

# Explainable Machine Learning Prediction for Mortality of COVID-19 in the Colombian Population

**Gabriel Ricardo Vásquez Morales**

Ministry of Health and Social Protection

**Sergio Mauricio Martínez Monterrubio**

Universidad Complutense de Madrid

**Juan Antonio Recio García** (✉ [jareciog@ucm.es](mailto:jareciog@ucm.es))

Universidad Complutense de Madrid

**Pablo Moreno Ger**

Universidad Internacional de La Rioja

---

## Research Article

**Keywords:** COVID-19 pandemic, global public health problem, transparency, COVID-19 prognosis

**Posted Date:** April 30th, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-404848/v1>

**License:**  This work is licensed under a Creative Commons Attribution 4.0 International License. [Read Full License](#)

---

# Abstract

The COVID-19 pandemic, which began in late 2019, has become a global public health problem, resulting in large numbers of people infected and dead. One of the greatest challenges in dealing with the disease is to identify those people who are most at risk of becoming infected, seriously ill and dying from the virus, so that they can be isolated in a targeted manner and thus reduce mortality rates. This article proposes the use of machine learning, and specifically of neural networks and random forest to build two complementary models that identify the probability that a person has of dying because of COVID-19. The models are trained with the demographic information and medical history of two population groups: on the one hand, 43,000 people who died from COVID-19 in Colombia during 2020, and on the other hand, a random sample of 43,000 people who became ill with COVID-19 during the same period of time, but later recovered. After training the neural network classification model, evaluation metrics were applied that yielded an 88% accuracy value. However, transparency is a major requirement for the explicability of COVID-19 prognosis. Therefore, a complementary random forest model is trained that allows the identification of the most significant predictors of mortality by COVID-19.

## Introduction

Artificial intelligence has recently been used in medicine for the detection and treatment of diseases. One of the most successful fields of application of artificial intelligence in disease detection has been machine learning. Machine learning consists of the use of computer algorithms to train mathematical and statistical models from large volumes of data that can include diagnostic images, laboratory tests and medical records. Through this training, the models can identify patterns in the data, which can then be applied to analyze new data sets. For example, a machine learning model can be trained with a large set of patient cell images telling the algorithm which ones belong to cancer patients and which ones do not. After training the model, it is expected that it will be able to identify from a new image of the cells, whether the patient has cancer or not<sup>1</sup>. Some machine learning algorithms such as SVM, random forest (RF) and neural networks (NN)<sup>2–4</sup> have been successfully used in the diagnosis of diseases such as diabetes<sup>5,6</sup>, Alzheimer's<sup>7</sup>, heart disease<sup>8</sup>, cancer<sup>9,10</sup>, liver cirrhosis<sup>11</sup> and chronic kidney disease<sup>12–14</sup>, among others<sup>15–19</sup>.

The COVID 19 pandemic, which originated in late 2019 in China caused by a coronavirus similar to that which causes the common cold and severe acute respiratory syndrome (SARS), has become a major public health problem and has infected more than 82.7 million people worldwide, causing more than 1.8 million deaths by the end of 2020<sup>20</sup>. The general isolation measures taken by governments worldwide have not been sufficient to contain the advance of the disease and have had a negative impact on the local and global economy. Figure 1 graphically presents the growth in the number of daily cases of COVID-19 worldwide<sup>20</sup>. One of the great challenges in dealing with the disease is to identify the people who are most at risk of becoming infected, seriously ill and dying from the virus, in order to isolate them selectively and thus reduce mortality rates.

This article proposes the use of machine learning, and specifically of neural networks to build a model that identifies the probability that a person has of dying from COVID-19. The model has as an input data set the demographic information and the diagnosis history of diseases of the individuals, coded according to the International Classification of Diseases (ICD). In previous works<sup>21</sup> models of neuronal networks capable of predicting the risk of presenting pathologies such as chronic kidney disease has been presented, taking a similar set of data<sup>22–25</sup>. These models have managed to obtain values of accuracy greater than 95%. Other works related to the use of machine learning in the diagnosis of COVID-19 propose models for the detection of new cases<sup>26,27</sup>, the use of neural networks for case detection by chest X-rays<sup>28–30</sup>, the use of models to identify factors that influence patient mortality<sup>31,32</sup>, as well as to identify other environmental factors that may affect the spread of the epidemic<sup>33</sup>.

Despite the tremendous performance of neural networks, they work as black-box systems and their effectiveness is limited by their inability to explain their predictions to the experts. The problem of explainability in Artificial Intelligence is not new<sup>34</sup> but the rise of the machine learning as a very successful classification technique has created the necessity to understand how these systems make a prediction in order to increase user's reliability and trust<sup>35</sup>. Therefore, this paper proposes an alternative prediction model using random forests that enables the identification of the most significant predictors of mortality by COVID-19.

## Results

In this section the evaluation of the two previously trained machine learning models will be carried out. For this, a set of classifier evaluation metrics derived from the confusion matrix technique was applied.

### Neural network evaluation

The objective of metrics for the evaluation of classification algorithms is to identify the predictive ability of the model. To achieve this goal, the classes predicted by the algorithm for each example in the test set are compared with the actual value of the class and thus it is identified whether the model was able to correctly classify the data. A widely used technique is the confusion matrix, in which the examples classified correctly and incorrectly are counted, grouping them into true positives, true negatives, false positives and false negatives<sup>36</sup>.

Figure 2 graphically shows the confusion matrix for the final neural network model. The x-axis corresponds to the real values, and the y-axis to the values predicted by the model. The position (1,1) corresponds to the number of true positives, the position (0,0) to the true negatives, the position (0,1) to the false positives, and the position (1,0) to the false negatives. As can be seen in the graph, the network correctly classified 88% of the cases and failed in 12%. Of this percentage of errors, most correspond to false positives, that is, the model is predicting that the person is going to die when he or she recovers.

From the confusion matrix, the set of metrics presented in Table 1 can be obtained. These metrics confirm, on the one hand, the predictive ability of the classifier obtained through the accuracy metric, and on the other hand, a tendency of the model to predict more accurately the positive examples with respect to the negative ones. This can be observed by considering that the sensitivity value, which measures the proportion of correctly classified positive examples, is higher than the specificity value, which measures the proportion of correctly classified negative examples.

For binary classification algorithms, a metric known as area under the curve (AUC) can be used<sup>37</sup>. This metric is used to determine the balance between detecting true positives and avoiding false positives. To do this, it shows the detection ratio of true positives on the y-axis, and the ratio of false positives on the x-axis. Figure 3 shows the ROC curve obtained along with its AUC value.

### Comparison with random forest model

Table 2 presents a comparison of the main metrics obtained after applying the neural network and random forest models. The table shows that the best classifier is the neural network, considering an accuracy value of 88%.

The sensitivity metric, which measures the proportion of positive examples correctly classified, shows a better performance of the neural network against random forest. The same situation occurs for specificity, where the neural network gets a higher value than the random forest model. The precision metric indicates the proportion of examples that are truly positive, and the recall metric measures how complete the results are and is like the sensitivity of the model. The value of F-measure corresponds to a balance between precision and recall and simplifies the performance

of a classification algorithm in a single metric. Finally, the area under the curve (AUC) of the neural network model again exceeds random forest because of its tendency to correctly identify true positives and avoid false positives.

### Significant predictors

A very important feature of Random Forest is that, although it is a black box model, it is possible to know the importance that the algorithm gives to each input variable. Importance measures the impact that each variable has on the final prediction of the model. In the case of the model trained with the demographic and health care data, the distribution of values indicated in Table 3 was obtained. In this table are the 10 variables with the greatest importance for the model. In first place is the variable Age, followed by a set of diagnoses among which are hypertension (I10), diabetes mellitus (E10.9, E11.9 and E10.8), obesity (E66.9 and E66.0), chronic obstructive pulmonary disease (J44.9) and chronic renal disease (N18.9). These diagnoses coincide with the medical theory that points to these variables as the main risk factors for death from COVID-19<sup>38</sup>. Other diagnoses such as prostate hyperplasia (N40) may be associated with aging, considering that the main risk factor is the age of the patients.

## Discussion

The neural network and the model trained with random forest were able to identify patients at risk of dying from COVID-19, with values of accuracy of 88% and 87% respectively. This demonstrates the predictive capacity of both models and their effectiveness in identifying patterns in large data sets, as well as their application in the early detection of diseases and other health risk conditions. Future work includes the implementation of the model for predicting the total population of Colombia to identify the people with the highest level of risk within the health system and to take the necessary measures for their protection. It is also planned to create a web application that, applying the trained model, allows people to consult their individual risk level.

## Methods

This section describes the techniques used following the CRISP-DM methodology in order to perform the data capture and treatment, training and optimization of the machine learning models built with the aim of predicting the risk of death by COVID-19. The CRISP-DM methodology is considered a standard for the life cycle of data analysis projects and includes descriptions of the basic phases of a project, the tasks required in each phase and the relationships between the tasks. Among its advantages over other methodologies is its flexibility, since it can be easily adapted to any data exploitation project, such as automatic learning<sup>39</sup>. A general overview of this methodology is presented in Fig. 4.

## Data collection

The data set required for the creation of the machine learning models was obtained from the SEGCOVID and RIPS databases of the Colombian Ministry of Health and Social Protection. SEGCOVID is a web application that records the follow-up of suspected and confirmed cases of COVID-19. The RIPS database (*Registro Individual de Prestación de Servicios de Salud*) contains information on medical care provided to all members of the health system in Colombia since 2009<sup>41</sup>. Two samples were taken from the population: one corresponding to 43,000 people who died in Colombia from COVID-19 during 2020, and the other to a group of 43,000 people who fell ill from COVID-19 during the same period of time but subsequently recovered. The next step was to integrate the two groups of people in the same table called "Patients", adding for each record the fields: ID or unique anonymized identifier of the person, sex, age, ethnicity, place of residence, and the label "Dead" which indicates with a "1" if the person belongs to the group of deceased or with a "0" if he or she belongs to the group of recovered patients. Figure 5 shows some example data taken from the "Patients" table. The total number of records in this table is 86,000.

Finally, the "Diagnoses" table was created, in which all the diagnoses identified in the RIPS database were added for each person in the "Patients" table, as well as the number of medical cares provided for this diagnosis. This set of diagnoses included the diseases detected in these individuals up to December 31, 2020. This table has 1,076,718 records, corresponding to 86,000 persons and 8,111 diagnoses of diseases. Figure 6 shows some example data taken from the "Diagnoses" table.

## Data cleaning and preparation

This process transforms the data set obtained from the database and generates a new model with which to train and test the neural network. This transformation includes feature selection operations, table joining, row to column transformation, categorical variable handling, null value treatment and other cleaning operations required to obtain the final dataset.

From the initial exploration of the data, several variables were considered to make up the data set; however, they were not included due to the lack of data, as in the case of the results of laboratory tests and family history, or due to data quality problems, as in the case of variables related to people's weight and height. Finally, the following variables were identified with which the model was trained:

- **Sex:** categorical variable with two domain values: "F" (Female) and "M" (Male).
- **Age:** numerical variable with integer values between 0 and 130, which stores the age of the patient.
- **Ethnicity:** categorical variable with 4 domain values: "INDIGENOUS", "AFROCOLOMBIAN", "OTHER ETHNICS" and "NONE".
- **Department:** categorical variable with 32 domain values corresponding to the codes of the departments of Colombia. It represents the geographical location of the patient.
- **Diagnoses:** categorical variable with the ICD codes of the 8,111 diseases identified in the "Diagnoses" table. For each diagnosis, the number of treatments reported in the RIPS by each person is stored. If the person does not have the diagnosis, the number of cares provided will be equal to 0.
- **Deceased:** numerical variable that contains two values: 1 if the person died and 0 otherwise. This variable was used as a label for each element of the data set.

To create the data model required for the training of the neural network, it is necessary to have all the patient's information in the same record. For this reason, it was necessary to perform a transformation from rows to columns so that all the pathologies that a patient has are in the same row. Once the function is executed, a single dataframe is generated containing a record for each of the 86,000 persons in the sample, with 8,112 columns corresponding to each possible disease within the database, in addition to the person's unique identifier. The final step is to fill in the blank values with zeros. This occurs in cases where the person does not have any of the diseases.

Categorical variables contain descriptions within a set of finite elements that cannot be converted into numerical values. This is the case with the variables sex, ethnicity and department in the table "Patients". For these cases it is necessary to create as many columns as possible domain values contained in each variable and fill in with values "1" or "0" depending on the option that applies to each person.

A final step required for the creation of the final data set consists of joining the two tables worked on so far: the "Patients" and the "Diagnoses" tables. Each table contains only one record for each person, so the process consists in joining the row of the "Patients" table with the row of the "Diagnoses" table for the same person. Figure 7 shows graphically the creation of the final dataset.

# Training and test datasets

Once the data was cleaned and prepared, it was used to train the neural network. At this stage of the process it is necessary to separate the data set into three subsets: training, validation, and testing. The training data set is composed of those examples that will allow the model to learn from the characteristics and identify the patterns hidden in the data. The validation set allows you to identify if the model is learning correctly as it is being trained. This is important considering that machine learning models can be overfitted. Before proceeding with the creation of the datasets, three additional operations are required. The first consists of a review of the variables used for training the model. The set of variables up to this moment is composed of 8,152 variables obtained after applying data cleaning and preparation processes. As a result of this review it was identified that the ID variable is not relevant for the training of the model. Once the column is removed, the resulting data set is composed of 86,000 records and 8,151 variables. The following operation consists of separating the data into two different structures that within the Python language are known as dataframes. On the one hand, we have the X dataframe, which contains the characteristics or input variables of the model, and on the other hand we have the Y dataframe, which contains the class or output variable of the model. The third operation is to normalize the data. Each variable of the X dataframe contains a range of values different from the other variables or characteristics. This can be a problem for training the model because variables with higher values may have more weight than the others. To solve this situation, a process of data normalization is performed, so that all variables are in a range of values between 0 and 1. Finally, the separation of the training and test data sets is performed using the `train_test_split` function, belonging to the scikit-learn library. In this case 30% of the data was used for testing and the remaining 70% of the examples were used to train the model. The validation data set will be obtained from the training set during the construction of the neural network.

## Definition of the neural network topology

The topology or architecture of the network defines the number of layers, as well as the number of neurons per layer and how they are connected. The topology of a network is directly related to the complexity of the tasks that can be learned by it<sup>42</sup>. Generally, networks with a greater number of layers and neurons can identify more complex patterns, although they consume a greater amount of computational resources, especially processing capacity and memory space. The proposed neural network is composed of 5 layers according to the diagram in Fig. 8. The input layer corresponds to the characteristics or input variables of the network, composed of 8.150 nodes or neurons. The following 3 layers correspond to the hidden layers of the model and contain 500, 100 and 50 neurons, respectively. The last layer corresponds to the neuron that represents the only class of the binary classification problem. The objective of the training is to obtain the values corresponding to the optimal weights ( $W$ ) for each layer of the network, as well as the bias values<sup>43</sup>.

## Model training

The initial model of the neural network was implemented using the Keras library<sup>44</sup> and the TensorFlow framework<sup>45</sup> in Python. This library facilitates the creation and evaluation of classifiers with neural networks. The `Sequential` class allows to add new layers to the network, indicating for each one the number of neurons, as well as the activation function to be implemented. Once the model was defined, it was compiled using an optimization algorithm. The use of the gradient descent algorithm is common, although in practice a variation of this algorithm known as stochastic gradient descent (SGD) is used which is less computationally costly<sup>42</sup>. Although stochastic gradient descent is an algorithm that works well in general, it has several problems that sometimes make it difficult to train neural networks. One of the algorithms that has been created to solve these problems is Adam<sup>46</sup>. This algorithm combines techniques taken from other methods such as RMSProp<sup>47</sup> and SGD with momentum to improve the speed with which it converges in the search for an optimal solution. Adam also reduces the probability that the algorithm stops in an intermediate

position and is not able to advance in the search for a local or global minimum. For these reasons Adam was used for the training of the neural network proposed in this work.

One of the most critical elements for neural network training is the activation function. The activation function is the mechanism by which artificial neurons process information and this information is propagated through the network. The activation function chosen for the neural network training was the RELU function (rectified linear unit). This is the most widely used activation unit in practice today<sup>48</sup>. Neural networks are models with great power of representation. The large number of parameters and layers added to a neural network makes it easy for the network to learn too well the data it is training with. Sometimes, the network may be able to perfectly memorize a set of training data, even classifying them perfectly. However, when this is done, the model loses its ability to generalize and, when evaluated on a set of data not seen during the training or test set, a low capacity for prediction can be observed<sup>42</sup>. This phenomenon is known as overfitting and is a common phenomenon in machine learning: the algorithm is modeling the training data too well while losing ability to generalize on unseen data. The regularization techniques try to solve this problem. Dropout is a quite modern technique that has found a great reception<sup>49</sup>. Dropout prevents the memorization of variables. In this model, the dropout technique was applied to the output of each layer of the network, with a hyper parameter value equal to 0.5. After training the neural network with 10 iterations or epochs it was found that the model converged in the third iteration and obtained an accuracy value of 88% in both the training and the assessment datasets. Figure 9 shows the point of convergence of the model, as well as a possible overfitting that occurs if the model is continued to be trained beyond the third iteration.

## Declarations

## Acknowledgements

This work was supported by the Spanish Committee on Economy and Competitiveness (TIN2017-87330-R) and SECTEI (Subsecretaría de Ciencia, Tecnología e Innovación de la Ciudad de México).

## References

1. Alarabeyyat, A. & Alhanahnah, M. Breast cancer detection using k-nearest neighbor machine learning algorithm 2016 *9th International Conference on Developments in eSystems Engineering (DeSE)*. 35–39 (2016).
2. McCulloch, W. S. & Pitts, W. A logical calculus of ideas immanent in nervous. *Bulletin of Mathematical Biophysics*. **5**, 115–133 (1943).
3. Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*. **2**, 359–366 (1989).
4. Hinton, G. E., Osindero, S. & Teh, Y. A fast learning algorithm for deep belief nets. *Neural Computation*. **18**, 1527–1554 (2006).
5. Yu, W., Liu, T., Valdez, R., Gwinn, M. & Khoury, M. J. Application of support vector Machine modeling for prediction of common diseases: the case of diabetes and pre-diabetes. *BMC medical informatics and decision making*. **10** (1), 16 (2010).
6. Zou, Q. *et al.* Predicting diabetes mellitus with machine learning techniques. *Frontiers in genetics*. **9**, 515 (2018).
7. Magnin, B. *et al.* Support vector machine-based classification of Alzheimer's disease from whole-brain anatomical MRI. *Neuroradiology*. **51** (2), 73–83 (2009).
8. Dessai, I. S. Intelligent heart disease prediction system using probabilistic neural network. *International Journal on Advanced Computer Theory and Engineering (IJACTE)*. **2** (3), 2319–2526 (2013).

9. Zhou, Z. H., Jiang, Y., Yang, Y. B. & Chen, S. F. Lung cancer cell identification based on artificial neural network ensembles. *Artificial Intelligence in Medicine*. **24** (1), 25–36 (2002).
10. Bejnordi, B. E. *et al.* Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*. **318**, 2199–2210 (2017).
11. Cao, Y., Hu, Z. D., Liu, X. F., Deng, A. M. & Hu, C. J. An MLP classifier for prediction of HBV-induced liver cirrhosis using routinely available clinical parameters. *Disease markers*. **35** (6), 653–660 (2013).
12. Rady, E. H. & Anwar, A. S. Prediction of kidney disease stages using data mining algorithms. *Informatics in Medicine Unlocked*. **15**, 100178 (2019).
13. Xiao, J. *et al.* Comparison and development of machine learning tools in the prediction of chronic kidney disease progression. *Journal of translational medicine*. **17** (1), 119 (2019).
14. Soldevila, A. *Análisis de la progresión de la enfermedad renal crónica avanzada mediante técnicas de aprendizaje máquina*. <https://roderic.uv.es/handle/10550/61346> (2017).
15. West, D. & West, V. Improving diagnostic accuracy using a hierarchical neural network to model decision subtasks. *International journal of medical informatics*. **57** (1), 41–55 (2000).
16. Lakhani, P. & Sundaram, B. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology*. **284**, 574–582 (2017).
17. Oh, S. L. *et al.* A deep learning approach for Parkinson's disease diagnosis from EEG signals. *Neural Computing and Applications* 1–7(2018).
18. Kermany, D. S. *et al.* Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*. **172**, 1122–1131 (2018).
19. Poostchi, M., Silamut, K., Maude, R. J., Jaeger, S. & Thoma, G. Image analysis and machine learning for detecting malaria. *Translational Research*. **194**, 36–55 (2018).
20. Johns Hopkins University. *Coronavirus Resource Center*. <https://coronavirus.jhu.edu/map.html> (2021).
21. Vásquez-Morales, G. R., Martínez-Monterrubio, S. M. & Moreno-Ger, P. & Recio-García, J. A. Explainable prediction of chronic renal disease in the colombian population using neural networks and case-based reasoning. *IEEE Access*. **7**, 152900–152910 (2019).
22. Hore, S., Chatterjee, S., Shaw, R. K., Dey, N. & Virmani, J. Detection of chronic kidney disease: A NN-GA-based approach. *Nature Inspired Computing*. 109–115(2018).
23. Al Imran, A., Amin, M. N. & Johora, F. T. Classification of Chronic Kidney Disease using Logistic Regression, Feedforward Neural Network and Wide & Deep Learning 2018 *International Conference on Innovation in Engineering and Technology*. 1–6 (2018).
24. Chatterjee, S., Banerjee, S., Basu, P., Debnath, M. & Sen, S. Cuckoo search coupled artificial neural network in detection of chronic kidney disease 2017 *1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech)*. 1–4 (2017).
25. Ren, Y., Fei, H., Liang, X., Ji, D. & Cheng, M. A hybrid neural network model for predicting kidney disease in hypertension patients based on electronic health records. *BMC Medical Informatics and Decision Making*. **19** (2), 51 (2019).
26. Distante, C., Pereira, I. G., Goncalves, L. M., Piscitelli, P. & Miani, A. Forecasting Covid-19 Outbreak Progression in Italian Regions: A model based on neural network training from Chinese data. medRxiv(2020).
27. Rustam, F. *et al.* COVID-19 Future Forecasting Using Supervised Machine Learning Models. *IEEE Access*. **8**, 101489–101499 (2020).

28. Narin, A., Kaya, C. & Pamuk, Z. Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *arXiv preprint arXiv: 2003.10849* (2020).
29. Apostolopoulos, I. D. & Mpesiana, T. A. Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*. **43**, 635–640 (2020).
30. Khan, A. I., Shah, J. L., Bhat, M. M. & Coronet A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. *Computer Methods and Programs in Biomedicine*. **196**, 105581 (2020).
31. Yan, L. *et al.* An interpretable mortality prediction model for COVID-19 patients. *Nature Machine Intelligence*. **2** (5), 283–288 (2020).
32. Zhu, J. S. *et al.* Deep-learning artificial intelligence analysis of clinical variables predicts mortality in COVID-19 patients. *Journal of the American College of Emergency Physicians Open*. **1** (6), 1364–1373 (2020).
33. Magazzino, C., Mele, M. & Schneider, N. The relationship between air pollution and COVID-19-related deaths: an application to three French cities. *Applied Energy*. **279**, 115835 (2020).
34. Arrieta, A. B. *et al.* Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*. **58**, 82–115 (2020).
35. Miller, T. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*. **267**, 1–38 (2019).
36. Fawcett, T. An introduction to ROC analysis. *Pattern Recognition Letters*. **27** (8), 861–874 (2006).
37. Bradley, A. P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*. **30**, 1145–1159 (1997).
38. Centers for Disease Control and Prevention. *People with Certain Medical Conditions*. <https://www.cdc.gov/coronavirus/2019-ncov/need-extra-precautions/people-with-medical-conditions.html> (2020)
39. Chapman, P. *et al.* The CRISP-DM user guide. In 4th CRISP-DM SIG Workshop in Brussels in March(1999).
40. Ministerio de Salud y Protección Social. *Resolución 676 de 2020*. <https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/DIJ/resolucion-676-de-2020.pdf> (2020).
41. Ministerio de Salud y Protección Social. *Resolución 3374 de 2000*. [https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/DIJ/Resoluci%C3%B3n\\_3374\\_de\\_2000.pdf](https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/DIJ/Resoluci%C3%B3n_3374_de_2000.pdf) (2000).
42. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016).
43. Nielsen, M. *Neural Networks and Deep Learning* (Determination press, 2015).
44. Gulli, A. & Pal, S. *Deep Learning with Keras* (Packt Publishing Ltd, 2017).
45. Abrahams, S., Hafner, D., Erwit, E. & Scarpinelli, A. *TensorFlow for Machine Intelligence: A Hands-on Introduction to Learning Algorithms* (Bleeding Edge Press, 2016).
46. Kingma, D. P., Ba, J. & Adam A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*(2014).
47. Hinton, G. Neural networks for machine learning. Coursera, video lectures. 264(1)(2012).
48. Jarrett, K., Kavukcuoglu, K. & LeCun, Y. What is the best multi-stage architecture for object recognition? 2009 *IEEE 12th international conference on computer vision*. 2146–2153 (2009).
49. Srivastava, N. Improving neural networks with dropout. University of Toronto. 182(566)(2013).

## Tables

**Table 1:** Neural network evaluation metrics.

Metric	Value
Sensibility	89%
Specificity	87%
Precision	87%
Recall	89%
F-Measure	88%
<b>Accuracy</b>	<b>88%</b>
AUC	95%

**Table 2:** Model comparison metrics. Source: own.

Metric	Neural Network	Random Forest
Sensibility	89%	87%
Specificity	87%	86%
Precision	87%	86%
Recall	89%	87%
F-Measure	88%	86%
<b>Accuracy</b>	<b>88%</b>	<b>87%</b>
AUC	95%	87%

**Table 3:** Importance of variables.

Variable	Description	Relevance
Age	Patient age in years	0.273151
I10	Essential (primary) hypertension	0.094111
E10.9	Type 1 diabetes mellitus without complications	0.027660
E66.9	Obesity, unspecified	0.022811
J44.9	Chronic obstructive pulmonary disease, unspecified	0.022095
N40	Benign prostatic hyperplasia	0.010785
E11.9	Type 2 diabetes mellitus without complications	0.010263
E66.0	Obesity due to excess calories	0.007944
N18.9	Chronic kidney disease, unspecified	0.007802
E10.8	Type 1 diabetes mellitus with unspecified complications	0.007176

# Figures

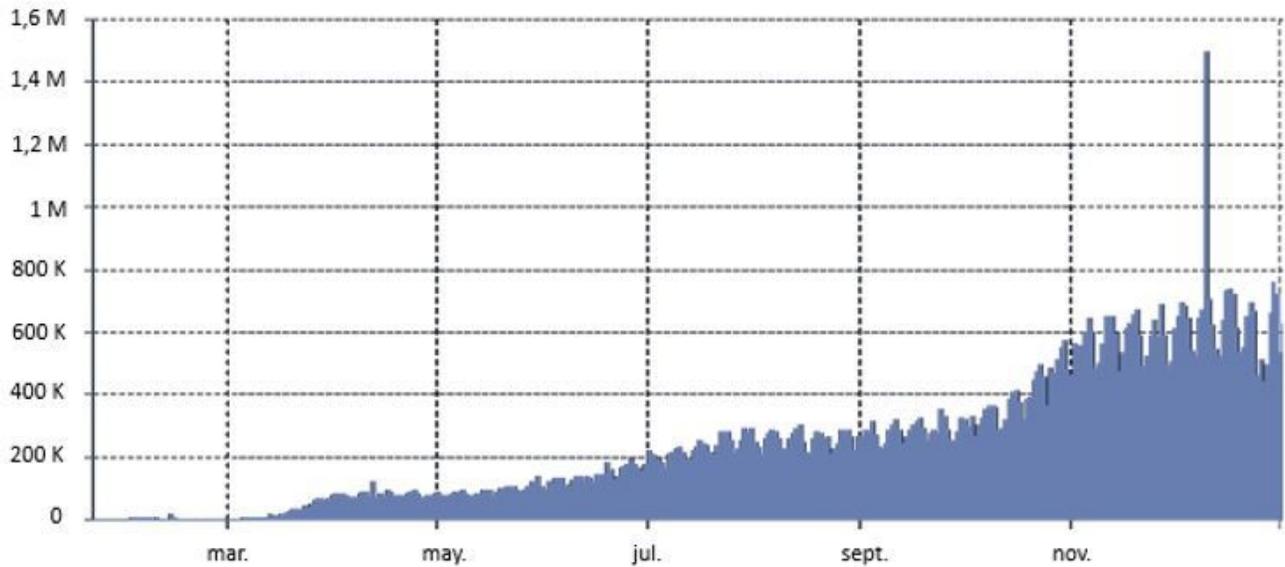


Figure 1

Daily growth in the number of COVID-19 cases worldwide20.

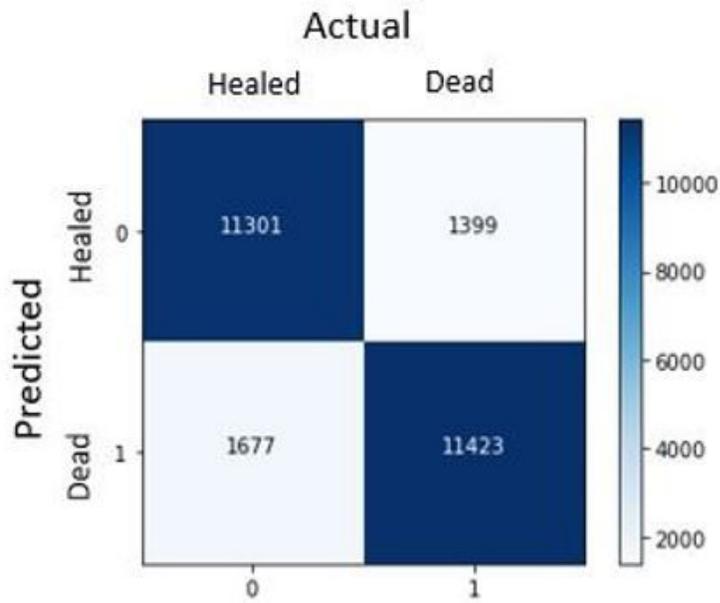
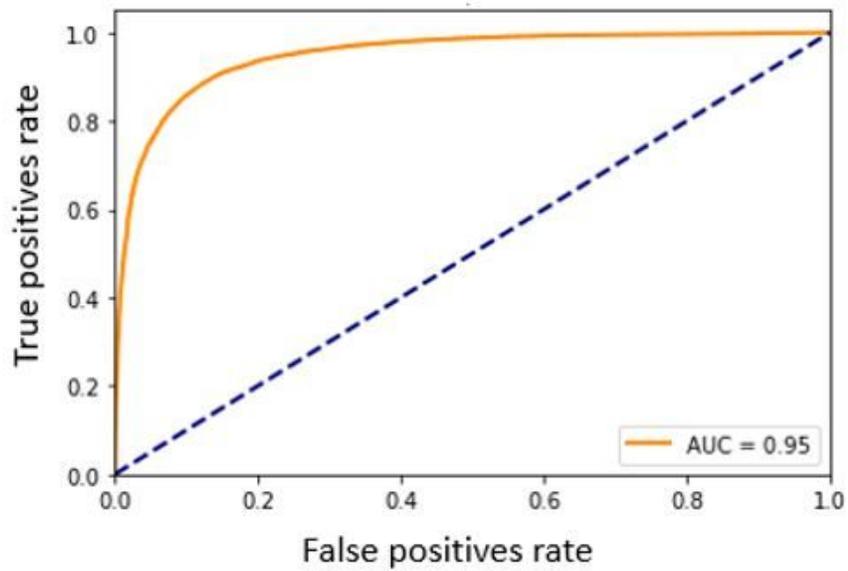


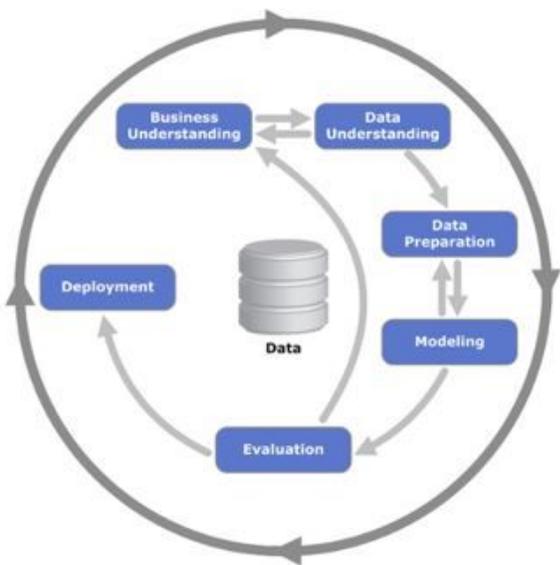
Figure 2

Confusion matrix of the neural network model.



**Figure 3**

ROC/AUC curve.



**Figure 4**

Phase diagram of the CRISP-DM methodology

ID	Sex	Age	Ethnicity	State	Dead
9935729	M	59	None	08	1
4983364	M	56	Indigenous	11	1
125555375	F	79	None	11	1
26720085	M	42	None	08	0
110262900	F	24	Indigenous	17	0
50977015	M	47	None	70	0

**Figure 5**

Patients table.

ID	DiagnosisCode	ServicesNumber
27894106	H259	1
42146268	K044	4
30355352	E785	1
30850286	R42X	1
27573225	M419	2
50512705	R42X	1

**Figure 6**

Diagnoses table.

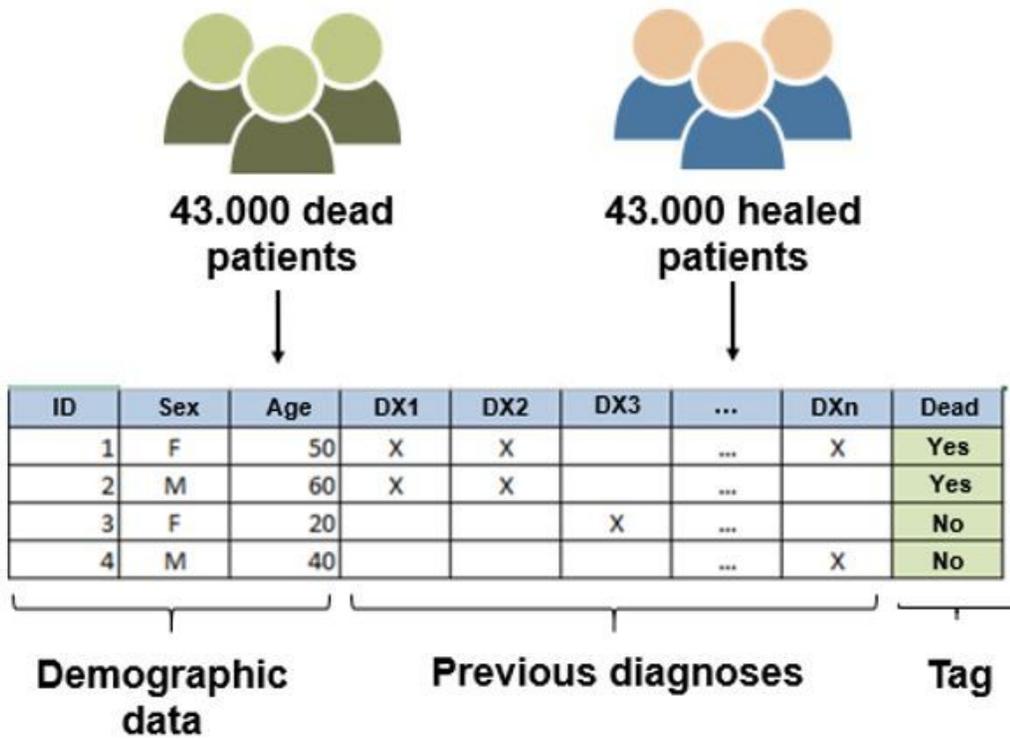


Figure 7

Covid-19 Data Set.

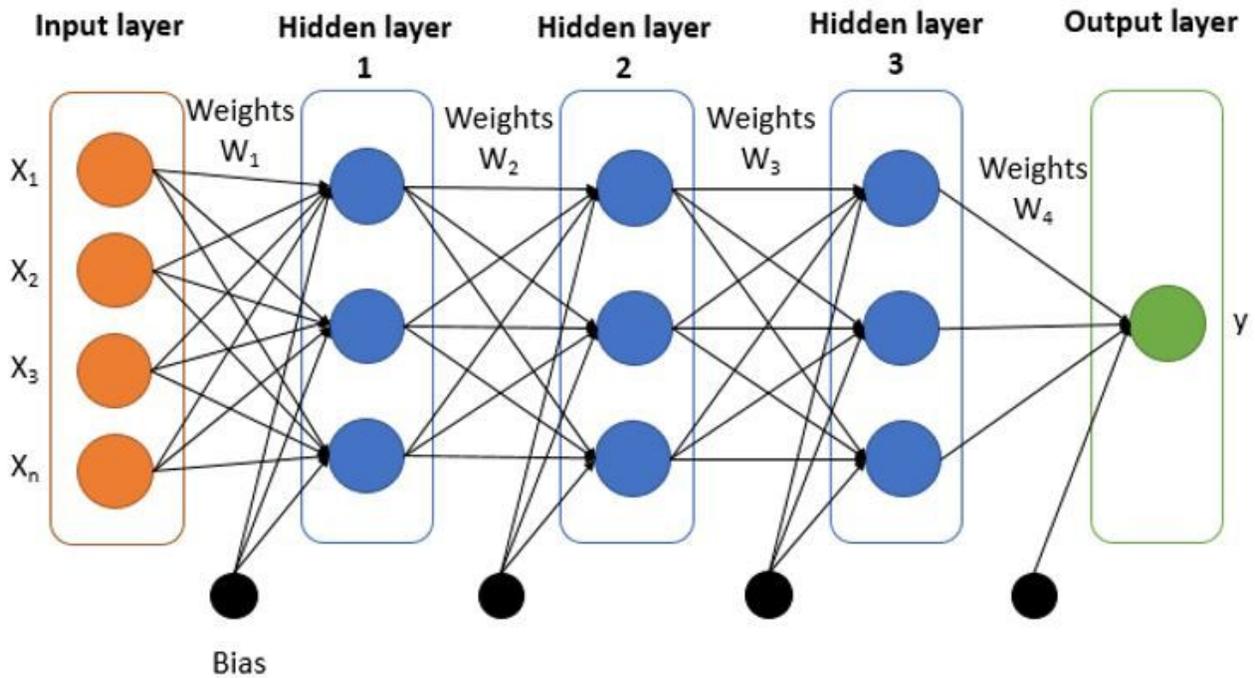
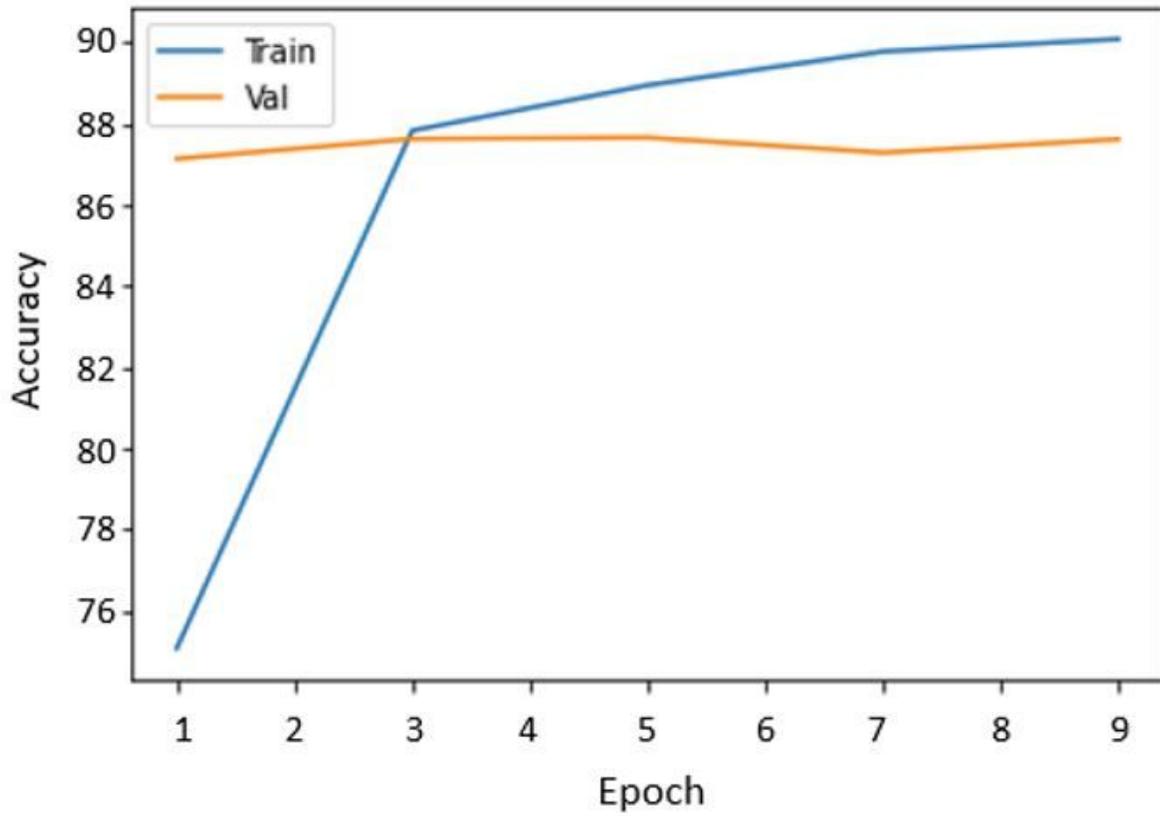


Figure 8

Topology of the neural network with 5 layers.



**Figure 9**

Accuracy in the final neural network model.