring Descriptive and Approximate Fuzzy Rules for Credit Scoring Using Evolutionary Algorithms. *European Journal of Operational Research*, 2007, vol. 177, iss. 1, pp. 540–555. URL: https://doi.org/10.1016/j.ejor.2005.09.044

41. Ignatius J., Hatami-Marbini A., Rahman A. et al. A Fuzzy Decision Support System for Credit Scoring. *Neural Computing and Applications*, 2016, vol. 27, no. 1, pp. 1–17.
URL: https://doi.org/10.1007/s00521-016-2592-1

42. Lahsasna A., Ainon R.N., Wah T.Y. Credit Risk Evaluation Decision Modeling Through Optimized Fuzzy Classifier. Proc. International Symposium on Information Technology, 2008. *IEEE*, 2008, vol. 1, pp. 1–8.

43. Kaur A. et al. Fuzzy Rule-based Expert System for Evaluating Defaulter Risk in Banking Sector. *Indian Journal of Science and Technology*, 2016, vol. 9, iss. 28, pp. 1–6.
URL: https://doi.org/10.17485/ijst/2016/v9i28/98395

44. Malhotra R., Malhotra D.K. Differentiating Between Good Credits and Bad Credits Using Neuro-Fuzzy Systems. *European Journal of Operational Research*, 2002, vol. 136, iss. 1, pp. 190–211.
URL: https://doi.org/10.1016/S0377-2217(01)00052-2

45. Clemen R.T., Murphy A.H., Winkler R.L. Screening Probability Forecasts: Contrasts Between Choosing and Combining. *International Journal of Forecasting*, 1995, vol. 11, iss. 1, pp. 133–145.
URL: https://doi.org/10.1016/0169-2070(94)02007-C

46. DeGroot M.H., Fienberg S.E. The Comparison and Evaluation of Forecasters. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 1983, vol. 32, no. 1/2, pp. 12–22.
Stable URL: http://www.jstor.org/stable/2987588

47. DeGroot M.H., Eriksson E.A. Probability Forecasting, Stochastic Dominance, and the Lorenz Curve. J.M. Bernardo, M.H. DeGroot, D.V. Lindley and A.F.M. Smith (eds). Amsterdam, North-Holland, Bayesian Statistics, 1985, vol. 2, pp. 99–118.

Conflict-of-interest notification

We, the authors of this article, bindingly and explicitly declare of the partial and total lack of actual or potential conflict of interest with any other third party whatsoever, which may arise as a result of the publication of this article. This statement relates to the study, data collection and interpretation, writing and preparation of the article, and the decision to submit the manuscript for publication.

Please cite this article as: Volkova E.S., Gisin V.B., Solov'ev V.I. Data Mining Techniques: Modern Approaches to Application in Credit Scoring. *Digest Finance*, 2017, vol. 22, iss. 4, pp. 400–412. https://doi.org/10.24891/df.22.4.400