Application Analysis of Supervised and Unsupervised Methods for Clustering and Classification of a Wheat Seeds Dataset

Mirella M. de O. Carneiro [*] **Milena F. Pinto** [**]
Alessandro R. L. Zachi [**]

[*] *Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Brazil, (e-mail: mirella.carneiro@coppe.ufrj.br)*
[**] *Federal Center for Technological Education of Rio de Janeiro, Rio de Janeiro, RJ , Brazil, (e-mail: milena.pinto@cefet-rj.br)(e-mail: alessandro.zachi@cefet-rj.br)*

Abstract: This project aims to apply easy-to-implement supervised and unsupervised learning methods to do a cluster and classification analysis of the information from a wheat seeds dataset. In addition, it intends to thoroughly evaluate, employing post-processing techniques, the efficiency of the models produced from them, proving that it is not necessary to use complex procedures in this database, since the obtained results for clustering and classification were highly similar to the dataset original labels. The decision tree was chosen as the classification algorithm. Furthermore, *k-means* and Kohonen neural network were selected as the clustering methods. Additionally, pre-processing and exploratory data analysis techniques were explained in detail and employed in order to maximize the final quality of the model developed.

Resumo: O objetivo deste trabalho é aplicar métodos de aprendizagem supervisionada e não supervisionada de fácil implementação com o intuito de agrupar e classificar informações pertencentes a um banco de dados de sementes de trigo, e avaliar minuciosamente, por meio de técnicas de pós-processamento, a eficiência dos modelos produzidos a partir dos mesmos, provando que não é necessário a utilização de procedimentos complexos nessa base de dados, visto que os resultados obtidos com o agrupamento e classificação foram bastante similares aos *labels* originais da mesma. O algoritmo de árvore de decisão foi escolhido como o classificador utilizado e o *k-means* e a rede neural de Kohonen foram os métodos de *clusterização* selecionados. Além disso, técnicas de pré-processamento e análise exploratória dos dados foram explicadas em detalhes e aplicadas com a intenção de maximizar a qualidade final do modelo desenvolvido.

Keywords: Supervised Learning, Decision Tree, Unsupervised Learning, *K-Means*, Kohonen Neural Network, Wheat Seeds.

Palavras-chaves: Aprendizado Supervisionado, Árvore de Decisão, Aprendizado Não Supervisionado, *K-Means*, Rede Neural de Kohonen, Sementes de Trigo.

1. INTRODUCTION

Machine learning is applied in many research fields, as stated in Bevilacqua et al. (2006), Lima and Minussi (2011), Bianchi et al. (2007). However, it is being largely implemented in agriculture too, as shown in Wei et al. (2020), Elmetwalli et al. (2022), Yao et al. (2020). Agriculture satisfies one of the humans' primary demands, in other words, foodstuff. Furthermore, it is one of the world's economic needful pillars. Taking Brazil into consideration, which is an agricultural power, employing machine learning in this field is extremely important in order to improve the gain, allowing the producers to keep track of its steps precisely, which are pre-harvesting, harvesting, and post-harvesting tasks (Meshram et al. (2021)).

Some examples of pre-harvesting are seeds quality, disease detection, and environmental conditions. Post-harvesting activities include gases employed in fruit containers, seed handling processes to preserve quality, and others. Harvesting exemplifications includes determining crop characteristics like maturity stage, size, detection and classification (Prange (2010)). The second step of the process will be dealt with in this project, precisely wheat seeds detection and classification, because of a huge issue around wheat seeds is that occasionally their varieties look so similar that characterizing them turns into an exhausting task when executed manually (Punn and Bhalla (2013)).

One of the unsupervised learning algorithms applied in this project was *k-means*, which partition the observations into clusters with similar characteristics, taking into account pre-determined criteria, allowing to discover patterns between data. In this algorithm, the number of clusters must be specified preliminarily. For that, some techniques can be employed, such as the elbow method or the silhouette method, to discover the natural number of clusters. An important observation about *k-means* is the fact that it is

ISSN: 2525-8311 DOI: 10.20906/CBA2022/3508

an algorithm of easy implementation and interpretation, taking into account other methods. In addition, it is also computationally attractive (Charytanowicz et al. (2010)).

The other unsupervised learning algorithm employed with the aim of clustering data was the Kohonen neural network that reproduces the organization of the cerebral cortex, which is able to learn from experience (Lima and Minussi (2011)). Kohonen neural network is composed of two layers, the input, and the competition layer. In this case, the neuron is a similarity meter. Moreover, the competitive learning methodology “winner-take-all” is used since only the winner neuron is trained (Bevilacqua et al. (2006)). The network’s output consists of a vector of weights linked to each of the neurons.

The classifier employed in this work was the decision tree, which consists of a supervised learning algorithm widely applied in data classification and prediction problems. The classifier uses data samples characteristics in order to delimit information in a set of rules that can be carried out to make choice-generating decision guidelines in a tree model (Escovedo and Koshiyama (2020)).

This work used a dataset named Seeds Data Set, obtained from UCI Machine Learning Repository [1]. The selected wheat seeds database has about 1400 elements and seven attributes, thus being considered a small database. The attributes consist of the geometric characteristics of the internal part of the wheat grains, which were obtained through a high-quality visualization of the inner kernel structure, using a soft X-ray technique (Charytanowicz et al. (2010)). The wheat grains were obtained from experimental fields exploited at the Institute of Agrophysics of the Polish Academy of Sciences in Lublin [2].

It is critical to point out that in many articles published in the literature involving machine learning, pre-processing techniques are not performed or not presented/explained, as shown in (Yao et al. (2020); Bianchi et al. (2007); Charytanowicz et al. (2010)). Nevertheless, in many machine learning applications, this step can improve the results.

The purpose of this project is to find out how many varieties of wheat seeds are present in the set of observations obtained. Moreover, it is also necessary to predict to which cluster a new element that get in the database belongs. In order to achieve these objectives, two methods of unsupervised learning were applied and the results provided by the algorithms were compared. A classifier, which consists of a supervised learning algorithm, was also used alongside exploratory data analysis and different pre and post-processing techniques, which will be discussed in details later on so as to provide a complete understanding about the procedures carried out.

2. METHODOLOGY

A general overview of the proposed methodology is shown in Figure 1. As can be seen, an exploratory data analysis was employed to summarize the main characteristics of the database through graphs, tables, or numerical measures, making it simpler to identify abnormal behaviors, detect patterns, or verify hypotheses. After the first step, some pre-processing techniques were performed as long as this step is extremely relevant to the final quality of the model developed. Pre-processing is responsible for tasks that include preparation, organization, and structuring the observations.

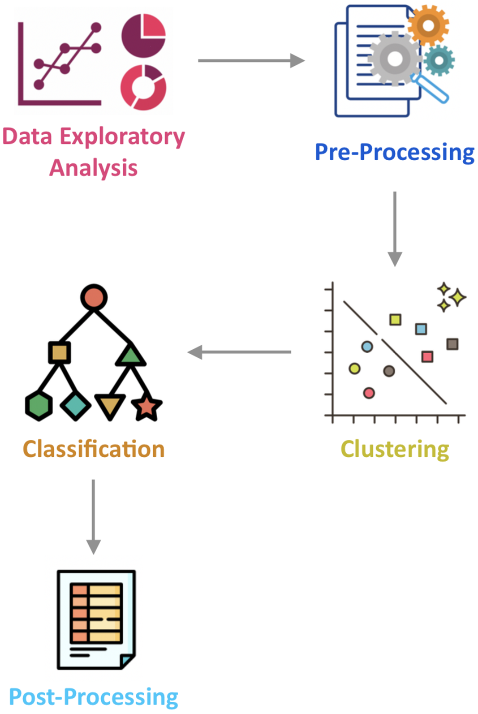

Figure 1. Proposed methodology.

Subsequently, the clustering process was carried out employing *k-means* algorithm and the one-dimensional Kohonen neural network, in which the database was clustered into different classes. It is needful to highlight that intending to know the optimal number of classes, the elbow method was used.

After the cluster analysis, the decision tree classifier was applied, as after employing *k-means* or Kohonen neural network, the clusters are known and are provided to the classifier along with the input variables. The classifier aims to predict a cluster related to an input variable with specific characteristics. In order to finish, data post-processing techniques were performed. This stage is responsible for evaluating the quality of the model developed. The entire project was programmed in *Python* using PyCharm IDE.

2.1 Dataset and Libraries

The database has 1470 elements and seven attributes related to the geometric characteristics of wheat kernels that are defined by: Area, Perimeter, Compactness, Kernel Length, Kernel Width, Asymmetry Coefficient, and Kernel Groove Length. All the values of these parameters are numeric, continuous, and real.

The dataset already came with its respective labels. However, they were not used since unsupervised learning methods are being carried out. These labels will be used only to validate the results achieved.

[1] http://archive.ics.uci.edu/ml/datasets/seeds
[2] https://www.ipan.lublin.pl/en/

Some *Python* libraries were employed in order to deal with the data and producing the model. The packages used in this project were:

- Pandas: Data analysis.
- NumPy: Working with arrays.
- Seaborn and Matplotlib: Data visualization using graphs.
- Scikit-learn: Module for applying machine learning.

2.2 Exploratory Data Analysis

Missing Values: An important thing to point out about this database is the fact that there are no missing observations. Thus, it is not necessary to use techniques that deal with missing observations, such as replacing with mean/mode/median of the parameter or removing rows/columns if an attribute has missing values, among other procedures (Escovedo and Koshiyama (2020)). However, it is indispensable to emphasize that each method has pros and cons. It is decisive to analyze the selected database and choose the best strategy to be applied. Note that each dataset variable has got 210 observations.

Outliers: The boxplots of the seven attributes were analyzed in order to check if there are possible outliers in the dataset (Escovedo and Koshiyama (2020)).

It was decided not to eliminate the few elements that could be potential anomalies, since when they were removed, taking into account that 95% of the data are within two standard deviations of the mean. Any element outside this range was considered a possible outlier. This action also removed crucial information related to other variables from the database.

Correlated Features: By means of the Pearson correlation coefficients of the variables (Brownlee (2016)), shown in Figure 2, it is possible to conclude that there is a strong correlation between several attributes, which is undesirable, since highly correlated variables only add redundant information to the database, causing unnecessary memory occupation and slowing down the algorithm.

Statistical Analysis: Since unsupervised learning algorithms are being performed and clusters' labels are not being taken into consideration, there is no way to apply techniques, like the correlation matrix between input/output variables, which provide the importance of each feature for the classification. So, it is fundamental to carry out a graphical analysis of the parameters to investigate which contributes the most to the correct data classification.

The probability density function of each dataset feature was plotted and is shown in Figure 3. It is feasible to conclude that the attributes with well-differentiated modes are Area, Perimeter, Kernel Length, and Kernel Groove Length. Thus, it is possible to assume that such parameters may be the ones that most contribute to the classification.

Furthermore, associating variables' modes makes it viable to have an idea of the number of initial centroids. However, another methodology was applied for this purpose.

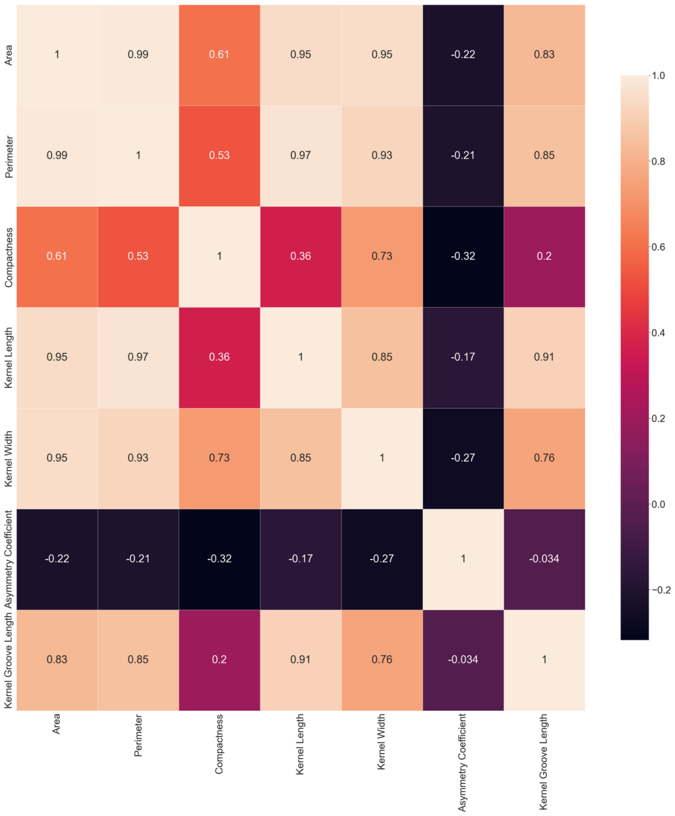

Figure 2. Pearson correlation coefficients of the attributes.

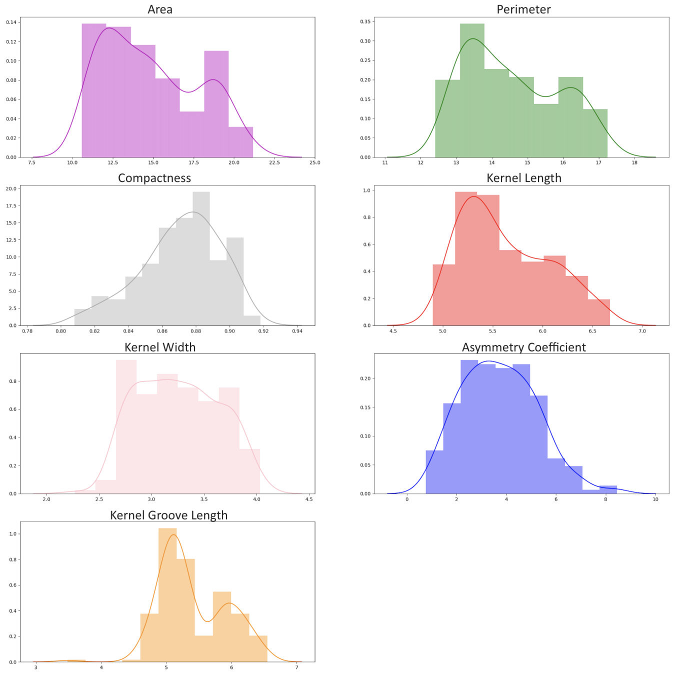

Figure 3. Probability density function for each parameter.

2.3 Pre-Processing

Data Standardization: Each feature has different scales. Therefore, with the intention that this discrepancy between the scales does not influence the cluster analysis methods being used, it is needful to standardize data. Additionally, taking into account principal components analysis, with the purpose of not giving greater importance to one attribute than to another, leading to an erroneous assessment of which parameters most contribute to the representation of the database, it is crucial to perform data standardization technique.

This procedure is extremely relevant since it is employed to transform variables that hold a Gaussian distribution and standard deviations and unequal means into a Gaussian distribution with a standard deviation of 1 and mean 0 (Brownlee (2016)).

To illustrate how distinct variables' scales can influence the results obtained in the diverse processes applied to the database, Table 1 presents the values of each parameter variance ratio in principal components analysis before and after data standardization.

Table 1. Variance ratio for each feature before and after data standardization.

	Before	After
Area	82.67%	70.96%
Perimeter	16.41%	16.84%
Compactness	0.67%	9.92%
Kernel Length	0.18%	1.57%
Kernel Width	0.04%	0.49%
Asymmetry Coefficient	0.02%	0.19%
Kernel Groove Length	0.0002%	0.01%

Principal Component Analysis: A set of features, initially correlated with each other, is linearly transformed into a considerably smaller set of uncorrelated variables, which hold most of the information of the original dataset, through the statistical technique of principal component analysis (Jolliffe and Cadima (2016)).

The statistical analysis carried out in Section 2.3 concluded that Area, Perimeter, Kernel Length, and Kernel Groove Length are the attributes that probably, most contribute to the classification. Nonetheless, instead of selecting this approach to settle on which features will be used, it was chosen to perform the principal component analysis method and, later, analyze the classifier's performance employing this procedure.

The advantage of applying this technique, instead of the features selecting method, is that it is possible to represent most of the relevant material by choosing the principal components with the greatest variability. It is important to remember that all the components have information about each parameter. Nevertheless, when performing the second procedure, some variables are entirely removed from the database, and all of their corresponding material is lost.

By analyzing the graph of Figure 4, it was found that the principal components which explain 99% of the original dataset are 0, 1, 2, and 3.

The variance ratio is defined by: the variance of each principal component/variance sum of all principal components. Features' scatter plots from the new dataset, which has only the principal components that in total represent 99% of the original database, are shown in Figure 5. It is plausible to verify that after applying this method, dataset size was reduced, and the data is uncorrelated. Figure 6 shows the correlation between each principal component and the initial database attributes.

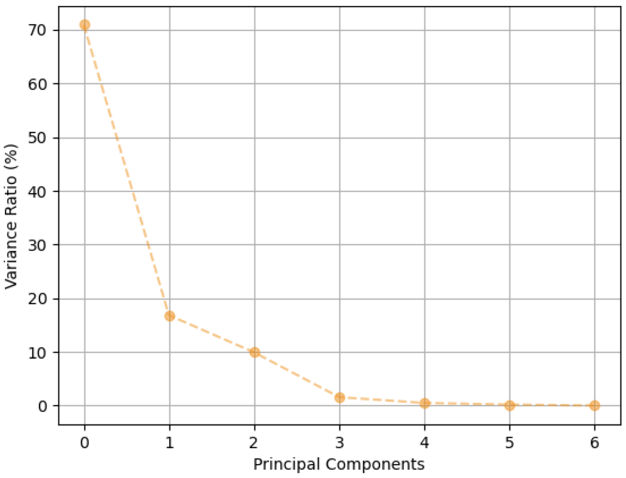

Figure 4. Principal components variance ratio.

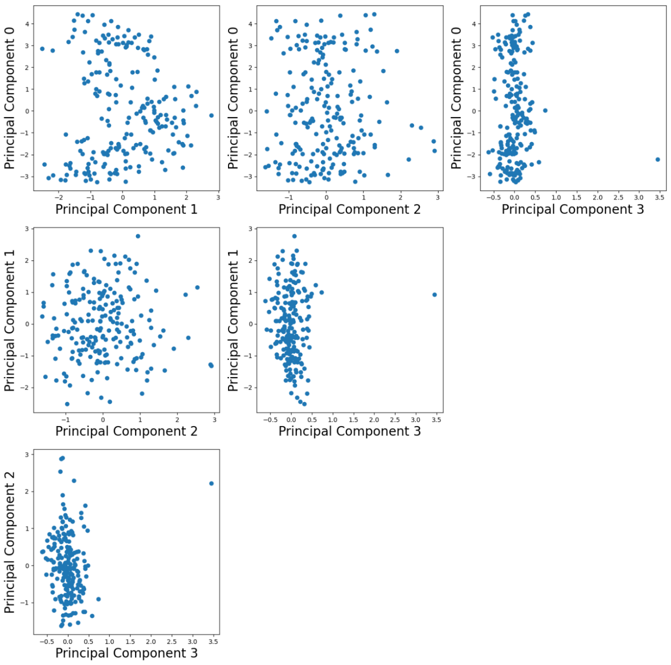

Figure 5. Scatter plots of 0, 1, 2 and 3 principal components.

2.4 Clustering

As discussed earlier, one of the unsupervised learning algorithms used in this work was *k-means*. This algorithm is flexbile, fast and easy-to-implement. *K-means* measures and compares the Euclidean distance between the database elements and the center of the classes, relating each element to the cluster that configure the smallest distance. The centroid of a cluster is the arithmetic mean of all observations that belong to it.

Taking into account *k-means*, it is necessary to insert the desired number of classes as a hyperparameter. So, the elbow method was applied to find the optimal number of classes to obtain a good result after cluster analysis.

The elbow method consists of calculating the total within-cluster sum of squares, considering different numbers of classes. The optimal number of clusters is reached in the

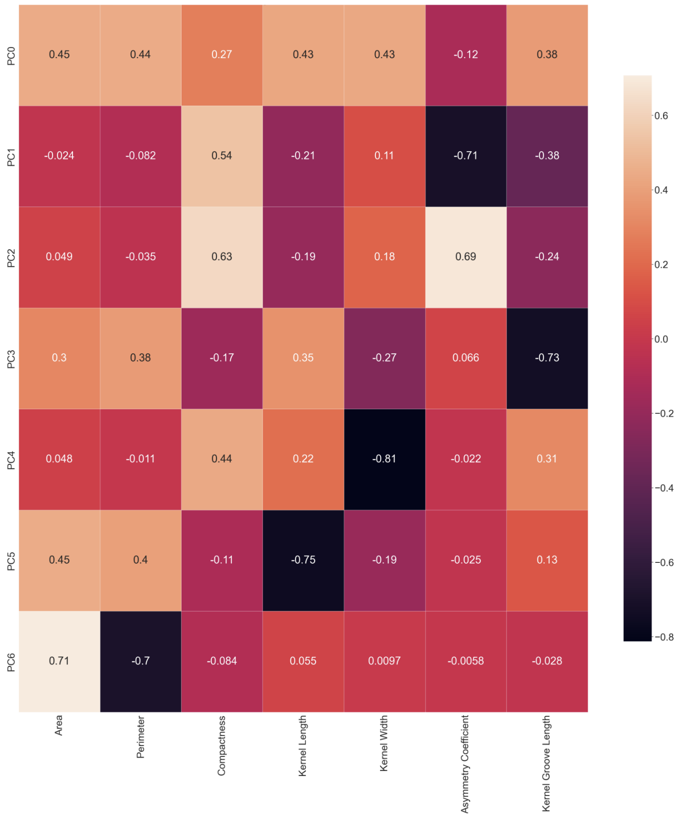

Figure 6. Correlation between principal components and features.

immediate vicinity of a wide variation of the total within-cluster sum of squares. Thus, analyzing Figure 7, it is plausible to conclude that the optimal number of classes is three.

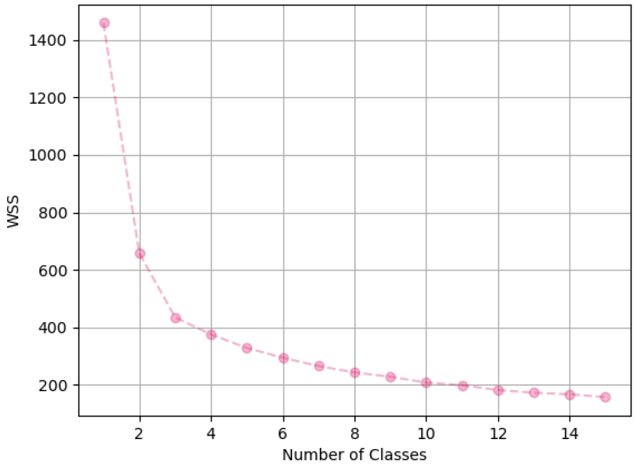

Figure 7. Elbow method.

The other unsupervised learning method carried out was Kohonen neural network. In this algorithm, each neuron of the competitive layer is associated with a weight vector, which is modified throughout the learning process, and whose dimension is equal to the number of variables that will be used in the classification, i.e., four. The characteristic vector of a specific subset of data is characterized by the weight vector connected to each neuron (Bianchi et al. (2007)). Input variables and synapses are defined in the input layer, where the competition occurs.

In the one-dimensional Kohonen neural network simplified algorithm (Bianchi et al. (2007)) applied in this work, the training procedure comprises searching for the neuron which the Euclidean distance between the input vector and its weight vector is the smallest among all neurons in the layer, as shown in Equation 1. N is the number of neurons. $\boldsymbol{w_i}$ is defined as the weight vector of the winning neuron. $\boldsymbol{x}$ is the input vector. $\boldsymbol{w_j}$ represents the weight vector of the other neurons in the layer (Bevilacqua et al. (2006)).

$$||\boldsymbol{w_i} - \boldsymbol{x}|| \leq ||\boldsymbol{w_j} - \boldsymbol{x}||, \forall j = 1...N \tag{1}$$

The next step, after determining the winning neuron, is to update its weight vector, according to Equation 2. The variables $\alpha \in [0,1]$ represents the learning rate, which decreases monotonically over time, n denotes the current state and $n+1$ the next state (Bianchi et al. (2007)).

$$\boldsymbol{w_i}(n+1) = \boldsymbol{w_i}(n) + \alpha(n)[\boldsymbol{x}(n) - \boldsymbol{w_i}(n)] \tag{2}$$

It is relevant to mention that α and N are parameters that must be defined preliminarily. The weight vector of the neurons located in the winning neuron neighborhood will remain with the same values, as shown in Equation 3.

$$\boldsymbol{w_j}(n+1) = \boldsymbol{w_j}(n), \forall j \neq i \tag{3}$$

2.5 Classifier

The decision tree classifier is fast, simple to interpret and visualize, besides, it is more efficient than some classification algorithms like k-nearest neighbor. Decision tree algorithm creates a binary tree from the training data. By means of checking each feature and each attribute value in the training data, to minimize a cost function, like Gini Index, split points are selected (Brownlee (2016)). In this project, Gini Index was applied as a splitting criteria. It measures the degree of data heterogeneity, as a consequence, it can be used with the intention of measuring the impurity of a node (Onoda and Ebecken (2001)). The Gini Index of a certain node is defined by Equation 4.

$$Gini = 1 - \sum_{j=1}^{c} p_j^2 \tag{4}$$

where c is the number of classes and p_j is the proportion of samples that belong to a class c for a determined node.

When the index approaches the value 1, the node is considered impure (the quantity of equally distributed classes in this node expands). The moment it is equal to 0, the node can be defined as pure. The decision tree algorithm is very prone to be affected by overfitting, which occurs when the training data is classified perfectly. However, the predictive process applied to the test data will be degraded. Intending to avoid this problem, it was necessary to specify the maximum depth of the tree as a hyperparameter, which was set to a value of 6.

Nevertheless, other parameters can also be defined for this purpose.

2.6 Post-Processing

Classification Report: In order to determine the quality of the predictions made by a classification algorithm, the classification report is used. This metric, along with other model validation metrics, can narrowly analyze the classifier's efficiency.

This report provides the parameters of accuracy, precision, recall, and f1-score of each class formed. These metrics are found using false positive (FP), true positive (TP), false negative (FN), and true negative (TN) values.

The accuracy gives how many elements the classification model correctly classified concerning the total of data that were classified. To summarize, this parameter measures the classifier's overall performance and is defined by Equation 5.

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Precision indicates how many correct positive classifications the classifier performed, among all the elements considered positive. Equation 6 determines this metric.

$$P = \frac{TP}{TP + FP} \tag{6}$$

Recall is responsible for pointing out the relationship between correct positive predictions and all the predictions that are really positive. This parameter is given by Equation 7.

$$R = \frac{TP}{TP + FN} \tag{7}$$

The f1-score metric is calculated from the harmonic mean of recall and precision values, as shown in Equation 8.

$$F = 2 * \frac{R * P}{R + P} \tag{8}$$

Confusion Matrix: It is a metric that aims to calculate class frequency distribution, in other words, the number of FPs, TPs, FNs and TNs (Escovedo and Koshiyama (2020)). It is possible to check the number of times the classifier got the predictions right and wrong.

Cross Validation: This procedure is carried out with the purpose of examining the classifier generalization ability given a certain database. To sum up, cross-validation evaluates the performance of the model for new observations, investigating how accurate the model is in practice.

First, the hold-out validation technique was employed, in which data is split into training and test sets. The problem with this procedure is that the algorithm randomly selects data within the defined percentages, and it can occur that very similar training and test samples may be chosen, which will lead to a good evaluation of the classifier. Nonetheless, the classification results will be unsatisfactory when the observations are very dissimilar from the training set applied.

The k-fold cross-validation method is necessary since data is split into k subsets, and in each iteration, a data portion is chosen to be the training and test set. This process continues until every k fold in the dataset is selected for testing. After completing the process, k values of accuracy are provided in a table, including their mean and standard deviation (Brownlee (2016)).

3. RESULTS AND DISCUSSION

3.1 K-means

After using the elbow method, the *k-means* algorithm was applied. Figure 8 shows this result. Note that the classes are well defined. Moreover, the furthest point that is present in some of the graphs will not be removed because of it is not an anomaly.

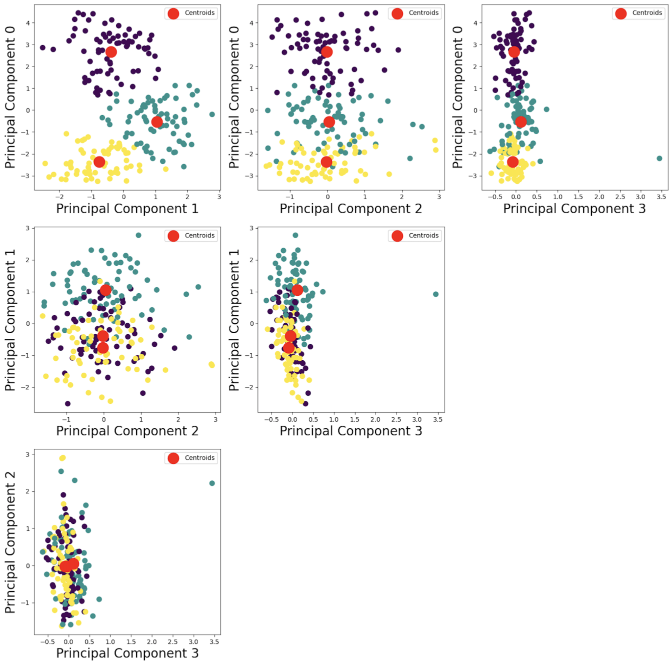

Figure 8. Created classes and their respective centroids.

Considering the original labels that came with the database, it has three classes, as predicted by the elbow method performed in this project. It is conceivable to conclude, analyzing the graph of Figure 9, that the number of elements per class obtained with *k-means* is very close to what was expected, considering the original labels from the database. The original clusters of the dataset have 70 elements, and according to the result obtained in *k-means*, each of the classes 0, 1, and 2 have 72, 73, and 65 elements, respectively, which shows that the algorithm grouped data satisfactorily.

After employing the decision tree classifier, with a percentage of training data defined as 70%, and 30% for testing, the classification report was generated and is presented in Table 2.

Cross-validation was also employed, with a number of k folds equal to 5. Table 3 presents the accuracy of each

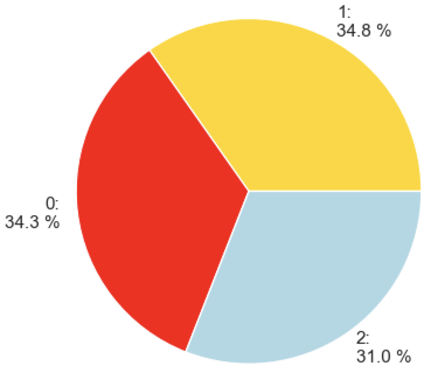

Figure 9. Percentage of elements per cluster.

Table 2. Classification report.

	Precision	Recall	F1-score	Accuracy
0	100%	100%	100%	95.24%
1	92%	96%	94%	
2	95%	91%	93%	

Table 3. k folds metrics.

	$K1$	$K2$	$K3$	$K4$	$K5$
Accuracy	93.33%	96.67%	100%	93.10%	86.21%
Mean	94%				
Standard Deviation	4.6%				

iteration performed in the algorithm in addition to the mean and standard deviation.

Finally, the confusion matrix was generated, which is in Figure 10.

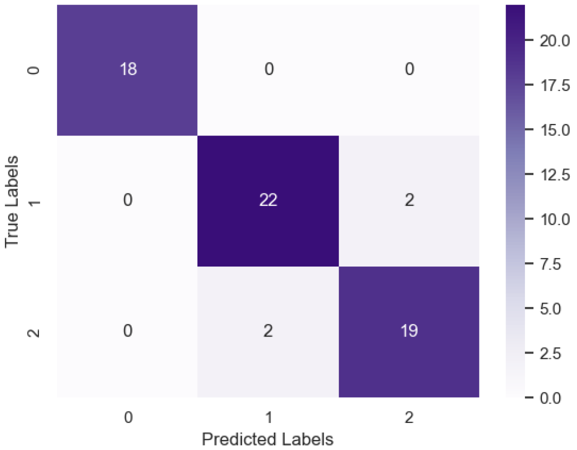

Figure 10. Confusion matrix.

According to the confusion matrix, the number of correct classifications was 59 out of 63 predictions made, proving the efficiency of the classifier used.

3.2 Kohonen Neural Network

Three weight vectors were initially defined for the Kohonen neural network algorithm, and the result is shown in Figure 11. It is possible to identify that the classes were well defined, as occurred applying *k-means*.

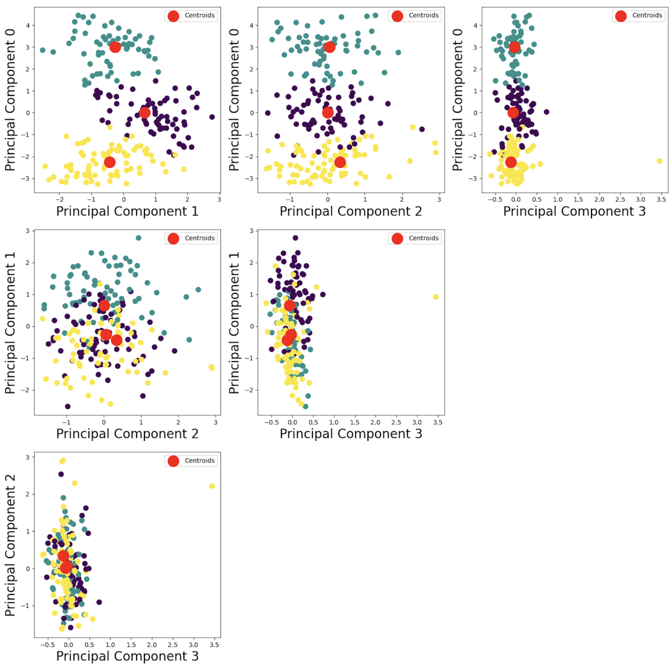

Figure 11. Created clusters and their respective centroids.

Table 4. Classification report.

	Precision	Recall	F1-score	Accuracy
3	90%	100%	95%	96.83%
4	100%	100%	100%	
5	100%	93%	96%	

It is possible to verify in Figure 12 that the number of elements per class obtained was very close to the expected, as occurred in the *k-means* algorithm.

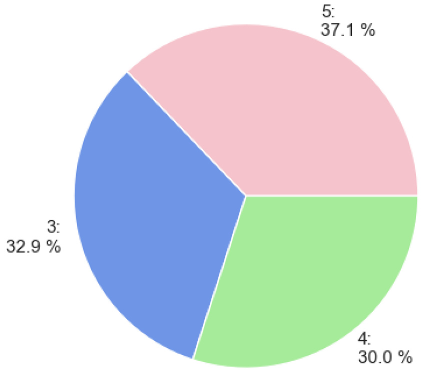

Figure 12. Percentage of elements per class.

According to the result obtained in the Kohonen neural network, each of the classes 3, 4, and 5 have 69, 63, and 78 elements, respectively, showing that the algorithm grouped them properly. However, applying *k-means*, the number of elements per cluster was a little closer to the number of elements per class when taking into consideration the original labels of the database.

After using the decision tree classifier, with a percentage of the dataset for training as 70%, and 30% for testing, the classification report was generated, which is shown in Table 4.

Table 5. k folds metrics.

	$K1$	$K2$	$K3$	$K4$	$K5$
Accuracy	100%	100%	93.10%	93.10%	93.10%
Mean	96%				
Standard Deviation	3.4%				

Cross-validation was applied, and the number of k folds selected was equal to 5. Table 5 shows the accuracy of each iteration performed in the algorithm in addition to the mean and standard deviation.

At last, Figure 13 presents the confusion matrix generated.

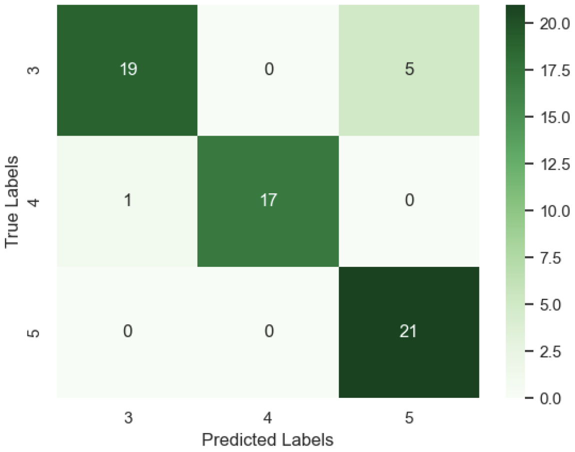

Figure 13. Confusion Matrix.

By analyzing the confusion matrix, it can be assumed that the number of correct classifications was 57, out of a total of 63 predictions made, providing a satisfactory result.

4. CONCLUSIONS AND FUTURE WORKS

The results obtained make it practicable to verify that the classifier adopted in this work provided efficient results since the post-processing metrics used to evaluate it offered promising results. The pre-processing techniques also contributed significantly to the effectiveness of the model. The two unsupervised learning methods provided similar outcomes after clustering and classification. Moreover, they were compatible with the original labels of the database, validating the entire project developed.

Future works comprehend the need to select a more complex database involving wheat seeds to cluster and classify it, carrying out simple interpretation and implementation unsupervised learning algorithms, which leads to highly relevant results similar to the ones achieved in this project.

ACKNOWLEDGMENT

The authors would like to thank CEFET/RJ, UFRJ and the brazilian research agencies CAPES, CNPq and FAPERJ for supporting this work.

REFERENCES

Bevilacqua, V., Mastronardi, G., and Marinelli, M. (2006). A neural network approach to medical image segmentation and three-dimensional reconstruction. In International Conference on Intelligent Computing, 22–31. Springer.

Bianchi, D., Calogero, R., and Tirozzi, B. (2007). Kohonen neural networks and genetic classification. Mathematical and Computer Modelling, 45(1-2), 34–60.

Brownlee, J. (2016). Machine learning mastery with python. Machine Learning Mastery Pty Ltd, 527, 100–120.

Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P.A., Łukasik, S., and Żak, S. (2010). Complete gradient clustering algorithm for features analysis of x-ray images. In Information technologies in biomedicine, 15–24. Springer.

Elmetwalli, A.H., Mazrou, Y.S., Tyler, A.N., Hunter, P.D., Elsherbiny, O., Yaseen, Z.M., and Elsayed, S. (2022). Assessing the efficiency of remote sensing and machine learning algorithms to quantify wheat characteristics in the nile delta region of egypt. Agriculture, 12(3), 332.

Escovedo, T. and Koshiyama, A. (2020). Introdução a Data Science: Algoritmos de Machine Learning e métodos de análise. Casa do Código.

Jolliffe, I.T. and Cadima, J. (2016). Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202.

Lima, F.P.d.A. and Minussi, C.R. (2011). Usando uma rede neural de kohonen para reconhecimento de padrões de som. Revista OMNIA Exatas, 4(2), 19–30.

Meshram, V., Patil, K., Meshram, V., Hanchate, D., and Ramkteke, S. (2021). Machine learning in agriculture domain: a state-of-art survey. Artificial Intelligence in the Life Sciences, 1, 100010.

Onoda, M. and Ebecken, N.F. (2001). Implementação em java de um algoritmo de árvore de decisão acoplado a um sgbd relacional. In SBBD, 55–64.

Prange, R. (2010). Pre-harvest, harvest and post-harvest strategies for organic production of fruits and vegetables. In XXVIII International Horticultural Congress on Science and Horticulture for People (IHC2010): International Symposium on 933, 43–50.

Punn, M. and Bhalla, N. (2013). Classification of wheat grains using machine algorithms. International Journal of Science and Research (IJSR), 2(8), 363–366.

Wei, W., YANG, T.l., Rui, L., Chen, C., Tao, L., Kai, Z., SUN, C.m., LI, C.y., ZHU, X.k., and GUO, W.s. (2020). Detection and enumeration of wheat grains based on a deep learning method under various scenarios and scales. Journal of Integrative Agriculture, 19(8), 1998–2008.

Yao, Y., Li, Y., Jiang, B., and Chen, H. (2020). Multiple kernel k-means clustering by selecting representative kernels. IEEE Transactions on Neural Networks and Learning Systems, 32(11), 4983–4996.