https://doi.org/10.46335/IJIES.2022.7.9.24 e-ISSN: 2456-3463
Vol. 7, No. 9, 2022, PP. 119-126

International Journal of Innovations in Engineering and Science, www.ijies.net

Severity Prediction of Covid-19 Patient Using Machine Learning Techniques

Dr. Pushpa Ravikumar[1], Arun Kumar N R[2], Prajwal Moudgalya[3], Shriman A S[4], Srihari N U[5]

[1] Prof. & Head of Department, Department of CS&E Adichunchanagiri Institute of Technology. Chikkamagaluru.
[2345] Department of CS&E, Adichunchanagiri Institute of Technology, Chikkamagaluru.

prajwalmoudgalya@gmail.com

***Received on**: 11 June ,2022* ***Revised on**: 08 August ,2022,* ***Published on**: 10 August,2022*

***Abstract-** A COVID-19 (coronavirus disease 2019) was detected and has spread rapidly across almost all countries throughout the world science end of the year 2019.It is an unusual viral pneumonia in patients. It was declared a pandemic by WHO because of its deadly effect on public health. Due to increase in the COVID-19 cases in present, we are here to predict the severity of the COVID-19 patients. We are performing this analysis based on the cases occurring in different states of India in present dates. Our dataset contains multiple classes so we are performing multi-class classification. This analysis contains four classifier Decision -tree, Random-Forest, Gradient boosted trees and Artificial Neural Network. This analysis will help the patients to know the amount of severity of COVID-19 which may reduce the crowd in Hospital, so that more severe patient can get the treatment as soon as possible.*

***Keywords:** Viral pneumonia, multi-*class *classification.*

I-INTRODUCTION

The virus of coronaviruses is a special kind of virus that itself is a disease and it enhances the existing disease in humans body which makes it a very dangerous virus. This virus results in wheezing, hard to breathe, bad digestive system, and liverwort, effects badly human nervous system, and also harms animals like cows, horses, and pigs that are kept, raised, and used by people and different wild animals. Coronavirus disease 2019 (COVID-19) is a highly contagious viral illness caused by severe acute respiratory syndrome SARS-CoV-2. It has had a devastating effect on the world's demographics resulting in more than 5.3 million deaths worldwide. After the first cases of this predominantly respiratory viral illness were first reported in Wuhan, Hubei Province, China, in late December 2019, SARS-CoV-2 rapidly disseminated across the world in a short span of time, compelling the World Health Organization (WHO) to declare it as a global pandemic on March 11, 2020. A COVID-19 diagnostic testing kit has been developed and is available in clinical testing labs. The gold standard for testing for COVID-19 is Reverse Transcription Polymerase Chain Reaction (RT-PCR). However, current data suggest that RT-PCR is only 30-70% effective for acute infection, this may be due to incorrect use of lab kits or not enough virus in the blood at the early stages of testing. Plus, the availability of testing will vary from country to country. In the proposed system which is able to predict the state of COVID-19 patient. For prediction Machine Learning algorithms Decision tree, Random Forest, Gradient Boosted Tree and Artificial Neural Network is used. Therefore, more accuracy is achieved. Fig. 1 Structure of coronavirus, showing the structure of COVID-19, this structure looks like a crown. The different parts of this virus are also introduced in this diagram .

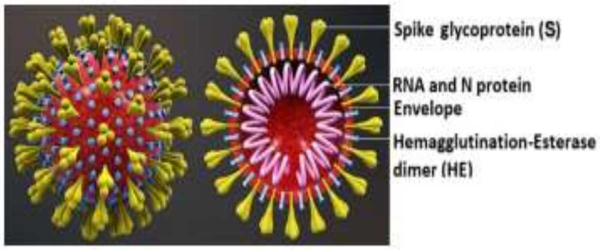


Fig. 1 -Structure of coronavirus.

The objectives of the Severity Prediction of Covid-19 Patient are:

- To predict severity of covid-19 patient using 4 different Machine Learning models such as Decision Tree, Random Forest, Gradient Boosted Trees and Neural Networks.
- To predict high-accuracy by interpreting Decision Tree, Random Forest, Gradient Boosted Trees and Neural Networks.
- To reveal the most significant indicators in early diagnosis of patient.
- To reduce the death rate in COVID Pandemic.

II- METHODOLOGIES

2.1 Data Collection

Machine learning needs two things to work, data (lots of it) and models. When acquiring the data, be sure to have enough features (aspect of data that can help for a prediction, like the surface of the house to predict its price) populated to train correctly your learning model. In general, the more data you have the better so make to come with enough rows.

Theprimarydatacollectedfromtheonlinesourcesremainsintherawformofstatements, digits and qualitative terms. The raw data contains error, omissions and inconsistencies. It requires corrections after careful scrutinizing the completed questionnaires. The following steps are involved in the processing of primary data. A huge volume of raw data collected through field survey needs to be grouped for similar details of individual responses.

2.2 Data pre-processing

Machine learning needs two things to work, data (lots of it) and models. When acquiring the data, be sure to have enough features (aspect of data that can help for a prediction, like the surface of the house to predict its price) populated to train correctly your learning model. In general, the more data you have the better so make to come with enough rows.

Data Pre-processing is a technique that is used to convert the raw data into a clean data set.In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. Therefore, certain steps are executed to convert the data into a small clean data set. This technique is performed before the execution of Iterative Analysis. The set of steps is known as Data Pre-processing.

It includes –

- Data Cleaning (K-Nearest Neighbor)
- Data Transformation
- Data Reduction

2.3 Data visualization

The process of organizing data into groups and classes on the basis of certain characteristics is known as the classification of data. Classification helps in making comparisons among the categories of observations. It can be either according to numerical characteristics or according to attributes. So here we need to visualize the prepared data to find whether the training data contains the correct label, which is known as a target or target attribute.

Next, we will slice a single data set into a training set and test set.

Training set—a subset to train a model.

Test set—a subset to test the trained model.

2.4Model building and training

The process of training an ML model involves providing an ML algorithm (that is, the learning algorithm) with training data to learn from. The term ML model refers to the model artefact that is created by the training process. The training data must contain the correct answer, which is known as a target or target attribute. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict), and it outputs an ML model that captures this pattern.

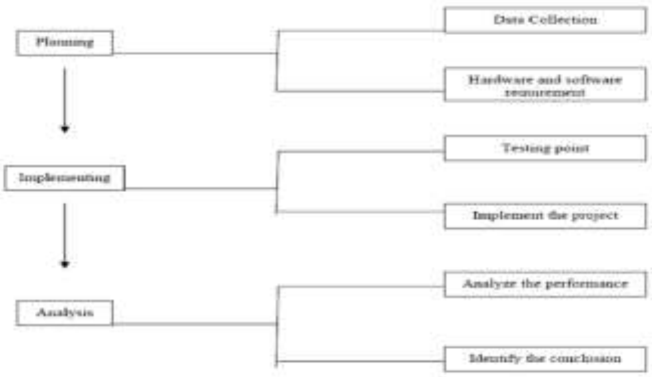


Fig. 2-Methodology of work.

2.5Model classification and Result Analysis

In testing phase the model is applied to new set of data. The training and test data are two different datasets. The goal in building a predictive model is to have the model perform well. On the training set, as well as generalize well on new data in the test set. Once the build model, classification is done using Decision tree, Random Forest, Gradient Boosted tree and Artificial Neural Network algorithms in which and classified into levels of severity in which we can classify whether the given patient's severity.

I. MACHINE LEARNING MODELS USED IN THIS STUDY

These are the models used for the Severity Prediction of Covid-19 Patient Using Machine Learning classification.

3.1 K-Nearest Neighbor

Missing values must be marked with NaN (not a number) values and can be replaced with
nearest neighbor estimated values.

Distance calculation in the presence of missing values

In the presence of missing coordinates, the Euclidean distance is calculated by ignoring the missing values and scaling up the weight of the non-missing coordinates

$$d_{xy} = \sqrt{weight * squared\ distance\ from\ present\ coordinates}$$

Equation 1 is the formula to calculate the Euclidian distance.

$$weight = \frac{Total\ number\ of\ coordinates}{Number\ of\ present\ coordinates}$$

Equation 2 is used to find the weight

For example, the Euclidean distances between two points (3, **NA**, 5) and (1, 0, 0) is:

$$\sqrt{\frac{3}{2}[(3-1)^2+(5-0)^2]} = 6.595453 \text{ ——— } (3)$$

Table 4.1: Sample dataset containing missing values

Using equation 1, the Euclidean distances between two points (5, **NA**, 1.6,0.2) and (5.2, 3.5, 1.5,0.2) is:

= 4/3 * {(5-5.2)^2 +(1.6-3.5)^2 + (0.2-0.2)^2 }
= 2.6

Similarly for (4.8,3.1,1.6,NaN) (5.4,3.4,1.5,0.4)

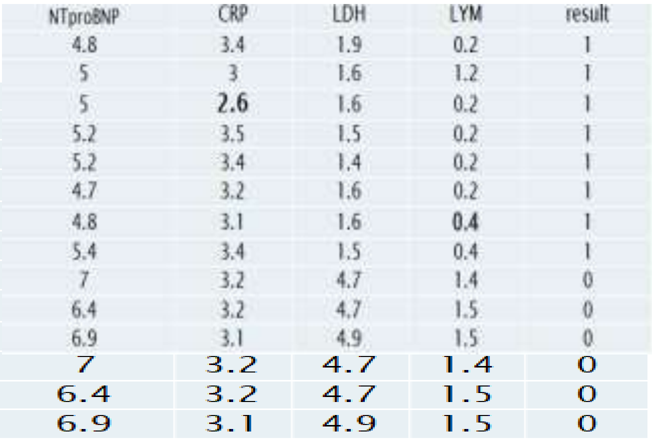
= 4/3 * {(4.8-5.4)^2 + (3.1-3.4)^2 + (1.6-1.5)^2}
=0.4

All the missing value is calculated and is represented in the table 4.2

Table 4.2-Table after finding new value of NaN

3.2 Decision Tree

Root Nodes – It is the node present at the beginning of a decision tree from this node the population starts dividing according to various features.



NTproBNP	CRP	LDH	LYM	result
4.8	3.4	1.9	0.2	1
5	3	1.6	1.2	1
5	2.6	1.6	0.2	1
5.2	3.5	1.5	0.2	1
5.2	3.4	1.4	0.2	1
4.7	3.2	1.6	0.2	1
4.8	3.1	1.6	0.4	1
5.4	3.4	1.5	0.4	1
7	3.2	4.7	1.4	0
6.4	3.2	4.7	1.5	0
6.9	3.1	4.9	1.5	0
7	3.2	4.7	1.4	0
6.4	3.2	4.7	1.5	0
6.9	3.1	4.9	1.5	0

Leaf Nodes – the nodes where further splitting is not possible are called leaf nodes or terminal nodes

- Information Gain
- Gini Index

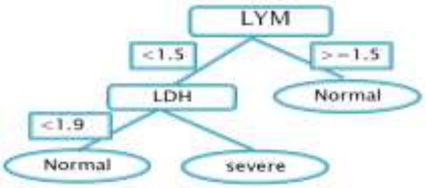
Entropy

Entropy is nothing but the uncertainty in our dataset or measure of disorder.
The formula for Entropy is shown below

$$E(S) = -p_{(+)}\log p_{(+)} - p_{(-)}\log p_{(-)}$$

Here,

- p_+ is the probability of positive class
- p_- is the probability of negative class
- S is the subset of the training example



Gini Index

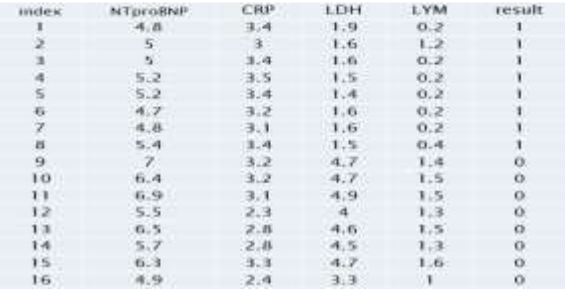
It favours larger partitions and easy to implement. It is

calculated by subtracting the sum of squared probabilities of each class from one.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

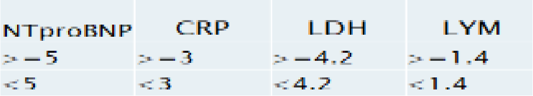
Equation 4 is used to calculate the Gini index of the attributes.

Table 4.3: Sample dataset



index	NTproBNP	CRP	LDH	LYM	result
1	4.8	3.4	1.9	0.2	1
2	5	3	1.6	1.2	1
3	5	3.4	1.6	0.2	1
4	5.2	3.5	1.5	0.2	1
5	5.2	3.4	1.4	0.2	1
6	4.7	3.2	1.6	0.2	1
7	4.8	3.1	1.6	0.2	1
8	5.4	3.4	1.5	0.4	1
9	7	3.2	4.7	1.4	0
10	6.4	3.2	4.7	1.5	0
11	6.9	3.1	4.9	1.5	0
12	5.5	2.3	4	1.3	0
13	6.5	2.8	4.6	1.5	0
14	5.7	2.8	4.5	1.3	0
15	6.3	3.3	4.7	1.6	0
16	4.9	2.4	3.3	1	0

After finding the average point of all the attribute values, we consider the two parts i.e. greater than or equal to that average point and less than that average point which is as shown below.



NTproBNP	CRP	LDH	LYM
>=5	>=3	>=4.2	>=1.4
<5	<3	<4.2	<1.4

Calculating Gini Index using equation 4 for Var A:

Value >= 5: 12

NTproBNP A >= 5 & class = sever: 5/12

NTproBNP A >= 5 & class = normal: 7/12

Gini(5, 7) = 1 – [(5/12)^2 +7/12)^{2}] = 0.4860

Value < 5: 4

Attribute NTproBNP < 5 & class = Sever: 3/4

Attribute NTproBNP < 5 & class = Normal: ¼

Gini(3, 1) = 1 – [(3/4)^{2} + (1/4)^{2}] = 0.375

By adding weight and sum each of the gini indices:

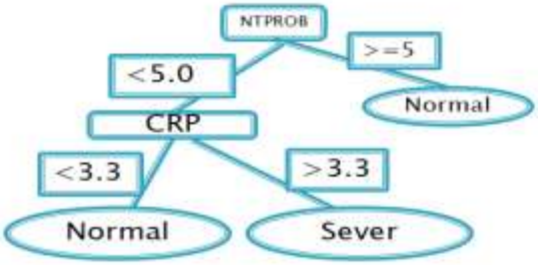
gini(Target, A) = (12/16) * (0.486) + (4/16) * (0.375) = 0.45825

Gini Index of CRP = 0.33

Gini Index of LDH = 0.2

Gini Index of LYM = 0.2

We consider the attribute of least Gini index value as the root node i.e LDH = 0.2 and construct the Decision tree which is as shown in below figure 4.4.



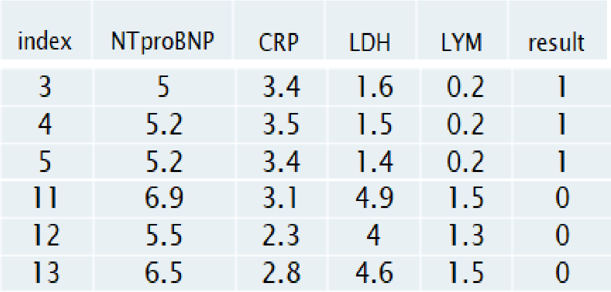
.

Table 4.6: Training Sub-dataset

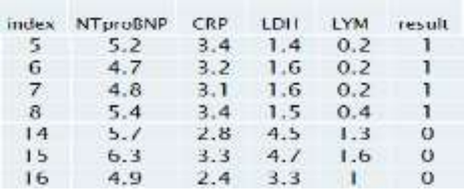
The above table 4.6 is the sub-set of table 4.4, we construct decision tree for the above sub-dataset 2 which is as shown in below figure 4.7

Figure 4.7: Decision tree for sub-dataset 2

Table 4.7: Training Sub data-set 3

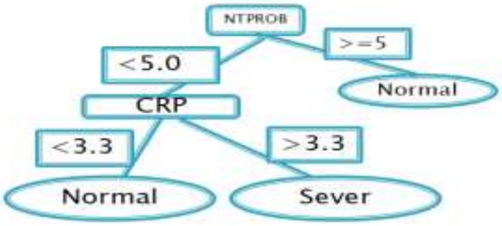


index	NTproBNP	CRP	LDH	LYM	result
3	5	3.4	1.6	0.2	1
4	5.2	3.5	1.5	0.2	1
5	5.2	3.4	1.4	0.2	1
11	6.9	3.1	4.9	1.5	0
12	5.5	2.3	4	1.3	0
13	6.5	2.8	4.6	1.5	0



index	NTproBNP	CRP	LDH	LYM	result
5	5.2	3.4	1.4	0.2	1
6	4.7	3.2	1.6	0.2	1
7	4.8	3.1	1.6	0.2	1
8	5.4	3.4	1.5	0.4	1
14	5.7	2.8	4.5	1.3	0
15	6.3	3.3	4.7	1.6	0
16	4.9	2.4	3.3	1	0

The above table 4.7 is the sub-set of table 4.4, we construct decision tree for the above sub-dataset 1 which is as shown

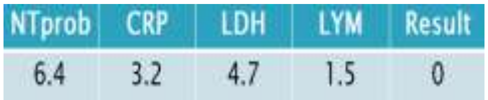


in below figure 4.8

Figure 4.8: Decision tree for sub-dataset 3.

We classify the new instances by selected decision tree.

Classifying New Instance:



NTprob	CRP	LDH	LYM	Result
6.4	3.2	4.7	1.5	0

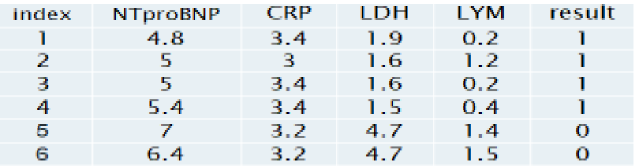
Classifying from all Decision trees = Normal=0

3.4 Gradient Boosted Tree

Shrinkage refers to the fact that the prediction of each tree in the ensemble is shrunk after it is multiplied by the learning rate (eta) which ranges between 0 to 1 which is calculated using the below equation .

y(pred) = y1 + (eta * r1) + (eta * r2) + ... + (eta * rN)

Table 4.8: Sub-dataset for Gradient boosted tree



index	NTproBNP	CRP	LDH	LYM	result
1	4.8	3.4	1.9	0.2	1
2	5	3	1.6	1.2	1
3	5	3.4	1.6	0.2	1
4	5.4	3.4	1.5	0.4	1
5	7	3.2	4.7	1.4	0
6	6.4	3.2	4.7	1.5	0

For the above table 4.8, we start with one leaf node that predicts the initial value for every individual passenger.

Log (Sever/normal)

Log(4/2) = 0.7

0.7 becomes our initial leaf.

If the probability of Sever is greater than 0.5, then we first classify everyone in the training dataset

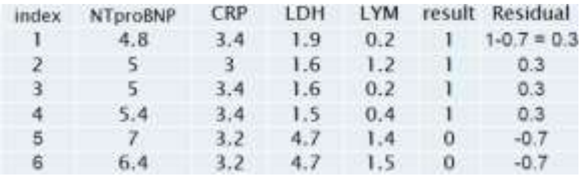
as sever

We need to find the residual which would be:

Residual = Observed – predicted

After calculating the residual values for all the instances in table 4.8 we construct table 4.7 with residual value column which is as shown below

Table 4.9: Sample dataset with residual value attribute



index	NTproBNP	CRP	LDH	LYM	result	Residual
1	4.8	3.4	1.9	0.2	1	1-0.7 = 0.3
2	5	3	1.6	1.2	1	0.3
3	5	3.4	1.6	0.2	1	0.3
4	5.4	3.4	1.5	0.4	1	0.3
5	7	3.2	4.7	1.4	0	-0.7
6	6.4	3.2	4.7	1.5	0	-0.7

We will use this residual to get the next tree. It may seem absurd that we are considering the residual instead of the actual value.

$$\frac{0.3+0.3+0.3+0.3}{[0.7*(1-0.7)]+[0.7*(1-0.7)]+[0.7*(1-0.7)]+[0.7*(1-0.7)]} = 1.428$$

We use a limit of two leaves here to simplify our example, but in reality, Gradient Boost has a range between 8 leaves to 32 leaves.

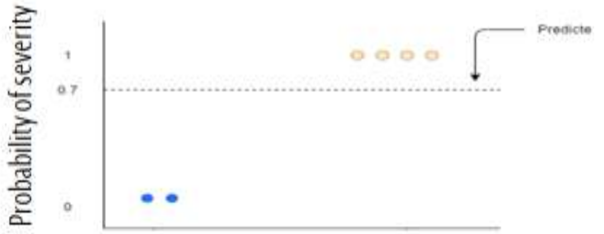


Figure 4.10: Graph of prediction of severity in Gradient boosted tree

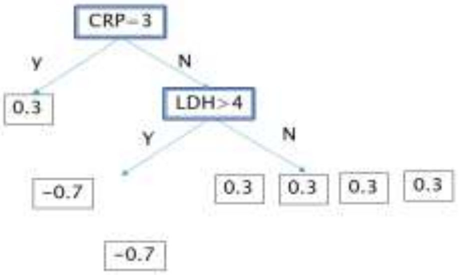


Figure 4.11: Gradient boosted tree

Because of the limit on leaves, one leaf can have multiple values. Predictions are in terms of log(odds) but these leaves are derived from probability which cause disparity. So, we can't just add the single leaf we got earlier and this tree to get new predictions because they're derived from different sources. We have to use some kind of transformation. This is as shown in below figure 4.10.

The numerator in this equation is sum of residuals in that particular leaf.

The denominator is sum of (previous prediction probability for each residual) * (1 - same previous prediction probability).

The first leaf has only one residual value that is 0.3, and since this is the first tree, the previous probability will be the value from the initial leaf

$$\frac{0.3}{[0.7 * (1 - 0.7)]} = 1.43$$

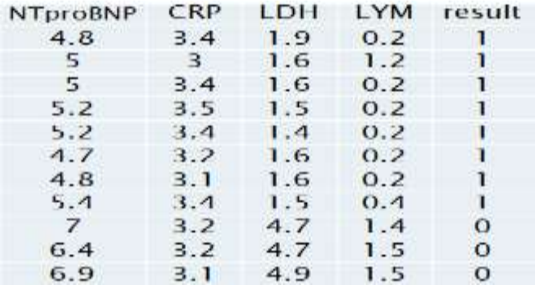
$$\frac{-0.7-0.7}{[0.7 * (1 - 0.7)] + [0.7 * (1 - 0.7)]} = -3.33$$

Figure 4.13: Gradient boosted tree

3.5 Artificial Neural Network

Consider the sample dataset which is as shown below table 4.11

Table 4.10: Sub-dataset for calculating weights in ANN



NTproBNP	CRP	LDH	LYM	result
4.8	3.4	1.9	0.2	1
5	3	1.6	1.2	1
5	3.4	1.6	0.2	1
5.2	3.5	1.5	0.2	1
5.2	3.4	1.4	0.2	1
4.7	3.2	1.6	0.2	1
4.8	3.1	1.6	0.2	1
5.4	3.4	1.5	0.4	1
7	3.2	4.7	1.4	0
6.4	3.2	4.7	1.5	0
6.9	3.1	4.9	1.5	0

For below table 4.10, consider: x1=4.8, x2=3.4, Y=1 for the first Record

W is weighted edge vary from 0.1 to 0.3 which considered as constant

F = 0.1 *4.8 + 0.1 * 3.4+0.1*1.9 = 0.19

F(x) = 1/(1+E(0.19)= 0.3

This output will become the input for the next neuron

Initially W1=0.2 and x1=1

t1=w1*x1=0.2

Applying Sigmoid activation function i.e equation 8:

y1=1/(1+e^(-0.2))=0.73

w2=0.7

t2=0.73*0.7= 0.511

y2=1/(1+e^(-0.511))=0.625

Loss=0.9-0.62=0.27

where **target =0.9**

For updating weights in ANN we use Back propagation algorithm which constantly updates weights minimizing the error which leads to high accuracy.

Back propagation:

We consider sample data present in the table 4.10 for back propagation.

Propagate error from output layer to hidden layer

δ2 = y2 * (1-y2) * error

= 0.625(1-0.625) * 0.27

= 0.072

Δw2 = α * δ2 * y1

= 0.9 * 0.072 * 0.73

= 0.0477

w2 = w2 + Δw2

= 0.7 + 0.034

= 0.734

Propagate error from hidden layer to input layer

δ1 = y1 * (1-y1) * (δ2 * w2)

= 0.73(1-0.73) * (0.072 * 0.7)

= 0.0099

Δw1 = α * δ1 * x1

= 0.8 * 0.0099

= 0.008

w1 = 0.2 + Δw1

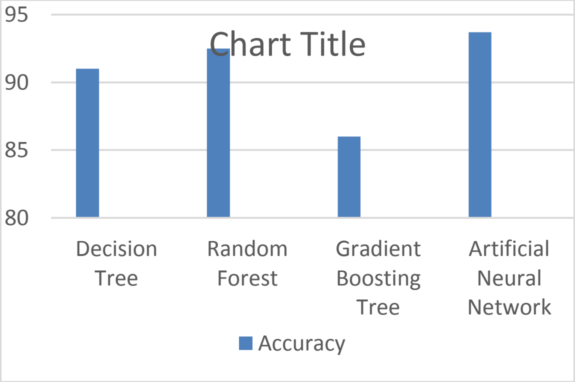
= 0.208

Similarly the process repeats until the error becomes zero, finally resulting in optimal weights which leads to high accuracy.

Artificial Neural network gives accurate value when compared to other algorithms.

IV-RESULT ANALYSIS AND RESULT ANALYSIS

Classification is done using 4 algorithms: Decision tree, Random Forest, Gradient Boosted tree, Artificial Neural Network. The result is predicted using the algorithm which gives highest accuracy and will classify severity level of COVID-19 patient.



REFERENCES

[1] T. Singhal, “A review of coronavirus disease-2019 (covid-19),” Then Indian Journal of Pediatrics, vol. 87, no. 4, pp. 1–6, 2020.

[2] S. Basu, S. Mitra, and N. Saha, “Deep learning for screening covid- 19 using chest x-ray images,” in 2020 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2020, pp. 2521–2527.

[3] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, “Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission,” in KDD ’15, 2015.

[4] A. Ramchandani, C. Fan, and A. Mostafavi, “Deepcovidnet: An interpretable deep learning model for predictive surveillance of covid-19using heterogeneous features and their interactions,” IEEE Access, vol. 8, pp. 159 915–159 930, 2020.

[5] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, “Deepfm: A factorizationmachine based neural network for ctr prediction,” in International Joint Conference on Artificial Intelligence, 08 2017, pp. 1725–1731.

[6] F. Doshi-Velez and B. Kim, “Towards a rigorous science of interpretable machine learning,” arXiv preprint arXiv:1702.08608, 2017.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” arXiv preprint arXiv:1412.6572, 2014.

[8] L. Yan, H.-T. Zhang, J. Goncalves, Y. Xiao, M. Wang, Y. Guo, C. Sun, X. Tang, L. Jing, M. Zhang, X. Huang, Y. Xiao, H. Cao, Y. Chen, T. Ren, F. Wang, Y. Xiao, S. Huang, X. Tan, N. Huang, B. Jiao,C. Cheng, Y. Zhang, A. Luo, L. Mombaerts, J. Jin, Z. Cao, S. Li, H. Xu, and Y. Yuan, “An interpretable mortality prediction model for covid-19 patients,” Nature Machine Intelligence, vol. 2, no. 5, pp. 283–288, May 2020.

[9] E. Matsuyama et al., “A deep learning interpretable model for novel coronavirus disease (covid-19) screening with chest ct images,” Journal of Biomedical Science and Engineering, vol. 13, no. 07, p. 140, 2020.

[10] J. H. Friedman, “Greedy function approximation: A gradient boosting machine.” Ann. Statist., vol. 29, no. 5, pp. 1189–1232, 10 2001.

[11] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, “Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation,” Journal of Computational and Graphical Statistics, vol. 24, 09 2013.

[12] D. W. Apley and J. Zhu, “Visualizing the effects of predictor variables in black box supervised learning models,” Journal of the Royal Statistical Society Series B, vol. 82, no. 4, pp. 1059–1086, September 2020.

[13] A. Fisher, C. Rudin, and F. Dominici, “All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously,” Journal of Machine Learning Research, vol. 20, no. 177, pp. 1–81, 2019.

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, “”why should i trust you?”: Explaining the predictions of any classifier,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144.

[15] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Advances in Neural Information Processing Systems

30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774.

[16] M. T. Ribeiro, S. Singh, and C. Guestrin, “Anchors: High-precision model-agnostic explanations,” in AAAI, 2018.

[17] D. Alvarez-Melis and T. S. Jaakkola, “Towards robust interpretability with self-explaining neural networks,” arXiv preprint arXiv:1806.07538,2018.

[18] R. Luss, P.-Y. Chen, A. Dhurandhar, P. Sattigeri, Y. Zhang, K. Shanmugam, and C.-C. Tu, “Generating contrastive explanations with monotonic attribute functions,” arXiv preprint arXiv:1905.12698, 2019.

[19] M. T. Ribeiro, S. Singh, and C. Guestrin, “Model-agnostic interpretability of machine learning,” arXiv preprint arXiv:1606.05386, 2016. [20] C. Molnar, Interpretable machine learning. Lulu. com, 2020.

[21] L. S. Shapley, “17. a value for n-person games,” Contributions to the Theory of Games (AM-28), Volume II, p. 307–318, 1953.

[22] L. Breiman, J. Friedman, R. Olshen, and C. Stone, “Classification and regression trees. belmont, ca: Wadsworth international group.” Encyclopedia of Ecology, vol. 57, no. 1, pp. 582–588, 2015.

[23] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[24] S. Schaal and C. C. Atkeson, “From isolation to cooperation: An alternative view of a system of experts,” in Advances in Neural Information Processing Systems 8. MIT Press, 1996, pp. 605–611.

[25] A. Maier, C. Syben, T. Lasser, and C. Riess, “A gentle introduction to deep learning in medical image processing,” Zeitschrift f ̈ur Medizinische Physik, vol. 29, no. 2, pp. 86 – 101, 2019, special Issue: Deep Learning in Medical Physics.

[26] G. Montavon, W. Samek, and K.-R. M ̈uller, “Methods for interpreting and understanding deep neural networks,” Digital Signal Processing,vol. 73, p. 1–15, Feb 2018.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas,A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[28] L. Wang, “C-reactive protein levels in the early stage of covid-19,” Medecine et maladies infectieuses, vol. 50, no. 4, pp. 332–334, 2020.

[29] *L. Gao, D. Jiang, X.-s. Wen, X.-c. Cheng, M. Sun, B. He, L.-n. You, P. Lei, X.-w. Tan, S. Qin et al., "Prognostic value of nt-probnp in patientswith severe covid-19," Respiratory research, vol. 21, pp. 1–7, 2020.*

[30] *R. Pranata, I. Huang, A. A. Lukito, and S. B. Raharjo, "Elevated nterminal pro-brain natriuretic peptide is associated with increased mortalityin patients with covid-19: systematic review and meta-analysis," Postgraduate Medical Journal, vol. 96, no. 1137, pp. 387–391, 2020.*

[31] *V. Arya, R. K. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S. C. Hoffman, S. Houde, Q. V. Liao, R. Luss, A. Mojsilovi´c et al., "Oneexplanation does not fit all: A toolkit and taxonomy of ai explainability techniques," arXiv preprint arXiv:1909.03012, 2019.*

[32] *E. K. Bajwa, U. A. Khan, J. L. Januzzi, M. N. Gong, B. T. Thompson, and D. C. Christiani, "Plasma C-reactive protein levels are associated with improved outcome in ARDS," Chest, vol. 136, no. 2, pp. 471–480, Aug 2009.*

[33] *N. Chen, M. Zhou, X. Dong, J. Qu, F. Gong, Y. Han, Y. Qiu, J. Wang, Y. Liu, Y. Wei et al., "Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in wuhan, china: a descriptive study," The lancet, vol. 395, no. 10223, pp. 507–513, 2020.*

[34] *K. Zhao, R. Li, X. Wu, Y. Zhao, T. Wang, Z. Zheng, S. Zeng, X. Ding, and H. Nie, "Clinical features in 52 patients with covid-19 who have increased leukocyte count: a retrospective analysis," European Journal of Clinical Microbiology & Infectious Diseases, vol. 39, no. 12, pp. 2279–2287, 2020.*

[35] *Y. Du, L. Tu, P. Zhu, M. Mu, R. Wang, P. Yang, X. Wang, C. Hu, R. Ping, P. Hu et al., "Clinical features of 85 fatal cases of covid-19 from wuhan. a retrospective observational study," American journal of respiratory and critical care medicine, vol. 201, no. 11, pp. 1372–1379, 2020.*

[36] *G. D. Wool and J. L. Miller, "The impact of covid-19 disease on platelets and coagulation," Pathobiology, vol. 88, no. 1, pp. 14–26, 2021.*