Chapter 13

Examples of the Use of Data Mining Methods in Animal Breeding

Wilhelm Grzesiak and Daniel Zaborski

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/50893

1. Introduction

Data mining techniques involve mainly searching for various relationships in large data sets. However, they can also be used in a much narrower range, sometimes as an alternative to classical statistics. The characteristic feature of these models is the use of a specific strategy, usually requiring the division of data into training set, sometimes also verification set, which enable the evaluation of the model quality as well as a test set for checking its prognostic or classification abilities. Among many different methods belonging to data mining, the following can be distinguished: the general models of classification and regression trees (G_Trees), general CHAID (Chi-square Automatic Interaction Detection) models, interactive classification and regression trees (also with boosting – Boosted Trees), random forest, MARS (Multivariate Adaptive Regression Splines), artificial neural networks (ANN), other machine learning methods such as: naïve Bayes classifier (NBC), support vector machines (SVM), k-nearest neighbors (k-NN) and other regarded (or not) by different authors as data mining techniques. These methods are more and more frequently applied to various issues associated with animal breeding and husbandry.

2. Various methods used in data mining – Multivariate adaptive regression splines, naïve Bayes classifier, artificial neural networks, decision trees

2.1. Multivariate adaptive regression splines

MARS, introduced by Jerome Friedman in 1991 [1], is mainly used for solving regression-type problems. It is "a nonparametric regression method that approximates a complex non-linear relationship with a series of spline functions defined on different intervals of the independent (predictor) variable" [2]. Moreover, MARS makes it possible to fit non-linear

multivariate functions. In this method, no assumptions about the analyzed functional relationship between variables are made. Instead, this relationship is determined based on regression data [3, 4]. Contrary to the global parametric models, MARS operates locally. It can be considered as a generalization of the binary recursive partitioning, in which the problem of the occurrence of the disjoint subregions and thus discontinuity of the approximating functions at the boundaries of these subregions, has been eliminated [2]. It utilizes left-sided and right-sided truncated power functions as spline functions:

$$(t-x)_+^q = \begin{cases} (t-x)^q, for\_x < t \\ 0, otherwise \end{cases},$$

$$(x-t)_+^q = \begin{cases} (x-t)^q, for\_x > t \\ 0, otherwise \end{cases},$$

where q $(q \geq 0)$ is a power, to which the spline functions are raised to enable the adjustment of the smoothness of the obtained function estimate and t is a knot [5]. Basis functions in MARS can be a single spline function or a product of two (or more) such functions. The main idea behind MARS is the use of the combination of basis functions for the approximation of the relationship between the dependent variable and predictors:

$$\hat{y} = a_0 + \sum_{m=1}^{M} a_m B_m(\mathbf{x}),$$

where: $\hat{y}$ is a dependent variable, a_0 is a coefficient of the constant basis function, $B_m(\boldsymbol{x})$ is an mth basis function, a_m is a coefficient of the mth basis function and M is a number of basis functions in a model [4, 6].

An optimal MARS model is constructed in two stages. First, the model containing too many basis functions that lead to its overfitting is created. At this stage, it is also possible to take into account interactions between predictors or they can constitute only additive components [7]. At the second stage of the algorithm execution (pruning), these basis functions that contribute least to the goodness-of-fit are removed [8]. Elimination of these functions is based on the generalized cross-validation error (GCV):

$$GCV = \frac{\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2}{\left(1 - \frac{C}{n}\right)^2}, C = 1 + cd,$$

where: n is a number of cases in a data set, d is degrees of freedom equal to the number of independent basis functions, c is a "penalty" for the addition of the next basis function to the model, y_i is an actual value of the dependent variable, $\hat{y}_i$ is the value predicted by the model [2, 4, 9].

MARS, apart from regression tasks, can be used for classification. In the case of only two classes, dependent variable is coded as a binary one and further procedure is the same as in regression problems, whereas with more categories, the indicator variables are used and a model with a multivariate dependent variable is applied [10].

2.2. Naïve Bayes classifier

Naïve Bayes is a very simple and, at the same time, effective classifier. It can handle an arbitrary number of continuous and categorical variables [4]. It is based on the following Bayes' rule for conditional probability:

$$P(A \mid B) = P(B \mid A)\frac{P(A)}{P(B)},$$

where: $P(A)$ and $P(B)$ are probabilities of events A and B, respectively [11]. In terms of classification problems this rule can be expressed as:

$$P(y \mid \mathbf{x}) = P(y)\frac{P(\mathbf{x} \mid y)}{P(\mathbf{x})},$$

where: $\boldsymbol{x}=(x_1, x_2,.., x_N)$ is a feature vector and y is the class [11, 12].

In general, Bayesian classifiers determine the class to which an observation described by its feature vector belongs and the training process of such classifiers can be much simplified by assuming that the features are independent given class [4, 13]:

$$P(\mathbf{x} \mid y) = \prod_{j=1}^{N} P(x_j \mid y).$$

In practical applications, this assumption is often not fulfilled, however, it turns out that this fact does not significantly affect the quality and precision of classification [4, 11, 14]. Since $P(\boldsymbol{x})$ is the same for all classes, it can be omitted and thus the *a posteriori* probability according to which maximum value the observations are assigned to a given class takes the following form [12, 13]:

$$P(y \mid \mathbf{x}) = P(y)\prod_{j=1}^{N} P(x_j \mid y).$$

The main advantage of NBC is its simplicity and speed, whereas the disadvantage is the lack of any explanation of the decision made by the classifier [14].

2.3. Artificial neural networks

Artificial neural network (ANN) is an information processing system inspired by the biological systems such as the human brain. The characteristic features of the brain include

incremental information processing, learning new concepts, taking decisions and drawing conclusions based on complex, sometimes irrelevant or incomplete data. The popularity of ANNs results from their ability to reproduce the processes occurring in the brain, although to a limited extent [15]. Therefore, ANNs represent different approach than traditional statistical methods in which it is necessary to define an algorithm and record it in the form of a computer program. Instead, ANNs are presented with exemplary tasks and the connections between the network elements as well as their weight coefficients are modified automatically according to the assumed training strategy. Besides the ability of self-programming, ANNs also show reduced sensitivity to the damages of their structure elements and are capable of the parallel data processing [16, 17].

The basic element of ANN is an artificial neuron, which is a very simplified model of a living nerve cell (Fig. 1) [18]. The so-called input signals (in the form of independent, explanatory variables) are sent to the inputs of an artificial neuron. They are subsequently multiplied by the corresponding weight coefficients (equivalents of synaptic potentials in the living nerve cells). The next stage of the artificial neuron functioning is obtaining an auxiliary internal signal, the so-called postsynaptic potential s [19]. This potential can be expressed using the following equation:

$$s = \sum_{j=1}^{N} w_j x_j,$$

where: x_j is a jth input signal, w_j – weight associated with a jth neuron input, N – number of neuron inputs [20].

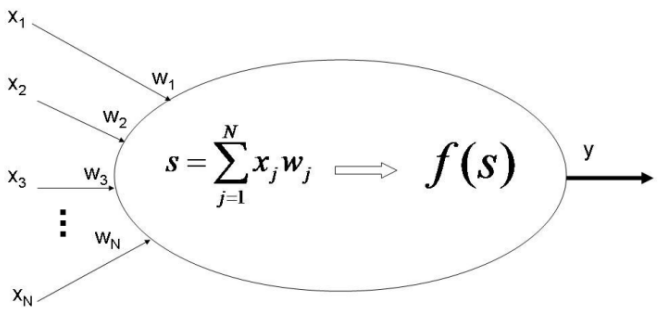

Figure 1. Schematic representation of an artificial neuron (without bias weight)

To this sum of signals, the neuron sometimes adds an additional component called bias weight, which is independent from the input signals and, if taken into account, also undergoes the learning process. Bias weight, which is associated with the constant input $x_0=1$, makes it possible to define the properties of a neuron more freely [16,18, 21]. If a neuron represents a multiple regression model, bias weight can be regarded as an intercept [22]. The weighted sum of input signals with an added (possibly) bias weight can sometimes be passed directly to the output of an artificial neuron constituting its output signal. In the

more complex types of ANNs the output signal is, however, calculated using the so-called activation function [16,18]. Activation function can be a linear function – then the output signal y is calculated as: $y=bs$, where b is a given coefficient [23]. Another type of activation function is the unit step function. Then the output signal takes the following form:

$$y = \begin{cases} 0\_when\_s < 0 \\ 1\_when\_s \geq 0 \end{cases},$$

where s - the postsynaptic potential value [20]. To describe more precisely the non-linear characteristics of the biological neuron, sigmoid functions, including logistic and hyperbolic tangent can be used. They are frequently applied, especially to solve more complex issues [15]. The logistic function can be expressed with the following formula:

$$y = \frac{1}{1 + e^{(-bs)}},$$

where: b – a coefficient determining the shape of the logistic function, most often equal to 1, s – the value of the postsynaptic potential, e – base of the natural logarithm [20, 21, 24].

The algorithm used to train a single neuron (supervised method) assumes that with each input vector $\boldsymbol{x}_i$ presented to the neuron, a desired or real output value y_i corresponding to this input vector is also presented. In response to the input vector, the neuron produces the output signal $\hat{y}_i$. However, if the neuron is not fully trained, this signal differs from the desired one. Therefore, the error is calculated, which is then used to modify the weights w_j so that the neuron better approximates the relationship between input and output values [16]. This process is repeated many times until the lowest possible error is obtained. The initial weight values are usually selected at random, and they are modified in the successive iterations of the algorithm according to the gradient of an error function in the space defined by the number of neuron inputs [18].

Perceptrons are one of the ANNs types (Fig. 2). Initially, the name was reserved only for feed-forward neural networks with neurons using threshold activation function. Later, this name included also multilayer feed-forward networks with neurons having continuous activation functions [20]. In perceptrons, the signals are sent only in one direction, that is, from the network input, from which it takes the input data, to the network output, in which the network returns solution [25]. The neurons are organized in layers and the neurons of one layer are connected with all the neurons of the next layer. The neurons of the same layer cannot connect with each other and there is no feed-back to preceding layers [21, 26]. The task of the neurons of the input layer is the preprocessing of input data, which usually involves normalization or scaling. The main processing takes place, however, in the hidden and output layers [25]. The name "hidden layer" results from the fact that it does not have a direct contact with the inputs or outputs of the network [18]. The presence of the hidden layers (in the ANNs with neurons having non-linear activation functions) significantly extends the range of mapping that the network can realize. A single hidden layer is sufficient in such networks to realize any mapping relating input to output signals [21, 25].

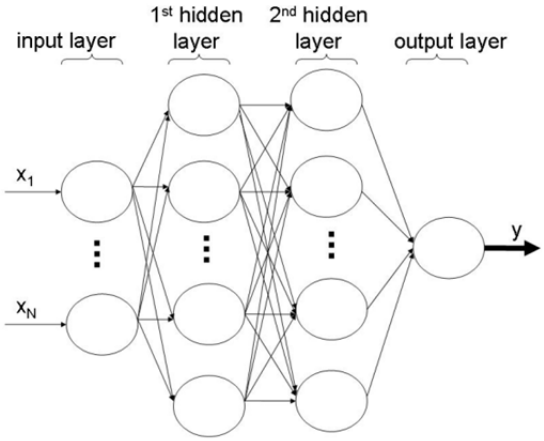

Figure 2. Schematic representation of the feed-forward artificial neural network with two hidden layers

The second frequently used ANN type is radial basis function (RBF) networks. In the case of the RBF networks, the input signals making the vector $\boldsymbol{x}$ are fed to each neuron of the hidden layer [26]. Thus, unlike in MLPs, the connections between input-layer and hidden-layer neurons are direct connections without weights [21]. In the hidden-layer, activation functions of the neurons perform the following mapping:

$$\mathbf{x} \to \varphi(\|\mathbf{x}-\mathbf{c}\|), \mathbf{x} \in \mathbf{R}^n$$

where $(||\cdot||)$ most often denotes Euclidean norm and $\boldsymbol{R}^n$ is n-dimensional space [21, 26]. The functions $\varphi(||\boldsymbol{x}\text{-}\boldsymbol{c}||)$ are called radial basis functions and $\boldsymbol{c}$ denotes the center of a given radial basis function. The number of these neurons is equal to the number of cases in the training set or lower. The neuron of the output layer computes the weighted sum of the output signals from the hidden-layer neurons [20, 26]:

$$\hat{y} = \sum_i w_i \varphi(\|\mathbf{x}-\mathbf{c}_i\|).$$

The most frequently used basis function is the Gaussian function of the form:

$$\varphi(\|\mathbf{x}-\mathbf{c}_i\|) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{c}_i\|^2}{2\sigma_i^2}\right)$$

where $\sigma>0$ is a parameter [24].

An important issue in the practical application of the ANNs is the scaling of input signals to the range appropriate for the network with the aim of their standardization. The methods used in this case are min-max or mean-deviation. The result of the first method can be expressed with the following formula:

$$x_i^* = \frac{x_i - \min(x_i)}{range(x_i)} = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)},$$

where x_i* ranges from 0 to 1 [15, 27]. The output signal from the network is also scaled appropriately. For categorical predictors, it is first necessary to convert them to the numeric form. Two methods are commonly used for this purpose: one-of-N encoding and numerical encoding. In the first method, one neuron is allocated for each possible nominal value. During the learning process, one of the neurons is on and the others are off [4]. On the other hand, in the numerical representation, numerical values that are fed to the network inputs are assigned to the consecutive categories of the nominal variable. The use of this representation causes that one neuron in the input layer corresponds to one nominal variable, however, by numbering the values of the nominal variable, the user defines its ordering, which is not always justified [18].

In the process of ANN learning, the basic role is played by weight vectors. A single weight vector determines the behavior of an artificial neuron, whereas the weight matrix – the behavior of the whole network. The main algorithm of the MLP training is an error back-propagation [28]. During the optimization process, weights are modified each time after the presentation of a given training case. The learning process is based on a training sequence consisting of the pairs $<\boldsymbol{x}_i, y_i>$, where $\boldsymbol{x}_i$ is an ith vector of input values, y_i is a desired output value defined for each $i=1,..,n$, i is the number of the training vector and n is the number of training cases [16]. In the MLP functioning, the following stages can be distinguished [23]:

1. feeding the ith input vector $\boldsymbol{x}_i$ to the input layer of the network,
2. computation of the s_{ik}^h value for each neuron of the hidden layer according to the following formula:

$$s_{ik}^h = \sum_{j=0}^{N} w_{kj}^h x_{ij},$$

where: s_{ik}^h - weighted sum of the input signals for the kth neuron of the hidden layer, h – label of the hidden-layer neuron, $j=0,\ldots,N$, N – number of input-layer neurons, w_{kj}^h - weight from jth neuron of the input layer to the kth neuron of the hidden layer, x_{ij} – jth input signal for the ith training case.

3. computation of the output value y_{ik}^h for each neuron of the hidden layer :

$$y_{ik}^h = f_k^h(s_{ik}^h),$$

where: $f_k^h(\cdot)$ is an activation function of the hidden-layer neuron.

4. computation of the s_{il}^o value for each neuron of the output layer:

$$s_{il}^o = \sum_{k=0}^{K} w_{lk}^o y_{ik}^h,$$

where: s_{il}^{o} is a weighted sum of signals from the hidden-layer neurons for the lth neuron of the output layer, o - the label of the output-layer neuron, $k=0,..,K$, K – the number of hidden-layer neurons, w_{lk}^{o} - the weight from the kth neuron of the hidden layer to the lth neuron of the output layer,

5. calculation of the output value $\hat{y}_i$ of the output-layer neuron:

$$\hat{y}_i = f_l^o(s_{il}^o),$$

where: $f_l^o(\cdot)$ is an activation function of the output-layer neuron.

After performing all the above-mentioned phases, the network determines its output signal $\hat{y}_i$. This signal can be correct or incorrect but the role of the learning process is to make it as similar as possible (or identical in an ideal case) to the desired output signal y_i [28]. This can be achieved by appropriately modifying network weights so that the following error function E (for a single neuron in an output layer) is minimized [15, 16, 23]:

$$E = \frac{1}{2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2.$$

The optimization method used for this purpose is a gradient descent. The error function gradient is evaluated for each training case at a time and the weights are updated using the following formula [20]:

$$\Delta \mathbf{w}^{(t)} = -\eta \nabla E_i(\mathbf{w}^{(t)}),$$

where: $\Delta \mathbf{w}^{(t)}$ is a weight vector update at step t, η is a learning rate in the range [0,1], $\nabla E_i(\mathbf{w}^{(t)})$ is a gradient of the function E_i in point $\boldsymbol{w}^{(t)}$, E_i is an error for the ith training case:

$$E_i = \frac{1}{2}(y_i - \hat{y}_i)^2.$$

Both the weights of the output neuron and those of the hidden-layer neurons are updated during this process. The weight modification requires the calculation of the partial derivatives of an error with respect to each weight [23, 28]:

$$\Delta w_{lk}^{o} = -\eta \frac{\partial E_{\mathbf{i}}}{\partial w_{lk}^{o}}, \Delta w_{kj}^{h} = -\eta \frac{\partial E_{\mathbf{i}}}{\partial w_{kj}^{h}}$$

In order to make the back-propagation algorithm more effective, the momentum term α is often added to the equation for the weight modification:

$$\Delta w(t+1) = -\eta \frac{\partial E_{\mathbf{i}}}{\partial w(t+1)} + \alpha \Delta w(t)$$

where $\Delta w(t+1)$ is a weight update at step *t+1* and $\Delta w(t)$ is a weight update at step *t* [27].

The RBF network learning algorithm consists of two stages: (1) first, the position and shape of the basis functions are determined using one of the following methods: random selection, self-organization process, error back-propagation; (2) next, the weight matrix of the output layer is obtained in one step using the pseudoinversion method [26].

An important issue in the classification and regression by means of ANNs is to establish which variables in the model contribute most to the class determination or prediction of the value of continuous variable. An ANN sensitivity analysis is used for this purpose [15]. Elimination of individual variables affects the total network error and thus it is possible to evaluate the importance of these variables. The following indices are used [4, 29]:

1. error – determines how much the network's quality deteriorates without including a given variable in the model; the larger the error, the more important the variable;
2. ratio – the ratio of the above mentioned error to an error obtained using all variables, the higher the ratio, the more important the variable; the ratio below 1 indicates the variables that should be excluded from the model to improve the network quality;
3. rank – orders the variables according to decreasing error, the higher the rank, the more important the variable.

2.4. Decision trees

In mathematical terms, decision tree can be defined as a directed, acyclic and connected graph, having only one distinguishable vertex called a root node [30]. The tree structure consists of nodes and branches connecting these nodes [4]. If a node has branches leading to other nodes, it is called a parent node and the nodes to which these branches lead are called children of this node. The terminal nodes are called leaves [30]. Classification and regression trees (CART) are one of the types of decision trees.

CART were proposed by Leo Breiman et al. in 1984 [31]. The characteristic feature of CART is that the decision trees constructed by this algorithm are strictly binary. The cases from the training set are recursively partitioned into subsets with similar values of the target variable and the tree is built through the thorough search of all available variables and all possible divisions for each decision node, and the selection of the optimal division according to a given criterion [27].

The splitting criterions have always the following form: the case is moved to the left child if the condition is met, and goes to the right child otherwise. For continuous variables the condition is defined as "explanatory variable $x_j \leq C$". For the nominal variables, the condition expresses the fact that the variable takes on specific values [32]. For instance, for the variable "season" the division can be defined as follows: a case goes to the left child if "season" is in {spring, summer} and goes to the right child otherwise.

Different impurity functions $\varphi(\boldsymbol{p})$ can be used in decision nodes but the two most commonly applied for classification are Gini index and entropy:

$$\varphi(\mathbf{p}) = \sum_j p_j(1-p_j),$$
$$\varphi(\mathbf{p}) = -\sum_j p_j \log p_j,$$

where: $\boldsymbol{p}=(p_1, p_2,\ldots, p_J)$ are the proportions of classes *1, 2,.., J* in a given node [33].

In order to avoid overtraining, which leads to reduced generalization ability, the CART algorithm must initiate the procedure of pruning nodes and branches. This can be done using the test set or the V-fold cross-validation [27].

3. Classification example – The use of various data mining methods for the analysis of artificial inseminations and dystocia in cattle

An example of the application of data mining methods in the animal husbandry can be the detection of dairy cows with problems at artificial insemination by means of ANNs. The effectiveness of artificial insemination depends on meeting the following conditions: cow has healthy reproductive organs and is in the appropriate phase of reproductive cycle, artificial insemination is performed within 12 – 18 hours since the occurrence of the external estrus symptoms, the bull semen has appropriate quality, artificial insemination is performed correctly [34]. The possibility of identifying cows that can have problems at artificial insemination allows the farmer to more carefully treat such animals and eliminate potential risks associated with conception. A larger number of artificial inseminations increases the costs of this process and affects various reproductive indices, which in turn reduces the effectiveness of cattle farming.

In the aforementioned work [35], the set of 10 input variables determining potential difficulties at artificial insemination was used. They included, among other things, percentage of Holstein-Friesian genes in cow genotype, lactation number, artificial insemination season, age at artificial insemination, calf sex, the length of calving interval and pregnancy, body condition score and selected production indices. The output variable was dichotomous and described the class of conception ease: (1) conception occurred after 1 - 2 services or (2) after 3 or more services (3 - 11 services). The whole set of artificial insemination records (918) was randomly divided into 3 subsets: training (618 records), validation (150 records) and test (150 records) sets. To ensure appropriate generalization abilities of ANNs, a 10-fold cross-validation was applied. ANNs were built and trained by means of Statistica® Neural Networks PL v 4.0F software. The search for the best network from among many ANN categories was performed. The best network from each category (selected on the basis of the root-mean-square error – RMS) was utilized for the detection process. An MLP with 10 and 7 neurons in the first and the second hidden layers, respectively, trained with the back-propagation method was characterized by the best results of such detection. The percentages of correct indications of cows from both distinguished categories (altogether) as well as those of the correct detection of cows with difficulties at conception and without them were similar and amounted to approx. 85%. The ANN sensitivity analysis was applied to identify the variables with the greatest influence on

the value of the output variable (category of conception ease). Of the variables used, the following were the most significant: length of calving interval, lactation number, body condition score, pregnancy length and percentage of Holstein-Friesian genes in cow genotype.

Another method from the data mining field applied to the detection of cows with artificial insemination problems is MARS [35]. The effectiveness of this method was verified on the data set with analogous variables as those used for ANN analysis. From the whole set of records, two subsets were formed: training (768 records) and test (150 records) sets, without the validation set. In the model construction, up to 150 spline functions were applied, some of which were subsequently removed in the pruning process so as not to cause the overfitting of the model to the training data, which results in the loss of generalization abilities. The generalized cross-validation (GCV) error enabled the evaluation of the analyzed MARS models. The best model selected according to this criterion was used to perform the detection of cows with difficult conception. The percentages of correct detection of cows from both categories as well as percentages of correct indication of cows with difficulties at artificial insemination and those without such problems amounted to 88, 82 and 91%, respectively. Based on the number of references, it was also possible to indicate variables with the greatest contribution to the determination of conception class (length of calving interval, body condition score, pregnancy length, age at artificial insemination, milk yield, milk fat and protein content and lactation number).

Other data mining methods, CART and NBC, applied to the detection of cows with conception problems also turned out to be useful [36]. Based on the similar set of input data (the percentage of Holstein-Friesian genes in cow genotype, age at artificial insemination, length of calving-to-conception interval, calving interval and pregnancy, body condition score, milk yield, milk fat and protein content) and a similar dichotomous output variable in the form of the conception class (difficult or easy), 1006 cases were divided into training (812 records) and test (194 records) sets. Using Statistica ® Data Miner 9.0 software, the Gini index was used as an impurity measure in the construction of the CART models. The obtained models were characterized by quite a high sensitivity, specificity and accuracy of detection on the test set (0.72, 0.90, 0.85 for NBC and 0.83, 0.86, 0.90 for CART). In the case of CART, it was also possible to indicate the key variables for the determination of the conception class: the length of calving and calving-to-conception intervals and body condition score. The presented data mining methods used to support the monitoring of cows selected for artificial insemination can be an ideal tool for a farmer wishing to improve breeding and economic indices in a herd.

Another example of the application of such methods is the use of ANNs for the detection of difficult calvings (dystocia) in heifers [37]. Dystocia is an undesired phenomenon in cattle reproduction, whose consequence is, among other things, an increased risk of disease states in calves, their higher perinatal mortality, reduced fertility and milk yield in cows as well as their lower survival rate [38]. Dystocia also contributes to increased management costs, which result from the necessity of ensuring the permanent supervision of cows during parturition. Financial losses associated with dystocia can reach even 500 Euro per case [39]. According to various estimates, the frequency of dystocia in Holstein cows ranges from

approx. 5% to approx. 23% depending on the level of its severity and the parity [40]. The reasons for dystocia in cattle can be divided into direct and indirect. The former include, among other things, insufficient dilation of the vulva and cervix, uterine torsion and inertia, small pelvic area, ventral hernia, too large or dead fetus, fetal malposition and malpresentation, fetal monstrosities [41,42]. These factors are difficult to account for and can occur without clear reasons. Because of that, their potential use for prediction purposes is limited. On the other hand, indirect factors such as: age and body weight of cow at calving, parity, body condition score, nutrition during gestation, cow and calf breed, calving year and season, management and diseases can be used to some extent as predictors of calving difficulty in dairy cows. Susceptibility to dystocia has also genetic background [42]. This is mainly a quantitative trait, although some major genes, which can determine calving quality and constitute additional predictors of calving difficulty class, have been identified. The limitation of the occurrence of dystocia can be achieved using various prediction models, constructed on the basis of different variables. By means of such models, it is possible to indicate in advance animals with calving difficulties, which often allows the farmer to take action against dystocia. In the cited study [37], the authors used the following input variables: percentage of Holstein-Friesian genes in heifer genotype, pregnancy length, body conditions score, calving season, age at calving and three previously selected genotypes. The dichotomous output variable was the class of calving difficulty: difficult or easy. The whole set of calving records (531) was divided into training, validation and test sets of 330, 100 and 101 records, respectively. The authors selected the best networks from among MLP and RBF network types based on the RMS error. The networks were trained and validated using Statistica ® Neural Networks PL v 4.0F software. An analysis of the results obtained on a test set including cases not previously presented to the network showed that the MLP was characterized by the highest sensitivity (83%). This network had one hidden layer with four neurons. Specificity and accuracy were similar and amounted to 82%. The ANN sensitivity analysis showed that calving ease was the most strongly affected by pregnancy length, body condition score and percentage of Holstein-Friesian genes in heifer genotype.

Besides detecting dystocia in heifers, ANNs were also successfully applied to the detection of difficult calvings in Polish Holstein-Friesian cows [43]. In this case, the following predictors were used: percentage of Holstein-Friesian genes in cow genotype, gestation length, body condition score, calving season, cow age, calving and calving-to-conception intervals, milk yield for 305-day lactation and at three different lactation stages, milk fat and protein content as well as the same three genotypes as those for heifers. The whole data set of calving records (1221) was divided into three parts of 811, 205, and 205 records for the training, validation and test sets, respectively. Using Statistica Neural Networks ® PL v 4.0F software, the best ANN from each category (MLP with one and two hidden layers, RBF networks) was searched for on the basis of its RMS error. Then the selected networks were verified on the test set. Taking into account sensitivity on this set, the MLP with one hidden layer had the best performance (80% correctly detected dystotic cows), followed by the MLP with two hidden layers (73% correctly diagnosed cows with dystocia). The ability of the RBF network to detect cows with calving difficulties was smaller (sensitivity of 67%). Sensitivity

analysis showed that the most significant variables in the neural model were: calving season, one of the analyzed genotypes and gestation length.

4. Regression tasks - Milk yield prediction in cattle

The use of an important data mining method, ANN, in regression problems can be briefly presented on the basis of predicting lactation milk yield in cows. Such a prediction is significant both for farmers and milk processors. It makes it possible to appropriately plan milk production in a herd and is the basis for taking decisions on culling or retaining an animal already at an early lactation stage [44]. The commercial value of a cow is estimated by comparing its milk yield with the results of cows from the same herd, in the same lactation and calving year-season. Moreover, obtaining information on the potential course of lactation allows the farmer to appropriately select the diet, more precisely estimate production costs and profits, diagnose mastitis and ketosis [45]. Milk yield prediction is also important for breeding reasons. The selection of genetically superior bulls is, to a large extent, dependent on their ability to produce high-yielding daughters. Therefore, the sooner these bulls are identified, the sooner the collection of their semen and artificial insemination can begin. In the species like cattle, in which the generation interval is approx. 5 years, every method that can contribute to the milk yield prediction in cows before the completion of lactation will speed up the process of bull identification and increase genetic progress [46]. In the cited work [47], the input variables in the neural models were the evaluation results from the first four test-day milkings, mean milk yield of a barn, lactation length, calving month, lactation number, proportion of Holstein-Friesian genes in animal genotype. Linear networks (LNs) and MLPs were designed using Statistica ® Neural Networks PL v 4.0F software. A total set of milk yield records included 1547 cases and was appropriately divided into subsets (training, validation and test sets). The RMS errors of the models ranged between 436.5 kg and 558.2 kg. The obtained values of the correlation coefficient between the actual and predicted milk yield ranged from 0.90 to 0.96. The mean milk yield predictions generated using ANNs did not deviate significantly from those made by SYMLEK (the computer system for the comprehensive milk recording in Poland) for the analyzed herd of cows. However, the mean prediction by the one-hidden-layer MLP was closer to the values obtained from SYMLEK than those generated with the remaining models.

A similar study on the use of ANNs for regression problems concerned predictions for 305-day lactation yield in Polish Holstein-Friesian cows based on monthly test-day results [48]. The following 7 input variables were used: mean 305-day milk yield of the barns in which the cows were utilized, days in milk, mean test-day milk yield in the first, second, third and fourth month of the research period and calving month. MLP with 10 neurons in the hidden layer was designed using Statistica ® Neural Networks PL v 4.0F software. The whole data set (1390 records) was appropriately divided into training, validation and test set of 700, 345 and 345 records, respectively. However, an additional set of records from 49 cows that completed their lactation was utilized to further verify the prognostic abilities of the ANN. The RMS error calculated based on the training and validation sets was 477 and 502 kg, respectively. The mean milk yield for 305-day lactation predicted by the ANN was 13.12 kg

lower than the real milk yield of the 49 cows used for verification purposes but this difference was statistically non-significant.

The next successful attempt at using ANNs for predicting milk yield in dairy cows was based on daily milk yields recorded between 5 and 305 days in milk [49]. The following predictor variables were used in the ANN model: proportion of Holstein-Friesian genes in cow genotype, age at calving, days in milk and lactation number. The dependent variable was the milk yield on a given day. Predictions made by ANNs were compared with the observed yields and those generated by the SYMLEK system. The data set (137,507 records) was divided into subsets for network training, validation (108,931 records) and testing (28,576). 25 MLPs were built and trained using Statistica ® Neural Networks PL v 4.0F software. MLP with 10 and 6 neurons in the first and second hidden layer, respectively, showed the best performance (RMS error of 3.04 kg) and was selected for further analysis. The correlation coefficients between the real yields and those predicted by the ANN ranged from 0.84 to 0.89 depending on lactation number. The correlation coefficients between the actual cumulative yields and predictions ranged between 0.94 and 0.96 depending on lactation. ANN was more effective in predicting milk yield than the SYMLEK system. The most important variables revealed by the ANN sensitivity analysis were days in milk followed by month of calving and lactation number.

Another study on milk yield prediction involved the use of ANNs to predict milk yield for complete and standard lactations in Polish Holstein-Friesian cows [29]. A total of 108,931 daily milk yield records (set A) for three lactations in cows from a particular barn as well as 38,254 test-day records (set B) for cows from 12 barns located in the West Pomeranian Province in Poland were analyzed. ANNs quality was evaluated with the coefficient of determination (R^2), relative approximation error (*RAE*) and root mean squared error *(RMS*). To verify the prognostic ability of the models, 28,576 daily milk yield records (set A') and 3,249 test-day records (set B') were randomly selected. For the cows for which these records were obtained, the predictions of the daily and lactation milk yields were generated and compared with their real milk yields and those from the official milk recording system SYMLEK. The RMS errors on sets A and B were 2.77 - 3.39 kg and 2.43 – 3.79 kg, respectively, depending on the analyzed lactation. Similarly, the RAE values ranged from 0.13 to 0.15 and from 0.11 to 0.15, whereas the R^2 values were 0.75 – 0.79 and 0.75 – 0.78 for sets A and B, respectively. The correlation coefficients between the actual (or generated by the SYMLEK system) and predicted milk yields calculated on the basis of the test sets were 0.84 - 0.89 and 0.88 – 0.90 for sets A' and B', respectively, depending on lactation. These predictions were closer to the real values than those made by the SYMLEK system. The most important variables in the model determined on the basis of sensitivity analysis were lactation day and calving month as well as lactation day and percentage of Holstein-Friesian genes for the daily milk yield and test-day records, respectively.

5. Model quality

For the evaluation of the classification and regression model quality, the indices described below, calculated on the basis of the training set or combined training and validation sets, are used.

5.1. Classification model quality

The evaluation of the classification model quality is performed using the indices such as: sensitivity, specificity, probability of false positive results *P(FP)*, probability of false negative results *P(FN)* and accuracy. Moreover, the *a posteriori* probability of true positive results *P(PSTP)* and *a posteriori* probability of true negative results *P(PSTN)* are used. All the above-mentioned probabilities are calculated for the two-class classification based on the classification matrix (Table 1).

Predicted class	Actual class		Total
	Positive result	Negative result	
Positive result	A	B	A+B
Negative result	C	D	C+D
Total	A+C	B+D	A+B+C+D

Table 1. The general form of classification matrix

Sensitivity is defined as a percentage of correctly identified individuals belonging to the distinguished class (e.g. individuals with dystocia or conception difficulties):

$$Sensitivity = \frac{A}{A+C}.$$

Specificity is a percentage of correctly recognized individuals belonging to the second (undistinguished) class (e.g. individuals with easy calvings or conception):

$$Specificity = \frac{D}{B+D}.$$

The probability of false negative results *P(FN)* defines the percentage of incorrectly classified individuals belonging to the distinguished class (e.g. indicating dystotic cow as one with an easy calving or cow with conception problems as one without such difficulties):

$$P(FN) = \frac{C}{A+C},$$

whereas the probability of false positive results *P(FP)* corresponds to the proportion of incorrectly recognized individuals belonging to the second analyzed class (e.g. diagnosing cow with an easy calving as a dystotic one or a cow without conception problems as one with such difficulties):

$$P(FP) = \frac{B}{B+D}.$$

The *a posteriori* probabilities make it possible to answer the question about the proportion of individuals assigned by the model to a given class that really belonged to that class. They are calculated according to the following formulae:

$$P(PSTP) = \frac{A}{A+B} \text{ and } P(PSTN) = \frac{D}{C+D}.$$

In the case of some classification models it is also possible to calculate additional quality indices, such as root mean squared error RMS (for ANN and MARS):

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2},$$

where: n – the number of cases, y_i – the real value of the analyzed trait, $\hat{y}_i$ - the value of this trait predicted by a given classification model.

5.2. Regression model quality

For the evaluation of the regression model quality, the following indices are mainly used: Pearson's coefficient of correlation between the actual values and those calculated by the model (r), the ratio of standard deviation of error to the standard deviation of variable (SD_{ratio}), error standard deviation (S_E) and the mean of error moduli ($\overline{E}_{MB}$) [29].

Moreover, the relative approximation error (RAE), adjusted coefficient of determination (R_p^2) and the aforementioned root mean squared error (RMS) are used. The first two indices are calculated according to the following equations [49]:

$$RAE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} y_i^2}} \text{ and } R_p^2 = 1 - \frac{MS_E}{MS_T},$$

where: MS_E – the estimated variance of a model error, MS_T – the estimated variance of the total variability.

In the evaluation of the regression model, special attention should be paid to two of the aforementioned parameters [17]:

1. SD_{ratio} – always takes on non-negative values and its lower value indicates a better model quality. For a very good model SD_{ratio} takes on the values in the range from 0 to 0.1. SD_{ratio} over 1 indicates very poor quality of the model.
2. Pearson's correlation coefficient – takes on the values in the range between 0 and 1. The higher the value of this coefficient, the better the model quality.

6. Prediction quality

For the evaluation of predictions made by the developed classification models, the above-mentioned probabilities calculated for the test set can be used. It is also possible to apply the receiver operating characteristic (ROC) curves, which describe the relationship between

sensitivity and specificity for the models in which dependent variable has only two categories (Fig. 3).

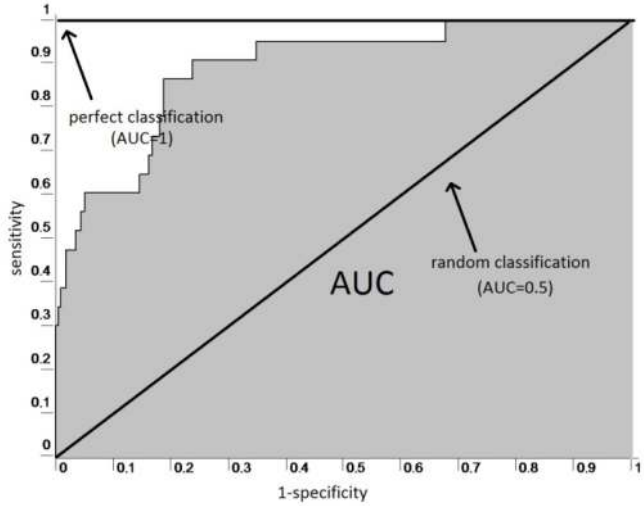

Figure 3. The receiver operating characteristic (ROC) curve and the area under curve (AUC) (from Statistica ® Neural Networks, modified)

The ROC curve is obtained in the following steps. For each value of a predictor, which can be a single variable or model result, a decision rule is created using this value as a cut-off point. Then, for each of the possible cut-off points, sensitivity and specificity are calculated and presented on the plot. In the Cartesian coordinate system, 1-specificity (equal to false positive rate) is plotted on the horizontal axis and sensitivity on the vertical axis. Next, all the points are joined. The larger the number of different values of a given parameter, the smoother the curve [50]. For the equal costs of misclassification, the ideal situation is when the ROC curve rises vertically from (0,0) to (0,1), then horizontally to (1,1). Such a curve represents perfect detection performance on the test set. On the other hand, if the curve is a diagonal line going from (0,0) to (1,1), the predictive ability of the classifier is none, and a better prediction can be obtained simply by chance [51].

The ROC curves are often used to compare the performance of different models, so it would be advantageous to represent the shape of the curve as one parameter. This parameter is called area under curve (AUC) and can be regarded as a measure of goodness-of-fit and accuracy of the model [50, 52]. AUC takes on the values in the range [0,1]. The higher the AUC, the better the model but no realistic classifier should have an AUC less than 0.5 because this corresponds to the random guessing producing the diagonal line between (0,0) and (1,1), which has an area of 0.5 [51].

For the evaluation of predictions made by regression models, the following parameters calculated for the test set can be applied [49]:

1. Pearson's coefficient of correlation between the actual values and those predicted by the model (r)
2. Mean relative prediction error Ψ calculated according to the following formula:

$$\Psi = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \cdot 100\%$$

3. Theil's coefficient I^2 expressed by the following equation [53]:

$$I^2 = \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n} y_i^2}.$$

7. Model comparison

At least two basic criteria can be used for making comparisons between various models. These are: Akaike information criterion (*AIC*) and Bayesian information criterion (*BIC*). *AIC* can be defined as:

$$AIC = -2\ln L_{\max} + 2k,$$

where L_{max} is the maximum likelihood achievable by the model, and k is the number of free parameters in the model [54]. The term k in the above equation plays a role of the "penalty" for the inclusion of new variables in the model and serves as compensation for the obviously decreasing model deviation. The model with a minimum *AIC* is selected as the best model to fit the data [30].

Bayesian information criterion (*BIC*) is defined as [54]:

$$BIC = -2\ln L_{\max} + k\ln n,$$

where n – the number of observations (data points) used in the fit. Both criteria are used to select a "good model" but their definition of this model differs. Bayesian approach, reflected in the *BIC* formulation, aims at finding the model with the highest probabilities of being the true model for a given data set, with an assumption that one of the considered models is true. On the other hand, the approach associated with *AIC* uses the expected prediction of future data as the most important criterion of the model adequacy, denying the existence of any true model [55].

8. Summary

Data mining methods can be an economic stimulus for discovering unknown rules or associations in the object domains. No knowledge will be discovered without potential and significant economic benefits. Much acquired knowledge can be used for improving

currently functioning models. These methods are capable of finding certain patterns that are rather inaccessible for conventional statistical techniques. These techniques are usually used for the verification of specific hypotheses, whereas the application of data mining methods is associated with impossibility of formulating preliminary hypotheses and the associations within data are often unexpected. Discoveries or results obtained for individual models should be an introduction to further analyzes forming the appropriate picture of the problem being explored.

Author details

Wilhelm Grzesiak* and Daniel Zaborski
Laboratory of Biostatistics, Department of Ruminant Science, West Pomeranian University of Technology, Szczecin, Poland

9. References

[1] Friedman J H (1991) Multivariate adaptive regression splines (with discussion). Annals of Statistics 19: 1-141.
[2] Zakeri I F, Adolph A L, Puyau M R, Vohra F A, Butte N F (2010) Multivariate adaptive regression splines models for the prediction of energy expenditure in children and adolescents. Journal of Applied Physiology 108: 128–136.
[3] Taylan P, Weber G-H, Yerlikaya F (2008) Continuous optimization applied in MARS for modern applications in finance, science and technology. 20th EURO Mini Conference "Continuous Optimization and Knowledge-Based Technologies" (EurOPT-2008), May 20–23, 2008, Neringa, Lithuania, pp. 317-322.
[4] StatSoft Electronic Statistics Textbook. http://www.statsoft.com/textbook/ (last accessed 14.04.2012)
[5] Xu Q-S, Massart D L, Liang Y-Z, Fang K-T (2003) Two-step multivariate adaptive regression splines for modeling a quantitative relationship between gas chromatography retention indices and molecular descriptors. Journal of Chromatography A 998: 155–167.
[6] Put R, Xu Q S, Massart D L, Vander Heyden Y (2004) Multivariate adaptive regression splines (MARS) in chromatographic quantitative structure-retention relationship studies. Journal of Chromatography A 1055: 11-19.
[7] Lee T-S, Chiu C-C, Chou Y-C, Lu C-J (2006) Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. Computational Statistics and Data Analysis 50: 1113-1130.
[8] Zareipour H, Bhattacharya K, Canizares C A (2006) Forecasting the hourly Ontario energy price by multivariate adaptive regression splines. IEEE, Power Engineering Society General Meeting, pp. 1-7.

* Corresponding Author

[9] Sokołowski A, Pasztyła A (2004) Data mining in forecasting the requirement for energy carriers. StatSoft Poland, Kraków, pp. 91 – 102 [in Polish]
[10] Hastie T, Tibshirani R, Friedman J (2006) The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York, p. 328.
[11] Glick M, Klon A E, Acklin P, Davies J W (2004) Enrichment of extremely noisy high-throughput screening data using a naïve Bayes classifier. Journal of Molecular Screening 9: 32-36.
[12] Lewis D D (1998) Naïve (Bayes) at forty: The independence assumption in information retrieval. Machine Learning ECML-98. Lecture Notes in Computer Science 1398/1998: 4-15.
[13] Rish I (2001) An empirical study on the naïve Bayes classifier. The IJCAI-01 Workshop on empirical methods in artificial intelligence. August 4, 2001, Seattle, USA, pp. 41-46.
[14] Morzy M (2006) Data mining – review of methods and application domains. In: 6th Edition: Data Warehouse and Business Intelligence, CPI, Warsaw, pp. 1–10 [in Polish].
[15] Samarasinghe S (2007) Neural Networks for Applied Science and Engineering. From Fundamentals to Complex Pattern Recognition. Auerbach Neural Publications, Boca Raton, New York, pp. 2, 75, 97, 254, 311.
[16] Tadeusiewicz R (1993) Neural Networks. AOW, Warsaw, pp. 8, 19, 28, 49, 55, 56-57,59-61 [in Polish],
[17] Tadeusiewicz R, Lula P (2007) Neural Networks. StatSoft Poland, Kraków, pp. 8-20,35 [in Polish].
[18] Tadeusiewicz R, Gąciarz T, Borowik B, Leper B (2007) Discovering the Properties of Neural Networks Using C# Programs. PAU, Kraków, pp. 55, 70-72, 91-92,101 [in Polish].
[19] Tadeusiewicz R. 2000. Introduction to neural networks. In: Duch W, Korbicz J, Rutkowski L, Tadeusiewicz R (Eds.) Neural Networks, AOW Exit, Warsaw, p. 15 [in Polish] .
[20] Bishop C M (2005) Neural Networks for Pattern Recognition. Oxford University Press, Cambridge, pp. 78, 80, 82, 116, 122, 141, 165, 233, 263.
[21] Haykin S (2009) Neural Networks and Learning Machines. (3rd ed.), Pearson, Upper Saddle River, pp. 41,43-44,154,197,267.
[22] Cheng B, Titterington D M (1994) Neural networks: A review from a statistical perspective. Statistical Science 9: 2-54.
[23] Boniecki P (2008) Elements of Neural Modeling in Agriculture. University of Life Sciences in Poznań, Poznań, pp. 38, 93-96 [in Polish].
[24] Osowski S (1996) Algorithmic Approach to Neural Networks. WNT, Warsaw [in Polish].
[25] Witkowska D (2002) Artificial Neural Networks and Statistical Methods. Selected Financial Issues. C.H. Beck, Warsaw, pp. 10,11 [in Polish].
[26] Rutkowski R (2006) Artificial Intelligence Methods and Techniques. PWN, Warsaw, pp. 179-180,220,222-223 [in Polish].
[27] Larose D T (2006) Discovering Knowledge in Data. PWN, Warsaw, pp. 111-118,132,144 [in Polish].

[28] Rumelhart D E, Hinton G E, Williams R J (1986) Learning representations by back-propagating errors. Nature 323: 533-536.
[29] Grzesiak W (2004) Prediction of dairy cow milk yield based on selected regression models and artificial neural networks. Post-doctoral thesis. Agricultural University of Szczecin, Szczecin, pp. 37, 49-70 [in Polish].
[30] Koronacki J, Ćwik J (2005) Statistical Learning Systems. WNT, Warsaw, pp. 59, 122-123 [in Polish].
[31] Breiman L, Friedman J, Olshen L, Stone C (1984) Classification and Regression Trees, Chapman and Hall/CRC Press, Boca Raton
[32] Steinberg D (2009) Classification and Regression Trees. In: Wu X., Kumar V. (Eds.) The Top Ten Algorithms in Data Mining. Chapman and Hall/CRC Press, Boca Raton, London, New York, pp. 179-202.
[33] Breiman L (1996) Technical note: Some properties of splitting criteria. Machine Learning 24: 41-47.
[34] Dorynek Z (2005) Reproduction in cattle. In: Litwińczuk Z, Szulc T (Eds.) Breeding and Utilization of Cattle. PWRiL, Warsaw, p. 198 [in Polish].
[35] Grzesiak W, Zaborski D, Sablik P, Żukiewicz A, Dybus A, Szatkowska I (2010) Detection of cows with insemination problems using selected classification models. Computers and Electronics in Agriculture 74: 265-273.
[36] Grzesiak W, Zaborski D, Sablik P, Pilarczyk R (2011) Detection of difficult conceptions in dairy cows using selected data mining methods. Animal Science Papers and Reports 29: 293-302.
[37] Zaborski D, Grzesiak W (2011) Detection of heifers with dystocia using artificial neural networks with regard to *ERα-Bgl*I, *ERα-Sna*BI and *CYP19-Pvu*II genotypes. Acta Scientiarum Polonorum s. Zootechnica 10: 105-116.
[38] Zaborski D, Grzesiak W, Szatkowska I, Dybus A, Muszyńska M, Jędrzejczak M (2009) Factors affecting dystocia in cattle. Reproduction in Domestic Animals 44: 540- 551.
[39] Mee J F, Berry D P, Cromie A R (2009) Risk factors for calving assistance and dystocia in pasture-based Holstein–Friesian heifers and cows in Ireland. The Veterinary Journal 187: 189-194.
[40] Johanson J M, Berger P J, Tsuruta S, Misztal I (2011) A Bayesian threshold-linear model evaluation of perinatal mortality, dystocia, birth weight, and gestation length in a Holstein herd. Journal of Dairy Science 94: 450–460.
[41] Meijering A (1984) Dystocia and stillbirth in cattle – a review of causes, relations and implications. Livestock Production Science 11: 143-177.
[42] Zaborski D (2010) Dystocia detection in cows using neural classifier. Doctoral thesis. West Pomeranian University of Technology, Szczecin, pp. 5-21 [in Polish].
[43] Zaborski D, Grzesiak W (2011) Detection of difficult calvings in dairy cows using neural classifier. Archiv Tierzucht 54: 477-489.
[44] Park B, Lee D (2006) Prediction of future milk yield with random regression model using test-day records in Holstein cows. Asian- Australian Journal of Animal Science 19: 915-921.

[45] Grzesiak W, Wójcik J, Binerowska B (2003) Prediction of 305-day first lactation milk yield in cows with selected regression models. Archiv Tierzucht 3: 215-226.
[46] Sharma A K, Sharma R K, Kasana H S (2006) Empirical comparisons of feed-forward connectionist and conventional regression models for prediction of first lactation 305-day milk yield in Karan Fries dairy cows. Neural Computing and Applications 15: 359–365.
[47] Grzesiak W (2003) Milk yield prediction in cows with artificial neural network. Prace i Materiały Zootechniczne. Monografie i Rozprawy No. 61: 71-89 [in Polish].
[48] Grzesiak W, Lacroix R, Wójcik J, Blaszczyk P (2003) A comparison of neural network and multiple regression prediction for 305-day lactation yield using partial lactation records. Canadian Journal of Animal Science 83: 307-310.
[49] Grzesiak W, Błaszczyk P, Lacroix R (2006) Methods of predicting milk yield in dairy cows – Predictive capabilities of Wood's lactation curve and artificial neural networks (ANNs). Computers and Electronics in Agriculture 54: 69-83.
[50] Harańczyk G (2010) The ROC curves – evaluation of the classifier quality and searching for the optimum cut-off point. StatSoft Poland, Kraków, pp. 79-89 [in Polish].
[51] Fawcett T (2004) ROC Graphs: Notes and Practical Considerations for Researchers. Technical Report HPL-2003-4. HP Labs, Palo Alto, CA, USA. http://www.hpl.hp.com/techreports/2003/HPL-2003-4.pdf (last accessed 14.04.2012)
[52] Bradley A P (1997) The use of the area under the ROC curve in the evaluation of the machine learning algorithms. Pattern Recognition 30: 1145-1159.
[53] Theil H (1979) World income inequality. Economic Letters 2: 99-102.
[54] Liddle A R (2007) Information criteria for astrophysical model selection. Monthly Notices of the Royal Astronomical Society: Letters 377: L74-L78.
[55] Kuha J (2004) AIC and BIC. Comparisons of assumptions and performance. Sociological Methods and Research 33: 188-229.