COMPARATIVE ANALYSIS OF SUPERVISED MACHINE LEARNING ALGORITHMS FOR HEART DISEASE DETECTION

Hector Daniel Huapaya
Member of the artificial intelligence research group of the faculty of systems engineering, department of software engineering at the National University Mayor de San Marcos, Lima, (Perú).
E-mail: hector.huapaya@unmsm.edu.pe ORCID: https://orcid.org/0000-0003-3616-9046

Ciro Rodriguez
Professor at the School of Software Engineering at the National University Mayor de San Marcos, Lima, (Perú).
E-mail: crodriguezro@unmsm.edu.pe ORCID: https://orcid.org/0000-0003-2112-1349

Doris Esenarro
Professor at the Faculty of Environmental Engineering and Graduate School of the National University Federico Villarreal, Lima, (Perú).
E-mail: desenarro@unfv.edu.pe ORCID: https://orcid.org/0000-0002-7186-9614

Recepción: 27/12/2019 **Aceptación:** 30/03/2020 **Publicación:** 30/04/2020

Citación sugerida Suggested citation

Hector Daniel Huapaya, H. D., Rodriguez, C., y Esenarro, D. (2020). Comparative analysis of supervised machine learning algorithms for heart disease detection. *3C Tecnología. Glosas de innovación aplicadas a la pyme. Edición Especial, Abril 2020,* 233-247. http://doi.org/10.17993/3ctecno.2020.specialissue5.233-247

ABSTRACT

This paper describes the most prominent algorithms of Supervised Machine Learning (SML), their characteristics, and comparatives in the way of treating data. The Heart Disease dataset obtained from Kaggle was used to determine and test its highest percentage of accuracy. To achieve the objective, Python sklearn libraries were used to implement the selected algorithms, evaluate and determine which algorithm is the one that obtains the best results, applying decision tree algorithms achieved the best prediction results.

KEYWORDS

Supervised machine learning, Heart disease, Decision tree algorithms, Prediction.

http://doi.org/10.17993/3ctecno.2020.specialissue5.233-247

1. INTRODUCTION

Machine learning is one of the fastest-growing areas of computer science (Srivastava *et al.*, 2014), with long-range applications, which refers to the automatic detection of significant patterns in data with machine learning tools, which give programs the ability to learn and adapt.

Machine learning has become one of the pillars of information technology and, with that, a reasonably central, though generally hidden, part of our life. With the increasing amount of data available, there is a good reason to believe that intelligent data analysis will be even more widespread as a necessary ingredient for technological progress.

There are several applications for Machine Learning (ML), being one of the most important data mining (Bustamante, Rodríguez, & Esenarro, 2019). The handling of a large amount of data makes people more likely to make mistakes during analyzes or, possibly, when trying to establish relationships between multiple characteristics.

Data mining and machine learning go hand in hand with which several ideas can be derived through appropriate learning algorithms. There has been significant progress in data mining and machine learning as a result of the evolution of nanotechnology, which generated curiosity to find hidden patterns in the data to obtain results. The fusion of math and statistics, machine learning and artificial intelligence, information theory and big data, and hight processing computation, has created a reliable science, with a firm mathematical base and compelling tools.

This paper focuses on the classification of ML algorithms and the determination of the most efficient algorithm with the best accuracy and precision. In addition to establishing the performance of different algorithms in large and small datasets with one view, classify them correctly, and provide information on how to build supervised machine learning models.

2. CONCEPTUAL FRAMEWORK

2.1. CLASSIFICATION OF SUPERVISED LEARNING ALGORITHMS

Supervised machine learning algorithms deal more with the classification of data that includes the following algorithms: Linear Classifiers, Logistic Regression, Naive Bayes Classifier, Perceptron, Support Vector Machine; Quadratic classifiers, K-Means grouping, Reinforcement, Decision Tree, Random Forest (RF); Neural networks, Bayesian networks.

1) **Linear Classifiers**: Linear models for classification separate input vectors into classes using linear decision limits (hyperplane). The objective of linear classifiers in machine learning is to group elements that have similar characteristic values into groups (Ray, 2018). A linear classifier achieves this objective by making a classification decision based on the value of the linear combination of the characteristics. A linear classifier is often used in situations where classification speed is a problem since it is classified as the fastest classifier. Besides, linear classifiers often work very well when the number of dimensions is significant, as in the classification of documents, where each element is typically the number of counts of a word in a report. However, the rate of convergence between the variables in the data set depends on the margin. In general terms, the margin quantifies how linearly separable a collection of data is and, therefore, how easy it is to solve a given classification problem.

2) **Naive Bayesian Networks**: These are elementary Bayesian networks that are composed of acyclic graphs directed with a single parent (representing the unobserved node) and several children (corresponding to the observed nodes) with a strong assumption of independence between nodes children in the context of their father. Thus, the independence model (Naive Bayes) is based on the estimate. Bayes classifiers tend to be less accurate than other more sophisticated learning algorithms (such as Artificial Neural Networks). However, in a large-scale comparison of the Bayes naive classifier with state-of-the-art algorithms for decision tree induction, instance-based learning and rule induction in standard reference data sets, and discovered that it is sometimes superior to the other learning schemes, even in data sets with dependencies of substantial characteristics. The Bayes classifier has an attribute independence problem that was addressed with the average estimators of a dependence.

http://doi.org/10.17993/3ctecno.2020.specialissue5.233-247

3) **Support Vector Machines**: This is the most recent supervised machine learning supervised technique. Support vector machine models (SVM) are closely related to classical multilayer perceptron neural networks. SVMs revolve around the notion of a "margin" on each side of a hyperplane that separates two kinds of data. It has been shown that maximizing the margin and, therefore, creating the most significant possible distance between the separation hyperplane and the instances on each side thereof reduces an upper limit on the expected generalization error.

4) **K-means**: It is one of the simplest unsupervised learning algorithms that solve the known clustering problem. The procedure follows a simple and straightforward way to classify a given set of data through a certain number of groups (suppose k groups) set a priori. The K-Means algorithm is used when tagged data is not available (Bhavsar & Ganatra, 2012). General method of conversion approximate general rules into a highly accurate prediction rule. Given the "weak" learning algorithm that you can consistently find classifiers ("general rules") at least slightly better than random, say 55% accuracy, with sufficient data, a reinforcing algorithm can build a single classifier with very high precision, say 99%.

5) **Decree Tree**: Decision trees (DT) are trees that classify instances by ordering them according to characteristic values. Each node in a decision tree represents a characteristic in an example that will be organized, and each branch represents a value that the node can assume. Instances are arranged from the root node and are sorted based on their characteristic values. The decision tree learning, used in data mining and machine learning, uses a decision tree as a predictive model that assigns observations on an element to conclusions about the objective value element.

6) **Neural Networks**: They can perform several regressions and classification tasks at the same time, although commonly, each network performs only one (Sethi *et al.*, 2019). Therefore, in the vast majority of cases, the network will have a single output variable. However, in the case of classification problems of many states, this may correspond to several output units (the post-processing stage is responsible for the assignment of output units to output variables) (Mureșan & Oltean, 2018).

http://doi.org/10.17993/3ctecno.2020.specialissue5.233-247

2.2. CHARACTERISTICS OF MACHINE LEARNING ALGORITHMS

Supervised machine learning techniques are applicable in numerous domains. In general, Support Vector Machines and neural networks tend to work much better when it comes to multidimensional and continuous features (Agarwal & Sagar, 2019). On the other hand, logic-based systems tend to work better when it comes to discrete/categorical features. For neural network models and Support Vector Machines, the large sample size is required to achieve maximum prediction accuracy, while Bayesian networks may need a relatively small data set.

There is a general agreement that the K nearest neighbor algorithm is very sensitive to irrelevant characteristics: this characteristic can be explained by the way the algorithm works. Besides, the presence of irrelevant characteristics can make the training of the neural network very inefficient, even impractical. The most decision tree algorithms cannot work well with problems that require diagonal partitions (Sathya & Abraham, 2013). The division of the instance space is orthogonal to the axis of a variable and parallel to all other axes. Therefore, the resulting regions after separation are all hyper-angles. Artificial neural networks and support vector machines work well when multicollinearity is present, and there is a non-linear relationship between the input and output characteristics.

Naive Bayes (NB) requires little storage space during the training and classification stages: the strict minimum is the memory needed to store prior and conditional probabilities. The basic kNN algorithm uses a large amount of storage space for the training phase (Cao *et al.*, 2019), and its execution space is at least as ample as its training space. On the contrary, for all non-lazy learners, the execution space is usually much smaller than the training space, since the resulting classifier is often a very condensed summary of the data. Besides, Naive Bayes and CNN can easily be used as incremental learners, while rule algorithms cannot. Naive Bayes is naturally robust to missing values since these are ignored in the probabilities of calculation and, therefore, have no impact on the final decision. On the contrary, kNN and neural networks require complete records to do their job.

Finally, the decision trees and NB generally have different operational profiles, when one is very precise, and the other is not, and vice versa. In contrast, decision trees and rule classifiers have a similar operational profile. SVM and ANN also have a similar operational profile. No single learning algorithm can uniformly outperform other algorithms in all data sets.

Different data sets with different types of variables and the number of instances determine the kind of algorithm that will work well (Manzoor & Singla, 2019). There is no single learning algorithm that exceeds other algorithms based on all data sets according to the free lunch theorem. The following table presents a comparative analysis of several learning algorithms.

3. METHODOLOGY

The methodology to determine the best-supervised algorithm applied in the heart disease dataset will begin with the interpretation of the data, the preprocessing of the data, and the application of the algorithms to determine the best accuracy.

A. Dataset

The dataset used for this research will be "Heart Disease" which was found in the Kaggle repository, this database contains 76 attributes, but all published experiments refer to the use of a subset of 14 of them. In particular, the Cleveland database is the only one that ML researchers have used to date. The "goal" field refers to the presence of heart disease in the patient. It has an integer value of 0 (no presence) to 4. Experiments with the Cleveland database have concentrated on the simple attempt to distinguish presence (values 1, 2, 3, 4) from absence (value 0) (Ray, 2018; Sethi *et al.*, 2019; Agarwal & Sagar, 2019).

B. Interpretation of the data

Next, the data extracted is interpreted from the empirically chosen database.

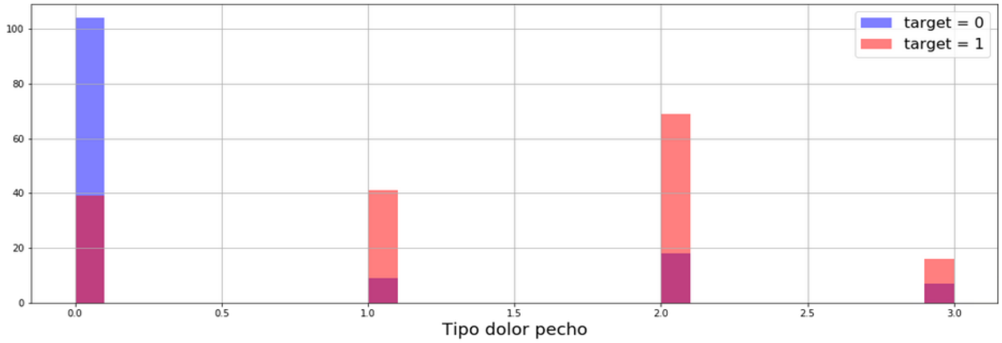


Figure 1. Type of chest pain.

http://doi.org/10.17993/3ctecno.2020.specialissue5.233-247

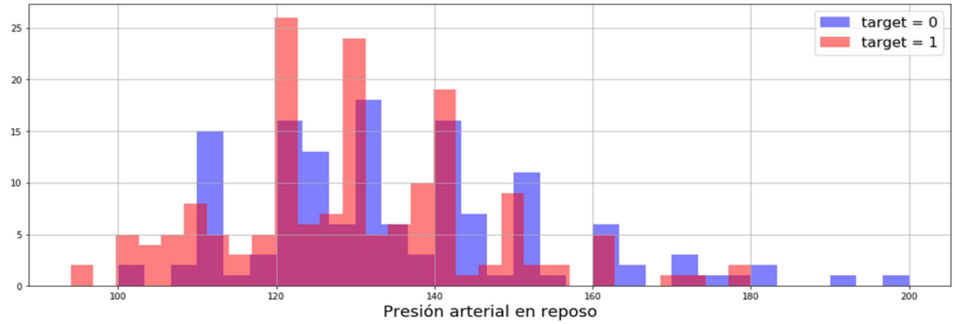


Figure 2. Resting blood pressure.

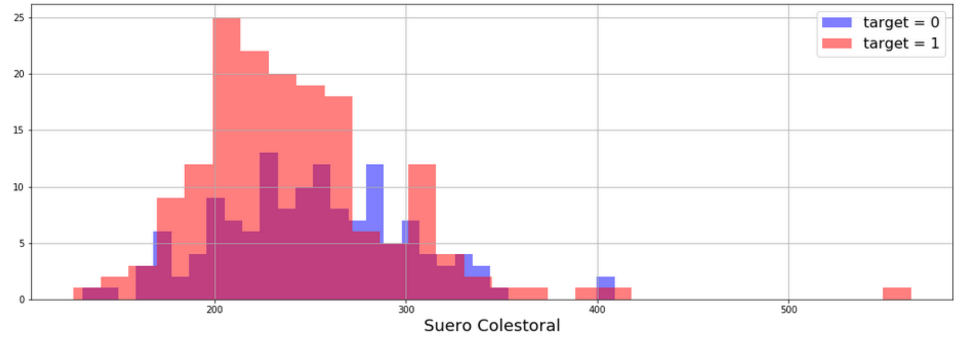


Figure 3. Serum cholesterol.

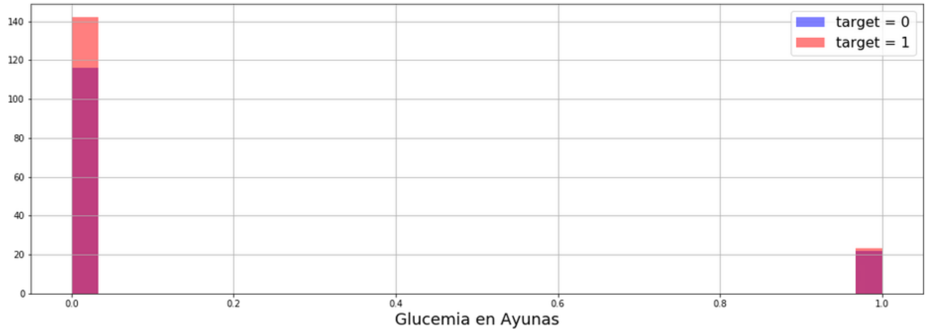


Figure 4. Fasting blood sugar.

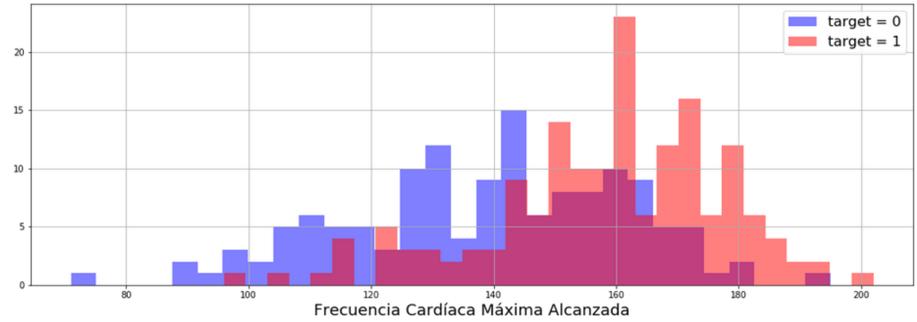


Figure 5. Maximum heart rate reached.

http://doi.org/10.17993/3ctecno.2020.specialissue5.233-247

From the visualization of figures 1,2,3,4, and 5 by category is possible to observe how the data are expressed, which makes it possible to detect if there is a probability of heart disease.

C. Application of algorithms

After understanding the data and interpreting the information to be generated, the following algorithms will be applied.

1. *K Nearest Neighbors (KNN)*

Because the KNN algorithm classifier predicts the class of a given test observation by identifying the observations that are closest to it, the scale of the variables is essential. Any variable that has a large scale will have a much more significant effect on the distance between the observations than the variables that are on a small scale, and therefore on the KNN classifier (Sethi *et al.*, 2019; Agarwal & Sagar, 2019; Cao *et al.*, 2019; Manzoor & Singla, 2019).

After determining the training and test data with the preprocessing processes, let's use the elbow method to choose a good value of K.

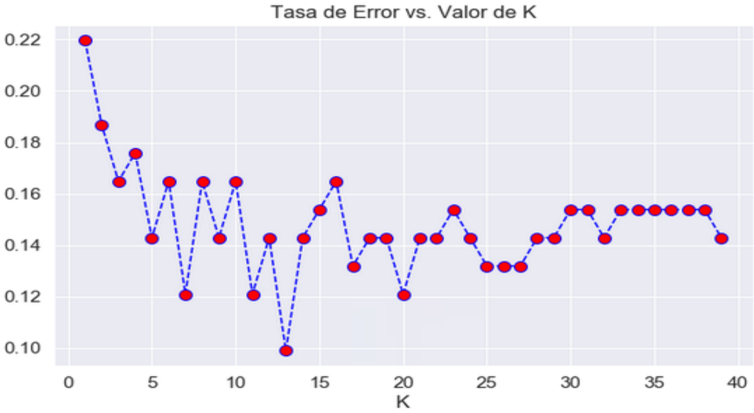


Figure 6. Error rate vs. K-value.

Here we can see the error rate after applying K = 13, let's re-enter the model with this data, and this information is reached.

1. **Decision trees:** The data is divided into a training set and a test set, then a single decision tree will be trained, using the sklearn library, to evaluate the created decision tree.

http://doi.org/10.17993/3ctecno.2020.specialissue5.233-247

2. **Random Forest:** The data is preprocessed, and the training and test variables are separated to train the model.
3. **Neural Network:** The sklearn library will be used to preprocess the data to prepare for training.
4. **Support Vector Machines:** The data is preprocessed to apply the algorithm, the training and test variables are separated; we train the model using the sklearn library.

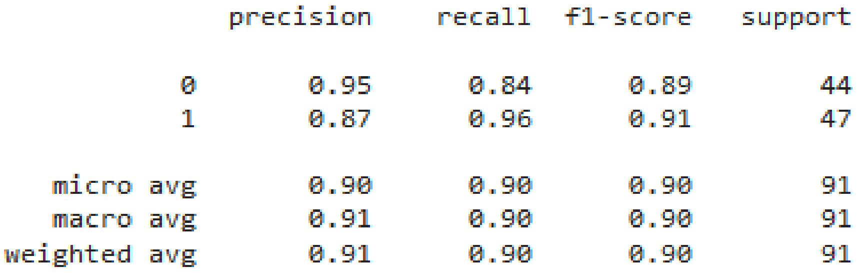
5. RESULTS

After applying the selected supervised learning algorithms to the dataset chosen for comparison, the following algorithm results are obtained.

A. K Nearest Neighbors (KNN)

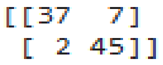
To evaluate the model test data was used to find the confusion matrix, with which we can calculate the accuracy, precision, recall, and f1-score metrics, the following information is available:

Table 1. Result of applying the KNN algorithm.



```
             precision    recall  f1-score   support

           0       0.95      0.84      0.89        44
           1       0.87      0.96      0.91        47

   micro avg       0.90      0.90      0.90        91
   macro avg       0.91      0.90      0.90        91
weighted avg       0.91      0.90      0.90        91
```

Table 1 shows the average weight as 0.91, and the accuracy formula that is the sum of the real positives with the true negatives among the total population is applied, an accuracy of 45,614 is reached, and confusion matrix as :



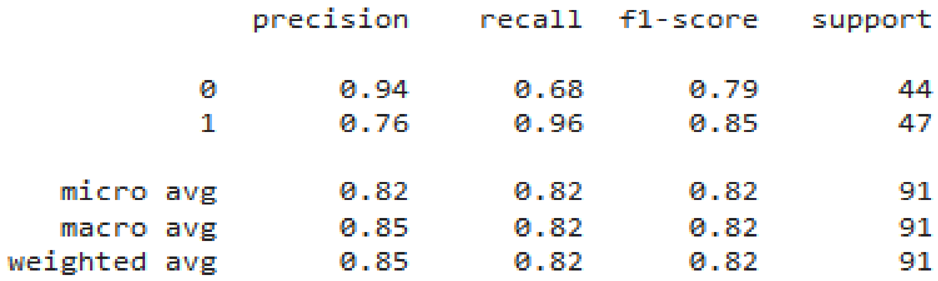
```
[[37  7]
 [ 2 45]]
```

http://doi.org/10.17993/3ctecno.2020.specialissue5.233-247

B. Decision Trees

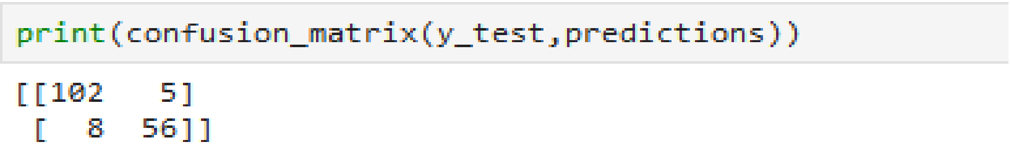
Applying the decision tree, we get the following results.

Table 2. Result of applying the Decision Trees algorithm.



```
             precision    recall  f1-score   support

           0       0.94      0.68      0.79        44
           1       0.76      0.96      0.85        47

   micro avg       0.82      0.82      0.82        91
   macro avg       0.85      0.82      0.82        91
weighted avg       0.85      0.82      0.82        91
```

Table 2 shows the average weight as 0.85, and confusion matrix as:



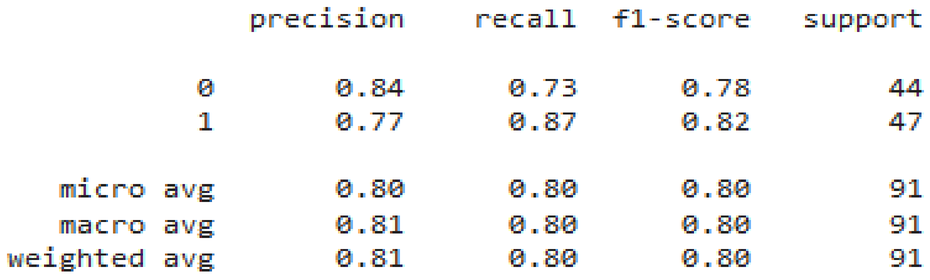
```
print(confusion_matrix(y_test,predictions))
```

```
[[102   5]
 [  8  56]]
```

C. Random Forest

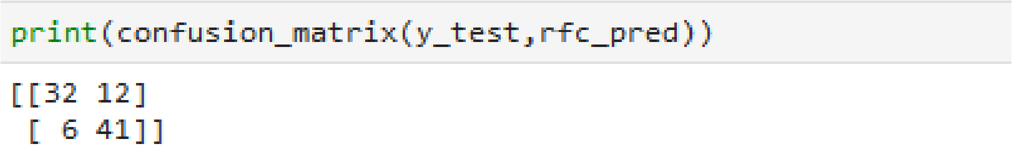
We evaluate the random forest model according to the data already preprocessed and trained with several estimates of 100.

Table 3. Result of applying the Random Forest algorithm.



```
             precision    recall  f1-score   support

           0       0.84      0.73      0.78        44
           1       0.77      0.87      0.82        47

   micro avg       0.80      0.80      0.80        91
   macro avg       0.81      0.80      0.80        91
weighted avg       0.81      0.80      0.80        91
```

It has an average weight of 0.81, and the confusion matrix as:



```
print(confusion_matrix(y_test,rfc_pred))
```

```
[[32 12]
 [ 6 41]]
```

http://doi.org/10.17993/3ctecno.2020.specialissue5.233-247

D. Neural Network

Training and test data are separated, to train the model using Keras dataset, then the model will be evaluated. Figures 7 and 8 show the models.

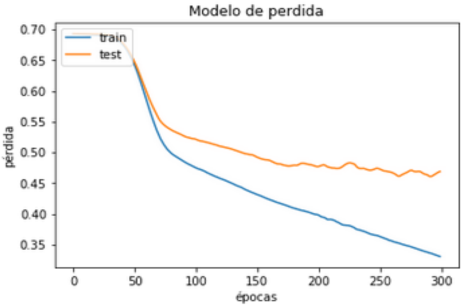


Figure 7. Loss model.

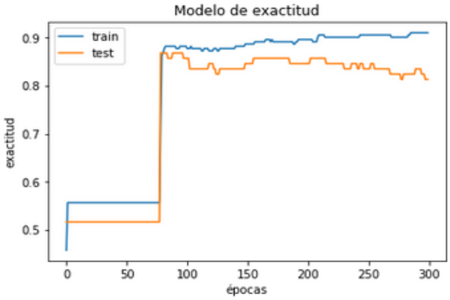
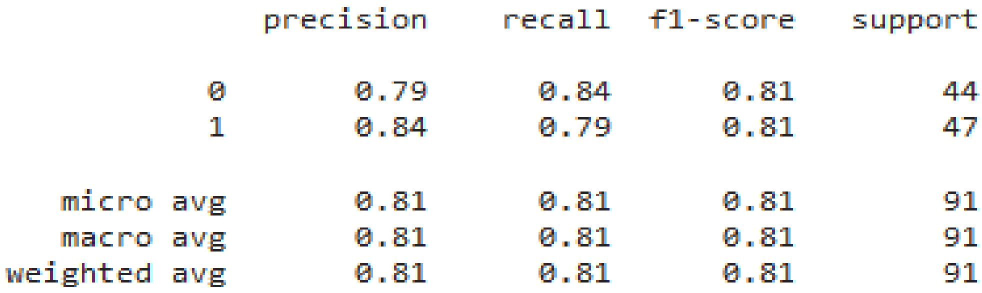


Figure 8. Accuracy model.

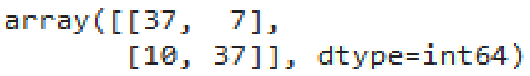
Table 4. Result of applying Neural Network algorithm.



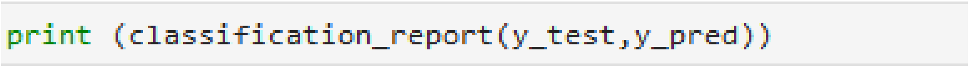
```
              precision    recall  f1-score   support

           0       0.79      0.84      0.81        44
           1       0.84      0.79      0.81        47

   micro avg       0.81      0.81      0.81        91
   macro avg       0.81      0.81      0.81        91
weighted avg       0.81      0.81      0.81        91
```

Table 4 shows the weight average accuracy obtained of 0.81.

The following confusion and information matrix are obtained:



```
array([[37,  7],
       [10, 37]], dtype=int64)
```

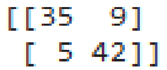


```
print (classification_report(y_test,y_pred))
```

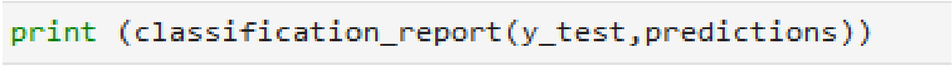
E. Support Vector Machines (SVM)

The model will be evaluated according to the preprocessed data, and the following is obtained, and the report classification and matrix are:

http://doi.org/10.17993/3ctecno.2020.specialissue5.233-247

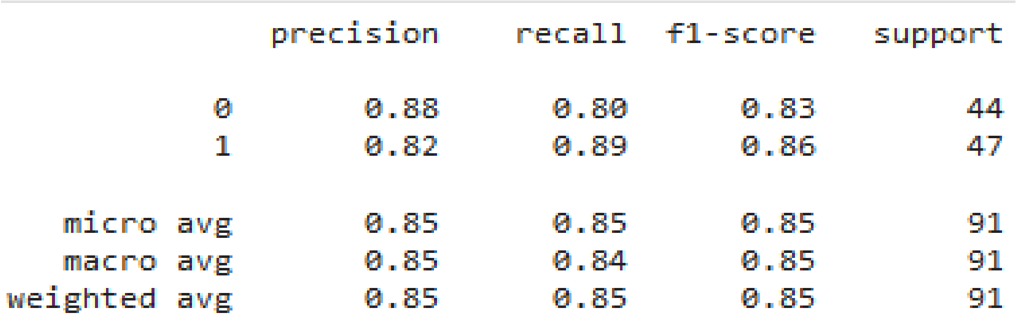


```
[[35  9]
 [ 5 42]]
```



```
print (classification_report(y_test,predictions))
```

Table 5. Result of applying Support Vector Machines algorithm.



```
              precision    recall  f1-score   support

           0       0.88      0.80      0.83        44
           1       0.82      0.89      0.86        47

   micro avg       0.85      0.85      0.85        91
   macro avg       0.85      0.84      0.85        91
weighted avg       0.85      0.85      0.85        91
```

Table 5 shows the weighted average accuracy of 0.85.

6. CONCLUSION

As was observed in the results, the model of k nearest neighbors has obtained better results in precision with an average accuracy of 0.91 for the heart disease dataset. For future work, other types of classification or segmentation can be applied to achieve a better prediction of the chosen dataset.

ACKNOWLEDGMENTS

This paper has been possible to carry out as research due to the need to obtain and generate knowledge from different professionals. The authors wish to thank our university mentors for their support and guidance.

REFERENCES

Agarwal, R., & Sagar, P. (2019). A Comparative Study of Supervised Machine Learning Algorithms for Fruit Prediction. *Journal of Web Development and Web Designing, 4*(1), 14-18. https://zenodo.org/record/2621205#.XoRZtYgzZPY

Bhavsar, H., & Ganatra, A. (2012). A Comparative Study of Training Algorithms for Supervised Machine Learning. *International Journal of Soft Computing and Engineering (IJSCE), 2*(4), 74-81. http://www.ijsce.org/wp-content/uploads/papers/v2i4/D0887072412.pdf

Bustamante, J. C., Rodríguez, C., & Esenarro, D. (2019). Real Time Facial Expression Recognition System Based on Deep Learning. International Journal of Recent Technology and Engineering (IJRTE), 8(2S11), 4047-4051. https://www.ijrte.org/wp-content/uploads/papers/v8i2S11/B15910982S1119.pdf

Cao, Y., Fang, X., Ottosson, J., Näslund, E., & Stenberg, E. (2019). A Comparative Study of Machine Learning Algorithms in Predicting Severe Complications after Bariatric Surgery. *Journal of Clinical Medicine, 8*(5), 668. https://doi.org/10.3390/jcm8050668

Manzoor, S. I., & Singla, J. (2019). A Comparative Analysis of Machine Learning Techniques for Spam Detection. *International Journal of Advanced Trends in Computer Science and Engineering, 8*(3), 810-814. http://www.warse.org/IJATCSE/static/pdf/file/ijatcse73832019.pdf

Mureșan, H., & Oltean, M. (2018). Fruit recognition from images using deep learning. *Acta Universitatis Sapientiae, Informatica, 10*(1), 26-42. https://www.researchgate.net/publication/321475443_Fruit_recognition_from_images_using_deep_learning

Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology (IJCTT), 48*(3), 128-138. https://doi.org/10.14445/22312803/IJCTT-V48P126

Ray, S. (2018). *A Comparative Analysis and Testing of Supervised Machine Learning Algorithms*. https://doi.org/10.13140/RG.2.2.16803.60967

Sathya, R., & Abraham, A. (2013). Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. *International Journal of Advanced Research in Artificial Intelligence (IJARAI), 2*(2). http://dx.doi.org/10.14569/IJARAI.2013.020206

Sethi, K., Gupta, A., Gupta, G., & Jaiswal, V. (2019). Comparative Analysis of Machine Learning Algorithms on Different Datasets. In *Circulation in Computer Science International Conference on Innovations in Computing (ICIC 2017), 87-91.* https://www.researchgate.net/publication/332223901_Comparative_Analysis_of_Machine_Learning_Algorithms_on_Different_Datasets

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research, 15*(1), 1929-1958. https://www.researchgate.net/publication/286794765_Dropout_A_Simple_Way_to_Prevent_Neural_Networks_from_Overfitting

http://doi.org/10.17993/3ctecno.2020.specialissue5.233-247