INFORMATION
SYSTEMS IN
MANAGEMENT

Information Systems in Management (2016) Vol. 5 (1) 36–48

APPLICATION OF SELECTED SUPERVISED CLASSIFICATION METHODS TO BANK MARKETING CAMPAIGN

DANIEL GRZONKA [a)], GRAŻYNA SUCHACKA [b)], BARBARA BOROWIK [a)]

[a)] *Institute of Computer Science, Cracow University of Technology*
[b)] *Institute of Mathematics and Informatics, Opole University*

Supervised classification covers a number of data mining methods based on training data. These methods have been successfully applied to solve multi-criteria complex classification problems in many domains, including economical issues. In this paper we discuss features of some supervised classification methods based on decision trees and apply them to the direct marketing campaigns data of a Portuguese banking institution. We discuss and compare the following classification methods: decision trees, bagging, boosting, and random forests. A classification problem in our approach is defined in a scenario where a bank's clients make decisions about the activation of their deposits. The obtained results are used for evaluating the effectiveness of the classification rules.

Keywords: Classification, Supervised Learning, Data Mining, Decision trees, Bagging, Boosting, Random Forests, Bank Marketing, R Project

1. Introduction

Nowadays marketing has become an integral part of a companies' activities for looking for ways to promote goods and services focused on the consumer. Undoubtedly, it is also an important phenomenon in social and economic sciences. In economics, marketing issues have been studied using multivariate statistical analysis methods. The proper use of suitable methods for a particular problem has been an ongoing challenge that requires the utilization of knowledge about the possibilities of common techniques.

A significant increase in the computing power and memory has made it possible to collect and analyze large amounts of data. As a result a rapid development of knowledge discovery methods took place. The choice of a suitable tool for data analysis is not an easy task. This problem is still valid. The basis for intelligent data analysis (i.e. data mining) has become machine learning (ML) methods [1]. ML is an interdisciplinary science that with the help of artificial intelligence aims to create automated systems that can improve their operation taking advantage of gained experience and acquired new knowledge. ML methods have been widely and successfully used in all sectors - industry, services, research, economics, medicine, and others. Depending on the approach and the nature of applied methods, ML systems can be divided into three groups: supervised, unsupervised, and semi-supervised learning systems [2]. In this paper, we consider the issue of classification, which is a particular case of supervised machine learning. In a supervised learning system each observation (instance) is a pair consisting of an input vector of predictor variables and a desired output value (target variable). Data is provided by a "teacher" and the goal is to create a general model that links inputs with outputs. In the case of a classification problem this model is called a classifier. The goal of a classification process is to assign the appropriate category (the set of categories is known a priori) for an observation. A popular example is the classification of incoming mail as "spam" or "non-spam" [3]. Classification methods have been also applied to WWW, e.g., to identify and detect automated and malicious software in computer networks [4] or on Web servers [5]. They have been also successfully applied to text analysis, including website content analysis [6, 7]. Other popular area of application of supervised classification has been the electronic commerce, e.g. online sales prediction [8, 9, 10], and customer relationship management [11].

One of the most successful approaches for building classification models is decision tree learning, which became the basis for many other classification models. Decision trees are built using recursive partitioning which aims to divide the variable space until the target variable reaches a minimum level of differentiation in each subspace.

Classification trees have been mentioned for the first time in [12], but they gained popularity thanks to the work of Breiman et al. [13], which gave the name to the whole family of methods and algorithms based on the idea of Classification and Regression Trees (CART).

In this paper we consider a problem of predicting the effectiveness of a marketing campaign. The marketing campaign is a typical strategy for acquiring new customers or promoting new products. Knowledge about the effectiveness of marketing methods and the susceptibility of recipients is extremely valuable in many sectors. Without a doubt, this is also an important issue from the standpoint of statistical science. The problem of choosing the best set of customers is considered as NP-hard problem [14]. Based on data from a telemarketing campaign of one of the Portuguese banks [15] we propose classification models which predict the client's decision whether to deposit or not their savings in the bank. The proposed models are based on the idea of a classification tree.

The paper is organized as follows. Subsequent sections describe: decision trees (Section 2), ensemble methods: bagging (Section 3.1), boosting (Section 3.2), and random forests (Section 3.3). Finally, in Section 4 we discuss the results of our experiments. The paper is summarized and concluded in Section 5.

2. Decision Trees

Decision Trees (DTs) are a non-parametric supervised learning method used to build discrimination and regression models. In a graph theory, a tree is an undirected graph, which is connected and acyclic, that is a graph in which any two vertices are connected by exactly one path. In the case of a decision tree we have to deal with a directed tree in which the initial node is called the root. The nodes correspond to tests on attributes and the branches represent decisions. The whole learning set is initially cumulated in the root and then it is tested and passed to the appropriate node. Thus, in all nodes (except the last one), a split with the best optimization criterion is selected. Split criterion is the same on each node. Leaf nodes represent classes assigned to them and they correspond to the last phase of the classification process. In other words, for each new observation to which we want to assign a class, we must answer a series of questions related to the values of variables - the answers to these questions determine the choice of the appropriate class for that instance.

According to [16] in discrimination trees next to the branch splitting conditions are often given that determine the next node (a level below) for a considered sample. The nodes give a dominant class which contains elements of the subsample training set that were in that node.

A method for construction of discrimination models is the combination of local models built in each subspace. Splitting of the subspaces occurs sequentially (based on recursive partitioning) until it reaches a predetermined minimum level of differentiation. The process of building a classification tree is done in stages, starting with the distribution of elements of the learning set. This division is based on the best split of data into two parts, which are then passed to the child nodes. An example of a classification tree model is shown in Fig. 1.

An important issue is the choice of a splitting method. Input data at a node is characterized by the homogeneity of the target variable within the subsets. The aim of the division is to minimize this homogeneity. For this purpose, functions determining the homogeneity are used. The most popular are [16]:

1. Misclassification error:

$$Q_m(T)_1 = 1 - \hat{p}_{mk(m)} \tag{1}$$

2. Gini index:

$$Q_m(T)_2 = \sum_{k=1}^{g} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{2}$$

3. Entropy:

$$Q_m(T)_3 = -\sum_{k=1}^{g} \hat{p}_{mk} \log \hat{p}_{mk} \tag{3}$$

where *$Q_m(T)$* is the homogeneity ratio of node *m* of tree *T*, *k* means a class, *g* is the number of classes, and $\hat{p}_{mk}$ is the ratio of the number of instances of class *k* in node *m*, which can be calculated by the formula:

$$\hat{p}_{mk} = \frac{n_{mk}}{n_m} \tag{4}$$

where n_m is the number of instances in node *m* and n_{mk} is the number of instances of class *k* in node *m*.

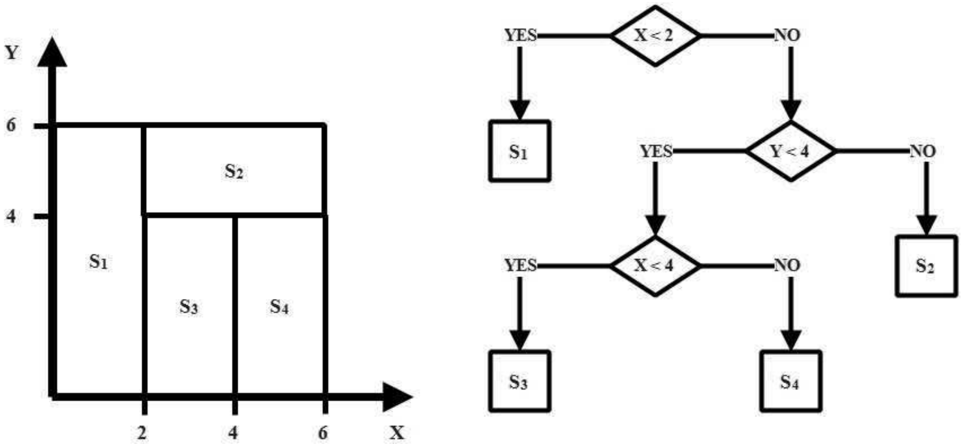

Figure 1. The decision tree corresponding to the division of space into subspaces.
Source: own elaboration on the basis of [17]

Observations considered in node *m* are classified into the most often represented class. If node *m* is a leaf, then it is the end result of the classification of the input vector. Otherwise, the process continues.

In the case of a two-class problem the above equations will be the following [16]:

$$Q_{m1}(T) = 1 - \max(p, 1 - p) \tag{5}$$

$$Q_{m2}(T) = 2p(1 - p) \tag{6}$$

$$Q_{m3}(T) = -p \log p - (1 - p)\log(1 - p) \tag{7}$$

where $Q_m(T)$ is the homogeneity ratio of node m of tree T and p is the ratio of the number of instances of the remaining class in node m.

The Gini index and entropy are most commonly used in CART methods as they allow for a locally optimal division of a sample. They do not guarantee finding a globally optimal solution. Due to the computational complexity a globally optimal solution is impossible to obtain in a finite time [16, 17].

Another important issue is determining the moment when the construction of the tree should be terminated. A disadvantage of this method is the excessive growth of the tree (over-fitting) causing a poor preparation for the future classification of new objects. This problem can be solved by pruning algorithms. Various approaches may be applied to deal with this problem, e.g. [18]:

1. All instances in the node belong to a single category.
2. The maximum tree depth has been reached.
3. The number of instances in the node is less than the pre-established minimum.
4. The best splitting criteria is not greater than a certain threshold.

Knowledge on the most important aspects of DT modelling is helpful in identifying their advantages. The trees are both flexible and capable of dealing with missing attribute values. Other advantages are the independence of attributes and insensitivity to irrelevant attributes. DTs have a high readability so they can be easily analyzed by an expert.

Unfortunately, classification trees also have a significant disadvantage. They are considered to be unstable - small changes in the learning data may yield substantially different trees, which increases the probability of misclassification [16].

3. Ensemble methods

Ensemble methods may use different learning algorithms to predict a proper class. The idea is to aggregate multiple classifiers in one model. The term “ensemble” is usually reserved for methods that generate multiple hypotheses using the same base learner.

The idea of joining classifiers dates back to 1977 [19] but the increased interest in this type of approach appeared only in 1990, when Hansen and Salomon in their work [20] presented the proof of improving the efficiency of classification through the aggregation of classifiers [17]. Algorithms for classifiers' families are usually based on decision trees that have been discussed in detail in the previous section. An ensemble learning approach involves combining weak classifiers, whose operation is little better than a random decision-making. At the same time, weak classifiers are characterized by the simplicity of construction and high speed of operation. It should be noted that the usage of a large number of different models (trained with the same method) makes a classification result more reliable. Unfortunately, in practice, classifiers created from the same training sample are statistically dependent on one another, which is the main drawback of this method, nevertheless they give good results [13].

3.1. Bagging

In 1996 L. Breiman [21] proposed one of the first ensemble methods, involving the bootstrap aggregation, proving at the same time that the error of the aggregated discrimination model is smaller than the average error of models that make up the aggregated model. This method is called bagging (*bootstrap aggregating*). As previously mentioned, methods based on families of classifiers use mainly decision trees - and in the rest of this paper we consider methods in which only decision trees are used [16].

Training of V decision trees requires V training samples U_1, ..., U_V. Every n-element sample comes from drawing with replacement from the training set U whose cardinality is N As one can notice, the probability of selecting a given observation is always constant and it equals $\frac{1}{n}$ [17].

The algorithm takes the following steps [17, 22]. We assume that a dataset has N observations and the target variable has a binary value.

1. Take a bootstrap sample from the data (i.e. a random sample of size n with replacement).
2. Construct a classification tree (the tree should not be pruned yet).
3. Assign a class to every leaf node. For every observation the class attached to every case coupled with the predictor values should be stored.
4. The steps from 1 to 3 need to be repeated a defined earlier large number of times.
5. For every observation in the dataset, the number of trees classifying this observation to one given category is counted over the number of trees.

Each observation needs to be assigned to a resulting final class using a majority vote method over the set of trees.

Unlike a single classification tree, a family of trees does not behave unstably and gives significantly better classification possibilities compared to a single tree.

3.2. Boosting

Another algorithm that is based on the idea of families of classifiers, which was created independently from the bagging method, is the boosting method being to a certain degree an improvement of the bagging method. As in the previously discussed algorithm, the boosting method is also based on drawing random training samples of size *n* with replacement from the training set - the difference is that the probability distribution (weights' distribution), according to which elements are drawn, changes from sample to sample. Then the classifier is constructed and its quality is verified [16].

The algorithm uses two types of weights. The first type refers to observations that have been wrongly classified by a given classifier - their weight is being increased. The second type of weights refers to classifiers, assigning to each one of them a weight value that is proportional to the prediction error that the given classifier makes. This means that weights of less accurate models are being reduced and weights of more accurate models are being increased [17].

The basic boosting algorithm is called a Discrete Adaboost (*Discrete Adaptive Boosting*). Similarly to bagging, this method requires *V* *n*-element training samples $U_1, ..., U_V$ from the training set *U*. The algorithm takes the following steps [23, 24]:

1. Set a number of training samples.
2. Set the initial weights $w_i = \frac{1}{n}$, where $i = 1,...,n$.
3. Repeat for $v = 1, ..., V$:
   a. Take a sample from the training set *U*.
   b. Train a weak classifier $f_v(x)$ and compute:

$$err_v = \sum_{i=1}^{n} w_i^{(v)} I(f_v(x) \neq y_i), \tag{8}$$

$$\alpha_v = \frac{1}{2}\log(\frac{1-err_v}{err_v}). \tag{9}$$

   c. Set $w_i^{(v+1)} = \frac{w_i^{(v)}}{2(1-err_v)}$ if $f_v(x_i) = y_i$, else $w_i^{(v+1)} = \frac{w_i^{(v)}}{2err_v}$. (10)

4. The output is the aggregated classifier: $\sum_{v=1}^{V} \alpha_v f_v(x)$. (11)

3.3. Random forests

Random forests, like the bagging and boosting algorithms, are based on families of classifiers but random forests can use only decision trees as individual classifiers.

The random forests algorithm was proposed by L. Breiman in 2001 [25]. It combines the bagging method and the idea of promoting good classifiers by seeking the best division (division rules have been mentioned in Chapter 2) using the best attributes (variables) of an observation.

The random forests algorithm is very similar to the bagging algorithm. It is relatively straightforward and is as follows [22]:

Let us assume that the target variable has a binary value and *N* is the number of observations.

1. Take a bootstrap sample from the data (i.e. a random sample of size *n* with replacement). From the set of predictors take a random sample without replacement.
2. Using predictors chosen in Step 2 construct a split within the tree.
3. For each subsequent split repeat Steps 2 and 3 until the tree has the required number of levels, without pruning the tree yet. In this way, every tree is random as during generating every tree obtained here at each split a random sample of predictors has been used.
4. Test the classification abilities of the tree for the out-of-bag data. The class assigned to every observation needs to be saved along with every observation's predictor values.
5. Steps 1 through 5 are repeated a required number of times, defined at the beginning.
6. For every observation in the dataset, the number of trees classifying this observation to one given category is counted over the number of trees.

Each observation needs to be assigned to a resulting final class using a majority vote method over the set of trees.

It is worth noting that due to the use of the bootstrap sampling, approximately 1/3 of training set elements is not involved in the process of building a family of trees. Thereby, a dependence between trees decreases and operations on sets with a big number of elements become easier [16].

4. Experimental Analysis

Experiments were conducted using data obtained from direct marketing campaigns of a Portuguese banking institution [15]. Data was collected during the campaign from May 2008 to November 2010 based on phone calls. S. Moro et al. have shared two datasets: a set with all examples and a set with 10% of the full dataset. In our research the second set was used.

The data set used to build classification models consists of 4521 instances. Each observation is defined by 17 attributes (an input vector of 16 predictor variables and a target variable). An input vector has both nominal and numerical values.

A target variable takes one of two values (classes). All the attributes are specified (there is no missing attribute value). The classification goal is to predict if a client will subscribe to a term deposit. From the 4521 samples, only 521 ended in a decision to open a deposit. Table 1 specifies all attributes.

Table 1. Specification of bank marketing campaign dataset attributes

Attribute name	Type	Values
Age	Numeric	19 to 87
Job	Categorical	admin., unknown, unemployed, management, housemaid, entrepreneur, student, blue-collar, self-employed, retired, technician, services
Marital (marital status)	Categorical	married, divorced (widowed), single
Education	Categorical	unknown, secondary, primary, tertiary
Default (has credit in default?)	Binary	yes, no
Balance (average yearly balance, in euros)	Numeric	-3 313 to 71 188
Housing (has housing loan?)	Binary	yes, no
Loan (has personal loan?)	Binary	yes, no
Contact (contact communication type)	Categorical	unknown, telephone, cellular
Day (last contact day of the month)	Numeric	1 to 31
Month (last contact month of year)	Categorical	Jan., Feb., Mar., ..., Nov., Dec.
Duration (last contact duration, in seconds)	Numeric	4 to 3 025
Campaign (number of contacts performed during this campaign and for this client)	Numeric	1 to 50
pDays (number of days that passed by after the client was last contacted from a previous campaign)	Numeric	-1 (first time) to 871
Previous (number of contacts performed before this campaign and for this client)	Numeric	0 to 25
pOutcome (outcome of the previous marketing campaign)	Categorical	unknown, other, failure, success
Target variable (has the client subscribed a term deposit?)	Binary	yes, no

In order to create classification models we used R project. R is a popular programming language and software environment for data analysis, statistical computing and modelling.

First, we analyzed the significance of individual attributes that define observations. For this purpose a decision tree was created based on the complete set of data. Using the Gini index we determined the most significant attributes. Each attribute received a value from 0 to 100. The total value of the weights for all attributes is equal to 100. The results are shown in Table 2.

Table 2. Significance of attributes in a single decision tree trained on the basis of the entire set of attributes

Attribute:	Duration	Day	Job	Month	Age	pOutcome	Balance	Education
Significance:	24	12	10	10	9	9	8	4
Attribute:	pDays	Marital	Campaign	Contact	Housing	Previous	Loan	Default
Significance:	3	3	3	3	1	1	0	0

In practice, some of the attributes are known a posteriori (after a telephone conversation with the customer). Unlike S. Moro et al. in [15], we decided to reduce the attributes to those that are known a priori and have the greatest impact on the process of classification. Analysis of the significance of the attributes and creation of classification models were done on the basis of eight selected features, shown in Table 3. As is apparent, the most important factor influencing the customer's decision is the success of previous campaigns. Other important factors are the month in which the campaign takes place, job and age of the customer.

Table 3. Significance of attributes in a single decision tree trained on the basis of the reduced set of attributes

Attribute:	pOutcome	Month	Job	Age	Balance	Education	Campaign	Marital
Significance:	47	19	15	12	4	2	1	<1

In the next part of our research, we built, tested, and compared selected classifiers based on the idea of decision trees. Each model was built based on attributes presented in Table 3. A common part of training methods configuration was based on the same parameters (model complexity: 0.001, the minimum number of observations for splitting: 5). For each method, the training set and the test set consisted of the same observations. 1/3 of all samples were designed for the training set and the rest - for a test set (which contained 3014 observations, including 334 ones with a decision to open a deposit). Results can be presented in the form of a confusion matrix and classification errors (they are explained in Table 4).

First, we trained a single decision tree. For this purpose, we used the *rpart* function (*rpart* library for R project). 84% of the observations were correctly classified. The decision tree coped well in terms of true positives (83 predictions) and slightly worse in terms of true negatives (2453 predictions). The second classifier based on bagging was created by *bagging* function (*ipred* library). This method classified negative customers' decisions well (2632 correct predictions). Unfortunately, it dealt much worse with positive decisions (only 51 correct predictions). Bagging was effective in 89% test cases. Comparable results were obtained for the boosting method (*boosting* function from *adabag* library) that incorrectly classified about 12% of the observations. The best overall results were obtained for random forests (function and library named *randomForest*) with the classification error equal to 0.1055. 60 observations were correctly classified as positive decisions, 2636 as negative. The detailed results of all experiments are presented in Table 5.

Table 4. Form of the confusion matrix with a classification error

	Expected results		
Prediction		No	Yes
	No	TN (true negative)	FN (false negative)
	Yes	FP (false positive)	TP (true positive)
Classification error: (FP+FN)/(Number of Instances)			

Table 5. Confusion matrix with classification errors for all tested methods

	Expected results								
Prediction		Decision tree		Bagging		Boosting		Random forests	
		No	Yes	No	Yes	No	Yes	No	Yes
	No	2453	251	2632	283	2614	288	2636	274
	Yes	227	83	48	51	66	46	44	60
Classification error:		0.159		0.110		0.117		0.106	

5. Conclusions

In our study we reviewed common tree-based classification methods. Using data on the effectiveness of real marketing campaigns we selected the most significant decision-making attributes describing the customers. Considering the full set of data, the most significant attribute was the duration of a call. Unfortunately, in reality this parameter is known only after performing a direct marketing operation. Therefore, in contrast to S. Moro et al. research presented in [15], we decided to omit such parameters in our study and not to include them in the construction of discriminant models. We used eight selected attributes to train classifiers. Compared to the results in [15], this treatment had a negative impact on the quality of classifiers, but instead it allowed for the prediction basing solely on a priori known attributes. According to our analysis, in a reduced attributes scenario the most significant parameter was the effectiveness of previous campaigns.

Four classification methods were applied: decision trees, bagging, boosting, and random forests. The trained classifiers were used to predict consumer decisions to open or not a deposit in the bank. Based on classification results presented in the form of confusion matrices and misclassification errors we evaluated the effectiveness of the methods. The best results were obtained for random forests. However, the largest percentage of true positive classifications was obtained for a single decision tree.

It should be mentioned that a very important factor for the obtained results could be the randomness of bootstrap samples used to build models. Due to the use of the bootstrap sampling, approximately 1/3 of training set elements is not involved in the processes of building a family of trees. Therefore, in any attempt to build classifiers another model could be obtained. However, despite of the difficulty of the considered issues, the obtained results suggest a sense of using decision tree-based methods to support planning and management of bank marketing campaigns.

Acknowledgement

Research presented in the paper was partly supported by the Polish Ministry of Science and Higher Education through the Faculty of Physics, Mathematics and Computer Science (Cracow University of Technology), grant no. F3/600/2015/DS-M.

Daniel Grzonka and Grażyna Suchacka are members of the consortium of ICT COST Action IC1406 High-Performance Modelling and Simulation for Big Data Applications (cHiPSet).

REFERENCES

[1] Odrzygóźdź Z., Szczęsny W. (2011) *Data Mining - how to select the tools for analysts in a large bank*, Information Systems in Management XIII, WULS Press, Warszawa, Poland, 97-103.

[2] Chapelle O. et al. (2006) *Semi-Supervised Learning*, The MIT Press, UK.

[3] Liu G., Yang F. (2012) *The application of data mining in the classification of spam messages*, Proceedings of CSIP'12, 1315-1317.

[4] Kruczkowski M., Niewiadomska-Szynkiewicz E. (2014) *Comparative study of supervised learning methods for malware analysis*, JTIT, Vol. 4, 24-33.

[5] Suchacka G., Sobków M. (2015) *Detection of Internet robots using a Bayesian approach*, Proceedings of CYBCONF'15, Gdynia, Poland, 365-370.

[6] Soiraya B., Mingkhwan A., Haruechaiyasak C. (2008) *E-commerce web site trust assessment based on text analysis*, International Journal of Business and Information, vol. 3, Issue 1, 86-114.

[7] Shen D., Ruvini J.-D., Sarwar B. (2012) *Large-scale item categorization for e-commerce*, Proceedings of CIKM'12, Maui, HI, USA, 595-604.

[8] Suchacka G., Skolimowska-Kulig M., Potempa A. (2015) *A k-Nearest Neighbors method for classifying user sessions in e-commerce scenario*, JTIT, Vol. 3, 64-69.

[9] Suchacka G., Skolimowska-Kulig M., Potempa A. (2015) *Classification of e-customer sessions based on Support Vector Machine*, Proceedings of ECMS'15, Albena, Bulgaria, 594-600.

[10] Hop W. (2013) *Web-shop order prediction using machine learning*, Masters Thesis, Erasmus University Rotterdam.

[11] Ngai E. W. T., Xiu L., Chau D. C .K. (2009) *Application of data mining techniques in customer relationship management: A literature review and classification*, Expert Systems with Applications, Vol. 36, Issue 2, Part 2, 2592-2602.

[12] Morgan J. N., Sonquist J. A. (1963) *Problems in the analysis of survey data, and a proposal*, Journal of the American Statistical Association, Vol. 58, Issue 302, 415-434.

[13] Breiman L. et al. (1984) *Classification and Regression Trees*, Wadsworth, CA.

[14] Nobibon F et al. (2011) *Optimization models for targeted offers in direct marketing: Exact and heuristic algorithms*, European Journal of Operational Research, Vol. 210, Issue 3, 670–683.

[15] Moro S. et al. (2011) *Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology*, Proceedings of the European Simulation and Modelling Conference - ESM'2011, Portugal, 117-121.

[16] Koronacki J., Ćwik J. (2005) *Statystyczne systemy uczące się*, WNT, Warszawa, Poland.

[17] Walesiak M., Gatnar E. (2009) *Statystyczna analiza danych z wykorzystaniem programu R*, PWN, Warszawa, Poland.

[18] Rokach L., Maimon O. (2005) *Decision trees* in The Data Mining and Knowledge Discovery Handbook, Springer, 165–192.

[19] Tukey J.W. (1977) *Exploratory data analysis*, Addison-Wesley, Reading.

[20] Hansen L. K., Salamon P. (1990) *Neural network ensembles*, IEEE Transactions on Pattern Analysis & Machine Intelligence, Vol. 12, Issue 10, 993-1001.

[21] Breiman L. (1996) *Bagging predictors*, Machine learning, Vol. 24, Issue 2, 123-140.

[22] STAT 897D: Applied Data Mining and Statistical Learning course (available online: https://onlinecourses.science.psu.edu/stat857/).

[23] Freund, Y., Schapire R. E. (1996) *Experiments with a new boosting algorithm* in ICML, Vol. 96, 148-156.

[24] Babenko B. *Note: A Derivation of Discrete AdaBoost* (available online: http://vision.ucsd.edu/~bbabenko/data/boosting_note.pdf).

[25] Breiman, L. (2001) *Random forests*, Machine learning, Vol. 45, Issue 1, 5-32.