DOI: 10.2507/33rd.daaam.proceedings.067

PEACH FIRMNESS PREDICTION USING OPTIMIZED REGRESSION TREES MODELS

Tomislav Ivanovski, Xiaoshuan Zhang, Tomislav Jemrić,
Marko Gulić & Maja Matetić

This Publication has to be referred as: Ivanovski, T[omislav]; Zhang, X[iaoshuan]; Jemric, T[omislav]; Gulic, M[arko] & Matetic, M[aja] (2022). Peach Firmness Prediction Using Optimized Regression Trees Models, Proceedings of the 33rd DAAAM International Symposium, pp.0480-0489, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-36-5, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/33rd.daaam.proceedings.067

Abstract

The paper focuses on creating accurate model for peach firmness prediction. For this purpose, multiple machine learning models and their optimized variations are developed and compared. Because of its simplicity and robustness, multiple linear regression is used as a base-line model for predicting peach firmness. It assumes a linear relationship between numerical predictors and the outcome. Regression trees is the second developed model. It is a flexible data-driven model that can be used for predicting numerical outcome. The experiment aims to investigate the possibility of improving regression tree model using various metaheuristic optimization techniques implemented in *metaheuristicOpt* and *GA* R packages. As a proof of concept, prediction accuracy between multiple linear regression, regression trees and optimized regression trees models is compared. The results shows that it is possible to improve the peach firmness prediction accuracy of regression trees model using metaheuristic algorithms.

Keywords: Peach firmness prediction; regression trees; global optimization; machine learning; metaheuristics

1. Introduction

Reduction of food waste is a crucial part of global food security and leads to more sustainable food production. Determining the correct fruit harvest time is critical for food waste reduction along the supply chain, and it ensures that the fruit meets the ripeness specifications [1]. Therefore, an early assessment of fruit ripeness using machine learning models trained on data obtained from non-destructive measurement techniques can lead to precise prediction of harvest date and extension of fruit shelf life. Colorimetry, visible imaging, electrical impedance spectroscopy, and spectroscopy imaging [2] are some of the non-destructive techniques that produce high-quality data. Hardware innovations, quality datasets and algorithmic advances are three main factors driving advances in machine learning [3]. Research on sensing technologies in the agriculture sector is being conducted [4] which will result in more quality data for analysis. With increased availability of quality sensor data, variations of machine learning models can be created for data analysis and future predictions [4]. The work presented in this paper aims to extend the research conducted in [29], where simple linear regression, multiple linear regression, and backpropagation neural network models were developed for peach firmness prediction.

In this experiment, a larger dataset is used for the development of the machine learning models, and the new regression tree model is created for firmness prediction. In addition, a special focus is given to the application of nature inspired metaheuristic algorithms as global optimizers for the machine learning models. Evolutionary algorithms (EA) are global optimization techniques used to solve optimization problems. All variants of EAs have the common underlying logic, where an initial random set of candidate solutions is created, and fitness function is applied to each candidate. Based on the fitness values, the best candidate solutions are selected as parents for the offspring. This paper is organized in seven chapters. The review of the papers with the similar topic of research is given in Ch. 2. Features that make up the data set are described in Ch. 3, as well as the machine learning and metaheuristic algorithms used in the experiment. In Ch. 4 the results of the experiment are described, and the most accurate model is presented. The conclusion is given in Ch. 5.

2. Related works

In the previous research [29] simple linear regression, multiple linear regression, and backpropagation neural network models were developed for peach firmness prediction. The dataset used consisted of four measured features: density, firmness, ratio of soluble solids content and titratable acidity, and magnitude of impedance. Experiment results show that multiple linear regression model is the most accurate, which is expected since the dataset is small and the features in it have large variance. This research extends the [29] by including two additional features: angle of impedance and colour. In addition, regression trees model was created, and various nature inspired metaheuristic algorithms were applied in order to additionally optimize the model. Another peach firmness prediction research by G. Zhang et al. is described in [7]. The correlation analysis in [7] shows that the highest correlation coefficients are between the dielectric properties and firmness of the peach. Using the measured peach dielectric properties as inputs to multi-layer neural network model, peach firmness is accurately predicted. The research also shows that the proportion of peach red surface area has small relationship with the quality scores calculated by the grading model developed in [7]. Similar to [29], in [8] D. Jamaludin et al. study the dielectric characteristic of banana fruit at different ripening stages using impedance measurement. Using the measured impedance, it is possible to differentiate the unripe, ripe, and overripe banana [8]. The experiment shows that the impedance correlates with the soluble solid content. This discovery is used to develop simple linear regression model to predict soluble solid content value based on measured impedance, which indicates that banana ripeness can be predicted by its dielectric properties. The study described in [9] uses image data of melon skin colour to distinguish ripe melons from unripe ones. The developed linear discriminant analysis model can be used to accurately predict melon harvesting time using the colour image data. Such model will prove useful as a preharvest tool for making decisions about whether the melon is ready to harvest. Like the study described in [9], in [10] areal images of apples are used for development of an apple ripeness classifier model. The described classifier model is based on the artificial neural network model optimized using the genetic algorithm. The model classifies the apples to one of the following stages of ripeness: unripe, half-ripe, ripe, and overripe.

Nature inspired metaheuristic algorithms are applicable in wide range of applications. Optimization of liner and exponential regression models to forecast global CO2 emissions using bat algorithm is presented in [11], while in [12] the bat algorithm is used to optimize fuzzy PD speed controller for brushless DC motor. Another application of BA is described in [13], where BA is used to optimize photo-voltic systems to operate at the maximum power point. Optimization of photo-voltic model is also described in [14], where optimization is realized using combination of DE algorithm and reinforcement learning. The study in [16] describes the use of DE algorithm for determining the appropriate number of wells and the maximum oil processing capacity in an oil and gas industry. In [15] an improved adaptive differential evolution (IADE) algorithm which adopts its parameters during execution is used to optimize the flight path trajectory of unmanned arial vehicles. The application of DE for CoCoMo and CoCoMo II model optimization is described in [17], while a modified DE algorithm that is used to optimize recommendation systems is described in [18]. In [19] the PSO algorithm is used to improve the accuracy and convergence speed of radial basis function neural network (RBFNN) for estimating the battery pack SOC. Similar in [20] the improved-hybrid particle swarm optimization (IH-PSO) algorithm is used to optimize the batch size in batch processing, while in [21] a modified PSO algorithm is used to optimize collaborative beam folding process in wireless sensor networks. Another application of metaheuristic algorithms in wireless sensor networks is described in [22], where the energy consumption was successfully optimized by using GA. The demonstration of GA use in the optimization of nuclear reactor core load, including fuel reloads is described in [23], while the study in [24] uses a GA as an optimization technique for creating a household level hybrid renewable energy system. In [25] the neuro-fuzzy model in combination with the GA are used to optimize the train routes on the railway network, where in [26] an optimization method based on GA is used to optimize ship voyage and therefore minimizing fuel consumption and greenhouse gas emissions. The study in [27] gives insight on the use of GA for optimizing structural design of deck concrete arch bridges, while the study in [40] utilizes GA for obtaining optimal design for I-beams.

3. Dataset description and methods

3.1. Data set description

The dataset on which the experiment is conducted consists of various peach feature measurements. Peach impedance is one of the measured features and it is measured with 500 mV and 10 kHz frequency alternate voltage.

Insight into the structural characteristics of biological tissues is obtained using electrical impedance measurements [28]. The dielectric properties are related to changes in the membrane structure [29], and changes in the membrane structure reflect changes in fruit ripening [30][31][32]. Since impedance is a complex number and can be represented in the polar form using magnitude and phase angle, the two main predictors in the dataset are *Zs* and *Angle* respectively. By combining juice extracted from a peach with alkaline solution, the titratable acidity *TA* of the peach is obtained. The soluble solids content *SSC* is obtained using a refractometer and juice extracted from a peach, while peach firmness is measured using penetrometer. Because the firmness is a good maturity indicator [33], it is used as outcome in the machine learning models described in the paper. The colour of the peach is represented by Delta E value. Initial dataset consisted of twenty-one measured features. To increase the efficiency of machine learning algorithms the dimension of the data set had to be reduced [34] using the expert's domain knowledge. An example of feature combination is quotient of SSC and TA which is used in the final dataset. The final dataset used in the experiment consists of 200 observations and 6 features. Fig. 1 shows the distribution of the measured features used in the experiment. It is possible to see that the colour feature follows symmetric normal distribution, while the ratio of soluble solids content and titratable acidity and density are slightly skewed to the right. Firmness, phase angle of impedance, and the impedance magnitude follow bimodal distribution. By examining the distribution of firmness, it is possible to get an insight into the characteristics of peaches that make up the data set. The firmness distribution is skewed towards lower values, and peaches with lower firmness tend to be riper [33].

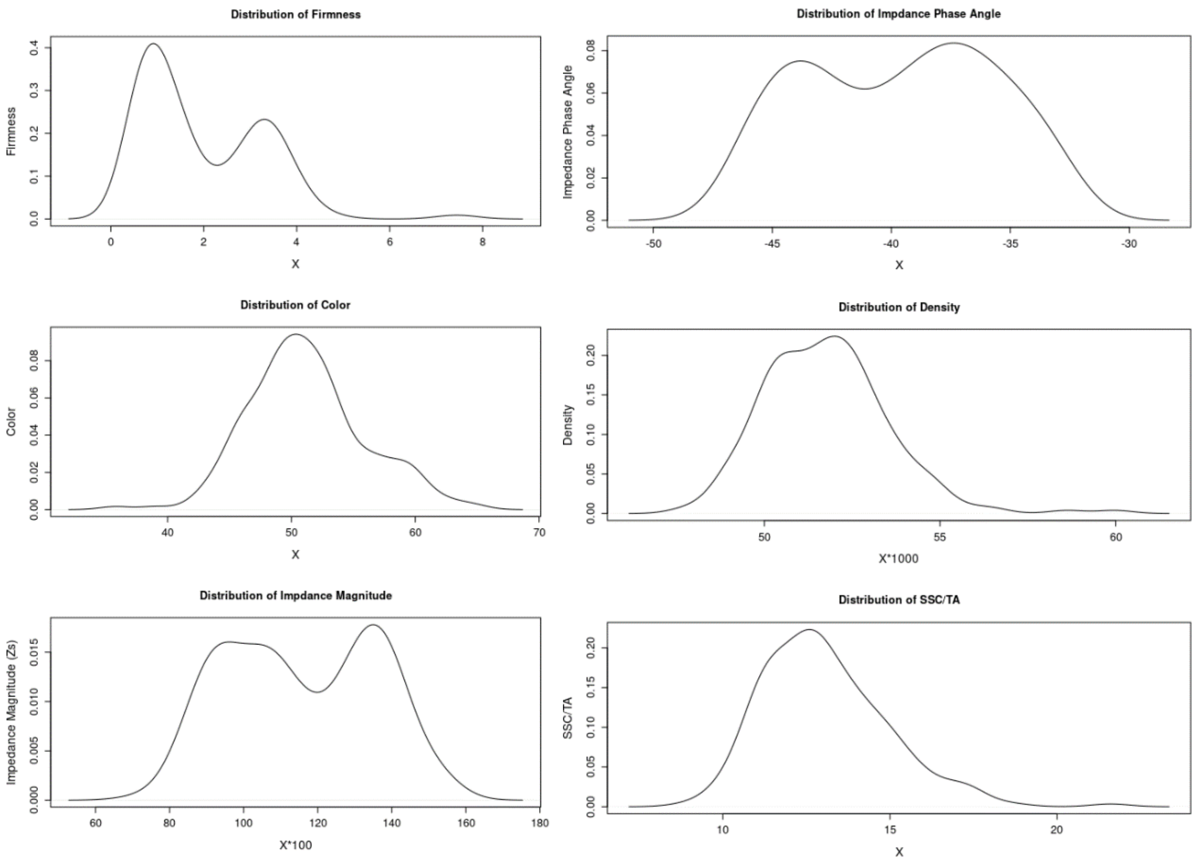

Fig. 1. Distribution of features in dataset

3.2. Classification and regression trees

Classification and regression trees (CART) are a flexible data driven method that can be used for classification or numerical prediction [34]. Segmentation of the predictor space into multiple simple regions using recursive partitioning and pruning processes is the main idea behind CART algorithm [35]. The resulting subgroups should be more homogeneous in terms of the outcome feature, thereby creating useful prediction rules that are easily interpreted by humans. The most prominent advantage of CART is its simplicity and transparency. CART model consists of terminal and decision nodes. The decision nodes give the splitting value on the specific predictor while the terminal nodes contain predicted value. For the regression program, the values of terminal nodes are the mean of the outcome feature of training observations in the adequate region to which the observations belong. Fig. 2 shows an example of regression tree model created using *Hitters* dataset included in *ISLR* R library. Terminal nodes are represented by round rectangles, while sharp rectangles represent decision nodes.

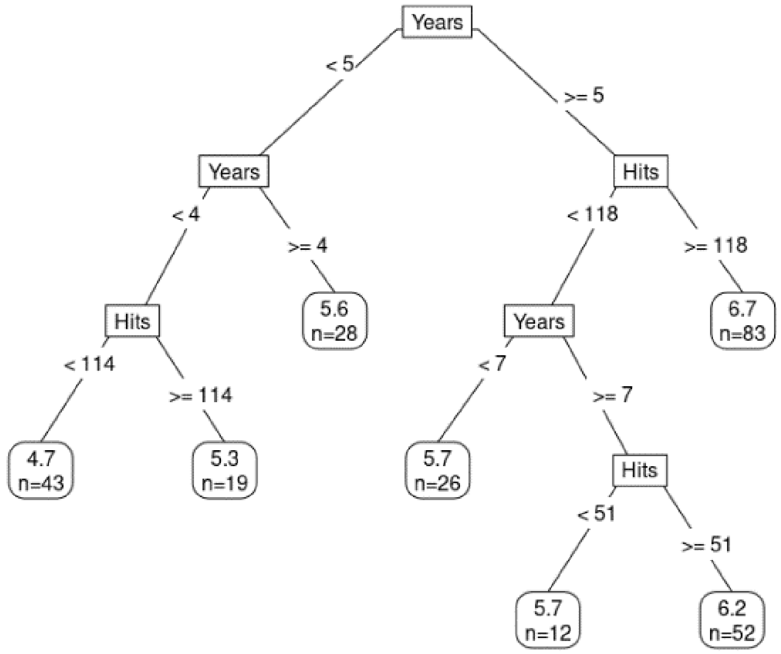

Fig. 2. Regression tree model example

Regression tree models are not very robust, meaning a small change in the data can cause a large change in the final estimated tree. However, by aggregating many decision trees, using methods like bagging, random forests, and boosting, the predictive performance of trees can be substantially improved [35].

3.3. Multiple linear regression

Multiple linear regression (MLR) method assumes a linear relationship between multiple predictors and outcome feature. It is a straightforward approach for predicting a numerical outcome based on multiple predictors [35]. Multiple linear regression, just like simple linear regression uses least square approach for adjusting the coefficients [35]. A generic expression for multiple linear regression model that approximates relationship between the predictors and the outcome is an extension of the simple linear regression model [35] and is shown in Eq. (1):

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \ldots + \beta_n X_n + e \tag{1}$$

Where:
- $\beta_{0\ldots n}$: Multiple linear regression coefficients
- $X_{1\ldots n}$: Feature values
- e: Mean-zero random error term

3.4. Genetic algorithm

Genetic algorithm is a metaheuristic algorithm that belongs to a family of evolutionary algorithms and can be applied in solving global optimization problems. Inherent characteristic of all metaheuristics algorithms is that they make few or no assumptions about the problem being optimized, and therefore can easily be applied to a wide range of optimization problems. GA is one of multiple variants of evolutionary algorithms that is inspired by Charles Darwin's theory of evolution. Given a population of individuals within some environment that has limited resources, competition for those resources causes natural selection, which in turn causes a rise in the fitness of the population [5]. By defining a quality function to be maximized and creating a population of initial candidate solutions, it is possible to apply the quality function to candidate solutions as an abstract fitness measure - the higher the better [5]. Based on calculated fitness values some of the better candidates are chosen as parents to the next generation. The offspring is created by applying recombination and mutation operations on parent candidates. Recombination operator is applied to two or more selected parents, producing one or more new candidates. The mutation operator is applied to a single candidate and results in one new candidate. Both parents and offspring have their fitness evaluated and then compete based on their fitness (and possibly age) for a place in the next generation. This process can be iterated until a candidate with sufficient quality is found or a previously set computational limit is reached [5]. Fig. 3 shows rise in fitness value of the best candidate solution (green) and the population (blue) during the run time of genetic algorithm used in decision tree model optimization, which is described in Ch. 4.

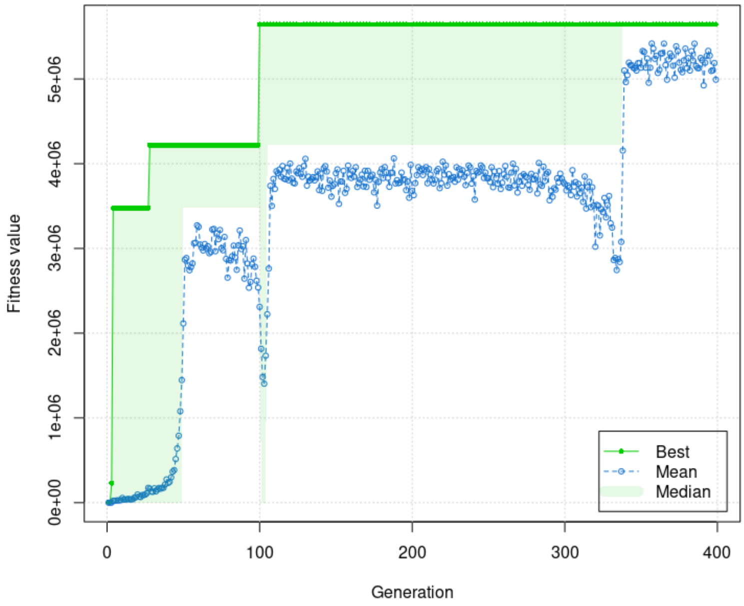

Fig. 3. Change in fitness value over time

3.5. Bat algorithm

Like the genetic algorithm, the bat algorithm (BA) is a metaheuristic algorithm for global optimization. The bat algorithm and particle swarm optimization algorithm can be classified as swarm intelligence (SI) algorithms. Like EA, SI algorithms are global optimization techniques that use a swarm of multiple, interacting agents to generate search moves in the search space. A wide spectrum of SI based algorithms have emerged in the last decades, but the lack of mathematical framework and in-depth understanding of how such algorithms may converge are still some of the important issues [6]. The bat algorithm was inspired by the echolocation behaviour of microbats, with varying pulse rates of emission and loudness. BA essentially uses a frequency tuning technique to increase the diversity of the solutions in the population, while the balance between exploration and exploitation can be controlled by tuning algorithm-dependent parameters [36]. Each bat *i* is associated with a velocity v_i^t and a location x_i^t, at iteration t, in a d-dimensional search space. The best solution in a population in any given iteration is marked with x^*. Following are equations for x_i^t and velocities v_i^t. The B [0,1] is a random vector drawn from a uniform distribution.

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{2}$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x^*)f_i \tag{3}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{4}$$

The BA can be considered as a frequency-tuning algorithm to provide a balanced combination of exploration and exploitation of the search space. The loudness and pulse emission rates essentially provide a mechanism for automatic control and auto-zooming into the region with promising solutions. In order to provide an effective mechanism to control the exploration and exploitation and switch to exploitation stage, when necessary, the loudness A_i and the rate of pulse emissions r_i must change during algorithm iterations. In following equations α and γ are constants.

$$A_i^{t+1} = \alpha A_i^t \tag{5}$$

$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \tag{6}$$

3.6. Differential evolution

Differential evolution (DE) algorithm is part of the family of evolutionary algorithms and is used for solving real parameter global optimization problems. DE is robust and simple algorithm and like other EAs. It can produce new offspring solutions through three mechanisms: mutation, crossover, and selection [37]. It utilizes directional information from the population, where each individual in the current generation is allowed to breed with multiple other randomly selected individuals from the population.

The algorithm starts by initializing all agents in the search space with random positions, after which mutation is applied to create a vector v_i^t [38]. The DE standard mutation operator needs three randomly selected different individuals from the current population for each individual to create a mutated vector. The goal of standard mutation operator is to recognize good variation directions and to increase the number of generations having fitness improvement [38].

$$v_i^t = x_j^t + F(x_k^t + x_l^t) \tag{7}$$

Where F is a differential weight real constant whose values is between 0 and 1. Offspring individual u_i^t is created by mating of the mutated individual v_i^t with x_i^t. The genes *m* of u_i^t are determined by the crossover probability $C_r \in [0, 1]$ and are inherited from both x_i^t and v_i^t [38].

$$u_{i,m}^t = \begin{cases} v_{i,m}^t\,,\ if\ rand(m) \leq C_r\ or\ m = rn(i) \\ x_{i,m}^t\,, if\ rand(m) > C_r\ and\ m \neq rn(i) \end{cases} \tag{8}$$

Where m=1,…, N corresponds to the m^{th} gene of an individual vector. The expression rand(m) $\in [0, 1]$ is the m^{th} evaluation of a uniform random number generator and rn(i) $\in \{1,...,N\}$ is a randomly chosen index which ensures that u_i^t gets at least one element from v_i^t [38].

The selection process in which fittest individuals are selected as parents for the next generation is conducted between each individual x_i^t and its offspring u_i^t. The winner is selected based on objective function values and promoted to the next generation. The old generation is replaced by the new one and the search process continues until the stopping condition is fulfilled.

3.7. Particle swarm optimization

Particle swarm optimization (PSO) is a metaheuristics algorithm inspired by the social cooperation of organisms observed in nature [39]. It solves an optimization problem by having a population of candidate solutions (particles) that work under social behaviour in swarms. The most prominent characteristic of PSO algorithm is the socio-cognitive learning process that is based on a particle's own experience and the experience of the most successful particle in the swarm. For an optimization problem of *n* variables, a swarm of N_P particles is defined, where each particle is assigned an initial random position in the n-dimensional space. Each particle has its own trajectory, namely position x_i and velocity v_i. In every iteration, each particle is updated by following the two best values: the best solution each particle has achieved so far x_i^*, and the best value obtained so far by any particle in the population x_g [38]. At iteration $t + 1$, the swarm can be updated by following equations.

$$v_i(t+1) = v_i(t) + cr_1[x_i^*(t) - x_i(t)] + cr_2[x_g(t) - x_i(t)] \tag{9}$$

$$x_i(t+1) = x_i(t) + v_i(t+1), i = 1, \dots, N_P \tag{10}$$

Where the acceleration constant $c > 0$, and r_1 and r_2 are uniform random numbers within [0, 1]. The intuitively simple representation and low number of adjustable parameters of PSO algorithm make it a popular choice for solving optimization problems [39]. Prominent characteristic of PSO algorithm is that it can locate the optimum region faster than EAs, but once in this region it progresses slowly due to the fixed velocity step size [38].

4. Results

Eighty percent of data set observations make the training and validation data, while the remaining twenty percent make the test data. The training data consists of 160 observations that are used to create the machine learning models, while the test data consists of 40 observations used to estimate their accuracy. Multiple linear regression (MLR) is used as a baseline machine learning model for predicting peach firmness. The root-mean-square error (RMSE) is used to estimate the accuracy of the model. The RMSE value for multiple linear regression model on test data is 1.717526. The GA, BA, DE, and PSO global optimization algorithms were applied to the regression model, but no significant change in the β coefficients and final test RMSE was obtained. Next, the CART model is trained. The default pruned tree is automatically built by selecting the complexity parameter that yields smallest error on validation data based on cross validation procedure using *rpart* R library. Using the *rpart* R library the cross validation and pruning of the tree is done automatically. From the Fig. 4 it is possible to see that the default pruned tree consists of seven decision nodes and eight terminal nodes. Note that recursive partitioning and pruning algorithms responsible for creating the default pruned tree did not include the colour feature in the decision nodes of the final model. By including it, additional complexity would be added to the model, and it would be prone to overfitting [35]. Below each decision node is the yes-no conditional based on which the split is made. Inside each decision and terminal node is the mean value of firmness that is calculated using observations that satisfy conditions specified by the decision nodes. The RMSE of default pruned tree on test data is 1.747593, which is slightly higher then base-line linear regression model.

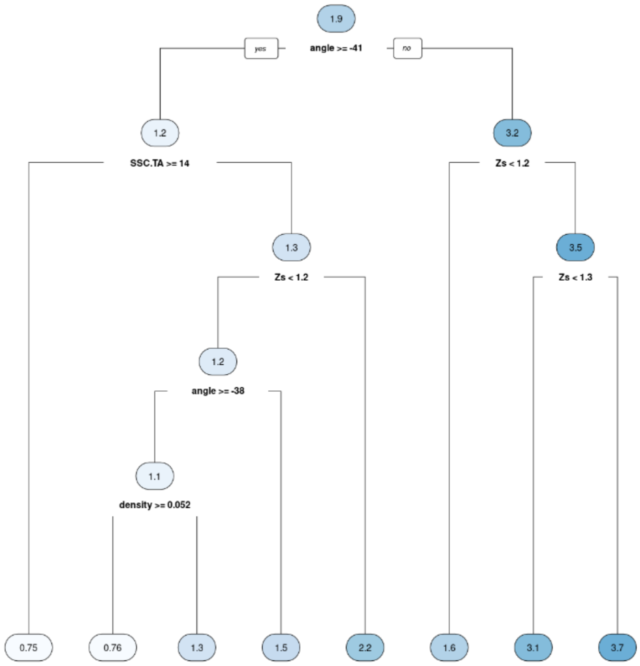

Fig. 4. Default pruned tree created using *rpart* R library

Using various metaheuristic algorithms, it is possible to optimize the splitting value on the predictors in the default pruned tree, thus obtaining modified CART model. By modifying the splitting value, the mean value of firmness in terminal nodes will change, hence predictions on test data will differ. For GA the upper bounds of the search space are defined as 50 percent above the splitting value of decision node in the default pruned tree, while the lower bounds are defined as 50 percent below the splitting value. The goal is the maximization of the objective function, thus the individual with the highest fitness value will be selected as the best one. The population size is set to 200 and the number of best individuals to survive at each generation is set to 40 percent. The *GA* R library is used to implement the genetic algorithm optimization of the splitting values. The model modified using GA is shown on Fig. 5, and its estimated accuracy is 1.589032, which is an improvement from the default pruned tree.

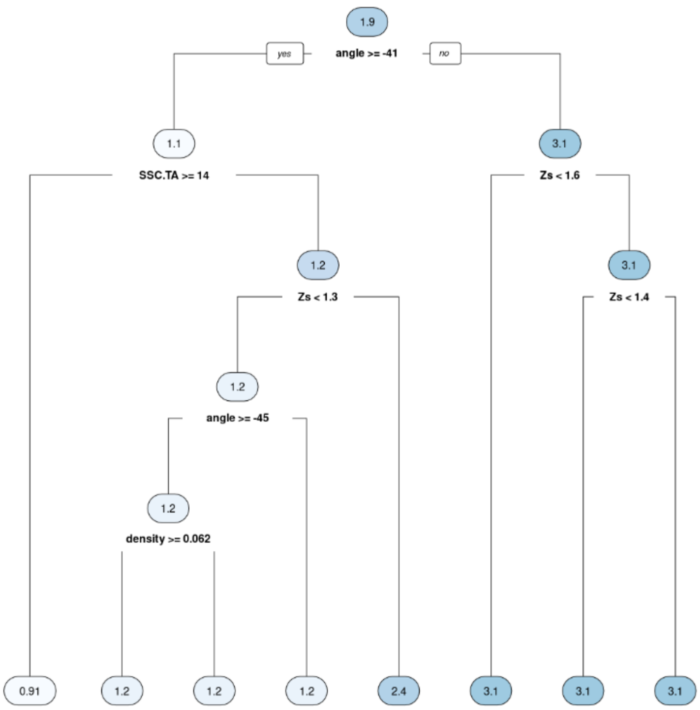

Fig. 5. Default pruned tree modified using GA

For BA, DE and PSO optimization techniques the upper bounds of the search space are defined as 10 percent above the splitting value of decision node in the default pruned tree, while the lower bounds are defined as 10 percent below the splitting value. For the BA, the population size is set to 40, while the maximum and minimum frequencies are set to the value of 0.1. The factor responsible for increasing the pulse rate is set to its default value of 1, and the factor used to decrease loudness is set to 0.1. The *metaheuristicOpt* R library is used to implement the BA, DE and PSO optimizations of the splitting values. The estimated accuracy for the model modified using BA is 1.741407, which is approximately the same as the default pruned tree.

The DE evolutionary algorithm defines that each individual in the generation is allowed to mate with other randomly selected individuals. The scaling factor for mutation operator is set to 0.8, while the factor determining the crossover probability is set to 0.5. The population size is set to 20. The estimated accuracy for the model modified using DE is 1.635962, which is an improvement from the default pruned tree.

When implementing PSO as an optimization algorithm for splitting values of default pruned tree model, the inertia weight is set to 0.729. Both the individual and group acceleration constants are set to 1.49445. As with DE, the population size is set to 20. The estimated accuracy for the model modified using PSO is 1.735599, which is slightly better than the default pruned tree. The Table 1 summarizes all the created models.

Model	RMSE
MLR	1.717526
Default CART	1.747593
GA Optimized CART	1.589032
BA Optimized CART	1.741407
DE Optimized CART	1.635962
PSO Optimized CART	1.735599

Table 1. Test RMSE summary

5. Conclusion

The aim of this research is to successfully predict the correct peach harvest time by predicting its firmness using machine learning models and measured peach features. Multiple linear regression model, default CART model, and CART model optimized using GA, BA, DE, and PSO metaheuristic algorithms are used for predicting peach firmness. Prediction results are compared, and the most accurate model was found. The results showed that CART model optimized using GA gave the most accurate predictions on the new data. This empirical experiment shows that by using various metaheuristic optimization techniques implemented in *metaheuristicOpt* and *GA R* libraries it is possible to improve the accuracy of the default CART model. It is important to note that the experiment is limited by the used *R* libraries and the optimization approach previously described. Hence different results could be obtained if other software packages or optimization approaches are used.

The future research will focus on applying additional nature inspired metaheuristic algorithms as optimizers for CART and other machine learning models. In addition, an adaptive network-based fuzzy inference system (ANFIS) model will be created, and its performance will be compared to the performance of models described in this paper.

6. Acknowledgement

This research was funded by the University of Rijeka, grant number uniri-drustv-18-122.

7. References

[1] Bo Li, J. Lecourt, G. Bishop. (2018). Advances in Non-Destructive Early Assessment of Fruit Ripeness towards Defining Optimal Time of Harvest and Yield Prediction - A Review. Plants 2018, 7, 3

[2] Satyam Srivastava, Shashikant Sadistap. (2017). Non-destructive sensing methods for quality assessment of on-tree fruits: a review. Springer Science+Business Media, LLC 2017

[3] F. Chollet., J.J. Allaire. (2017). Deep Learning with R - Version 1. Manning Publications (2017)

[4] Sonka, Steve. (2016). Big data: fueling the next evolution of agricultural innovation. Journal of Innovation Management 4.1: 114.

[5] A.E. Eiben, J.E. Smith. (2015). Introduction to Evolutionary Computing Second Edition. Springer (2015)

[6] Xin-She Yang. (2020). Nature-inspired optimization algorithms: Challenges and open problems, Journal of Computational Science 46 (2020)

[7] G. Zhang, Q. Fu, Z. Fu, X. Li, M. Matetić, M. B. Bakaric, T. Jemrić (2020). A Comprehensive Peach Fruit Quality Evaluation Method for Grading and Consumption. Appl. Sci. 2020, 10, 1348;

[8] D. Jamaludin, S. Abd Aziz, N. U. A. Ibrahim. (2014). Dielectric Based Sensing System for Banana Ripeness Assessment. International Journal of Environmental Science and Development, Vol. 5, No. 3, June 2014

[9] U. Ahmad, D. P. Bermani, Mardison. (2018). Color Distribution Analysis for Ripeness Prediction of Golden Apollo Melon. Telkomnika, Vol.16, No.4, August 2018, pp. 1659 1666

[10] S. Sabzi, Y. Abbaspour-Gilandeh , G. García-Mateos, A. Ruiz-Canales, J. M. Molina-Martínez, J. I. Arribas. (2019) An Automatic Non- Destructive Method for the Classification of the Ripeness Stage of Red Delicious Apples in Orchards Using Aerial Video. Agronomy 2019, 9, 84

[11] M. Bahmani, A. GhasemiNejad, F. N. Robati, N. A. Zarin. (2020). A novel approach to forecast global CO 2 emission using Bat and Cuckoo optimization algorithms, MethodsX 7 (2020) 100986

[12] K. Premkumar, B.V. Manikandan. (2016). Bat algorithm optimized fuzzy PD based speed controller for brushless direct current motor. Engineering Science and Technology, an International Journal 19 (2016) 818–840

[13] Ali M. Eltamaly, M. S. Al-Saud, Ahmed G. Abokhalil. (2020). A Novel Bat Algorithm Strategy for Maximum Power Point Tracker of Photovoltaic Energy Systems Under Dynamic Partial Shading. IEEE Access, Digital Object Identifier 10.1109/ACCESS.2020.2964759

[14] Z. Hu, W. Gong, S. Li. (2021). Reinforcement learning-based differential evolution for parameters extraction of photo-voltaic models. Elsevier, Energy Reports 7 (2021) 916–928

[15] Jian-li Su, H. Wang. (2021). An improved adaptive differential evolution algorithm for single unmanned aerial vehicle multitasking. KeAi, Defence Technology 17 (2021) 1967 - 1975

[16] Bilal, M. Pant, M. Stanko, L. Sales. (2021). Differential evolution for early-phase offshore oilfield design considering uncertainties in initial oil-in-place and well productivity. Elsevier, Upstream Oil and Gas Technology 7 (2021) 100055

[17] P. Singala, A C. Kumarib, P. Sharmaa. (2020). Estimation of software development effort: A differential evolution approach. Elsevier, Procedia Computer Science 167 (2020) 2643–2652

[18] U. Boryczka, M. Bałchanowski. (2021). Speed up differential evolution for ranking of items in recommendation systems. Elsevier, Procedia Computer Science 192 (2021) 2229–2238

[19] G. Zhang , B. Xia, J. Wang , Bo Ye , Y. Chen , Z. Yu , Y. Li. (2022). Intelligent state of charge estimation of battery pack based on particle swarm optimization algorithm improved radical basis function neural network. Elsevier, Journal of Energy Storage 50 (2022) 104211

[20] J. M. Powera, C. Ou-Yanga, Y. Juanb. (2021). Optimize batch size combination using improved hybrid particle swarm optimization. Elsevier, Sixth Information Systems International Conference (ISICO 2021)

[21] R. M. Mainaa, P. K. Lang'ata, P. K. Kihatoa. (2021). Collaborative beamforming in wireless sensor networks using a novel particle swarm optimization algorithm variant. Elsevier, Heliyon 7 (2021) e08247

[22] W. Yan, S. Xin-xin, S. Yan-ming. (2011). Study on the application of Genetic Algorithms in the optimization of wireless network. Elsevier, Procedia Engineering 16 (2011) 348-355

[23] A.V. Sobolev, A. S. Gazetdinov, D.S. Samokhin. (2017). Genetic algorithms for nuclear reactor fuel load and reload optimization problems. Elsevier, Nuclear Energy and Technology 3 (2017) 231–235

[24] M. J. Mayer, A. Szilágyi, G. Gróf. (2020). Environmental and economic multiobjective optimization of a household level hybrid renewable energy system by genetic algorithm. Elsevier, Applied Energy 269 (2020) 115058

[25] P. Dolgopolov, D. Konstantinov, L. Rybalchenko, R. Muhitovs. (2019). Optimization of train routes based on neuro-fuzzy modeling and genetic algorithms. Elsevier, Procedia Computer Science 149 (2019) 11–18

[26] H. Wang , X. Lang, W. Mao. (2021). Voyage optimization combining genetic algorithm and dynamic programming for fuel/emissions reduction. Elsevier, Transportation Research Part D 90 (2021) 102670

[27] M. Z. Abd Elrehim, M. A. Eid, M. G. Sayed. (2019). Structural optimization of concrete arch bridges using Genetic Algorithms. Elsevier, Ain Shams Engineering Journal 10 (2019) 507–516

[28] I. Jocsak, G. Vegvari, E. Vozary. (2019) Electrical impedance measurement on plants: a review with some insights to other fields. Theor. Exp. Plant Physiol. (2019) 31:359–375

[29] T. Ivanovski, Z. Guoxiang, T. Jemrić, M. Gulić, M. Matetić. (2020). Fruit firmness prediction using multiple linear regression. Proceedings of the 43rd International Convention MIPRO, (2020) 1570-1575

[30] Pietro Ibba, Aniello Falco, Biresaw Demelash Abera, Giuseppe Cantarella, Luisa Petti, Paolo Lugli. (2020). Bio-impedance and circuit parameters: An analysis for tracking fruit ripening. Postharvest Biology and Technology 159 (2020) 110978

[31] Marco Grossi, Bruno Riccò. (2017) Electrical impedance spectroscopy (EIS) for biological analysis and food characterization: a review. J. Sens. Sens. Syst., 6, 303–325, 2017

[32] G. M. Weaver, H.O. Jackson. (1966). Electric impedance, in objective index of maturity in peach. Canadian journal of plant science Vol. 46

[33] F. Roger Harker, John H. Maindonald. (1994). Ripening of Nectarine Fruit Changes in the Cell Wall, Vacuole, and Membranes Detected Using Electrical Impedance Measurements. Plant Physiol. (1994) 106: 165-171

[34] Galit Shmueli, Peter C. Bruce, Inbal Yahov, Nitin R. Patel, Kenneth C. Lichtendahl Jr. (2018). Data mining for business analytics. John Wiley & Sons, Inc. (2018)

[35] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. (2017). An Introduction to Statistical Learning with Applications in R. Springer Science+Business Media New York 2013 (Corrected at the printing 2017)

[36] Xin-She Yang. (2013). Bat Algorithm: Literature Review and Applications. International Journal of Bio-Inspired Computation, August 2013. DOI: 10.1504/IJBIC.2013.055093

[37] M. F. Ahmad, N. A. Mat Isa, W. H. Lim, K. M. Ang. (2022). Differential evolution: A recent review based on state-of-the-art works. Elsevier, Alexandria Engineering Journal (2022) 61, 3831–3872

[38] Ke-Lin Du, M.N.S. Swamy. (2016). Search and Optimization by Metaheuristics - Techniques and Algorithms Inspired by Nature. Birkhäuser (2016)

[39] S. Sengupta, S. Basak, R. A. Peters. (2019). Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives. Machine learning and knowledge extraction 2019, 1, 157–191; doi:10.3390

[40] Raspudic, V. (2020). Optimal Design of Laterally Unrestrained I-Beams Using Genetic Algorithm, Proceedings of the 31st DAAAM International Symposium, pp.0683-0691, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-29-7, ISSN 1726-9679, Vienna, Austria DOI: 10.2507/31st.daaam.proceedings.095