AGRONOMY SCIENCE AND BIOTECHNOLOGY

REVIEW ARTICLE

Prediction and importance of predictors in approaches based on computational intelligence and machine learning

Antônio Carlos da Silva Júnior[1,*], Waldênia de Melo Moura[2], Leonardo Lopes Bhering[1], Michele Jorge da Silva Siqueira[1], Weverton Gomes da Costa[1], Moysés Nascimento[3] and Cosme Damião Cruz[1]

[1]Department of General Biology, Federal University of Viçosa, Viçosa, Avenida Peter Henry Rolfs, s/n, Viçosa, MG, Brazil, CEP 36570-000. [2]Empresa de Pesquisa Agropecuária de Minas Gerais (EPAMIG), Av. José Candido da Silveira, 1647, Belo Horizonte, MG, Brazil, CEP 31170-495. [3]Department of Statistic, Federal University of Viçosa, Bioagro, Computational Intelligence Laboratory and Statistical Learning (LICAE), Avenida Peter Henry Rolfs, s/n, Viçosa, MG, Brazil, CEP 36570-000. *Corresponding author, E-mail: antonio.silva.c.junior@gmail.com

Check for updates

OPEN ACCESS

How to cite: Silva Júnior, A. C., Moura, W. M., Bhering, L. L., Siqueira, M. J. S., Costa, W. G., Nascimento, M., & Cruz, C. D. (2023). Prediction and importance of predictors in approaches based on computational intelligence and machine learning *Agronomy Science and Biotechnology*, 9, 1-24 https://doi.org/10.33158/ASB.r179.v9.2023

Received: December 3rd, 2022.
Accepted: January 11, 2023.
Published: March 08, 2023.

English by: Antônio Carlos da Silva Júnior

Copyright: © 2023 Agronomy Science and Biotechnology. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, since the original author and source are credited.

ABSTRACT

Machine learning and computational intelligence are rapidly emerging in plant breeding, allowing the exploration of big data concepts and predicting the importance of predictors. In this context, the main challenges are how to analyze datasets and extract new knowledge at all levels of research. Predicting the importance of variables in genetic improvement programs allows for faster progress, carrying out an extensive phenotypic evaluation of the germplasm, and selecting and predicting traits that present low heritability and/or measurement difficulties. Although, simultaneous evaluation of traits provides a wide variety of information, identifying which predictor variable is most important is a challenge for the breeder. The traditional approach to variable selection is based on multiple linear regression. It evaluates the relationship between a response variable and two or more independent variables. However, this approach has limitations regarding its ability to analyze high-dimensional data and not capture complex and multivariate relationships between traits. In summary, machine learning and computational intelligence approaches allow inferences about complex interactions in plant breeding. Given this, a systematic review to disentangle machine learning and computational intelligence approaches is relevant to breeders and was considered in this review. We present the main steps for developing each strategy (from data selection to evaluating classification/prediction models and quantifying the best predictor).

Keywords: Plant breeding, big data, artificial neural networks, decision tree, bagging, random forest, boosting.

INTRODUCTION

Plant breeding is an effective way to increase crop productivity. Its objective is the development of high-yielding varieties with specific grain qualities, resistance to abiotic and biotic stresses, and superior adaptation to the target environment (Yu et al., 2019; Silva Junior et al., 2021).

Quantifying the importance of variables in genetic improvement programs allows for faster progress, carrying out an extensive phenotypic evaluation of the germplasm, and selecting and predicting traits that present low heritability and/or measurement difficulties. Although, simultaneous evaluation of traits provides a wide variety of information, identifying which predictor variable is most important is a challenge for the breeder. The traditional approach to variable selection is based on multiple linear regression. It evaluates the relationship between a response variable and two or more independent variables (Skawsang et al., 2019). However, this approach has limitations regarding its ability to analyze high-dimensional data, in addition to not capturing complex and multivariate relationships between variables (Paswan, 2013; Parmley et al., 2019; Skawsang et al., 2019).

The application of computational intelligence can be an alternative for the selection of variables and has been used in prediction studies (Ventura et al., 2012; Silva et al., 2014, 2017; Sant'Anna et al., 2019), classification (Sant'Anna et al., 2015), pattern recognition (Beucher et al., 2019), in the decision-making process (Carneiro et al., 2018, 2019; Silva Junior et al., 2021) and also for minimizing the number of predictors without compromising model performance (Parmley et al., 2019). An alternative is machine learning, which is efficient for exploring large sets of data and contrasting information, in addition to identifying predictor variables of better performance (Parmley et al., 2019; Silva Junior et al., 2021).

Artificial neural networks (ANNs) are highly parameterized non-linear models, with sets of processing units called neurons, which can be used to approximate the relationship between the input and output signals of a complex system (Stefaniak et al., 2005). ANNs are powerful prediction tools compared to conventional models such as linear regression (Paruelo & Tomasel, 1997; Olden et al., 2002; Beck, 2018). In addition, they reproduce the importance of each predictor, making it easily interpretable (Zhang et al., 2018). However, the importance of traits during network tuning is often overlooked.

The quantification of the importance of variables through the Multilayer Perception Network (PMC) can be obtained through (i) Garson's algorithm (1991) modified by Goh (1995), which consists of partitioning the neural network connection weights to determine the importance relative value of each input variable in the network. This algorithm describes the relative magnitude of the importance of the descriptors (predictor) through the dissection of the synaptic weights of the neural network. (ii) Evaluation of the importance of variables (input) through the impact of destructuring or disturbing the information of a given input on the estimate of the coefficient of determination.

Radial Base Networks (RBF) also aim to estimate the importance of predictors using the technique of destructuring the coefficient of determination (Santos et al., 2018; Yadav et al., 2018; Beucher et al., 2019; Silva Junior et al., 2021). The RBF network, compared to other neural networks, has a simpler structure and a faster learning algorithm (Sreekanth et al., 2010; Basheer & Hajmeer, 2000). This network consists of three layers, namely, the input layer, the hidden layer, and the output layer (Silva Junior et al., 2021).

Other interesting alternatives for studies of prediction and importance of variables are methodologies based on machine learning, such as decision trees (Beucher et al., 2019; Parmley et al., 2019) and their refinements, such as bagging, random forest, and boosting (Degenhardt et al., 2019; Silva Junior et al., 2021). Such methodologies allow for obtaining good predictions and the importance of characteristics through measures based on, for example, the Gini and Entropy index (Hastie, 2009). These methodologies allow the quantification of the impact of destructuring or disturbing the information of a given input on the estimation of the coefficient of determination.

Methodologies based on regression, artificial intelligence, and machine learning have been successfully used in prediction studies. Parmley et al. (2019) evaluated high-dimensional phenotypic traits in soybeans using a machine-learning approach to predict seed yield in terms of the prescriptive development of cultivars for agricultural practices. Skawsang et al. (2019) applied such methodologies to predict the population of insect pests using climatic and phenological factors of the host plant. Silva Junior et al. (2023) used it to predict productivity and verify the importance of variables for grain yield in rice.

COMPUTATIONAL INTELLIGENCE IN PLANT BREEDING

Computational intelligence is the area of computer science that aims to simulate, in machines, the ability to solve problems and perform tasks, which are a skill of man's natural intelligence (Norvig & Russell, 2013). Currently, computational intelligence has been applied in the areas of autonomous planning, games, language recognition, and problem-solving (Fernandes, 2003).

In plant breeding, the application of computational intelligence has been used in the selection of traits (Silva Junior et al., 2022; Silva Junior et al., 2023), prediction (Ventura et al., 2012; Silva et al., 2014, 2017; Sant'Anna et al., 2019), classification (Sant'Anna et al., 2015), pattern recognition (Beucher et al., 2019), and in the decision-making process (Carneiro et al., 2018, 2019; Silva Junior et al., 2021). Among the various techniques of computational intelligence, there are artificial neural networks - ANN, which are tools with great potential for application in the genetic improvement of plants.

Artificial neural networks

Artificial Neural Networks (ANN) work conceptually similarly to the human brain, trying to recognize regularities and data patterns, and can learn from experience and make generalizations based on their previously accumulated knowledge (Cruz & Nascimento, 2018). Because ANNs are able general problems, such as approximation, classification, categorization, and prediction (Braga et al., 2007), and are tolerant to missing and non-linear data, which represents progress for statistical studies and genetic improvement (Cruz & Nascimento, 2018).

Multilayer perceptron

The Multilayer Perceptron networks, which present a structure of neural networks characterized by the existence of hidden (or intermediate) layers, are based, in their learning processes, on the backpropagation training algorithm, that is, multiple layer networks (Braga et al., 2007; Haykin, 2001; Cruz & Nascimento, 2018). This new model was called Multilayer Perceptron (MLP), as shown in Figure 1.

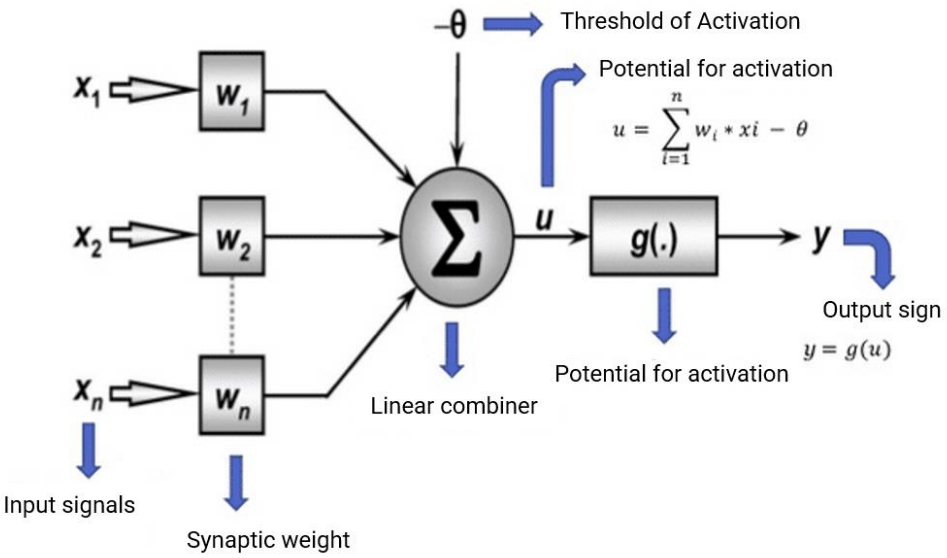

Figure 1. Nonlinear model of an artificial neuron, in which X_1, X_2,..., X_n are the network inputs; W_1, W_2,..., We are the weights, or synaptic weights, associated with each input; -Θ is the activation threshold (bias); μ is the linear combination of the input signals; g (.) is the activation function, and y is the output of the network.

Figure 2 presents a single-layer perceptron neural network, in which m and n represent the numbers of inputs and the numbers of neurons, respectively, is connected to all neurons in only one layer and yi is the output of each neuron

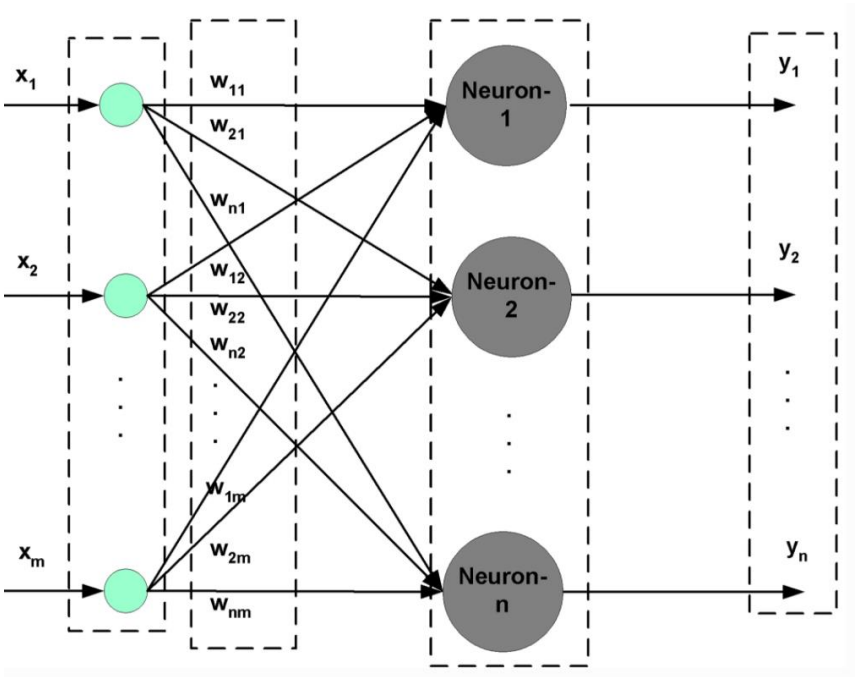

Figure 2. Structure of the Perceptron Neural Network. The number of outputs is the same as the number of intermediate neurons (n). The input signals (x_1, x_2... x_m) are

connected to single-layer neurons through their corresponding weights **(**w_{11}**,** w_{12} **…** w_{nm}**).**

Input signals (x_1, x_2… x_m) are connected to single-layer neurons through their corresponding weights (w_{11}, w_{12} ... w_{nm}), so that each neuron has a corresponding weight value for each input signal, respectively.

The input signal X and the weights can be expressed in a vector form:

$$\begin{pmatrix} x_1 \\ x_2 \\ : \\ x_m \end{pmatrix} e \begin{pmatrix} w_{11} & w_{12} & w_{1m} \\ w_{21} & w_{22} & w_{2m} \\ : & : & : \\ w_{n1} & w_{n2} & w_{nm} \end{pmatrix},$$

Where n is the number of neurons and m is the number of inputs.

Activation function

The activation function embedded in a network structure gives each neuron the ability to extract non-linear information and the learning potential of the network. Its role is to determine the form and intensity of alteration of the values transmitted from one neuron to another. Therefore, it is important to choose the correct activation function for better network performance. These functions can be classified by the activation functions used by the neurons: in a homogeneous network, in which the activation functions of the neurons in the network are the same, and in a heterogeneous network, in which the neurons in the network use different activation functions.

The activation functions widely used in neural network architectures are summarized in Table 1. Among them, the most used activation functions are linear and sigmoid functions.

Table 1. Main activation functions used in Artificial Neural Networks.

Function	Formula	Interval
linear	$f(x) = x$	(-∞, ∞)
semi-linear	$f(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases}$	(0, ∞)
logistics (sigmoidal)	$f(x) = \frac{1}{1 + e^{-ax}}$	(0,1)
hyperbolic tangent	$f(x) = \tanh\left(\frac{x}{2}\right) = \frac{1 + e^{-x}}{1 + e^{x}}$	(-1,1)
gaussian	$f(x) = e^{-x^2}$	(0,1)
exponential	$f(x) = e^{-x}$	(0, ∞)

Layers

In PMC networks, three layers are identified: input, output, and intermediate (or hidden), each of which has a specific function. The input layer corresponds to the information available to be presented to network structures for training purposes. If there are the right connections between the input layers and a sufficiently large set of intermediate layers, one can always find the representation that will produce the correct mapping from the input layer to the output layer, making use of tweaks. weights in the middle layers.

The intermediate layer works as an extractor of characteristics contained in the set of data that are presented. Their weights translate the importance of an aggregate of features extracted from the input patterns and allow the network to create its own, richer and more complex representation of the problem. The output layer receives stimuli from the intermediate layers and builds the pattern that will be the response.

Regarding the number of neurons in the intermediate layers, there is no consensus. Generally, the best number of neurons for solving a problem in the agricultural area is defined empirically, that is, a certain architecture is established with a fixed number of layers and the number of neurons is varied, in increasing ways, until finding a suitable one optimal solution (Cruz & Nascimento, 2018).

When it comes to establishing the number of neurons per layer, care must be taken not to use too many units, which can lead to serious problems in the training phase. In this case, the network, instead of learning, memorizes the available data (overfitting), memorizing the specific input and output pattern, which generally includes a true value plus a certain amount of noise. On the other hand, the use of a very small number of neurons per layer may require the network to spend too much time trying to find an optimal representation and, as in the previous case, provide low effectiveness in validation with the solution found in training.

Network architecture

The network architecture can be described as (numbers of neurons in the input layer - middle layer - output layer); eg (5-8-1) structure means 5 neurons in the input layer, and 8 hidden neurons in the middle layer. Considering a structure with 5-8-8-1; means two hidden layers with 8 hidden neurons, respectively, and 1 neuron in the output layer.

Radial Base Function Network - RBF

The radial basis function network consists of a simpler structure than the multilayer networks since it constitutes an input layer, only an intermediate layer, and a neuron output, which is fed forward, a procedure known as feedforward. With only one intermediate layer in the network, it is already possible to calculate any arbitrary function from given data (Hecht-Nielsen, 1989).

Regarding the RBF topology, which is composed of a radial basis activation function in its intermediate layer. Generally, these functions return smaller and smaller values as the distance between the observed point and the center of the function increases. Among the activation functions available for RBF networks, the multi-quadratic, inverse multi-quadratic, and gaussian functions stand out. The output layer generally adopts linear functions (Park & Sandberg, 1991).

The description of the gaussian activation function can be found in Eq.

$$g(u) = e^{\frac{-(u-c)^2}{2\sigma^2}}$$

where c is the center of the gaussian function, σ^2 is the variance of the gaussian function and u is the activation potential.

The linear activation function is defined in Eq.

$$y_{ri} = g\left(x_0 w_0 + \sum_{j=1}^{q} f_{xj}(x_i) w_j\right)$$

where x_i is the i^{th} entry; w_j is the synaptic weight; f_{xj} is the hidden layer activation function associated with input x_i weighted by its respective weight

In Figure 3, the scheme of the RBF feedforward network is presented. The entries X_1 to X_n in the input layer refer to phenotypic traits. A hidden layer with a radius ranging from 1 to r and a number of neurons ranging from 1 to n. On output, the network returns the vector of predicted values (y).

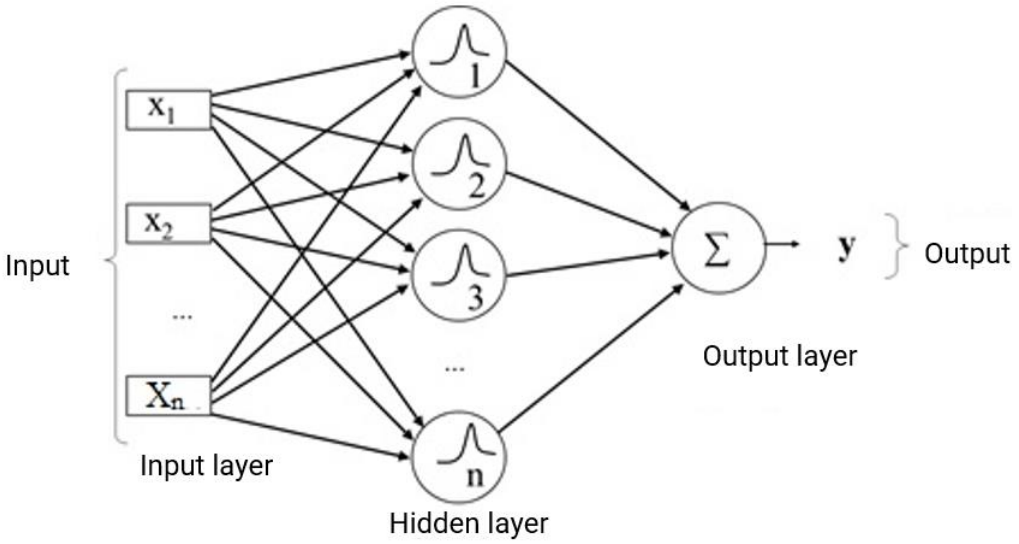

Figure 3. Scheme of the feedforward RBF network. Inputs X_1 to X_n in the phenotypic trait input layer. A hidden layer with a radius ranging from 1 to r and several neurons ranging from 1 to n. On output, the network returns the vector of predicted values (y).

MACHINE LEARNING IN PLANT BREEDING

Machine learning, also known as statistical learning, is a subfield of artificial intelligence dedicated to the study of algorithms for prediction and inference (Morota et al., 2018; Liakos et al., 2018). However, in practice, machine learning aims to learn or choose a set of models that can best predict unobserved data. The process of predicting the phenotype from a set of genotypes in which we have a dataset composed of pairs of phenotypes and corresponding genotypes is known as supervised learning.

For the choice of the model with good predictive capacity in supervised learning, we started by dividing the data set into two sets, training, and testing, where the latter is related to the unavailable data set (Morota et al., 2018). However, model selection uses information exclusively from the training dataset.

Decision tree

The decision tree (DT) is a methodology that partitions the predictor space into sub-regions through some criteria, for each formed sub-region a value is assigned that will be used as a predicted value for the new individuals that will be allocated to these sub-regions regions. The DT structure is composed of internal nodes, branches, and external/leaf nodes. The node is said to be internal when the data contained in this node are divided according to a division criterion, thus forming two new groups of data, these new groups being linked to the old group by the branches, the node is said to be external (leaf). When there are no further divisions of individuals belonging to this node (Figure 4 and Figure 5).

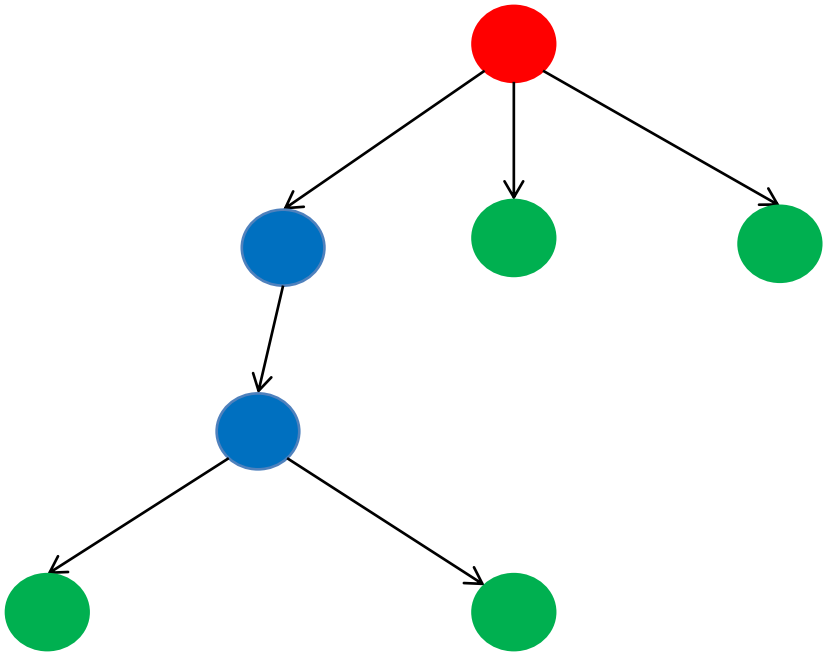

Figure 4. Decision tree, with the internal node (red), branches (blue), and leaves (green).

The DT can be classified as a regression tree when the response variable is of the quantitative type (continuous distribution), whereas when the dependent variable assumes qualitative values (discrete distribution), the DT is classified as a classification tree (Figure 5).

Regression trees

The construction of the regression tree aims to build regions R_1, R_2,....., R_M, in which it minimizes the Sum of Squares of the Residues, described below:

$$\sum_{m=1}^{M} \sum_{i \epsilon R_m} (y_i - \hat{y}_{R_m})^2,$$

Where, $\hat{y}_{R_m}$: average of the response variable of the training observations belonging to the mth region.

The computational cost is very high and it is not feasible to consider each possible partition of space in M regions to obtain the smallest mean squared error.

To get around the computational cost, (James et al., 2013) recommend a procedure based on recursive binary divisions, in which the objective is to obtain the variable X_p and the point s, which divide the space into two regions, such as:

$$R_1(p,s) = \{X|X_p \leq s\} \; e \; R_2(p,s) = \{X|X_p > s\}$$

where the point s divides the p[th] variable into two regions that obtain the smallest mean squared error, finally we use the variable that obtained the smallest mean squared error for the first division, then we repeat the process for each generated region.

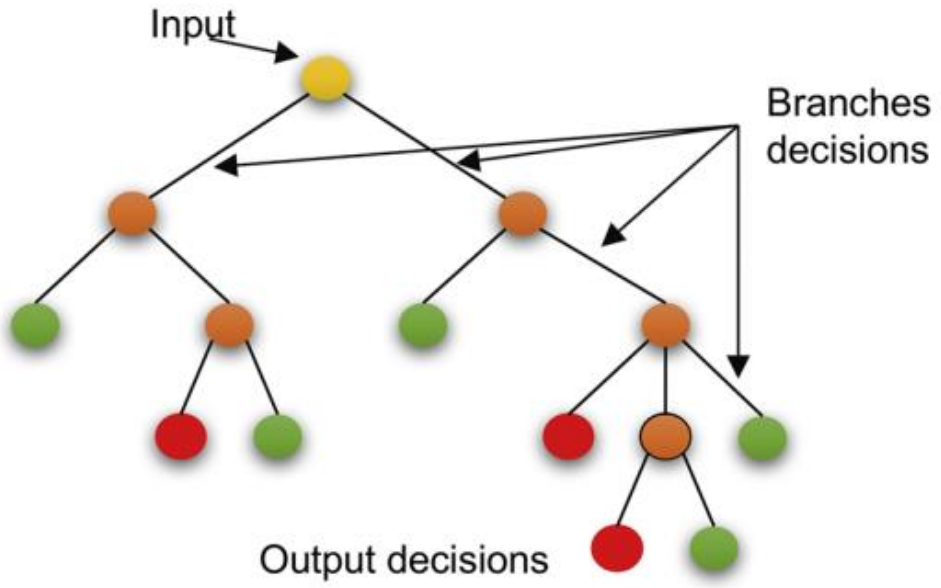

Figure 5. Decision tree. Red dots show instances of negative classes, while green dots indicate positive classes. The yellow dot designates a new data instance to be sorted.

When decision trees are built, many of the edges or subtrees can reflect noise or errors. While a very large tree may over fit the data, a small tree may not capture a good structure. To detect and exclude these edges and sub-trees, tree pruning methods are used, whose objective is to improve the model's success rate for new examples, which were not used in the training set (Han, 2001).

One approach to choosing tree size would be to build a tree until no region gets more than 5 individuals and then prune it using the pruning complexity cost (Hastie et al., 2009). Thus, in the second step, pruning is performed to make the regression tree smaller and less complex, to reduce the variance of this estimator. In this step, each node is removed, one at a time, observing how the prediction error varies in the validation set, and, later, based on the observations, it is decided which nodes remained in the tree (Hastie et al., 2009).

Generally, a single tree does not have good predictive accuracy when compared to other approaches (Sousa et al., 2020). Some refinements to improve the performance of the decision tree model are presented in the literature. The worst performance of the decision tree when compared to its refinements can be explained because this methodology suffers from high variation in terms of prediction (James et al., 2013). Hastie et al. (2009) emphasized that the low predictive accuracy of the

decision tree can be improved by using methods such as bootstrap aggregation (Bagging), random forest, and boosting (Breiman, 2001). These strategies combine multiple decision trees to reduce variability (Sousa et al., 2020).

Classification trees

The classification tree aims to obtain regions R_1, R_2,....., R_M that minimize one of the 3 criteria presented below (James et al., 2013):

- Apparent Error Rate:

$$TEA = 1 - MAX_k(\hat{p}_{mk})$$

- Gini Index:

$$G = \sum_{k=1}^{k} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

- Deviance:

$$D = -\sum_{k=1}^{k} \hat{p}_{mk} log \hat{p}_{mk}$$

where, $\hat{p}_{mk}$: represents the proportion of observations in the mth region belonging to the kth class.

Regarding the construction of the classification tree, it is indicated to use the Gini or Deviance index, since these are more sensitive to analyzing the purity of the node. The indices decrease according to the growth of the tree that occurs through recursive binary division (Sousa et al., 2020). To avoid overfitting the model, it is recommended that no region obtain more than 5 individuals and then prune it using any of the criteria as a guide to the cost and complexity of pruning (Hastie et al., 2009).

The ANNs despite presenting satisfactory efficiency demand a lot of computational resources. On the other hand, decision trees and their refinements (boosting, bagging, random forest) require fewer computational resources (James et al., 2013; Sousa et al., 2020). Furthermore, like ANNs, decision trees, and their refinements do not require assumptions about the model (Sousa et al., 2020; Silva Junior et al., 2021). However, such methodologies present good predictive performance (James et al., 2013), allowing the non-linearity of the data, and are also easy to interpret (Prasad et al., 2006), as they provide information about which attributes are most important for prediction or classification (Ebrahimi et al., 2011; Beiki et al., 2012; Hosseinzadeh et al., 2012).

DECISION TREE REFINEMENT

Bagging

The problem presented by the decision tree is the high variability between the results obtained since we use part of the database to build a tree and then use the other part of the same database to build the second tree. Its construction obtains two trees with different structures. One way around this problem is to obtain several samples from the same population, build several trees and then obtain the mean/mode of the predicted values.

Obtaining multiple training sets from a population is not an easy task. An alternative is the use of bootstrap aggregation (bagging) (Breiman, 2001) is a method

that applies the bootstrap technique. This technique consists of obtaining B samples with replacement of the available sampling, thus obtaining B models $\hat{f}^1(x), \hat{f}^2(x), \ldots, \hat{f}^B(x)$ (Efron, 1992). Sampling is done by replacing the original data and forming new data sets. New datasets can have a fraction of the columns and rows that are often hyperparameters in a model. Thus, the generated models are used to obtain an average, and to reduce the variability obtained in the decision trees (Breiman, 2001). This average of these models will be the final model and is given by:

$$\hat{f}_{médio}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^b(x)$$

The bagging technique consists of reducing the forecast variance, which combines the result of several classifiers, modeled on different sub-samples of the same data set (Breiman, 2001). In this way, it provides row and column fractions smaller than 1, which helps in the assembly of robust models, less prone to overfitting. The number of trees used in bagging is not a parameter that will result in overfitting of the model, in practice a number where the error has stabilized is used (James et al., 2013).

Random forest

From the point of view of using all the variables in each partition in the bagging, the predictions obtained in the decision trees will be highly correlated, since the created trees will have similar structures. In addition, it is almost always subject to the same variable being at the top of the tree (Hastie et al., 2009; James et al., 2013). The average of highly correlated values does not result in a large reduction in variance, as occurs when it is done with uncorrelated values (James et al., 2013). However, by improving the accuracy estimate in the classification of individuals, Ho (1995) proposed the random forest (RF) (Figure 6). Random forest is a versatile machine learning method capable of performing regression and classification (CART) tasks. This methodology also applies dimensional reduction methods and treats missing values, anomalous values ('outliers'), and other essential steps of data exploration. It is a type of learning method where a group of weak models is combined to form a stronger model (Ho, 1995).

The random forest presents the same bagging principle since the data set changes the observations and the number of predictive variables (m<p) used in each partition. Therefore, random forest obtains the most independent predicted values, since it reduces the variability found in decision trees. Hastie et al. (2009) suggest that the number of predictor variables used in each partition be $m = \sqrt{p}$ for classification tree and $m = p/3$ for regression trees. Thus, the predictions of the trees become less correlated and even correct the fact that only one variable is always at the top of the tree.

The modeling process of the RF algorithm is as follows: adopt a bootstrap sampling technique to extract information from the training data about N estimators of the original dataset. The training set is about 2/3 the size of the original dataset. The random forest bootstrap sample during the training process will have about 1/3 of the data not extracted. This part of the data is called out-of-the-bag data. The next step is to create a regression tree for each bootstrap training set.

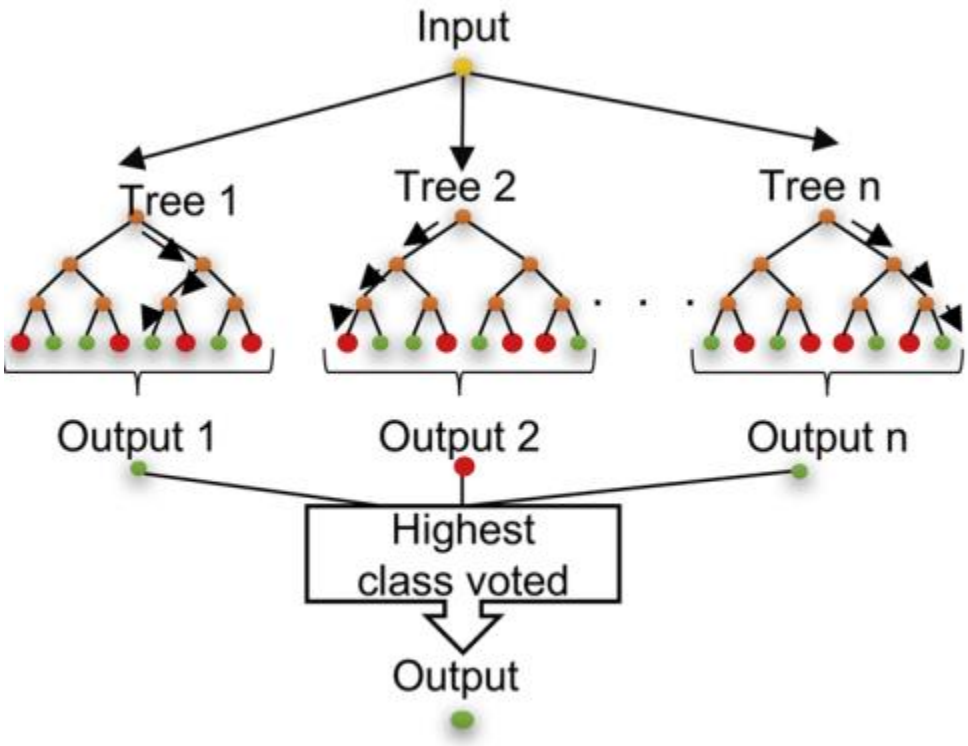

Figure 6. Random Forest. The yellow dot represents the input data. Following several different random decisions, trees are built. The orange circle shows the intermediate decision criterion. The red circle represents the incorrect classification. The green circle indicates the correct classification (Silva et al., 2019).

A total of N estimator regression trees are built to form a "forest", but these regression trees are not pruned. In the process of growing each tree, all ideal attributes are not selected as the inner node for branching, alternatively, the ideal attribute is selected from the randomly selected Max depth attributes for branching. Thus, the RF algorithm increases the difference between the regression models by building different training sets, thus improving the extrapolation prediction capability of the combined regression model. Through n-time model training, a regression model sequence is obtained, which is used to form a multi-regression model system (forest). And then, collecting the prediction results from the regression tree of the N estimators, adopting a simple average strategy to calculate the value of the new sample. The final regression decision formula is as follows: where represents the combined regression model, is a single decision tree regression model, K is the number of regression trees (N estimators). The random modeling process of the forest algorithm is shown in Figure 7.

$$\{t_1(x), t_2(x), \dots, k(x)\}$$

$$\hat{f}_{rf}^{k}(x) = \frac{1}{k}\sum_{k=1}^{k} t_i(x)$$

Boosting

Boosting is a methodology used to improve the performance obtained by a single tree. Unlike bagging, which creates multiple independent trees, boosting

creates trees sequentially using prior information from the previous tree (Sousa et al., 2020). This methodology adjusts a large number of decision trees, $\hat{f}^1, \hat{f}^2, \ldots, \hat{f}^B$, for the current residual (Freund and Schapire, 1999), instead of adjusting the model for the response variable Y.

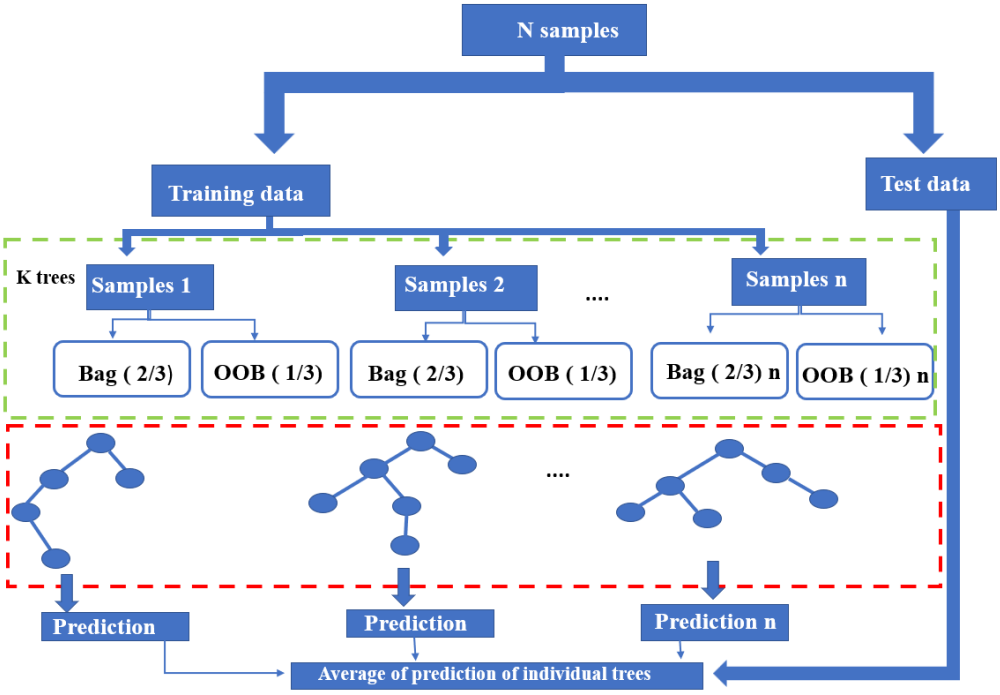

Figure 7. Random forest regression flow.

The learning process of this methodology is considered slow since it requires several models (B). However, care must be taken not to overfit the model. Thus, in boosting cross-validation is used to choose the number of trees to be built, this reduces the possibility of overfitting since all individuals will participate in the validation set (Bengio & Grandvalet, 2004).

The boosting classifier is described in the Equation below:

$$\mathrm{H(x)} = \sum_t \propto_t h_t(x),$$

where it aims to minimize the loss function by optimizing the scalar $\propto_t$ (importance assigned to $\mathrm{h_t(x)}$) and the individual classifier $\mathrm{h_t(x)}$ (individual decision tree) at each iteration t (Freund & Schapire, 1999).

The individual classifiers $\mathrm{h_t(x)}$ have low classifying power, but when used together with H(x), they show good results (Martins et al., 2009; Souza et al., 2020).

Support vector machines

Support Vector Machines (SVM) are part of a methodology with feedforward networks based on learning algorithms (Figure 8). Its operation is based on the theory of statistical learning and pattern recognition, and like multilayer neural networks, it is indicated to solve problems involving pattern classification and linear regression (Haykin, 2001). SVM is a classification and regression approach capable of separating

different classes of data. To find a separation line (hyperplane) between the data, the SVM maximizes the distance between the closest points about each class (Silva et al., 2019). The distance between the hyperplane and the first point of each class is defined as the margin. Thus, the SVM operates margin-maximizing ratings. Each classification step is performed through a predefined kernel function, which can be linear, polynomial, Gaussian, or sigmoidal (Silva et al., 2019).

The SVM is a linear machine for which the main idea is to build a hyperplane – through statistical learning theory and implementation of the structural risk minimization method – as a decision surface so that the separation between positive and negative examples is maximum (Haykin, 2001). Statistical learning theory aims to establish mathematical conditions that allow choosing a classifier with good performance, that is, one that provides the smallest error for the training dataset for better performance, and this error is measured by the percentage of incorrect classifications, called empirical risk R_{emp} (Lorena & Carvalho, 2003).

Regarding the risk minimization method, in turn, it consists of the fact that the error rate of a learning machine on test data is limited by the training error rate and the Vapnik-Chervonenkis dimension (V-C), which measures the complexity arising from limits on functional risk (Haykin, 2001).

The SVM learning algorithm consists of a feature vector, which we will denote by support vectors, and their product by the vector x taken from the input space. The classifier is built according to a set of patterns acquired in training. The support vector helps in the decision-making process (whether the input or individual belongs to a given class or not). The goal of SVM is to find the particular hyperplane for which the separation margin is maximum. Under this condition, the decision surface is referred to as the optimal hyperplane.

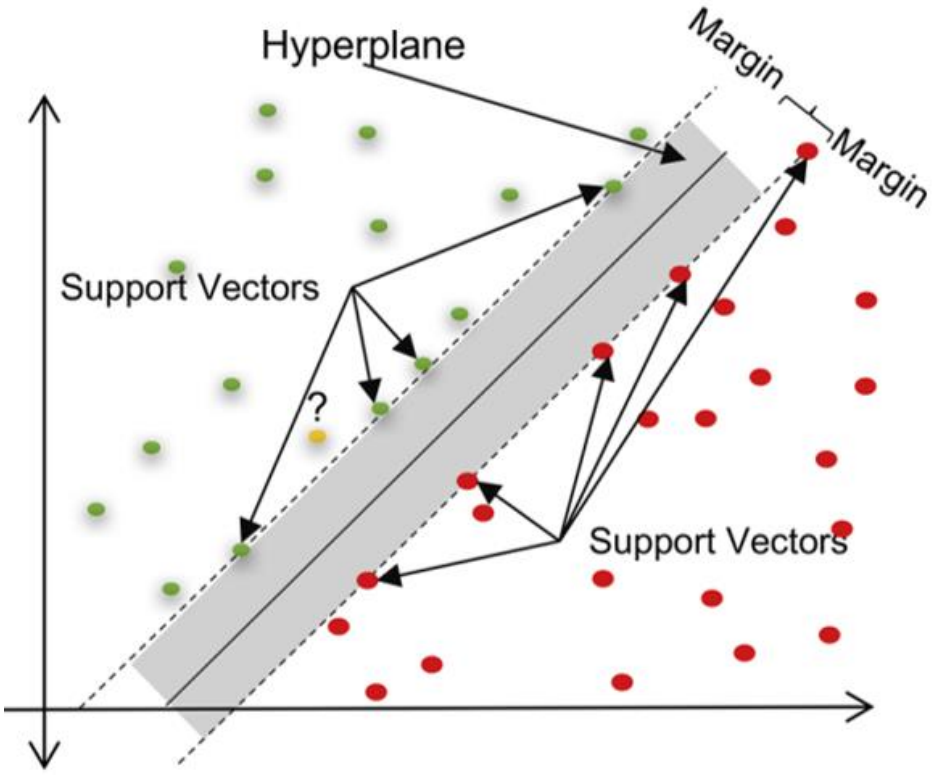

Figure 8. Support vector machine. Green dots represent classes with positive examples and red, negative examples. The yellow dots indicate a new instance to be classified (Silva et al., 2019).

Multivariate adaptive regression splines

Multivariate Adaptive Regression Splines (MARS) (Friedman, 1991) provide a convenient approach to capture the nonlinearity aspect of polynomial regression, where it evaluates the cutoff points (nodes). The definition of the spline is a mathematical curve related by two or more control points, where the control points that are on the curve are called nodes and the other points define the tangent to the curve at their respective nodes (Hastie et al., 2009).

MARS is a methodology for adjusting adaptive nonlinear regression that uses basic piecewise functions to define relationships between a response variable and some set of predictors (Friedman, 1991). Basis functions are defined in pairs, using a node or value of a variable that defines the inflection point along the predicted range. The first step (forward) selects all possible separation points and determines the basic functions. The other step (backward) eliminates redundant basis functions the development to automate all aspects of regression modeling consists of MARS, which aims to use large amounts of apparently uncorrelated variables (York & Eaves, 2001). Therefore, it is an automated and flexible data mining tool that combines the advantages of recursive partitioning and spline adjustment (Lin et al., 2008).

MARS is based on a non-parametric regression method, aimed at partitioning the training data and modeled on linear functions (Zheng et al., 2019), and like ANNs and decision trees, it also makes no assumptions about the distribution of variables predictors (Motsinger et al., 2007). Given this, it combines the strengths of decision trees and the adjustment of splines, in which they replace the step functions normally associated with decision trees with piecewise linear basis functions (Leathwick et al., 2006). This allows for the modeling of complex relationships between a response variable and its predictors. Thus, unlike models based on artificial intelligence, such as ANNs, and decision trees, MARS automatically models nonlinearities and interactions between input variables (Zheng et al., 2019) and presents, at the end of the process, an adjusted model, that is, it presents how the effects are determined in the characteristic under study. In this sense, MARS possibly increases the prediction accuracy for traits that present different types of complex non-additive effects, allowing the researcher to obtain information about the genetic architecture of the trait.

Given this, MARS has some advantages, which are: the ability to separate relevant and irrelevant predictors of a large number of independent variables; any non-linear predictor variables included in the model are automatically transformed (in the form of basic spline functions) concerning the outcome variable; all possible interactions are tested in a user-defined order and certain variable interaction combinations may also be prohibited; user-specified cross-validation protects against model overfitting (York & Eaves, 2001). The result is a reliable non-parametric algorithm that does not require a priori knowledge of the form of the relationship between the result and the predictor variables.

METHODOLOGIES FOR PREDICTING AND VERIFYING THE IMPORTANCE OF TRAIT

Multiple Regression

Stepwise multiple regression is the trait selection method that aims to explain

the relationship between a set of independent traits (x) and a dependent trait (y). The description of the regression model is presented in Eq.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon$$

where y is the response trait, x_1 to x_k are the explanatory trait, β_0 represents the intercept, β_1 and β_k are the slopes related between y and x_1 to x_k, and ε residual error.

The coefficient of determination (R^2) aims to estimate how much of the independent trait is explained by the total variation of the dependent trait. Thus, there is

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

Where y are the real values and e $\hat{y}$ are predicted values.

COMPUTATIONAL INTELLIGENCE FOR THE IMPORTANCE OF TRAITS

Multilayer Perceptron - PMC

The importance of variables through the PMC network can be used in two techniques. The first is based on Garson's (1991) algorithm modified by Goh (1995) which consists of partitioning the neural network connection weights to determine the relative importance of each input trait within the network. This algorithm describes the relative magnitude of importance of descriptors (predictors) in their connection with outcome traits by dissecting the synaptic weights of the neural network.

The equation for the relative importance of the trait is the same:

$$IR = WV$$

Matrix, we have:

$$\begin{pmatrix} IR_1 \\ IR_2 \\ \vdots \\ IR_n \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & w_{1m} \\ w_{21} & w_{22} & w_{2m} \\ \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & w_{nm} \end{pmatrix} \ldots \begin{pmatrix} w_{11} & w_{12} & w_{1m} \\ w_{21} & w_{22} & w_{2m} \\ \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & w_{nm} \end{pmatrix} \ldots \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$

Where, w_n represents the weight of the input neuron in nth neuron; IR: relative importance of the trait in the nth neuron; v_n is the weight from the middle neuron to the output at the nth neuron and each n x m matrix demonstrates the middle layer.

In summary, we have:

$$IR = \begin{pmatrix} IR_1 \\ IR_2 \\ \vdots \\ IR_k \end{pmatrix} = \left(W_{N_1 E}^{1}\right)' \left(W_{N_2 N_1}^{2}\right)' \ldots \left(W_{N_{c-1} y}^{c}\right)'$$

Where, W_x^c represents the neuron weight matrix of layer c, considering N_j neurons and N_{j-1} inputs; E is the first neuron that starts from inputs; y refers to the

desired output layer, and IR: relative importance of the variable.

These weights that connect to neurons in an ANN are partially analogous to the coefficients in a generalized linear model (Olden et al., 2004). The combined effects of the weights represent the relative importance of the predictors in their associations with the response variable. The weights correspond to the relative influence of the information that is processed in the network so that input variables that are not relevant in their correlation with a response variable are suppressed by the weights. On the other hand, the opposite effect is seen for weights assigned to explanatory variables that have strong and positive associations with a response variable.

The method proposed by Garson 1991 modified by Goh 1995 identifies the relative importance of explanatory variables for specific response variables in a supervised neural network. The relative importance (or strength of association) of a specific explanatory variable for a specific response variable can be determined by identifying all weighted connections between nodes of interest.

Figure 9 shows a PMC network with three input neurons (1, 2, and 3), two hidden neurons (A and B), and one output neuron (o) (3-2-1). The relative importance of variables across artificial neural networks using Garson's (1991) algorithm modified by Goh (1995) will be described below.

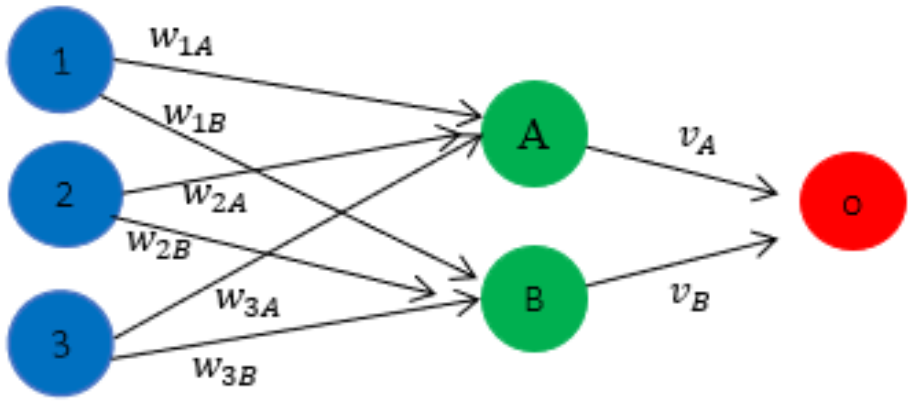

Figure 9. Algorithm by Garson (1991), modified by Goh (1995), to partition and quantify neural network connection weights. Sample calculations are shown for three input neurons (1, 2, and 3)(blue), two hidden neurons (A and B)(green), and one output neuron (o)(red).

The matrix containing the connection weights of neurons in the input-hidden-output layers is shown in Table 2. In this Table, the contribution of each input neuron to the output through each hidden neuron will be estimated, and the product of the input connections -hidden and hidden-output.

Table 3 represents the relative contribution of each neuron and the sum of the relative contributions of each input. Through the sum of the relative contributions, it is possible to estimate the percentage of the relative importance of each variable. The estimate of the relative importance of an entry is estimated by the sum of the relative contribution of this entry over the other sum of the relative contributions.

Table 2. Matrix containing the connection weights of neurons in the input-hidden-output layers.

Input	Hidden	
	A	B
1	w_{1A}= - 2.65	w_{1B}= - 1.25
2	w_{2A}= 0.15	w_{2B}= - 0.95
3	w_{3A}= - 0.70	w_{3B}= -0.40
Output	v_A= 1.10	v_B= -0.40

$$C_{1A} = w_{1A}\, v_A = -2.65 \times 1.10 = -2.91$$

$$C_{1A} = w_{2A}\, v_A = 0.15 \times 1.10 = 0.165$$

$$C_{3A} = w_{3A}\, v_A = -0.70 \times 1.10 = -0.77$$

$$C_{1B} = w_{1B}\, v_B = -1.25 \times (-0.40) = 0.50$$

$$C_{1B} = w_{2B}\, v_B = -0.95 \times (-0.40) = 0.38$$

$$C_{3B} = w_{3B}\, v_B = -0.40 \times (-0.40) = 0.16$$

The relative contribution of each neuron is estimated by the module of each input contribution in the hidden layer over the module of the sum of all contributions, as described:

$$r_{A1} = \frac{|C_{1A}|}{|C_{1A} + C_{2A} + C_{3A}|} = \frac{|-2.91|}{(|-2.91| + |0.165| + |-0.77|)} = 0.76$$

$$r_{A2} = \frac{|C_{2A}|}{|C_{1A} + C_{2A} + C_{3A}|} = \frac{|0.165|}{(|-2.91| + |0.165| + |-0.77|)} = 0.04$$

$$r_{A3} = \frac{|C_{3A}|}{|C_{1A} + C_{2A} + C_{3A}|} = \frac{|0.77|}{(|-2.91| + |0.165| + |-0.77|)} = 0.20$$

$$r_{B1} = \frac{|C_{1B}|}{|C_{1B} + C_{2B} + C_{3B}|} = \frac{|0.50|}{(|0.50| + |0.38| + |0.16|)} = 0.48$$

$$r_{B2} = \frac{|C_{2B}|}{|C_{1B} + C_{2B} + C_{3B}|} = \frac{|0.38|}{(|0.50| + |0.38| + |0.16|)} = 0.37$$

$$r_{B3} = \frac{|C_{3B}|}{|C_{1B} + C_{2B} + C_{3B}|} = \frac{|0.16|}{(|0.50| + |0.38| + |0.16|)} = 0.15$$

Table 3. Relative contribution of each neuron and sum of relative contributions.

Input	Hidden		Sum
	A	B	
1	0.76	0.48	S_1= 1.24
2	0.04	0.37	S_2= 0.41
3	0.20	0.15	S_3= 0.35

$$RI_1 = \frac{S_1}{S_1 + S_2 + S_3} x\ 100 = \frac{1.24}{1.24 + 0.41 + 0.35}\ x\ 100 = 62.0\ \%$$

$$RI_2 = \frac{S_2}{S_1 + S_2 + S_3} x\ 100 = \frac{0.41}{1.24 + 0.41 + 0.35}\ x\ 100 = 20.50\ \%$$

$$RI_3 = \frac{S_3}{S_1 + S_2 + S_3} x\ 100 = \frac{0.35}{1.24 + 0.41 + 0.35}\ x\ 100 = 77.70\%$$

Relative importance of trait by the coefficient of determination - R^2

The importance of variables (inputs) through the impact of destructuring or disturbing the information of a given input on the estimation of the coefficient of determination. This impact causes a reduction in the variance explained by the model (R^2). Swapping a feature breaks the association between the predictor and the target, and provides a reduction in the overall R^2 of the model. The magnitude of the reduction in R^2 when a predictor characteristic is swapped reflects the strength of the association between that predictor characteristic and the response. Consequently, the estimate of R^2 decreases, indicating that the variable is more important than the others, for prediction purposes with the network already established.

The relative importance of the variable by the permutation of R^2 will be described in the following Eq.:

$$pVR_{x_i} = R^2_{obs} - \bar{R}^2_{perm,x_i}$$

where, R^2_{obs} is the R^2 of the ANN model adjusted to the observed predictor and response variables; R^2_{perm,x_i} is the R^2 of the ANN model fitted to the modified dataset where x_i es permuted; $\bar{R}^2_{perm,x_i}$: is the average value of R^2_{perm,x_i} after the mth permutation of the data sets.

Radial Base Function Network – RBF

The radial basis function network is characterized by having only one hidden layer and making use of the Gaussian activation function (Cruz & Nascimento, 2018). The prediction efficiency is measured by the coefficient of determination and the relative importance of each input estimated by the technique of destructuring the information of each explanatory variable, as already described for PMC.

Machine learning for the importance of traits

The quantification of the importance of variables through a machine learning approach using the decision tree and its refinements, random forest, bagging, and boosting. The coefficient of determination measures the goodness of fit of the predictive model and the least squared error (MSE) information is used to quantify the importance of variables. The least squared error was estimated as described in the following Eq.:

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

where, y_i and $\hat{y}_i$ correspond to the observed and predicted value of the observation in genotype *i*, and *n* respectively, and n the total number of observations (trait).

In these techniques, the importance of the explanatory variable is made by quantifying the mean decrease in the prediction accuracy, which consists of estimating the percentage of least squared error increment (IMSE), which is constructed when we permute the values of each variable in the data set and comparing with the prediction of the variable's original non-permuted data set. In analogy to regression analysis, it is the mean increase in the squares of the dataset residuals when the variable is permuted (Li & Zhan, 2019). Higher IMSE values represent higher variable importance.

The importance of the variable by the IMSE permutation will be described in the following Equation:

$$IV_{x_i} = MSE_{perm,x_i} - MSE_{nperm}$$

Where, MSE_{perm,x_i} is the permutation of the values of each variable in the dataset where x_i is permutated; MSE_{nperm}: estimate values from the variable's original non-permuted data.

FINAL COMMENTS

Machine learning and computational intelligence approaches allow inferences about complex interactions in plant breeding. Given this, a systematic review to disentangle machine learning and computational intelligence approaches is relevant to breeders, and it was considered in this review. We present the main steps for developing each strategy (from data selection to evaluating classification/prediction models and quantifying the best predictor).

ACKNOWLEDGMENT

The authors would like to thank the Research Support Foundation of the State of Minas Gerais, the National Council for Scientific and Technological Development, and the Coordination for the Improvement of Higher Education Personnel for the financial support. This study was financed in part by the Coordination for the Improvement of Higher Education Personnel - Brazil (CAPES) - Financial Code 001.

REFERENCES

Beck, M. **(**2018). *Neural Net Tools: Visualization and Analysis Tools for Neural Networks**.** R package version 1.5.2.* http://dx.doi.org/10.18637/jss.v085.i11

Beucher, A., Møller, A. B., & Greve, M. H. (2019). Artificial neural networks and decision tree classification for predicting soil drainage classes in Denmark, *Geoderma,* 352, 351-359. http://dx.doi.org/10.1016/j.geoderma.2017.11.004

Carneiro, A. R. T., Sanglard, D. A., Azevedo, A. M., Souza, T. L. P. O., Pereira, H. S., & Melo, L. C. (2019). Fuzzy logic in automation for interpretation of adaptability and stability in plant breeding studies. *Scientia Agricola*, 76, 123-129. https://doi.org10.1590/1678-992x-2017-0207

Carneiro, V. Q., Prado, A. L., Cruz, C. D., Carneiro, P. C. S., Nascimento, M., & Carneiro, J. E.S. (2018). Fuzzy control systems for decision-making in cultivars recommendation. *Acta Scientiarum. Agronomy,* 40, 1-8. http://dx.doi.org/10.4025/actasciagron.v40i1.39314

Cruz, C. D., & Nascimento, M. (2018). *Inteligência Computacional aplicada ao melhoramento genético*. 1st ed. Vicosa: Editora UFV.

Friedman, J. H. (1991). Multivariate Adaptative regression Splines. *The Annals of Statistics,* 19, 1–141.

Garson, G. D. (1991). *Interpreting neural network connection weights*. *Artificial Intelligence Expert,* 6, 46-51.

Goh, A. T. C. (1995). Back-propagation neural networks for modeling complex systems. *Artificial Intelligence in Engineering*, 9, 143-151. http://dx.doi.org/10.1016/0954-1810(94)00011-S

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction.* 2nd ed. New York, NY, USA: Springer.

Haykin, S. (2001). *Redes neurais – princípios e prática.* 2nd ed. Porto Alegre, RS: Bookman.

Leathwick, J.R., Elith, J., & Hastie, T. (2006). Comparative performance of generalized additive models and multivariate adaptive regression splines for statistical modelling of species distributions. *Ecological Modelling.* 199, 188–196.

Li, L., & Zha, Y. (2019). Estimating monthly average temperature by remote sensing in China. *Advances in Space Research* 63(8), 2345-2357. https://doi.org/10.1016/j.asr.2018.12.039

Lin, H.Y., Wang, W., Liu, Y.H., Soong, S.J., York, T.P., Myers, L, & Hu, J.J. (2008). Comparison of multivariate adaptive regression splines and logistic regression in detecting SNP-SNP interactions and their application in prostate cancer. *Journal of Human Genetics,* 53, 802–811.

Lorena, A. C., & Carvalho, A. C. P. L. F. (2003). *Introdução às Máquinas de Vetores Suporte*. São Carlos, SP: ICMC - USP.

Motsinger, A. A., Ritchie, M. D., & Reif, D. M. (2007). Novel methods for detecting epistasis in pharmacogenomics studies. *Pharmacogenomics,* 8, 1229–1241.

Olden, J. D., & Jackson, D. A. (2002). "Illuminating the "Black Box": A Randomization Approach for Understanding Variable Contributions in Artifical Neural Networks." *Ecological Modelling,* 154, 135–150. http://dx.doi.org/10.1016/s0304-3800(02)00064-9

Paliwal, M., & Kumar, U. A. (2011). Assessing the contribution of variables in feed forward neural network. *Applied Soft Computing*, 11, 3690-3696

Park, J., & Sandberg, I. W. (1991). Universal approximation using radial basis function networks, *Neural Comput*., 3, 246–259. DOI: 10.1162/neco.1991.3.2.246

Parmley, K. A., Higgins, R. H., &Ganapathysubramanian, B., Sarkar, S., & Singh, A. K. (2019). Machine Learning Approach for Prescriptive Plant Breeding. *Scientific Report*, 9, Article number: 17132. http://dx.doi.org/10.1038/s41598-019-53451-4

Paruelo, J. M., & Tomasel, F. (1997). "Prediction of Functional Characteristics of Ecosystems: A Comparison of Artificial Neural Networks and Regression Models." *Ecological Modelling*, 98, 173–186. http://dx.doi.org/10.1016/s0304-3800(96)01913-8

Paswan, R. P., & Begum, S. A. (2013). Regression and Neural Networks Models for Prediction of Crop Production. International Journal of Scientific & Engineering Research, 4, 98-108.Sant'Anna, I. C., Ferreira, R. A. D. C., Nascimento, M., Carneiro, V. Q., Silva, G. N., Cruz, C. D., Oliveira, M. S., & Chagas, F. E. O. (2019). Multigenerational prediction of genetic values using genome-enabled prediction. *PLoS One*, 14, e0210531. http://dx.doi.org/10.1371/journal.pone.0210531

Sant'Anna, I. C., Tomaz, R. S., Silva, G. N., Nascimento, M., Bhering, L. L., & Cruz, C. D. (2015). Superiority of artificial neural networks for a genetic classification procedure. *Genetic and Molecular Research*, 14, 9898–9906.

Silva, G. N., Tomaz, R. S., Sant'Anna, I. C., Carneiro, V. Q., Cruz, C. D., & Nascimento, M. (2016). Evaluation of the efficiency of artificial neural networks for genetic value prediction. *Genetic and Molecular Research,* 15, 1–11. http://hdl.handle.net/11449/158805

Silva, G. N., Tomaz, R. S., Sant'anna, I. C., Nascimento, M., Bhering, L. L., & Cruz, C.D. (2014). Neural networks for predicting breeding values and genetic gains. *Scientia Agricola*, 71, 494-498. http://dx.doi.org/10.1590/0103-9016-2014-0057

Silva, J. C. F., Teixeira, R. M., Silva, F. F., Brommonschenkel, S. H., & Fontes, E. P. B. (2019). Machine learning approaches and their current application in plant molecular biology: A systematic review. *Plant Science*, 284, 37-47. http://dx.doi.org/10.1016/j.plantsci.2019.03.020

Silva Júnior, A. C., Sant'Anna, I. C., Silva, G. N., Cruz, C. D., Nascimento, M., Lopes, L.B., & Soares, P. C. (2023). Computational intelligence and machine learning to study the importance of characteristics in flood-irrigated rice. *Acta Scientiarum-Agronomy*, 45, e57209. http://dx.doi.org/10.4025/actasciagron.v45i1.57209

Silva Júnior, A. C., Silva, M. J., Cruz, C. D., Santanna, I. C., Silva, G. N., Nascimento, M., & Azevedo, C.F. (2021). Prediction of the importance of auxiliary traits using computational intelligence and machine learning: A simulation study. *PLoS One*, 16, e0257213. https://doi.org/10.1371/journal.pone.0257213

Silva Júnior, A. C., Silva, M. J., Sousa, I., Costa, W. G., Cruz, C. D., Nascimento, M., & Soares, P. C. (2021). Fuzzy logic for adaptability and stability studies in irrigated rice (*Oryza Sativa* L.) genotypes. *Plant Breeding*, v. 140, p. 719-980. https://doi.org/10.1111/pbr.12973

Skawsang, S., Nagai, M., Nitin, K., & Soni, P. (2019). Predicting Rice Pest Population Occurrence with Satellite-Derived Crop Phenology, Ground Meteorological Observation, and Machine Learning: A Case Study for the Central Plain of Thailand. *Appl. Sci*. 9:4846. http://dx.doi.org/10.3390/app9224846.

Sousa, I. C., Nascimento, M., Silva, G. N., Nascimento, A. C. C., Cruz, C. D., Fonseca, F., Almeida, D. P., Pestana, K. N., Azevedo, C. F., Zambolim, L., & Caixeita, E.T. (2020). Genomic prediction of leaf rust resistance to Arabica coffee using machine learning algorithms. *Scientia Agricola*, 78, 1–8. https://doi.org/10.1590/1678-992x-2020-0021

Sreekanth, S., Ramaswamy, H. S., Sablani, S. S., & Prasher, S. O. (2010). A neural network approach for evaluation of surface heat transfer coefficient. *Journal of Food Processing and Preserv*ation, 23, 329-348. https://doi.org/10.1111/j.1745-4549.1999.tb00389.x

Stefaniak, B., Cholewiński, W., & Tarkowska, A. (2005). Algorithms of Artificial Neural Networks - Practical application in medical science. *Polski Merkuriusz Lekarski*. 19, 819-822.

Tan, K., Li, E., Du, Q., & Du, P. (2014). An efficient semi-supervised classification approach for hyperspectral imagery. ISPRS *Journal of Photogrammetry and Remote Sensing*, 97, 36–45. http://dx.doi.org/10.1016/j.isprsjprs.2014.08.003

Ventura, R. V., Silva, M. A., Medeiros, T. H., Dionello, N. L., Madalena, F. E., Fridrich, A. B., Valente, B. D., Santos, G. G., Freitas, L. S., Wenceslau, R. R., Felipe, V. P. S., & Corrêa, G. S. S. (2012). Use of artificial neural networks in breeding values prediction for weight at 205 days in Tabapuã beef cattle. *Arquivo Brasileiro de Medicina Veterinária e Zootecnia,* 64, 411-418. http://dx.doi.org/10.1590/S0102-09352012000200022.

York, T. P., & Eaves, L. J. (2001). Common Disease Analysis Using Multivariate Adaptive Regression Splines (MARS): Genetic Analysis Workshop 12 Simulated Sequence Data. *Genetic Epidemiology,* 21, S649–S654.

Yu, H., Campbell, M.T., Zhang, Q., Walia, H., & Morota, G. (2019). Genomic Bayesian confirmatory factor analysis and Bayesian network to characterize a wide spectrum of rice phenotypes. *G3: Genes, Genomes, Genetics*, 9, 1975-1986. http://dx.doi.org/10.1101/435792.

Zheng, G., Yang, P., Zhou, H., Zeng, C., Yang, X., He, X., & Yu, X. (2019). Evaluation of the earthquake induced uplift displacement of tunnels using multivariate adaptive regression splines. *Computers and Geotechnics,* 113, 103099.